

# ACTIVE MODELLING OF VIRTUAL HUMANS

EAMONN BOYLE BA, BAI, MENG

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY  
(ELECTRONIC ENGINEERING)

SUPERVISED BY DR. DEREK MOLLOY

SCHOOL OF ELECTRONIC ENGINEERING  
DUBLIN CITY UNIVERSITY  
JUNE 2006

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: 

ID No.: 99145103

Date: 23 / June / 2006

## Acknowledgements

I want to thank my supervisor Dr. Derek Molloy for his support and encouragement. I also want to thank Jim Dowling for enabling me to complete this research. Additionally, want to thank all those who I have worked with during the course of this work, in particular, Bartek Uscilowski and Saman Cooray.

I would like to thank my parents for providing the basis for my education, for supporting me in the tough initial years and throughout my studies. I would like to thank my wife Stéphanie for her constant love, encouragement, proof reading skills and for putting up with me.

I also want to thank both the external and internal examiners, Prof. Adrian Hilton and Prof. Paul Whelan for the time they put into the examination of this work and their useful suggestions.

## **Abstract**

### **Active Modelling of Virtual Humans**

**Eamonn Boyle BA, BAI, MENG**

This thesis provides a complete framework that enables the creation of photorealistic 3D human models in real-world environments. The approach allows a non-expert user to use any digital capture device to obtain four images of an individual and create a personalised 3D model, for multimedia applications. To achieve this, it is necessary that the system is automatic and that the reconstruction process is flexible to account for information that is not available or incorrectly captured. In this approach the individual is automatically extracted from the environment using constrained active B-spline templates that are scaled and automatically initialised using only image information. These templates incorporate the energy minimising framework for Active Contour Models, providing a suitable and flexible method to deal with the adjustments in pose an individual can adopt. The final states of the templates describe the individual's shape. The contours in each view are combined to form a 3D B-spline surface that characterises an individual's maximal silhouette equivalent.

The surface provides a mould that contains sufficient information to allow for the active deformation of an underlying generic human model. This modelling approach is performed using a novel technique that evolves active-meshes to 3D for deforming the underlying human model, while adaptively constraining it to preserve its existing structure. The active-mesh approach incorporates internal constraints that maintain the structural relationship of the vertices of the human model, while external forces deform the model congruous to the 3D surface mould. The strength of the internal constraints can be reduced to allow the model to adopt the exact shape of the bounding volume or strengthened to preserve the internal structure, particularly in areas of high detail. This novel implementation provides a uniform framework that can be simply and automatically applied to the entire human model.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Motivation . . . . .	2
1.3	Contributions . . . . .	3
1.4	Organisation . . . . .	5
<b>2</b>	<b>Active Contour Models</b>	<b>6</b>
2.1	Introduction . . . . .	6
2.2	Active Contour Model Definition . . . . .	7
2.2.1	Contour Representation . . . . .	8
2.2.2	Energy Minimisation . . . . .	13
2.3	Problems with the Snakes . . . . .	17
2.3.1	Initialisation of the Snakes . . . . .	17
2.3.2	Regularisation of the Snakes . . . . .	18
2.3.3	The Balloon Model . . . . .	19
2.3.4	Fourier Snakes . . . . .	19
2.3.5	Dual Active Contours . . . . .	20
2.3.6	Topologically Adaptable Snakes . . . . .	21
2.4	G-Snakes - Deformable contours: Modelling and Extraction . . . . .	22
2.5	Reformulation of the Active Contour Model . . . . .	24
2.5.1	B-spline Snakes . . . . .	24
2.5.2	NURBS Snakes . . . . .	26
2.6	Geometric Active Contour Model . . . . .	28
2.6.1	Curve Evolution . . . . .	29
2.6.2	Relationships between Parametric and Geometric Models . . . . .	30
2.7	Templates . . . . .	31
2.7.1	Deformable Templates . . . . .	31
2.7.2	Active Shape Models . . . . .	34
2.7.3	Tracking using Active Contour Models . . . . .	35
2.8	Active-meshes . . . . .	35

2.9	Discussion . . . . .	37
<b>3</b>	<b>Building Virtual Humans</b>	<b>40</b>
3.1	Introduction . . . . .	40
3.2	General Approaches to 3D Reconstruction . . . . .	41
3.2.1	The Eight-point Algorithm . . . . .	42
3.2.2	Silhouette Based Reconstruction . . . . .	43
3.2.3	Volumetric (Scene) Reconstruction . . . . .	45
3.2.4	Scanning Techniques . . . . .	49
3.3	<b>3D</b> Reconstruction (Vision) . . . . .	50
3.4	<b>3D</b> Object Pose Estimation . . . . .	53
3.5	Creation of Virtual Humans . . . . .	55
3.5.1	Data Acquisition . . . . .	56
3.5.2	Building a Virtual Human . . . . .	60
3.5.3	3D Modelling Environments . . . . .	69
3.6	Discussion . . . . .	71
3.6.1	Discussion on Human Modelling . . . . .	72
<b>4</b>	<b>Design Approaches</b>	<b>74</b>
4.1	Introduction . . . . .	74
4.2	<i>Approach 1: Towards the Creation and Animation of Virtual Humans</i> . . . . .	75
4.2.1	Silhouette Creation . . . . .	75
4.2.2	Image Capture . . . . .	77
4.2.3	Texturing . . . . .	79
4.2.4	Issues Highlighted in the Approach . . . . .	81
4.3	<i>Approach 2: Creating Active B-Spline Templates</i> . . . . .	83
4.3.1	Image Capture and Definition of the Individual Pose . . . . .	83
4.3.2	Template Generation using Active Contours . . . . .	86
4.3.3	Template Initialisation . . . . .	88
4.3.4	Constraints to Control the Evolution of the Templates . . . . .	94
4.3.5	Minimisation of the Templates Energy . . . . .	96
4.3.6	Issues Highlighted in this Approach . . . . .	97
4.4	<i>Approach 3: Using Facial Feature Extraction to Enhance 3D Human Models</i> . . . . .	99
4.4.1	Image Capture and Template Fitting . . . . .	100
4.4.2	Face Localisation . . . . .	100
4.4.3	Texturing and Personalising the Model . . . . .	101
4.4.4	Outcome of Texturing the Model Using the Facial Components . . . . .	102
4.4.5	Issues Highlighted in this Approach . . . . .	103
4.5	<i>Approach 4: Silhouette Based Models of an Individual</i> . . . . .	105
4.5.1	Alignment of views in 3D . . . . .	105
4.5.2	Combining of the Body Parts . . . . .	109

4.5.3	Texturing the Final Volumetric Model . . . . .	110
4.5.4	Issues Highlighted in this Approach . . . . .	110
4.6	<i>Approach 5: The extension of Active-Meshes to 3D</i> . . . . .	113
4.6.1	Specification of the Internal and External Constraints . . . . .	113
4.6.2	Termination Process . . . . .	118
4.6.3	Issues Highlighted in this Approach . . . . .	118
4.7	Discussion . . . . .	119
<b>5</b>	<b>Implementation, Testing and Results</b>	<b>121</b>
5.1	Introduction . . . . .	121
5.2	Image Capture . . . . .	122
5.3	Extraction of an Individual from a Real Environment . . . . .	127
5.3.1	Background Subtraction . . . . .	130
5.3.2	Application of Edge Detectors . . . . .	130
5.3.3	Testing of the Active Contour Implementation . . . . .	131
5.3.4	Template Generation and the Testing of the Constraints . . . . .	139
5.3.5	Template Initialisation and Minimisation . . . . .	144
5.3.6	Assessment of the Template Fitting . . . . .	151
5.4	Texturing of the Underlying Model . . . . .	152
5.4.1	2D to 2D Texture Mapping . . . . .	152
5.4.2	2D to 3D Texturing of the Underlying Model . . . . .	156
5.4.3	Enhancement of the model using facial features . . . . .	158
5.5	Generation of the Bounding Volume . . . . .	164
5.6	Testing of the 3D Active-Mesh Implementation . . . . .	169
5.6.1	Testing the Internal Constraints in 3D . . . . .	169
5.6.2	Testing with Primitive Shapes . . . . .	170
5.6.3	Application of Active-Meshes to Human Modelling . . . . .	180
5.7	Discussion on Results . . . . .	196
<b>6</b>	<b>Conclusions</b>	<b>202</b>
6.1	A Brief Review . . . . .	203
6.2	Major Contributions . . . . .	205
6.2.1	Associated contributions . . . . .	205
6.2.2	Limitations Identified In This Research . . . . .	206
6.3	Publications associated with this research . . . . .	207
6.4	Future Directions for Research . . . . .	208
6.4.1	Incorporation of NURBS into the Active-Mesh Formulation . . . . .	208
6.4.2	Automatic Selection of Templates . . . . .	209
6.4.3	Template Initialisation Possibilities . . . . .	209
6.4.4	Skin Animation using B-spline Control Points . . . . .	209

**A Splines and the Representing of Curves and Surfaces** . . . . . 220

A.1 Introduction . . . . . 220

A.2 Parametric Curves . . . . . 221

    A.2.1 Parametric Cubic Curves . . . . . 221

A.3 B-Splines Curves . . . . . 223

    A.3.1 B-Spline Surfaces . . . . . 229

    A.3.2 NURBS Surfaces . . . . . 229

A.4 Lines, Planes and Intersections in 3-Space . . . . . 230



# List of Figures

2.1	Examples of Initial Contours . . . . .	8
2.2	Gradient and normal to the curve. $\nabla(I(s))$ denotes the gradient at $s$ , $dr$ is the normal along the contour, $C$ , and $n$ is the inward unit normal vector. . . . .	11
2.3	(a) the volcano pushing the contour away and in (b) the spring force attached to a control point on the contour. . . . .	13
2.4	This figure depicts the correspondence between a decision set and image pixels for dynamic programming with number of states, $m = 9$ . The curve indicates the optimum position of each control point based on a particular criteria set. Each of the control points can move to one of the nine squares in the grid at a particular iteration. The position that is adopted minimises the energy configuration for a particular iteration. This is indicated by the curved arrows. . . . .	16
2.5	(a) neighbourhood searched when pattern = 1 in fast greedy algorithm, (b) neighbourhood searched when pattern = -1 in fast greedy algorithm. This approach reduces the number of operations at each iteration, assuming that the area around the minimum will also contain small values. . . . .	16
2.6	(a) simplicial approximation of the contour model constructed by subdividing the image using a cubic grid, (b) the cell classification. . . . .	22
2.7	The movement of a point on the curve $v(s)$ towards the desired control point as the associated weight is increased. . . . .	28
2.8	Illustration of the Level sets approach. (a) the original curve and (b) equivalent level set surface. . . . .	29
2.9	Example of the application of active-meshes. (a) shows the generated mesh that is used to track the position of the ambulance in an image sequence and (b) shows the motion vectors that are generated by the ambulance. . . . .	36
3.1	Volume intersection approach to reconstruction. (a) shows a simple shape reconstructed from two silhouettes. (b) shows a situation when the true shape of the object can not be reconstructed. . . . .	43
3.2	Illustration of an object's visual hull from 4 views <sup>4</sup> . . . . .	44
3.3	Illustration of how simple decision are made to decide if a voxel is inside the visual hull. (a) shows a sphere within the field of view of three cameras. The intersection of lines indicating the field of view of each camera forms the visual hull of the object. (b) shows a coarse reconstruction of the object the shaded squares indicate the voxels inside the visual hull of the object. . . . .	46

3.4	(a) the voxel projects in two views to background. (b) the voxel projects to the same color in all three views . . . . .	47
3.5	This figure illustrates how the pixel data in each image can be used to determine if the reconstruction is consistent with the captured data. . . . .	48
3.6	(a) A photograph of Berkeley's clock tower, with edges marked in green. (b) The model recovered by the method of Debevec et al. (1996). Although only the left pinnacle was marked, the remaining three, including the one not visible in the captured image were recovered from symmetrical constraints in the model. (c) shows that the accuracy of the model can be verified by projecting it into the original photograph. In (d) a synthetic view of the clock tower generated using the view-dependent texture-mapping method. . . . .	51
3.7	(a) contains four images of the set used in (Pollefeys et al. 2000) for the reconstruction of the Arenberg castle with the reconstruction shown in part (b). . . . .	52
3.8	Example of pose of an object detected using the process in (Zerroug & Nevatia 1995). Each detected object is described as a graph where nodes are parts and arcs labeled joint relationships between parts. . . . .	54
3.9	The Gypsy Motion capture System is shown. This motion capture system is an electro mechanical system that consist of an exoskeleton made of lightweight aluminium rods that follow the motion of the performer's bones. . . . .	57
3.10	Real-time Motion capture System that captures the individuals movements and animates a model to produce the same motions. . . . .	57
3.11	Example of a model created with the cyberware system. . . . .	59
3.12	The Cybreware whole body scanner system. . . . .	60
3.13	Example of the camera set up in the approach of Kakadiaris and Metaxas. . . . .	61
3.14	Example of the types of movements an individual must undertake in the approach of Kakadiaris and Metaxas. . . . .	62
3.15	Example of the reconstruction of the leg and the general approach to combining the 2D shape information. . . . .	62
3.16	Results of the Cheung et al. approach applied to synthetic data (Cheung et al. 2003).(a) one of the input images (b) unaligned color surface points and (c) shows the aligned colour surface points and (d) refined visual hull. . . . .	63
3.17	Results of the Cheung et al. approach applied to real human body (Cheung et al. 2003).(a) one of the input images (b) unaligned colour surface points and (c) shows the aligned colour surface points and (d) refined visual hull. . . . .	64
3.18	Models created for video conferencing systems. (a) shows an example of the flexible underlying mesh and (b) the final textured mesh. . . . .	65
3.19	The Hilton et al. system for the reconstruction of an individual. . . . .	67
3.20	The creation of human models in real environments (Lee, Goto & Magnenat-Thalmann 2000). (a)showa original images with the silhouettes super-imposed marked in yellow, (b) The final H-anim model combined with the seperately reconstruced face and (c) shows the untextured updated mmodel. . . . .	68
3.21	Segmentation of the scanned body into individual body parts. . . . .	69
3.22	The Sculpter character creation tool (Kalra & Magnenat-Thalmann 1998). Example of how the facial image is combined with an underlying model to sculpt the model to take on the appearance of the individual. . . . .	70
4.1	Flowchart with the main elements in the system for approach 1. . . . .	76

4.2	The default H-Anim model used for the creation of virtual humans. (a) shows a front view of the model, (b) and (c) show front and side views generated using the projection matrix in Equation 4.1 . . . . .	77
4.3	Examples of images captured for the creation of the virtual humans. . . . .	78
4.4	Illustration of the effects of applying a Gaussian filter to the images. (a) shows the original image, (b) shows 402 regions in the original image, (c) shows the Gaussian smoothed image and (d) shows the 37 regions in the smoothed image. . . . .	79
4.5	2D to 2D texture mapping of the body (a) shows the image data of the arm is mapped to the arm silhouette of the arm. In (b) the combined data is shown for the front and back views. . . . .	80
4.6	This figure illustrates the role of the normal vectors have in determining the image that is used to texture a face of the model. . . . .	81
4.7	Example of the textured model using the 2D texture map in Figure 4.5. (a) contains three views of the static model textured with the captured data for the sequence shown in Figure 4.3, (b) shows three views of the model in (a) as it is animated using a walking sequence. . . . .	82
4.8	Flowchart with the main components of Approach 2. . . . .	83
4.9	A set of captured images for an individual. . . . .	85
4.10	Examples of the captured images used for the generation of the template. . . . .	87
4.11	The key points that are identified by automatically examining the B-spline contour silhouette. (a) shows the key points identified on front and back views and (b) shows the key points found on the side views. . . . .	88
4.12	The mean template generated. (a) shows the mean template with the control under the arms and between the legs. (b) shows an approximation of the skeleton that is automatically fitted to the skeleton data. . . . .	89
4.13	The initial positions that the individual has selected for the features. . . . .	90
4.14	Effects of applying the Canny edge detector to images capture in a cluttered environment Canny (1986) . (a) and (d) show the original images, (b) and (e) illustrate the edges generated when the Canny edge detector has $\sigma = 1$ and the lower threshold, $T1$ set to 100 and the upper threshold set to 255, (c) and (f) illustrate the edges generated when the Canny edge detector has $\sigma = 2$ and the lower threshold, $T1$ set to 100 and the upper threshold set to 255. The images illustrate that when significant number of edges are extracted that it is difficult to initially position the template. . . . .	91
4.15	(a) shows the front and back subtraction that produces the difference map and (b) shows the difference map produced for the subtraction of the side views. . . . .	92
4.16	(a) and (d) show manual fitting of the template to the front view, (b) and (c) show automatic fitting of the template to the same images and (e) shows the automatic fitting of the template a side view. . . . .	93
4.17	The incorrect convergence of the template control points. In (a), the initial position of the template is shown. In (b), the position of the contour after 20 iterations is shown, it can be seen that under the right arm and between the legs the contour is converging to the same edge. In (c), the situation is shown after 60 iterations. . . . .	95
4.18	Starting with the contour after 20 iterations shown in Figure 4.17 and for the purpose of demonstrating the effects of the constraints, the constraints are manually introduced between the legs and under the arms by selecting each node and changing the direction that the control points can move. (b) shows the effect of introducing the constraints after a further 20 iterations. . . . .	96
4.19	Facial features localisation algorithm. . . . .	99
4.20	Examples of the capture images used to for testing this approach. . . . .	100

4.21	Results of subtraction of the front and back views and the subtraction of the left and right views. . . . .	100
4.22	Facial features locations in (a) front view and (b) side view head with the key features marked. (c) shows the front view of the model with the key features marked and (d) shows the key features identified on the side view of the head. image. . . . .	101
4.23	The locations of the facial features and the distances between them used for deforming the underlying model. . . . .	102
4.24	The created human model with (a) misplaced facial features and (b) aligned facial features. . . . .	103
4.25	The second human model textured with the facial image in (a). (b) shows the misplaced facial features and (c) (d) show two views of the model with aligned facial features. . . . .	104
4.26	Illustration of the main components of the system in approach 4. . . . .	106
4.27	parts (a) and (b) show examples of the 2D front and left silhouettes and part (c) shows the two silhouettes aligned in 3D space. . . . .	107
4.28	A schematic of how the leg is rotated about the $y$ -axis to construct the legs. . . . .	108
4.29	(a) shows the simple reconstruction of the arms from two views. It is clearly seen that shape of the arm does not approximate the shape of the individual's arm(In this case only two control points are interpolated between points on the control points extracted in the front view). In (b) the use of the depth from the upper body is shown from two views. . . . .	109
4.30	Two views of the 3D B-spline surface created from Figure 4.9 using volume intersection. . . . .	110
4.31	Two views of a textured model in Figure 4.30. . . . .	111
4.32	The main components of the Active-Mesh modelling tool. . . . .	113
4.33	Internal and external energies. The internal forces act along the mesh lines and are indicated by the vector $\overline{F}_{Line}$ and the external forces are generated using the normal vector, $n$ , to the mesh element. . . . .	114
4.34	Visual marking of the rigidity on the underlying model. The red parts have strong rigidity and tightly bound vertices. The lighter red parts indicate weaker internal forces. . . . .	115
4.35	Combining the Forces at a single node. This adapted to 3D and is based on a diagram in (Molloy & Whelan 2000). . . . .	116
5.1	(a) shows the essence of the capture process and how the four views of the individual captured, although in reality the capture system consists of a single camera, as in (b) where the images are captured against the same background. . . . .	124
5.2	The images in this Figure constitute a core set that are used within the section. . . . .	126
5.3	Example of individuals against backgrounds with different levels of clutter. In each case the number of regions also accounts for regions in the individual clothes. In (a) the number of regions is 196, in (b) the number of regions is 348, in (c) the number of regions is 271, in (d) the number of regions is 418 and in (e) the number of regions is 280. . . . .	128
5.4	(a) shows an image with three measures of clutter made at three different height levels. In (b), (c) and (d) the level in variation of the colour components at a particular height are shown. In each case the three colours, red, green and blue represent the respective components and the average value is also indicated using the black line. . . . .	129
5.5	(a) shows an image with three measures of clutter made at three different height levels. In (b), (c) and (d) the level in variation of the colour components at a particular height are shown. In each case the three colours, red, green and blue represent the respective components and the average value is also indicated using the black line. . . . .	130

5.6	The results for background subtraction. (a) shows the background image (b) shows the individual against the same background (c) shows the result of a direct subtraction (inverted for clarity) and (d) shows the same image as (c) with the brightness and contrast manually adjusted to try and improve the separation of the individual from the background. . . . .	131
5.7	The results of applying the Canny edge detector to the input images. In each case the parameters for the Canny are $\sigma = 1$ , $T_1 = 100$ and $T_2 = 255$ . . . . .	132
5.8	A schematic showing the information contained in the snake and the information associated with a control point. . . . .	133
5.9	Demonstration on the effects of the internal energy. (a) to (d) show that if no constraints are included then the internal energy has the effect of collapsing the contour to a point (e) to (g) Show the effects of constraining the internal energy by encouraging an even spacing between the control points. . . . .	134
5.10	(a) shows the initial position of the contour, (b) and (c) show the progression of the contour and the effect of the external energy on the minimisation process. This minimisation took 41 iterations. . . . .	134
5.11	(a) shows the initial position of the active contour inside the circle and (b), (c) and (d) show that the attraction to high intensity edges counteract against the internal energy constraints and it expands to find the boundary. This minimisation took 51 iterations. . . . .	135
5.12	(a) shows the directions that are searched for high intensity edges (b), (c) and (d) show the progression of the active contour over weak edges to the strongest edges. The outer and middle rectangle has an edge intensity of 155 and 200 respectively. . . . .	135
5.13	(a) shows the active contour at rest on the boundary of the square, (b) shows four points dragged from the equilibrium position, (c) shows the position after 20 iterations and (d) shows the position after 32 iterations. . . . .	136
5.14	(a) shows the initial position of the contour which is distant from the edges, (b) shows the effects of the internal energy as it minimises a constant rate until the search space contains high intensity edges, in (c) the external forces start to dictate the minimisation process. (d) shows the final position of the snake and the approximately even spacing of the control points. . . . .	136
5.15	A demonstration of how additional control points are inserted. (a) shows the initial position of the contour, (b) shows the final position of the contour, (c) shows the four inserted control points, marked yellow, (d) shows the results of the minimisation considering the new points. Note: in (d) the control point in the bottom right corner is located at the corner of the square but the B-spline does not interpolate this point. . . . .	137
5.16	A demonstration of how particular control points are removed from the contour formation. (a) shows the initial position of the contour, (b) shows an intermediate position of the contour, (c) shows the final position of the contour and (d) shows the results of removing control points along the sides of the square. . . . .	138
5.17	In (a) and (c) the default position of the arms is shown. Then in (b) and (d) the position of the arms after they have rotated about the y-axis is shown then scaled rotated back through the angle provided from the bounding box or from user initialisation . . . . .	142
5.18	(a) shows the position of the springs between the legs of an individual. In this situation the and also illustrates that the forces associated will have both a horizontal and vertical element that is not suitable for constraining the position of the control points. (b) illustrates the directional effects of the controls introduced . . . . .	143

5.19	The results of the subtraction process applied to the front and back images in Figure 5.2. (a) corresponds to the subtraction of Figure 5.2 (a) and (c). (b) corresponds to the subtraction of Figure 5.2 (e) and (g). (c) corresponds to the subtraction of Figure 5.2 (i) and (k). (d) corresponds to the subtraction of Figure 5.2 (i) and (j). (e) corresponds to the subtraction of Figure 5.2 (m) and (o). (f) corresponds to the subtraction of Figure 5.2 (q) and (s). . . . .	145
5.20	(a) to (e) show the results of applying the Canny edge detector to the difference map generated using the front and back images in Figure 5.2. (f) to (i) show the results for the side images. . . . .	146
5.21	The bounding box and the correctly initialised templates. (a) to (e) show the bounding boxes generated for the front views in Figure 5.20. (f) to (i) show the results for the side images in 5.20. . . . .	147
5.22	The initial position of the contour is shown in parts (a) to (d). The intermediate position of the contour is shown in images (e) to (h) and the final position of the contour is shown in parts (i) to (l). . . . .	149
5.23	(a) shows the initial position of the contour on the edge map, (b) shows the position of the contour after 10 iterations, (c) shows the position of the contour after 20 iterations and (d) shows the position of the contour after 40 iterations. . . . .	150
5.24	Graphs illustrating the mean square error (MSE) calculated in the template fitting procedure. 151	
5.25	Example of simple 2D to 2D texturing. (a) shows the region to be textured with the texture map in (b). (c) shows the effects of texturing if the image is only scaled horizontally and (d) shows the correct fitting of the texture map. . . . .	153
5.26	This Figure illustrates how the individual's silhouettes are separated into different parts for texturing. Parts (a) and (d) show the silhouette of the front of the individual split into six parts and seven parts. Parts (b) and (e) show the original image split into six and seven parts. Parts (c) and (f) show the side view of the individual split into four parts. . . . .	154
5.27	(a) and (c) contain the front and side silhouettes of the individual shown in Figure 5.2 (e) to (h), (b) and (d) show the equivalent silhouettes extracted using the template. . . . .	155
5.28	(a) contains the silhouette of model produced by projecting the model through the back camera centre, parts (b) and (c) show the equivalent silhouette and textured image. . . . .	155
5.29	This Figure illustrates how the model's silhouette is separated into different parts for texturing. In part (b) the model's silhouette is split into seven parts and in part (c) the model's silhouette is split into six parts. . . . .	156
5.30	(a) and (b) show mapping of the captured data in Figure 5.26 to the silhouettes in Figure 5.29. (c) contains the result of the 2D to 2D texturing process for Figures 5.2 (a) to (d). . .	157
5.31	Example of how the head is textured. The parts of the head that are textured red are textured using the front view, the parts of the head that are textured green are textured using the back view and the blue and magenta represent the side views. . . . .	157
5.32	the results of the 2D to 2D texturing process for the images in Figure 5.2, part (a) shows the results of the texturing with parts (e) to (h), part (b) shows the results of the texturing with parts (i) to (l), part (c) shows the results of the texturing with parts (m) to (p) and part (d) shows the results of the texturing with parts (q) to (t). . . . .	159
5.33	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	160

5.34	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	161
5.35	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	161
5.36	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	162
5.37	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	162
5.38	Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint. . . . .	163
5.39	Bounding volumes for the images (a) to (d) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.	164
5.40	Bounding volumes for the images (e) to (h) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.	165
5.41	Bounding volumes for the images (i) to (l) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.	166
5.42	Bounding volumes for the images (m) to (p) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.	167
5.43	Textured bounding volumes for the images (q) to (t) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model. . . . .	168
5.44	Application of internal constraints at a single point on the spheres surface. (a) shows the point of intersection on the surface and the effect this has on the current length of the mesh lines connected to the point. (b) shows the position after three iterations. (c) and (d) show the position after 20 iterations and 40 iterations respectively. . . . .	170
5.45	The application of internal constraints on a sphere, (a) shows the initial sphere (b), (c) and (d) show the effects of strong internal constraints on the sphere, and (f), (g) and (h) show the effects of weaker internal forces on the structure of the sphere. . . . .	171
5.46	The results for the first active-mesh trial involving a sphere placed inside and at the centre of a cube. (a) shows the initial shapes, (b) shows the evolution after 2 iterations, (c) shows the iterations after 10 iterations and (d) shows the final shape of the sphere after 46 iterations.	173
5.47	The results for the second active-mesh trial involving a sphere placed inside a cube and offset from the origin in the x and y direction. (a) shows the initial shapes, (b) shows the evolution after 2 iterations, (c) shows the iterations after 10 iterations and (d) shows the final shape of the sphere after 58 iterations. . . . .	174

5.48	The results for the third active-mesh trial involving a sphere placed partially inside and outside the cube. (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations and (d) shows the final shape of the sphere after 64 iterations. . . . .	175
5.49	The results obtained with an alternative method for calculating the external energy. (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations, (d) shows the final shape of the sphere after 51 iterations and (e) shows the uneven distribution of the vertices of the deformed sphere. . . . .	176
5.50	The modelling of a sphere with half the vertices having strong rigidity and half having high elasticity (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations, (d) shows the iterations after 30 iterations, (e) shows the final shape after 37 iterations and shows (f) shows the structure of the moulded sphere. . .	177
5.51	The modelling of a sphere starting with a cube (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 30 iterations, (d) shows the final shape. . . . .	178
5.52	The modelling of a sphere to approximate the head of the underlying model. The initial position of the sphere and the head are shown in (a) and the moulding at 5, 10 and 20 iterations is shown in (b), (c) and (d). (e) and (f) show two additional views of the moulded sphere after 20 iterations. . . . .	179
5.53	The class structure for the super-mesh class and the associated mesh class. The vectors in the super-mesh class contain lists of data that is important for each mesh but is not directly incorporated in the mesh class structure. . . . .	181
5.54	Three views of the aligned underlying model and the bounding volume generated in Figure 5.39. . . . .	182
5.55	(a) and (b) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.40. (c) and (d) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.41 (e) and (f) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.42 (g) and (h) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.43. . . . .	183
5.56	(a) shows an example of how the angle of rotation is calculated on the model, (b) shows the equivalent angles on the captured individual, (c) shows the original position of the arms before their position has been corrected and (d) shows the corrected arm position. . . . .	184
5.57	(a) shows the effects of modelling the human without constraints, (b) shows the constraints introduced as planes and (c) shows the effects of the planes when only the external energy is used to deform the underlying model. . . . .	184
5.58	(a) shows the default initialisation of the constraints on the human model, (b) shows an alternative set of internal constraints. In (a) the areas highlighted in red have strong rigidity and the areas in white are free to deform. In (b) the areas highlighted in blue have strong rigidity, in green have strong rigidity and strong elasticity and the areas highlighted in red have weak rigidity and strong elasticity. . . . .	185
5.59	The application of active-meshes to moulding the head into a cube and a sphere while maintaining the internal structure of the face. (a) shows the initial head. The blue region is highly constrained and maintains its structure, (b) and (c) show two views of the final shape that approximates the square. In (d) the shape of the head deformed partially to the sphere and in (e) and (f) show two views of head deformed to the sphere. . . . .	186



5.60	Illustration of how the shape of the individual body parts are well preserved in the active-mesh implementation. (a) shows the left upper leg the lower leg and the foot and (b) shows the upper arm, the forearm and the hand. . . . .	189
5.61	An example of the deformation of the model to approximate the bounding volume in Figure 5.39. . . . .	190
5.62	An example of the deformation of the model to approximate the bounding volume in Figure 5.40. . . . .	191
5.63	An example of the deformation of the model to approximate the bounding volume in Figure 5.41. . . . .	192
5.64	An example of the deformation of the model to approximate the bounding volume in Figure 5.42. . . . .	193
5.65	An example of the deformation of the model to approximate the bounding volume in Figure 5.43. . . . .	194
5.66	The modelling of the underlying model. (a) shows the front view of the bounding volume created for the model, (b) shows a side view of the models bounding volume, (c) shows the initial model aligned with the bounding volume and (d) shows the moulded underlying model after 20 iterations. From the 2D silhouettes of the model, the height of the model is 615 pixels, the width from left hand to right hand is 220 pixels and the depth of the model 80 pixels. . . . .	195
5.67	Illustration of how the MSE error decreases over 20 iterations. . . . .	196
5.68	Animation of the models. (a) to (c) show the animation of the models created in approaches 2 and 3. (d) to (g) show the animation the active mesh models. . . . .	198
A.1	point on a parametric cubic curve. . . . .	222
A.2	Piecewise polynomial curve with three segments. . . . .	224
A.3	A uniform cubic B-spline, basis functions $N_i(t)$ . . . . .	225
A.4	A uniform cubic B-spline shown with control points and control polygon. . . . .	226
A.5	Nonperiodic B-splines with varying degrees: from linear to cubic. . . . .	228
A.6	Nonuniform cubic B-spline with knot vector $[0, 1, 1, 2, 3, 3, 4, 5, 5, 6]$ . . . . .	228
A.7	Illustration of the equation of the line parallel to vector $v$ . . . . .	231
A.8	Illustration of the point normal form of the equation of a plane. . . . .	232
A.9	(a) the vectors used to calculate the cross product for a triangle face and (b) the vector normal that results from the cross product of the vectors $v_1$ and $v_2$ . . . . .	233
A.10	Illustration of the calculation of the normal per vertex . . . . .	234

# Introduction

## 1.1 Introduction

Virtual worlds have been in existence for several years, providing users with new experiences through the development of innovative computer games, the reconstruction of ancient cities and virtual reality applications, to name but a few. Although full immersion is still not feasible, a user's experience can be greatly enhanced using personalised human models. To satisfy the demand for models, the creation of realistic human models has infiltrated the world of computer games and virtual worlds. Moreover, increasingly applications are facilitating the creation of personalised characters, avatars, etc. to renew interest in existing applications and to draw attention to new ones. This has resulted in continuous development of techniques for the creation of virtual, realistic, and anatomically correct human models for use in both real-time and offline applications. In film studios where there exist bigger budgets, large number of cameras can be used to capture images that are combined in a user-assisted reconstruction process to create high-quality realistic human models. Presently, not every user has access to this type of technology or the technical expertise necessary to create such models, but still requires the same level of realism. The most common methods that are currently used in games involve the capture of images and texturing a default model with the face of the captured individual, for example in FIFA05<sup>1</sup> and Quake<sup>2</sup>.

There are numerous different approaches to the creation and animation of virtual humans, depending primarily on the final application but also on the input data, the quality of the model required and the destination device (which in many cases is not limited to a single device). The input data consists of either images captured from a single or multi-camera setup (Hilton et al. 1999), data from a whole body scanner (Ju & Siebert 2001) or models for sculpting in modelling environments (Kalra et al. 1998).

In addition, to provide standard representation of human models, the Motion Picture Experts Group (MPEG) and the Web3D organisations have collaborated to produce the humanoid animation standard which forms part of both the MPEG-4 standard (MPEG4 1998) and the VRML (Virtual Reality Modelling Language) 2.0 standard (VRML 1997).

The approach described in this thesis provides a flexible automated low constrained image

---

<sup>1</sup>EA Games: [www.ea.com](http://www.ea.com)

<sup>2</sup>ID Games: [www.idsoftware.com](http://www.idsoftware.com)

based method that enables a home-user with the tools to create realistic human models with non-specialised equipment that can be seamlessly immersed into existing worlds and animated using existing animation streams. The approach builds on previous research in the area of active contours models (ACMs), also known as snakes, introduced by Kass et al. (1987) as a method to solve a variety image and machine vision tasks. ACMs are a general technique for matching a deformable model to an image using energy minimisation.

In particular, this approach presents a constrained ACM in the form of a deformable template that is automatically initialised close to the individual in the captured images and that deforms to minimise the ACM's energy while adhering to the predefined constraints. The extracted 2D contours describe accurately the individual's shape. The contour extracted in each view is then combined to form a 3D bounding volume using a silhouette based reconstruction technique. The bounding volume represents the maximal silhouette equivalent of the individual and contains sufficient information to allow for active deformation of an underlying generic human model. Our modelling approach is performed using a novel technique that extends active-meshes to 3D, to deform the underlying human model, while adaptively constraining it to preserve its existing structure (Molloy & Whelan 2000). The active-mesh approach incorporates internal constraints that maintain the structural relationship of the vertices of the human model, while external forces deform the model congruous to the bounding volume mould. The strength of the internal constraints can be reduced to allow the model to adopt the exact shape of the bounding volume or strengthened to preserve the internal structure, particularly around the face and in areas of high detail. This novel implementation provides a uniform framework that can be simply and automatically applied to the entire human model.

## 1.2 Motivation

The main motivation behind this research is to provide the home-user<sup>3</sup> with any digital camera (including web-cams and camera enhanced mobile phones) the ability to create their own personalised human models that can be used in a variety of interactive virtual environments. In providing such a facility, it will enhance the individual's experience in the virtual world and encourage the creation of virtual communities where individuals can interact with and recognise one another.

In developing this system, it is necessary to provide a set of tools that a home-user can simply use to create a realistic model that can be incorporated easily into existing and future virtual environments. The system should be automated or provide a user interface that requires a low-level of user interaction to produce the model. The greater the level of automation in the system, the more universally applicable the system will be.

The system should facilitate the accurate extraction of shape information to ensure that the models can be personalised and easily integrated into different environments. Thus the model should conform to a standard representation. The extraction of the shape information should be robust, to extract the shape from a minimum number of views in any environment, and should not be dependent on the individual adopting a pose for a prolonged period.

---

<sup>3</sup>A home-user (or non expert user) is classified as a user with little or no expertise in creating 3D content and who does not have access to any sophisticated equipment.

The reconstruction process must be flexible to allow the home-user to change the appearance of their model for different environments. In gaming environments, this is important to allow the individual to take on the attributes of a character and to reuse existing animation data. It is essential that the deformation of the default character, to take on the shape of the individual, should be automatic and preserve the detail of the underlying model to reconstruct the non-convex contours is achieved while deforming to the extracted shape of the individual to provide a highly personalised model. This is particularly important when such data cannot be readily extracted from the image data.

By providing a flexible method for the creation of human models, it is possible to provide greater interactive experience in new and diverse on-line 3D applications. These applications include personalised fashion shows, self-diagnosis for certain medical conditions and the visualisation of the user in a new house or apartment, etc. To realise this, the models must be flexible and crucially conform to a standard representation.

Further motivation for this research is to explore the minimum requirements for the extraction of an individual from a real environment, to determine the minimal requirements for building a realistic human model, to demonstrate that it is possible to create accurate human models from a limited set of views and that silhouette-based reconstruction is a valid approach to human modelling. Thus, existing approaches for the creation of human models are examined in order to determine what aspects are most suitable for the creation of human models in real environments and to establish the key elements that constitute a flexible reconstruction process. In particular, the examination of existing techniques indicates what restrictions are placed on the individual, the quality of the 3D models and the cost in the system in terms of setup, expertise required to interpret the data and the flexibility of the created models.

An additional motivation that was not explicitly considered at the outset of this research, but which grew in significance as the research progressed, is the provision of an approach with a low-overhead that seamlessly permits the use of the captured data to modify an underlying model while ensuring that the existing internal structure is maintained. This has implications beyond the modelling of an individual, for example in morphing one object to approximate another shape; this is a common method in existing animation packages requiring considerable interaction and cannot in general, be applied to the object as a whole unless the object has a simple structure.

### **1.3 Contributions**

This thesis makes a number of important contributions in the two fields of image processing and machine vision. In particular, this thesis describes a complete system for the capture, creation and animation of virtual human models that can be used in various virtual environments. It begins by completing a review of active contour models and current reconstruction techniques that facilitates the determination of minimum requirements for the creation of human models using images captured in real environments. The theories that are expounded are validated with extensive testing using real-world data.

The significant aspects of this research are identified, and they consist of the following major contributions:

- This approach is innovative in that it attempts to impose a minimum number of constraints at all stages, from image capture through to the 3D modelling of the individual. In particular, the system captures only four images of the individual who adopts a static pose in each image. The extraction of the individual is achieved using a template that is automatically initialised and requires a limited amount of user interaction. Furthermore, the bounding volume is automatically created, as is the application of active-meshes. The provision of a system capable of enabling a non-expert user to create and modify their own models is an important achievement and step in providing greater access to new virtual experiences. A prerequisite for such system is that it permits the use of off-the-shelf technology and supports the capture of data in unconstrained real environments.
- In achieving the overall goal, there are two interwoven themes that contribute to a number of significant developments.
  1. The first is the development of a generally applicable template that can be used to automatically extract an individual from their environment using constrained active B-spline contours that are automatically initialised within the captured images. The final position of this template accurately describes the shape of the individual. The development and application of such a template for the accurate extraction of human shape information has not previously been identified in the extensive literature reviewed.
  2. The second contribution provides a logical step in generating the B-splines extracted from each view to perform the 3D reconstruction of an object as a 3D B-spline surface. It integrates the image data that is extracted in each view to create a 3D bounding surface using silhouette based reconstruction. This is combined with an underlying model to create a human model which can be easily animated with varying levels of realism and that conforms to the humanoid animation standard. This validates the use of silhouette-based approach to the reconstruction of human models.
  3. Thirdly, this approach culminates in a novel formulation of active-meshes for the active deformation of an underlying model to rebuild the fine data that cannot be extracted using silhouette based reconstruction from four views. This formulation provides a 3D active framework that uses the active contours extracted from 2D images, which contain important shape data, to create a 3D active surface. The active surface formulation is represented as a B-spline surface and incorporates large and small-scale deformations in a single formulation. Internal constraints maintain the structure of the vertices of the human model, while external forces deform the model according to the bounding volume mould. The strength of the internal constraints can be varied to allow the model to adopt the exact shape of the bounding volume or to preserve the internal structure. This novel implementation provides a uniform framework that can be simply and automatically applied to the entire human model.

In addition, auxiliary contributions associated with this research include:

- A method for the automated initialisation of parametric active contour models.

- The specification of criteria for the automatic insertion or removal of control points.
- The creation of different human models with different levels of realism and detail.
- The evaluation of the level of clutter in images.
- A simple texturing technique that can be applied to modelling photorealistic humans.
- A simplified method for storing 3D shape information.

## 1.4 Organisation

The organisation of this thesis is as follows:

**Chapter 2** discusses active contour models; in particular, it provides a complete review of the evolution of the active contour models since their inception in (Kass et al. 1987) and discusses the different formulations including parametric and geometric active contour models. This description contains a detailed mathematical description of the active contour model and how this has been optimised and reformulated to enable active contours to be used in diverse applications. This includes the reformulation in terms of B-splines and their general applicability for use as templates. This review facilitates the development of techniques that are described in chapter 4 for the extraction of the individual from their surroundings.

**Chapter 3** provides a description of available 3D reconstruction techniques for the creation of both rigid and articulate 3D objects. Following this, the techniques that have been applied to the creation of human models are discussed in detail with particular emphasis on photographic techniques which stress the importance of devising a complete system. This review provides a basis for the modelling techniques that are formulated in chapter 4.

**Chapter 4** describes the main design approaches that have been considered for the creation of virtual human models and the different techniques that have been developed for the personalisation of an underlying model and how the active contour models applied automatically in the form of pre-constrained templates for the extraction of the individual from the scene and the creation of the human models. This chapter also indicates the necessity for the provision of the active deformation of the underlying model and how this is formulated in 3D to seamlessly modify the underlying model while conforming to internal constraints.

**Chapter 5** contains an extensive set of tests and results that highlight the success and failures of the methods proposed in Chapter 4. This section contains a series of tests that were undertaken to verify the correct implementation of the techniques. It culminates by showing that the reformulation of the active-meshes enables significant enhancement for the modelling of human models and improves the photo-realism of the underlying model. Additionally, it shows that the process can be applied in general for the creation of advanced 3D content.

**Chapter 6** summarises the research undertaken and provides a review of the results. A list of publications stemming from this research is furnished. Finally, suggested future research directions are discussed.

# Active Contour Models

## 2.1 Introduction

Active contours models (ACMs) are a popular method used to solve a variety of image analysis and machine vision tasks. They are often called “snakes” because of the fact that they appear to slither across images to arrive at the desired solution. ACMs are an example of the general technique of matching a deformable model to an image using energy minimisation. The fundamental theory of the ACMs was largely developed by Kass et al. (1987). Snakes have been applied in various guises and in various imaging domains, most notably in the medical imaging domain (McInerney & Trezopoulos 2000), and the use of templates has enabled snakes to be applied successfully in other image domains for the analysis of dynamic image data and 3D image data (Curwen & Blake 1993).

ACMs are energy minimising curves that continuously deform and minimise to fit desired image features. They provide a low-level means of detecting simple image information, including light and dark lines, edges and terminations. Furthermore, ACMs facilitate the combining of this image information in a meaningful manner that enables its use in higher-level processes. In particular, the use of snakes facilitates the accurate definition of object boundaries. One of the significant advantages that ACMs and, in general, deformable templates have over other competing low-level imaging techniques is that they facilitate the incorporation of higher-level information in the definition of the template, encompassing elements of image data and model-based control strategies (Sonka et al. 1999).

The classification of ACMs as deformable templates can, in part, be achieved by analysing how they locate features of interest, how they represent the actual contour and by the target applications. In (Kass et al. 1987), the features are located by placing either an open or closed contour in the vicinity of the feature (or object) that is to be extracted. The minimisation process then guides the contour to locate the particular feature, while the balloon model, introduced by Cohen (1991), locates the particular feature by expanding to the strongest edge or feature in the region about the contour.

There are two common methods for the representation of active contour models. The first and the original is that of the parametric snake which is essentially an ordered set of discrete points

(also termed *snaxels*<sup>1</sup>). The second, which is more recent development and independent of the contour parameterisation, is the geometric active contour model. Both of these models facilitate the representation of the various forces that enable the contours to deform/evolve to the correct solution (Malladi et al. 1995, Caselles et al. 1997).

In the late 1980's and the early 1990's, the idea of using active contour model as templates was considered by many authors including Blake & Isard (1992), Blake & Isard (1998) and Cootes et al. (1992). They used the general energy minimising technique to allow the creation of deformable templates. The templates describe the general or average shape of the object to be extracted and then based on training sets the allowable movement of the control points was formulated. These templates were redefined by Blake & Isard (1998) using B-splines and applied successfully to motion tracking and other real-time applications.

The objective of this chapter is to study and compare the different theoretical frameworks of active contour models. It starts by considering the original ACM proposed by Kass et al. (1987). Then, parametric and geometric active contour models are described. In addition, the different variations on the original model are explored and assessed in this chapter along with the different elements that have been incorporated to facilitate the creation of templates including the reformulation of the active contour models to incorporate non-rational uniform B-splines (NURBS).

## 2.2 Active Contour Model Definition

Snakes as proposed by Kass et al. (1987) are designed to provide a unified approach to low-level vision tasks including: edge and line detection, motion capture and stereo matching. The overall objective is to design an energy minimising function whose local minima comprise the set of alternative solutions available to higher-level processes. The local image minima correspond to desired image properties. The snake is a model-based active contour initialised close to or around the object of interest by manually placing a discrete set of points called control points. The snake's energy depends on its shape and its location in the image. The fundamental difference between this method and any that have preceded is that it intends to provide higher-level processes with information about the image that is not confined to a single solution. In essence, low-level image information is extracted and then higher-level image processing applications can group or interpret the information in a manner dependant on the desired output. This is realised using an energy minimisation framework in which constraints are used to ensure that the minimum solution is obtained based on local information. In addition to this, snakes have the advantages that when, a gap exists in the edge image, the contour can still determine the objects boundary, and similarly when several smaller objects (or regions) constitute a whole object, the ACMs can be used to provide a single contour to outline the entire object.

There are two main elements in the definition of active contour models. The first is the definition of a deformable contour model that is used to extract objects from a scene. This includes the definition of forces that are used to control the deformation of the contour. The second is the energy minimisation process that is undertaken to arrive at the desired solution.

---

<sup>1</sup>The term *snaxel* is derived from the contraction of the term *snake elements*.



## 2.2.1 Contour Representation

In (Kass et al. 1987) the snake is defined parametrically using an ordered number of control points that are placed close to the object of interest in the image. The snake is created by a spline that interpolates all the control points. It is possible to have either an open or closed snake, for example see Figure 2.1. The positions of the control points define the shape and energy of the snake. The contour is represented as an ordered set of  $n$  points  $v_i = (x_i, y_i)$   $i = 0 \dots n - 1$ . Thus every control point has a 2D position in the image.

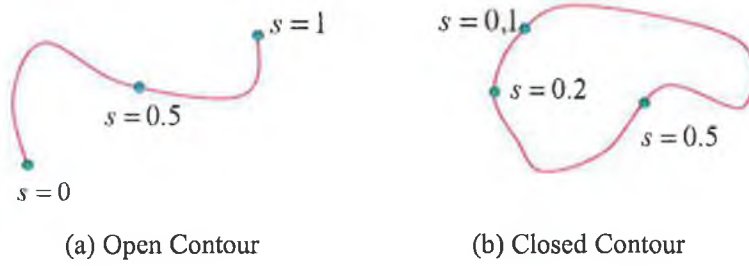


Figure 2.1: Examples of Initial Contours

The original snakes are modelled

$$E_{Snake} = \int_0^1 E_{Snake}(v(s)) ds \quad (2.1)$$

$$E_{Snake} = \int_0^1 [E_{Internal}(v(s)) + E_{External}(v(s)) + E_{Constraint}(v(s))] ds \quad (2.2)$$

Defining the snake as a closed contour has the advantage of periodicity and also eliminates some of the difficulties that are encountered in calculating the energy at the first and last points. This will be more evident in the discussion on internal energy below. The energy contained in the contour consists of two parts: the internal and the external energy. The internal energy will move the contour towards a smooth curve and the external energy will pull the contour towards regions or features with desired properties.

### The Internal Energy

The internal energy is composed of a first and a second order terms which control the elasticity and the stiffness of the snake. The internal energy works to minimise the distance between control points. This minimisation is counteracted by the external forces which attract the control points to features in the image. The relative significance of these forces is controlled by weighting parameters. The internal energy is sometimes referred to as the regularisation term and using the thin plate model, it can be expressed as

$$E_{Internal} = (\alpha(s)|\mathbf{v}_s(s)|^2 + \beta(s)|\mathbf{v}_{ss}(s)|^2)/2 \quad (2.3)$$

where  $v_s \equiv \frac{\partial v}{\partial s}$  and  $v_{ss} \equiv \frac{\partial^2 v}{\partial s^2}$  and  $\alpha$  and  $\beta$  are arbitrary functions that determine the contours tension and rigidity respectively. The functions  $\alpha$  and  $\beta$  provide the possibility to change the topology of the elastic contours. There are no guidelines for setting  $\alpha$  and  $\beta$  (Williams & Shah 1992). In several implementations, their values are taken as constant and (Fua & leclerc 1990) (Neuenchwander et al. 1994) show that the values chosen are fairly image independent. In (Samadani 1989)  $\alpha$  and  $\beta$  are redefined as space-varying functions, which facilitates changes in the continuity. The first order term in Equation 2.3,  $v_s$  makes the snake behave like a thin membrane and the second order,  $v_{ss}$ , term acts like a thin plate and  $\alpha(s)$  and  $\beta(s)$  control the relative significance of each term. Setting  $\beta(s) = 0$  at a point allows the snake to become second order discontinuous and develop a corner. The value of the first order term will tend to be larger when the average spacing between the control points is large and the second order term will tend to be larger when the curve is bending rapidly.

The discrete representation of the internal energy is important for determining the snake's energy and for the implementation of the energy minimisation process in (Amini et al. 1988). Using the discrete approximation allows the image to be considered as a discrete grid, as opposed to the formulation of (Kass et al. 1987) which permits control points to lie between discrete coordinates.

$$E_{Internal} = \frac{1}{2} \sum_{i=1}^n \left\{ \alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i-1} - 2v_i + v_{i+1}|^2 \right\} \quad (2.4)$$

If the snake is defined as an open contour then some restrictions or boundary conditions have to be introduced to accurately update the position of the end points. In particular, the second order term relies on three points and thus it cannot be reliably calculated at the end points. If this term is omitted from the calculation of the energy at the end points then the end points, will move towards its nearest point causing the length of the snake to contract. On closer inspection of Equation 2.4, it can be seen that to minimise the first order term, the distance between two points must be minimised. This causes the contour to shrink, possibly to a point. Using the hard constraints that are introduced by Amini et al. (1988), the effects of this problem can be reduced.

In (Williams & Shah 1992), the first order term is reformulated to cause the control points to be evenly spaced on the contour and removes the shrinking behaviour of the contour. This is achieved by incorporating the average distance,  $d_{ave}$ , between control points into the first order term as follows:

$$d = d_{ave} - |\mathbf{v}_i - \mathbf{v}_{i-1}| \quad (2.5)$$

Where  $d$  is the regularised distance between the control points  $\mathbf{v}_i$  and  $\mathbf{v}_{i-1}$

This ensures that control points having distance near the average distance will cause the first order term to be closer to zero. This can be normalised by dividing by the largest value of the

neighbourhood that a control point can move to in a single iteration. Williams & Shah (1992) calculate the second order term as a measure of curvature and if the curvature is above a defined threshold then a corner is allowed to develop by setting  $\beta = 0$ .

### The External Energy

This discussion on the external forces will consider the image forces and constraints that influence the minimisation of the snakes. The external forces counteract the minimisation of the contour by increasing or reducing its attraction to a particular feature in the image. The image forces are derived from the image data over which the snake lies. Several examples of these forces are found in the literature and some of the most common are described in this section.

Each of the external forces that are used to control the evolution of the snake are combined using the following formulation:

$$E_{External} = \omega_{Line} E_{Line} + \omega_{Edge} E_{Edge} + \omega_{Termination} E_{Termination} \quad (2.6)$$

where  $\omega_{line}$ ,  $\omega_{edge}$  and  $\omega_{termination}$  are used to control the significance of each element, i.e. to control its attraction to a particular feature. If additional external forces are required, they are simply added to the above equation.

#### 1. Line Functional

The line-based functionality is used to attract the snake to lines of a particular intensity. By adjusting the sign of  $\omega_{line}$  the snake can be either attracted to light or dark lines.

$$E_{line} = I(x, y) \quad (2.7)$$

where  $I(x, y)$  is the image intensity at a particular point.

#### 2. Edge Functional

In general, the edge information is calculated in the pre-processing stage. A description of the different operators available for the creation of edge maps are described in (Sonka et al. 1999). The edge-based functional is the most widely used external force. These forces are used to guide the contour to areas where edge information is strongest. In (Kass et al. 1987), no specific edge detection technique is specified and the edge functional is defined as:

$$E_{edge} = -|\nabla I(x, y)|^2 \quad (2.8)$$

where  $\nabla$  is defined as the gradient and the negative sign means that the contour is attracted to strong edges.

In (Williams & Shah 1992), the discrepancies in gradient magnitude are addressed by scaling the values of the gradient-based on its neighbourhood. This results in the following edge functional:

$$\left( \frac{min - mag(v_i)}{max - min} \right) \quad (2.9)$$

where  $min = min|grad(x_i, y_i)|$  and  $max = max|grad(x_i, y_i)|$  are the minimum and maximum value in the gradient in the neighbourhood of a control point respectively and  $mag(v_i) = |grad(x_i, y_i)|$  is the magnitude of the edge at the control point  $v_i$  and is calculated using Equation 2.8. This gradient term is negative so that points with large gradient values will have small values attracting the contour to these points. If the difference between minimum and maximum values in the neighbourhood is less than 5, then the denominator in Equation 2.9 will be set to  $5^2$ . This provides a measure that prevents large differences in the value of the image energy when the gradient magnitude is measured in a nearly uniform neighbourhood.

Jacob et al. (2004) highlight that the gradient-based edge energy is heavily dependent on the parameterisation of the snake, i.e. if the snake is re-parameterised in terms of a parameter  $s' = w(s)$ , where  $w$  is a monotonically increasing one-to-one warping function, it will result in a different value for the gradient-based edge energy. In addition to this, the use of the scalar gradient energy will result in the control points being attracted to regions of high gradient. In (Jacob et al. 2004), a new gradient-based image energy is proposed that takes the integral of the scalar field derived from the gradient vector field. This is expressed as

$$E_{grad} = \oint_C \mathbf{k} \cdot (\nabla(I(s)) \times d\mathbf{r}) \quad (2.10)$$

$$= \oint_C \nabla(I(s)) \cdot \underbrace{(d\mathbf{r} \times \mathbf{k})}_{\|d\mathbf{r}\|\hat{\mathbf{n}}(r)} \quad (2.11)$$

where  $\mathbf{k}$  is the unit vector orthogonal to the image plane<sup>3</sup> and  $\hat{\mathbf{n}}(r)$  denotes the inward unit normal to the curve at  $r$ . This is illustrated in Figure 2.2.

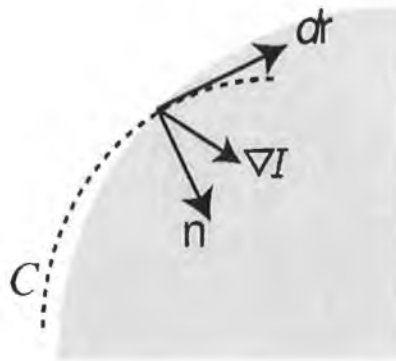


Figure 2.2: Gradient and normal to the curve.  $\nabla(I(s))$  denotes the gradient at  $s$ ,  $d\mathbf{r}$  is the normal along the contour,  $C$ , and  $n$  is the inward unit normal vector (Jacob et al. 2004).

<sup>2</sup>The main reason for having a cut-off value of 5 can be seen using the following values: 47, 48, 49. Using these Figures the gradient magnitude would be 0, -0.5, or -1.0 for points with essentially the same gradient magnitude. On the other hand with 5 as a minimum value the gradient term would be -0.6, -0.8, or -1.0 which is a more accurate representation of the similarity of the values.

<sup>3</sup>The vector  $\mathbf{k}$  is chosen depending on the direction in which the curve is described, such that  $\hat{\mathbf{n}}(r) = \frac{d\mathbf{r} \times \mathbf{k}}{\|d\mathbf{r}\|}$  is the inward unit normal (Jacob et al. 2004).

It has been proposed by Fua & leclerc (1990) to replace the gradient magnitude by its logarithm since the external energy is proportional to the gradient information, which can vary rather rapidly due to image contrast or noise. Other approaches to the problem have suggested that the direction as well as the magnitude should be considered as a solution to getting more precise edge information (Xu & Prince 1998b), but in (Neuenschwander et al. 1994) it is claimed that the result of the different methods, including using the Euclidean distance from the control point to the nearest edge, provide very similar results.

### 3. Termination Functional

The termination functional sometimes referred to as the corner functional, is used to guide the snake to line terminations and corners within the image. This is achieved using a slightly smoothed version of the original image<sup>4</sup>,  $H(x, y) = G_n(x, y) * I(x, y)$ , where  $G_n(x, y)$  is a Gaussian operator applied to the original image. Let  $\theta(x, y)$  be the gradient directions along the snake in the slightly smoothed image and let

$$\mathbf{n} = (\cos\theta, \sin\theta) \quad \text{and} \quad \mathbf{n}_\perp = (-\sin\theta, \cos\theta) \quad (2.12)$$

be unit vectors along and perpendicular to the gradient directions  $\theta(x, y)$ . The curvature of the level contours in the smoothed image can then be written as:

$$E_{Term} = \frac{\partial\theta}{\partial\mathbf{n}_\perp} = \frac{\partial^2 H / \partial\mathbf{n}_\perp^2}{\partial H / \partial\mathbf{n}} \quad (2.13)$$

$$E_{Term} = \frac{H_{yy}H_x^2 - 2H_{xy}H_xH_y + H_{xx}H_y^2}{(H_x^2 + H_y^2)^{\frac{3}{2}}} \quad (2.14)$$

In (Kass et al. 1987), the line terminations and corners are extracted in subjective contour illusions to illustrate that when insufficient image information can be reliably extracted from the image the active contours can be used to generate an accurate contour and the internal constraints of the active contours provide a smooth curve. In general, when sufficient edge information exists, attraction to line terminations can be incorporated into the edge energy functional.

### 4. Springs and Volcanoes

In addition to using image information, it is possible that user specified elements can be used to constrain the movement of the snakes (Kass et al. 1987). These forces are implemented as springs and volcanoes and provide forces that push and pull the snake towards or away from certain features or regions respectively. The spring forces are usually used to force the contour to move towards a desired feature, see Figure 2.3 (b). One end of the spring is attached to the contour and the other connected to a fixed point, connected to another point

---

<sup>4</sup>The smoothing of the image is an important step in the reliable generation of the second derivative and establishing the zero-crossings. The effect of the smoothing is to reduce the noise (Sonka et al. 1999)

on the snake or dragged by the mouse. The spring force between two points  $v_1$  and  $v_2$  is defined as  $-k(v_1 - v_2)^2$  and this term is simply added to Equation 2.6. The volcano is used to push the contour away from one region in the image and towards another region or to push it out of one local minimum and into another, see Figure 2.3 (a). This force is defined as a repulsion force and can be defined as  $\frac{1}{|r|}$  where  $r$  is the distance between a control point  $v_i$  and a reference point  $v$ . It is possible that more than one spring force or volcano can be used to guide the snake to the correct solution.

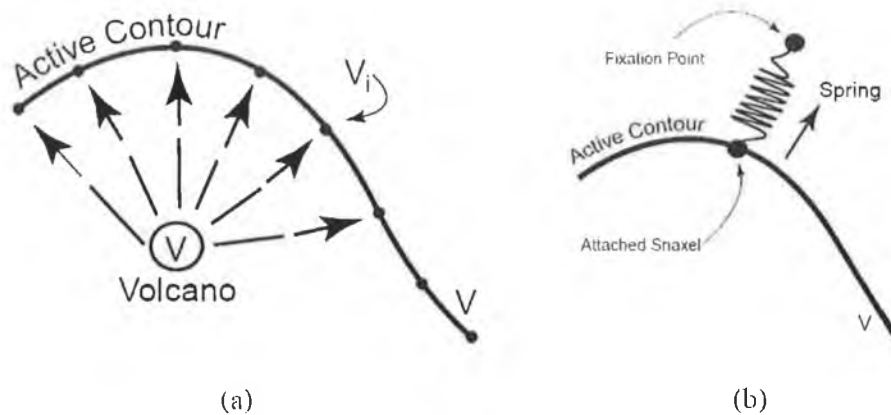


Figure 2.3: (a) the volcano pushing the contour away and in (b) the spring force attached to a control point on the contour (Molloy 2000).

## 2.2.2 Energy Minimisation

The energy minimisation procedure adopted plays a critical role in the success of a particular algorithm. The original algorithm of Kass et al. (1987) obtains a minimum using the Euler-Lagrange method with finite differences. Some of the potential problems associated with this method were highlighted in (Amini et al. 1988, 1990) regarding the stability of the algorithm. In this section, different minimisation techniques that have been applied to minimising the energy of active contours are discussed in relation to a number of parameters relating to the stability of the method, the correctness of the solution, the running time, memory allocation and the robustness of the algorithms.

The minimization process in (Kass et al. 1987) is a finite difference method (FDM). This is an iterative technique based on the calculus of variations and takes implicit Euler steps with respect to the internal energy and explicit Euler steps with respect to the image and external constraint energy. This approach can be described as a finite difference method in which the contour behaves as a set of masses linked by a zero length string (Cohen 1991). This results in the contour shrinking to a point if no external or image forces act on the snake. The system of equations that results from the Euler-lagrange equation requires  $O(n)$  time which is typical of an iterative method. In addition to the stability issues, this approach is sensitive to local noise because the energy is only calculated at each of the control points and not at all points along the contour. However, if the number of

control points in the contour are increased, then the gap between the control points decreases which causes the stiffness of the snake to increase<sup>5</sup>.

Other approaches have been proposed. These include:

- Finite Element Method (FEM),
- The dynamic programming approach and
- The fast algorithm and the greedy fast algorithm - both of which are variations on the dynamic programming approach.

### Finite Element Method

A finite element method is proposed by Karaolani et al. (1992) as an improvement on the finite differences method proposed by Kass et al. (1987). This method makes the active contours more sensitive to fine detail in the image by sampling the image forces that act along the length of the active contour. This is in contrast to the original implementation in which the image is only sampled at the control points.

The contour  $v(s)$  is divided into a number of elements<sup>6</sup>, in which the elastic, stiffness and image forces seek their own local minima. The snake is expressed as a hermite cubic spline with low order continuity at joints. Splitting the contour into different elements results in the computation of  $2n_e$  equations, where  $n_e$  is the number of elements. At each element, an approximation to the length of the element (in the  $(x - y)$  space) is computed and the number of elements is proportional to the length in pixels.

This approach shows that by sampling the contour along its length it can provide more accurate minimisation and, moreover, sampling between the control points is more computationally efficient than increasing the number of control points. One of the possible drawbacks (limitations) of this method is that it requires the individual to specify *a priori* the number of control points based on the expected size of object of interest. This requires the use of prior knowledge or to provide an initial segmentation of the object and means that the sample locations are based on the initial contour. This can be overcome in the dynamic programming approach where it is possible to increase the number of control points as the contour evolves.

In the paper by Cohen & Cohen (1993), the finite element method is put forward as a method that can reduce the number of control points, which is important in the extension of active contours to 3D. A series of similar experiments to Karaolani et al. (1992) is described highlighting the fact that FEM has a lower complexity than FDM and provides more stable results.

### Dynamic Programming Approach

Since the snake is active, it is always trying to minimize its energy and thus it displays dynamic behaviour and lends itself to dynamic programming. Amini et al. (1990) proposed an approach

---

<sup>5</sup>This decision must be taken prior to the initialisation of the active contour because the Euler-lagrange minimisation does not facilitate the dynamic insertion or removal of control points

<sup>6</sup>An element is described as the distance between two control points.

to solve the minimisation procedure that overcomes instabilities that exist in the original Euler-Lagrange method and the fact that the control points tend to bunch together at a strong portion of the edge. In addition to this, the dynamic programming approach allows hard constraints to be hard coded into the movement of the snake. This ensures that the algorithm is independently stable. This approach takes into account the discrete nature of the problem that is faced when working with digital images.

In general, in dynamic programming solutions to the minimisation problem the contour is considered as the position vector  $\mathbf{v} = v_1, v_2, \dots, v_n$ , where  $v_i = (x_i, y_i)$  and the energy of the snake is given by

$$E_{Snake} = \sum_{i=1}^n E_i \quad (2.15)$$

where

$$E_i = \frac{\left\{ \alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i-1} - 2v_i + v_{i+1}|^2 \right\}}{2} + E_{External}(v_i) \quad (2.16)$$

The approach assumes that the position vector  $v_i$  can have at most  $m$  degrees of freedom. The vector therefore can take only a finite set of values. This introduces a computational limitation on the algorithm, because by increasing the size of the neighbourhood (degrees of freedom), the number of operations increases substantially, i.e. the operations at each iteration are of the order of  $O(nm^3)$  (Chandran & Potty 1998). This situation with 9 degrees of freedom is shown in Figure 2.4. The situation illustrated shows the minimal path within the search space about the respective control points. This method, like that of Kass et al. (1987), will cause the snake to converge to a single point if it is not initialised correctly or if no image forces are available.

The dynamic programming approach was considered the most suitable approach for the extraction of the individual from a real environment. The primary reasons for this are the ability to incorporate hard constraints in the model and the ability to add additional control points. These reasons are further detailed in Section 4.3.5 and tested in Section 5.3.3

### The Fast and Greedy Fast Algorithms

Williams & Shah (1992) detail a fast method for solving the minimisation process that includes the hard constraints introduced by Amini et al. (1988). This method is an order of magnitude faster than the previous approaches. This approach includes a continuity term and a curvature term in addition to the image and external energy terms. A different formulation is used for the continuity term that ensures that the control points are more evenly spaced on the contour. Alternatively, in the fast greedy algorithm proposed by Lam & Yan (1994), which is a fast iterative method for calculating the minimal energy, relies on the assumption that the neighbours of the control point having minimum value also have small values, and thus the search space can be halved by considering alternative searches of the neighbourhood<sup>7</sup>. These alternative search patterns are shown in Figure 2.5. By alternating these searches, the whole neighbourhood will be searched.

<sup>7</sup>The patterns 1 and  $-1$  are used by Lam & Yan (1994) to signify alternative search patterns.



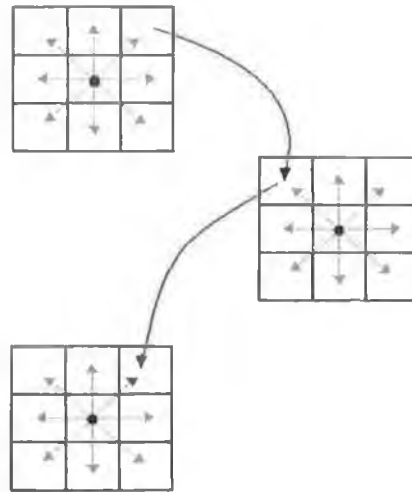


Figure 2.4: This figure depicts the correspondence between a decision set and image pixels for dynamic programming with number of states,  $m = 9$ . The curve indicates the optimum position of each control point based on a particular criteria set. Each of the control points can move to one of the nine squares in the grid at a particular iteration. The position that is adopted minimises the energy configuration for a particular iteration. This is indicated by the curved arrows.

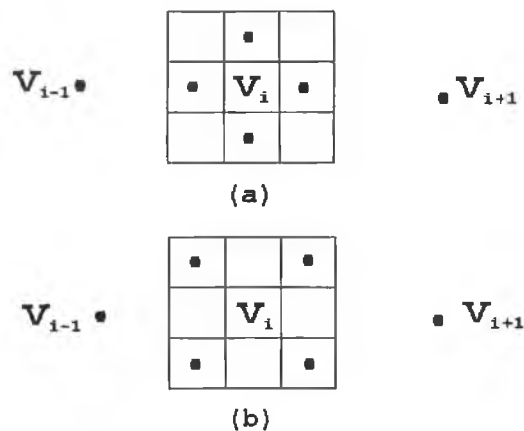


Figure 2.5: (a) neighbourhood searched when pattern = 1 in fast greedy algorithm, (b) neighbourhood searched when pattern = -1 in fast greedy algorithm. This approach reduces the number of operations at each iteration, assuming that the area around the minimum will also contain small values (Lam & Yan 1994).

## 2.3 Problems with the Snakes

There are several elements in the formulation of active contours models proposed by Kass et al. (1987) that reduce the automated application of snakes. This section follows on from Section 2.2.2, which described alternative methods for minimising the contour's energy. It describes some of the problems of active contours and details techniques that have been proposed to overcome some of these issues.

In (Kass et al. 1987), the positioning of the snake requires an expert user to position the snake close to the contour of interest, and automated positioning of the contour was not considered. Automated approaches to the initialisation of the active contour are discussed in Section 2.3.1. This is followed by a discussion on the use of a regularisation term to determine the relative importance of the internal and external energy terms. In Section 2.3.3, the balloon model of Cohen (1991) provides a method to move the contour out of local minima with the inclusion of an inflation force. Dual Snakes described in Section 2.3.5 use two snakes to improve fitting active contour in face detection applications. Further to this, topologically adaptive snakes attempt to address the issues related to the change of topology of the active contours which is not possible in the original formulation and is also considered in Section 2.6 where geometric active contours are discussed.

In Kass et al. (1987), an iterative technique involving the calculus of variations has been applied to minimise the energy in the contour. There are a number of issues that are associated with this technique, including: stability and convergence (Chandran & Potty 1998). In particular, Amini et al. (1990) state that in the theory of calculus of variations, the concept of the absolute minimum is not clear defined and that the best solution to the iterative problem is a relative minimum. In addition, instability can be introduced in the calculation of higher order derivatives, since the problem is formulated on a continuous plane and solved using an approximate method.

### 2.3.1 Initialisation of the Snakes

The objective of initialisation is to place the contour in close proximity to the contour, so as to facilitate speedy convergence. This is one of the most important aspects of any approach because they rely on local image information to deform to the desired object. Snakes do not solve the entire problem of finding salient image contours and require other mechanisms to position them in the vicinity of the desired contours. Different methodologies involving the use of standard imaging techniques<sup>8</sup> have been developed to facilitate the automated initialisation of the contour (Neuenchwander et al. 1994). To date, these techniques are application specific and generally result in (or from) templates that are applied to extract particular objects from the scene. In addition to placing the contour somewhere near the desired solution, it is necessary to specify the approximate shape of the contour. This effectively amounts to an initial course segmentation of the image.

One of the most general and widely used techniques is that of the generalised Hough Transform that is described in (Ballard 1981). In image space, a pair of quantised parameters can represent a line segment. The complete set of possible lines for the whole image can be represented by an

---

<sup>8</sup>Standard imaging techniques involve the application of filters and smoothing algorithms to the image in an attempt to extract more information from the image. (Sonka et al. 1999)

accumulator array whose axes are the parameters characterising the line. Thus at each  $(x_i, y_i)$  edge image point, the accumulator array is incremented for every possible line through the point  $(x_i, y_i)$ . If many edge responses lie on the same line, then this results in a high value at the position in the accumulator array corresponding to that line.

An alternative method for the initialisation of the snakes is proposed by Neuenschwander et al. (1994), which enables the initialisation of the snake by specifying only the first and last control points. The user chooses these points. This reduces the complexity of the initial contour that is used to extract a particular feature. Then the image component at the end points is switched on. This is followed by the switching on of other control points successively from both ends moving towards the centre. This enables the snake to find the smoothest path. These are termed “static snakes” by the authors. Although this approach applies primarily to open contours, it highlights the advantages of using a reduced number of control points until the contour approaches a minimum, because in certain instances, when the initial contour is far from the final position the snake can get stuck in undesired local minima due to irrelevant edge information.

A fast method for the initialisation of active contours is proposed in (Mobahi et al. 2004) for the development of robots that can interact naturally with humans. This is achieved by using so called “Self Organised Contours” (SOC), which localise a region of interest (ROI). Rather than search the whole image, only a small fraction of the image is searched. The method first initialises a number of agents randomly over the image. This is the self-organisation phase of the approach. Regions that change between frames and edges are treated with higher significance and the behaviour of each agent depends on three parameters, namely position, velocity and energy. If the energy of an agent reaches zero, then the agent is discarded. When the self-organisation is complete, agents are only scattered over feature regions. The energy of an agent is initialised as a non-zero value which continually decreases over time and to zero if it is not in an active region.

### 2.3.2 Regularisation of the Snakes

The performance of the active contour model can be encouraged to be more robust through the regularisation of the internal and external energy terms in the form of a dynamic equation. As previously described, the internal energy of the snake imposes continuity and smoothness constraints on the snake while the external energy is used to attract the snake to salient image features. The introduction of a regularisation term can be used to control the significance of each of these constraints. The snake model can be regularised by the introduction of a regularisation parameter  $\lambda$ . From the discrete form of the internal energy in Equation 2.4, and following the notation in (Lai 1994), the regularisation parameter can be incorporated as follows:

$$V_{\Lambda} = \arg \min_v \left( \sum_{i=1}^n \lambda_i E_{Internal} + (1 - \lambda_i) E_{External} \right) \quad (2.17)$$

where  $\lambda_i \in [0, 1]$  are the regularisation parameters. By setting  $\lambda \gg (1 - \lambda)$  encourages regularisation and strong models that are resistant to noise while  $\lambda \ll (1 - \lambda)$  will increase the attractiveness of the snake to image information including noise.

In (Lai 1994), a technique for the regularisation of the snake through the use of the minmax principle permits the automatic determination of values of the regularisation parameter along the contour. This results in a trade-off at each iteration along the contour and is expressed in the form:

$$e(\mathbf{V}^*, \Lambda^*) = \min_{\mathbf{v}} \left( \sum_{i=1}^n \max(\mathbf{E}(\mathbf{v}_i)) \right) \quad (2.18)$$

where the solution snake  $\mathbf{V}^*$  and the solution regularisation parameters  $\Lambda^*$  ( $= \lambda_1, \lambda_2, \dots, \lambda_n$ ) are calculated by finding the snakes  $\mathbf{V}$  with the minimum energy determined by the sum of the maximum of Equation 2.17 for each value of  $\lambda_i$  appropriate to the control point.

### 2.3.3 The Balloon Model

Another approach to the energy minimisation process was suggested by Cohen (1991) based on the Galerkin solution of the FEM. The Galerkin solution of the FEM has the advantage of numerical stability and better efficiency (Press et al. 1992). This approach is applied to the closed contour case and finds exceptionally good stability and in addition, attempts to overcome the problems associated with the original snake, including the behaviour of minimising to a straight line when not placed in close proximity to a desired object. The balloon model uses an additional inflation force, so the balloon constantly inflates, passing through edge fragments that are too weak to contain the inflation but resting on stronger edge features. The applications are clear for the balloon model, such as medical imaging, tracking closed internal organs (such as the cavities in the heart) in noisy image scans (using noisy ultrasound and magnetic resonance images). It is very important in dealing with the balloon model that it is prevented from overstepping the edges of the feature of interest. Cohen (1991) used the FEM and prevents a step size of more than two pixels. The algorithm works very accurately in the example of tracking the contractions of the left ventricle of the human heart. This approach is particularly important in the case of closed or nearly closed contours. This formulation permits the snake to overcome isolated energy valleys resulting from spurious edge points.

The second problem that has been highlighted is that active contours have difficulties progressing into boundary concavities. Different methods have been proposed to address this problem, including multi-resolution methods, pressure forces, and distance potentials. One idea to overcome this problem is to increase the capture range of the external force fields and to guide the contour toward the desired boundary.

### 2.3.4 Fourier Snakes

In the article by Staib & Duncan (1992), a method for segmentation using of parametrically deformable models that use boundary and global shape information is proposed. The parametric model is based on elliptical Fourier decomposition of the boundary. This method is developed primarily for use in the extraction of natural objects and those found in biomedical images. The objects in general have a tendency towards some average shape with numerous variations near this average shape.

The Fourier representation expresses the curve in terms of an orthogonal basis. This allows the representation of any object as a weighted sum of a set of known functions and makes the parameters distinct and avoids redundancy. These can be used to generate all types of closed curves using relatively few parameters. The contour can be viewed as being composed as the sum of rotating phasors, each individually defining an ellipse and rotating with a speed proportional to their harmonic number. It is important for the curve representation that the Fourier components are continuous and periodic<sup>9</sup>. The Fourier descriptors can be used to describe open contours, although a straightforward representation of this would cause discontinuities. This is avoided by tracing along the curve and then retracing back to the start and forming a closed path. In this approach, the internal energy is calculated analytically, and the external energy is calculated by sampling the curve at regular intervals as well as computing image features (e.g. gradient magnitude and direction) at each point. The direction of the gradient is important as it determines the direction of greatest increase of the function value, and the iterative approach takes steps in the direction of the gradient.

### 2.3.5 Dual Active Contours

Gunn & Nixon (1994) introduced the notion of dual snakes to solve problems associated with face detection. In this approach, one active contour is initialised inside the desired feature and one outside it. This reduces the sensitivity of the initial contours to initialisation by constraining the space in which the active contour should lie. The two contours are coupled using spring forces that cause the two contours to be attracted to each other and to image features. This approach also has the advantage that additional higher-level information, such as geometric shape can be incorporated into the snake model (Gunn & Nixon 1994).

The mean contour is generated from the mean position of points on the inner and outer contours.

$$mean(s) = \frac{1}{2} [inner(s) + outer(s)] \quad (2.19)$$

Further to this, dual snakes have been generalised in (Gunn & Nixon 1995) and showed that they can improve the snake's ability to move into non concave areas. In addition, the dual snakes can be combined with a model in which the contours have no tendency to expand or contract other than to attain a prior shape. In (Gunn & Nixon 1996), the dual shapes are applied to accurately segment the boundary of the head using a dynamic programming approach in which each contour point is constrained to lie along a line joining the two initial contours. Each line is discretised into  $M$  points. In this approach, it is assumed that the frontal face image is captured against a plain (uniform) background and an initial estimate is required to place the snake. In (Gunn & Nixon 1997), the dual snakes are combined with a local shape model<sup>10</sup> to improve the parameterisation, and in (Nixon et al. 1997) are used to enable the extraction of parameter for face recognition by the establishment of a boundary that located the chin and the upper hairline.

<sup>9</sup>This is important as it ensures that the shape can be described in a single cycle through 360°.

<sup>10</sup>The local information is incorporated to make the two contours rotation, translation and scale independent, additionally the contour should be in equilibrium when it is similar to an estimated contour so that it has no preference to contract or expand other than to acquire its natural shape.

### 2.3.6 Topologically Adaptable Snakes

Topologically Adaptable Snakes (T-snakes), introduced by McInerney & Trezopoulos (1995), extend the functionality of the parametric snakes while retaining all of the features of traditional snakes, such as user interaction <sup>11</sup>, constraint forces and volcanoes etc. The additional functionality is achieved by superimposing a regular grid over the image. This grid is used to re-parameterise the deforming snake model iteratively. The deformation scheme extends the geometrical flexibility of the snake to deal with complex shapes.

The T-snakes are defined as closed contours and a set of control points connected by adjustable springs. Users can interact with this model by using spring forces and other constraints. The model's deformation is controlled by discrete Lagrangian equations of motion. One of the attractive features of this approach is that the control points and the interconnections do not remain constant as the snake moves towards the image features. The re-parameterisation is controlled by a grid, which is superimposed over the image. This makes the snake model relatively independent of its initial position.

#### T-Snake Model

The T-snake is defined as discrete set of  $N$  control points indexed  $i = 1, \dots, N$  connected in series by a set of  $N$  elements. Associated with each control point are time varying positions  $\mathbf{x}_i(t) = [x_i(t), y_i(t)]$ , tensile forces  $\alpha_i(t)$ , rigidity forces  $\beta_i(t)$ , and  $f_i$  external forces. This is expressed as a first order differential equation of motion:

$$m_i \ddot{\mathbf{x}}_i + \gamma_i \dot{\mathbf{x}}_i + \alpha_i + \beta_i = \mathbf{f}_i \quad (2.20)$$

where  $\ddot{\mathbf{x}}_i$  is the acceleration of node  $i$ ,  $\dot{\mathbf{x}}_i$  is its velocity,  $m_i$  is the mass,  $\gamma_i$  is the damping coefficient that controls the rate of dissipation of the kinematic energy of the control points and  $\mathbf{f}_i$  is an external force that attracts the model towards image edges. The term  $\alpha_i$  is a force that attempts to preserve the length of the snake and  $\beta_i$  is a rigidity force that resists bending. This equation is simplified by setting the mass density  $m_i$  equal to zero,  $m_i = 0 \forall i$ . This preserves the dynamic nature of the model and it comes to rest when the applied force balances the internal forces.

#### Simplicial Decomposition

The grid that is used to approximate the snakes motion is an example of the space partitioning by simplicial decomposition <sup>12</sup>. The simplicial decomposition provides an unambiguous framework for the creation of local polygon approximations of a contour or surface model. An example of the grid is shown in Figure 2.6 (a). The set of triangles in the grid that intersect with the contour form a 2D combinatorial manifold. The contour intersects each triangle at two distinct points, each of which is located on different edges. This means the edges of the intersected triangles to be used to approximate the contour. The vertices of the triangles can be classified based on the position of

<sup>11</sup>User interaction is described as a features as it is not generally incorporated in a geometric snake formulation see Section 2.6

<sup>12</sup>A simplicial cell decomposition is also called a triangulation.

the contour. The vertices of the triangles that are inside the contour have the same “sign”. There are three possibilities for sign assignment, these are shown in Figure 2.6 (b). All three vertices will have a negative sign if they are completely outside the contour or all positive if the triangle is completely inside the contour or a combination of positive and negative signs if the contour intersects the triangle.

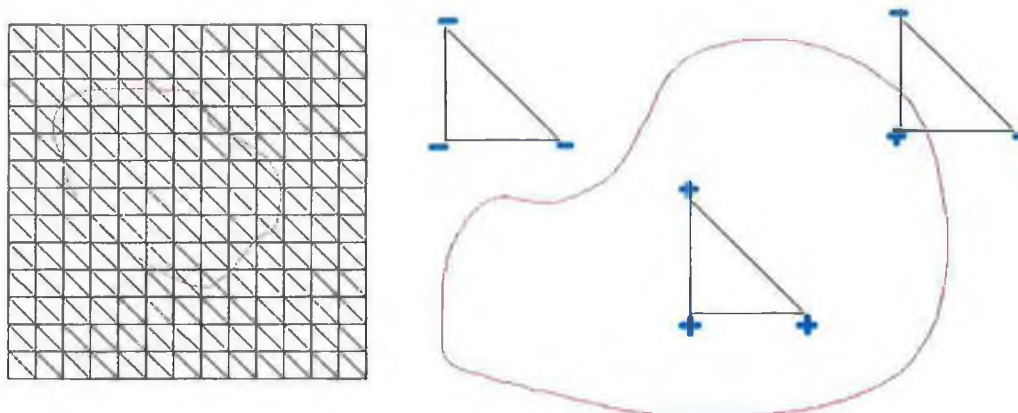


Figure 2.6: (a) simplicial approximation of the contour model constructed by subdividing the image using a cubic grid, (b) the cell classification (McInerney & Trezopoulos 1995).

This model is useful when contours intersect and topological changes are required. In addition to its positional information, each control point stores the edge and cell number it intersects and each boundary cell stores a reference to the control points which form the line intersecting the triangle. Therefore, when a snake intersects with another snake or collides with itself, a topological change is required (McInerney & Trezopoulos 1995). A decision is required when more than one line intersects a triangle. Then two line segment endpoints are chosen on different edges of these boundary triangles and they are connected to form a new line segment.

Both discrete and T-snakes suffer from the drawback that the movement of a single control point affects the entire length of the contour; a small change in the position of a control point tends to strongly propagate throughout the entire snake. This can be described as a global propagation of the change of a particular control point. Moreover T-snakes are not immune to initialisation, and while the ability to split and merge improves the flexibility of the original parametric snake model, it is not as applicable as implicit methods which can be used when the topology is unknown.

## 2.4 G-Snakes - Deformable contours: Modelling and Extraction

G-snakes is a general method that addresses the problems of modelling and extracting arbitrary deformable contours from noisy images. This approach attempts to combine global and local deformations. The motivation behind this is that global templates contain fewer parameters and cannot exercise local control along the contour. In contrast, local models generally contain more parameters and have good local control but are not suitable for the incorporation in global models as they tend to get trapped in local features (Lai 1994).

The G-snake or general approach is designed for the representation of arbitrary shapes. The contour model is based on a stable and regenerative shape matrix that is invariant and unique under rigid motions. This approach is applied to the image to locate objects that are similar to the initial contour. The similarity is modelled using Markov random fields<sup>13</sup> (Lai & Chin 1995, Lai 1994). In particular, this is achieved by representing a point as a linear combination of two independent vectors.

Global deformations correspond to the effects of rigid motion such as scaling, rotation, stretching and dilation. These operations can be represented by affine transformations. (Lai & Chin 1995, 1998) show that the regenerative shape matrix (or shape matrix) is unaffected by affine transformations. To represent a family of contours that exhibit small shape irregularities, an internal energy is induced from the shape matrix and a variance matrix, which contains location dependent weighted deformations. Then the Gibbs measure<sup>14</sup> is used to express the conditional probability of a given point as a conditional probability of its two basis points which is more realistic for modelling local deformations (Lai 1994).

In Lai & Chin (1995) state that the range of possibilities that exist when using a rigid template means that it may cover all possible transformations, although this can be improved by restrictions introduced by learning or by using prior knowledge. Moreover, the following assertion is made in favour of deformable templates:

*The expected correlation of a matched template decreases with deformation*

The proof of this assertion can be found in (Lai 1994) which illustrates that the use of rigid templates yields poor performance as the variation of expected deformation increases. Consequently, the proposed deformable template is generalised to account for this global variation caused by rigid motion while retaining the ability for local control.

The active contours are generalised by redefining the internal and external energies to incorporate the global deformations. This formulation is analogous to that of Kass et al. (1987), although the original internal energy only constrains the solution to the class of controlled continuity splines<sup>15</sup>. This formulation generalises  $E_{int}$  by allowing for the incorporation of prior models to create an attraction towards a particular type of contour. This model is termed a generalised active contour model. The model is initialised through the generalised Hough transform.

The minimisation process that is employed in this method uses the properties of a random Markov field to localise the operations, i.e. the minimisation is decomposed into  $n$  independent

<sup>13</sup>Markov random field theory is a branch of probability theory that provides a foundation for the characterisation of contextual constraints and the derivation of the probability distribution of interacting features. The theory attempts to build the very general probabilistic model of the Ising model

<sup>14</sup>The Gibbs model is a probabilistic measure derived from the Ising model and defines the energy function that is used to measure the entropy (a measure of the amount of uncertainty in the outcome of an event or output of a system) of a system. The Gibbs distribution expresses the neighbourhood relationship i.e. the probability of a certain value at a point  $j$ , given all the values of probability at all other points of a lattice is the same as the probability of the value at that point by considering the neighbours of the point  $j$ . This is expressed mathematically as  $P(\sigma_j = a | \sigma_k, k \neq j) = P(\sigma_j = a | \sigma_k, k \in N_j)$  where  $N_j$  is the neighbourhood of point  $j$ . This property is also indicative of a Markov random field.

<sup>15</sup>Handling local contour deformations is based on learning from a set of training examples. The absence of domain knowledge about the object shape may cause inaccuracies in the tracking, especially when the motion is non-rigid (Lai & Chin 1995).



stages, where each stage considers only three neighbouring points. This idea was first proposed by Amini et al. (1990) using dynamic programming.

A linear search is used in the minimisation algorithm. The search region must be sufficiently large to include at least part of the solution otherwise the contour will not deform to the correct solution. There are a number of search strategies considered. These are designed to search large regions without increasing the complexity of the algorithm. In the initial stages, it is prudent to rapidly inflate or deflate the contour to locate the neighbourhoods of the global minima. This is achieved by searching in the normal directions at each control point (Lai & Chin 1998). The basic line search restricts the search region which contains all the points on the normal vector. The stratified line search extends this idea to encompass even larger search by breaking the region normal to the points into disjoint segments.

## 2.5 Reformulation of the Active Contour Model

To facilitate the use of active contour models in different applications, the original snake model has been reformulated to incorporate different external forces and different minimisation techniques. To facilitate the use of snakes in real-time, the original energy-minimising spline has been reformulated in terms of B-spline contours. This has several advantages including local control over the contour. The more recent reformulation in terms of non-rational uniform B-splines curves (NURBS) provides even greater local control by the inclusion of weights that can be adapted depending on the curvature of the curve (Meegama & Rajapakse 2003). Reformulation in terms of B-splines and NURBS is described in sections 2.5.1 and 2.5.2 respectively, including the particular advantages of each method.

These reformulations make it easy to increase the number of control points in the snake and to examine the external forces that act along the length of the contour, a problem which was not considered in the original paper of Kass et al. (1987) and which enables the model to increase the local flexibility of the contour. In addition to this, the NURBS snake has the added ability, through the use of weights, to increase the local flexibility without increasing the number of control points.

### 2.5.1 B-spline Snakes

In the original snake formulation, the snake's convergence was rather slow particularly in the isolation of corners. This led to the developments described in Section 2.2.2. In (Amini et al. 1988), the curve is approximated by a polygonal approximation of the curve, but this can no longer guarantee smoothness and the only way to include a corner is to set  $\alpha = 0$  at some locations. A better approach proposed by Menet et al. (1990) uses a parametric B-spline approximation of the curve. In this model, the curve is replaced by a B-spline approximation and the energy of the approximation is minimised. This model is referred to as the "*B-snake*" model. A description of B-spline curves and surfaces can be found in Appendix A.

Using the B-spline approximation, the curve is split into segments (spans) and the breakpoints between each segment, called knots, which are common to two neighbouring curve segments. Each of these segments is approximated by a piecewise continuous polynomial function which

can have any order<sup>16</sup>,  $k$ , and is obtained as a linear combination of basis functions  $N_i$  and a set of control points  $P_i$  expressed as

$$C(u) = \sum_{i=0}^m N_i(u)P_i = \sum_{i=0}^m (X_i N_i(u), Y_i N_i(u)) \quad (2.21)$$

The  $\{N_i(u), i = 0, 1, \dots, m\}$  are piecewise polynomial functions that form a basis for the vector space of all piecewise polynomial functions of the desired degree and continuity for a fixed knot sequence. The control vertices form a control polygon which exhibits a strong convex hull property and the curve is contained within the convex hull of its control polygon (Piegl & Tiller 1997).

The first stage involves finding a control polygon. This is achieved by performing a least squares fit of the data by the B-spline curve, effectively minimizing the distance between the original data and the approximation. This is achieved using a least squares approximation of the curve and is described using the following expression (Piegl & Tiller 1997):

$$R = \sum_{j=0}^p |C(u_j) - D_j|^2 = \sum_{j=0}^p ((x(u_j) - x_j)^2 + (y(u_j) - y_j)^2) \quad (2.22)$$

where  $p + 1$  is the number of discrete data points on the curve and  $u_j$  is some parameter associated with the  $j$ th data point and  $C(u_j)$  is given by Equation 2.21. This equation is then solved using LU decomposition to provide the approximation of the curve. The choice of the number of vertices,  $m + 1$ , determines how close the B-spline curve approximates the original data. The B-spline curve is then substituted into the discrete approximation of the snake Equation 2.4. Then the equation to be minimised becomes (Menet et al. 1990):

$$E = \sum_{j=0}^p \left\{ \frac{1}{2} \alpha(u_j) \left[ \left( \sum_{i=0}^m (X_i N_i'(u_j))^2 + \left( \sum_{i=0}^m Y_i N_i'(u_j) \right)^2 \right) + \frac{1}{2} \beta(u_j) \left[ \left( \sum_{i=0}^m (X_i N_i''(u_j))^2 + \left( \sum_{i=0}^m Y_i N_i''(u_j) \right)^2 \right) + F(v(u_j)) \right] \right\} \quad (2.23)$$

The local control over the B-spline means that moving the position of one control point only effects a small part of the curve. The continuity of a B-spline can be changed at a control point by the use of multiple knots. Give that the continuity at a particular knot is defined as  $C^{k-2}$  and with  $\mu$  the multiplicity degree at a knot the continuity is then reduced to  $C^{k-1-\mu}$ . When  $\mu = k - 1$  the corresponding control point is interpolated. This property enables the development of a corner by reducing the continuity at a particular knot.

In addition, it has been shown that in general, representations using B-spline basis functions require fewer parameters than point based approaches and thus result in faster optimisation algorithms. Also, such models have inherent regularity and do not require additional constraints to

<sup>16</sup>A  $k$ th order polynomial has  $k$  coefficients. Therefore a quadratic has order 3 and a cubic polynomial has order 4

force smoothness (Jacob et al. 2004). In (Kim 1999), the energy within the B-spline was discretised and minimised using a dynamic programming approach.

In (Tang & Zhuang 1998), an adaptive B-spline snake is proposed. An optimal edge detection filter is used to extract edge potentials at different resolutions. Then a non-uniform B-spline curve is used to represent the contour and to approximate the image edge as close as possible. The adaptive B-spline active contour model uses non-uniformly distributed control points although initially uniformly distributed points are used along the initial contour with interval  $t$  between the B-spline control points for fast computation. These control points form the initial vector  $v^o$ . During the evolution, a displacement measure  $v^t - v^{t-1}$  is used to adjust the control points, and then obtain a new active contour, along which a new group of uniformly distributed nodes with interval  $t$ . The procedure is continued until the energy of the active contour is minimized. Then the contour is searched for segments with high curvature or hard corners and redefined with  $N$  control points and  $M$  discrete curve points  $q(i)$  along the B-spline active contour. The points are re-distributed according to the curvature along the curve. A greater the density of control points is necessary to represent a curve with higher curvature. Thus the resolution of the B-spline curve is adapted to the curvature of the contour.

## 2.5.2 NURBS Snakes

In certain situations, the use of Non-Uniform Rational B-splines (NURBS) curves is considered to provide greater control using fewer control points. The greater flexibility is introduced to the snake model by the use of weighting parameters. Although B-spline snakes perform better than traditional snakes, individual control points need to be duplicated to achieve high curvature and force the curve to interpolate the control points. Meegama & Rajapakse (2003) propose a snake model that uses NURBS in which the weighting parameters are automatically adjusted to control the flexibility of the curve at each control point. The weighting parameters are adapted according to the curvature of the contour without increasing the number of control points.

(Meegama & Rajapakse 2003) provide a definition of the snake model in terms of NURBS and describe how the internal and external forces are included within this framework. In Appendix A, NURBS, which extends the flexibility of B-splines, are defined as

$$v(s, t) = \frac{\sum_{i=0}^{n-1} w_i^t N_{i,k}(s) \mathbf{p}_i^t}{\sum_{i=0}^{n-1} w_i^t N_{i,k}(s)} \quad (2.24)$$

$$= \sum_{i=0}^{n-1} R_i^t(s) \mathbf{p}_i^t \quad (2.25)$$

where

$$R_i^t(s) = \frac{w_i^t N_{i,k}(s) \mathbf{p}_i^t}{\sum_{j=0}^{n-1} w_j^t N_{j,k}(s)} \quad (2.26)$$

This can be expressed in matrix form as  $v(s) = \mathbf{p}^T \mathbf{R}(s)$  where  $p = (p_0^t, p_1^t, \dots, p_{n-1}^t)^T$  and  $R(s) = (R_0(s), R_1(s), \dots, R_{n-1}(s))^T$ .

This is incorporated into the snake model by defining the internal energy as:

$$E_{int}(\mathbf{v}(s)) = \alpha |\mathbf{p}^T \mathbf{R}'(s)|^2 + \beta |\mathbf{p}^T \mathbf{R}''(s)|^2 \quad (2.27)$$

and the external energy as

$$E_{ext}(\mathbf{v}(s)) = -\gamma |\nabla I \mathbf{p}^T \mathbf{R}(s)|^2 \quad (2.28)$$

The internal and external energy are combined within the energy minimisation formulation so that the position of a control point at time  $t + 1$ ,  $\mathbf{v}(s, t + 1)$  is given by

$$\mathbf{v}(s, t + 1) = \arg \min_{\mathbf{v}(s, t)} (E_{int}(\mathbf{v}(s, t)) + E_{ext}(\mathbf{v}(s, t))) \quad (2.29)$$

### Curvature based weight adjustment

The use of NURBS provides the possibility to have greater local control on how the contour can approximate the image information. This section describes how the weights of a NURBS snake effectively control the local shape of the contour based on its curvature properties, without moving or duplicating the relevant control points. Let  $\mathbf{v}(s, w_i)$  be the family of curves obtained by changing the weight  $w_i$  at a control point  $\mathbf{p}_i$ , and  $\mathbf{v}(s, w_i = 0)$  is the set of curves when  $w_i$  is set to zero. Then, from Piegl & Tiller (1997)  $\mathbf{v}$  can be expressed as:

$$\mathbf{v}(s, w_i) = \mathbf{v}(s, w_i = 0) + \tau (\mathbf{p}_i - \mathbf{v}(s, w_i = 0)) \quad (2.30)$$

where

$$\tau = \frac{N_{i,k}(s)w_i}{\sum_{j=0}^{n-1} N_{j,k}(s)w_j} \quad (2.31)$$

From Equations. 2.30 and 2.31, it is clear that as the weight  $w_i$  associated with the control point  $\mathbf{p}_i$  is increased,  $\tau$  is increased, and hence, the point  $\mathbf{v}(s, w_i)$  moves toward  $\mathbf{p}_i$  along the vector  $\mathbf{p}_i - \mathbf{v}(s, w_i = 0)$  as shown in Figure 2.7. Similarly,  $\mathbf{v}(s, w_i)$  moves away from  $\mathbf{p}_i$  if  $\tau$  is decreased.

Let  $\mathbf{p}_i^t$  denote the position of a control point with weight  $w_i^t$  at time  $t$  and  $\kappa(\mathbf{v}(s, t))$ , the curvature at a point  $\mathbf{v}(s, t)$  If the maximum curvature within the spline segment  $\mathbf{v}(s)$ ,  $s \in [s_i, s_{i+p+1})$ , at time  $t$ ,  $K^t$ , exceeds a pre-defined value (for example the average curvature along the NURBS curve), the new weight  $w_i^{t+1}$  of the control point  $\mathbf{p}_i^{t+1}$  is updated such that

$$w_i^{t+1} = w_i^t + \eta \frac{K^t}{\|\kappa(s, t)\|} \quad (2.32)$$

where the parameter  $\eta \in \Re$  controls the amount of attraction of the curve towards the control point and  $\|\kappa(s, t)\|$  is the maximum curvature of the closed NURBS curve. In Equation 2.32, the curvature  $K^t$  is divided by the maximum curvature in order to normalise the weight modification.

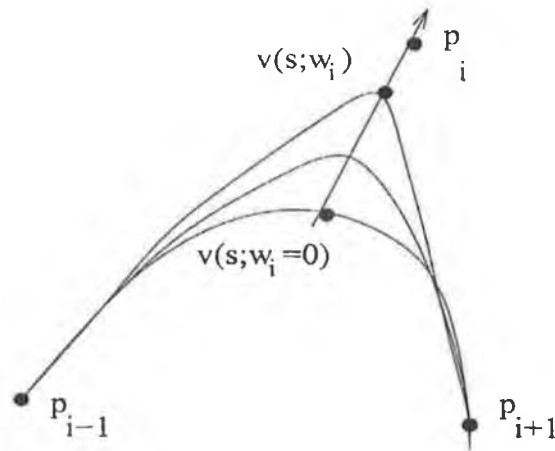


Figure 2.7: The movement of a point on the curve  $v(s)$  towards the desired control point as the associated weight is increased (Meegama & Rajapakse 2003).

## 2.6 Geometric Active Contour Model

Geometric active contours are based on the theory of curve evolution and level set methods. Curves evolve using only geometric measures <sup>17</sup>, resulting in a contour evolution independent of the curve's parameterisation, thus avoiding the need to re-parameterise the curve repeatedly or to explicitly handle topological changes (Caselles et al. 1997, Malladi et al. 1995). The parametric representations of the curves themselves are computed only after the evolution of the level set function is complete. Geometric active contours can be applied to model arbitrarily complex shapes, including shapes with significant protrusions and in situations where no *a priori* assumption about the objects boundary (topology) is made.

Geometric active contours have many advantages over parametric active contours, such as computational simplicity, the possibility to split and merge, the ability to change curve topology during evolution <sup>18</sup> and the ability to evolve in the presence of sharp corners in a seamless fashion. As previously described, parametric active contours are represented explicitly as parameterised curves. Now geometric active contours are introduced as level sets of two dimensional distance functions that evolve according to an Eulerian formulation.

The basis of the approach introduced by Malladi et al. (1995) is that it is not always possible to specify the topology of an object prior to its recovery. For example,

- this is an important issue in object tracking and motion detection applications where the topology can change rapidly depending on the position of the observer.
- When closed contours change their connectivity and split undergoing changes in topology.

<sup>17</sup>The geometric measurements are minimal distance curves (or geodesics) that are made in Riemannian space, whose metric is defined by the image content,

<sup>18</sup>It additionally removes the problems associated when points on the contour cross over each other causing the contour to kink.

## 2.6.1 Curve Evolution

### Level Set Methods

A level set starts with a given boundary separating one region from another and a speed that controls how each point on the interface can move. The speed can depend on a variety of physical effects. A level set approach does not track the motion of the individual points on the boundary, but takes the original curve and creates a cone-shaped surface, shown in Figure 2.8, that has the important property that it intersects the  $xy$  plane exactly where the curve sits. The surface on the right of Figure 2.8 is called the level set surface and the red front is called the *zero level set*.

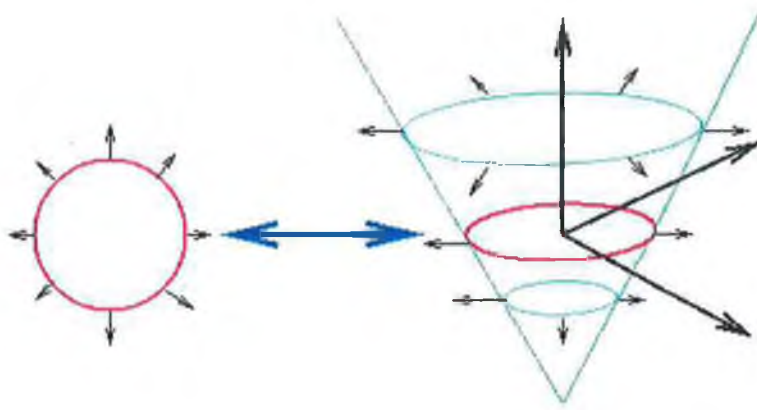


Figure 2.8: Illustration of the Level sets approach. (a) the original curve and (b) equivalent level set surface.

Fast Marching Methods provide an alternative means of evolution to the level set approach. They are designed for a problem set in which the sign of the speed function never changes, i.e. the front is always moving forward or backward. This converts the problem to a stationary formulation, because a front will only cross a point once. This can be used to speed up the evolution (Malladi & Sethian 1996). Level Set Methods are designed for problems in which the speed function can be positive in some places and negative in others, so that the front can move forwards in some places and backwards in others. While significantly slower than Fast Marching Methods, embedding the problem in one higher dimension gives the method tremendous generality.

### Contour Representation

Let  $\phi(\mathbf{x}, t)$  be a 2D scalar function whose zero level set defines the geometric active contour. In (Caselles et al. 1997, Malladi et al. 1995), the geometric active contour model evolves  $\phi$  using the following formulation:

$$\phi_t = c(\kappa + V_o)|\nabla\phi| \quad (2.33)$$

where  $\kappa$  is the curvature,  $V_o$  is a constant and

$$c \equiv c(\mathbf{x}) = \frac{1}{1 + |\nabla(G_\sigma(\mathbf{x} * I(\mathbf{x})))|} \quad (2.34)$$

is an edge potential derived from the image. According to Caselles (1995), the term  $c(\kappa + V_o)$  determines the overall speed of the level set of  $\phi(\mathbf{x}, t)$  along the direction normal to the curve. The value  $\kappa$  has the smoothing effect on the contour and  $V_o$  is the constant speed at which the contour either contracts or expands. The combination in Equation 2.33 combines internal and external constraints consistent with the parametric model and the external constraints have the effect of stopping the evolution in a particular direction.

In (Xu et al. 2000), this formulation works well with objects that have good contrast, because if high contrast is not guaranteed or if the boundary is incomplete, then the contour can pass over the boundary of the object. This problem is in part rectified by including an additional stopping term. Thus Equation 2.33 is reformulated as

$$\phi_t = c(\kappa + V_o)|\nabla\phi| + \nabla c \cdot \nabla\phi \quad (2.35)$$

This stopping term  $\nabla c \cdot \nabla\phi$  pulls the contour back if it passes the boundary. Although this additional term improves the boundary leaking problem, it does not provide a satisfactory solution. Several alternatives have been proposed by Caselles et al. (1997) and Kichenassamy et al. (1996).

### Boundary Detection

At an ideal edge,  $E_{ext}$  is expected to be zero since  $|\nabla I| = \infty$  and  $g(r) \rightarrow 0$  as  $r \rightarrow \infty$ , where  $g$  is an edge detector. Thus the goal is to send the edges to zeros of  $g$ . Ideally, it is also important to send the internal energy to zero. Since the images are not formed by ideal edges, the internal and external energy term are weighted identically and have an equal contribution to the minimisation process (Caselles et al. 1997). This is consistent with Fermat's Principle<sup>19</sup> and provides a link with between curve evolution active contours and this approach, i.e. minimising the geodesics measures is the same as minimising the energy in the active contour.

### Changing Topology

(Caselles et al. 1997) state that the classical energy of snakes can not directly deal with changes in topology. The topology of the initial curve will be the same as the possibly wrong final curve, i.e. errors in the initial parameterisation of the contour are propagated through the contours evolution.

## 2.6.2 Relationships between Parametric and Geometric Models

Similarities between the two active contours are apparent although the precise nature of these relationships is explored by Xu et al. (2000). (Xu et al. 2000) state that the majority of relationships between the two types of curves have been established by neglecting the rigid forces. In addition, it states that overall, the equivalences currently established in the literature do not relate a full family of parametric models to their geometric equivalent. As a result, it is difficult to design geometric active contours that take advantage of the wealth of parametric models that have been previously established. For example:

<sup>19</sup>In (Caselles et al. 1997) *Fermat's Principle* is states that: in an isotropic medium the path taken by light rays in passing from a point  $A$  to a point  $B$  are extrema corresponding to the traversal-time (as action). Such paths are geodesics with respect to the new metric  $(i, j = 1, 2)$ .

- it is not clear how one would incorporate non-conservative external forces, such as the forces defined in (Xu & Prince 1998a),
- it is not clear how to incorporate regional pressure forces.

It has been well documented that the use of elastic internal forces may cause an undesirable shrinking effect, whereas the use of rigid internal forces can smooth the contour without this adverse result with a relative low overhead (Williams & Shah 1992).

Although Xu & Prince (1998a), show that it is possible to find equivalences for the majority of external forces that are applied to parametric active contours, it is not apparent how external spring forces or variable tension and rigidity can be defined in the geometric representations.

## 2.7 Templates

Machine vision systems attempt to create internal models of the processed scene and update them using an appropriate sequence of processing steps that must be performed to achieve a given result for a particular task (Sonka et al. 1999). The attraction of using prior knowledge in machine vision is that it is hard to make progress without it. The use of prior information moves the task of interpretation from general to goal-directed processes (Blake & Isard 1998). Methods for fusing prior knowledge with observations are of crucial importance and any difference between the observation and the predicted shape can be classified as an error. The prior assumptions can be varied, by adjusting the elastic parameters of the template. In addition, more specific models can be constructed using flexible curves in which the parameters, such as kinematic variables, sizes of subparts and the angle-hinges which join them can be adjusted either interactively or automatically. A model that permits such deformation is known as deformable template and is a powerful tool in analysing images.

Often the smoothness and the constraints introduced in the original snake model are not sufficient to encourage the snake to converge, and prior knowledge needs to be introduced to the snake model to achieve stable behaviour. If all the control points are allowed to vary somewhat freely overtime, the tracked curve can rapidly tie itself into unrecoverable knots.

### 2.7.1 Deformable Templates

Inclusion of hard constraints in the default template can be achieved by using a parametric shape-model  $r(s; \mathbf{X})$  with relatively few degrees of freedom. This is known as a deformable template. The template is matched to the image in a manner similar to snakes by searching for the parameter vector  $\mathbf{X}$  that minimise the external energy. The internal energy is included as a regularisation term (Blake & Isard 1998). The internal energy  $E_{int}(\mathbf{X})$  contains a quadratic function of  $\mathbf{X}$  that encourages the template to relax back to a default shape. The external energy  $E_{ext}(\mathbf{X})$  comprises the sum of various integrals over image-feature maps.

Spline snakes are a common method of representing curves used in templates. These are a smooth curve between a set of control points, and the shape of the curve is completely defined by the knot positions. B-splines are the most common implementation of splines. A description of B-splines can be found in Appendix A. The use of B-splines enables the calculation of the bending



energy analytically and it is simple to sample the curve at multiple points where the external energy can be calculated and used to optimise and modify the position of the control points. In this section, the curves are expressed uniquely as B-splines. The notation that is used is consistent with that introduced by Blake & Isard (1998).

### The Control Vector

In reality, dealing with control points is not the most convenient method for representing B-spline data, and in (Blake & Isard 1998), control vectors  $\mathbf{Q}$  are introduced as a first stage towards defining shape-spaces. The control vector  $\mathbf{Q}$  consists of the control point coordinates. Thus for a parametric spline curve  $\mathbf{r}(s) = (x(s), y(s))$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix} \quad \text{where} \quad \mathbf{Q}^x = \begin{pmatrix} q_0^x \\ \dots \\ \dots \\ q_{N_B-1}^x \end{pmatrix} \quad \text{and} \quad \mathbf{Q}^y = \begin{pmatrix} q_0^y \\ \dots \\ \dots \\ q_{N_B-1}^y \end{pmatrix} \quad (2.36)$$

The coordinate functions can be expressed as  $x(s) = \mathbf{B}^T(s)\mathbf{Q}^x$  and  $y(s) = \mathbf{B}^T(s)\mathbf{Q}^y$  where  $\mathbf{B}(s)$  is a vector B-spline basis function such that

$$\mathbf{r}(s) = U(s)\mathbf{Q} \quad \text{for} \quad 0 \leq s \leq L \quad (2.37)$$

where

$$U(s) = I_2 \otimes \mathbf{B}^T(s) = \begin{pmatrix} \mathbf{B}^T(s) & 0 \\ 0 & \mathbf{B}^T(s) \end{pmatrix} \quad (2.38)$$

which is a matrix of size  $2 \times 2N_Q$  and  $I_M$  denotes an  $m \times m$  matrix.

The properties such as norm and inner product can be redefined in terms of the control vectors and a full derivation of how this is achieved is found in (Blake & Isard 1998).

### Shape-Space Models

Shape-space was introduced by Blake & Isard (1998) as a means to reduce the shape variability. A distinction is made between the spline-vector  $\mathbf{Q} \in \mathcal{S}_Q$  that describes the basic shape of an object, introduced above, and the shape vector  $\mathbf{X} \in \mathcal{S}$ , where  $\mathcal{S}$  is a shape space while  $\mathcal{S}_Q$  is a vector space of B-splines<sup>20</sup>. The Shape-space is defined as:

*Shape-space is the linear parameterisation of the set of allowable deformations of a base curve.*

The requirement that the shape space be linear is made to ensure simplicity in terms of computation. In mathematical terms, the shape space  $\mathcal{S} = \mathcal{L}(W, \mathbf{Q}_0)$  is a linear mapping of a shape-space

<sup>20</sup>In general, the shape-space has smaller dimension than the B-spline vector space. The dimension of the shape space and the B-spline vector space are denoted as  $N_X$  and  $N_Q$  respectively.

vector  $\mathbf{X} \in \mathbb{R}^{N_X}$  to a spline vector  $\mathbf{Q} \in \mathbb{R}^{N_Q}$

$$\mathbf{Q} = W\mathbf{X} + \mathbf{Q}_0 \quad (2.39)$$

where  $W$  is a  $N_Q \times N_X$  shape matrix. The constant offset  $\mathbf{Q}_0$  is a template curve against which shape variations are measured. The matrix  $W$  is composed of columns which are the vectors of the basis of the shape-space. An example of a shape-space is the space of Euclidean similarities with four degrees of freedom or the planer affine group of transforms with six degrees of freedom. Other transform groups do not readily form shape-spaces and it is not always possible to represent the shape-space in minimum dimensions. Euclidean similarity and Affine Transforms, are shape-spaces because there exists an  $\mathbf{X}$  for which  $\mathbf{Q} = W\mathbf{X}$

For example consider the *Euclidean similarities* (Blake & Isard 1998). Give a template curve  $r_0(s)$  represented by  $\mathbf{Q}_0$  form a shape space,  $\mathcal{S}$ , with shape matrix

$$W = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{Q}_0^x & -\mathbf{Q}_0^y \\ \mathbf{0} & \mathbf{1} & \mathbf{Q}_0^y & \mathbf{Q}_0^x \end{pmatrix} \quad (2.40)$$

The first two columns control the horizontal and vertical translations and the third and fourth columns cover rotation and scaling. Some Examples of shape represented in the space of Euclidean similarities are:

- $\mathbf{X} = (0, 0, 0, 0)^T$  represents the original template shape  $\mathbf{Q}_0$
- $\mathbf{X} = (1, 0, 0, 0)^T$  represents the template translated 1 unit to the right
- $\mathbf{X} = (0, 0, 1, 0)^T$  represents the template doubled in size
- $\mathbf{X} = (0, 0, \cos \theta - 1, \sin \theta)^T$  represents the template rotated through an angle  $\theta$

The properties such as norm and inner product can be redefined in terms of the shape-space and a full derivation of how this is achieved is found in (Blake & Isard 1998) where the centroid and additional moments are also redefined.

In the context of the problem of curve fitting, shape-space is combined with an energy landscape to encourage the curve to approximate the image-feature curve. This acts as a way of encouraging smoothness and reducing the shape variability. Suppose the image features were expressed in the form of a spline curve  $r_f$  where  $r_f(s) = U(s)\mathbf{Q}_f$ . If the fitted spline is restricted to shape space and using an edge-based energy landscape the fitting problem then becomes a problem of

$$\min_x \|\mathbf{W}\mathbf{X} + \mathbf{Q}_0 - \mathbf{Q}_f\|^2 \quad (2.41)$$

The solution  $\mathbf{X} = \hat{\mathbf{X}}$  is given by  $\hat{\mathbf{X}} = W^+(\mathbf{Q}_f - \mathbf{Q}_0)$  where  $W^+$  is the pseudo-inverse of  $W$ . This effectively is an expression in shape space that minimises the difference between the template,  $\mathbf{Q}_0$ , and the image curve,  $\mathbf{Q}_f$ , that it is fitted to. The fitting can be improved by biasing the curve towards the mean shape.

### 2.7.2 Active Shape Models

In this approach, a shape is represented as a set of points and training sets of samples are examined to determine the average position of the shape points. This approach has many advantages over rigid models when objects of the same class are not identical. In such cases, flexible models, or deformable templates, can be used to allow for some degree of variability in the shape of the imaged object (Cootes et al. 1992).

Cootes et al. (1992) describe a method of shape modelling based on the statistics of labelled points placed on a set of “training” examples. This model consists of the mean positions of the points and a number of vectors describing the modes of variation. In this approach, the labelling of the points is important as each labelled point represents a particular part of the object or its boundary. To ensure that the points are correctly labelled, it generally requires someone familiar with the model to place the points.

To create the mean statistics for the shapes, it is essential that the training shapes are aligned. This is achieved by scaling, rotating and translating the training shapes so that they correspond as closely as possible. Each of the points are weighted and in general, the weights are used to give more significance to those points that tend to be most stable over the set of training sets, i.e. the points that move least with respect to the other points on the shape. This ensures that minimisation process will be dictated by the points that move least and reduce the influence of stray points.

Once the shapes are aligned, the mean shape and variability are calculated. This enables the modes of variation, the ways which the points tend to move together, to be found. These are established by applying principal component analysis to the deviations from the mean. For each point,  $1, \dots, n$ , its deviation from the normal is calculated and its covariance is presented in a  $2n \times 2n$  matrix. The modes of variation correspond to the eigenvectors of the covariance matrix. The largest eigenvalues describe the most significant modes of variations of the elements used to calculate the covariance matrix. The majority of the variation can usually be described by a small number of modes (Cootes et al. 1995) and any shape in the training set can be approximated using the mean shape and a weighted sum of the deviations

The points on the mean model do not have to lie on the boundary of object as they can represent internal features or sub-components. The models are linear, like those described in 2.7.1. This method is inefficient at modelling bending or rotation of one subcomponent about another.

### Learning Deformable Models for Tracking Human Motion

The task of surveillance involves observing a scene for a considerable length of time. In the situation that a single camera, monitors a scene, it is possible to extract dynamic information from an image sequence. Baumberg (1995) present such an approach that attempts to extract human motion. The difference observed over a series of frames is thresholded. It is important that the thresholding is based on the expected size of the object in the image. This enables the identification of the silhouette of the object of interest. This difference is then blurred and thresholded to reduce the effects of noise. This enables the extraction of the individual’s silhouette from the image sequence.

In (Baumberg & Hogg 1994), the exact shape of the individual is not sought but a reasonably

consistent shape vector is extracted. The boundary of the silhouette is described by a B-spline contour that interpolates 40 uniformly distributed control points. The B-spline contour is fitted to the silhouette providing a reasonably close approximation. The control points are treated in the same manner as described in (Cootes et al. 1992). Tracking the movement of regions from one frame to the next enables a direction for motion to be extracted. This information is then associated to each control point. Using this model, it is possible to infer 2D direction of the motion of a person, in terms of image coordinates.

### 2.7.3 Tracking using Active Contour Models

When the scene, or objects within it, start to move, the task of locating the moving objects becomes more difficult. This problem is known as tracking and can be solved by introducing dynamic elements in the active contour model. In (Blake & Isard 1998), the dynamic models are referred to as “dynamic contours”. Active Contours can be applied dynamically to temporal image sequences. In dynamic applications, an additional layer of modelling is required to convey any prior knowledge about object motions and deformations. For this purpose, the active contours are redefined as time and space varying curves and terms to account for inertia and viscosity. This is expressed mathematically as:

$$\underbrace{\rho r_{tt}}_{\text{inertial force}} = - \underbrace{\left( \gamma r_t - \frac{\partial(w_1 r)}{\partial s} + \frac{\partial^2(w_2 r)}{\partial s^2} \right)}_{\text{internal forces}} + \underbrace{\nabla F}_{\text{external force}} \quad (2.42)$$

where  $\rho$  is the mass density and  $\gamma$  is the viscous resistance from the medium surrounding the snake. In this formulation, there is a large degree of freedom and without imposing prior knowledge the snake can potentially move and deform to any shape.

When tracking movements between frames, it is not sufficient to compare the previous position of the curve with its current position. This is because the discretisation caused the curve to undergo small perturbations around the equilibrium points. However, tracking the position of the curve for a few iterations distinguishes between perturbations around the equilibrium, and motion towards equilibrium by the curve.

## 2.8 Active-meshes

ACMs have been reformulated as active-meshes that are used in unconstrained environments to perform tracking of objects in an image sequence (Molloy & Whelan 2000). The approach uses meshes to track strong features in images. An example is shown in Figure 2.9 in which an ambulance is tracked. The strong features, such as edges and corners, are used as vertices of a mesh. Setting the constraints involves generating internal forces between the mesh vertices. In (Molloy & Whelan 2000), the forces applied to each node are proportional to the distance between the current length of the interconnecting lines and the reference length giving the mesh its elastic properties. Additional rigid forces are applied to encourage the distance between vertices to return to the reference distance which can expand slowly over a number of iterations if it is subjected to a consistent external force.

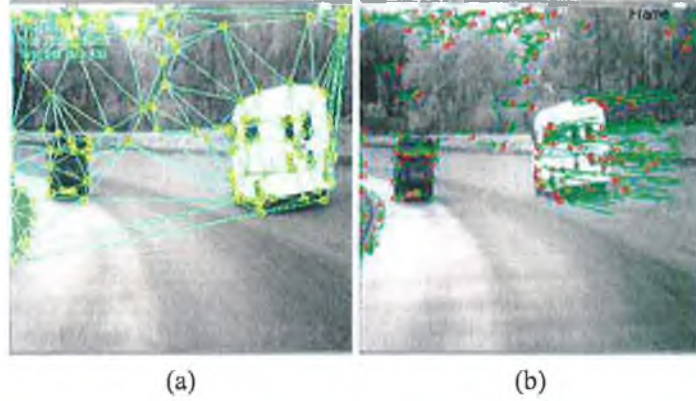


Figure 2.9: Example of the application of active-meshes. (a) shows the generated mesh that is used to track the position of the ambulance in an image sequence and (b) shows the motion vectors that are generated by the ambulance (Molloy 2000).

### The Internal Forces

The mesh structure is generated using a Delaunay triangulation that connects the mesh points to their natural neighbours. The mesh lines have elastic properties that enable the mesh to deform and track the movement of the mesh points. The internal forces are proportional to the difference between the current length and the reference length of the interconnecting line. The rigid properties of the mesh attempt to force the mesh lines to return to their reference length while the elastic properties give the mesh the flexibility to track points. The reference length is not fixed and can change and adopt a new length if the mesh is stretched by a number of consistent external forces over a significant number of iterations. The internal energy is expressed as:

$$\vec{F}_{Line} = L_{cur}(x)\beta_L\vec{i}_x + L_{cur}(y)\beta_L\vec{i}_y \quad (2.43)$$

where mesh lines have a current length ( $L_{cur}$ ), a set length ( $L_{set}$ ) and  $\beta_L = (L_{set} - L_{cur}) / \alpha_L L_{cur}$ .  $L_{cur}(x)$  and  $L_{cur}(y)$  represent the x and y components of the current mesh line lengths which determine the internal energy,  $\vec{i}$  is a unit vector with same axis as the mesh model,  $\alpha_L$  is a user defined factor to limit the effect of forces.

At each iteration  $L_{set} = L_{set} + \alpha_I(L_{cur} - L_{set})$  where  $\alpha_I$  is a user defined factor that limits the change in length of the line length.

### The External Forces

The External forces are applied to the mesh nodes independent of the mesh lines and are derived from image data. The image forces pull the mesh nodes towards image features points. The forces are determined by the Euclidean distance between the mesh node and the location of the feature and a scale factor that is determined by suitability of the match feature.

The most suitable match is found by comparing the  $3 \times 3$  area surrounding the current node and the  $3 \times 3$  area surrounding the possible match corners detected within a circular search space with a predefined radius,  $r$ . Then for all of the corner matches with Euclidean distance  $d < r$ , to

the current node are considered. The total intensity difference is

$$I_T = \sum_{i,j=-1}^1 |I(x_{n_o} + i + y_{n_o} + j) - I(x_{c_n} + i + y_{c_n} + j)| \quad (2.44)$$

where  $n_o$  is the current node and  $c_n$  is a corner within the radius  $r$ . The corner point is chosen that minimises the value of  $I_T$ . It is subsequently normalised to establish a match strength,  $S_M$ . The indices  $i$  and  $j$  are the horizontal and vertical coordinates in the circular search space.

Thus for a single node the external force is given by:

$$\vec{F}_{ext} = \beta_{ext} d(x) \vec{i}_x + \beta_{ext} d(y) \vec{i}_y \quad (2.45)$$

where  $\beta_{ext} = \alpha_E (r - d/r)$  and  $\alpha_E$  is a user defined property that determines the significance of the force that the external forces have over the mesh.

### Force Combination

These forces are combined using a weighting factor that is inversely dependent on the Euclidean distance separating two connected nodes and is expressed as:

$$\beta_i = 1 - D_i / \left( \sum_{i=1}^N D_i \right) \quad (2.46)$$

Thus the force on the centre node is

$$\vec{F}_0 = \sum_{i=1}^N \beta_i \vec{F}_i \quad (2.47)$$

## 2.9 Discussion

This section provides a review of previous active contour models. In particular, it has been shown that active contours have undergone and continue to undergo change since their inception; in some cases, to overcome apparent weaknesses<sup>21</sup> in the original implementation and in others, the reformulation is necessary to enable the active contours to be applied in new domains. Although geometric active contours, as described in this chapter, are perhaps fundamentally separated from the original active contour model, they provide enhancements that are not easily integrated into the original model while the development of deformable templates based on active contour models is potentially the principal technique to enable higher-level information to be brought to bear on the task of extracting known objects from an image.

The section also draws attention to the semblance of the energy minimization framework in different forms, including the active mesh formulation discussed in Section 2.8 and the incorporation of additional constraints in the framework to solve specific problems in the domain of medical imaging. It also details different reformulations in terms of B-splines and NURBS, both of which

---

<sup>21</sup>The weaknesses are not related to the algorithmic methodology of active contours but how it can be applied to specific problems or how the process can be automated

are key to application of snakes in real-time applications and can provide greater local control over the positioning of the final curve.

This review provided a vital step in deciding how the active contours can be used for the extraction of an individual from their environment. It follows from an examination of existing techniques for the identification and extraction of objects from images. Moreover, this review has highlighted and confirmed that implicit (geometric) models are best suited for situations in the recovery of unknown topologies and extraction of complex shapes, although they are not as convenient as parametric models for known shape and visual presentation of intermediate stages or for user interaction.

The following reasons are considered as validation in the choice of parametric active contour models for the extraction of individuals from real environments:

- In (Jacob et al. 2004), it is shown that since the parametric snakes represent the curve explicitly, it is easy to introduce *a priori* shape constraints into the snake framework. This is important to enable the incorporation of shape information on the structure of the contour. To date, it has not been possible to introduce constraints in the geometric active contour framework that can dictate how the contour should evolve as this is entirely dependent on parameterisation of the contour and the image forces.
- McNerney & Trezopoulos (1995) highlight that parametric snakes as opposed to geometric snakes introduced by Caselles et al. (1997) and Malladi et al. (1995) are well suited when the topology of the object is fixed and known *a priori*. Parametric active contours are best as they provide greater control and enable the contour to deform in a pre-described manner.
- In addition, to incorporate the flexibility to adapt to any topology, geometric active contours tend to be computationally more complex as they evolve a surface as opposed to a curve.
- The final position of a geometric contour does not provide an easily interpretable shape. If a contour description is required, it is necessary to fit a curve to describe the extracted shape. Moreover, during the iterative process comparisons, with exiting shapes is difficult. This is in contrast to the parametric methodology that can be easily examined at any iteration.

Parametric definitions are used because a pixel free representation is sought which is necessary to develop a template that can be easily adapted to the extraction of any individual. Additionally, the use of parametric contours is considered appropriate since:

- It is important to take advantage of the fact that the shape of the object to be extracted is known in advance.
- It is necessary to introduce constraints to force the contour to move in the desired fashion, and while constraints cannot be incorporated in the approach proposed by Kass et al. (1987), they have been successfully introduced under the auspices of dynamic programming by Amini et al. (1988).
- The boundary of the individual will have different characteristics, for example the texture or colour of the clothing the individual wears. Thus if a level sets approach is employed,

the contour will have to evolve over multiple boundaries within the image to successfully extract the individual as a single region.



# Building Virtual Humans

## 3.1 Introduction

The creation of virtual human models that are used to populate virtual worlds is a diverse area which has received significant interest in recent years. This has led to the development of various techniques for the creation of realistic human models that are used in numerous applications over an increasingly wide range of devices. These applications have made it possible for individuals to interact in new mixed and virtual reality applications including video conferencing (Weik et al. 2000) (Kompatsiaris et al. 1998) (Wingbermhle et al. 1997), virtual worlds (Prince et al. 2002) and network games. The virtual humans are also used in diverse applications from virtual tourism (Papagiannakis et al. 2004) to fashion (Cordier et al. 2003) and sports (Klein et al. 2002). Additionally, new applications are being continually devised to take advantage of the 3D human models (Sobreviela et al. 2000).

All of these techniques have different requirements in terms of the quality of the final content and how this content should be delivered (in real-time or offline), and the specification of the terminal device. In general, the techniques that are used for the creation of the models depend on the type of applications that the models are created for. In computer games, motion capture systems are the preferred method for data capture because of the real-time constraints giving interaction and animation a higher precedence over the fine detail, while in the creation of characters for films the quality of the model is paramount for successful integration of the model and thus a combination of technologies including the use of range scanners, or multiple images, and often additional sculpting is required to generate the final character. In web based and increasingly mobile applications, different Level of Detail (LOD) models are being produced to take advantage of available bandwidth (Collins & Hilton 2001).

This chapter starts by providing a review of various approaches that exist for the 3D reconstruction of rigid objects and scenes. The different approaches that are described are assessed in terms of their suitability for use by non-expert users and their practicality in reconstructing non-static objects such as humans. This is supplemented by a review of the existing techniques that have been applied to the creation of “Virtual Humans”. This includes a description of the techniques for the estimation of pose and shape information which build on the Active Shape Models

and the templates that are described in Chapter 2. Finally, this chapter concludes by highlighting the aspects that are deemed important by the author in the development of a flexible approach to the creation of human models.

## 3.2 General Approaches to 3D Reconstruction

Early computer vision aimed at understanding how explicit geometric representations of a 3D world may be reconstructed from 2D images. There are a number of aspects related to this problem of representation and they resulted in a number of different reconstruction techniques based on: intensity gradient and flow fields, reconstruction of 3D surface depth and orientation and motion fields (Trezopoulos 1998). Under Marr's paradigm, 3D vision is formulated as the 3D reconstruction of an object from an image (or a series of images) of a scene. The first two stages of the vision process involve converting an image to a primal sketch and converting that primal sketch to a  $2\frac{1}{2}D$  sketch. This is followed by the conversion of the sketch to a 3D model. This involves the extraction of 3D geometric description of the scene and the quantitative determination of the properties of the object in the scene.

The task of 3D reconstruction involves solving three interrelated problems:

1. Feature visibility in images: i.e. how to determine whether or not the relevant object is contained in the image (even partially contained in the image).
2. Representation: this relates to the choice of model for the observed world, at various levels of complexity.
3. Interpretation: this covers how the data is mapped to the 3D (real) world (Sonka et al. 1999).

These problems can be solved in either a bottom-up or top-down (model based) approach based on the amount of prior information available:

1. **Reconstruction, bottom-up:** This aims at reconstructing 3D shape of an object from an image or set of images when very little *a priori* information is available. In this approach, the idea is to create a 3D model from real world objects,
2. **Recognition, top-down, model based vision:** The available *a priori* information about the objects is expressed by means of models (templates), where 3D models are of particular interest. The inclusion of additional constraints in the model makes it possible to infer data that is not available in an underdetermined vision task.

In general, 3D Reconstruction is a difficult task for a number of reasons including:

- The fact that 3D reconstruction is an ill-posed problem that requires the extrapolation from less dense 2D information to a richer 3D domain.
- The extraction of correspondences between images which form the basis for 3D geometrical properties are based on image intensity values that are subject to variation.
- The existence of occlusion and the presence of noise in the images.

### 3.2.1 The Eight-point Algorithm

The premise for 3D reconstruction of scenes and objects from two images using calibrated cameras is presented by Longuet-Higgins (1981), in which it is shown that the essential matrix<sup>1</sup> contains sufficient information to compute the structure of a scene if eight point matches are known. The solution is obtained from solving a set of simultaneous linear equations. This approach does not detail how the eight-points should be determined although it highlights degenerate configurations that will cause the algorithm to fail. These include the following configurations:

- if as many as four points lie in a straight line,
- if as many as seven of them lie in a plane,
- if six points lie at the vertices of a regular hexagon,
- or if 8 points are located at the vertices of a cube.

Thus in general, it is necessary to calculate more than eight-point correspondences between the images and then use a subset of all points. Hartley (1997) applies the same algorithm to compute the fundamental matrix from images captured with uncalibrated cameras. The fundamental matrix can be used to reconstruct a scene but only up to a projective transformation.

It has long been established that it is possible to extract the structure of a scene with the establishment of five point correspondences, but this results in a set of non linear equations and involves an iterative solution. This work has been improved by Roach & Aggrawal (1979) and by Faugeras & Maybank (1990).

The calculation of the points of correspondence can be achieved by using a number of well known approaches including point matching, line matching and corner detection, all of which are important in tracking methods and in the recovery of 3D (Sonka et al. 1999).

It is necessary that a series of checks needs to be carried out on the points to ensure that no degenerate configurations exist. In addition, to establish points of correspondence necessary for a complete 3D reconstruction of an object requires numerous images captured close together. This provides a high correlation between the images. All points of correspondence will not be available in each image (Dyer 2001). This increases the complexity of the reconstruction process and, in general, requires an expert user to ensure that sufficient information is captured in each view. Alternatively, if multiple cameras are used, then it is imperative that the cameras are calibrated or that correspondences between the images can be obtained to ensure that accurate reconstruction can be completed. Many different partial models may result, which must be combined to form a single consistent model, and there is no way of handling occlusions or differences between views (Dyer 2001).

---

<sup>1</sup>The essential matrix conveniently encapsulates the epipolar geometry of the imaging configuration. The same algorithm may be used to compute a matrix with this property from uncalibrated cameras. In this case of uncalibrated cameras, it has become customary to refer to the matrix so derived as the fundamental matrix. See (Hartley & Zisserman 2000) for more details.

### 3.2.2 Silhouette Based Reconstruction

Silhouette based techniques provide an alternative approach to reconstruct objects from images when it is not possible to extract sufficient points of correspondence. Although silhouette based approaches are unable to reconstruct non-convex (concave) parts of surfaces, there exist numerous methods for construction of models from a set of silhouette images. A comprehensive list is described in (Dyer 2001) and the references therein.

A silhouette is a 2D projection of an object from a particular viewpoint. In image terms, the silhouette is the part of the image that contains the projection of the object as opposed to the background. In general, it is possible to recognise a convex object from its silhouette. The basis for examining volume intersection, as a method of 3D reconstruction, is that the silhouettes can be simply and reliably obtained from intensity images, and it is not necessary to find multiple correspondences between all images. The only requirement is that the positions of the viewpoints are known (Laurentini 1994). The basis idea of silhouette based reconstruction is shown in Figure 3.1.

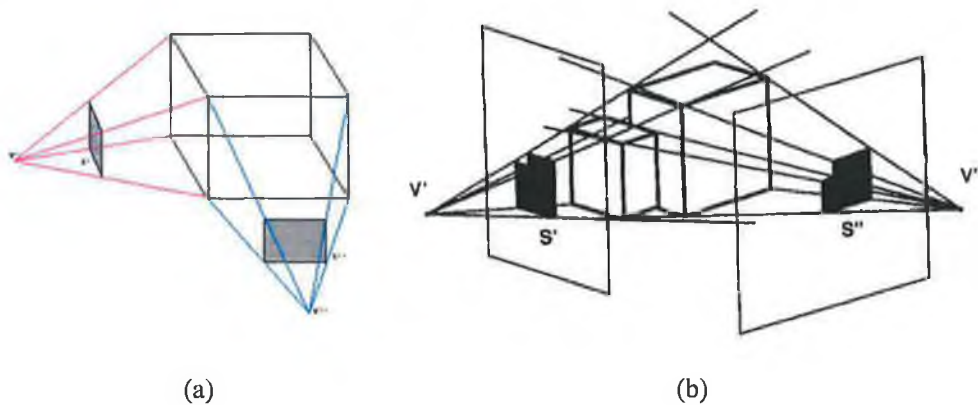


Figure 3.1: Volume intersection approach to reconstruction. (a) shows a simple shape reconstructed from two silhouettes. (b) shows a situation when the true shape of the object can not be reconstructed (Laurentini 1994).

Although silhouette based reconstruction is proposed as a method for the construction of 3D objects, it is not possible to construct every convex object from a series of silhouettes and, in general, the closest approximation that can be obtained is called the “visual hull” of the object. Only objects coincident with the visual hull, can be reconstructed. The visual hull as introduced by Laurentini (1994), is defined as:

*Definition 1:* The visual hull  $VH(S, R)$  of an object  $S$  relative to a viewing region  $R$  is a region of  $E^3$  such that, for each point  $P \in (S, R)$  and each viewpoint  $V \in R$ , the half-line starting at  $V$  and passing through  $P$  contains at least a point of  $S$ .

This highlights an important role that the viewing region  $R$  plays in the reconstruction of an object. Laurentini (1994) defines the terms silhouette-active surface and inactive surface are defined. The silhouette-inactive surface can take on any shape without affecting the silhouette of the object while any point on the silhouette-active surface is part of the surface of the object and on

the boundary of  $VH(S, R)$ . Following on from this, Laurentini states that there is a unique visual hull not exceeding the convex hull of  $S$ , relative to all viewing regions that enclose  $S$  and do not enter the convex hull. Thus any point  $Q$  belonging to  $VH(S, R')$  also belongs to  $VH(S, R'')$  and vice versa, since a half-line from a particular point on the object will intersect the viewing regions  $R'$  and  $R''$ . As the number of viewpoints is increased, the object can be reconstructed with greater accuracy (higher precision), although if the viewpoints do not contain sufficient variation then it is impossible to accurately extract the 3D shape information.

The visual hull can be computed based on the simple observation that a point does not belong to the visual hull if there are lines passing through this point that do not intersect the object since its image cannot be found in all silhouettes of the object. This is illustrated in Figure 3.2 The visual hull for 2D scenes is equal to the convex hull of the object, and in 3D scenes it is contained in the convex hull (Dyer 2001). The visual hull is illustrated in Figure 3.2.

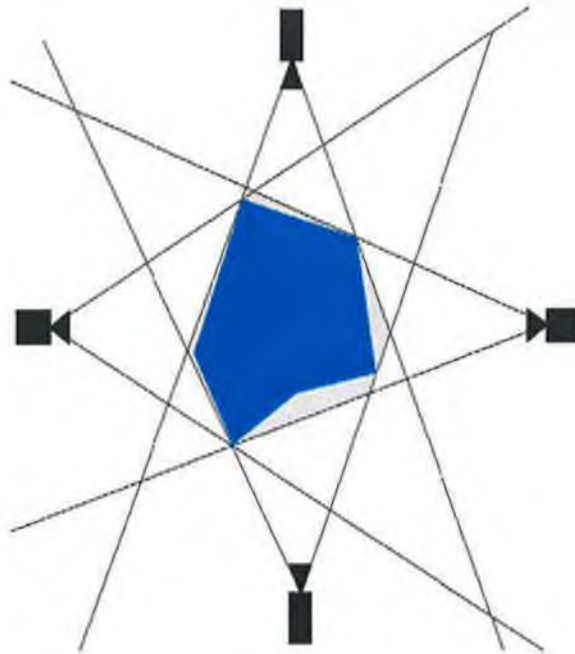


Figure 3.2: Illustration of an object's visual hull from 4 views<sup>4</sup>.

A 2D silhouette of a 3D object constrains the object inside the volume obtained by back projecting the silhouette from the viewpoint and a number of silhouettes specify a bounding volume that is created by the intersection of the silhouettes. This technique is known as volume intersection (VI) and it recovers the closest volumetric description of the object that can be recovered using silhouette-based reconstruction. The reconstructed object provides information about the external boundary of the object, but this does not provide complete shape information.

To determine which parts of the visual hull are coincident with the surface of the original object and those parts that may contain un-reconstructable concavities, it is necessary to consider each point in the visual hull. A line passing through any point of the visual hull must share at least one point with the object. Any point in the visual hull with at least one line that passes through the point without intersecting the visual hull at any other point is defined as a surface point. Further to

this, only edges with this property can be reconstructed. This enables the establishment of points that play an important role in estimating the shape of the object. Cheung et al. (2003) extend this with the concept of bounding edges in. A bounding edge is defined as:

*Definition 2:* A bounding edge  $E_j^i$  is the portion of a ray  $r_j^i$  such that the projection of  $E_j^i$  on the image planes of all other cameras lies completely inside the silhouettes<sup>2</sup>.

While staying within the bounds of the silhouettes, it is impossible to accurately extract all the finer details on areas such as the face, and thus the concavity or so called “complementary part” of the surface can take on different shapes without affecting the silhouette. This provides a basis for the use of an underlying model that can be scaled based on the information in the extracted silhouettes and then provides the necessary constraints for the reconstruction of the finer details.

Laurentini (1997) discusses the necessary considerations that should be considered in the reconstruction of objects from silhouettes. In addition to this, the theoretical minimum number of silhouettes required to reconstruct an object is examined. The result of the discussion is that a curved patch cannot be reconstructed from a finite number of intersections and that a concave polyhedral with  $n$  faces and viewpoints outside the convex hull may require an unbounded number of intersections to be successfully reconstructed.

### 3.2.3 Volumetric (Scene) Reconstruction

The construction of volumes or surfaces that are consistent with the input images is an alternative approach to traditional correspondence based methods for scene reconstruction and is based on computations in three-dimensional scene space. Volumetric methods offer flexible visibility models and explicit handling of occlusions. The space in which the scene occurs is represented through a discretised volume of voxels and occupancy decisions are made about whether a volumetric element belongs to the objects in the scene. Figure 3.3 shows a simple 2D example to determine which parts of the scene are outside the visual hull of the object and those that are inside the visual hull. In Figure 3.3 (b), the squares that are filled indicate that the complete pixel is inside the visual hull of the object.

There are a wide variety of methods for the construction of volumetric models, including the construction from a set of silhouette images discussed in (Dyer 2001) and the references therein. A volumetric scene is modelled explicitly in a world coordinate frame and the volume of space in which the scene resides. Volumetric modelling of scene space assumes there is a known, bounded area in which the objects of interest lie. This area is frequently assumed to be a cube surrounding the scene. A method for establishing this volume is presented in (Martin & Aggarwal 1983). The most common approach to representing this volume is as a regular tessellation of cubes, called voxels, in Euclidean 3D-space. Octrees<sup>3</sup> can be used to make the implementation more efficient (Srivastava & Ahuja 1990). Octree can be used in Figure 3.3 (b) to subdivide the voxels on the

<sup>2</sup>Assume that there are  $K$  cameras and that  $u_j^k$  is a point on the boundary silhouette in view  $k$ . Then  $r_j^k$  is the ray through camera centre  $k$  passing through  $u_j^k$ .

<sup>3</sup>Octrees are a hierarchical variation on the spatial occupancy enumeration designed to address the demanding storage requirements of dense 3D data. An octree is obtained by successfully subdividing a volume into eight equal octants along each 3D axis.

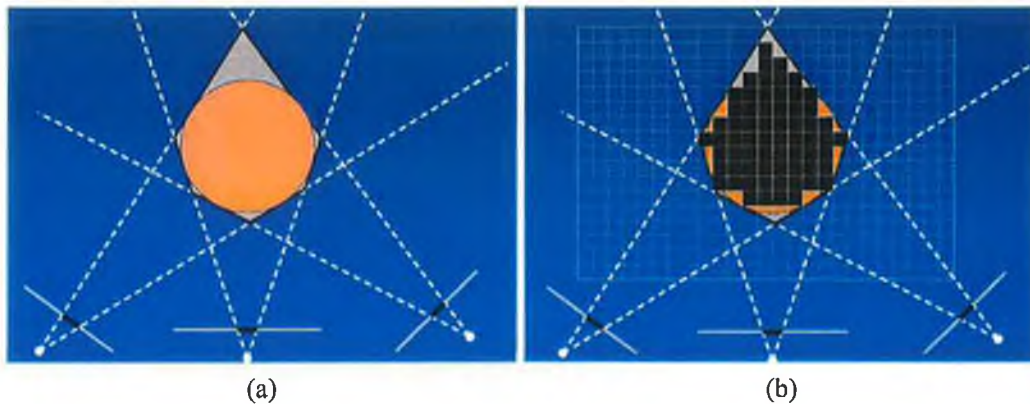


Figure 3.3: Illustration of how simple decision are made to decide if a voxel is inside the visual hull. (a) shows a sphere within the field of view of three cameras. The intersection of lines indicating the field of view of each camera forms the visual hull of the object. (b) shows a coarse reconstruction of the object the shaded squares indicate the voxels inside the visual hull of the object<sup>4</sup>.

boundary of the object's visual hull that are not filled to determine if part of pixel is contained within the visual hull or not.

Many algorithms have been developed for constructing volumetric models from a set of silhouette images, including (Martin & Aggarwal 1983) which attempts to derive a 3D object description from images that do not depend entirely on feature point correspondences. If the visual hull is not calculated accurately, the photo-realism of the scene will be significantly reduced when new views are generated. To increase the accuracy, more information from silhouettes can be used during the reconstruction. The main source of such information is colour (Slabaugh et al. 2001).

In volumetric reconstruction, it is necessary to start from a bounding volume that encloses the entire scene. This volume is then discretised into voxels and a voxel occupancy description is defined based on the intersection of the back projected silhouette cones. The intersection test is the most important task in the voxel-based algorithms. This is achieved in a number of different ways:

- In (Noborio et al. 1988), the silhouettes are back projected, producing a set of cones that intersect in 3D.
- In (Srivastava & Ahuja 1990), the intersection detection is achieved efficiently by decomposing it into a coarse-to-fine sequence of intersection tests
- In (Szeliski 1993), the intersection is determined by projecting each voxel into all of the images and seeing if it is contained in all of the silhouettes. This process is illustrated in Figure 3.4, in part (a) the voxel does not correspond to the object in the tree images and in part (b) the voxel projects to the object in each image.

To increase the efficiency of the voxel testing procedure, most methods use octree representation and implement a coarse-to-fine hierarchy (Dyer 2001, Srivastava & Ahuja 1990). This hi-

<sup>4</sup>From S. Seitz presentation 'From Images to Voxels', SIGGRAPH 2000 Course on 3D Photography.

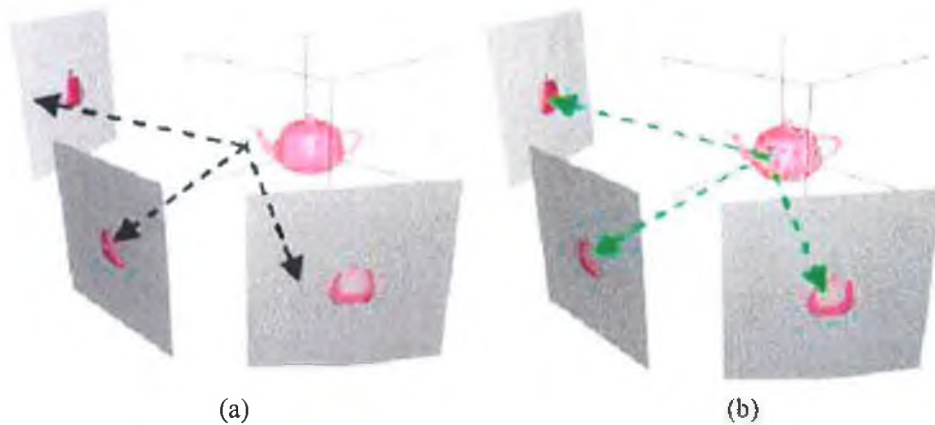


Figure 3.4: (a) the voxel projects in two views to background. (b) the voxel projects to the same color in all three views (Brisc 2004).

erarchy enables the volumetric construction of scenes and objects with different resolution (Brisc 2004). A decision on the occupancy decides whether a volumetric primitive contains objects in the scene or not. If a projected voxel does not intersect a silhouette in at least one view, then it is removed (marked transparent) and if a projected voxel intersects only silhouette pixels in each image, then it is marked opaque. Otherwise the voxel projection intersects both background and silhouette and the octree representation is sub-divided.

When the images are not binary, then additional photographic information can be used to improve the 3D reconstruction process. These photo-consistent approaches can be used to additionally constraint the reconstruction, such that a valid 3D scene model that is projected using the camera matrices associated with the input images must produce synthetic images that are the same as the corresponding real input images. The photo-consistency checks the colour similarity of the pixels that a visible voxel projects onto, e.g. Figure 3.5. If the voxel colour is the same in all images or within an agreed level of deviation, the voxel is consistent and will be kept, otherwise it is carved<sup>5</sup> from the reconstruction. It is possible that many 3D scenes will be consistent with the images and so the image consistency does not guarantee a unique solution (Dyer 2001, Kutulakos & Seitz 2000). In most photo-consistent implementations, the surfaces are assumed to be Lambertian<sup>6</sup> and the voxel's centroid is projected into each of the images. In addition, the limits of photo-consistency need to be set, and this can have a major influence on the outcome of the reconstruction of the scene.

Voxel colouring is an approach to volumetric reconstruction of scenes that reconstructs the photo hull of the scene. It is an efficient method that visits the voxels in a particular order to perform photo-consistent checks on each voxel. To achieve this, it requires specific placement of cameras, particularly if it is not possible to surround the scene with the cameras (Seitz & Dyer 1999). This is improved upon using the generalised voxel colouring (GVC) algorithm proposed by

<sup>5</sup>Carving of voxels effectively removes them from the reconstruction process. This is achieved by setting their opacity to be transparent. The carving of one voxel generally changes the visibility of other opaque voxels

<sup>6</sup>The Lambertian reflectance model occurs when every surface appears equally bright in all directions regardless of the illumination



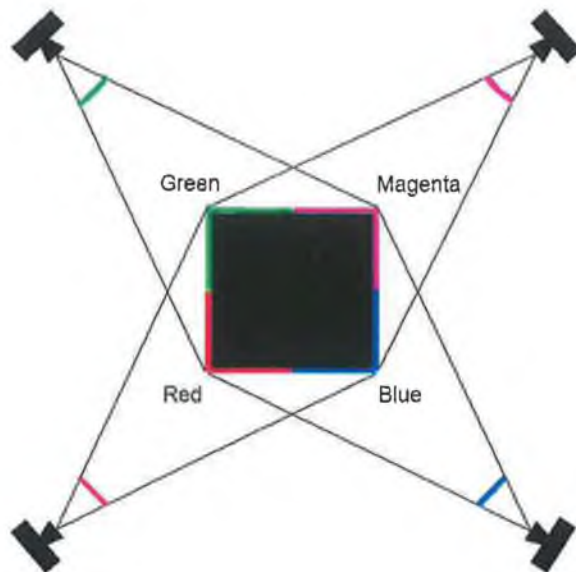


Figure 3.5: This figure illustrates how the pixel data in each image can be used to determine if the reconstruction is consistent with the captured data.

Slabaugh et al. (2004) which supports arbitrarily placed cameras and have minimal requirements on the order in which the voxels are processed. Moreover, Slabaugh et al. (2004) provide a comprehensive description of volumetric reconstructions using multiple arbitrarily placed calibrated cameras. This approach attempts to provide few restrictions on the type of camera used, or on the position from which the images are captured.

Voxel visibility is important when one voxel is occluded by another in any one of the camera viewpoints and is defined as the line segment connecting the centre of voxel  $x$  and the optical centre of any one of the cameras intersects voxel  $y$ , then  $x$  occurs after  $y$  in the ordering. This ordering ensures that all possible *occluders* of the voxel with respect to every camera have previously been visited. Thus, visibility testing is dependent only on the labels of voxels visited previously, enabling a one-pass algorithm (Dyer 2001).

In general, volumetric scene-space methods allow widely-separated views, but generally depend on calibrated cameras to determine the absolute relationship between points in space and visual rays. There exist different techniques to convert from a voxel representation to surface meshes, surfels<sup>7</sup>, etc., increasing the flexibility of the volumetric reconstruction process (Slabaugh et al. 2001). The complexity of the model and its accuracy depend on the coarse-to-fine definition and depending on the size of the images and the amount of detail that is available. It may not be possible to determine in advance what amount of detail is available and this could lead to a blocky representation of parts of a scene that require a fine resolution or a very fine resolution for parts of the scene that can be better represented using a coarse resolution.

Volumetric scene reconstruction is an important element in the creation of easily navigable scenes and provides techniques to build scenes from off-the-shelf cameras and the ability to integrate the different views sequentially into the existing model. Although the approach can operate

<sup>7</sup>A surfel is a contraction of the words surface and element

in real environments, the building of the volumes is highly dependent on the photo-consistency that is dependent on the lighting conditions. This means that some prior processing may be needed to ensure that the correct model is extracted. The volumetric scene reconstruction provides an ideal way to build scenes, although it does not detail how specific objects are segmented from the scene. In addition, in (Carranza et al. 2003) polygon based reconstruction is favoured over volumetric reconstruction because 3D graphics engines are best suited for rendering polygons.

### 3.2.4 Scanning Techniques

It is difficult to extract 3D shape information from intensity image of real scenes directly. Another approach is to measure the distance from the viewer to points on the surfaces in the 3D scene explicitly; such measurements are called geometric signals, i.e. a collection of 3D points in a known co-ordinate system. If the surface is measured from a single viewpoint, it is called a range image or depth map. Such explicit 3D information, being closer to the geometric model that is sought, makes geometry recovery easier.

Two steps are needed to obtain geometric information from the range image:

1. The range image must be captured,
2. Geometric information must be extracted from the range image. Features are sought and compared to a selected 3D model. The selection of features and geometric models leads to one of the most fundamental problems in computer vision: how to represent a solid shape.

According to Neugebauer<sup>8</sup>, it is important to capture a suitable number of range images to enable the complete reconstruction of an object. Attempting to predict where the most appropriate views should be captured from is an on going task. One approach to predicting the next view is described in (Klein & Sequeira 2000), although no method is detailed how this number should be suitably determined. The approach of Neugebauer also highlights that if the scanning device does not capture images at the same instant as scanning the object, then depending on the object the images can be captured subsequently using a common digital camera.

While scanners provide advantages over photographic techniques, they still do not overcome the problems associated with occlusions and extraction of dynamic 3D data. Thus, additional methods need to be considered because the extraction of 3D data is not exclusively used for the reconstruction of scenes or objects. For instance, 3D data can be used to categorise how a particular individual moves. While an estimate of this motion can be extracted from a single image or a scan in general, the most accurate information is extracted using multiple images or using sensors that form part of a motion capture system. These and other issues are described in Section 3.5.1 where in addition to the capturing of 3D data for the purposes of 3D reconstruction, additional data is captured to establish the pose and animation information associated with an object.

---

<sup>8</sup>Reconstructing 3D models of Real-World Objects from Range Data and Colour images Peter Neugebauer, CG topics 5/99

### 3.3 3D Reconstruction (Vision)

This section provides a review of some of the foremost techniques that exist for the creation of 3D objects and scenes, primarily from photographs, although some approaches that construct objects using range data are considered. The central problem addressed is concerned with multiple images: i.e. when given two or more images of a scene, possibly a camera model, points in these images which correspond to the same point in the world coordinate system, it is possible to construct a description of the 3D spatial relations between the points in the world (Faugeras & Loung 2001).

#### 3D Modelling and Rendering Scenes from Photographs

Modelling of 3D scenes from images is a challenging problem that has been a research topic for many years. A number of algorithms have been proposed that allow the extraction of complex 3D scenes from a sequence of images. Initial approaches related to robot guidance and how to extract sufficient information to allow the robots to move around an environment. This required only an estimate of the scene structure and, in general, did not perform a complete 3D reconstruction. Recently, the emphasis of the research has changed, focusing on obtaining accurate scene information and the generation of 3D objects that are present in the environments. These objects can be used in computer graphics and virtual reality applications (Pollefeys et al. 2000).

A significant amount of research has been devoted to the problems encountered by the large amount of calibration that is required and the restrictions that are placed, in particular on the camera motion. Using calibrated systems requires a high degree of expertise. In unconstrained environments, when the cameras are not used in calibrated systems, it is necessary to recalibrate the system each time that it is used. This reduces the flexibility of the system and the acquisition of the information. Projective reconstruction is used to reconstruct the object. This technique is based on previous research by Faugeras & Loung (2001) and Hartley & Zisserman (2000). These techniques proved that it was possible to reliably reconstruct an object up to an arbitrary projective transformation. In the techniques, the fundamental matrix was estimated from image pairs.

Debevec et al. (1996) proposed a hybrid method for modelling and rendering of architecture models using a small set of images. This approach exploits the constraints that are characteristic of architectural scenes and is illustrated in Figure 3.6. This enables the construction of parts of the buildings that are occluded, and the simplification of geometrical elements and their replacement with primitives. This is combined with a model-based stereo algorithm that enables the recovery of real scene information which is used to adjust the underlying model of the scene. In addition, the textures that are used when rendering the scene are view dependent, and this reduces the complexity of the final rendered model. The use of the model approach enables the reconstruction process to overcome some of the weaknesses of a purely image based reconstruction process, primarily the fact that the information in the photographs needs to be similar for reliable results to be obtained. This requires the use of many images from similar positions. This can require significant supervision and would require the capture of an unrealistic number of images and subsequent processing to derive depth and correspondence information. The distance between viewpoints also limits the number of new viewpoints that can be created (Debevec et al. 1996).

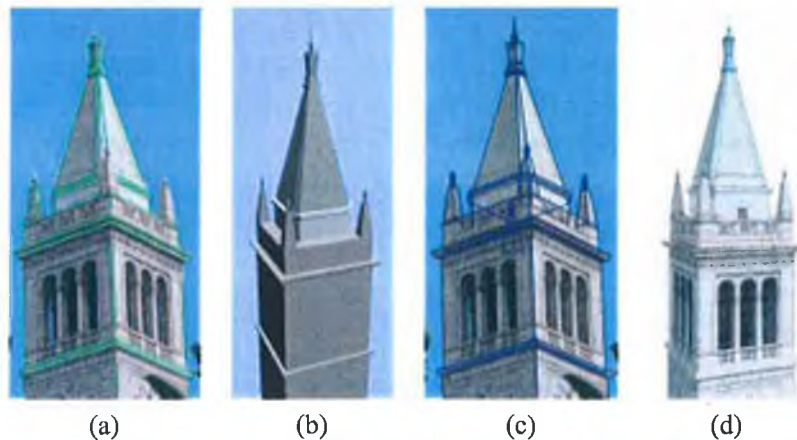


Figure 3.6: (a) A photograph of Berkeley's clock tower, with edges marked in green. (b) The model recovered by the method of Debevec et al. (1996). Although only the left pinnacle was marked, the remaining three, including the one not visible in the captured image were recovered from symmetrical constraints in the model. (c) shows that the accuracy of the model can be verified by projecting it into the original photograph. In (d) a synthetic view of the clock tower generated using the view-dependent texture-mapping method.

The task relies on a certain amount of user interaction to build the model, to process the input images and to select the relevant images for the construction of a particular building. In addition, the user carries out the initial modelling of the scene. Features, such as edges that correspond between the images and the model are marked. Then a task of minimising the difference between the edges marked in the images and the model edges is automatically undertaken. The minimisation process enables the computation of the camera positions and facilitates the texture mapping. The results that are achieved by Debevec et al. (1996) show that by using the hybrid approach, architecture can be reconstructed to a high level of detail and that the symmetry, which is strongly evident in the building design can be used to compensate when detail is missing. This approach is not as general as other approaches for modelling 3D scenes but shows the advantages that specific systems, particularly model-based system have.

Pollefeys et al. (2000) present a more general approach to scene reconstruction. The main objective is to allow off-the-shelf cameras to be used to acquire images by freely moving the camera around the object. Neither the camera motion nor the camera parameters are known in advance. The 3D model that is created is a scaled model of the original object that is captured. The textures used to texture the surfaces are also obtained from the images. The self-calibration of the camera system is important in many applications to produce a complete Euclidean reconstruction. This is a step up from the projective reconstruction. One assumption is that the same camera is used for the capture of the entire sequence. In addition to this, it is assumed that the same intrinsic parameters hold for all images captured. The system gradually obtains more information about the scene and the camera setup. The reconstruction starts with two images and calculates a projective frame of reference, and for the subsequent images the projective frame defined by the first two images is used to extend the projective reconstruction. To automate the projective reconstruction process, it is assumed that the images form a sequence in which consecutive images do not differ

too much. Thus the local neighbourhood about a scene point should look similar if the images are close together in the sequence. An example of this reconstruction process is shown in Figure 3.7 which contains four images that are taken from the Arenberg Castle in (Pollefeys et al. 2000) and a reconstruction of the castle. Based on this assumption, it is possible to find point correspondences between consecutive images, although the correspondences are not maintained through all the captured images. The disadvantage with this is that it is necessary to have a large number of images to do a complete 3D reconstruction of a large object.

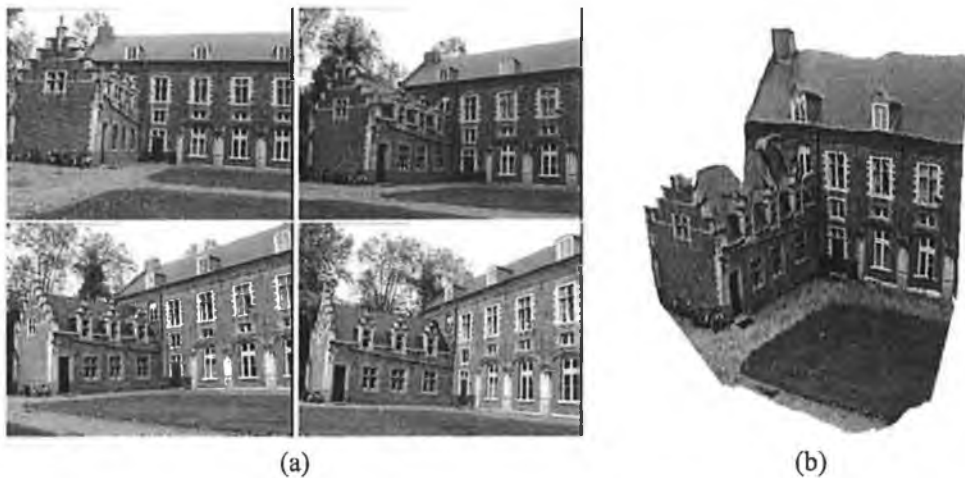


Figure 3.7: (a) contains four images of the set used in (Pollefeys et al. 2000) for the reconstruction of the Arenberg castle with the reconstruction shown in part (b).

Once the complete projective reconstruction is completed the position and orientation of the cameras are known for all the viewpoints. This information is obtained by calculating the epipolar geometry assuming some basic intrinsic parameters such as rectangular pixels, principal point at the centre of the image, etc are known. The Euclidean reconstruction provides accurate results of the scene that is reconstructed, and while the detail that is extracted is highly refined in comparison to Debevec et al. (1996), only the part of the building that appears in the photographs is visible and the range of view points is limited. Nevertheless the approach in (Pollefeys et al. 2000) lends itself to the reconstruction of small scale objects more easily.

Other correspondence based approaches include that of Beardsley et al. (1997) that implements a sequential updating of the 3D scene and recovering the camera positions and providing an affine and projective structure of the scene. This is in contrast to the previously described methods that perform the reconstruction using all available frames. This approach uses corner detection to identify the important points of correspondence within the images.

Klein & Sequeira (2000) apply scanners to 3D modelling of real indoor scenes. This approach attempts to overcome the problem of suitably scalable planning algorithms for the acquisition of different sized data. This approach uses a specialised capture unit that is composed of a laser scanner mounted on an tripod that can be moved from one location to the next to capture the necessary data maps. In general, the tripod can have variable height and scanning angle, and thus eight parameters need to be determined (three for position, two for the direction angle, one for

each of the horizontal and vertical field of view and one for the resolution). The results of Klein & Sequeira (2000) show that it is possible to automate the capture of data for the construction of 3D scenes, although this system is not suitable for use by a non-expert user as it requires the setting of different parameters depending on the data that is to be captured, and to ensure registration between different range images, it is necessary to have 0% of the pixels over-lapping thus limiting the motion between each range image.

The effectiveness of the above approaches rely on the accurate extraction of correspondences, and the matching parameters tend to fail as the base-line increases the effects of occlusion become more pronounced and the possibility of modelling the scene geometry becomes more complex. As a consequence, these methods are not well suited to the arbitrary positioning of cameras (Slabaugh et al. 2004).

The approaches described in this section provide accurate reconstruction of scenes and can be used to enhance an individual's virtual experience, but they require the user to have a significant level of expertise to create the models. The expertise is necessary to ensure that the information contained in each view does not vary significantly from the previous captured image, or the calibration of the cameras each time that they are used to account for the particular settings that are chosen, or finally, require the use of particular modelling tools to simplify the reconstruction process.

### 3.4 3D Object Pose Estimation

The recognition of 3D objects<sup>9</sup> from 2D images is an important element in many vision systems. To accurately recognise the object, it is first necessary to establish the pose of the object. To date, no generic approach has been established that can perform the task of recognising objects that humans, can in general, easily undertake. In particular, according to Chang & Ghosh (2000), the approach that humans use indicates that implicit 3D information is used to recognise the objects. In this section, some of the general approaches to pose estimation and object recognition are discussed. This acts as a forerunner to the discussion in Section 3.5.1, on the extraction from images of human pose and animation information.

According to Bergevin & Levine (1993), the task of generic pose estimation should be considered at a high-level, particularly when trying to extract the pose from a single view. This is important in building a general system, but in the majority of situations systems are developed for a particular task, and a general system adds unnecessary complexity. They highlight that humans can recognise objects from simple line drawings and use this as a basis for the development of a generic object recognition system. This is implemented in a manner that attempts to build coarse descriptions of the objects from single view edge maps. In (Bergevin & Levine 1993), the problem of recognising unexpected 3D objects from single 2D views is investigated. The PARVO (Primal Access Recognition of Visual Objects) system is discussed as a beginning in the goal of obtaining a systematic (or computational) implementation of pose estimation. This system is applied to simple geometrical objects such as cups or other objects composed of primitive shapes.

---

<sup>9</sup>The objects that are considered in this section are primarily rigid objects

A major problem associated with object recognition in unconstrained and complex environments is the determination of the pose. To enable the extraction of sufficient information, it is necessary to constrain the motion or the pose of an object. It should also be noted that the constraints should not place a major restriction on possible viewpoints or the pose of the object to ensure that a true estimate of pose and thus shape can be reliably extracted (Brophy et al. 2004).

An alternative approach to the task of recognising objects or estimating the pose of the object can be obtained using a generalised cylinder based approach described in (Zerroug & Nevatia 1995). Two approaches are presented based on SHGCs (straight homogeneous generalised cylinders), a straight-axis primitive and PRGCs (planar right generalised cylinders) and a curved (planar) axis primitive which form a large class of man-made objects. The different generalised cylinder elements are used to describe the different classes of objects based on information extracted from the image. The pose is estimated using a matching procedure that attempts to match the image information to an equivalent model.

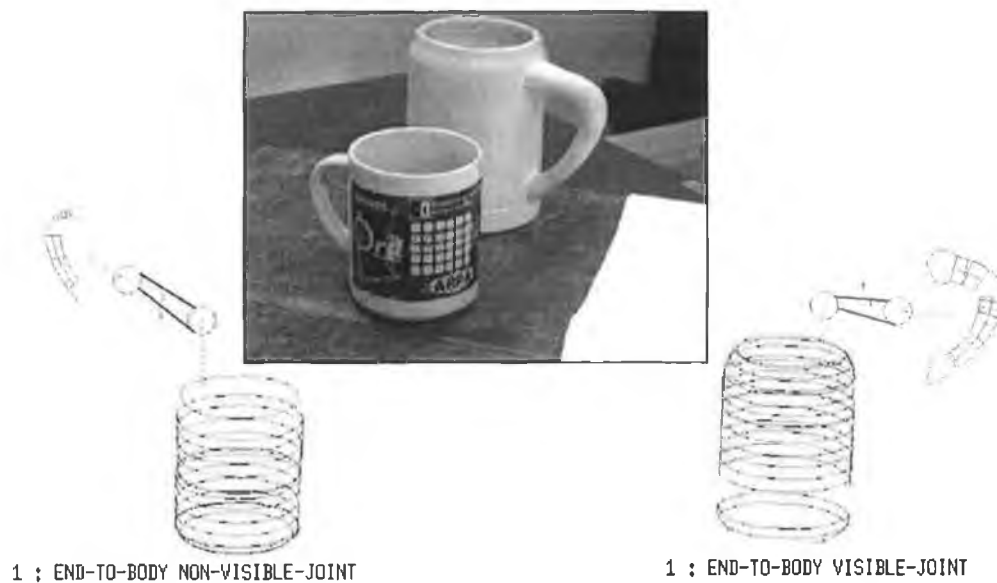


Figure 3.8: Example of pose of an object detected using the process in (Zerroug & Nevatia 1995). Each detected object is described as a graph where nodes are parts and arcs labeled joint relationships between parts.

Brophy et al. (2004) present a number of approaches to determine the pose of an object using various labels. The basic idea is that if easily identifiable labels are placed on an object, then it should be possible to identify the pose of the object through image processing and machine vision analysis. The use of a known label enables the establishment of the 3D pose from a single 2D image. In a similar manner, Martin & Aggarwal (1983) discuss the properties of special illumination conditions that can indicate surface orientations.

A learning stage is important for training a pose estimation system. This is achieved by considering the object from a number of different views, in a manner similar to that employed by humans

to recognise a 3D object. Chang & Ghosh (2000) use spherical manifolds<sup>10</sup> to represent the poses of the object that is to be determined. This approach is designed to automatically identify aircraft. A total of 684 different poses of each aeroplane are used as training sets. This is then used to establish the recognition of other views not used in the training of the model and on real objects with a high degree of accuracy. Chang & Ghosh (2000) advocate the use of 3D models when possible to generate the necessary views of the model for training purposes. This is also discussed in (Hutenlocher & Ullman 1990) in conjunction with a description of a model based approach to pose estimation and recognition of solid objects with unknown 3D position and orientation from a single 2D image. The transformation in the image is calculated by using correspondence pairs on the model and the image. The approach firstly computes possible alignments using a minimum number of correspondences between the model and the image features. The alignments are then verified by transforming the model into image coordinates and comparing the results.

It is important to realise that the extraction of the pose of an object initially depends on the ability to recognise features in the captured image. As discussed above, this involves a certain amount of learning. The estimation of the pose can be enhanced when information from additional views is considered. This is of particular importance when considering articulate objects which can undergo non-rigid deformations, which is discussed in terms of human pose analysis in Section 3.5.1.

### 3.5 Creation of Virtual Humans

With the increased availability of digital cameras, powerful graphics cards, 3D graphic engines and increased processor speeds, the possibility for a home-user to create and modify their own model is not unforeseen. This will facilitate the personalisation of various applications including interactive games. The key element necessary to enable this is the flexibility of both the capture and the reconstruction process. The provision of the techniques that enable an individual to create their own model will inherently be automated or require at most very limited user interaction. Moreover, it is imperative that the models can be created using off-the-shelf digital cameras and that there are no (or very few) restrictions imposed on the individual in terms of camera set up or the use of a controlled environment. In addition, the 3D reconstruction technique must be able to overcome inaccuracies that may result from simplifying capture procedure. This can be surmounted through the use of an underlying model (Hilton et al. 1998) (Lee, Goto & Magnenat-Thalmann 2000) (Cohen. & Lee 2002) which is modified using shape information extracted from the captured data, or through the use of texturing that can enhance the appearance of the model (Boyle 2004) (Boyle et al. 2005).

This section is organised as follows: the key stages in a system creating 3D human models are

---

<sup>10</sup>In real world applications it is more than often not required to decide if we have an image of a specific object, but given an image recognise the object relating to it from a large data base. This is possible using an index of some shape invariant that can be calculated from image measurements. Because of the ambiguity formed when projecting 3D onto 2D there is no unique shape invariant function. However it is possible to find invariant functions such that the set of all points corresponding to a feasible image of the object is a manifold. This manifold, given the invariant function used, is unique to each object and an image corresponding to this object must be on the manifold. H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," International Journal of Computer Vision, vol. 14, pp. 5-24, 1995.



discussed and reviewed. Firstly, the techniques for acquiring the data are described and assessed in terms of the quality of the captured data, cost and level of knowledge required to set up the capture system and to interpret the data. The processing of the data is then described in particular, for the photogrammetric approaches as they offer the most flexible approach to creating virtual humans. The next section then examines how the realism of the models may be increased by texturing and by using modelling tools.

### 3.5.1 Data Acquisition

The acquisition of data is fundamental to the creation of any 3D content and in particular, the capture of non-rigid complex objects introduces additional problems such as requiring an experienced user. The capture process must be robust and some amount of post processing may be required to extract the relevant information from the captured data. The most common methods for recovery of 3D human data are from range (or whole body) scanners, motion capture devices and photogrammetric/optical systems. A distinction is made between motion capture systems that require the individual to wear sensors and photographic systems that require the individual to undergo particular motions to identify particular joint information.

#### Motion Capture

Motion capture plays an important role in the abstraction of motion information from particular environments. This is an invasive method that requires the individual to wear sensors which are tracked. This tracking information is then used to provide a model with the same motion (Gleicher & Ferrier 2000). This is popular within computer games because it provides the characters with realistic movements. Two examples of motion capture systems are shown in figures 3.9 and 3.10. Motion capture systems are used primarily for the extraction of pose information and animation. The information that is captured can be used to personalise an underlying model. The sensor data can initially be used to alter the topology of the model using the position of sensors and the distance between sensors.

(Welch & Foxlin 2002) review various motion tracking systems for view control, navigation, avatar animation, etc. In effect, there are various techniques ranging from mechanical, acoustic to magnetic and optical, none of which is suitable to solve problems of every technique and application.

There are two types of motion capture; the first is classified as on-line motion capture and this uses magnetic sensors to plot the movement of joints and this can be directly fed to the model to mimic that animation (Babski & Thalmann 2000). Other on-line methods include the use of ultrasonic or mechanical sensors. The second type of motion capture systems are off-line systems which obtain the pose information through multiple views. It takes longer to extract the motion but provides more accurate results, in particular, for extraction of subtle gestures and also facilitates accurate joint location and the extraction of data for the creation of 3D models (Plankers & Fua 2003).

---

<sup>11</sup>Gypsy Motion Capture Systems, Meta Motion Capture; [www.metamotion.com](http://www.metamotion.com), last accessed April 2005.



Figure 3.9: The Gypsy Motion capture System is shown. This motion capture system is an electro mechanical system that consist of an exoskeleton made of lightweight aluminium rods that follow the motion of the performer's bones <sup>11</sup>.



Figure 3.10: Real-time Motion capture System that captures the individuals movements and animates a model to produce the same motions. <sup>11</sup>

The main weaknesses in these approaches for the creation of flexible models relate to the tracking of the individual sensors that require a large amount of post-processing by a skilled user to isolate and locate individual sensors. This has been improved by introducing a skeleton that is used to help predict the location of particular sensors (Boulic et al. 1998) (Plankers & Fua 2003) (Herda et al. 2000). Both on-line and off-line motion capture systems can take advantage of the skeleton, although the on-line model lacks the finer movements. The input 2D sensor locations are expressed in multiple-camera image space and between two and seven cameras are used to track the sensors giving extrapolated sensor trajectories which help to resolve ambiguities by predicting the future locations of the sensors and aid the construction of a 3D model.

Magnetic, ultrasonic and acoustic sensors are ideal for capturing the motion information that is important for realistic movements, but it does not provide sufficient means for the creation of a personalised and photo-realistic human model. It is not suitable for low-cost flexible implementations because of the cost of such a system, in terms of both equipment and processing time. In addition to tracking the sensors, optical motion capture systems facilitate the extraction of image data that can be used to create realistic human models. However, to accurately extract the shape and motion information at the same time would require a system of (possibly calibrated) cameras thus reducing the possibility of using this technique in a flexible framework. A major advantage of using the sensors is that particular information can be easily extracted based on the locations of the sensors although they must be accurately positioned and can possibly move. Motion capture systems, while reasonably accurate, do not fit into widely and easily accessible immersive environments (Cohen. & Lee 2002). (Plankers & Fua 2003).

### Range Scanners

3D whole body range scanning provides some of the best results for the creation of models but the cost and the quantity of data provided makes it difficult to use in practical applications. It has the advantage that the surface variation of the object is readily known and can be easily separated from its environment although the environment is generally constrained (Sonka et al. 1999). Another advantage that range sensors have is that they are non-invasive. The operation of a range scanner involves the projection of a structured light pattern onto the surface of an object and capturing a digital image of the projected pattern (Collins & Hilton 2001). Optical range sensors measure the 3D location of points on the surface and produce a cloud of points. It is then necessary to triangulate between the projector and camera to reconstruct the distance of points on the surface from the camera producing a depth map.

There are several types of range scanners that are currently available. Examples range from the whole body colour 3D scanner produced by Cyberware<sup>TM</sup><sup>12</sup> to the Polhemus 3Space FastSCAN<sup>TM</sup> handheld laser scanner<sup>13</sup>. The whole body scanner is a complete system designed specifically for capturing the shape of the human body in a single scan (Ju et al. 2000, Buxton et al. 2000). The capture system is shown in Figure 3.11. It produces high quality models (see Figure 3.12) although such a system is expensive and it is not suitably portable for flexible use. The Polhemus handheld laser scanner emits a laser beam from a wand as it is smoothly swept over an object. In both

<sup>12</sup>Cyberware Whole Body Colour 3D Scanner: [www.cyberware.com](http://www.cyberware.com), last accessed April 2005

<sup>13</sup>Polhemus FastSCAN: [www.fastscan3d.com](http://www.fastscan3d.com), last accessed April 2005

systems, it is necessary to combine the different sweeps over the object. This combination is made easier in the case of the whole body scanner, as the location of each scanner is known. Registration is required between views to generate a complete surface of a 3D object. One software tool available for this task is Rapidform 2004 produced by INUS technology Inc.<sup>14</sup>, which is available to compute the registration, to complete the triangulation and generate the complete model. The final model requires further processing to split it into individual body parts to enable animation of the model (Ju et al. 2000).

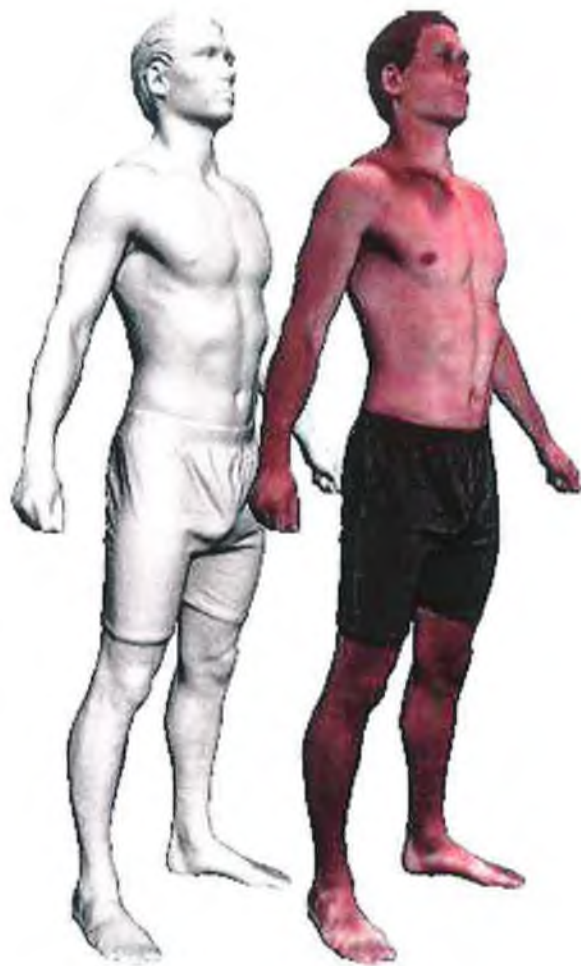


Figure 3.11: Example of a model created with the cyberware system<sup>12</sup>.

Apart from whole body scanners, range scanners have not been used for capturing the shape of an individual as the number of sweeps that are required would require the individual to stand still for an unacceptable length of time. Therefore, it is not currently feasible to use hand held scanners to create a human model although the process has been successfully applied to the reconstruction of statues (Curless & Levoy 1996) (Hilton & Illingworth 2000).

<sup>14</sup>RapidForm 2004: [www.rapidform.com](http://www.rapidform.com), last accessed April 2005



Figure 3.12: The Cybeware whole body scanner system<sup>12</sup>.

### Photogrammetric/Optical Systems

The capture of data using a single camera or a number of cameras encompass what are regarded as the most difficult approaches to the creation of 3D human models. In particular, the information can be captured from a single camera (Hilton et al. 1998) (Brisic & Whelan 2004), from a system of cameras (Kakadiaris & Metaxas 1998) (Cohen. & Lee 2002) or from a video sequence. Unlike a range image, it is not possible to extract depth information from a single view, and thus it is necessary to use two or more different views of an individual or object to extract any depth information which can be used to produce a projective reconstruction of the object. In general, it is necessary to calibrate the cameras either prior to the capture of the data or from the data that is obtained during capture to be able to produce a Euclidean reconstruction. This is a difficult task that is described in Section 3.3 and well documented in (Faugeras 1993, Hartley & Zisserman 2000, Pollefeys et al. 2000, Brisc 2004, Han & Kanade 2000). It is also necessary to identify key features in each view and find correspondences between each image. Having established the correspondences, it is possible to extract 3D information. There are various techniques available for the establishment of correspondences, including the use of corner detectors (Sonka et al. 1999).

Photogrammetric systems are the most promising approaches to the creation of flexible capture systems because of the large availability of digital cameras, including camera enabled mobile phones and web cams, although the success of such a system depends on both the quality of the images that are captured and the usability/flexibility of the reconstruction process (Hilton & Fua 2001). Moreover, in a flexible system the minimum number of images necessary to create the photo-realistic human model should be captured, but reducing the number of images makes it increasingly difficult to extract correspondence and introduces occlusions.

### 3.5.2 Building a Virtual Human

This section provides a review of existing 3D reconstruction techniques, primarily photographic, which have been applied to the extraction of pose and shape information for the creation of human models. The concept of building a virtual human is a complex task that incorporates a certain

amount of prior knowledge about the structure of a human. This information is necessary for automation, to accurately extract the shape information, to overcome occlusions and for facilitating the prediction of where certain parts of the body are likely to be found.

Kakadiaris & Metaxas (1998) propose a system for the acquisition of the human pose and shape information from multiple views. This approach uses three cameras to capture sequences of images and it does not use a prior model. It achieves automated joint localisation and can subsequently extract shape information to facilitate the creation of a 3D human model. In this approach, the individual is required to undergo two different sets of movement to extract firstly the pose and then to extract the 3D shape information and to overcome the occlusions among the body parts. The position of the cameras leads to scaling issues as only three cameras are used, one in front of the individual, one above and one to the left of the individual. The camera setup is shown in Figure 3.13. All the shape information from the right of the body is therefore biased.



Figure 3.13: Example of the camera set up in the approach of Kakadiaris and Metaxas (Kakadiaris & Metaxas 1998).

Requiring the individual to undergo a series of movements can lead to errors in the capture procedure if a step is omitted. Thus the system is best operated under the supervision of an expert user. The movements undertaken to identify the left arm using this process is illustrated in Figure 3.14. In addition, requiring a set up with three cameras makes this approach unsuitable for use in a flexible system.

The 3D reconstruction is achieved by firstly building a model of the human standing and to incrementally refine this by extracting the different body parts as they become visible in the different views. The 3D shape of a body part is obtained at the end of the appropriate set of movements. The reconstruction of the leg is shown in Figure 3.15. This means that the shape fitting is carried out in 2D and the 3D estimation is performed only once. Two mutually orthogonal views and their spatial relation are used to create the 3D model. This enables the intersection of the two views that form the meridians whose planes are parallel to the planes of the contours. The remainder of the 3D shape is then interpolated.

The technique of Cheung et al. (2003) creates a human model from multiple silhouette im-



Figure 3.14: Example of the types of movements an individual must undertake in the approach of Kakadiaris and Metaxas (Kakadiaris & Metaxas 1998).

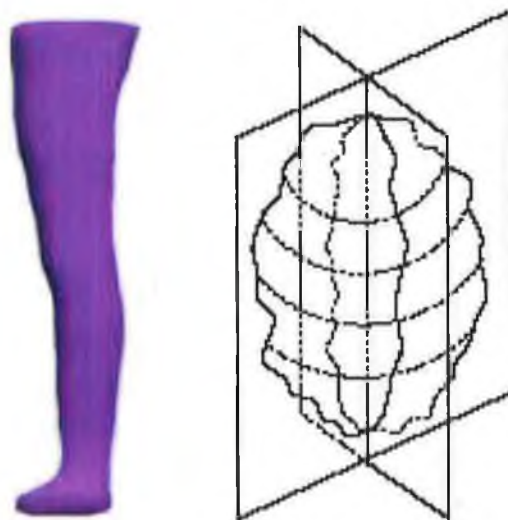


Figure 3.15: Example of the reconstruction of the leg and the general approach to combining the 2D shape information (Kakadiaris & Metaxas 1998).

ages. This approach extends the classical notion that silhouettes have to be extracted at the same instance of time or relate to a static object to be relevant in the computation of a valid visual hull. This enables the silhouettes extracted at different time instances to be used to extract the shape information of an articulate human. This paper builds on the visual hull concept introduced by Laurentini (1994) (that describes the maximal shape of the object created by volume intersection) by defining that the edges are the lines that correspond to points on the object boundary that can be projected into every view to a varying degree.

Cheung et al. (2003) tested their approach with both synthetic and real sequences of rigidly moving individuals. The synthetic images are captured using a virtual scene with 8 cameras and the model is made to rotate around the z-axis. In the real sequences, the individual stands on a turntable with unknown speed and 30 frames per second are captured with 8 calibrated and colour balanced cameras. The results that are presented seem to lack the real detail that is obtained from other multiple camera reconstruction techniques, since it is not possible to reconstruct all the cavities from a limited set of views (Laurentini 1997), although the texture adds to the appearance, particularly around the face where the eyes can be seen. This approach consists of two interlaced stages that correctly segment the silhouettes of each articulated part and estimate the motion of each individual part using the segmented silhouette. No apparent constraints are imposed on the individual and like (Kakadiaris & Metaxas 1998), the joints and the body parts are extracted sequentially. This system is unrealistic for a low-cost or flexible approach and requires a large amount of overhead in terms of timing, and it is also impractical to have an 8 camera setup. In addition, the environment used appears to be controlled allowing easy extraction of the individual silhouettes. The reconstruction of a synthetic model and the reconstruction of an individual are shown in figures 3.16 and 3.17.

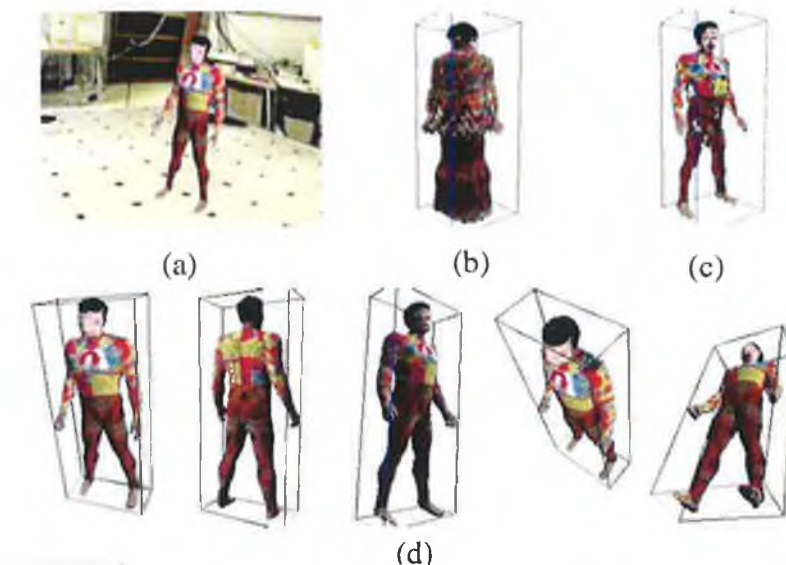


Figure 3.16: Results of the Cheung et al. approach applied to synthetic data (Cheung et al. 2003). (a) one of the input images (b) unaligned color surface points and (c) shows the aligned colour surface points and (d) refined visual hull.

Other multi-camera approaches for the estimation of pose that could be used for the creation



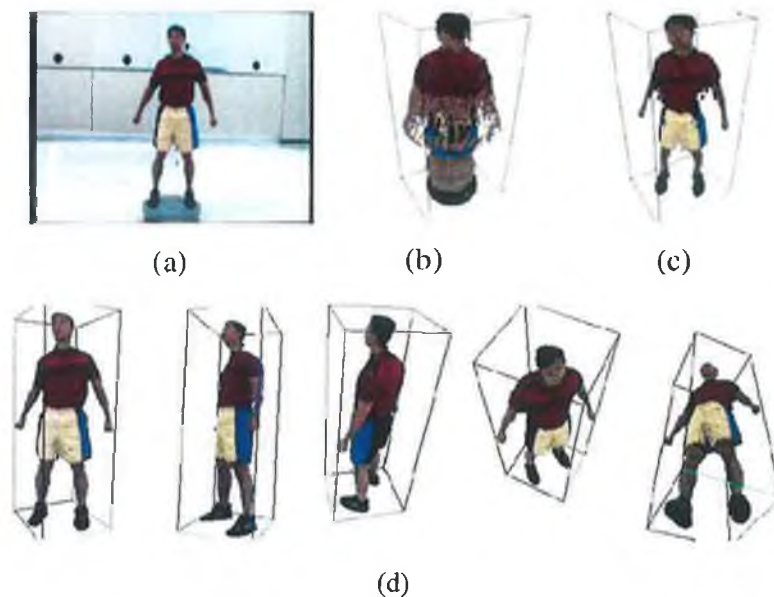


Figure 3.17: Results of the Cheung et al. approach applied to real human body (Cheung et al. 2003). (a) one of the input images (b) unaligned colour surface points and (c) shows the aligned colour surface points and (d) refined visual hull.

of a 3D human model include (Cohen et al. 2001), (Iwasawa et al. 2000) and (Bottino & Laurentini 2001). In (Cohen et al. 2001), an approach is taken for finding the body posture of an individual from a multiple camera set-up. The silhouettes are extracted using a background learning technique which requires a large set of images to generate a Gaussian distributed model of the static part of the scene. Correspondences between the different views are extracted using epipolar geometrical properties, and the integration of the different views enables the 3D representation to be inferred from synchronised video streams and mapped to a generic articulated body model. In (Cohen et al. 2001) and (Iwasawa et al. 2000), it is not clear what the user is required to undertake to achieve accurate reconstruction of an individual and what parts are automated.

This research is extended in (Cohen. & Lee 2002) where a system using two or more cameras tracks an individual and fits a skeleton which can be adjusted over time to the captured data. The tracking method is based on particle filtering without requiring the estimation of widths of body segments that vary among individuals. This is achieved by using a similarity measure between the articulated model and the captured data. The process of 3D reconstruction included in (Cohen. & Lee 2002) does not capture the individuals complete data, thus this can only be an estimate.

Wingbermhle et al. (1997), propose a method for the creation of human models for video conferencing systems. It requires the use of a stereoscopic system in a controlled environment to extract the individual's shape information and only the upper part of the body is created, which makes the model unsuitable for alternative applications. The use of flexible triangular mesh that adapts to the shape of an individual provides an efficient method for the modification of the shape in real-time. One of the major problems with this work is that the initial joint positions need to be manually extracted from calibrated views of the individual. In an extension to this work by Weik et al. (2000), the individual is required to sit on a turntable in a controlled environment and

the necessary shape information is extracted for the creation of a human model and while this is successful it is not suitable for general use. Figure 3.18 shows the underlying mesh structure and the textured mesh with data for a particular individual.

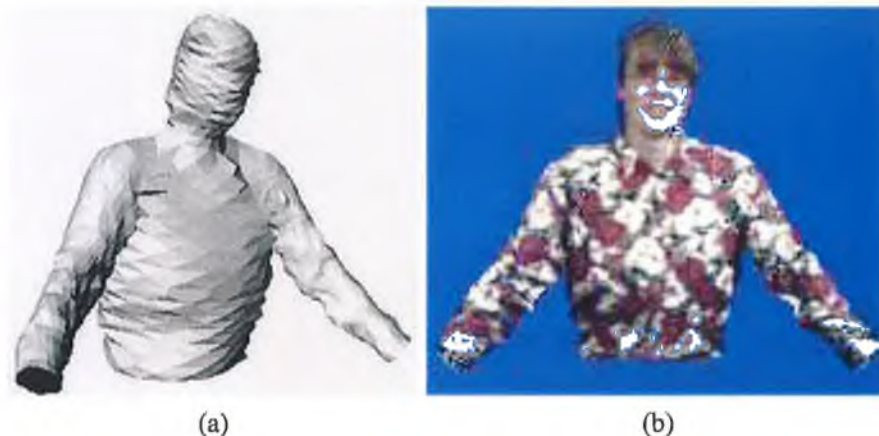


Figure 3.18: Models created for video conferencing systems (Wingbermhle et al. 1997). (a) shows an example of the flexible underlying mesh and (b) the final textured mesh.

Brisic & Whelan (2004) attempt a complete voxel based reconstruction of the individual. The individual is required to wear markers and stand still for a period of time while a series of images are captured from different viewpoints. To take account for the unknown viewpoints, calibration patterns are placed at different locations around the individual. It is not explained how possible variations that occur between the different images are incorporated into the model. Using the voxels enables the user to specify the resolution that a particular part of the individual is reconstructed.

### Model Based Approaches

Model based approaches take advantage of prior shape information to simplify and aid the extraction of shape and motion information. Model based approaches are popular in the development of characters for computer games where the game play has priority over the detail, as, in general, games are designed for the movements of a default or generic models. In this situation, texture mapping can be used to give the underlying model the appearance of a particular individual (Hilton et al. 1999), (Lee, Goto & Magnenat-Thalmann 2000), (Villa-Uriol et al. 2003). Using an underlying model offers advantages over competing methods including:

- The model can help to overcome problems caused by occlusions.
- The final textured and/or deformed model can be easily immersed into existing virtual worlds.
- The joint positions are known in advance and thus the final model can be easily animated using existing animation streams (Hilton 2003).
- In mobile applications or in virtual worlds, the model can be locally stored and the particular textures associated with an individual can be communicated reducing the data that is

communicated.

If non-deformable models are used which cannot adapt to differently sized bodies the final models will be the same size and they can be very similar in appearance. In addition, during the capture process the motions that the individual can undertake may be restricted and the final model will not truly reflect the motion of the individual (Kakadiaris & Metaxas 1998).

One of the most complete model based systems is that of Hilton et al. (1999). In their approach, the individual is required stand against a photo-reflective blue screen. A single camera is used to capture four images of the individual. The individual adopts a particular pose to enable the accurate location of important features and turns 90° degrees between each capture allowing four orthographic projections to be obtained. The images are then used to modify an underlying model by generating four corresponding silhouettes of the underlying model and identifying key features on the silhouettes and captured images. These key features allow the captured data of the front and back views to be split into seven different parts. This is important in the texturing of the model, as it ensures that the texturing is carried out part by part, and ensures that scaling is maintained. The results of this approach show that it is possible to create realistic models with a relatively flexible set up. The images are captured with a standard digital camera in a controlled environment. In addition, the model is deformed to take on the appearance of the individual that is captured. This is achieved by estimating the displacement of 3D points between the projection of the surface of the underlying models and the surface of the captured individual. The complete set of steps undertaken to create the model are shown in Figure 3.19.

The work of Hilton et al. (1999) was extended by Lee, Goto & Magnenat-Thalmann (2000), Lee, Gu & Magnenat-Thalmann (2000), Lee (2002). In Lee's approach, the main emphasis is on the creation of H-Anim models from data captured in real environments. Three images of the individual are captured in a real environment using a single camera. To simplify the camera calibration process, a method of direct estimation based on the distance between the camera and the individual or using the individual's height is used to scale the model in each view. An interactive feature extraction process is initiated to permit the individual to be located in the images. These features correspond to those extracted by Hilton et al. (1999) and additional estimates are used for the localisation of joints like the knees and elbows. The location of these joints is important for modifying the body parts of the underlying model to match the captured shape information. A heuristics based silhouette approach is used to extract the individual from the background. In this approach, a Canny edge detector (Canny 1986) is applied to the image, then a colouring-like linking algorithm is used to link the edges into connected segments. To avoid potential errors, the line segments are split into short segments. The segments are then evaluated to decide if the boundaries are correctly extracted and form an ordered set of edge segments. In this procedure, it is not clear how much user interaction is required to accurately segment the background. Following this, an edge growing technique in combination with texture blending is used, to account for possible errors in the extraction process. This is important because it ensures that the background is not textured to an individual's body model and overcomes problems caused by different illumination conditions and errors in the boundary extraction process. This process also uses separate images for the creation of the face and thus enables the complete reconstruction of the face that ensures a high resolution when examined closely. Examples of the final model are shown in Figure 3.20.

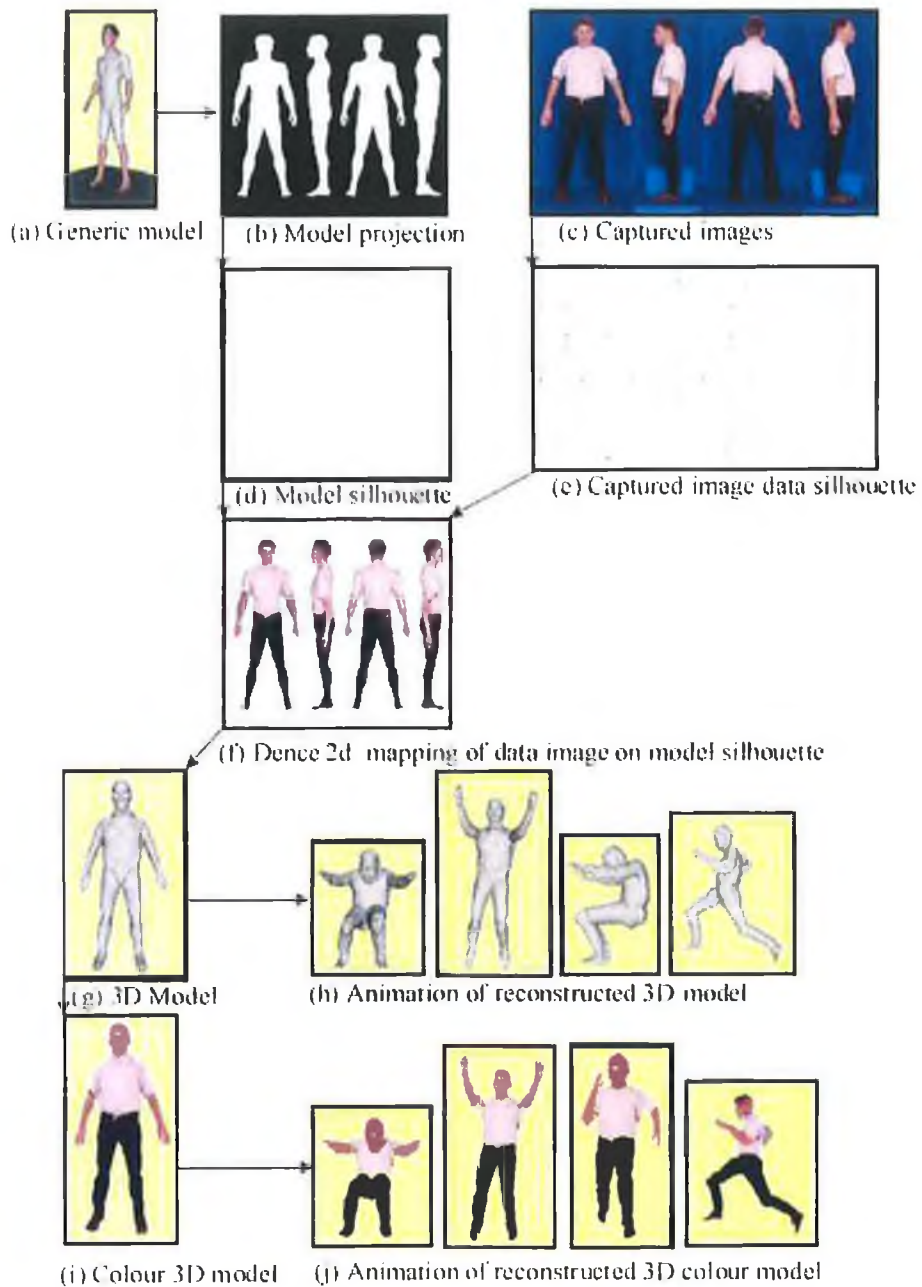


Figure 3.19: The Hilton et al. system for the reconstruction of an individual. (Hilton et al. 1999)

It has been shown in (Boyle et al. 2005) that highly realistic faces can be obtained from low resolution images that capture the whole body by using facial features to correctly position the facial image on the underlying model.

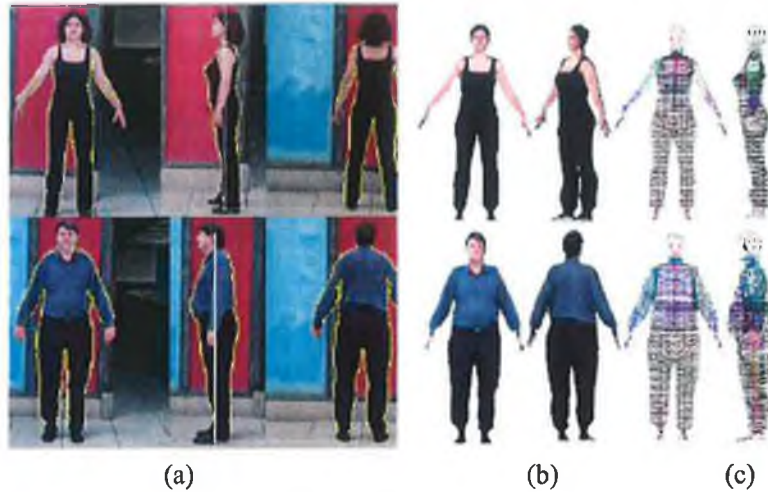


Figure 3.20: The creation of human models in real environments (Lee, Goto & Magnenat-Thalmann 2000). (a) shows original images with the silhouettes super-imposed marked in yellow, (b) The final H-anim model combined with the separately reconstructed face and (c) shows the untextured updated model.

Villa-Uriol et al. (2003) extend the approach of Hilton et al. (1999) to create models of individuals with unknown poses. This approach makes use of a video sequence and constrains the individual to stand on a turntable reducing the flexibility of the approach. Moreover, the approach requires the calibration of the camera. In this approach, it is not stated if the sequence is captured in a real or constrained environment. This approach introduces greater flexibility into the capture process, but it increases the complexity of the reconstruction process. This may provide a better fit of the captured data to the underlying model, although no additional kinematic information is obtained.

Ju et al. (2000) take advantage of the high quality models created by a whole body scanner by combining them with available human animation models to enable a particular individual to be immersed in a virtual world. This approach starts by developing a technique for segmenting the 3D model into individual body parts as the scanner provides a single surface with thousands (and possibly millions) of points with little or no semantic information. This is an automated approach that provides accurate segmentation of the body into the different parts and allows the identification of possible joint locations that are used to animate the model, (see Figure 3.21). This technique is sufficient in the cases where the individual wears tight fitting clothes as it is possible to take advantage of crevasses in the body that are not obvious when the individual wears clothes. This work is extended in (Ju & Siebert 2001) (Sibiriyakov et al. 2003) to enable the personalisation of the underlying model and the generation and animation of the characters.

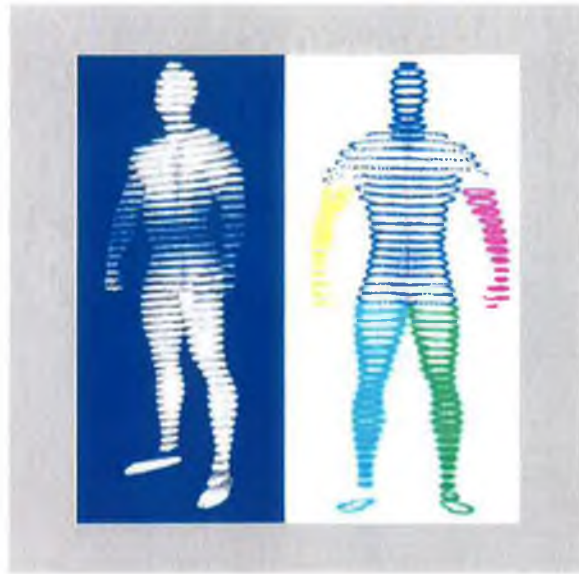


Figure 3.21: Segmentation of the scanned body into individual body parts (Ju et al. 2000).

### 3.5.3 3D Modelling Environments

The generation of human models using 3D modelling environments is an effective tool that can be used to overcome some of the problems encountered in adding realism to the model and in the development of correct muscle simulation. This information in general, can not be accurately extracted from a 3D body-scan or from a series of images captured at any time instance. Thus, to reliably extract and generate human models that deform as real individuals, it is important to have different simulation environments available to create the small movements that are associated with an individual. Moreover, modelling environments provide valuable tools when:

1. It is not possible to get the exact shape of the face, hands or other body parts,
2. A player wants to change the appearance of a default character in a game,
3. Capturing data may be used as basis for human model (when only a limited number of views are available),
4. The number of polygons that are required is set prior to reconstruction, for instance when real-time constraints are important (Kalra & Magnenat-Thalmann 1998),
5. A single image or images from a set of old images are only available, then a modelling environment can be used to reconstruct missing information and provide an approximation to the human shape.

Moreover, modelling environments are important in developing predefined movements or gestures that are important when immersing real-time simulated human models in games and virtual environments (in particular multi-user environments) where effective interaction between users adds to the sense of presence.

In (Kalra & Magnenat-Thalmann 1998), a detailed description of different tools that are available for the creation of accurate models are described as well as the way the information captured in images can be successfully manipulated to provide greater realism to the face and hands of the model, ensuring that the skin can be modelled and deformed. Research in this domain has been extended and recent publications such as (Seo & Thalmann 2004, Gutierrez et al. 2004, Magnenat-Thalmann & H.Seo 2004) provide human models for use in diverse applications.

The use of modelling environments that enable the specification of the smallest joints and the finer movements that have been generated based on close examination of various individuals, offering the possibility to take flexibly created models and enhance their appearance by a skilled user. This had been highlighted on the face and on the hands, as it is difficult to extract accurate information based on the captured data using either a photogrammetric approach or using a body scanning approach. In addition, once the model is created these additional features can be added to increase the realism and generate movements that will be able to distinguish one individual from another.



Figure 3.22: The *Sculptor* character creation tool (Kalra & Magnenat-Thalmann 1998). Example of how the facial image is combined with an underlying model to sculpt the model to take on the appearance of the individual.

The approach of Kalra et al. in (Kalra & Magnenat-Thalmann 1998) involves capturing the basic human information and creating a framework that is capable of deforming (modelling) and animating humans in real-time. They propose an interactive method for modelling the face and this increases the speed of creating a face model that conforms to the real-time restrictions. This technique involves the use of several images of each part of the body, as the modelling is done in different stages, including particular modelling for the head and the hands. This is beyond the scope of an average user. Then the body is built in a number of stages that conforms to the layered approach. At each layer the complexity and realism increases. The final stage requires interactive texture fitting to ensure that the models are textured correctly (Kalra & Magnenat-Thalmann 1998). This is the procedure that is followed in the character studio plug-in used for character creation in 3D Studio Max.

### 3.6 Discussion

Having examined the different methods available for the creation or 3D reconstruction of objects, it is apparent that the quality of the reconstruction depends on two principal factors. The first is that the shape of the final model and the realism is only as accurate as the initial segmentation of the object from the background, which is greatly simplified, when a plain or known background is used. Secondly, the level of detail within digital images is limited to pixel accuracy. Various approaches were examined for the creation of 3D models, primarily focusing on photographic techniques because the techniques using scanners and motion capture devices are beyond the means of most users. The photographic techniques that were described ranged from single to multiple camera systems capturing either still images or video sequences. The captured data has been used to provide photorealistic modelling to deform underlying models or to animate an existing model.

The first part of this chapter focused on the existing techniques that are used to reconstruct objects from real environments. Each of the techniques was examined to determine which approach or what parts could be used in a flexible and automated approach to human modelling.

- Approaches using multiple images captured from unknown viewpoints required the establishment of correspondences between consecutive frames, which is only possible if the frames are taken from a similar viewpoint. These constraints make it difficult for a non-expert user to capture the images and any applications based on this method of reconstruction require the capture of many images. Moreover, this requires the individual being captured to maintain the same position during the capture phase, and any movements can introduce errors in the reconstruction process. Thus such methods are not considered flexible enough.
- Silhouette and volumetric approaches to reconstruction using images provide an alternative that does not rely on the establishment of correspondences between images. When only greyscale or black and white images are considered, the extraction of silhouettes needed for the reconstruction is reduced and facilitates a complete reconstruction. The use of greyscale images is not viable with current virtual environments. With colour images the versatility of the approaches can be increased, and using techniques such as photo-consistency are valuable in the extraction of the silhouette and the determination of the associated volume.

Both of these approaches require knowledge of the centre of projection for each image captured to create the visual hull of the object to be reconstructed. Volumetric approaches are well suited for the reconstruction of scenes and when a large number of images are available, but they are sensitive to movements between each image captured. The uses of space carving techniques are computationally expensive, and it is not specified how a particular object can be separated from its environment. Silhouette based approaches are not ideally suited to the reconstruction of scenes because they focus on the extraction of an object from the scene to establish the visual hull of the object. In silhouette based approaches the resulting visual hull contains the maximum information that can be obtained without



capturing images inside the complex hull of the object <sup>15</sup>. Importantly, silhouette based approaches are computationally less expensive and can generate a visual hull from a limited set of images.

- In the approaches discussed, using multiple cameras required the calibration of each camera and the establishment of a common frame of reference for each image captured. In addition, specialised hardware is required to implement simultaneous capture from multiple cameras. The extraction of calibration data is essential for an accurate Euclidean reconstruction of the object, although a projective reconstruction can be achieved without the calibration. Furthermore, the use of a single camera, as opposed to multiple camera set-ups, greatly simplifies the calibration process and facilitates the creation of a flexible system, as home-users should not be expected to have several cameras or be required use a multiple camera system.

### 3.6.1 Discussion on Human Modelling

In this chapter, the creation and use of virtual human models has been presented. In particular, they are shown as an important process for enhancing an individuals experience in a virtual world. Human models are used in a wide range of applications, and several techniques exist for the creation of such models. This provides several challenges that are not present in the 3D reconstruction of other static objects. Their importance is also evident in the number of different approaches that are tailored towards the creation of realistic human models.

This chapter also highlighted that numerous techniques have been developed for the extraction of an individual's pose and animation data. This is a challenging task and is not suitable in the development of a flexible automated approach to the creation of human models that can be used by a non-expert user. As a consequence, the individual should adopt a single predefined pose similar to that detailed in (Hilton et al. 1999), for the reason that requiring the user to undergo such movements reduces the flexibility of the technique and to get accurate information requires a multiple camera set-up or the use of a video cameras. Significantly, having a particular pose enables the automation of the reconstruction process.

To develop a flexible approach for the creation of human models for virtual worlds, silhouette based approaches provide the simplest approach to extract the relevant shape information, and in (Leon & Sucar 2000) it has been shown that it is possible to distinguish one individual from another based on information contained in a single silhouette and as stated above this reduces the reliance on the establishment of correspondences. Significantly, the vast majority of techniques used for the extraction of shape information relied on the extraction of the individual's silhouette, although some of the finer detail is lost when the silhouettes are combined. Thus when using silhouette based methods, it is necessary to provide a method that can re-create the fine detail that cannot be reconstructed.

One approach to model of the fine detail combines the silhouettes with an underlying model which can be easily deformed using the shape information in the silhouettes. This provides a

---

<sup>15</sup>The complex hull of the object is defined as the region that is intersected by any line between two points in the object. In situations that the surface contains concave surfaces the complex hull includes regions that are not part of the object.

flexible approach to personalising the underlying model. The shape information can be used to create models of varying level of detail. At the lowest level of detail the silhouettes can be used to simply texture the model and provide models that are suitable for use in application destined for mobile devices. Importantly, combining the captured data with an underlying model permits the use of a predefined number of polygons. Additionally, combining the silhouettes with underlying models makes it easier to integrate the final model into existing virtual worlds or can be readily made so. Thus the models that are created should be consistent with existing standards, including VRML (1997) and MPEG4 (1998). This provides the advantage that existing animation streams can be used to animate the models in the virtual environments. Moreover, in the provision of a flexible system, it is not reasonable to expect an individual to undergo a complete set of movements for complete personal characterisation, and network restriction and gaming environments may force the communication of predefined motions to enable efficient movements.

This review highlights that there is no obvious flexible approach for the creation of realistic human models that can be guaranteed to provide the desired results. The best approach to creating an accurate human model will probably require the combination of a number of the described techniques to ensure that all complexities are extracted. This is beyond the means of most users in terms of software and expertise. However, to obtain greater realism it is necessary to provide further methods that enable the texture (or outer skin level) to take on the movements that would ordinarily be caused by muscle movements. Approaches to achieving this are introduced in Section 3.5.3 but a complete description of such techniques is currently beyond this research. This is perhaps the greatest challenge in providing realistic models, and to do this successfully, a lot can be learned from the research that is being carried out in facial animation to provide greater realism (Kshirsagar et al. 2003) (Lin et al. 2002).

# Chapter 4

## Design Approaches

### 4.1 Introduction

In this chapter, five approaches towards the flexible creation of virtual human models developed by the author for the purposes of populating virtual worlds are presented. In each case, the individual is captured in a real environment. These approaches illustrate how the segmentation of the individual progresses from simple backgrounds to more complex backgrounds. In addition, the personalisation of the model progresses from the simple texturing of a default model to the deformation of this model using the captured image data.

The primary goal in devising a flexible system is that non-expert users<sup>1</sup> will be able to automatically create realistic human models and subsequently simply modify the models. Thus it is imperative that the system is automatic or requires a minimum amount of user interaction.

The two main aspects of this research are interspersed within this chapter. The first is the segmentation of the individual from the background, and the second is the development of a framework that enables the information in the captured images to be combined to create a photo-realistic model of the individual, a so called “virtual twin”. In the development of the system, the number of restrictions that are imposed on the individual are reduced, as are the constraints on the environment and type of camera used in the capture. In addition to this, the system developed must be robust and overcome any occlusions or errors that are introduced to the system during the capture phase. This chapter also highlights the development of the constrained B-spline templates and the validation of silhouette based reconstruction for the creation of photorealistic human models.

As discussed in the previous chapter, there are several photographic based approaches for the creation of human models. These can be loosely categorised as: multi-resolution, multi-camera approaches giving high quality accurate 3D models and low-cost and flexible systems using restricted views to provide photo-realistic 3D models. The author’s approaches described in this chapter fall largely under the latter category, where from a limited set of views, individuals are captured to create photo-realistic human models.

The first technique that is described in this chapter is an initial approach and an extension of the approach of Hilton et al. (1999) to real environments and uses a simplified texturing procedure

<sup>1</sup>In the context of this research a non-expert user or home-user is classified as an individual who is not trained or experienced in the area of 3D model creation, or in the area of image analysis.

to produce photo-realistic models. The second approach facilitates the extraction of the individual from cluttered environments using a constrained B-spline template that is derived from the active contour models described in Chapter 2. The third approach builds on the two preceding approaches and uses facial features in conjunction with the B-spline templates to improve the realism of the human model. Then the fourth approach provides a silhouette based reconstruction technique that enables the individual to generate his or her own models that can be used to incorporate an accurate shape description of the individual. The final approach extends the silhouette based approach and enables the user to undertake small and large scale deformations of an underlying model to approximate the bounding volume created using the silhouette based reconstruction process.

## 4.2 Approach 1: Towards the Creation and Animation of Virtual Humans

This approach, as described in (Boyle 2004), was designed as a low-cost technique for the automated creation of virtual humans using a number of images of a person taken in a uncluttered background. The approach involves the creation of personalised 3D models by combining the captured images with an underlying H-Anim model. Silhouettes of the model are created that correspond approximately to the captured images of the person. These are used to define a 2D-2D texture mapping<sup>2</sup>. The normal vector for each tri-face of the model is used to determine which image is used in texturing a particular face in a 2D-3D mapping. This approach builds on the approach of Hilton et al. (1999) by extending the approach to operate in real environments and extends that of Lee, Goto & Magnenat-Thalmann (2000) by providing an automated approach to extract the individual from the images.

The system is composed of four main elements that form a chain for the creation of virtual humans and is illustrated in Figure 4.1. The next section describes how the silhouettes of the models are generated. This can be carried out without prior knowledge of the human to be captured. Then the capture process is described and in particular how the user is segmented from the background. This is followed by a description of how the 2D-2D texturing is achieved and finally a description of how the normal vectors are used to texture the underlying model.

### 4.2.1 Silhouette Creation

The creation of silhouettes for the model is achieved by placing the model at the origin in a 3D coordinate system. The 3D vertices that form the mesh of the model are projected to a 2D plane  $3m$  away<sup>3</sup>, effectively forming a virtual camera. This is achieved through the use of a camera projection matrix (Sonka et al. 1999, Faugeras 1993). In Equation 4.1, the camera projection matrix,  $P$ , is created by multiplying the scaled orthographic projection matrix by the transform

---

<sup>2</sup>2D-2D texture mapping involves transforming data that is contained in one 2D shape or region to another while maintaining the pixel order in the original shape or region.

<sup>3</sup>A distance of approximately  $3m$  is chosen because in the real-world setup this is the minimum distance from the individual that the camera can be placed to capture the all of the individuals data. At this distance a plane  $2m$  high and  $1.5m$  wide can be projected to the image plane.

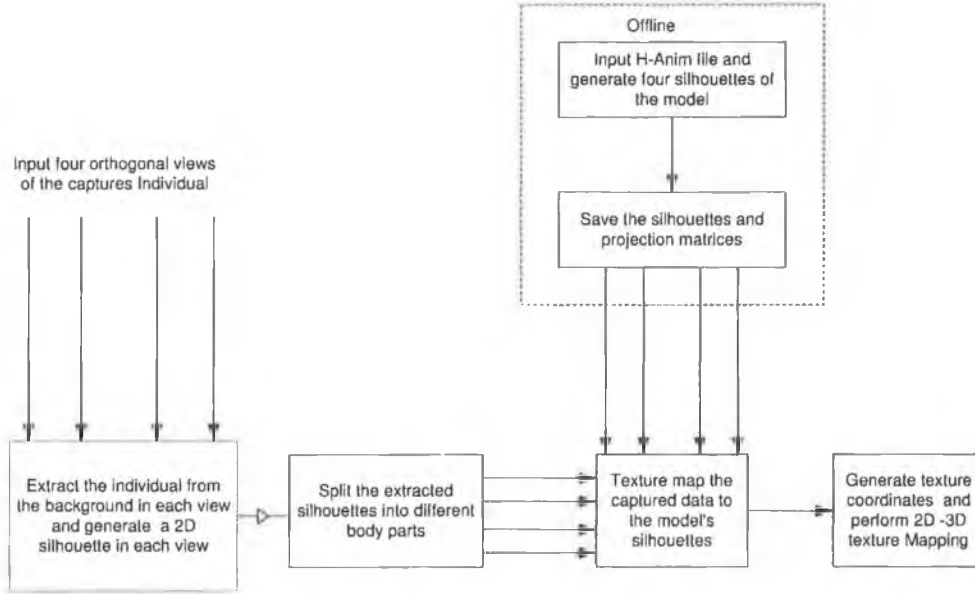


Figure 4.1: Flowchart with the main elements in the system for approach 1.

matrix,  $\mathbf{D}$ , where  $f_u$  and  $f_v$  are the horizontal and vertical focal lengths.

$$\mathbf{P} = \begin{bmatrix} f_u & 0 & o_u & 0 \\ 0 & f_v & 0 & o_v \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix}}_{\mathbf{D}} \quad (4.1)$$

The coordinates  $(o_u, o_v)$  represent the coordinates of the principal point or the image centre. The matrix  $\mathbf{D}$  is composed of a sub-matrix:  $\mathbf{R}$ , a  $3 \times 3$  rotational matrix, and two vectors  $\mathbf{t}$ , a  $3 \times 1$  translational vector, and  $\mathbf{0}$ , a  $3 \times 1$  zero valued vector.

The 2D projections can be represented in VRML (1997) but they require 3D coordinates for their representation, the third dimension being the same for all points. MPEG4 (1998) facilitates the representation of 2D scenes within 3D worlds and vice-versa. This 2D representation can be used to provide a preview in the situation when the user has a number of 3D models for use in an application.

To facilitate the texture mapping, the points are projected to an image plane using a modified camera projection matrix. The parameters of this matrix are determined by the size of the required image. The image centre is located at  $(\frac{image\_width}{2}, \frac{image\_height}{2})$  in pixels. The values for  $f_u$  and  $f_v$  are obtained iteratively by projecting a  $2m \times 1.6m$  plane so that its projection is completely captured by the image and the boundary of the cube corresponds to the boundary of the plane. The values  $f_u$  and  $f_v$  are multiplied by the image width and height respectively to convert from camera coordinates to image coordinates.

A silhouette is created corresponding to each available image. The modified projection matrix is used and the points are projected to the image plane. Though the basic shape of the model is discernable, there is no defined boundary. Thus, before the image is rendered, the information,

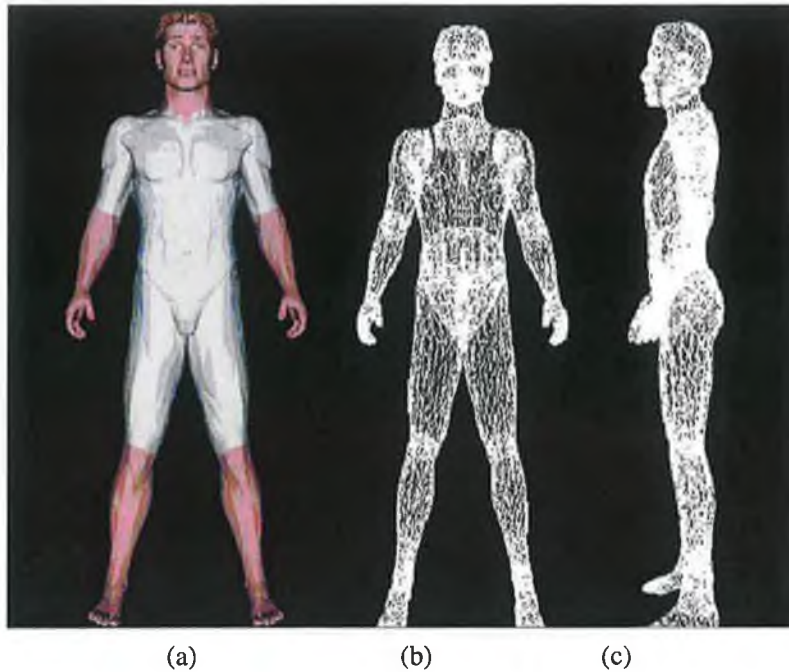


Figure 4.2: The default H-Anim model used for the creation of virtual humans. (a) shows a front view of the model, (b) and (c) show front and side views generated using the projection matrix in Equation 4.1

which is contained in the H-Anim file for connecting the vertices, is used to join the points in the image plane (see Figure 4.2 (b) and (c)).

Restricting the views makes it possible to create the silhouettes once and use them every time the procedure is called. This enables the automatic creation of virtual humans when the individual adopts a set pose because if the camera is in the same position, then the centre of projection of each image to be used is the same. Additionally, it is possible to adjust the pose of the model to approximate that of the individual in a manner similar to that described in (Hilton et al. 1999), although this results in the silhouettes of the model being created each time an individual is captured.

#### 4.2.2 Image Capture

A variable approach to capture the images was adopted to facilitate the capture of images from different devices. The research undertaken by Hilton et al. (1999) used  $756 \times 582$  images giving a resolution of  $40 \times 40$  pixels for the individual's face. This approach uses images with  $640 \times 480$  pixels as the default size because some current webcams, camera enhanced mobile phones and new enhanced phones offer the possibility of creating the images of this size and greater. Figure 4.3 shows examples of captured images. The images contain an individual in a standard pose taken against an uncluttered background. Other approaches that separate an individual from a simple background are discussed in Section 5.3. This pose is essential to allow accurate identification of the body parts. The background is removed by smoothing the image using a Gaussian filter with

standard deviation,  $\sigma = 0.5^4$ , and assuming that the four corners in the image contain background information (Sonka et al. 1999). The Gaussian filter is used to suppress noise in the images and to make it easier to generate a region map. In Figure 4.3 (a), using the four corners of the image will not provide sufficient information to accurately remove the background. In this situation, the boundary of the images is traced and the values along the boundary of the image are used to develop more regions that are classified as background. This background is then subtracted from the image, resulting in a silhouette of the individual. This is repeated for each view of the individual. The resulting silhouettes contain the information necessary to personalise the H-Anim model.

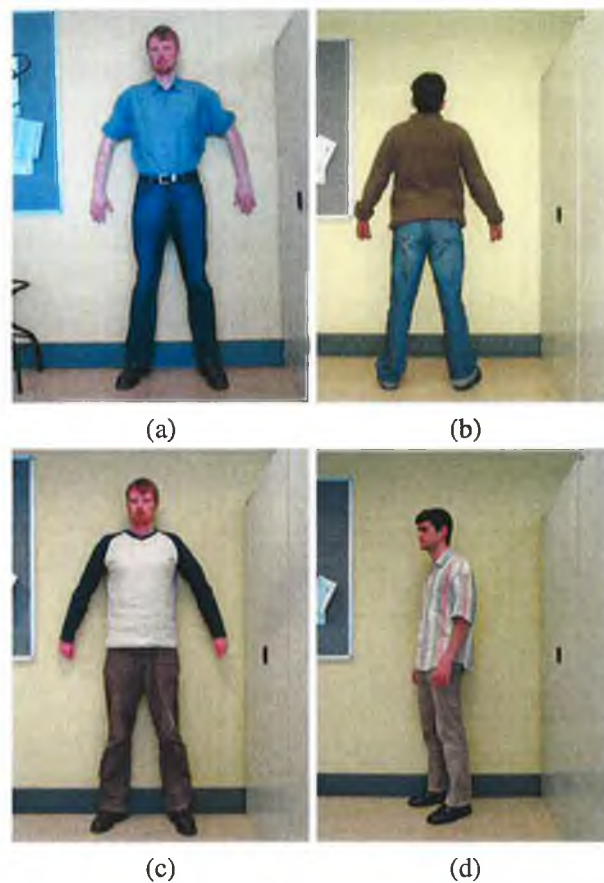


Figure 4.3: Examples of images captured for the creation of the virtual humans.

In Figure 4.4 (b), the different regions in the Figure 4.3 (a) are shown. The application of the Gaussian filter to this image is shown in Figure 4.4 (b). The number of regions is significantly decreased in Figure 4.4 (d). Grouping of these regions can be used to estimate the individual's location. In fact, the number of regions decreases from over 400 to 37 regions. Although the background is not one uniform region in Figure 4.4 (b), tracing boundary of the image, the major regions are connected to the edge of the image and the same colours are observed between the legs. In Figure 4.4 (d), the right arm of the individual is shaded the same colour as the background

<sup>4</sup>Setting the standard deviation to 0.5 is an approximate method and in general is dependent on the image and can not be reliably automatically set. In addition, it is not expected that the home-user should have to set  $\sigma$ .

beside it. This occurred because the number of regions is unknown prior to the segmentation and thus random colours are used to distinguish the regions visually.

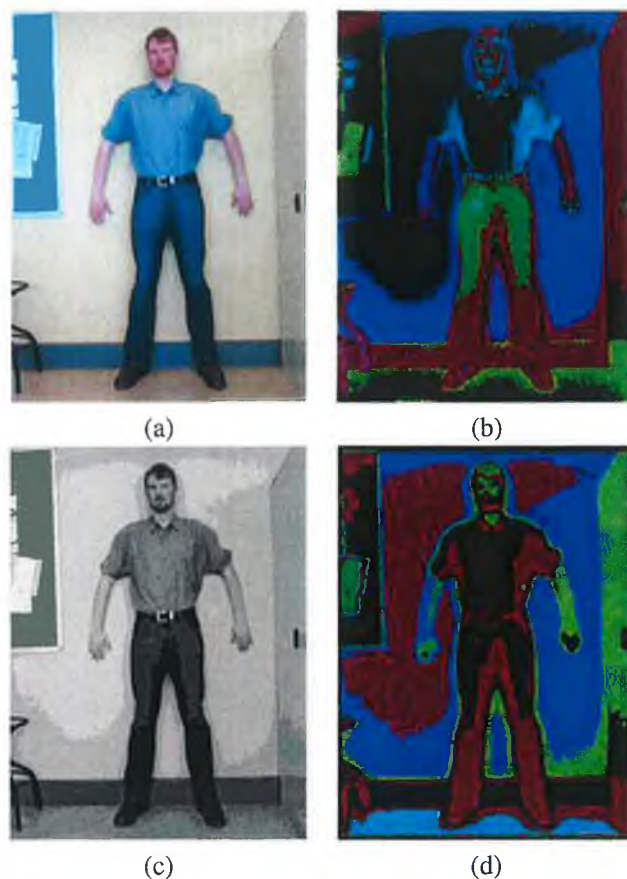


Figure 4.4: Illustration of the effects of applying a Gaussian filter to the images. (a) shows the original image, (b) shows 402 regions in the original image, (c) shows the Gaussian smoothed image and (d) shows the 37 regions in the smoothed image.

### 4.2.3 Texturing

The front and back views of the individual are used to establish correspondences between the model silhouettes and the individual. This is based on the algorithm presented in the paper by Hilton et al. (1999). This algorithm forms an important part in the identification of key joints. Using the position of the joints and the key features, the main components of the body are identified in the back and front views.

The 2D-to-2D mapping is carried out using two images, one of which contains either a body part extracted using the feature extraction (Hilton et al. 1999) or a complete image submitted by the user, for example a side view. The second image contains the equivalent model silhouette for example. The texture mapping uses scale factors to ensure that the information is mapped accurately. While the vertical scale factor is calculated based on the height of the two images, the horizontal scale factor is calculated for each row in the image. This ensures no background information is mapped. Thus the horizontal scaling factor is defined in terms of the number of



pixels in a particular row that are not background elements, and the horizontal index at which the first non-background pixel is encountered is stored. This 2D-to-2D mapping is illustrated in Figure 4.5 for the left arm using the front view and the combined textures for the front and back views are shown in Figure 4.5.

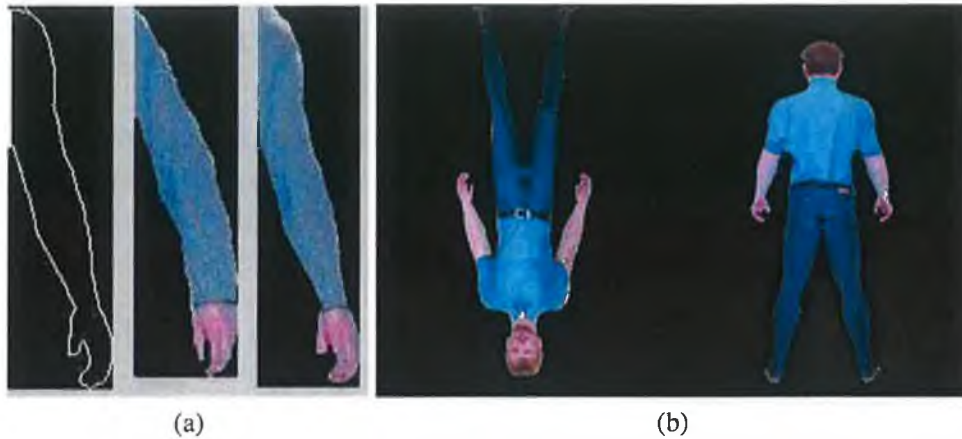


Figure 4.5: 2D to 2D texture mapping of the body (a) shows the image data of the arm is mapped to the arm silhouette of the arm. In (b) the combined data is shown for the front and back views.

When the 2D-to-2D texture mapping is complete for each body part in the front and back views, the individual components are recombined to form a complete textured silhouette for the back and front views. Then these are combined with the side views into a single image. This provides an alternative to producing a 3D image<sup>5</sup> of the individual images. This is essential, as VRML only has a single field for specifying the texture coordinates. In generating texture coordinates, each sub-image is accessed through an offset equal to the sub-images width. The selection of the images and the generation of the texture coordinates, are determined by the normal vectors. The normal vectors are created so that they all project away from the tri-faces of the mesh. Thus, the tri-face of the model with a normal vector projecting in the direction of a camera centre uses the image produced from that viewpoint. In a real scenario, the normal vector may project between two or more cameras. Equation 4.2 is used to determine which image is used to texture a tri-face. This equation is derived from the vector dot product with  $\mathbf{u}$  and  $\mathbf{v}$  nonzero vectors and  $\theta$  the angle between them satisfying  $0 \leq \theta \leq \pi$  (Anton 1994).

$$\cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad (4.2)$$

In Equation 4.2,  $\mathbf{u}$  represents the translational vector of a camera centre and  $\mathbf{v}$  represents the normal vector for a tri-face. For each tri-face, angle  $\theta$  is calculated for each camera centre and the minimum value of  $\theta$  establishes the image used to texture the tri-face. Thus for each tri-face, a value indicating the camera centre is stored. Using this approach, it is possible to texture the complete model using two images, provided that the angle of separation between the two cameras is sufficiently large to give good coverage of the individual, for example using the front and back

<sup>5</sup>The 3D image is equivalent to the integrated texture map described in (Hilton et al. 1999).

views. In addition, it may simply be adapted to use any number of cameras. This principle is illustrated in Figure 4.6. In this figure the normal vectors project away from the faces of the object and closest camera centre is determined by Equation 4.2

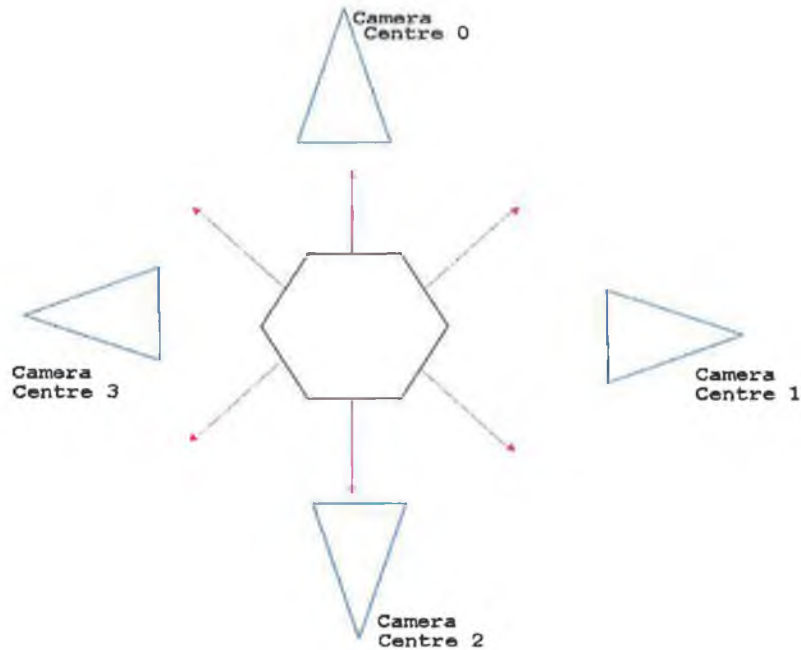


Figure 4.6: This figure illustrates the role of the normal vectors have in determining the image that is used to texture a face of the model.

The texture coordinates are generated by projecting the 3D vertices to a 2D plane using the modified projection matrix. The 2D coordinates correspond to points inside the silhouettes on the reconstructed texture image. Then depending on which projection matrix is used the 2D points are translated to correspond to their location in the combined image. The 2D coordinates are then normalised by dividing the  $x$ -coordinates by the effective width of the combined image and the  $y$ -coordinates by the effective height because texture coordinates are normalised in VRML. The final textured model using the front and back images in Figure 4.5 is shown in Figure 4.7.

#### 4.2.4 Issues Highlighted in the Approach

- In this approach, it can be seen that it is necessary to improve the realism of the model and that the clarity of the face is important in the recognition of the model in the real environment. To practically use this approach, it is imperative that the face is textured with a minimum of images to ensure that the photo-realism is maintained.
- The extraction process that is described only permits the individual to be extracted from the environment when the background is non-complex and consisting of very few features. Thus a more general approach to the segmentation of the individual from the background is required. If the same underlying model is textured, then all the individuals created will have the same shape in the virtual worlds. A possible alternative is to have a selection of models available to the user. The user can then choose the model that best matches their shape and



(a)



(b)

Figure 4.7: Example of the textured model using the 2D texture map in Figure 4.5. (a) contains three views of the static model textured with the captured data for the sequence shown in Figure 4.3, (b) shows three views of the model in (a) as it is animated using a walking sequence.

then use this to create the corresponding 3D model.

- The advantage of this method is that once the silhouette information is extracted and textured to the final model, the model can be easily animated and this enables the incorporation of the models into virtual worlds.
- The use of different templates or the modification of the templates in response to the input data may be more realistic if different models are used to model different individuals and take account of differences in clothing, hair and body shape.

### 4.3 Approach 2: Creating Active B-Spline Templates

This approach is an important stage for the automated extraction of the individual from their environment for the creation of virtual humans for use in virtual worlds. This method uses the same capture procedure as in Approach 1. Then constrained deformable B-splines templates are used in each view to automatically extract the user from a real environment. While minimising the snake's energy, a skeleton is automatically incorporated into the model.

Firstly, the image capturing process is described detailing the pose that the individual should adopt and how this influences the development of the templates. Then the process of extracting the user from a real environment is described, and this includes a description of the B-spline templates and how they are generated, constrained to accurately extract the users shape/silhouette and automatically fit the skeleton to the model. A system overview is shown in Figure 4.8. This figure shows the complete system for the initialisation, the fitting of the templates and how the extracted image data is combined with the underlying model to provide a personalised human model.

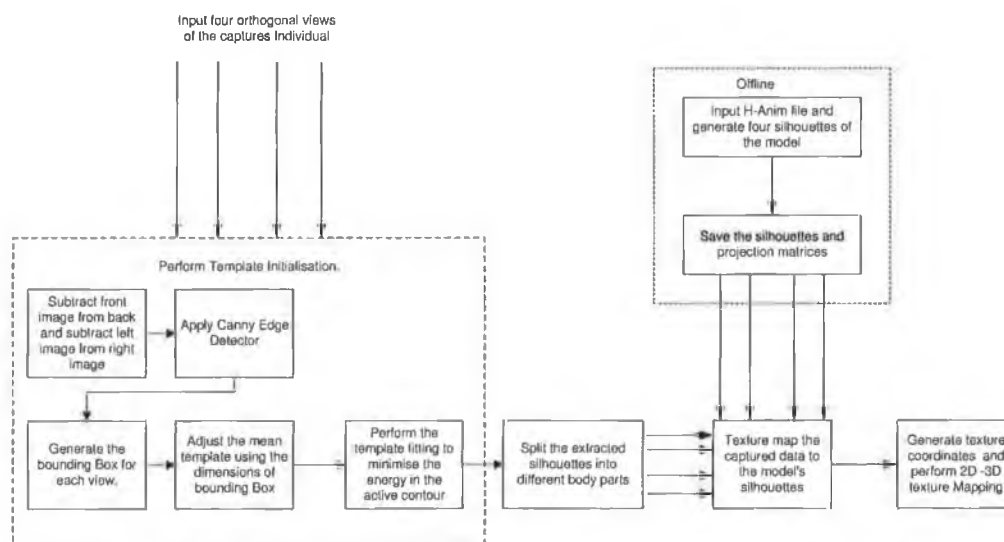


Figure 4.8: Flowchart with the main components of Approach 2.

#### 4.3.1 Image Capture and Definition of the Individual Pose

The image capturing process is implemented in such a manner that does not impose strict constraints on an individual. The process involves capturing four images of the individual. Each image has a resolution of  $640 \times 480$ . The images are captured with a single fixed low-cost off-the-shelf camera, and between the capture of each image the individual rotates  $90^\circ$ . This enables the capture of the four orthographic projections which contain sufficient information to create a realistic human model. This capturing technique is based on that proposed by Hilton et al. (1999). In each situation, the individual stands approximately  $3m$  from the camera. An alternative approach, as adopted in (Lee, Goto & Magnenat-Thalmann 2000) allows the user to specify the distance

between the camera and the individual or else indicate an approximate height of the individual in the images because it is felt that, in certain situations, valuable information that adds to the realism is lost when the individual does not occupy a significant portion of the image, for example when capturing the images of a child.

In the approach described in (Hilton et al. 1999), the images are captured against a photo-reflective blue screen backdrop enabling easy segmentation of the background using a chroma-key technique that identifies the background pixels based on the percentage of blue in each pixel. In our approach, the images are captured against a real background. The segmentation of the individual from the background is described in Section 4.3.2. Our method constrains the individual to occupy the centre of the image in each capture. Even in the simplest of real-world situations, capturing a complete image of an individual will additionally capture the floor, wall and other elements that appear in the background.

In order to reliably extract and locate particular features, the person should adopt the pose similar to that shown in Figure 4.9 (hereafter referred to as the standard pose). This pose requires the individual to stand with their feet apart and with their arms raised. This ensures that the armpits and the crotch area can be easily (accurately) located. Another requirement that has become apparent from initial testing is that the individual should look directly at the camera and if possible a little above the camera. This is to ensure that all the features of an individual's face are visible. This is important because the face is a highly detailed region and can significantly enhance the realism of the model, and in the side views it is used to gain more defined profile information. In the side views, the individual is required to keep their hands at their side and to ensure that the hands are not in front or behind the rest of the body, as this will affect the silhouette of the individual.

To get an accurate model it is important that the individual wears tight fitting clothes. the tighter the clothes the closer the final model will approximate the true form of the individual. This is in part justified, since when the final model is animated the effects of the animation will be more evident on the clothing than on the body itself, i.e. if the individual moves then we see the movements through the clothes. This is important as the templates should be flexible enough to deform to the particular body shape of every individual. The possibility of creating models of the individual wearing other clothes is discussed in Section 6.4.2.

### **Rational for Pose Restriction**

As described in Chapter 3, the determination of the pose of any object, even an inarticulate object, is difficult from a set of images, moreover from a limited set of images. Thus in developing a sufficiently flexible system that can be used by a non-expert user, it is necessary to:

- either restrict the pose of the individual,
- constrain the environment (including lighting),
- constrain the movement of the individual,
- use additional cameras to capture numerous images from a large number of viewpoints,

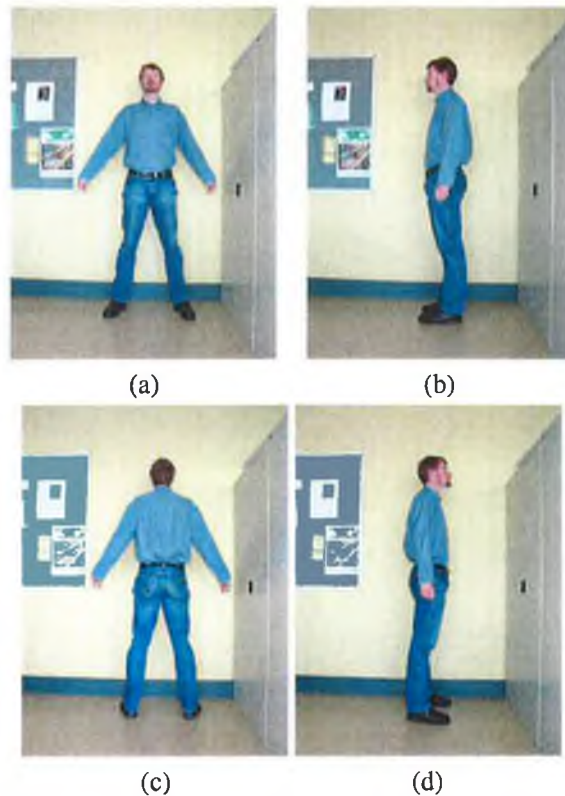


Figure 4.9: A set of captured images for an individual.

- rely on significant interaction to locate the joint and associate the features to a default view of a similar object,
- or enforce a combination of these constraints.

Using the capture set up described previously, if it was permitted that the individual could adopt an arbitrary pose, then the hands or arms could possibly occlude vital information, for example if the arm was across the face or behind the individual. If the hand was in front of the body, then it would be subjected to a different projection resulting in disparities and scaling problems. In addition, the difference in pose between each capture would make it extremely difficult to establish correspondences between the silhouettes that are captured. Thus restricting the pose ensures that the template can be generally defined with sufficient flexibility to automatically extract the individual in any environment, and using a standard pose, it enforces scaling constraints on the model as all the body parts that are identified in the front, back and side views can be assumed to undergo approximately the same projection to the image plane.

Using a standard pose reduces the variation in the position that feature points have relative to each other. The alignment of the individual silhouettes can be achieved by the application of the method introduced by Cootes et al. (1992) and detailed Section 2.7.2. In addition, if the pose in Figure 4.9 (a) is adopted the variation in position of the key features identified in Figure 4.11 is significantly reduced. The position of the arms is the only exception, but this is addressed in Section 5.3.4.

The variation in the pose is low and limited to the change in position of the arms. However, in the initialisation processes involving the use of the bounding box or the user-assisted approach the location of the arms are identified, this facilitates the adjusting of the template. In addition, the key features are the main points and the intermediate points identified when extracting the individuals silhouette are all relevant to the location of the key points. Thus using either of the initialisation procedures, the location of the head, feet and arms are known. This is sufficient to initialise the template and reduces the need to establish the modes of variation.

### 4.3.2 Template Generation using Active Contours

Requiring the individual to adopt a standard pose means that the variation between captures of different individuals will be small, and this facilitates the use of templates that are well suited to the extraction of objects that undergo small variation from the mean shape. The templates that are used to extract an individual's shape are derived from the active contour models and are implemented as active B-spline templates (Boyle & Molloy 2005a). The decision to use B-splines results from the fact that they offer greater local control than the original spline based snakes and that B-splines are in common use in computer graphics.

The use of the active contours is considered because it gives the template the ability to adapt to any contour and can reliably and simply describe any boundary. Moreover, it can define boundaries when sufficient edge information is not available. This is essential as the boundary contour is highly complex. In addition, defining the initial contour as a template enables the inclusion of constraints that can control how the snake evolves and ensure that the template is modified in a suitable manner.

Splines are ideal for defining the contour that describes the shape of an individual because unless they are severely stressed, they can maintain second order continuity. Splines can be defined mathematically as continuous cubic polynomials that interpolate a number of control points. The polynomial coefficients for natural splines are dependent on all  $n$  control points. Thus, changing the position of one of the control points affects the entire curve and involves operating on an  $n + 1$  by  $n + 1$  matrix (Foley et al. 1990, Piegl & Tiller 1997). B-splines have been chosen because they consist of curve segments that are only dependent on a few of the control points. This offers an enhancement over the use of splines, as the movement of an individual point does not require the modification of the complete curve (template) and thus greater local control is achieved over the deformation of the template. In addition to this, it makes the B-spline templates suitable for real-time applications (Blake & Isard 1998) and for animation.

#### Generation of the Mean Template

The mean templates are generated initially by taking a set of images of 10 distinct individuals and interactively fitting a B-spline contour to define the basic shape of the individual in each view. This process is similar to the training sets used by Cootes et al. (1995) to find the mean position of points. Examples of the images captured are shown in Figure 4.10.

The fitting of the template to the image information is carried out in a two-stage process. Initially, a user will place the main control points around the individual in the images. Then the

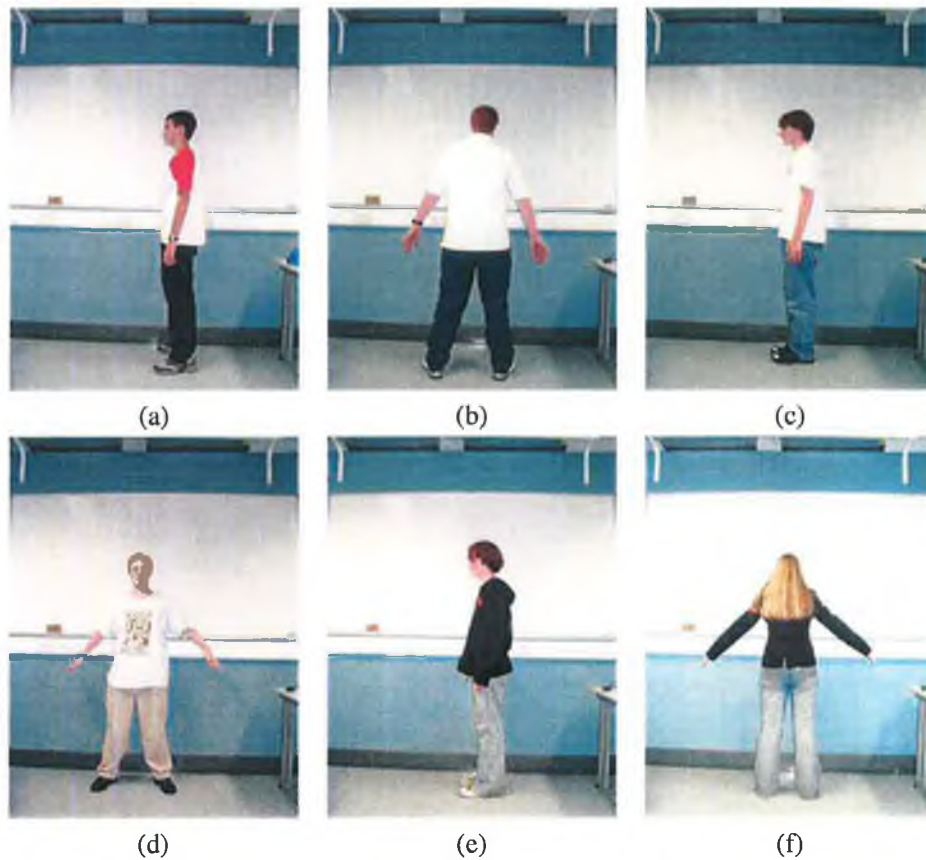


Figure 4.10: Examples of the captured images used for the generation of the template.

control points are automatically interpolated with a B-spline curve. This is performed to permit the user to determine how well the template fits the captured data. The contour is closed to ensure that continuity is maintained at the ends. At this stage, the user has the option to modify the template by either moving or removing the existing control points or adding new control points. The second stage involves generating a contour using the final positions of the user positioned control points, and using the energy minimisation process to ensure that the control points defining the template are on the correct edges in the image. This is completed in each of the four views for an individual. The user has the option to use the front template on the back image and interactively adjust this to fit the template to the image information, or alternatively, the user has the option to use previously generated templates and adjust them to fit the shape of the captured individual.

Following this, the contours are examined to enable the identification of important features, including the minimum and maximum values for the horizontal and vertical control points. This is used to define scaling factors and in the normalisation of the templates. Other key features identified include the armpits, the crotch, the hands, feet and the centroid, which are important landmark points used in aligning the templates and in fitting the skeleton to the final model. These points are identified in Figure 4.11. Identifying these points in each set allows a mean template to be created.

Unlike the process described in (Cootes et al. 1992), when the user is specifying the control



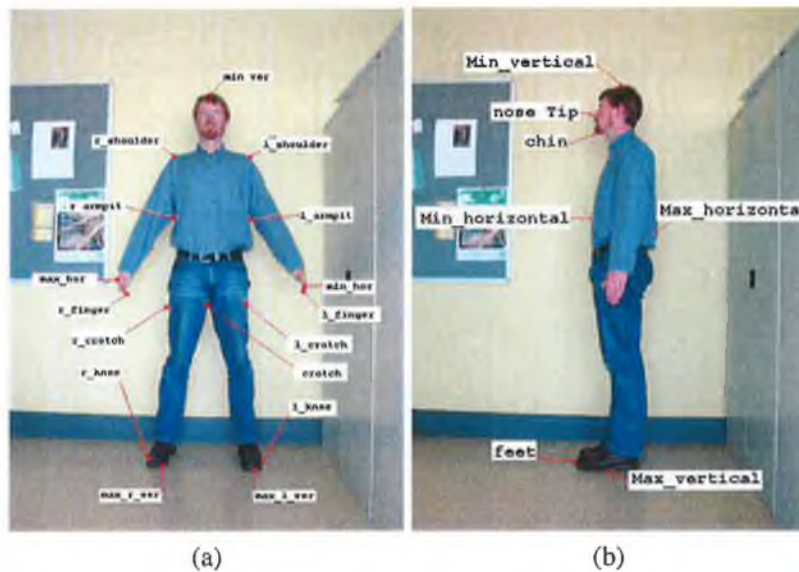


Figure 4.11: The key points that are identified by automatically examining the B-spline contour silhouette. (a) shows the key points identified on front and back views and (b) shows the key points found on the side views.

points to describe the shape of the captured individual the number of control points used is not predetermined. This is necessary to account for variations in the shape of the captured individuals. In (Baumberg & Hogg 1994), when the contour is initialised, a set of control points are equally spaced around the boundary of the individual, which is possible since the variation of the silhouette over a number of frames is sought and not a description of the individual's shape. However, the key features identified in Figure 4.11 have a fixed position relative to each other, thus it is not possible to evenly position the control points along the contour. The generation of the mean template is controlled by the key features and not the points between them. In addition, establishing the key features permits the initialisation of the template. The various approaches to the initialisation and establishment of the key features are discussed in Section 4.3.3. These approaches show that it is only necessary to scale the generic template for its correct initialisation.

The mean template is shown in Figure 4.12, it shows the key features, the controls and an approximation of the skeleton that can be generated based on the key feature locations.

In completing this task it was observed that the greatest variation in the template structure was the position of the arms. This can be seen in the difference between Figure 4.10 (b) and Figure 4.10 (f). Thus one of the most important aspects in the placing and initialisation of the template involves locating the arms. In general, the same template is used for the back and front views, while in the side views the same template is used although it is flipped about its central axis to reflect the difference in pose.

### 4.3.3 Template Initialisation

The initialisation of the template (active contour) is one of the most important aspects to ensure that the minimisation process converges to the correct solution. In (Kass et al. 1987), the positioning of

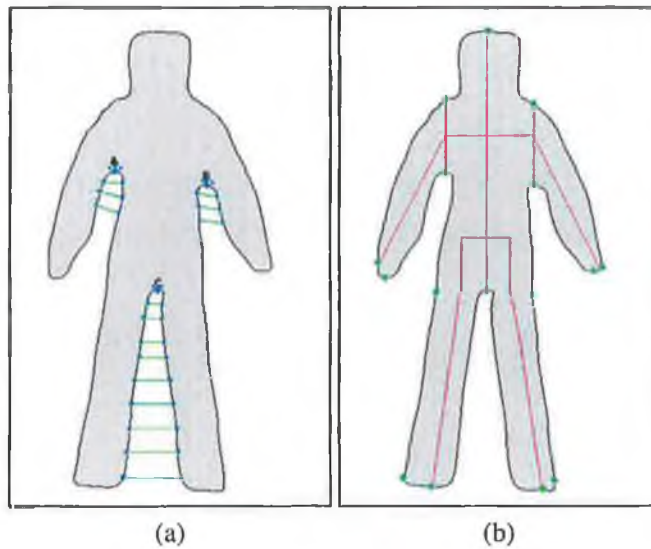


Figure 4.12: The mean template generated. (a) shows the mean template with the control under the arms and between the legs. (b) shows an approximation of the skeleton that is automatically fitted to the skeleton data.

the snake relied on an expert user to position the snake close to the object (region) of interest and automated positioning of the contour was not considered. Since then, different techniques have been developed for the initialisation of active contours and in general are application specific as discussed in Section 2.3.1.

In this section, different approaches to the initialisation of the template are described and discussed as to their applicability to different types of captured image data. The first approach that is developed is a user-assisted initialisation of the template. The second uses an edge map to enable the positioning of the template based on the centroid of the edge information. The third approach employs the subtraction of the captured views to define a difference map that has the additional advantage that the template can be simply scaled. A comparison between the different approaches is then presented. Finally, other approaches that were considered are detailed including: using the difference between frames in a video sequence, to initialise the template, or the use of face detection techniques to isolate the face.

### User Assisted Initialisation

Allowing the user of the system to initialise the template in each view is the simplest case to consider and, provided that the user specifies the key points accurately, then the initialisation will be successful. To enable the correct initialisation the user must specify the position of the top of the head, the feet, the hands, the armpits and the crotch. This information is sufficient to permit the additional points to be interpolated. This process is illustrated in Figure 4.13.

The mean template is initially aligned by calculating the centroid of the points identified by the user and positioning the centroid of the mean template at that location. The mean template is scaled vertically, based on the difference between the top of the head and the feet, and scaled horizontally in two stages. The first horizontal scaling is based on the position of the armpits. This is used

to scale the horizontal dimension of the body and legs of the template. Then, depending on the relative position of the hands to the armpits, the arms will be scaled and positioned appropriately.

Figure 4.13 shows the initial positions that the individual has selected for the features and the template generated based on the user selected points.

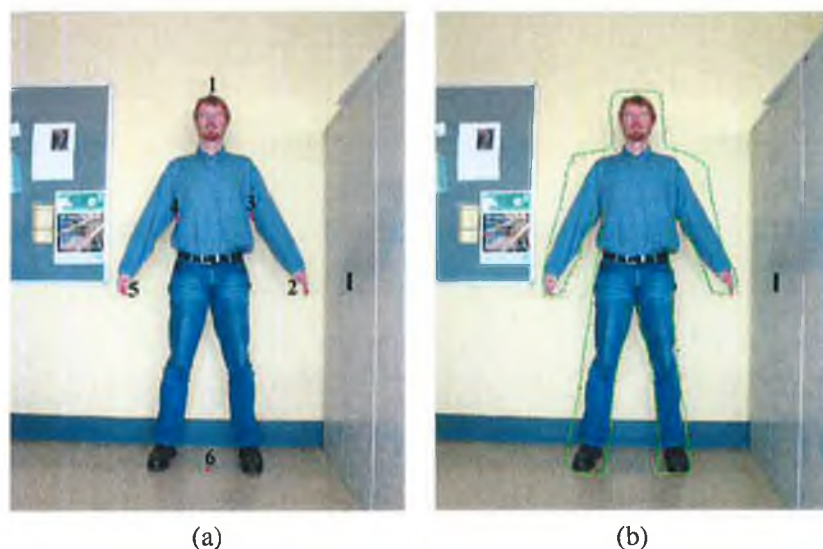


Figure 4.13: The initial positions that the individual has selected for the features.

In the side views, the side template can then be initialised by the user in one of two approaches. The first involves the specification of the key points at the top of the head, the feet and the left and right most points on the individual's body. These points are indicated in Figure 4.11 (b). These points can then be interpolated using the mean template. This template is easier to initialise as its silhouette appears as a simple object. The second approach involves selecting an approximate centre of the individual and then specifying another point either the maximum or minimum vertical value.

### Initialisation Using Edge Information

This approach involves the use of the Canny edge detector to generate an edge map of the input image (Canny 1986). The edge information can be invaluable in defining different structural elements within the scene and also in guiding the active contour. Although, in certain situations, the number of edges can outweigh the information that is in the edge map (Ballard & Brown 1982). Thus before the edge information is calculated, a smoothing technique, such as a Gaussian filter used in Section 4.2.2 is used to smooth the image. This is inherent in the Canny edge detector (Canny 1986). This reduces the effects of weak edges in the image. The centroid of the edge information in the image acts as a centre for the positioning of the template, although it does not provide sufficient information to scale the template to fit the image data. In this case, it requires the user to drag the key points into position. This approach was tested on several types of images, but in the situations where the background clutter was considerable the quality of the initialisation was reduced. This is illustrated in Figure 4.14, where the number of edges in the image makes it difficult to isolate the exact shape of the individual in the image. The effects of changing the

parameters of the Canny edge detector are also illustrated in Figure 4.14. This illustrates a need to have a measure to determine the level of clutter to set the parameters of the Canny edge detector. Such a measure is discussed in Section 5.3.

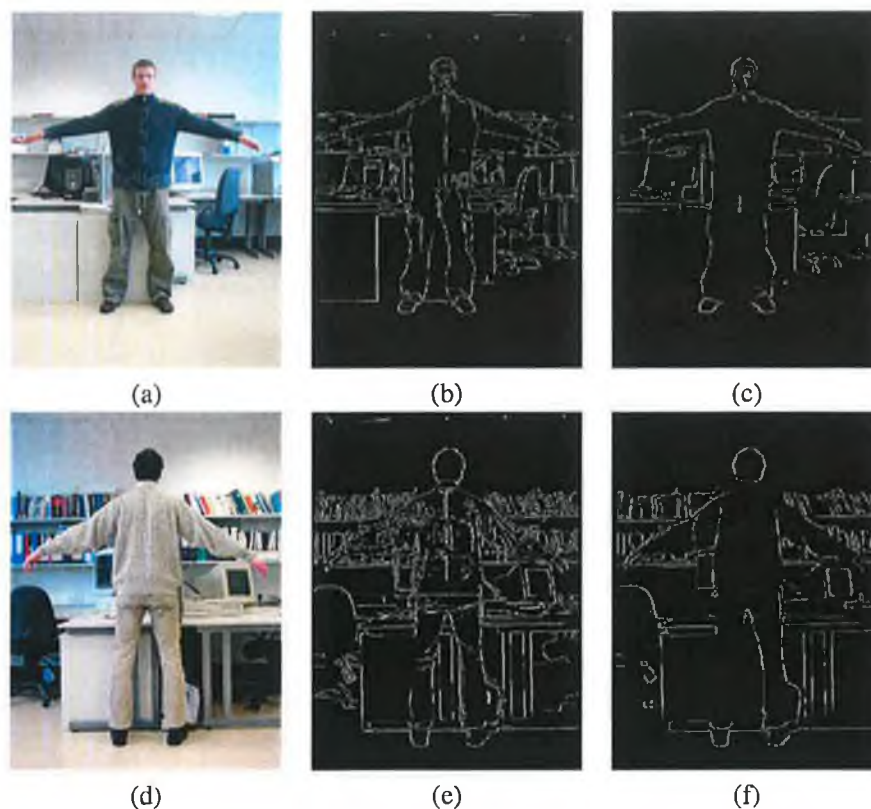


Figure 4.14: Effects of applying the Canny edge detector to images capture in a cluttered environment Canny (1986) . (a) and (d) show the original images, (b) and (e) illustrate the edges generated when the Canny edge detector has  $\sigma = 1$  and the lower threshold,  $T_1$  set to 100 and the upper threshold set to 255, (c) and (f) illustrate the edges generated when the Canny edge detector has  $\sigma = 2$  and the lower threshold,  $T_1$  set to 100 and the upper threshold set to 255. The images illustrate that when significant number of edges are extracted that it is difficult to initially position the template.

### Automatic Initialisation using Difference Map

In this approach, the template is initialised using a subtraction technique in which the back and front images are subtracted from each other and similarly the side views are subtracted. Background subtraction techniques using the capture of an additional image were also considered. Some of the tests associated with background subtraction are described in Section 5.3.1. There are two main reasons that the use of another image was not considered. The first relates to the fact that the individual casts a shadow in a real environment, meaning background subtraction will not accurately extract the shape of the individual. The second reason is that it was not considered that an individual should capture an additional image. Since the individual is positioned in the centre of the image and that the camera is assumed to be in the same position for each capture, the

background will have high correlation, and thus when subtracted, the areas in the image with the greatest difference relate to the individual. This enables the positioning of the bounding box that facilitates the horizontal and vertical scaling of the template.

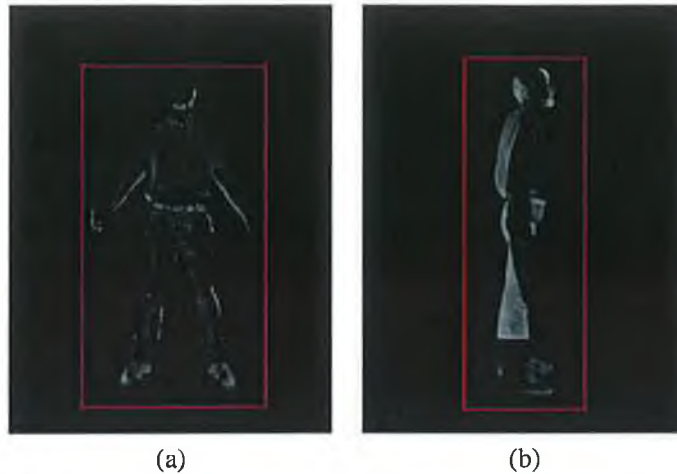


Figure 4.15: (a) shows the front and back subtraction that produces the difference map and (b) shows the difference map produced for the subtraction of the side views.

The initial bounding box (100 pixels by 300 pixels) is positioned using the centroid of difference pixels in the subtracted images. This box automatically expands independently in four directions to encompass the area of the difference pixels. Apart from scaling the template, the bounding box provides additional information for the positioning of the arms. In particular, the left and right sides of the bounding box correspond to the extremes of the arms. The final positions of the bounding boxes are shown in Figure 4.15 (a) and (b). Thus using this information, it is possible to modify the position of the arms in the templates and thus accurately initialise the template.

To improve the minimisation process and to reduce the effects of unrelated edges, the edge pixels in the edge image that lie outside the bounding box are negatively weighted to encourage the contour to move to the correct solution. This reduces the effect that these edges have on the minimisation process. It is important that the edge information is not completely discarded, because if the bounding box is not accurately placed, then it is possible that some vital edge information will be located outside the bounding box.

### Comparison of Initialisation Techniques

The images in Figure 4.16 show the automatic and manual positioning of the templates. In Figure 4.16 (a) and (d) the template is manually positioned, and in Figure 4.16 (b), (c) and (e) the template is automatically positioned. The main difference is in the localisation of the hands, which is highly dependant on the colour of the background and may not be accurately determined using automated initialisation. Furthermore, in the manual fitting, the contour fits the data more accurately, this because the key features are clearly identified and thus the contour is better scaled to approximate the captured data.

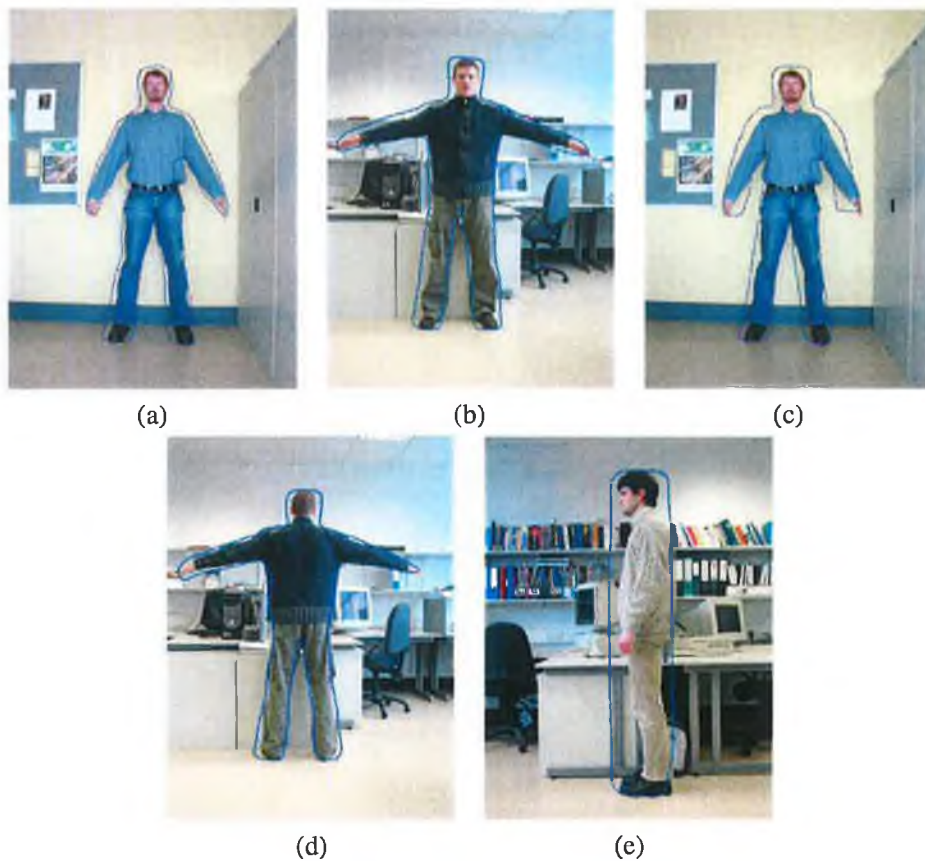


Figure 4.16: (a) and (d) show manual fitting of the template to the front view, (b) and (c) show automatic fitting of the template to the same images and (e) shows the automatic fitting of the template a side view.

In the side view shown in Figure 4.16 (e), the initialisation process is simpler because the individual stands with their arms at their side and initialisation is determined by the four extremes, the head, maximum and minimum horizontal coordinates. The manual and automatic initialisation produce identical results for the side views.

### Alternative Approaches to Initialisation

In addition to the approaches described in this section, alternative approaches to the initialisation of the template have been considered.

- Hilton et al. (1999) propose the use of a photo-reflective blue screen, although the use of the blue screen on its own does not eliminate problems associated with shadows and thus the requirement of special lighting is needed to enable the individual to be accurately segmented from the background. This was not considered, as it is not practical in a flexible system designed to operate in a home environment.
- One alternative approach in real environments is to use a face detection method such as described in (Cooray & Uscilowski 2004), which can provide an accurate location of the

face within the frontal images and possible estimates in the side images. This approach would not provide sufficient information in the back image but the initialisation in the front image could be used to initialise the procedure in the back image. Although, skin detection algorithms can be used to provide information on the location of the hands in the image, it is also highly dependant on the clothing worn by the individual. This could cause difficulties for initialisation, e.g. if the individual was wearing short trousers or had a short-sleeved shirt.

- Motion-tuned active contours offer the possibility to reliably locate the individual in any environment by capturing the individual and the slight movements that he or she either consciously or unconsciously makes over a few seconds using a video camera. An approach that extracts the boundary of an individual using a video sequence is described in (Bompis et al. 2005). The idea of using an image sequence for the extraction of an individual from a real environment has been previously presented in (Baumberg & Hogg 1994). In particular, it was used for the tracking of an individual and not for accurate extraction of an individual's shape information. This approach attempts to extract the complete shape requiring the whole body to move during the capture phase. If the template previously described is used in conjunction with the motion data it will be possible to accurately position it even if complete motion data is not available. This approach was not presently considered because only still images are used, but it is discussed in greater detail in Chapter 6.

#### 4.3.4 Constraints to Control the Evolution of the Templates

The initial template that is used for the front and back views is shown in Figure 4.12. The inclusion of the constraints was considered necessary after initial testing of the automated fitting process in the front and back images because the control points moved towards the closest edge. This was particularly evident under the arms and between the legs where some control points defining the template converged to the same edge depending on how the templates are initialised. This is shown in Figure 4.17.

In the case of the automatic initialisation of the contour, it is impossible to know the location of specific control points relative to the edges in the image. Thus constraints are introduced between the legs and under the arms. These constraints take the following form:

- The direction in which the control points between the legs can move is limited, i.e. the control points were forced to move away from each other. Rather than enforcing a direction in which the control points can move, it was initially considered that the control points could not come within a certain distance of each other, relative to the initial distance or the inclusion of a volcano<sup>6</sup> between the legs and under the arms (Kass et al. 1987). The simplest and most effective approach involved the introduction of a repulsion force between the nearest control point on the opposite side of the template.
- A second linearity (or approximately linear) constraint was introduced to ensure that the control points between the legs and under the arms moved uniformly towards the correct

---

<sup>6</sup>A volcano as introduced in (Kass et al. 1987). Volcanoes act as a repulsion force between a point on the image at a distance from a point on the snake. The larger the value of the peak of the volcano the stronger the repulsion force.

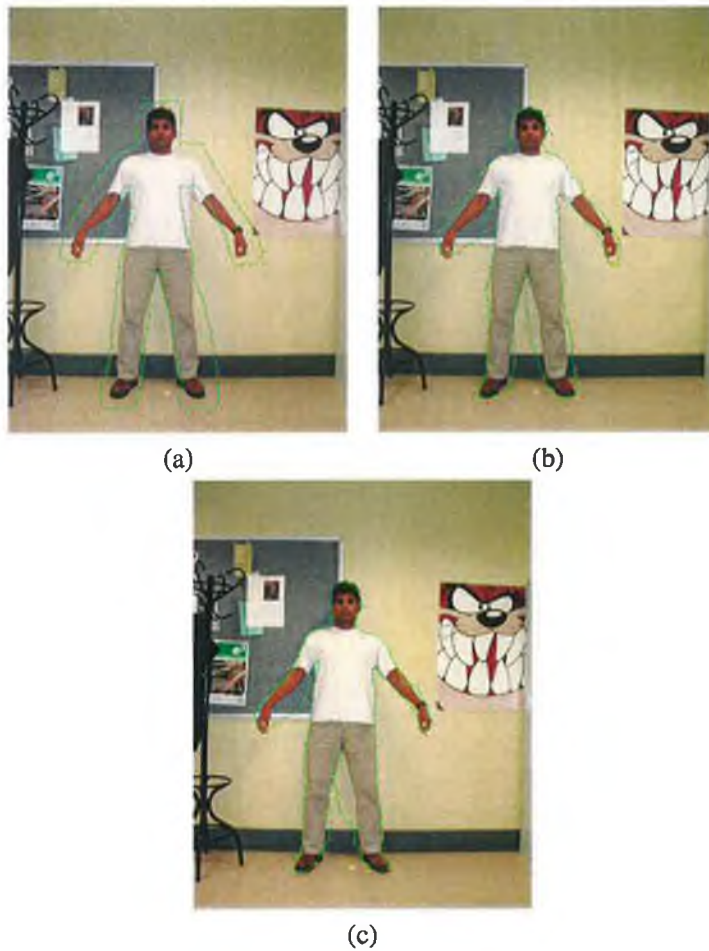


Figure 4.17: The incorrect convergence of the template control points. In (a), the initial position of the template is shown. In (b), the position of the contour after 20 iterations is shown, it can be seen that under the right arm and between the legs the contour is converging to the same edge. In (c), the situation is shown after 60 iterations.

solution and, in some cases, speeded up the convergence process. The linearity constraint involves taking a series of control points under the arm and between the legs and examining the slope of the line that passes through these control points to establish so as to determine if the points are converging to the same edge or if one of the control points is trapped in a local minimum.

The linearity is also used to validate the location of the armpits and the crotch as at this point, the slope of the lines should change dramatically, and if the control point identified in the template to correspond to this feature is not positioned at an effective point of inflection, then a valid location of these points is not considered to be found.

These constraints are shown in Figure 4.18.



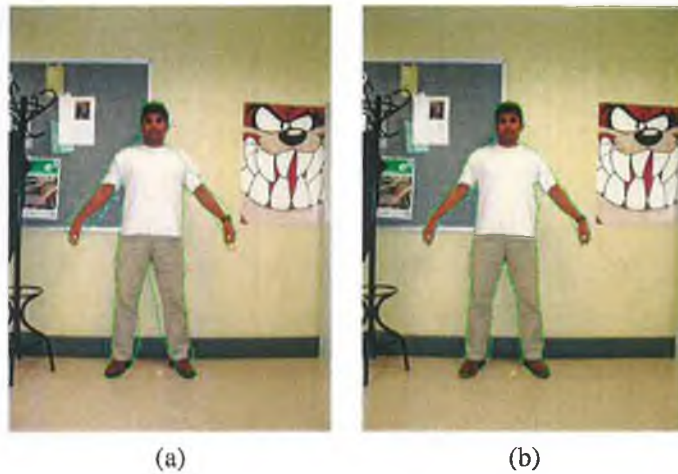


Figure 4.18: Starting with the contour after 20 iterations shown in Figure 4.17 and for the purpose of demonstrating the effects of the constraints, the constraints are manually introduced between the legs and under the arms by selecting each node and changing the direction that the control points can move. (b) shows the effect of introducing the constraints after a further 20 iterations.

#### 4.3.5 Minimisation of the Templates Energy

The energy minimisation paradigm defined for active contours is ideally suited to the problem of fitting the templates to image features. In particular, active contours enable the inclusion of a variety of external energies to control how the contour evolves in the presence of specific features. The final version of the contour also ensures that even when an edge or other boundary information is missing, a complete boundary description of the individual can be obtained and the use of control points makes for efficient storage of the final contour.

As described in Section 2.2.2, there are a number of approaches to the minimisation of the energy in the active contour. In the original paper by Kass et al. (1987), the minimisation was achieved using an iterative technique. At each iteration, the implicit Euler steps with respect to the internal and external energy are taken. An alternative to this procedure, which is adopted in this research, is the dynamic programming approach of Amini et al. (1990). This method is guaranteed to find the minima within a predefined search space. The search space about a particular control point is used to define the possible positions that the control point can move to. The search space is specified as a  $3 \times 3$  grid around a control point. In addition to this, an extra one dimensional search space for the external energy calculations is introduced. This search space is defined perpendicular to the slope of the line passing through the two neighbouring control points of the current control points<sup>7</sup>. The edge feature map is examined along this line to find the closest edge. It is possible to set the size of this search space and in practical situations it is set to a limit of 10 pixels from the current control points. Any edges outside of this range are considered to have no influence on the minimisation procedure at that iteration. This search space is illustrated in Figure 5.12 (a). Since the template is defined with a strong attraction to high intensity edges, if it occurs that the template lies inside the boundary of the individual then the attraction to the edges will cause the

<sup>7</sup>The perpendicular or normal direction is searched to reduce the effects of contraction on the contour as detailed in (Blake & Isard 1998, Jacob et al. 2004). Also it is considerably more efficient than a spiral search.

snake to expand towards the individual's boundary, and thus it is possible to accurately extract the boundaries, even when the template is not well positioned. Thus, it is not necessary to define an expansion force like that in (Cohen 1991). In particular, the contribution of the edges in the minimisation procedure is based on the strength of the edge pixel found within the search space of a current control point and its distance from the current control point.

The approach of Amini et al. (1990) was considered a superior approach to the original implementation for our application. The major reason is that it facilitates the inclusion of constraints in the definition of the minimisation procedure. Similarly, in the case of digital images, little information is gained by considering positions that are between two coordinate locations on the image grid. Moreover, it is possible to include measures to restrict the distance between control points, i.e. if two control points move within a certain distance of each other then the next best location that minimises the energy is assumed to be the minimum energy for that control point.

Parametric active contours are considered a more appropriate approach than geometric active contours. Indeed, the template is parametrically defined and the controls that are introduced to control the deformation and the evolution of the contour cannot be incorporated into the geometric active contour framework (Xu et al. 2000). Other reason related to the decision to use parametric snakes are described in Section 2.9.

### **Termination Conditions**

The contour is assumed to have extracted the correct boundary when the minimisation process stops or the difference between the current energy level and the previous level has reduced below a threshold. It is necessary to determine if the correct boundary has been extracted. This is achieved automatically by examining each of the control points and ensuring that they lie on an edge in the edge feature map. Each control point that is not located on an edge is further examined. Algorithm 1, determines how to encourage a control point to move to an edge. Algorithm 2, determines if it possible to reduce the number of control points to describe the boundary.

#### **4.3.6 Issues Highlighted in this Approach**

- The image information that is extracted using the contours is projected or mapped to the silhouette of the model using the texture mapping procedure in sections 4.2.2 and 4.2.3. Active contours provide a more accurate boundary extraction process that results in a more accurate mapping of the image information to the model.
- The quality of the face (of the model) on the model significantly reduces its quality (photo-realism). This is primarily because the texture that is produced for the face is based on a significantly small area of the image, and in some instances, is further scaled to map it to the underlying model.
- In mapping the information to the underlying model, significant shape information is being sacrificed and the personalisation of the models is being reduced. The final positions of the four B-spline contours contain information and the possibility exists to use this information to create a model of the individual.

---

**Algorithm 1** Automatic Insertion of Control Points

---

```
loop
  for each control point
    {first pass}
    Check if all points are located on an edge
    if control point not on edge then
      check neighbouring control points
      if neighbouring control points on edges then
        add new control point either side of current control point
      else if neighbouring control points not on edges then
        search space normal to the control point, is increased
      end if
    end if
  end loop
end loop

{second pass}
loop
  for each control point
    if control point not on edge then
      check if its position has changed
      if Position has not changed then
        ask for user assistance
      else
        minimise contour's energy and update current control position
        start second pass
      end if
    end if
  end loop
end loop
```

---

---

**Algorithm 2** Automatic Removal of Control Points

---

```
Determine if three control points are linear
loop
  for each control point  $n$ 
    if control points  $n - 1$ ,  $n$  and  $n + 1$  are linear then
      remove the central control point
      minimise the energy in the contour
      if neighbouring control points on edges then
        add new control point either side of current control point
      else if neighbouring control points not on edges then
        search space normal to the control point, is increased
      end if
    end if
  end loop
end loop
```

---

#### 4.4 Approach 3: Using Facial Feature Extraction to Enhance 3D Human Models

As discussed in Section 3.5, the face of an individual provides significant detail that is important in determining the quality and realism of the model that is produced. Thus to provide photo-realism of the model, it is important that the face is highly detailed. This can be achieved by finding correspondences between features on the face of the individual and the face of the model. The predominant features on an individual's face are the nose, the eyes and the mouth, which provide geometric dependencies and constraints for precise face localisation. These dependencies are used to accurately position the individual's face on the model in order to enhance the realism of the 3D human model.

However, the locations of these features are commonly used in other applications, e.g. the normalised facial image for the creation of the MPEG-7 Face Recognition (FR) descriptor is obtained using the predefined eye locations (MPEG7 2002). The majority of applications use only the frontal image for the purposes of extracting facial information, and in general, the larger the image area containing the face the more reliable the feature localisation. In the approach described in this section, the side view is also considered as it provides additional information for locating the facial features. Significant methods developed for the localisation of facial features are discussed in (Boyle et al. 2005, Chelappa et al. 1995).

The structure of the system for creating a personalised 3D model is presented in Figure 4.19. Elements of the process are described in details in the following sections; firstly the necessary requirements enabling the accurate capture of the facial image data are described as well as any issues arising while fitting the template. Finally the texturing process that incorporates the facial features is detailed and some results are shown.

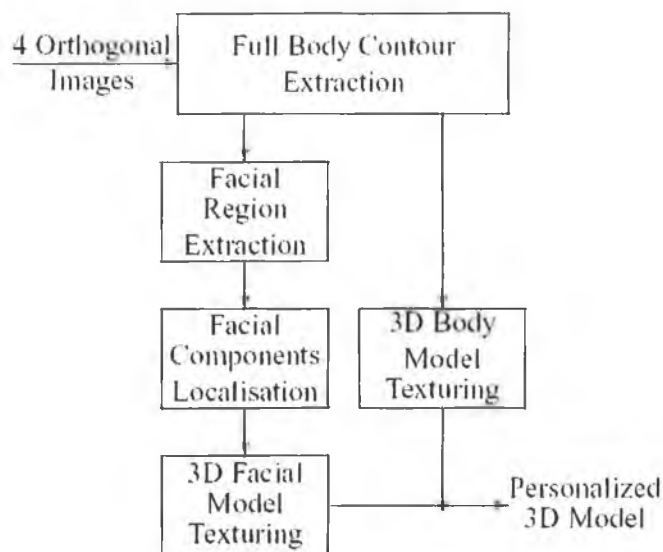


Figure 4.19: Facial features localisation algorithm.

#### 4.4.1 Image Capture and Template Fitting

The basis of the image capture process is the same as previously described in Section 4.3.1. In addition to this, the individual should look directly at the camera or little above the camera. This is to ensure that all the features of an individual's face are visible which is important for the extraction of the facial features. This should also be the case in the side view to provide accurate profile information. Figure 4.20 shows examples of the images that were used to test this approach. The background in the images, shown in Figure 4.20, are not extremely cluttered because the principle is to demonstrate that the location of the facial features can be used to enhance the model, but the same principles can be applied to images captured in more complex environments.

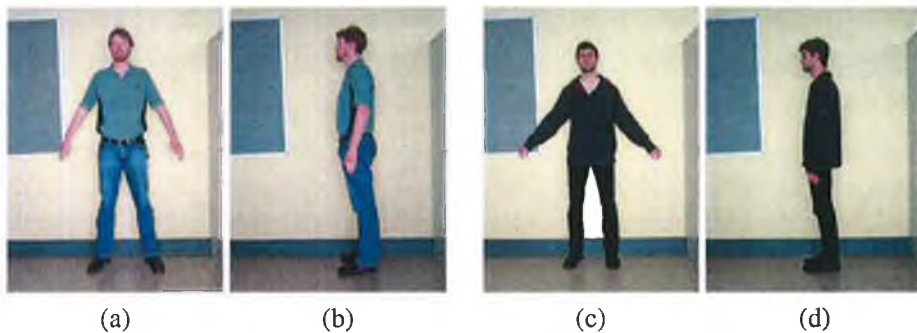


Figure 4.20: Examples of the capture images used for testing this approach.

It is necessary to have accurate segmentation of the individual from the background to ensure that it is not textured to the underlying model. This is achieved using the templates described in the previous section. The subtraction of the images used in the initialisation of the template is shown in Figure 4.21. The estimate of the bounding box is shown in each of the cases.

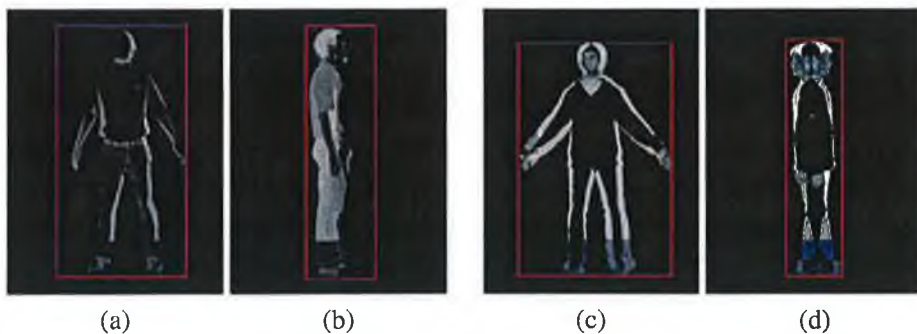


Figure 4.21: Results of subtraction of the front and back views and the subtraction of the left and right views.

#### 4.4.2 Face Localisation

The facial region is well localised by the final body contour and it can be used for the precise localisation of the facial features. In the case of the frontal view of the face, the colour segmentation is applied to the facial region and the facial components are found. The segmentation is

carried out in three steps: an initial segmentation, followed by feature extraction and classification (Boyle et al. 2005). The results of the feature localisation in the front view is shown in Figure 4.22 (a). The extraction of the facial features is detailed in work undertaken by Cooray & Uscilowski (2004).

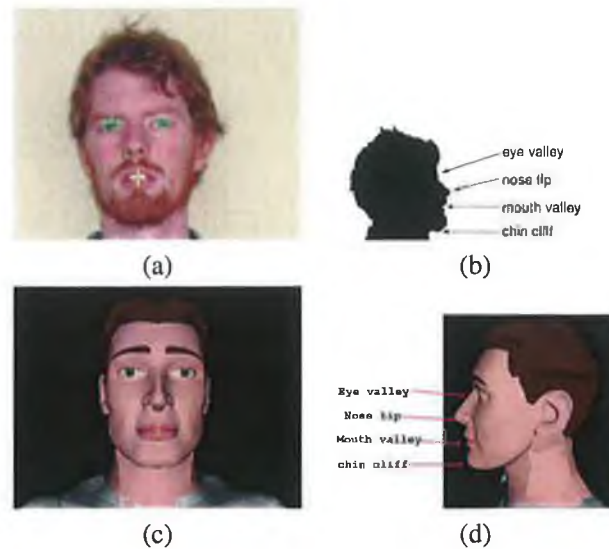


Figure 4.22: Facial features locations in (a) front view and (b) side view head with the key features marked. (c) shows the front view of the model with the key features marked and (d) shows the key features identified on the side view of the head. image.

The location of the shoulders in the front view is used to provide a initial estimate of the base boundary of the head in the side view. The feature location in the side views of the head is achieved by scanning the contour, defining the head from the top to the bottom and searching for the dominant valleys and peaks in the head contour using gradient analysis. When considering the right-hand side view of the head as shown in Figure 4.22 (b), the nose tip can be found as the peak on the right hand side boundary, the eyes should be placed in the valley above the nose peak and the lips form small peaks below the nose tip. The valley of the chin provides information about the bottom boundary of the face, which in general cannot be reliably located in the front view because of shadow. If the ear in the side view of the head is visible and not covered with hair, its location can be found using the segmentation technique used for the features extraction from frontal view images (Boyle et al. 2005).

#### 4.4.3 Texturing and Personalising the Model

To texture the underlying model, four silhouettes corresponding to the captured images are generated. These are used for establishing correspondences between the captured images and the underlying model. The approach is based on feature extraction algorithm in (Hilton et al. 1999). The establishment of features is essential to enable the accurate texturing of the model. Having the correspondences enables the texturing of the model on a part-by-part basis, ensuring that the scale of the different body parts is preserved. In the texturing algorithm proposed in (Boyle 2004)

and described in a Section 4.2.3, the normal vectors for each tri-face of the model are used to determine which image is used to texture that face of the model. The major limitation with the approach in Section 4.2.3 is that the face is not accurately textured, and this reduces the realism of the model. To overcome the limitation of this approach, the facial features are located on the face of the model and on the face in the captured images. These are used to align the individual's face with that of the model. These features are indicated in Figure 4.22 and the geometric relationships are shown in Figure 4.23. These relationships and distances are used for scaling and validation.



Figure 4.23: The locations of the facial features and the distances between them used for deforming the underlying model.

The distance between the eyes  $d_e$  (see Figure 4.23 (a)) is used for scaling the texture image in the horizontal direction. This distance is also used for the creation of the MPEG-7 FR descriptor and is essential for defining the size of the head (MPEG7 2002). This ensures a high level of recognition of the model in a virtual world. The vertical size of the texturing image is adjusted using the distance between the eyes and the centre of the mouth  $d_m$ . The three points representing the eyes and mouth locations are used to calculate the centre of the facial region and to position the facial texture on the underlying model in the front view.

The side view images deliver information required for the enhancement of the head model viewed from the sides of the head. As shown in Figure 4.23 (b), the triangle drawn between the eye, the mouth and the ear is used for determining the scaling factor for side images. The distance between the ear and the eye  $d_d$  determines the depth of the head model whilst the distance between mouth and eye should be equal to the distance  $d_m$  in the frontal view and can be used for the validation of the vertical dimensions of the image. When the ear is covered with hair and not visible, the head boundary is used for finding the depth of the head model. Once the facial information has been aligned, the texturing technique in Section 4.2.3 is used to texture the facial region of the model.

#### 4.4.4 Outcome of Texturing the Model Using the Facial Components

The results obtained using the proposed method to enhance the realism of the human models, using the facial features, are presented in Figure 4.24 and Figure 4.25. It can be clearly seen that the quality of the models that use the facial features in the texturing of the model provides superior results. In Section 5.4.3, additional models are generated and observed at different depths from

the viewer to observe the differences in the quality of the texturing. The images in Figure 4.20 are used to texture the underlying model in Figure 4.24. In this figure, the complete model is textured using the information from the four orthogonal views. Figure 4.24 (a) shows the model textured without using the facial features to position and scale the facial texture, and Figure 4.24 (b) shows the model with the aligned facial features.

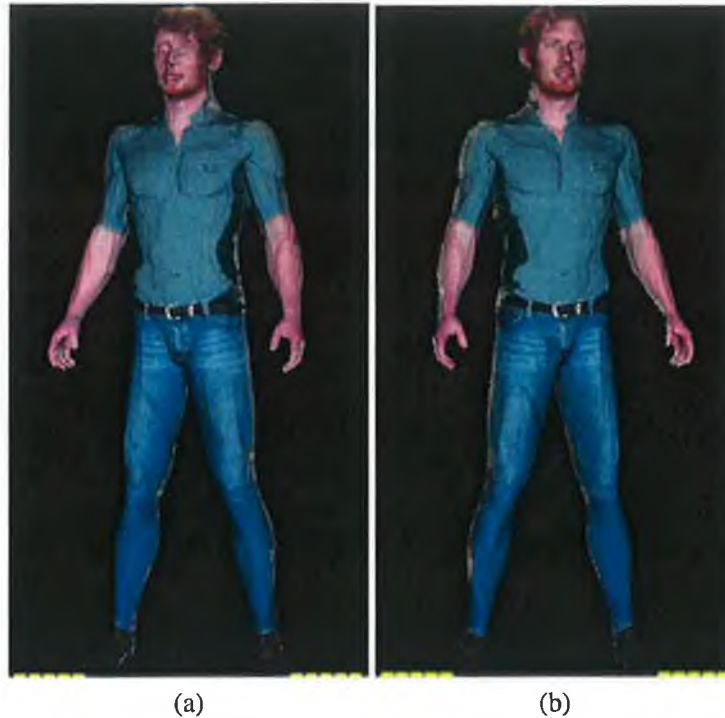


Figure 4.24: The created human model with (a) misplaced facial features and (b) aligned facial features.

In Figure 4.25, the results for a second set of images are combined to texture the same underlying model. In this set of results, only the upper body and the head are shown and the difference can be easily seen when the features on the model are positioned accurately. In Figure 4.25 (a), the frontal image is shown. In Figure 4.25 (b), the simply textured model is shown and Figure 4.25 (c) and (d) show two views of the textured model when the facial features are used to position the facial texture. In Figure 4.25 (a) it can be seen that the head of the individual is slightly tilted and in parts (c) and (d) the face is correctly aligned. This is achieved by calculating the measure  $d_m$  in Figure 4.23 from the average vertical coordinate for each eye.

#### 4.4.5 Issues Highlighted in this Approach

- The enhanced model still does not provide high detail of the eye regions, although this technique can be used to texture the model using a separate high resolution facial image (Lee, Goto & Magnenat-Thalmann 2000) or mapping each of the facial feature separately.
- The information in the final contour is used to accurately locate the head of the individual but the scaling of the images still reduces the quality of the final model and the shape in-



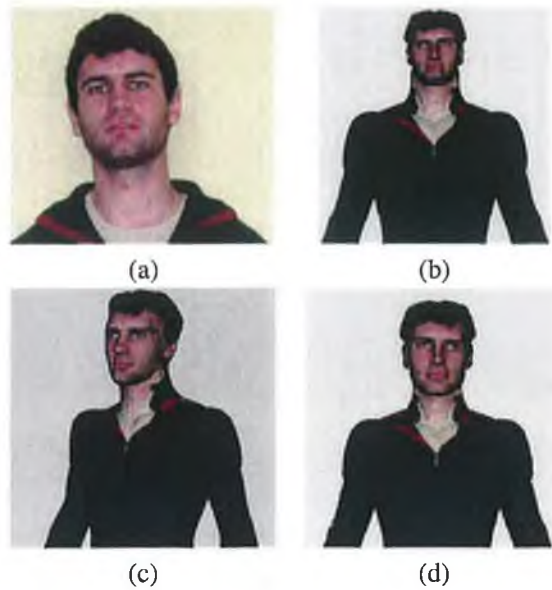


Figure 4.25: The second human model textured with the facial image in (a). (b) shows the misplaced facial features and (c) (d) show two views of the model with aligned facial features.

formation that is extracted is not being fully utilised. The main reason that this information is not fully utilised, is that this approach shows that improving the quality of the texturing of the face can add greatly to the personalisation of the model, even without modifying the body.

- The presented method can improve the realism of the 3D human models for low-resolution devices such as PDAs or mobile phones, providing a low-cost solution to the creation of personalised 3D models without requiring costly full 3D reconstruction.

## 4.5 Approach 4: Silhouette Based Models of an Individual

The silhouette based approach described in this section provides a convenient and flexible method for the 3D reconstruction of human models. In particular, based on the method described in Chapter 3, a silhouette based approach provides the most flexible method for the reconstruction of an object provided that the viewpoints are known in advance. The information in the silhouettes can be combined to create a model. Such a model would include information that is discarded in previously described approaches. This relates primarily, to the shape information that is extracted using the active contours (Boyle & Molloy 2005*b*).

Passive techniques for capturing shape information and/or inferring 3D shape from a set of multiple images provide the basic input for shape-from-silhouette approaches. They have established themselves as a technique for recovering the approximate surface shape of an object by capturing the subject against a known background (Laurentini 1994), such as a blue-screen. The images from multiple views are combined to determine the spatial volume occupied by the object and reconstruct a surface model. This approach can be used to produce highly realistic object models when combined with a texture mapping technique.

In (Slabaugh et al. 2004), silhouette based reconstruction is described as the simplest form of volumetric multi-view reconstruction. Using silhouette-based reconstruction of an individual is a difficult task, but it can be achieved provided that foreground/background segmentation at each reference view is possible and relatively simple to implement. In this case it is not necessary to implement measures to establish the visibility of points in each image. In the approach described in this section, the silhouettes are extracted from each image using the template created in approach 2 and described in Section 4.3. The viewpoints are known in advance since the camera position does not change between views and the individual rotates  $90^\circ$  between each capture.

The final position of the active contours described in detail in Section 4.3 define four silhouettes of the individual that are expressed in terms of control points in a 2D plane. The 2D B-splines provide a minimal representation of the body contours and, in Section 4.5.1, are combined to create a 3D B-spline surface representing the maximal object silhouette equivalent of the individual that acts as a bounding volume and is used as an approximate human model. An overview of the main system elements are shown in Figure 4.26. Moreover, the photo-realism of the model can be increased if the image data used to texture the model is not scaled.

### 4.5.1 Alignment of views in 3D

In the construction of the human model, the first stage involves aligning the extracted silhouettes in 3D. The two views, namely the left and the front views, are positioned in 3D by aligning the minimum vertical point on the front silhouette with the minimum vertical point in the left silhouette<sup>8</sup>. The minimum vertical values correspond to the top of the head. These points are chosen to ensure that the photo-realism of the face is achieved and thus that of the model. Aligning the silhouettes using the minimum vertical values explicitly incorporates a higher priority for the alignment of the head.

---

<sup>8</sup>The minimum vertical points on the silhouettes are used because the image coordinates are measured from the top left corner of the image.

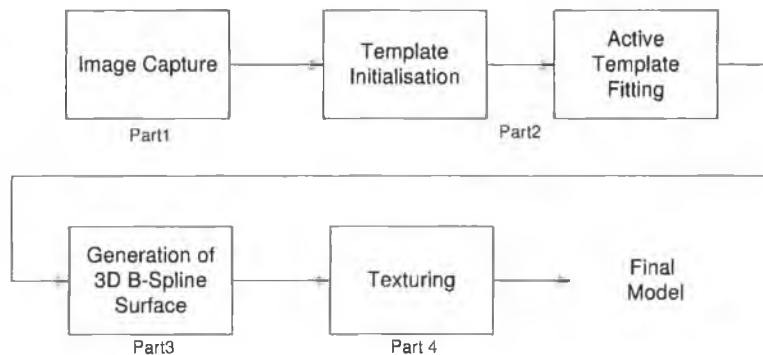


Figure 4.26: Illustration of the main components of the system in approach 4.

In respect to the capture process described in 4.3.1, the silhouette of the side view has a larger vertical length. The difference in length is typically of the order of five or six pixels. This results from the fact that the individual stands with their legs apart in the front view and stands with their legs together in the side view. After the alignment of the views, the vertical difference only affects the feet of the model.

### Combining the silhouettes

The 2D silhouettes are expressed in terms of the 2D control points, and after the alignment of the silhouettes in 3D it is necessary to transform the control points to 3D this is achieved based on the view alignment. The front and back silhouettes are parallel to the  $x - y$  axis in 3D and the side silhouettes are rotated to be parallel to the  $y - z$  axis in 3D. Thus the axis passing through the  $x$  value for the minimum vertical point is used as a central axis for the rotation of the left view. The control points in each 2D view are then explicitly transformed into 3D by setting  $z$ -value of the control points in the front view equal to 0 and setting the  $x$ -value of the control points in the left view to zero.

The alignment of the views means that at each vertical location four values are available. If the explicit vertical values are required, these can be calculated by interpolating the B-spline curves (Moore et al. 2005). The four values correspond to the maximum silhouette equivalent of the individual, from two views and is illustrated in Figure 4.27. This figure also illustrates that the finer detail that is inside the silhouette of the individual cannot be recovered from the silhouettes. This is particularly evident on the face, which is the most detailed part of the body and has a high number of concavities. On other body parts, this is not as evident, for example the legs.

Having aligned the contours, it is possible to generate a simple B-spline surface passing through the four extremes at a particular vertical height<sup>9</sup>. This is completed for the head and the entire upper body of the individual. This results in a series of parallel elliptical B-spline curves. To increase the accuracy of the curves at a vertical height, prior information about the human shape is used to interpolate new control points at a vertical level. This prior information is designed

<sup>9</sup>The repetition of the knots in the knot vector is used to insure that the B-Spline curve passes through the extremes at a vertical height.

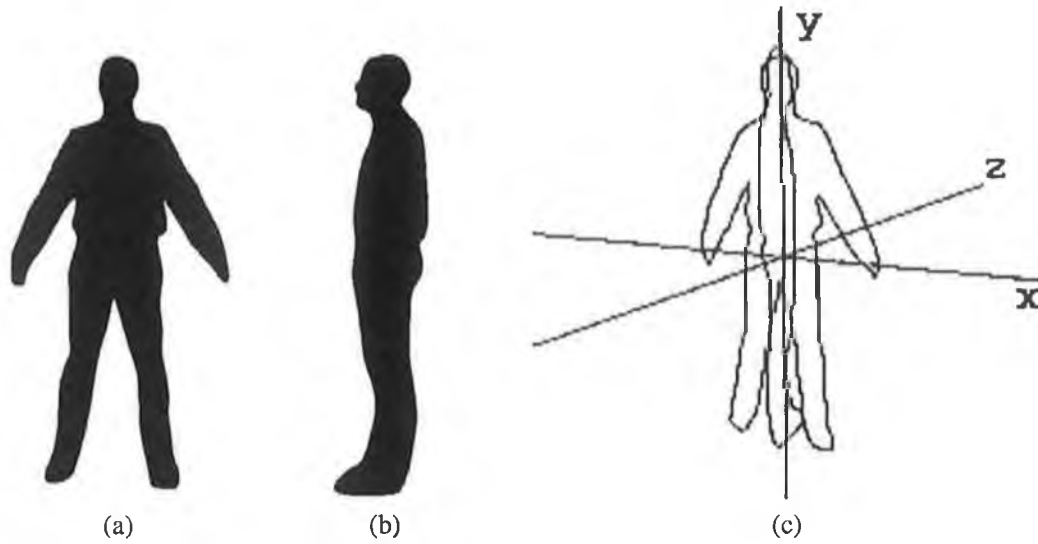


Figure 4.27: parts (a) and (b) show examples of the 2D front and left silhouettes and part (c) shows the two silhouettes aligned in 3D space.

to make the front of the face and the upper body relatively planar in comparison to the elliptical surface. This process does not attempt to rebuild the complexities of the individual's face, as this task would result in an exhaustive procedure that would reduce the flexibility of the approach.

The 3D B-spline surface is generated by considering the control points as a bidirectional web of control points, two knot vectors<sup>10</sup> and two univariate B-spline basis functions,  $N$ . This is expressed as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} \quad (4.3)$$

where  $\mathbf{P}_{i,j}$  are the 3D control points and  $U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}$  and  $V = \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}$  are two knot vectors and  $p$  and  $q$  are the degree of the curve.  $U$  has  $p + 1$  knots and  $V$  has  $s + 1$  knots. The number of iterations between control points can be increased or decreased depending on the required level of detail. This is illustrated in Figure 4.30 where the contours for the head are closer together than on the rest of the body. Also the body parts are shown in different colours in the front view.

The new control points that are interpolated at a particular height do not extend past the visual hull of the combined silhouettes. This is in line with Laurentini (1994), where it is stated that points and surfaces inside the visual hull can take on any form, provided that they do not exceed the visual hull of the individual.

The silhouette based reconstructions of the legs and the arms are treated separately and described in the next two subsections. The legs require the introduction of a rotation element to align the legs in each of the views (Moore et al. 2005). The arms are treated separately because the depth information of the arms cannot be reliably extracted in the side views.

<sup>10</sup>The knot vectors determine how close the B-spline curve approximates the control polygon.

### Reconstruction of the Legs

There are two approaches to the reconstruction of the legs. The first approach involves ignoring the fact that, in the front views, the legs are apart. In this approach, the legs are reconstructed by using four points at vertical height and, in a manner similar to that described above, basic shape information is incorporated. This results in a mis-alignment of the side and front views, but as the structure of the legs is close to uniform, the shape difference is not adversely affected.

The second approach uses the approximate location of the bone passing down the leg of the individual (see Figure 4.28). The axis is aligned by rotating the leg through  $\theta^\circ$  such that it is parallel with the  $y$ -axis. Then it is aligned with the information in the side view. This is used to create a series of ellipses parallel to the  $x - z$  plane. When the ellipses are completely reconstructed, each of the control points that are interpolated, are rotated back through  $\theta^\circ$  to realign the legs. This process generates a more accurate reconstruction of the leg. This is illustrated in Figure 4.28.

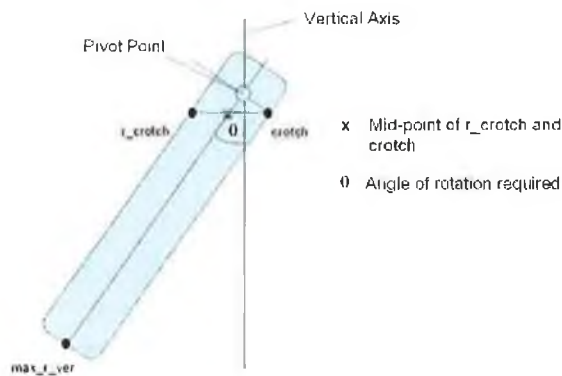


Figure 4.28: A schematic of how the leg is rotated about the  $y$ -axis to construct the legs.

### Reconstruction of the Arms

The reconstruction of the arms is a more challenging process because it is only possible to reliably extract information from the front view. To account for the information that is missing in the side view, a simple approach would assume that the arms are approximately circular and that the diameter extracted in the front view can be used as a solid estimate of the depth information. This results in arms that have highly cylindrical shape, see Figure 4.29 (a) where the true shape of the arms is not accounted for. Furthermore, the upper arms that are extracted in the front and back view do not have a complete cylindrical shape as they connected to the upper body of the individual and when combined with the reconstructed body do not align correctly with the rest of the body.

In (Cohen. & Lee 2002), the information related to a missing dimension of a body part is estimated using the information based on a generic human model that is scaled based on the available information. This approach is incorporated into the silhouette based reconstruction process for the arms and results in an accurate reconstruction of the hand. The second and more successful technique applied to reconstructing the arms take into account depth information from the upper body

and uses the generic shape of the model. The clothing an individual wears covers the shoulder and the upper arm, thus any movement of the upper arm would be observed in the clothes. Thus using the depth of the upper body to reconstruct the parts of the arm that are above the armpit results in a better shape of the final model.

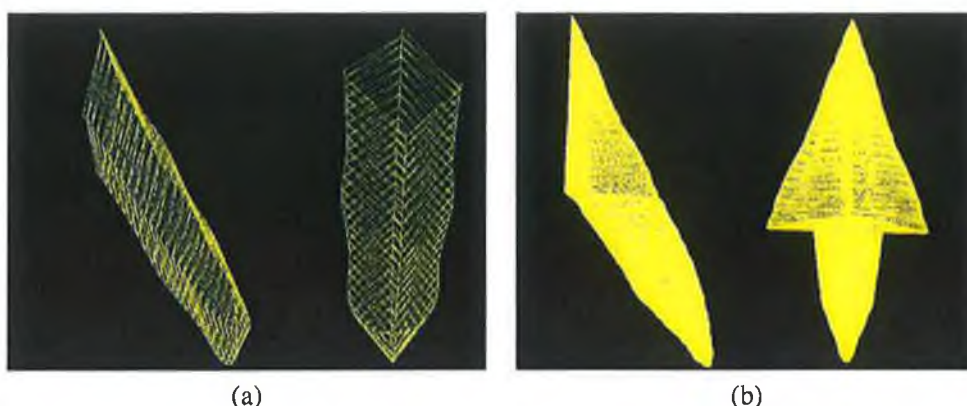


Figure 4.29: (a) shows the simple reconstruction of the arms from two views. It is clearly seen that shape of the arm does not approximate the shape of the individual's arm (In this case only two control points are interpolated between points on the control points extracted in the front view). In (b) the use of the depth from the upper body is shown from two views.

The reconstruction of the arms using this technique is shown in Figure 4.29 (b). The width of the upper arm is created using the information available in the front silhouette and the depth information is created using that width of the silhouette in the side view. The remainder of the arms, below the armpit, is created using scale information from the model underlying model and the width of the arm in the front view. The use B-spline curves to approximate the shape of the arms means that the fine detail of the hands is sacrificed.

#### 4.5.2 Combining of the Body Parts

The body parts are combined using the key features that are extracted in Section 4.3.2. The key features allow the individual's body parts to be recombined and correctly aligned with each other. The key features also allow the estimation of the skeleton of the individual, which can be equally used for the positioning on the body parts. This is more significant in terms of the animation of the model and can be interpolated after the body parts are created.

The results of the combination are shown in Figure 4.30. In this figure, the arms are connected to the upper body using the location of the shoulder and the armpit. To ensure that the arms are connected to the upper body the control points on the inside<sup>11</sup> of the shoulders and armpits are used in the reconstruction of the arms. This permits an overlapping of the B-spline surfaces. The legs are joined to the body using the information about the position of the crotch and the l.crotch and r.crotch key points.

<sup>11</sup>On the right hand side of the model, inside is used to indicate a control point to the left of the armpit and the shoulder. On the left hand side, inside is used to indicate a control point to the right of the armpit and the shoulder

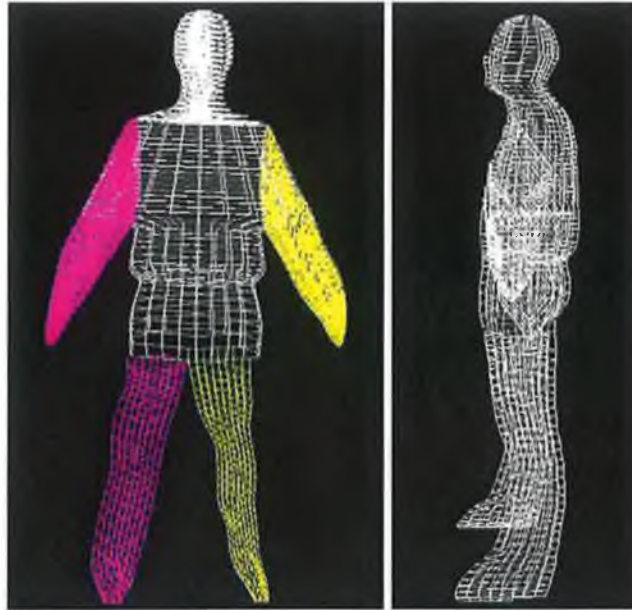


Figure 4.30: Two views of the 3D B-spline surface created from Figure 4.9 using volume intersection.

### 4.5.3 Texturing the Final Volumetric Model

This involves re-using the techniques developed in Approach 1 where the information inside the extracted silhouette of the individual is mapped onto the silhouette of the model. In this case, the extracted silhouette of the back of the model is mapped onto the extracted front silhouette of the individual. This is necessary to ensure that the extracted texture in the back view is mapped onto the bounding volume of the individual.

The texturing of the bounding volume is seen as an intermediate stage in generating a personalised model of the individual. While the extracted silhouettes of the individual contain accurate shape information when placed in 3D, it lacks the finer details. This approach shows that while it is possible to personalise the bounding volume with a texture map, the missing detail cannot be completely compensated for.

In applications designed for mobile devices, the human model can be created by directly texturing the bounding surface. High-resolution views of such a model are shown in Figure 4.30. This model is created using the four images from which Figure 4.9 belongs. This model has significant advantages over other low-cost solutions (Hilton et al. 1999, Lee, Goto & Magnenat-Thalmann 2000). In particular, since the textures are not transformed the photo-realism of the model is improved even when the detail on the underlying model is missing and when textured the face of the model is textured with the individual in the front view (see Figure 4.31 (a)). The size of the model is representative of the captured individual.

### 4.5.4 Issues Highlighted in this Approach

In addition, if the visual hull is not calculated correctly (accurately) the photo-realism of the model will be significantly reduced when new views are generated. To increase the accuracy, more



Figure 4.31: Two views of a textured model in Figure 4.30.

information from silhouettes can be used during the reconstruction. The main source of such information is colour. In practice silhouette based modelling is confined to modelling objects that can be segmented from their environment rather than the creation of 3D scenes. According to Laurentini (1994), the viewing region contains the viewpoint that are allowed for observing the object as well as the object itself. Our approach is designed to avoid scale problems and through the capturing of images from different positions, it was observed that the visual hull does not vary significantly and thus the four views that are captured are sufficient to get maximal silhouette equivalent.

- However, silhouette based approach alone cannot reconstruct surface concavities. Close-range photogrammetric approaches have been developed, which reconstruct 3D shape from matches between images. To generate the close range images involves capturing images inside the complex hull of the object and establishing image features matches between images is obtained either manually or automatically to recover models of surface shape (Collins & Hilton 2001). To extract this additional information the individual would be required to stand in the same position for an extended period of time or else a system would need to be devised to have a multiple camera set-up. Both of these reduce the flexibility of a system and increase the complexity involved in the reconstruction of the model.
- The quality of the reconstruction depends on the accuracy of the segmentation. In addition, the silhouettes that are extracted only contain the visual hull of the individual. Thus the finer details are lost in the reconstruction. Although some of the finer detail is lost in the reconstruction process, it has been shown in (Cheung et al. 2003) that texturing can be used to enhance the appearance of the model.
- The reconstruction is specific to the individual that is captured but the intermediate values depend on the shape information that is pre-coded. This depends on the basic idea that the



human body can be modelled as elliptical shapes.

- The depth information for the arms cannot be recovered accurately from the silhouettes and is obtained in a method based on that in (Cohen. & Lee 2002), using a method of similarity between the dimensions of the extracted body parts and a generic model.
- The prior knowledge related to the animation information in the underlying model is lost.
- The complexity of the model can be predefined and is related to the number of control points used.

## 4.6 Approach 5: The extension of Active-Meshes to 3D

In approach 4, the 3D bounding volume is automatically created using volume intersection that combines the front contour and a side contour to form a 3D B-spline surface. The four contours are used to position the skeletal information that is extracted using the 2D feature detection algorithm. Approach 4 is a step closer to complete personalisation of the model, as it is generated using shape information that is specific to the individual being created. As highlighted in Section 4.5.4, the model is not easily animated and lacks the fine detail that can be used to distinguish one individual from another. An innovative reformulation of the active-meshes is presented in 3D as a method to overcome both issues. The technique is based on the energy framework developed by Molloy & Whelan (2000) for motion tracking and extended to actively deform an underlying model to take on the shape of the bounding volume. This approach combines the notion of rigidity and elasticity introduced in (Molloy & Whelan 2000) to permit a 3D model to deform to the shape of the individual while retaining the internal structure using internal constraints between the mesh vertices. Preserving the internal structure is important in maintaining the fine detail that cannot be extracted using the silhouette based approach, described in the previous section. In addition, for the animation of the model, the positions of the joints are known and ensures that the model can be easily animated.

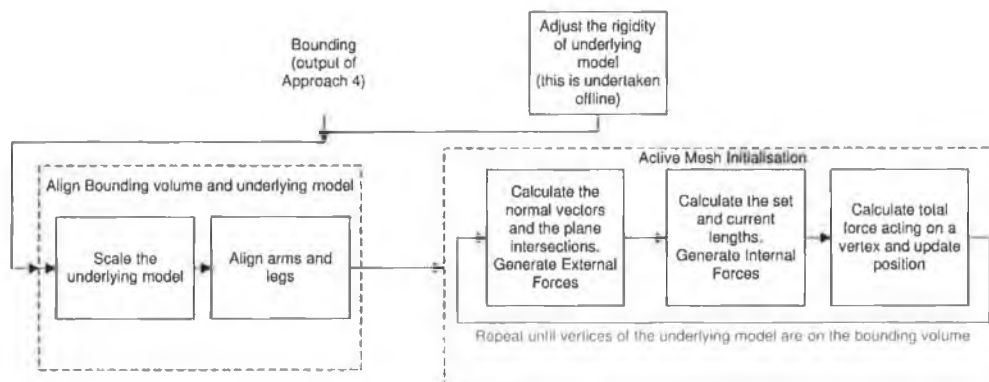


Figure 4.32: The main components of the Active-Mesh modelling tool.

In this approach, the bounding surface and the underlying model can be considered as active surfaces, although in the descriptions that follow the term, active surface is used to refer to the underlying model, since it is being actively deformed by the bounding surface. In terms of the traditional notion of snakes, the bounding surface acts as an external force. The bounding surface is a B-spline surface that is generated in Section 4.5. The shape of the bounding surface approximates that of the captured individual.

### 4.6.1 Specification of the Internal and External Constraints

The user has the option to specify the internal and external constraints that act on particular sections of the model or else the defaults can be used. The default internal constraints are used to ensure highly irregular parts of the underlying mesh, e.g. the face, retain the same structure while the external forces attempt to pull the vertices towards the bounding surface.

The internal constraints determine how the mesh can deform. If the internal constraints are strong, the mesh will attempt to preserve its original shape and structure at each iteration. This is particularly important around the face of the individual because if the internal constraints do not preserve the structure then the concavities on the face will be deformed under the influence of the external forces. Thus strong internal forces are needed for parts of the model that contain strong detail. For other parts of the body where the fine detail does not affect the realism of the model, for example on the legs, the internal forces can permit greater elasticity to enable the underlying model to deform to the shape of the bounding volume. Within the active-mesh formulation the incorporation of the internal constraints must be sufficiently flexible to allow different parts of the mesh to have different internal constraints. This is achieved by specifying the rigidity for each element of the mesh. The internal forces can be applied uniformly across a particular mesh or at individual points depending on how the mesh is to be deformed.

This is expressed in the following equations and illustrated in Figure 4.33:

$$\vec{F}_{Line} = L_{cur}(x)\beta_L\vec{i}_x + L_{cur}(y)\beta_L\vec{i}_y + L_{cur}(z)\beta_L\vec{i}_z \quad (4.4)$$

where  $L_{set}$  and  $L_{cur}$  are the set and current length of the lines joining to vertices of the mesh at

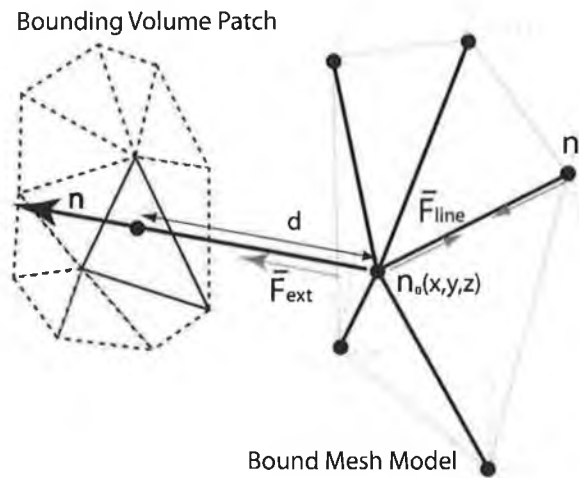


Figure 4.33: Internal and external energies. The internal forces act along the mesh lines and are indicated by the vector  $\vec{F}_{Line}$  and the external forces are generated using the normal vector,  $n$ , to the mesh element.

particular iteration,

$$\beta_L = \frac{(L_{set} - L_{cur})}{\alpha_L L_{cur}}, \quad (4.5)$$

and  $L_{cur}(x)$ ,  $L_{cur}(y)$  and  $L_{cur}(z)$  represent the x, y and z components of the current mesh line lengths, which determine the internal energy,  $\vec{i}$  is a unit vector with same axis as the mesh model,  $\alpha_L$  is a user defined factor to limit the effect of forces and  $\alpha_L$  is represented visually on the model and has the visual range (0, 255) that is normalised to ensure that the internal forces are not under biased. A closer inspection of the  $\beta_L$  parameter shows that it is related to the length of the mesh-line connecting two vertices in the mesh and is independent of the position of each of the vertices. This is important in 3D because it permits the mesh to be globally transformed, possibly under

the influence of a strong external force and still retain its structure. In Equation 4.5, the numerator encompasses a central element in the forces that act on a particular node. If the current length is bigger than the set length, the  $\beta_L$  parameter will be negative and will act to reduce increase in length of the mesh-line. The parameter  $\alpha_L$ , determines the significance of the  $\bar{F}_{Line}$ , if  $\alpha_L$  is small then  $\beta_L$  will be large and will permit only small changes in length. If  $\alpha_L$  is large then  $\beta_L$  will be small and will permit significant changes in length. The effects of  $\alpha_L$  and  $\beta_L$  are tested in Section 5.6.

Visually representing the internal forces on the model is important because in 3D it can be difficult to select a vertex to examine its connections. Thus using different colours to represent the strength of the internal constraints provides an intuitive way of visualising the areas of the model that are tightly bound. Moreover, since the rigidity of a vertex also depends on the rigidity of the neighbouring vertices, it is usual to consider the rigidity of an area of the surface as opposed to an individual vertex. This is also an important feature in providing a user assisted determination of the internal constraints. An example of the visually represented internal forces is shown in Figure 4.34. The red areas represent areas with strong rigidity and tightly bound vertices. The lighter red parts of the mesh that surrounds the red areas indicates weaker internal forces which are formed where tightly bound vertices are connected to weaker bound vertices.



Figure 4.34: Visual marking of the rigidity on the underlying model. The red parts have strong rigidity and tightly bound vertices. The lighter red parts indicate weaker internal forces.

The intersection of the normal vector from each vertex of the underlying model and the 3D bounding surface is used to determine the external forces that affect vertices of the model:

$$\bar{F}_{ext} = \beta_{ext}d(x)\bar{i}_x + \beta_{ext}d(y)\bar{i}_y + \beta_{ext}d(z)\bar{i}_z \quad (4.6)$$

where

$$\beta_{ext} = \frac{\alpha_E(d_{max} - d)}{d_{max}} \quad (4.7)$$

and  $d$  is the distance normal from  $n_i$  to the intersection of the normal with the bounding volume and  $d_{max}$  is the maximum normal distance from the underlying model vertices controlled by a single patch on the bounding volume. This is necessary to ensure that the vertices in a particular part of the mesh are deformed relative to the strongest force affecting this part of the mesh. This is important in preserving the structure of the mesh.

The combination of the forces involves multiple line connections at each vertex of the model, and thus there are multiple forces at each vertex. This is illustrated in Figure 4.35. These forces are combined using a weighting factor that is inversely dependent on the Euclidean distance separating two connected nodes and is expressed as

$$\beta_i = 1 - \frac{D_i}{\sum_{i=1}^N D_i} \quad (4.8)$$

Thus the force on the centre node is

$$\bar{F}_0 = \sum_{i=1}^N \beta_i \bar{F}_i \quad (4.9)$$

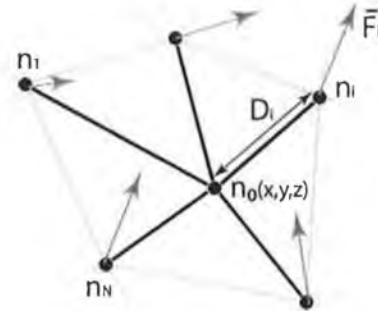


Figure 4.35: Combining the Forces at a single node. This adapted to 3D and is based on a diagram in (Molloy & Whelan 2000).

The resultant force determines the maximum change in length of a particular line between two vertices. In a highly elastic case, this can be used to position the new vertices. Although in general, at each iteration the length of the line is scaled to a fraction of this change.

$$L_{set} = L_{set} + \alpha_l(L_{cur} - L_{set}) \quad (4.10)$$

where  $\alpha_l$  is a user defined factor that limits the change in the line length. This introduces elasticity that can be set to determine the rate of expansion of the model. This is an important factor if the underlying model is not completely within the bounds of the active surface.

### Relative Elasticity

The internal energy term in Equation 4.4 provides a strong method for ensuring the rigidity of the model is preserved. As the external forces pull the vertices on the internal model towards the active surface, the strength of the internal forces increases because the average distance between each of

the vertices increases and thus the internal forces have a greater influence in the force combination equation, Equation 4.9. This causes the minimisation process to prematurely end without reaching the desired solution.

In the original implementation of the active meshes in (Molloy & Whelan 2000), maintaining the internal structure is an important feature as parts of the object can disappear or reappear between consecutive frames. Also in tracking applications the shape of the object will change slowly over a number of frames, possibly due to parts of the shape that are occluded in one frame coming into view in the next, while in 3D the underlying objective is to mould the internal surface approximates the external surface. Thus the strict rigidity requirement in 2D is relaxed to permit a gradual expansion towards the external surface. This is defined as “relative rigidity” and is introduced by preserving the rigidity relative to the average length of each connecting line in the mesh. This ensures that the strength of the internal forces does not increase as the model is moulded to approximate the bounding surface.

When the rigidity is strong, the internal surface will be pulled towards the active surface but will maintain its original structure that is the relative distance between the vertices will remain constant. This ensures that areas that contain strong rigidity will be maintained.

### Consistent External Forces

Initially, the external forces are calculated based on the normal distance from a vertex on the internal model to where it intersects the active surface. This is calculated by representing the line in parametric form and then using the following equation for the intersection of a line and a plane<sup>12</sup>

$$t = -\frac{ax_o + by_o + cz_o + d}{an_x + bn_y + cn_z} \quad (4.11)$$

where  $ax + by + cz + d = 0$  is the equation of the plane,  $n = (n_x, n_y, n_z)$  is the equation of the normal vector passing through the point  $P_o = (x_o, y_o, z_o)$ .

Unlike the 2D case when two lines intersect, the possibility exists, in 3D, that the normal will intersect more than one plane. This can occur since three points define an unbounded plane. To check that the normal intersects between the points, it is necessary to perform a number of checks.

1. From Equation 4.11, it is first necessary to ensure that the value  $t$  that is calculated is greater than zero to ensure that the intersection is in the direction of the normal.
2. Then a ray-tracing algorithm is employed to ensure that the particular combination of vertices is intersected. Two approaches are considered. The first involves projecting the tri-face into 2D along the normal direction with greatest magnitude. This method is not considered practical in cases when the size of the tri-face is small according Watt (2000). The second approach involves summing the angles between each of the vertices and the point of intersection of the ray in the plane. If the sum of the angles equals  $360^\circ$ , the point lies on the particular tri-face. This operation is computationally more expensive but guarantees a unique intersection.

<sup>12</sup>The parametric equation of the line and the equation of the plane are derived in Appendix A.

The initial formulation of the external forces described above does not have sufficient strength to deform the underlying surface sufficiently. This occurred because the normal distance from the vertex to the active surface was continually decreasing as the vertex moved towards the active surface. Thus the internal forces become dominant and halt the deformation process. This is overcome by calculating the centre of the current body part that is being modelled and projecting a line from this centre through the vertex on the internal surface and finding where this intersects the active surface. This distance results in a stronger external force that remains constant over a number of iterations. This is consistent with the approach described in (Molloy & Whelan 2000) in which it is stated that the mesh will deform when subjected to a consistent external force over a significant number of iterations.

#### **4.6.2 Termination Process**

The minimisation process is terminated when the magnitude of the force exerted over a vertex is insufficient to change the position of the vertex. In general, this occurs when the vertex on the internal surface lies on the external surface and thus the effect of the external force is reduced to zero. When this occurs, the internal forces can result in the vertex moving on the active plane since the neighbouring vertices may be moved resulting in small internal forces moving the vertex.

This termination condition will not ensure that the shape will deform exactly to the active surface. This is particularly evident when the shape of the active surface is substantially different from the underlying model, for example when a sphere is being actively deformed into a cube, see Section 5.6.2. This occurs because the closest point from the underlying model to the active surface is not necessarily located on the boundary of two tri-faces. Thus once a vertex lies on a tri-face, it does not necessarily move towards a control point on the active surface. Furthermore, the goal is to approximate the bounding volume because when the internal constraints bind the underlying surface tightly the vertices of the underlying mesh will not rest on the bounding volume. This means that the termination condition has to be modified based on the rigidity of all the vertices of the underlying model. This is discussed in greater detail in Section 5.6.2.

It is possible to force this constraint, but in the situation that the underlying model has more vertices than the active surface, more than one point would be forced to the same point on the 3D active surface, causing undesired effects.

#### **4.6.3 Issues Highlighted in this Approach**

This approach illustrates a complete system that captures the individual shape and provides a flexible method that can generate an accurate model of the individual. In addition, this approach provides a complete active method that can be applied to mould any shape into another shape.

In respect to silhouette based human modelling, this approach enables the creation of an accurate human model from a limited number of views. The underlying human model is moulded to take on the shape of the bounding volume. In particular, this approach enables the model to be simply animated and incorporate, shape information specific to the individual, both of which are clear advantages over Approach 4.

This approach provides the home-user with a unified method for deforming an underlying

model through the inclusion of adjustable constraints that can be used simultaneously to limit the deformation of parts of the model while relaxing the constraints on other parts of the model to permit the model to deform and take on the shape of the bounding volume. This approach is simple to apply, and for the purposes of modelling an individual, it permits the incorporation of default constraints that enable the underlying model to be automatically deformed to take on the shape of the individual.

## 4.7 Discussion

In this chapter, the different approaches that are developed to extract the human shape information and construct human models from a limited number of images was described. This culminated in a definition of a flexible method to actively mould any shape into another shape and, in particular, remould an underlying model to take on the reconstructed volume. Each approach described in this chapter can be used to provide a human model that can be used in different environments. The validation of these techniques is detailed in Section 5.6.

In the first approach, the primary object is to investigate how existing techniques can be extended to separate the individual from a real environment and to provide a simple method to texture an underlying model. This model could be easily incorporated into mobile application or used as a low resolution model to replace impostors<sup>13</sup> in certain mobile applications (Boyle et al. 2004).

The second approach focuses on the extraction of accurate shape information from a limited set of images captured in a cluttered environment. This approach encapsulates the constraints necessary to extract an individual from a real environment in the form of a full body template. This template is unique, in that no other method attempts the accurate extraction of a human from a real environment in a single operation. The added accuracy introduced with this method enables the accurate determination of the shape of an individual and a better texturing of an underlying model. The accuracy of the template fitting is discussed in Section 5.3.6. Other applications of this technique enable comparison over time or demographic analysis of individuals (Boyle & Molloy 2005a).

The third approach is intended for use in gaming environments or in virtual worlds where the characters are predetermined and limited modifications can be made to personalise the model. This approach, unlike current methods used in gaming environments, permits the texturing of the underlying model, not just the face, to take on the appearance of the individual. This approach combines the active templates developed in Approach 2 with the geometric relationships between facial features to improve the level of personalisation of the model (Boyle et al. 2005). This approach also facilitates the use of automated methods for the recognition of an individual in a virtual environment based on the 3D model.

The fourth approach provides a method for recombining the silhouettes and building a bounding volume that is representative of the captured individual from a limited set of views. This is important in providing accurate shape information and demonstrates that, although a 3D model

---

<sup>13</sup>An impostor is a simple rectangle with a texture image attached. It is used to replace high-detail geometric models that are made up of thousands of triangles and are expensive to draw. Impostors are used for highly detailed buildings and virtual humans. They are also used to replace geometry of far-off buildings, as the extra detail afforded by geometric models cannot be seen across large (virtual) distances.



can be created; it does not provide the full realism that the end-user requires and motivates the approach developed in Approach 5. In addition to this, the method validates the use of silhouette based reconstruction of human models and shows that combining the silhouettes with some prior shape information, a representative 3D structure can be created. In terms, of practically using this model, it is a non-trivial task to animate these models because unlike the preceding approaches, no underlying model is used and thus only limited joint information can be extracted.

The final approach encompasses aspects of the earlier approaches and combines them with a innovative implementation of active-meshes in 3D that seamlessly combines the information in the bounding volume as an active surface that can be used to modify the underlying model. This approach lends itself to automatic and interactive application to various 3D modelling tasks. In this approach it is applied specifically to rebuild the fine details of an individual and to preserve the skeletal information of the underlying model. The application of the active deformation to 3D is an important achievement of this approach and overcomes complexities in other approaches that limits its use by non expert-users. In particular, the visual representation of rigidity simplifies the specification of constraints and is important because users not familiar with 3D navigation may prefer to apply the constraints on different 2D views of the model.

The methodologies that have been developed are fully automated and thus are suited for use by non-expert users. The approaches can facilitate user interaction. This has a number of uses:

- In situations where the level of clutter<sup>14</sup> makes it impossible to reliably extract the individual from the background and manual guiding of the active contours is necessary.
- Similar interaction is required when the individual is wearing clothing that is the same colour as the background.
- A user may want to reformulate the templates to extract other objects from a real environment.
- When using the active modelling of a 3D object, the user may want to set the rigidity or elasticity of the underlying model to create the desired model.

All the approaches permit movement between captures to generate the human models based on limited images captured using a single camera in a home environment. This is important because the individual is not required to adopt the same pose for a long period of time and there are no issues related to setting up and synchronising a system for multiple captures. In the approaches, it is possible to accurately extract joint locations and identify key features that are essential. The capture process only considers the capture of clothed models, thus, at best, the final model will be an approximation of an individual's shape.

---

<sup>14</sup>As assessment of the level of clutter is made in Section 5.3.

## Implementation, Testing and Results

### 5.1 Introduction

This chapter provides details of the different experiments undertaken to verify and test the approaches developed in Chapter 4. The primary objective within this chapter is to present results for each approach and to indicate in which situations each approach is applicable as well as to enforce the rationale for the progression of the research.

In each case, the individual was captured in a real environment with varying levels of clutter. The level of clutter was classified to allow comparison, when possible, with alternative approaches. Following this, various filters and pre-processing operations that were considered to improve the separation (extraction) of the individuals from their environment are discussed. This culminated in the active B-spline templates developed in Approach 2 (see Section 4.3). These templates are tested in different environments and examined to verify their correct operation.

In the previous chapter, certain results are presented to explain the rationale behind the design process. These results constitute a subset of the results set that are presented in this chapter. The complete set of results shows how each approach performs in different real environments and analyses the creation stages: from image capture through to the final model creation. To achieve this, numerous sets of images were captured in diverse environments to create the template and to assess each approach. To illustrate the progression through each of the approaches, a core set of images was used to enable a suitable comparison to be undertaken between each approach.

At each stage, the final models are discussed from a number of perspectives including the realism of the models, the complexity, in terms of the number of polygons used, the reconstruction process and how of the model is animated.

The implementation is carried out in the Java programming language. The underlying reasons for the choice and Java's practicality are discussed throughout this chapter. The primary reasons behind this decision include the fact that Java is platform independent and thus users with different devices can avail of the software. Moreover, it is envisaged that a final application could be made available to a home-user via a web interface as a Java Applet.

Finally, the results are analysed in comparison to some other human modelling approaches detailed in Chapter 3. Although direct comparison is not possible, the comments of the other

authors are useful to highlight areas that have been improved upon.

## 5.2 Image Capture

The capture of the images is important in determining the quality of the model. In general the larger the image, the higher the quality of the captured data which is available. Currently, the quality of images that can be captured with off-the-shelf digital camera, camera phones etc, is continually increasing. Examples of the image sizes available with current camera enhanced mobile phones and webcams are presented in the table 5.1.

Manufacturer	Model	Image Resolution	Photo Album Size/ Camera Type
Nokia	6630	1280 × 960	10MB shared memory
	7650	640 × 480	3.6MB
	6820	352 × 288	N/A
Motorola	V1050	1280 × 960	up to 256MB shared memory
Samsung	Z140v	640 × 480	50MB shared dynamic memory
	E335	640 × 480	3MB shared memory
Logitech	Quickcam	640 × 480	webcam
	Pro 5000	640 × 480	web cam <sup>1</sup>
Creative	several	640 × 480	webcam

Table 5.1: Image sizes from a sample of currently available, camera phones and webcams.

In the experiments undertaken in this section, a “*Canon PowerShot G2*” camera is used to capture the images used in the reconstruction process. The option to use the raw format was not available because, in the majority of off-the-shelf cameras, this is not a standard option and thus it would not be generally applicable. A selection of the camera’s attributes is presented in table 5.2. The images that are used in the creation of the models are in the JPEG format<sup>2</sup>, as this is the most common format used to store the captured images. This format is not immune to noise and the compression process used can introduce blocking effects to the images.

The size of the images that are used for testing is based on knowledge that was available at the commencement of this research. This knowledge was gained by examining the specification of the then available camera enhanced mobile phones, web cams and digital cameras. Additionally, the size of the images used by Hilton et al. (1999) provided a guideline for the size of images required. Hilton et al. (1999) used 756 × 582 pixels images. This provided a facial resolution of approximately 40 × 40 pixels. Images of this size are not available on the majority of off-the-shelf digital cameras and in the specifications for camera phones. The then next generation camera enhanced mobile phones specification included specification for images with 640 × 480 pixels. This improved on the exiting common image size of 320 × 240 pixels. In addition to this, most

<sup>2</sup>Joint Pictures Expert Group. The JPEG standardises the compact representation of image data. Methods exist within Java for reading in JPEG image data, accessing the pixel level information and rendering it to the screen

Resolution		Compression		
		superfine	fine	normal
Large	2272 × 1704 pixels	2002 KB	1162 KB	556 KB
Medium 1	1600 × 1200 pixels	1002 KB	558 KB	278 KB
Medium 2	1024 × 786 pixels	570 KB	320 KB	170 KB
Small	640 × 480 pixels	249 KB	150 KB	84 KB
Raw	2272 × 1704 pixels	2862 KB		

Table 5.2: Image sizes and associated compression ratios for the Canon PowerShot G2 camera. The camera has a fast f2.0 3x optical zoom lens (34-102mm).

digital cameras and web cams examined provided images with  $640 \times 480$  pixels<sup>3</sup>. This image size was adopted as the default image size for the creation of virtual humans. The image size of  $320 \times 240$  pixels was not considered, as it does not provide a suitable level of detail to reliably extract the shape of an individual.

In the approaches discussed in Chapter 4, the camera position is assumed to be fixed between each capture. To achieve this, the camera is placed on a tripod to ensure that its position remains the same for each image in a data set. This is not unrealistic, as in a home environment; an individual can place the camera on a table or other item of furniture and capture the data, for example using a webcam.

In the described approaches, the individual stands approximately  $3m$  from the camera. Specifying the distance between the camera and the individual is important for a set of images. This ensures that each of the images is created under approximately the same projection. This is illustrated in Figure 5.1. This takes account of the standard projection that is available with cameras and permits the capture of the individual in a single image. It is possible to change this distance depending on the particular individual that is being captured, for example capturing the images of a child will result in a large area about the individual that is not important in the reconstruction process but the shape of the individual can be clearly seen within the captured images. If the distance from the camera is changed it is possible that more accurate information will be extracted, although this will result in the individual appearing bigger in the virtual world, unless additional information such as the height of the individual is provided. This underpins the approach taken by Lee, Goto & Magnenat-Thalmann (2000). Moreover, if the captured data is used primarily to texture the model of a character in a game, then the captured data can be scaled appropriately without requiring an additional distance measure.

<sup>3</sup>Web cams have not been used in the experiments to date since the use of the wide-angle lens introduces distortion at the edges of the image. In addition, different projection models would be required.

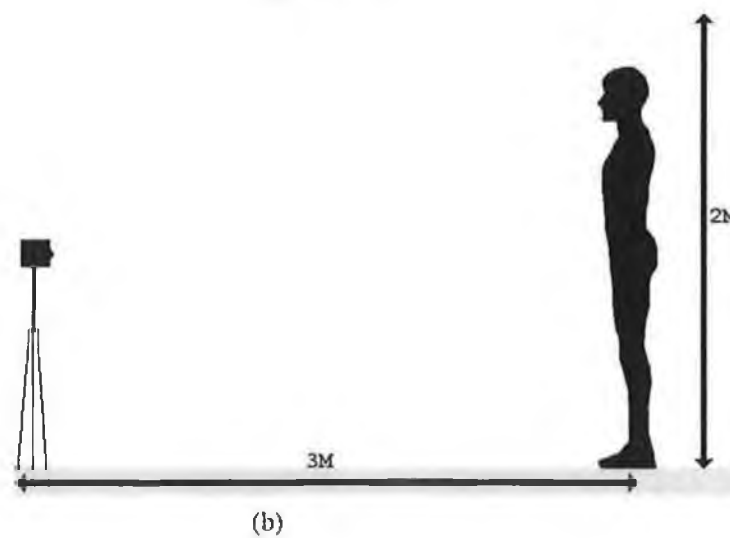
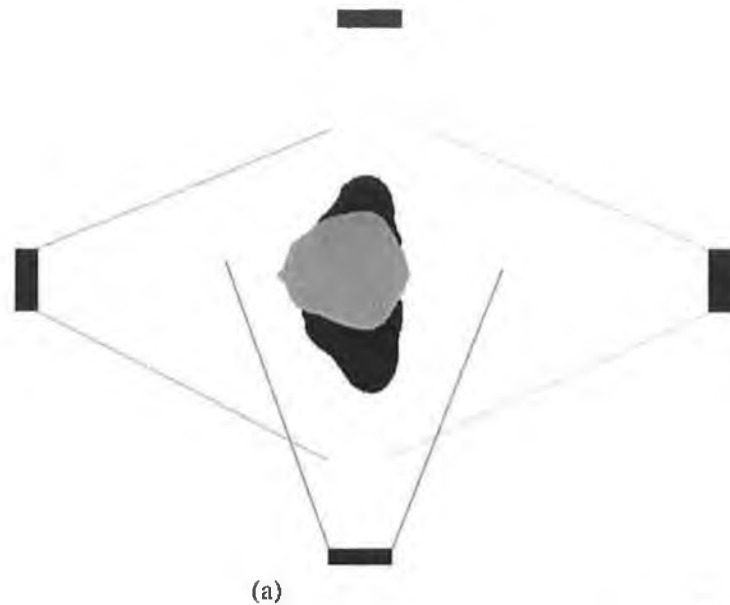


Figure 5.1: (a) shows the essence of the capture process and how the four views of the individual captured, although in reality the capture system consists of a single camera, as in (b) where the images are captured against the same background.

Working with images of  $480 \times 640$  pixels presents some difficulties regarding the accurate extraction of the individual from the background because, in general, a large part of the image is occupied with background and the boundaries between regions are not clearly defined. These effects at the boundaries result from shadows or effects introduced by the image compression algorithms, making it difficult to have a clean segmentation of the individual from the background. This can result in parts of the background being textured to the individual. If higher resolution images are used, then the effects of this problem are reduced. In addition, dealing with compressed images presents other problems, including the fact that some of the finer details are lost. As previously mentioned, the size of the image makes it difficult to extract an individual's facial features as the size of the facial region is reduced below that used by Hilton et al. (1999) and, unlike the approach of Lee, Goto & Magnenat-Thalmann (2000), an additional facial image is not used to enhance the model.

To provide a flexible solution to the capture, the number of images required for the creation of a virtual human was reduced to a minimum. To achieve this, it is necessary to capture the maximum information from a limited number of views. The priority information is the shape information and the texture. The shape information is important to enable the creation of models that are unique to a particular individual and the texture information is important in creating a realistic human model. In approaches 1 and 3 described in Chapter 4, the shape information is sacrificed in respect of the texture information because this is used to texture a generic human model. In the other approaches the active contours are used to reliably extract the individual's shape. In the approach of Hilton et al. (1999), four views of the individual are captured. The most important view is the front image, as this contains the face. The other images provide additional texture and shape information. In the approach of Lee, Goto & Magnenat-Thalmann (2000), three images are captured, the front, back and a side view for texturing the body. The side view is used to texture both sides of the individual.

In Figure 5.2, the images that are throughout the remainder of this chapter are presented<sup>4</sup>.

---

<sup>4</sup>It should be noted that other images will be introduced to illustrate particular aspects of an approach

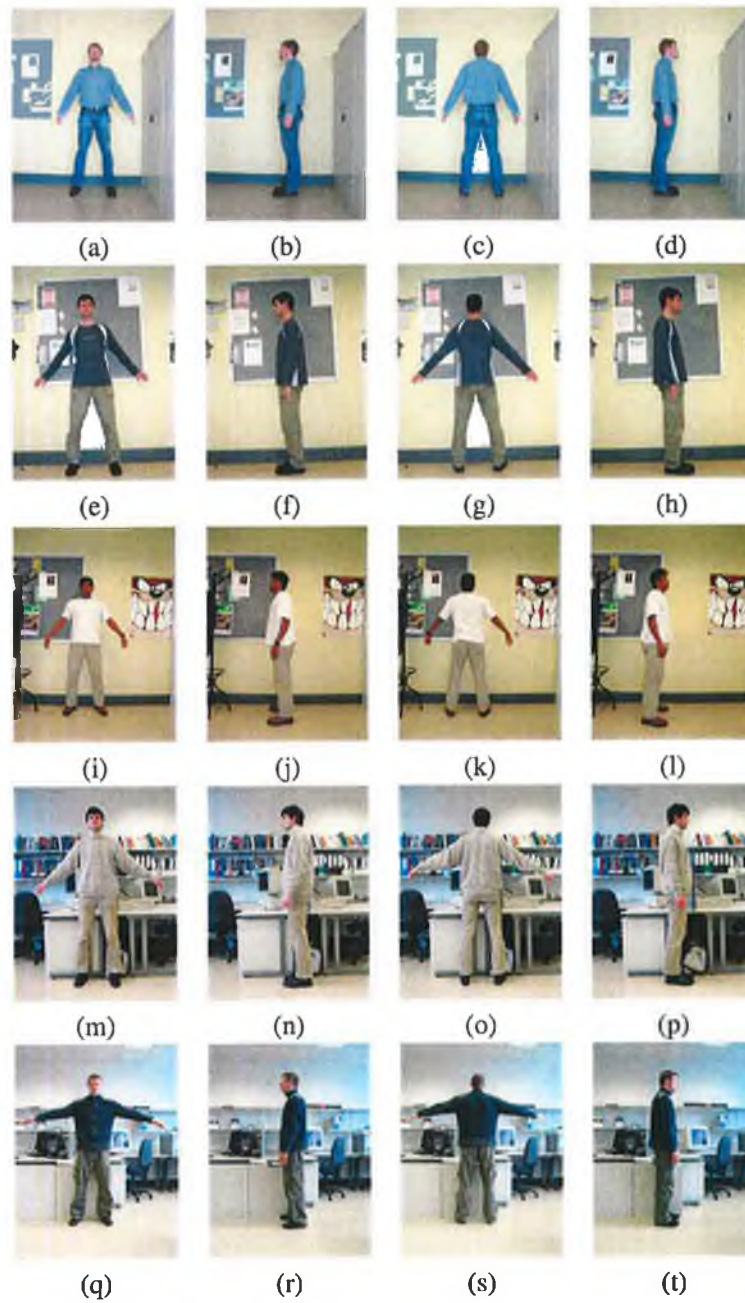


Figure 5.2: The images in this Figure constitute a core set that are used within the section.

### 5.3 Extraction of an Individual from a Real Environment

To generalise the approach of Hilton et al. (1999) and facilitate the automatic creation of human models in any environment it was first necessary to examine how the individual and the background can be separated and in what conditions an individual can be reliably extracted from the environment. This resulted in the classification of different backgrounds and the examination of the effects of different filters and pre-processing steps that can be incorporated to extend the approach to more cluttered backgrounds.

The extraction of an object from a cluttered background can be simplified if prior knowledge is available (Sonka et al. 1999). The prior knowledge that is considered important in this situation is that the individual is positioned at approximately the centre of the captured images and adopts a standard pose. The front pose is illustrated in Figure 4.9. Each individual has characteristics that can be classified as common and features that are more specific and while these features may be used to distinguish one individual from another and to locate facial and skin regions in images they, in general, cannot be used to reliably extract a complete individual from an image.

It is assumed that the individual is positioned at the centre of the image then it is valid to assume that the information at the edge of the images can be classified as background. The importance of this information depends on the level of clutter in the background. If the background is primarily uniform then the information at the edge of the image will share a high correlation with the remainder of the background image data. Thus in this situation it is possible to use simple background and foreground classification to reliably extract the individual from the background. As a first stage to classify the level of clutter in the images a Gaussian filter was used to smooth the captured images and to enable a simple region growing algorithm to be implemented, see Algorithm 3. In particular, the smoothing of the image ensured the effects of noise were reduced, thus reducing the number of small regions in the images.

---

**Algorithm 3** Simple Region growing Algorithm

---

Convert the image data to greyscale image  
Apply Gaussian filter to smooth the image  
Represent the image as a 2D matrix

```
regionCounter = 0
for i = 1 to image_width - 1 do
  for j = 1 to image_height - 1 do
    if pixel (i, j) not part of region then
      regionCounter = regionCounter + 1
      Call regionCreation function
    end if
  end for
end for
```

---

One measure to classify the clutter is assigned based on the number of region that are in a particular image. The results for the images in 5.2 are shown in Figure 5.3. In comparison to the images used in Figure 4.4, in which the number of regions is 37, the number of regions<sup>5</sup> is

---

<sup>5</sup>The colours of the regions should not taken as an indication of a properties of a particular region because the colour



---

**Algorithm 4** Simple Region growing Algorithm: regionCreation function

---

input data: pixel, regionCounter

Examine the 8 neighbours of this pixel

**if** neighbouring pixel has not been assigned a regionCounter **then**

**if** neighbouring pixel has same value as current pixel **then**

        neighbouring pixel is assigned same regionCounter value

        Call regionCreation function

**end if**

**end if**

---

substantially higher in cases shown Figure 5.3. As the level of clutter increases this is no longer considered a valid approach to separate the individual from the background.

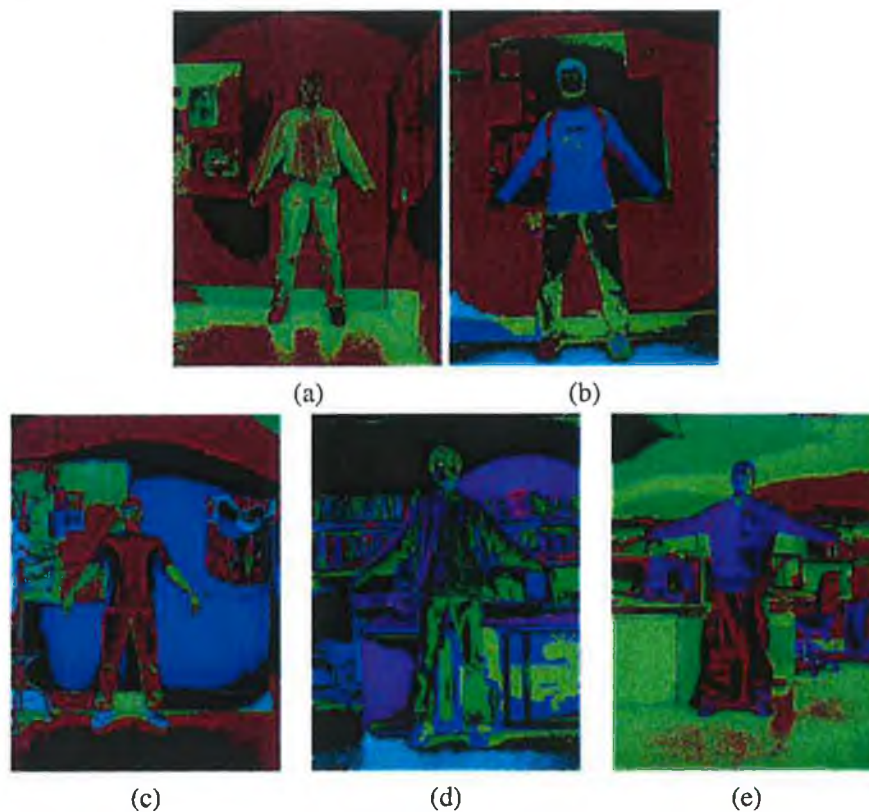


Figure 5.3: Example of individuals against backgrounds with different levels of clutter. In each case the number of regions also accounts for regions in the individual clothes. In (a) the number of regions is 196, in (b) the number of regions is 348, in (c) the number of regions is 271, in (d) the number of regions is 418 and in (e) the number of regions is 280.

Even in the simple realistic situations as shown in Figure 5.2 (a) to (l) the background includes two principal regions the floor and the wall. Other smaller regions exist such as the skirting board that is broken in to two or three smaller regions because of the pose adopted by the individual. All of these regions can be accounted for, if the boundary of the image is traced and the regions are allowed to grow. The only exception is the parts of the background that are between the legs. This can be simply established by analysing the mean colour of each region and grouping the regions is assigned randomly since the number of regions is not know prior to applying the algorithm.

that have low colour variation.

In the situation shown in Figure 4.4 applying a Gaussian filter to the image provides a reliable approach to locate the individual in the image. Although depending on the lighting conditions and the individual's distance from the wall in the background the effect of shadows have a more or less noticeable effect on the extraction. In certain situations this results in parts of the background being classified as being part of the foreground (individual). In addition, to this smoothing of the image blurs the boundary between the foreground and background. As the level of clutter increased these effects are more evident and other shadows were introduced and the number of regions increased and it became increasingly difficult to separate the individual from the background.

An alternative, automatic, classification of the background is achieved by analysing the image at different horizontal and vertical locations. If the colour variation across a particular line is high then it is assumed that the background is cluttered. Alternatively, if the colour variation is low then the background can be classified as uncluttered. Enabling the application of the simple background segmentation algorithm, see Algorithm 3.

An measure of the level of clutter is illustrated in Figures 5.4 and 5.5. The image variation was extracted using the neatvision<sup>6</sup> image analysis software (Whelan & Molloy 2000)

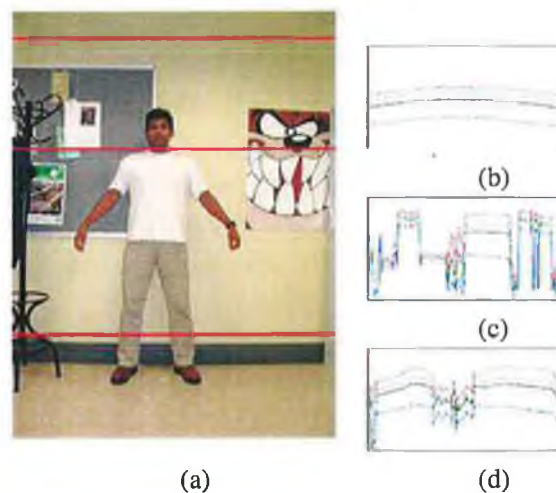


Figure 5.4: (a) shows an image with three measures of clutter made at three different height levels. In (b), (c) and (d) the level in variation of the colour components at a particular height are shown. In each case the three colours, red, green and blue represent the respective components and the average value is also indicated using the black line.

The approaches described in this section aim to provide a semi-automated approach to assessing the level of clutter in the images. This is important to determine if it is possible to apply a simple region technique. The number of regions in Figure 5.3 makes it impractical to extract the background using the method in Algorithm 4. In Figures 5.4 and 5.5, a simpler technique to establish the level of clutter is shown and although it does not facilitate the use of the region algorithm, it can be used to set the  $\sigma$  parameter and  $T_1$  and  $T_2$  for the Canny edge detector. If the level of clutter is high then the lower threshold  $T_1$  is raised and the  $\sigma$  parameter is also increased.

<sup>6</sup>NeatVision is an Image Analysis & Software Development Environment, [www.neatvision.com](http://www.neatvision.com)

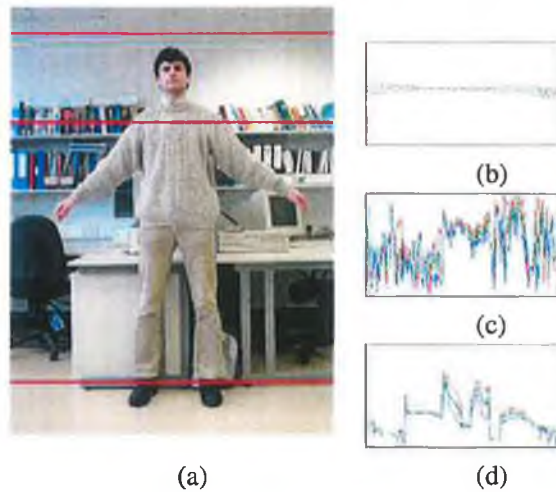


Figure 5.5: (a) shows an image with three measures of clutter made at three different height levels. In (b), (c) and (d) the level in variation of the colour components at a particular height are shown. In each case the three colours, red, green and blue represent the respective components and the average value is also indicated using the black line.

### 5.3.1 Background Subtraction

At this point the possibility of using background subtraction was considered. In this approach an additional image of the background is captured before the images of the individual were captured. The use of background subtraction is analogous to the use a blue screen background and depending on the lighting effects and shadows introduced when the individual stands in front of the camera and it makes it difficult to reliably extract the individual's shape. This is illustrated in Figure 5.6. In addition, it required the user to capture an additional image and reduces the flexibility of the approach.

The use of background subtraction eliminates large parts of the background and makes it easier to locate the individual in the image. It does not provide accurate definition of boundaries and thus the data cannot be reliably used to describe the shape of the individual. Moreover, with the application of additional image processing tools it is a challenging task to identify the arms and other parts of the individual.

### 5.3.2 Application of Edge Detectors

Edges provide a powerful tool for examining the information that is contained in the images. Examining the edges within the image provides an alternative approach to extracting the shape information. The edges can be used to provide a coarse and generally disjointed view of the image. In particular, the edges within images generally constitute the boundary of one region with another. This edge information can be corrupted by noise and the strength of the edges can change depending on the difference in colour between one region and its neighbouring region.

In the uncluttered background the edges provide a detailed boundary description of the individual. Although on close examination the edges are disjoint and in situations where the individual is wearing clothing that is similar in colour to the background the edges are non-existent. Areas

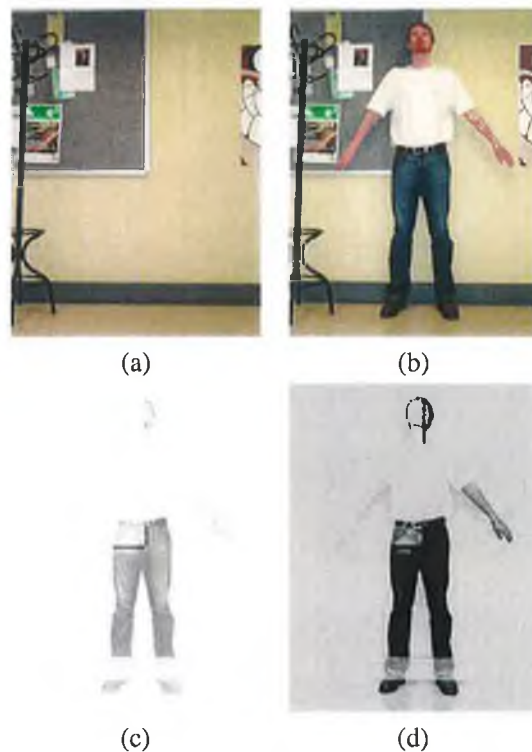


Figure 5.6: The results for background subtraction. (a) shows the background image (b) shows the individual against the same background (c) shows the result of a direct subtraction (inverted for clarity) and (d) shows the same image as (c) with the brightness and contrast manually adjusted to try and improve the separation of the individual from the background.

around the head provide a large number of disjoint edges but these cannot be easily combined. In the situation of the cluttered background the number of images is dramatically increased and the edges defining the shape of the individual are not easily identifiable.

The edge maps generated for the front view for each set of images in Figure 5.2 are shown in Figure 5.7. In all cases shown some post processing of the edges is necessary to remove the edges in the background. The most appropriate approach is the application of a template to group the relevant edges and to provide a more complete boundary description. This is the subject of the next section.

### 5.3.3 Testing of the Active Contour Implementation

The active contours as introduced in Chapter 2 provide an ideal method of combining the edges in Figure 5.7 to generate a boundary that describes an individual. The active contours were implemented using a dynamic programming approach as described in Section 4.3.2. This section provides a description of the development and the testing that was undertaken to ensure that firstly the basic implementation of the active contours was working as expected. Then additional constraints that are essential for the template to reliably extract the individual from the background are tested.

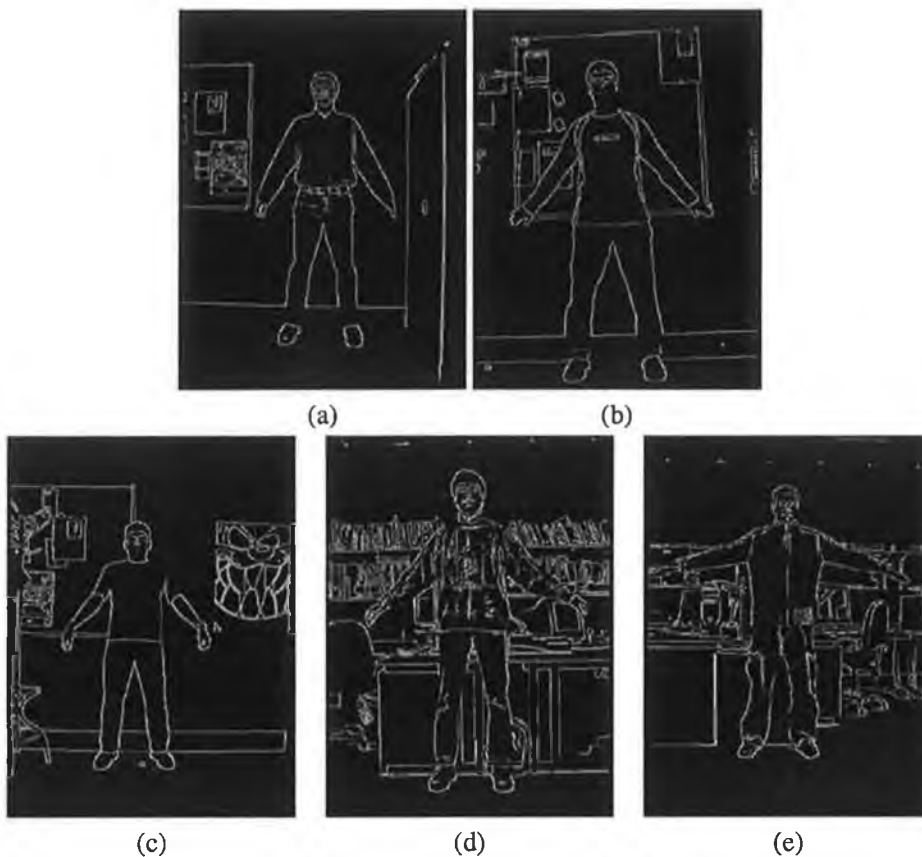


Figure 5.7: The results of applying the Canny edge detector to the input images. In each case the parameters for the Canny are  $\sigma = 1$ ,  $T_1 = 100$  and  $T_2 = 255$

The active contour is constructed as a series of nodes that are stored in a Java vector list <sup>7</sup>. Each element in the vector list contains the 2D position of a control point, the *alpha* and *beta* weights associated with the internal energy, a control value that is used to indicate if there are any restriction on the directions that a particular control value can move. A schematic of this information is shown in Figure 5.8.

Within the general implementation, the values of *alpha* and *beta* in Equation 2.3 are set to be 0.5. The values are included within structure for a particular node to facilitate the incorporation of additional information such as the bending energy that can be used to determine a corner within a particular implementation (Lam & Yan 1994) where the value of the *beta* parameter is altered if the bending energy is above a set threshold. Within this implementation corners are not considered an important feature because the shape of the individual has in general, no right-angled or strong corners.

<sup>7</sup>A vector list is a way of storing unformatted objects in a link-list like structure, only more carefully and in a more structured manner. The Vector class is used to replace stack operations that would possibly be required in other language implementations of the algorithm. The Vector Class has several methods that facilitate the updating of the list including the ability to add and remove elements at particular locations when required.

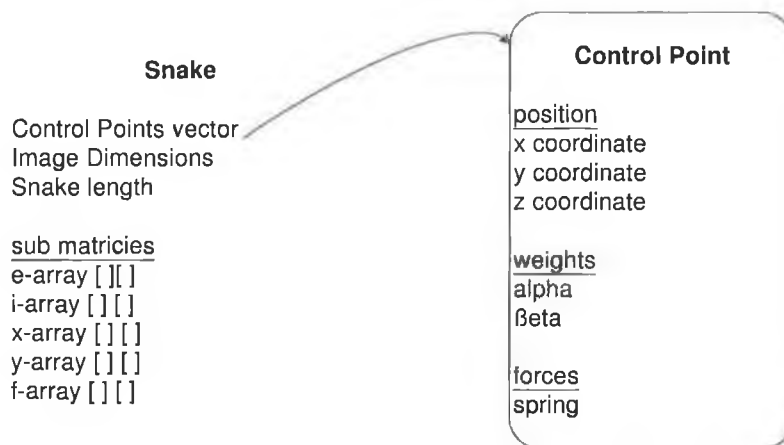


Figure 5.8: A schematic showing the information contained in the snake and the information associated with a control point.

### The Internal Energy

To verify that the internal energy term in the implementation is operating correctly the active contour was constructed with the external energy at each location set to zero. Thus according to Kass et al. (1987), the internal energy should move to minimise the distance between each node. The initial position of a contour is shown in Figure 5.9. In this test the image data has no effect on the movement of the control points. The stages in Figure 5.9 show the initial contour and additional snapshots as the minimisation process progresses. Figure 5.9 (d) shows the image one of last views of the contour before it reaches the minimal position<sup>8</sup>. This minimum is reached after 60 iterations.

Williams & Shah (1992) introduce a constraint to reduce the convergence of the active contour to a point in. This is achieved by including the average distance between each of the nodes within the internal energy term. This results in the internal energy reaching a minimum when the distance between each point is equal to the average distance. The effects of this constraint are shown in Figure 5.9 (e) to (g). In this Figure the control points firstly arrange into two lines, as this is the shortest distance between two points. Then the distance between the points approaches the average distance. Since only the internal effects are considered at this stage, the two lines converge to each other, although the contour does not converge to a point. The effects of this constraint are shown in Figure 5.14 where the control points are evenly distributed along the length of the contour.

### The External Energy

The external energy term in the active contour combines the forces that attract the contour to particular intensity values in the image, to areas or points with high edge intensity values. The attraction to these features reduces or increases the minimising effects of the internal energy. In particular, the external energy is defined as having a high attraction to high intensity edges in the edge feature map. This is illustrated in Figure 5.10.

<sup>8</sup>The last view is not shown as all the control points have contracted to a single point.

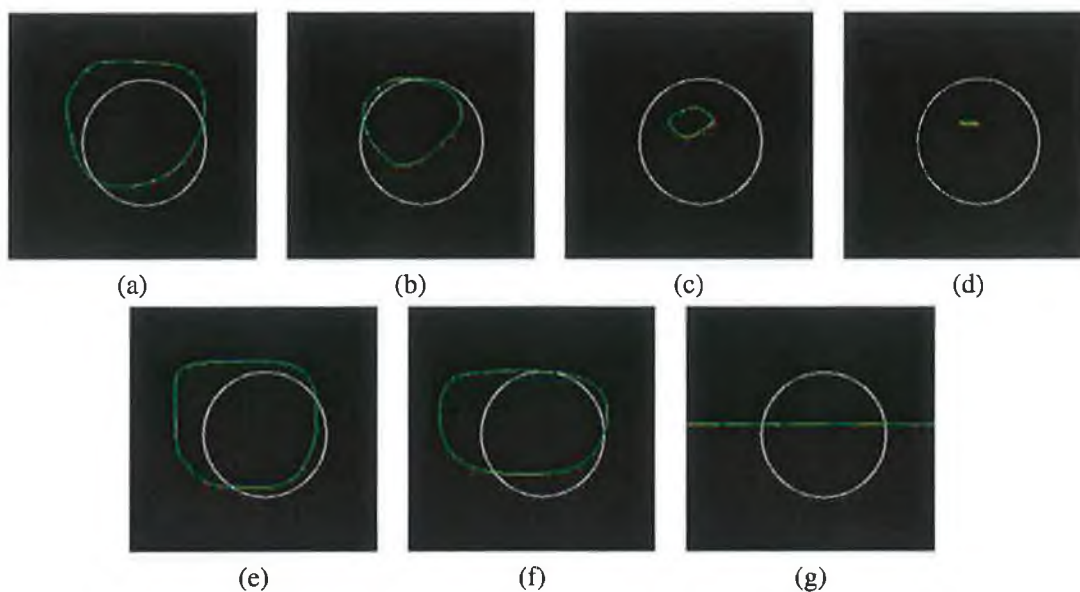


Figure 5.9: Demonstration on the effects of the internal energy. (a) to (d) show that if no constraints are included then the internal energy has the effect of collapsing the contour to a point (e) to (g) Show the effects of constraining the internal energy by encouraging an even spacing between the control points.

The expansion or balloon force introduced by Cohen (1991) is not directly incorporated into the external energy term because the template may be placed inside or outside the boundary contour of the individual and minimises to the correct solution. By constructing the external energy to have strong attraction to the high intensity edges the active contour can expand if it is placed within the contour. This is illustrated in Figure 5.11.

To encourage a control point to move from one edge pixel to another pixel with a higher intensity value the value of the external value is weighted by the intensity value. Thus, if an edge pixel in the search area has a higher intensity value than the current value, the control point will move to this location. This ensures that the control point moves towards the edge with the strongest intensity.

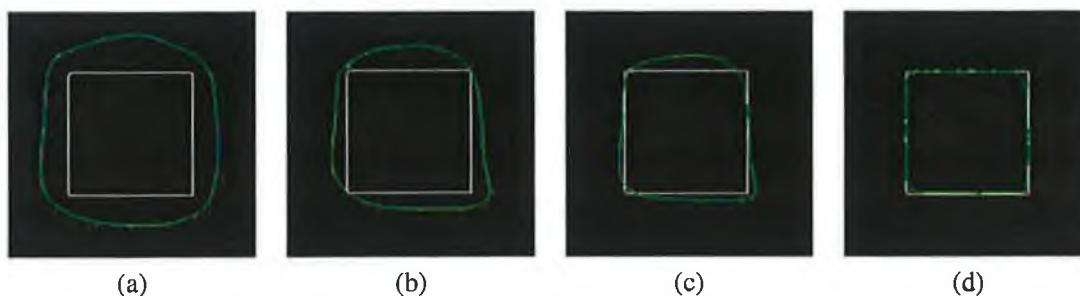


Figure 5.10: (a) shows the initial position of the contour, (b) and (c) show the progression of the contour and the effect of the external energy on the minimisation process. This minimisation took 41 iterations.

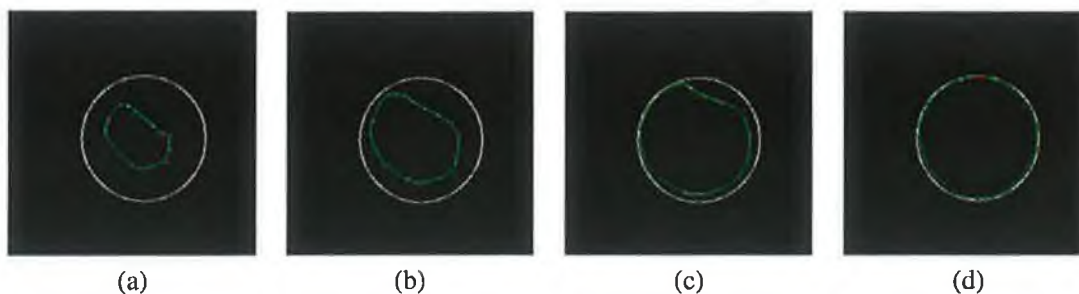


Figure 5.11: (a) shows the initial position of the active contour inside the circle and (b), (c) and (d) show that the attraction to high intensity edges counteract against the internal energy constraints and it expands to find the boundary. This minimisation took 51 iterations.

In the original dynamic programming implementation of Amini et al. (1990) the edge information was only examined within an  $n \times n$  search space about a control point. This added an additional  $O(n^2)$  operations at each iteration. In addition, this procedure ensured that the contour converges slowly to the final state and that the control point can move along the length of the contour as opposed to perpendicular to the contour. To counteract against these effects our implementation of the active contour employs a linear search space to locate the nearest high intensity edge pixel. This ensures that the active contour is always moving to minimise the length of the contour and increases the convergence time of the snake. The search space is further reduced by the introduction of controls that reduces the direction that the contour can move. This is discussed in greater detail in Section 5.3.4. If at a particular iteration, no high intensity edge is found within the search space then the external energy term is set equal to zero and the internal energy term dominates the minimisation procedure. The directions that are searched at a particular control points are shown in Figure 5.12 (a). In Figure 5.12 parts (b), (c) and (d) show an instance of an active contour moving over weaker edges towards the strongest edges. In Figure 5.12 (a) the search space was extended to be 50 pixels for the purposes of illustration. The final position of the contour is achieved after 49 iterations.

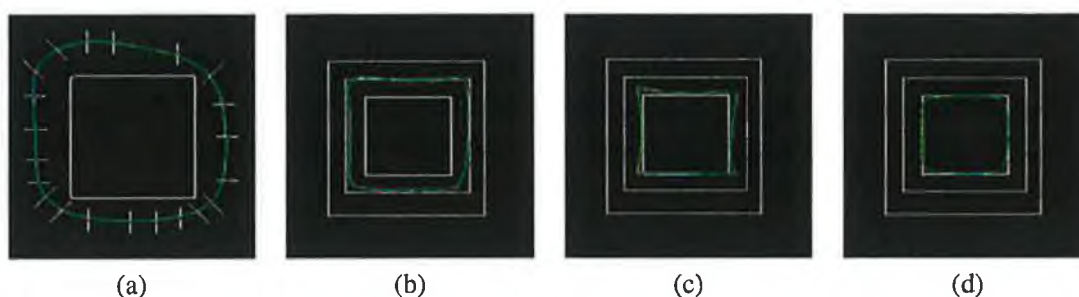


Figure 5.12: (a) shows the directions that are searched for high intensity edges (b), (c) and (d) show the progression of the active contour over weak edges to the strongest edges. The outer and middle rectangle has an edge intensity of 155 and 200 respectively.

Figure 5.13 shows the effects of moving a point from the equilibrium position. The points are at rest on the boundary of the square and when dragged away from this position they return to



equilibrium after 32 iterations.

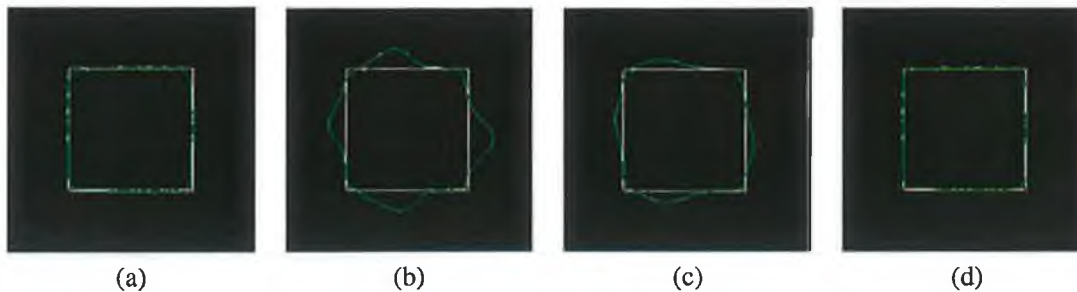


Figure 5.13: (a) shows the active contour at rest on the boundary of the square, (b) shows four points dragged from the equilibrium position, (c) shows the position after 20 iterations and (d) shows the position after 32 iterations.

### Combining The Internal and External Energies

The effects of combining the energy terms is shown in Figure 5.14. The effects of combining the internal and external energies are that when the external search is unable to find an edge in a particular location the internal energy will provide a new location to search for the energy and cause the length of the contour to reduce. This has the advantage that the external energy search space can be reduced. In addition, with the constraint forcing the distance between the control points towards the average, the structure of the active contour becomes more regular. This can be seen in Figure 5.14 (c) where the distance between the control points tends to be evenly spaced.

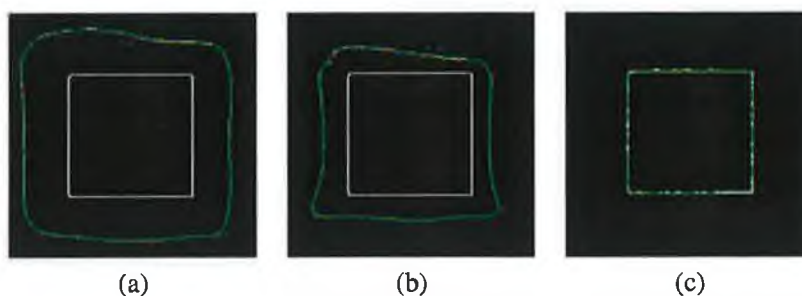


Figure 5.14: (a) shows the initial position of the contour which is distant from the edges, (b) shows the effects of the internal energy as it minimises a constant rate until the search space contains high intensity edges, in (c) the external forces start to dictate the minimisation process. (d) shows the final position of the snake and the approximately even spacing of the control points.

### Addition and Removal of Control Points

In this section the control points of the active contour are examined to determine if it is necessary to introduce additional control points to improve the convergence of the active contour or to remove control points if the contour is representing a straight edge. Figure 5.15 shows the situation when the energy in the contour has reached a minimum but the active contour does not accurately

describe the shape in the image. This is particularly evident at the corners of the square in Figure 5.15. The principle that is applied in this situation can be applied in the situation that not all the control points are located on areas of high intensity edges.

The active contour is examined at each location. If a control point is located on a strong edge then the second order term of the internal energy is examined. If this value is above the average value for the contour then the interpolated positions of the contour are examined to ensure that they are located on a significant edge. This results in the insertion of four new control points that are marked yellow in Figure 5.15 parts (c) and (d), i.e. one in each corner of the square. The minimisation process is reactivated to improve the attraction to the boundary and the improved accuracy is shown in Figure 5.15 (d).

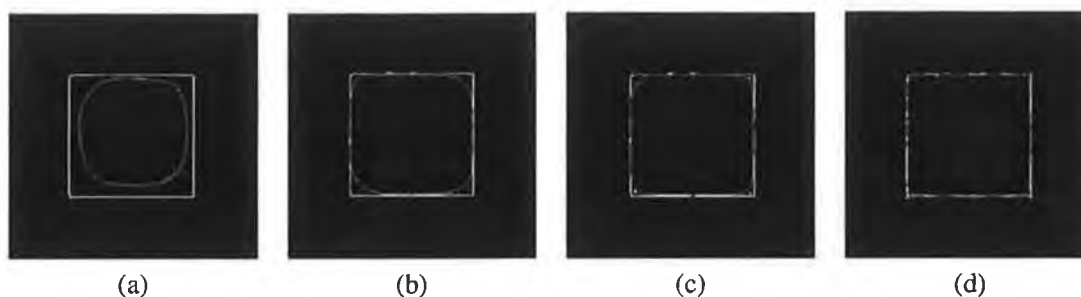


Figure 5.15: A demonstration of how additional control points are inserted. (a) shows the initial position of the contour, (b) shows the final position of the contour, (c) shows the four inserted control points, marked yellow, (d) shows the results of the minimisation considering the new points. Note: in (d) the control point in the bottom right corner is located at the corner of the square but the B-spline does not interpolate this point.

The removal of control points is facilitated by the examination of the control point's position. If a number of control points are located on an edge and have approximately<sup>9</sup> the same  $x$  or  $y$  coordinate then it is possible to reduce the number of control points along a particular edge. Although this procedure has been implemented, it is generally not used within the final application because this situation rarely occurs. The removal of control points is shown in Figure 5.16. Figure 5.16 (a) and (b) show the evolution of the contour and the final position of the contour is shown in part (c). Then between parts (c) and (d) the control points along the sides of the square are reduced. This technique can be extended to describe a line at any orientation by considering the slope through a number of consecutive control points.

To ensure that the contour does not kink during the evolution of the contour a procedure is implemented at each iteration to decide if two control points occupy the same location. If this is the case then one of the control points is removed to ensure that the contour does not kink. Before a control point can be removed it is necessary to check the index of the control points. If the control points are consecutive in the ordering of the contour then a control point can be removed. Otherwise the control points are permitted to occupy the same position, but in general, when this situation occurs the possibility exists to split the contour into two contours. However, in the simple shapes shown in this section the possibility of the contour splitting is not considered as the shapes

<sup>9</sup>A difference of one pixel is tolerated.

extracted are not composed of multiple objects and when extracting the individual from the image a single object is sought.

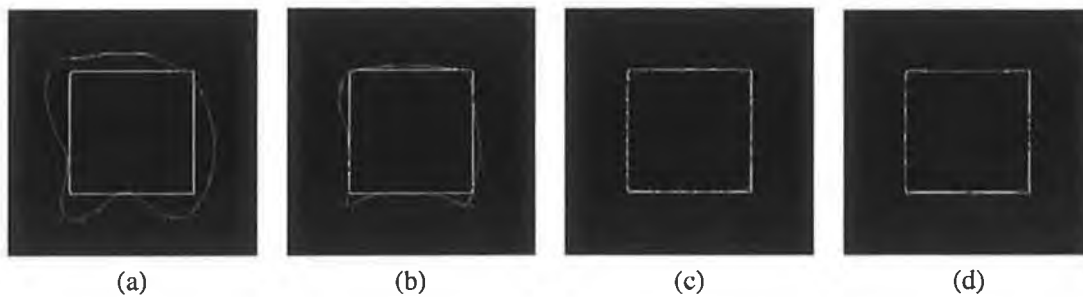


Figure 5.16: A demonstration of how particular control points are removed from the contour formation. (a) shows the initial position of the contour, (b) shows an intermediate position of the contour, (c) shows the final position of the contour and (d) shows the results of removing control points along the sides of the square.

### 5.3.4 Template Generation and the Testing of the Constraints

The pose that the individual adopts is important to reliably identify key features of the body. This provides the possibility of defining a template that incorporates general shape information about the individual. This template has to be flexible to adjust to the shape of different individuals and to adapt to the different environments. The template needs to reliably extract an individual's shape and to overcome information that is missing, for example faint edges that occur when part of the individual's clothing has a similar colour to the background.

The pose that the individual adopts is shown in Figure 4.9 in Chapter 4. The template is generated by interactively fitting a B-spline curve around the individual in sample images. This is achieved by specifying the control point locations on either the original image or on the associated edge map. Table 5.3 shows examples of the images captured to create the template. The middle column in Table 5.3 contains the points that are interactively fitted to the captured data. During the fitting process the user has the option to toggle between the edge map and the original image. This enables the user to match the control points on the exact edge. Once the points are placed a B-spline curve interpolates the control points. To ensure that the control points are on the highest edge in the region in which they are placed the energy in the active contour is minimised to accurately locate the boundary.

The active contour is converted to a boundary map and it is analysed to find the key features such as those in the third column of Table 5.3. A full list of key values are shown in Figure 4.11. These values are combined with the centroid of the contour to provide the scaling values that are used to define the mean template. All scaling and positioning of the control points is done relative to the centroid of the template. Not all the key features are used to fit the template but they are used to separate the individual's silhouette into different parts for the texturing.

In Sections 4.3.1 and 4.3.2, the rationale for restricting the pose of the individual are discussed. Restricting the pose significantly reduces the variations in poses adopted and thus enables the initialisation of the template by simply scaling the default template. The exception to this is the initialisation of the arms, which are initialised separately. In (Cootes et al. 1992), each of the models that constitute the training set are first aligned and then normalised to generate the mean template. This is not necessary for the generation of the mean template since in each case the individual is directly in front of the camera. Furthermore, in (Cootes et al. 1992) and (Baumberg & Hogg 1994), the fine detail is not extracted, and only the position of landmark points are considered suitable for the generation of the mean shape.

During the initial fitting of the template to the images data, different ratios were experimented with, to enable both the manual and automated fitting of the template to the image data. As described in Section 4.3.3 using the manual initialisation the horizontal scaling factor can be extracted using six points. However, with the automated approach the same information is not available. The height of the bounding box provide the vertical scaling factor and width of the bounding box provides the location of the hands but this is not sufficient to provide an accurate horizontal scaling factor as the position of the arms varies between individuals.

A selection of ratios that were experimented with to determine appropriate scaling values of the template are presented next. The mean of the *max\_vertical\_left* and *max\_vertical\_right*

is combined with the *min\_vertical* to provide a vertical scaling of the template. The horizontal scaling is initially calculated using the *max\_horizontal* and *min\_horizontal* values which correspond to the left and right most position of the arms but since the relative distance between these two points and the width of the upper body can vary substantially from one individual to the next this measure was not considered reliable.


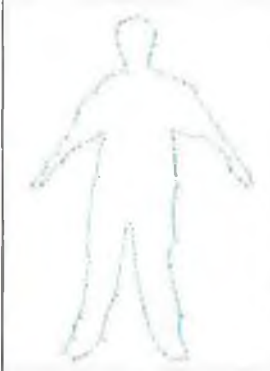

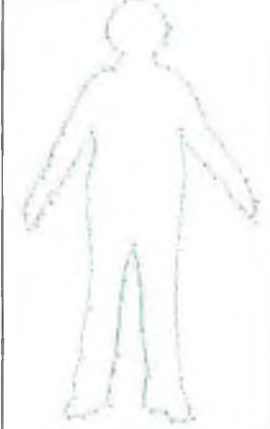


		<p>min vertical = (214, 193)  max left vertical = (270, 595)  max right vertical = (142, 593)  min horizontal = (348, 384)  max horizontal = (94, 391)  left armpit = (262, 328)  right armpit = (175, 329)  crotch = (209, 438)</p>
		<p>min vertical = (231, 176)  max left vertical = (282, 597)  max right vertical = (166, 593)  min horizontal = (353, 384)  max horizontal = (122, 393)  left armpit = (280, 313)  right armpit = (188, 314)  crotch = (232, 427)</p>
		<p>min vertical = (239, 164)  max left vertical = (319, 604)  max right vertical = (183, 600)  min horizontal = (371, 383)  max horizontal = (110, 405)  left armpit = (293, 309)  right armpit = (190, 303)  crotch = (244, 428)</p>

Table 5.3: This table shows some of the images used to generate the template.

To scale the template using the automated initialisation technique, it is necessary to extract information that can be used to automatically scale the template to fit the captured data. Significant data is available from the bounding box that is discussed in Section 4.3.3. This bounding box enables the location of the head, the feet and the arms. As stated above, the location of the arms is not sufficient to determine a suitable horizontal scaling factor. Thus different ratios are used to determine a horizontal scaling factor for the template. This is illustrated with the ratios for the three examples shown in Table 5.3.

$$\begin{aligned}
 \text{horizontal\_ratio} &= \frac{\text{min\_horizontal.x} - \text{max\_horizontal.x}}{\text{l\_armpit.x} - \text{r\_armpit.x}} \\
 Ex_1 &= \frac{348 - 94}{262 - 175} = 2.919 \\
 Ex_2 &= \frac{353 - 122}{262 - 188} = 2.675 \\
 Ex_3 &= \frac{371 - 110}{293 - 190} = 2.533
 \end{aligned} \tag{5.1}$$

where the *.x* notation indicates the *x* coordinate of a particular feature point. This is one way in indicating the variation of the position of the hands and the effect that they have on the horizontal scaling. If the same ratios are used in the creation of the horizontal to vertical ratios then the following results:

$$\begin{aligned}
 \text{ratio} &= \frac{\text{mean}(\text{max\_left\_vertical.y}, \text{max\_right\_vertical.y}) - \text{min\_vertical.y}}{\text{l\_armpit.x} - \text{r\_armpit.x}} \\
 R_1 &= \frac{0.5(595 + 593) - 193}{262 - 175} = 4.609 \\
 R_2 &= \frac{0.5(597 + 593) - 176}{262 - 188} = 5.662 \\
 R_3 &= \frac{0.5(604 + 600) - 164}{293 - 190} = 4.252
 \end{aligned} \tag{5.2}$$

where the *.y* notation indicates the *y* coordinate of a particular feature point.

The two ratios above indicate that extremes of the model are not suitable in determining a horizontal scaling factor that can be universally applied to the templates. If the distance between the armpits is considered separately then it is observed that the distance is nearly independent of the height of the individual. In effect, the average distance between the armpits of the individuals is 94 pixels in the examples shown in Table 5.3. Additionally, the average distance between the feet can be easily calculated from the bounding box because it expands to encompass the individual and the position of the feet can be determined during the initialisation procedure described in Section 4.3.3. The average distance between the feet for the three examples shown in Table 5.3 is 127 pixels. This provides the following ratio:

$$\begin{aligned}
 \text{ratio} &= \frac{\text{mean}(\text{left\_armpit}.x, \text{right\_armpit}.x)}{\text{mean}(\text{max\_left\_vertical}.x - \text{max\_right\_vertical}.x)} \\
 &= \frac{94}{127} = 0.74
 \end{aligned}
 \tag{5.3}$$

Using this ratio an estimate of the distance between the armpits can be automatically extracted. The corresponding distance between the armpits is used to horizontally scale the default template. The position of the armpits is important in the fitting process because of the significant variation of the arms. This is possibly because different individuals are more comfortable with their hands at different levels. In particular the position of the armpit provides an axis for rotating the arms to better approximate the captured data. Ideally the greater the separation between the position of the hands and the side of the body the more accurately the location of the armpits can be determined.

The positions of the armpit and the *max\_horizontal* and *min\_horizontal* are used to position the arms independently of the rest of the template. In particular, a line from the armpit to the either extreme defines an axis about which the points defining the arm can be rotated to better fit the captured data. The distance between the two points is used to scale the template. This is illustrated in Figure 5.17.

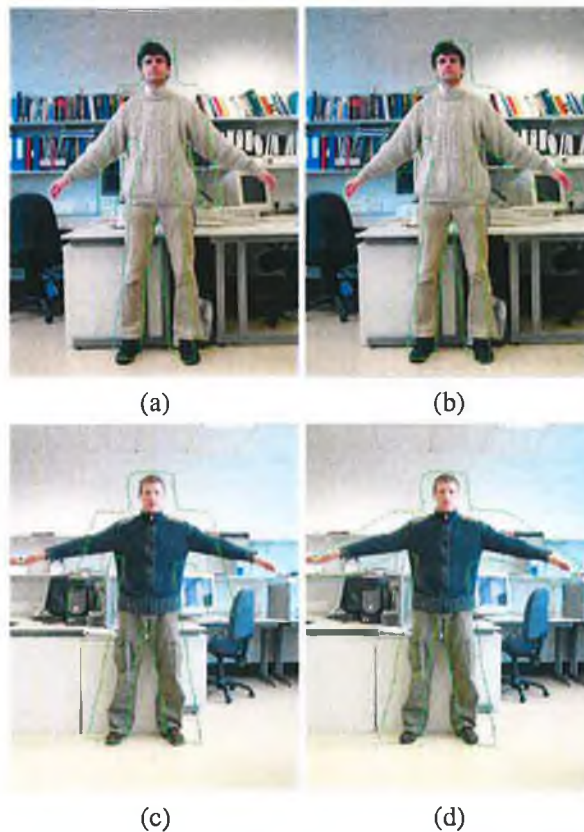


Figure 5.17: In (a) and (c) the default position of the arms is shown. Then in (b) and (d) the position of the arms after they have rotated about the y-axis is shown then scaled rotated back through the angle provided from the bounding box or from user initialisation .

### Inclusion of Constraints

This section illustrates the testing of the constraints that are described in Section 4.3.4. The constraints are introduced to stop the contours from converging to the wrong edges. The first approach to solving this problem involved the use of constraint forces between two points that could converge to the same edge as illustrated in Figure 5.18. The main difficulties that are associated with this implementation are firstly that the positions of the points on each side are not uniquely defined and they change between each individual and as the snake energy is minimised. Secondly, the force that is calculated includes a strong directional element that may cause the control point to move towards a neighbouring control point as opposed to the direction of the correct edge. The effects of the constraints are illustrated in Figure 5.18. Figure 5.18 (a) shows the initial position of the springs is shown and it can be seen that the springs are not completely horizontal and in part (b) the effects of the directional component of the force is illustrated as the control points move towards each other along the inside leg.



Figure 5.18: (a) shows the position of the springs between the legs of an individual. In this situation the and also illustrates that the forces associated will have both a horizontal and vertical element that is not suitable for constraining the position of the control points. (b) illustrates the directional effects of the controls introduced

To overcome such difficulties a second approach is considered that involves imposing restrictions on the direction that a particular point can move. There are four restrictions that can be used:

- stopping the control point moving up,
- stopping the control point moving down,
- stopping the control point moving right,
- stopping the control point moving left,

This approach reduces the directional element of the constraint forces and also means that no additional forces are introduced in the minimisation process. The effects of these forces were tested on simple images and then on real images. The results proved inconclusive, because in particular, when the template is initialised the location of the key features, namely the armpits and the crotch cannot be guaranteed to be in the correct location. Thus they may have to move in any direction to find their correct position. Moreover, when the template is initialised it cannot be guaranteed that the location of the control points along the legs or under the arms will all lie on



the same side of the edges. Thus the control points may have to move towards the opposite edge to find the minimum position.

Consequently, in the final implementation the direction restrictions are not imposed but the concept of the volcano, as introduced in (Kass et al. 1987) is used to move restrict the movement of control points. This improves the minimisation by permitting the control points to move in any direction while actively discouraging the movement towards the wrong edges. The Volcano is initialised using the bounding box dimension in the automatic initialisation. A line is plotted from the centre of the bounding box to the bottom of the box. This line provides the centre latitude for the volcano between the legs. Under the arms the initialisation is more difficult the first point that is used is the arm pit location and then a point between the inside of hand and the left or right crotch equivalent. This provides the necessary constraint to prevent the contour from moving to the wrong edge.

### 5.3.5 Template Initialisation and Minimisation

In Section 5.3.1 background subtraction involving the capture of an additional image was ruled out as a possible automatic method of separating the individual from the background because of the effects of lighting and shadow meant that the boundary of the individual is not easily identified and requires further processing for correct identification and the fact that it required the capture of an additional image. In Section 4.3.3 a method for the initialisation of the contour was presented for the front and back templates and the side templates. The front and back templates are initialised, by subtracting the front and back images from each other. The subtraction process that is undertaken does not try to accurately describe the boundary of the individual, but provides an approximation of the individual's location within the images. The results of this subtraction applied to the images in Figure 5.2 are shown in Figure 5.19. It can be seen that the subtraction removes a large part of the background and isolates the region where the individual is located. In Figure 5.19 (c), due to movement of the camera between two captures the background subtraction contains part of the background. Thus, to enable the correct initialisation the front and back views are subtracted from a side view to initialise the template. The results of this are shown in Figure 5.19 (d).

The results of this subtraction process are further improved by applying the Canny edge detector to the difference images (Canny 1986). The location of the individual is localised in Figure 5.20. In parts (a), (b), (d) and (e) the difference in the individual corresponds to the individual's position. In (c) parts of the background are also contained in the image. This occurred because the camera moved slightly between the capture of the front and back images. Figure 5.20 (f) to (j) show the same results for the side views in Figure 5.2.

The results in Figure 5.20 enable the generation of a bounding box. The bounding box is used to scale the default template and to position it correctly. All control points are not positioned edges but the positions are sufficient to enable the active contours to be attracted to the edges in each view. The bounding box is shown in Figure 5.21 with the position of the template. The arms are adjusted for each in the cases using the technique described in Figure 5.17.

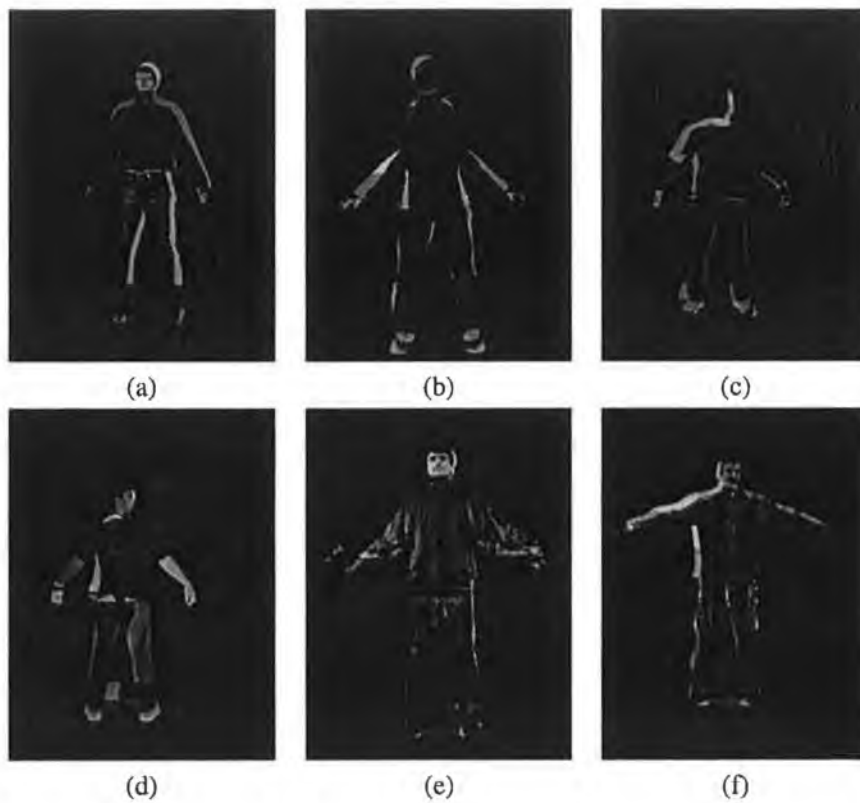


Figure 5.19: The results of the subtraction process applied to the front and back images in Figure 5.2. (a) corresponds to the subtraction of Figure 5.2 (a) and (c). (b) corresponds to the subtraction of Figure 5.2 (e) and (g). (c) corresponds to the subtraction of Figure 5.2 (i) and (k). (d) corresponds to the subtraction of Figure 5.2 (i) and (j). (e) corresponds to the subtraction of Figure 5.2 (m) and (o). (f) corresponds to the subtraction of Figure 5.2 (q) and (s).

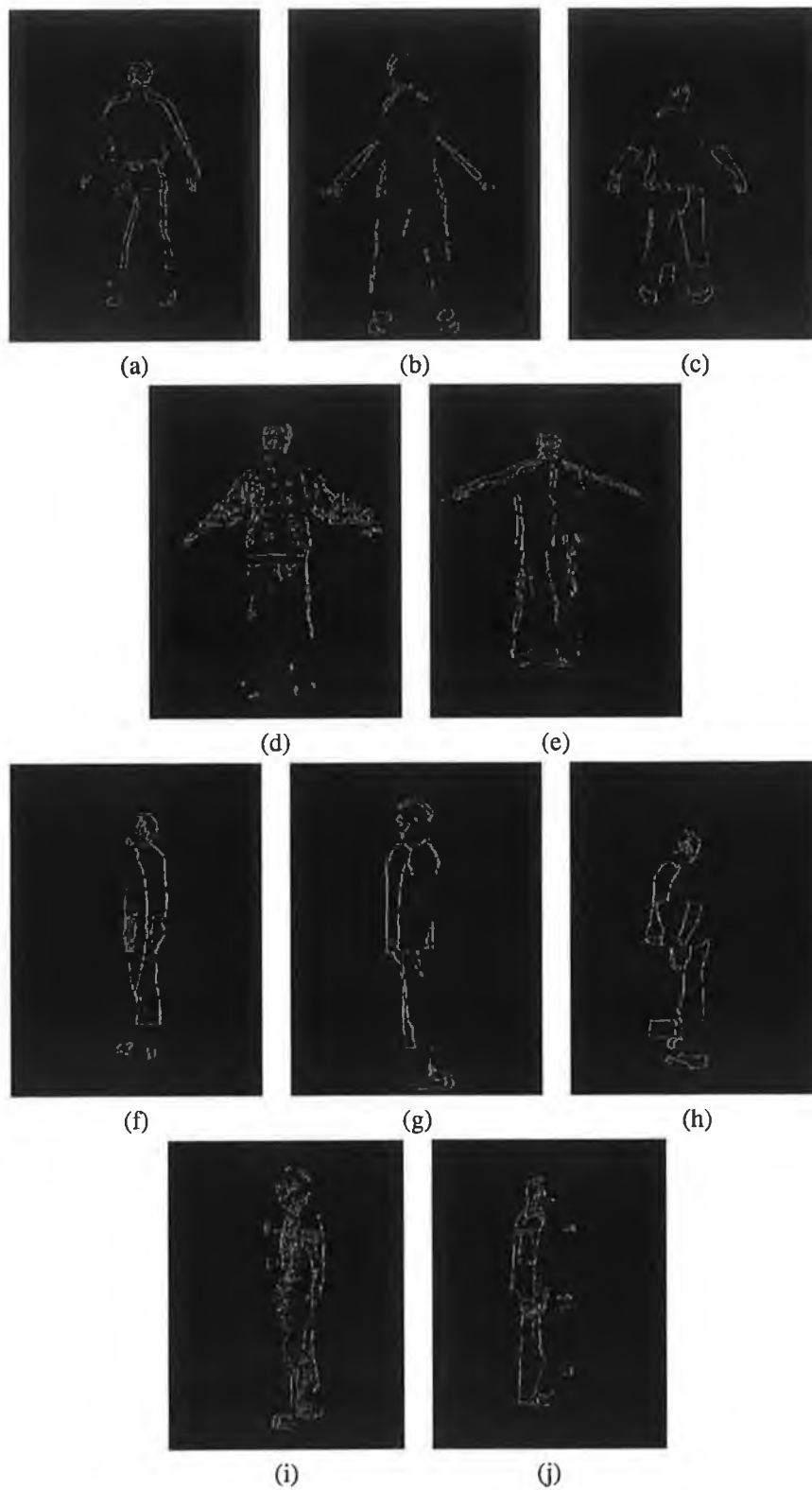


Figure 5.20: (a) to (e) show the results of applying the Canny edge detector to the difference map generated using the front and back images in Figure 5.2. (f) to (i) show the results for the side images.

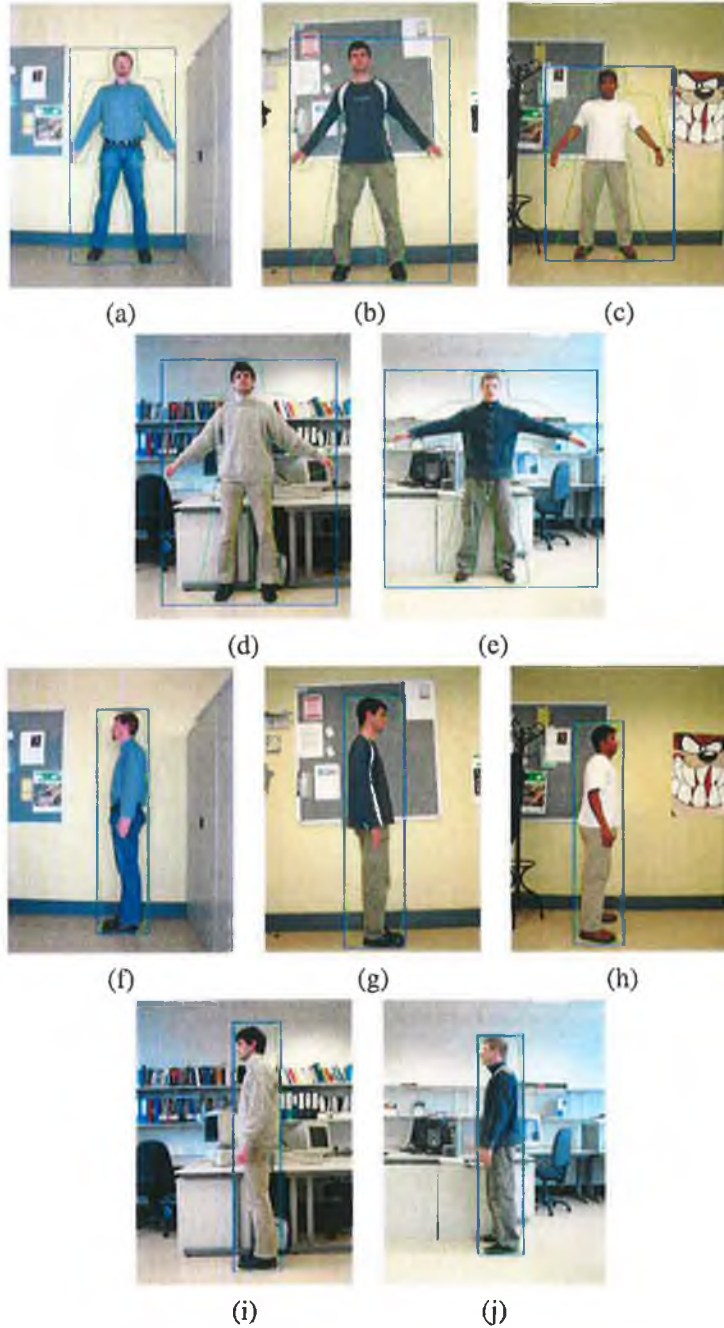


Figure 5.21: The bounding box and the correctly initialised templates. (a) to (e) show the bounding boxes generated for the front views in Figure 5.20. (f) to (i) show the results for the side images in 5.20.

The minimisation procedure is initiated in each case and the then different stages of evolution are shown after 20 iterations and 40 iterations and then the final initialisation of contours before the constraints are used to make sure that the final position matches as closely as possible the pre-defined constraints. The first phase of testing involves determining if the points are approximately linear. These points are selected using the key features that are extracted by tracing the final contour. The crotch point is chosen as the first point and then a control point with an index less than the index of the crotch is chosen.

The first example of the minimisation process is shown in Figure 5.22. In this Figure the initial position of the templates are shown in parts (a) to (d) on the images resulting from applying the Canny edge detector to images. In parts (e) to (h) the position of the contour at an intermediate stage shown, in parts (e) and (f) the contour is shown after 30 iterations and in (g) and (h) the contour is shown after 40 and 35 iterations respectively. In parts (i) to (l) the final position of the contour is shown for parts (i), (k) and (l) the final position is reached after 50 iterations and for part (j) the final position is reached after 65 iterations.

The second example using the template shows the situation when the level of clutter is increased. This has the effect of increasing the number of edges that are present in the background thus making it increasingly difficult for the active contour to converge to the correct solution. The evolution of the contour over 40 iterations for the image in Figure 5.2 (q) is shown in Figure 5.23.

In this situation when the edges defining the shape of the individual are not easily distinguishable from the edges in the background, an alternative approach to encourage the correct convergence was attempted using the difference map. This did not have a significant effect on the overall minimisation process to justify its general use. This is because although the individual occupies the centre of the image the left to right variation of the person can change substantially between the front and back or between the side views.

In this particular example and that of Figure 5.2 (m) to (p) required user assistance to move particular control points to reach the correct solution. This reduces the generality of the template extraction process. The automated initialisation is important as it reduces the amount of interaction that a home-user has to undertake. Although to reduce the effects of clutter the individual is advised to stand against a relatively clutter free background.

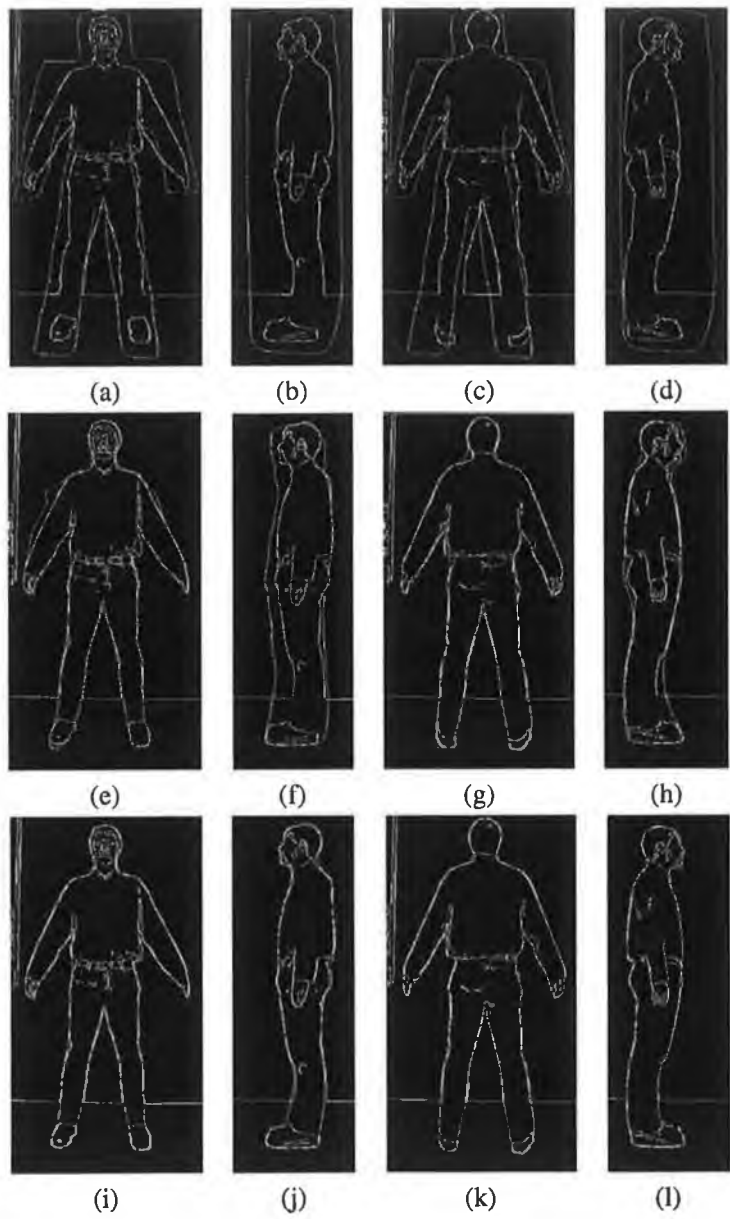


Figure 5.22: The initial position of the contour is shown in parts (a) to (d). The intermediate position of the contour is shown in images (e) to (h) and the final position of the contour is shown in parts (i) to (l).

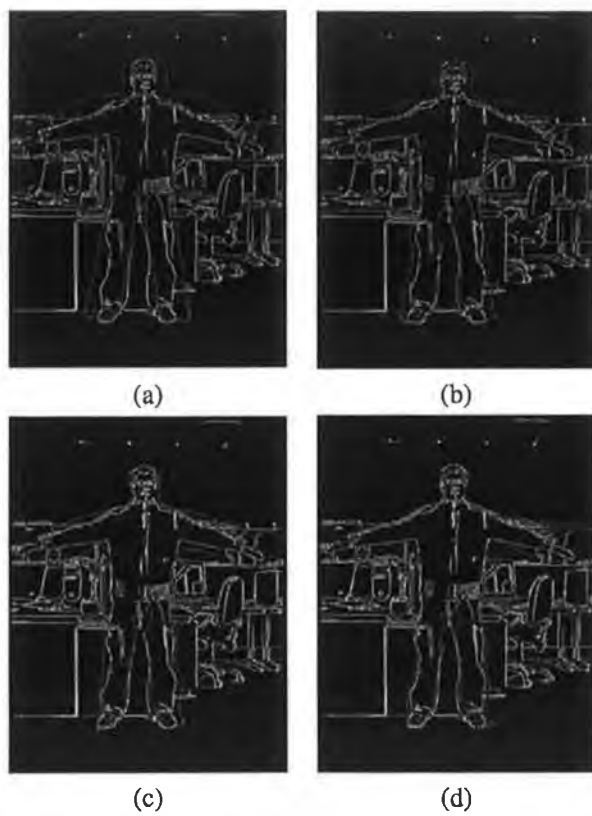
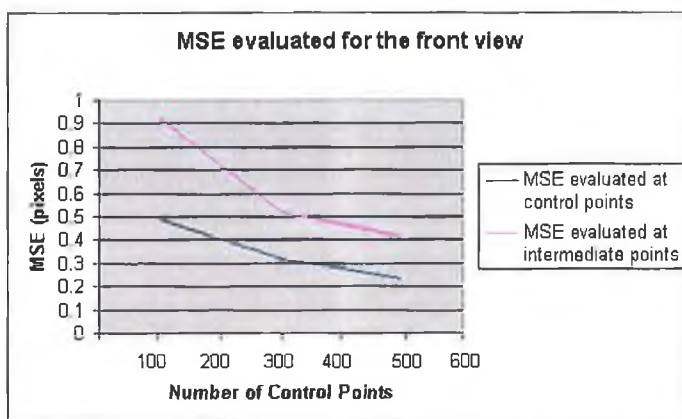


Figure 5.23: (a) shows the initial position of the contour on the edge map, (b) shows the position of the contour after 10 iterations, (c) shows the position of the contour after 20 iterations and (d) shows the position of the contour after 40 iterations.

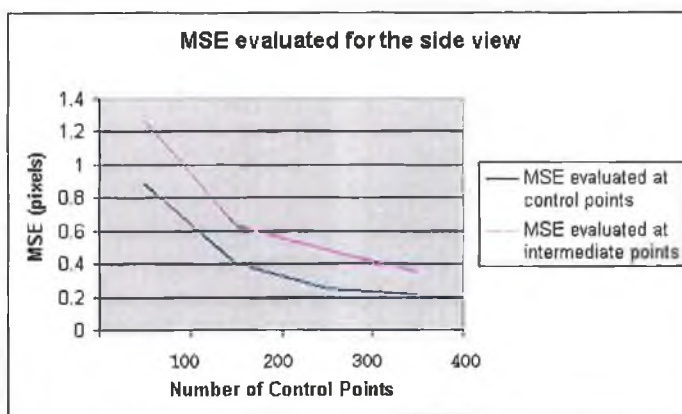
### 5.3.6 Assessment of the Template Fitting

This section describes analytically how accurately the final position of the templates describes the actual shape of the captured individual. To get a true measure of accuracy it is important that the measure is free of subjectivity. The subjectivity can be introduced in the manual generation of ground truth measures. In the case of manually segmenting the individual from the background a person carries out the segmentation and makes decisions regarding each pixel on the boundary, i.e. which pixel forms part of the background and which is part of the foreground object.

To reduce the subjectivity in determining this measure it is necessary to operate in a noise-free environment. To achieve this the underlying H-Anim model 4.2 is projected to an image plane using the scaled projection matrix in Equation 4.1, producing four silhouettes of the model in a noise free environment. These silhouettes are used as a ground truth from which the accuracy of the fitting procedure can be measured. An edge map for each silhouette is produced using the Canny edge detector (Canny 1986). The front and side templates are fitted to the image data. The



(a) Evaluation of the the contour fitting in the front view.



(b) Evaluation of the contour fitting in the side view.

Figure 5.24: Graphs illustrating the mean square error (MSE) calculated in the template fitting procedure.

front template is defined initially with 126 control points and the side template has 48 pixels. Each



of these templates are manually initialised. In each graph illustrated in Figure 5.24, the mean square error (MSE) decreases as the number of control points increases. The MSE calculated at each control point is lower than that calculated at the intermediate points between the control points. However, as the number of control points increases this difference decreases, this is in line with expectation. In the coarsest fitting of the front template using only the default number of control points, the MSE is below 1 pixel for errors calculated at the control points and at the intermediate points.

This test highlights that it is possible to accurately extract the silhouette of an individual using the constrained B-spline template. However, since the (photo-)realism of the final model is central to the application, the accuracy of the template fitting can be sacrificed and still achieve the same level of realism. However, in respect of the final application the realism of the human model is more important than the accuracy of the final model. Thus, it is possible to create a realistic human model even with a coarse fitting of the template.

## 5.4 Texturing of the Underlying Model

In approaches 1, 2 and 3 discussed in Chapter 4 the automatic texturing of the underlying model is achieved in two stages. Firstly, the individual's texture information that was extracted is mapped inside the silhouette of the underlying model. This information is then back projected, using normal vectors, to the underlying model. In this section, the process is initially tested on simple primitive objects; it is then applied to the underlying model. In particular, the texturing results highlight the importance of texturing the underlying model on a part-by-part basis and the important role that the normal vectors play in selecting which image should be used to texture a particular part of the objects when multiple views are available.

### 5.4.1 2D to 2D Texture Mapping

The 2D to 2D texture mapping is designed to permit any shaped texture map to be mapped to any shaped object while maintaining as accurately as possible the original information, including scale. On simple objects this can be achieved by generating a single mapping of all the information on to the shape but with more complex shapes and textures, particularly, when the objects have multiple boundaries or have holes within the complex hull<sup>10</sup>, the texturing is best achieved on a part by part basis.

An example of simple texture mapping is shown in Figure 5.25. In this figure, the information contained in the square is mapped to the circle. In Figure 5.25 (c) the information appears to be stretched across the horizontal dimension of circle. This occurs because the texture mapping at each vertical level is scaled using a constant vertical scale factor and a horizontal scale factor based on the relative widths of the circle and the square. When the vertical information is scaled appropriately as in Figure 5.25 (d) the texture map appears to take on the shape of the object being textured. The example of the mapping from a square to a circle was chosen as it illustrates the need to incorporate both the appropriate horizontal and vertical scaling. If on the other hand the texture

---

<sup>10</sup>The convex hull is the smallest convex region which contains the object such that any two points of the region can be connected by a straight line all points of which belong to the region.

mapping is carried out on objects that are similar in structure then scaling in a single dimension is sufficient.

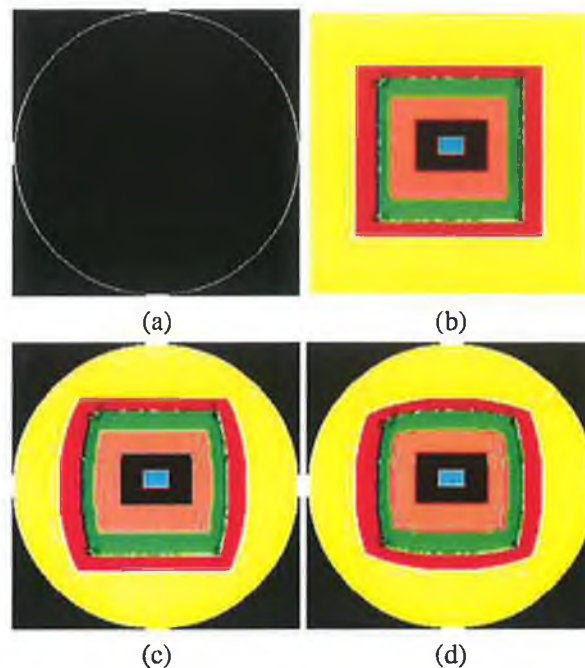


Figure 5.25: Example of simple 2D to 2D texturing. (a) shows the region to be textured with the texture map in (b). (c) shows the effects of texturing if the image is only scaled horizontally and (d) shows the correct fitting of the texture map.

It was not possible to apply this process to the body as a whole because of the pose that the individual adopts means that the convex hull of the individual contains gaps that do not form part of the shape of the individual. Thus as proposed in the approach of Hilton et al. (1999) the body was broken into different parts to ensure that the texture is mapped correctly. The key features that are automatically extracted are illustrated in Figure 4.11 are used to split the underlying model and the captured data into the different body parts.

The line from the armpit to the shoulder is used to separate the arms from the upper body; similarly the line joining the two shoulder points separates the head from the upper body. A line drawn horizontally through the crotch separates the legs from the upper body. This results in the body being split into six parts. This is illustrated in Figure 5.26 (a) and (b). In (Hilton et al. 1999) the body is split into seven parts where the upper body is separated into two parts separated at the armpits, see Figure 5.26 (d) and (e). In Figure 5.26 parts (b) and (e) the texture maps contain parts of the background and other parts of the body that are not related to the particular part. This is not significant since the silhouettes in parts (a) and (d) are the primary input data for the texturing method. In Figure 5.27 the same procedure of splitting the front view into six parts and the side view into four parts is shown for another individual.

In the left hand side views shown in Figure 5.26 (c) and (f) are split into four parts. Although the side view of the individual does not contain any holes if it is examined by plotting a vertical line through the silhouette at different positions, it can be seen that the line passes from inside

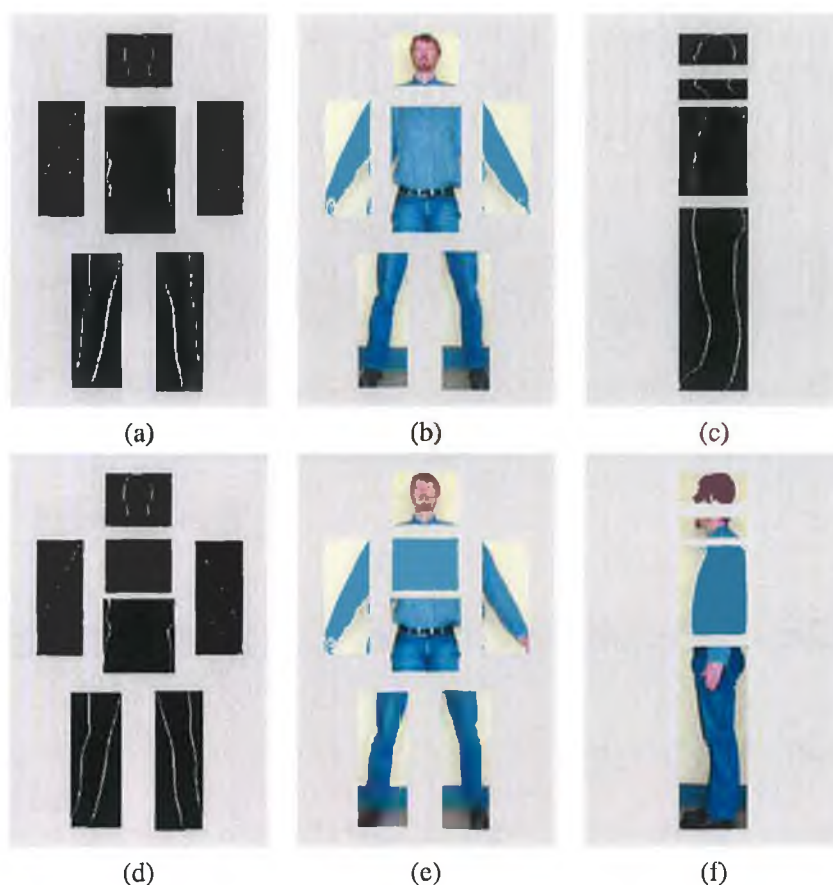


Figure 5.26: This Figure illustrates how the individual's silhouettes are separated into different parts for texturing. Parts (a) and (d) show the silhouette of the front of the individual split into six parts and seven parts. Parts (b) and (e) show the original image split into six and seven parts. Parts (c) and (f) show the side view of the individual split into four parts.

the silhouette through the background and into the silhouette again. This is particularly evident under the chin and at the small of the back depending on the individual's posture and clothing they are wearing. Thus the model is split into four parts to achieve the desired texturing. The side view is split into four parts by examining the contour. The width of each segment is the same and corresponds to the maximum and minimum horizontal coordinates that are extracted by tracing around the contour. The contour is further examined to find the centroid. Control points at the equivalent height on the left and right of the contour are automatically identified and the upper part of the contour is then examined to find the nose that is the extreme maximum or minimum above the centroid depending on the view. The throat is identified as the deepest valley between the nose and the centroid equivalent. It is also necessary to separate the legs from the upper body because depending on the build of an individual their stomach may cause a similar problem like when texturing under the chin because all points below the centroid inside the visual hull of the individual are not part of the individual. Thus the final division is at the small of the back or if that cannot be identified then centroid is used to separate the lower body. Additionally, it was considered to use information that was extracted in the front view to aid the segmentation of the

side view, for example and estimate of the height of the throat could be obtained from the height of the shoulder in the front or back view.

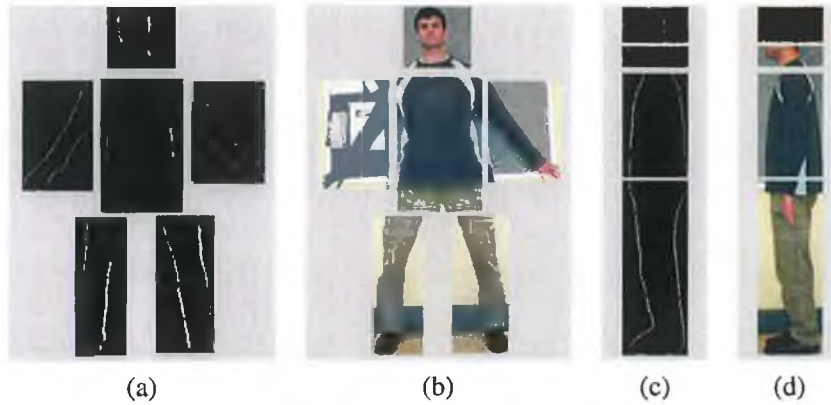


Figure 5.27: (a) and (c) contain the front and side silhouettes of the individual shown in Figure 5.2 (e) to (h), (b) and (d) show the equivalent silhouettes extracted using the template.

A separate mapping procedure is applied for the back view since in general the individual does not have the exact same silhouette from the front and behind. The same procedure for breaking the body into different parts can be applied to the back view of the individual and the results are shown in Figure 5.28.

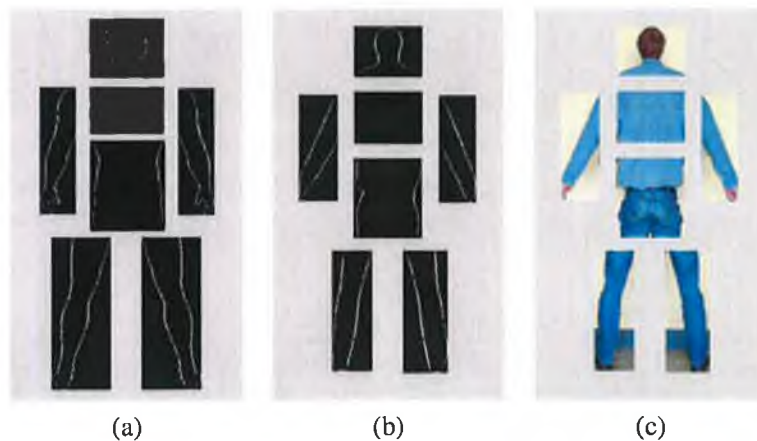


Figure 5.28: (a) contains the silhouette of model produced by projecting the model through the back camera centre, parts (b) and (c) show the equivalent silhouette and textured image.

The corresponding silhouette for the model is generated by projecting the 3D coordinates of the model to a plane  $3m$  from the model in the virtual environment. This process is repeated for each view corresponding to the captured data. This ensures that the scale of the model is proportional to the data captured. This can be seen in Figure 5.29 by examining the vertical and horizontal dimensions of the model's projection.

On close examination of the upper body it is not necessary to split it into two parts because in general the texture is similar across the two parts of the body and in some cases splitting the

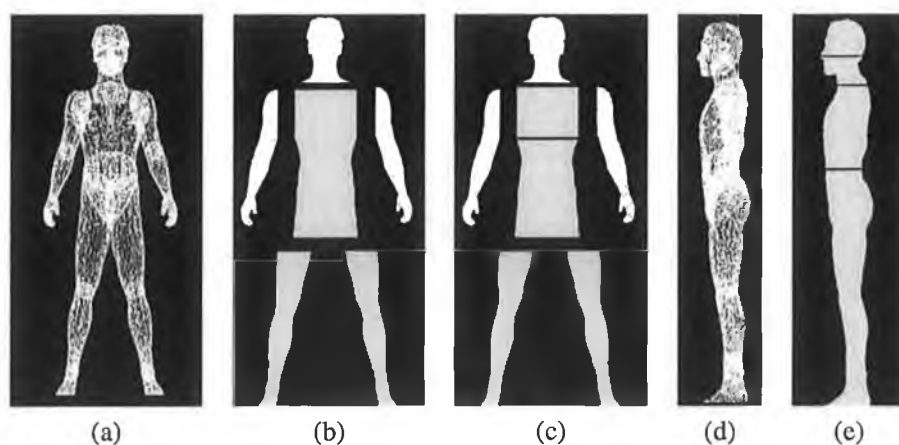


Figure 5.29: This Figure illustrates how the model's silhouette is separated into different parts for texturing. In part (b) the model's silhouette is split into seven parts and in part (c) the model's silhouette is split into six parts.

upper body into two parts can distort the design when the parts are recombined as they are scaled by different amounts.

Having separated the model and the captured data into different parts it is possible to implement the texture mapping procedure. The texturing method takes in three images; the first is the silhouette of the model which is split into the different body parts using the key features illustrated in Figure 4.11. The second image is generated using the B-spline contour of the individual that is separated into different parts using the key features and the third image is the captured image of the individual. The results of this texture mapping are shown in Figure 5.30. The information in each of the parts of the individual are mapped to the underlying model on a part-by-part basis using the corresponding key features on the model and the captured individual. This ensures that the texture map is continuous. It can be seen that the information captured in the images of the individual is accurately mapped inside the boundaries of the model. This ensures that all the data can be mapped to the underlying model in Section 5.4.2.

A new texture map is created by combining the sub-images in Figure 5.30 (a) and (b)<sup>11</sup>. The individual parts are combined using the key feature on the underlying model's silhouette. This results in the combined images in Figure 5.30 (c). More results are shown in Figure 5.32.

#### 5.4.2 2D to 3D Texturing of the Underlying Model

This section provides a detailed description of the testing that was undertaken to ensure that the underlying model is textured accurately using the information in the available views. The combined images that are generated in Section 5.4.1 are used to texture the model. As stated in Section 4.2.3 VRML texture nodes only accept a single image for the texturing of any primitive or complex object (VRML 1997). Thus it was first necessary that the four images that are generated are combined into a single image file. The texture coordinates must reflect this.

<sup>11</sup>In general, the images for each body part are not generated because the textured image data is mapped to its position inside the silhouette of the model

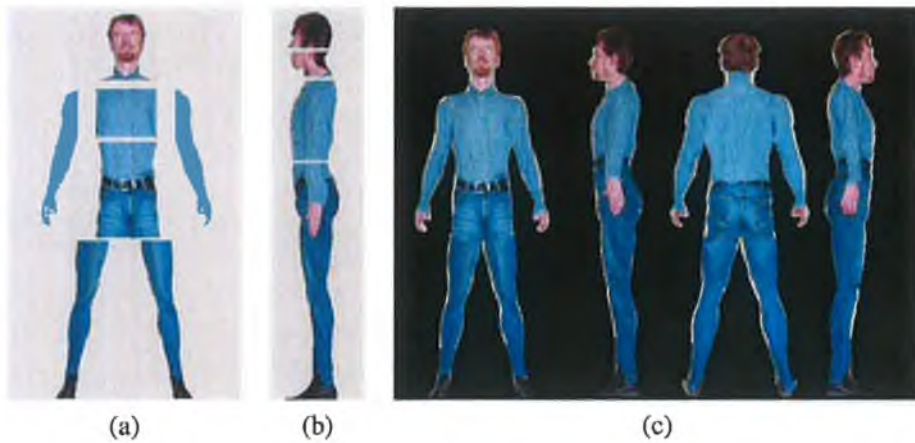


Figure 5.30: (a) and (b) show mapping of the captured data in Figure 5.26 to the silhouettes in Figure 5.29. (c) contains the result of the 2D to 2D texturing process for Figures 5.2 (a) to (d).

The texture coordinates range from 0 to 1 in the  $x$  direction and 0 to 1 in the  $y$  direction and the combined image has dimensions  $640 \times 1920$  pixels as the four images are placed side by side. It is possible to reduce the dimensions of this image by using the maximum and minimum values that are identified as key features in each view. This is not necessary and makes the generation of the texture coordinates more difficult because the projection of the 3D coordinates using the projection matrix in Section 4.2.1 projects the points to an image frame with dimensions  $640 \times 480$  pixels.

Before the final texture coordinates are generated it is necessary to establish which parts of the model should be textured with a particular part of the image. This is established using the normal vectors that project from each tri-face of the model and comparing them to the various centres of projection using the Equation 4.2 in Section 4.2.3. This technique enables the model to be textured using any number of views. To ensure that the maximum information was textured different criteria were used to make sure that the texture is not stretched by trying to texture too much of the model with a single image. This is of particular importance on the face and resulted in the following limits for the selection of the appropriate image to texture a tri-face. The results of this are shown in Figure 5.31 where the head of the model is textured using four different colours to represent the different images that are used to texture a particular tri-face.

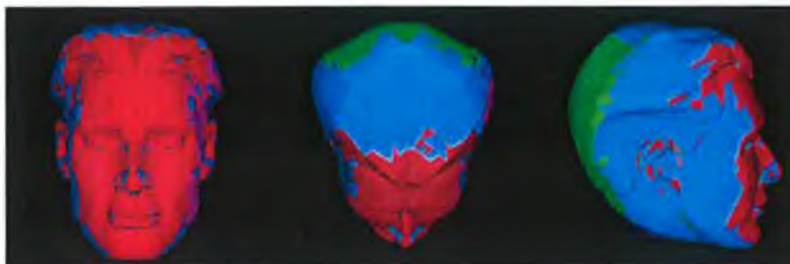


Figure 5.31: Example of how the head is textured. The parts of the head that are textured red are textured using the front view, the parts of the head that are textured green are textured using the back view and the blue and magenta represent the side views.

---

**Algorithm 5** Texture Selection Algorithm

---

```
if angle between camera centre[0] and normal from a face is less than 60° then
    Texture with front view
    Generate appropriate texture coordinates
else if angle between camera centre[1] and normal from a face is less than 60° then
    Texture with back view
    Generate appropriate texture coordinates
else if angle between camera centre[2] and normal from a face is less than 30° then
    Texture with left view
    Generate appropriate texture coordinates
else
    Texture with right view
    Generate appropriate texture coordinates
end if
```

---

Once the appropriate camera has been selected it is then necessary to generate the appropriate texture coordinates. In this situation with four views the images are arranged from left to right: front, left, back and right view. Thus if a tri-face is to be textured with an image from the front view then the coordinates are generated by taking the vertices of the face that are projected to the image frame in the creation of the model's silhouette. This provides coordinates in the range of 0 to 480 pixels in the horizontal direction and 0 to 640 pixels in the vertical direction. The horizontal values are divided by 1920 to convert the coordinates to a values in the range 0 to 1 and the vertical values are divided by 640 to convert them to a value between 0 and 1. If the right image is used the projected horizontal values are shifted by 1440 pixels before they are divided by 1920 to ensure that the correct part of the combined image is used to texture the model. The textured 3D models are shown in Figure 5.34 to 5.38.

### 5.4.3 Enhancement of the model using facial features

In this section the results are presented for the improvement of the texturing process using the extracted facial features. The features are first identified and then the scaling of the images is achieved relative to the distance between the eyes (for horizontal scaling) and the distance between the eyes and the mouth (for vertical scaling). The images are then positioned using the location of the eyes and the mouth in the front view. In figures 5.34 to 5.38 the results are presented. The coordinates of the key facial features in the front images are listed in the Table 5.4. The image dimensions are based on the size of the head images extracted using the feature extraction process.

The results that are presented to allow a comparison between the enhanced model and the model that is simple textured. In each case four views of the models are presented. The situations are chosen to represent possible views of the model within a virtual environment. The idea behind this method of presentation is to enable a direct examination of the realism of the model at two depths i.e. when the model close to the viewing location and when the model is far from the viewpoint. In addition, the models are examined with two additional views in which the models are not parallel to the viewpoint.

In Figures 5.37 and 5.38 the quality of the texturing of the face is reduced due to the effects of natural light that illuminates one side of the face more than the other. This is particularly prevalent

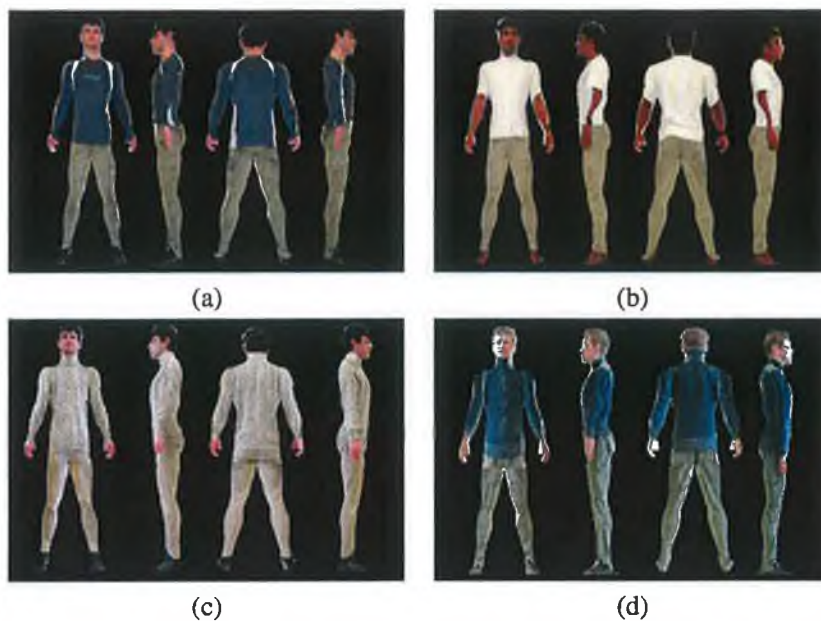


Figure 5.32: the results of the 2D to 2D texturing process for the images in Figure 5.2, part (a) shows the results of the texturing with parts (e) to (h), part (b) shows the results of the texturing with parts (i) to (l), part (c) shows the results of the texturing with parts (m) to (p) and part (d) shows the results of the texturing with parts (q) to (t).

Images	left eye (x,y)	right eye (x,y)	mouth (x,y)	Image Dimension
set 1	(37, 24)	(54, 24)	(46, 40)	95 × 77 pixels
set 2	(33, 35)	(51, 33)	(44, 52)	105 × 100 pixels
set 3	(39, 29)	(55, 28)	(47, 44)	90 × 76 pixels
set 4	(39, 32)	(58, 31)	(49, 48)	109 × 98 pixels
set 5	(60, 33)	(78, 33)	(96, 51)	142 × 93 pixels

Table 5.4: The location of the key facial features in the images shown in Figure 5.2. The image dimension refers to the size of the head image that results from applying the body separation algorithm.

in 5.38 where one side of the face appears to be paler than the other. In the other Figures a stronger resemblance between the enhanced models and the captured data at both distances can be seen. However this is a very subjective measure. Figure 5.37 was split into a left and right image for analysis, the right image being closest to the source of natural light. The average pixel intensity and the standard deviation for each part of the image are determined. For the right-hand side image the average pixel intensity is 127 and the standard deviation is 51.90 and for the left-hand side image the average pixel intensity is 119 and the standard deviation is 56.87. In Figure 5.38 in the left hand image, which is closest to the natural light source, the average pixel intensity is 167 and the standard deviation is 56.71 and in the right hand image the average pixel intensity is 160 and the standard deviation is 52.22. If the images are examined in HSI<sup>12</sup> colour, it is possible to

<sup>12</sup>HSI: Hue Saturation and Intensity. Hue refers to the perceived colour (the dominant wavelength), saturation refers to the dilution by white light.



see the contrast in the hue between different parts of the image, see Figure 5.33. In Figure 5.33 (b) the area circled, the hue shows a high local variation on the face.

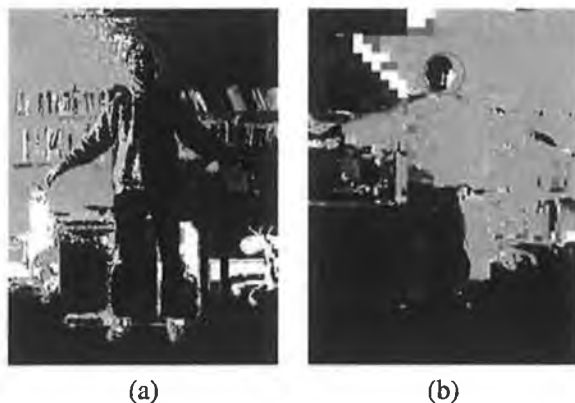


Figure 5.33: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

In each situation the boundary of the 2D textures contains parts of the background. This accounts for the fact that at the edge of the model that the texture appears to be disjoint. In the texturing algorithm proposed for the texturing of the model as described in Section 5.4.2 the angles used to determine which image is used aims to minimise these effects by trying to select parts of the image that are not at the boundary of the 2D texture.

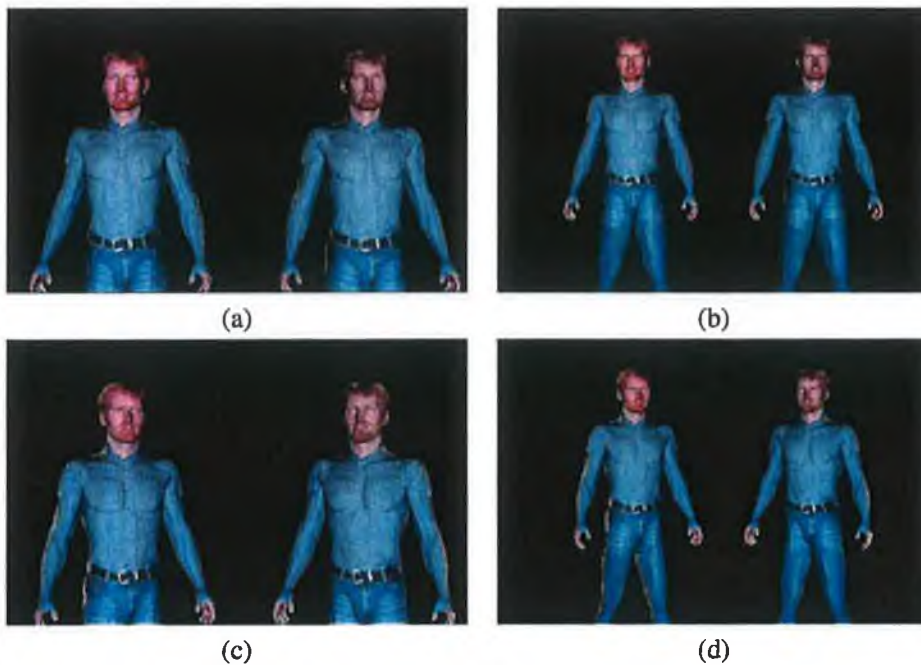


Figure 5.34: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

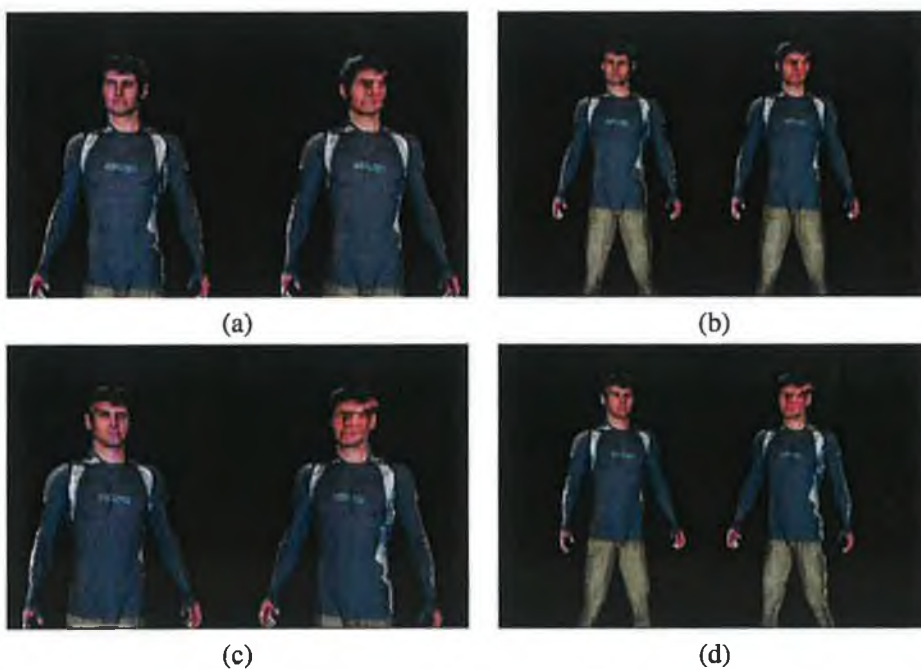


Figure 5.35: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

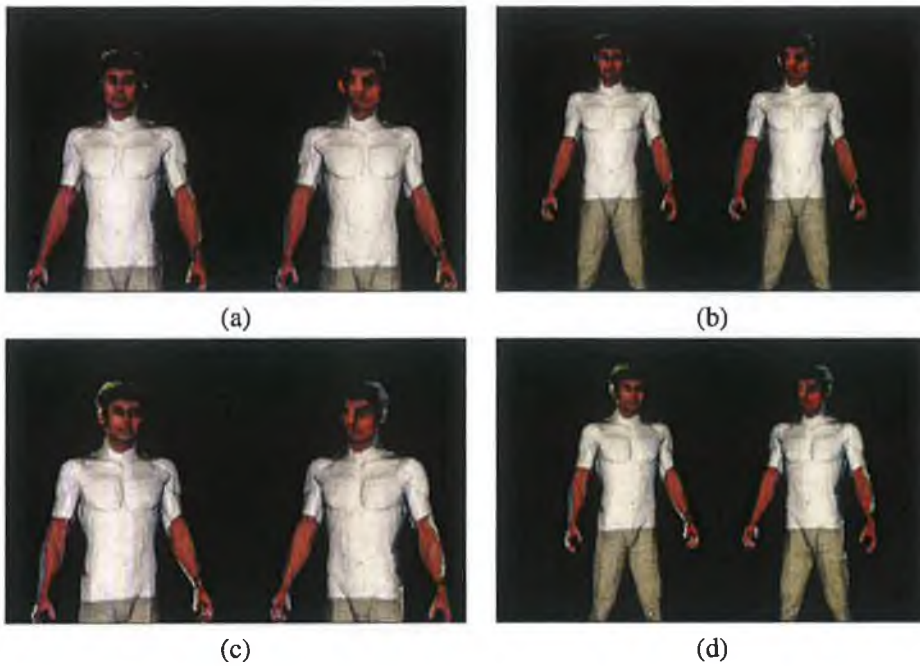


Figure 5.36: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

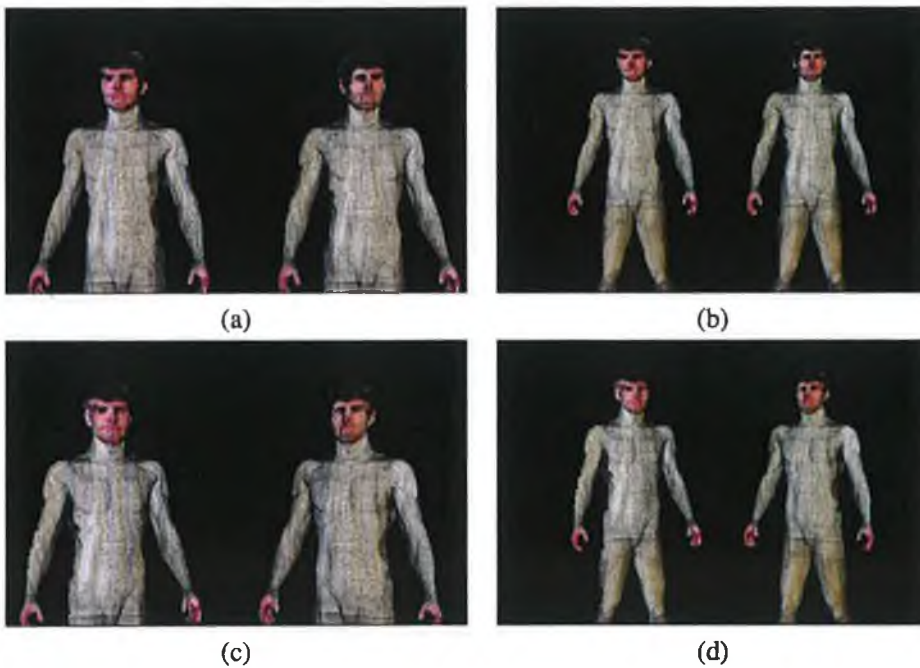


Figure 5.37: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

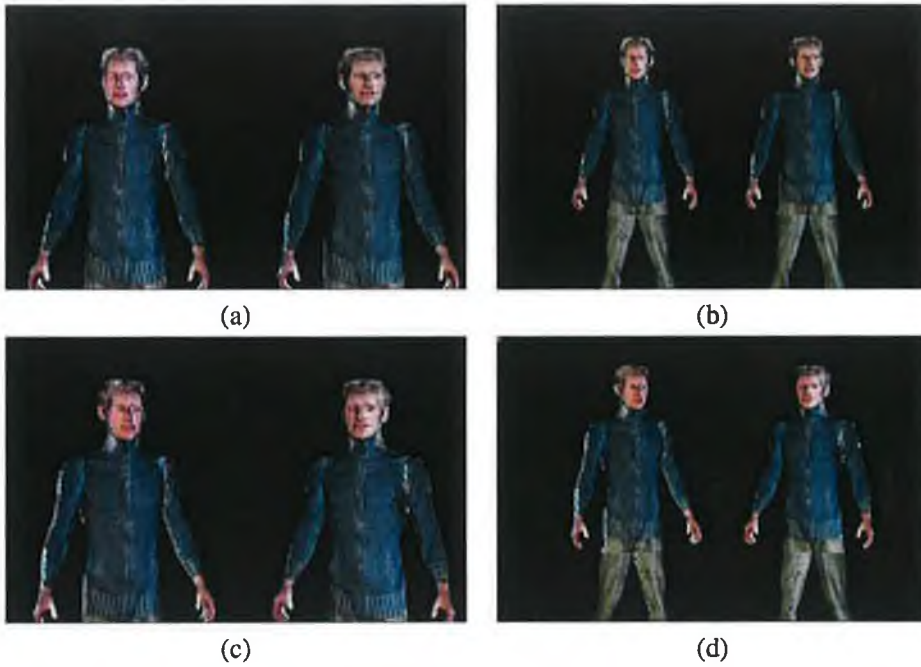


Figure 5.38: Four views of the model. In each case the model on the left is the model that is textured using the facial features. (a) shows the models close to the viewpoint and (b) shows the models far from the viewpoint. (c) and (d) show the models at an angle to the viewpoint and at two distances from the viewpoint.

## 5.5 Generation of the Bounding Volume

The generation of the bounding volume is an important element in the personalisation of the final model. In the results displayed in Figure 5.39 to 5.43 the bounding volumes incorporate specific shape information associated with the captured individual. In each case two views of the bounding volume are provided: one from the front and one from a location between the front and the back. Three views of each textured volume are shown as well. In both the front and the side the silhouette of the individual is well defined but in the third view the particular features of the individual are not easily distinguished.

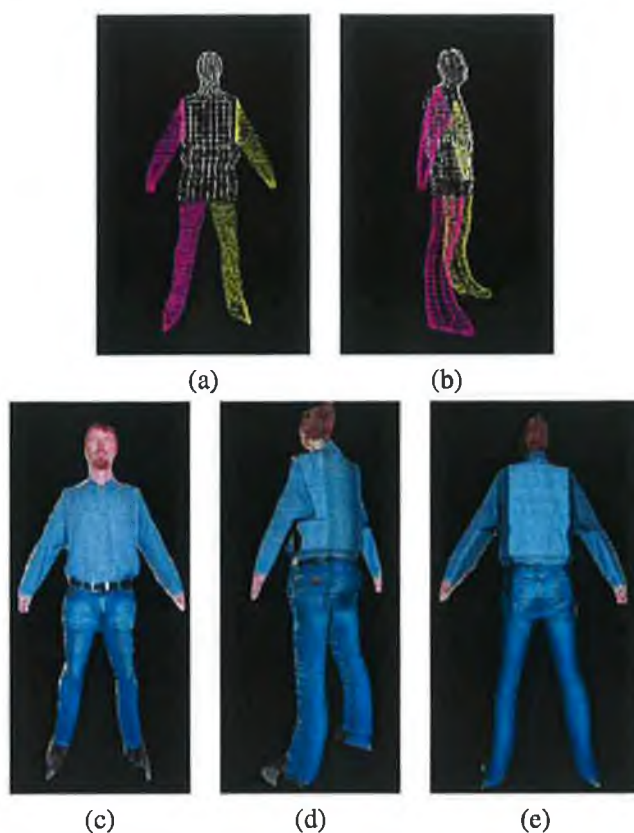


Figure 5.39: Bounding volumes for the images (a) to (d) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.

When generating the bounding volume the most important element is to automatically provide a model that incorporates the shape information that is essential for the generation of personalised models. The models are created using the front and one of the side views of the model. This created the maximal visual hull of the individual and thus the fine details of the individual cannot be reconstructed. The models that are shown in Figures 5.39 to 5.43 are created using simple ellipses<sup>13</sup> that interpolate the silhouettes at different vertical levels. This limits the personalisation that is possible, since the individuals are not perfectly elliptical. In addition, if more extensive

<sup>13</sup>The ellipses are generated by a B-spline that interpolates the control points at a vertical level.

modelling of the individual was considered then it reduces the flexibility of the approach and would require a certain amount of interaction from the home user to identify particular features that should be enhanced. The back of the models appear to be more rectangular instead of elliptical to match the general shape of an individual's upper body. This is achieved by introducing additional control points and reducing the continuity of the B-spline curve by having multiple knots.

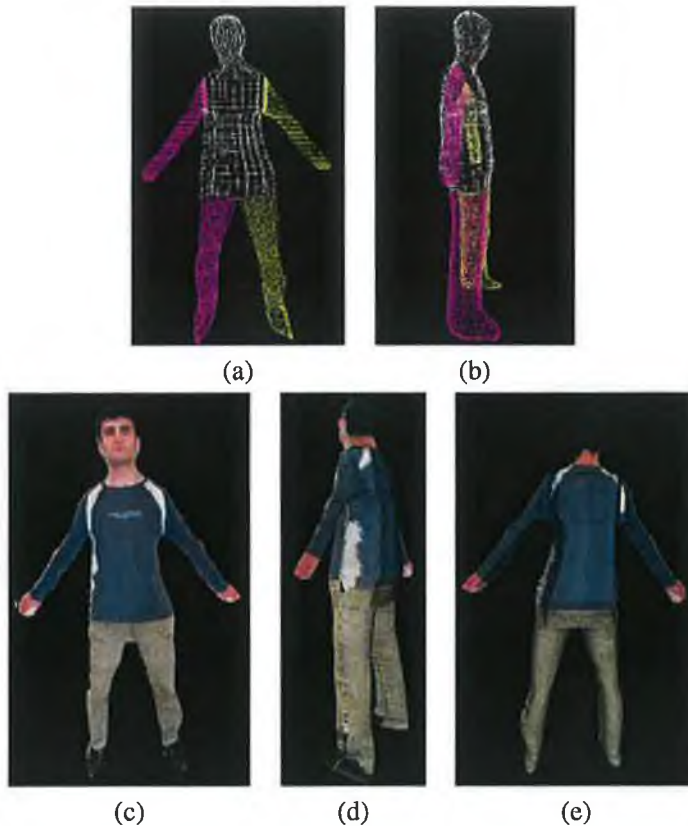


Figure 5.40: Bounding volumes for the images (e) to (h) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.

The texturing of the bounding volume is a first attempt to use the bounding volume for representing the individual in low-cost applications. The texturing of the model is carried out using the front and back views. The front of the model is directly textured without scaling by projecting the front view on to the bounding volume. The back of the bounding volume is textured by mapping the information inside the back silhouette on to the front silhouette in a similar manner to that described in Section 5.4.1 for the mapping of the captured data into the model's silhouette. This introduces a small amount of scaling of the image data. This is necessary because in general, the size of the front and back silhouette are not the same.

In each case the number of vertices that are used to create the model are presented in Table 5.5. The maximum value is 2956 vertices for the creation of the bounding volume in Figure 5.40. This is in contrast to 11103 vertices in the Hiro H-Anim model (H-Anim 1997).

In each case the number of polygons that are used to generate the model are listed in Table

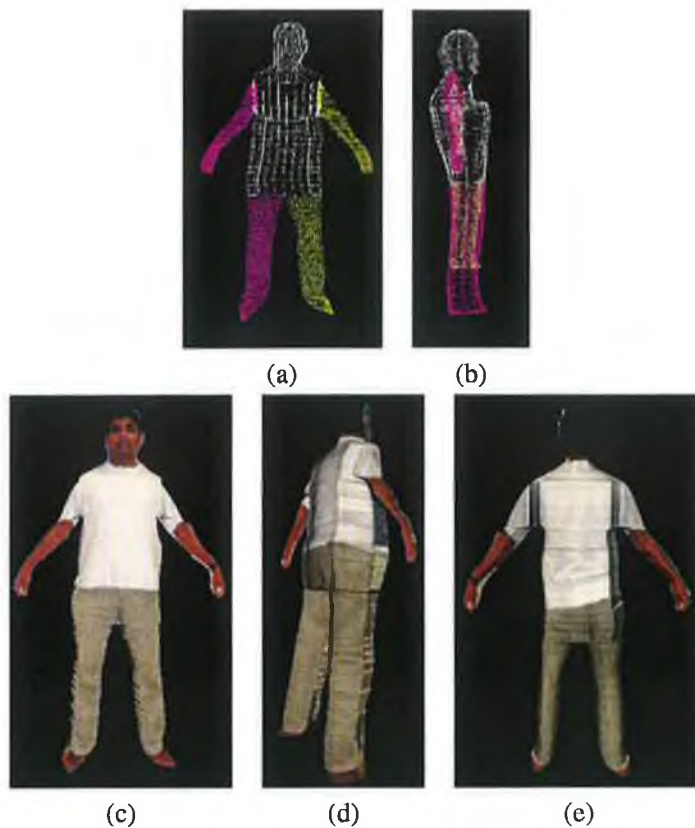


Figure 5.41: Bounding volumes for the images (i) to (l) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.

5.6. The maximum value is 5732 polygons for the creation of the bounding volume in Figure 5.40. This is in contrast to 21422 in the Hiro H-Anim model (H-Anim 1997).

When the bounding volumes are textured their shape shows a greater resemblance to that of the captured individuals than the models created by texturing an underlying model. The texturing of the bounding volume has the advantage that the front image is not scaled and can be directly textured to the surface. This improves the realism of the individual model when viewed from the front. The significant limitation of this method is that the joint positions cannot be reliably located and thus existing animation data cannot be applied to this model. This reduces the generality of this model in virtual environments.

Images	Head	Upper body	Left leg	Right leg	Left arm	Right arm	Total
Figure 5.39	419	524	541	473	371	388	2716
Figure 5.40	524	608	558	541	337	388	2956
Figure 5.41	398	524	490	473	337	337	2559
Figure 5.42	482	545	609	541	371	315	2863
Figure 5.43	503	524	534	473	337	371	2742

Table 5.5: Details of the number of vertices that are used to create the bounding volumes.

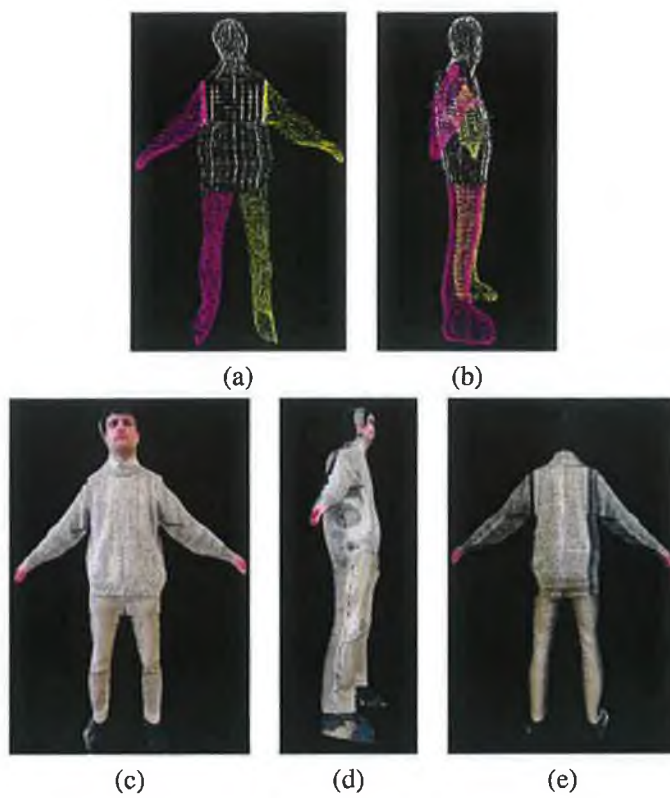


Figure 5.42: Bounding volumes for the images (m) to (p) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.

Using the skeletal information extracted in 2D from the knowledge of the key features in Figure 4.11 it is possible to position a skeleton within the bounding volume. This skeleton provides possible locations for the bones but does not facilitate the extraction of joints, such as elbows or knees. This lack of specific joint information makes it difficult to animate the bounding volume making it difficult to use it in real world applications since an estimate of some joints is not sufficient to give the model a realistic movement. Although, in comparison to impostors that are used in certain mobile application to provide low-cost models of distant objects the animation of the personalised models is not considered (Boyle et al. 2004). The lack of 3D information associated with the impostors means that their range is limited. However, the use of a bounding volume has a relatively low number of polygons that can provide a 3D model with low overhead

Images	Head	Upper body	Left leg	Right leg	Left arm	Right arm	Total
Figure 5.39	796	1006	1048	908	708	742	5207
Figure 5.40	1006	1216	1082	1046	640	742	5732
Figure 5.41	754	1006	946	910	640	640	4896
Figure 5.42	964	1048	1182	1046	708	606	5554
Figure 5.43	964	1006	1048	916	640	708	5282

Table 5.6: The number of polygons that are used to create the bounding volumes.



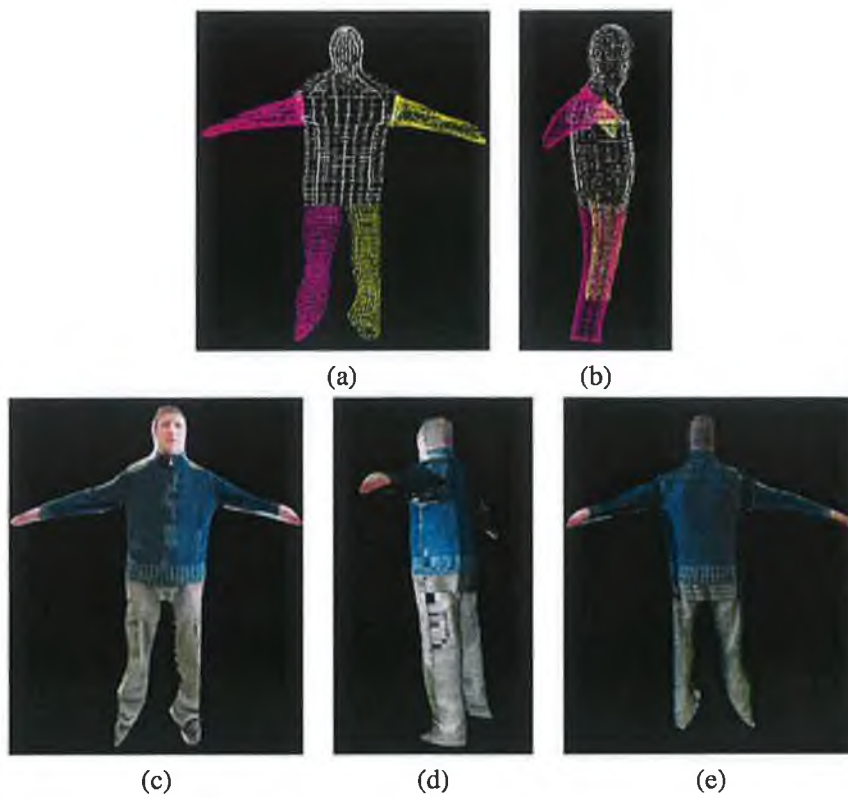


Figure 5.43: Textured bounding volumes for the images (q) to (t) in Figure 5.2. (a) and (b) show the generated mesh structure for the individual and (c), (d) and (e) show three views of the textured model.

than can be used to represent the individual in certain applications and provide more personalised models for 3D applications.

Trying to ensure that the front image is well matched to the face make it difficult to incorporate the side views. Thus at close distances it is necessary that the orientation of the model is kept constant with respect to a view point.

## 5.6 Testing of the 3D Active-Mesh Implementation

This section illustrates the procedures that were undertaken to verify the active-mesh implementation and to illustrate how active-meshes can be applied to the modelling of 3D shapes and in particular, how this can be extended to model 3D virtual humans. The testing starts with a single object being deformed to approximate another. A number of different scenarios are presented to see how the objects deform. This is then extended to permit multiple objects to be deformed by a single object or a group of objects. This is important extension for the modelling of human models since the internal (underlying) model is specified as the constitute body parts which are combined to form the human model.

### 5.6.1 Testing the Internal Constraints in 3D

In active-meshes the internal constraints are important in determining how the mesh deforms. If the internal constraints are strong then the mesh will attempt to retain its original shape at each iteration, but if the internal constraints are weak the mesh is free to deform under the influence of the external constraints. The internal forces can be applied uniformly across a particular mesh or at individual points depending on how the mesh is to be deformed. The following tests are designed to show how the model deforms under the influence of the internal constraints. In the first set of tests a point on the surface is pulled a significant distance from the surface in the direction of its normal. This has the effect of changing the length of the mesh-lines connected to this point, see Equation 4.4. In each case, only the internal constraints determine how the point is pulled back to the surface of the sphere.

In Figure 5.44 the original position of the sphere is shown with a point pulled from its surface. This point is pulled a significant distance from the surface. This results in a large internal force at the point of intersection and smaller internal forces acting on the connecting points. This permits the sphere to deform and at these points only because all other mesh-lines have the same length and thus the internal constraint at the other mesh elements is equal to zero, thus their position remains unchanged. In this situation, the internal constraint  $\alpha_L = 0.01$  and  $\alpha_i = 0.01$ .

At each iteration, the subsequent point is pulled to the same point location and thus the internal constraint still affects the same point and the connected mesh points. In Figure 5.44 (d) the position is shown after 40 iterations and the point of intersection has moved and the set length of the lines joining the mesh points connected to this point have changed. The points connected to this have also been pulled in the same direction. If the surface is examined closely, then it is observed that the remainder of the mesh elements do not move. The reason for this is that at the end of each iteration the set position and the current position of the mesh points are set to be the equilibrium position at each iteration. This results in the set and current positions being the same at the start of the next iteration and thus the range of the internal constraints is limited.

In the second situation, shown in Figure 5.45, the same force is applied to the sphere and at the point of intersection the point is pulled from the surface. Observing the shape of the sphere after the first iteration the point of intersection and the connected points have been pulled from the surface. After the subsequent iterations, the effects of the initial deformation at a single point are seen rippling across the surface of the sphere. After a certain number of iterations, the effects

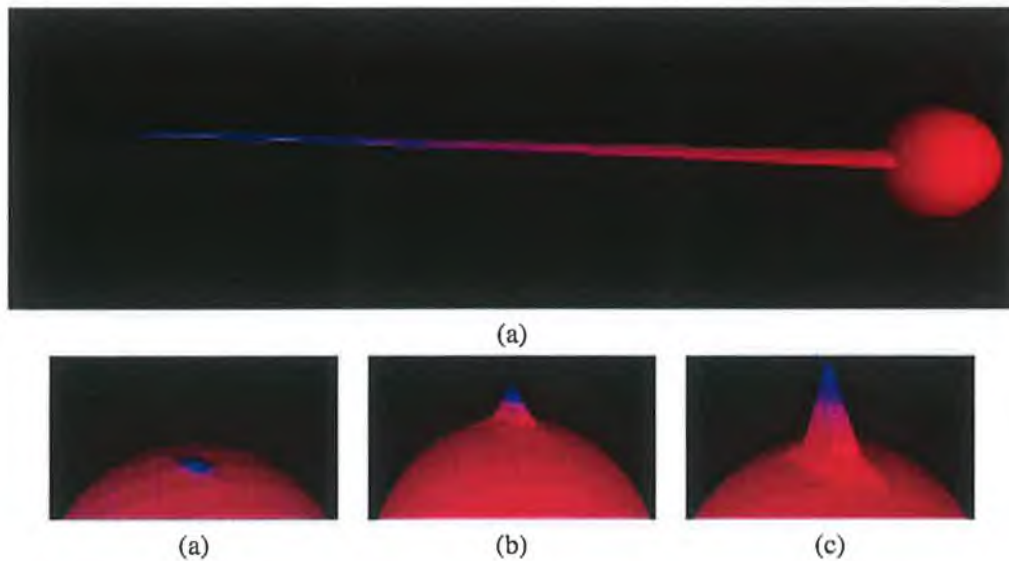


Figure 5.44: Application of internal constraints at a single point on the spheres surface. (a) shows the point of intersection on the surface and the effect this has on the current length of the mesh lines connected to the point. (b) shows the position after three iterations. (c) and (d) show the position after 20 iterations and 40 iterations respectively.

of this are reduced and the points on the surface that are further from the point of intersection are less severely effected. It can be also observed that the effects do not influence the whole surface of the sphere for small  $\alpha_i$  values. In Figure 5.45, (a) the initial sphere, and (b), (c) and (d) the  $\alpha_i$  and the  $\alpha_L$  values are set at 0.01 the effects of the internal forces are not observed far from the point of intersection. In Figure 5.45 (e), (f) and (g) the  $\alpha_i$  and the  $\alpha_L$  values are set at 0.3 and the effects of the internal constraints have a ripple effect along the surface of the sphere and in (g) the surface of the sphere has returned to a nearly smooth surface but it has been pulled in the direction of vertex that was moved. Smaller values for  $\alpha_i$  and the  $\alpha_L$  are not shown because the effects are not observed on the surface unless examined very closely or over a large number of iterations.

### 5.6.2 Testing with Primitive Shapes

The first set of tests show how the active-mesh approach to modelling can be used to mould one object to take on the characteristics of another. In particular, the case shown is a sphere being moulded into a cube to illustrate the complexities involved. The sphere is characterised as having no edges or corners and all points in the surface of the sphere are equidistant from the centre of the sphere, while the cube has twelve edges and eight corners and the points on the surface are at varying distances from the centre of the cube. Thus it is difficult to find similarities between the two shapes. In the Figures shown in this section, different views of the evolving scene are taken to firstly illustrate the 3D nature of the scenes and secondly to show particular attributes of the active mesh implementation.

One of the initial tests is illustrated in Figure 5.46. In this situation the sphere and the cube are centred at the origin, thus the sphere is at the centre of the cube. The external forces acting

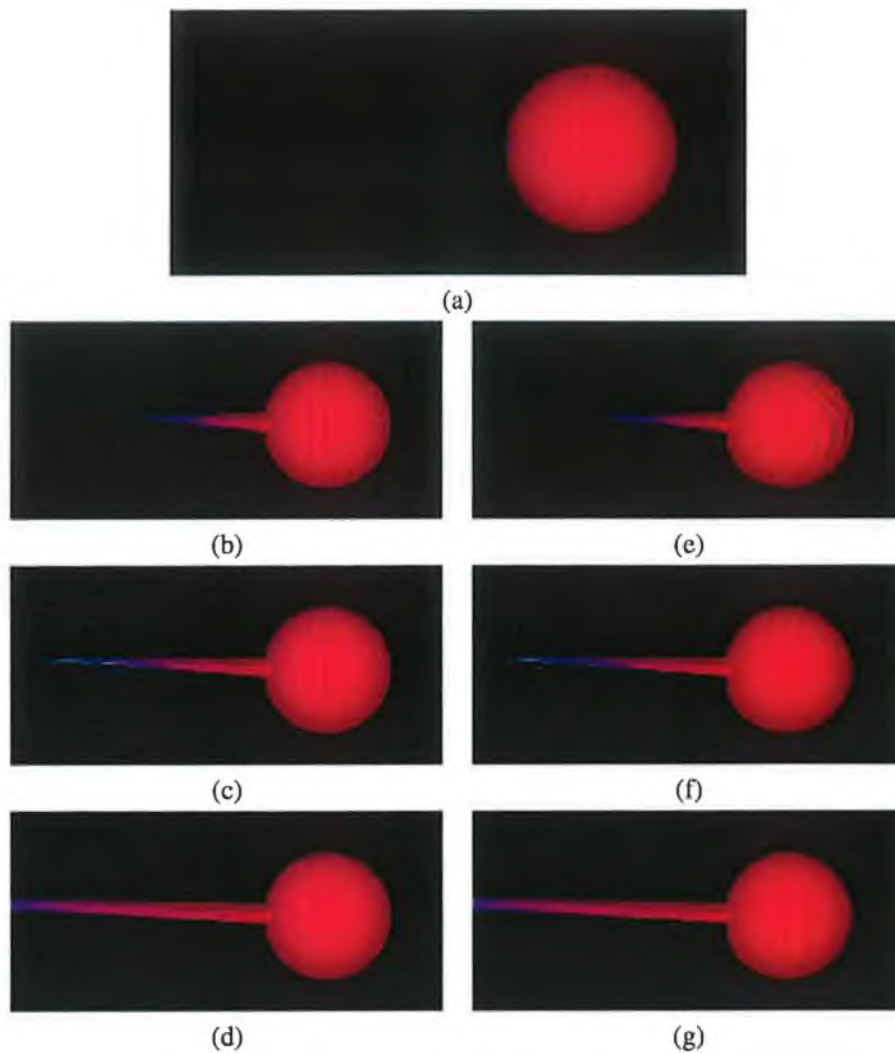


Figure 5.45: The application of internal constraints on a sphere, (a) shows the initial sphere (b), (c) and (d) show the effects of strong internal constraints on the sphere, and (f), (g) and (h) show the effects of weaker internal forces on the structure of the sphere.

on a vertex of the sphere are generated using the normal vector<sup>14</sup> from a vertex on the sphere to its point of intersection on the surface of the cube. The point of intersection is calculated using Equation A.33 in Appendix A. In initial tests the external energy was calculated using the absolute distance from a point on the surface of the sphere to where it intersects the cube. This resulted in the situation that the external energy progressively got smaller as the sphere was deformed to approximate the cube. Thus to provide a consistent external force the decision was taken to find the centroid of the object and then the distance from the centroid of the sphere to the point of intersection on the cube is used to provide the external force.

The calculation of the internal energy resulted in a significant modification of the concept of internal energy as defined in (Molloy & Whelan 2000) because the internal energy provided

<sup>14</sup>The normal for each face is determined such that it projects away from the surface of the object and the normal per vertex is calculated using formula A.32 in Appendix A

a method of incorporating the elastic and rigid properties of mesh-lines<sup>15</sup>. The elastic properties provide the mesh with its flexibility to deform and the rigid constraints gives the mesh its structure. In addition, the rigid properties of the mesh-lines cause the lines to attempt to return to their reference length. In 3D, the elasticity of the mesh is essential to allow one shape to deform to approximate another and the rigidity is important to ensure that the structure is maintained and to preserve the characteristics of the shape that is being deformed. In contrast to the initial concept of the external energy, as the sphere is modelled the distance between the vertices on the sphere will increase at each iteration as they are pulled towards the cube. Thus at each successive iteration, the calculation of the internal energy will have a greater effect in the energy Equation and will reduce the significance of the external energy and ultimately stopping the minimisation process. To provide a constant internal force, the concept of relative internal energy is introduced, i.e. the significance of the internal energy is relative to the average distance between the vertices of the sphere.

The effects of the internal and external energies are fully investigated in this section using the following tests where the sphere is actively deformed to approximate the cube:

- The sphere and the cube centred at the origin with the sphere completely enclosed by the cube,
- The sphere completely enclosed in the cube but offset from the origin.
- The sphere offset from the origin but not completely enclosed within the cube.
- The sphere with different vertices having different rigidity
- The sphere completely enclosing the cube.

In Figure 5.46 the sphere and the cube are centred at the origin and the cube completely encompasses the sphere. The rigidity of each vertex of the sphere is the same and is set to be 255 (see Chapter 4) permitting significant deformation at each iteration. In addition, the value of  $\alpha_i$  is 0.1. This corresponds to the step size at each iteration. It should be noted that in each Figure shown in this section that the shape being deformed has the faces of each polygon shaded and the shape that it is being approximated is represented as a mesh. In Figure 5.46 (a) the initial position of the sphere and the cube are illustrated then the position after 2, 10 and 46 iterations are shown. It is observed in Figure 5.46 (b) that the effects of the deformation process are not uniform. The reason for this is that the normal vector calculated at a vertex on the surface of the sphere will intersect the cube at different angles resulting in different calculations for the external energy resulting in a non-uniform deformation of the sphere. After 10 iterations the sphere has started to take on the properties of a cube although no clearly defined corners exist and its size does not closely approximate the cube. The final position of the deformed sphere is shown in Figure 5.46 (d). The process is halted when the distance between the vertices on the surface of the sphere and cube is reduced to zero or approximately zero.

Analysing the results it is seen that the corners and the edges of the cube are not well approximated. The reason for this can be found in general nature of the modelling tool, i.e. that in certain

<sup>15</sup>A mesh-line is defined as the line that connects two vertices in the mesh

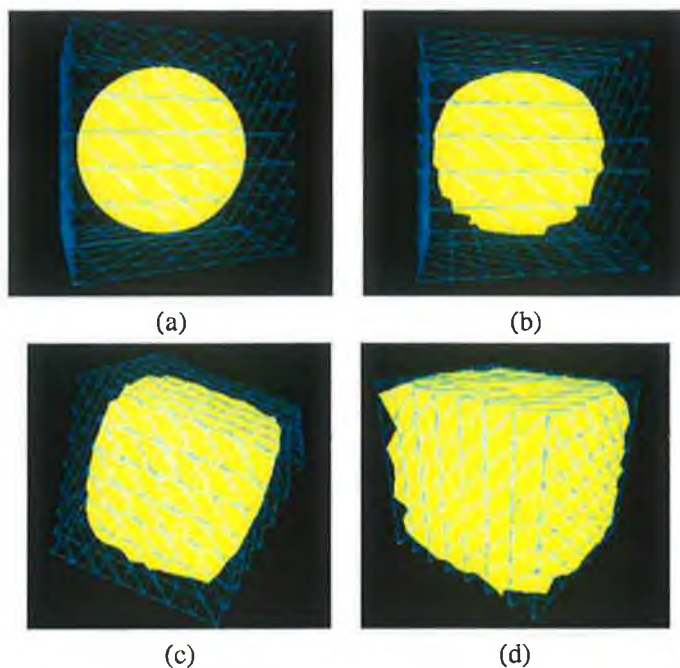


Figure 5.46: The results for the first active-mesh trial involving a sphere placed inside and at the centre of a cube. (a) shows the initial shapes, (b) shows the evolution after 2 iterations, (c) shows the iterations after 10 iterations and (d) shows the final shape of the sphere after 46 iterations.

instances the shapes that is being deformed will not have any edges and if the nearest vertex to the point of intersection on the cube was chosen this would result in multiple vertices being pulled to the same point. In the results generated, the vertex of the object being deformed is pulled to its point of intersection on the surface of the bounding object. This was deemed appropriate, because for example, in the situation that the cube is defined using six planes one for each side<sup>16</sup>, all the points on the sides of the cube would be pulled to the eight corners. If the accurate approximation of the cube is required this could be achieved by having an additional processing stage applied when all the vertices are located on the surface requiring the vertices to move to the nearest vertex on the bounding surface.

The second test procedure is illustrated in Figure 5.47. In this situation the sphere is completely contained within the cube although offset from the origin. The rigidity of each vertex of the sphere is the same and is set to be 255, see Section 4.6.1. This permits significant deformation at each iteration. In addition, the value of  $\alpha_i$  is 0.1. This corresponds to the step size at each iteration. In this situation, the modelling terminates after 58 iterations. Again in this situation the effects of the external forces result in a non-uniform deformation of the sphere. In particular after 10 iterations as illustrated in Figure 5.47 (c), the external forces have pulled the vertices of the sphere down towards the cube surface at a faster rate than the upper part of the sphere. This is because the update at each iteration controlled by the  $\alpha_i$  parameter is relative to the difference between the current and the set length of the mesh-lines, see Equation 4.10 in Chapter 4.

<sup>16</sup>In this section, the same cube and the sphere are used for the generation of the results shown in Figures 5.46 to 5.51. These can be achieved with a cube defined with fewer vertices. This is to make it easier to visualise the results

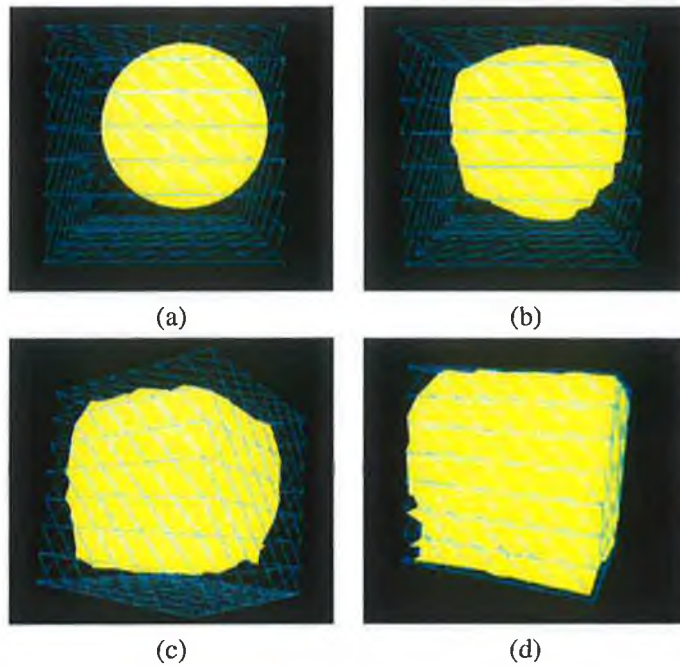


Figure 5.47: The results for the second active-mesh trial involving a sphere placed inside a cube and offset from the origin in the x and y direction. (a) shows the initial shapes, (b) shows the evolution after 2 iterations, (c) shows the iterations after 10 iterations and (d) shows the final shape of the sphere after 58 iterations.

The final surface is not the same as in Figure 5.46 although it shares similar characteristics, including the fact that all the points of the sphere rest on the surface of the cube and that the points on the surface do not attract to edges or corners and the point of intersection on the surface of the cube is different than in Figure 5.46.

The third set of results illustrates the situation when the sphere is not completely enclosed by the cube. In analysing this situation it was necessary to modify calculation of the external energy because as described above the normal to a vertex on the sphere is projected away from the surface to intersect the cube and thus when a vertex of the sphere is outside the cube no intersection can be determined, reducing the external energy to zero at a vertex. If no intersection can be determined using the normal vector to a vertex then the normal vector is inverted. The intersection of the inverted normal vector and the cube is then determined. In this situation the determination of the correct intersection has the added complexity that the normal vector will in general intersect the cube at more than a single location, i.e. it will pass through one side of the cube and out the other side. Thus a distance measure is necessary to determine the closest point of intersection. The Equation to calculate the distance between a 3D point and a plane, see Equation A.33 in appendix A, is used to establish the closest point of intersection.

The results for this are shown in Figure 5.48. The situation is shown after 10, 20 and 64 iterations. Here the sphere approximates the shape of the cube although it does not clearly take on the true shape of the cube all the vertices of the sphere lie on the surface of the cube.

and to show in Figure 5.51 how the cube is moulded into a sphere

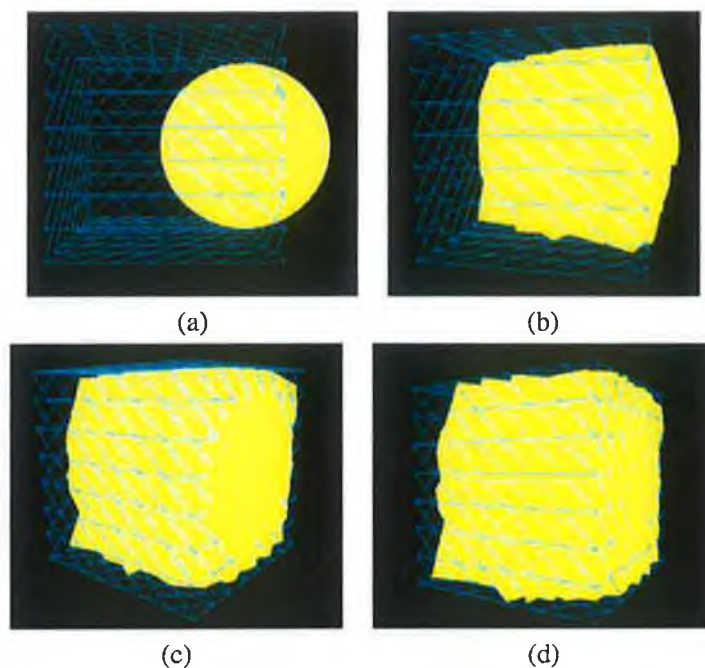


Figure 5.48: The results for the third active-mesh trial involving a sphere placed partially inside and outside the cube. (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations and (d) shows the final shape of the sphere after 64 iterations.

Figure 5.49 illustrates an alternative approach towards the calculation of the external energies. In this situation it was proposed that rather than using the normal vectors to determine the point of intersection with the cube that the vector projecting from the centroid of the sphere through the vertex would be used to establish the external force acting on a vertex. This provided equally good results for the situations shown in Figures 5.46 and 5.47 but when applied to the situation when the sphere was not completely bounded by the cube the sphere did not provide a close approximation of the cube. In Figure 5.49 the initial position is shown in part (a) and then after 2, 10 and the final position are shown in parts (b), (c) and (d) respectively. It is clear that the sphere does not approximate the cube. In Figure 5.49 (e) the position of the vertices of the sphere are shown and it can be seen that a high concentration of vertices are found on the side where the sphere was initially placed.

In each of the tests illustrated in Figures 5.46 to 5.49, each mesh element had the same rigidity. This permitted the mesh to be moulded to take on the shape of the bounding volume. In each of these cases the termination of the moulding process occurred when all of the mesh vertices on the sphere coincided with the bounding surface. However, when the rigidity varies across the surface of the object being deformed, all of the vertices will not reach the surface of the bounding shape. The reason behind this is that if the number of iteration is permitted to run indefinitely, even the highly rigid parts of the object being deformed would coincide with the bounding surface. If this happens then the internal structure of the mesh is not preserved. Thus in Figure 5.50, where the sphere is defined with varying rigidity, and in subsequent tests with varying rigidity, the moulding



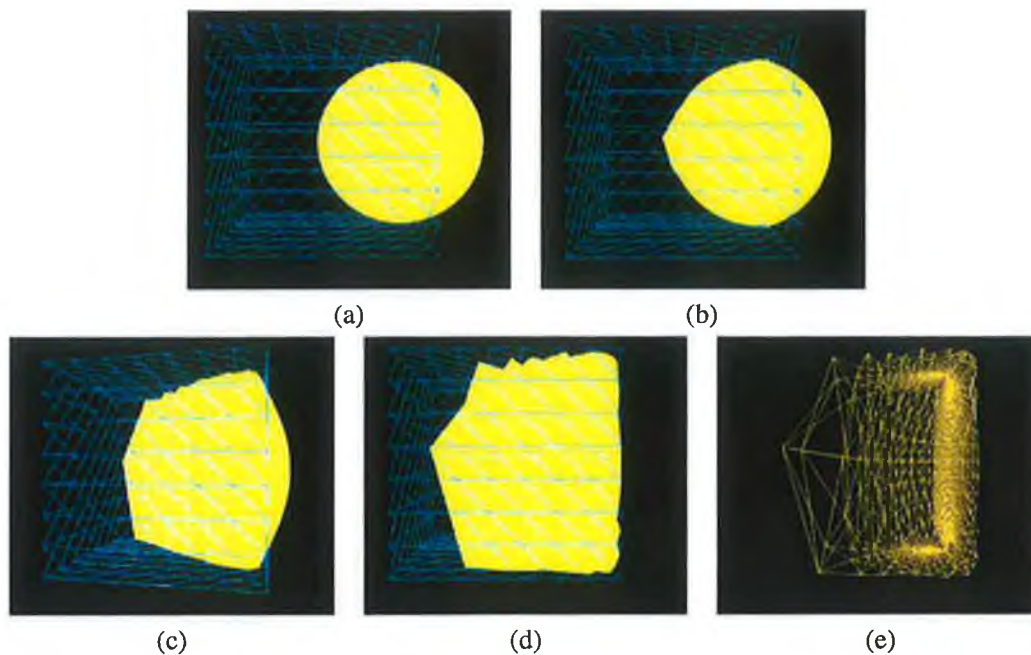


Figure 5.49: The results obtained with an alternative method for calculating the external energy. (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations, (d) shows the final shape of the sphere after 51 iterations and (e) shows the uneven distribution of the vertices of the deformed sphere.

process is terminated when the most elastic points are on the surface of the bounding surface. This ensures that the shape approximates the bounding volume but importantly that the structure of the highly rigid parts of the mesh are preserved.

Changing the value of the  $\beta_L$  parameter in Equation 4.4 in Section 4.6 has the effect of changing the rigidity of the surface of the sphere and limits the effects of the external forces acting on a vertex. In Figure 5.50 half the sphere has its  $\alpha_L$  parameter in Equation 4.4 set to 0.001 which is one hundred times smaller than that used in the previous examples. The initial position of the sphere is shown in Figure 5.50 with the red part of the sphere representing the part that is free to deform and the blue part representing the part that is highly constrained to keep its original shape. In Figure 5.50 (a) the initial position of the sphere inside the cube is shown. In Figure 5.50 parts (b), (c) and (d) subsequent iterations are shown and in part (e) and the final position is shown and in part (f) the structure of the moulded sphere is shown.

In the active-mesh implementation of Molloy & Whelan (2000) setting the  $\beta_L$  parameter to 0.001 would result in a very small movement of the mesh elements and a small change in the length of the mesh-lines because the effects of the internal forces are significantly strong in comparison to the external forces. However in 3D using relative internal energies results in the fact that the internal energies can find equilibrium at different distances from the centroid of the shape. Thus as the external forces pull vertices towards the bounding shape the effects of the strong internal forces will result in the shape maintaining its original structure while expanding or contracting towards the bounding shape.

In the following set of tests the cube is deformed to take on the shape of the sphere. Firstly

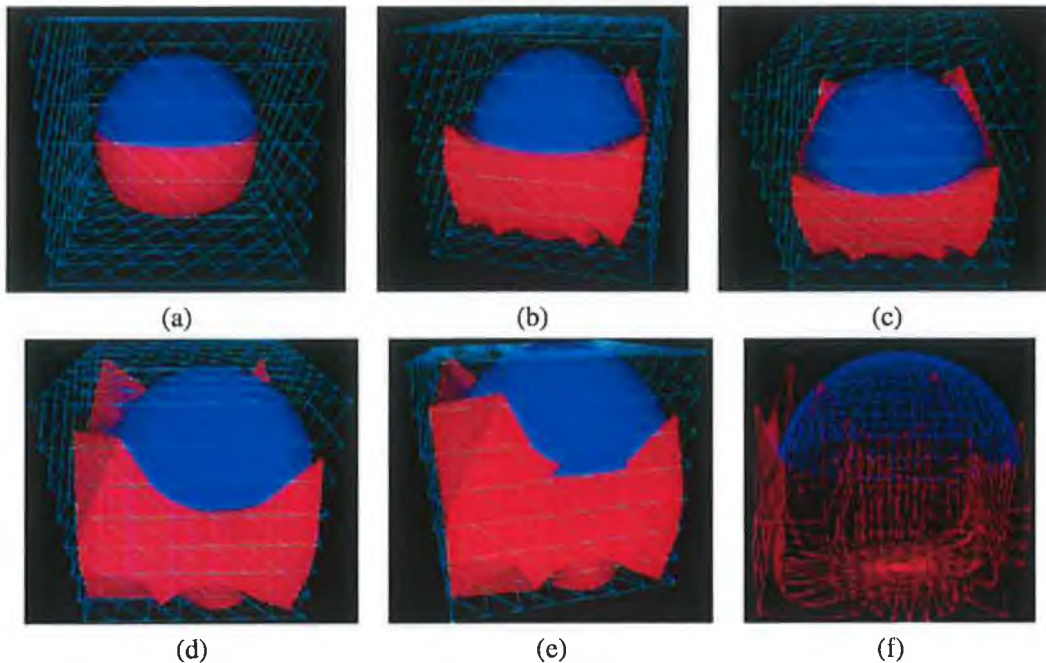


Figure 5.50: The modelling of a sphere with half the vertices having strong rigidity and half having high elasticity (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 20 iterations, (d) shows the iterations after 30 iterations, (e) shows the final shape after 37 iterations and shows (f) shows the structure of the moulded sphere.

the sphere and the cube have been scaled to be approximately the same size, resulting in parts of the cube being inside the sphere. Different stages of the modelling process are shown in Figure 5.51. In Figure 5.51 (a) the initial positions of the sphere and the cube are shown, in part (b) the positions are shown after 10 iterations and after 30 iterations in part (c) and final position is shown in part (d).

The results of the deformation show that the cube can be deformed to take on the shape of the sphere and that with such a shape it is possible to get a good approximation. In particular, with no edges and corners the sphere is well approximated by the deformed cube which has less vertices than the sphere<sup>17</sup>.

In Figure 5.52 an example of how a primitive shape can be moulded to take on a complex shape is shown. In particular, this example shows how a sphere can be deformed to take on the shape of the head of the underlying model. In Figure 5.52 (a) the initial geometry of the sphere and the head are shown. In parts (b), (c) and (d) the objects are shown after 5, 10 and 20 iterations respectively. The sphere has high elasticity and low rigidity allowing significant surface deformation.

### Assessment of the 3D Fitting

The moulding of the sphere to take on the shape of the cube is assessed from two perspectives. The first measure, quantifies the difference in volume of the sphere and the cube. The second measure determines the distance from each tri-face on the sphere to the nearest point on the surface of

<sup>17</sup>The sphere has 946 vertices and 1890 polygons while the cube has 295 vertices and 588 polygons.

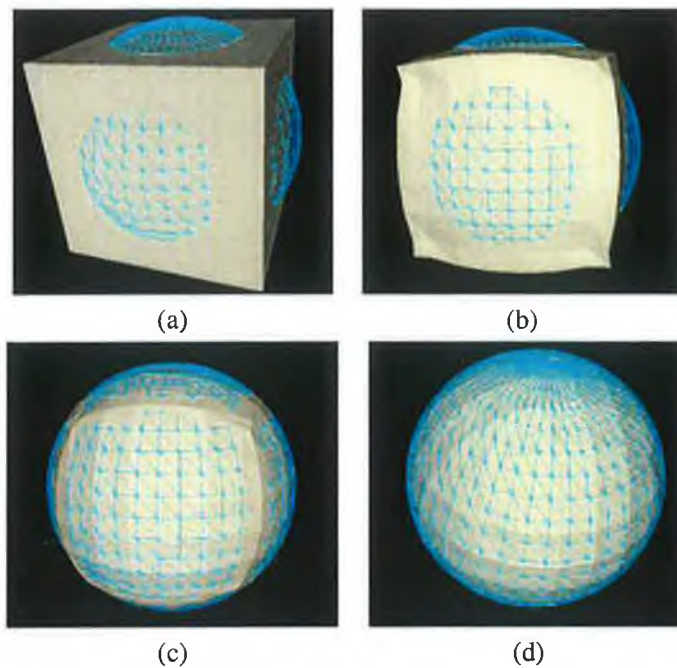


Figure 5.51: The modelling of a sphere starting with a cube (a) shows the initial shapes, (b) shows the evolution after 10 iterations, (c) shows the iterations after 30 iterations, (d) shows the final shape.

the cube. The accuracy of the fitting can be determined by the values in Table 5.7. In each case the difference in volume is less than 10%. The 10% difference corresponds to the fitting of the constrained sphere to the cube. In this case part of the sphere is constrained to preserve its shape. In Figure 5.46 the difference is 7% this is accounted for because the sphere can not completely deform to the corners of the cube. In the other examples, the fitting is closer to the bounding volume. In particular, in Figure 5.52. The difference has reduced to 2%, which indicates a highly accurate fitting of the sphere to the head. This is important because it provides the flexibility of the active-mesh technique to mould to approximate a particular shape even when starting with a geometric primitive. In Table 5.7 the

	initial sphere volume /voxels	cube volume /voxels	final sphere volume /voxels	difference
Figure 5.46	5.57 (radius = 1.15)	13.52	12.53	7%
Figure 5.47	5.57 (radius = 1.15)	13.52	13.12	3%
Figure 5.48	5.57 (radius = 1.15)	13.52	12.15	10%
Figure 5.50	5.57 (radius = 1.15)	13.52	10.81	2%
	initial cube volume	sphere volume	final cube volume	difference
Figure 5.51	13.52	11.84	11.52	3%
	initial sphere volume	sphere volume	final head volume	difference
Figure 5.52	7.05(radius = 1.29)	11.84	6.95	2%

Table 5.7: The difference in the shapes volume before and after fitting process. In each case the dimensions of the cube are  $2.382 \times 2.382 \times 2.382$ .

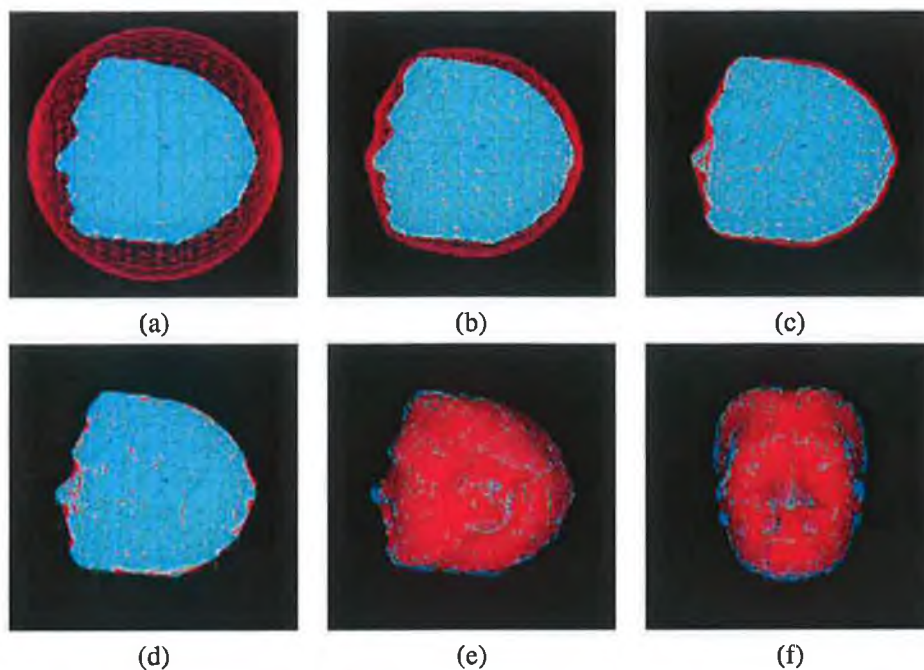


Figure 5.52: The modelling of a sphere to approximate the head of the underlying model. The initial position of the sphere and the head are shown in (a) and the moulding at 5, 10 and 20 iterations is shown in (b), (c) and (d). (e) and (f) show two additional views of the moulded sphere after 20 iterations.

In Table 5.8, the difference between the centre of each tri-face on the underlying model is presented. The centre of each tri-face is chosen because in the highly elastic cases the vertices are located on the surface of the bounding shape and thus a zero difference measure would result. The results in Table 5.8 provide a measure of this distance. In particular, these results highlight that the final distance between the centre of each tri-face and the bounding volume is less than one pixel in each case, illustrating an accurate fitting to the bounding volume and in particular, the difference between the model and the bounding volume in Figure 5.52 is 0.0097 which indicates a very accurate fitting. This illustrates that it is possible for the sphere to be deformed to accurately represent the head which concave and non-concave surfaces.

	average distance at 0 iterations	average distance at final iteration
Figure 5.46	0.5316 pixels	0.0123 pixels
Figure 5.47	0.5316 pixels	0.0137 pixels
Figure 5.48	0.7316 pixels	0.1837 pixels
Figure 5.50	0.4374 pixels	0.1207 pixels 0.0208 pixels(for red part of sphere) 0.2413 pixels(for blue part of sphere)
Figure 5.51	0.6023 pixels	0.0120 pixels
Figure 5.52	0.3786 pixels	0.0097 pixels

Table 5.8: The average distance, measured in pixels, from the surface of the underlying shape to the bounding volume, measured before and after the moulding process.

The overall result of the fitting shows that it is possible for one shape to moulded to take on the shape of another and that the accuracy of the approximation is determined by the user defined parameters  $\alpha_i$  and  $\alpha_L$ . This particularly evident in Figure 5.50 where the highly constrained part of the sphere with  $\alpha_i$  set to 0.001 and  $\alpha_L$  set to 0.03 which gives a larger error than the unconstrained part of the sphere.

### 5.6.3 Application of Active-Meshes to Human Modelling

This section provides a description of how the active-meshes are applied to the modelling of humans. This describes how the constraints are included and how the shape deforms under the influence of multiple objects. This occurs because the bounding volume can either be a continuous surface or can be formed using a number of sub-surfaces, and also each of the body parts of the underlying model is a separate object. Thus, it was first necessary to define a super-mesh class that can group several sub-objects that combine to form some overall object. An illustration of the class structure is shown in Figure 5.53.

The super-mesh contains three variables that are used for scaling and positioning of all the objects relative to each other. In particular the *max\_vals* and *min\_vals* contain the maximum and minimum coordinates for all the coordinates that make up the sub-objects in the group. This is used to establish scale-factors for the object. The centroid is used for the positioning of the objects. The super-mesh also contains a set of vectors that contain information for a sub-mesh, which is important in establishing the forces and maintaining the structure of the meshes. This is not directly incorporated in the mesh structure, as it is shared by each of the vertices that form an object. The mesh class structure contains the information associated with a single mesh element (or vertex). The position coordinates contain the set and current positions of the vertex and the *old\_set* position is used when the set position is updated in the following iteration. The *index\_pointer* is a reference list to the index of the vertices connected to the current vertex.

The effects of this are shown in Figure 5.54 where the underlying model which consists of 16 separate meshes and the bounding surface which consists of six different elements are aligned in 3D. The underlying model is first vertically scaled to be approximately the same size as the bounding volume. Then, if necessary, it is scaled horizontally and then the depth is scaled. This ensures that the bounding volume and the underlying model approximate each other. The alignment of the four other bounding volumes and the underlying model are shown in Figure 5.55.

In the back and front view in Figure 5.54 the model and the bounding volume the arms are not perfectly aligned with the arms of the model. The alignment of the arms is important, because if the arms are badly aligned then the external forces will pull all points to one side of the arm on the bounding volume. The alignment is performed by rotating the arm about the  $z$ -axis. The angles of rotation are determined from key features that are illustrated in Figure 4.11 in Chapter 4. The same parameters are available for the underlying model. The angle of rotation is the angle between the vertical line passing through the left or right shoulder and the line joining the shoulder and either the *max\_horizontal* or *min\_horizontal* points. An example of how the angles are calculated is shown in Figure 5.56 parts (a) and (b). The vertices of the model are transformed under the transformation that maps the model's shoulder's highest vertex to the origin. The vertices are first rotated through the angle obtained for the model and then rotated back through the angle

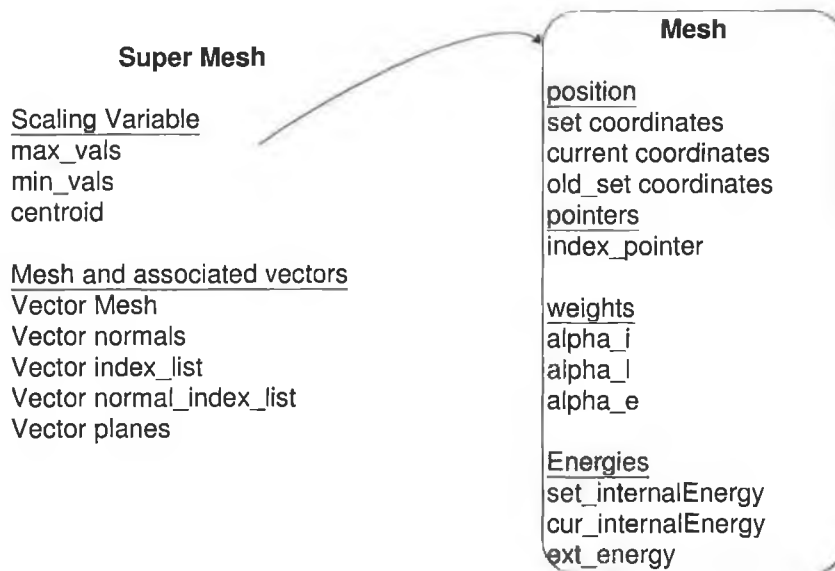


Figure 5.53: The class structure for the super-mesh class and the associated mesh class. The vectors in the super-mesh class contain lists of data that is important for each mesh but is not directly incorporated in the mesh class structure.

established from the individual's silhouette. The results of the rotation are shown in Figure 5.56 part (d). The angles between the vertical for each of the arms on the model and the five silhouettes are detailed in Table 5.9;

Bounding volume	left arm (degrees)	right arm (degrees)
model Figure 5.56 (a)	38°	38°
Figure 5.56 (c)	16°	18°
Figure 5.55 (a) & (b)	24°	18°
Figure 5.55 (c) & (d)	18°	20°
Figure 5.55 (e) & (f)	24°	28°
Figure 5.55 (g) & (h)	63°	59°

Table 5.9: The angles between the vertical and each of the arms for the bounding volumes generated in Section 5.5.

Furthermore, the width of the arms is not clearly identifiable in the side views thus, there is an uncertainty in the shape of the arm consequently the arms must have a strong rigidity to ensure that the shape of the arms is preserved. The  $\alpha_L$  for the arms is set to 0.3 and indicated in Figure 5.58 by the parts of the model shaded green.

The first test to see how the underlying model deforms to the bounding volume produced the result in Figure 5.57 (a). In this situation, the values of  $\alpha_i$  and  $\alpha_L$  are set to 0.1 for each mesh element. To try and improve the results and to ensure that the mesh vertices moved towards the correct surface the idea of constraint planes are introduced through which a vector normal to a vertex cannot pass. If the normal vector passes through the plane before it intersects a mesh face, then it is not considered as a valid intersection. The planes are shown in Figure 5.57 (b). The

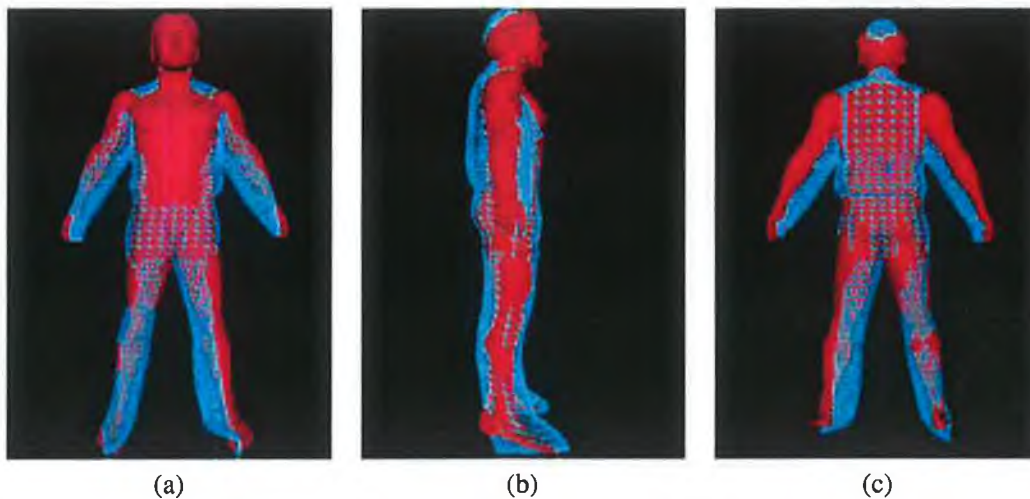


Figure 5.54: Three views of the aligned underlying model and the bounding volume generated in Figure 5.39.

planes are introduced using the key feature points illustrated in Figure 4.11 in Chapter 4. In Figure 5.57 (c) the result of the deformation are shown when the planes are included.

The application of the internal constraints to determine how the model can deform is important to preserve the realism of the model while deforming to the shape of the bounding volume. The user has the option to specify the strength of the internal and external constraints that act on particular section of the object being deformed or else the defaults can be used (Defaults are used for the modelling of particular shapes and are preserve highly irregular parts of the underlying mesh e.g. the face attempts to retain the same structure counteracting the effects of the external forces attempting to pull the vertices towards the bounding surface.).

Two methods are proposed to enable the specification of the internal constraints. The first, involves projecting the mesh to a 2D plane and the user clicks on the regions that are to have particular rigid and elastic properties. The user also has the option to specify how many levels<sup>18</sup> should be effected by this change. This results in the changing of the colour of that vertex. Projecting the mesh to a 2D plane is important, because when navigating around a 3D structure some parts are occluded. The second approach, enables the user to specify the internal parameters directly on the mesh in 3D.

The specification of the constraints is made using an interface that is provides the user with the ability to select certain vertices and assign it a particular value of  $\alpha_i$  and  $\alpha_L$ . Two examples of how the constraints are assigned are shown in Figure 5.58.

<sup>18</sup>A level in this context relates to the vertices connect to the selected vertex. For example, if the level is zero then the selected vertex has its values of  $\alpha_i$  and  $\alpha_L$  set, if the level is one then the vertex selected and the vertices connected to it are set, etc.

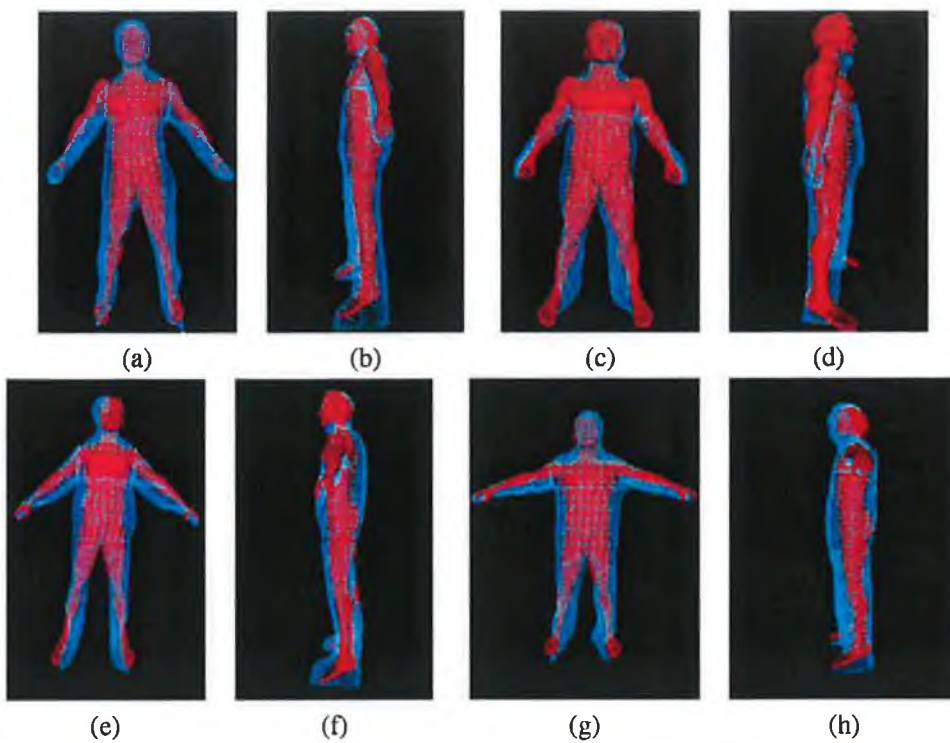


Figure 5.55: (a) and (b) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.40. (c) and (d) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.41 (e) and (f) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.42 (g) and (h) show two views of the aligned bounding volume and the underlying model for the bounding volume in Figure 5.43.



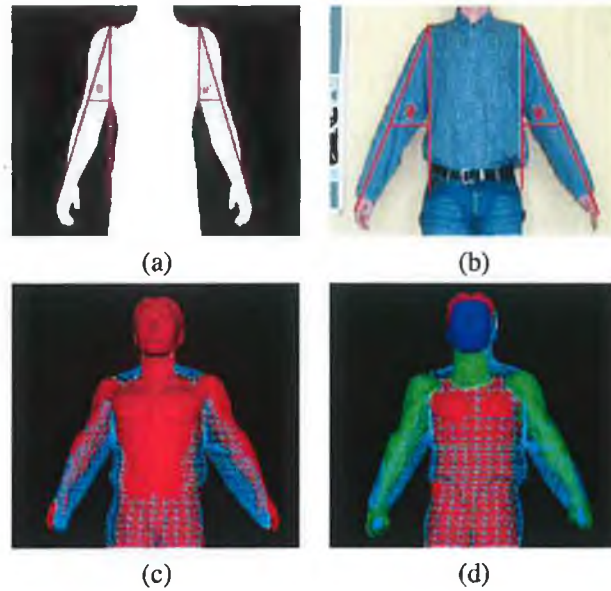


Figure 5.56: (a) shows an example of how the angle of rotation is calculated on the model, (b) shows the equivalent angles on the captured individual, (c) shows the original position of the arms before their position has been corrected and (d) shows the corrected arm position.

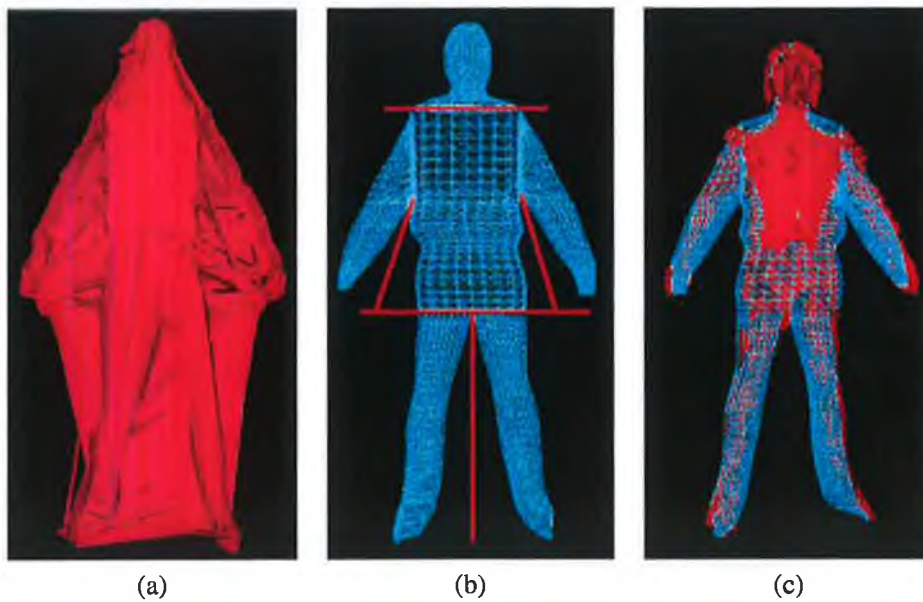


Figure 5.57: (a) shows the effects of modelling the human without constraints, (b) shows the constraints introduced as planes and (c) shows the effects of the planes when only the external energy is used to deform the underlying model.

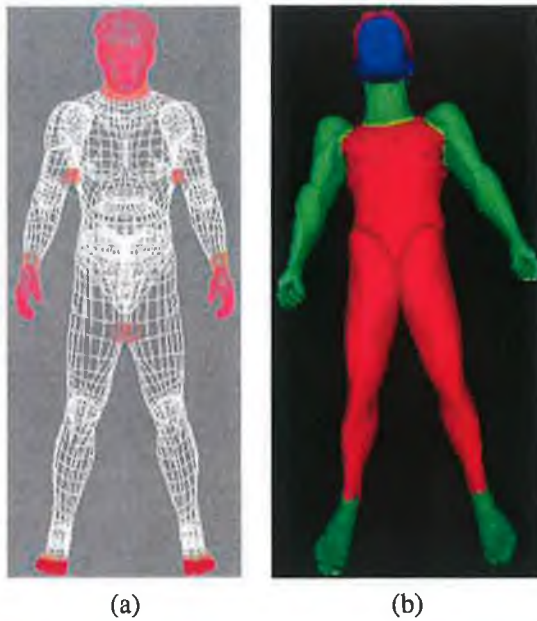


Figure 5.58: (a) shows the default initialisation of the constraints on the human model, (b) shows an alternative set of internal constraints. In (a) the areas highlighted in red have strong rigidity and the areas in white are free to deform. In (b) the areas highlighted in blue have strong rigidity, in green have strong rigidity and strong elasticity and the areas highlighted in red have weak rigidity and strong elasticity.

## Modelling of the Head

In Figure 5.54 (c) it is clear that the head requires strong constraints to maintain the internal structure. Different combinations of internal forces have been applied to the face to maintain the structure. Initially, the area around the eyes nose and mouth had strong rigid parameters but the parts of the face surrounding the features deformed to make the face unrecognisable. Thus, the entire face is strongly bound together with strong internal constraints. Examples of the head deforming to take on the shape of a cube and a sphere are shown in Figure 5.59. In both situations, the initial head is shown in Figure 5.59 (a) and the internal constraints are  $\alpha_l$  and  $\alpha_i$  are set to 0.001 for the face, and  $\alpha_l$  is set to 0.01 and  $\alpha_i$  is set to 0.01 for the rest of the head.

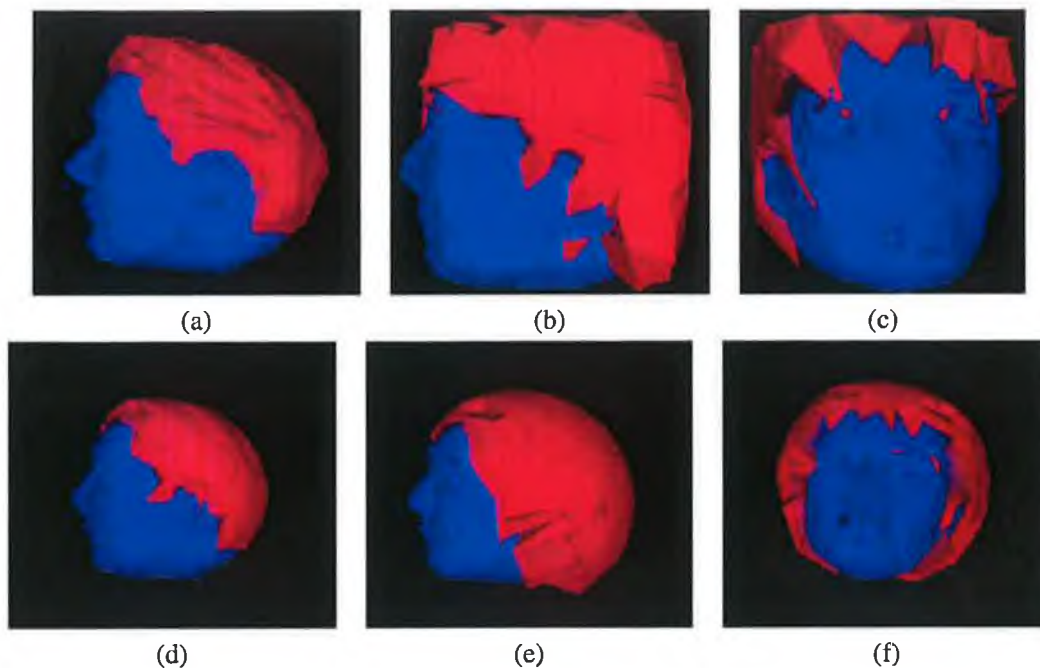


Figure 5.59: The application of active-meshes to moulding the head into a cube and a sphere while maintaining the internal structure of the face. (a) shows the initial head. The blue region is highly constrained and maintains its structure, (b) and (c) show two views of the final shape that approximates the square. In (d) the shape of the head deformed partially to the sphere and in (e) and (f) show two views of head deformed to the sphere.

Table 5.10 contains the results of moulding the head to take on the shape of a cube and a sphere. The first set of results measures the difference in volume between the head and the cube and the sphere. This provides a measure of how the head has been moulded. The second set of test provides an error measure based on the distance from the centre of each tri-face on the head to the bounding surface. This second measure permits the average error across the surface to be measured. The distance measure shows that while the vertices, shaded in red, on the head are on the surface of the cube and sphere respectively, the blue parts are not on the surface. This results in a larger average difference in volume than in Table 5.8. The tri-faces generated using the vertices are not on the surface although in both cases illustrated in Table 5.10 the average difference is small, at sub pixel level. In both cases, the average distance from each tri-face is greater for the

blue parts of the head because the internal constraints preserve the internal structure. Examining the distance from each tri-face the distance is substantially less than one pixel.

	initial head volume /voxel	bounding volume /voxel	final head volume /voxel	difference
head to cube	7.18	13.52	11.87	12%
head to sphere	7.18	10.55	9.5	10%
	average distance at 0 iterations	average distance final iteration	average distance for blue part	average distance for red part
head to cube	0.419	0.196	0.2893	0.2269
head to sphere	0.688	0.141	0.2569	0.1293

Table 5.10: The difference in the shapes volume before and after fitting process and the distance between each tri-face on the head and the bounding volume. The dimensions of the cube are  $2.382 \times 2.382 \times 2.382$ . and the sphere has radius of 1.58

### Modelling of Body

The modelling of the human is carried out as a single iterative task. The only requirements before the models can be moulded is that the models are aligned in 3D and that the internal constraints have been assigned to highly detailed parts of the mesh and any additional constraints that are required to limit the deformation of the model have been defined. If the above-mentioned constraints are not satisfied then the deformation of the model cannot be predicted.

In Figure 5.61, the internal constraints are set according to Table 5.11. In this situation, the face has strong rigidity and the arms have weaker value of rigidity where as the rest of the body is highly elastic. The corresponding values are detailed in Table 5.11. In 5.61 (a), the initial model and its bounding volume are shown, in (b) an intermediate stage in the deformation of the underlying model is shown and in (c) the final position of the underlying model is shown with its bounding volume.

body part	$\alpha_l$	$\alpha_i$	body part	$\alpha_i$	$\alpha_i$
head	0.1	0.1	face	0.01	0.01
left upper arm	0.05	0.03	upper body	0.1	0.1
left fore arm	0.05	0.03	pelvis	0.1	0.1
left hand	0.05	0.03	thighs	0.1	0.1
right upper arm	0.05	0.03	calf	0.1	0.1
right fore arm	0.05	0.03	feet	0.1	0.1
right hand	0.05	0.03	neck	0.05	0.03

Table 5.11: The constraints that are applied to the different body parts of the underlying model.

The deformation of the underlying model can be assessed from a number of perspectives. The parts of the model that have high elasticity deform to adopt the shape of the bounding volume. These correspond to the parts of the models that are shaded in red. The parts of the model that are shaded in green have adopted the shape defined by the bounding volume, but do not converge to the bounding volume. The primary reason for this is that the process of moulding the underlying

model is terminated after a certain number of iterations. The user can determine this or it can be determined when the parts of the model that are highly elastic have been moulded to adopt the shape of the bounding volume. If the moulding process is permitted to run till all control points are positioned on the bounding volume, the influence of the internal constraints would be reduced. In the results shown for the modelling of the underlying model in this section the results are shown after 10 iterations. The parts of the model that are shaded blue have high rigidity it can be seen that features on the face are preserved while closely approximating the bounding volume.

To enable the face to be successfully deformed to approximate the individual's bounding volume the head is realigned independently of the rest of the body. This is necessary, because of the fine detail of the face. Initial tests highlighted that because of the strong rigidity between the mesh elements of the face that more than 40 iteration were required for the head to approximate the mould of the bounding volume.

The realignment is achieved automatically by using the key features in the side view of the individual's silhouette. In particular the position of the nose is used to set the depth and height at which the head of the underlying model is placed. This alignment ensures that the deformation of the underlying model's head will be consistent with the bounding volume and that the number of iterations required to have convergence will be reduced.

Observing the deformation of the hands and feet, using the active-mesh implementation, it can be seen that with the rigidity parameter  $\alpha_L$  set to 0.05 and the elasticity  $\alpha_e$  set to 0.03 the hands and feet are free to deform to the bounding volume, but still retain characteristics of the underlying model, i.e. the fingers and toes can still be recognised. If the elasticity is reduced then the hands can retain more closely the shape of the underlying models hands.

Examining the structure of each of the body parts, to see how they deform under the influence of the bounding volume it is noticed, that the internal constraint that bind the mesh elements together, stop individual mesh elements from deforming sporadically to distant parts of the bounding volume. This reduced the need to use more constraint planes across at each of the joints to preserve each of the body parts. On a static model the importance of this may be overlooked but when the model are to be animated this is particularly important. Examples of this are shown in Figure 5.60.

The results of the application of active meshes to human models created corresponding to the images in Figure 5.2 are shown in Figures 5.62 to 5.65. In these figures parts of the model are not equally well preserved, for instance, around the crotch in Figure 5.62 the surface is not continuous. The constraint planes are the primary reason for this because if the nearest point on the surface of the bounding volume is at the behind of the plane then the line normal to a vertex will not locate the nearest point and the next nearest point on the back of the bounding volume. This can result in a cavity in the surface. This illustrates the importance of accurately positioning the planes.



(a)



(b)

Figure 5.60: Illustration of how the shape of the individual body parts are well preserved in the active-mesh implementation. (a) shows the left upper leg the lower leg and the foot and (b) shows the upper arm, the forearm and the hand.

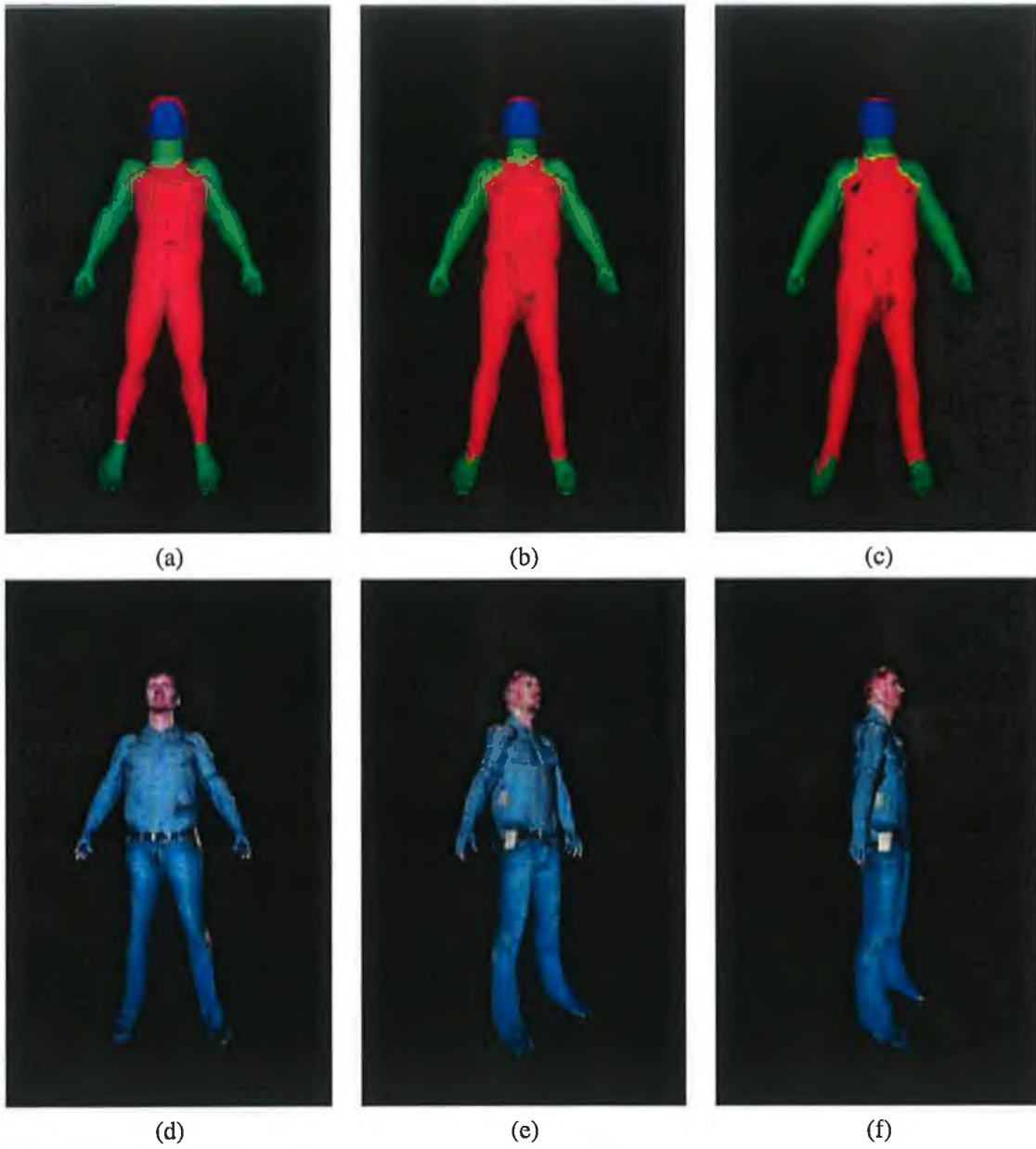


Figure 5.61: An example of the deformation of the model to approximate the bounding volume in Figure 5.39.

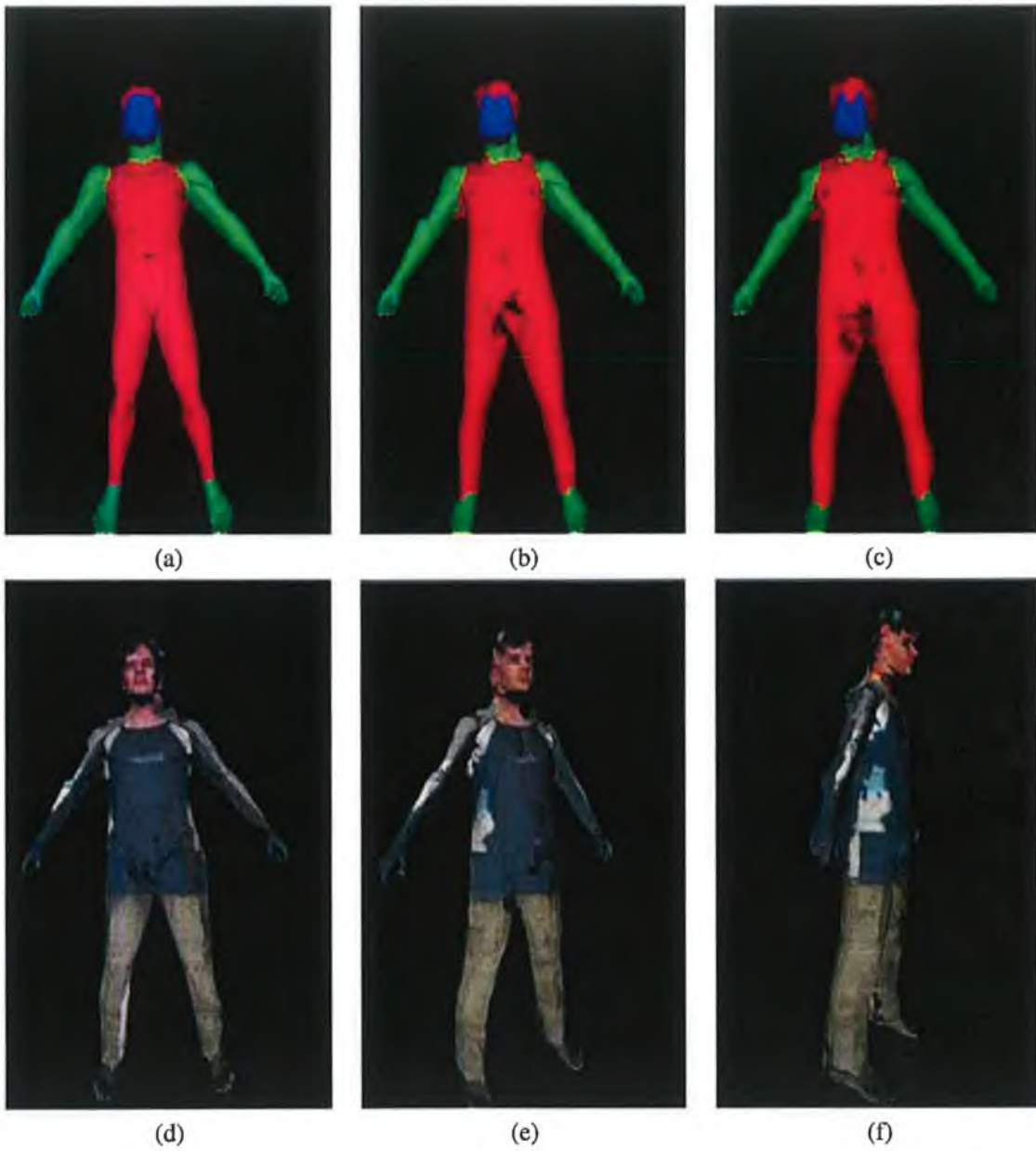


Figure 5.62: An example of the deformation of the model to approximate the bounding volume in Figure 5.40.



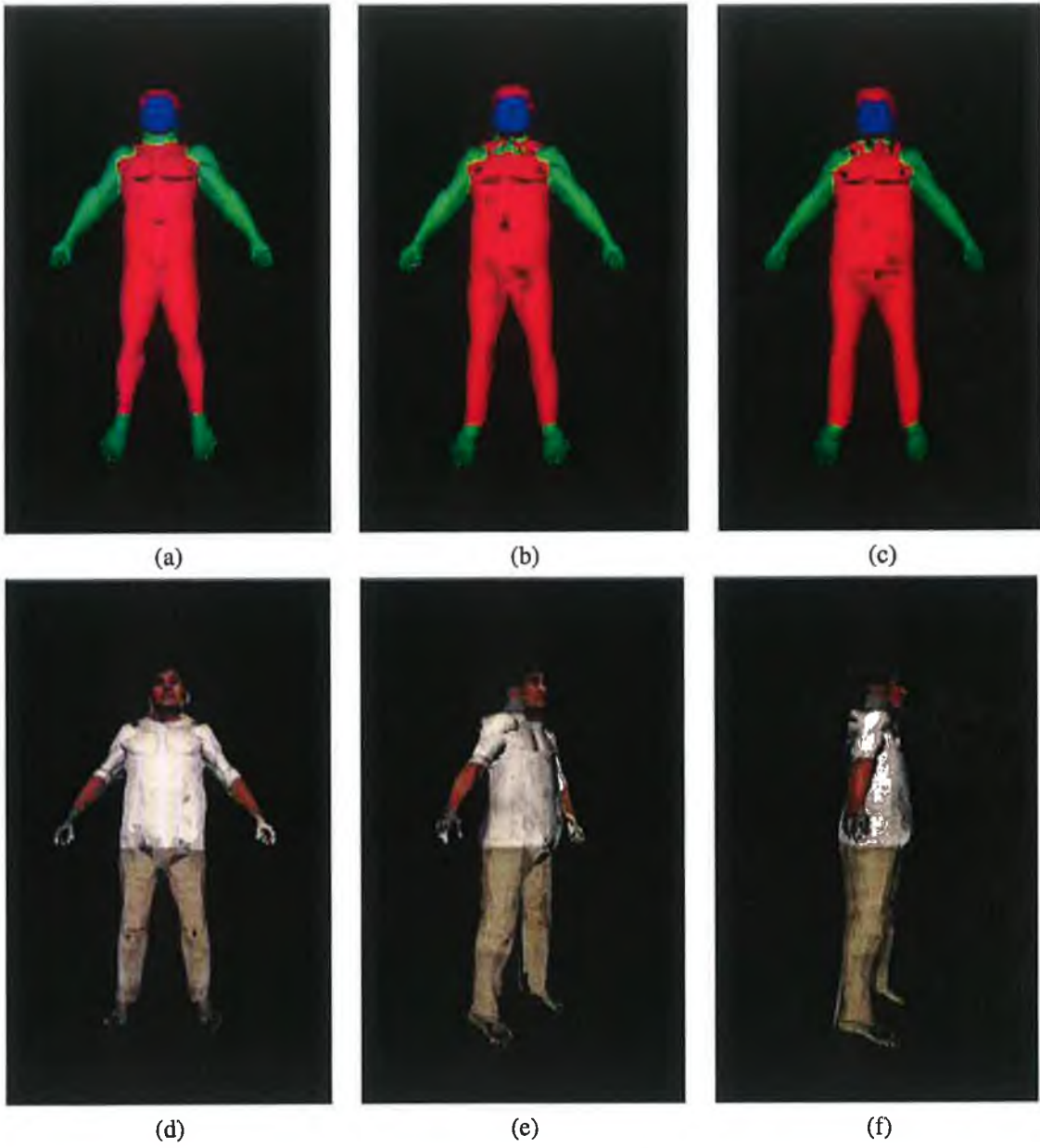


Figure 5.63: An example of the deformation of the model to approximate the bounding volume in Figure 5.41.

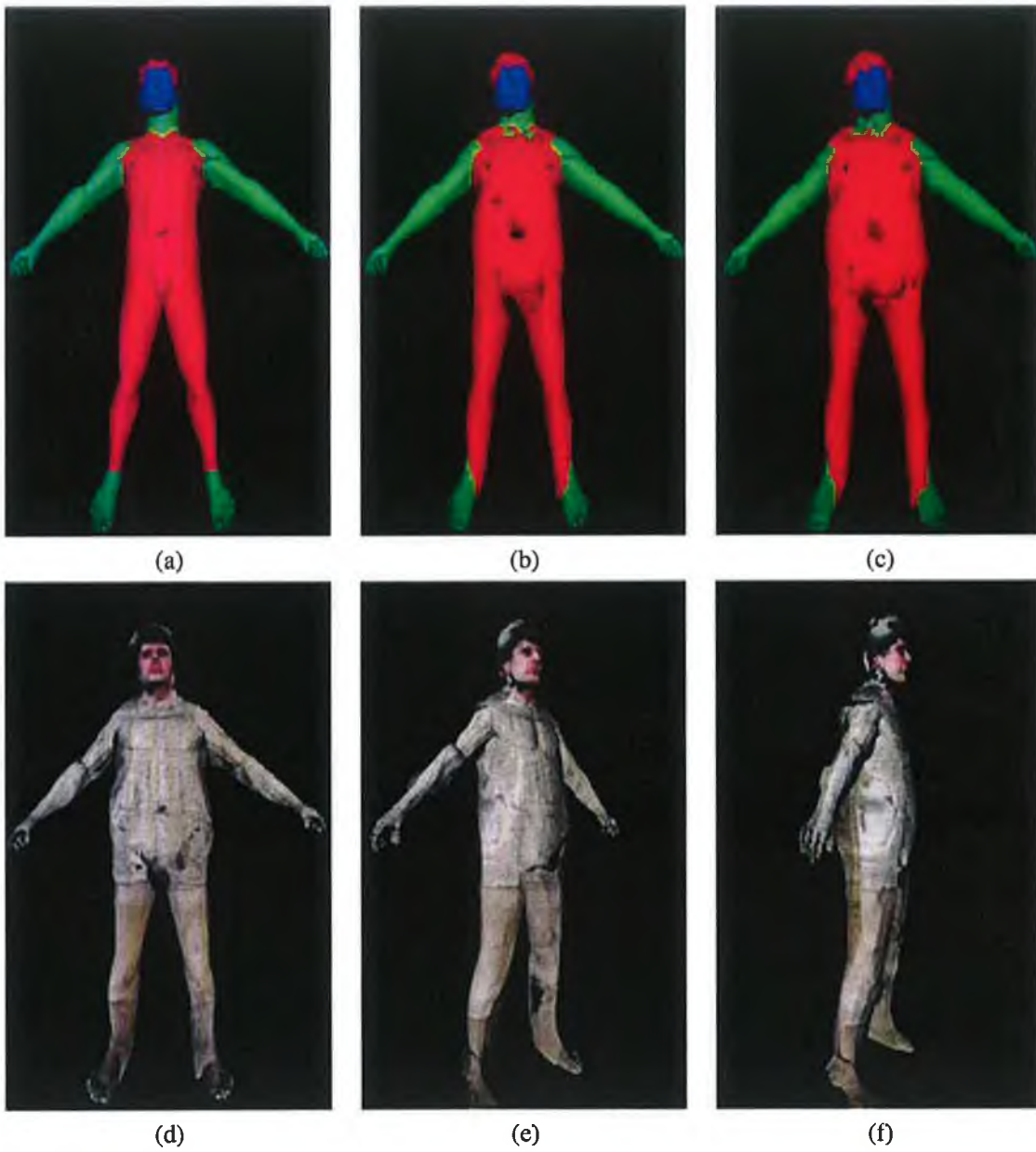


Figure 5.64: An example of the deformation of the model to approximate the bounding volume in Figure 5.42.

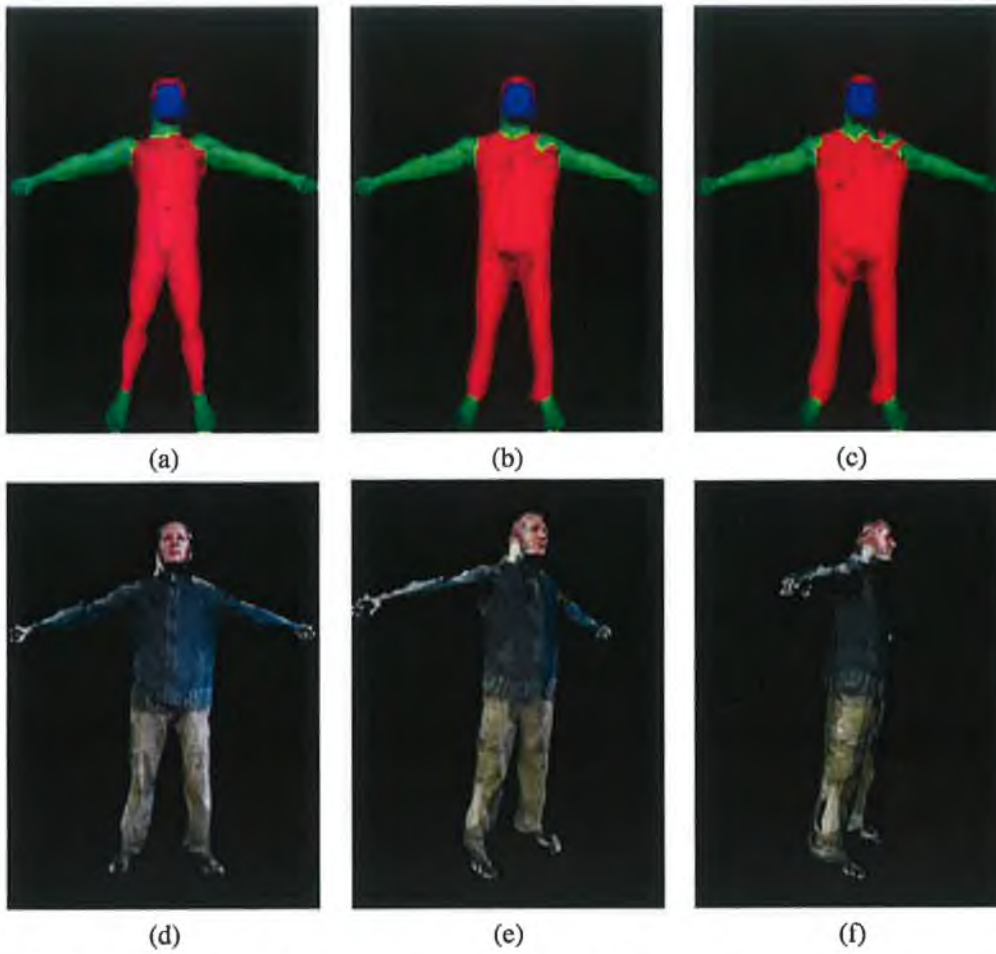


Figure 5.65: An example of the deformation of the model to approximate the bounding volume in Figure 5.43.

### Accuracy of Active Meshes Applied to Human models

This section describes how the underlying model is deformed to take on the shape of the bounding volume. The use of the volume measure in Section 5.6.2 cannot be reliably used to determine how close the model approximates the bounding volume because the structure of the underlying model has a certain amount of overlap where the body parts are connected. Thus, the combined volume for each of the parts would be greater than the volume of the underlying model. Therefore this section presents the accuracy of the moulding process using the distance from the centre of each tri-face on the underlying model to the bounding volume. Figure 5.66 (a) and (b) show the bounding volume that is produced when the silhouettes of the underlying model are recombined. Figure 5.66 (c) shows the underlying model aligned with the bounding volume and Figure 5.66 (d) shows the final model after 20 iterations.

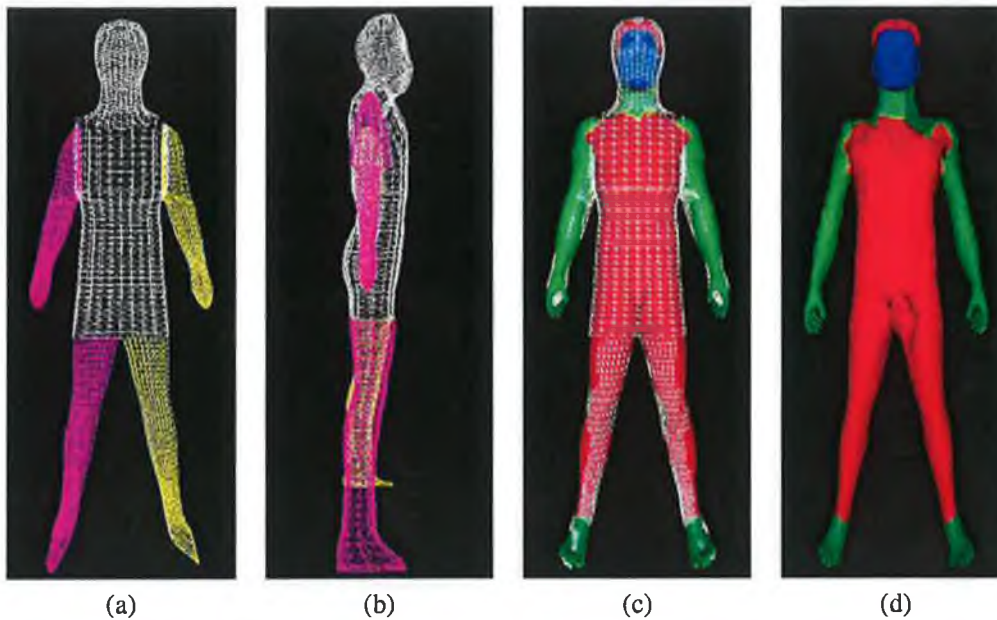


Figure 5.66: The modelling of the underlying model. (a) shows the front view of the bounding volume created for the model, (b) shows a side view of the models bounding volume, (c) shows the initial model aligned with the bounding volume and (d) shows the moulded underlying model after 20 iterations. From the 2D silhouettes of the model, the height of the model is 615 pixels, the width from left hand to right hand is 220 pixels and the depth of the model 80 pixels.

Figure 5.67 depicts the decrease in error over 20 iterations. The figure shows the combined error for each of the body parts at each iteration. The light blue line shows the total MSE error at each iteration. The pink line shows the MSE error related to the parts of the model that are shaded in green. The MSE error associated with the red parts of the model are illustrated with yellow line and the dark blue line shows the MSE error associated blue parts of the model. The total MSE error is generated by adding each of the MSE error for the red, green and blue parts of the model. At each iteration the error progressively decreases by a smaller amount. This occurs because the movement of each point is related to its distance from the bounding surface.

The errors illustrate that the MSE error decreases at each iteration and the role of the user

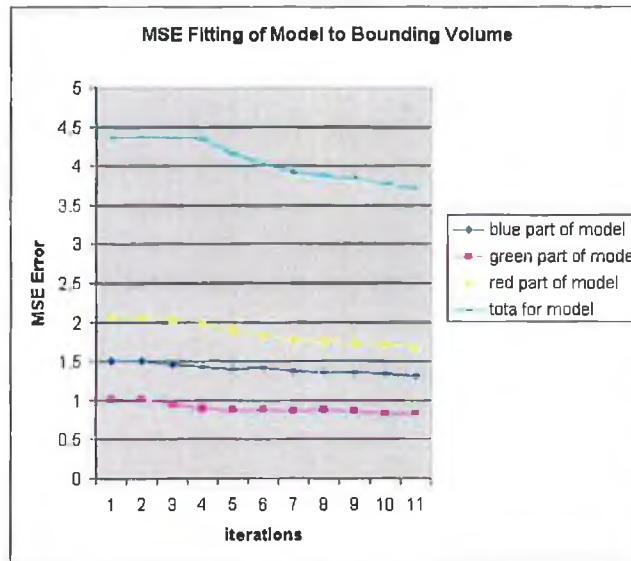


Figure 5.67: Illustration of how the MSE error decreases over 20 iterations.

defined constants dictate how this decrease proceeds. For example, the error associated with the red parts of the model is calculated with  $\alpha_L$  set to 0.1 and  $\alpha_i$  set to 0.1 providing significant deformation and thus the MSE error decreases faster than that for the green<sup>19</sup> and blue<sup>20</sup> parts of the model. In the results shown in Figure 5.67. The variance at each iteration was calculated. The variance for the parts of the model shaded in red decreased from 1.70 pixels to 1.57 pixels over the 20 iterations. For the blue part of the model the variance decreased from 1.30 pixels to 0.98 pixels and the parts the model shaded in green the variance decreased from 0.86 pixels to 0.82 pixels.

## 5.7 Discussion on Results

The approaches described in Chapter 4 have been fully implemented and the testing was completed at each stage using real-world images. In addition, a core set of images has been selected to highlight the creation of the models. Using these images allows a comparison to be made between each of the models that are created. The results presented in this chapter show an evolution from the texturing of a default model using the captured data through to an implementation of the active-meshes that enables the moulding of an underlying model to take on the shape of the bounding volume that is generated using two of the captured views.

The results at each stage provide a greater personalisation of the underlying models using the captured data. In the first approach, the personalisation of the final model is achieved using the captured images to texture the underlying model. This is suitable for use in virtual worlds and in computer games where the characters in the game can be personalised to take on the appearance of the individual. In the approach, these results were enhanced using the localisation of facial features to improve the texturing on the face. The improvement of the texturing of the face was

<sup>19</sup>The green parts of the model shaded green have  $\alpha_L$  set to 0.5 and  $\alpha_i$  set to 0.3

<sup>20</sup>The blue parts of the model shaded green have  $\alpha_L$  set to 0.1 and  $\alpha_i$  set to 0.1

chosen, as the face contains significant detail that can be used to identify the individual in a virtual world. While the use of the facial features improves the texturing of the underlying model, it does not use the shape information that is extracted using the active template.

A first approach to incorporate the shape information is achieved through the creation of a bounding volume using the extracted silhouettes. The bounding volume lacks the fine detail that is necessary to provide true personalisation of the model. The texturing of the bounding volume improves its quality and in certain applications, it can provide a suitable representation of an individual. Apart from the texturing of the bounding volume, the main element that is associated with this method is that the shape of the individual is strongly incorporated in the model.

Finally, to improve the quality of the model and to incorporate the shape information, the application of the active-meshes is used to mould the underlying model to take on the shape of the bounding volume. A series of tests are detailed that show how the internal, external energy and combination of the energies effect the deformation of the objects. The strength of the internal forces have been varied to preserve parts of the model that have strong rigidity. This is an important procedure that enables the reconstruction of fine features that are not present in the bounding volume. Furthermore, the use of the underlying model ensures that joint positions, which are essential for animating the model, can be located and thus, the final model can be animated with existing animation streams.

### **Animation of the models**

In each case, the animation of the models was considered. This is important for the integration of the model into a virtual environment. With the exception of approach 4, the models can be animated using existing animation streams. This is possible because the joint locations on the underlying models are known. In Figure 5.68 (a), (b) and (c), different views of the animated models produced in approach 3 are shown. In each case, the models are seen undergoing a series of movements. In Figure 5.68 (d), (e), (f) and (g), different views of the animated models produced in approach 5 are shown. In each case, the models are seen walking across the screen in different directions.

### **Template Minimisation Issues**

The role of the template minimisation described in Section 5.3.5 illustrates that, although the initialisation of the template is successful even in cluttered backgrounds, the correct minimisation cannot always be automatically obtained. Against simpler backgrounds, the minimisation procedure performs very well. This is particularly important because the final position of the template provides the input for the 3D reconstruction of the individual. Thus to ensure that the template can accurately extract an individual from the environment, it is realised that a high contrast background is best suited to the use of the template if limited set of images are used for the extraction of the individual. Since the extraction of the individual from a highly cluttered environment is a challenging task, the template fitting procedure facilitates user interaction if certain control points get trapped in local minima. In Section 5.3.6, the fitting of the 2D template to the silhouette of the model was tested. The results indicated that the template can accurately define the boundary of

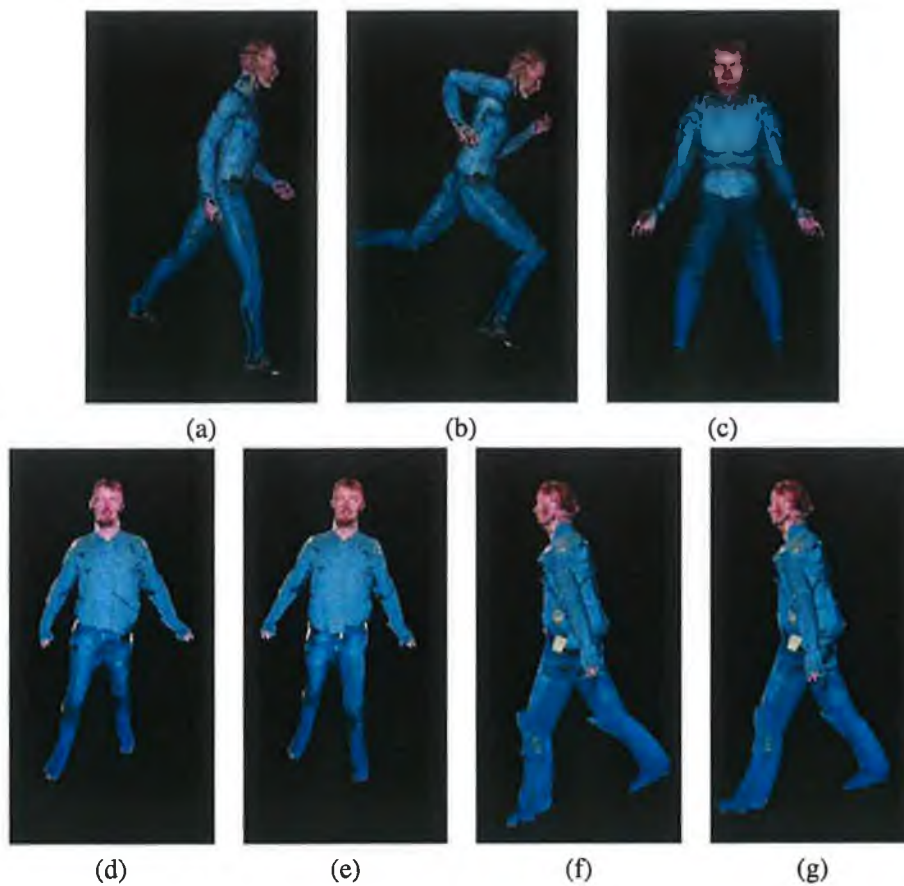


Figure 5.68: Animation of the models. (a) to (c) show the animation of the models created in approaches 2 and 3. (d) to (g) show the animation the active mesh models.

the an individual. The accuracy increases as the number of control points increses.

### Texturing of the Models

The texturing of the underlying model using the normal vectors to determine which of the captured images should be used to texture a particular part of the model proved successful in approaches 1–4. The limitations of the texturing procedure are evident in approach 5 when the direction of the normals on the underlying model can undergo significant variation from the original orientation, and in certain situations texturing using the normals will result in the front of the model being textured with image information in the back image if the normal is not calculated correctly.

As highlighted in approach 3, the effects of natural light on the images can cause significant variation in the quality of the final models. This is particularly evident on the faces in Figure 5.64 and 5.65. Thus, the use of image blending as used in (Lee, Goto & Magnenat-Thalmann 2000) would improve the quality of the final models. This is particularly evident on the side of the models.

## Active-Meshes

The implementation of active-meshes, developed in Chapter 4 were tested for the moulding of different shaped objects. In particular, it was shown that it is possible to rebuild fine detail that is not possible to reconstruct from a limited number of views. To determine the accuracy of the moulding process, two measures were devised, based on a volume measure and distance measure. The volume measure was used in modelling simple objects, where a comparison between the volume of the deformed object and the volume of bounding object were made. These results illustrated that the internal constraints determined how close the deformed object approximates the bounding volume. In certain situations when the internal constraints are weak, the moulded object volume was within 2% of the bounding volume. However, when the two objects were not correctly aligned this difference in volume grew to 10%. The distance measure provides a means to show how close the underlying model approximates the bounding volume. In the simple cases illustrated in Figures 5.46 to 5.51 the difference is below 1 pixel. However when the sample measure is applied to the human model as a whole the difference increases because the model has fine detail around the face, hands and feet, thus there are significant model vertices needed to describe these parts of the model. Thus the difference between each of these parts of the body and the bounding volume increases. Furthermore, the error increases because the model has overlapping body parts and some of the vertices are not on the surface, thus the distance between these vertices and the bounding volume surface is greater.

## Comparison with other Techniques

The results presented provide a selection of models that can be incorporated in different virtual worlds and improve an individual's virtual experience. The key elements that can be used to assess the quality of the model are:

- The complexity of the reconstruction and capture process
- The photo-realism of the final models,
- How accurately the shape of the model approximates that of the individual,
- The number of polygons used<sup>21</sup>,
- The animation of the model and how realistic the animation is.

In the procedures described, the complexity of the reconstruction process is reduced to what is believed is a minimum. In particular, the reconstruction of the model is achieved from two views. The resulting bounding volume can be combined with an underlying model to incorporate the animation information. Alternative approaches have been presented for improving the photorealism of the models through the use of facial features. This provides a low-cost method to texturing the face of the model. This is in comparison to the techniques of Hilton et al. (1999) and Lee, Goto & Magnenat-Thalmann (2000), where a complete 3D image of the head is created. In (Lee, Goto & Magnenat-Thalmann 2000), a separate facial reconstruction is used to improve the quality of

---

<sup>21</sup>This is not possible since other approaches reviewed do not discuss the number of polygons used.



the model. This requires the capture of additional images and makes it unsuitable for a home-user to create an accurate head model, and in (Lee & Magnenat-Thalmann 2000), a semi-automated feature localisation process is required to identify the key features on the face. In Section 5.4.3, a method using facial features is presented to improve the reconstruction of the face of the model. This is a technique that can be simple and reliably carried out.

Hilton et al. (1999) discuss some limitations of the low-cost reconstruction process. These limitations include the fact that fingers cannot be localised in either the side or front views, and with the size of the images the width of the hand is only measures a few pixels. Thus, in the final reconstruction, the hands of the underlying models are preserved. A similar issue exists with the reconstruction of the feet. These concerns are evident in the work of Lee, Goto & Magnenat-Thalmann (2000), and in (Wingbermhle et al. 1997, Weik et al. 2000), the reconstruction of the hands is not attempted, and in certain situations the created models have no hands. In the approaches developed in this thesis, it is not possible to automatically extract the arms in the side views because they are within the visual hull of the silhouette generated from the side view, but an accurate approximation of the arms achieved. If the user permits the hands to deform, it is possible that they can approximate the hand of the captured individual. Moreover, additional shape information is extracted to attempt a better reconstruction of the feet.

In (Hilton et al. 1999), the capture process is simplified to enable the automated creation of human models and in (Lee, Goto & Magnenat-Thalmann 2000), the process is extended to real environments but requires user assisted identification of key features, and in (Lee, Gu & Magnenat-Thalmann 2000), it is stated that only simple environments are considered. Both of these approaches have simpler capture set up than those proposed in (Weik et al. 2000) and (Kakadiaris & Metaxas 1998). In the capture process presented in this thesis this is further simplified permitting the use of a standard digital camera in any environment without the use of any special lighting requirements. In (Lee, Goto & Magnenat-Thalmann 2000), the possibility of using energy minimisation techniques are ruled out because of the time to convergence to the correct solution. However, it is shown in Section 5.3.3 that when the individual is localised in the image and the active contour is expressed in the form of a template, active contours provide a suitable method for extracting the individual from the background, and in highly cluttered environments, the use of the template significantly reduces the user interaction for feature identification.

A summary of other aspects that can be compared between the existing methods and the current implementation are shown in Table 5.12

	Boyle	Hilton et al. (1999)	Lee et al. (2000) Lee et al. (2000a)	Villa-Uriol et al. (2003)	Wingbermuhle et al. (1997) Weik et al. (1998)
Number of views required	4	4	3	video capture	video capture
size of images	640 × 480	756 × 582	N/A	N/A	N/A
Segmentation from background	Automatic Template Fitting	chroma-keying	heuristic based edge growing	frame difference	background subtraction
Method of segmentation	automatic	automatic	manual	automatic	automatic
Environmental constraints	indoor	lighting blue screen	none	turntables multiple cameras studio	turntables multiple cameras studio
Feature identification	Automatic	Automatic	Manual	N/A	Manual
Method of reconstruction	silhouette based with 3D active-meshes	deforming model using captured data	deforming model using captured data	silhouette based	silhouette based
Animation of the final model	default streams	default streams	default streams	N/A	N/A
Intended application	Virtual Human	Virtual Human	3D Avatar	3D Avatar	video conferencing

Table 5.12: Further aspects of comparison between various approaches.

## Conclusions

The creation of virtual human models for the population of virtual worlds and the enhancement of a user's virtual experiences is a challenge that has been met with the provision of an array of techniques that facilitate the creation of numerous models. The quality of the model depends on the quality of the captured data, which is influenced by the capture equipment and the environment in which the data is captured. This is one of the key innovations of this research, as none of the reviewed literature attempts to enable a home-user to create their own models in unconstrained environments. This is realised in a system that imposes no environmental or equipment constraints on the capture of the data and no pre-required level of computational competency to complete the creation or modification of the 3D model.

Two main themes are present and interspersed within the bounds of this research. The first is the extraction of the individual from real environments. This resulted in the creation of a template that can be automatically used to extract an individual from a real environment. Alternative approaches, using a combination of filters and imposing restrictions on the environment were considered, but ultimately rejected, as they were not deemed robust enough to extract an individual from more complex backgrounds.

The second major theme that is fundamental to this research, is the personalisation of the human model from the extraction of sufficient data from a limited set of images, to create a realistic 3D model that can be integrated into any environment. This was initially achieved by the texturing of an underlying model and expanded to provide greater personalisation of the model by providing a silhouette based reconstruction of the individual using the extracted silhouettes of the individual in two views. This provided a bounding surface that is unique to the individual. This bounding surface was considered as an active surface that was used to actively mould an underlying model to take on the shape of the individual. The approach developed incorporated both external and internal constraints, appropriately weighted and combined in an energy minimisation framework that provides the home-user with a modelling tool that can be uniformly applied to the modelling of any object with extensive control over the deformation of the model.

## 6.1 A Brief Review

In chapter two, active contour models are described as a method for extracting the boundaries of arbitrarily shaped objects from complex backgrounds. The review focused on the development of the active contour models, as a method of incorporating both high-level notions of object segmentation and low-level aspects of feature detection to generate a sophisticated method that is active in nature. The review also identified alternative formulations of the active contour model that incorporate additional energy functionals or alternative solutions to the energy minimisation equations. This review concludes with an investigation of templates that incorporate the active framework and a description of active-meshes that are applied to tracking objects.

Chapter three describes a range of techniques, that have been applied to the 3D reconstruction of objects. The techniques described have contrasting requirements facilitating the creation of high quality models with the incorporation of LoD information and photo-realistic models that are generally lower quality models, but can be enhanced using photographic information available from captured images. It is stated that no unified approach exists for the reconstruction of an individual and that technology and applications are the major factors that influence the quality of the final model. In this chapter, the minimum requirements necessary to enable a home-user to capture sufficient data to create a photorealistic model are presented. These requirements impose strict limitation in terms of the capture device (number of cameras, and image size) and reconstruction techniques that can be used while generalising the capture environment.

Chapter four sets out the approaches developed to generate human models in real environments. The approaches detail the progression from the capture and extraction of individuals from simple environments to the more complex backgrounds with a high level of clutter. In addition, the personalisation of the final models is continually increased starting with a simple texturing of an underlying model and progressing to incorporate an individual's shape into the modelling of the final model. Each of the approaches developed is automated to facilitate a home-user to simply create a photo-realistic model. Within this chapter the major contributions of this research are set forth, and it is clearly outlined how the objective of providing a generally accessible system for the creation of photo-realistic human models in any environment is achieved.

The texturing method developed in the first approach enables the simple texturing of the underlying model to take on photo-realistic appearance of the individual, using the data captured from either two or four views. This extraction of the image data is enhanced in the second approach to enable the extraction of the individual from real cluttered environments by the introduction of an innovative whole body constrained template, automatically initialised close to the individual in any environment and that can adjust to the pose the individual adopts. The personal data that is extracted using this template is textured to the underlying model using the previously developed texturing approach. This approach improves the quality of the models generated, although the models generated have the same body. Thus, an approach that enables the greater personalisation of the final model is required.

This is first achieved in approach three, where the facial features of the individual are considered to improve the quality of the texturing of the face. This was considered important, as it is the most detailed part of the model and contains significant detail, most importantly to humans, to

allow the recognition or distinction of the individual.

The personalisation of the model is extended in approach four, by providing a silhouette-based reconstruction of the individual, using the silhouettes that are extracted in each view. This approach facilitates the incorporation of the shape information that is extracted using the human template in approach two. This shape information is sufficient to create the maximum silhouette equivalent of the individual, and when textured, exhibits the photo-realistic appearance of the individual, facilitating the creation of models that are specific to an individual.

In approach five, the task of animating the bounding volume is considered in conjunction with the reconstruction of the non-convex surfaces that cannot be recovered using silhouette based reconstruction techniques. This manifested in a novel implementation of active-meshes and their extension to 3D to deform an underlying model in an active framework. The net result of the active-mesh moulding of the underlying model facilitates the reconstruction of the non-convex parts of the model and the incorporation of joint information that cannot be reliably extracted from the silhouettes<sup>1</sup>.

In chapter five, the approaches are verified through extensive testing of each part of the approaches developed in chapter four. The first phase of the testing involved the capture of the images to create the personalised models of the individual. In particular, this involved an examination of the images to determine a level of clutter and the application of filters and other segmentation techniques to the extraction of the individual from different environments. This was preceded by the testing of the human template. This involved firstly determining that the underlying active contour model exhibited the properties of general active contour models. Then the incorporation of specific constraints to enable the generation of the template were verified. The initialisation of the template and the minimisation of the energy within were examined to show how the individuals were extracted from their environment. The fitting of the template was then compared to ground truth measures, by fitting the template to the silhouette of the underlying model. This provided a consistent method to verify that the templates can accurately extract the individual from their environment.

Subsequently, the texturing process developed in approach one was tested and a selection of models were created. These models were enhanced using the facial feature information. Following this, validation of silhouette-based reconstruction of an individual was provided, by generating and texturing of personalised bounding volumes. The generation of active-mesh models was considered and different scenarios were examined to see how the model could be deformed to take on the appearance of the bounding volume. The active-mesh implementation incorporates both large and small-scale deformations and the framework permits parts of the mesh to preserve the internal structure and parts of the mesh to deform to approximate the bounding volume. This approach provided a uniform way for modelling the underlying model to take on the appearance of the bounding volume but it was also demonstrated that it could be used to mould any shape to take on the appearance of another. This was extensively tested and the results are analysed to determine how accurately the bounding volume is approximated.

In particular, the quality of the models was considered by placing images of the individual in 3D space to examine the success of the reconstruction process.

---

<sup>1</sup>More generally, the active-meshes framework can be applied to mould any surface to approximate another.

## 6.2 Major Contributions

There are five key contributions central to this research and a number of associated contributions. The major contributions are:

- The provision of an automatic system for the creation of photo-realistic human models using images captured in real environments. This system does not require specialised equipment and does not require the images to be captured in a studio. Equally, the system does not require expert knowledge to create the models. This system is a flexible approach that can overcome possible errors in the capture process and provides a simplified approach to reconstructing the individual's model.
- The creation of a constrained human template that can be used to accurately extract an individual from any environment. This is not described or presented in any of the literature reviewed. This template incorporates the energy minimisation framework of active contours and uses dynamic programming to include constraints to control how it deformed to the local image data. The rate of convergence is improved by using a search space perpendicular to the contour. In addition, the template is automatically scaled, adjusted and initialised within the image using a novel subtraction technique to enable its automatic application in the captured images.
- A silhouette-based reconstruction of the individual from two views was achieved, thus validating it as a technique for human model creation. This was achieved in approach four, in which the silhouettes are combined using simple elliptical contours to approximate the non-convex data of the individual, while remaining within the visual-hull. This ensures that the shape information that is contained in the captured images is used to personalise the shape of the model.
- The implementation of active-meshes in 3D provides a tool that can be generically and uniformly applied to the deformation of any object. In particular, defining the underlying model as an active surface facilitates greater personalisation of the individual's model, through the provision of internal constraints, that control how rigidly the internal structure is connected, and external forces that work to mould the underlying model to the bounding volume. The balance of the constraints is shown in its uniform application to the human model with different parts of the body having different constraints. This ensures that the fine features on the face can be remoulded to approximate those of the captured individual. In general terms, this technique can be applied to the deformation of any shape to approximate any other shape. It facilitates the incorporation of constraints that can be simply defined to set the rigidity and elastic parameters at a particular mesh point. The internal and external constraints were fully tested on a range of shapes.

### 6.2.1 Associated contributions

Some of the minor contributions associated with this research are:

- An extensive review of active contours as a method for the extraction of both known and unknown objects from any environment and a review of techniques that have been successfully applied to the creation of human models. The template developed defined criteria for the automatic insertion or removal of control points to improve the extraction of data from a real environment.
- A review of existing techniques for the reconstruction of objects from a limited set of views and an examination of existing techniques for the creation of photo-realistic human models. This resulted in the identification of the key elements that are essential for the development of a flexible low-cost human modelling technique.
- The use of the B-spline contour as a method for the definition of the silhouettes provides a convenient method for storing 3D shape information.
- The texturing of the underlying model and the final models in the different approach is achieved using the normal vectors to determine which image should be used for the texturing of a particular mesh-face. This provides a simplified method for the texturing of the final models without requiring the generation of a 3D image.
- The active-mesh implementation provides a technique that is used to mould one shape to approximate another and can be used as a technique for reducing the number of polygons required to represent a particular object.
- The use of a 2D view of the object in the active-mesh approach to modelling provides a simplified method for the assignment of constraints to the object being deformed.

### **6.2.2 Limitations Identified In This Research**

The quality of the models is relative to the clothing that the individual is wearing. If the individual is wearing loose fitting clothes, then it is difficult to accurately extract the key features that are essential for the texturing of the models in the initial approaches and the creation of the bounding volume in approaches four and five.

The creation of a photorealistic human model is important for the enhancement of a home-users experience in a virtual environment, but from a limited set of views captured with a single camera, it is not possible to accurately reconstruct an individual. In particular, it was identified in Laurentini (1997) that is not possible to reconstruct a concave surface from a finite set of views. Additionally, the individual is wearing clothes that define the shape of the model, and thus if the clothes of the individual are changed then the model has to be modelled appropriately to account for the change of clothing.

The creation of the models is only one aspect that is important in the personalisation of a human model. Capturing the animation data associated with an individual is vital in personalising the individual's model and to give true realism to the model. This is not possible from a limited set of views.

### 6.3 Publications associated with this research

The following publications stem directly from this research. Each publication describes a particular aspect of this research:

**Adaptive Active Human Model Reconstruction** E. Boyle and D. Molloy, *IADAT Journal of Advanced Technology on Imaging and Graphics*, Vol. 1(1), September 2005, pp. 16-18, ISSN 1885-6411.

**Flexible Extraction of Human Shape in Real-Time Using Motion-Tuned Active-Contour Based Segmentation** O. Bompis, M. Sorin, E. Boyle, D. Molloy, *In Proc. IMVIP2005 - Irish Machine Vision & Image Processing Conference*, pp. 243-244, 30-31 August 2005.

**Flexible Silhouette Based Creation of Virtual Humans** E. Boyle and D. Molloy, *In Proc. IMVIP2005 - Irish Machine Vision & Image Processing Conference*, pp. 143-150, 30-31 August 2005.

**Automatic Construction of 3D Human Models from Only Two Orthographic Projections** P. Moore, E. Boyle, D. Molloy, *In Proc. Eurographics Workshop (Ireland)*, Institute of Technology Blanchardstown (ITB), Dublin, pp. 33-40, 3rd June 2005.

**A Review of Techniques for the Extraction of Shape Information for the Creation of Virtual Humans** E. Boyle and D. Molloy, *In Proc. Eurographics Workshop (Ireland)*, Institute of Technology Blanchardstown (ITB), pp. 25-32, Dublin, 3rd June 2005.

**Definition of B-Spline Templates for the Automatic Extraction of Human Shape Information** E. Boyle and D. Molloy, *In Proc. IADAT-micv2005*, Madrid, Spain, pp. 123-127, March 30 - April 1 2005.

**Using Facial Feature Extraction to Enhance the Creation of 3D Human Models** E. Boyle, B. Uscilowski, D. Molloy, N. Murphy, *In Proc. WIAMIS05 - 6th International Workshop on Image Analysis for Multimedia Interactive Services*, Montreux, Switzerland, 13-15 April 2005.

**Framework and Applications for Mobile Networks Using Synthetic Multimedia** E. Boyle, F. Brisc, S. Cooray, B. Uscilowski, A. Brosnan, R. Sadleir, C. O'Sullivan, *In Proc. 3G2004*, London, UK, pp. 644-648, 18-20 October 2004.

**Generation and Animation of Virtual Humans** E. Boyle, *In Proc. IWSSIP04*, Poznan, Poland, pp.143-146, 13-15 September 2004.



## **Auxiliary Presentations**

**An Adaptive Model Based Approach to the Creation of Virtual Humans** In RINCE Research Seminar Series, Dublin, Ireland, November 2004.

## **6.4 Future Directions for Research**

In this final section, a number of natural extensions to this research are considered that could be used to enhance the usefulness of the methodology. In conception, this thesis focused on utilising aspects from a wide range of image processing and computer vision techniques and thus the future work described in this chapter is varied relating to improvements in both domains.

Firstly, an alternative implementation of the 3D active meshes is considered that uses NURBS, introduced in Chapter 2 and appendix A. In particular, this approach is discussed as a possibility to improve the shape of the bounding volume and integrating the weighting factors into the calculation of the external forces in the 3D active meshes. Secondly, the possibility to use different templates that are generated based on the information used to initialise the template. This could, in turn, be used to provide a method for automatic selection of the underlying model that closely approximates the individual. Thirdly, in Chapter 4, the possibility of using different initialisation techniques was discussed. Motion tuned active contours provides an interesting alternative for initialising the contour and are discussed in greater detail. Another further possibility is examining the shape of the bounding contours that are extracted and combined in 3D. This opens the possibility of using the shape information in medical applications as a first stage diagnosis in weight related and posture analysis. Finally, another extension to this research is to investigate the possibility of animating the skin or the clothing of the model using the control points that define the surface of the model.

### **6.4.1 Incorporation of NURBS into the Active-Mesh Formulation**

To enable the extension of the active mesh formulation to 3D, the external surface was expressed in terms of B-spline control points interpolated with a cubic polynomial curve. In Meegama & Rajapakse (2003), the advantages of using NURBS representations were highlighted. In particular, the use of the weighting factor enables the curve to interpolate certain control points and to ensure that the B-spline curve can be used to accurately progress into cavities in the models.

The introduction of the weighting factors in the active-mesh formulation could be used to increase or decrease the effects of an external force at a point. In particular, the current system used the Euclidean distance to determine the external force that is exerted on the underlying model. This could be extended to exert a greater force to pull the shape being deformed into deep cavities on the object to be approximated.

In addition, the re-formulation of the underlying model as a B-spline or NURBS surface could be attempted to allow for greater deformation of the underlying model. This also presents the possibility of introducing additional control points when the surface is strongly deformed under the influence of the bounding surface.

## **6.4.2 Automatic Selection of Templates**

Apart from the initial scaling of the template, the bounding body gives additional positional information that can be used in the choice of template that is used. When the minimisation process is terminated, the relative positions of the key control points can be used as an indication to the best template to use. In particular, if an individual is wearing a dress then the position of the key point representing the crotch will be considerably lower than the centroid and this could be used to select a model that wears a dress.

Alternatively, the information in the side views could be used to select the best template based on side profiles of the models. This could be used improve the fit of the individual to the model and to improve the convergence time in the active mesh approach.

## **6.4.3 Template Initialisation Possibilities**

With the advances in the consumer technology market, the possibility exists that an individual will take advantage of video camera and the new video enabled mobile phones, and this provides an ideal cost effective method to implement the motion tuned active contours discussed in Chapter 4 and Bompis et al. (2005). The information that is extracted provides an ideal method for the initialisation of the template and then to apply the templates created within this thesis to accurately locate the boundary of the individual in the images.

In addition, the requirement for a fixed viewing position could be removed and gradient based or block-matching method utilised to align the different views. This would still require that the distance between the individual and the camera remains approximately constant between captures or else some scaling information should be included to ensure that the correct shape information is extracted.

## **6.4.4 Skin Animation using B-spline Control Points**

In this thesis one of the goals was to produce a human model that could be realistically animated within a virtual environment. While this has been achieved with a model that can be predefined animation streams, the fact is that to provide realistic animation, it is necessary to animate the skin or clothing of the model. This is a time consuming procedure that involves the calculation of the movement of each element of the model's surface. Determining this could be improved by firstly generating the 3D model that approximates the individual and then using this in conjunction with a real-time active contour tracker to feed the information about the moving control points to the surface model. This could be used to detail how the individuals skin moves. This approach to modelling has influence beyond the scope of human modelling and could be used to model how a piece of fabric moves.

# Bibliography

- Amini, A., Tehrani, S. & Weymouth, T. (1988), Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints, *in* 'The Second International Conference on Computer Vision', pp. 95–99.
- Amini, A., Weymouth, T. & Jain, R. (1990), 'Using dynamic programming for solving variational problems in vision', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(9), 855–867.
- Anand, V. (1993), *Computer Graphics and Geometric Modeling for Engineers*, John Wiley and Sons, Inc.
- Anton, H. (1994), *Elementary Linear Algebra, 7th edition*, Wiley and Sons, Inc.
- Babski, C. & Thalmann, D. (2000), Real-time animation and motion capture in web human director (whd), *in* 'Proc. Web3D and VRML 2000 Symposium', pp. 139–145.
- Ballard, D. (1981), 'Generalising the hough transform to detect arbitrary shapes', *Pattern Recognition* **13**, 111–122.
- Ballard, D. & Brown, C. (1982), *Computer Vision*, Prentice-Hall (Englewood Cliffs, New Jersey).
- Baumberg, A. (1995), Learning Deformable Models for Tracking Human Motion, PhD thesis, School of Computer Studies, University of Leeds, Leeds UK.
- Baumberg, A. & Hogg, D. (1994), An efficient method of contour tracking using active shape models, *in* 'Proc. IEEE Workshop on Motion on Non-rigid and Articulated Objects', pp. 194–199. Texas.
- Beardsley, P., Zisserman, A. & Murray, D. (1997), 'Sequential updating of projective and affine structure from motion', *International Journal of Computer Vision* **23**(3), 235–259.
- Bergevin, R. & Levine, M. (1993), 'Generic object recognition: building and matching coarse descriptions from line drawings', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(1), 19–36.
- Blake, A. & Isard, M. (1992), *Active Vision*, MIT Press, Cambridge, Massachusetts.

- Blake, A. & Isard, M. (1998), *Active Contours*, Springer Verlag.
- Bompis, O., Sorin, M., Boyle, E. & Molloy, D. (2005), Flexible extraction of human shape in real-time using motion-tuned active-contour based segmentation, in 'Proc. Irish Machine Vision & Image Processing Conference 2005'. IMVIP2005.
- Bottino, A. & Laurentini, A. (2001), 'A silhouette based technique for the reconstruction of human movement', *Computer Vision and Image Understanding* **83**(1), 79–95.
- Boulic, R., Fua, P., Herda, L., Silaghi, M., Monzani, J., Nedel, L. & Thalmann, D. (1998), An anatomic human body for motion capture, in 'Proc. EMMSEC 98'. Bordeaux, France.
- Boyle, E. (2004), Generation and animation of virtual humans, in 'IWSSIP04', pp. 143–146. Poznan, Poland.
- Boyle, E., Brisc, F., Cooray, S., Uscilowski, B., Brosnan, A., Sadleir, R. & O'Sullivan, C. (2004), Framework and applications for mobile networks using synthetic multimedia, in '3G2004', pp. 644–648. London, UK.
- Boyle, E. & Molloy, D. (2005a), Definition of B-spline templates for the automatic extraction of human shape information, in 'Proc. IADAT-micv2005', pp. 123–127. Madrid, Spain.
- Boyle, E. & Molloy, D. (2005b), Flexible silhouette based creation of virtual humans, in 'Proc. Irish Machine Vision & Image Processing Conference 2005'. IMVIP2005.
- Boyle, E., Uscilowski, B., Molloy, D. & Murphy, N. (2005), Using facial feature extraction to enhance the creation of 3D human models, in 'Proc. WIAMIS05'. Montreux, Switzerland.
- Brisc, F. (2004), Multi-resolution volumetric reconstruction using labelled regions, in 'Proc. 6th IEEE Southwest Symposium and Image Analysis and Interpretation', pp. 114–118. Lake Tahoe, USA.
- Brisc, F. & Whelan, P. (2004), Creating virtual models from uncalibrated camera views, in 'Proc. Irish EuroGraphics Workshop 2004'. University College Cork.
- Brophy, C., Dawson, K. & O'Neill, P. (2004), 3-D object pose determination, in 'Proc. IMAGE'COM 93, Bordeaux, France', pp. 363–366. March 23-25.
- Buxton, B., Dekker, L., Douros, I. & Vassilev, T. (2000), Reconstruction and interpretation of 3D whole body surface images, in 'Proc. Scanning 2000'. Paris, France.
- Canny, J. (1986), 'A computational approach to edge detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 679–698.
- Carranza, J., Theobalt, C., Magnor, M. & Seidel, H.-P. (2003), Free-viewpoint video of human actors, in 'Proc. SIGGRAPH '2003', Vol. 22, pp. 569–577.
- Caselles, V. (1995), Geometric models for active contours, in 'Proc. International Conference on Image Processing'. Washington.

- Caselles, V., Kimmel, R. & Sapiro, G. (1997), 'Geodesic active contours', *International Journal of Computer Vision* **22**(1), 61–79.
- Chandran, S. & Potty, A. (1998), 'Energy minimisation of contours using boundary conditions', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(5), 546–549.
- Chang, K. & Ghosh, J. (2000), Three-dimensional model-based object recognition and pose estimation using probabilistic principal surfaces, in 'Proc. SPIE: Applications of Artificial Neural Networks in Image Processing V', Vol. 3962, pp. 192–203.
- Chelappa, R., Wilson, C. & Sirohey, S. (1995), 'Human and machine recognition of faces: A survey', *Proceedings of the IEEE* **83**(5), 705–740.
- Cheung, K., Baker, S. & Kanade, T. (2003), Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture, in 'Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition', Vol. 1, pp. 77–84. CVPR03.
- Cohen, I. & Lee, M. W. (2002), 3D body reconstruction for immersive interaction, in 'Proc. Second International Workshop on Articulated Motion and Deformable Objects'. Palma de Mallorca, Spain.
- Cohen, I., Medioni, G. & Gu, H. (2001), Inference of 3D human body posture from multiple cameras for vision-based user interface, in 'Proc. 5th World Multi-Conference on Systemics, Cybernetics and Informatics'. Orlando Florida.
- Cohen, L. (1991), 'Note on active contour models and ballons', *Computer Vision Graphics Image Processing: Image Understanding* **53**(2), 211–218.
- Cohen, L. & Cohen, I. (1993), 'Finite-element methods for active contour models and ballons for 2-D and 3-D images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11), 1131–1147.
- Collins, G. & Hilton, A. (2001), *Software Focus*, Addison-Wiley. Modelling for Character Animation.
- Cooray, S. & Uscilowski, B. (2004), Effect of eye localization on the MPEG-7 face recognition descriptor, in 'VIIP'. Marbella, Spain, 6-8 Sept.
- Cootes, T., Cooper, D., Taylor, C. & Graham, J. (1995), 'Active shape models - their training and application', *Computer Vision and Image Understanding* **61**(1), 38–59.
- Cootes, T., Taylor, C., Cooper, D. & Graham, J. (1992), Training models of shape from sets of examples, in 'D.C Hogg and R.D. Boyle, editors, Proc. British Machine Vision Conference, Leeds, UK', pp. 9–18. Springer Verlag.
- Cordier, F., Seo, H. & Magnenat-Thalmann, N. (2003), 'Made-to-measure technologies for online clothing store', *IEEE Computer Graphics and Applications* **23**(1), 38–48.

- Curless, B. & Levoy, M. (1996), A volumetric method for building complex models from range images, in 'Proc. SIGGRAPH '96'.
- Curwen, R. & Blake, A. (1993), Dynamic contours: Real-time active splines, in 'Active Vision edited by A. Blake and A. Yuille', pp. pp 39–57. MIT Press, Cambridge, MA.
- Debevec, P., Taylor, C. & Malik, J. (1996), Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach, in 'Proc. ACM SIGGRAPH'.
- Dyer, C. (2001), 'Volumetric scene reconstruction from multiple views', *Foundations of Image Understanding*, Editor: L.S. Davis pp. 469–489. Kluwer, Boston.
- Faugeras, O. (1993), *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press.
- Faugeras, O. & Loung, Q.-T. (2001), *The Geometry of Multiple Images*, Cambridge University Press.
- Faugeras, O. & Maybank, S. (1990), Motion from point matches: Multiplicity of solutions, in 'rapport de Recherche de L'INRIA, RR1157'. 37 pages, published in *International Journal of computer Vision*, 4, 225-246.
- Foley, J., van Dam, A., Feiner, S. & Hughes, J. (1990), *Computer Graphics Principles and Practice*, Addison-Wesley publishing Company.
- Fua, P. & leclerc, Y. (1990), 'Model driven edge detection', *Machine Vision and Applications* 3, 45–56.
- Gleicher, M. & Ferrier, N. (2000), Evaluating video-based motion capture, in 'Proc. Computer Animation 2002', pp. 75 – 80.
- Gunn, S. & Nixon, M. (1994), A dual active contour for head boundary extraction, in 'Proc. IEE Colloquium on Image Processing for Biometric Measurement', Vol. 6, pp. 1–4.
- Gunn, S. & Nixon, M. (1995), 'Improving snake performance via a dual active contour', In V. Hlavac and R. Sara, editors, *Computer Analysis of Images and Patterns* 970, 600–605. Lecture Notes in Computer Science.
- Gunn, S. & Nixon, M. (1996), Snake head boundary extraction using global and local energy minimisation, in 'Proc. of the 13th International Conference on Pattern Recognition', pp. 581 – 585.
- Gunn, S. & Nixon, M. (1997), 'A robust snake implementation; a dual active contour', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(1), 63–68.
- Gutierrez, M., Vexo, F. & Thalmann, D. (2004), The mobile animator: Interactive character animation in collaborative virtual environments, in 'Proc. IEEE Virtual Reality 2004 conference', pp. 125–132.

- H-Anim (1997), *Information technology Computer graphics and image processing Humanoid animation (H-Anim)*, Humanoid Animation. ISO/IEC FCD 19774 Humanoid Animation Standard.
- Han, M. & Kanade, T. (2000), Creating 3D models with uncalibrated cameras, in 'Proc. Fifth IEEE Workshop on Applications of Computer Vision', pp. 178–185.
- Hartley, R. (1997), 'In defense of the eight-point algorithm', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(6), 580–593.
- Hartley, R. & Zisserman, A. (2000), *Multiple View Geometry in Computer Vision*, Cambridge University Press.
- Herda, L., Fua, P., Plankers, R., Boulic, R. & Thalmann, D. (2000), Skeleton-based motion capture for robust reconstruction of human motion, in 'Proc. Computer Animation 2000', pp. 77–83.
- Hilton, A. (2003), 'Computer vision for human modelling and analysis', *Journal of Machine Vision Applications* **14**(4), 206–209.
- Hilton, A., Beresford, D., Gentils, T., Smith, R. & Sun, W. (1999), Virtual people: Capturing human models to populate virtual worlds, in 'IEEE International Conference on Computer Animation', pp. 174–185.
- Hilton, A. & Fua, P. (2001), 'Modeling people toward vision-based understanding of a person's shape, appearance, and movement', *Computer Vision and Image Understanding* **81**(3), 227–230.
- Hilton, A., Gentils, T. & Beresford, D. (1998), Popup people: Capturing 3D articulated models of individual people, in 'Proc. In IEE Colloquium on Computer Vision for Virtual Human Modelling', pp. 1–6.
- Hilton, A. & Illingworth, J. (2000), 'Geometric fusion for a hand-held 3D sensor', *Machine Vision Applications* **12**(1), 44–51.
- Hutenlocher, D. & Ullman, S. (1990), 'Recognizing solid objects by alignment with an image', *International Journal of Computer Vision* **5**(2), 195–212.
- Iwasawa, S., Ohya, J., Takahashi, K., Sakaguchi, T. & Morishima, K. E. S. (2000), Human body postures from trinocular camera images, in 'Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition', pp. 326 – 331. Manchester, UK.
- Jacob, M., Blu, T. & Unser, M. (2004), 'Efficient energies and algorithms for parametric snakes', *IEEE Transactions on Image Processing* **13**(9), 1231–1244.
- Ju, X. & Siebert, P. (2001), Individualising human animation models, in 'Proc. Eurographics 2001'. Manchester, UK.
- Ju, X., Werghi, N. & Siebert, P. (2000), Automatic segmentation of 3D human body scans, in 'Proc. IASTED Int. conf. On Computer Graphics and Imaging 2000'. Las Vegas, USA.

- Kakadiaris, I. & Metaxas, D. (1998), 'Three-dimensional human body model acquisition from multiple views', *The International Journal of Computer Vision* **30**(3), 191–218.
- Kalra, N. & Magnenat-Thalmann, N. (1998), 'Real-time animation of virtual humans', *IEEE Computer Graphics and Applications* **18**(5), 42–56.
- Kalra, N., Thalmann, N., Moccozet, L., Sannier, G., Aubel, A. & Thalmann, D. (1998), 'Real-time animation of virtual humans', *IEEE Computer Graphics and Applications* **18**(5), 42–56.
- Karaolani, P., Sullivan, G. & Baker, K. (1992), Active contours using finite elements to control local scale, in 'Proc. British Machine Vision Conference, 1992', pp. 481, 488. Leeds, UK.
- Kass, M., Watkin, A. & Trezopoulos, D. (1987), 'Snakes: Active contour models', *International Journal of Computer Vision* **1**(4), 231–331.
- Kichenassamy, S., Kumar, A., P. Olver, A. T. & Yezzi, A. (1996), 'Conformal curvature flows: from phase transitions to active vision', *Arch. Rational Mech. Anal* **134**, 275–301.
- Kim, D. (1999), B-spline representation of active contours, in 'Proc. Fifth International Symposium on Signal Processing and Its Applications, 1999', Vol. 2, pp. 813–816. ISSPA '99.
- Klein, K., Malerczyk, C., Wiebesiek, T. & Wingbermhle, J. (2002), Creating a "personalised, immersive sports tv experience" via 3D reconstruction of moving athletes, in 'BAVR 2002 - Workshop on Business Applications of Virtual Reality', pp. 353–357. Poznan Poland.
- Klein, K. & Sequeira, V. (2000), View planning for the 3d modelling of real world scenes, in 'Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems'.
- Koivunen, V. & Bajcsy, R. (1995), Spline representations in 3-d vision, in 'in Springer-Verlag Lecture Notes on Computer Science', pp. 177–190.
- Kompatsiaris, I., Tzovaras, D. & Strintzis, M. (1998), 'C3D model-based segmentation of video-conference image sequences', *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5), 547–561.
- Kshirsagar, S., Garchery, S., Sannier, G. & Magnenat-Thalmann, N. (2003), 'Synthetic faces : Analysis and applications', *International Journal of Imaging Systems and Technology* **13**(1), 65–73.
- Kutulakos, K. & Seitz, S. (2000), 'A theory of shape by space carving', *International Journal of Computer Vision* **38**(3), 199218.
- Lai, K. (1994), Deformable Contours: Modelling, Extraction, Detection and Classification, PhD thesis, University of Wisconsin-Madison.
- Lai, K. & Chin, R. (1995), 'Deformable contours: modeling and extraction', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(11), 1084–1090.
- Lai, K. & Chin, R. (1998), 'On modelling, extraction, detection and classification of deformable contours from noisy images', *Image and Vision computing* **16**(1), 55–62.



- Lam, K.-M. & Yan, H. (1994), 'Fast greedy algorithm for active contours', *Electronic Letters* **30**(1), 21–23.
- Laurentini, A. (1994), 'The visual hull concept for silhouette-based image understanding', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(2), 150–162.
- Laurentini, A. (1997), 'How many 2D silhouettes does it take to reconstruct a 3D object?', *Computer Vision and Image Understanding* **67**(1), 81–87.
- Lee, W. (2002), Making one structured body, in 'Proc. Visual Systems and Multimedia, VSMM 2002', pp. 558–567. Lausanne, Switzerland.
- Lee, W., Goto, T. & Magnenat-Thalmann, N. (2000), Making H-Anim bodies, in 'Proc. Avatars2000'. Lausanne, Switzerland.
- Lee, W., Gu, J. & Magnenat-Thalmann, N. (2000), 'Generating animatable 3D virtual humans from photographs', *Eurographics2000* **19**(3), 1–10.
- Lee, W. & Magnenat-Thalmann, N. (2000), 'Fast head modeling for animation', *Journal Image and Vision Computing* **18**(4), 355–364.
- Leon, R. D. D. & Sucar, L. (2000), Human silhouette recognition with fourier descriptors, in 'Proc. 15th International Conference on Pattern Recognition 2000', Vol. 3, pp. 709–712.
- Lin, I.-C., Yeh, J.-S. & Ouhyoung, M. (2002), 'Extracting 3D facial animation parameters from multiview video clips', *IEEE Computer Graphics and Applications* **22**(6), 72–80.
- Longuet-Higgins, H. (1981), 'A computer algorithm for reconstructing a scene from two projections', *Nature* **293**, 133–135.
- Magnenat-Thalmann, N. & H.Seo (2004), Data-driven approaches to digital human modeling, in 'Proc. 2nd International Symposium on 3D Data Processing, Visualization, and Transmission'. Thessalonica, Greece.
- Malladi, R. & Sethian, J. A. (1996), Level set and fast marching methods in image processing and computer vision, in 'Proc. International Conference on Image Processing', Vol. 1, pp. 489–492.
- Malladi, R., Sethian, J. A. & Vemuri, B. C. (1995), 'Shape modelling with front propagation: a level set approach', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(2), 158–175.
- Martin, W. & Aggarwal, J. (1983), 'Volumetric description of objects from multiple views', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5**(2), 150 – 158.
- McInerney, T. & Trezopoulos, D. (1995), Topologically adaptive snakes, in 'Proc. Fifth International Conference on Computer Vision (ICCV'95)', pp. 840–845. Cambridge, MA, USA.
- McInerney, T. & Trezopoulos, D. (2000), 'T-snakes: Topology adaptive snakes', *Medical Image Analysis* **4**, 73–91.

- Meegama, R. & Rajapakse, J. (2003), 'NURBS snakes', *Journal of Image and Vision Computing* **21**, 551–563.
- Menet, S., Saint-Marc, P. & Medioni, G. (1990), B-snakes: implementation and application to stereo, in 'proc. Image Understand Workshop 1990', pp. 720 – 726.
- Mobahi, H., M.N.Ahmadabadi & Araabi, B. (2004), Fast initialization of active contours, in 'proc. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004', Vol. 1, pp. 546–551.
- Molloy, D. (2000), Active Meshes for Motion Tracking, PhD thesis, Dublin City University.
- Molloy, D. & Whelan, P. F. (2000), 'Active-meshes', *Patt. Recognition Letters* **21**, 1071–1080.
- Moore, P., Boyle, E. & Molloy, D. (2005), Automatic construction of 3d human models from only two orthographic projections, in 'Proc. Eurographics Workshop (Ireland)', pp. 33–40. Institute of Technology Blanchardstown (ITB), Dublin.
- MPEG4 (1998), *Coding of AudioVisual Objects: Systems*, ISO/IEC 14496-1 Final Draft International Standard, ISO/IEC JTC1/SC29/WG11 N2501.
- MPEG7 (2002), *Overview of the MPEG-7 Standard*, ISO/IEC JTC1/SC29/WG11, N4980.
- Neuenschwander, W., Fua, P., Székely, G. & Kübler, O. (1994), Initializing snakes, in 'Proceedings of the 1994 IEEE Computer Society on Computer Vision and Pattern Recognition', pp. 658–663. Seattle, USA.
- Nixon, M., Ng, L. & Gunn, D. B. S. (1997), Considerations on extended feature vectors in automatic face recognition systems, in 'Proc. 1997 IEEE International Conference on Man, and Cybernetics, 'Computational Cybernetics and Simulation'', Vol. 5, pp. 4075–4080.
- Noborio, H., Fukuda, S. & Arimoto, S. (1988), 'Construction of the octree approximating a three-dimensional object by using multiple views', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(6), 769–782.
- Papagiannakis, G., Schertenleib, S., Ponder, M., Arvalo, M., Thalmann, N. M. & Thalmann, D. (2004), Real-time virtual humans in AR sites, in 'Proc. IEE Visual Media Production(CVMP)', Vol. 3, pp. 273–276. London, UK.
- Piegl, L. & Tiller, W. (1997), *The NURBS Book, 2nd Edition*, Springer-Verlag.
- Plankers, R. & Fua, P. (2003), 'Articulated soft objects for multiview shape and motion capture', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(9), 1182–1187.
- Pollefeys, M., Koch, R., Vergawen, M., Deknuydt, B. & VanGool, L. (2000), 'Three-dimensional scene reconstruction from images', *Proceedings SPIE Electronic Imaging 2000* **1**(4). Three-Dimensional Image Capture and Applications III.
- Press, W., Teukolsky, S., Vetterling, W. & Flannery, B. (1992), *Numerical Recipes in C: The art of Scientific Computing, 2nd Edition*, Cambridge University Press.

- Prince, S., Cheok, A., Fabrizi, F., Williamson, T., Johnson, N., Billinghamurst, M. & Kato, H. (2002), 3D live: real time captured content for mixed reality, in 'Proc. International Symposium on Mixed and Augmented Reality'. (ISMAR).
- Qin, H. & Trezopoulos, D. (1996), 'D-NURBS a physical based framework for geometric design', *IEEE Transactions on Visualisations and Computer Graphics* **2**(1), 85–96.
- Roach, J. & Aggrawal, J. (1979), 'Computer tracking of objects moving in space', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1**(2), 127–135.
- Samadani, R. (1989), 'Changes in connectivity in active contour models', *In the Proceedings of the IEEE Workshop on Visual Motion* pp. 337–343.
- Seitz, S. & Dyer, C. (1999), 'Photorealistic scene reconstruction by voxel coloring', *International Journal of Computer Vision* **35**(2), 151–173.
- Seo, H. & Thalmann, N. M. (2004), 'An example-based approach to human body manipulation', *Graphical Models, Academic Press* **66**(1), 1–23.
- Sibiriyakov, A., Ju, X. & Nebel, J.-C. (2003), A new automated workflow for 3D character creation based on 3D scanned data, in 'In Proc. 2nd International Conference on Virtual Storytelling', pp. 155–162. Toulouse, France.
- Slabaugh, G., Culbertson, W., Malzbender, T. & Schafer, R. (2001), A survey of volumetric scene reconstruction methods from photographs, in 'In Proc. International Workshop on Volume Graphics', pp. 81–100.
- Slabaugh, G., Culbertson, W., Malzbender, T., Stevens, M. & Schafer, R. (2004), 'Methods for volumetric reconstruction of visual scenes', *International Journal of Computer Vision* **57**(3), 179–199.
- Sobreviela, E. J., Gutierrez, D., Gomez, F. & Seron, F. J. (2000), Assessment of an interactive 3D video experience aimed at large audiences, in 'In Proc. IADAT-micv2005', Vol. 3, pp. 151–155. Madrid, Spain.
- Sonka, M., Hlavac, V. & Boyle, R. (1999), *Image Processing, Analysis and Machine Vision*, PWS Publishing.
- Srivastava, S. & Ahuja, N. (1990), 'Note: Octree generation from object silhouettes in perspective views', *Computer Vision, Graphics and Image Processing* **49**, 68–84.
- Staib, L. & Duncan, J. (1992), 'Boundary finding with parametrically deformable models', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(11), 1061–1075.
- Szeliski, R. (1993), 'Rapid octree construction from image sequences', *CVGIP : Image Understanding* **58**(1), 23–32.
- Tang, H. & Zhuang, T. (1998), An improved adaptive b-spline active contour model, in 'Proc. 20th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society', Vol. 2, pp. 990–993.

- Trezopoulos, D. (1998), 'Regularization of inverse visual problems involving discontinuities', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(4), 413–424.
- Villa-Uriol, M., Kuester, F. & Bagherzadeh, N. (2003), Image-based avatar reconstruction, in 'Proc. Lake Tahoe Workshop on Collaborative Virtual Reality and Visualization 2003.'
- VRML (1997), *Virtual Reality Modeling Language*, ISO/IEC DIS 14772-1.
- Watt, A. (2000), *3D Computer Graphics, Third Edition*, Addison-Wesley.
- Weik, S., Wingbermhle, J. & Niem, W. (2000), Automatic creation of flexible antropomorphic models for 3D videoconferencing, in 'Proc. Computer Graphics International (OGI) 1998', pp. 520–527. Hanover Germany.
- Welch, G. & Foxlin, E. (2002), 'Motion tracking: No silver bullet, but a respectable arsenal', *IEEE Computer Graphics and Applications, special issue on Tracking* **22**(6), 24–38.
- Whelan, P. & Molloy, D. (2000), *Machine Vision Algorithms in Java: Techniques and Implementation*, Springer-Verlag, September.
- Williams, D. & Shah, M. (1992), 'A fast algorithm for active contours and curvature estimation', *CVGIP: Image Understanding* **55**(1), 14–26.
- Wingbermhle, J., Weik, S. & Kopernik, A. (1997), 'Highly realistic modelling of persons for 3D videoconferencing systems', *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing* pp. 286–291. Princeton, New Jersey, USA.
- Xu, C. & Prince, J. (1998a), 'Generalized gradient vector flow external forces for active contours', *Signal Processing - An International Journal* **71**(2), 131–139.
- Xu, C. & Prince, J. (1998b), 'Snakes, shapes, and gradient vector flow', *IEEE Transactions on Image Processing* **7**(3), 359–369.
- Xu, C., Yezzi, A. & Prince, J. (2000), On relationships between parametric and geometric active contours, in 'Proceedings of the 34th Asilomar Conference on Signals a, Systems and Computing', pp. 483–489.
- Zerroug, M. & Nevatia, R. (1995), Pose estimation of mulit-part curved objects, in 'Proc. SCV 1995 and nd IUW 1996', pp. 431–436.

# Splines and the Representing of Curves and Surfaces

## A.1 Introduction

This appendix provides a description of the existing methods for the representing of curves and surfaces. It also describes how the different curves are used in different applications. There is no single representation that is appropriate for all classes of shapes. Consequently, multiple representations have been used to describe different shapes efficiently (Koivunen & Bajcsy 1995). Unless the curves or surfaces being approximated are piecewise linear, large numbers of endpoint coordinates must be created and stored to achieve reasonable accuracy. In general, the idea is to use functions that are of higher degree than linear functions for representing (approximating) curves (Foley et al. 1990).

Any curve can be described as an array of points although this is not the most efficient means of storing the information about the curve and it is difficult to determine the exact shape of the curve and finding integral properties. Analytic equations when possible are generally used and provide the designer with better control over the shape and behaviour of the curves.

A polynomial function is an example of a function that is popular because they are convenient for computational purposes. Its general form is:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 = \sum_{i=0}^n a_i x^i \quad (\text{A.1})$$

where  $n$  is a nonnegative integer and  $a_0, a_1, \dots, a_n$  are real numbers. The polynomial  $p(x)$  is said to be of degree  $n$  if it has this representation and  $a_n \neq 0$ .

Additionally, curves can be expressed either explicitly<sup>1</sup> or implicitly<sup>2</sup>. These methods are suitable for representing certain types of curves and surfaces but they are not applicable for describing all types of curves. In particular, the use of explicit functions do not permit the multiple values of  $y$  for a single value of  $x$ . The functions are not rotationally invariant. Implicit representations can

<sup>1</sup>Explicit functions express  $y$  and  $z$  as functions of  $x$ , i.e.  $y = f(x)$  and  $z = g(z)$

<sup>2</sup>Implicit functions are expressed in the form  $f(x, y, z) = 0$

give rise to several solutions for a single value and additional constraints are needed to control the curve and these cannot be incorporated into the implicit equation (Koivunen & Bajcsy 1995, Foley et al. 1990).

## A.2 Parametric Curves

Curves are in general represented in parametric or implicit form. In the parametric form points on a 3D curve are defined by using three polynomials in a parameter,  $t$ , one for each of  $x$ ,  $y$  and  $z$ . The coefficients of the curve are selected such that the curve follows the desired path. Various degrees of polynomials can be used and at present the most practical are cubic polynomials although in certain applications it is necessary to use higher degree polynomials. Implicit representation of curves is not as computationally convenient a representation but it is sometimes required and there exist methods for converting from parametric representation to implicit representation (Anand 1993). The additional smoothness is achieved by approximating the curve by a piecewise polynomial curve instead of piecewise linear curve used in the implicit and explicit representations.

### A.2.1 Parametric Cubic Curves

Cubic polynomials are most frequently used because lower degree polynomials do not offer sufficient flexibility in controlling the shape of the curve, and higher-degree polynomials can introduce unwanted wiggles and also require more computation. No Lower-degree representation allows a curve segment to interpolate (pass through) two specified end points with specified derivatives at each endpoint.

A parametric cubic curve is defined as

$$P(t) = \sum_{i=0}^3 a_i t^i \quad (\text{A.2})$$

where  $P(t)$  is a point on the curve as shown in Figure A.1

Given a polynomial with its four coefficients, four known values are used to solve for the unknown coefficients. The four known coefficients may be the two end points and the derivative at the endpoints. This allows the creation of four independent equations that can be solved to find the correct solutions.

Parametric cubic polynomials that define a curve segment  $P(t) = [x(t), y(t), z(t)]$  are of the form:

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \end{aligned} \quad 0 \leq t \leq 1 \quad (\text{A.3})$$

To deal with finite segments of the curve, without loss of generality, we restrict the parameter  $t$  to the  $[0, 1]$  interval. This can be expressed in matrix format,  $P(t) = T.C$ , where  $C$  is a  $4 \times 3$  coefficient matrix. Another important reason for using parametric cubic polynomials is that they are the lowest-degree curves that are non planer in 3D. The derivative of  $P(t)$  is a parametric

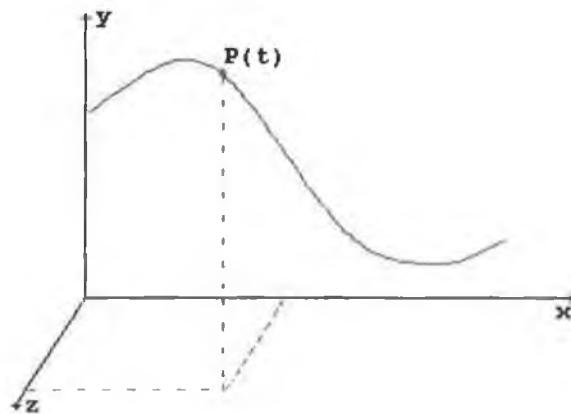


Figure A.1: point on a parametric cubic curve.

tangent vector of the curve.

### Continuity in Parametric Curves at the Joints

There are two type of continuity discussed in relation to parametric curves. The first is geometric continuity which exists if the directions of the two segments tangent vectors are equal at a joint point. For geometric continuity it is not necessary that the magnitudes of the tangent vectors are the same. If the tangent vectors of two vectors are equal in magnitude and direction a joint point the curves are said to have first-degree continuity,  $C^1$  in  $t$  and similarly if  $n$ th derivative are equal at a joint the curve is  $C^n$  continuous. In general,  $C^1$  implies geometric continuity  $G^1$  (Foley et al. 1990).

### Types of Parametric Curves

There are three main types of curves that are created using parametric cubic polynomials, these are:

- Hermite curves are defined using two endpoints and two endpoint tangent vectors. Each curve segment is defined from 0 to 1. This is to ensure that the end point correspond to the parametric variable  $t$ . This can be expressed compactly in matrix form.
- Splines are piecewise parametric representation of geometry with a specified level of parametric continuity. The cubic spline is represented by a piecewise cubic polynomial with second order derivative continuity at the common joints between segments and is defined by four control points. Splines have  $C^1$  and  $C^2$  continuity at the join points and come close to their control points but generally do not interpolate the points.
- Bézier curves are defined by two endpoints and two other points that control the endpoint tangent vectors. Unlike Hermite curves and splines which interpolate the control points Bézier curves provide an approximate curve that closely matches the control points.

Splines are long flexible strips of metal used by draftsmen to layout the surfaces of airplanes etc. They are particularly useful, if the focus is on describing of geometry of manufactured parts or free-form shapes (Koivunen & Bajcsy 1995). They have second order continuity. The mathematical equivalent of these strips, the natural cubic splines, is a  $C^0$ ,  $C^1$ ,  $C^2$  continuous cubic polynomial that interpolates (passes through) the control points. This is one more degree of continuity than in either Bézier or Hermite curves thus splines are smoother than either of these.

If the segments of cubic spline are parameterised separately, so that the parameter  $t$  varies between 0 and 1 for all segments. This is termed a normalised cubic spline and is in fact a particular case of Hermite interpolation.

### A.3 B-Splines Curves

Splines are ideal for defining the shape of an object because unless they are severely stressed they can maintain second order continuity. Splines can be defined mathematically as continuous cubic polynomials that interpolate a number of control points. The polynomial coefficients for natural splines are dependent on all  $n$  control points. Thus changing the position of one of the control points affects the entire curve and involves operating on an  $n + 1$  by  $n + 1$  matrix (Foley et al. 1990, Piegl & Tiller 1997, Koivunen & Bajcsy 1995). An alternative to the general spline representation is that of B-splines which consist of curve segments that are only dependent on a few of the control points. This offers an enhancement over the use of splines as the movement of an individual point does not require the modification of the complete curve and thus greater local control is achieved over how a contour deforms and moving a control point affects only a small portion of the curve. In addition to this it makes the B-spline contours suitable for real-time applications (Blake & Isard 1998). B-splines have the same continuity as natural splines, but do not interpolate the control points. The two continuity conditions on a segment come from the adjacent segments. This is achieved by sharing control points between segments.

When using splines to describe shapes it is possible to increase the polynomial order  $d$  but it is preferable to increase the number of spans (segments) used. Usually the polynomial order is fixed at quadratic ( $d = 3$ ) or cubic ( $d = 4$ ). This is important and leads to computational stability. B-splines are essentially piecewise polynomial curves. In Figure A.2 a curve  $C(u)$  consists of  $m(= 3)$   $n$ th-degree polynomial segments.  $C(u)$  is defined on  $u \in [0, 1]$ . The parameter values are called break points.

B-splines are constructed as a weighted sum of  $N_B$  basis functions  $B_n(s)$ ,  $n = 0, 1, \dots, N_B - 1$ . In the simplest case each basis function consists of  $d$  polynomials each defined over a span of the  $s$  axis. Spans can be taken to have unit length and they are joined at knots see Figure A.3.

The constructed spline function is of the form:

$$\sum_{n=0}^{N_B-1} x_n B_n(s) \quad (\text{A.4})$$

where  $x_n$  are the weights applied to the respective basis functions  $B_n(s)$ . This can be expressed compactly in matrix notation as  $x(s) = \mathbf{B}(s)^T \mathbf{Q}^x$



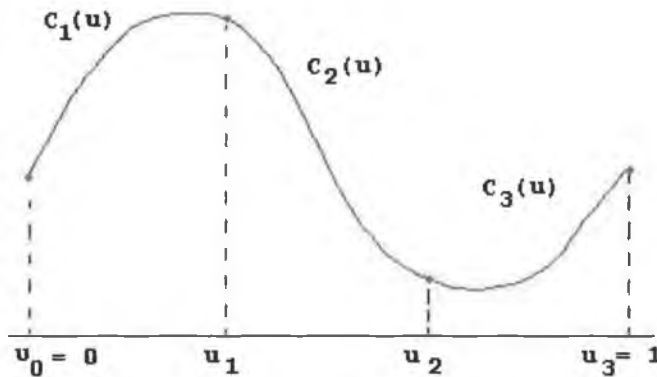


Figure A.2: Piecewise polynomial curve with three segments.

Where  $\mathbf{B}(s) = (B_0(s), B_1(s), \dots, B_{NB-1}(s))^T$  and vector weights

$$\mathbf{Q}_x = \begin{bmatrix} x_n \\ \vdots \\ x_{NB-1} \end{bmatrix} \quad (\text{A.5})$$

By convention B-spline basis functions are constructed in such a way that they sum to 1 at all points:

$$\sum_{n=0}^{N_i-1} B_n(s) = 1 \text{ for all } s \quad (\text{A.6})$$

In the simple case of a quadratic B-spline with knots spaced regularly at unit intervals the first B-spline basis function has the form

$$B_0(s) = \begin{cases} \frac{s^2}{2} & \text{if } 0 \leq s \leq 1 \\ \frac{3}{4} - (s - \frac{3}{2})^2 & \text{if } 1 \leq s \leq 2 \\ \frac{(s-3)^2}{2} & \text{if } 2 \leq s \leq 3 \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.7})$$

and the others are simply translated copies  $B_n(s) = B_0(s - n)$  (Blake & Isard 1998).

### Uniform Periodic B-Splines

A uniform periodic B-spline is the simplest form of the B-spline curve. In general it provides a good approximation of the curve based on the control points. A uniform cubic B-spline,  $N_i(t)$  is a cubic  $C^2$  basis function as shown in Figure A.3.

The parametric intervals or knots,  $t$ , within which the basis function is defined are equal. Thus the name uniform B-splines. The knots form a vector of real numbers called the knot vector, in non decreasing order. The function is centred at  $t_i + 2$  and has a zero value  $t < t_i$  and  $t > t_{i+4}$ . The non zero part of the function is composed of four polynomials,  $N_{0,3}$ ,  $N_{1,3}$ ,  $N_{2,3}$ ,  $N_{3,3}$ . A

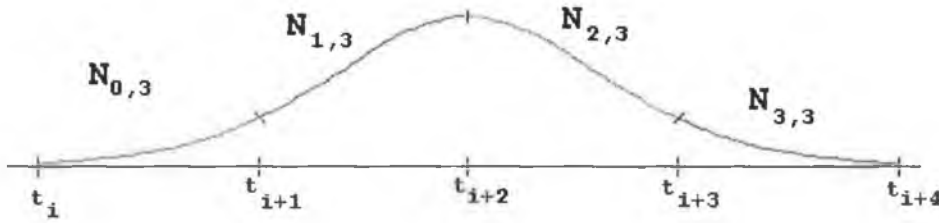


Figure A.3: A uniform cubic B-spline, basis functions  $N_i(t)$ .

B-spline curve is obtained by multiplying this approximation function by a subset of four control points in the vicinity of the curve and can be represented by the following equation:

$$P_i(t) = N_{0,3}V_i + N_{1,3}V_{i+1} + N_{2,3}V_{i+2} + N_{3,3}V_{i+3} \quad (\text{A.8})$$

where  $V_i$  are the control points defining the B-spline curve.

For B-splines the control points are not actually interpolated but approximated and thus the curve does not always pass through the individual control points, see Figure A.4. This can be forced by changing the continuity at the control points and gives a situation where the control points are doubled at each point and then the B-spline curve passes through each point. Although like the other curves the possibility to reuse the same blending function at each interval ensures that the curves can be rapidly calculated (Piegl & Tiller 1997).

Each of the four cubic polynomials has the form

$$N_{j,3} = a_j + b_j t + c_j t^2 + d_j t^3 \quad (\text{A.9})$$

This results in sixteen unknowns that are solved using equations that are generated using the fact the derivatives exist and imposing continuity constraints at the joint points. This results in the following B-spline basis functions.

$$\begin{aligned} N_{0,3} &= \frac{1}{6}t^3 \\ N_{1,3} &= \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ N_{2,3} &= \frac{1}{6}(3t^3 - 6t^2 + 4) \\ N_{3,3} &= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1) \end{aligned} \quad (\text{A.10})$$

### General B-spline Representation

The constructed spline function is

$$P(t) = \sum_{i=0}^{N_n} N_{i,k}V_i \quad (\text{A.11})$$

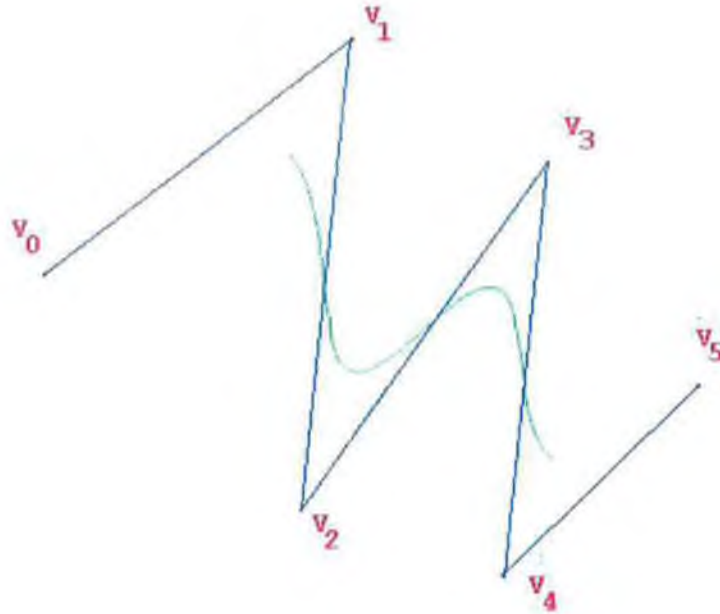


Figure A.4: A uniform cubic B-spline shown with control points and control polygon.

where  $V_i$  is the set of control points and  $N_{i,k}$  represents the appropriate blending functions of degree  $(k - 1)$ . A spline is said to be of order  $k$  or degree  $(k - 1)$  when defined mathematically as a piecewise  $(k - 1)$ st degree polynomial that is  $C^{k-2}$  continuous. In other words, the degree of the polynomial does not exceed  $(k - 1)$  inside each  $[t_i, t_{i+1}]$  interval and the position and  $[1 \text{ to } (k - 2)]$  derivatives are continuous.

Thus for the case of the cubic B-spline  $k = 4$ , the degree  $= (k - 1) = 3$  and second degree continuity is satisfied. In general the  $i^{\text{th}}$  blending function  $N_{i,k}(t)$  is defined by the following recursive equation

$$N_{i,1}(t) = \begin{cases} 1 & \text{for } t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.12})$$

$$N_{i,k}(t) = \frac{(t - t_i)}{(t_{i+k-1} - t_i)} N_{i,k-1}(t) + \frac{(t_{i+k} - t)}{(t_{i+k} - t_{i+1})} N_{i+1,k-1}(t)$$

where the knot vector is  $[t_i, \dots, t_{i+k}]$ . The only constraint on the knot vector is that it must be non-descending, i.e the values of the elements of  $t_i$  must satisfy the relationship  $t_i \leq t_{i+1}$  and the same value should not appear more than  $k$  times (higher than the order of the spline).

In any B-Spline curve, the degree  $(k - 1)$ , the number of control points and the number of

knots are related to each other by the following formula

$$\begin{array}{ccccccc}
 (m + 1) & = & (n + 1) & + & k & & \\
 \downarrow & & \downarrow & & \downarrow & & \\
 \text{no. knots} & & \text{no. control points} & & \text{order of curve} & & \text{(A.13)}
 \end{array}$$

or

$$m = n + k \quad \text{(A.14)}$$

the knot vector is therefore  $[t_0, \dots, t_{n+k}]$  and there are three types of knot vectors

### Periodic/Uniform B-Splines

These B-splines are characterised by equal spaced knot values. This equal spacing is implicit in the definition in Equation A.10. Uniform knot vectors are all periodic and each segment of the B-spline function is a translated copy of the first. The uniform B-spline curves do not interpolate the control points and if the curve is open ended then the uniform B-spline curves do not interpolate the first and last points except in the linear case.

### Non-Periodic B-Splines

A non-periodic basis on a finite interval is more complex than the periodic basis and permits the inclusion of multiple knots<sup>3</sup> at its ends to reduce the continuity at the endpoints and force the B-Spline curve to interpolated the endpoints. Each knot that coincides reduces the continuity at a point by one degree. When the continuity decreases to  $C^0$  discontinuities are introduced in (Blake & Isard 1998).

In (Anand 1993) the non-periodic knot vector has repeated knot values at its ends with multiplicity equal to the order of the parametric function thus with a control polygon consisting of four control points the knot vector has the following form:

order	No. of knots	Nonperiodic	
k	$m = n + k$	knot vector	
2	6	[001233]	(A.15)
3	7	[0001222]	
4	8	[0000111]	

and in general, these conditions must be satisfied for a knot  $t_i$  in a non-periodic knot vector starting at  $t_0$ :

$$\begin{aligned}
 t_i = 0 &\rightarrow i < k \\
 t_i = i - k + 1 &\rightarrow k \leq i \leq n \\
 t_i = n - k + 2 &\rightarrow i > n
 \end{aligned} \quad \text{(A.16)}$$

<sup>3</sup>A multiple knot is classified as two knots that approach one another or coincide

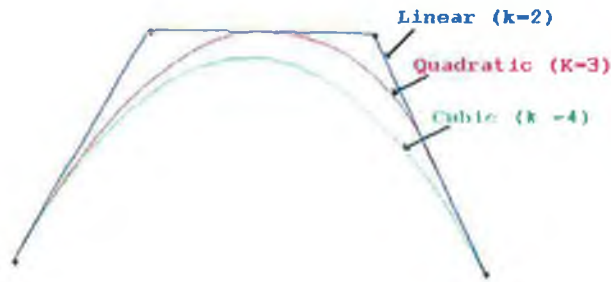


Figure A.5: Nonperiodic B-splines with varying degrees: from linear to cubic.

### Non-Uniform B-splines

A non-uniform knot sequence can be obtained by introducing multiple knots at interior knot values and thus reducing the continuity. Additionally, the spacing between the knots may be unequal, for example the knot sequence  $[0 \ 1 \ 2 \ 4 \ 5]$ . Non-uniform spacing has certain advantages over uniform spacing, these include:

- greater control over the shape of the curve
- overcomes the possible oscillations that may occur in uniform B-Spline curves as a result of unevenly spaced control points

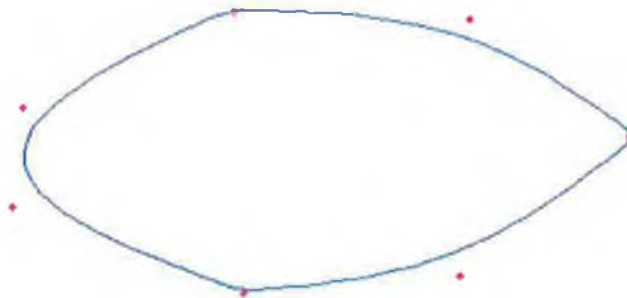


Figure A.6: Nonuniform cubic B-spline with knot vector  $[0, 1, 1, 2, 3, 3, 4, 5, 5, 6]$ .

### Non-Uniform Rational B-Splines (NURBS)

NURBS have become one of the most important geometric elements in design and are standard within most cad and animation packages (Koivunen & Bajcsy 1995, Qin & Trezopoulos 1996). NURBS include Bézier and B-Splines as special cases. NURBS are characterised by the introduction of weights in the non-uniform representation of B-spline. This is the most general form

of B-spline curve and provides the greatest variety of shapes and has the potential to pull the B-spline curve close to or push away from the control points by the introduction of multiple knots, non uniform spacing of the knots or the use of the weights.

$$v(s, t) = \frac{\sum_{i=0}^{n-1} w_i^t N_{i,k}(s) \mathbf{P}_i^t}{\sum_{i=0}^{n-1} w_i^t N_{i,k}(s)} \quad (\text{A.17})$$

$$= \sum_{i=0}^{n-1} R_i^t(s) \mathbf{P}_i^t \quad (\text{A.18})$$

where

$$R_i^t(s) = \frac{w_i^t N_{i,k}(s) \mathbf{P}_i^t}{\sum_{j=0}^{n-1} w_j^t N_{j,k}(s)} \quad (\text{A.19})$$

The weights are incorporated in the B-Spline representation by using homogeneous coordinates  $(x_i, y_i, z_i), w_i$

### A.3.1 B-Spline Surfaces

It is often desirable to represent shapes using surfaces. B-Spline curves are a desirable means of representing surfaces because they provide high quality surface approximation and in addition to the continuity property they process local control so changes are limited to a small region which makes them suitable in animation applications (Koivunen & Bajcsy 1995). The surfaces lie within the convex hull of the control point mesh.

The 3D B-spline surface is generated by considering the control points as a bidirectional web of control points. This is expressed as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} \quad (\text{A.20})$$

where  $\mathbf{P}_{i,j}$  are the 3D control points and  $U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}$  and  $V = \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}$  are two knot vectors that determine how closely the B-spline curve approximates the control points.  $U$  has  $p + r$  knots and  $V$  has  $s + q$  knots.

### A.3.2 NURBS Surfaces

The NURBS surface is generated by considering the control points as a bidirectional web of control points. This is expressed as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{B}_{i,j}^h \quad (\text{A.21})$$

where  $\mathbf{B}_{i,j}^h$  are the homogeneous coordinates  $(x_{i,j}, y_{i,j}, z_{i,j}, h_{i,j})$  of the control points and  $U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}$  and  $V = \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}$  are two knot vectors that determine how closely the B-spline curve approximates the control points.  $U$  has  $p + r$  knots and  $V$  has  $s + q$  knots. The basis functions are defined recursively and the spacing of the

knots can be non-uniform and to introduce discontinuities multiple knots have to be introduced in the same location.

**Polygon Meshes:** a polygon mesh is a set of connected polygon bounded planar surfaces. Polygon meshes can easily represent open boxes, cabinets and building exteriors.

Some properties of B-Spline and NURBS surfaces include:

- if  $U = [\overbrace{0, \dots, 0}^{p+1}, \overbrace{1, \dots, 1}^{p+1}]$  and if  $V = [\overbrace{0, \dots, 0}^{q+1}, \overbrace{1, \dots, 1}^{q+1}]$  then the surface interpolates the four corners.
- Affine invariance: is maintained if the affine transform is applied to the control points
- the surface is completely contained within the convex hull of the control points
- if a control point  $B_{i,j}^h$  is moved it affects the part of the surface bounded by the rectangle  $[u_i, u_{i+p+1}) \times [v_i, v_{i+q+1})$

## A.4 Lines, Planes and Intersections in 3-Space

This section contains a description of geometric entities that are used for representing lines and planes in 3D. The equations in this section are based on those in (Anton 1994).

### Parametric Equation of a Line

If a line,  $l$ , in 3-space passes through the point  $P_o = (x_o, y_o, z_o)$  and parallel to the non-zero vector  $\mathbf{v} = (a, b, c)$  then  $l$  consists precisely of those points  $P(x, y, z)$  for which the vector  $\vec{P_oP}$  is parallel to  $\mathbf{v}$  i.e for the scalar  $t$  such that

$$\vec{P_oP} = t\mathbf{v} \quad (\text{A.22})$$

This can be rewritten in parametric form as:

$$x = x_o + ta, \quad y = y_o + tb, \quad z = z_o + tc \quad (-\infty < t < +\infty) \quad (\text{A.23})$$

### Parametric Equation of a plane

The equation of a plane is calculated in one of two ways. Firstly, if to find the equation of a plane passing through a point  $P_o = (x_o, y_o, z_o)$  with a non-zero vector  $\mathbf{n} = (a, b, c)$  as its normal. Then the plane consists precisely of those points  $P(x, y, z)$  for which the vector  $\vec{P_oP}$  is orthogonal to  $\mathbf{n}$  i.e

$$\mathbf{n} \cdot \vec{P_oP} = 0 \quad (\text{A.24})$$

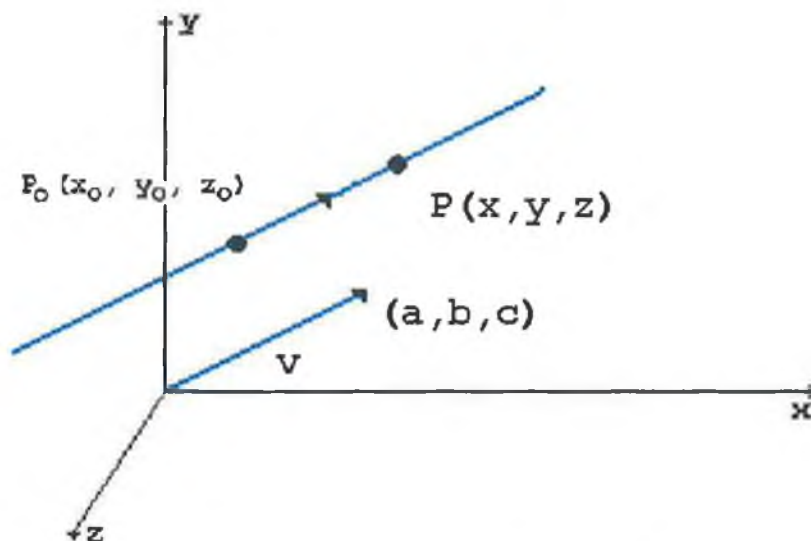


Figure A.7: Illustration of the equation of the line parallel to vector  $v$ .

where  $\cdot$  represents the dot product<sup>4</sup> of two vectors.

Since  $P_0P = (x - x_0, y - y_0, z - z_0)$ , Equation A.24 can be written as

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \quad (\text{A.25})$$

This is called the point normal form of the equation of a plane and is illustrated in Figure A.8.

A second method for calculating the equation of the plane when the coordinates of three points are known is as follows. Given three points  $P_0 = (P_{0x}, P_{0y}, P_{0z})$ ,  $P_1 = (P_{1x}, P_{1y}, P_{1z})$ , and  $P_2 = (P_{2x}, P_{2y}, P_{2z})$ . The three points lie in a plane and the vectors  $P_0P_1$  and  $P_0P_2$  are parallel to the plane. Thus  $P_0P_1 \times P_0P_2 = (n_x, n_y, n_z)$  is normal to the plane as it is perpendicular to both  $P_0P_1$  and  $P_0P_2$ , where  $\times$  is the cross product<sup>5</sup>. This defines the normal to the plane at point  $P_0$  and the equation is formed as

$$n_x(x - P_{0x}) + n_y(y - P_{0y}) + n_z(z - P_{0z}) = 0 \quad (\text{A.26})$$

### Generation of the Vector Normal to a Plane

The normal vector is a vector orthogonal to a point, a line or a plane. Two vectors are considered to be orthogonal if their dot product is zero. In computer Graphics a normal vector can be used to define

<sup>4</sup>The dot product or inner product of two vectors is expressed mathematically as

$$\mathbf{u} \cdot \mathbf{v} = \begin{cases} \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta \\ 0 \end{cases}$$

<sup>5</sup>if  $\mathbf{u} = (u_0, u_1, u_2)$  and  $\mathbf{v} = (v_0, v_1, v_2)$  are vectors in 3-space, then the cross product  $\mathbf{u} \times \mathbf{v}$  is the vector defined by  $\mathbf{u} \times \mathbf{v} = (u_2v_3 - u_3v_2, u_3v_1 - u_1v_3, u_1v_2 - u_2v_1)$



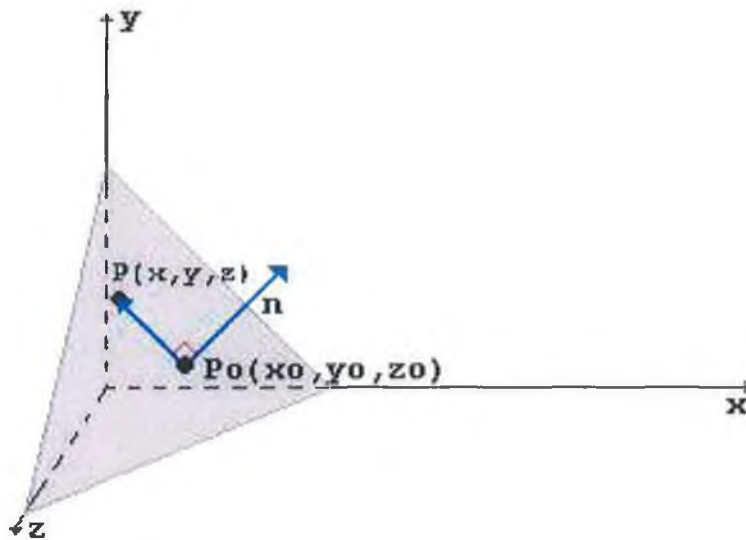


Figure A.8: Illustration of the point normal form of the equation of a plane.

how a surface responds to lighting. The amount of light reflected by a surface is proportional to the angle between the lights direction and the normal.

To calculate a normal for a plane or surface defined by three points,  $t_1, t_2, t_3$ . It is then possible to define two vectors  $v_1 = t_2 - t_1$  and  $v_2 = t_3 - t_1$ . See Figure A.9(a). With these two vectors, it is possible to compute the cross product between them to find a perpendicular vector to the face. The resulting vector is then normalised

Normalisation is calculated by first calculating the length of the vector and dividing each component of the vector by the vectors length.

This results in the normal vector  $n$  which is defined as

$$n = (n_x, n_y, n_z) \quad (\text{A.27})$$

where

$$n_x = c_x/L \quad (\text{A.28})$$

$$n_y = c_y/L \quad (\text{A.29})$$

$$n_z = c_z/L \quad (\text{A.30})$$

where  $c$  is the vector resulting from the cross product of vectors  $v_1$  and  $v_2$  and  $L$  is the length of the vector, defined as

$$L = \sqrt{c_x \times c_x + c_y \times c_y + c_z \times c_z} \quad (\text{A.31})$$

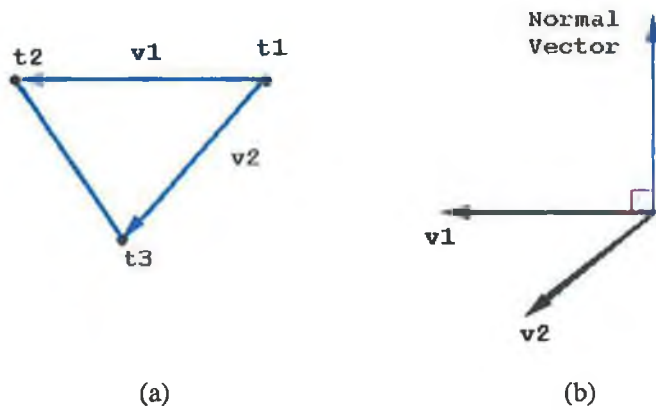


Figure A.9: (a) the vectors used to calculate the cross product for a triangle face and (b) the vector normal that results from the cross product of the vectors  $v_1$  and  $v_2$ .

#### The Vector Normal to a Point or Vertex

A problem with assigning a normal per face is that the surfaces looks faceted, i.e. the brightness of each face is constant, and there is a clear difference between faces with different orientations. Computing a normal per vertex, and not per face provides a smoother surface. An example of this is shown in Figure A.10. Each vertex (excluding the corner and border vertices), is shared by four triangle faces. The normal at a vertex should be computed as the normalised sum of all the unit length normal for each face the vertex shares. This is expressed mathematically

$$v = \text{normalised}(\text{sum}(v_{12} + v_{23} + v_{34} + v_{41})) \quad (\text{A.32})$$

where  $v_{ij}$  is the normalised cross product of  $v_i$  and  $v_j$

#### Distance from a point to a Plane

The distance  $D$  between a point  $P_o(x_o, y_o, z_o)$  and the plane  $ax + by + cz + d = 0$  is

$$D = \frac{|ax_o + by_o + cz_o + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (\text{A.33})$$

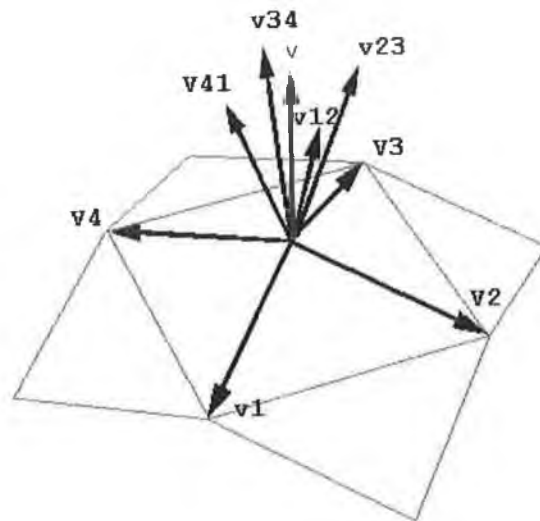


Figure A.10: Illustration of the calculation of the normal per vertex