D. Molloy and P.F. Whelan (1999), "Active-Mesh Self-Initialisation", Proc. Irish Machine Vision and Image Processing Conference 1999, Dublin City University, September 8-9, pp 116-130.

Active-Mesh Self-Initialisation

Derek Molloy, Paul F. Whelan

Vision Systems Laboratory, Dublin City University,
Glasnevin, Dublin 9, Ireland.
{molloyd, whelanp}@eeng.dcu.ie

Abstract. This paper describes a method of initialising an Active-Mesh that is to be automatically created and configured directly from a single frame of an image sequence. The aim of this approach is to use the derived mesh to perform visual tracking in unconstrained motion environments, allowing movement of the camera, the scene and even the inclusion of background-independent moving objects. The problem in initialising this mesh comes from the fact that there is no *a priori* information about the scene available. The paper will discuss methods that are currently available for determining the initial position of active contour models within images, then suggesting a method of initialising an active mesh.

1. Introduction

Active Contour Models are a popular method for tracking 'regions' as features through image sequences. Developed largely by Kass *et al* (1), *snakes* are active contour models, using an energy-minimising spline that can help solve numerous computer vision problems, such as the analysis of dynamic image data, image segmentation and image understanding. The model is described as active since it is always attempting to minimise its energy function, hence showing dynamic behaviour. Snakes are an example of the generalised technique of matching a deformable model to an image using energy minimisation techniques. The shape of the contour determines its internal energy and its external energy is determined by the spatial location of the contour within the image. External forces may be used to attract these contours towards or away from salient image features, such as edges, lines and corners.

There are several models of deformable contours that may be used, such as the snake model suggested by Kass *et al* (1) that 'wraps around' image features, or the balloon model introduced by Cohen (2) that expands to locate desired image features. Staib and Duncan (3) deal with other methods such as elliptic Fourier decomposition for objects with shape irregularities. They use a Fourier shape model that represents a closed boundary as a sum of trigonometric functions of various frequencies. They then use an iterative energy minimisation technique to fit the model within the image. This technique is limited to closed boundaries and does not always provide an appropriate basis for capturing shape variability. Some of these methods deal with different problems, for example, Kass *et al* (1) deal only with local deformations while Staib and Duncan (3) deal only with global deformations, such as those that might be described by scaling, rotation, stretching or dilation of a contour (rigid motion). However, local deformations might be caused by some higher level complex motion, such as the movement of human lips, living cell deformation and other deformations related to the shape of the feature and not just its location in space. There are indeed two separate problems to be examined, global deformations of rigid objects are too varied to be described adequately by single shape attributes such as bending energy or elasticity, while these descriptions may be adequate for local deformations in deformable objects.

2. Formulisation

Active Contour Models were proposed initially as an approach to the ill-posed edge detection problem. It was proposed that the low-level process of edge detection should provide sets of possible alternate solutions for the higher-level process of edge linking, rather than forcing forward a single unique solution. An energy minimisation framework was developed as a solution, designing energy functions with local minima that provide alternate solutions for the higher-level edge linking, resulting in an active model that minimises to the desired solution when placed spatially near that solution.

The structure of a snake is an ordered set of control points (or snaxels [1]) of the form $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ where each snaxel $\mathbf{v}_i \in \mathbf{I} = \{i_1(x,y), i_2(x,y), \ldots, i_n(x,y)\}$ and $x, y = 1, 2, \ldots, M$, allowing every snaxel to have a 2-D coordinate position on the image plane. Alterations to the location or shape of the snake are possible by moving the positions of the individual snaxels. The number of snaxels is chosen by selecting an appropriate internal distance h. This value is chosen on an application specific basis, where the coarsness of the fit of the snake to the object is the defining factor. The smaller the value of h, the greater the number of snaxels that are required and the more tightly defined the minimised snake will be to the desired contour.

A snake can either be open or closed, with a closed snake having the end points connected, so $\mathbf{v}_n$ is connected to $\mathbf{v}_1$. Allowing the snake to be open leads to difficulties in determining the desired energy at the first and last snaxels. Sometimes however, for specific vision applications, it may be necessary to require the end points to remain at specific predefined spatial locations.

[1] For the rest of this paper ‘snaxel’ will be used to refer to such points or elements of the snake. The term is derived from a contraction of the term “snake elements”.

3. Initialisation

Many methods for developing active contour models have emerged in recent years, since the introduction of snakes by Kass *et al* (1), with many applications including Staib and Duncan (3) and Leymarie and Levine (4). However, in most of these applications it is assumed that the initial position of the snake is relatively close to the desired solution, in fact often initialised by a human operator. While this might be a suitable assumption, on a frame by frame basis in the motion tracking problem, it is not always acceptable when initialising the active contour.

The initialisation of the snake is a difficult problem that has a significant impact on the outcome of the snake minimisation. If the snakes initial position is far from the desired solution it is quite common for the snake to become trapped in local energy minima, due to irrelevant edge information or noise. The snake is also limited in spatial movement since the snakes own potential energy prevents the snake from moving far from its current position (Neuenschwander *et al* (5)). Many methods require the user or other mechanisims to place the initial snaxels near the desired boundaries of the object to be tracked. If the snake is placed close to an intended contour, its energy minimisation will force the correct solution. Snakes do not attempt to solve the problem of detecting prominent image contours, but rely on other methods to place the snake near the desired contour. Some of these methods include: (i) The Hough transform is a common method for the extraction of the initial estimates of the contour position for *rigid* objects. These rigid templates cannot account for deformations that may occur, thus a rigid template chosen *a priori* cannot produce satisfactory results in all cases. Lai (6) shows that performance actually degrades with deformation, so using the generalised Hough transform to provide the initial contours when substantial prior knowledge is available. (ii) Short snakes may be initialised at strong edges and allowed to expand and even overlap until the entire boundary is covered. (iii) If enough computational power is available thousands of randomly initialised snakes may be placed on the image, until a suitable solution is found, however this is rarely practical.

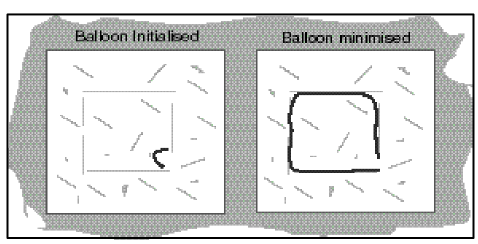

Fig. 1. The Balloon Active Contour Model (Cohen and Cohen (1992)).

Cohen and Cohen (8) suggested another approach to the energy minimisation problem (see Fig. 1.) based on the Galerkin solution of the Finite Element Method. This approach is applied to the closed contour case and finds remarkably good stability. An additional pressure force is added to the contour, in which they consider the contour as a balloon. The balloon is inflated from the inside and expands, overcoming isolated valleys and noise giving better results than possible with snakes in particular cases. This approach allows a less accurately defined initial position to expand to the correct shape, without becoming trapped on local discontinuities; however, it cannot be used in all applications.

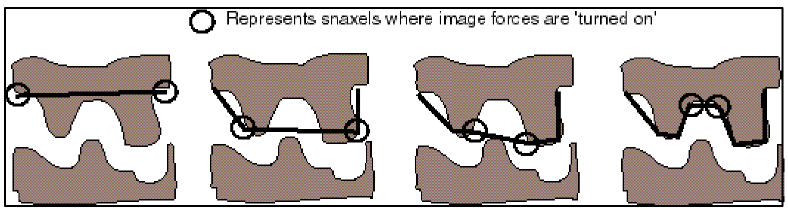

Fig. 2. The method of Neuenschwander *et al* (5) begins by placing the snaxels at each end of the snake, as close to the desired object as possible, then allowing each successive pair of snaxels to converge in an ordered fashion towards the center snaxel of the snake. When the two active snaxels meet at the same snaxel location the optimisation method is complete. All snaxels are frozen in place except for the two active snaxels.

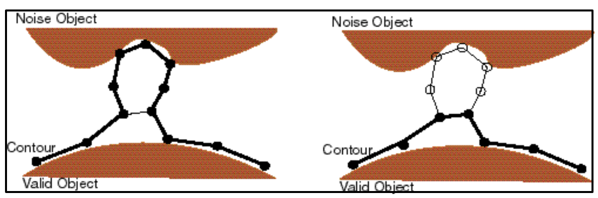

Fig. 3. An example algorithm for noise reduction in the snake.

Hashimoto *et al* (9), suggest a quick technique for the reduction of noise in the snake structure, by 'cutting off' sections of the snake that are not useful to the snake. In Fig.3, we see that when non-neighbour snaxels are within a small pre-determined distance we have a method of determining a noisy condition. By removing the snaxels that are between these two snaxels we are removing noise from the snake, hopefully allowing the snake to converge more closely. This removes noise from the initialis ation of the initial snake description.

4. IMPLEMENTATION

The method presented here initialises the mesh using feature points extracted from the initial image frame to provide the initial node locations of the mesh. Not only is it necessary to place the snake in its initial position, the energy equations must be initialised such that the mesh is created in equilibrium. A fundamental stage in computer vision is the generation of descriptions of images, more useful than a large set of pixels. The main aim of this feature extraction is to reduce this set of pixels to a list of features that are distinct from surrounding portions of the image so that the information available in the scene becomes more manageable. For example, in the case of feature matching, the distinctiveness of these points limits the potential matches in the following frames. These points more than likely correspond to significant features in the real-world scene, such as ‘real corners’, ‘real boundaries or edges’ or textured areas. Most ‘image segmentation’ techniques are based on the search for local discontinuities or on the detection of regions in the image with homogeneous properties.

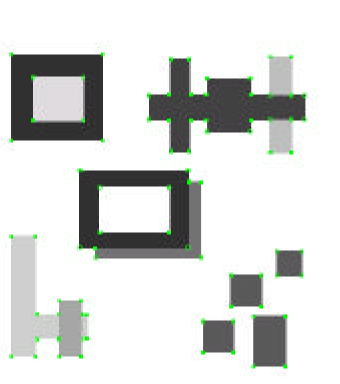

Fig. 4. A Standard Template used for the test initialisation, showing the corners clearly detected by the SUSAN algorithm.

The trajectories derived from locating particular feature points on an object, through time, are popular because they are relatively simple to extract. The generation of motion trajectories from a sequence of images typically involves the detection of tokens in each frame, and the correspondence of such tokens from one frame to another. These tokens need to be distinct enough for detection and stable enough to be found in each frame. Tokens may include edges, corners, interest points, and regions.

The Smallest Univalue Segment Assimilating Nucleus (SUSAN) corner detector as introduced by Smith (7) is used with varying thresholds to provide feature points that are suitable, such that the extracted features are: (i) Consistent; in that features to be used as tokens for motion analysis must be detected consistently through image frames, if they are to be used as the basis for subsequent higher level processing. (ii)

Accurate, in that features must be located precisely from frame-to-frame. (iii) Non-complex, in that computational speed is a very important issue, where this primary stage of corner detection must be performed at each iteration of the algorithm.

In Fig. 5, the SUSAN corner detection algorithm is applied to the same image with differing thresholds to show how this has an effect on the number of corners detected. In (b) the standard values are used of a threshold value of 16 and a distance value of 16. In this case 570 corners are detected. When these values are reduced to 8 and 8 the number of corners increases to 930 as in (c) and when this value is increased to 32 the number of corners reduces down to 305 as in (a). The general quality of the corners detected at lower thresholds is poorer as the intensity difference that is required to classify as a corner is reduced, becoming more affected by lighting conditions and noise. The determination of these values for the brightness and distance thresholds for the active mesh can be related directly to the determination of the internal spacing value, *h* for active contours, as discussed previously.

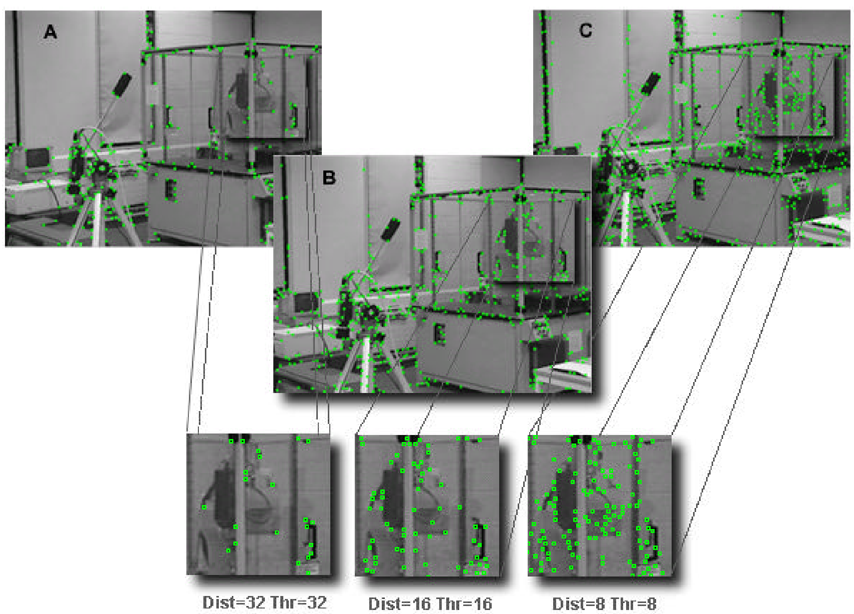

Fig. 5. The SUSAN algorithm applied to the same image using different threshold values. In (a) Distance = 32, Thr = 32, (b) Dist = 16, Thr = 16, and in (c) Dist = 8, Thr = 8.

The corners that are detected with the highest difference are the strongest, i.e. they have a large gradient on the corner and so are possibly the best corners for matching, in that with the exception of occlusion they are likely to be consistent from one frame to the next.

The system should provide a reasonable number of corners for feature matching. The larger the number of tokens, the more computationally intensive the matching becomes, and the weaker the tokens become, in terms of consistently and reliably detectable tokens. Corners detected from texture information, or image quantisation effects are likely to be unreliable. An insufficient number of corners will cause regions in the image to contain no tokens for the feature-matching algorithm, resulting in sparse motion information. However, having no tokens detected in a region of uniform intensity may not be too much of a problem, as the features surrounding this area will allow an estimation to propagate towards the centre of the uniform region. A specific number of corners could be chosen, say 1000, however this would not be suitable in many cases, again the corners could suffer from the two problems above. The number of corners may be forced to be too high, giving poor tokens, or too low, giving a non-uniform image description. Filtering and other methods are being examined to provide a more dynamic and consistent choice of threshold values.

The interconnecting physical structure of the mesh is then created using a modified iterative Delaunay triangulation algorithm.

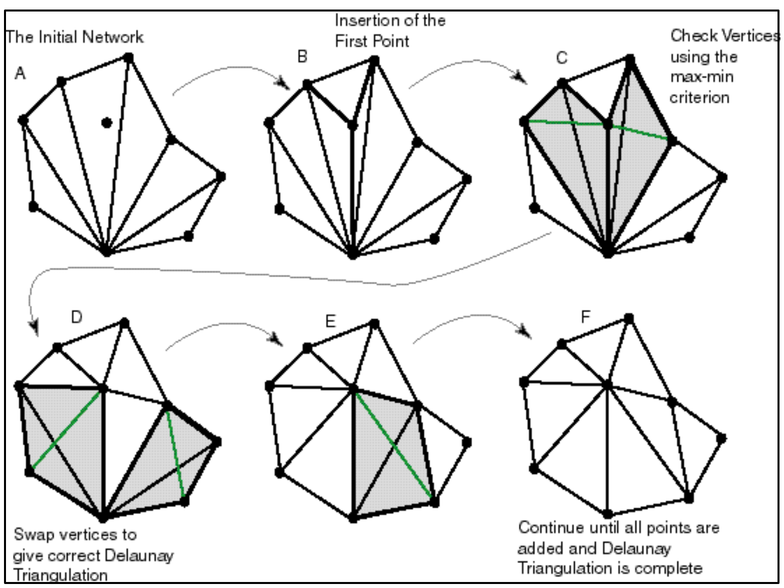

Fig. 6. The creation of the Delaunay Triangulation using the Incremental Algorithm.

Given a set of data points, the Delaunay triangulation produces a set of lines connecting each point to its natural neighbours. A Delaunay triangulation is desirable for approximation applications because of its general property that most of the triangles are almost equiangular and also because there is a unique triangulation for a given set of points. Unlike many other algorithms for determining the Delaunay Triangulation, the incremental algorithm has the main advantage of keeping the triangular network as a Delaunay triangular network during the actual triangulation process. In the case of the active-mesh being described, cases can arise where mesh nodes become occluded, leave the scene, or become unreliable. In this case, or in the case where nodes new nodes can appear, the incremental Delaunay Triangulation algorithm allows the addition or subtraction of mesh nodes. Fig.7 shows an example test scene, in which this initialisation has been performed. It can be seen from this figure that the feature corner points are detected accurately and the mesh is well constructed by the Delaunay algorithm.

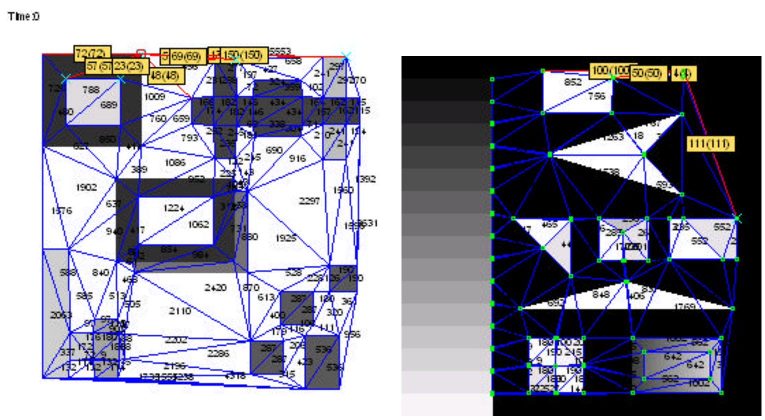

Fig. 7. The mesh created using the modified Delaunay triangulation on random shapes with the detected corners as the node points (The image background on the right is from (7))

5. Formulisation of the Energy Equations

Once the mesh structure is available the internal and external energies within the mesh are established at each node and mesh line, so that the mesh is initially in equilibrium with no internal or external forces being applied. The mesh will then remain in equilibrium until a change in the underlying image structure occurs as the 'active-mesh' is designed so that it deforms in response to salient image features.

To allow for the complexities involved in dealing with 'active-meshes' as opposed to the more simple contours, the mesh algorithm must allow for varying numbers of line connections to each node and provide an algorithm for dealing with the numerous forces that will be applied at each node.

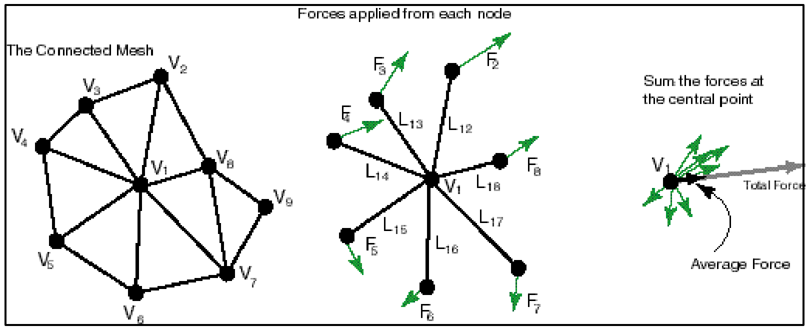

Fig. 8. Multiple forces being applied to a single node, the combination of which is performed using the force strength and the distance of the forcing nodes from the centre node.

An algorithm was developed to deal with the varying number of connected nodes, resulting from the initialisation algorithm, with a method of combining the effects of the forces from the connected nodes on a particular node. It was determined experimentally that the closer a connected node was, the more likely that this node would be present on the same real-world object. So examining centre node $n_0(x, y)$ with connected nodes $n_1, n_2, \ldots n_N$ the combination is of the form:

$$a_i = 1 - \frac{D_i}{\sum_{i=1}^{N} D_i} \text{ where, } D_i = \sqrt{(n_0(x) - n_i(x))^2 + (n_0(y) - n_i(y))^2} \tag{1}$$

$$F_0(x) = \sum_{i=1}^{N} a_i F_i(x) \tag{2}$$

$$F_0(y) = \sum_{i=1}^{N} a_i F_i(y) \tag{3}$$

The larger the distance of the node applying the force from the current node, the smaller the effect it has on the movement of the current node. These forces being applied from the surrounding nodes can be due to those nodes being pulled away from or towards salient image features.

5.1 The Internal Energy

For the internal energy an elastic form is used, where every node pulled or pushed by the connected nodes.

The mesh-lines have elastic properties so that the mesh can deform when required over a number of time iterations, to track deformations in the scene, the scene objects or deformations due to scaling. The elastic properties give the mesh its flexibility while the rigid properties give the mesh structure. The rigid properties of the mesh lines cause the lines to attempt to return to their determined length. This determined length is permitted to expand or contract slowly over a time period, and is influenced by the elastic properties of lines. In other words, if the mesh is stretched by a number of consistent external forces for a significant number of iterations then the mesh will slowly assume a new default shape. This default shape is now the rigid shape of the mesh and will remain so, until similar forcing conditions arise.

For each line in the mesh:

$$F_x = L(x)_{cur}\left(\frac{L_{set} - L_{cur}}{a_{Line} L_{cur}}\right) \tag{4}$$

$$F_y = L(y)_{cur}\left(\frac{L_{set} - L_{cur}}{a_{Line} L_{cur}}\right) \tag{5}$$

where *L(x)*, *L(y)* represent the *x* and *y* components of the mesh line lengths. The internal forces are determined by the current-length of each individual mesh line in comparison to the set-length of that mesh line. The mesh line has two nodes n_1 and n_2 where,

$$F_{n_1} = \left(-F_x, -F_y\right) \text{ and } F_{n_2} = \left(F_x, F_y\right) \tag{6}$$

At each iteration:

$$L_{set} = L_{set} + a_I\left(L_{cur} - L_{set}\right) \tag{7}$$

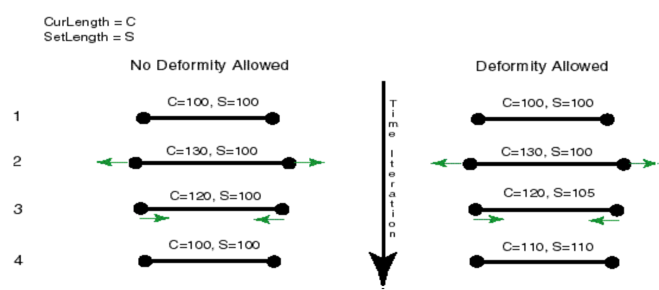

Fig. 9. An example of a mesh line deformity in (a) how a rigid mesh would react and in (b) how a deformable mesh might react.

5.2 The External Energies

The external forces are applied to the mesh nodes independent of the mesh lines and are derived from the image data. These image forces pull the mesh nodes towards suitable feature match points that are found within the circular image search space of the mesh nodes. If a suitable match feature appears within the circular search space then the node is pulled towards that feature point by a force magnitude determined by the suitability of the match feature. This in turn pulls the connected mesh nodes (due to the internal forces of the interconnecting mesh lines) in the direction of the new feature.

The best match corner is found by comparing the 3x3 area surrounding the current node n_0 with the 3x3 area surrounding the possible match corners c_n detected within the circular search space of radius a_S. So, $\forall c_n(x, y)$ where the distance from n_0 the current mesh node,

$$d = \sqrt{(c_n(x) - n_0(x))^2 + (c_n(y) - n_0(y))^2} \tag{8}$$

is less than the search space of radius a_S, the total intensity difference is:

$$I_T = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \left| I(n_0(x+i, y+j)) - I(c_n(x+i, y+j)) \right| \tag{9}$$

The corner point c_n is chosen that minimises the value of I_T in the range 0 to 2295(i.e. 255x9).

Based on this intensity difference, match strength is established. The larger this value the weaker the match strength and a factor is established:

$$S_M = 1 - \left(\frac{I_T}{9 \times 255} \right) \tag{10}$$

I_T is larger the smaller the difference, averaged over 9 pixels and over the maximum intensity value 255, so S_M = 1 for the best match and 0 for the worst match.

$$F_{ext(x)} = a_E S_M d(x) \left(\frac{a_S - d}{a_S} \right) \tag{11}$$

$$F_{ext(y)} = a_E S_M d(y) \left(\frac{a_S - d}{a_S} \right) \tag{12}$$

Where a_E is a user-defined factor to allow the external forces to have a larger or smaller effect on the mesh. The last term weights the distance of the force as weaker the larger the distance from the examined node.

The active mesh allows constrained feature matching to take place on a frame-by-frame basis in an image sequence. Once the features have been correctly matched the real-motion of the image of selected points in the image plane are available as vectors.

6. Initialisation and Tracking Results

A specialised software application (written in Java) was developed to implement the 'active-mesh' algorithm. A number of sequences with both artificial and real-world scenes are shown here with the algorithm providing initialisation information and tracking information of the objects in the scene. Results are available and very encouraging, with the extracted vector fields being displayed.

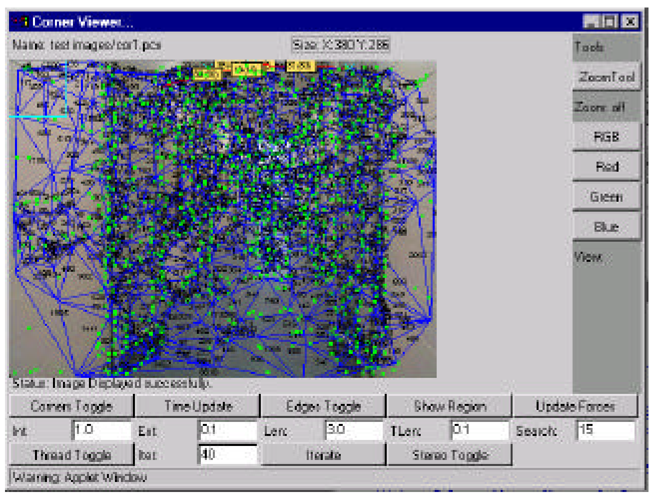

Fig. 10. Showing a more complex Active Mesh initialised on a real-world image frame having tracked image features through several frames.

Fig.11(a), shows the resulting vector field from the rotation of the mesh in Fig.7(a), by about 2° clockwise around the centre of the image. Only two frames were used and the results were taken after 60 iterations. As can be seen the results are very accurate except for some noise at the very centre of the image. The structure of the mesh is perfectly preserved through the rotation.

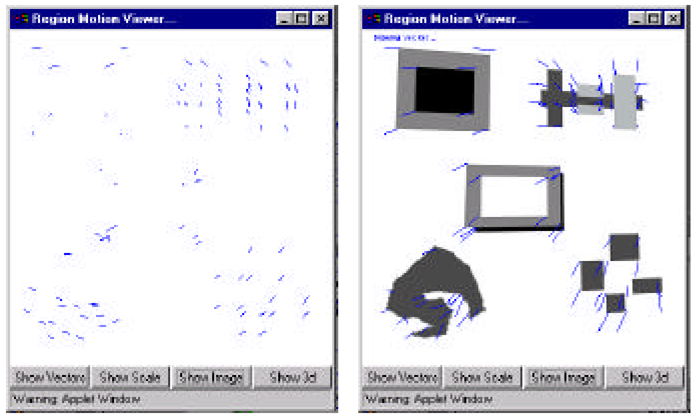

Fig. 11. (a) Shows results from the rotation of the scene as given in Fig.7(a). (b) Shows the results from a substantial distortion of the image, causing non-rigid mesh motion.

The top left corner has been pulled out of shape slightly but this would converge if more than 60 iterations were allowed. In this case the results are derived from a rotation of a *rigid* mesh, but one of the strengths of this method is that the mesh need not be rigid on a frame-to-frame basis. The method is applied to a distorted image, in which many corners appear and disappear due to the distortion or occlusion.

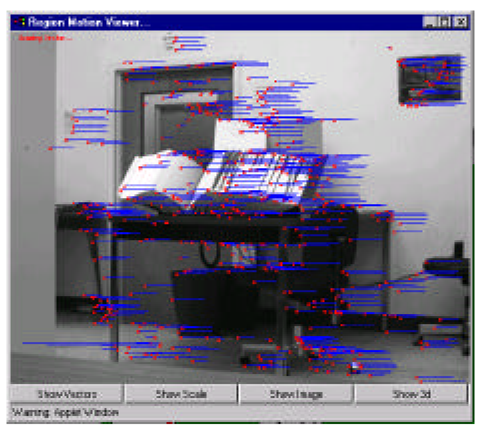

Fig. 12 Vector results from a standard real-world image stereo pair.

This is shown in Fig.11(b), the distortion that is applied is substantial throughout the entire image but the only area that returns slightly problematic results is the object in the bottom left of the image. The extra corners that are detected in the distorted image, along with the fact that the intensity values at these new corners are very similar to the 'correct' corner matches at these points. It is unlikely that such uniform intensity levels (to exact pixel value) exist in real-world scenes. The vector field shown in Fig. 11(b), gives a clear indication of the real motion of the scene, nearly like a 'black-hole' effect in the top right hand corner, with all objects being 'sucked in'.

7. Conclusions

This method shows the use of a self-initialising 'active-mesh', that functions with promising results. It has primary application in scenes where no *a priori* knowledge is available, or where unknown motion events can occur in subsequent image frames. The initialisation of the mesh location is based on an incremental Delaunay triangulation so that node points may be added and removed dynamically as they appear and disappear on a frame-by-frame basis. The SUSAN algorithm was used for choosing mesh nodes, however the active mesh was designed to initialise using and track any form of strong motion features. The energy equations that were used were specifically designed for the initialisation technique developed. The technique provides promising tracking results, providing information about the 'real' motion in the scene, which can be difficult for techniques such as optical flow. Improvements are being made to the algorithm on an on-going basis and this method is currently being applied to the 3D scene analysis problem.

For more information on this work and for an interactive Java Applet see:

http://www.eeng.dcu.ie/~molloyd/phd/

8. Acknowledgements

We wish to thank the School of Electronic Engineering, Dublin City University, Ireland for supporting this research.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. Proceedings of the 1st International Conference on Computer Vision, (1987) 259-268
2. Cohen, L. D.: NOTE On Active Contour Models and Balloons. Computer Vision Graphics Image Processing: Image Understanding, 53, 2, (1991) 211-218
3. Staib, L. H., Duncan, J. S.: Boundary Finding with Parametrically Deformable Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14, 11, (1992) 1061-1075
4. Leymarie, F., Levine, D.: Tracking Deformable Objects in the Plane Using an Active Contour Model. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15, 6, (1993) 617-633

5. Neuenschwander, W., Fua, P., Szekely, G., Kubler, O., Initialising Snakes. Proc of the 1994 IEEE Computer Society on Computer Vision and Pattern Recognition, Seattle, (1994) 658-663
6. Lai, K. F.: Deformable Contours: Modelling, Extraction, Detection and Classification. Ph.D. Thesis at the University of Wisconsin-Madison (1994)
7. Smith, S.: A New Class of Corner Finder. British Machine Vision Conference, Leeds, (1992) 139-148
8. Cohen, L.D., Cohen, I.: Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15, 11 (1993) 1131-1147
9. Hashimoto, M., Kinoshita, H., Sakai, Y.: Sampled Active Contour Model, Transactions of the Institute of Electronics, Information and Communication Engineers, J77D-2, (1994) 2171-2178