A REAL-TIME LOW-COST VISION SENSOR FOR ROBOTIC BIN PICKING

BY

OVIDIU GHITA (M.ENG.)
(GHITAO@EENG.DCU.IE)

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
(ELECTRONIC ENGINEERING)

SUPERVISED BY DR. PAUL F. WHELAN

SCHOOL OF ELECTRONIC ENGINEERING
DEPARTMENT OF ENGINEERING AND DESIGN
DUBLIN CITY UNIVERSITY

JANUARY 2001

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

Signed: ……………………………………. ID No: ………………………………….

Date: ………………………………

Acknowledgements

I would like to thank Dr. Paul Whelan for his guidance and help throughout this project. I wish to thank Prof. Charles McCorkell and the School of Electronic Engineering for supporting this research. I would like to express my gratitude to Motorola BV, Ireland for funding this project. Special thanks goes to Peter Shortt for his constructive comments regarding the issues associated with the implementation of this work.

I would like to thank the members of the Vision Systems Laboratory at DCU, especially Derek Molloy, Robert Sadleir, Alexandru Drimbarean and John Mallon for their friendship and listening to the many problems that occurred during this time. Special thanks must go to my parents, who supported my education and for their constant encouragement.

I would also wish to thank the external examiner Prof. David Vernon for the time that he put into the examination of this work and for the suggestions that he made.

Abstract

A real-time low-cost vision sensor for robotic bin picking

Ovidiu Ghita

Under the supervision of Dr. Paul Whelan

Dublin City University

Submitted in partial fulfilment of the requirements for the degree
of Doctor of Philosophy at Dublin City University, 2001

This thesis presents an integrated approach of a vision sensor for bin picking. The vision system that has been devised consists of three major components. The first addresses the implementation of a bifocal range sensor which estimates the depth by measuring the relative blurring between two images captured with different focal settings. A key element in the success of this approach is that it overcomes some of the limitations that were associated with other related implementations and the experimental results indicate that the precision offered by the sensor discussed in this thesis is precise enough for a large variety of industrial applications. The second component deals with the implementation of an edge-based segmentation technique which is applied in order to detect the boundaries of the objects that define the scene. An important issue related to this segmentation technique consists of minimising the errors in the edge detected output, an operation that is carried out by analysing the information associated with the singular edge points. The last component addresses the object recognition and pose estimation using the information resulting from the application of the segmentation algorithm. The recognition stage consists of matching the primitives derived from the scene regions, while the pose estimation is addressed using an appearance-based approach augmented with a range data analysis. The developed system is suitable for real-time operation and in order to demonstrate the validity of the proposed approach it has been examined under varying real-world scenes.

“And the strength of the mental image which impresses and moves them comes either from the magnitude or the number of the *preceding* perceptions. For often a strong impression produces all at once the same effect as a *long-formed* habit, or as many and *oft-repeated* ordinary perceptions…and supposing there were a machine, so constructed as to think, feel and have perception, it might be conceived as increased in size, while keeping the same proportions, so that one might go into it as into a mill.”

The Monadology,
Gottfried Wilhelm Leibniz, 1714

Table of Contents

Table of Figures

Chapter 1 - Introduction

The objective of this chapter is to introduce the motivation for the investigation of a vision system that provides sufficient information for a bin picking robot to locate, recognise and manipulate industrial objects. The background to this research will be outlined and the problems that are related to current approaches will be discussed. This chapter concludes by presenting an overview of the thesis.

1.1 The problem and motivation

In recent times the presence of vision and robotic systems in industry has become commonplace, but in spite of many achievements, a large range of industrial tasks still remain unsolved due to a lack of flexibility of vision systems when dealing with the manufacture of products in small batches (Whelan and Batchelor, 1996).

Robots are used to deliver a wide range of products and services in an integrated manufacturing environment. In such an environment, the workpieces to be processed or inspected are commonly supplied in bins and before these workpieces are manipulated it is necessary to have full knowledge of their identity, location, size, shape and orientation. Prior to the availability of flexible robotic systems, human labour was employed to perform material handling, but in today's competitive market the use of robots for such industrial tasks is playing a more important role than ever before. This issue was specifically addressed by Porter *et al* (1985) when they clearly demonstrate that the use of human labour for such tasks has no economical or social justification while "the development of robot technologies will require technicians, programmers, maintenance workers, and operators", jobs that offer better work conditions but require higher skills[1]. Thus, the incorporation of robots in manufacturing industry is not necessarily about the displacement of labour, but more as an answer to the expectation of an increasingly educated labour force and the compliance to the recent realities of the world economy.

[1] There are important social issues addressed by this paper. However, a discussion on this topic is beyond the scope of this thesis.

The task of locating, recognising and evaluating the position of the objects jumbled in a bin is referred in the vision literature as automatic *bin picking*. Over the last two decades vision engineers have tried to address this problem, their efforts being concentrated in the development of an adaptable system capable of handling a wide range of products.

Historically, this problem was tackled using mechanical vibratory feeders where vision feedback was unavailable. This solution has certain problems especially with parts jamming. Very often, workpieces have shapes that make them impossible to orient or can tangle with one another (Kelley *et al*, 1982). Some parts can be damaged or scratched against each other or the orienting device. Furthermore, the mechanical feeders are very noisy and the vibrations can be a problem for other parts of the system. Another important disadvantage is the fact that these machines are highly dedicated. If a design change is made in how the product is manufactured, the changeover may include an extensive re-tooling and a total revision of the system control strategy. Due to the aforementioned disadvantages, robotic[2] vision systems represent a cost effective solution (Yoshimi and Allen, 1994).

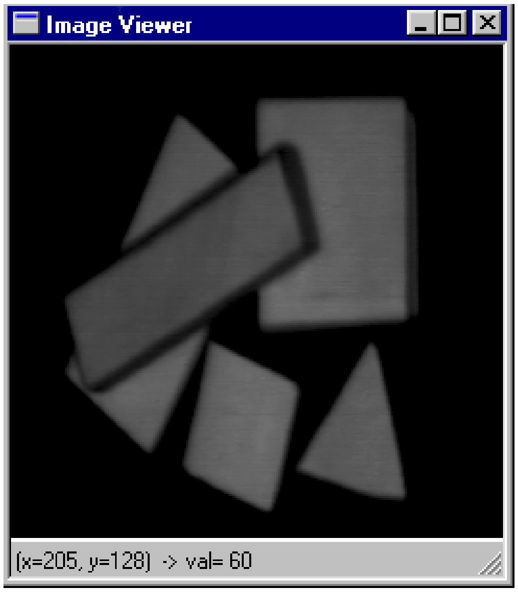
The ability to manipulate objects under *visual control* is one of the key issues required by an automatic bin picking system. With the incorporation of vision systems, some information about the scene can be obtained by analysing the images taken from the working environment and as a result a robot can be used to perform the operations previously performed by mechanical vibratory feeders, while avoiding the inconveniencies associated with them. This solution represents a clear step forward because due to its increased adaptability it is better suited for batch applications.

The bin picking problem has been the subject of research for quite a few years and reviewing the existing systems, none of them so far indicates a solution to solve this classic vision problem in its generality. While there are many possible ways to explain this circumstance, the main facts that hamper the implementation of a generic bin picking system are:

- An image is a *two-dimensional* (2-D) projection of the *three-dimensional* (3-D) scene.

[2] According to the Robot Institute of America an *industrial* robot is a "re-programmable multifunctional manipulator design to move materials, parts, tools or specialised devices through variable programmed motion for the performance of a variety tasks".

- Dealing with *clutter* and *occlusions* generated by the randomly placed objects in the scene (see Figure 1.1).
- Recovering the depth (3-D) information of the scene.
- A meaningful 2-D or 3-D scene interpretation.
- An invariant to translation, rotation and scaling scheme able to recognise and estimate the 3-D pose for general objects.
- The reflections generated by the object's surfaces are material and viewpoint dependent.
- Various technical limitations imposed by the precision of sensorial information or mechanical constraints introduced to the system by the gripper.



(x=205, y=128) -> val= 60

Figure 1.1 A typical scene for a bin picking application.

The abovementioned problems clearly emphasize that the bin picking task is extremely complex. In addition to scene interpretation, the task of dealing with object occlusion and the generation of precise 3-D estimation of the scene represent the key issues for a bin picking implementation.

1.2 Human and robotic bin picking

One question that constantly arises is how the human operator can adapt to a variety of situations that include bin picking? It is well known that humans have the ability to recognise and manipulate objects that are partially occluded. In this process the human operator uses a large amount of perceptual information under normal operating conditions[3]. Depending on the technology, a robotic system would be expected to perform better than the human operator at some relatively uncomplicated quantitative tasks. While for human labour the efficiency decreases as the time passes, a robotic vision system can theoretically work for 24 hours a day, 7 days a week. Also it is important to note that the manufacturing environment is often unhealthy if not even dangerous, a situation that makes the presence of robots mandatory.

To implement a bin picking system with a human-like flexibility *is not* a realistic approach. While some analogies may be useful in the implementation of a robotic system, the danger in relying on such human driven approaches is that simpler and possibly more efficient solutions can be overlooked (Whelan and Batchelor, 1996). This does not mean that vision researchers should abandon the goal of developing human-like vision systems. However, no bin picking system in the foreseeable future can approach the perceptual and decisional abilities of a human operator and this issue will be emphasised in the literature survey that will be presented in the next section.

1.3 Bin picking literature survey

Many vision researchers have tried to solve the bin picking problem for a specific application. Thus, Kelley *et al* (1983) proposed a heuristic approach to workpiece acquisition without any visual information. A "blind" robot with only tactile-type sensorial information could acquire a workpiece by scanning a bin until a contact with a workpiece is sensed. Although simple, this technique is inadequate due to the amount of time required to find a graspable object. Also, the risk of damaging the workpiece or the gripper is significant. Some improvements were obtained by using visual feedback. Birk *et al* (1981) developed a bin picking system able to locate a

[3] It is acknowledged that the performance of the human operator is not constant and is highly influenced by the external factors. Try to imagine an exhausted, ill or drunk person that attempts to perform a bin picking task.

graspable surface with good reflecting characteristics. The developed robotic system was able to manipulate a *single-class* of plastic boxes. The key component of their system is a simple shrinking algorithm which is applied in order to locate a graspable surface situated near the top of the bin. This approach does not use any 3-D information to compute the position and orientation of the object in question while it is “realistic to compute the pose of a piece in the hand after the workpiece is acquired”. This observation might be true as long as during the acquisition the workpiece may shift relative to the gripper. Because the workpieces have a simple configuration, the pose is estimated when the object is grasped by using one or two images. The system can manipulate objects that can be grasped only on planar surfaces using a vacuum gripper. Two years later, Kelley *et al* (1983) extended the shrinking algorithm to grey-scale image processing in order to increase its robustness. In the same paper they propose an alternative solution to decompose the scene in parts by employing an algorithm based on edge propagation.

A similar approach was employed by Kelley *et al* (1982) in the implementation of a bin picking system for acquiring cylindrical objects. The key part of the system is the jaw gripper which is able to handle a family of objects such as cylinders and pieces obtained in the subsequent stages of the manufacturing process. This implementation can handle a wider range of applications than the systems equipped with vacuum grippers.

Dessimoz *et al* (1984) developed a conceptually related system based on *matched filtering* technique. The principle of this method is straightforward and consists of matching *local* patterns *p(i,j)* defined on the original image *f(i,j)*. To accomplish this goal, a filter *h(i,j)* is applied and a match with *p(i,j)* occurs if it produces a peak in the output image where *p(i,j)* is located. The main problem associated with this approach is the fact that the pattern to be matched varies with the viewing angle and object rotation. Therefore, this technique is practical only when dealing with objects with simple shapes, a case when the number of filters necessary to sample the object’s appearance is relatively low.

The main drawback of the systems described above is that they rely on simple algorithms restricted to their applications and consequently their solutions are over-constrained. Since the systems do not perform the recognition task, the objects contained in the bin have to be similar. Also, 3-D information is not available and consequently the position of the graspable region is only approximately known. Thus, these systems can only be applied to rigid objects with simple shapes that cannot be damaged when they are manipulated. While these assumptions may be acceptable for a small range of applications, in most practical cases a flexible system must deal with more than one type of object with a wide scale of shapes.

Other areas of related research include the work of Rechsteiner *et al* (1992) in which they discuss the implementation of a system for sorting postal parcels. The proposed system performs very well for cluttered scenes when the objects are in contact but fails when they are occluded. The contour of parcels is identified by using the edge structure returned by the Kirsch-compass operators. The 3-D information (provided by a range sensor based on optical triangulation) is employed to select a suitable parcel for manipulation. If there are no graspable parcels the robot rearranges the scene in order to obtain a better configuration.

In the paper by Rahardja and Kosaka (1996) a bin picking system for handling alternator covers is discussed. Since the alternator covers have a very complex shape, it would be difficult to analyse them as independent entities. To cope with this situation, the authors highlight the importance of simple entities such as circular and polygonal surface patches called *landmark features*. Therefore, they define the recognition and pose estimation of the target objects as those of their landmark features. To carry out this concept they developed a scheme for region extraction using a split and merge algorithm. Nevertheless, the landmark recognition process has to accommodate the anticipated appearance distortion due to viewpoint changes. For the purpose of simplifying this process, the algorithm selects a list of landmark features that fulfil an aspect criterion. The pose estimation for the *selected* feature is determined using the depth information supplied by a stereo range sensor. Although the system is designed for a particular application it is a good example which demonstrates the effectiveness of combining heuristic vision algorithms and sensor equipped robots to handle a range of robotic applications.

1.3.1 2-D object recognition approaches

As noted earlier, a bin picking system has to address three difficult problems: scene interpretation, object recognition and pose estimation. These problems were found to be more difficult than originally anticipated and despite many attempts still represent a challenge for vision researchers. Henderson (1983) highlighted the object representation as an important step in solving many problems in scene analysis. As might be expected, the choice of object representation will determine the method required for surface extraction. The vision literature indicates that shapes can be represented in various ways, but ultimately they can be divided into two main categories: 2-D and 3-D representations.

2-D surface representations describe the contour or the periphery of a planar shape while the 3-D representations describe the objects using various volumetric primitives. In this regard, Wechsler and Zimmerman (1988) proposed to use *distributed associative memory* (DAM) as a 2-D recognition component for a bin picking system. Their implementation consists of two subsystems. The first component derives an invariant representation based on the use of complex-log mapping in order to transform an image from cartesian coordinates into polar exponential coordinates. This representation is very convenient as long as the rotations and changes in scale are transformed into translation in the complex-log domain. The second component builds and interprets the DAM by projecting the stimulus vector onto the memory space. For this approach, the object under investigation has to be placed in the middle of the image while small shifts from the centre cause severe distortions in the complex-log mapped image. This issue can be a serious problem especially when the objects are overlapped.

After a number of years, Tsang and Yuen (1993) suggested the use of *difference chain code* (DCC) for recognising partially occluded objects. When dealing with object occlusion the boundaries associated with the scene objects are not completely described. To handle this problem, the authors proposed to search only for *segments* of the object’s contour instead of searching for a complete boundary. Because the variations in viewing angles determine distortions, a solution to overcome this problem relies on using *a non-linear elastic matching* algorithm based on dynamic programming. The results were found to be encouraging but the difficulty in selecting relevant contour segments and the computational overhead are the main limitations of this approach.

The paper by Forsyth *et al* (1991) investigates different aspects of the application of invariant theory to model-based vision. They proposed a recognition scheme using the shape invariants (i.e. conics) which are not affected by the transformation between the object and the image plane. In order to be useful, these invariants must be accurate and stable. The authors decomposed the task into two sub-problems. The first component consists of conic extraction using the information returned by an edge detector. Usually, due to occlusion the curves contained in the image are not completely described. If algebraic invariants are used it may be possible to overcome this problem by fitting a conic to an image curve. The remaining problem deals with matching a model from the database. A problem associated with this approach is the large number of hypotheses created which restrict the efficiency of the proposed scheme. To cope with this issue, a mechanism for *grouping* and *indexing* was proposed. Using this mechanism, the number of hypotheses is drastically reduced and the experimental results demonstrate the potential of this approach to recognise *planar* objects in cluttered scenes.

Bose *et al* (1996) investigated the use of affine invariant moments to recognise rigid flat objects. Since the recognition of partially occluded objects is investigated, the authors allocate a higher importance to the local features while the global features are used only in the verification stage. In this regard, the boundary points resulting after the application of the Canny edge detector are employed as local features. Then, from the edge information the shape is approximated with polygons and the invariant moments are computed. This approach was shown to be reliable for recognising simple planar objects but it is not applicable when dealing with 3-D objects.

A different approach was proposed by Kriegman and Ponce (1990) where they employed the shape of image contours for recognising real objects and estimate their pose. The elimination theory was employed to determine the implicit equations of the image contours formed by the projections of edges and occluding contours. This approach was applied only to objects with a surface of revolution and the edges were hand selected. The authors acknowledged that computing the implicit equations for generic objects remains a long term goal. Some years later, Vijayakumar *et al* (1998) tried to find local invariants such as contour bi-tangents associated with the projection of the object's surface taken under different viewpoints. A bi-tangent should be seen as a line which touches the surface of the object at two distinct points that belong to the same tangent plane. These invariants can be computed from a single 2-D image and the recognition process consists of matching the scene bi-tangents with those contained in the database. In contrast with the implementation suggested by Kriegman and Ponce (1990) this scheme works for objects which have a complex shape and the authors suggested that it should be considered as a *component* of a complete recognition system.

Dickinson *et al* (1992) proposed a solution to represent 3-D objects from a single 2-D image. They suggested a set of ten volumetric primitives called *geons* which are considered flexible enough to model a large number of objects. These primitives are the *projections* of the following 3-D models: block, truncated pyramid, pyramid, bent block, cylinder, truncated cone, cone, truncated ellipsoid, ellipsoid and bent cylinder. Nevertheless, these primitives cannot directly represent the object under investigation due to self and mutual occlusions. To tackle this problem, they developed a hierarchical representation called *aspect hierarchy*. In other words, a primitive is decomposed in subcomponents in a hierarchical fashion. *Aspects* which describe a distinct primitive are placed at the top level of this hierarchy. Each aspect consists of a set of 2-D faces. Due to occlusion some of the faces contained by an aspect can be partially or completely missing. This introduces the motivation of the second level of the hierarchy, a level where *faces* which are closed 2-D contours are situated. Again due to occlusion a face can be only partially recovered. Therefore, the lowest level of hierarchy is represented by *boundary groups*. This representation is appropriate because the resulting graph which describes the object of interest is independent of translation, rotation and scaling. The major problem posed by this approach is the bottom-up primitive extraction and the verification of possible groupings in order to match a model contained in the database. The strength of this approach is given by the hierarchical representation and the use of relatively complex primitives which drastically reduce the number of possible groupings.

Bergevin and Levine (1993) addressed the *generic* recognition of *unexpected* 3-D objects from single 2-D views. The developed system called PARVO is closely related to the implementation suggested by Dickinson *et al* (1992). Instead of dealing directly with volumetric primitives or with the subcomponents that are hierarchically derived from them, PARVO extracts pairs of line segments that have close endpoints and grouping them into junctions such as *arrow*, *fork*, *peak*, *T type* and *tree*. The next stage of the algorithm (hypothesis generation and validation) determines the geons that can be compatible with the extracted junctions. The set of geons utilised in their experiments is very similar with that proposed by Dickinson *et al* (1992). The final object description consists of a graph of geons augmented with a qualitative *aspect ratio* for each extracted part. The aspect criterion assigns a symbolic value in agreement with the shape of the geon in question which can be elongated, flat or bloblike. Despite its increased robustness, this formulation shares the same merits and limitations as the previous implementation.

A common problem related to the aforementioned approaches is the difficulty associated with extracting the scene primitives which usually consists of analysing the edge structures. If the objects contained in the scene are highly textured, the effort to extract meaningful primitives is cumbersome. Consequently, the recognition problem can be formulated as one of matching *appearance* rather than shape.

In this regard, Turk and Pentland (1991a) developed a face recognition system that uses *principal component analysis* (PCA) to learn and recognise images of human faces. Then, Murase and Nayar (1995) extended this approach by developing an appearance-based system suitable to learn, recognise and estimate the position of complex objects using 2-D images. This approach is suitable for the recognition of multiple objects but cannot handle occlusions.

To cope with this problem, Ohba and Ikeuchi (1997) proposed to *divide* the appearance into small windows and to apply eigenspace analysis to each window. Because the number of windows that are necessary to be stored is extremely large, a framework using criteria such as detectability, uniqueness and reliability was developed in order to select only relevant windows. This approach is effective if the objects present pronounced textural characteristics.

An alternative approach using *colour histograms* was suggested by Swain and Ballard (1991). Their implementation consists of a colour-indexing scheme where recognition is achieved by histogram matching. Nevertheless, representing the histogram using the entire colour space makes this approach computationally inefficient. To address this issue, they proposed to reduce the histogram's resolution by using a restricted colour space. As long as this approach uses only global properties which are sensitive to partial occlusion, the proposed implementation can handle the recognition of a single object within the image. The results proved to be encouraging when the objects to be analysed have a complex appearance and the illumination conditions are maintained at constant level.

Schiele and Crowley (1996) extended this approach by using *multidimensional receptive field histograms*. Their implementation consists of a general statistical object representation framework, where the multidimensional histograms are used to approximate the probability density function for *local* appearance. The authors outlined some of the concerns associated with the sensitivity of local features such as invariant, equivariant and robust properties. Most of the invariant local properties are based on the calculation of higher order derivatives. Because these derivatives amplify the high frequencies, this issue can create problems related to instability due to sampling and digitising noise. Consequently, the authors found it necessary to "weaken" the requirement of invariance. Because the equivariant properties are a function of some transformations such as scaling and rotation, their use is restricted to certain classes of objects. Therefore, the robust local properties such as those derived from the local appearance are more appropriate because they change slowly and in a predictable manner with respect to viewpoint transformations. For this purpose the authors employed Gaussian derivatives because they are robust to image plane rotations. This scheme proved to be relatively robust to viewpoint changes. Although the proposed system can handle only the recognition of objects with different textural characteristics, the reported results are impressive.

1.3.2 3-D object recognition approaches

An inherent problem derived from 2-D object representation is the fact that the primitives used in the recognition stage are directly derived from the information contained in a grey-scale (or colour) image. As mentioned earlier, the 2-D geometric features contained in the image represent the projection of 3-D objects. This is a serious issue as long as these features are viewpoint dependent. In addition, when dealing with textured objects, the 2-D representation may be inappropriate in some application areas. Many of the problems outlined above have been successfully addressed by techniques that employ range images to recognise objects and estimate their spatial locations.

Bolles and Horaud (1987) developed a system known as 3DPO for recognising and locating 3-D objects from the range data. Their system is a two-part recognition model: a low-level analysis of range data followed by a recognition scheme based on model matching. The first component of the recognition system locates edges from the range data and classifies them into circular arcs and straight lines. Resulting edges are partitioned and classified in a multi-step process. In this way, a circular edge is expected to have one planar surface and one cylindrical surface adjacent to it while a straight edge may be the intersection of two planes. After the edge classification process is completed, the low-level analysis continues with indexing the visible surfaces that are adjacent to these edge features. The second component of the system recognises an unknown object by searching the model database for features that match those that are associated with the object to be recognised. The proposed system works well for complex scenes containing clutter and occlusion but with very few models. Furthermore, the system is better suited for recognition of curved objects.

Horn's (1979) paper introduces the *Extended Gaussian Image* (EGI) representation to recognise 3-D objects. The EGI model is obtained by mapping the normals of the object's surface onto a unit sphere called the Gaussian sphere. If a unit mass is attached to each point where a normal is erected, the result will be a distribution of mass over the Gaussian sphere. This distribution represents the EGI of a 3-D object. In other words, the EGI representation is a histogram which records the variation of surface area with surface orientation information. This approach assumes that the object's surface is divided into a fixed number of faces per surface unit and a normal is erected on each face. Obviously, the bigger the number of faces the more accurate the object representation. The recognition process consists of comparing the EGI of the unknown object with those contained in the model database. There are a few problems associated with this model such as its sensitivity to occlusion and the fact that it assures a unique representation only for convex objects (for more details refer to Appendix C).

Using the same concept, Ikeuchi (1983) employed the *needle map* to determine the attitude of an object. To achieve this goal by searching the entire space is not a realistic approach since all the possible combinations have to be verified in order to match a model from the database. Therefore, it is necessary to reduce the degrees of freedom for a plausible attitude by constraining the search space. Because matching an observed EGI with a model EGI involves three degrees of freedom, the author considered that two constraints are sufficient to solve the problem. The first is the EGI mass centre position while the second deals with the least EGI mass inertia direction. The application of these constraints greatly reduces the number of possible attitudes and the model that maximises the fitting measure is selected as the observed attitude of the object. A number of years later, Distante *et al* (1988) used the previous approach in the implementation of a model–based bin picking system.

As mentioned earlier, the EGI concept is not applicable in the representation of non-convex 3-D objects. To address this restriction, Kang and Ikeuchi (1990) proposed the *Complex* EGI (CEGI) concept that extends the conventional EGI representation. As in the conventional case, the CEGI of an object is a spatial histogram in which a weight value (which is in this case a complex number) is associated with each normal to the surface. In this way, the normal distance from a predefined origin to the face to be analysed represents the *phase* while the *magnitude* is the area of the face. This scheme overcomes some of the problems associated with the conventional EGI representation.

Krishnapuram and Casasent (1989) attempted to determine the location and orientation of general 3-D objects using an approach based on the 3-D Hough transform. The authors discuss the effects of translation, rotation and scaling on the 3-D Hough space. Since the translation and rotation effects are separable, they implemented a hierarchical algorithm to compute the object distortion parameters. The developed system was applied to scenes containing a single object and the reported results were reliable only for objects with a simple configuration.

An alternative approach was developed by Kim and Kak (1991) and consists of a novel *discrete relaxation and bipartite matching algorithm*. The first stage of this algorithm involves range data segmentation, primitive classification and generation of a scene graph. The primitives (features) resulting from the segmentation process are classified to reduce the complexity of matching, since the recognition of primitive blocks can be used to eliminate inapplicable model objects. The problem of recognition is addressed using a bipartite matching algorithm. Initially, bipartite matching is employed to establish the existence of at least one complete matching. In other words, the matching process attempts to identify a plausible group of primitives that possibly match a model from the database using bipartite graphs. The next phase is represented by a *fine-tuning* feature correspondence that verifies the entire set of features that are adjacent in order to recover a complete matching. The last stage of the algorithm deals with pose estimation. To reduce the computational burden only a small number of extracted features are used to form a set of pose transformations.

This method provides robustness and was successfully implemented for a single-object scene. When the algorithm was applied to a multi-object scene, the task to find a *plausible* complete matching becomes increasingly difficult in relation to the number of different objects contained in the scene.

Kak and Edwards (1995) proposed an object representation scheme that incorporates three concepts such as *feature sphere*, *local feature set* and *multi-attribute hash table*. The purpose of this representational scheme is to reduce the complexity of scene-to-model hypothesis generation and verification. As in the previous case, this system performs well when dealing with a single-object scene but the reported results are unsatisfactory for scenes containing a large number of objects.

Stein and Medioni (1992) introduced a novel approach for recognising 3-D *free-form* objects in presence of clutter and occlusion. Their system employs two different primitives for matching. The first primitive consists of small surface patches where differential properties can be reliably computed (*splashes*) while the second primitive employs *3-D curves*. For some objects such as polyhedra, the depth discontinuities represented by edges from the range data (3-D curves) are the natural primitives to represent the object. When dealing with flat highly textured objects this scheme is inappropriate and a better representation relies on the use of splashes. The proposed representation is a very convenient scheme to represent a large range of real 3-D objects. The main problem associated with this approach is the difficulty to extract relevant viewpoint independent splashes.

Many researchers have suggested a structural shape description by using high-level volumetric primitives such as *polyhedra*, *generalised cones or cylinders and super-quadrics*. Brooks (1981) developed an image understanding system called ACRONYM. This system employs a surface representational scheme using generalised cones where the model objects are described in terms of primitives such as *ribbons* and *ellipses*. These structural primitives are indexed into a hierarchical graph constructed by the user. The user intervention is inopportune because restricts the possibility of automatic model building. Furthermore, since both models and scenes are modelled using only generalised cones, this scheme can represent only objects with simple shapes.

Some of these limitations were addressed in the paper by Terzopoulos and Metaxas (1991) where a new family of adaptable volumetric primitives called *deformable super-quadrics* was introduced. A deformable super-quadric is a dynamic model that encapsulates local and global properties inherited from *membrane splines* and *super-quadric ellipsoids*. This formulation is useful to model and recognise 3-D objects or parts of the objects with irregular shapes from the range data. The authors suggested an improved model able to accommodate with other global deformations such as bends, shears and tapers.

Recently, Johnson and Hebert (1999) proposed a framework for simultaneous recognition of multiple objects in scenes containing clutter and occlusion. The recognition method is based on matching surfaces by matching points using the *spin image* representation. The spin image is a 3-D shape descriptor which depends on the object surface curvature. Since the spin images are associated with the 3-D information, in order to be efficient, this approach requires a precise range sensor. The developed recognition system demonstrates reliability when the objects of interest present complex 3-D shapes.

1.4 Conclusions drawn from the literature review

Vision research has a long tradition in trying to find a solution to the generic bin picking problem. From the literature survey, it can be concluded that despite a number of partially successful implementations *an optimal* generic solution to this classical vision problem has not been found. However, there are many algorithmic and technological issues that hamper the implementation of a generic bin picking system and some were already mentioned in Section 1.1.

Early research has tried to solve the bin picking problem strictly related to a specific application. The implementations derived from this approach involves little or no visual information and are suitable only for simple applications where there is no need to recognise the objects contained in the scene. Therefore, these approaches constrain the problem and can deal only with scenes containing similar objects.

A flexible bin picking system has to address some issues such as scene understanding, object recognition and pose estimation. A key element in the design of the system is the choice of shape description. As seen from the literature review, some approaches employ a 2-D shape representation while others approached the problem using methods based on range images. The choice of object representation scheme will determine the method required to compute the pose of the object of interest.

A key aspect in the implementation of a bin picking system is its ability to cope with environmental changes. In line with environmental changes such as illumination conditions, level of dust, etc, the constraints introduced by the gripper and other mechanical limitations should all be considered. Very often the specific (or context) of the application may give invaluable clues in the design of industrial vision systems

The aim of this survey was to introduce the bin picking problem by presenting a large number of approaches relevant to this area of research. The review also indicated that the likely area where a bin picking system should be used is a batch manufacturing environment which deals with a relative small numbers of different components. This is mainly motivated by the fact that 98% of products are made of fewer than 25 parts (Delchambre, 1992).

1.5 Overview of the present implementation

This thesis presents the theoretical framework and describes the development of a real-time *integrated* low cost vision sensor for bin picking. The literature survey stressed that even for a well defined problem the implementation of a bin picking system is difficult. In this sense, the main objective of this research is to develop a theoretical framework suitable to be used in the implementation of an integrated system that provides sufficient information for a bin picking robot to manipulate various polyhedral objects that are randomly placed in the scene.

The approach presented in this thesis consists of three major components. The first component deals with theoretical and practical issues related to the implementation of a bifocal range sensor based on defocusing techniques. A key element in the success of this approach is that it removes some limitations that were associated with other related implementations. Furthermore, the developed range sensor is mechanically robust which is a key requirement for a robotic application.

The second component is concerned with scene segmentation. The aim of the segmentation process is to decompose the image into disjointed meaningful regions that are used in the recognition process.

The last component deals with object recognition and pose estimation. The proposed implementation introduces a novel scheme that addresses the recognition and pose estimation at different stages. Thus, the recognition scheme employs an approach based on the use of global geometrical primitives while the pose estimation is addressed using a PCA technique augmented with a range data analysis. The proposed formulation is feasible to operate in real-time and an intuitive *graphical user interface* (GUI) is provided.

1.6 Organisation of the thesis

The aim of first chapter is to assess the role of a vision system in an integrated sorting/packing industrial environment. This discussion introduces the bin picking problem and a review of the related work. The main conclusions drawn from the literature survey are discussed. Finally, an overview of the thesis is presented.

Chapter 2 details the theoretical and practical aspects related to the implementation of a bifocal range sensor. The main issues associated with this approach such as active illumination, inverse filtering and the calibration procedure are then discussed in detail. Real-time implementation issues and some experimental results are also examined.

In Chapter 3 the image segmentation problem is introduced. Several techniques are presented and the problems that are associated with them are discussed. The discussion continues with the edge-based segmentation approach followed by a detailed presentation of some popular edge detection operators. An important issue related to this segmentation technique consists of minimising the errors in the edge detected output. In this regard, a method to close the gaps in edges using the information derived from endpoints is discussed. Finally, an extensive set of results are presented and analysed.

Chapter 4 begins with a review of some popular methods for object recognition. A discussion that includes the practical issues that must be considered for the successful development of a robust object recognition system is presented. This section outlines the current implementation where each component of the proposed algorithm is discussed and analysed. In the last section of this chapter, a number of experimental results are presented and analysed.

In Chapter 5 the implementation of the entire system is described. For each component a block diagram is presented and its role in the system is explained. The features provided by the graphical user interface are also highlighted.

Finally, Chapter 6 summarises this work and outlines the research contributions that can be drawn from this investigation. This chapter also highlights the parts of the study that require further development and a number of suggestions are presented.

Chapter 2 - Range sensing

2.1 Introduction

It has generally been accepted that a versatile bin picking system cannot be implemented without resorting to 3-D information. For example, the grasping and manipulation of an object implies an understanding of the spatial relationship between the gripper and the relevant object. It is important to note that this relationship involves 3-D scene understanding, i.e. the spatial relationship between the objects that defines the scene. To support 3-D scene understanding, various methods to reconstruct the shapes of complex 3-D objects have been investigated in recent years. Depending on the application, the choice of the range acquisition technique will rely on the task being performed. Each task will differ in terms of the range and the size of the object to be examined; hence, there is no single sensor that is capable of performing satisfactorily in all the necessary applications.

The range acquisition methods can be divided into two main categories: *passive* and *active*. Passive approaches do not interact with the object while active methods make contact with the object or project a structured light onto it. For passive ranging techniques such as *stereo* and *depth from motion*, the 3-D information is obtained by solving the correspondence between different features contained in a sequence of images. Other passive ranging techniques are represented by the depth from focus/defocus methods that use two or more images taken by modifying the focal settings in small steps. It should be noticed that the *stereo* and *depth from defocus* methods can be transformed into active techniques by projecting a special pattern of light onto the scene. Among active range sensing methods two major approaches can be identified: *contact* and *non-contact* 3-D sensors. The contact sensor consists of touch probes which follow the object’s contour with a pointer and usually are mounted on a robotic arm. In terms of precision, these sensors represent the best solution but are slow and very costly. Another drawback is the fact that they make contact with the object and this approach cannot be used when dealing with fragile or non-rigid objects. The non-contact approaches can be further divided into *optical* and *non-optical*. The optical category includes approaches such as methods based on *triangulation*, *active stereo, active depth from defocus*, *Moiré interferometry* and *infrared* (IR) scanning. The non-optical category includes methods such as *microwave radar* and *sonar*.

The next section will describe the theoretical and practical aspects associated with the DFD range sensing technique. Sections 2.2.1 to 2.2.9 describe the actual implementation of a real-time active DFD range sensor and the main contributions of this work are detailed in Sections 2.3 and 6.1.1. Also, a number of popular range acquisition techniques are briefly discussed in Appendix A.

2.2 A bifocal range sensor based on depth from defocus

The aim of this section is to outline an approach capable of extracting 3-D information from the scene by measuring the relative blurring between two images captured with different focal settings. Krotkov (1987), Pentland (1987) and Grossman (1987), independently investigated this method for the first time, attempting to make a connection with human vision. The *depth from defocus* (DFD) methods use the direct relationship between the depth of the scene, camera parameters and the degree of blurring in several images (in the current implementation only two images are used). In contrast with other techniques such as stereo or motion parallax where solving the correspondences between different local features represents a difficult problem, DFD relies only on simple local operators.

Historically, the DFD techniques have evolved as a *passive* sensing strategy. In this regard, Xiong and Shafer (1993) proposed a novel approach to determine dense and accurate depth structure using the maximal resemblance estimation. Subbarao and Surya (1994) reformulate the problem as a one of regularised deconvolution where the depth is obtained by analysing the local information contained in a sequence of images acquired under different camera parameters. Later, Watanabe and Nayar (1995b) argue that the use of focus operators such as the Laplacian of Gaussian results in poor depth estimation. In order to address this problem, they developed a set of broadband rational operators to produce accurate and dense depth estimation. However, if the scene under investigation has a weak texture or is textureless, the depth estimation achieved when passive DFD is employed is far from accurate.

Fortunately, there is a solution to this problem and is offered by *active* DFD, when a structured light is projected on the scene. Consequently, a strong texture derived from the illumination pattern is forced on the imaged surfaces and as an immediate result the spectrum will contain a dominant frequency. The use of active illumination was initially suggested by Pentland *et al* (1994) where the apparent blurring of a pattern generated by a slide projector is measured to obtain the range information. Then, Nayar *et al* (1995) developed a symmetrical pattern organised as a rectangular grid which was optimised for a specific camera. Active DFD is very attractive because it can be successfully applied to both textured and textureless objects. Also, it is worth mentioning that the problems associated with the scene shadows are considerably alleviated when structured light is employed. Obviously, the scene shadows are closely related to object occlusion and this issue is detailed in the paper by Asada *et al* (1998). They proposed an effective solution to compensate for this problem by using the *reversed projection blurring* (RPB) model which is very common in ray tracing techniques. This approach employs the photometric properties of occluded edges when the surface behind the nearer object is partially observed. As expected, due to shadows the blurring model based on convolution becomes inconsistent around the occluding edges. To overcome this limitation, they use the radiance of the near and far surfaces, followed by mapping the occluded region. In the implementation described in this thesis, the occluded region is assigned to be equal to that from the nearer side of the depth discontinuity, which in most of the situations (except cases where the radiance distribution is uniform around the occluded region) is a correct assumption.

2.2.1 Theoretical approach of depth from defocus

If the object to be imaged is placed in the focal plane, the image formed on the sensing element is sharp since every point *P* from the object plane is refracted by the lens into a point *p* on the sensor plane. Alternatively, if the object is shifted from the focal plane, the points situated in the object plane are distributed over a patch on the sensing element. As a consequence, the image formed on the sensing element is blurred. From this observation, the distance from sensor to each object point can be determined by the size of the patch formed on the sensing element. This can be observed in Figure 2.1 where the image formation process is illustrated.

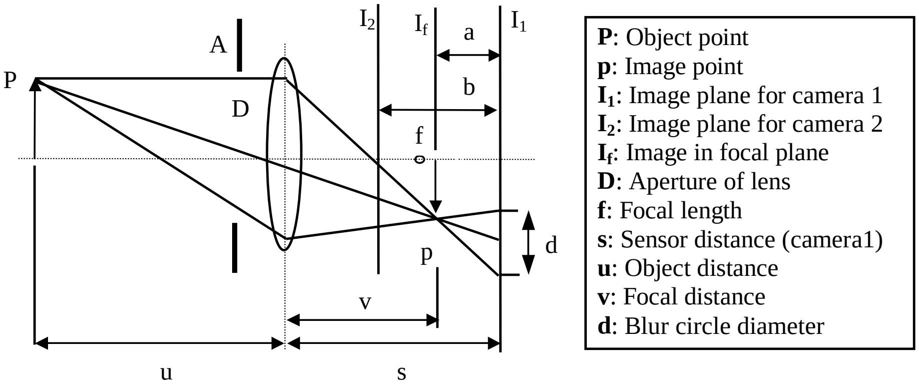

A
P
D
I_2
I_f
a
I_1
b
f
d
p
v
u
s
P: Object point
p: Image point
I_1: Image plane for camera 1
I_2: Image plane for camera 2
I_f: Image in focal plane
D: Aperture of lens
f: Focal length
s: Sensor distance (camera1)
u: Object distance
v: Focal distance
d: Blur circle diameter

Figure 2.1. The image formation process. The depth can be determined by measuring the level of blurring.

Thus, the diameter of the patch (blur circle) *d* is of interest and can be easily determined by the use of similar triangles:

$$\frac{D/2}{v} = \frac{d/2}{s-v} \quad \Rightarrow \quad d = Ds\,(\frac{1}{v} - \frac{1}{s}) \tag{2.1}$$

where *v* is the focal distance, *D* is the aperture of the lens and *s* is the sensor distance. Because the parameter *v* can be expressed as a function of the focal length *f* and the object distance *u* (Gaussian lens law), Equation 2.1 becomes:

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \quad \Rightarrow \quad d = Ds(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}) \tag{2.2}$$

It is important to note that *d* can be positive or negative depending on whether the image plane is behind or in front of the focused image. Consequently for certain sensor displacements, the level of blurring and the resulting images are identical. To overcome this uncertainty, it is necessary to either constrain the sensor distance *s* to be always greater than the image distance *v* (Pentland, 1987) or to employ two images captured with different focal settings (Pentland *et al*, 1989; Turk *et al*, 1989; Nayar *et al*, 1995). It is worth noting that for the former case the depth can be determined accurately and uniquely only for places in the image with known characteristics (e.g. sharp edges). The latter is not hampered by this restriction and for the implementation described in this thesis a pair of images, i.e. the near and the far focused images, separated by a known distance *b* are employed to determine the blur circle.

2.2.2 Constant magnification. Telecentric lens

Unfortunately with the common lens, a variation of the defocus position (*a* or *b*) will produce a variation in image magnification. A solution to compensate for this problem resides in using a telecentric lens. In the image system shown in Figure 2.1, if the image location at point *p* moves parallel to the optical axis as the sensor plane is displaced, as a result a shift in the image co-ordinates of *P* will be produced. This is a problem as it produces a different magnification for images I_1 and I_2. In the telecentric case, the modification between the image system described in Figure 2.1 is an external aperture *A*’ placed in the front focal plane[4]. This addition (external aperture) solves the problem, because if a small aperture is placed in the front of the lens, the *entire scene* appears to be in perfect focus. A simple geometrical analysis (see Figure 2.2) reveals that a ray from any point passing through the centre of the lens’ aperture *A*’, emerges parallel to the optical axis on the image side of the lens (Watanabe and Nayar, 1995a). The result is that (ignoring the blurring) the effective co-ordinates of point *P* in both images (I_1 and I_2) are the same, with the co-ordinates of point *p* for the focused image I_f, proving the anterior observation.

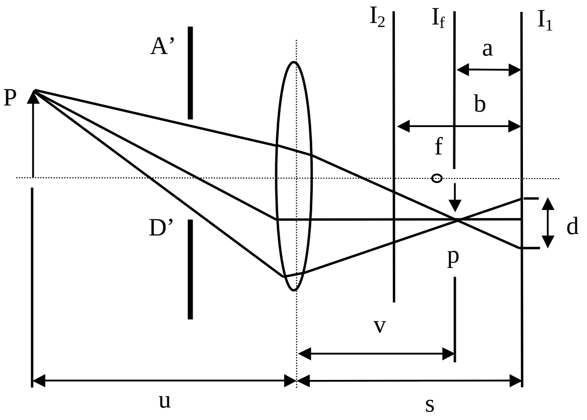


Figure 2.2. The image formation for telecentric case (from Watanabe and Nayar, 1995a).

Initially, for the current implementation a Computar 55mm telecentric lens (this lens is telecentric with respect to the view) was used, which eliminates the perspective distortions but the magnification errors were not significantly corrected. To overcome

[4] The focal length in front of the principal point of the lens (the centre of the lens).

this, using the procedure detailed in Watanabe and Nayar (1995a) an AF MICRO NIKKOR lens (*f* = 60mm) was transformed into a telecentric lens. The magnification errors were greatly reduced but since the diameter of the external aperture has to be very small (approximately 3mm) also resulted in a severe reduction in brightness. To compensate for this problem a very powerful source of light has to be used. This solution was dismissed because it is costly and difficult to be applied to robotic applications due to the size of the light generator. Consequently, for this implementation the magnification errors are minimised by using image interpolation as will be shown in Section 2.2.7.

2.2.3 The blur model

An image *g(x,y)* produced in a position other than the focal plane can be thought of as a processed version of the image *i(x,y)* obtained in the focal plane. From this observation, the blurring effect can be modelled as a convolution between the focused image and the blurring function.

$$g(x,y) = \int\int i(u,v)h(x-u, y-v)dudv \qquad (2.3)$$

where *i* is the focused image and *h* is the blurring function.

The blurring function, also called the *point spread function* (PSF), gives an indication regarding the amount of defocusing. It is clear that this function depends on the diameter *d* of the patch of each pixel obtained in the sensor plane. If only paraxial geometric optics is used (Subbarao and Surya, 1994), the PSF can be approximated in the spatial domain by simple ray tracing as follows:

$$h_p(x,y) = \begin{cases} \frac{4}{\pi d^2}, & x^2 + y^2 \le \frac{d^2}{4} \\ 0 & otherwise \end{cases} \qquad (2.4)$$

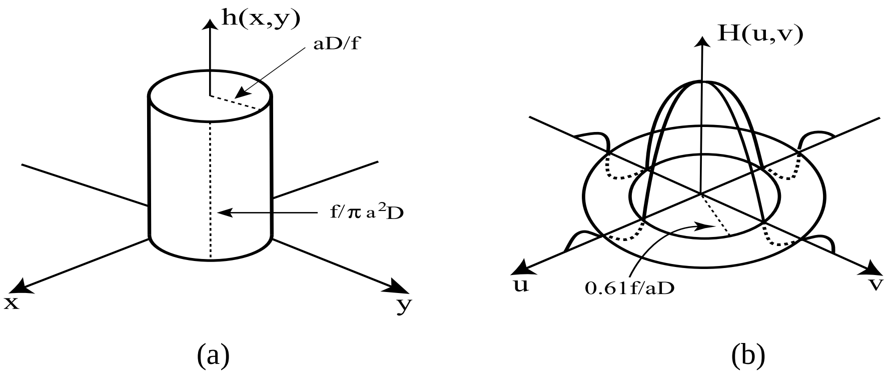
where h_p is the *pillbox* function and can be seen as the cone of light transmitted from the lens at the focal point. It is important to note that the relationship illustrated in Equation 2.4 defines the ideal case when the optical equipment (lens and sensing element) does not produce any supplementary blur. This issue is explicitly addressed in the paper by Nayar *et al* (1995) where the PSF is re-modelled by including some optical parameters in its expression. In Figure 2.1, a photometric analysis reveals that the light energy radiated by the scene and collected by the lens is uniformly distributed over a circular patch with a radius of *aD/f* on the sensor plane, where *a* is the distance between the focal plane I_f and the sensor plane I_1, *D* is the diameter of the aperture and *f* is the focal length. If these parameters are taken into account, the PSF becomes:

$$h(x,y) = h(x,y,a,D,f) = \frac{4f^2}{\pi D^2 a^2} \Pi(\frac{d}{Da}\sqrt{x^2+y^2}) \qquad (2.5)$$

where Π(*r*) is a rectangular function which has a value 1 for $|r| < \frac{1}{2}$, 0 otherwise. In the Fourier domain, the PSF is approximated by:

$$H(u,v) = H(u,v,a,D,f) = \frac{2f}{\pi Da\sqrt{u^2+v^2}} J_1(\frac{\pi Da}{f}\sqrt{u^2+v^2}) \qquad (2.6)$$

where J_1 is the first-order Bessel function. As is evident from the above expression, the PSF implements a low pass filter where the bandwidth of the filter increases as *a* decreases, or in other words, when the sensor plane is closer to the focus plane. This can be observed in Figure 2.4 where the high frequencies derived from the scene's texture are attenuated in accordance with the degree of blurring.



(a) (b)

Figure 2.3. The point spread function. (a) Representation in the spatial domain. (b) Representation in the frequency domain (from Nayar *et al*, 1995).

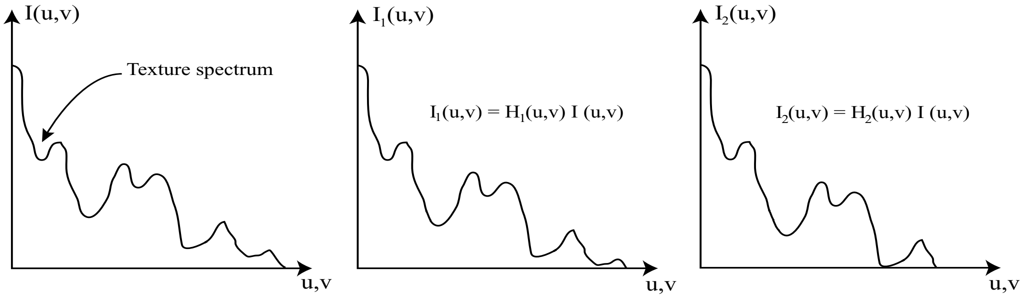


Figure 2.4. The blurring influence and the focus measure.

The blur model described above is valid only if the brightness is constant over the blur circle. As mentioned earlier this model defines the ideal case, but in practice due to the effects of diffraction, diffusion and other non-idealities such as polychromatic illumination and lens curvature (see also Pentland, 1987; Bhatia, 1996; Subbarao, 1988; Subbarao, 1991), the image blurring is not uniform within the blur circle and is better modelled by a two dimensional Gaussian:

$$h(x,y) = \frac{1}{\sqrt{2\pi\,\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.7}$$

where σ is the standard deviation of the Gaussian. The function shown in Equation 2.7 is rotationally symmetric and its expression in the Fourier domain is given by:

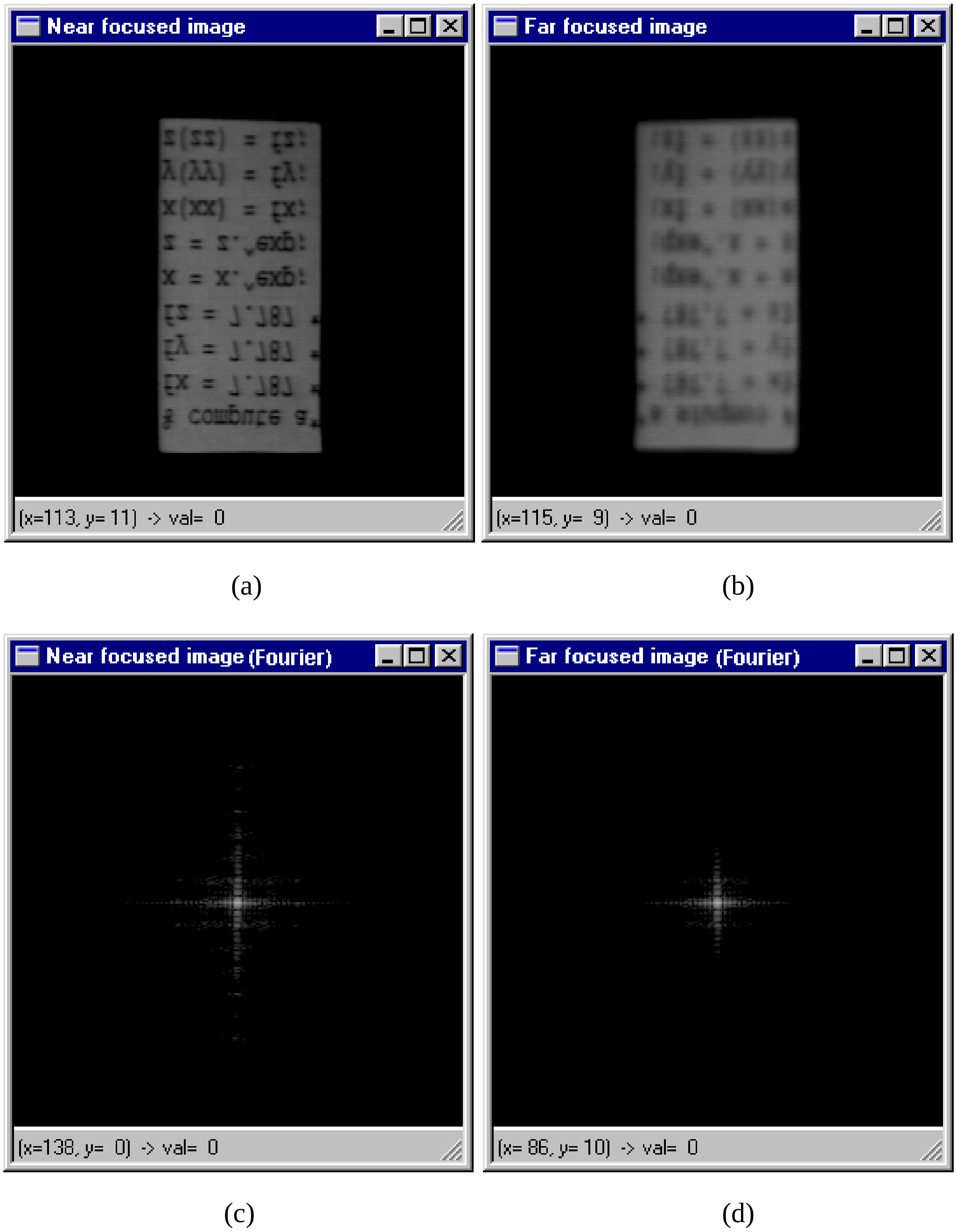
$$H(u,v) = e^{-\frac{u^2+v^2}{2}\sigma^2} \tag{2.8}$$

where (*u,v*) is the spatial frequency. The Gaussian is a low pass filter where the low frequencies are passed unaltered, whilst higher frequencies are reduced in amplitude (especially the frequencies above 1/σ). If σ increases, the result is an increased blurring as more of the higher frequencies are attenuated. Therefore, this parameter (also referred to as blur parameter) is of interest because it gives an indication about the amount of blurring contained in the image and can be estimated using the following relationship:

$$\sigma = k\,d\ ,\quad for\ \ k > 0 \tag{2.9}$$

where *k* is a constant of proportionality which is characteristic for each camera and in general can be determined by a calibration procedure (Subbarao, 1988).

Except the situations when σ is very small (when diffraction effects dominate), Equation 2.9 represents the actual situation and as a consequence it can be assumed that the blur parameter σ is proportional with *d*.



(x=113, y= 11) -> val= 0

(x=115, y= 9) -> val= 0

(x=138, y= 0) -> val= 0

(x= 86, y= 10) -> val= 0

(a) (b)

(c) (d)

Figure 2.5. The blurring effect. (a) The near focused image. (b) The far focused image. (c) The Fourier spectrum of the near focused image. (d) The Fourier spectrum of the far focused image.

To analyse the practical results associated with the blurring effect, let us consider the optical set-up described in Figure 2.1. When the sensor plane is moved with *a* we obtain the near focused image $i_1(x,y)$, which is the result of the convolution between the focused image $i(x,y)$ and the point spread function $h_1(x,y)$. In the frequency domain it is the product of the Fourier spectrum of the focused image $I(u,v)$ and the point spread function $H_1(u,v)$. When the sensor is placed at a distance *b-a* from the focal plane it can be observed that the Fourier spectrum of the far focused image $I_2(u,v)$ is altered to a greater extent. This result was expected because the distance *b-a* is greater than *a*. This can be observed in Figure 2.5.

Spatial domain *Fourier domain*

$$\begin{array}{ll} i_1(x,y) = i(x,y) \circ h_1(x,y) & I_1(u,v) = I(u,v)\,H_1(u,v) \\ i_2(x,y) = i(x,y) \circ h_2(x,y) & I_2(u,v) = I(u,v)\,H_2(u,v) \end{array}$$

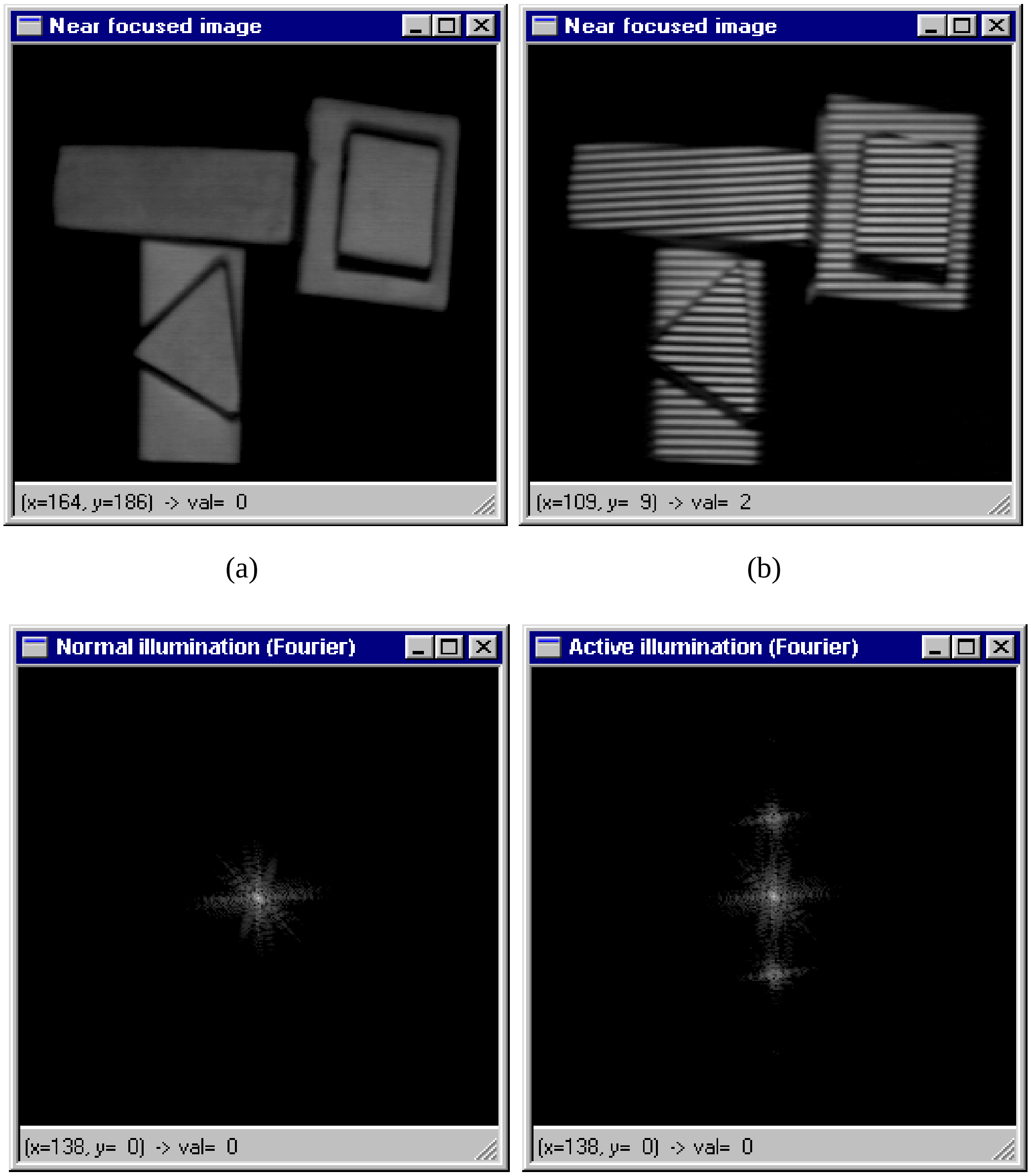
where $i(x,y)$ is the focused image, $\circ$ defines the convolution operator and $h_1(x,y)$ and $h_2(x,y)$ are the point spread functions for distances out of focus *a* and *b-a* respectively.

2.2.4 Active illumination

The high frequencies derived from the scene determine the accuracy of the depth estimation. If the scene has a weak texture or is textureless (like a blank sheet of plain paper) the depth recovery is far from accurate. Consequently, the applicability of passive DFD is restricted to scenes with high textural information.

To overcome this restriction, Pentland *et al* (1994) proposed to project a known pattern of light on the scene. As a result, an artificial texture is forced on the visible surfaces of the scene and the depth can be obtained by measuring the apparent blurring of the projected pattern. The illumination pattern was generated by a slide projector and selected in an arbitrary manner. In Figure 2.6-d it can be observed the textural frequency derived from an illumination pattern organised as a striped grid.

Later, Nayar *et al* (1995) developed a symmetrical pattern optimised for a specific camera. They used the assumption that the image sensor is organised as a rectangular grid. The optimisation procedure presented in the same paper, consists of a detailed Fourier analysis and the resulting model of illumination is a rectangular cell with uniform intensity which is repeated on a two dimensional grid to obtain a periodic pattern. The resulting pattern is very dense and difficult to fabricate and during experimentation it was found that the problems caused by a sub-optimal pattern are significantly alleviated when image interpolation was applied. This will be presented later.



(a) (b)

(c) (d)

Figure 2.6. Normal[5] illumination versus active illumination. (a) Image captured using normal illumination. (b) Image captured when active illumination is employed. (c) The Fourier spectrum of image (a). (d) The Fourier spectrum of image (b).

[5] Natural or ring type illumination.

2.2.5 The focus operator

Since the defocus function is a low pass filter, the effect is a suppression of high frequencies from the focused image. Therefore, to isolate the effect of blurring it is necessary to extract the high frequency information derived from the scene. Hence, the focus operator has to approximate a high pass filter. The goal of this operator is to estimate the blur parameter σ by inverse filtering the near and far focused images. Since the blur circle is uniform only for small regions, the kernel of the focus operator has to be small in order to preserve locality although the windowing operation introduces supplementary errors. To address this issue, Xiong and Shafer (1994) proposed a solution to select the window size for Gabor filters. They employed a simple criterion where the window size is selected to be as small as possible, while the error caused by noise and windowing is smaller than a preset value. Aside from window size, every focus operator must be rotationally symmetric and must not respond to any DC component (a DC component can be a change in image brightness). This condition is satisfied if the sum of all elements of the focus operator is equal to zero. Watanabe and Nayar (1995b) suggested an approach for passive DFD based on the use of rational filters. They proposed a method to compute a set of broadband rational operators. The first operator performs prefiltering (for removing DC components) and the remaining three operators are involved in depth estimation. Finally, the depth errors caused by spurious frequencies are minimised by applying a smoothing operator.

In the implementation described in this thesis, the experiments were conducted using the Laplacian operator. The kernels of the Laplacian operator are illustrated in Figure 2.7 (the 5 x 5 and 7 x 7 kernels are *ad hoc* approximations of the Laplacian operator and are constructed by analogy with the 3 x 3 kernel). It is well known that the Laplacian operator enhances the high frequency noise and this may cause significant errors when the depth is computed. In addition, supplementary errors are caused by quantisation and the misalignment between the cells of the sensing elements and the illumination pattern. To cope with these inconveniences, after the application of the focus operator, a 5 x 5 Gaussian is applied. Another aim of this operation is to minimise the error caused by local maxima which is due to surface reflections. The experimental data indicates that the resulting depth map is significantly smoother especially for scenes that contain specular objects.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(a) (b)

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 16 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

(c) (d)

$$\begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 20 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

(e) (f)

Figure 2.7. The Laplacian operator. (a), (b) The 3 x 3 kernel (4 and 8 neigbourhood). (c), (d) The 5 x 5 kernel (4 and 8 neighbourhood). (e), (f) The 7 x 7 kernel (4 and 8 neighbourhood).

2.2.6 Depth estimation from two images

The depth information can be estimated by taking a small number of images under different camera or optical settings. Since the PSF is a rotationally symmetric function, the relationship between the focused and defocused images is illustrated by the following expression (Subbarao and Surya, 1994).

$$f(x,y) = g(x,y) - \frac{\sigma^2}{4} \nabla^2 g(x,y) \qquad (2.10)$$

where *f* is the focused image, *g* is the defocused image, σ is the standard deviation of the PSF and ∇^2 is the Laplacian operator. Equation 2.10 represents the deconvolution formula. It expresses the focused image *f* in terms of the defocused image *g*, its derivatives and the blur parameter σ. If two images g_1 and g_2 are taken under different camera settings and the term *f*(*x*,*y*) is substituted, the result is the expression illustrated in Equation 2.11.

$$g_1(x,y) - g_2(x,y) = \frac{1}{4}(\sigma_1^2 - \sigma_2^2)\nabla^2 g, \qquad \nabla^2 g = \frac{\nabla^2 g_1 + \nabla^2 g_2}{2} \tag{2.11}$$

From Equation 2.11 it can be observed that no terms depend on scene's textural information. Since σ is proportional with the blur circle, the depth can be estimated using the difference between the standard deviation of the near focused image g_1 and the standard deviation of the far focused image g_2. As mentioned earlier, to determine the blur parameters it is necessary to filter the near and far focused images with the focus operator which gives an indication of the focus level.

Once this operation is accomplished, the next step consists of determining the depth from two images. Pentland suggested to attempt the depth recovery process by using the edge information returned by the focus operator. In this way, if a strong edge is returned, the corresponding point must be in focus (or very close). Whilst the edge is weak the point is out of focus. This can be observed in Figure 2.8 where the relation between the outputs of the focus operator and the depth estimation is illustrated.

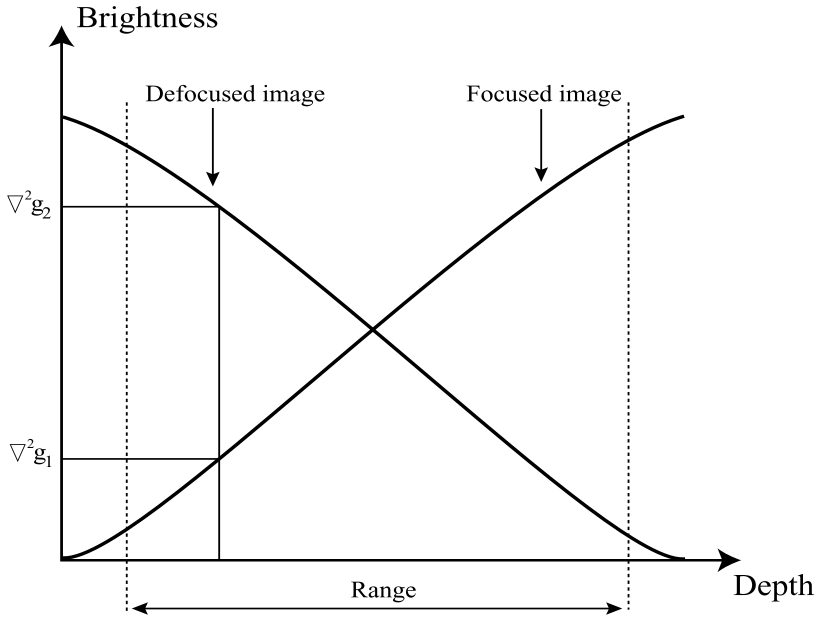


Figure 2.8. Estimating the depth from two images captured under different camera settings.

As Figure 2.8 illustrates, an effective solution to estimate the depth is to use the ratio between the $\nabla^2 g_1$ and $\nabla^2 g_2$. This ratio implements the defocus function and its profile is depicted in Figure 2.9. It is clear from this diagram that the defocus function is not bounded, but this fact is not a major drawback as long as the depth is investigated only for a restricted domain (see Figure 2.8). As this function has a linear profile, the ranging distance is estimated using the following relationship: $depth = \chi_1 \frac{\nabla^2 g_1}{\nabla^2 g_2} - \chi_2$, where χ_1 performs the gain correction and χ_2 eliminates the offset. These constants of proportionality are determined by calibration.

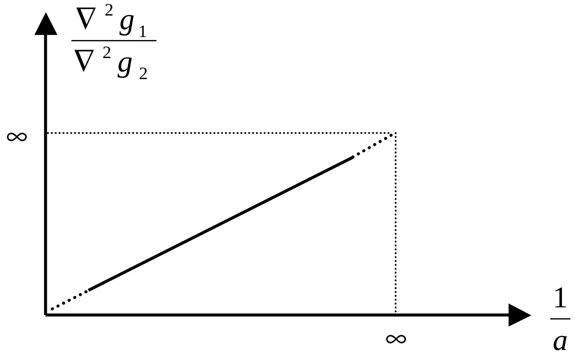


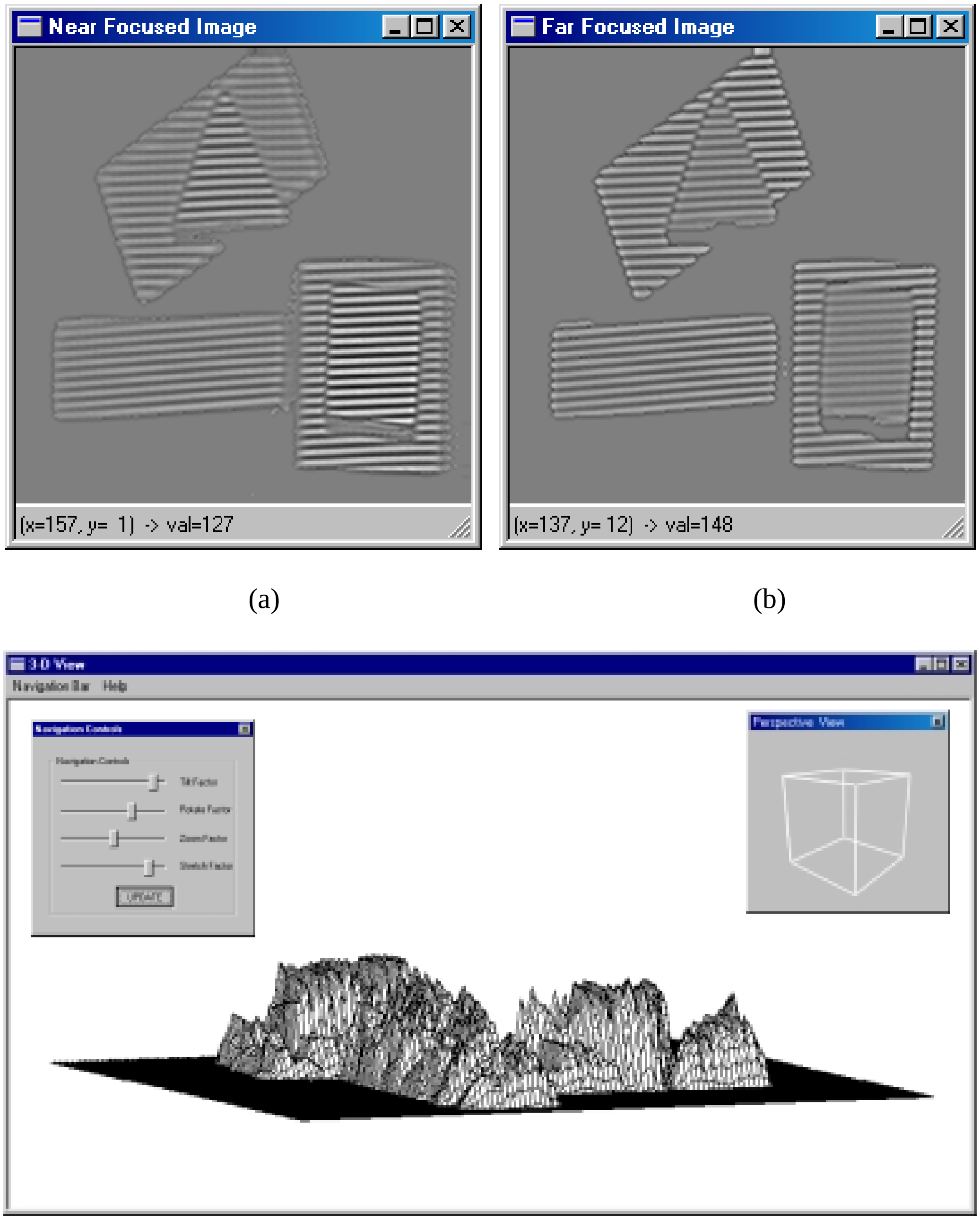
Figure 2.9. The defocus function.

2.2.7 Image interpolation

Since active illumination is employed, the depth estimation will have the same pattern as the structured light. Due to magnification changes between the near focused image and the far focused image, the stripes do not match perfectly together. As a consequence, the depth estimation is unreliable especially around the stripes borders. This can be observed in Figure 2.10 where the depth recovery is not continuous. Also, the errors caused by changes in magnification are evident.

To compensate for this problem, Watanabe and Nayar (1995a) proposed to use a telecentric lens. This solution is elegant and effective but since the telecentric lens requires a small external aperture, the illumination source necessary to image the scene has to be very powerful. To avoid this complication, for the present implementation the dark regions are mapped using image interpolation. Linear interpolation was found to be sufficient in the case where a dense (10 lines per mm) illumination pattern was used.

The effect of image interpolation is depicted in Figure 2.11 where the quality of the depth estimation is significantly improved.



(x=157, y= 1) -> val=127

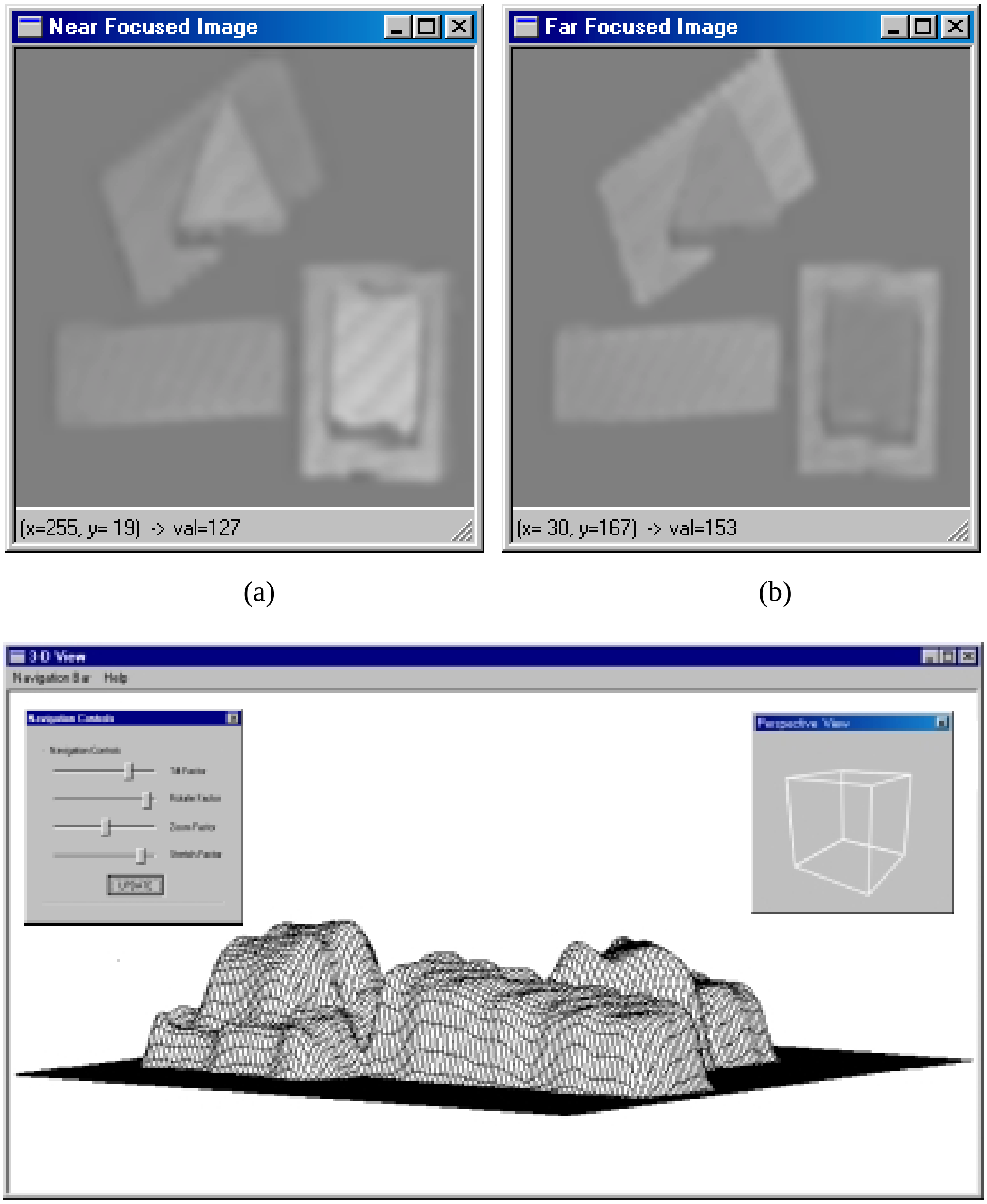
(x=137, y= 12) -> val=148

(a)

(b)

(c)

Figure 2.10. The near (a) and far (b) focused images resulting after the application of the focused operator. (c) The resulting depth map.



(x=255, y= 19) -> val=127

(x= 30, y=167) -> val=153

(a)

(b)

(c)

Figure 2.11. (a,b) The effect of interpolation when this operation is applied to the images illustrated in Figure 2.10 (a,b). (c) The resulting depth map.

2.2.8 Physical implementation

The aim of this implementation is to build a range sensor able to extract the depth information derived from dynamic scenes. Thus, the key issue is to capture the near and far focused images at the same time. For this purpose, two OFG VISION*plus* –

AT frame grabbers were utilised. The scene is imaged using an AF MICRO NIKKOR 60 mm F 2.8 (Nikon) lens. Between the NIKKOR lens and the sensing elements a 22 mm beam splitter cube is placed. The sensing elements used for this implementation are two low cost 256 x 256 VVL 1011C (VLSI Vision Ltd.) CMOS sensors. Nevertheless, the beam splitter introduced a supplementary distance between the CMOS sensors and the lens that image the scene. Furthermore, the distances added by the C-mount adapters (are used to attach the cameras to the beam splitter case) and the lens' mount further increase the distance between the CMOS sensors and the lens. As a result, the images projected on the sensors' active surface will be significantly out of focus. To overcome this problem, the camera head was opened and the first sensor was set in contact with the beam splitter inside the case. The second sensor was positioned with a small gap (approximately 0.8 mm) from the beam splitter surface using a multi-axis translator. The distance between the lens and the beam splitter is 1 mm. These settings offer a detectable ranging distance between 0 and 7 cm when the sensor is placed at a distance of 86 cm from the baseline of the workspace.

The structured light is projected on the scene using a MP-1000 Projector with a MGP-10 Moire gratings (stripes with density of 10 lines per mm). The lens attached to the light projector is the same type as that used to image the scene. Note that all equipment required by this implementation is low cost and furthermore the calibration procedure as outlined in Section 2.2.9 is relatively simple. The components required by this implementation and a diagram of the developed sensor are illustrated in Figures 2.12 and 2.13, while Figure 2.14 depicts the actual set-up.

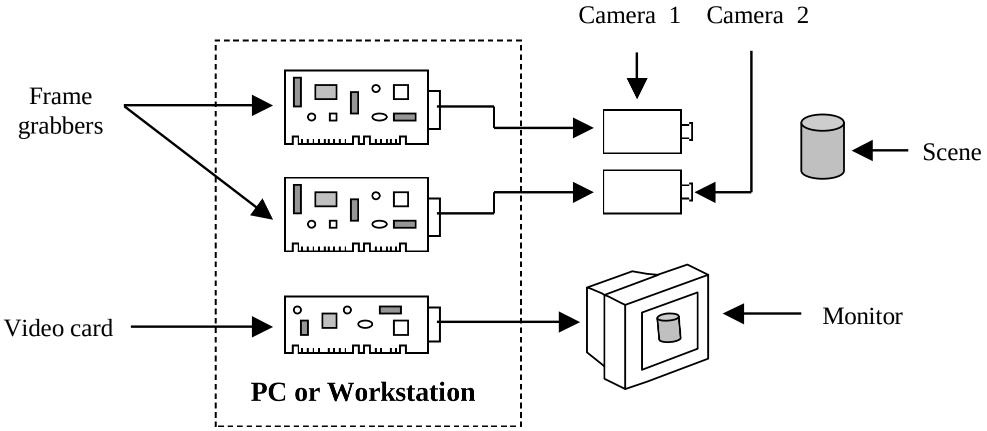


Figure 2.12. DFD system block diagram.

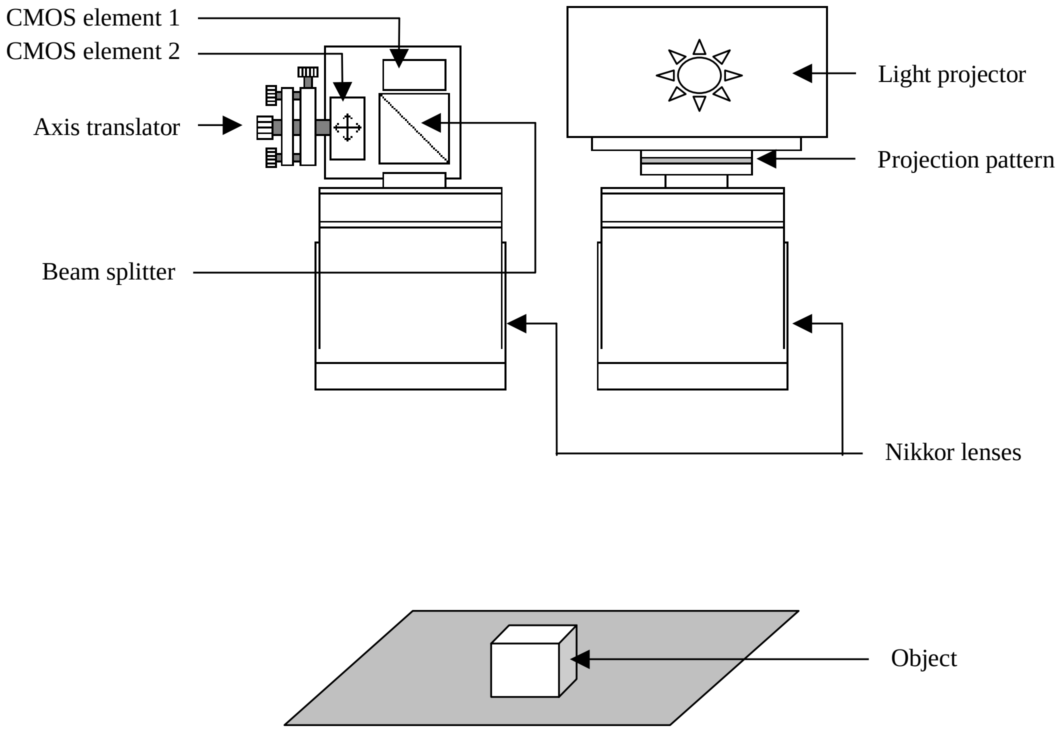
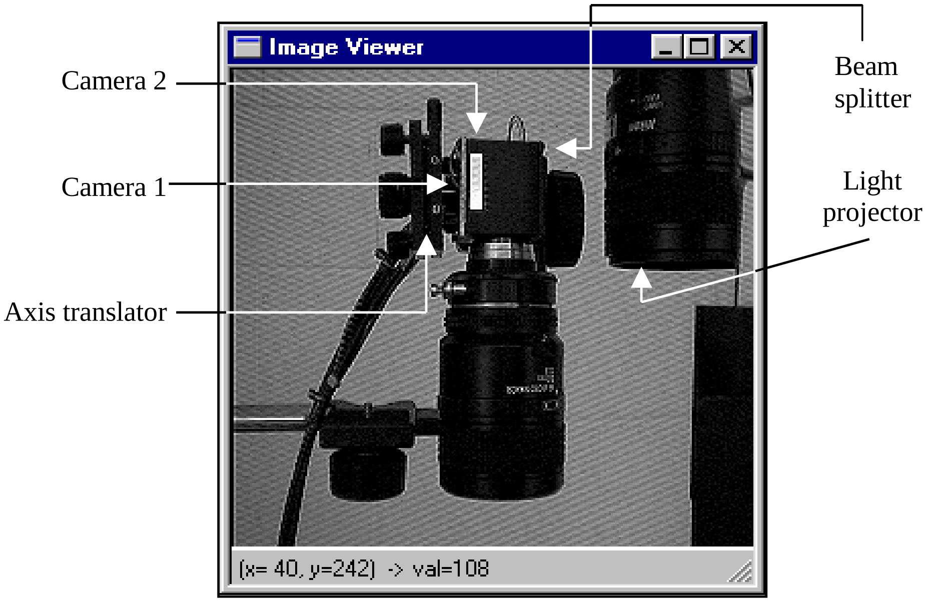


Figure 2.13. The diagram of the 3-D sensor.



(x= 40, y=242) -> val=108

Figure 2.14. The 3-D sensor and its principal components.

The software is simple as long as it includes only local operators. The flowchart illustrated in Figure 2.15 describes the main operations required to compute the depth map of a 256 x 256 resolution. The depth map is computed in 95 ms on a Pentium 133, 32 Mb RAM and running Windows 98.

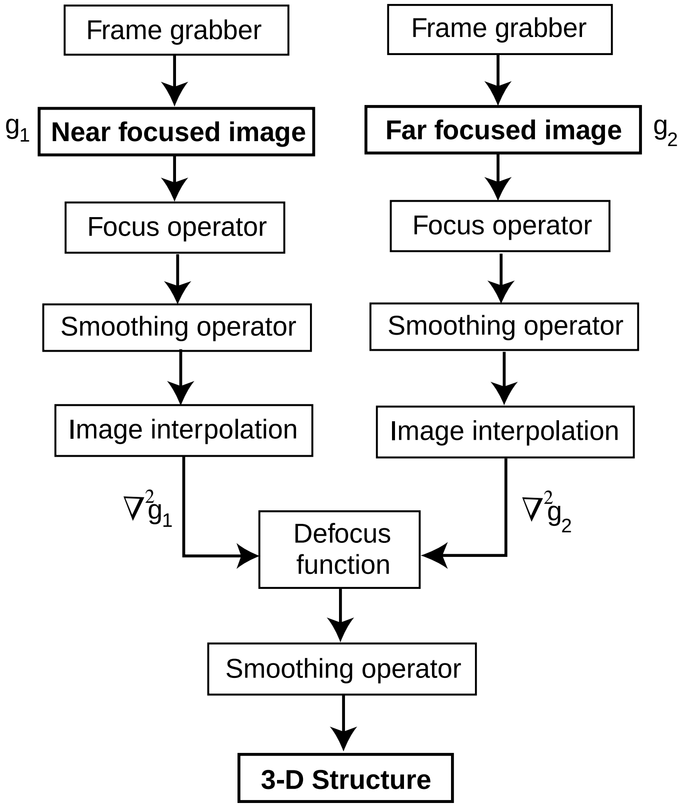
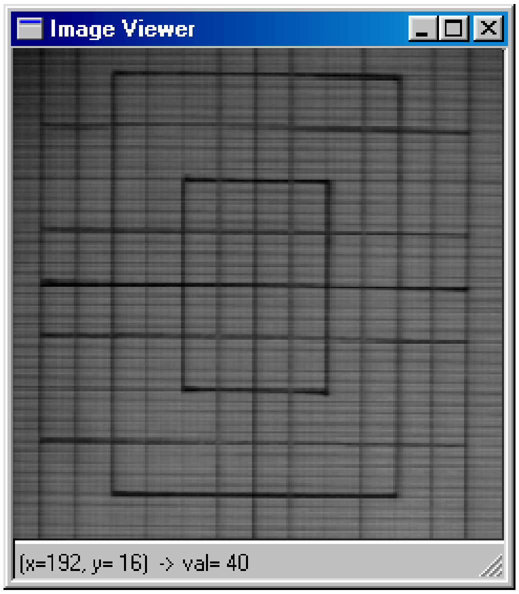


Figure 2.15. Data flow during the computation process.

2.2.9 Calibration procedure

Like for any other range sensor, the calibration procedure represents an important operation. This sensor requires a two-stage calibration procedure. The first stage involves obtaining a precise alignment between the near and the far focused sensing elements. To achieve this goal, the calibration is performed step by step using the multi-axis translator which is attached to one of the CMOS sensors. This procedure continues until the mis-registrations between the near and far focused images are smaller than the errors caused by changes in magnification due to different focal settings.

Because even a sub-pixel mis-registration may cause errors when the depth is computed, for the purpose of obtaining a precise alignment between the CMOS sensors, a grey level rectangular grid pattern is proposed as calibration pattern. This pattern is illustrated in Figure 2.16.



```
(x=192, y= 16)  -> val= 40
```

Figure 2.16. The calibration pattern.

The second step performs a pixel by pixel gain calibration in order to compensate for the errors caused by the imperfection of the optical equipment. This operation consists of the following procedure: a planar target is perpendicularly placed to the optical axis of the sensor at precise known distances. Then, the depth map is computed and the differences between the resulting depth values and the real ones are recorded for each elevation. Then, these depth errors were averaged and recorded in a table which defines the gain offset map. The gain compensation was carried out by subtracting the depth offset values from the detected depth map.

Nevertheless, this procedure holds only if the errors introduced by the optical equipment are linear. The experimental results proved that most of the errors were caused by image curvature, errors that are constant and easy to correct. Therefore, the proposed pixel by pixel gain calibration is an adequate procedure for this current implementation.

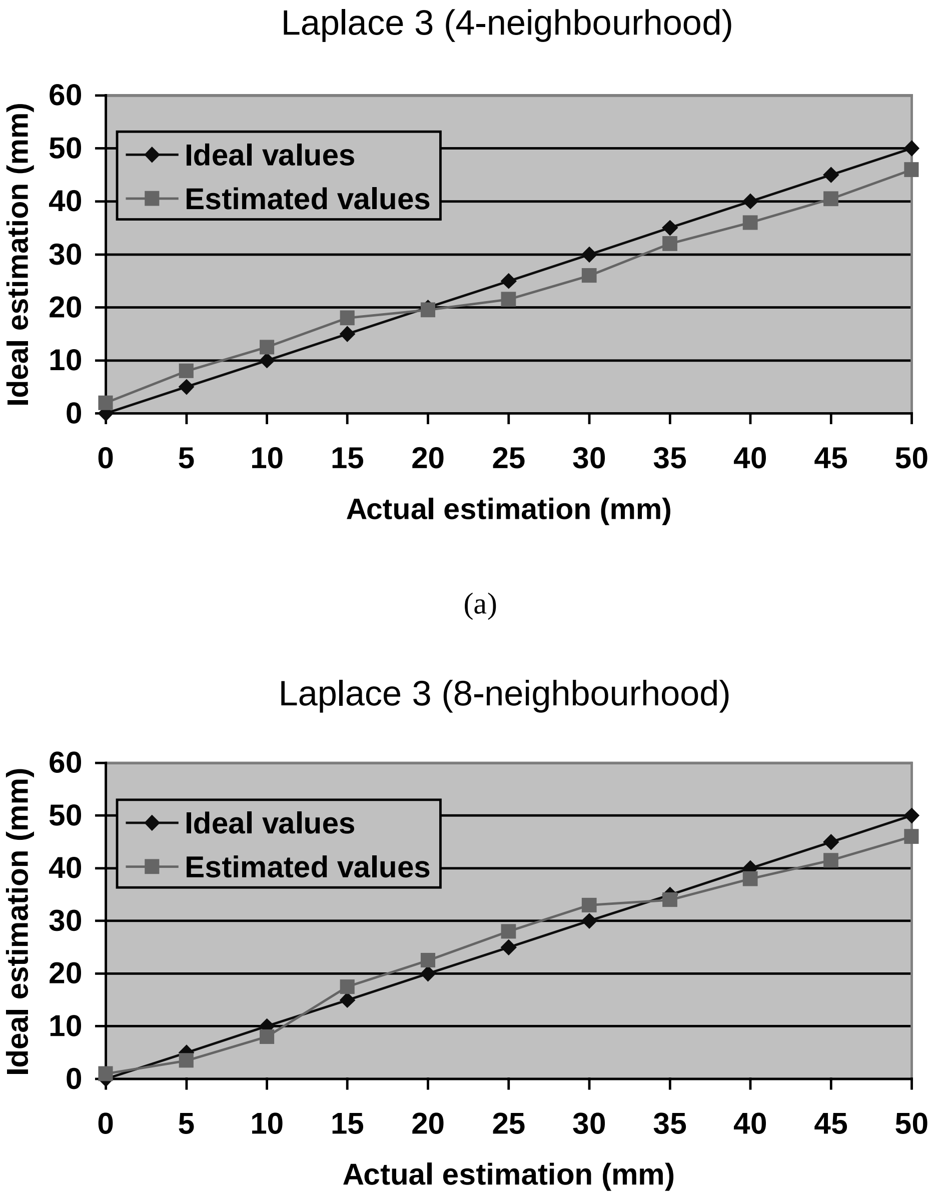
2.2.10 Experiments and results

The major problem concerning the usefulness of focal gradient information is whether this information is sufficiently accurate. This information can be accurate only if the optical settings are well known and are used as parameters when the depth is computed. Due to the fact that active illumination is used, the following experiments were carried out exclusively on indoor scenes.

Initially, this sensor was evaluated using simple targets in order to verify the accuracy of the implemented range sensor. The accuracy and linearity is estimated when the sensor is placed at a distance of 86 cm from the baseline of the workspace.

The reported results were obtained for a planar textureless test object i.e. a plain sheet of paper. To determine the sensor's accuracy, the test object was placed at several distances from the workspace baseline. These distances were measured using a simple ruler and are marked in the graphs illustrated in Figures 2.17 and 2.18 as ideal values. The actual estimation for each elevation was obtained by averaging the depth values contained in a test area which was obtained by sub-sampling the depth map to a 32 x 32 pixels area. Finally, the actual estimations are plotted against the ideal values. It should be noted that the first point (marked with zero) in the graphs shown in Figures 2.17 and 2.18 represents the calibration plane.

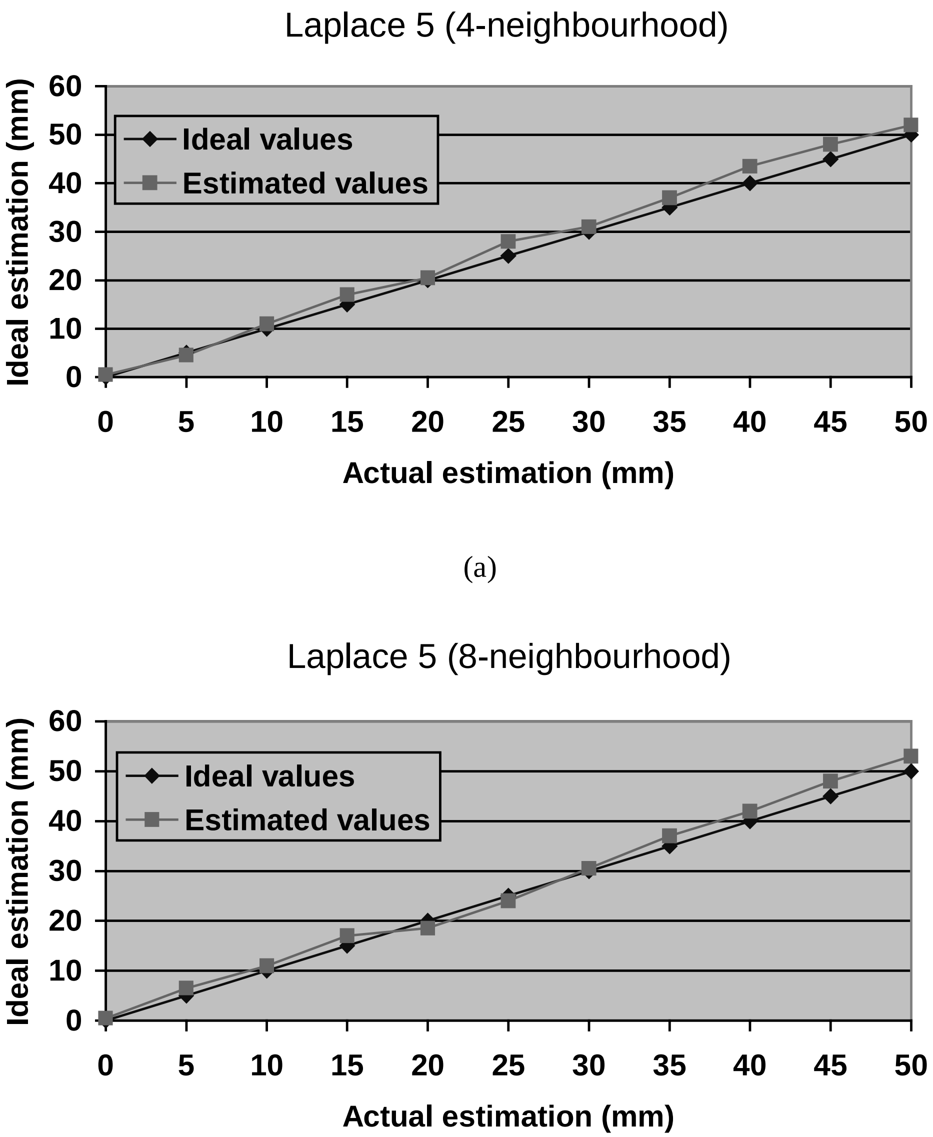
The results depicted in Figures 2.17 and 2.18 need to be further discussed. During experimentation it was found that most of the errors were caused by the sensing elements. Apart from quantisation noise, the most difficult problems were generated by the image intensity offset. The level of offset is non-uniform and is dependent upon the brightness distribution contained in the image and furthermore is different from sensor to sensor. Nevertheless, the intensity offset generates significant errors when the depth is computed. On the other hand, the offset compensation causes loss of information that may also lead to imprecise depth estimation. As can be easily observed, an ideal solution to this problem is not possible and to address this problem efficiently a procedure similar to that utilised for gain calibration is employed. Thus, the offset errors were minimised as follows: a number of target objects with different colors were utilised and for each CMOS sensor the errors caused by the offset were averaged and recorded in a table. These tables implement the offset map for each sensor. The offset compensation was carried out by subtracting the offset map from the captured image according to the sensor in question. Although this procedure does not completely eliminate the offset, the experimental results illustrated in Figures 2.17 and 2.18 indicate that the accuracy of this proposed sensor compares well with that offered by other methods such as stereo and motion parallax.



(a)

(b)

Figure 2.17. The accuracy and linearity when the 3 x 3 Laplacian operator is applied. (a) 4-neighbourhood. (b) 8-neighbourhood.

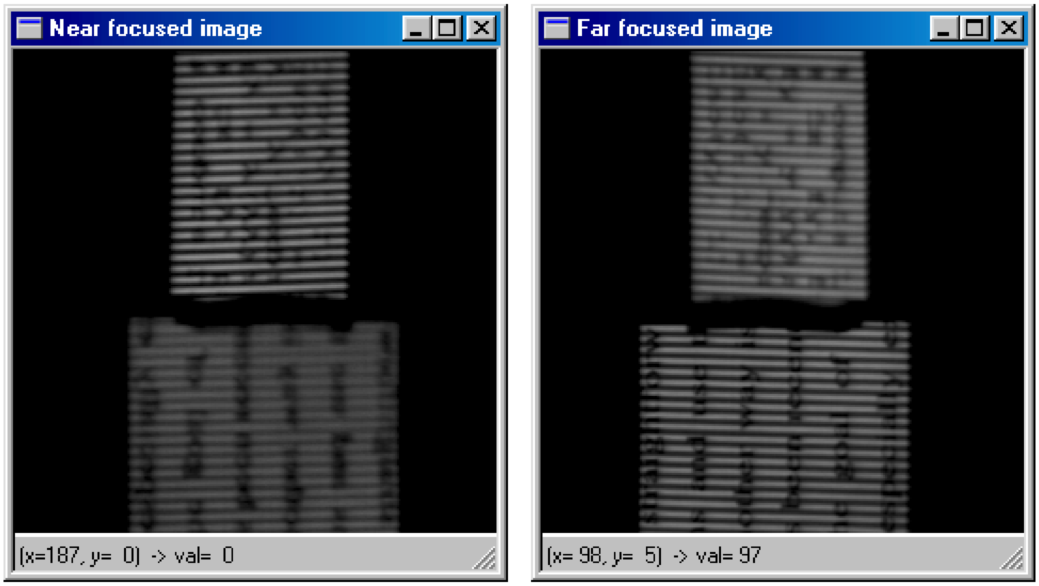


(a)

(b)

Figure 2.18. The accuracy and linearity when the 5 x 5 Laplacian operator is applied. (a) 4-neighbourhood. (b) 8-neighbourhood.

Next, the performance of the range sensor was evaluated using complex scenes. Figure 2.20 illustrates the depth recovery for two textured planar objects (see Figure 2.19) situated at different distances from the sensor.



(x=187, y= 0) -> val= 0

(x= 98, y= 5) -> val= 97

(a) (b)

Figure 2.19. The near (a) and far (b) focused images for a scene which contains two planar objects situated at different distances from the sensor.

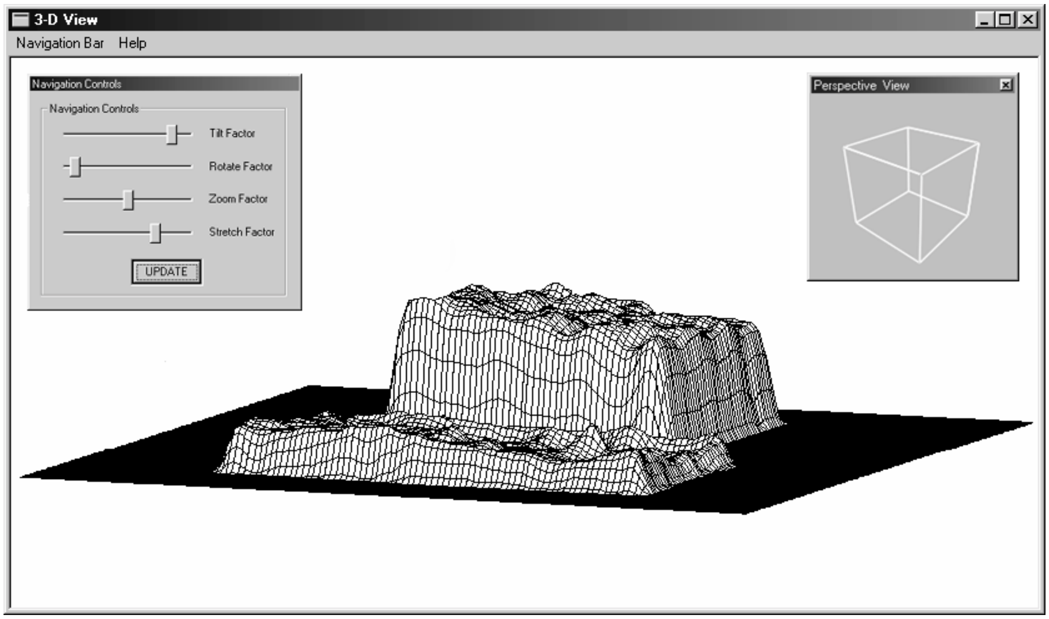
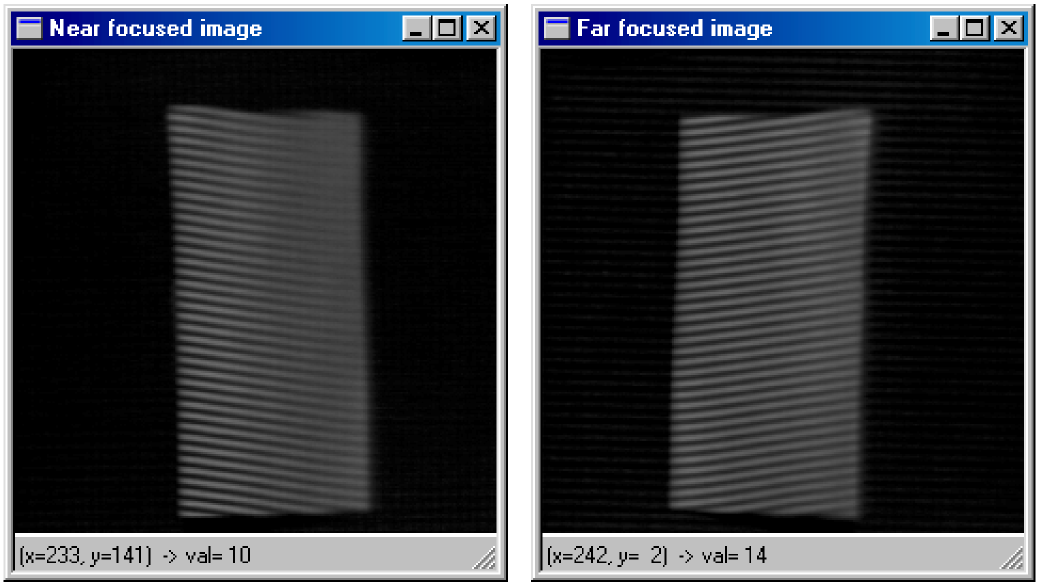


Figure 2.20. The depth estimation for the two planar objects illustrated in Figure 2.19 situated at different distances from the sensor.

Figure 2.22 illustrates the depth map for a slanted planar object and in Figure 2.24 a more complex scene containing textureless objects with different shape is shown.



```
(x=233, y=141) -> val= 10
```

```
(x=242, y= 2) -> val= 14
```

(a) (b)

Figure 2.21. The near (a) and far (b) focused images for a scene which contains a slanted planar object.

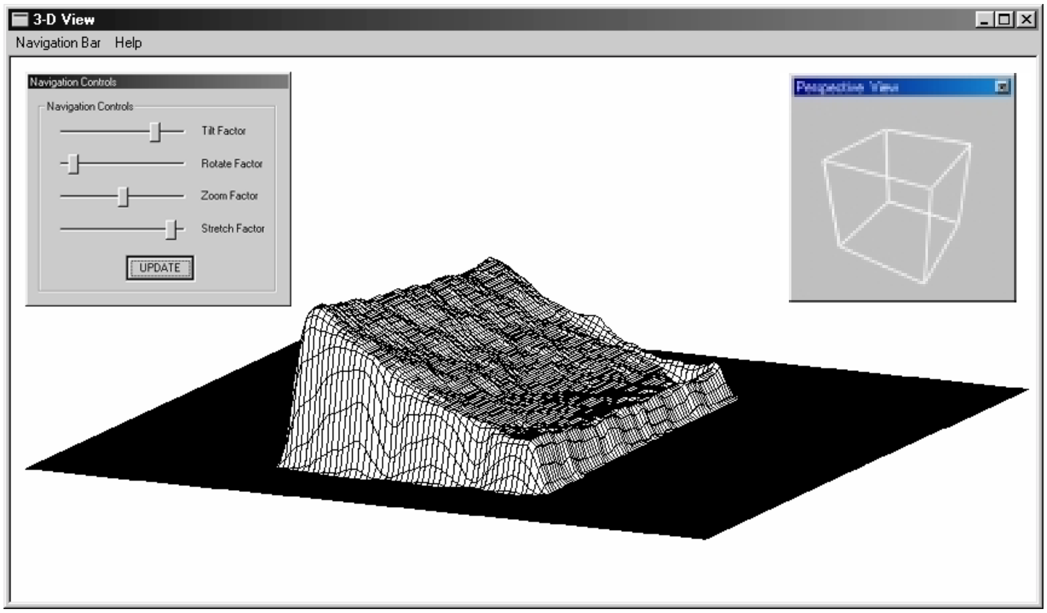
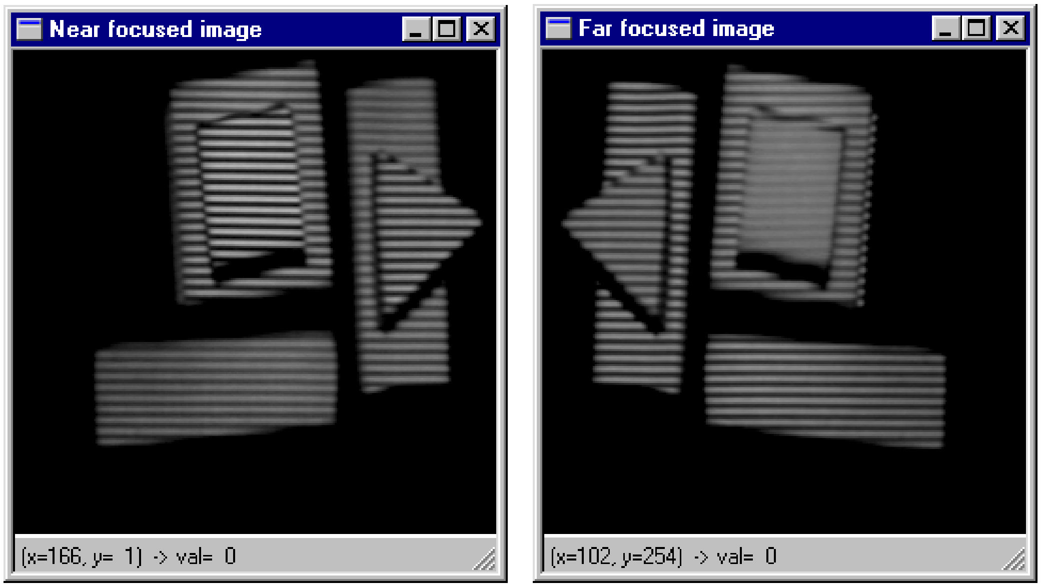


Figure 2.22. The depth estimation for the scene illustrated in Figure 2.21 which contains a slanted planar object.



(x=166, y= 1) -> val= 0

(x=102, y=254) -> val= 0

(a)

(b)

Figure 2.23. The near (a) and far (b) focused images for a scene which contains various textureless objects.

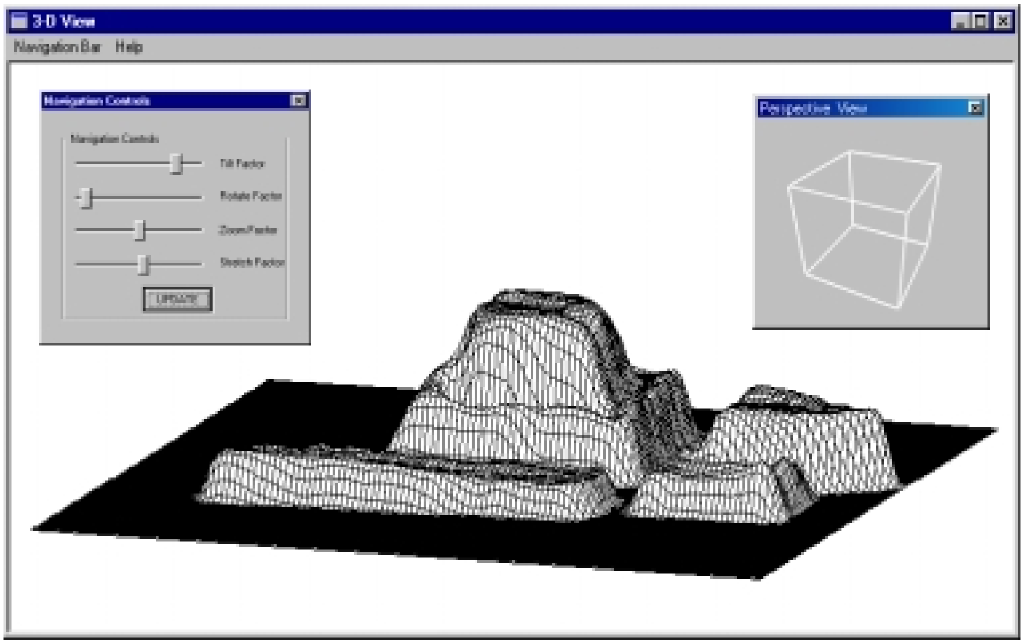
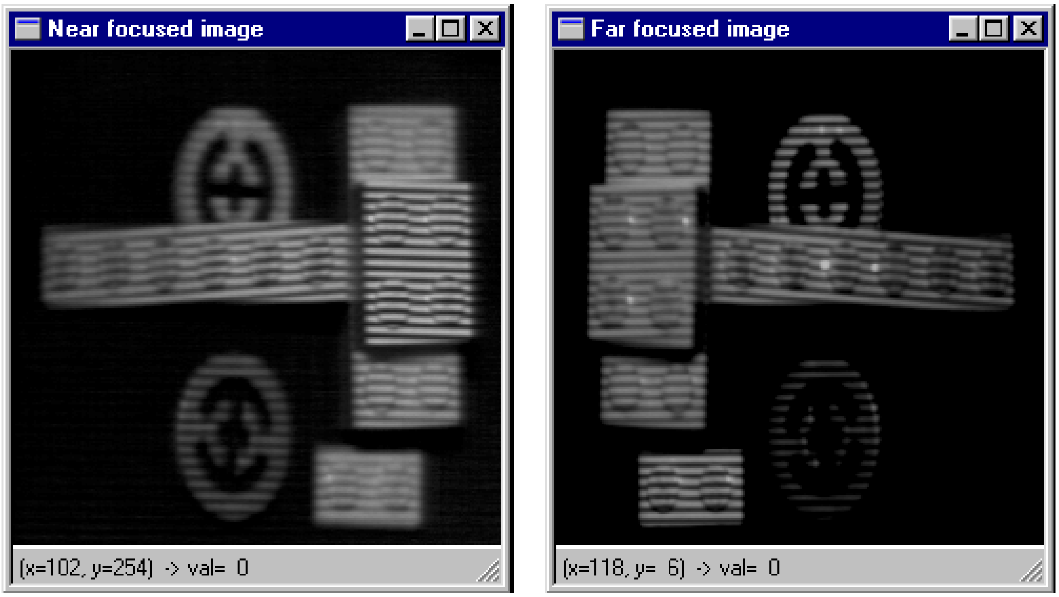


Figure 2.24. The depth estimation for the scene illustrated in Figure 2.23 which contains various textureless objects (a post-smoothing operation has been applied).



(x=102, y=254) -> val= 0

(x=118, y= 6) -> val= 0

(a)

(b)

Figure 2.25. The near (a) and far (b) focused images for a scene which contains various LEGO® objects.

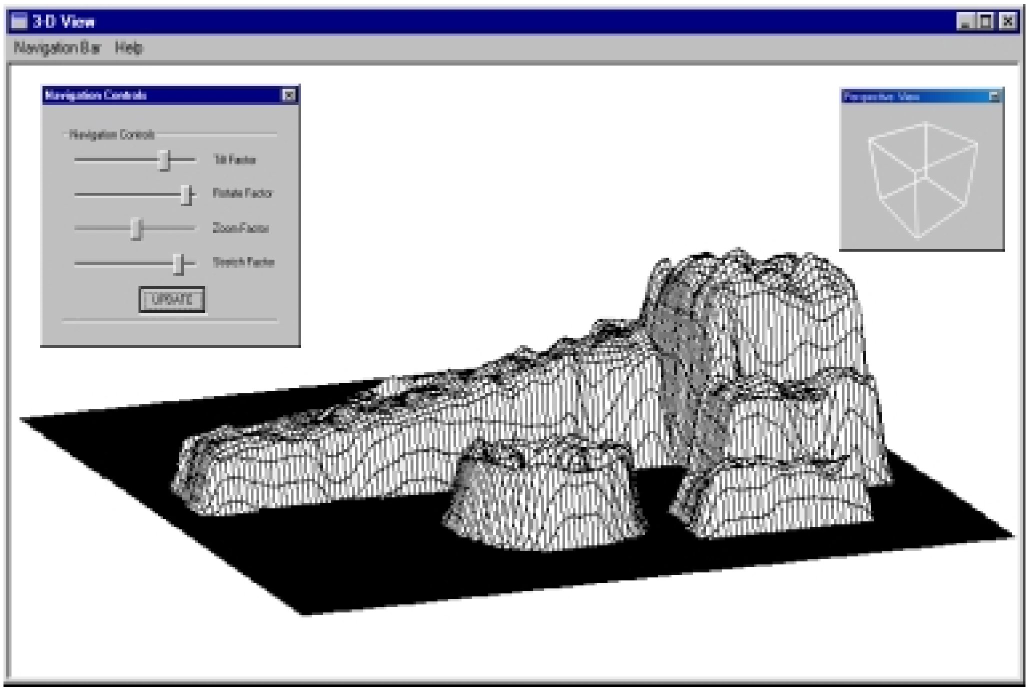
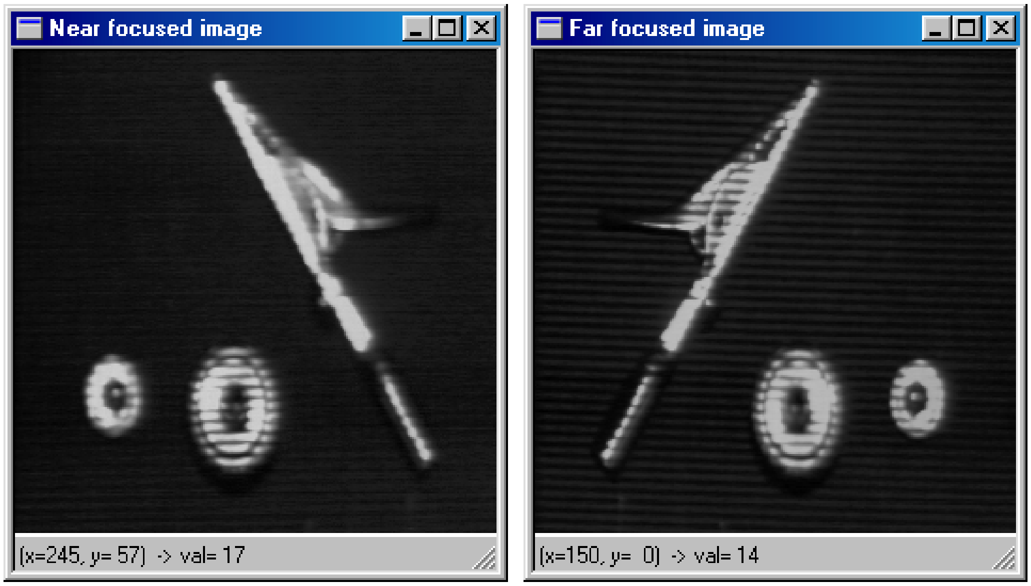


Figure 2.26. The depth estimation for the scene illustrated in Figure 2.25 which contains various LEGO® objects (a post-smoothing operation has been applied).



(a) (b)

Figure 2.27. The near (a) and far (b) focused images for a scene which contains objects with prominent specular characteristics.

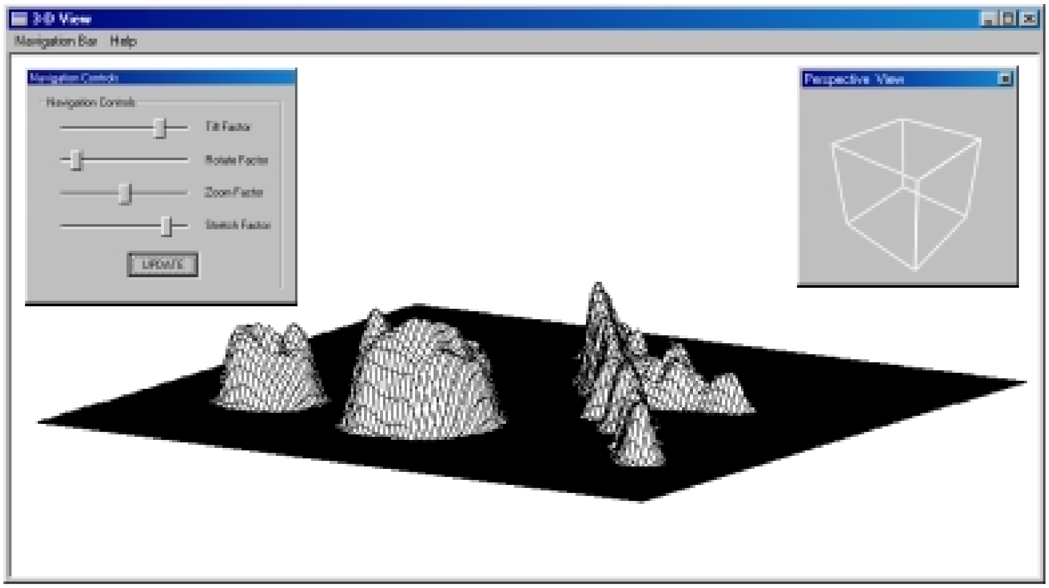


Figure 2.28. The depth estimation for the scene illustrated in Figure 2.27 which contains objects with prominent specular characteristics (a post-smoothing operation has been applied).

Figure 2.26 depicts the depth map for a scene which contains mildly specular LEGO® objects with different shapes and a large scale of colours. Figure 2.28 shows the depth recovery for a scene which contains objects with pronounced specular characteristics. For the last scenes, in order to reduce the errors caused by the reflections associated with specular surfaces the depth map is smoothed by applying a 5 x 5 Gaussian operator.

One of the aims of this section was to identify an optimal solution for the focus operator. As mentioned in Section 2.2.5, six focus operators were used. The best results with respect to the gain were obtained for a 7 x 7 (8 neighbourhood) Laplacian operator but the depth estimation was not very linear. The results were more linear when the 3 x 3 Laplacian operator was used as focus operator but the discontinuities in depth were not as well recovered. A trade-off between gain and linearity was given by the 5 x 5 Laplacian (8 neighbourhood) operator.

The relative accuracy and repeatability of the sensor are estimated relative to the overall ranging distance. Accuracy is measured by the standard deviation of the depth values contained in a 32 x 32 pixels area obtained by sub-sampling the depth map, while repeatability is determined by the standard deviation of the depth values measured at the same position at different times. The results depicted in Table 2.1 are reported for both textured and textureless planar objects when the 5 x 5 Laplacian 8-neighbourhood was used as focus operator.

Depth accuracy – max error (%)	3.4
Depth accuracy – standard deviation	2.62
Repeatability – standard deviation	1.46
Depth map area (pixels)	256 x 256
Test area (pixels)	32 x 32
Number of cycles	50
Delay between two successive cycles (sec)	5

Table 2.1. The accuracy and repeatability parameters of the developed range sensor.

For scenes containing non-specular objects (with elevations ranging between 0.7 and 2.5 cm) the error rate is 3.4% of the overall ranging distance from the sensor. When the scene contains objects with specular properties the accuracy is affected in relation to the degree of specularity. This can be observed in Figure 2.26 when some fine details such as the bumps on the LEGO blocks are not always accurately recovered.

The results were found to be very encouraging and the recovered shape is precise enough for a large variety of visual applications including object recognition and advanced inspection.

2.3 Discussion

This thesis was authored with the specific intention of exploring active DFD and in order to expand this topic, Appendix A surveys a variety of range sensing techniques and presents the problems associated with them. From this survey, it can be concluded that all these techniques have problems and limitations. For instance, passive techniques such as stereo and motion parallax are mostly used for outdoor scenes where the depth discontinuities are significant. In contrast, active techniques such as depth from defocus or methods based on triangulation are employed when the objects are situated nearby.

This chapter has focused on describing the implementation of a real-time bifocal range sensor. Since the depth is estimated by measuring the relative blurring between two images captured with different focal settings, this approach in contrast with methods such as stereo and motion parallax is not restricted by problems such as detecting the correspondence between features contained in a sequence of images or missing parts. Also it is worth mentioning that DFD offers the possibility of obtaining real-time depth estimation at a low cost.

Active DFD is preferred in many applications because it can reliably estimate the depth even in cases when the scene is textureless. However, accurate depth estimation requires practical solutions to a variety of problems including active illumination which was identified to be the key issue for this approach, optical and sensing equipment and the physical implementation.

In contrast with other implementations based on defocusing where the depth range is relatively large, the current implementation estimates the depth within a small range (between 0-7 cm). In addition, the current approach has another advantage over other implementations suggested by Pentland *et al* (1994) and Nayar *et al* (1995) because it does not contain any sensitive equipment to movements or vibrations, therefore it can be easily utilised in robotics applications. The consistency between theory and experimental results has indicated that the implementation outlined in this chapter is an attractive solution to estimate the depth quickly and accurately.

Chapter 3 - Image segmentation

3.1 Introduction

Accurate image segmentation is one of the key issues in computer and machine vision. The aim of the segmentation process is to decompose the image into disjointed meaningful regions that have strong correlation with the objects from the real world. The segmentation is *complete* when the objects from the input image are completely described by computed regions, or *partial* when the objects cannot be directly represented by these regions. Achieving complete segmentation using only simple algorithms is difficult because the regions contained in the image are not homogenous. Sometimes, it is better to achieve partial segmentation and then using some properties such as brightness, colour, texture etc., the segmentation process can be improved by further processing. In other words, a reasonable way to obtain complete segmentation is to use the partial segmentation results as input to higher level processing.

The range of literature on image segmentation and clustering is extensive. Many authors consider that the segmentation techniques can be divided into different categories according to the feature in question. The segmentation techniques included in the first category use only local features that describe the image content (pixel intensities, histograms etc.). The second category is represented by edge-based segmentation techniques. For these methods the quality of segmentation is given by the precision of the edge detector involved. The third category includes region-based segmentation techniques and commonly these methods are based on region growing algorithms.

Depending on the availability of *a priori* information describing the image content, the segmentation techniques can be divided into *supervised* and *unsupervised*. In general, the segmentation process is unsupervised since no *a priori* information about the number or type of regions is available. The supervised segmentation is used when we *have knowledge* about the scene, a typical example is represented by inspection systems.

Many segmentation techniques have been developed only for 2-D images. For some particular applications such as robotic bin picking where the 3-D information is required by the robot to identify the position and orientation of the objects (refer to Chapter 1), a practical approach is represented by edge-based segmentation. This approach is suggested by the observation that edges are determined in general by changes of the geometrical properties in the scene. The most relevant segmentation techniques will be briefly discussed, with emphasis on the edge–based approach that is the topic of the implementation outlined in this thesis.

3.2 Region growing segmentation

The edge-based segmentation technique tries to exploit the dissimilarities between the regions existent in the image and consequently this approach is efficient if the borders between different regions are well defined. If the scene is highly textured or the image is affected by noise, the edge structures become very complex and it is very difficult to determine the boundaries of the objects contained in the scene.

Therefore, region growing techniques are usually better suited when the meaningful edges are difficult to detect. As opposed to edge-based approaches, these segmentation techniques try to identify the regions with the same properties. Generally, these techniques consist of two stages: the split and the merge. The first stage (the splitting stage) divides the image into initial regions with maximum homogeneity of properties. Then, two or more adjacent regions will join together if the conditions established by the merge criterion are upheld.

The simplest algorithm for region growing considers that each pixel contained by the input image represents an initial region. Next, the algorithm starts the merging stage using regions of 2 x 2, 4 x 4 and so on. The merging stage is finished when there are no changes between the computed regions. This segmentation algorithm known as *single-linkage region growing* (Haralick and Shapiro, 1992) is attractive due to its simplicity but is not very accurate (especially when the image is corrupted by noise).

More advanced segmentation techniques rely on split and merge algorithms. The algorithm starts with splitting the image sequentially into sub-regions. Some resulted regions may be homogenous during this process and will not split any more. It is worth mentioning that the resulting output after the application of the splitting process *does not* represent a segmented image. This output represents the input for the merging process that uses *different criteria* with respect to the region's homogeneity. In general, the split and merge algorithms are organised as segmentation trees where the nodes represent regions while the leaves describe sub-regions. The quality of these algorithms is dictated by the homogeneity criteria employed. If the input image is reasonably simple, a split and merge approach can use only local properties. If the image is complex, an approach that uses only local information may not give acceptable results and more consistent properties have to be considered.

Along with the edge-base and region growing segmentation techniques that use only a single image as an input, for motion-based segmentation techniques the input data is represented by a sequence of images. The aim of the motion segmentation is to separate the objects of differing velocity contained by the scene. Next, the surfaces resulting after motion segmentation can be further segmented into regions by other segmentation techniques. A prominent example of this approach is the ASSET-2 system developed at Oxford University by Smith and Brady (1995).

3.3 Edge-based segmentation

Edge-based segmentation uses the information returned by edge detecting operators. Typically, an edge can be seen as a discontinuity in grey (or colour) levels. Also, edges are determined by abrupt changes in depth structure, a situation where they are related to the geometrical properties of the scene. Therefore, depending on the representation of the input data, the segmentation process can be approached using 2-D or 3-D primitives. For some applications the 3-D information is not available and consequently the segmentation algorithm deals with raw images (grey scale or colour). Other applications such as robotic bin picking or robot navigation and obstacle avoidance require 3-D analysis, thus a realistic way to approach the segmentation process relies on the use of range images. The accuracy of the range sensor and the geometrical properties of the scene will have a great influence on the overall segmentation results. There is no doubt that this approach will be highly successful if the relative depth between the objects is significant. In contrast, if the scene contains planar objects which are in contact, hence situated at the same elevation, the range image will contain very little information and the results returned by the segmentation algorithm are in this case unreliable.

Sometimes, better results are obtained if this problem is addressed using raw images and the 3-D analysis is employed only to estimate the object's orientation (for more details refer to Section 3.6.3).

The key issue associated with the edge-based segmentation (independent of the representation of the input data) is the choice of edge operator. The precision of this segmentation technique is highly dependent on the quality of the edge operator used. Ideally, every object should be represented by a closed region, but unfortunately this situation is very difficult to achieve using only the edge information. Therefore, to obtain a meaningful segmentation further processing that takes into consideration the local information has to be performed.

The most common problems related to edge-based segmentation are caused by image noise and the small changes in the grey level distribution. Certainly, these issues may have a negative effect on segmentation results and an optimal solution that compensates for these problems can vary from case to case.

3.3.1 Edge detectors

The successful detection of edge information in an image is an important precursor to many image processing and analysis operations. Since edges are determined by sharp changes in the grey level transitions, their extraction entails a two-stage process. Initially the edges are enhanced using partial derivatives, then, the edge detected output is analysed in order to decide whether a particular pixel is an edge or not. Based on this concept the following two types of detection operators are introduced. The first category includes the gradient operators and the second category evaluates the zero crossings of the image second derivative. The first derivative of an image *f*(*x*,*y*) is defined by the expression illustrated in Equation 3.1.

$$\nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y} \qquad (3.1)$$

The abruptness of the edge is given by the *magnitude* of the gradient which is illustrated in Equation 3.2 and the *direction* of maximal grey-level change can be evaluated using the relationship presented in Equation 3.3.

$$\left|\nabla f(x,y)\right| = \sqrt{\left(\frac{\partial\, f(x,y)}{\partial\, x}\right)^2 + \left(\frac{\partial\, f(x,y)}{\partial\, y}\right)^2} \qquad (3.2)$$

$$\vartheta = \arg\left(\frac{\partial\, f(x,y)}{\partial\, x}, \frac{\partial\, f(x,y)}{\partial\, y}\right) \qquad (3.3)$$

It can be noticed that the edge operators that use the first derivatives do not present the same properties in all directions. Usually, these operators consist of a pair of masks which measure the gradient in two orthogonal directions. In contrast, the second derivative operator (also known as Laplacian) presents the same properties for all directions (rotationally invariant) and is defined by the relationship illustrated in Equation 3.4.

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial\, x^2} + \frac{\partial^2 f(x,y)}{\partial\, y^2} \qquad (3.4)$$

An analysis of Equations 3.2 and 3.4 reveals that the magnitude of the first derivative and the profile of the second derivative are given by the profile of the edge.

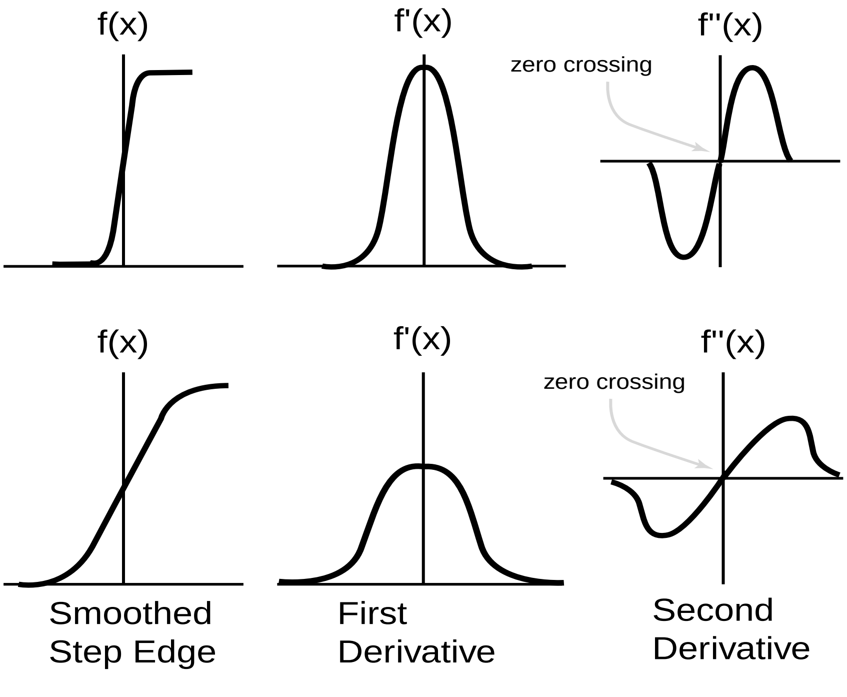


Figure 3.1. A 1-D edge and its derivative profiles (from Sonka *et al*, 1993).

To support this observation, Figure 3.1 depicts the edge profiles for the original image, first derivative and second derivative in two situations: the first for a step-like edge and the second for a smoother edge profile. As can be easily observed, the maximum values for first and second derivatives are obtained for a step-like edge profile. The next section will introduce some operators based on the evaluation of the first and the second derivatives.

3.3.2 Gradient operators

For a discrete image, the gradient can be calculated by simply computing the difference of grey values between adjacent pixels. The edge operators are described by a collection of masks (kernels) which measure the gradient for certain directions.

The simplest gradient edge detector is the Roberts cross operator. It has a pair of 2 x 2 neighbourhood masks which compute the first derivatives in two orthogonal directions. The convolution masks of the Roberts operator are:

$$H_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The gradient’s magnitude and orientation are given by next expressions:

$$Mag(x,y) = \sqrt{(f(x,y) \circ H_1)^2 + (f(x,y) \circ H_2)^2} \tag{3.5}$$

$$\vartheta = \tan^{-1}\left(\frac{f(x,y) \circ H_2}{f(x,y) \circ H_1} \right) \tag{3.6}$$

where *f(x,y)* is the input image and $\circ$ represents the convolution operator.

This operator is very convenient to be used because it is simple and fast. The main disadvantage is the sensitivity to noise because the masks use only few pixels to compute the first derivative. The Prewitt operator gives a 3 x 3 approximation to gradient and its convolution masks for *x* and *y* directions are:

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

The Sobel operator is a version of Prewitt operator and its convolution masks are illustrated below.

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The Prewitt and Sobel operators determine the vertical and horizontal edge components. Another popular gradient operator is the Frei-Chen edge detector. The convolution masks for this edge operator are defined as follows:

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$$

Because 3 x 3 neighbourhood masks are used, the gradient can be approximated for eight possible directions (very often called compass operators). Other common compass operators include Robinson and Kirsch (Haralick and Shapiro, 1992).

3.3.3 Second order derivative operator

The Laplacian (∇^2) is an edge detector that approximates the second derivative in the same way that the gradient is an approximation to the first partial derivatives. The Laplacian is a rotationally symmetric operator and usually is approximated by a 3 x 3 mask (for 4 and 8 neighbourhood).

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

The Laplacian presents some disadvantages.

- Estimates only the magnitude, is not given any directional information.
- It responds doubly to some edges in the image.
- Since it is an approximation of the second derivative, it enhances the high frequency noise from the input image even more than the gradient operators.

3.3.4 The Marr-Hildreth edge detector

The main disadvantage of the aforementioned operators is their dependence on the *size* of the object (because the masks perform convolution only for a small area of the image) and sensitivity to noise. A popular method based on the zero crossings of the second derivative is the Marr-Hildreth edge detector (Marr and Hildreth, 1980). This approach is based on the observation that a step edge corresponds to an abrupt change in the image grey levels. Therefore, the first derivatives have the extreme values where the edges are positioned in the image and consequently the second derivative should be zero at the same position. Thus, it is easier to search for the zero crossings of the image that is first smoothed with a Gaussian mask in order to reduce the noise and then the second derivative is computed by applying the Laplacian. In other words, the image is convolved with the Laplacian of the Gaussian also known as the LoG operator ($\nabla^2(G \circ f(x,y))$, where *G* is the two-dimensional Gaussian and *f*(*x*,*y*) is the input image). The two-dimensional Gaussian has the following expression:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.7}$$

where *x*, *y* are the image co-ordinates and σ is the standard deviation. The practical implication of using Gaussian filtering is that edges are recovered reliably even in the presence of noise. It is worth mentioning, that the standard deviation plays an important role because it will determine the shape of the Gaussian filter and ultimately the scale of the operator. To choose the correct scale for this operator is difficult because it depends on the size of the objects contained in the image, information that is usually unavailable. If only strong edges are required, the standard deviation has to be increased accordingly with the edge significance. As a result, the less evident edges are suppressed, a situation that may cause a significant loss of useful information. A solution to this problem is to use multiple scales and aggregate the information between them. Unfortunately, this approach is computationally intensive since the convolution masks become larger when σ increases.

Although a powerful edge detector, the Marr-Hildreth operator has some disadvantages such as the fact that it smoothes the shape significantly and due to the Laplacian operator creates closed loops of edges.

3.3.5 The Canny edge detector

Due to its performance, the Canny edge detector (Canny, 1986) is considered by many vision researchers as an optimal approach for step edges corrupted by noise.

This edge detector was developed to meet several constraints:

- To maximise the signal to noise ratio. This criterion expresses the fact that important edges should not be missed and the spurious responses have to be suppressed.
- The distance between the actual and located position of the edge should be minimal.
- Minimise multiple responses to a single edge. This criterion is very important especially when the input image is corrupted by noise.

The Canny edge detector in contrast with the Marr-Hildreth operator returns not only the magnitude but also the direction of the gradient. Canny developed an exponential function that is very similar to the first derivative of a Gaussian. The normal to an edge *n* can be expressed as:

$$n = \frac{\nabla(G \circ f(x,y))}{\left|\nabla(G \circ f(x,y))\right|} \tag{3.8}$$

where ∇ is the gradient operator, *f(x,y)* is the input image and *G* is a two dimensional Gaussian. The edge is located at the maximum in the direction *n* of the first derivative of *G* in the same direction.

$$\frac{\partial}{\partial n}\left(\frac{\partial}{\partial n}\left(G \circ f(x,y)\right)\right) = \frac{\partial^2}{\partial n^2}\left(G \circ f(x,y)\right) = 0 \tag{3.9}$$

Equation 3.9 highlights the operations required by the Canny edge detector. The first operation convolves the input image with a Gaussian, then the partial derivatives are computed and the magnitude and orientation results are recorded. The output of the edge detector is thresholded in order to select only significant edges. Canny (1986) proposed to remove the spurious responses by thresholding with hysteresis. This technique evaluates the output of the edge detector using two threshold values (referred to as low and high thresholds) and works as follows: if an edge response is greater than the higher threshold it is considered as a valid edge point. Any candidate edge pixels that are connected to valid edge points and are above the lower threshold are also considered as edge points. The low and high thresholds are chosen according to an estimated signal to noise ratio. Also, an important issue is to choose the correct scale and since the input image is initially smoothed with a Gaussian operator, the problems discussed for the Marr-Hildreth operator (see Section 3.3.4) are valid for the Canny edge detector as well.

3.3.6 The Shen-Castan (ISEF) edge detector

The *Infinite Symmetric Exponential Filter* (ISEF) detects the edges from maxima of the gradient (or equivalently, to a zero crossing of the second derivative in the gradient direction) by using differential operators based on exponential filters (Shen and Castan, 1992). The ISEF filter for the 1-D case is implemented as a cascade of two recursive filters *$h_1(x)$* and *$h_2(x)$*.

$$h(x) = ca_0(1-a_o)^{|x|} = h_1(x)h_2(x) = c[h_1(x) + h_2(x) - a_0 d(x)] \tag{3.10}$$

$$h_1(x) = a_0(1-a_0)^x u(x), \quad h_2(x) = a_0(1-a_0)^{-x} u(-x) \tag{3.11}$$

where $0 < a_0 < 1$, $c = \frac{1}{2-a_0}$, *$d(x)$* is the Dirac function and *$u(x)$* is the Heaviside function. The first derivative of the function *$h(x)$* is given by:

$$h^{'}(x) = \frac{d}{dx}h(x) = \begin{cases} c\ln(1-a_0)[(1-a_0)^x] & if\ x > 0 \\ -c\ln(1-a_0)[(1-a_0)^{-x}] & if\ x < 0 \end{cases} \tag{3.12}$$

Using Equations 3.10 and 3.11, the first derivative can be rewritten as follows:

$$h^{'}(x) = \frac{\ln(1-a_0)}{2-a_0}[h_1(x) - h_2(x)] \tag{3.13}$$

Figure 3.2 illustrates the ISEF function and the profile of its first derivative. As could be easily observed from Equation 3.14, the profile of the second derivative is very similar with the profile of the ISEF function.

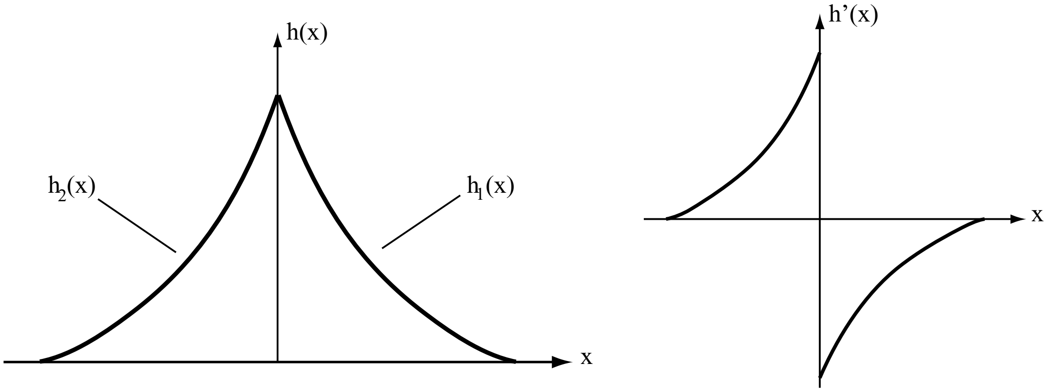


Figure 3.2. The ISEF function and its first derivative.

The second derivative is given by the following expression:

$$h^{''}(x) = \frac{2\ln(1-a_0)}{2-a_0}[h_1(x) + h_2(x) - 2d(x)] \qquad (3.14)$$

The generalisation to 2-D is straightforward since the exponential function is separable. Thus, the 2-D exponential filter can be written as follows:

$$h(x,y) = h(x)h(y) \qquad (3.15)$$

The horizontal and vertical partial derivatives of the input image *f(x,y)* are given by the following relations:

$$\begin{aligned} \frac{\partial}{\partial x}[h(x,y)\,\mathrm{o}\,f(x,y)] &= \frac{\partial h(x)}{\partial x}\,\mathrm{o}_H\,[h(y)\,\mathrm{o}_V\,f(x,y)] \\ \frac{\partial}{\partial y}[h(x,y)\,\mathrm{o}\,f(x,y)] &= \frac{\partial h(y)}{\partial y}\,\mathrm{o}_V\,[h(x)\,\mathrm{o}_H\,f(x,y)] \end{aligned} \qquad (3.16)$$

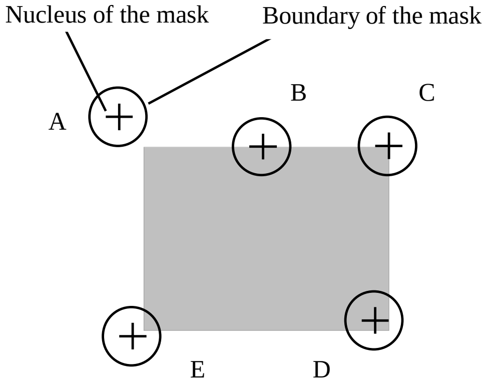
where o represents a two-dimensional convolution, o_H denotes the convolution in the horizontal direction and o_V defines the convolution in the vertical direction. These equations implement the *Gradient Exponential Filter* (GEF). Shen and Castan (1992) proposed another implementation that examines the zero crossings of the second derivative (also known as *Second Derivative Exponential Filter* (SDEF)). The equations that implement the SDEF operator are shown below.

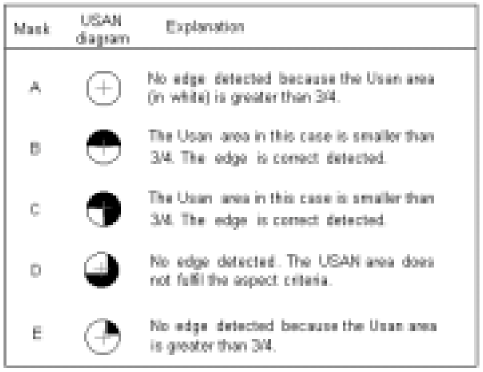
$$\frac{\partial^2}{\partial x^2}[h(x,y) \circ f(x,y)] = \frac{\partial^2 h(x)}{\partial x^2} \circ_H [h(y) \circ_V f(x,y) - 2h(y) \circ_V f(x,y)]$$
$$\frac{\partial^2}{\partial y^2}[h(x,y) \circ f(x,y)] = \frac{\partial^2 h(y)}{\partial y^2} \circ_V [h(x) \circ_H f(x,y) - 2h(x) \circ_H f(x,y)] \qquad (3.17)$$

It is important to note that the computational burden required by the edge detectors based on ISEF (GEF and SDEF) is significantly lower than the burden associated with the Canny edge detector. The qualitative results depicted in Figures 3.6 to 3.9 show that the ISEF-based operators represent an attractive solution to recover *step-like* edges quickly and accurately.

3.3.7 The SUSAN edge detector

The *Smallest Univalue Segment Assimilating Nucleus* (SUSAN) algorithm developed by Smith (1992) uses a circular mask and in correlation with a set of rules determines the edges in the image. The principle of this algorithm is illustrated in the next figure.





Mask	USAN diagram	Explanation
A	⊕	No edge detected because the Usan area (in white) is greater than 3/4.
B	◓	The Usan area in this case is smaller than 3/4. The edge is correct detected.
C	◕	The Usan area in this case is smaller than 3/4. The edge is correct detected.
D	◕	No edge detected. The USAN area does not fulfil the aspect criteria.
E	◔	No edge detected because the Usan area is greater than 3/4.

Figure 3.3. Description of SUSAN edge detector algorithm.

The mask is placed at each point in the input image and the brightness for every pixel inside the mask is compared with the one given by the nucleus (the centre of the mask).

The comparison function is implemented by the relationship illustrated in Equation 3.18.

$$c\left(\vec{r},\vec{r}_0\right)=\begin{cases}1 & if\ \left|I(\vec{r})-I(\vec{r}_0)\right|\le t\\ 0 & if\ \left|I(\vec{r})-I(\vec{r}_0)\right|>t\end{cases}\tag{3.18}$$

where $\vec{r}_o$ is the position of the nucleus, $\vec{r}$ is the position of any other pixel within the mask, $I(\vec{r})$ is the brightness of corresponding pixel, t is the brightness difference threshold and c is the output of the comparison. If the input image is corrupted by noise, better results may be obtained if the expression illustrated in Equation 3.19 is employed as comparison function.

$$c\left(\vec{r},\vec{r}_0\right)=e^{-\left(\frac{I(\vec{r})-I(\vec{r}_0)}{t}\right)^6}\tag{3.19}$$

The next operation consists of counting the pixels values inside USAN area.

$$n(\vec{r}_0)=\sum_{\vec{r}}c(\vec{r},\vec{r}_0)\tag{3.20}$$

The resulting value is compared with a *geometric* threshold which is set to $\frac{3n_{\max}}{4}$, n_{max} being the maximum value that n can take (if only step edges are considered this value should be set to $\frac{n_{\max}}{2}$). This comparison is carried out by a simple function which gives the edgeness of the nucleus (see Equation 3.21).

$$R(\vec{r}_0)=\begin{cases}\dfrac{255(g-n(\vec{r}_0))}{g} & if\ n(\vec{r}_0)<g\\ 0 & if\ n(\vec{r}_0)\ge g\end{cases}\tag{3.21}$$

where g is the geometric threshold and $n(\vec{r}_0)$ is the number of pixels of the USAN area. In order to eliminate false responses, an aspect constraint with respect to the USAN area is applied. As can be seen in Figure 3.4, for a step edge the inflection point where the second derivative is equal to zero is the right position for the edge point. This result was expected, because for that point the USAN area has the minimum value.

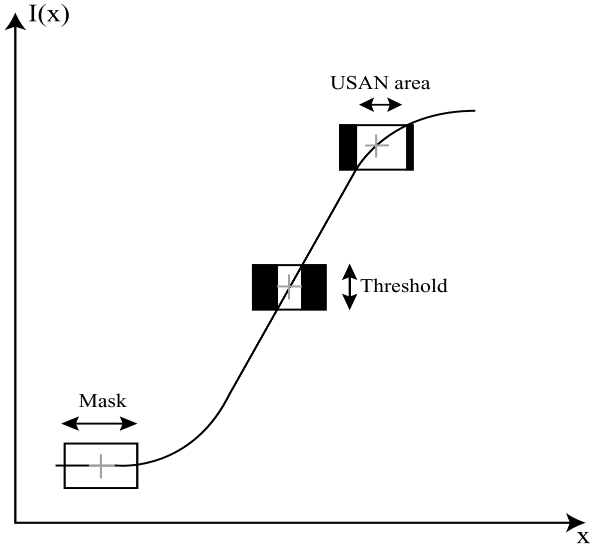


Figure 3.4. The aspect criteria (the minimum area of USAN determines the place of the edge point) Smith (1992).

For some applications the information related to edge direction may be of interest. The direction of the edge point is given by the vector which lies between the nucleus of the mask and the centre of gravity of the USAN.

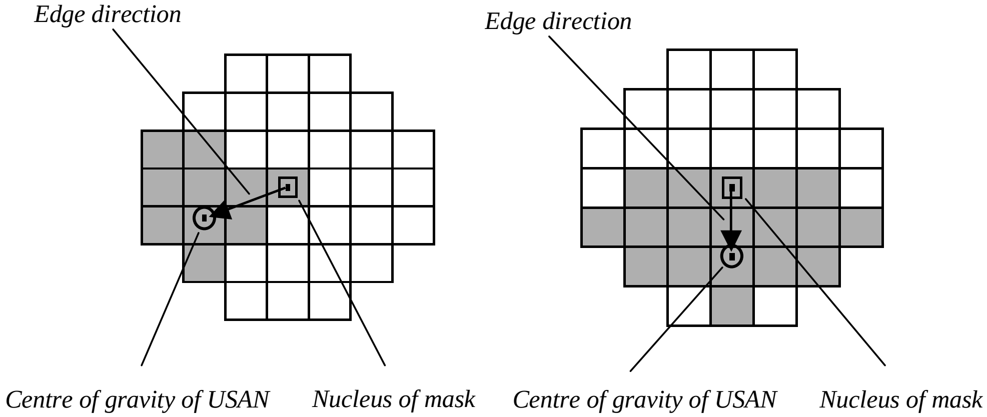


Figure 3.5. The edge direction for two different situations.

3.4 Comparison of edge detectors performances

The aim of this section consists of testing the performances of the edge operators presented previously. This task is more difficult than it appears because a qualitative estimation involves different criteria that are not always convergent. Therefore, it is difficult to establish an analytical expression that gives a competent evaluation and a common practice to compare the edge detectors relies on presenting visual results side by side. Nevertheless, this approach is subjective as long as it implies the human perception that is different from subject to subject. Heath *et al* (1997) agreed that the performance of the edge detectors has to be evaluated in the context of a visual task because an “objective evaluation of an early vision algorithm is difficult without specifying the purpose of a total system which includes the algorithm…”.

One possible way to attempt the rating of edge detectors was suggested in Ramesh and Haralick’s (1992) paper when they consider edge detection related to object recognition. This approach stems from the assumption that object boundaries are described by changes of the geometrical properties in the scene, an observation that is closely related to the human perception[6]. Unfortunately, the scenes do not always provide such convenient clues when the objects are separable and in this case the evaluation among the edge detectors has to be approached more systematically. However, an experimental study based on visual evaluation has to address some problems such as:

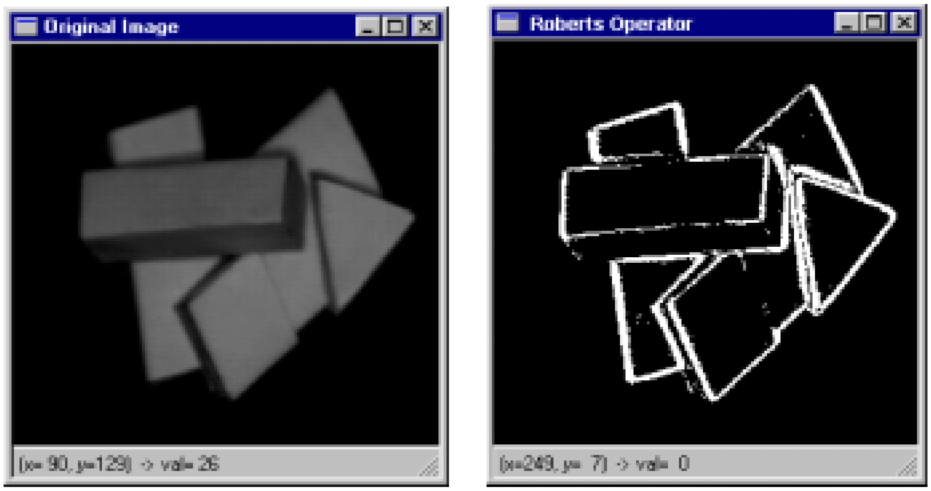
- To evaluate how the quality of the edge detection is affected when the images are artificially corrupted with noise.
- To verify how much the edge detection results associated with an operator are influenced by the choice of its parameters.

Typical quantitative measures that have to be employed by a visual evaluation are:

- Missing edges.
- Localisation errors.
- Various distortions (problems around corners and junctions, errors in the estimation of the edges orientation, gaps in edges, etc…).

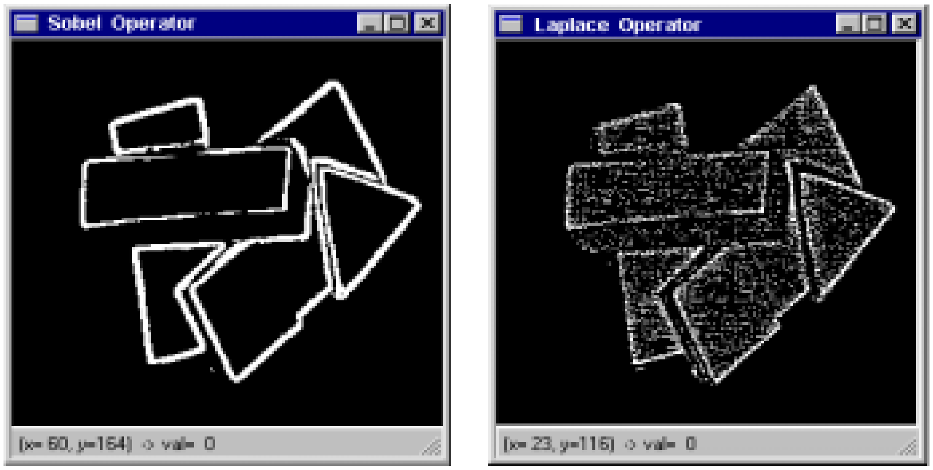
[6] Commonly, humans judge the performance of the edge detectors based on how well they are able to capture the salient features of real objects.

Figures 3.6 to 3.9 illustrate a comparison between edge detectors when the image is not corrupted by noise. The parameters *t* for SUSAN, σ for Canny, a_0 for GEF and SDEF are selected to give optimal[7] results.



(a) (b)

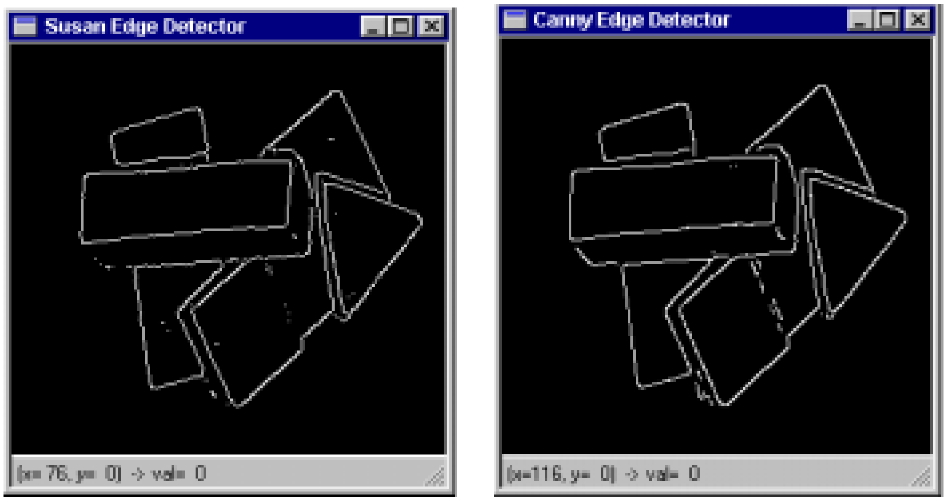
Figure 3.6. The application of the Roberts edge operator to a noiseless image. (a) Original noiseless image. (b) The resulting image when the Roberts operator is applied to image (a).



(a) (b)

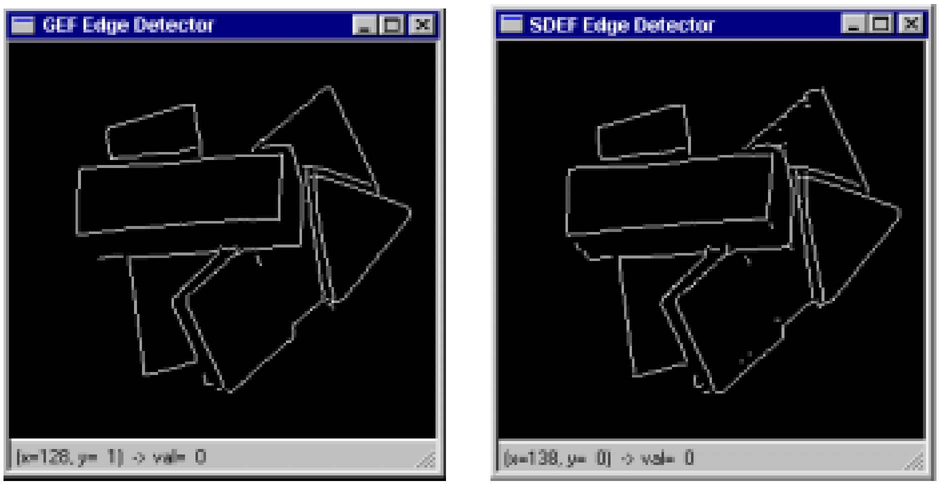
Figure 3.7. Other edge detection results. (a) The resulting image when the Sobel operator is applied to Figure 3.6-a. (b) The resulting image when the Laplace operator is applied to Figure 3.6-a.

[7] Heath *et al* (1997) demonstrated that an optimal set of parameters for an edge detector always exists. But in practice, the optimal parameters are unlikely to be found due to the amount of experiments necessary to achieve this goal. Very often "optimal " is referred to as the best solution to a problem obtained in a reasonable amount of time.



(a) (b)

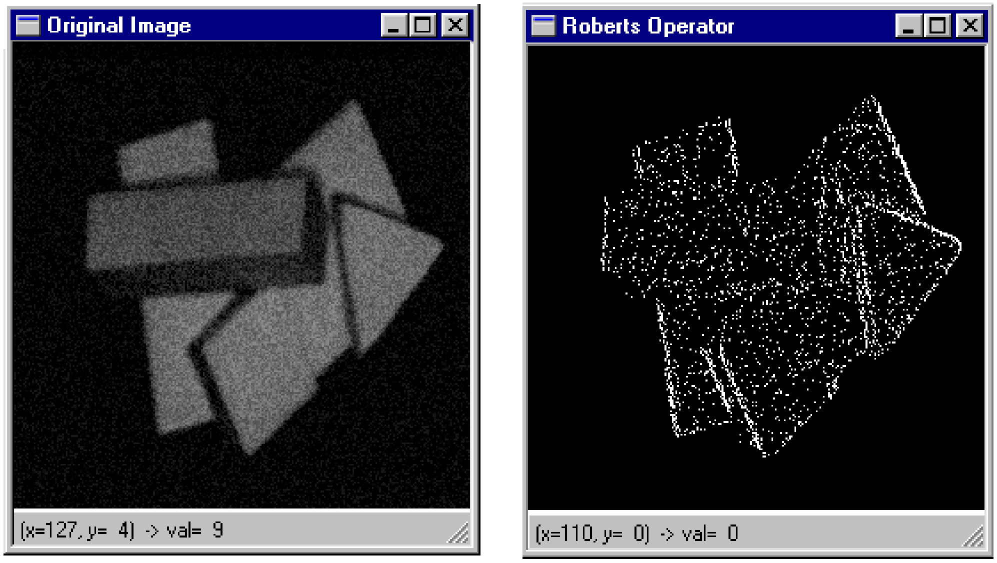
Figure 3.8. Other edge detection results. (a) The resulting image when the SUSAN edge detector (*t* = 10) is applied to Figure 3.6-a. (b) The resulting image when the Canny edge detector (σ = 1.0) is applied to Figure 3.6-a.



(a) (b)

Figure 3.9. Other edge detection results. (a) The resulting image when the GEF edge detector (a_0 = 0.45) is applied to Figure 3.6-a. (b) The resulting image when the SDEF edge detector (a_0 = 0.55) is applied to Figure 3.6-a.

Figures 3.10 to 3.13 illustrate the behaviour of different edge detectors when the image is corrupted with Gaussian noise (mean = 0, variance = 50).

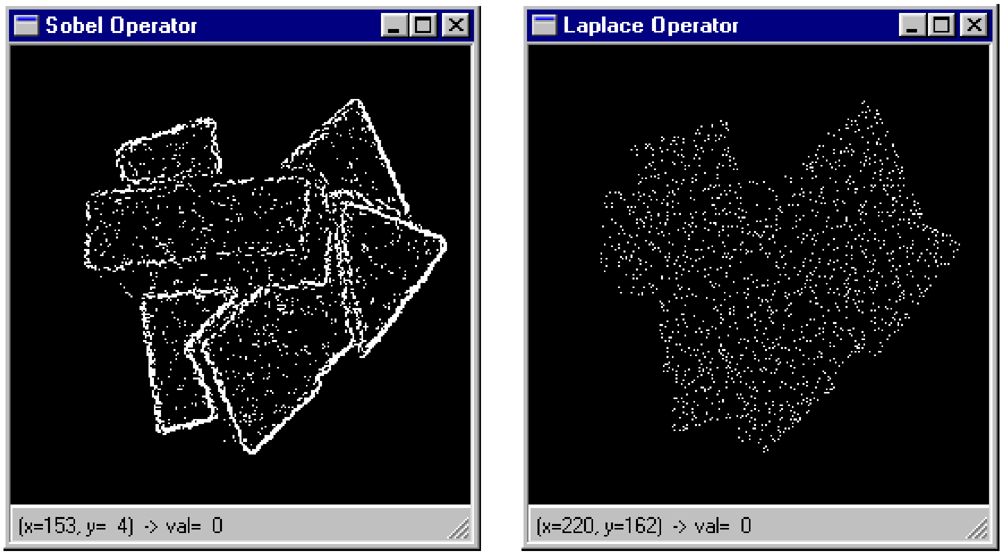


(x=127, y= 4) -> val= 9

(x=110, y= 0) -> val= 0

(a) (b)

Figure 3.10. The application of the Roberts operator to an image corrupted with noise. (a) Original image corrupted with noise. (b) The resulting image when the Roberts operator is applied to image (a).

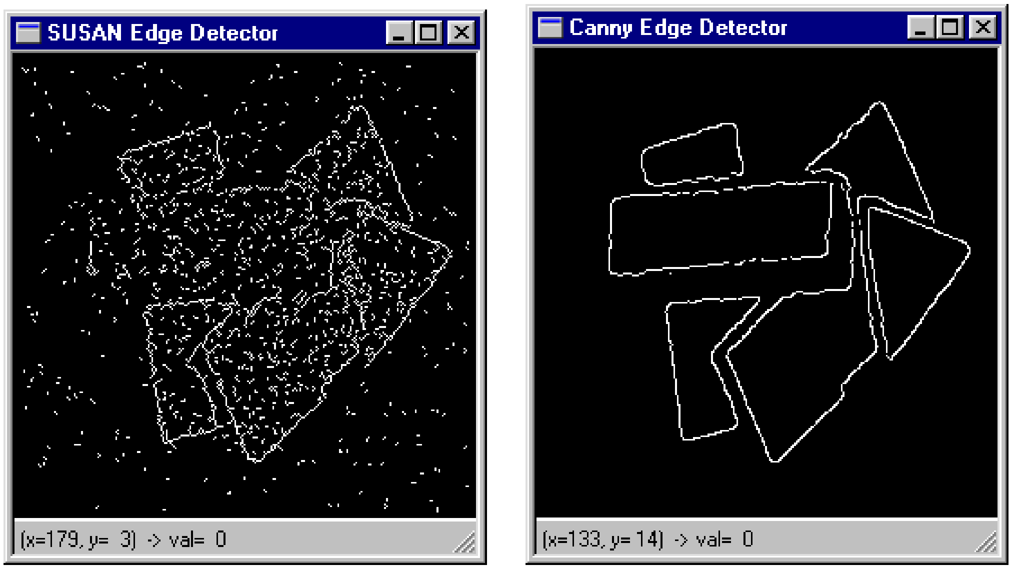


(x=153, y= 4) -> val= 0

(x=220, y=162) -> val= 0

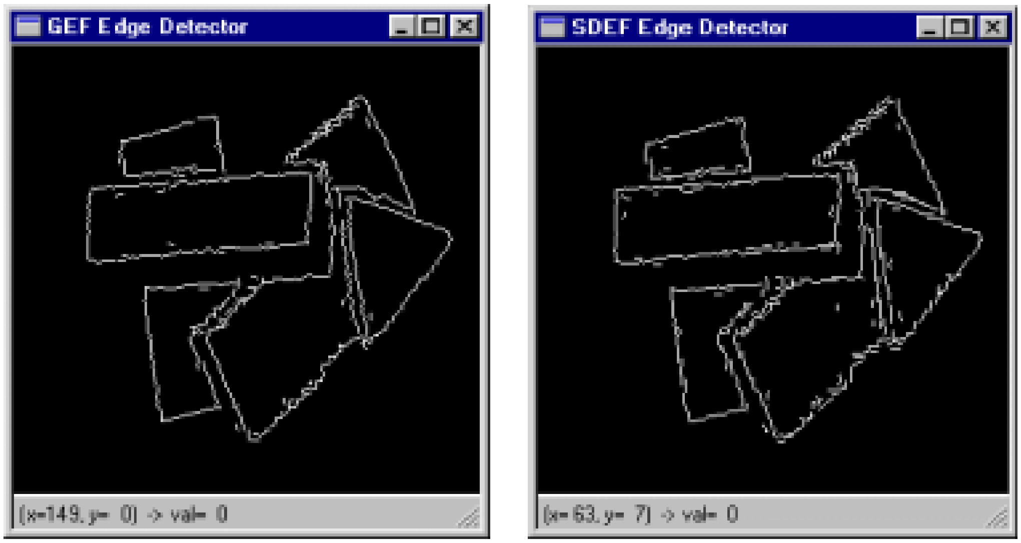
(a) (b)

Figure 3.11. Other edge detection results. (a) The resulting image when the Sobel operator is applied to Figure 3.10-a. (b) The resulting image when the Laplace operator is applied to Figure 3.10-a.



(a) (b)

Figure 3.12. Other edge detection results. (a) The resulting image when the SUSAN Edge Detector (t = 20) is applied to Figure 3.10-a. (b) The resulting image when the Canny edge detector (σ = 1.5) is applied to Figure 3.10-a.



(a) (b)

Figure 3.13. Other edge detection results. (a) The resulting image when the GEF edge detector (a_0 = 0.45) is applied to Figure 3.10-a. (b) The resulting image when the SDEF edge detector (a_0 = 0.55) is applied to Figure 3.10-a.

Edge Operator	Edge Localisation	Edge Recovering	Noise Distortion	Computation[8] (ms)	Personal Ranking
Roberts	Poor	Poor	Very poor	9	6
Sobel	Good	Poor	Poor	12	5
Laplace	Poor	Very poor	Very poor	7	7
SUSAN	Very good	Good	Poor	710	4
Canny	Very good	Very good	Very good	4922	1
GEF	Very good	Very good	Good	545	2
SDEF	Very good	Very good	Good	610	3

Table 3.1. The rating of the edge operators based on the results depicted in Figures 3.6 to 3.13.

The evaluation of the performance of the edge detectors illustrated in Table 3.1 is obtained based on the visual estimation. This evaluation may be subjective, as a statistical approach to evaluate the performance of the edge detectors based on the observation of more than one subject is beyond the scope of this thesis. One of the aims of this research is to identify the edge detector that maximises the ratio quality in edge detection versus computational load. To rate the edge detectors included in this experimental framework some measures such as edge localisation errors, missing edges and immunity to noise were employed. The Canny edge detector despite its problems associated with connectivity (especially for junctions) appears to be the best option. Unfortunately, this edge detector is complex and computationally inefficient. The edge operators that use only simple kernels such as Roberts, Sobel and Laplace perform only modestly and furthermore are very sensitive to noise. A better approach is represented by the SUSAN edge detector but it is outperformed by edge detectors based on ISEF (GEF and SDEF). These operators represent an attractive alternative because they are fast and the edge estimation is qualitatively close to one returned by the Canny edge detector. In addition, their insensitivity to noise is impressive. From aforementioned observations for the implementation outlined in this thesis the edge operators based on ISEF represent the optimal solution. Also, it may be advantageous to apply some post-processing, for example to close the gaps between edges, a situation when the quality of the overall edge detection is notably improved. A relatively simple and efficient algorithm that addresses this issue will be discussed in the next section.

[8] These measurements were performed on a Pentium 133MHz, 32 MB RAM and running Windows 98.

3.5 Post-processing the edge detection output

The segmentation process is highly influenced by the quality of the edge detector used. Although robust edge detection has been a goal of computer vision for many decades, the current range of edge operators fail to correctly recover the entire edge structure associated with a given image[9]. This is due to the presence of image noise and to the small variations in the grey level (or colour) distribution. Thus, the image noise will generate extraneous edges while a small variation of the image intensity distribution will contribute to gaps in edges. As an immediate result, the segmentation process will fail to identify the meaningful regions derived from the image under analysis. Therefore to achieve meaningful segmentation, further processing that takes into account the local information revealed in the edge detection output has to be considered.

There are various techniques which address the problem of improving the quality of the edge detection. Approaches that have been used include morphological methods (Casadei and Mitter, 1996; Snyder *et al*, 1992; Vincent, 1993), Hough transform (Gupta *et al*, 1993), probabilistic relaxation techniques (Hancock and Kittler, 1990), multiresolution methods (Bergholm, 1987; Eichel and Delp, 1985; Lindeberg, 1993; Vincken *et al*, 1996) and the use of extra information such as colour (Saber *et al*, 1997). In general, morphological approaches offer a fast solution and they attempt to maximally exploit the local information, which unfortunately is not always sufficient. In contrast, multiresolution and multiscale methods try to enhance the edge structure by aggregating the information contained in a stack of images with different spatial resolutions. These methods also referred to as pyramidal techniques usually outperform morphological techniques, but this is obtained at a high computational cost.

The implementation outlined in this thesis uses a morphological-based algorithm for edge thinning and linking that consists of two phases. The first phase deals with integration of the edge structure by aggregating in a hierarchical manner the edge information contained in a relatively small collection of edge images of different resolutions. The second phase performs edge thinning and linking using the

[9] It is worth noting that what is not detected by the edge operators does not represent an edge as a discontinuity in the optical signal and it is rather part of an inferred "perceptual" edge associated with geometrical properties of the scene.

information associated with the singular points (also referred to as endpoints or edge terminators).

3.5.1 Sequential edge reconstruction

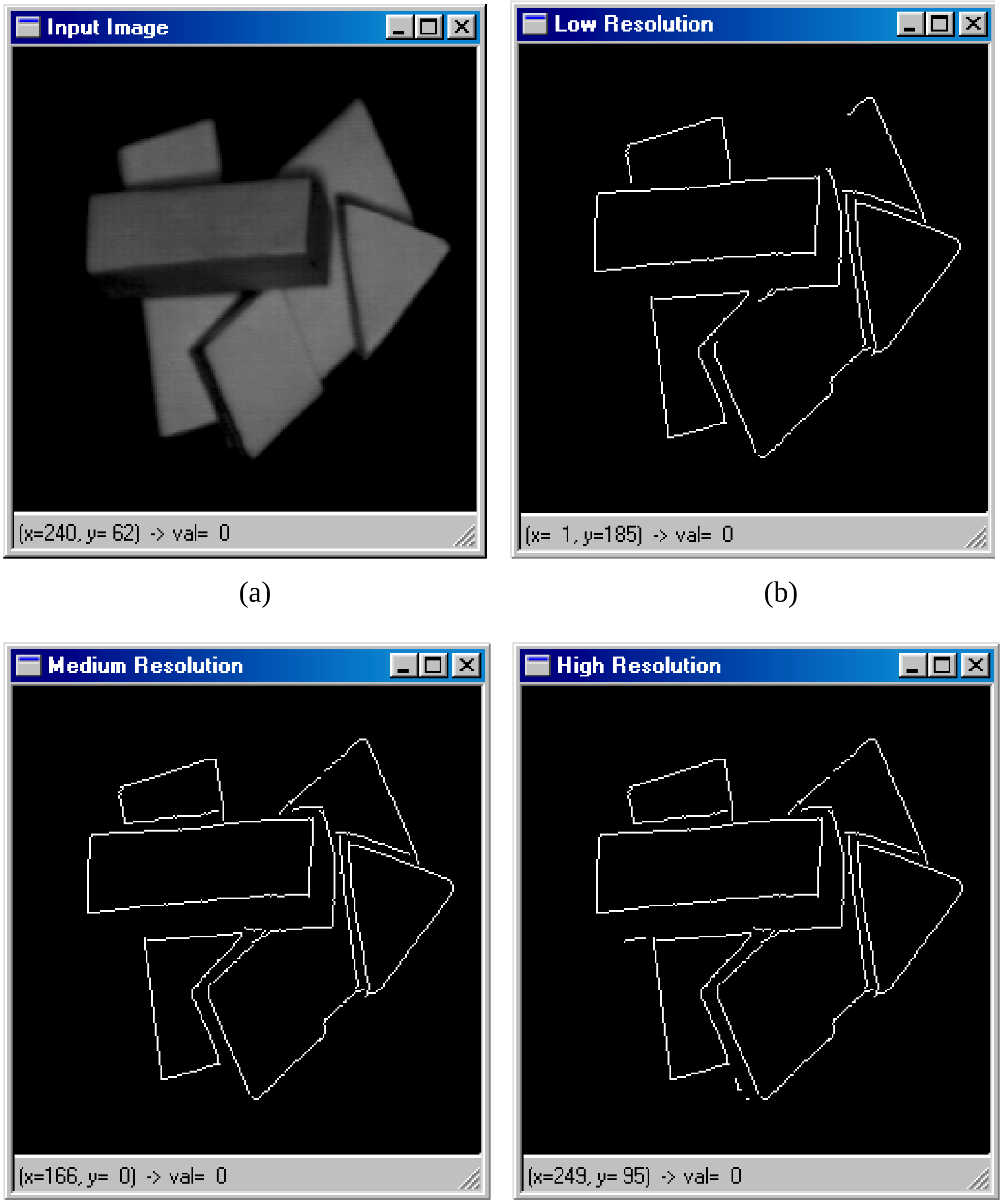
A general problem related to morphological approaches is the choice of optimal parameters for an advanced edge operator. In order to reduce the spurious responses generated by image noise, the input image is usually smoothed by applying a Gaussian filter (Marr and Hildreth, 1980). Consequently the first parameter is the standard deviation σ, a parameter that determines the scale of the Gaussian operator. As mentioned in Section 3.3.5, to further improve the edge detection output, Canny proposed a method based on thresholding with hysteresis using two threshold levels (referred to as low and high thresholds). A similar approach was employed by Shen and Castan when they developed the ISEF edge operators. Since the optimal set of these parameters is dependent on the input image, it will be difficult to apply simple criteria to consistently determine these parameters. As most developed systems have been designed to perform a specific task, it makes it difficult to use them in other applications.

To address this problem, many researchers have tried to tackle this on a global basis by building a stack of images in which the scale parameter is varied. However it makes sense to improve the edge structure by aggregating the edge information starting from images with low resolutions towards those with higher resolutions, but this entails a high computational cost since the convolution masks become larger when σ increases. Also choosing the right scale parameters is not a simple issue (Lindeberg, 1993; Vincken *et al*, 1996). In addition, the appearance and the localisation of edges within the image are increasingly disturbed when σ increases and this complication may cause a real problem when edges are reconstructed.

To avoid such problems and to maintain a low computational overhead, the threshold parameters are varied while the standard deviation is kept constant to the default value (for example $\sigma = 1.0$ for Canny and $a_0 = 0.45$ for the ISEF-based GEF edge detector). This approach has the advantage that the edge operator has to be applied only once while the hysteresis threshold is sequentially applied to obtain the stack of images of different resolutions[10]. At this stage, a key problem consists of

[10] The term resolution is not used in the normal accepted scale-space sense. Here, the term resolution defines the level of edge detail presented in the images generated at different cut-off thresholds.

selecting the optimal range for the threshold parameters. Obviously, the aim is to have the edge segments presented in the output image as large as possible. Smaller segments (less than 4 pixels) are generally due to noise. In this regard, the lower threshold is selected by analysing the level of small edge segments that are present in the edge detection output.



(a)

(b)

(c)

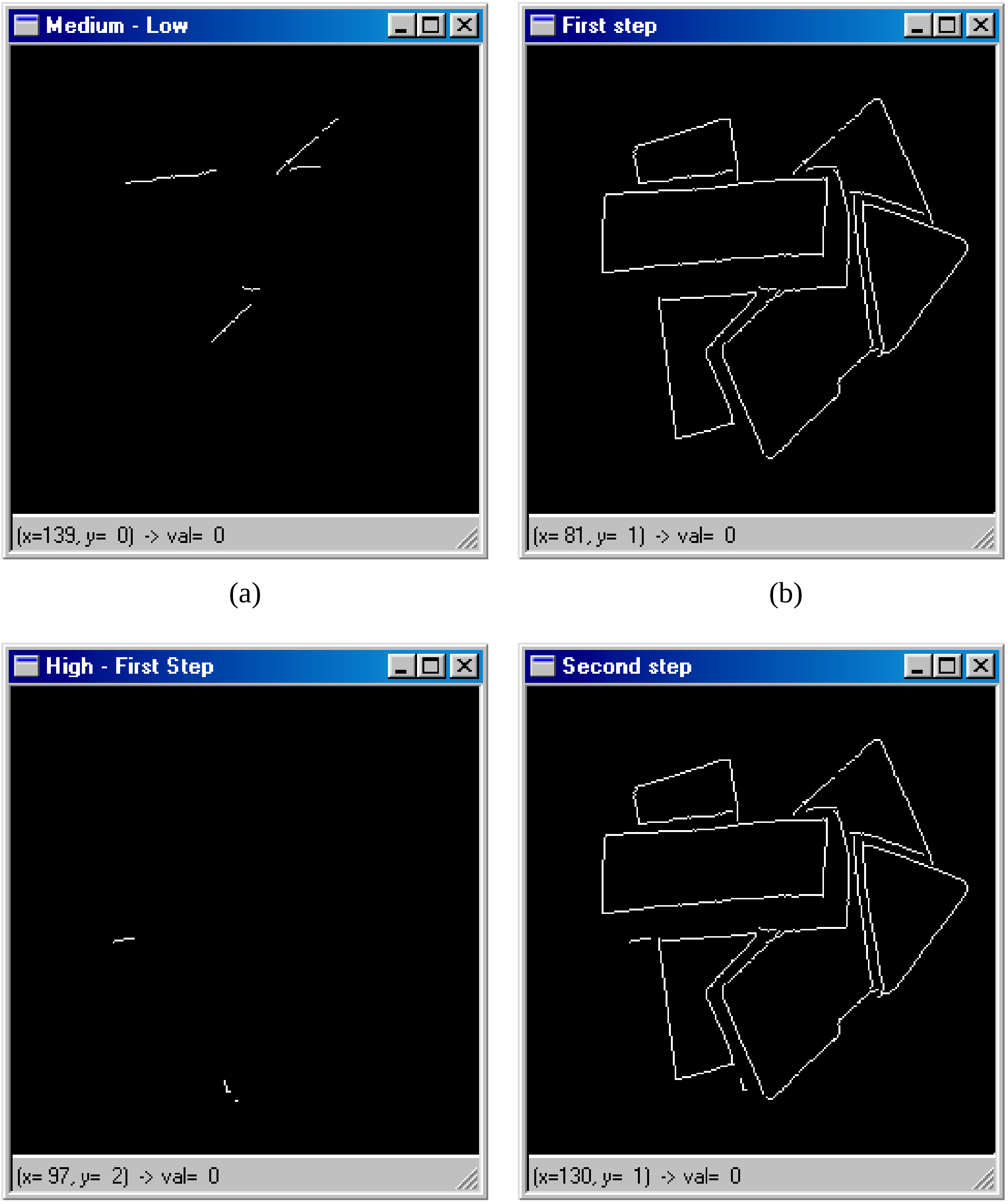
(d)

Figure 3.14. The image stack. (a) Input image. (b) The low resolution image. (c) The medium resolution image. (d) The high resolution image.

The algorithm is initialised with the minimal value for the lower threshold (during this stage, the higher threshold is set at the same value as the lower threshold) and it is incrementally increased until the ratio between the number of edge pixels derived from small segments and the number of edge pixels derived from large segments is smaller than a preset value. When this criterion is upheld, the lower threshold value is fixed and by increasing the value of the higher threshold the images with a coarser resolution are obtained. The maximum value of the higher threshold is dependent upon the edge detector used (for example it takes a value of 20 for the GEF edge operator). To increase robustness to noise at least 2 stack images are required, but to better combine the edge structure an image stack which contains three images (referred to as low resolution image, medium resolution image and high resolution image) of different resolutions are considered. This solution is advantageous as long as it allows the removal of noise at each iteration. Also, the scene may contain objects that have a low contrast against the background; consequently they will not be present in the image with the lowest resolution. Once the stack of images are processed (see Figure 3.14). The algorithm attempts to combine the edge information between them by using the following procedure:

1 Initially the image with lowest resolution is subtracted from the medium resolution image.
2 The resulting image is labelled using a graph-based algorithm (refer to Section 3.6.2) and the length of each edge segment is computed.
3 The next step involves analysing the edge structure contained in both images, namely the image with the lowest resolution and the labelled image. If any edge pixel is connected with an edge segment from the labelled image, the edge segment is added to the edge structure in the image with lower resolution.
4 The remaining edge segments from the labelled image are analysed in order to decide if they are valid edge segments. If the labelled segments under examination contain more than 4 edge pixels, they are added to the edge structure.
5 The remaining pixels are assigned as edge segments only if there is a valid edge segment with a length greater than 4 pixels in their 3 x 3 vicinity. The aim of this operation is to remove the isolated and the small undesirable edge responses that are caused by noise.

When this process is completed, the image resulting from the first step is subtracted from the high resolution image. Then, the edges are aggregated by using the same procedure outlined above.



(a) (b)

(c) (d)

Figure 3.15. The edge reconstruction process for images illustrated in Figure 3.14. (a) The subtraction of the lowest resolution image from the medium resolution image. (b) The resulting image after the first iteration. (c) The subtraction of the resulting image from the highest resolution image. (d) The output image.

Figure 3.15-d shows the result obtained after the application of the proposed reconstruction scheme.

3.5.2 Iterative edge thinning

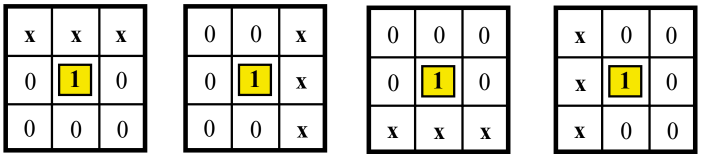
Multiple edge replications represent another typical error associated with edge detecting operators. Thus, there are cases when the edge responses are several pixels wide. Since the goal of this approach consists of reconnecting the interrupted edges using only the local information, multiple edge responses may generate incorrect linking decisions. Therefore to use the local information more efficiently, a thinning algorithm has to be applied to remove the unnecessary edge responses. In this regard, an iterative morphological thinning algorithm based on the use of L-type structuring elements was implemented (Sonka *et al*, 1993). This algorithm is defined as follows:

$$I \varnothing S = I - (I \otimes S)$$

where I is the image containing the edge information, S is the structuring element, $\varnothing$ denotes the thinning operation and $\otimes$ defines the binary hit-or-miss transformation. The thinning process is convergent and stops when two successive images in the sequence are identical.

3.5.3 Recovering the endpoints and edge normalisation

Although the proposed scheme significantly improves the edge structure, there are situations where gaps in edges exist in the output image. To correct this problem, a method to bridge the gaps by analysing the endpoints has been developed. Extracting the endpoints entails a simple morphological analysis and consists of a set of 3 x 3 masks that are applied to the resultant image after the application of the edge reconstruction process.



x	x	x
0	1	0
0	0	0

0	0	x
0	1	x
0	0	x

0	0	0
0	1	0
x	x	x

x	0	0
x	1	0
x	0	0

Figure 3.16. The masks used to detect the endpoints.

Figure 3.16 illustrates the masks used to detect the endpoints, where the pixel under investigation is highlighted and mask entries indicated by ‘x’ can take any value (0 or 1) but at least one of them has the value 1. This ensures that the single edge pixels are not marked as endpoints.

0	0	**1**	0	**1**	0	**1**	0	0
0	**1**	0	**1**	0	**1**	0	**1**	0

(a)

0	0	0	0	0	0	0	0	0
0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0

(b)

Figure 3.17. (a) The chain of endpoints. (b) The normalised edge structure.

As can be easily observed, there is the possibility to have “chains of endpoints”. This situation is illustrated in Figure 3.17 where each edge pixel would be assigned as an endpoint. This is a common problem and is caused by the edge localisation errors that are generally due to noise. Since for each endpoint the algorithm evaluates its direction and analyses the possible connections with other edge points, the chains of endpoints will only increase the computational load as long as they are linked. Fortunately, the localisation errors are easy to detect. This situation occurs when more than 2 endpoints are connected and the algorithm shifts the pixels in order to obtain the linear configuration illustrated by Figure 3.17-b.

3.5.4 Endpoints labelling

To efficiently close the gaps in edges, the local knowledge has to be maximally exploited. Thus, it is necessary to determine the scanning direction for each endpoint by evaluating the linked edge pixels that generate it. It can be noticed that the masks illustrated by Figure 3.16 contain some information that gives a useful clue regarding the endpoint direction. Unfortunately, this gives only 4 scanning directions which is not sufficient to always find the correct result. To avoid such limitation, the search for edge links was extended to 8 directions, a situation when supplementary information has to be evaluated. As Figure 3.18 illustrates, there are cases when the endpoint is generated by a straight edge, a situation where the scanning direction can be easily established.

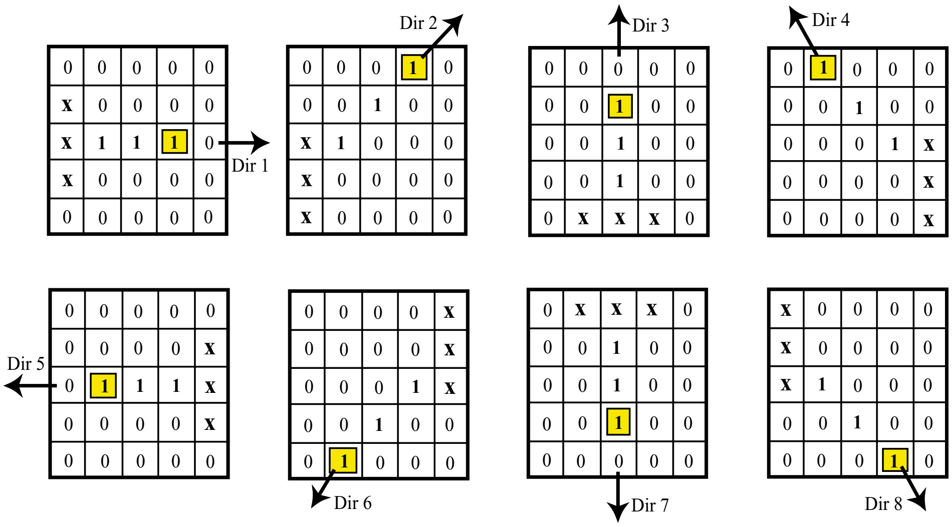


Figure 3.18. Situations where the edge direction (indicated with an arrow) is derived from straight edges.

This may not be the case for curved edges, when the edge direction is not as well defined. A typical situation is illustrated in Figure 3.19 where the endpoint direction is evaluated by analysing the local information for a larger neighbourhood. In Figure 3.19 only the first 3 directions are analysed, while the remaining directions can be obtained by rotating the masks.

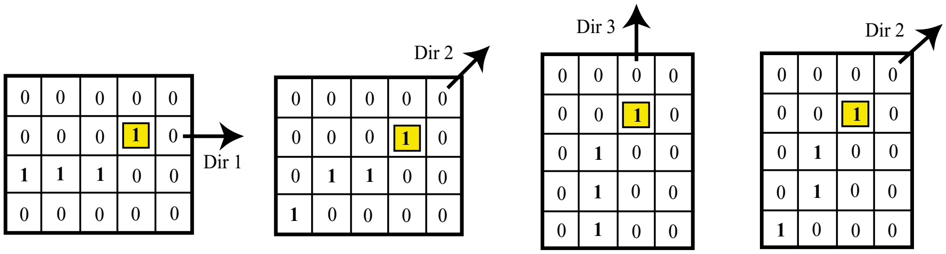


Figure 3.19. The edge direction derived from curved edges.

3.5.5 Edge linking

The next step of the algorithm deals with searching for possible edge paths by using the information derived from the endpoints. The scanning process is iterative and starts at the endpoint under investigation. This process is defined as follows:

1 Initially the algorithm evaluates the 3 x 3 neighbourhood at the side given by the endpoint direction. In order to avoid closed loops of edges, the pixels

situated in the endpoint's neighbourhood are evaluated in a strict order. Thus, the pixels which lie on the endpoint direction are evaluated first. If there are no edge pixels detected, the scanning continues by evaluating the remaining pixels, starting with those closer to the endpoint.

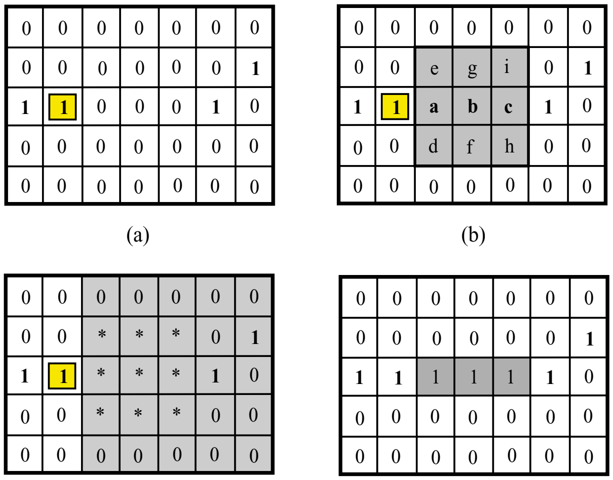
2 If no connections are detected, the algorithm evaluates the 5 x 5 neighbourhood while ignoring the 3 x 3 area which was already assessed.

3 If the scanning process fails to find an edge pixel, the algorithm analyses the 7 x 7 neighbourhood by using the same procedure outlined above.

4 If a connection is detected, a path is established between the endpoint and the detected edge point by using the Bresenham algorithm (Bresenham, 1965). In other words a line is drawn between the endpoint and the edge pixel.

5 There exists the possibility that the detected edge point to be also an endpoint. If the path given by the detected endpoint is the same like that given by the first endpoint, as a result the Bresenham algorithm will be applied twice. To avoid such situations, the co-ordinates of the path are stored into a table and each time a new path is found, the algorithm verifies if there is not another entry with the same co-ordinates.

6 If the scanning process fails to find any edge pixels, no entry is generated.



0	0	0	0	0	0	0
0	0	0	0	0	0	**1**
1	**1**	0	0	0	**1**	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(a)

0	0	0	0	0	0	0
0	0	e	g	i	0	**1**
1	**1**	**a**	**b**	**c**	**1**	0
0	0	d	f	h	0	0
0	0	0	0	0	0	0

(b)

0	0	0	0	0	0	0
0	0	*	*	*	0	**1**
1	**1**	*	*	*	**1**	0
0	0	*	*	*	0	0
0	0	0	0	0	0	0

(c)

0	0	0	0	0	0	0
0	0	0	0	0	0	**1**
1	**1**	1	1	1	**1**	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(d)

Figure 3.20. The edge linking process. (a) The edge structure around an endpoint. (b) Scanning the 3 x 3 neighbourhood (the pixels are evaluated in alphabetic order). (c) Scanning the 5 x 5 neighbourhood (the previous area is not taken into account). (d) The result after the Bresenham algorithm is applied.

Figure 3.20 illustrates the edge linking process described above. The mask entries marked with ‘∗’ indicate that they were already verified. To evaluate the performance of the proposed edge linking scheme it was tested on several images. Initially, the algorithm was tested on noiseless images and results of the complete process can be seen in Figure 3.21.

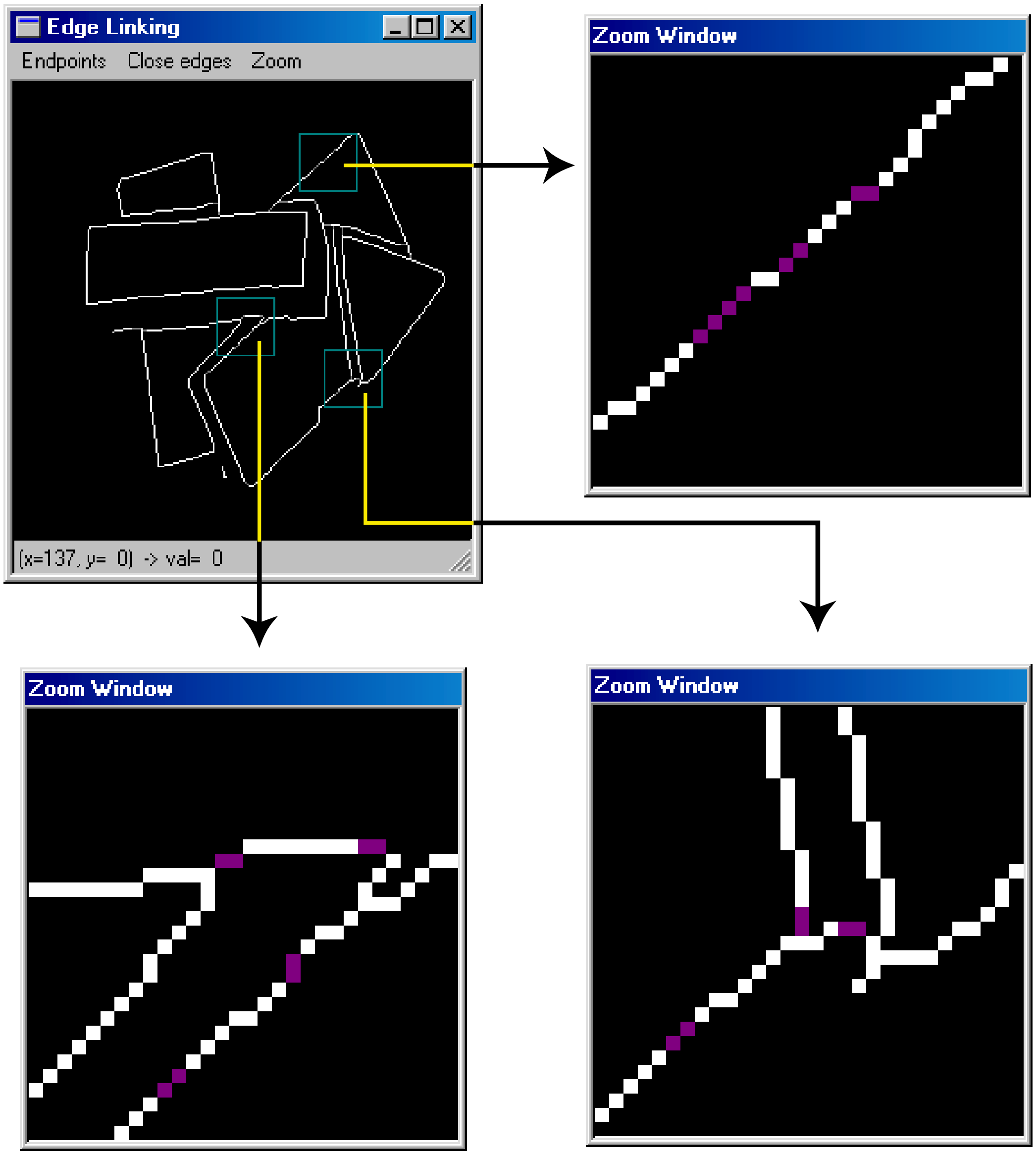
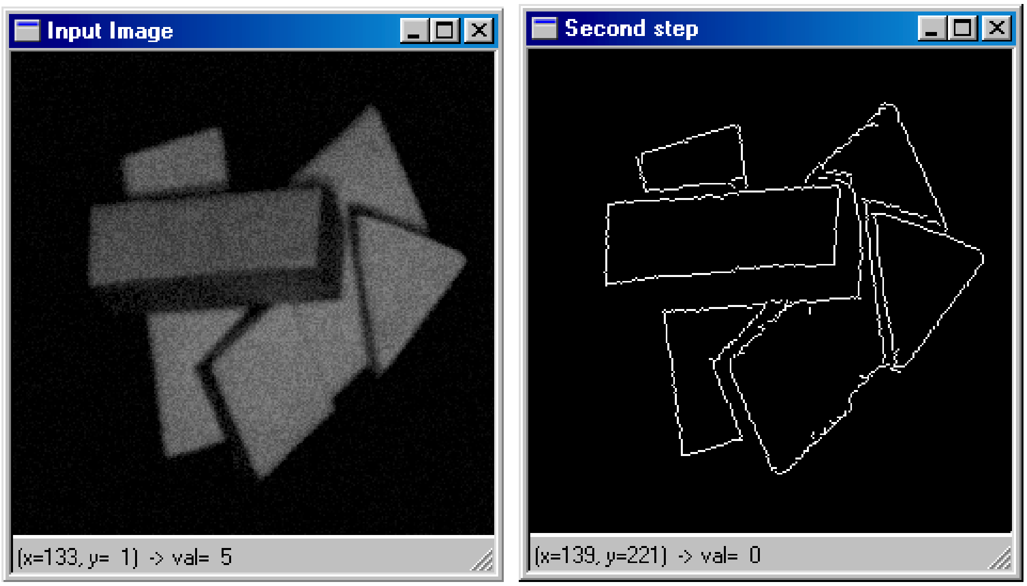


Figure 3.21. The edge linking results when the algorithm is applied to the image illustrated in Figure 3.15-d. The linking pixels are shaded and for clarity some details are magnified.

As Figure 3.21 illustrates, the algorithm was able to handle even difficult situations such as edge bifurcation. This can be observed in the last two image details.

To verify the algorithm's robustness to noise, the input image was corrupted with additive Gaussian noise (mean = 0, variance = 30). In Figure 3.22 the result of the edge reconstruction process is illustrated.



(a) (b)

Figure 3.22. The edge reconstruction process. (a) The input image corrupted with Gaussian noise. (b) The result after edge reconstruction.

It can be noticed that the edge segments caused by noise are removed except in the case they make contact with the edge structure presented in the lower resolution images. Figure 3.23 illustrates the output after the edge linking algorithm is applied. As expected, errors such as loops of edges occur due to extraneous edges caused by image noise. However, in practical applications this is not a major disadvantage as long as they generate small regions that can be easily removed or relabelled.

The experimental results indicate the ability of the proposed algorithm to reconnect edges even in cases when they are closely spaced. Also some limitations of this approach can be noticed. The first is derived from the fact that the algorithm is not able to cope with gaps larger than 7 pixels. The scanning process can be extended to search until a connection is found but this leads to incorrect linking decisions especially when dealing with complex scenes. Furthermore, since the gaps are bridged using the Bresenham algorithm the edge geometry for larger gaps is not preserved. However, large gaps cannot be efficiently closed using only the local edge information and to address this problem robustly supplementary knowledge has to be considered.

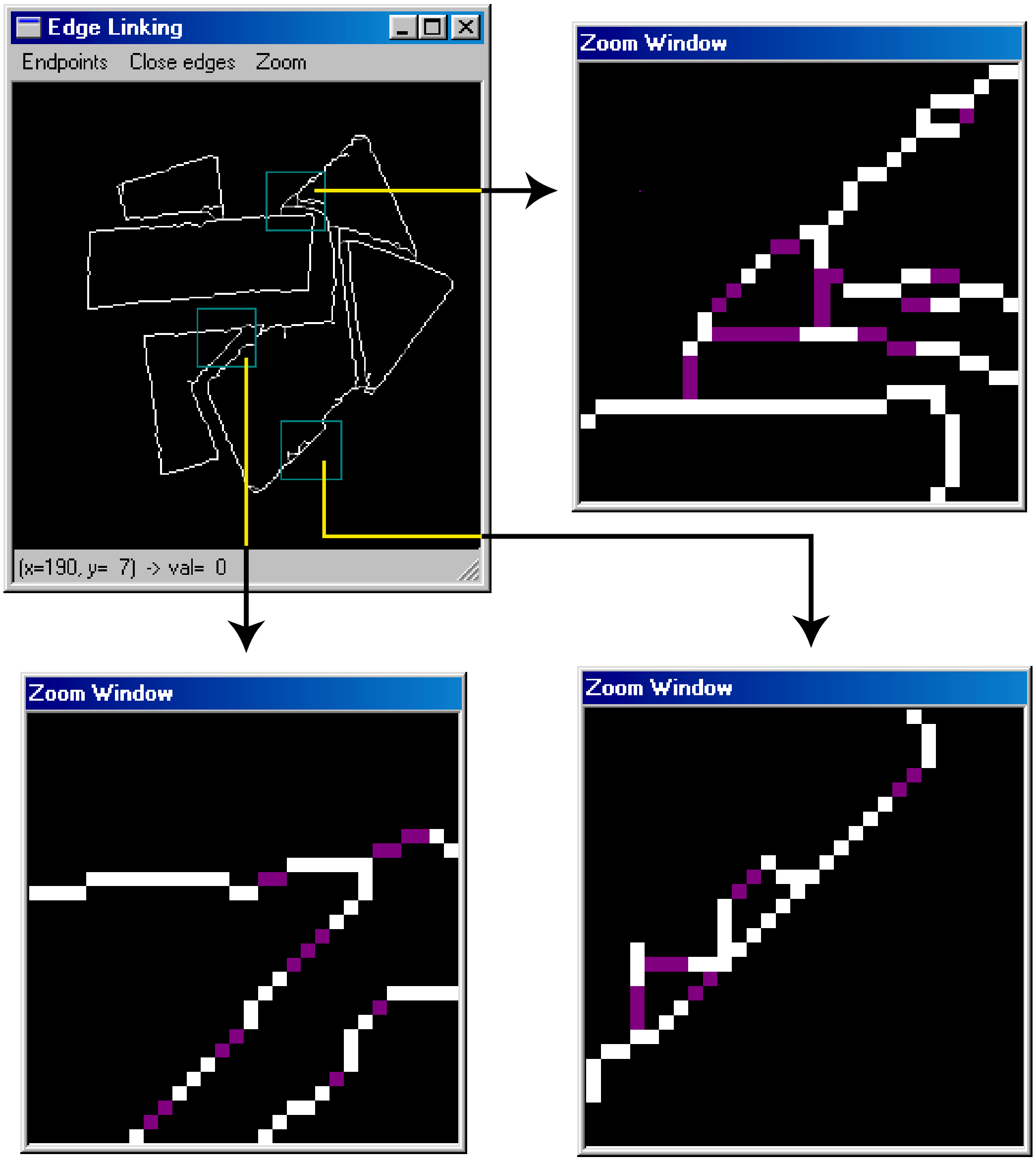


Figure 3.23. The edge linking results when the algorithm is applied to the image illustrated in Figure 3.22-d.

An important issue is the computational efficiency. Achieving reasonable timing using a complex edge operator such as Canny is difficult, since the computational time required to extract the edge structure derived from a 256 x 256 / 256 greyscale image is 4922 ms when running on a PC with a Pentium 133 processor. As mentioned in Section 3.4, the ISEF-based GEF operator represents an attractive solution since the processing time is 545 ms. Also, it is worth mentioning that this advantage is obtained without reducing significantly the edge recovering performance. The processing time associated with the edge reconstruction, thinning and linking algorithm depends on the complexity of the edge structure. Timings for images involved in the aforementioned experiments are depicted in Table 3.2.

Input image	Edge detection [ms]	Edge reconstruction [ms]	Edge thinning and linking [ms]
Figure 3.14-a	Table 3.1	450	100
Figure 3.22-a	Table 3.1	495	115

Table 3.2. Performance of the edge linking algorithm.

3.6 Image segmentation

The image is segmented using the result returned from the edge linking algorithm. This stage is very important because the resulted image after segmentation is used as input information for the object recognition algorithm. The operations required to disconnect the regions contained in the input image are:

1 To remove the small regions created by shadows, a morphological binary dilation with a square 3 x 3 structuring element is applied twice to the edge-linked image.

2 The resulting image is analysed in order to identify the regions’ boundaries. If an edge point is present, its position is marked in the input image as a point of discontinuity.

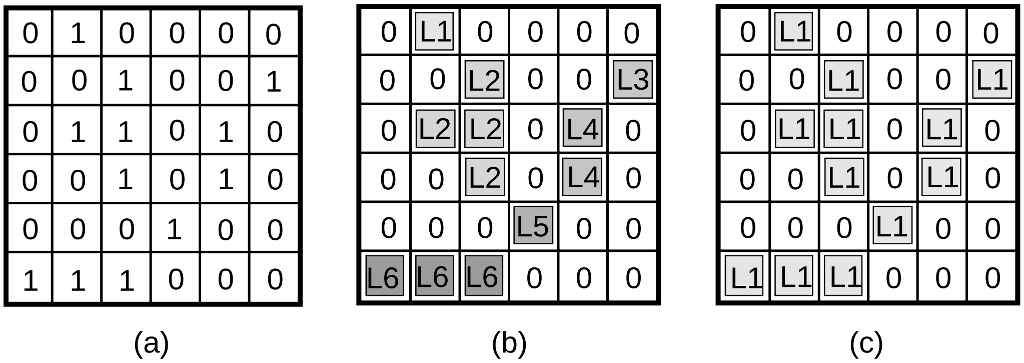
3 The image obtained from step 2 is thresholded against the background. The background is assumed to be the darkest region contained by the image. The threshold value is set just above this value.

4 The labelling algorithm (see Section 3.6.1) is applied to the image resulted from step 3. This assigns a unique label to each disjointed region.

3.6.1 The labelling process

The aim of the labelling process is to assign a unique index (label) to each region in a binary image and consists of connecting the pixels that have a value other than zero (which describes the background). The connectivity between pixels depends on the type of neighbourhood used.

When only the north, south, east, and west directions are considered, the connectivity in this case is based on a 4-connected neighbourhood. When the northeast, northwest, southeast, and southwest are also considered, the connectivity in this case consists of 8-connected neighbourhood. Figure 3.24 illustrates the pixel connectivity when the 4 and 8-connected neighbourhood are employed.



0	1	0	0	0	0
0	0	1	0	0	1
0	1	1	0	1	0
0	0	1	0	1	0
0	0	0	1	0	0
1	1	1	0	0	0

(a)

0	L1	0	0	0	0
0	0	L2	0	0	L3
0	L2	L2	0	L4	0
0	0	L2	0	L4	0
0	0	0	L5	0	0
L6	L6	L6	0	0	0

(b)

0	L1	0	0	0	0
0	0	L1	0	0	L1
0	L1	L1	0	L1	0
0	0	L1	0	L1	0
0	0	0	L1	0	0
L1	L1	L1	0	0	0

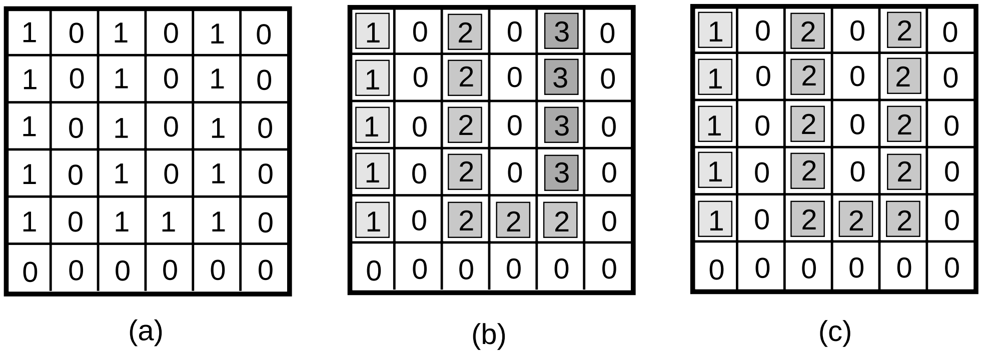
(c)

Figure 3.24. The labelling process. (a) The binary image. (b) Labelling the pixels using 4-connectivity. (c) Labelling the pixels using 8-connectivity.

As can be seen in Figure 3.24, 4-connectivity for certain situations fails in assigning a correct label for pixels which are grouped together. Therefore, most of the labelling algorithms use 8-connectivity.

3.6.2 The iterative labelling algorithm

This algorithm analyses the input image in a raster scan mode and assigns a new label each time an unconnected pixel is found. When two regions that are connected but have different labels are found, an iterative label propagation sequence will reassign both regions with the minimum label. Figure 3.25 illustrates how this algorithm works.



1	0	1	0	1	0
1	0	1	0	1	0
1	0	1	0	1	0
1	0	1	0	1	0
1	0	1	1	1	0
0	0	0	0	0	0

(a)

1	0	2	0	3	0
1	0	2	0	3	0
1	0	2	0	3	0
1	0	2	0	3	0
1	0	2	2	2	0
0	0	0	0	0	0

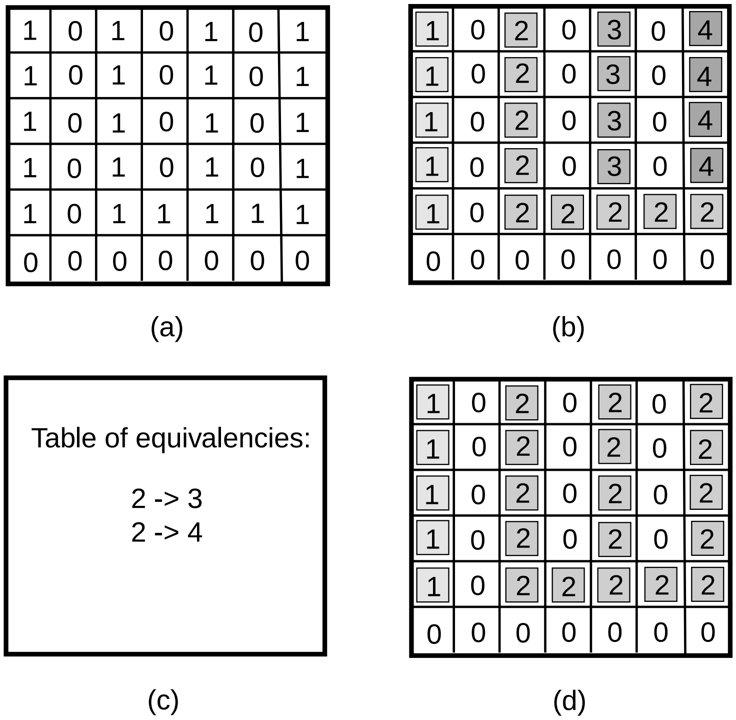
(b)

1	0	2	0	2	0
1	0	2	0	2	0
1	0	2	0	2	0
1	0	2	0	2	0
1	0	2	2	2	0
0	0	0	0	0	0

(c)

Figure 3.25. The iterative labelling algorithm. (a) The original binary image. (b) The results before label propagation. (c) The results after label propagation.

This algorithm is simple but unfortunately is slow especially when the image contains many objects with *U* type shape. A more efficient algorithm (Haralick and Shapiro, 1992) uses a table that records the equivalencies between assigned labels. Whenever a situation where two connected regions which have different labels is found, a new entry in the table of equivalencies is inserted.



1	0	1	0	1	0	1
1	0	1	0	1	0	1
1	0	1	0	1	0	1
1	0	1	0	1	0	1
1	0	1	1	1	1	1
0	0	0	0	0	0	0

(a)

1	0	2	0	3	0	4
1	0	2	0	3	0	4
1	0	2	0	3	0	4
1	0	2	0	3	0	4
1	0	2	2	2	2	2
0	0	0	0	0	0	0

(b)

Table of equivalencies:

2 -> 3
2 -> 4

(c)

1	0	2	0	2	0	2
1	0	2	0	2	0	2
1	0	2	0	2	0	2
1	0	2	0	2	0	2
1	0	2	2	2	2	2
0	0	0	0	0	0	0

(d)

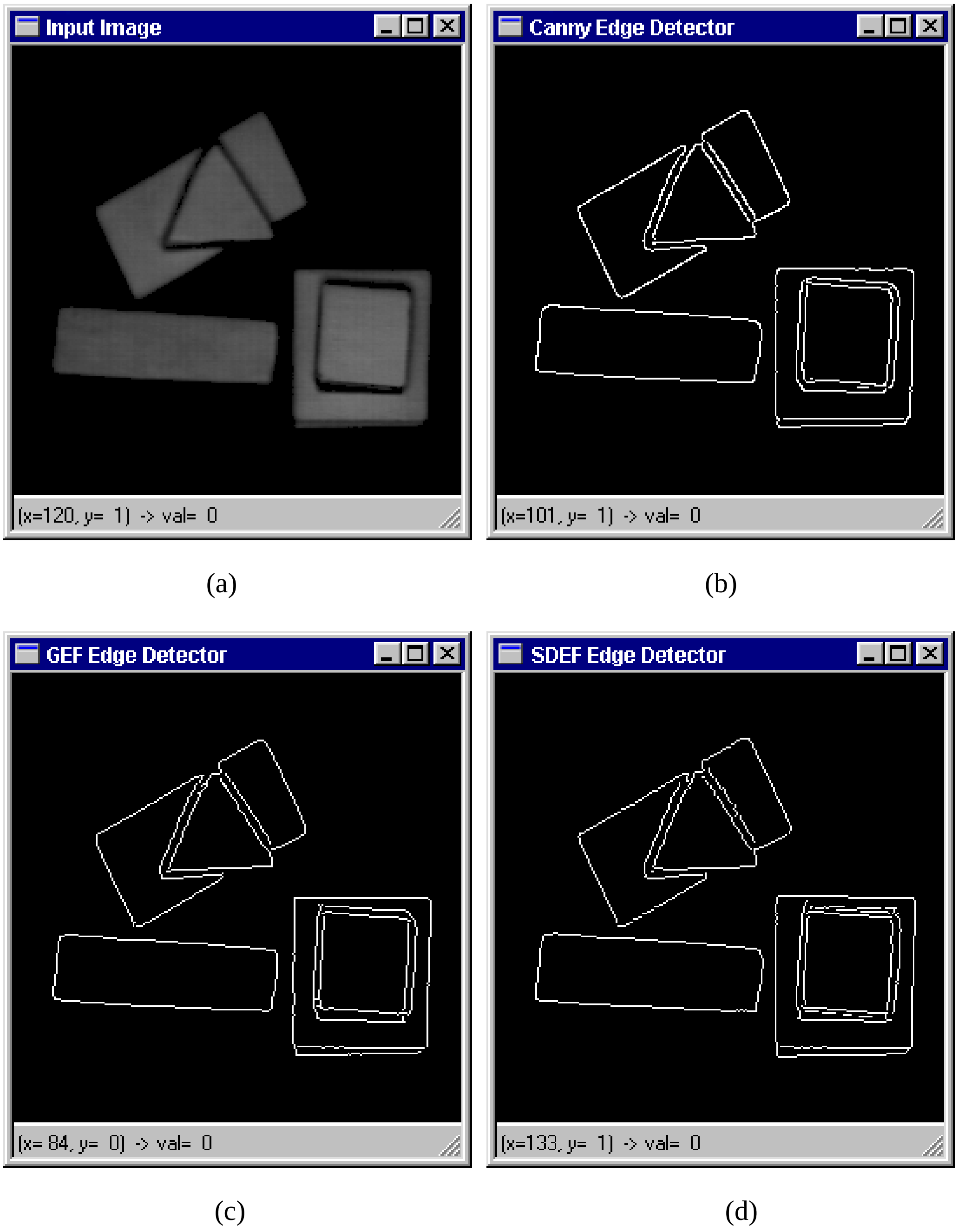
Figure 3.26. The efficient labelling algorithm. (a) The original binary image. (b) The results after first step. (c) The table of equivalencies. (d) The results after solving the equivalencies between labels.

The last stage consists of reassigning the labels using a graph searching technique (depth first search). This process is illustrated in Figure 3.26.

There is a significant difference in computation cost between these approaches. For example, a complex 256 by 256 binary image is labelled by the iterative algorithm in 3.4 seconds (this time is highly dependent on the shape of the objects contained in the image) while for the algorithm that uses the table of equivalencies the required time is about 0.5 seconds.

3.6.3 Input data representation

The choice of input data representation must be correlated with the edge interpretation. As mentioned previously, an edge can be seen as a discontinuity in the intensity function or alternatively, can be associated with the changes in the scene's depth structure. The approaches discussed above suggest the input data required to be analysed. Thus, in the first situation the edge information is detected from the raw images while in the second case the range images are used as input. A natural question can be formulated as: which approach gives better results? A realistic answer was suggested by Sonka *et al* (1993) when they tied the problem of input data representation with the context of the application where this information is analysed. For some applications such as that outlined in this thesis where the 3-D information is available, a practical way to approach the segmentation process relies on the use of range images. The main problem is whether the edge information detected from the range images is precise enough to obtain a meaningful segmentation. Obviously, the precision of the segmentation process is strictly correlated with the resolution of the range sensor. Rahardja and Kosaka (1996) acknowledged in their paper the difficulty of obtaining a precise segmentation if the range sensor is low resolution or the depth discontinuities revealed in the range image are not significant. If this is the case, better results may be obtained if the edge-based segmentation scheme is applied to raw images. Because the implementation described in this thesis deals with a set of small polyhedral textureless objects, a realistic way to conduct the research is to investigate which approach returns better results. To carry out this investigation, it is necessary to evaluate the edge information associated with raw and range images that describe the same scene. The results are evaluated side by side using the same visual framework employed to rate the edge detectors (refer to Section 3.4). In these experiments only the top three edge detectors (Canny, GEF and SDEF[11]) resulted from the visual evaluation carried out in Section 3.4 were utilised.



(a)

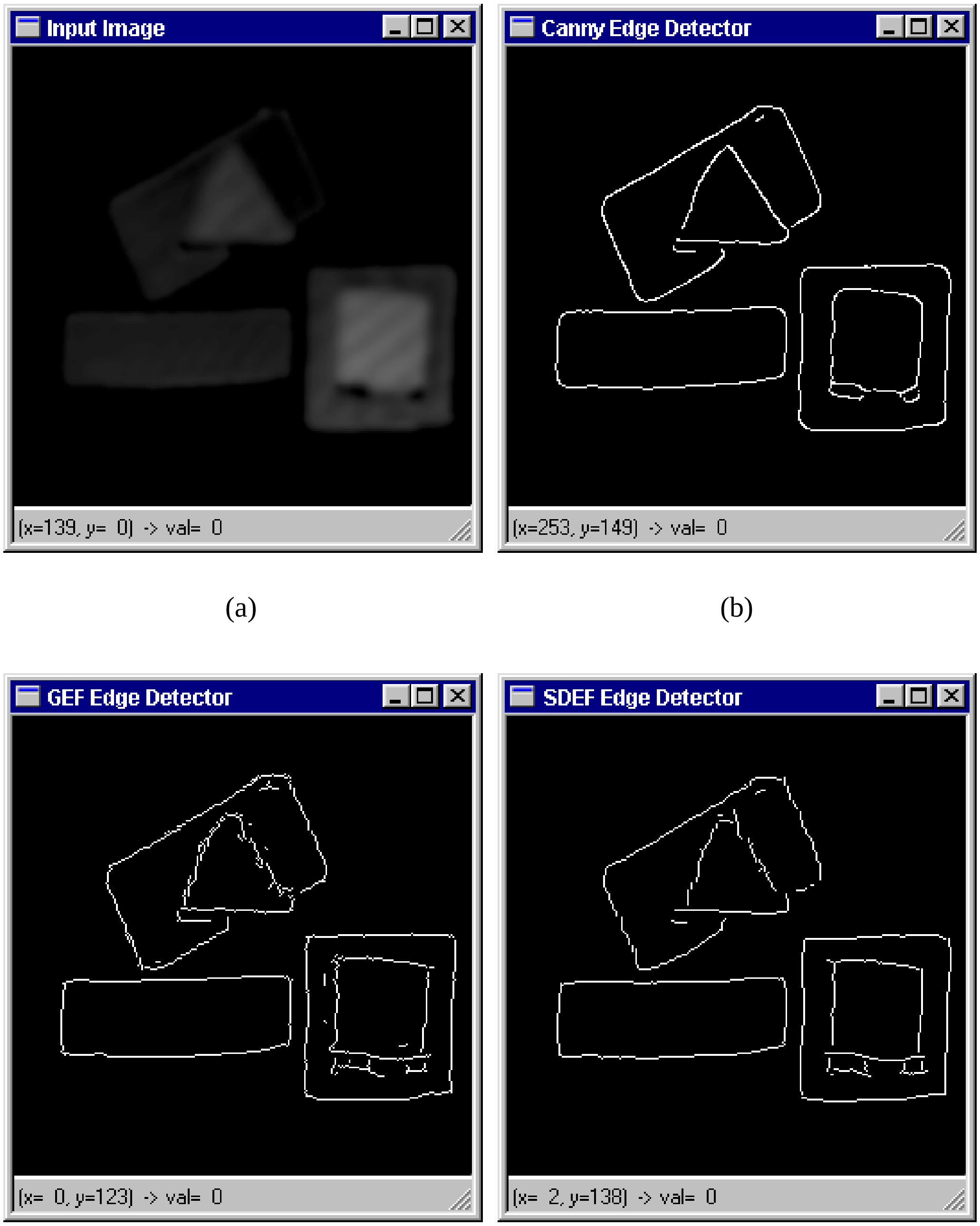
(b)

(c)

(d)

Figure 3.27. Edge estimation when the input is a raw image. (a) The input image. (b) The output from the Canny edge detector (σ = 1.0). (c) The output from the GEF edge detector (a_0 = 0.45). (d) The output from the SDEF edge detector (a_0 = 0.55).

[11] The parameters for these edge detectors were selected for optimal results.



(a) (b)

(c) (d)

Figure 3.28. Edge estimation when the input is a range image. (a) The range image of the scene illustrated in Figure 3.21-a. (b) The output from the Canny edge detector (σ = 1.0). (c) The output from the GEF edge detector (a_0 = 0.5). (d) The output from the SDEF edge detector (a_0 = 0.6).

Figure 3.27 illustrates the edge estimation results when a raw image is analysed. These results can be compared with those depicted in Figure 3.28 when the range image is obtained from the scene described by the raw image. A simple visual comparison indicates that the edge detection is significantly better in the case when the raw image is analysed. In addition, Figure 3.27 illustrates the fact that the quality of edge detection is almost independent of the edge detector employed. In contrast, when the range image is investigated there is a significant difference between the results returned by the Canny and the ISEF edge detectors. There are many rationales that explain these results. As might be expected, the most important one is derived from the fact that the depth discontinuities between objects are not very significant. A different source of errors is generated by the imperfections that are associated with the range sensor, the most important problems being caused by shadows (which can be observed at the bottom of the square object) and a limited resolution of the range sensor.

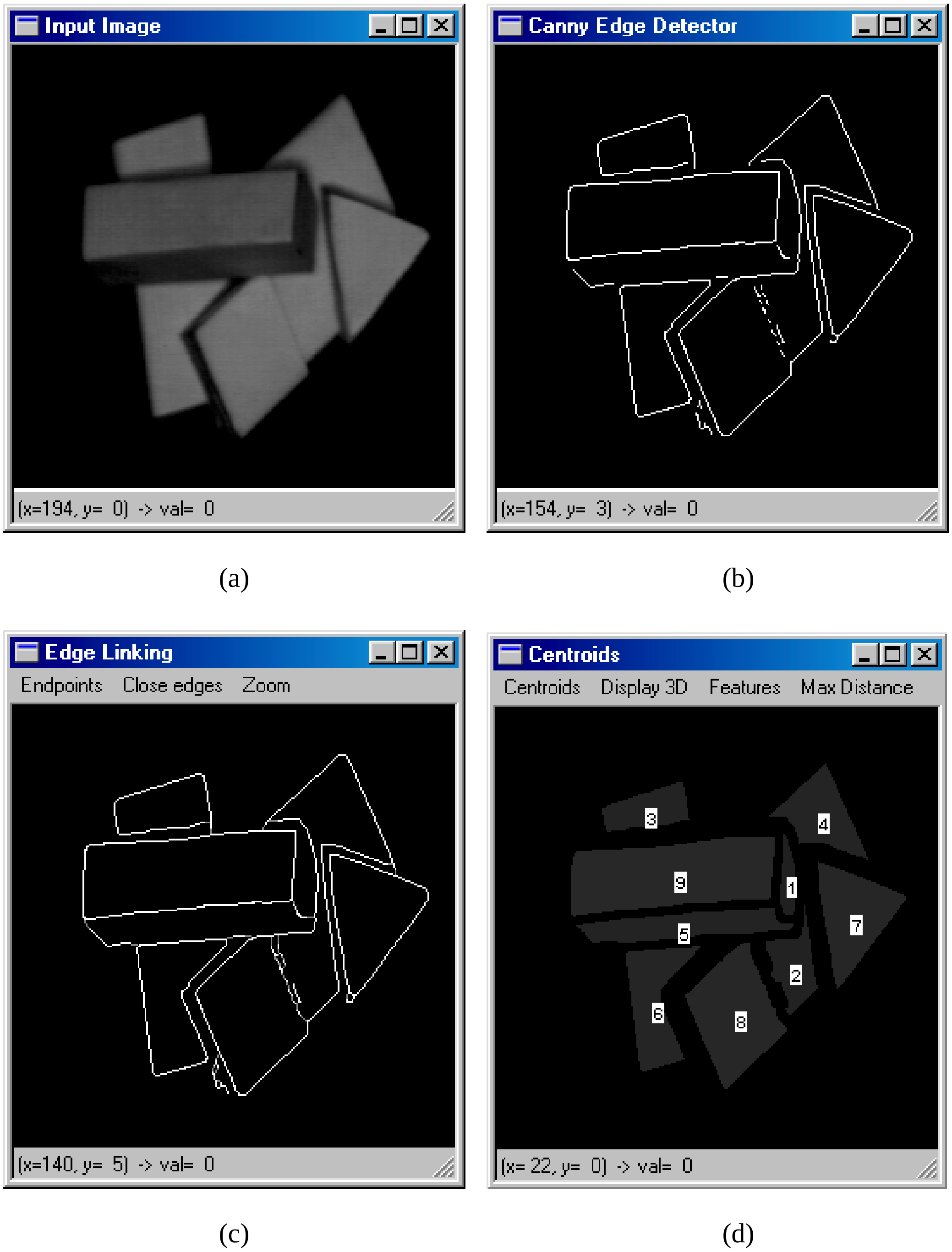
Analysing the edge information returned in both cases, the visual evaluation of the results suggests that the optimal solution relies on the use of raw images as inputs for the segmentation framework outlined in this chapter[12]. Thus, the analysis of the segmentation results that will be presented in the next section will be carried out only on raw images.

3.7 Segmentation results

To evaluate the performance of the proposed segmentation algorithm, it was tested on several images which contain textureless objects. Initially, the algorithm was tested on noiseless images to verify the validity of the proposed segmentation scheme. The next test uses input images which are artificially corrupted with noise for the purpose of evaluating the robustness of the algorithm when dealing with real-world non-idealities.

For these experiments the Canny, GEF and SDEF edge detectors were employed. Figures 3.29 to 3.31 depict the segmentation results when the input of the algorithm is a noiseless image.

[12] This does not mean that is always the case. For example, other similar implementations which involve a higher resolution range sensor or deal with objects with a larger size, more consistent results may be given if the range images are analysed.



```
(x=194, y= 0) -> val= 0
(x=154, y= 3) -> val= 0
(x=140, y= 5) -> val= 0
(x= 22, y= 0) -> val= 0
```

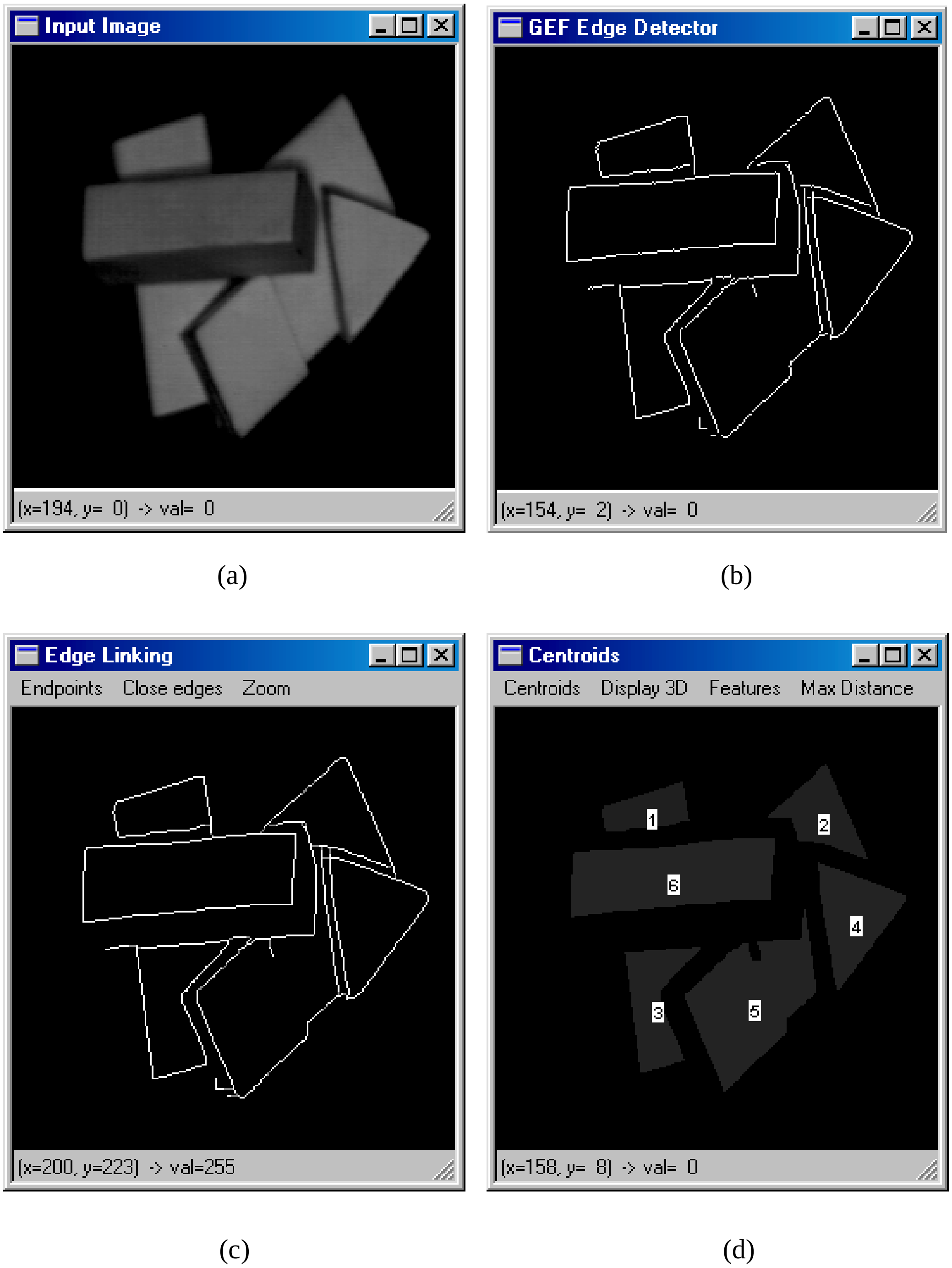
(a)

(b)

(c)

(d)

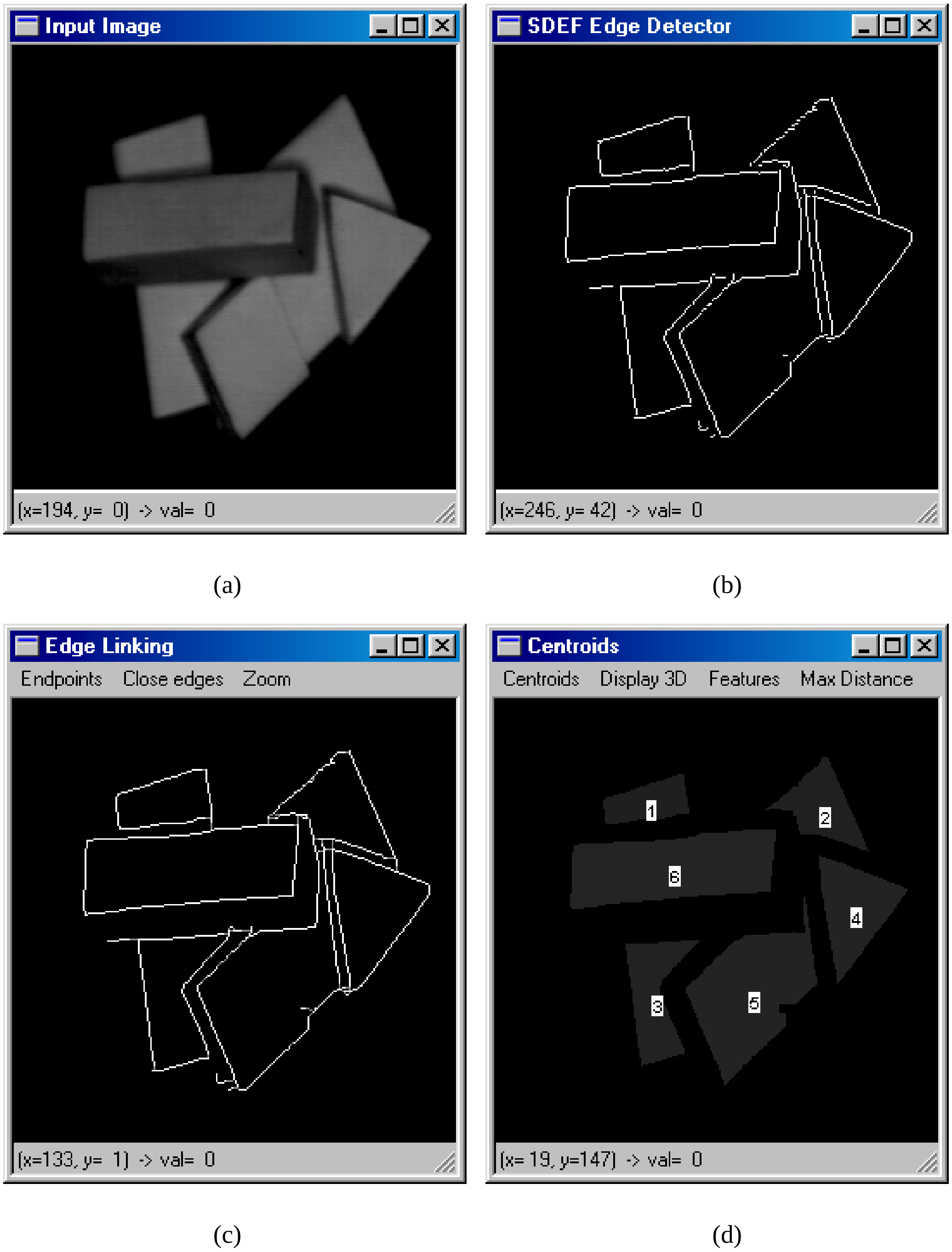
Figure 3.29. Segmentation results using the Canny edge detector. (a) The input noiseless image. (b) The output from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions. Note that a morphological binary dilation with a 3 x 3 square structured element is applied twice to the edge linked image illustrated in image (c).



(a) (b)

(c) (d)

Figure 3.30. Segmentation results using the GEF edge detector. (a) The input noiseless image. (b) The output from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.

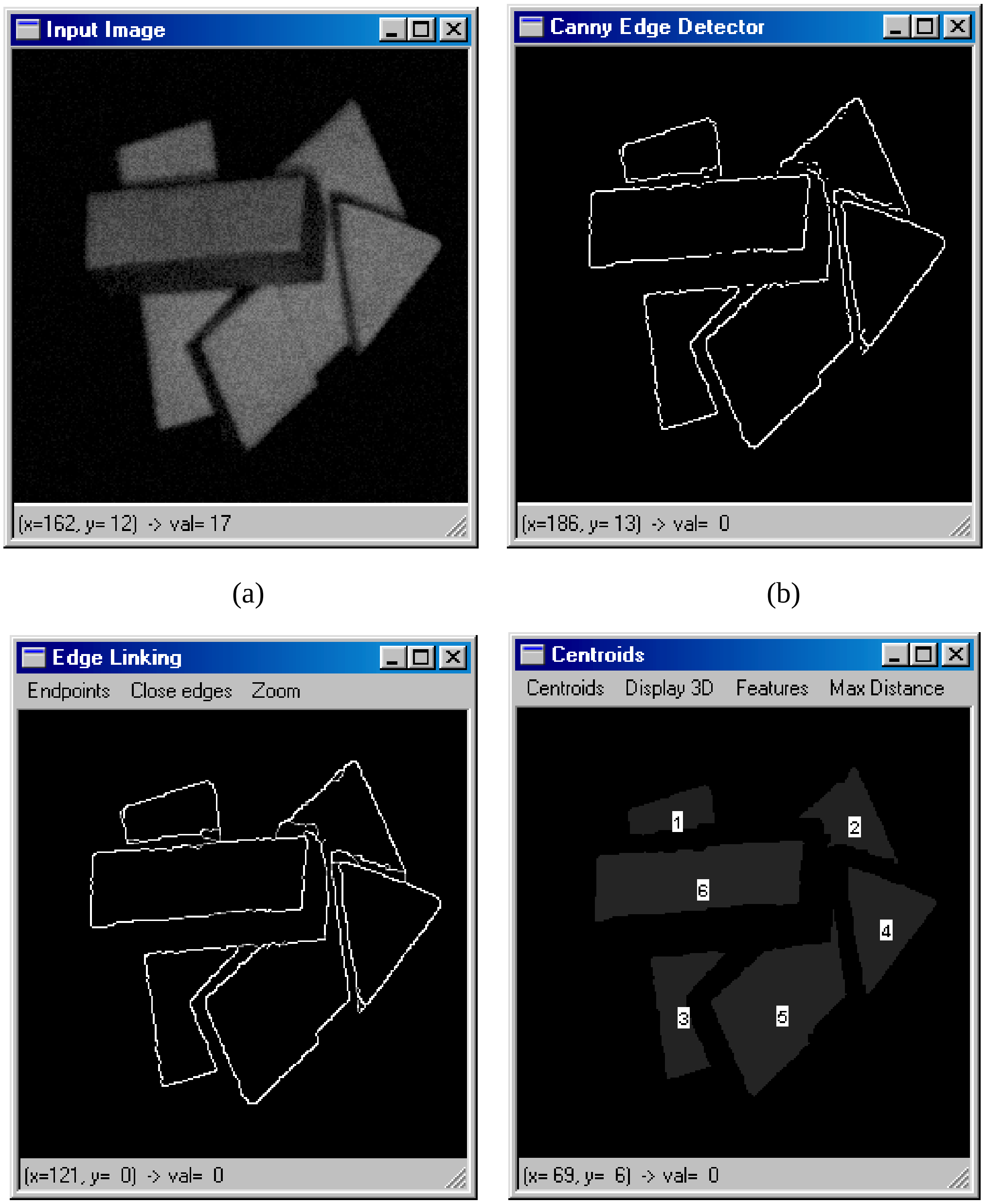


(a) (b)

(c) (d)

Figure 3.31. Segmentation results using the SDEF edge detector. (a) The input noiseless image. (b) The output from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.

It can be noted that the computed regions illustrated in Figures 3.29-d, 3.30-d and 3.31-d are marked according to the number of pixels that are contained in each region. The regions are sorted with respect to the size. When the Canny edge detector is applied to the input noiseless image, the result is a complete meaningful segmentation (all regions are correctly identified). When the ISEF (GEF or SDEF) edge detector is applied, the regions marked with 1 and 5 disappeared because they are not closed by the edge structure.



(a)

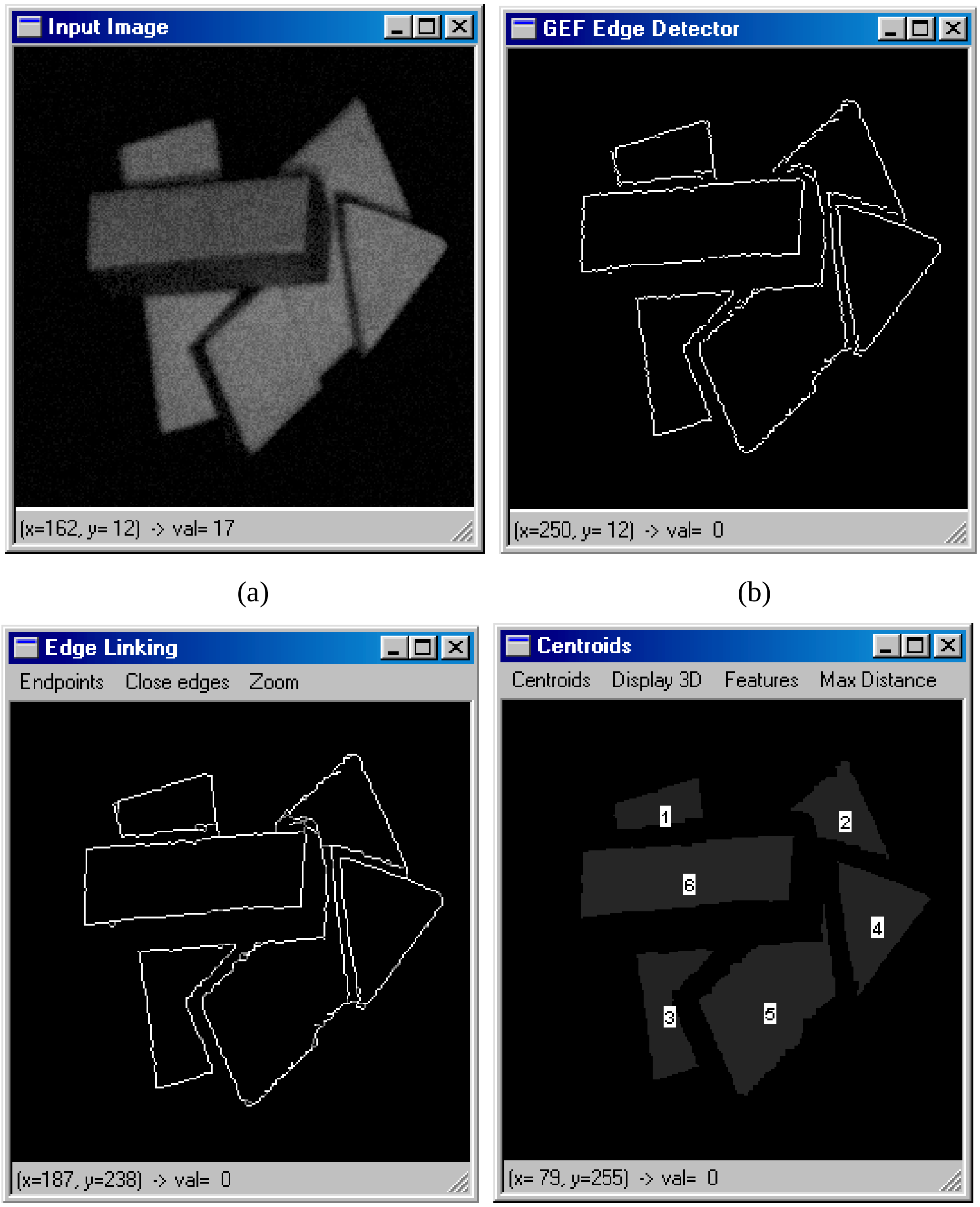
(b)

(c)

(d)

Figure 3.32. Segmentation results using the Canny edge detector. (a) The input image corrupted with Gaussian noise (variance = 30). (b) The output from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.

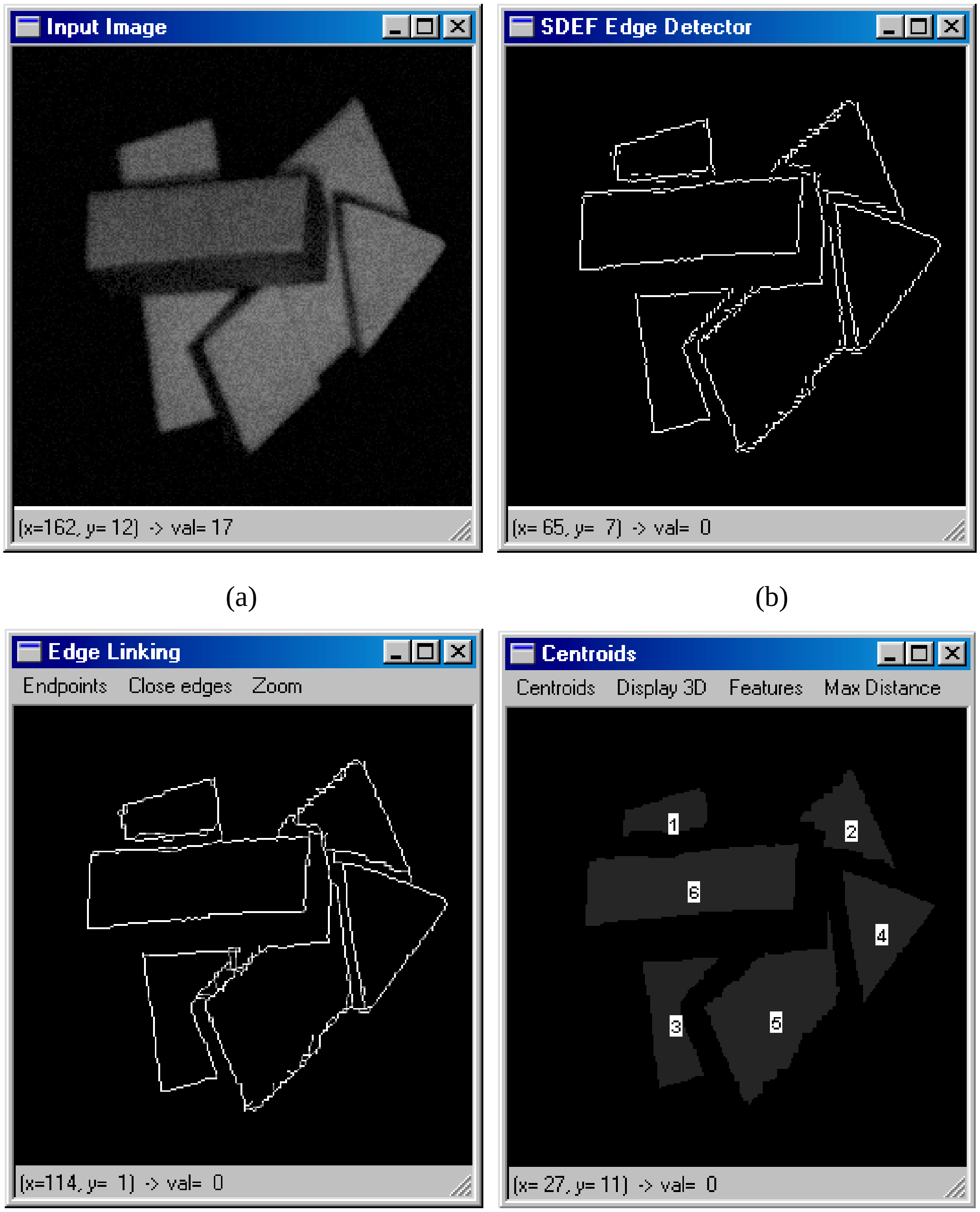
In addition, the regions marked with 2 and 8 in Figure 3.29-d are linked together in Figures 3.30-d and 3.31-d because they are not separated. Figures 3.32 to 3.37 illustrate the behaviour of the algorithm when the input image is corrupted with Gaussian noise (mean = 0, variance = 30).



(a) (b)

(c) (d)

Figure 3.33. Segmentation results using the GEF edge detector. (a) The input image corrupted with Gaussian noise (variance = 30). (b) The output image from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.

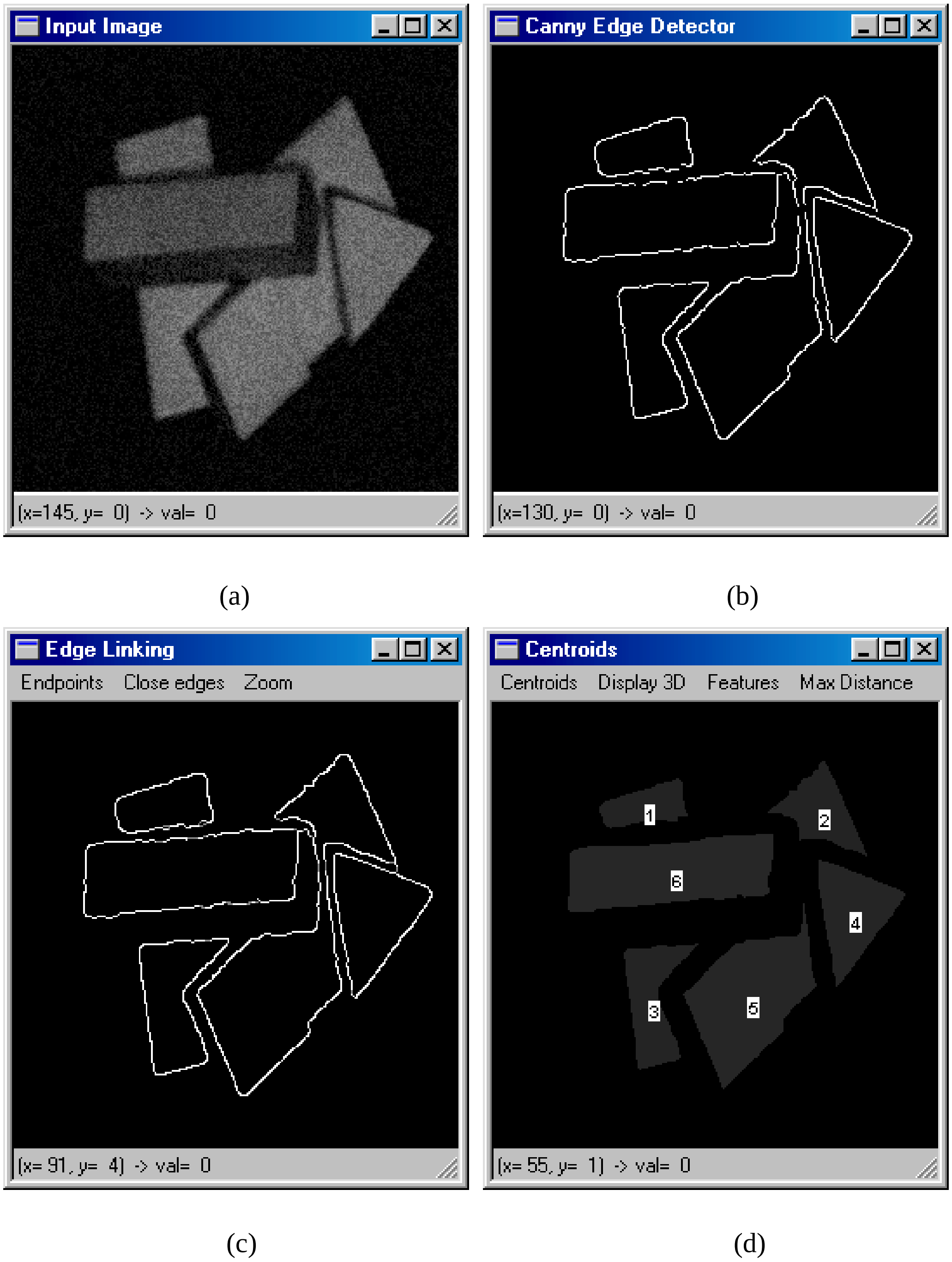


(a) (b)

(c) (d)

Figure 3.34. Segmentation results using the SDEF edge detector. (a) The input image corrupted with Gaussian noise (variance = 30). (b) The output image from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.

Figures 3.32-d, 3.33-d and 3.34-d show that the number of detected regions and their positions within the image returned by the segmentation algorithm is not modified when the input image is corrupted by noise. The next test was to verify the stability of the algorithm when the input image was corrupted with noise to a larger extent (mean =0, variance = 50).



(x=145, y= 0) -> val= 0

(x=130, y= 0) -> val= 0

(a)

(b)

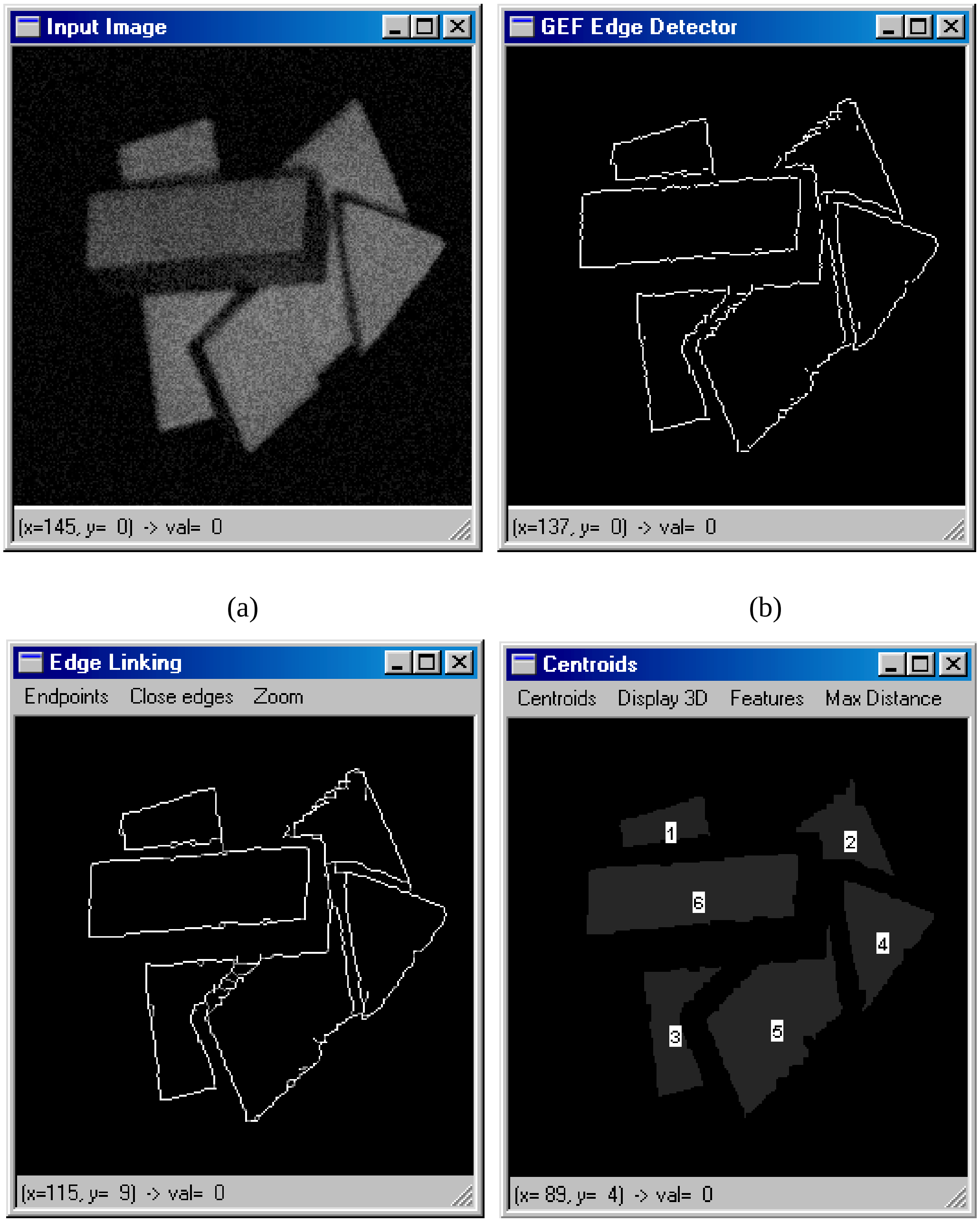
(x= 91, y= 4) -> val= 0

(x= 55, y= 1) -> val= 0

(c)

(d)

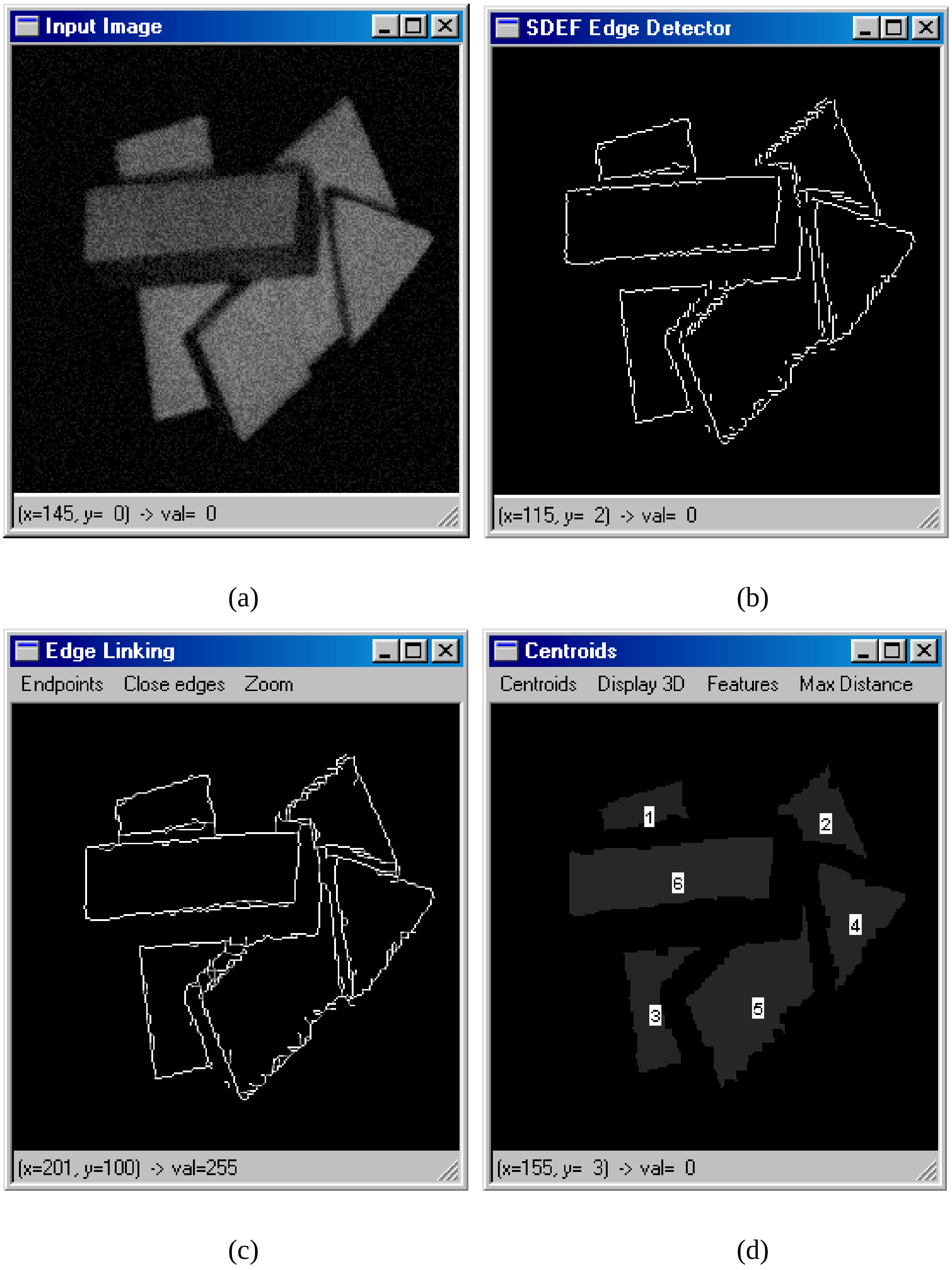
Figure 3.35. Segmentation results using the Canny edge detector. (a) The input image corrupted with Gaussian noise (variance = 50). (b) The output image from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.



(a) (b)

(c) (d)

Figure 3.36. Segmentation results using the GEF edge detector. (a) The input image corrupted with Gaussian noise (variance = 50). (b) The output image from the edge detector. (c) The output after edge linking operation. (d) The centroids of the computed regions.



(a)

(b)

(c)

(d)

Figure 3.37. Segmentation results using the SDEF edge detector. (a) The input image corrupted with Gaussian noise (variance = 50). (b) The output image from the edge detector. (c) The output image after edge linking operation. (d) The centroids of the computed regions.

As can be noted from Figures 3.35-d, 3.36-d and 3.37-d the algorithm was well able to deal with this added complication, the number and the position of the computed regions being unchanged. As would be expected, the borders are not as well preserved (especially for the ISEF edge detectors) but the results are not significantly different from those depicted in the previous cases.

3.8 Discussion

The segmentation process is accurate only if the meaningful regions that describe the different objects contained in the image are precisely separated. A key question is: which method will return better results?

If the scene is very complex and the objects are highly textured the resulting image is over-segmented if there an edge-based approach is employed. In this case, better results are given by region growing techniques. If the objects are textureless and have almost the same colour there is very little information for region growing techniques and the image describing the scene will be under-segmented. The best approach is represented in this case by edge–based segmentation techniques. Because the textureless objects are the topic of this current research, much of the effort has been devoted to identify the optimal edge detector. In terms of efficiency, the Canny edge detector appears to be the best option but unfortunately it is computationally complex. Thus, a trade-off between the quality in edge estimation and the computational efficiency is given by the ISEF edge operators (GEF and SDEF). However, the results returned by the ISEF edge detectors (even when the image was corrupted by noise) are qualitatively close to those returned by the Canny edge detector.

An important issue is reconnecting the gaps between interrupted edges and performing fine enhancements in order to remove the isolate edge pixels that are due to noise. The implementation outlined in this chapter is a morphological-based approach which has two key components. The first component *globally* maximises the edge detection by aggregating the information contained in a small collection of images. The second component attempts to correct the *local* imperfections by exploiting the information around the singular points. The experimental data demonstrates the validity of the proposed segmentation scheme.

Chapter 4 - Object recognition and description techniques

4.1 Introduction

Image-based object recognition has been one of the key subjects of research for several decades. The motivation extends from the observation that the adaptive robotic applications cannot be developed without the help of recognition. In spite of the enormous effort that has been devoted to solve this problem, the field of computer vision has still not produced any clear understanding of how the *generic* recognition should be attempted. Nevertheless, there are many rationales to explain this circumstance and the most important is represented by the distortion of the object's appearance due to occlusions and the fact that the primitives associated with the object's shape are viewpoint dependent. Certainly, the first question that arises is: how does the human visual system accommodate these problems so well? It has generally been accepted that humans are able to recognise real objects contained in scenes with a complex scenario without difficulty. In addition, their ability to recognise 3-D surfaces from 2-D contours is extremely robust even in the case when the contours are incompletely described. Many psychological studies attempt to explain the human visual perception. In this sense, the excellent paper of Edelman (1995) details a psychological experiment in which the subjects were trained to discriminate between two classes of computer generated 3-D objects, the first resembling monkeys and the other dogs. Both classes were defined using an extensive set of parameters which encodes sizes, shapes and placement of limbs. One of the aims of this experiment was to evaluate how the recognition process is influenced when the parameters that describe the object (also called *geons* or parts) are randomly perturbed. A distinct area of interest was to investigate the subjects' performance when the objects were viewed from an unfamiliar perspective.

The results of this experiment were very surprising and can be summarised as follows:

- A recognition model based on the investigation of the geon-structure difference between objects was *not sufficient* to achieve viewpoint invariance.
- The rate of recognition of objects viewed from an unfamiliar perspective is relatively poor.
- As expected, increasing the inter-objects similarities the overall performance decreased accordingly.

One year earlier, Cutzu and Edelman (1994) analysed the usefulness of canonical views in the recognition of 3-D objects. They emphasised that "perceiving the shape of an object irrespective of the viewing conditions such as its orientation in space and its distance from the observer frequently incurs a certain information-processing cost, over and above what it takes to recognise the same object in the most familiar appearance". It is well known that the human visual system has the ability to cope better with some kind of object transformations than others (Biederman and Gerhardstein, 1993). A typical example is represented by scaling when the size of the objects does not affect visibly the recognition rate. In contrast, rotation in depth may have a significant influence on recognition rate. These issues were addressed in their paper when the experiments were conducted on synthetic randomised wire-like objects. The conclusion of the experiment states that the recognition of irregular complex objects relies *at least* in part "on schematic 2-D objects representations and a image-plane shape matching process".

A fundamental question is: what can be learnt from the human vision system? The most obvious answer is that no recognition is possible without knowledge (very often called as *a-priori* information). In this regard, Bergevin and Levine (1993) considered it useful to define the term *generic* recognition. In their assertion, the term generic emphasizes the fact that the recognition process is not based on accurately known object models, but rather on coarse, *qualitative* models representing *classes* of objects. This approach is closely related to the *recognition by parts* (RBC) theory developed by Biederman (1987), a theory that is based on the observation that humans can efficiently recognise objects from simple line drawings.

In contrast with this formulation, the “classic” model-based approaches require a detailed description of the objects contained in the model database. Nevertheless, the generic approaches are conceptually very attractive but unfortunately they have a single proof of feasibility: human vision. Thus, this approach may be useful to those working in perception psychology and computer vision, but it *is not* as relevant to vision researchers trying to implement industrial systems.

Along with knowledge understanding, the issue of object representation plays a central role in object recognition. The importance and the necessity to describe precisely the shapes associated with real objects were highlighted in Henderson’s (1983) paper. Historically, the shape properties used in object recognition were computed in 2-D. Ideally, the shape would be described by viewpoint invariant primitives (features). In some cases, invariants like circle transform into an ellipse when they are viewed from a non-perpendicular direction, a transformation that is relatively easy to evaluate. Also, other shape descriptors such as line segments, junctions or planar polygons show some projection invariant properties. These features are very suitable to describe planar objects when the distortions caused by the observer position or object occlusion are not considerably significant. Unfortunately, this assumption is very restrictive and not always the applications can be constrained to meet these conditions. In this regard, Kak and Edwards (1995) suggested to choose the shape descriptors after a detailed analysis of the shape recognition problem in the context of the given application and to decide whether or not the 2-D object representation provides sufficient information. For some real objects the 2-D representation may bear enough information for recognition while for other the 3-D representation should be considered. The main problem associated with the 2-D approaches is the fact that an image is a 2-D projection of a 3-D scene and there is a significant loss of information. There is no doubt that *conceptually* the 3-D object representation should remove the problems associated with the 2-D representation. Experience has demonstrated that in practice the problems related to the extraction of the 3-D volumetric primitives are more complex than initially believed. As mentioned in Section 3.6.3 the quality of the range sensor and the relative depth between objects will limit the applicability of this approach.

Independent of object representation, the recognition process refers to the classification of the primitives (features) that describe the objects of interest. Formally, the object recognition process can be divided into two main operations. The first consists of extracting the primitives contained in a raw or a range image while the role of the second operation is to classify the objects in classes.

A typical recognition scheme is illustrated in Figure 4.1. The first block (“Extract object features”) implements the first stage of the algorithm and is closely related to the type of object representation employed, while the second block (“Classifier”) assigns the feature vector (or pattern) to a class.

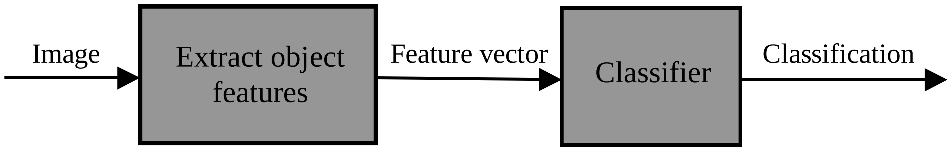


Figure 4.1. The block diagram for object recognition.

The classifier is the central part of the recognition scheme outlined in Figure 4.1 and can be seen as a deterministic machine that assigns the feature vector in question to a specific class in agreement with the decisional rule employed. It is very important to note that the result of classification is not always correct and a natural aim is to minimise this state by choosing the optimal classification parameters.

The operation that adjusts the classifier parameters using a well-defined set of examples (training set) is very often referred in the literature to as classifier learning stage. The training set consists of a collection of feature vectors that describe the models contained in the database and each feature vector is accompanied by information about its proper classification. The operations required in the classifier learning stage are depicted in Figure 4.2 and basically consist of a trial of the training set where the results of classification are assessed by a *supervisor* which adjusts the classifier parameters accordingly. The quality of the learning process is highly dependent on the accuracy of the training set. Naturally, the training set should be fully representative for the classes that are represented. The learning process is finished when the criterion with respect to the classification precision is upheld.

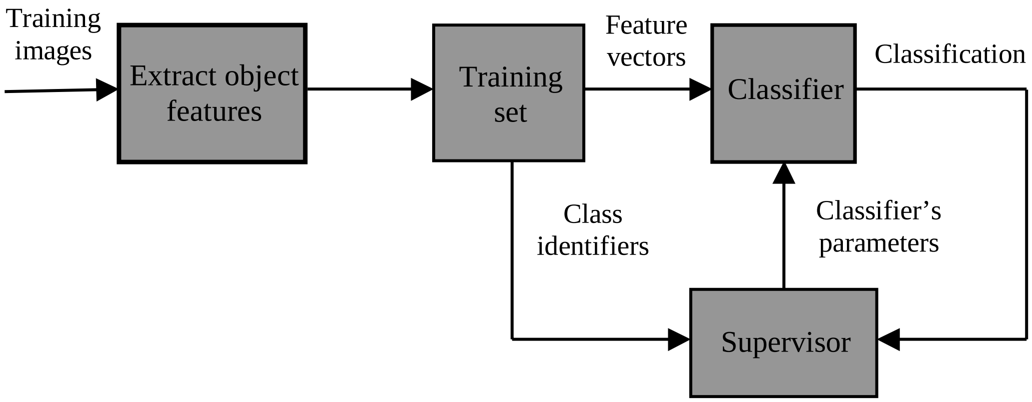


Figure 4.2. The classifier learning stage.

At the end of the learning stage, the classifier is able to partition the feature space using so-called "discrimination" surfaces into disjointed regions. Figure 4.3 illustrates the pattern space for four separable classes.

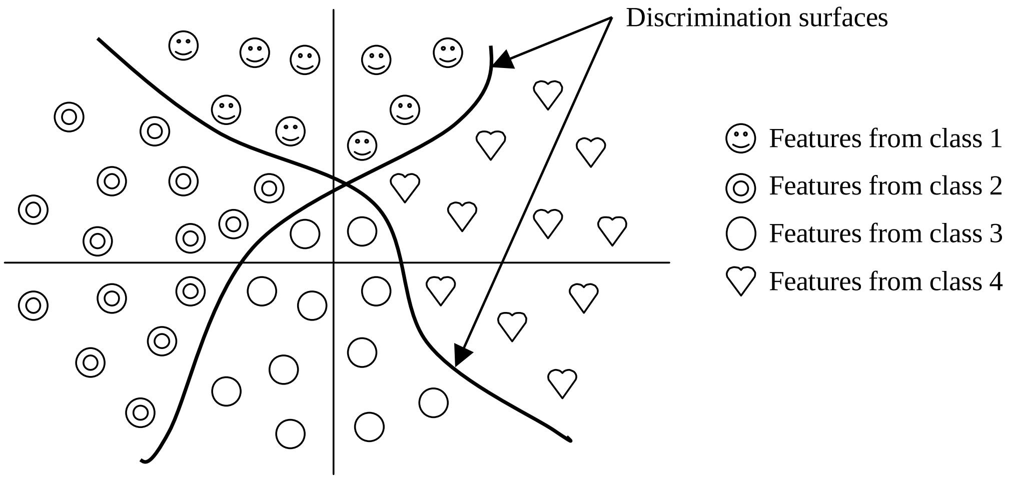


Figure 4.3. Feature space for four separable classes of objects.

Ideally, each resulting region will contain only feature vectors belonging to a single class but in some cases due to the difficulty associated with choosing the optimal training set, some regions will be incorrectly defined and therefore certain input feature vectors will be misclassified.

4.2 Analysis of the issues related to object recognition

The problems such as model acquisition and representation, feature extraction and matching represent the key issues related to object recognition. All of these issues are closely related, as the type of feature extraction employed depends directly on the object representation scheme adopted. As noted earlier, the problems associated with object representation play a central role in object recognition. In this regard, a number of object recognition schemes have been proposed to address this problem using 2-D views. Initially, as suggested in the paper by Lamdan *et al* (1988), the recognition scheme consists of analysing and matching simple 2-D features such as line segments, corners, junctions etc. The justification for using this approach is that these primitives are easy to determine and are fairly robust with respect to viewpoint invariance. Intuitively the main drawback is the large number of hypotheses created, a fact that makes the process of feature matching inefficient. An effective way to overcome this issue was presented in the paper by Forsyth *et al* (1991) and relies on grouping and indexing the features that are associated with an object model. The grouping and indexing mechanism is based on the observation that a number of features are extracted together when the model is analysed. Nevertheless, this approach produces a clear improvement in terms of reliability and efficiency but some problems such as partial occlusion when some of the features are unavailable and an untrustworthy and expensive feature verification procedure restricts the applicability of this recognition scheme.

Much recent debate has focused around including high-level primitives in the recognition process. The early work on inferring the 3-D shape from 2-D views was focused on analysing the *projections* of volumetric primitives. Commonly used classes of volumetric primitives include polyhedra, generalised cylinders and super-quadrics. The recognition schemes based on the use of these primitives were very successful when the scene was restricted to objects with simple geometrical properties. As we might anticipate, an inherent problem is the recovery of these primitives when dealing with occlusion. An efficient solution to compensate for this problem was proposed by Dickinson *et al* (1992). In their implementation they employed a set of ten primitives called geons which are augmented with a hierarchy of their component features (see Figure 4.3).

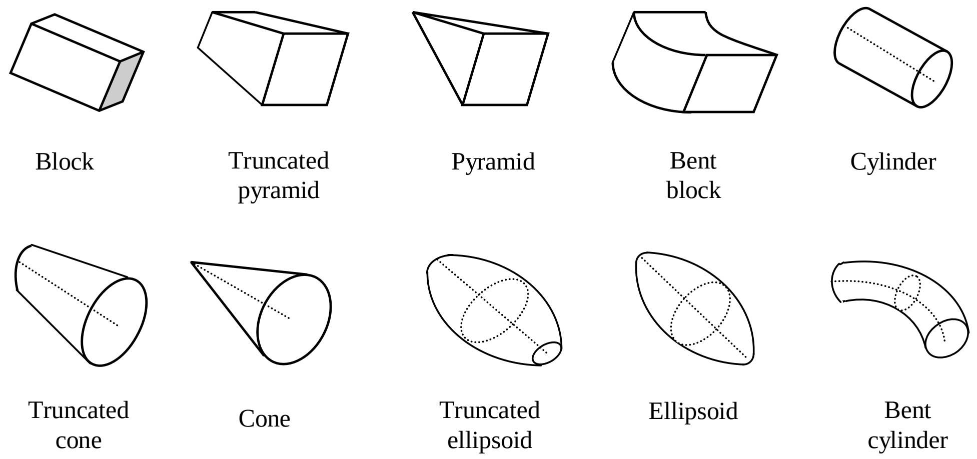


Figure 4.4. The set of volumetric primitives (from Dickinson *et al*, 1992).

The developed probabilistic framework (also called an *aspect hierarchy*), decompose the geons in subcomponents in a hierarchical manner with the purpose of increasing the formulation's robustness when dealing with occlusion. This observation is correct because due to occlusion some of the subcomponents can be partially or completely missing. At this stage a careful analysis is required to select a limited number of qualitative subcomponents in order to maintain a computationally efficient verification process. Consequently, the authors concluded that an appropriate primitive representation should be organised on three hierarchical levels (see Figure 4.5). At the top level of the aspect hierarchy reside *aspects* which completely describe a distinct primitive. As can be seen in Figure 4.4, the aspects consist of a collection of 2-D faces. Due to occlusion some of the faces may be unavailable and this observation introduces the motivation of the second level of the hierarchy. Thus, *faces* which describe 2-D closed contours are placed at the second level of the aspect hierarchy. Again due to occlusion some contours can be only partially recovered and therefore the aspect hierarchy has to be completed with the lowest level. *Boundary groups* represent the third and the last level of the hierarchy and consist of a collection of lines and curves that are parts of the closed contours.

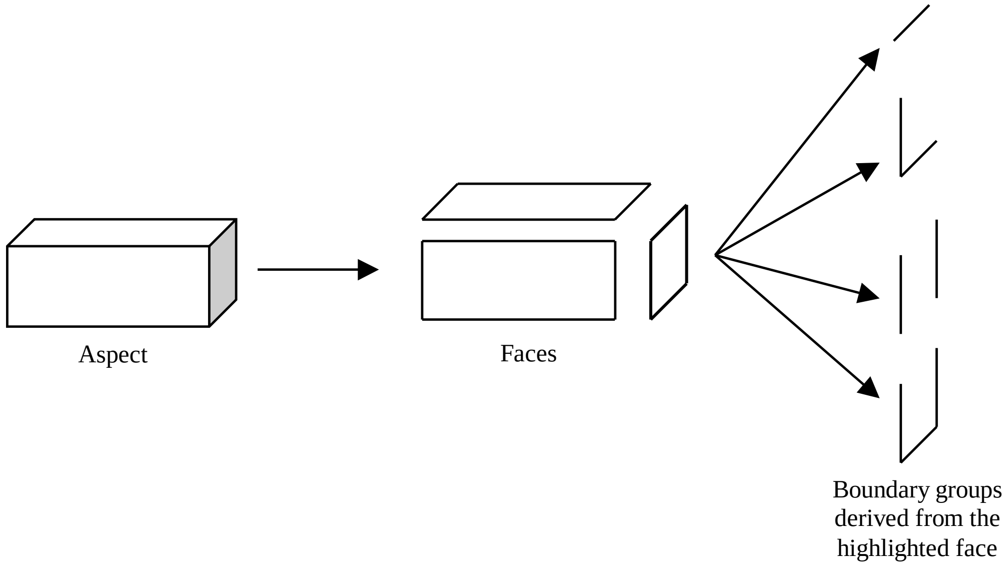


Figure 4.5. Primitive representation using the aspect hierarchy (modified from Dickinson *et al*, 1992).

The primitive representation illustrated in Figure 4.5 is very appealing because the resulting graph that describes the links between subcomponents has great viewpoint independence. In addition, this approach is convenient because it supports the creation of probabilistic rules for inferring complex primitives from relatively simple features. However, this formulation is accompanied by some problems such as a complex bottom-up primitive extraction scheme. In addition, the ambiguities created by occlusions can generate some problems when the groupings are verified to match a model from the database. Also, an inherent disadvantage is the fact that the aspect hierarchy is derived from a rather small number of primitives, a fact that makes this approach particularly appropriate to recognise *textureless* objects with distinct faces. Unfortunately, the objects to be analysed do not always meet these conditions. If the objects are highly textured or their shapes contain many irregularities, the effort to extract the meaningful primitives is immense. Consequently, the scheme developed by Dickinson *et al* (1992) may not be appropriate to handle the recognition of such objects.

To overcome this problem the recognition process can be formulated as one of matching the appearance rather than the shape. In this regard, Murase and Nayar (1995) developed a PCA appearance-based recognition system suitable for recognising and estimating the position of complex objects from 2-D images. This approach is suitable for the recognition of multiple objects but is not able to handle occlusion.

More recently, Ohba and Ikeuchi (1997) proposed to divide the appearance into small windows and to apply eigenimage analysis to each window. Nevertheless, for a large database the number of windows that are involved in the recognition process is very large and a framework using criteria such as detectability, uniqueness and reliability was developed in order to select only the relevant windows. The principle of this algorithm is illustrated in Figure 4.6.

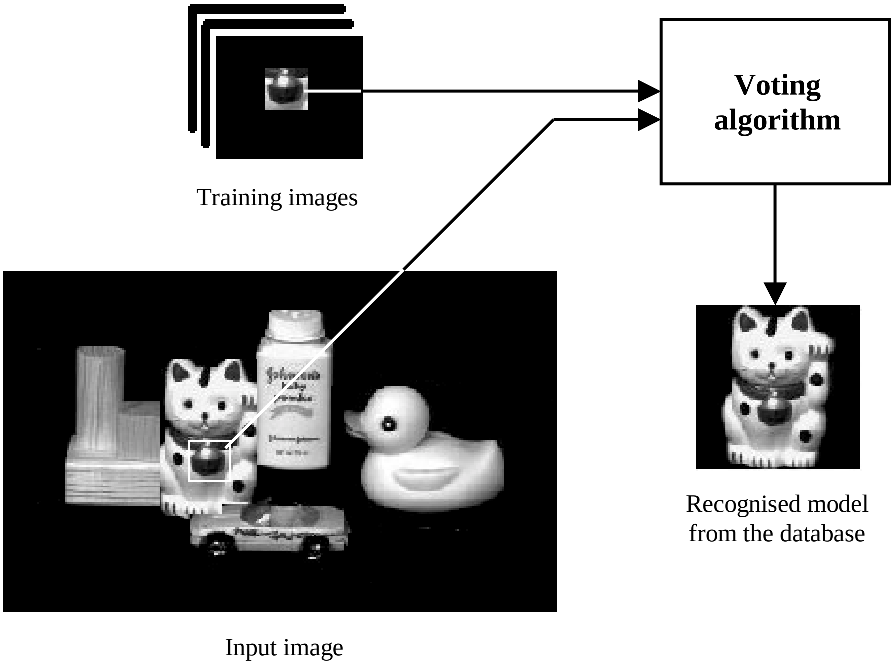


Figure 4.6. The principle of the recognition scheme based on matching the local appearance (modified from Ohba and Ikeuchi, 1997)[13].

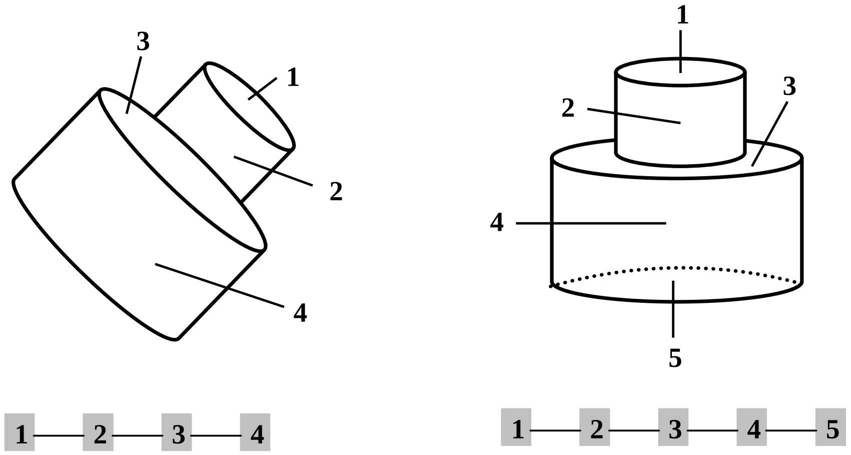
A conceptually related approach was proposed by Schiele and Crowley (1996) when they employed the *multidimensional receptive field histograms* to match the local appearance.

[13] The scene is synthetically created with objects from Columbia Object Image Library (COIL-20). This database is available at: http://www.cs.columbia.edu/CAVE/research/softlib/coil-20.html

In contrast with the formulation proposed by Dickinson *et al* (1992), the recognition schemes based on matching the appearance can handle only the recognition of objects with pronounced and different textural characteristics. For the purpose of reducing the disturbance effects, the appearance has to be divided into small windows. It should also be noted that these windows are viewpoint dependent and the recognition algorithm must search all possible positions in order to minimise the mismatch, a fact that restricts the application of this approach to real-time implementations.

During the last decade, many strategies have been developed to recognise 3-D objects using the information contained in range images. Nevertheless, using 3-D information in the recognition process was implied by the observation that the vision applications are geared specifically for dealing with the 3-D world. As for approaches based on 2-D object representations, one of the critical issues in designing a model-based vision system is the selection of features to be employed in the recognition process. In the late 80's, Bolles and Horaud (1987) developed the well-known 3DPO system for recognising 3-D objects from the range data. They proposed a recognition scheme based on analysing the edge information contained in the range image. The first component of the developed scheme performs a low-level analysis of the range data and classifies the resulting edges into two categories, straight and circular, followed by a characterisation of the surfaces that are adjacent to each edge. Using this strategy, a circular edge is expected to be the intersection between a planar surface and a cylindrical surface while a straight edge may be the intersection of two planes. This analysis is completed by indexing all the visible surfaces that are adjacent to these edge features. The high-level analysis is used in the matching and validation stage. In this sense, the hypotheses created by the resulting features associated with the objects to be recognised are verified if they match a model from the database. This system proved to be successful for recognising curved objects but the database is restricted to very few models.

Fan *et al* (1989) proposed an approach to recognise 3-D objects by using the visible surfaces and their topological relationships which resulted after the segmentation of the range image. The resulting surfaces are indexed into a graph, each node being assigned to a visible surface. If two surfaces share a common border the algorithm creates a link between the nodes that describe them. Figure 4.7 shows the surfaces associated with an object in a random pose and the corresponding object model.



(a) (b)

Figure 4.7. The surface indexing process. (a) Surfaces and the resulting graph associated with a scene object. (b) Surfaces and the resulting graph associated with the corresponding model from the database (modified from Fan *et al*, 1989).

As can be observed in Figure 4.7-b, all the surfaces that are associated with the object model are known while in the case of the scene object due to self-occlusion the region that is marked with 5 is not visible and therefore is not taken into account. In practice, the situation is even worse when in addition to self-occlusion the resulting graph is further disturbed as a result of an imperfect surface segmentation or more often due to occlusion caused by other objects from the scene. Consequently, the matching algorithm based on graph searching techniques has to handle such errors in description, a requirement that is not always easy to achieve. This approach proved to be very successful especially when the objects in question are polyhedral or have a sufficient number of distinct surfaces. Recently, Johnson and Hebert (1999) proposed a framework for simultaneous recognition of multiple objects. Their implementation is based on matching local 3-D features using so called spin images. The spin images are local 3-D descriptors which depend on the object surface curvature. This approach is very appealing because does not require any surface segmentation whilst only local features are employed. Nevertheless, the main problem associated with this approach is the large number of spin images that have to be analysed. In this sense, to reduce the computational overhead associated with the matching algorithm, the authors employed PCA to compress the spin images contained in the model database. As opposed to the approach developed by Fan *et al* (1989), this technique is suitable to recognise objects with complex shapes. In addition, the rate of success is highly influenced by the quality of the range sensor.

In this section the most relevant techniques that are the basis of discussion for the present implementation were detailed. Also, a number of popular techniques for object recognition based on 2-D object representation will be discussed in Appendix B. This discussion is continued in Appendix C where some relevant techniques based on analysing the range images are presented.

4.3 Current implementation for object recognition

The aim of the research outlined in this thesis is to develop a vision sensor for bin picking suitable for use in an integrated sorting/packing industrial environment. As noted by Batchelor (1991), more than 75% of industrial applications are in small to medium batches. This observation is further strengthened by the fact that 98% of products are made of fewer than 25 parts (Delchambre, 1992). However, in the design of an industrial vision system, criteria such as speed and precision has to be considered. From the previous considerations, for a practical system an efficient solution to this problem is to recognise the object first because the number of objects contained in the database is relatively small. Then, the orientation can be addressed at a later stage. The proposed recognition framework consists of analysing the resulting regions obtained after the application of the segmentation algorithm while the pose is precisely estimated using eigenimage analysis. Conceptually, the approach described in this thesis is related to the work detailed in the paper by Dickinson *et al* (1992). As described in the previous section they employed a hierarchical primitive representation to address the recognition of objects in scenes containing clutter and occlusions. A natural question is: why only use surfaces as primitives for recognition? The answer is based on experimental results when the behaviour of each component contained by the hierarchical representation is analysed when the scene contains self and mutual occlusions. At the highest level of the hierarchy reside aspects which are very complex primitives but unfortunately they are very sensitive to occlusion. Furthermore, they are appropriate for describing objects with simple geometry.

Boundary groups are located at the lowest level of the hierarchy. These groups are very appealing to use when the scene is heavily occluded. Unfortunately, they are very ambiguous and the model-to-scene verification procedure is computationally intensive, a fact that restricts their use from real-time implementation. Therefore, the best solution is to use faces (referred to as regions in this thesis) as primitives in the recognition process.

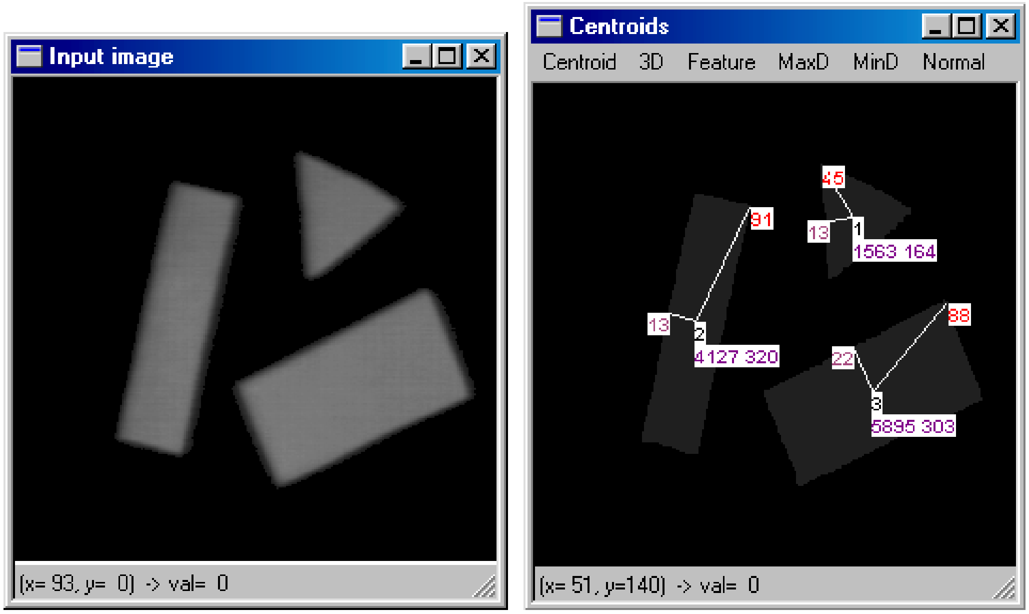
Also, an important issue is related to object representation. As detailed in Section 3.6.3 the objects of interest are small and consequently the relative depth revealed in the scene is not very significant. The analysis that was carried out on various scenes contributed to the conclusion that for this implementation the 2-D object representation may be the appropriate solution. Thus, the 3-D information is required only to select the most suitable regions and to estimate their pose.

In Section 4.3.1 the developed recognition scheme which analyses the regions resulting from the segmentation process is introduced. This discussion is continued in Section 4.3.2 where the pose estimation algorithm based on an eigenimage analysis approach is detailed.

4.3.1 Region-based object recognition

The aim of this section is to analyse the benefits associated with the use of regions (surfaces) that are related to the shape of the object as primitives in the recognition process. In the approach developed by Fan *et al* (1989) (see Section 4.2) the objects are described in terms of their surfaces. The surfaces resulted after the range image is segmented are indexed and the recognition process consists of graph matching. Kim and Kak (1991) extended this approach when they associated an attribute to each region. For example, the attribute for a planar surface is its centroid, while for a cylindrical surface attributes such as radius, centroid and axis are employed. Nevertheless, this formulation is very efficient when the objects have many and regular faces. If the objects in question have only a small number of faces this approach becomes inefficient.

An alternative approach that can handle such situations, consists of analysing the features that can be derived from the geometrical characteristics of the regions. The choice of the types of the features that are used depends on the reliability of their measurement. The criterion employed to select the optimal subset of features has to take into consideration factors such as consistency, accuracy and computation complexity. Another issue of interest is to analyse the immunity of the features in question when viewpoint changes occur. In this sense, the local features such as junctions, lines, strong corners, segments of the object's contour appear to be better suited but as mentioned earlier, these features are very ambiguous and the number of hypothesis created in the verification stage is extremely large. Also, the effects of self and mutual occlusion hinder the applicability of this approach. In contrast, the global features derived from the scene's regions offer good viewpoint invariance and the degree of ambiguity is drastically reduced. It should be noted that these features describe globally the regions and consequently this approach is efficient only if the region of interest is mildly occluded. Using the aforementioned criteria, for the present implementation features such as area, perimeter, the maximum and minimum distances from the region's centroid to the region's border are chosen. Figure 4.8 depicts the features employed in the developed recognition scheme.



(a) (b)

Figure 4.8. The region-based recognition process. (a) The input image. (b) The resulting image containing the geometrical features employed (the first row's figure represents the rank of the region with respect to the area, the second row contains the region's area and perimeter while the remaining figures indicate the maximum and minimum distances from the region's centroid to the region's border).

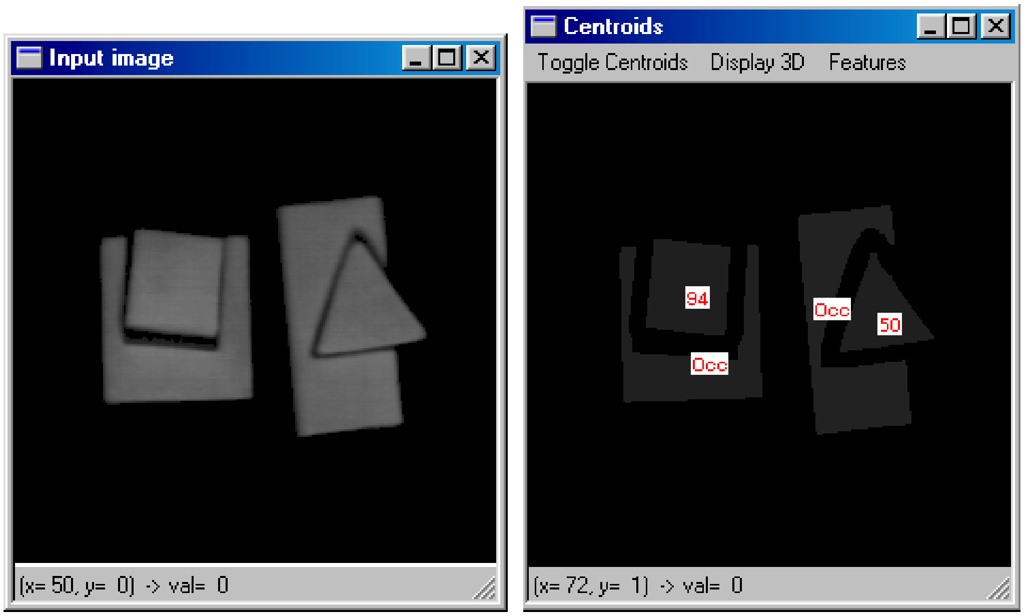
Because this implementation addresses a bin picking application where the scene changes each time an object is picked, at a certain time *one object* is of interest and this object has to be *on the top* of the object pile in order to allow easy manipulation. Certainly, at this stage an important role is played by the 3-D information that gives useful clues in determining the best placed object from the object pile. The initial stage consists of building the object database using the aforementioned features (region's area and perimeter, the maximum and minimum distances from the region's centroid to the region's border) for every object of interest. The algorithm was developed to deal with a variable number of features, each feature being normalised in order to limit the influence of the dominant features. In order to avoid the situations where the features with the largest values overpower the remaining ones, a range normalisation scheme is performed in which the feature mean is subtracted from each feature followed by dividing it with the feature variance (as described in Appendix D). In this case each feature is standardised to zero mean and unit variance. The recognition stage consists of computing the Euclidean distance between the input region and the regions contained in the database (for more details regarding the Euclidean distance refer to Appendix D).

$$\begin{aligned} dist\_g_i^2 &= [r\_in(area) - r\_dbase(area)_i]^2 + [r\_in(per) - r\_dbase(per)_i]^2 \\ &+ [r\_in(\min\_d) - r\_dbase(\min\_d)_i]^2 + [r\_in(\max\_d) - r\_dbase(\max\_d)_i]^2 \end{aligned}$$

where *i = 1,2,…, N*$_r$ (*N*$_r$ being the number of regions contained in the database). The input region is in the database if the minimum distance that gives the best approximation is smaller than a threshold φ. The value of this threshold was set experimentally and defines the maximum allowable distance for a positive recognition state.

$$\min(dist\_g_i) \le \varphi \tag{4.4}$$

If $\varphi < \min(dist\_g_i) < 2\varphi$ the region could be one of interest and the algorithm invokes a *global search operation*, a situation when the entire database is verified to find the best match (more details in Section 4.3.2.3). There is no doubt that the global search operation is computationally intensive and consequently an important goal is to minimise the cases where this operation is required. To achieve this goal, a framework that deals with a variable number of regions which fulfil the 3-D criteria was implemented. The developed algorithm can identify some regions which do not fulfil the selection criteria and are marked as *Occ*. This situation occurs when the scene reveals obvious occlusions as illustrated in Figure 4.9.



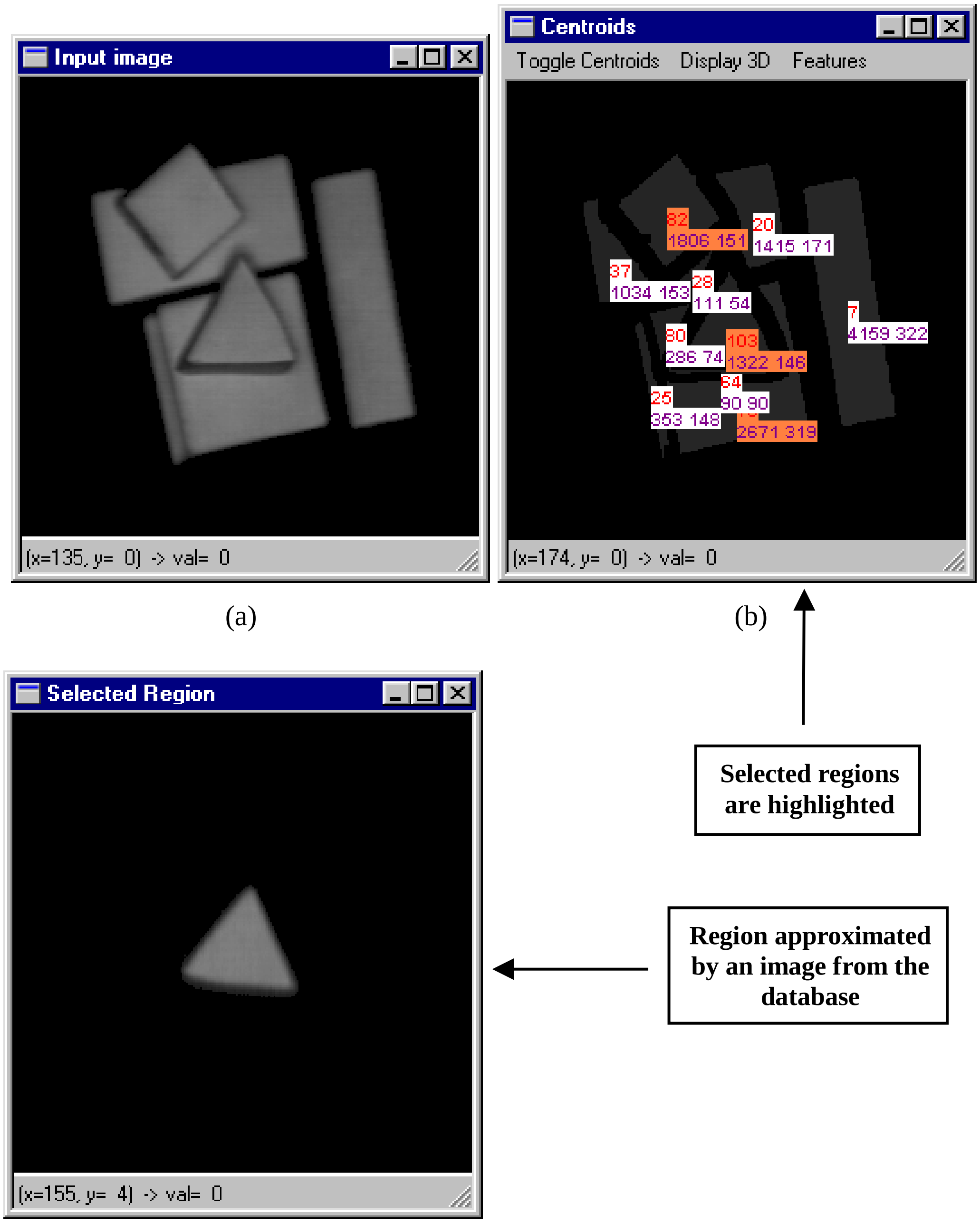
(a) (b)

Figure 4.9. Determining the occluded objects. (a) The input image. (b) The segmented image which highlights the object's elevations. The occluded objects are marked with *Occ*.

The number of remaining regions can be further decreased by applying other selection constraints. For this implementation the selected regions' area have to be bigger than a preset value that is 80% of the smallest region contained in the database. Also, the gripper's mechanical characteristics can impose further constraints such as the space between the object to be grasped and the objects situated in its neighbourhood.

Next, from the selected regions, the one that gives the best approximation with respect to the matching criteria is considered to be on the top of the object pile. If the minimum Euclidean distance for the selected region is higher than the threshold value the algorithm invokes the global search operation. Although this framework does not eliminate the need for the global search operation, the situations when this operation is required are greatly reduced.

If the region is not contained in the database the algorithm provides two options, the first is to remove the object from the pile and the second is to rearrange the scene. An example that shows how the developed framework operates is illustrated in Figure 4.10



(a)

(b)

Selected regions are highlighted

Region approximated by an image from the database

(c)

Figure 4.10. Selecting the best-placed objects. (a) The input image. (b) The resulting image data. (c) The selected region considered to be on the top of the object pile.

4.3.2 Pose estimation using eigenimage analysis

The implementation outlined in this chapter uses an eigenimage analysis approach for pose estimation. This can be briefly described as a method, which computes the eigenspace determined by processing the eigenvalues and eigenvectors of the image set (see also Moghaddam and Pentland, 1994; Murakami and Kumar, 1982; Sirovich and Kirby, 1987; Turk and Pentland, 1991b). For the current implementation, the image set is obtained by acquiring a collection of spatial positions by rotating the objects *around a single axis* and an eigenspace is computed for each object of interest. For the region resulted after recognition, the algorithm projects its image to the corresponding eigenspace and the object pose is estimated according to the number of images contained by the image set (Murase and Nayar, 1995).

To make this approach computationally efficient the image set is obtained by varying pose whilst maintaining a constant level of illumination. A major aim is to minimise the problems associated with the position of the object within the image. It is acknowledged that this approach is very sensitive to the location of the object, therefore to compensate for this problem the objects are centered within the image. Another key problem consists of eliminating the redundant information. In this regard, the image set is normalised in brightness and the background is discarded. For the sake of computational efficiency, the eigenspace for every image set can be constructed by computing only the larger eigenvalues (eigenvectors) of the covariance matrix. The next step involves projecting all the images contained in the image set onto the eigenspace and thereby obtain a set of points that characterise the object's position in space. In order to increase the resolution, the resulting points are connected in eigenspace using a linear interpolation. In this case, intermediary positions situated between two consecutive positions contained in the image set are better approximated.

The procedure described above parameterises the pose by only one degree of freedom (DOF) with respect to the camera. To address the full three DOF pose estimation the image set must sample all possible spatial orientations. This approach is impractical since a very large number of images are required. In contrast, the present approach constrains one DOF using eigenimage analysis as described above, while the remaining two DOF are addressed by analysing the range data, namely by computing the normal to the surface as will be shown in Section 4.3.3.

4.3.2.1 Calculating eigenspace

In general, an image *I(x,y)* is represented by a two-dimensional 256 by 256 array of 8-bit intensity values. Alternatively, this image can be considered as a vector of dimension 65536 when the brightness values are evaluated in a raster scan manner. Whichever representation is used, this information is too large to be used directly for pose estimation. The main concept of the *principal component analysis* (PCA or Karhunen-Loeve expansion) is to find the vectors that can efficiently describe the entire image set. Many research studies concluded that the optimal representational space entails computing the eigenvectors of the covariance matrix associated with the image set. These vectors describe an orthonormal space and this property is illustrated in Equation 4.5.

$$u_i u_j^T = \gamma_{ij} = \begin{cases} 1, & if \quad i = j \\ 0, & if \quad i \neq j \end{cases} \tag{4.5}$$

where u_i, u_j are two eigenvectors and γ_{ij} is the scalar product.

Let I be an image which describes a scene that contains a single object. As previously mentioned this image can be represented by the vector illustrated in Equation 4.6.

$$I= [i_1,i_2,i_3,\ldots,i_N] \tag{4.6}$$

where i_1, i_2,…,i_N are the pixel values. The idea of PCA consists of matching the appearance of the object contained in the image rather than its structural description. It is worth mentioning that the appearance of the object is affected only by its *spatial* position if the illumination condition are maintained constant. In this sense, it is necessary to construct an image set that encodes the *relevant* spatial positions. As can be seen in Equation 4.7 the image set associated with an object is organised as a matrix of images.

$$[I_1,I_2,I_3,\ldots,I_P]^T \tag{4.7}$$

where P is the number of considered positions in space. For the purpose of minimising the effects caused by the variations in the intensity of illumination, each image is normalised so that the total energy contained in the image is unity (Murase and Nayar, 1995). To accomplish this goal, each pixel intensity is divided by B as shown in Equation 4.8.

$$i_n^{'} = \frac{1}{B} i_n , \quad B = \sqrt{\sum\nolimits_{n=1}^{N} i_n^2} \qquad (4.8)$$

In Figure 4.11 is shown a part of the image set obtained by rotating the object manually in small increments. For each image, the object of interest is centered and the background is discarded.

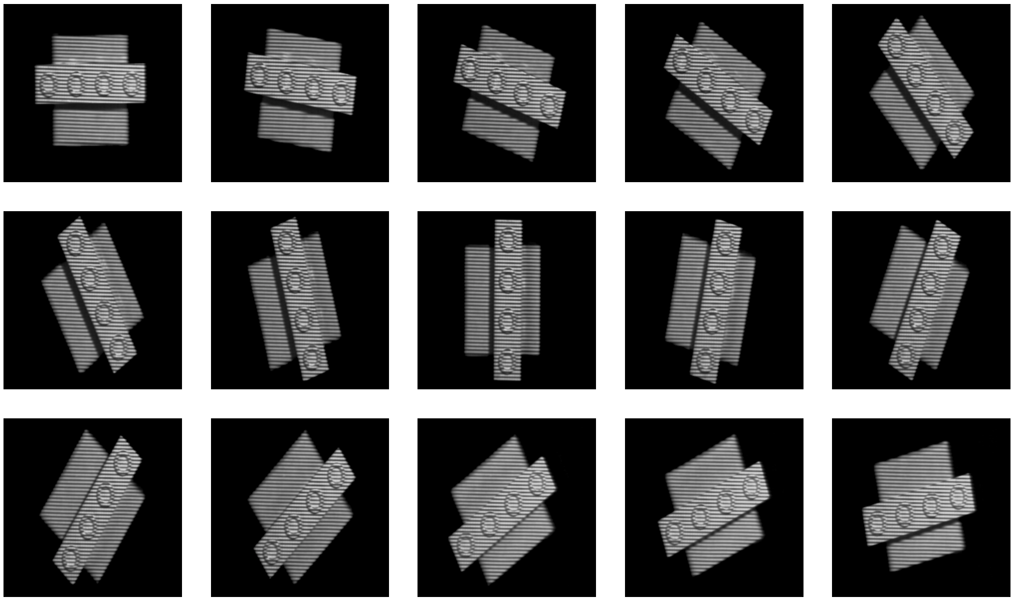


Figure 4.11. Part of the image set obtained by rotating the object around a single axis.

Before computing the eigenspace, it is necessary to compute the average image (*A*) of all the images from the image set:

$$A = \frac{1}{P} \sum_{i=1}^{P} I_i^{'} \qquad (4.9)$$

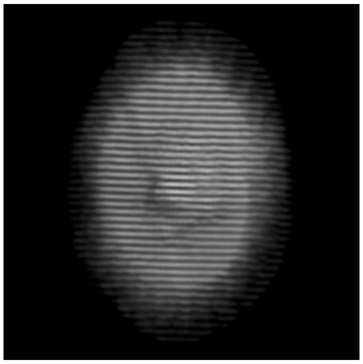


Figure 4.12. The average image of the image set illustrated in Figure 4.11.

The normalised image set will be obtained by subtracting the average image from each normalised image:

$$S = [I_1^{'} - A, I_2^{'} - A, I_3^{'} - A, \ldots, I_P^{'} - A]^T \qquad (4.10)$$

Equation 4.10 indicates that the dimension of matrix S is P x N, where P is the number of positions (images) and N is the number of pixels. The next step involves the computation of the covariance matrix.

$$C = S^T S \qquad (4.11)$$

The resulting matrix illustrated in Equation 4.11 is very large (65536 x 65536) and it will be extremely difficult to compute the eigenvalues and the corresponding eigenvectors. Alternatively, if the number of images P is smaller than N it is easier to construct the P x P matrix using $Q = SS^T$, but in this case the dimension of the space is maximum P. The eigenvalues and the corresponding eigenvectors associated with the reduced covariance matrix Q are computed by solving the following well known equation:

$$Qu_i = v_i u_i \qquad (4.12)$$

where u_i is the i^{th} eigenvector and v_i is the corresponding eigenvalue. It should be noted that the covariance matrix is symmetrical and this property considerably reduces the computational overhead when the eigenvalues are computed. Based on this observation, in the implementation described in this thesis, the eigenvalues and the eigenvectors are computed using the QL algorithm (see Appendix E). This algorithm is preceded by an application of the Householder transform with the aim of obtaining a simple *tridiagonal* form for the covariance matrix (for more details refer to Appendix E). The eigenspace is obtained by multiplying the matrix of eigenvectors (eigenmatrix) with the matrix S:

$$E = US \qquad (4.13)$$

where $U=[u_1,u_2,\ldots,u_P]^T$, U is P x P dimensional and E that represents the eigenspace is P x N dimensional. As an example, Figure 4.13 illustrates the first eight eigenvectors computed from the image set depicted in Figure 4.11.

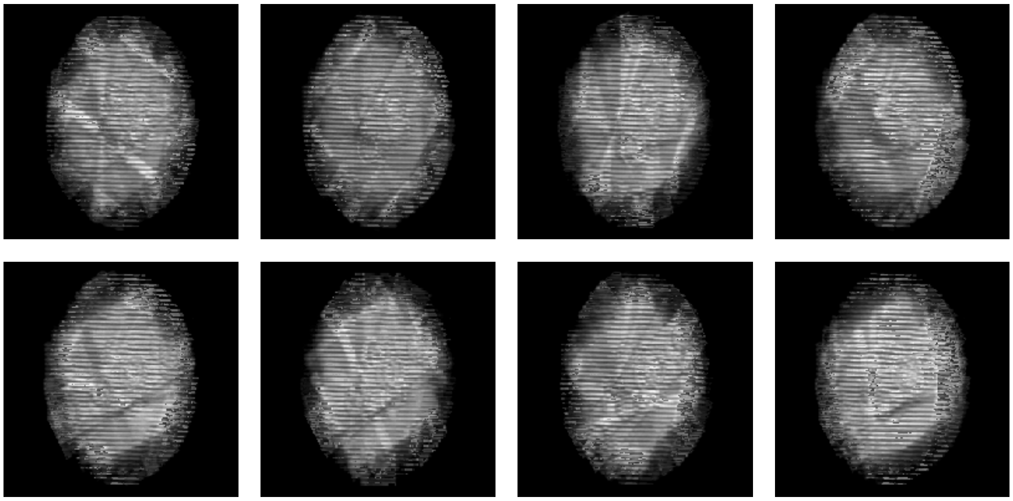


Figure 4.13. Eight of the eigenvectors corresponding to the largest eigenvalues calculated for the input image set.

Using this approach, the number of calculations is significantly reduced, but in this case the number of eigenvectors is relatively small (up to *P*).

4.3.2.2 Position estimation

Once the eigenspace is computed, the next step consists of projecting all images from the set on this subspace ($E=[e_1, e_2,\ldots,e_p]^T$). The result will be a collection of points which describe the object's position. Before projecting the image set onto eigenspace, it is necessary to subtract the average image from each image as illustrated in Equation 4.14.

$$h_i = [e_1, e_2, \ldots, e_P]^T (I_i^{'} - A) \tag{4.14}$$

where e_1, e_2, …, e_p are the eigenspace vectors and are *N* dimensional.

As predicted, each point is *P* dimensional and for the purpose of pose estimation, this allows a fairly accurate method under the condition of maintaining a constant illumination. Moreover since consecutive images are strongly correlated, a method to estimate the intermediary spatial position of the objects contained in the image set can be developed. A simple algorithm was proposed by Nene *et al* (1994) and consists of interpolating the points resulting after the projection of the input image set on the eigenspace. All the points from the PCA database are connected using a linear interpolation into a manifold by passing through the root node while the new points are added to the left child (the left child is the point derived from the root node). This process is applied from left to right until a spatial position is assigned to each point (projection). This continuous manifold depicted in Figure 4.14 allows the possibility to estimate with increased precision the spatial orientations of the object that are not included in the image set.

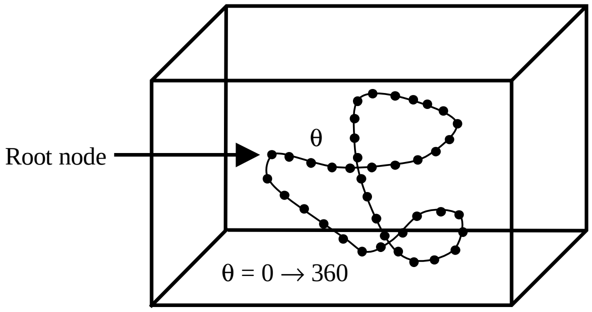


Figure 4.14. The projected points connected together into a manifold using the first three dimensions of the eigenspace.

If an unknown input image is projected on the eigenspace (see Figure 4.15), a *P* dimensional point will result and this vector can be used in a standard estimation algorithm. The simplest method to match an unknown image with an image contained in the image set relies on computing the Euclidean distance using the relationship illustrated in Equation 4.15.

$$d_i^2 = \left\| h - h_i \right\|^2 = (h - h_i)^T (h - h_i) = h^T h - h_i^T h_i \tag{4.15}$$

where $i=1,2,\ldots,P$ and h is the projection of the input image onto the eigenspace.

The input image approximates an image contained in the image set if the minimum distance between its projection on the eigenspace and the points contained in the PCA database is smaller than a threshold value. As in the previous case, the value of this threshold was set experimentally and defines the maximum allowable distance for a positive estimation state.

$$d_i = \min\|h - h_i\| \le \zeta \tag{4.16}$$

If the minimum distance is bigger than the threshold value, then the *general search operation* is invoked, this will be described later.

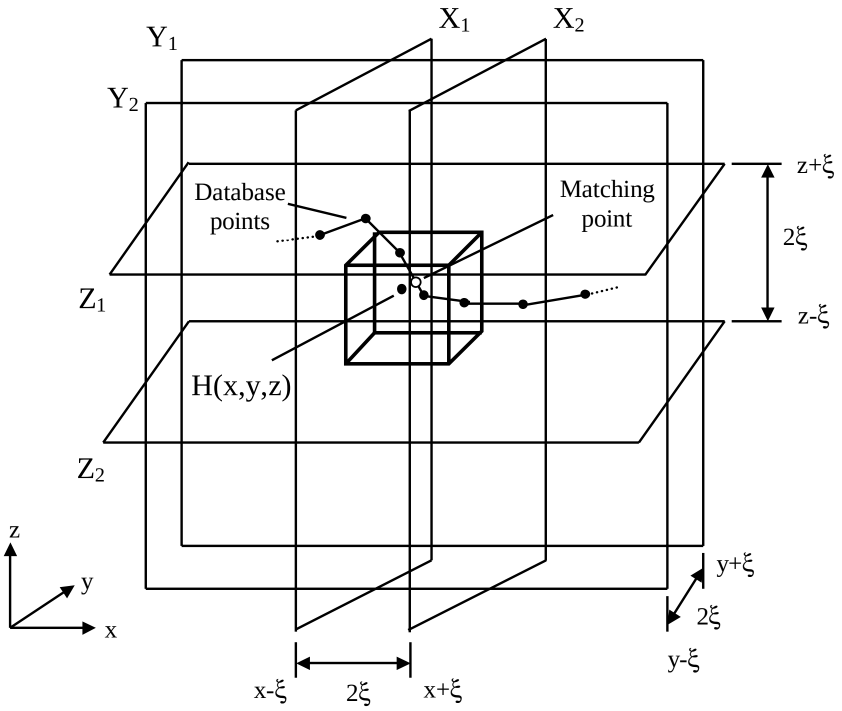


Figure 4.15. The matching process (modified from Nene and Nayar, 1995).

In order to simplify the representation in Figure 4.15 where the matching process is illustrated, the space is partitioned only for the first three co-ordinates *(x,y,z),* while the pose estimation algorithm uses a multi-dimensional space (up to *P*). In Figure 4.15, the matching point is inside the cube of size 2ξ and the corresponding position in space is approximated by the point's position in the hypercube. Generally the input point *H* (with rare exceptions when it matches perfectly a position from the PCA database) will lie between two consecutive points from the PCA database. In case if these points are not connected into a manifold, the input point will match the point from the database that is closest to it. Since the image set for each object contains 24 spatial orientations, the maximum error rate is 7.5 degrees. This error rate is significantly reduced when the database is connected into a manifold. This can be clearly observed in Figure 4.15 when the distance between the input point *H* and the nearest point situated on the manifold is smaller than the distance between the input point and the closest point contained in the database. In this case, the maximum error was reduced to 2.1 degrees when 24 orientations are used.

4.3.2.3 Global search operation

The global search is invoked only if the geometrical constraints are not precise enough or the initial recognition failed. A solution is to compute the universal eigenspace determined by using all images contained in the database. Unfortunately, the universal eigenspace is difficult to compute as the number of images for a large collection of objects is significant. For this reason, the input image is projected onto every object's eigenspace and if the minimum distance is smaller than ξ, the object is correctly recognised and its pose estimated within the algorithm error. Otherwise the object is not contained in the database. Certainly, this operation is slow as long as all possible situations are verified but the situation when the global search is required rarely occurs.

4.3.3 Object pose from 1 DOF to 3 DOF

The eigenimage analysis detailed in Sections 4.3.2.1 to 4.3.2.3 constrains only 1 DOF since the image set is generated by rotating the object around a single axis. In order to address the full 3 DOF object pose it is necessary to generate an image set that captures all possible orientations of the object under analysis. Nevertheless such approach is quite impractical since even at a coarse rate of object pose sampling, it would require a very large image set. Consequently, the 3 DOF object pose has to be reformulated in order to reduce the size of the image set. In this sense, for the present implementation 1 DOF is constrained using eigenimage analysis while the remaining 2 DOF are addressed by computing the normal to the surface resulting after recognition.

4.3.3.1 Computing the normal to a plane

The normal vector of a planar surface gives useful clues regarding the orientation in space of this surface. To compute the normal vector it is necessary to know the co-ordinates of 3 non-collinear points *A*, *B*, *C* that that belong to the planar surface.

These points will generate two independent vectors *p*, *q* (called vertex) and their vectorial product will generate a vector *n* perpendicular on the plane on which the two vectors lie.

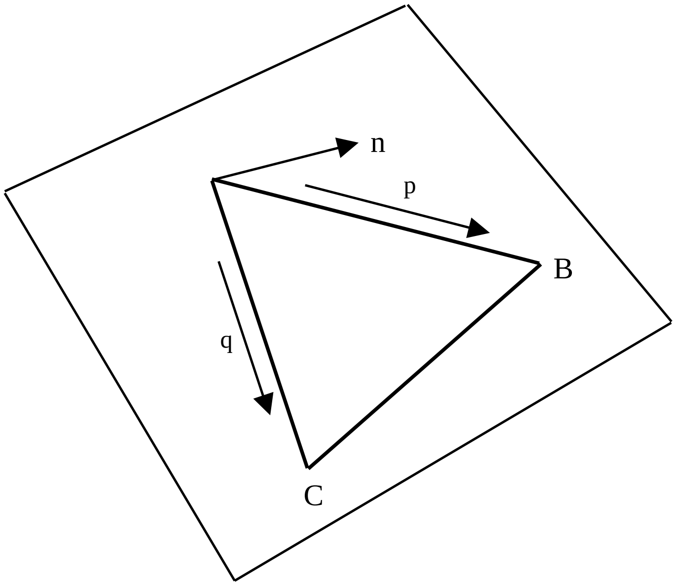


Figure 4.16. The normal vector to a plane.

The vectors *p* and *q* are obtained by subtracting the co-ordinates of point A from the co-ordinates of point B and C respectively. Once *p* and *q* are determined, the normal to the plane in point *A* is given by the vectorial product of the input vectors as illustrated in Equation 4.16.

$$\begin{aligned}
&p = (A - B); \quad p.x = A.x - B.x; \quad p.y = A.y - B.y; \quad p.z = A.z - B.z \\
&q = (A - C); \quad q.x = A.x - C.x; \quad q.y = A.y - C.y; \quad q.z = A.z - C.z \\
&n = (p.y * q.z - p.z * q.y)\vec{i} + (p.z * q.x - p.x * q.z)\vec{j} + (p.x * q.y - p.y * q.x)\vec{k}
\end{aligned} \tag{4.16}$$

where the symbol $*$ defines the arithmetic multiplication and $\vec{i}, \vec{j}, \vec{k}$ are the standard unit vectors. The resulting vector *n* has to be normalised in order to adjust its norm to unity. This operation is straightforward and is shown in Equation 4.17.

$$m = \sqrt{n.x^2 + n.y^2 + n.z^2}; \quad n.x = \frac{n.x}{m}; \quad n.y = \frac{n.y}{m}; \quad n.z = \frac{n.z}{m} \tag{4.17}$$

Another common problem is to check which side of the plane the normal vector is on. If we are looking only for normals pointing outward the surface, then the *z* component of the normal must be positive. Otherwise we have to multiply the *x*, *y* and *z* components of the normal by –1 in order to obtain the orientation of the normal from the other side of the plane.

4.3.3.2 Computing the 3 DOF object pose

Generally, the grasping position is relative to well-defined stable points such as the centroid of the recognised face. Consequently, the normal to the surface where the centroid is situated has to be determined. While the co-ordinates of the centroid are known, in order to compute the normal vector at least other two independent points situated in the same plane with the centroid are required. For this purpose the points on the border derived from the maximum and minimum distance from the region's centroid to the region's border are very appealing to use since the measures associated with them were used for recognising the topmost object.

Unfortunately, determining the normal to the surface using only a vertex (3 non-collinear points) is very sensitive to the errors in depth estimation. In addition it is worth noting that the highest error rate in depth estimation is around the objects borders. To compensate for this issue, four vertices v_1, v_2, v_3, v_4 adjacent to the centroid are chosen and the normal vector is obtained by averaging the vectors obtained for each vertex (see Figure 4.17).

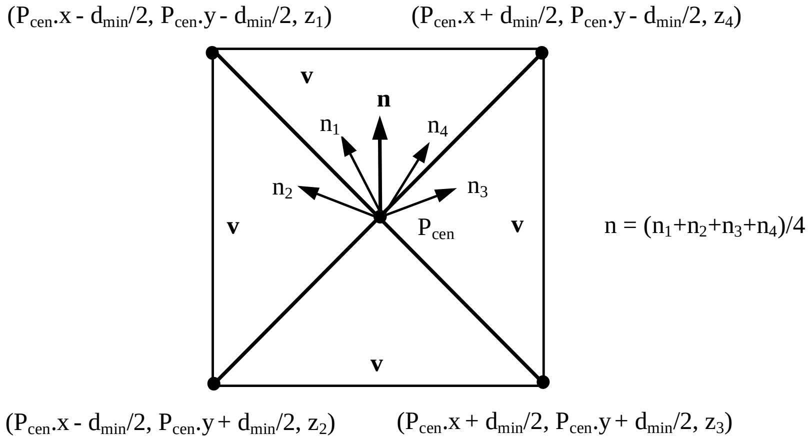
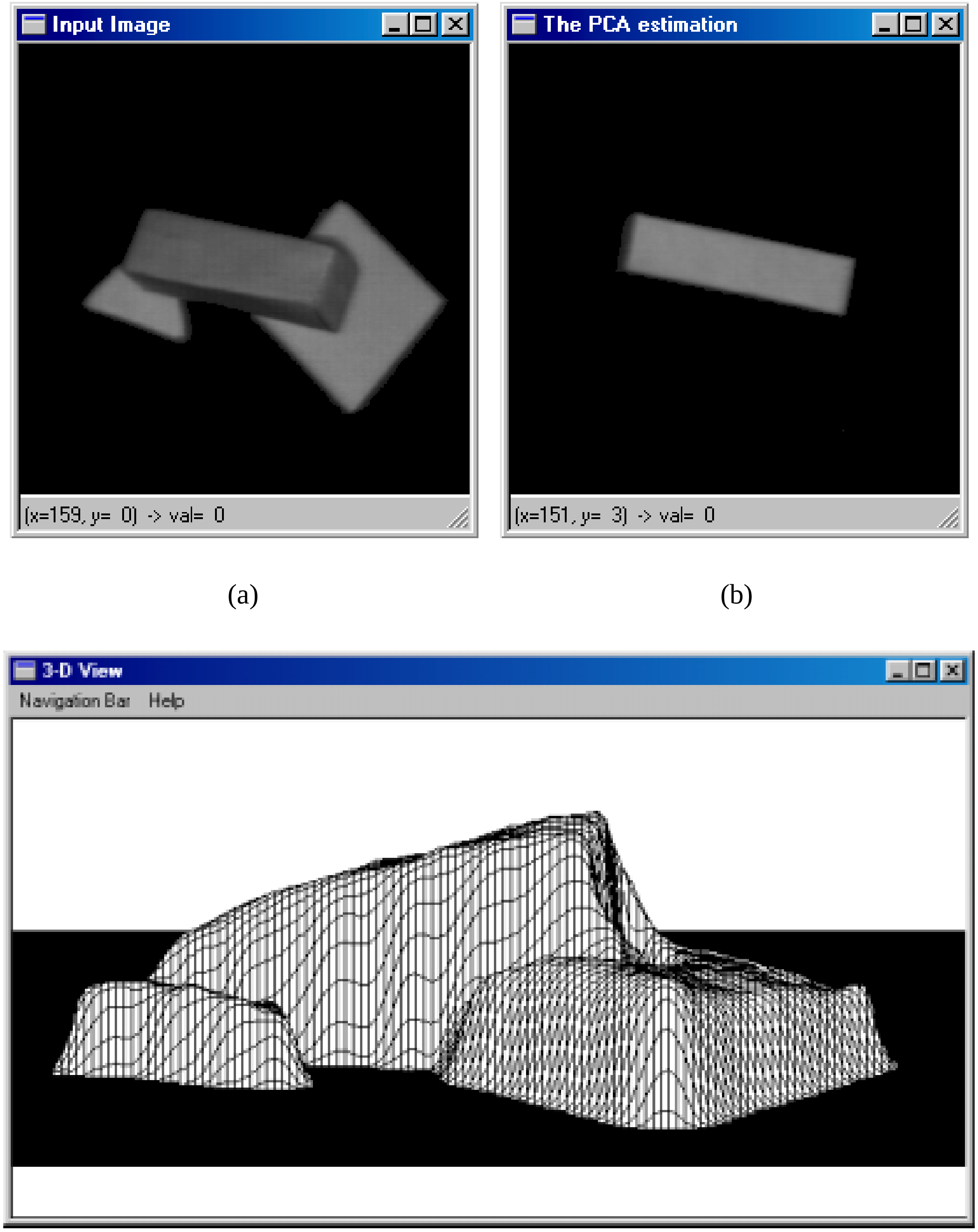

$(P_{cen}.x - d_{min}/2, P_{cen}.y - d_{min}/2, z_1)$
$(P_{cen}.x + d_{min}/2, P_{cen}.y - d_{min}/2, z_4)$
v
n
n_1
n_4
n_2
n_3
v
P_{cen}
v
$n = (n_1+n_2+n_3+n_4)/4$
v
$(P_{cen}.x - d_{min}/2, P_{cen}.y + d_{min}/2, z_2)$
$(P_{cen}.x + d_{min}/2, P_{cen}.y + d_{min}/2, z_3)$

Figure 4.17. Computing the normal to a surface from 4 adjacent vertices.

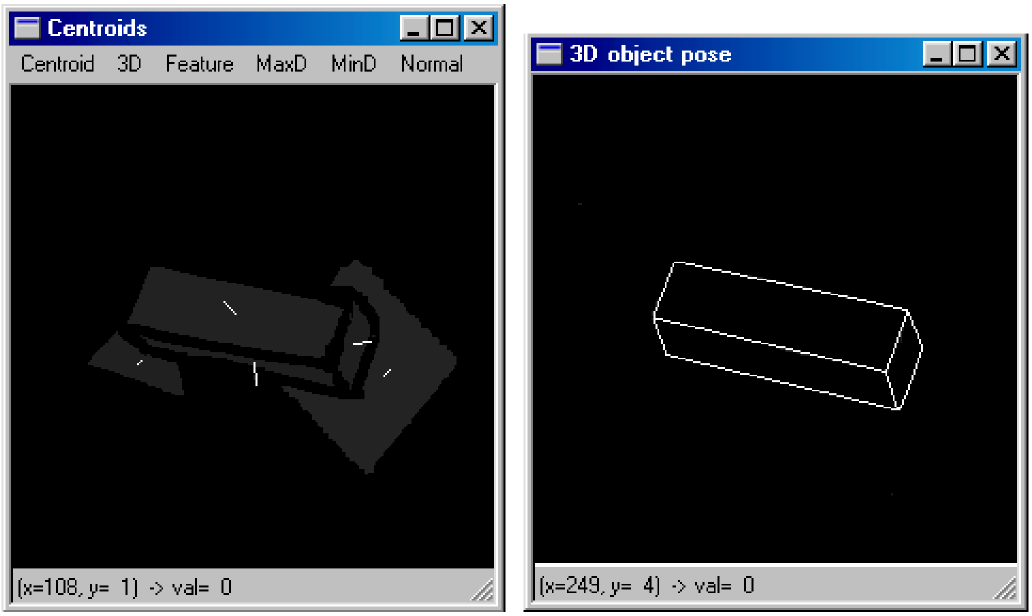
The vertices depicted in Figure 4.17 were chosen symmetrically relative to the centroid using only the minimum distance from the surface centroid to its border in order to avoid the situation when the selected points may fall outside the surface. Results of the developed pose estimation algorithm are depicted in Figure 4.18.



(a)

(b)

(c)



(x=108, y= 1) -> val= 0

(x=249, y= 4) -> val= 0

(d) (e)

Figure 4.18. The 3 DOF pose estimation. (a) The input image. (b) PCA pose estimation. (c) Depth estimation. (d) The normals computed from the depth map illustrated in image (c). (e) 3 DOF pose estimation for the topmost object. The model displayed in image (e) is synthetically generated using the parameters returned by the pose estimation algorithm (PCA and range data analysis).

4.4 Analysis of recognition in cluttered and occluded scenes

The recognition system was tested in the presence of clutter and mild occlusions. In order to test the recognition system, a database containing 5 objects was created. The objects contained in the database are shown in Figure 4.19.

The cluttered scenes are created by arranging the objects contained in the database in various ways. In this regard, the system was initially tested on a simple scene illustrated in Figure 4.20-a in which several objects are situated in convenient positions. Then, it was tested on a complex scene depicted in Figure 4.21-a where the objects are either slanted or occluded.

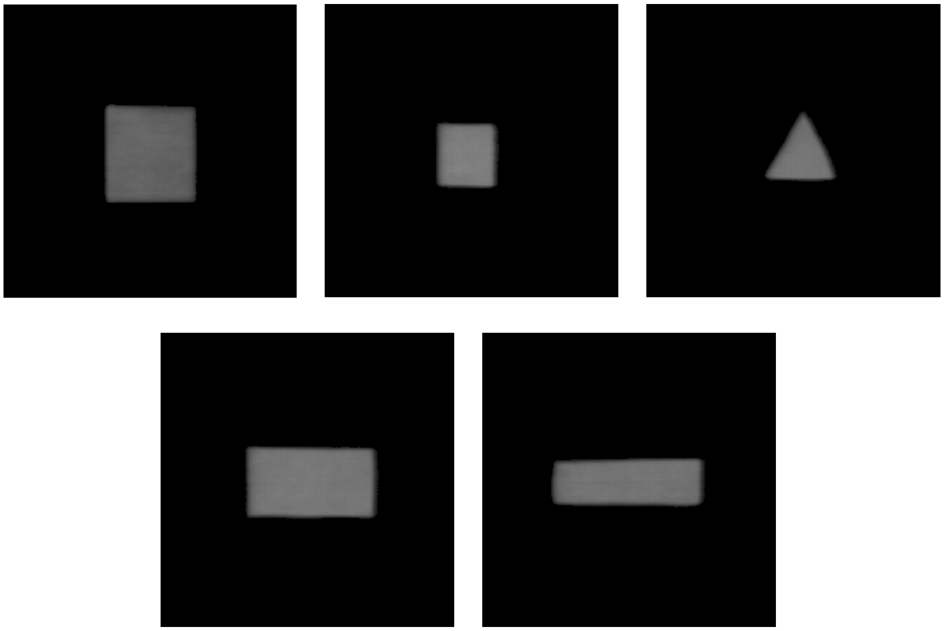
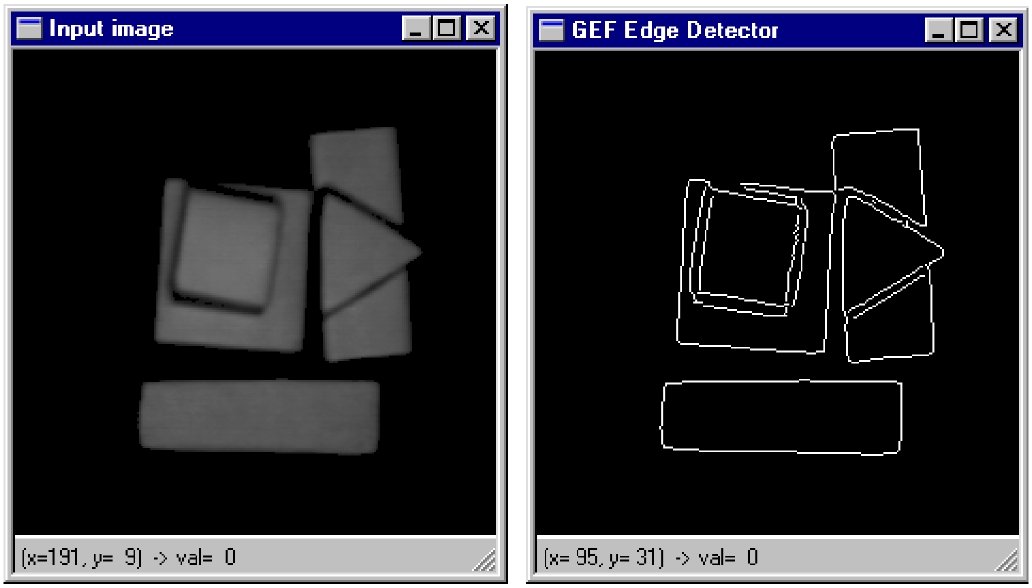


Figure 4.19. The object set utilised in experimentation.

The next test was performed on a difficult scene where all objects are occluded (see Figure 4.22).

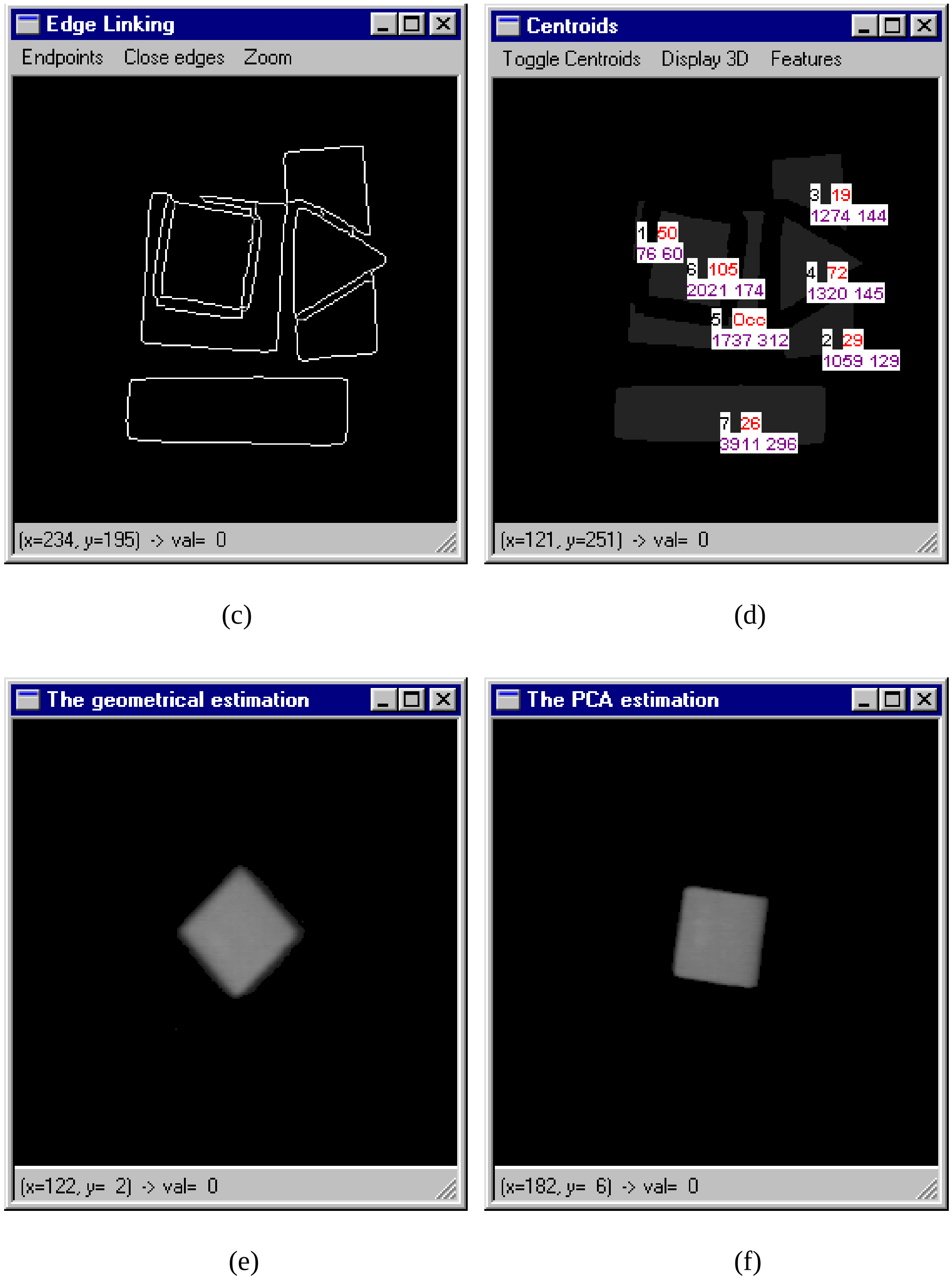


```
(x=191, y= 9) -> val= 0
```

```
(x= 95, y= 31) -> val= 0
```

(a)

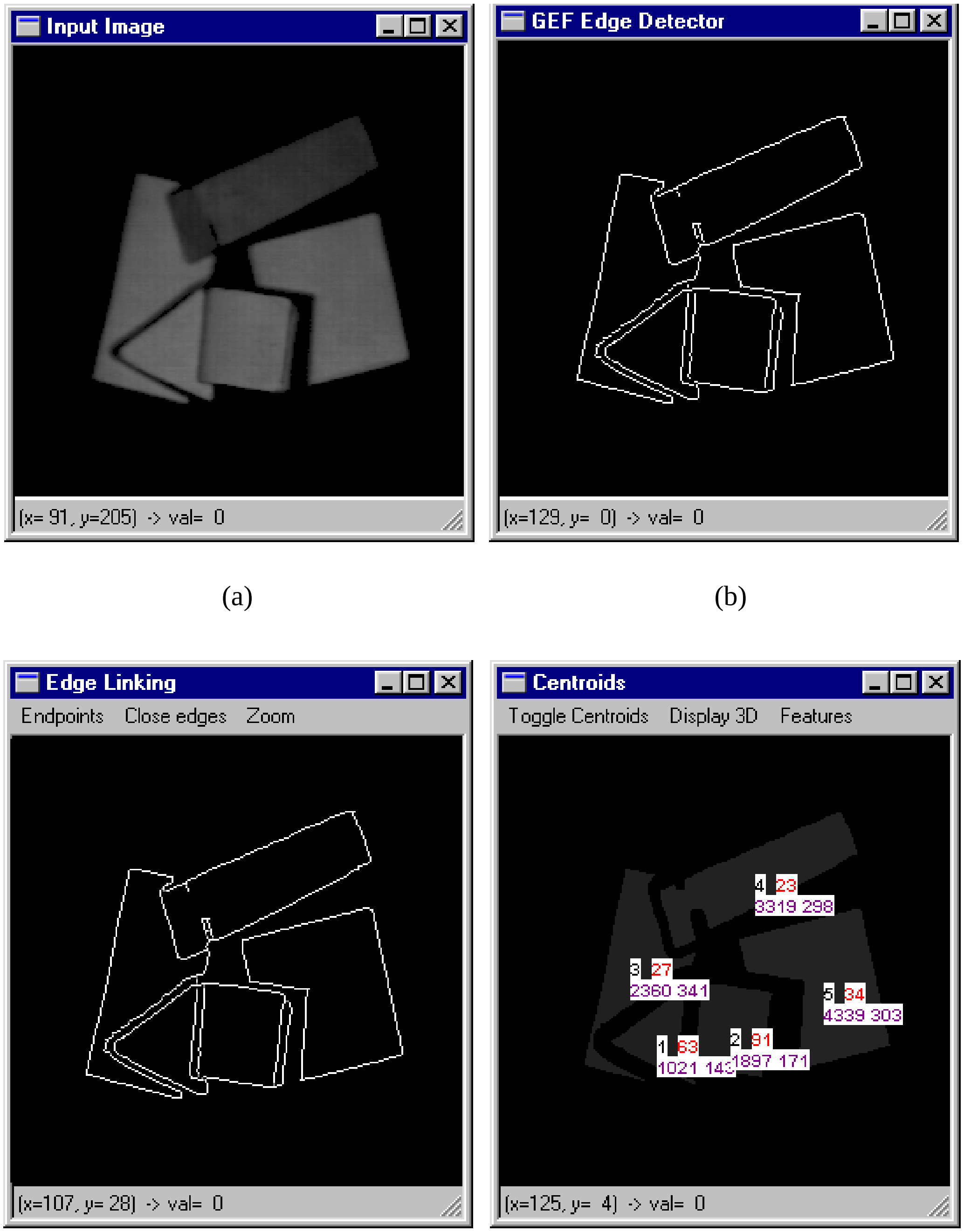
(b)



(c) (d)

(e) (f)

Figure 4.20. Results set 1. (a) The original image. (b) The image after the application of the GEF edge operator. (c) The image after edge linking operation. (d) The resulting image data (the first figure is the rank of the region with respect to the area, the second is the region's maximum elevation, and the last two represent the dominant features). (e) The estimation returned by the algorithm using geometrical constraints. (f) The estimation returned by the algorithm using eigenimage analysis.

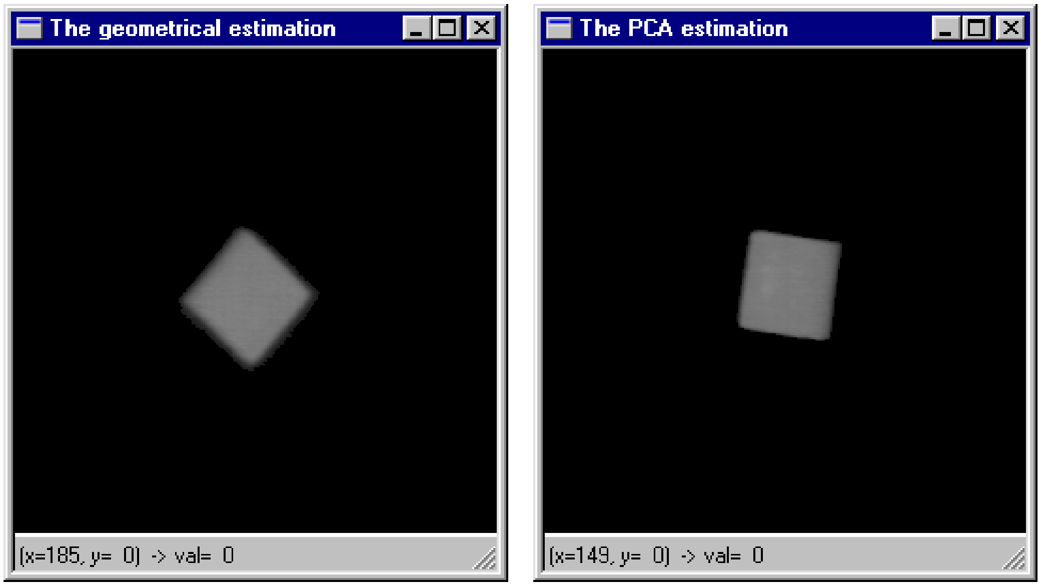

Input Image
(x= 91, y=205) -> val= 0
GEF Edge Detector
(x=129, y= 0) -> val= 0
Edge Linking
Endpoints
Close edges
Zoom
(x=107, y= 28) -> val= 0
Centroids
Toggle Centroids
Display 3D
Features
4 23
3319 298
3 27
2360 341
5 34
4339 303
1 63
1021 14
2 91
1897 171
(x=125, y= 4) -> val= 0

(a)

(b)

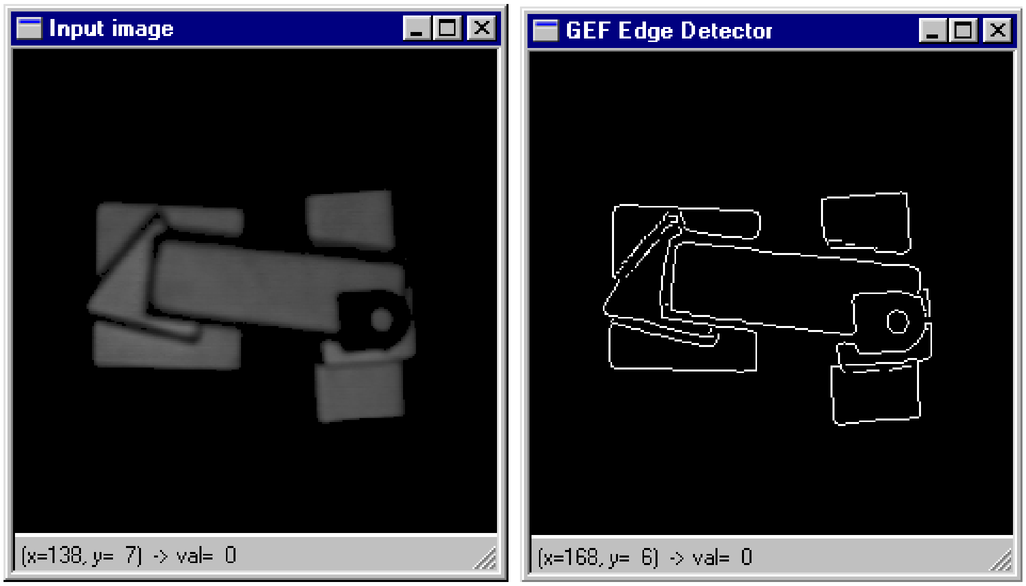
(c)

(d)

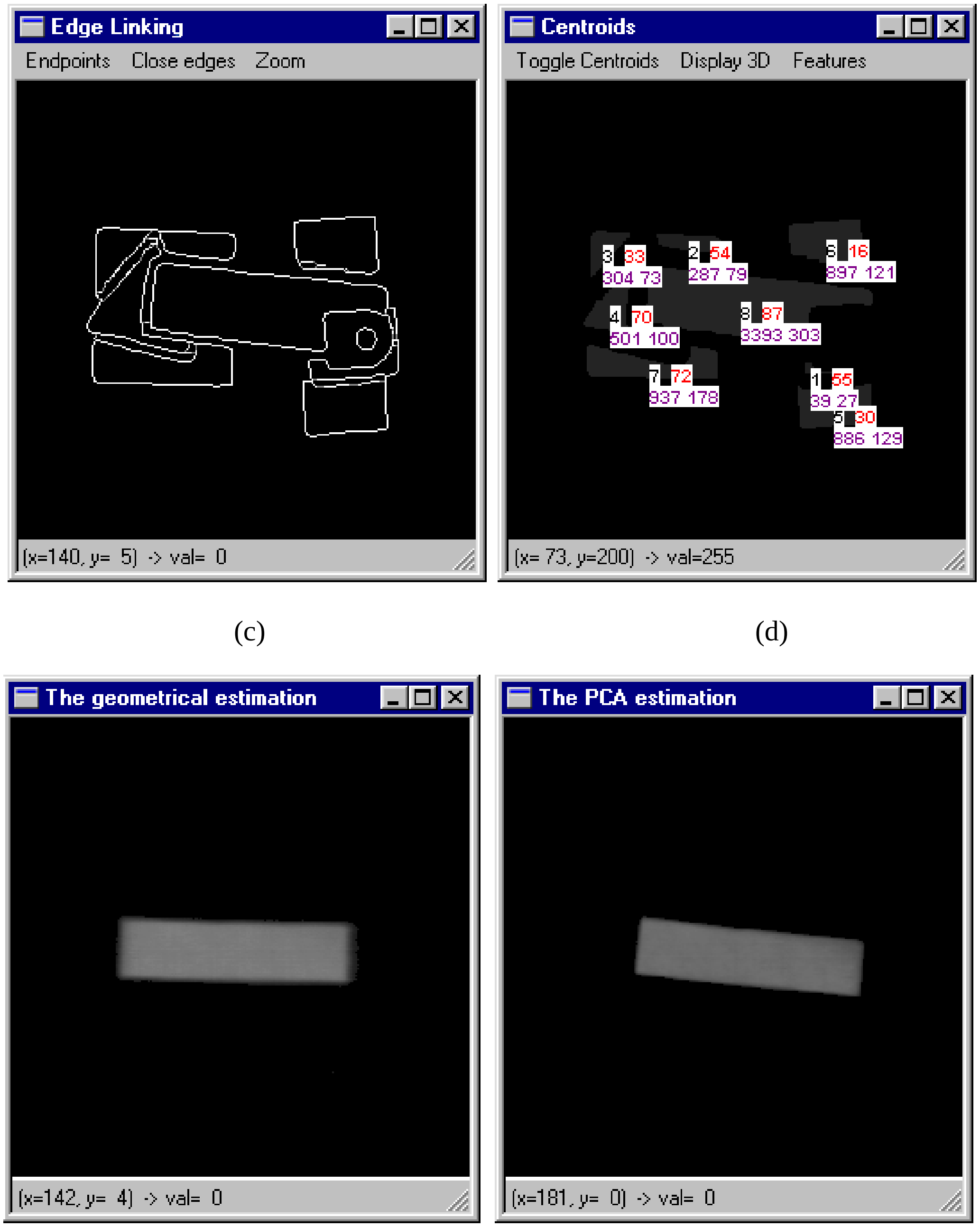


(e) (f)

Figure 4.21. Results set 2. (a) The original image. (b) The image after the application of the GEF edge operator. (c) The image after edge linking operation. (d) The resulting image data (the first figure is the rank of the region with respect to the area, the second is the region's maximum elevation, and the last two represent the dominant features). (e) The estimation returned by the algorithm using geometrical constraints. (f) The estimation returned by the algorithm using eigenimage analysis.



(a) (b)



(x=140, y= 5) -> val= 0

(x= 73, y=200) -> val=255

(c)

(d)

(x=142, y= 4) -> val= 0

(x=181, y= 0) -> val= 0

(e)

(f)

Figure 4.22. Results set 3. (a) The original image. (b) The image after the application of the GEF edge operator. (c) The image after edge linking operation. (d) The results (the first figure is the rank of the region with respect to the area, the second is the region's maximum elevation, and the last two represent the dominant features). (e) The estimation returned by the algorithm using geometrical constraints. (f) The estimation returned by the algorithm using eigenimage analysis.

A key problem is the dimension of the eigenspace. As noted earlier, the eigenspace dimension is limited to P, where P is the number of positions contained in the image set which is specific for each object. The experimental results indicated that this algorithm failed to return a precise estimation if the dimension is less than 8 (see Figure 4.19).

The eigenspace dimension was increased to 24 incrementally but the error rate was not affected visibly after 16, when the position is estimated within the algorithm error. For the present implementation the dimension used was 24, this generates the most precise results.

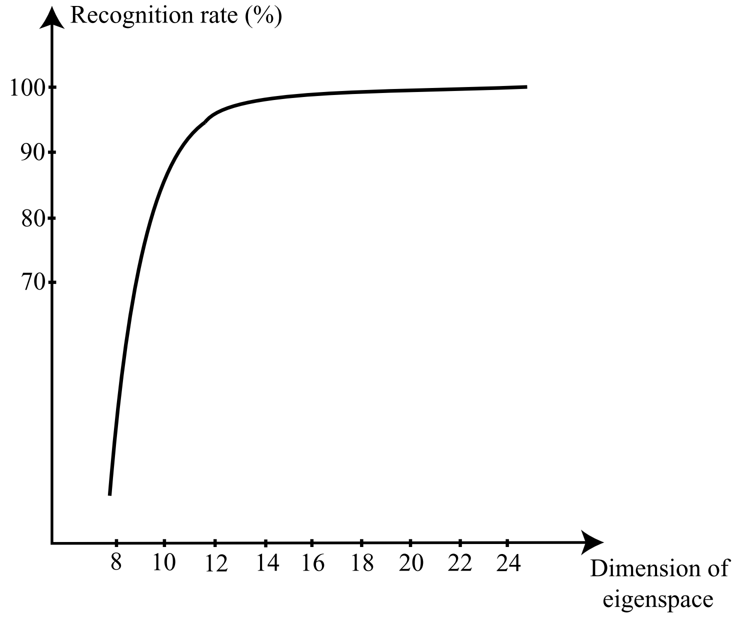


Figure 4.19. Estimation rate as a function of the dimension of eigenspace.

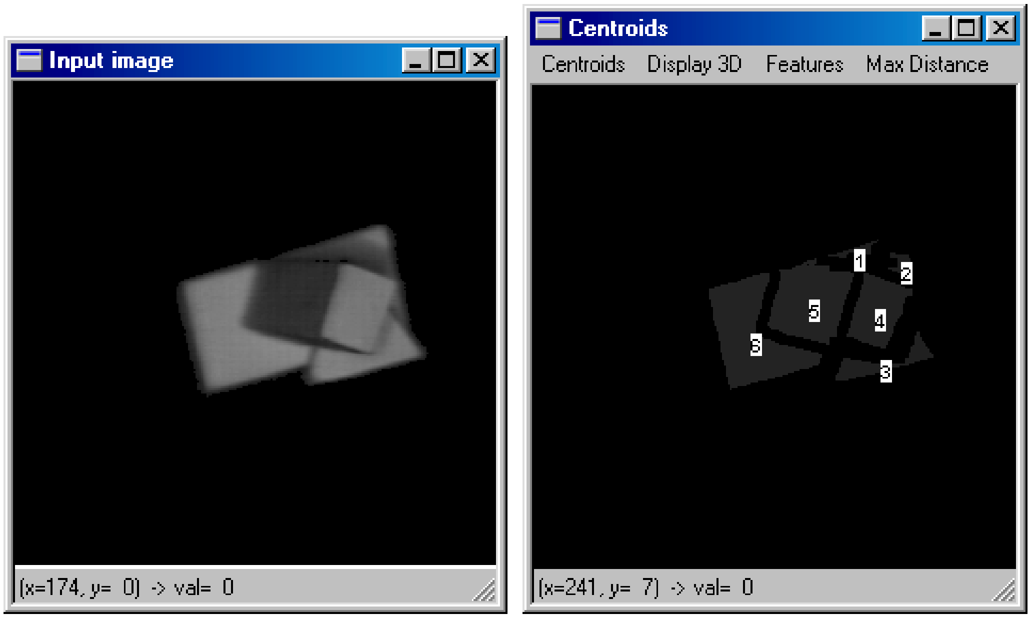
The algorithm was designed in order to minimise the *false-positive* recognition state when the object does not belong to the scene and the recognition algorithm concludes that the object exists. This situation appears when the segmentation algorithm does not decompose the input image into disjointed meaningful regions or the 3-D information is not precise enough in determining the object placed on the top of the object pile. These issues are minimised by using the *global search operation* which was described in Section 4.3.2.3. When the scene is affected only by clutter, the algorithm correctly recognises the object placed on the top of the pile. When the scene is affected by clutter and occlusion the object is correctly recognised only if the occluded region is smaller than 15% of the object's total area.

4.5 Discussion

A recognition algorithm suitable for bin picking has to tackle some important issues such as clutter and occlusion, problems that were discussed and analysed in Section 4.2. Reviewing the existing systems, the conclusion is that all of them have merits and limitations. For example, the geon-based recognition scheme is conceptually appealing since the primitive representation can be approached in a hierarchical manner. In spite of this, the practical implementation is hindered by a complex primitive extraction procedure. This approach is extremely well suited for the recognition of textureless objects with distinct faces. If the objects of interest are textured this technique may not be appropriate and a possible solution relies on using appearance-based recognition approaches. To be efficient when dealing with occlusion, the appearance has to be divided and the resulting windows are analysed to match a model object. This scheme is particularly appropriate to recognise flat objects with different textural characteristics. Because the bin picking application is specifically geared to dealing with the 3-D world, using 3-D information in the recognition process appears to be the natural approach. The 3-D methods are very successful if the relative depth between the objects in the scene is significant. Also, it is important to note that the quality of the range sensor plays a crucial role.

Because the objects of interest are textureless and small the recognition scheme outlined in this thesis is conceptually related to the approach detailed in the paper by Dickinson *et al* (1992). As mentioned in Section 4.2, their implementation is based on graph matching when the applicability of this approach is restricted to objects with a relatively large number of faces with a simple geometry. In contrast with this implementation, the proposed formulation is a region-based approach where the recognition process consists of matching the features derived from the region in question. It also should be noted that this approach can handle objects with arbitrary shapes and furthermore the implementation is simple, fast and reliable. Because the present approach addresses a bin picking implementation it is very important to remember that the scene changes each time an object is picked, thus only the object placed on the top of the object pile is of interest, an object which *is rarely* occluded.

Nevertheless, there are situations when all the objects are heavily occluded or they are positioned in such way that their appearance is significantly disturbed. Figure 4.20 shows an example where the resulting regions are not able to fulfil the matching criteria. In such cases the resulting regions after segmentation are too distorted and they will be either rejected or misclassified. Thus, a major aim was to minimise the occurrence of the false-positive recognition state when the object does not belong to the scene and the recognition algorithm concludes that the object exists. In this regard, a large number of features which are able to match the most relevant properties of the regions that describe the object were employed. Also in order to minimise the misclassification, the matching algorithm uses very strict selection criteria and this formulation is completed with a general search operation.



(x=174, y= 0) -> val= 0

(x=241, y= 7) -> val= 0

(a) (b)

Figure 4.20. An example when the recognition algorithm fails to identify the objects contained in the scene. (a) The input image. (b) The resulting regions after segmentation.

The second stage of the algorithm deals with pose estimation and the current approach is based on eigenimage analysis augmented with an analysis in the range data. The pose estimation is traditionally achieved by analysing the transformation caused due to viewpoint changes on the visible surfaces that belong to the object. This approach proved to be very efficient if the number of surfaces resulting after segmentation is significant. Because this research deals with objects with a small number of faces, the position of the object in question is estimated using an appearance-based approach applied to the region resulting after the application of the recognition algorithm. In conjunction with the appearance-based approach the proposed algorithm includes an analysis in the range data in order to obtain 3 DOF pose estimation. The experimental data demonstrates the effectiveness of the proposed strategy when dealing with polyhedral objects.

Chapter 5 - System implementation

This chapter describes the overall system. A detailed description of the block diagram is provided in order to clarify many aspects of the system implementation. This section begins with an introductory presentation of a bin picking system and is followed by the description of the current implementation.

5.1 The block diagram for a bin picking system

The block diagram illustrated in Figure 5.1 represents the outline of a bin picking system, where components have been classified under seven main blocks.

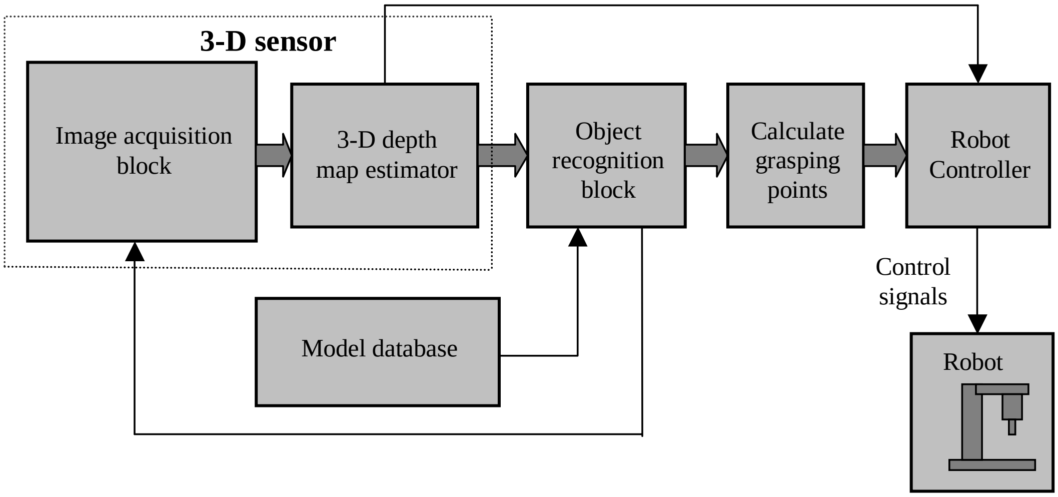


Figure 5.1. The block diagram of a bin picking system.

Starting on the left of Figure 5.1, the *3-D sensor* is the point in the diagram where data enters. As noted in Chapter 2, the 3-D sensor can be divided into two different components. The first component i.e. the "*Image acquisition block*" consists of the image acquisition equipment (sensing elements, lens and frame grabbers) and depending on the range sensing strategy employed, two or more images are captured in order to recover the 3-D information. These captured images are passed to the "*3-D depth map estimator*" block. This block is a software or hardware component which computes the depth map using the information contained in the captured images according to the range sensing strategy in question. Also, for some sensors a calibration procedure may be required and this is illustrated in Figure 5.1 by the loop between the “*3-D depth map estimator*” block and the “*Robot controller*” block. The output of the “*3-D depth map estimator*” block is the depth map of the scene and this information is passed to the next block.

The “*Object recognition*” block is the key component of an adaptive robotic application and deals with the recognition of similar or different objects contained in the scene. The recognition process consists of matching the input object with a model stored in the “*Model database*”.

The “*Calculate grasping points*” block computes the co-ordinates (*x*, *y*, *z*) of the graspable object. If the object is correctly matched, then it is graspable and the gripper will pick it up and perform manipulation operations.

The “*Robot controller*” block performs the communication between the host computer and the robot. This component will report all the errors that occur when the robot is running.

5.2 Current implementation

5.2.1 The image acquisition block

Intuitively, the image acquisition block contains the optical and sensing equipment (lens, beam splitter, CCD sensors and frame grabbers).

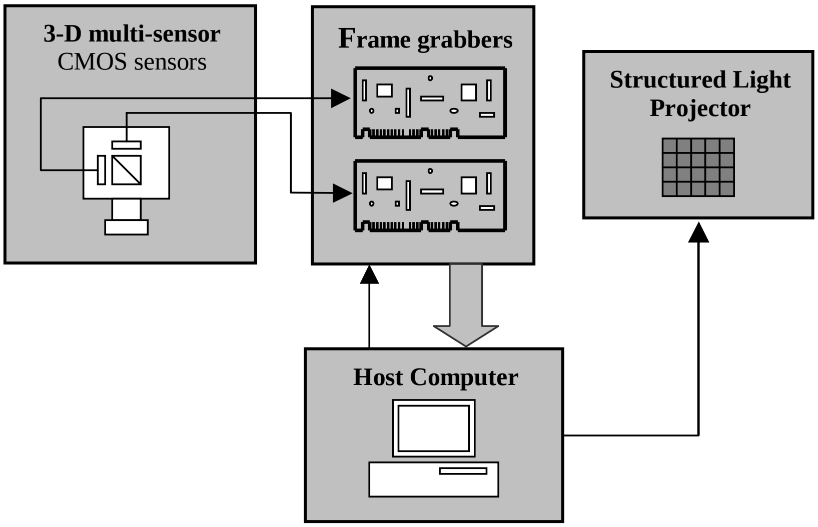


Figure 5.2. The image acquisition block diagram.

Along with the optical and sensing equipment, it is widely acknowledged that the illumination conditions play a crucial role for any vision system. This observation is further strengthened by this particular application where the active illumination was identified as the key issue in the implementation of a range sensor based on a defocusing technique. The diagram of the image acquisition block is shown in Figure 5.2. Initially, in the implementation of this sensor a single frame grabber was utilised, a situation when the near and far focused images were sequentially digitised. There is no doubt that this solution is restricted by a range of problems such as the differences between the near and far focused images when dealing with dynamic scenes. In addition, the time required to capture a pair of images sequentially is too long and hinders the implementation of a real-time range sensor. Due to the aforementioned problems, in the implementation of the current range sensor two frame grabbers were employed. Also, as mentioned in Section 2.2.4 a problem of interest consists of choosing the optimal pattern for structured light. Because a special pattern is difficult to manufacture and the achievements in terms of precision are not significant, for this sensor a simple striped grid used in Moiré contour detection was employed.

5.2.2 The 3-D depth map estimator

The "3-D depth map estimator" is a complex software block and performs the 3-D estimation using the information contained in two images captured with different focal settings.

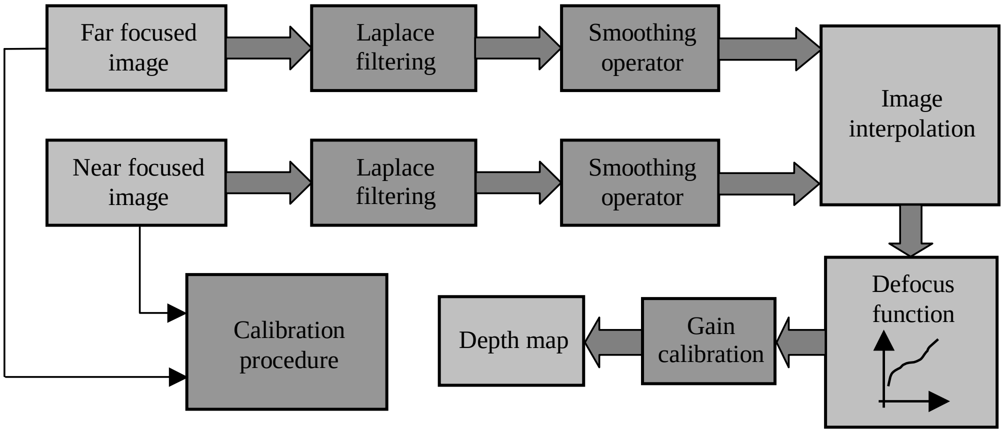


Figure 5.3. The 3-D depth estimator block diagram.

The principal operations required to compute the depth map are outlined in Figure 5.3. The near and far focused images are stored in the computer's memory after the image acquisition stage has been completed. The depth is estimated by isolating the blurring effect. To do this, it is necessary to extract the high frequency information derived from the scene by filtering the near and far focused images with a Laplacian operator. As mentioned in Section 2.2.5, the Laplacian operator enhances the high frequency noise and to compensate for this issue a smoothing Gaussian operator is applied. The image interpolation was discussed in Section 2.2.7 and its role consists of enhancing the quality of the depth map by interpolating the dark regions caused by the illumination pattern. The information resulted after image interpolation is used to determine the depth map. The gain correction compensates for the errors caused by the imperfection of the optical and sensing equipment. The calibration procedure was outlined in Section 2.2.9 and its goal is to align the sensing elements in order to minimise the mismatch between their spatial positions.

5.2.3 The object recognition block

The object recognition block indicates whether the object under investigation is contained in the database or not. Also, another task deals with estimating the relative position in space for the recognised object in order to provide the information required in the manipulation stage. The complete diagram of the current implementation is illustrated in Figure 5.4.

Depending on the strategy involved, the recognition process may be preceded by segmentation. If the recognition scheme deals with local invariants the segmentation process is not necessary. Alternatively, if the recognition consists of analysing high-level primitives, their extraction requires scene segmentation. Because textureless objects are the topic of this research, the segmentation process determines the meaningful regions by analysing the edge information provided by the input image.

As illustrated in Figure 5.4, the next operation attempts to select the object placed on the top of the object pile. This operation was described in detail in Section 4.3.1 and involves a framework that deals with a variable number of regions which fulfil some 3-D criteria. The aim of this framework is to select only the best placed regions that will be used in the recognition process. From these regions, the one that gives the best approximation with respect to the matching criteria is considered to be on the top of the pile. If the threshold conditions are upheld the region matches an object model from the database.

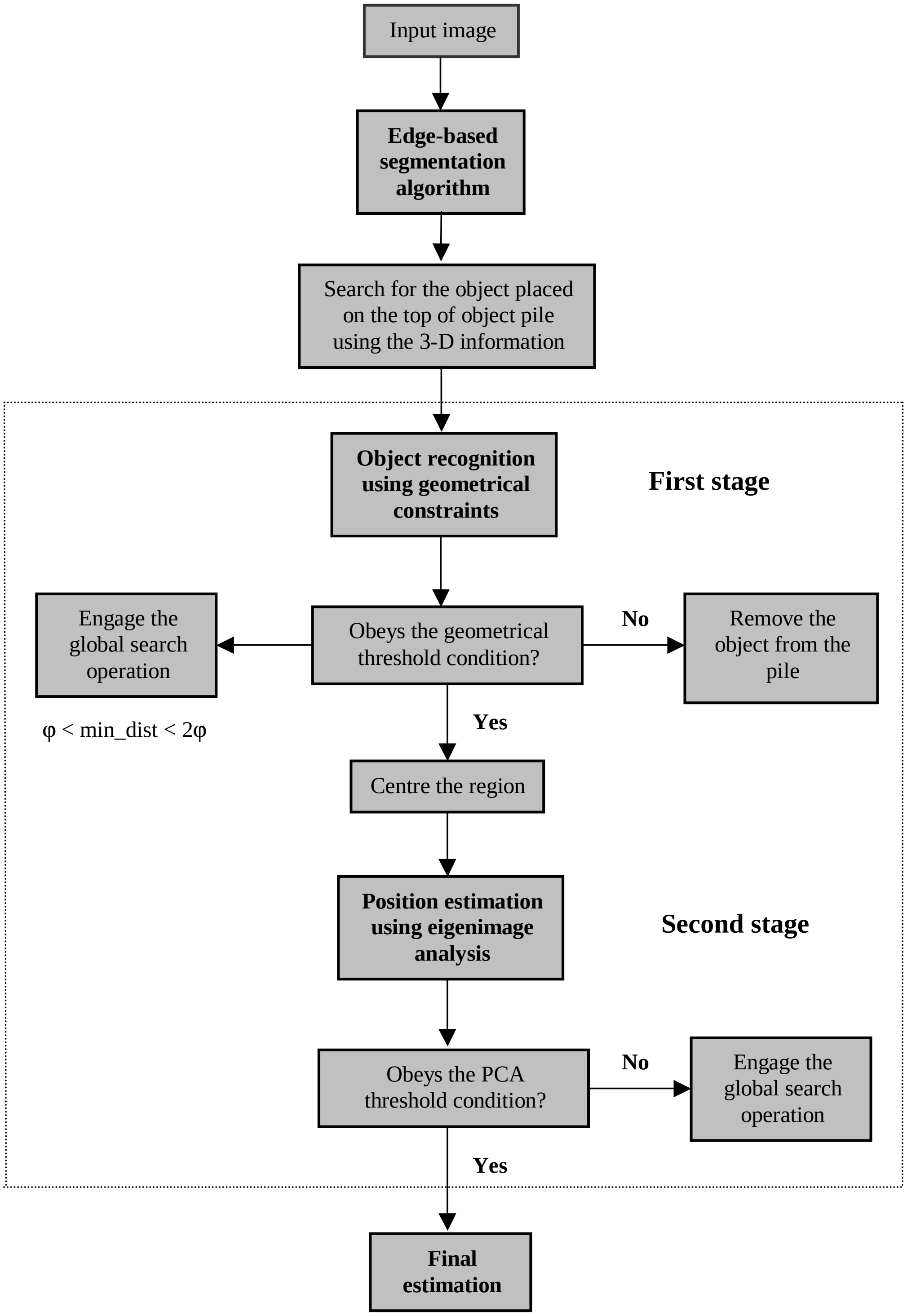

Input image
Edge-based segmentation algorithm
Search for the object placed on the top of object pile using the 3-D information
Object recognition using geometrical constraints
First stage
Engage the global search operation
Obeys the geometrical threshold condition?
No
Remove the object from the pile
φ < min_dist < 2φ
Yes
Centre the region
Position estimation using eigenimage analysis
Second stage
Obeys the PCA threshold condition?
No
Engage the global search operation
Yes
Final estimation

Figure 5.4. The outline of the object recognition and pose estimation algorithm.

The algorithm invokes a general search operation if the matching result exceeds the threshold but is smaller than a predefined value. If is not the case, the object is removed or the scene is rearranged. The complete list of operations required in the recognition stage is outlined in Figure 5.5.

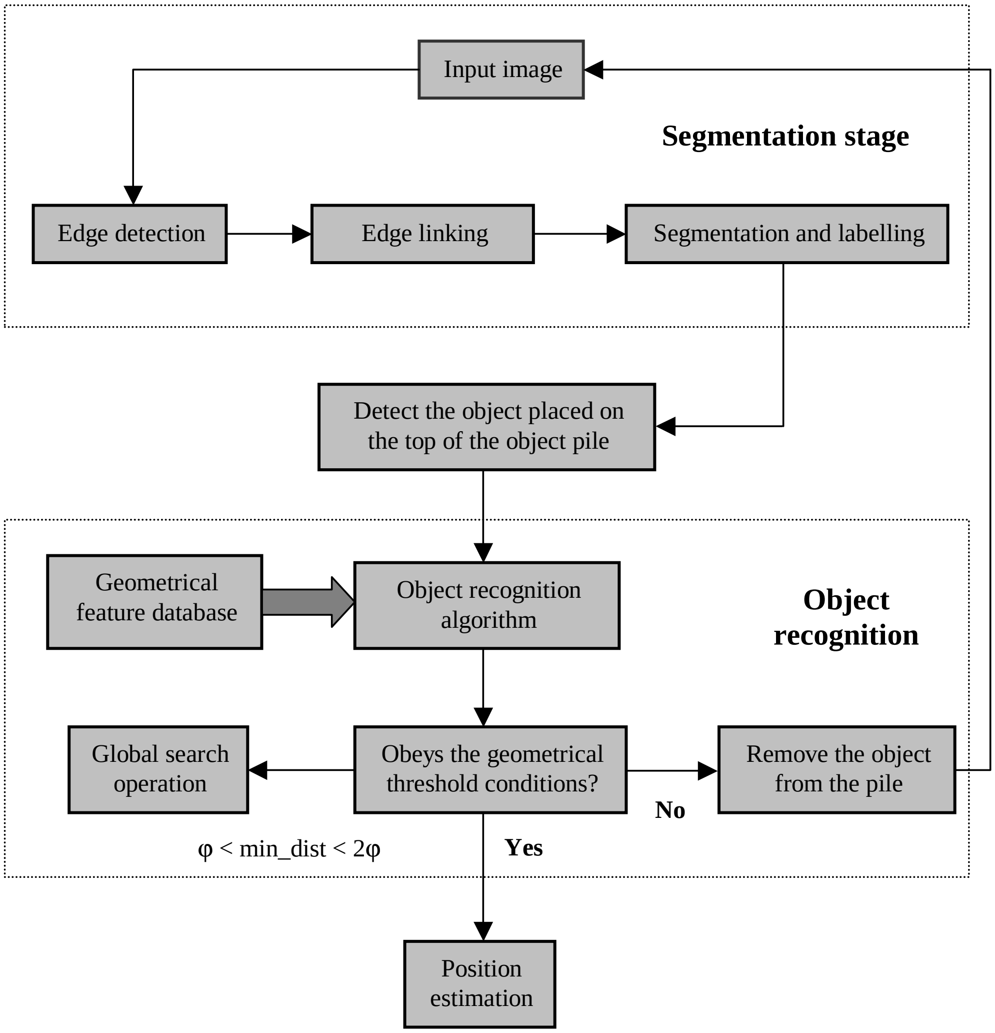
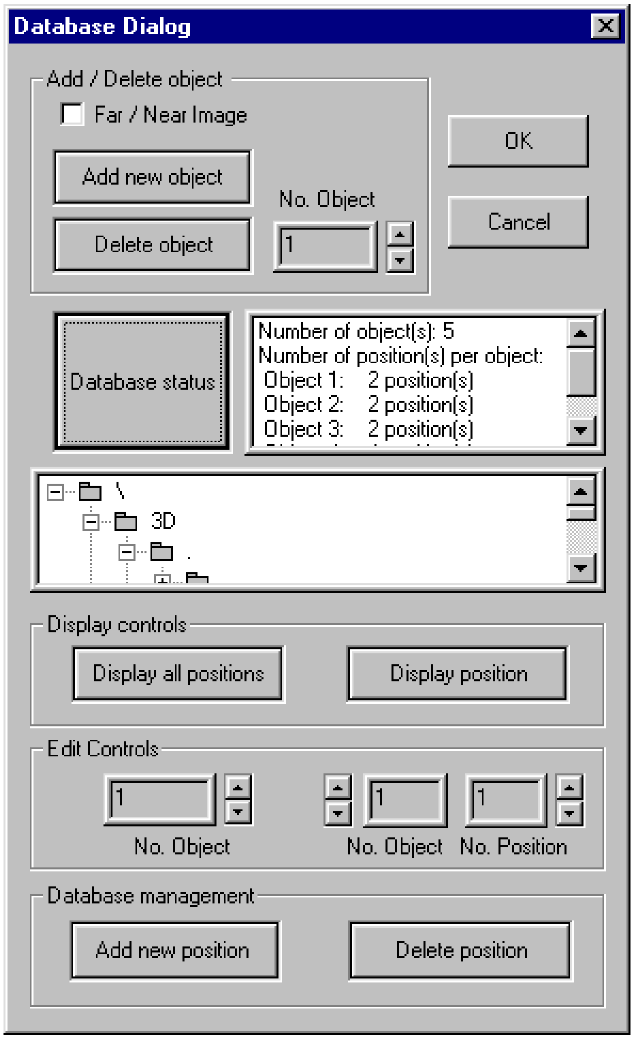

Input image
Segmentation stage
Edge detection
Edge linking
Segmentation and labelling
Detect the object placed on the top of the object pile
Geometrical feature database
Object recognition algorithm
Object recognition
Global search operation
Obeys the geometrical threshold conditions?
Remove the object from the pile
No
φ < min_dist < 2φ
Yes
Position estimation

Figure 5.5. The object recognition stage.

As can be seen in Figure 5.4, the recognition process is addressed in the first stage while the second stage deals with pose estimation. This scheme provides the ability to organise the model database required by the recognition algorithm independently. This independence is very convenient because it provides a great deal of flexibility where the user can easily add or remove any irrelevant views without affecting the database required by the pose estimation algorithm. In this regard, an intuitive graphical interface is provided; the database management interface is illustrated in Figure 5.6.



```
Number of object(s): 5
Number of position(s) per object:
 Object 1:    2 position(s)
 Object 2:    2 position(s)
 Object 3:    2 position(s)
```

Figure 5.6. The graphical environment for the geometrical database.

The output of the object recognition algorithm represents the input for the next stage, which addresses the pose estimation using eigenimage analysis (PCA). Because this method is sensitive to the position of the region within the image, therefore the first operation involves centering the region. The resulting region is projected onto the object's eigenspace and the output is a multi-dimensional point. The matching algorithm computes the Euclidean distance between this vector and the vectors from the *model database*. The minimum distance will represent the best estimation.

The operations required by the pose estimation algorithm are shown in Figure 5.7.

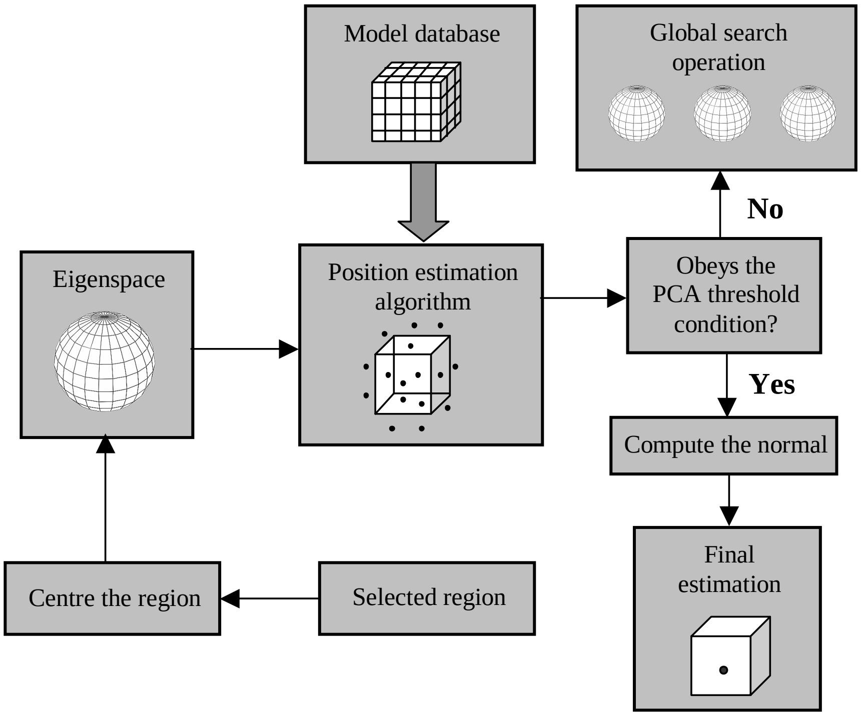


Figure 5.7. The position estimation stage.

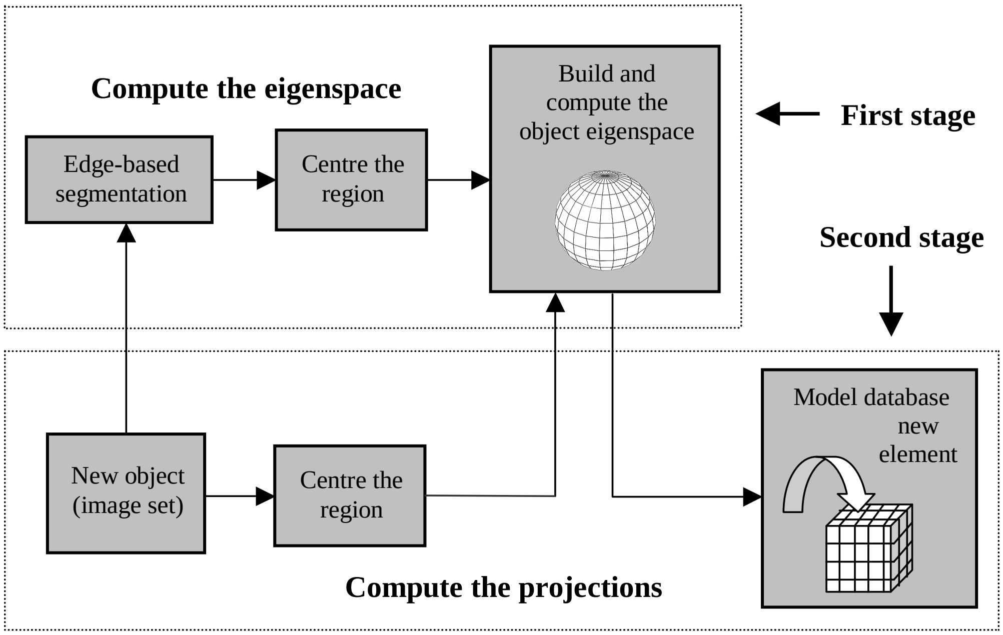
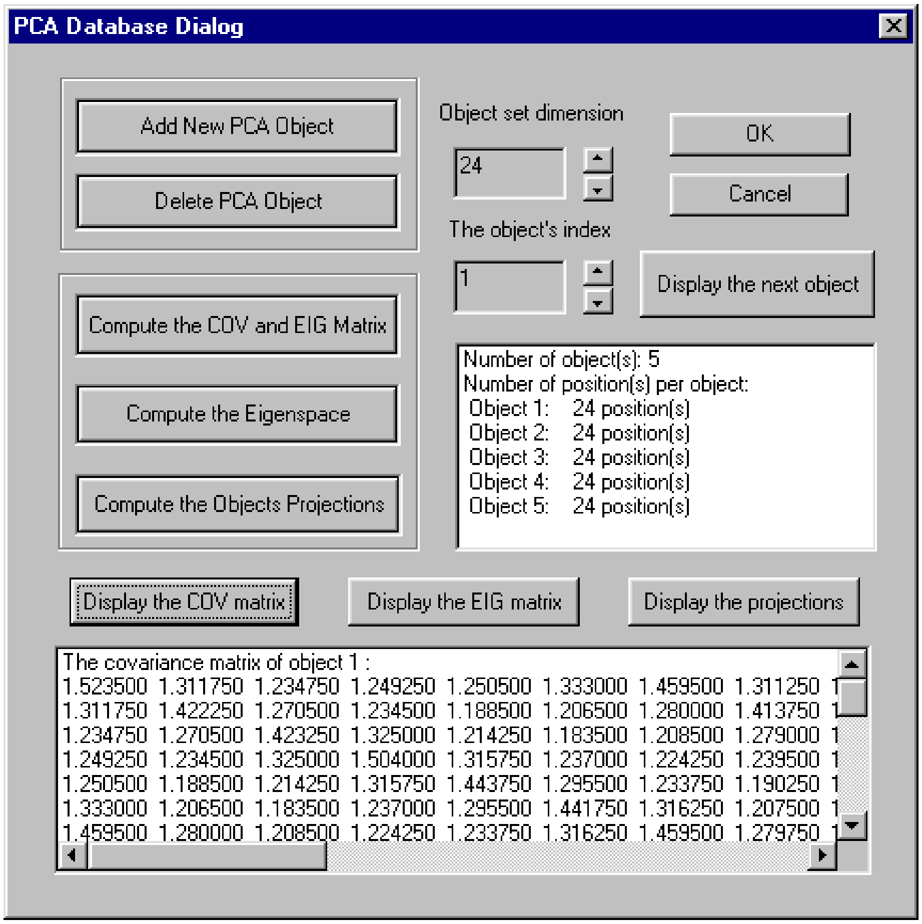


Figure 5.8. The training stage procedure.

Another key stage is the generation of the database, an operation that is implemented by the two-stage training procedure illustrated in Figure 5.8. The first stage deals with building and computing the object eigenspace and consists of following operations: determine the region for every image contained in the image set, compute the covariance matrix of the image set, compute the eigenvalues and the corresponding eigenvectors and select the eigenvectors that correspond to the biggest eigenvalues. The second stage computes the projections that are used in the estimation stage by projecting the image set on the object's eigenspace. This procedure is directly related to the number of images contained in the image set. As mentioned earlier, the PCA database is independent of the geometrical database.



Object set dimension: 24

The object's index: 1

```
Number of object(s): 5
Number of position(s) per object:
 Object 1:    24 position(s)
 Object 2:    24 position(s)
 Object 3:    24 position(s)
 Object 4:    24 position(s)
 Object 5:    24 position(s)
```

```
The covariance matrix of object 1 :
1.523500 1.311750 1.234750 1.249250 1.250500 1.333000 1.459500 1.311250 1
1.311750 1.422250 1.270500 1.234500 1.188500 1.206500 1.280000 1.413750 1
1.234750 1.270500 1.423250 1.325000 1.214250 1.183500 1.208500 1.279000 1
1.249250 1.234500 1.325000 1.504000 1.315750 1.237000 1.224250 1.239500 1
1.250500 1.188500 1.214250 1.315750 1.443750 1.295500 1.233750 1.190250 1
1.333000 1.206500 1.183500 1.237000 1.295500 1.441750 1.316250 1.207500 1
1.459500 1.280000 1.208500 1.224250 1.233750 1.316250 1.459500 1.279750 1
```

Figure 5.9. The graphical environment for the PCA database.

While the geometrical database is very flexible by allowing the user to add and remove irrelevant spatial orientation, the PCA database is more rigid because the entire image set is required to be processed in order to compute the object eigenspace. The user has two options to add or delete the object's eigenspace. The window that provides the management of the PCA database is depicted in Figure 5.9.

5.3 The system environment

The system was implemented using Visual C++ 5.0. The use of Visual C++ is very convenient because it supports a modular design, every module having its own graphical interface.

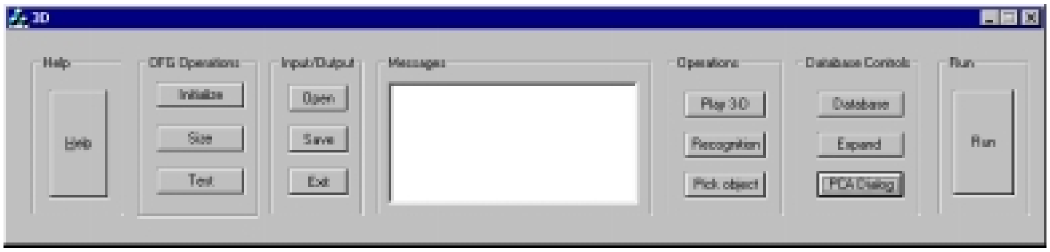
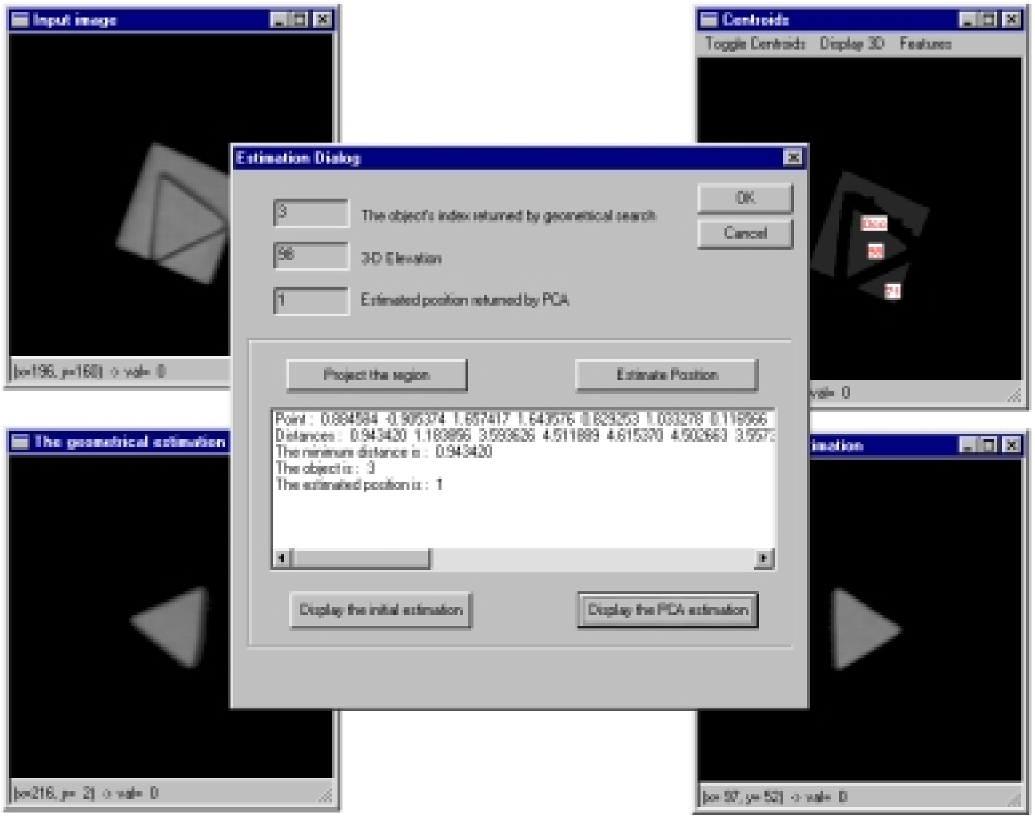


Figure 5.10. The application window.



3 The object's index returned by geometrical search
98 3-D Elevation
1 Estimated position returned by PCA

```
Point :  0.884584 -0.905374  1.657417  1.643576  0.829253  1.033278  0.116966
Distances :  0.943420  1.183896  3.593626  4.511889  4.615370  4.502663  3.957
The minimum distance is :  0.943420
The object is :  3
The estimated position is :  1
```

[x=196, y=168] -> val= 0

[x=216, y= 2] -> val= 0

[x= 97, y= 52] -> val= 0

Figure 5.11. The developed system environment.

The application window is illustrated in Figure 5.10 while some windows associated with a working sequence are shown in Figure 5.11.

5.4 Discussion

This section gives a detailed presentation of the system outline. In order to emphasise the implementation aspects, a block diagram was provided for each component which describes the main operations and its role in the overall system. One of the most important requirements for a bin picking system is real-time operation. Therefore, to accomplish this goal all the operations have to be computationally efficient. In this regard, the range sensor based on depth from defocus proved to be an appropriate solution due to its merits such as robustness, accuracy and speed. Prior to the object recognition stage, the segmentation algorithm must be applied. The current edge-based segmentation technique is particularly well suited for dealing with textureless objects which are the subject of this implementation.

There is no doubt that the recognition and pose estimation algorithm represents the most important component of the bin picking system. The two-stage adopted approach initially recognises the object using global geometrical constraints computed for the best placed region within the scene, while the position is estimated at a later stage using an appearance-based approach. An operation cycle varies between 3 to 10 seconds depending on whether the system invokes the general search operation as described in Section 4.3.2.3. Although this approach has been designed to be as general as possible, the strength of the current implementation becomes more evident when the issues associated with a specific object set are considered.

Significant emphasis was placed on system testing and evaluation. Each module was tested under various conditions in order to verify its robustness prior to integration. Next, to demonstrate the validity of the current approach, a large number of tests were conducted after the integration of the entire system. The reported results were found to be very encouraging and highlight the potential of the adopted approach. It also should be noticed that all the equipment involved in the development of this system is inexpensive.

Special attention was also given to the implementation aspects. In this regard, the current implementation was designed in order to provide a great deal of flexibility, the user can add or remove different objects easily using the tools provided by the graphical environment. In addition, the system was designed in a modular fashion, this assures the possibility of expanding the system's functionality at a later stage.

Chapter 6 - Contributions and further directions of research

6.1 Contributions

6.1.1 Contributions to the 3-D recovery using depth from defocus

One of the aims of this research was to develop a cost effective range sensor suitable for a robotic application. To be suitable for a robotic application, the depth estimator has to be mechanically robust and the 3-D information has to be estimated quickly and accurately. The approach outlined in this thesis is a bifocal range sensor based on depth from defocus. While it is acknowledged that passive DFD returns reliable depth estimation only when dealing with highly textured scenes, active DFD is preferred in many applications because it can accurately estimate the depth even in cases when the scene is textureless. However, some theoretical and implementation issues emerge when a range sensor based on active DFD is implemented. A difficult problem consists of choosing the illumination pattern. In this sense, Nayar *et al* (1995) developed a symmetrical pattern optimised for a specific camera. Their solution appears to be the best approach but several disadvantages can be mentioned:

- To fabricate a precise pattern is a difficult task and specialised technology is required.
- It requires a perfect registration between the illumination pattern and the sensors' cells.
- The changes in magnification between the near and far focused images determine unreliable depth estimation.

As a result, their implementation is costly and requires a sub-pixel sophisticated camera calibration. The problems caused by changes in magnification between images captured with different focal settings were alleviated by resorting to a telecentric lens. This solution is effective but in order to image the scene a very powerful source of light has to be utilised (see Section 2.3.3). Also, another disadvantage is the lack of flexibility as any change in the optical and sensing equipment involves redesigning the system. In addition, it is not feasible to apply this range sensor to robotic applications since it contains equipment sensitive to vibrations.

In contrast with the previous implementation, for the current approach a simple striped grid MGP-10 (10 lines per mm) was used (Ghita and Whelan, 1999a). Nevertheless, due to magnification changes between the near and far focused images the stripes do not match perfectly together. As mentioned above, this problem can be corrected on an optical basis by using a telecentric lens, but to avoid the complications associated with the use of telecentric lens, for the current implementation, the problems caused by changes in magnifications due to different focal setting are corrected by using image interpolation (Ghita and Whelan, 1999b). The experimental data indicated that the depth estimation is significantly improved when image interpolation is applied.

As noted in Section 2.2.8, several problems occur when this sensor is implemented. Among others, to compensate for the increased distance between the lens and the CMOS sensors due to the beam splitter and aligning the two CMOS sensors proved to be the most challenging. In order to facilitate an easy calibration, a multi-axis translator was attached to one of the sensors. The experimental results have indicated that the developed range sensor produces precise and fast depth estimation at very low cost.

6.1.2 Contributions to the segmentation algorithm

Because the resulting image (segmented image) represents the input of the recognition algorithm, the overall results are greatly influenced by the segmentation technique that is employed. Since the objects of interest are textureless, a natural way to approach the segmentation process is to rely on the information returned by an edge operator. Because the precision and the efficiency of this method is dependent on the edge detector, many efforts were dedicated to select the edge operator that maximise the ratio quality versus processing time. In this sense, a large variety of edge detectors were analysed under various conditions in order to evaluate their robustness.

Many similar implementations such as those suggested by Kak and Kosaka (1996) and Rahardja and Kosaka (1996) rely on the use of the Canny edge detector.

The visual framework employed in the evaluation of the edge detectors proved that the Canny edge detector due to its performance appears to be the best choice, but at a very high computational cost. The experimental data revealed that a trade-off between the quality in edge estimation and the computational efficiency is given by the ISEF edge operators (see Section 3.3.6).

Another problem relates to reconnecting the gaps between unlinked edges and eliminating the small regions caused by spurious edges. An efficient solution to close the gaps between interrupted edges relies on analysing the information derived from the singular edge points (endpoints). The particular novelty of this approach lies in the labelling scheme which assigns the directionality of the endpoints based only on local knowledge. As a consequence, it relaxes the demand of *a priori* information and assures an accurate and efficient search for edge paths in the image under investigation. The last stage involves applying a labelling algorithm that assigns a unique label to each disjointed region.

6.1.3 Contributions to the recognition algorithm

The immediate contribution of this work is the overall algorithm that provides an innovative approach which helps to solve the object recognition and pose estimation problem. To achieve this goal, the approach outlined in Chapter 4 deals with the recognition and pose estimation issues at different stages. The first stage addresses the object recognition and consists of the use of global geometrical features. Since this research deals with small polyhedral objects, only a small number of extracted faces are available. The global features derived from extracted faces are efficient only if object faces are mildly occluded, thus an important goal consists of detecting the topmost object. There are some rationales that motivate this approach and perhaps the most obvious being the observation that the object placed on top of the pile is rarely occluded. Because the developed system includes a range sensor, the algorithm selects the best placed object using 3-D information along with other selection criteria (see Section 4.3.1). Then, the recognition process consists of computing the Euclidean distance between the features that belong to the selected region and those contained in the database.

Nevertheless, this approach is appropriate if the selected object is only mildly occluded. Therefore, a main goal consists of minimising the occurrence of the false-positive recognition state. For this purpose, very strict threshold conditions were imposed on the system. If the threshold conditions are not upheld the algorithm invokes a global search operation (see Section 4.3.2.3).

The second stage estimates the position in space for the object placed on the top of the object pile. The algorithm outlined in Section 4.3.2 uses an approach based on eigenimage analysis which assures a compact and efficient representation for pose estimation. In contrast with other similar implementations, for the purpose of increasing the precision and flexibility, the present approach computes a low dimensional space (eigenspace) for every object contained in the database. This approach is very convenient because every object is individually considered while any modifications related to any object from the database will not affect the remaining ones.

6.1.4 Implementation of the entire system and the real-time approach

The aim of this thesis was to describe the theoretical framework and the development of an integrated vision system suitable for a bin picking application. The implementation was designed in a modular fashion, hence providing the possibility to expand the system's functionality at a later stage.

During the implementation a special attention was given to the real-time approach. There is no doubt that an industrial robotic application must always be as close to real time as is possible. In this sense, a hardware architecture gives the best results in terms of speed but offer very little flexibility and furthermore the overall implementation is expensive.

The aim of this research was to build a reliable and cost effective system. So, for the purpose of obtaining real-time depth estimation, a range sensor based on active depth from defocus was developed. A significant improvement was obtained when two frame grabbers were utilised to capture the near and far focused images. In addition, the segmentation algorithm is fast and all the computationally intensive operations required by the recognition and pose estimation algorithm are performed off-line.

6.2 Further directions and possible improvements

6.2.1 Improvements to the 3-D depth estimator

Because this sensor estimates the depth within a small range, the initial question was, if this sensor would be precise enough for a vision application. Although during experimentation the results were found to be very encouraging, the precision and the resolution of this sensor are restricted by two factors. The first includes technical limitations due to imperfections related to the optical and sensing equipment. The second problem is associated with the difficulty of determining an optimal solution for the illumination pattern and the focus operator. In this regard, the potential further developments are identified as follows:

- Using CCD elements with higher quality and resolution. This gives the possibility of using patterns of structured light with higher density.
- Using a very dense structured light pattern *correlated* with the resolution of the CCD sensor.
- Using a framework that allows a variable window for the focus operator in order to minimise the windowing errors.

The ideas outlined above are currently used in the implementation of a new sensor which is expected to provide better accuracy than that provided by the present bifocal range sensor.

6.2.2 Improvements to the segmentation algorithm

This current approach was motivated by the fact that textureless objects are widely used in industry and consequently it was employed for the purpose of obtaining precise segmentation for a scene that contains such objects.

At a future stage, this approach will be extended to objects with various textures. The main problem will be to discriminate between edges returned by the borders and those returned by the texture. This complication is further increased because the scene is affected by occlusions and it will be difficult to achieve meaningful segmentation using only the edge information. In the author’s view, a possible solution is to combine the developed approach with a range data segmentation technique which tries to identify the regions of different elevations.

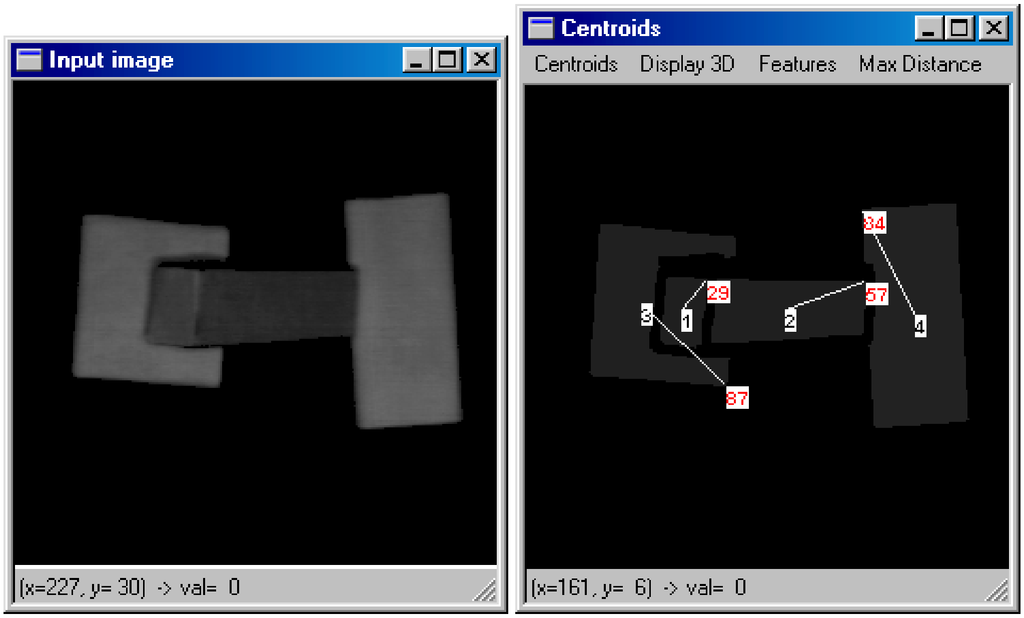
6.2.3 Improvements to the recognition and pose estimation algorithm

The proposed algorithm is based on the assumption that the object placed on the top of the object pile is rarely occluded. This assumption covers the practical cases very well, thus the algorithm performs the recognition task only for the object which is placed on the top of the object pile.

Nevertheless when all objects are highly occluded, the recognition algorithm fails in matching the correct object. The worst situation is the false-positive state when the object is not contained in the object pile and the recognition algorithm decided that it is present. To minimise the occurrence of this state, the developed algorithm uses very strict threshold conditions. If all selected regions do not fulfil the recognition conditions the scene is rearranged. Certainly, if the objects are not fragile or deformable this solution is effective. A clear disadvantage is the lack of efficiency because this operation is slow. A possible improvement relies on the use of local geometrical primitives or textural features. If the objects have different textures or colours the best approach relies on the use of textural features. In contrast, if the object set contains objects made from the same material (which is the topic of this research) the use of local geometrical primitives is a more appropriate approach. An immediate disadvantage is the fact that these features are highly influenced by the object's orientation. To compensate for this problem a database which contains a large number of orientations is required.

Also, the current approach can be further developed by using multiple regions along with other geometrical primitives (junctions, vertices and curves), this gives a better representation for objects in the recognition process. Clearly, the number of hypotheses in this case is extensive and the matching problem becomes increasingly difficult as long as the recognition of the primitives employed is dependent upon the viewing direction. It is strongly believed that a scheme that quickly *rejects* the non-plausible combinations of detected primitives represents an effective solution to reduce the number of possible hypotheses. Then, from remaining hypotheses, the plausible ones are determined using rigid matching constraints.

A typical example is illustrated in Figure 6.1 where the regions marked with 1 and 2 may give a better representation of the object in question.



(x=227, y= 30) -> val= 0

(x=161, y= 6) -> val= 0

(a) (b)

Figure 6.1. A typical scene that is successfully handled by an algorithm that employs a multiple-region recognition scheme. (a) The input image. (b) The resulting data image.

An alternative approach relies on the use of 3-D models for every object contained in the database. This solution is effective if the object or a part of it can be precisely identified in the object pile. However, to be successful, this approach requires a high quality range sensor.

The second stage of the algorithm performs the pose estimation. This approach gives a precise estimation only if a large number of positions are used in the training stage. This issue does not have a great influence on the processing time in the estimation stage because it is performed off-line. A possible improvement consists of using a flexible framework that is able to eliminate the redundant or irrelevant positions.

6.4 Conclusions

This thesis describes the theoretical framework and the development of a vision sensor for bin picking. A bin picking system needs sensorial information in order to understand and evaluate the spatial representation of the objects that form the scene. A key issue associated with a bin picking system is 3-D information acquisition. Chapter 2 discusses the DFD technique which is the topic of this current research. The DFD approach is precise only if the scene is highly textured. If the objects contained in the scene are textureless the depth estimation is inaccurate. In order to compensate for this problem, a practical solution relies on the use of active illumination which was identified to be the key issue of this implementation. Numerous experiments demonstrated that the developed bifocal sensor proved to be an attractive solution to estimate the depth quickly and accurately.

Chapter 3 describes the implementation of an edge-based segmentation technique. The aim of the segmentation process is to decompose the image into disjointed meaningful regions which have a strong correlation with the objects that define the scene. Since the edge operator plays the central role in this approach, thus a large section was dedicated to describing the most common edge operator implementations. Significant emphasis was placed on choosing the optimal edge detector. The experimental results concluded that the best option involves the use of the ISEF edge operators which gives the best trade-off between the quality in edge estimation and the computation efficiency. An important part of the segmentation algorithm consists of re-connecting the gaps between interrupted edges and eliminating the small regions created by false edges.

The next chapter describes a novel approach for object recognition and pose estimation. The developed algorithm addresses the recognition and the pose estimation tasks by using a two-stage implementation. The first stage performs the recognition task using the global geometrical parameters as features for matching. This approach can handle textureless objects with well-defined faces and proved to be efficient when the objects are not significantly occluded. Therefore, the algorithm attempts to perform the recognition task on the object placed on the top of the object pile. The second stage addresses the pose estimation and uses an approach based on eigenimage analysis. This approach estimates the position of the recognised object by using its appearance in conjunction with a range data analysis. The practical experiments have demonstrated that the proposed algorithm is a powerful and an efficient method for pose estimation.

In conclusion, the experimental data has reinforced the concepts presented in this thesis and has demonstrated that the proposed framework is a fast, accurate and inexpensive solution to the bin picking problem.

References

Asada N., Fujiwara H. and Matsuyama T. (1998), - "Seeing behind the scene: analysis photometric properties of occluding edges by the reversed projection blurring model", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 20, no. 2, pp. 157-166.*

Ballard D.H. (1981), - "Generalizing the Hough transform to detect arbitrary shapes", *Pattern Recognition, no. 13, pp. 111-122.*

Batchelor B.G. (1991), - Intelligent image processing in Prolog, *Springer-Verlag*, London.

Batchelor B.G. and Whelan P.F. (1997), - Intelligent Vision Systems for Industry, *Springer-Verlag*, London.

Bergevin R. and Levine M.D. (1993), - "Generic object recognition: building and matching coarse descriptions from line drawings", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 15, no. 1, pp. 19-36.*

Bergholm F. (1987), - "Edge Focusing", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 9, no. 6, pp. 726-741.*

Biederman I. (1987), - "Recognition by components: a theory of human image understanding", *Psychology Review, vol. 94, pp. 115-147.*

Biederman I. and Gerhardstein P.C. (1993), - "Recognizing depth-rotated objects: evidence and conditions for 3D viewpoint invariance", *Journal of Experimental Psychology: Human Perception and Performance.*

Bhatia M. (1996), - "Depth Estimation from Defocus Information", *Unpublished.*

Birk R.J., Kelley R.B. and Martins A.S. (1981), - “An orienting robot for feeding workpieces stored in bins”, *IEEE Transactions on Systems, Man and Cybernetics, vol. 11, no. 2, pp. 151-160.*

Bolles R.C. and Horaud R. (1987), - “3DPO: A Three-Dimensional Part Orientation System”, *Three Dimensional Machine Vision (Kanade T. - Editor), Kluwer Academic Publishers, pp. 399-450.*

Bosse S.K., Biswas K.K. and Gupta S.K. (1996), - “Model based object recognition – the role of affine invariants”, *Artificial Intelligence in Engineering, vol. 1, pp. 227-234.*

Brady M. and Wang H. (1992), - “Vision for mobile robots”. *Phil. Trans. Royal Soc. London B.337, pp. 341-350.*

Bresenham J.E. (1965), - “Algorithm for computer control of a digital plotter”, *IBM Systems Journal, vol. 4, no. 1, pp. 25-30.*

Brooks R.A. (1981), - “Symbolic reasoning among 3-D models and 2-D images”, *Artificial Intelligence Journal, vol. 17, pp. 268-348.*

Canny J. (1986), - “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 8, no. 6, pp. 679-698.*

Casadei S. and Mitter S.K. (1996), - “A hierarchical approach to high resolution edge contour reconstruction”, *Proceedings of the IEEE Conf. for Computer Vision and Pattern Recognition (CVPR’96), San Francisco, USA.*

Chen C., Hung Y., Chiang C. and Wu J. (1997), - “Range data acquisition using color structured lighting and stereo vision“, *Image and Vision Computing, vol. 15, pp. 445-456.*

Cutzu F. and Edelman S. (1994), - “Canonical views in object representation and recognition”, *Vision Research, vol. 34, pp. 3037-3056.*

Degunst M.E. (1990), - “Automatic Extraction of Roads from SPOT Images”, *PhD thesis, Delft University, Holland.*

Delchambre A. (1992), - Computer-aided assembly planning, *Chapman & Hall*.

Dessimoz J.D., Birk J.R., Kelley R.B. and Martins H.A. (1984), - “Matched filters for bin-picking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), no. 6, pp. 686-697.*

Dickinson S.J., Pentland A.P. and Rosenfeld A. (1992), - “3-D Shape Recovery Using Distributed Aspect Matching”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 2, pp. 174-198.*

Distante A., Ancona N., Attolico G., Caponetti L., Chiaradia M. and Stella E. (1988), - “A model-based 3-D vision system for bin-picking”, *IEEE Transactions on Circuits and Systems, vol. 35, no. 5, pp. 545-553.*

Edelman S. (1995), - “Class similarity and viewpoint invariance in the recognition of 3D objects”, *Biological Cybernetics, vol. 72, pp. 207-220.*

Eichel P.H. and Delp E.J. (1985), - “Sequential edge detection in correlated random fields”, *Proceedings of the IEEE Conf. for Computer Vision and Pattern Recognition, pp. 14-21, San Francisco, USA.*

Fan T.J., Medioni G. and Nevatia R. (1989), - “Recognising 3-D objects using surface descriptions”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 11, no. 11, pp. 1140-1157.*

Faugeras O.D. (1993), - Three-Dimensional Computer Vision, *MIT Press*.

Freeman H. (1961), - "On the encoding of arbitrary geometric configuration", *IRE Transactions on Electronic Computers, EC-10(2), pp. 260-268.*

Forsyth D., Mundy J.L., Zisserman A., Coelho C., Heller A. and Rothwell C. (1991), - "Invariant Descriptors for 3-D object recognition and pose", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 13, no. 10, pp. 971-991.*

Gonzales R. and Wintz P. (1987), - Digital image processing, *Addison – Wesley Publishing Company.*

Grossman P. (1987), - "Depth from focus", *Pattern Recognition Letters, vol. 5, no. 1, pp. 63-69.*

Gupta A.K., Chaudhury S. and Parthasarathy G. (1993), - "A new approach for aggregating edge points into edge segments", *Pattern Recognition, vol. 26, no. 7, pp. 1069-1086.*

Hancock E.R. and Kittler J. (1990), - "Edge-labelling using dictionary based relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 12, no. 2, pp. 165-181.*

Haralick R.M. and Shapiro L.G. (1992), - Computer and Robot Vision, *Addison - Wesley Publishing Company*.

Heath M., Sarkar S., Sanocki T. and Bowyer K. (1997), - "Comparison of edge detectors: a methodology and initial study", *Computer Science and Engineering, University of South Florida, Technical report.*

Henderson C.T. (1983), - "Efficient 3-D object representation for industrial vision systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 5, no. 6, pp. 609-617.*

Horii A. (1992), - “The Focusing Mechanism in the KTH Head Eye System”, *Royal Institute of Technology, Stockholm, Sweden, Technical report.*

Horn B.K.P. (1977), - “Understanding image intensities”, *Artificial Intelligence, vol. 8, pp. 201-231.*

Horn B.K.P. (1979), - “SEQUINS and QUILLS – representation for surface tomography”, *AI Memo no. 536, Artificial Intelligence Laboratory, MIT.*

Horn B.K.P. (1983), - “Extended Gaussian Images”, *AI Memo no. 740, Artificial Intelligence Laboratory, MIT.*

Ikeuchi K. (1983), - “Determining attitude of object from needle map using Extended Gaussian Image”, *AI Memo no. 714, Artificial Intelligence Laboratory, MIT.*

Jain A.K. and Dubes R.C. (1988), - Algorithms for Clustering Data, *Prentice Hall.*

Jarvis R.A. (1983), - “A Perspective on Range Finding Techniques for Computer Vision”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 5, no. 2, pp. 122-139.*

Johnson A.E. and Hebert M. (1999), - “Using spin images for efficient object recognition in cluttered 3D scenes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 21, no. 5, pp. 433-449.*

Jones A.G. and Taylor C.J. (1994), - “Robust shape from shading”, *University of Manchester, UK, Technical report.*

Kak E.A. and Edwards J.L. (1995), - “Experimental State of the Art in 3-D Object Recognition and Localization Using Range Data”, *Proc. of Workshop on Vision for Robots in IROS’95 Conference, Pittsburgh, PA.*

Kak A. and Kosaka A. (1996), - "Multisensor Fusion for Sensory Intelligence in Robotics", *Proceedings of Workshop on Foundations of Information/Decision Fusion: Applications to Engineering Problems*, August 7-9, Washington, D.C.

Kanade T., Hiroshi K., Kimura S. and Yoshida A. (1995), - "Development of a Video-Rate Stereo Machine*", Proceedings of International Robotics and Systems Conference (IROS '95), Pittsburgh, PA, USA.*

Kang S.B. and Ikeuchi K. (1990), - "3-D object pose determination using complex EGI", *Transfer report CMU-RI-TR-90-18, Carnegie Mellon University*.

Kelley B.P., Birk J.R., Martins A.S. and Tella R. (1982), - "A robot system which acquires cylindrical workpieces from bins", *IEEE Transactions on Systems, Man and Cybernetics, vol. 12, no. 2, pp. 204-213.*

Kelley B.P., Martins A.S., Birk J.R. and Dessimoz J.D. (1983), - "Three vision algorithms for acquiring workpieces from bins", *Proc. of the IEEE, vol. 71, no. 7.*

Kim W. and Kak A. (1991), - "3-D object recognition using bipartite matching embedded in discrete relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 3, pp. 224-251.*

Kriegman D. and Ponce J. (1990), - "On recognizing and positioning curved 3-D objects from image contours", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), no. 12, pp. 1127-1137.*

Krishnapuram R. and Casasent D. (1989), - "Determination of three-dimensional object location and orientation from range images", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 11, no. 11, pp. 1158-1167.*

Krotkov E. (1987), - "Focusing", *International Journal of Computer Vision, vol. 1, pp. 223-237.*

Lamdan Y., Schwartz J.T. and Wolfson H. (1988), - "Object recognition by affine invariant matching", *Proceedings of CVPR.*

Lindeberg T. (1993), - "On scale selection for differential operators", *Proceedings of 8th Scandinavian Conf. on Image Analysis, pp. 857-866, Tromso, Norway*.

Marr D. and Hildreth E. (1980), - "Theory of edge detection", *Proceedings of Royal Society, London, B 207, pp. 187-217.*

Mc Donald J. and Vernon D. (1998), - "A New Hough Transform for the Detection of Arbitrary 3-Dimensional Objects", *Proceedings of the Optical Engineers Society of Ireland and the Irish Machine Vision and Image Processing Joint Conference, National University of Ireland, Maynooth.*

Moghaddam B. and Pentland A. (1994), - "Face recognition using view-based and modular eigenspaces", *Automatic Systems for the Identification and Inspection of Humans, SPIE, vol. 2277.*

Murakami H. and Kumar V. (1982), - "Efficient calculation of primary images from a set of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 4, no. 5, pp. 511-515.*

Molloy D. and Whelan P.F. (1997), **-** "Self initialising active contours for motion discrimination", *IMVIP Irish Machine Vision and Image Processing Conference, pp. 139-146.*

Murase H. and Nayar S.K. (1995), - "Visual learning and recognition of 3-D objects from appearance", *International Journal of Computer Vision, 14, pp. 5-24.*

Nayar S.K., Watanabe M. and Noguchi M. (1995), - "Real-Time Focus Range Sensor", *Proceedings of International Conference on Computer Vision (ICCV 95), pp. 995-1001.*

Nene S.A. and Nayar S.K. (1995), - "A Simple Algorithm for Nearest Neighbour Search in High Dimensions", *Columbia University, Technical report CUCS-030-95.*

Nene S.A., Nayar S.K. and Murase H. (1994), - "SLAM: A Software Library for Appearance and Matching", *Proceedings of ARPA Image Understanding Workshop, Monterey.*

Ohba K. and Ikeuchi K. (1997), - "Detectability, Uniqueness and Reliability for Stable Verification of Partially Occluded Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 19, no. 9, pp. 1043-1048.*

Okutomi M. and Kanade T. (1993), - "A Multiple-Baseline Stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 15, no. 4, pp. 353-363.*

Pentland A.P. (1987), **-** "A new sense for depth of field", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 9, no. 4, pp. 523-531.*

Pentland A.P., Darrell T., Turk M. and Huang W. (1989), - "A simple, real-time range camera", *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 256-261.*

Pentland A.P., Scherock S., Darrell T. and Girod B. (1994), - "Simple range cameras based on focal error", *Journal of Optical Society of America, vol. 11, no. 11, pp. 2925-2935.*

Pollard B.S., Mayhew J.E.W. and Frisby J.P. (1985), - "PMF: A stereo correspondence algorithm using a disparity gradient limit", *Perception, no. 14, pp. 449-470.*

Porter A.L., Rossini F.A., Eshelman J., Jenkins D.D. and Cancelleri D.J. (1985), - "Industrial robots – A strategic forecast using the technological delivery system approach", *IEEE Transactions on Systems, Man and Cybernetics, vol. 15, no. 4, pp. 521-527.*

Press W.H., Teukolsky S.A., Vetterling W.T. and Flannery B.P. (1992), - Numerical Recipes in C, *Cambridge University Press.*

Pudney C.J. (1995), - “Surface Following for Manipulators with Proximity Sensors”, *University of Western Australia, Technical report.*

Rahardja K. and Kosaka A. (1996), - “Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues”, *Robot Vision Laboratory, Purdue University.*

Ramesh V. and Haralick R.M. (1992), - “Performance Characterization of Edge Detectors”, *Proceedings of SPIE, vol. 1708, pp. 252-266.*

Rechsteiner M., Schneuwly B. and Guggenbuhl W. (1992), **-** “Fast and precise 3-D sensor insensitive to ambient light”, *Proceedings of SPIE Optics, Illumination and Image Sensing for Machine Vision, vol. 1822.*

Saber E., Tekalp A.M. and Bozdagi G. (1997), - “Fusion of color and edge information for improved segmentation and edge linking”, *Image and Vision Computing, vol. 15, no. 10, pp. 769-780.*

Sato Y., Hasegawa K. and Hattori K. (1999), - “3D face measurement system with Cubicscope”, *Proceedings of Irish Machine Vision and Image Processing (IMVIP), Dublin City University, Dublin, Ireland.*

Schiele B. and Crowley J.L. (1996), - “Probabilistic object recognition using multidimensional receptive field histograms”, *International Conference on Pattern Recognition, Vienna, Austria, vol. B, pp. 50-54.*

Shen J. and Castan S. (1992), - “An Optimal Linear Operator for Step Edge Detection”, *CVGIP: Graphical Models Image Processing, vol. 54, no. 2, pp. 112-133.*

Sirovich L. and Kirby M. (1987), - “Low-dimensional procedure for the characterization of human faces”, *J. Opt. Soc. Amer., vol. 4, no. 3, pp. 519-524.*

Smith S.M. (1992), - “Feature Based Image Sequence Understanding”, *PhD thesis, Robotics Research Group, Department of Engineering Science, Oxford University.*

Smith S.M. and Brady J.M. (1995), - “ASSET-2: Real-time motion segmentation and shape tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 17, no. 8, pp. 814-820.*

Snyder W.E., Groshong R., Hsiao M., Boone K.L. and Hudacko T. (1992), - “Closing Gaps in Edges and Surfaces”, *Image and Vision Computing, vol. 10, no. 8, pp. 523-531.*

Sonka M., Hlavac V. and Boyle R. (1993), - Image Processing, Analysis and Machine Vision, *Chapman & Hall.*

Stein F. and Medioni G. (1992), - ”Structural indexing efficient 3-D object recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 14, no. 2, pp. 125-145.*

Subbarao M. (1988), - “Parallel depth recovery by changing camera parameters”, *Proceedings of the 2nd International Conference on Computer Vision, pp. 149-155.*

Subbarao M. (1991), - “Spatial-Domain Convolution/Deconvolution Transform”, *State University of New York, Technical report-910703.*

Subbarao M. and Surya G. (1994), - “Depth from Defocus: A Spatial Domain Approach”, *International Journal of Computer Vision, vol. 13, no. 3, pp. 271-294.*

Swain M. and Ballard D. (1991), - “Color indexing”, *International Journal of Computer Vision, vol. 7, no. 1, pp. 11-32.*

Tenenbaum J.M. (1970), - “Accommodation in Computer Vision”, *PhD thesis, Stanford University.*

Terzopoulus D. and Metaxas D. (1991), - “Dynamic 3-D models with local and global deformations: deformable superquadrics”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 13, no. 7, pp. 703-714.*

Tsang P.W. and Yuen P.C. (1993), - “Recognition of partially occluded objects”, *IEEE Transactions on Systems, Man and Cybernetics, vol. 23, no. 1, pp. 228-236.*

Turk M., Pentland A.P., Darrell T. and Huang W. (1989), -“A simple, real-time range camera”, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

Turk M. and Pentland A. (1991a), **-** “Eigenfaces for recognition”, *Journal of Cognitive Neuroscience, vol. 3, pp. 71-86.*

Turk M. and Pentland A. (1991b), **-** “Face recognition using eigenfaces”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586-591.*

Vernon D. (1991), - Machine Vision: Automated Visual Inspection and Robot Vision, *Prentice-Hall.*

Vijayakumar B., Kriegman D.G. and Ponce J. (1998), - “Invariant-based Recognition of Complex Curved 3-D Objects from Image Contours”, *Computer Vision and Image Understanding, vol.72, no. 3, pp. 287-303.*

Vincent L. (1993), - “Morphological greyscale reconstruction in image analysis: applications and efficient algorithms”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 2, no. 2, pp. 176-201.*

Vincken K.L., Niessen W.J. and Viergever M.A. (1996), - “Blurring strategies for image segmentation using multiscale linking model”, *Proceedings of the IEEE Conf. for Computer Vision and Pattern Recognition (CVPR’96), San Francisco, USA.*

Watanabe M. and Nayar S.K. (1995a), - “Telecentric Optics for Computational Vision”, *Columbia University, Technical report, CUCS-026-95.*

Watanabe M. and Nayar S.K. (1995b), - “Rational Filters for Passive Depth from Defocus”, *Columbia University, Technical report, CUCS-035-95.*

Wechsler H. and Zimmerman L. (1988), - “2-D invariant object recognition using distributed associative memory”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 10, no. 6, pp. 811-821.*

Whelan P.F. and Batchelor B.G. (1996), - "Automated Packing Systems - A Systems Engineering Approach", *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, vol. 26, no. 5, pp. 533-544.*

Xiong Y. and Shafer S.A. (1993), - “Depth from focusing and defocusing”, *Carnegie Mellon University, Technical report CMU-RI-TR-93-07.*

Xiong Y. and Shafer S.A. (1994), - “Variable window Gabor filters and their use in focus and correspondence”, *Proceedings of Computer Vision and Pattern Recognition, pp. 668-671.*

Xiong Y. and Shafer S.A. (1995), - “Dense structure from a dense optical flow sequence”, *Carnegie Mellon University, Technical report CMU-RI-TR-95-10.*

Yoshimi B.H. and Allen P. (1994), - “Visual control of grasping and manipulation tasks”, *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, NV.*

Yuille A. and Geiger D. (1990), - “Stereo and controlled movement”, *International Journal of Computer Vision, vol. 4, pp. 141-152.*

Zisserman A., Forsyth D., Mundy J., Rothwell C., Liu J. and Pillow N. (1994), - “3D Object Recognition Using Invariance”, *University of Oxford, Technical report OUEL 2027/94, UK.*

Bibliography

Al-Hujazi E. and Sood A. (1990), - "Range image segmentation with applications to robot bin-picking using vacuum gripper", *IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 6, pp. 1313-1324.*

Amit Y., Geman D. and Wilder K. (1995), - "Recognizing shapes from simple queries about geometry", *Technical report, University of Massachusetts.*

Baily T. and Jain A.K. (1978), - "A note on distance-weighted k-nearest neighbour rules", *IEEE Transactions on Systems Man and Cybernetics, vol. 8, no. 4, pp. 311-313.*

Besl P. and Jain R. (1985), - "Three-dimensional object recognition", *ACM Computing Surveys, vol. 17, no. 1, pp. 75-145.*

Butler P., O'Brion E. and Vernon D. (1998), - "A Hand-Activated White-Light Profilometry System to Effect the Automatic Recovery of Facial Shape", *Proceedings of the Optical Engineering Society of Ireland & Irish Machine Vision and Image Processing Joint Conference, National University of Ireland, Maynooth, Ireland.*

Chen C.H. and Kak A.C. (1989), - "A robot vision system for recognising 3-D objects in low-order polynomial time", *IEEE Transactions on Systems, Man and Cybernetics, vol. 19, no. 6, pp. 1535-1563.*

Darrell T. and Wohn K. (1988), - "Pyramid Based Depth from Focus", *Proceedings of IEEE Conf. for Computer Vision and Pattern Recognition (CVPR), pp. 504-509.*

Dupuis P. and Oliensis J. (1992), - "Direct method for reconstructing shape from shading", *IEEE Computer Vision and Pattern Recognition Conference, pp. 453-458.*

Ikeuchi K. (1987), - “Generating an interpretation tree from a CAD model for 3-D object recognition in bin-picking tasks”, *International Journal of Computer Vision, vol. 1, no. 2, pp. 145-165.*

Ikeuchi K. and Kanade T. (1988), - “Automatic generation of object recognition programs”, *Proceedings of the IEEE, vol. 76, no. 8, pp. 1016-1035.*

Jain A.K. (1989), - Fundamentals of Digital Image Processing, *Prentice Hall.*

Kriegman D., Vijayakumar B. and Ponce J. (1993), - “Reconstruction of HOT curves from images sequences”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 20-26.*

Lamdan Y. and Wolfson H. (1988), - “Geometric hashing: A general and efficient model-based recognition scheme”, *Proceedings of IEEE International Conferences for Robotics and Automation, Philadelphia, pp. 1407-1413.*

Murase H. and Nayar S.K. (1994), - “Illumination planning for object recognition in structured environments”, *IEEE Conference on Computer Vision and Pattern Recognition, pp. 31-38, Seattle.*

Nayar S.K. and Bolle R. (1996), - “Reflectance based object recognition”, *International Journal of Computer Vision, vol. 17, no. 3, pp. 219-240.*

Oxford (1995), - “Concise Oxford Dictionary”, *New Edition.*

Phong T.Q., Horaud R., Yassine A. and Tao P.D. (1995), - “Object pose from 2-D to 3-D point and line correspondences”, *International Journal of Computer Vision, vol. 15, no.3, pp. 225-243.*

Ponce J., Hoogs A. and Kriegman D.J. (1992), - "On using CAD models to compute the pose of curved 3-D objects", *CVGIP Image Understanding, no. 55, pp. 184-197.*

Pudney C.J. (1992), - "Surface Following and Modelling for Planar Robots", *University of Western Australia, Technical report.*

Rogers S.K. and Kabrisky M. (1991), - "An Introduction to Biological and Artificial Neural Networks for Pattern Recognition", *SPIE Optical Engineering Press, vol. T4.*

Subbarao M. and Gurmoorthy N. (1988), - "Depth Recovery from Blurred Edges", *Proceedings of IEEE Conf. for Computer Vision and Pattern Recognition (CVPR), pp. 498-503.*

Vernon D. (1991), - Machine Vision: Automated Visual Inspection and Robot Vision, *Prentice-Hall.*

Wechsler H. and Zimmerman L. (1989), - "Distributed Associative Memory (DAM) for Bin-Picking", *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 11, no. 8, pp. 814-822.*

Whelan P. and Molloy D. (2000), - Machine Vision Algorithms in Java: Techniques and Implementation, *Springer-Verlag, London.*

Xiong Y. and Shafer S.A. (1994), - "Moment Filters for Precision Computation of Focus and Stereo", *Carnegie Mellon University, Technical report CMU-RI-TR-94-28.*

Appendix A – Popular range sensing techniques

A.1 3-D dynamic laser scanning

Laser scanning is one of the most popular optical range acquisition approaches. This technique is also called light stripe sectioning, structured lighting or optical triangulation. The shape is recovered by translating or rotating the object while the laser beam (or structured light) is swept over the object under analysis. The number of images acquired during this stage determines the resolution of the resulting depth map. As would be expected for a high resolution 3-D estimation, a large numbers of images must be captured and consequently the scan period will be very long. In order to decrease the scanning time, more laser stripes with different frequencies can be used, but in this case the range sensor will be more sophisticated and costly.

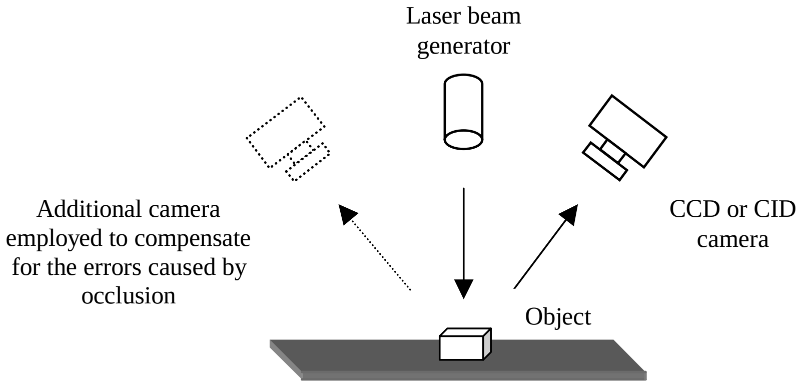

Laser beam generator
Additional camera employed to compensate for the errors caused by occlusion
CCD or CID camera
Object

Figure A.1. The 3-D laser scanning principle.

Also, it is worth mentioning that due to self-occlusions some points may only be seen when observed from particular angles. To compensate for this problem, multiple range images acquired from different viewpoints are necessary to be examined in order to capture the entire shape of the object. A typical system configuration is illustrated in Figure A.1. To compensate for deficiencies caused by self-occlusions, the set-up must also include an additional camera which is marked with a dashed line. In Figure A.2 a simple geometrical analysis reveals that the elevation of the scanned surface (which is in direct relation to the parameter *D*) can be easily computed by appling the relationship illustarated in Equation A.1.

$$H = \frac{D \times L(1 + \tan^2(A))}{S \times \tan(A) + D} \tag{A.1}$$

where *L*, *H*, *A*, *S*, *D* are the geometrical parameters shown in Figure A.2.

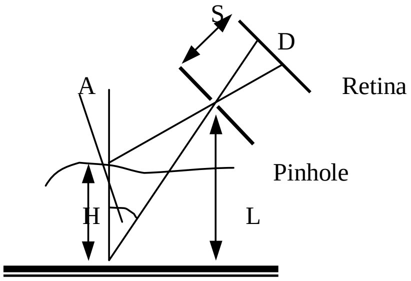


Figure A.2. The ray geometry (from Batchelor and Whelan, 1997).

Figure A.3 illustrates the image captured by camera when the laser beam is projected over the object. Obviously, from this image only the depth information for a single line is recovered.

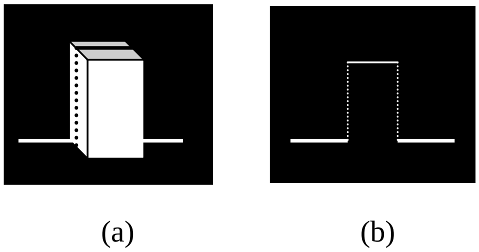


Figure A.3. The depth reconstruction process. (a) The real view. (b) The image captured by camera.

In order to recover a depth map of resolution 512 x 512 pixels, it is necessary to examine 512 images until the depth structure of the entire scene is completely determined. In this case, the depth map can be formed in 10.24 seconds if we consider that a typical frame scan period is 0.040 seconds (Batchelor and Whelan, 1997).

A number of techniques have been suggested for the purpose of reducing the scanning time. An elegant way to address this problem relies on the use of a space encoding method. This approach was employed by Sato *et al* (1999) in the implementation of a rangefinder system called Cubicscope[14]. They proposed an effective solution to generate the spatial stripe patterns using a laser diode and a polygonal mirror with 12 faces which is attached to a servomotor (see Figure A.4). This implementation needs 200 ms to generate a range image with a resolution of 512 x 242 pixels. A disadvantage of this technique is that it requires a complex hardware set-up to synchronise the rotational speed of the mirror with the video signal, a solution that makes this implementation expensive.

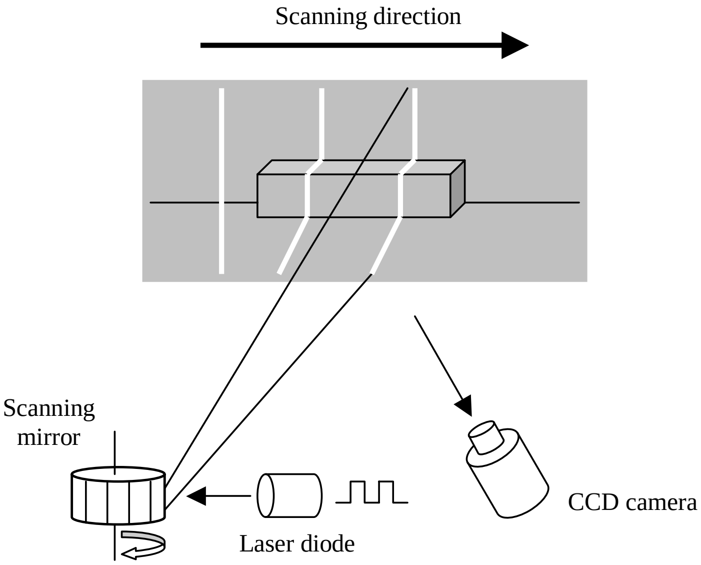


Figure A.4. Generation of spatial stripe patterns by scanning[15].

A different approach for improving the scanning time is to use a multicoloured band light projector while it is easier to solve the problem with line connectivity.

As opposed to the implementation proposed by Sato *et al* (1999), this approach does not require specialised hardware, but a dense illumination pattern has to be employed for high resolution range estimation. Also, the correlation between any two segments of a consecutive sequence of light stripes should be as small as possible in order to minimise the mismatch (Chen *et al*, 1997). Nevertheless, a dense illumination

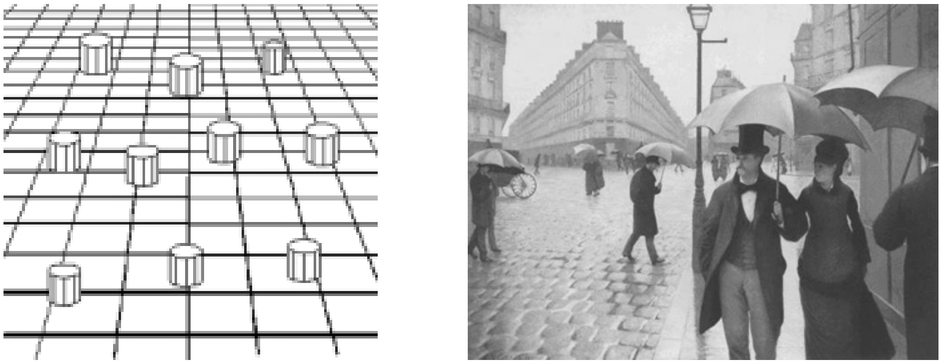
[14] The Cubiscope range finder was developed at Nagoya Institute of Technology.

[15] This image was modified from http://hilbert.elcom.nitech.ac.jp/CubicscopeHP/principle/index.html.

pattern that fulfils the abovementioned conditions is costly and furthermore several problems related to the object's reflections hinder the precision of this ranging technique. In spite of this, optical triangulation offers a powerful method for precise 3-D estimation and consequently this ranging technique is widely used in the implementation of various robotic applications.

A.2 Depth from texture

Cognitive scientists indicate that there is clear psycho-physical evidence that humans have the ability to extract depth from the views of image content in correlation with *a priori* information.



(a) (b)

Figure A.5. Depth from texture. (a) A simple example. (b) A complex scene[16].

To support this observation, in the first image it is easy enough for a human observer to estimate the depth from the floor texture. The cells on the floor are bigger for nearer distances and smaller (and distorted) for longer distances. In the second image[17], we can estimate the depth from the pavement as well, but we can also use the texture of buildings and the height of people on the street as an indication of depth. However, this example is more intuitive. In computer vision in order to evaluate the depth of the scene, it is necessary to extract the *texture primitive* (texel) which is usually recovered by applying low-level processing.

[16] These images were obtained from Hanover College, Department of Psychology. http://psych.hanover.edu/Krantz/art/texture.html.

[17] Rue de Paris, temps de pluie (Paris street, a rainy day) by Gustave Caillebotte (1848-1894).

The angle at which the surface is seen would cause a perspective distortion of the texel and the distance from the observer will change the size of the texel. This property is illustrated in Figure A.5-a where the floor pattern is distorted according to the distance from the observer. For the human observer there are three properties used in the depth recovery. The first is the *distance* from the observer, the second is the *slant* (the angle between the normal to the surface and the line of sight) and the third is the *tilt* (the direction in which the slant takes place).

To re-capture some of this information, one solution is to apply a *texture gradient* method in which the depth information is given by the direction of maximum rate of change of the perceived size of texel. Nevertheless, if the image under analysis has a simple texture, this ranging technique may give acceptable results. Alternatively, if the texture is complex (in this case it is difficult to extract the texel accurately), this approach will return unreliable depth estimation. In addition, it is important to note that for dynamic systems where objects appear and disappear in the scene, it is even more difficult to solve this problem. As a consequence, although psycho-physical studies proved that there is a close relationship between the depth and texture distortion, this ranging method due to its complexity and the fact that requires a large amount of *a-priori* information about the scene under analysis, it is rarely used in the implementation of current machine vision systems.

A.3 Infrared scanning

There are many types of non-contact *proximity* sensors that can be applied to surface following, but *ultra-sonic* (US) and *infrared sensors* (IR) are among the most popular. It is worth mentioning that due to the interference problems caused by industrial environment, the use of US sensors is restricted to some application areas. IR sensing devices contain a large array of 500 or more sensing elements and provide a relatively precise distance estimation.

Figure A.6 depicts the principle of this sensor where the sensing elements are placed at the intersection of two perpendicular lines, each containing an IR emitter and detector pair. This range sensor works as follows: the emitter projects a ray of IR light. If the object’s surface is intersected by the ray (up to a certain distance from the sensor) the reflected light is detected by the corresponding IR detector. The intensity (energy) of the reflected light is in direct correlation with the distance to the object.

An important technological problem associated with this range sensor is the finite number of sensing elements. Therefore, the resolution of the depth map is limited and it is much smaller than the resolution offered by a standard CCD camera.

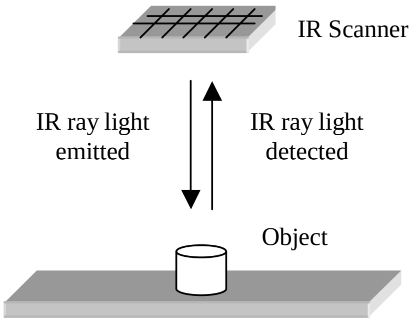


Figure A.6. Infrared scanning principle.

The best way to overcome this problem is to scan the shape of the object in sections and then to merge the sections in order to obtain the depth map of the entire object. As mentioned earlier, IR scanning is a non-contact proximity method and during the shape detection a safe minimal distance between the arm of the robot and the object's surface is maintained. After the depth map is computed, the object can be manipulated within the robot's workspace.

This method is sensitive to the reflectance of the material, hence it is necessary to perform a pre-calibration in accordance with the material in question. Pudney (1995) reported good results for shape recovery but he did not mention the computational time involved. For the purpose of minimising the processing time, a hardware implementation for a real time range sensor can be developed.

A.4 Depth from motion

This approach is based on the idea of recovering the depth information from a sequence of images using the relative motion between camera and the scene. Consequently, this method is based on analysing the optical flow and in this sense various techniques have been developed to solve this problem. The most popular techniques are: the *local-based* approach, *gradient-based* methods, *energy-based* techniques and *extended Kalman filtering* (EKF).

Usually, these methods are based on a two-stage implementation. The aim of the first stage is to extract the features from the images that may be useful in describing the optical flow, while the goal of the second stage is to compute the 3-D structure. The motion discontinuities in the optical flow are determined by analysing the correspondence between the features contained in a sequence of images. Brady and Wang (1992) developed an algorithm for scene reconstruction based on analysing the optical flow. They also presented an interesting approach to calculate 3-D structure using stereo disparity (stereopsis) in correlation with corner detection.

Molloy and Whelan (1997) present a novel approach based on corner detection, but their aim is to implement a navigation system detecting the image motion from the corners correspondence.

An interesting method based on EKF was developed by Xiong and Shafer (1995). The EKF is applied to update the 3-D structure using the information from the previous iteration until the output (depth map) is stable. The block diagram for this approach is shown in Figure A.7.

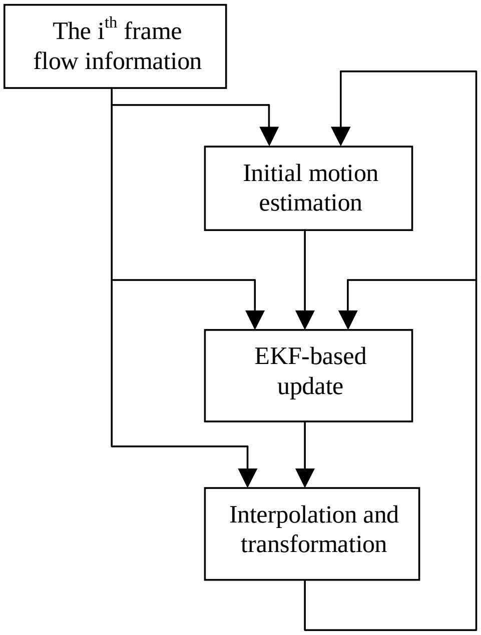

The ith frame flow information
Initial motion estimation
EKF-based update
Interpolation and transformation

Figure A.7. The block diagram of the system described in Xiong and Shafer (1995).

The block details depicted in Figure A.7 are outlined below:

- *Initial motion estimation*: This block uses the current information from the optical flow and predicted 3-D structure to compute the motion information for the current frame. Once this information is estimated, the motion parameters are re-calculated so that they are equally sensitive to flow variations.
- *EKF-based update*: This block uses the current flow information, predicted information and initial motion information to compute the *posteriori* motion information.
- *Interpolation and transformation*: This block converts the structural information from the *a priori* co-ordinate system into the *posteriori* co-ordinate system using geometrical transforms such as interpolations, rotations and translations.

The depth recovery using this method is precise (especially for objects with simple shapes), but the optical flow tends to contain a large number of frames. To solve this problem completely is computationally intensive and a real-time implementation still remains a challenge.

A.5 Shape from stereo

The key problem with this approach is that it must search for the correct match for a point within the image. This is called the *correspondence problem* and is the key problem in shape from stereo implementations. Differences between the left and right images of a stereo pair are used in computer vision for the purpose of recovering 3-D information of the scene. The algorithms employed are usually slow, restricted to a limited range of images and require *a priori* information. A simple diagram depicted in Figure A.8 demonstrates how the depth of the scene using two cameras separated by a known distance can be determined. The set-up shown in Figure A.8 consists of two cameras separated by a distance 2h, where P_l and P_r represent the left and the right projection of a scene point $P(x,y,z)$.

In the same figure, the variable $x = 0$ defines the position midway between the two cameras, the z-axis represents the distance to the object (y axis is into the page) and the variables x_l and x_r are the distances from the centre of images. From disparity between P_l and P_r in correlation with cameras positions, the z co-ordinate can be calculated using the relationship illustrated in Equation A.3.

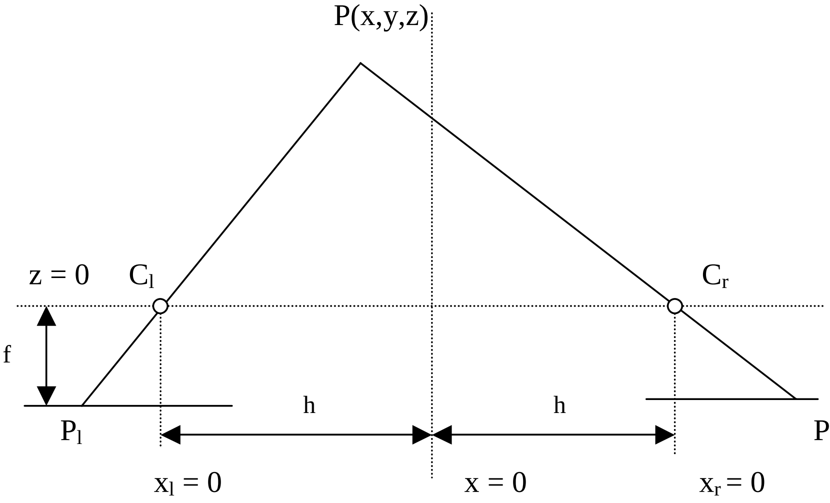


Figure 2.8. The basics of stereo geometry (Sonka *et al*, 1993).

As expected, if P_r - $P_l = 0$ denotes that the point in question is placed at a infinite distance from cameras.

$$\frac{P_l}{f} = -\frac{h+x}{z}; \qquad \frac{P_r}{f} = \frac{h-x}{z} \tag{A.2}$$

$$z(P_r - P_l) = 2hf \qquad \Rightarrow \qquad z = \frac{2hf}{P_r - P_l} \tag{A.3}$$

Before performing any 3-D reconstruction of the scene it is necessary to solve the problems related to camera calibration. The calibration procedure involves precisely determining the settings of the cameras such as the focal distance of the lenses and their relative spatial positions. The only remaining problem is to determine the correspondence between features contained in a pair of stereo images. These features can be edges, corners or other primitives. After the feature detection stage is completed, a matching algorithm has to be applied in order to solve the feature correspondence problem.

A well known solution to this problem is given by the PMF algorithm, named after its inventors (Pollard *et al*, 1985), where the edges from both images (left and right) are used as primitives for the matching algorithm. To deduce the correspondence, three constrains are applied: the first being *geometric* (epipolar constraint), the second is *intuitive* (this states that a pixel from left image can correspond to only one pixel in the right image) and the third is based on the *similarity* with human vision (disparity gradient limit). The epipolar constraint is based on the detection of the epipolar line which is the intersection between the epipolar plane (the plane defined by the optical centres C_1 and C_2 and the scene point under investigation X) and image planes. As can be easily observed in Figure A.9, for a non-parallel camera alignment the search space required to match the features in the left and right images is 2-D. To reduce the dimensionality of the search space it is necessary to apply a geometric transformation that changes the non-parallel camera alignment into a parallel configuration. This transformation is very often referred to as image rectification. Unfortunately, this transformation implies re-sampling that induces loss of information and in addition increases significantly the computational load associated with the matching algorithm. To avoid this complication and to simplify the matching algorithm, a parallel camera alignment (illustrated in Figure A.8) is commonly used. It is important to note that in this case the epipolar lines are parallel and as a consequence the search space is reduced to 1-D. Once the correspondence problem is solved, the depth structure can be easily computed as mentioned earlier by applying the relationship illustrated in Equation A.3.

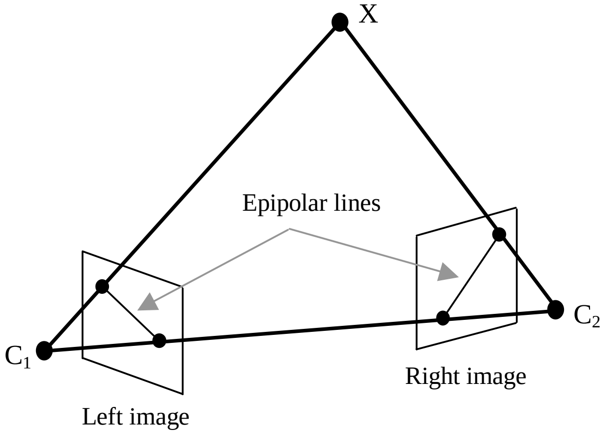


Figure A.9. The epipolar geometry.

The PMF algorithm is not the only solution to the stereo correspondence. Faugeras (1993) proposed a method based on *dynamic programming* (DP) to solve the problem of matching primitives between a pair of stereo images. The matching problem is formulated as a problem of minimising a cost function. Because DP is a way of efficiently minimising functions of a large number of discrete variables, thus matching primitives using this approach may represent a natural solution. Yuille and Geiger (1990) proposed the *Controlled Movement* approach to solve the stereo correspondence using multiple frames. The proposed stereo matching scheme consists of two parts. First the features in the left and right images are detected and an *initial stereo match* is performed. Then, a *rotation depth test* is applied where a match is accepted only if the estimated range error is compatible. This operation is followed by operations such as a *ratio test* and a *stereo test* that verify the matching consistency. This method is the most precise in terms of depth recovery, but unfortunately it is slow and additionally requires a precise calibration procedure for both cameras.

There are two problems that limit the accuracy offered by systems which employ only two cameras. The first problem is caused by the repetition of a similar pattern over a large region of the scene, a situation when due to ambiguity in feature matching the resulting depth is imprecise. The second problem is associated with the choice of the length of the baseline. It is well known that in stereo processing a short baseline will reduce the precision because the disparity between the features contained in the stereo images is narrowed. On the other hand, a larger baseline increases the disparity but other issues such as missing parts in the left and right images and the difficulty to find a reliable match within a larger disparity range have to be considered. Therefore, a trade-off between precision and accuracy in matching has to be established.

These problems were initially addressed by Okutomi and Kanade (1993) when they proposed a stereo system that uses multiple stereo pairs with various baselines. In this implementation the set-up consists of a number of CCD cameras (more than two) that are precisely aligned in a linear manner. Later, using this idea Kanade *et al* (1995) developed a video-rate stereo machine able to generate accurate depth maps at a rate of 30 frames/second. In this implementation the authors used a set-up that employs up to 6 cameras disposed in an L-type configuration. The developed stereo machine generates dense disparity maps but the system is very expensive as long as it is built on a hardware architecture. This system may be useful for a large scale of applications ranging from obstacle avoidance to robotic bin picking.

A.6 Shape from shading

Human vision has adapted to make very good use of clues from shadows and shading in general. Images of 3-D objects often show variations in brightness or shading across object surfaces and these variations provide useful information for recovering the shape of the object. Shape recovery using this approach is known as shape from shading. A good example is an artificially illuminated scene. Our brain can easily recover "the shape" of an object from degree of illumination. Using this approach, the shape of objects can be reconstructed from a single greyscale image (Horn, 1977).

A key problem for this approach is to associate the reflectance map with the surface of the object to be analysed. The reflectance map describes the relation between the intensity (brightness) of a particular pixel and the orientation of the surface which is given by the normal vector to the surface at the point of interest. The brightness of the surface under investigation is directly related to parameters such as: surface gradient, position of the light source and reflectance of the surface. Nonetheless, the reflectance properties of the surface vary from image to image. Thus, for a *specular* surface the reflected light depends on the incident angle of the light source. Alternatively, for a *matte* surface the reflected light is equal for all directions and depends only on the incident angle of the source light. An example is presented in Figure A.10.

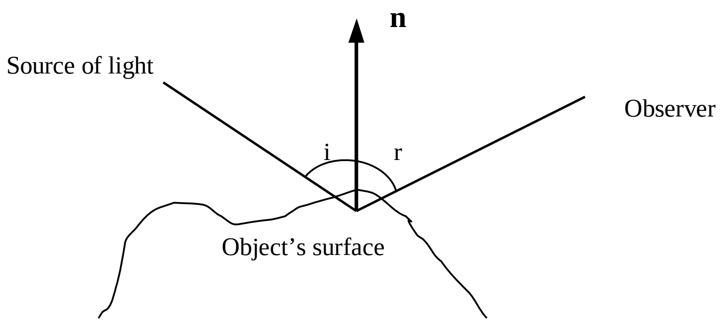


Figure A.10. Depth from shading. The elementary geometry.

Let us assume that the object's surface is described by the function $z(x,y)$. The *surface gradient* is given by the pair (p,q), where $p = \partial z/\partial x$, $q = \partial z/\partial y$. The surface gradient and the space orientation is expressed in Figure A.10 by the vector n which is the normal vector to the surface. The relationship between the image and the surface gradient is expressed by the image radiance equation $I(x,y) = R(p,q)$, where $I(x,y)$ is the intensity function derived from an image and $R(p,q)$ is the reflectance function. Using this relationship the function $z(x,y)$ can be determined, this describes the shape of the object when the reflectance function and the lighting model are known perfectly. A typical example is illustrated in Figure A.11, where the depth map is computed from a synthetic generated shaded image with the light source at an azimuth of 270 degrees and a zenith angle of 30 degrees.

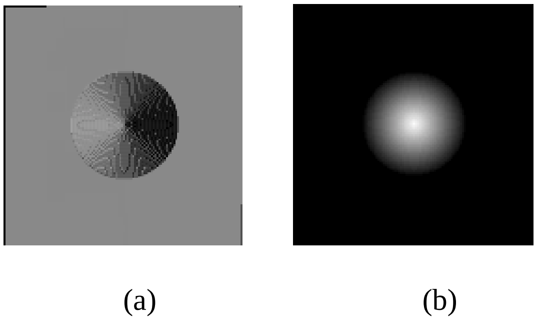


Figure A.11. A shape from shading example. (a) A synthetic shaded image. (b) The corresponding depth map[18].

Jones and Taylor (1994) developed a gradient-based algorithm, which returns a relatively precise depth map even in the presence of noise, but unfortunately for a single image their approach requires 20 minutes on a Sparc workstation.

In conclusion, shape from shading is computationally intensive and also requires *a priori* information regarding the reflectance function and lighting model and this method performs only modestly in comparison to other 3-D techniques when it is applied to objects with complex shapes. In addition, its applicability is constrained to Lambertian surfaces[19] with an albedo (defines the proportion of light radiated by the surface in question) similar over all the object and background.

[18] These images were obtained from: http://www.psrw.com/~markc/Articles/SFS/sfs.html.
[19] This is a surface with no specular properties.

A.7 Depth from focus

Depth from focus (DFF) means estimating the depth of the scene by taking multiple images when the focal level is modified in small increments. Since a lens has finite depth of the field, as a result only the objects placed at the correct distance are in focus. Others are blurred in relation to the focal error. A number of researchers called this ranging technique *auto-focus*.

This method has evolved as both a *passive* and an *active* sensing strategy. For passive focus the focal information is directly employed to calculate the range, whilst in active focus it is used to maximise the sharpness of the image. The active method is used widely in compact auto-focusing cameras and compact video cameras.

In principle, active auto-focus applies range detection with infrared or sonar sensors and subsequently the lens position is computed in direct correspondence with the distance acquired from the sensor. Passive focusing methods are used in SLR cameras and some compact cameras and consists of a stereo matching or a split prism method. It has been found that the passive focusing method is more precise, but this solution is less cost effective and it is generally more appropriate for static images.

A fundamental question is: how the focus level could be measured? To answer this, Tenenbaum (1970) proposed the Tenengrad criterion as a solution to estimate the focus level. This method will be briefly discussed in the next section. More details could be found in the papers by Horri (1992) and Xiong and Shafer (1993) where this technique was applied to both static and dynamic scenes.

A.7.1 The Tenengrad value

The Tenengrad criterion gives an estimation of the gradient $\nabla I(x,y)$ at each point by summing all magnitudes greater than a threshold value.

$$\sum \left|\nabla I(x,y)\right| = \sum S(x,y) = \sum \sqrt{I_x^2 + I_y^2} = \sum \sqrt{\left[i_x * I(x,y)\right]^2 + \left[i_y * I(x,y)\right]^2} \quad \text{(A.4)}$$

The best approximation for I_x and I_y is given by the Sobel operator with its specific kernels:

$$i_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad and \quad i_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

The next operation consists of calculating the Tenengrad value (*TN*) using the relationship illustrated in Equation A.5.

$$TN = \sum_{x=1}^{N} \sum_{y=1}^{N} S^2(x,y) \qquad for \qquad S(x,y) \geq T \tag{A.5}$$

where *N* is the dimension of the image and *T* is a threshold value set in direct correlation with the amount of noise in the image. The goal of the Tenengrad criterion is to maximise the *TN* values. This approach is appropriate for static images. When dealing with dynamic scenes, a stabilisation technique has to be applied in order to avoid the errors caused by local maxima during the focusing process. Because this criterion maximises the high frequency content, consequently this method is sensitive to noise and the aim of stabilisation is to obtain a smooth focussing curve without prominent local maxima. A solution to this problem was suggested in the Xiong and Shafer’s (1993) paper where they proposed a technique to detect the peak of the focus profile accurately using the Fibonacci search method. This approach proved to be effective for systems that have motor-driven lens with high motor resolution.

A.7.2 Grey level variance

Aside from the Tenengrad criterion, other approaches are possible such as *grey level variance* and *sum of modulus difference*. The grey level variance method is more intuitive where the level of focus is associated with the variance of the brightness in the image (i.e. grey level distribution). In this way, if the variance is high the image should be in focus, while the image is out of focus then the variance should be low. The variance σ is computed using the relationship presented in Equation A.6.

$$\sigma = \sqrt{\frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} \left[I(x,y) - \mu \right]^2} \tag{A.6}$$

where μ is the average of the grey level distribution. The aim of this criterion is to maximise the value of σ. This method performs badly for passive systems and furthermore its results are non-linear. In spite of these problems, this approach can be used with good results for active systems.

A.7.3 The sum of modulus difference

Jarvis (1983) proposed the *sum of modulus difference* (SMD) *criterion*. This method is based on the observation that if the image is in sharp focus, then the differences between neighbouring pixels are significant. When the image is blurred the value of pixels tends to be the same. The SMD value is computed for the *x* and *y* axes along a scan line.

$$SMD_x = \sum_{x=1}^{N}\sum_{y=1}^{N}\left|I(x,y) - I(x-1,y)\right| \tag{A.7}$$

$$SMD_y = \sum_{x=1}^{N}\sum_{y=1}^{N}\left|I(x,y) - I(x,y-1)\right| \tag{A.8}$$

Considering the sum $SMD = SMD_x + SMD_y$, to focus an image it is necessary to maximise the value of the *SMD*. Although simple, this technique performs well for both active and passive systems.

Appendix B – Popular 2-D object description techniques

B.1 Template matching

Template matching is a well known technique for shape recognition. This method involves moving a template across an image until a perfect match is found. The principle of template matching is shown in Figure B.1.

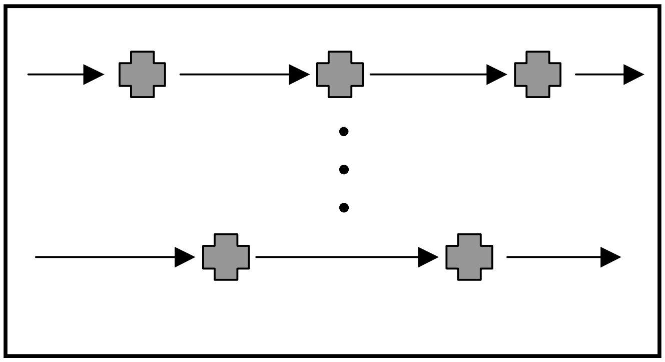


Figure B.1. Template matching algorithm.

The recognition process is slow and consequently will not find a match on the image if the template being searched for is rotated or the size is different. This method performs a very crude recognition. Improvements such as the creation of a large database where a template has different sizes and positions are possible. Unfortunately, using large databases significantly increases the processing time required to perform the recognition process (Vernon, 1991). Also, due to its simplicity, this recognition technique is suitable only for certain applications where the size and the position of the objects in the scene are known precisely.

B.2 Chain coding

Chain coding (also called Freeman coding) is a very popular method used in shape recognition. The aim of this method is to split an image into contours as it is scanned from the top to bottom and left to right in a raster scan manner. The resulting chains are described by a sequence of symbols (codes) and can be employed as primitives in the recognition process. The contours associated with the image in question are extracted progressively using a two-step strategy. The first step involves finding the location of the next contour while the second consists of extracting the chain code. For this purpose, it is necessary to settle the neighbourhood connectivity. Four (a) or eight (b) possible neighbourhood directions can be used in the extraction of the chain codes; this is illustrated in Figure B.2.

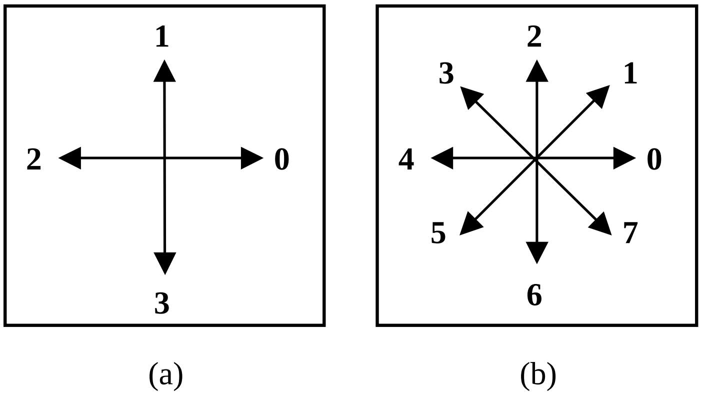


Figure B.2. Diagram showing the pixel neighbourhood. (a) 4 and (b) 8-connectivity.

Very often the chain codes are extracted from the image by following the edge structures (Freeman, 1961 and Gonzales and Wintz, 1987). In this case, the contour is defined by the co-ordinates of the starting point and the sequence of symbols that is obtained after following the edges that are associated with the objects in agreement with the adopted neighbourhood connectivity. The principle of the chain coding method is illustrated in Figure B.3.

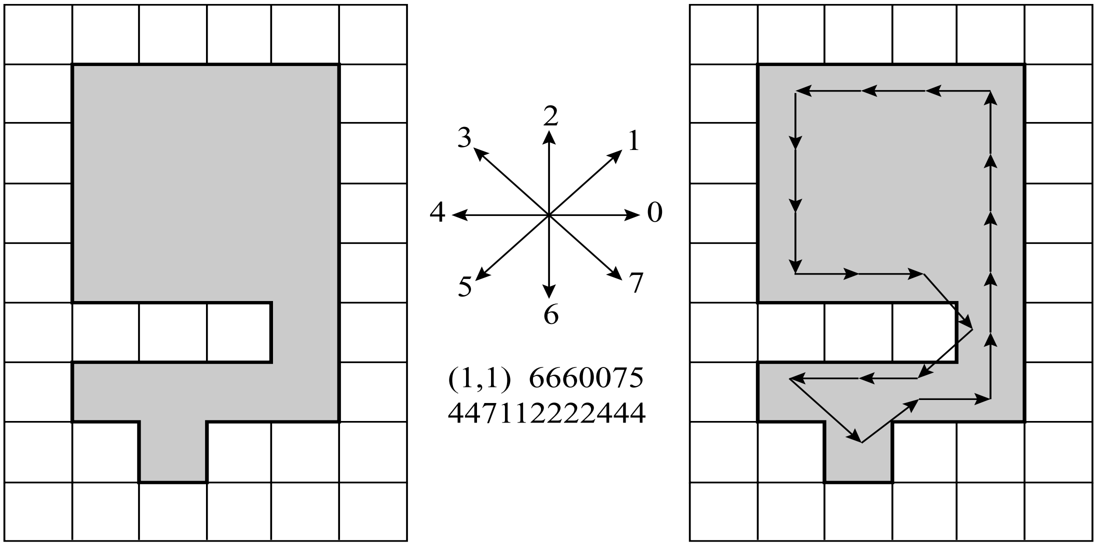


Figure B.3. A simple example for chain coding.

Chain coding has the same disadvantages regarding rotations and scale modification for objects as template matching. A disadvantage is the fact that the perimeters for large objects in the image, for example in a 256 x 256 resolution, the chains can exceed 700 pixels in length and this information is too large to be feasible in real-time object recognition. Furthermore, the resulting chain codes are dependent on the starting point and this uncertainty can arise as a major problem for recognition process.

B.3 Boundary length

The boundary length is a more detailed version of the chain coding method. Vertical and horizontal steps of the boundary are unity in length, the length of the diagonal in eight-connectivity is $\sqrt{2}$ units in length. A closed boundary (perimeter) can be described by the curvature that is the angle between a fixed line and the tangent to the surface at that point. The principle of this method is illustrated in Figure B.4.

For closed boundaries, the angle β is a function which can vary from 0 to 2π. The accuracy of this method is limited and the rotation and changes in scale for objects imply a complex algorithm that performs slowly.

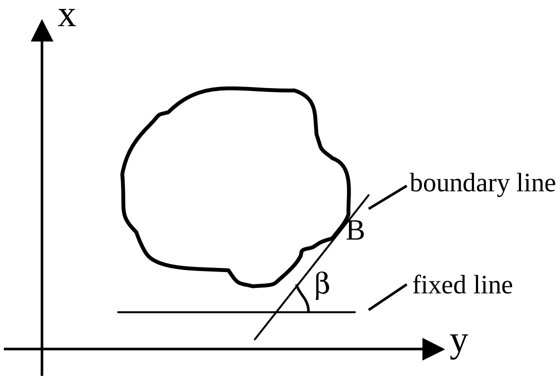


Figure B.4. Boundary length principle.

A closely related method is *bending energy*; bending energy is seen as the energy necessary to bend a rod to the desired shape. This energy can be computed as a sum of squares of the border curvature *c(i)* over the border length.

$$BE = \frac{1}{L}\sum_{i=1}^{L} c^2(i) \tag{B.1}$$

Both methods are feasible for object recognition but their applicability is restricted to non-occluded scenes where the contours can be precisely extracted. In addition as for previous method, for large objects the resulting information is not suitable for a real-time implementation.

B.4 Hough Transform

Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features to be specified in some parametric form (i.e. analytical equation), the *Classical Hough Transform* (CHT) is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. For example, the analytical equation of a line is: $y = ax+b$. A unique straight line can be represented by the starting point $A(x_1, y_1)$ accompanied with the parameters *a* and *b*. This problem can be viewed as a transformation from (*x, y*) space into (*a, b*) parameter space. The algorithm that implements the Hough transform consists of a search for each point in the image in all possible direction by sampling a limited number of parameters *a* and *b*. In other words, each different line through the point $A(x_1, y_1)$ correspond to one of the points on the line in the (a, b) space.

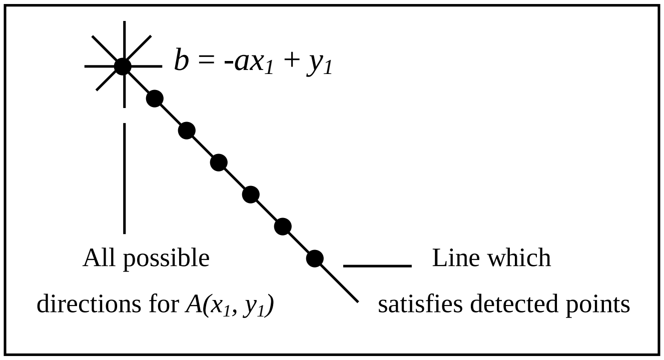

$b = -ax_1 + y_1$
All possible directions for $A(x_1, y_1)$
Line which satisfies detected points

Figure B.5. The process of line detection using the Hough transform.

Alternatively, all points which lie on the same line in (x, y) space are represented by lines which all pass through a single point in (a, b) space. This process is shown in Figure B.5.

A line can be also represented as $s = x\,cos(\alpha) + y\,sin(\alpha)$, where *s* is the distance from the origin of *x*, *y* axes and α is the angle between abscise and the line that connects the origin with the test point (see Figure B.6). In this case, the space (x, y) is transformed into (s, α) space (also referred to as polar space).

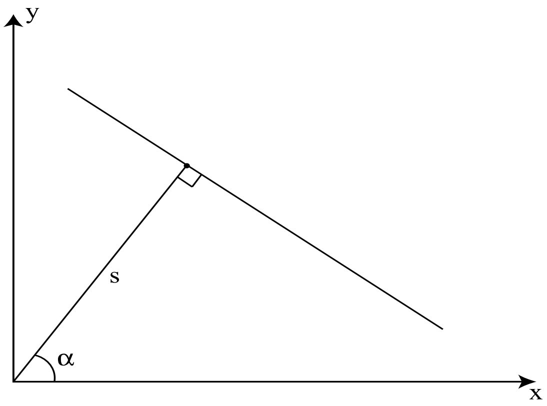


Figure B.6. Hough transform in (s,α) space.

It should be noticed that the previous line representation has the limitation that $a \rightarrow \infty$, the same situation does not appear when the line is represented using polar co-ordinates.

The Hough transform can be generalised to detect other features with an analytical description such as circles or ellipses (Ballard, 1981; Degunst, 1990; McDonald and Vernon, 1998). For instance, in the case of circles, the analytical equation is $(x_1\text{-}a)^2 + (y_1\text{-}b)^2 = r^2$, where (a, b) are the co-ordinates of the centre and r is the radius. Because the computational overhead is in line with the number of parameters contained in the analytical description (three for a circle), the processing time required by the algorithm to detect circles is significantly higher than that required for line extraction.

Arbitrary shapes can be rarely described using a parametric curve. In this case, the *Generalised Hough Transform* (GHT) can offer a solution to the problem. The GHT uses an internal representation when an arbitrary line is constructed by joining a predefined reference point (x_{ref}, y_{ref}) with each boundary point.

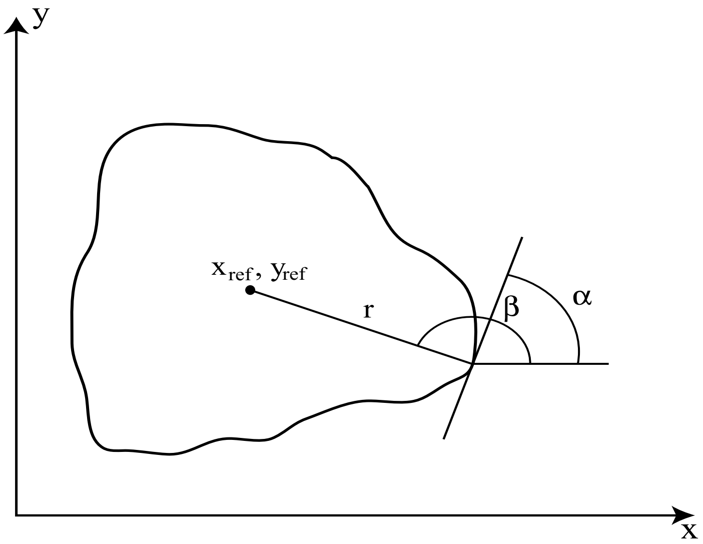


Figure B.7. Principle of the generalised Hough transform.

As illustrated in Figure B.7, the distance *r*, the orientation of the tangent line α and the angle β between the vector *r* and *x* axis are recorded for each boundary point. The resulting table (R-Table) can be ordered according to the orientation of the tangent line. An example of an R-Table is depicted in Figure B.8.

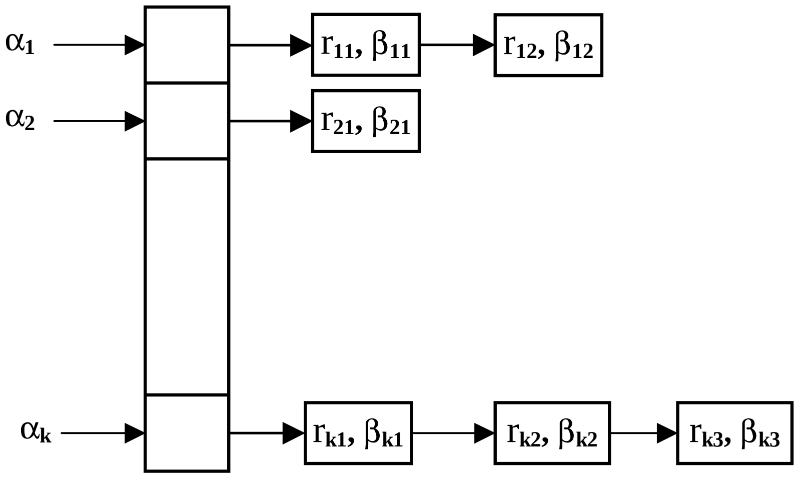


Figure B.8. The GHT R-Table.

The pair (*r*, β) for each boundary point can be computed using the following equations:

$$r_i = \sqrt{(x_{ref} - x_i)^2 + (y_{ref} - y_i)^2} \tag{B.2}$$

$$\beta_i = \tan^{-1}\left(\frac{y_{ref} - y_i}{x_{ref} - x_i}\right) \tag{B.3}$$

As can be seen in Figure B.8 there may be more than a pair (*r*, β) for each α, a typical example being the objects with 'S' type shapes. The Hough Transform is not invariant to rotation or scaling of the object. A solution to these problems is to extend the GHT from two to four parameters when the parameters for scale *s* and rotation ϕ are added.

The Hough transform is a very powerful tool in feature extraction. The vision literature indicates that many applications ranging from manufactured parts to medical imagery have been successfully approached using this technique. Its main advantage is that it is insensitive to gaps in the feature boundary description and is relatively unaffected by image noise. This approach is appealing especially in cases when the scene contains objects with a *known shape* and *size*. Nevertheless, this issue may hinder somehow the attractiveness of this feature extraction technique. In addition, the algorithm that implements the Hough transform requires a lot of storage and extensive computation, a fact which restricts the applicability of this approach to real-time systems.

B.5 Object recognition using invariant shape descriptors

As stated in Forsyth *et al* (1991), attempting object recognition using a single perspective view it is a difficult task and cannot be achieved without the use of shape descriptors derived from a geometrical description of the objects under various perspective directions. Thus, the object recognition relies on a model-driven approach in which a library of geometrical models is used to determine if the scene contains any objects of interest. Nevertheless, to be successful the information regarding object models has to be invariant to perspective projection. This observation introduced the necessity to use *invariant shape descriptors* in the recognition process because they are insensitive to perspective projection or object pose.

At this stage deciding which descriptors associated with the object's shape are viewpoint independent presents a major problem. This is not a trivial problem and a discussion on this topic is detailed in the paper by Zisserman *et al* (1994).

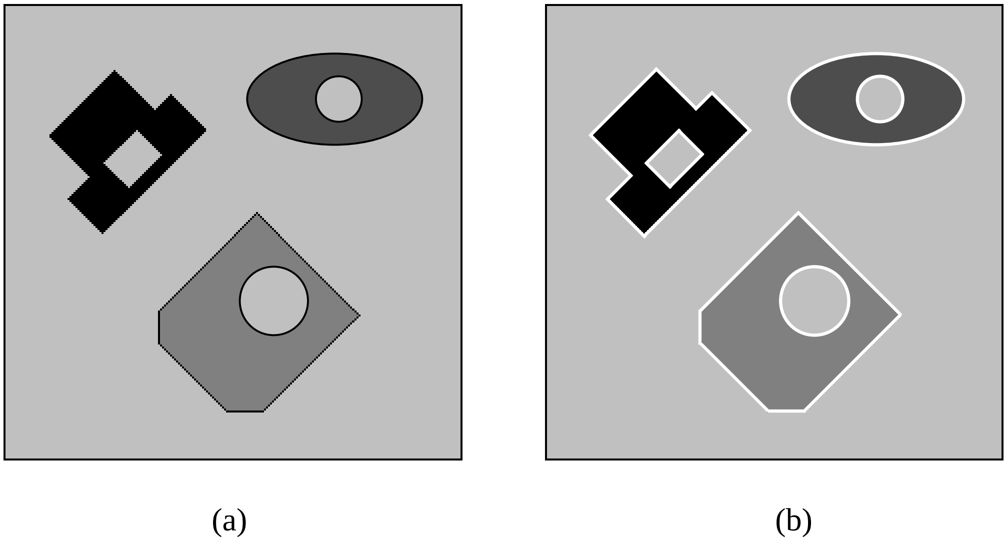


Figure B.9. Object recognition using shape invariants. (a) Input image containing simple planar objects and (b) the resulting image after the line and conic detection.

After extensive investigation, they concluded that the lines and conics are very stable invariants suitable to describe a relatively large range of at least planar objects. Because these algebraic invariants are described by analytical equations, they can be easily detected using the edge information or the Hough transform. In addition, they can be robustly recovered even in cases when they are not completely described. Figure B.9 illustrates the detection of the shape descriptors for a scene which contains only simple planar objects.

The remaining problem deals with matching the scene invariants with those associated with a model from the database. The resulting invariants derived from the scene are used to generate hypotheses required to match an object model. There is no doubt that for a complex scene the number of possible hypotheses is extremely large. Therefore, in order to reduce the computational burden only the hypotheses which are *geometrically compatible* are used. In other words, a hypothesis is created only if some rules concerning the type, the number and topology between invariants are upheld. For example, if an object consists of an elliptical plate with a circular hole in the middle, the only hypothesis that is accepted consists of two conics in which one is included in other. As a result, when this grouping mechanism is applied the number of plausible hypothesis is dramatically reduced. Once a model is matched, the last stage deals with pose determination using the *back projection* of a conic pair (for more details refer to Forsyth *et al*, 1991).

This formulation is extremely powerful when dealing with regular planar objects. If the objects of interest are described by non-algebraic curves, it is necessary to extract the *local* invariants which measure the curvature or the torsion derived from the shape. In addition, when this formulation was applied to 3-D objects the results were by far not as impressive as those reported when planar objects were considered.

Appendix C – Popular 3-D object description techniques

C.1 3-D chain coding

The idea of this description technique is to represent the shape of an object using a sequence of symbols (codes) obtained by scanning the surfaces in a raster scan manner. For this approach the chain code extraction is performed on range images, in contrast with the 2-D case when a row image is analysed. Figure C.1 illustrates the resulting chain code after a simple spatial representation is analysed.

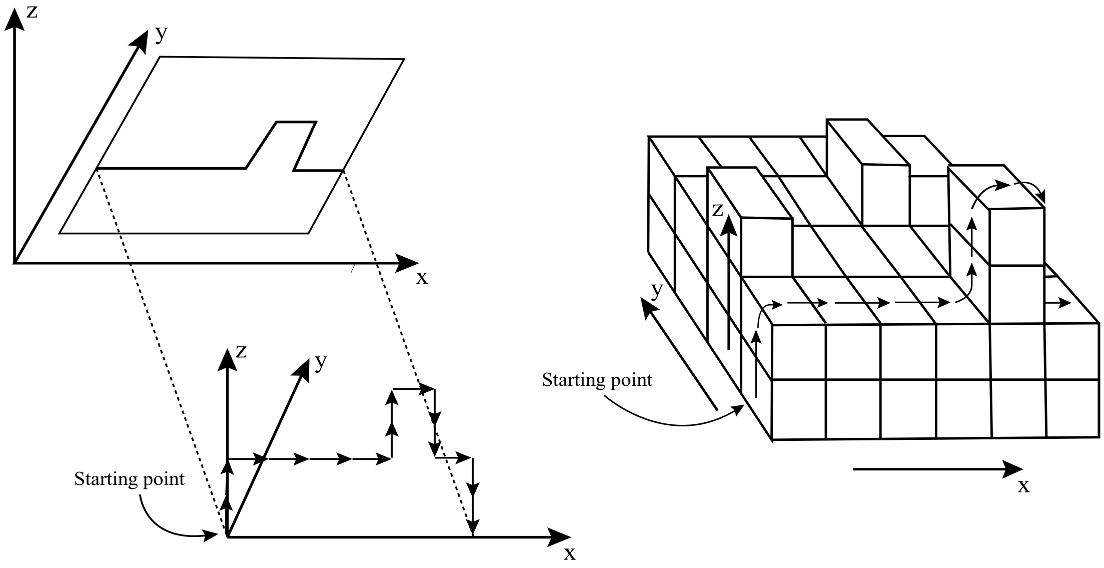


Figure C.1. An example for a 3-D connected chain code.

This method performs a very crude 3-D shape description and the main disadvantage is its sensitivity to rotations and scale modification of the object. Also, as for the 2-D case, the resulting chain codes are dependent on the starting point and for large objects the resulting information is too large to be feasible for real-time object recognition.

C.2 Extended Gaussian Image (EGI) representation

In contrast with the previous approach, the EGI is an example of the *global* representation of the 3-D space. This representation was proposed by Horn (1979) and consists of mapping the surface normals of an object onto the Gaussian sphere[20]. The surface normals for each point of the object are placed so that their tails lie at the centre of the Gaussian sphere while the heads lie on a point on the sphere according to the particular surface orientation.

This representation is further extended when a *weight* is attached to each point on the surface where a normal is erected. This weight value is proportional with the area of the surface given to the normal. The result is a distribution of weights over the Gaussian sphere and is called the *Extended Gaussian Image* (EGI). Alternatively, the EGI of an object can be thought as a spatial histogram of its surface normals. Figure C.2 illustrates the EGI of a cylindrical object.

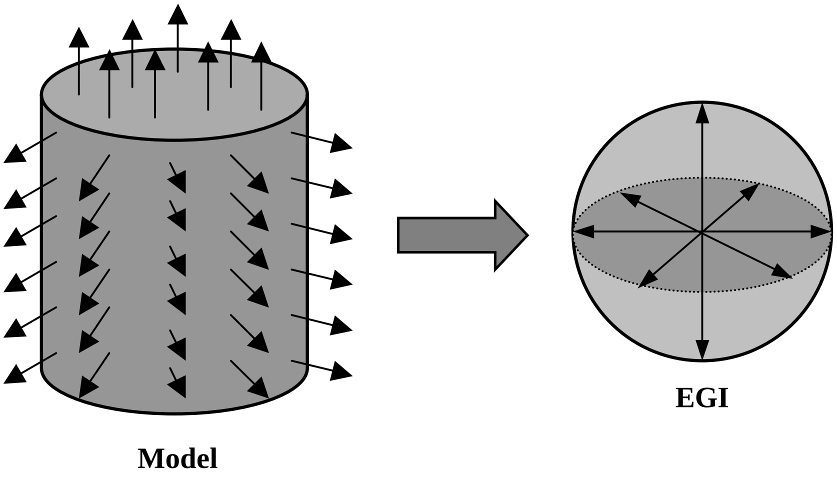


Figure C.2. The normals associated with a cylinder model and its corresponding EGI representation (modified from Ikeuchi, 1983).

Usually, the appearance of an object varies with the following factors: translation, size and rotation. It can be easily noticed that the EGI representation is independent of

[20] A detailed presentation of this concept is also available in Horn’s (1983) paper.

translation and size of the object. Theoretically, the EGI rotates in the same way as the object. But due to self-occlusion the EGI can be defined only for the visible hemisphere where its pole corresponds to the line of sights. However, this is not a major drawback as long as a *part* of the model's EGI can match the *attitude* of the object. Thus, this representation allows determination of the identity and orientation of the object at the same time. There is no doubt that a search for all possible attitudes is not a realistic approach (especially if the model database contains a large number of objects). In order to reduce the searching space Ikeuchi (1983) proposed to employ two constraints. The first is the EGI mass center which is used to constrain the line of sight. The use of this constraint is suggested by the obvious observation that the mass center is different for different visible hemisphere. The second constraint consists of minimising the EGI inertia direction in order to constrain the rotation around the line of sight when dealing with rotationally symmetric EGI distributions. The application of these constraints greatly reduces the search space and the model that maximises the fitting measure is chosen as the matched model.

Although very powerful this representation has some limitations such as its sensitivity to mutual occlusion when dealing with cluttered scenes and the fact that assures a unique representation only for convex objects. Figure C.3 illustrates three objects with the same EGI.

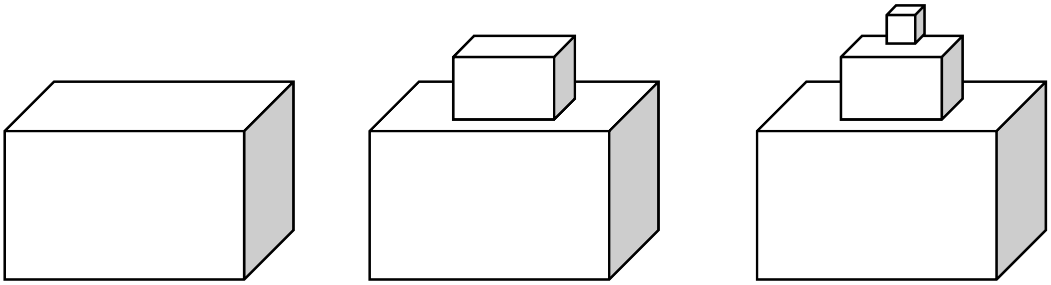
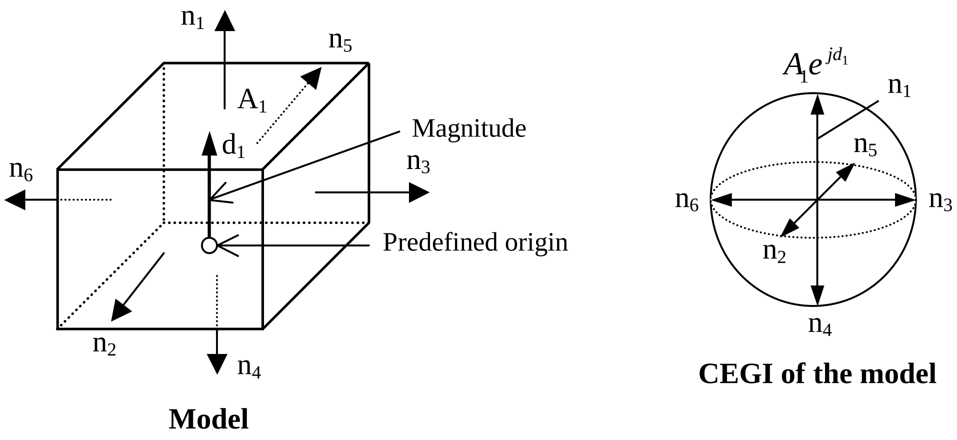


Figure C.3. Example of three objects having an identical EGI.

In the early 90's, Kang and Ikeuchi (1990) addressed the deficiency associated with the EGI representation, namely the inability to give a unique representation for non-convex objects. In this sense, they proposed the *Complex EGI* (CEGI) concept in which the weights associated with the points of the surfaces where the normals are erected are in this case complex numbers. As in the case of standard EGI, the *magnitude* is proportional to the area of the surface given to a normal vector while the *phase* is the distance from a predefined origin to the face to be analysed in the direction of the normal. The principle of this concept is illustrated in Figure C.4.



Model

CEGI of the model

Figure C.4. The CEGI representation for a cube (modified from Kang and Ikeuchi, 1990). For clarity, the weight is shown only for normal n_1.

Because CEGI encodes the object’s faces positions, the objects with similar EGI depicted in Figure C3 have different CEGI representations. This representational scheme is very convenient when dealing with single-object scenes, but due to its sensitivity to occlusion its application to multi-object scenes would be difficult.

C.3 Object representation using scene features

The goal of this approach is to extract a relatively small number of features in order to determine the possible model identities and poses for the scene object. In the paper by Kak and Edwards (1995) seven features are considered to be sufficient to represent a large range of objects. These features are: points, straight lines, elliptical curves, planar, cylindrical, conical and spherical surfaces. At this point a very important decision is how the resulting data is organised. From the literature survey presented in Chapter 1 it is clear that a proper model and data representation are keys to designing a computationally efficient object recognition system. Among other representational schemes, the *feature spheres* are powerful data structures used to reduce the computational burden in the verification stage.

The main idea of this concept relies on using the *principal direction* associated with the feature set. Basically, the principal direction Φ represents the characteristic position or orientation of a feature (point, line and surface) and is expressed by a unit vector in space. For example, the principal feature of a planar is the normal to that surface; for a cylindrical surface, it is the axis direction; and for a point, it is the normalised vector at the position of that point. To exemplify this representation, the rendering of a model object with respect to the principal direction is illustrated in Figure C.5. It should be noticed that the principal direction is defined only with respect to an object centered co-ordinates system.

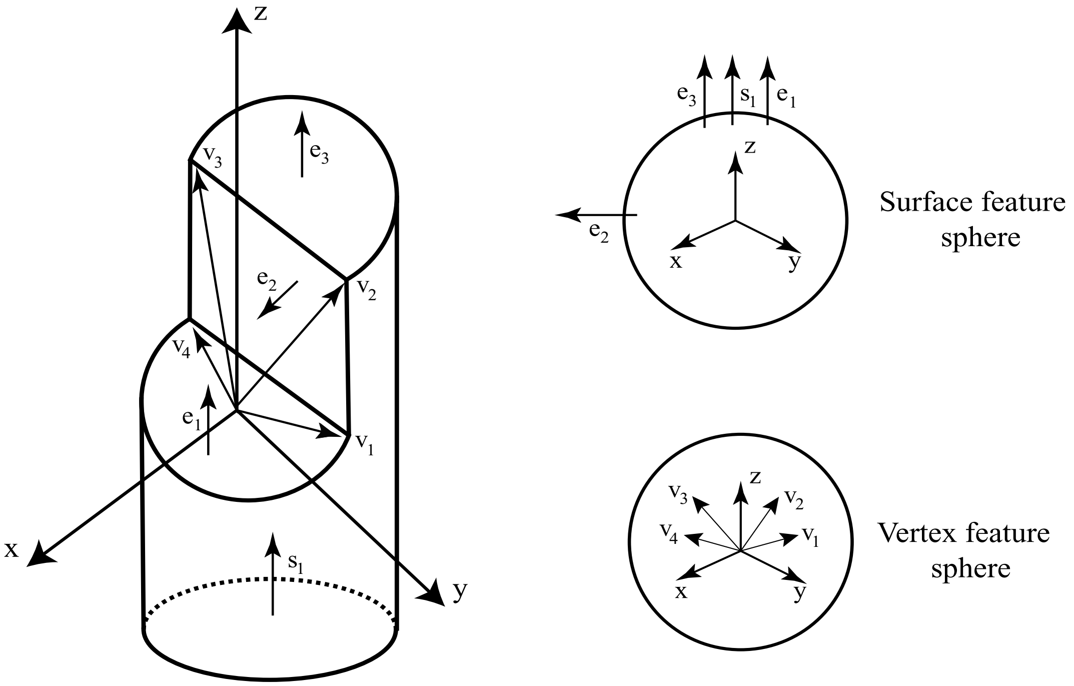


Figure C.5. A model object and the principal directions of its surfaces (modified from Kak and Edwards, 1995). With *e* is represented the principal direction (normal) of a plane, *v* is the normalised position of a point and *s* is the axis direction of a cylinder.

The next step consists of generating the pose transform hypotheses. In the same paper, two possible approaches suitable for organising the feature data for efficient hypotheses generation are presented.

The first approach uses *local feature sets* while the second relies on using *multiple attribute hash tables*[21]. Both of these techniques generate the pose transforms using three scene features employed to match the model features. Nevertheless, an inherent problem is how to select the most relevant three features associated with a scene object. In their experiments it was found that the shape attributes are ineffective for constraining the hypotheses generation stage. The main rationale that explains this situation is the fact that the principal directions associated with the feature set offer very little information to distinguish between different pose transformations. Thus, in the authors' opinion, the use of *relational* attributes between the scene features is more appropriate. To further restrict the computational burden associated with this stage only a set of adjacent scene features are used to generate pose hypotheses.

Once these hypotheses are formed, the recognition algorithm attempts to match the detected features with the model features. As would be expected, the precision of this technique is in line with the number of poses captured for the object model during the off-line model building procedure.

Although conceptually attractive, this approach raises many problems during its implementation. Kak and Edwards (1995) reported for their developed system a failure rate of 30 percent when the objects of interest contain only regular shapes. The main problems identified by the authors are associated with an improper surface segmentation scheme and an imprecise range sensor. In spite of these problems, the proposed formulation shows a lot of potential and at least in cases when dealing with simple geometrical objects with a homogenous surface appearance this approach is valid.

[21] For more details regarding the implementational issues associated with these approaches the reader can consult the paper of Kak and Edwards (1995).

Appendix D – A note on metric and feature normalisation

D.1 A discussion on distance metrics

Throughout our discussions, we must fix a *space* $\Re$ of points in which to work. The space may be the plane $\Re^2$ when described by two co-ordinates or any space $\Re^n$ the case when a distinct position in space is represented by an *n*-dimensional vector. Independent of space dimension, the distance is used in pattern recognition to evaluate the *closeness* between two *patterns* (vectors). Let's supose that our aim is to estimate the proximity between two *n*-dimensional patterns *x* and *y*, where *n* is the dimension of space. For this purpose, a metric which satisfies the conditions (often referred to as the *triangle inequality*) illustrated in Equation D.1 has to be settled.

$$\begin{gathered} d(x,y) = 0 \quad \text{only if} \quad x = y \\ d(x,z) \le d(x,y) + d(y,z) \quad \text{for all} \ (x,y,z) \end{gathered} \tag{D.1}$$

The proximity between these vectors can be estimated in several ways, but very often for this purpose the Minkovski metric is employed. The Minkovski metric is defined in Equation D.2.

$$d(x,y) = \left(\sum_{i=1}^{n} \left| x[i] - y[i] \right|^{r} \right)^{1/r} \quad \text{where} \quad r \ge 1 \tag{D.2}$$

It can be easily observed that the Minkovski metric obeys the metric conditions illustrated above independent of parameter *r*. Nevertheless, according to the relationship illustrated in Equation D.2 there are an infinite number of metrics but the most common are defined for *r* = 1 (Manhattan or taxicab distance), *r* = 2 (Euclidean distance) and

$r = \infty$ (Sup or Chessboard distance). The relationships for these metrics (distances) are depicted below.

$$\text{if } r = 1 \text{ (Manhattan distance)} \qquad d(x,y) = \sum_{i=1}^{n} \left| x[i] - y[i] \right| \qquad \text{(D.3)}$$

$$\text{if } r = 2 \text{ (Euclidean distance)} \qquad d(x,y) = \sqrt{\sum_{i=1}^{n} \left(\left| x[i] - y[i] \right| \right)^2} \qquad \text{(D.4)}$$

$$\text{if r} = \infty \text{ (Chessboard distance)} \qquad d(x,y) = \max_{i} \left(\left| x[i] - y[i] \right| \right) \qquad \text{(D.5)}$$

A simple numerical example is illustrated in Figure D.1.

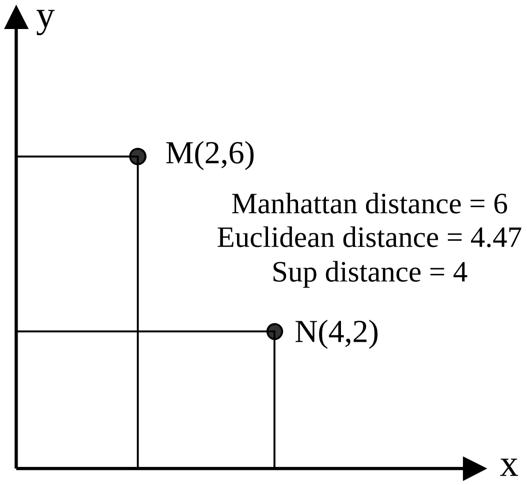


Figure D.1. A numerical example for three Minkovski metrics (from Jain and Dubes, 1988).

It can be observed from this simple example that contours of constant Manhattan distance describe a square (or hypercube for multi-dimensional cases) while for the Euclidean distance the contours are described by circles (or spheres for multi-dimensional cases). This geometrical approach shows that the Manhattan distance is not invariant to rotation. This issue is very inconvenient for a range of applications and this is the reason why the Euclidean distance is commonly employed. However, the Euclidean distance has some limitations. The first is its inability to cope with the scaling of the co-ordinate axes. In addition, for a minimum Euclidean distance classifier the resulting boundaries after the partition of the feature space are linear, a situation that is not always satisfactory.

These limitations are significantly alleviated when the *Mahalanobis* metric is considered. This distance incorporates the correlation between the features contained in a class and standardises each feature to zero mean and unit variance. The expression of Mahalanobis distance is illustrated in Equation D.6.

$$d = (x - m_i)^T C_i^{-1} (x - m_i) \qquad \text{(D.6)}$$

where x is the feature vector, m_i is the mean vector of class i, T denotes the vector transpose and C_i^{-1} is the inverse of the covariance matrix for class i. It has been shown that the contour in which d is constant describes an ellipse in the planar case or an ellipsoid when multi-dimensional vectors are considered. If the features are uncorrelated the covariance matrix is the identity matrix and as a result the Mahalanobis metric becomes the same as the Euclidean metric.

After these metrics are introduced a natural question is: which is better suited to our application? The answer is very simple and is dictated by the feature distribution on the feature space. If the features are uncorrelated and the discrimination surfaces are not very curved, there is no reason to use anything more than the Euclidean distance. In contrast, if the features are highly correlated the Euclidean metric may not be an appropriate solution and in this case the Mahalanobis distance is the best option. It is worth mentioning that the computational burden associated with Mahalanobis distance is much higher than the burden required by Euclidean distance.

D.2 A note on feature normalisation

As mentioned in the previous section the Mahalanobis metric performs an implicit normalisation but is not widely used because it is computationally inefficient. Therefore, commonly the Euclidean metric is used to evaluate the level of similarity in pattern recognition.

In practice, very often the features contained in the pattern vector have very different ranges. Let's imagine a simple situation where a pattern vector has only two features where the first describes the perimeter of a region while the second represents the area. In contrast with the perimeter which grows linearly with scale, the area grows quadratically and as a result the small feature is overpowered by the large feature when the proximity between two patterns is evaluated. Therefore, it is necessary to apply a normalisation scheme in order to compensate for this issue. The literature on clustering indicates that this operation can be performed in various ways. For example, one type of normalisation includes only range scaling. Others take into account the feature mean and a simple normalisation consists of subtracting the mean from each feature as illustrated in Equation D.8.

$$m_i = \frac{\sum_{j=1}^{k} x_j[i]}{k} \tag{D.7}$$

$$X_j[i] = x_j[i] - m_i \qquad \text{for } j = 1,\ldots,k \tag{D.8}$$

where m_i is the mean of the i^{th} feature, x_j is the unprocessed j^{th} pattern, k is the number of patterns and X_j is the normalised j^{th} pattern. This type of normalisation makes the features invariant to displacements of the co-ordinates. Commonly it is required that all features have zero mean and unit variance. To achieve this requirement it is necessary not only to subtract the mean from each feature but also to divide the result by the feature variance. The feature variance can be computed using the relationship presented in Equation D.9 and the normalisation is illustrated in Equation D.10.

$$s_i = \sqrt{\frac{\sum_{j=1}^{k} (x_j[i] - m_i)^2}{k}} \tag{D.9}$$

$$X_j[i] = \frac{x_j[i] - m_i}{s_i} \qquad \text{for } j = 1,\ldots,k \tag{D.10}$$

where s$_i$ is the variance of the i^{th} feature.

As stated in Jain and Dubes (1988), it is very important to mention that the effect of normalisation is not always positive. For example, the normalisation illustrated in Equation D.8 can change the distances between patterns and at the same time can alter the separation between natural clusters. Therefore, the type of normalisation has to be carefully chosen and practice has demonstrated that a successful scheme must be closely related to the context of the problem being evaluated.

Appendix E – A note on eigensystems[22]

E.1 Introduction

There are many instances in mathematics and physics in which only the vectors which are left essentially unchanged by the operation of the matrix are of interest. Specifically, we are interested in those vectors *x* for which $A \cdot x = \lambda \cdot x$ where *A* is a square *n* by *n* matrix and λ is a real number. A vector *x* (other than zero) for which this equation holds is called an eigenvector[23] of the matrix *A* and the associated constant λ is called the eigenvalue (or characteristic value) of the vector *x*. Obviously, λ is an eigenvalue of *A* if:

$$\det\left|A - \lambda I\right| = 0 \tag{E.1}$$

where *det* denotes the determinant of a square matrix and *I* is the *n* by *n* identity matrix. If this expression is expanded, the result is an *n*-order polynomial (also called characteristic polynomial) in λ whose roots are the eigenvalues. Nevertheless, there are situations when an eigenvalue is listed more than once, that means it is a multiple root of the characteristic polynomial. In these cases, the eigenvalues are called *degenerate*. Next, using the relationship $A \cdot x = \lambda \cdot x$ the corresponding eigenvectors are computed.

This section is focused only on the computation of the eigenvalues and the corresponding eigenvectors derived from the real matrices. For these matrices (except in rare cases) the resulting eigenvectors *do not* form an orthonormal vector space. In addition, if the characteristic polynomial has multiple roots, the resulting eigenvectors are not complete and the vector space is said to be defective. Fortunately, there is a very important class of matrices, namely the symmetric matrices, that have some interesting properties which are listed as follows:

[22] This section is mostly based on the book “Numerical Recipes in C” by Press *et al* (1992).

[23] Originally, eigen is a German word and means “self” or “own”.

- A symmetric matrix is identical to its transpose or $a_{ij} = a_{ji}$ for all i and j.
- The roots (eigenvalues) computed from the characteristic polynomial are all real.
- The resulting eigenvectors are mutually orthogonal, thus determine an n dimensional *linearly independent* vector space even in cases when dealing with multiple roots.

Due to its convenient properties, for a symmetric matrix the computational burden associated with the calculation of the eigenvalues and eigenvectors is significantly lower than the burden associated with non-symmetric matrices. Essentially, as stated in Press *et al* (1992) all modern algorithms involve transforming the input matrix A into a simpler special form by using a sequence of *similarity transformations*. A similarity transformation of the matrix A is presented in Equation E.2.

$$A \to Z^{-1} \cdot A \cdot Z \tag{E.2}$$

where Z is the transformation matrix. These transformations play a crucial role in the computation of eigenvalues because they leave the eigenvalues unchanged. This property is illustrated in Equation E.3.

$$\begin{gathered}\det\left|Z^{-1} \cdot A \cdot Z - \lambda \cdot I\right| = \det\left|Z^{-1} \cdot A \cdot Z - Z^{-1} \cdot Z \cdot \lambda \cdot I\right| = \\ \det\left|Z^{-1} \cdot A \cdot Z - Z^{-1} \cdot \lambda \cdot I \cdot Z\right| = \det\left|Z^{-1}(A - \lambda \cdot I) \cdot Z\right| = \\ \det\left|Z^{-1}\right| \cdot \det\left|A - \lambda \cdot I\right| \cdot \det\left|Z\right| = \det\left|A - \lambda \cdot I\right|\end{gathered} \tag{E.3}$$

When dealing with symmetric matrices which is the topic of this section, the eigenvectors are real and orthonormal, thus the transformation matrix is orthogonal. In this case the similarity transformation can be redefined as:

$$A \to Z^{T} \cdot A \cdot Z \tag{E.4}$$

where T defines the vector transpose.

At this stage, depending on the simplified form which is sought, there are two choices. The first requires transforming the symmetric matrix *A* into diagonal form which is the case of Jacobi reduction. The second possibility involves the transformation of the matrix *A* to tridiagonal form using the Householder reduction. Then, the eigenvalues are computed using the QL or the QR decomposition. In the remainder of this section the aforementioned techniques are detailed and some numerical examples are employed in order to evaluate their performance.

E.2 Jacobi transformations of a symmetric matrix[24]

The Jacobi method consists of a sequence of similarity transformations in order to convert the symmetric matrix *A* to diagonal form, where the elements which form the diagonal represent the desired eigenvalues. The main idea of this technique is to use a plane rotation for the purpose of annihilating one of the off-diagonal elements. Although successive transformations undo the previously set zeros, there is no doubt that with each iteration the off-diagonal elements get smaller and smaller and this process stops when the matrix is diagonal to machine precision. The basic Jacobi rotation P_{pq} is a matrix of the following form:

$$P_{pq} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \ddots & \vdots & 0 & \cdots & 0 & \vdots & \iddots & 0 \\ 0 & \cdots & c & 0 & \cdots & 0 & s & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 & \iddots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & 1 & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \iddots & 0 & \ddots & 0 & 0 & 0 \\ 0 & \cdots & -s & 0 & \cdots & 0 & c & \cdots & 0 \\ 0 & \iddots & \vdots & 0 & \cdots & 0 & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \tag{E.5}$$

where all the diagonal elements are unity except for the two elements marked with *c* and all off-diagonal elements are zero except the two elements marked with *s* and –*s*. The

[24] For a more detailed treatment of the algorithms presented in this section the reader can refer to: http://noir.ovpit.indiana.edu/B673/node22.html

numbers *c* and *s* are given by the rotation angle ψ and are defined as: *c* = *cos* ψ and *s* = *sin* ψ. A similarity transformation of matrix *A* according to Equation E.4 can be rewritten as follows:

$$A^{'} = P_{pq}^{T} \cdot A \cdot P_{pq} \tag{E.5}$$

As can be easily verified P_{pq}^{T} changes only the rows marked with *p* and *q* from *A* while P_{pq} affects only the columns *p* and *q*. When the Equation E.5 is expanding out the elements affected by similarity transformation are:

$$\begin{aligned} a_{mp}^{'} &= ca_{mp} - sa_{mq} \\ a_{mq}^{'} &= ca_{mq} + sa_{mp} \\ a_{pp}^{'} &= c^{2}a_{pp} + s^{2}a_{qq} - 2sca_{pq} \\ a_{qq}^{'} &= s^{2}a_{pp} + c^{2}a_{qq} - 2sca_{pq} \\ a_{pq}^{'} &= (c^{2} - s^{2})a_{pq} + sc(a_{pp} - a_{qq}) \end{aligned} \tag{E.6}$$

where $m \neq p,\ \ m \neq q$ and a_{ij} are the elements of matrix *A*. As mentioned earlier, the aim of this method is to annihilate the off-diagonal elements. Thus, to have $a_{pq}{}^{'}$ = 0, the last expression from Equation E.6 gives the expression for the rotation angle ψ.

$$\cot 2\psi = \frac{a_{qq} - a_{pp}}{2a_{pq}} = \frac{c^{2} - s^{2}}{2cs} = \frac{\cos^{2}\psi - \sin^{2}\psi}{2\cos\psi\sin\psi} = \frac{\cos 2\psi}{\sin 2\psi} \tag{E.7}$$

If the notation $z = \frac{s}{c}$ is employed, the Equation E.7 can be rewritten as follows:

$$z^{2} + 2z \cdot \cos 2\psi - 1 = 0 \tag{E.8}$$

$$
\begin{aligned}
z_1 &= -\cot 2\psi + \sqrt{\cot^2 2\psi + 1} = \frac{1}{\cot 2\psi + \sqrt{\cot^2 2\psi + 1}} \\
z_2 &= -\cot 2\psi - \sqrt{\cot^2 2\psi + 1} = -\frac{1}{-\cot 2\psi + \sqrt{\cot^2 2\psi + 1}}
\end{aligned} \qquad \text{(E.9)}
$$

As stated in Press *et al* (1992) the smaller root of Equation E.8 corresponds to a rotation angle smaller than $\frac{\pi}{4}$ and this choice at each stage assures a stable reduction. Therefore, the Equation E.9 can be rewritten as follows:

$$
z = \frac{\operatorname{sgn}(\cot 2\psi)}{\left|\cot 2\psi\right| + \sqrt{\cot^2 2\psi + 1}} \qquad \text{(E.10)}
$$

Once we have *z*, the numerical values for *c* and *s* are obtained by substituting *s* = *zc* in $c^2 + s^2 = 1$.

$$
c = \sqrt{\frac{1}{z^2 + 1}} \quad \textit{and} \quad s = zc \qquad \text{(E.11)}
$$

The only remaining problem is the strategy that should be adopted for the order in which the elements are annihilated. The simplest strategy consists of annihilating the largest off-diagonal elements at each stage. While this strategy is suitable for hand calculation, its algorithmical implementation is computationally inefficient with a complexity of N^2 per iteration. A more efficient strategy is the *cyclic Jacobi method*, where the off-diagonal elements are annihilated in a strict order. Commonly, it proceeds by analysing the matrix in a raster scan manner ($P_{12}, P_{13},\ldots,P_{1n}, P_{23}\ldots$). The convergence associated with the cyclic Jacobi method is generally quadratic when dealing with non-degenerate eigenvalues and can be easily evaluated by computing the sum of the squares of the off-diagonal elements.

The main advantage of this algorithm is its simplicity and it is recommended to be used when dealing with matrices of moderate order (usually smaller than 15). For larger matrices the processing time is significantly larger and more efficient methods have to be considered.

E.3 Reduction of a symmetric matrix to tridiagonal form

As mentioned earlier, the optimal strategy for computing the eigenvalues and eigenvectors relies on converting the input symmetric matrix to a simple form. One option was discussed in the previous section and requires converting the matrix in question to diagonal form. Another preferred form is tridiagonal, a situation where the processing time associated with the computation of the eigenvalues and the corresponding eigenvectors is significantly reduced.

E.3.1 Givens method

The Givens *reduction* consists of a modification of the Jacobi method but instead of attempting to reduce the symmetric matrix to diagonal form, the process stops when the resulting matrix is tridiagonal. To achieve this goal, the rotation angle is chosen in order to zero an element that is not one of the four corners, i.e. a_{pp}, a_{pq}, or a_{qq}. For example, to annihilate a_{31} is chosen P_{23}, to annihilate a_{41} is chosen P_{24} and so on. The sequence of similarity transformations required for reducing the matrix A to tridiagonal form is illustrated below.

$$P_{23}, P_{24}, \ldots, P_{2n}; P_{34}, \ldots, P_{3n}; \ldots P_{n-1n} \qquad \text{(E.12)}$$

where P_{mn} annihilates a_{nm-1}. The implementation of this reduction technique is straightforward but bears the same disadvantage as the Jacobi method namely its computational inefficiency. A more efficient method is the Householder reduction which will be discussed in the next section.

E.3.2 Householder method

The Householder reduction converts an n-order symmetric matrix into tridiagonal form by using n-2 orthogonal transformations. The idea of this method consists of annihilating at each stage the required part of the whole column and whole corresponding row. The basic Householder transformation is a matrix *P* of the following form:

$$P = 1 - 2w \cdot w^T \tag{E.13}$$

where *w* is a real vector with $|w|^2 = 1$. It can be easily demonstrated that the matrix *P* is orthogonal and this can be seen in Equation E.14.

$$\begin{gathered} P^2 = (1 - 2w \cdot w^T) \cdot (1 - 2w \cdot w^T) = \\ 1 - 4w \cdot w^T + 4w \cdot (w^T \cdot w) \cdot w^T = 1 \\ \Rightarrow P = P^{-1} \\ but \quad P = P^T \Rightarrow P^T = P^{-1} \end{gathered} \tag{E.14}$$

Then, *P* can be rewritten as follows:

$$P = 1 - \frac{u \cdot u^T}{H} \tag{E.15}$$

where $u = x \mu |x| \cdot e_1$, *x* is the vector composed of the first column of *A*, *e*$_1$ is the identity vector and *H* is a scalar defined as $H = \frac{1}{2}|u|^2$. Multiplying out the matrix *P* with the vector *x*, as a result the following expression is obtained:

$$\begin{gathered} P \cdot x = x - \frac{u}{H} \cdot (x \mu |x| \cdot e_1)^T \cdot x = \\ x - \frac{2u \cdot (|x|^2 \mu |x| \cdot x_1)}{2|x|^2 \mu 2|x| \cdot x_1} = x - u = \pm |x| \cdot e_1 \end{gathered} \tag{E.16}$$

From the relationship illustrated in Equation E.16, it can be observed that all the elements of the vector *x* except the first one are set to zero. Therefore, to reduce a symmetric matrix *A* to tridiagonal form, the vector *x* is chosen to be the lower *n*-1 elements of the first column. As a result the lower *n*-2 element will be set to zero as can be seen in equation E.17.

$$P_1 \cdot A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & & & & \\ 0 & & P_1^{n-1} & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & & & & \\ a_{31} & & \mathit{Irrelevant} & & \\ \vdots & & & & \\ a_{n1} & & & & \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ k & & & & \\ 0 & & \mathit{Irrelevant} & & \\ 0 & & & & \\ 0 & & & & \end{bmatrix} \tag{E.17}$$

where P_1^{n-1} is the n-1 by n-1 Householder matrix and k is the plus or minus magnitude of the vector [a_{21}, …, $a_{\text{n}1}$]. The complete orthogonal transform is shown in Equation E.18.

$$A^{'} = P_1 \cdot A \cdot P_1^T = \begin{bmatrix} a_{11} & k & 0 & \cdots & 0 \\ k & & & & \\ 0 & & \mathit{Irrelevant} & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix} \tag{E.18}$$

Then, the reduction process continue to $P_{\text{n-2}}$ where the vector x is composed by the bottom n-2 elements of the matrix A. It can be easily verified that this new transformation will not affect the results achieved in the first step. The reduction process continues by applying the remaining transformations: P_3, …, $P_{\text{n-2}}$.

E.4 The QL algorithm

Once the symmetric matrix has been reduced to tridiagonal form, the eigenvalues can be easily computed by applying the QL algorithm. The idea behind the QL algorithm is that any real matrix can be decomposed in the following form:

$$A = Q \cdot L \tag{E.19}$$

where *Q* is an orthogonal matrix and *L* is lower triangular. As seen in the previous section this decomposition is realised by applying the Householder reduction to *A*. Once we have the tridiagonal matrix resulting after the application of Householder reduction, the QL algorithm attempts to convert it to diagonal form. The QL algorithm consists of a sequence of orthogonal transformations and works as follows:

$$\begin{gathered} A = tridiagonal \quad matrix \\ find \quad the \quad decomposition \quad of \quad A \\ generate \quad A_1 = L \cdot Q \\ find \quad the \quad decomposition \quad of \quad A_1 \\ generate \quad A_2 = L_1 \cdot Q_1 \\ find \quad the \quad decomposition \quad of \quad A_2 \\ generate \quad A_3 = L_2 \cdot Q_2 \\ \mathrm{M} \end{gathered} \tag{E.20}$$

This sequence can be rewritten in a compact form as follows:

$$\begin{gathered} A_i = Q_i \cdot L_i \\ A_{i+1} = L_i \cdot Q_i = Q_i^T \cdot A_i \cdot Q_i \end{gathered} \tag{E.21}$$

As can be easily observed in Equation E.21, this process is iterative and continues until all off-diagonal elements are annihilated. This process is convergent (for more details refer to Press *et al*, 1992) and the elements placed on the diagonal are the sought eigenvalues. In contrast with Jacobi reduction where the workload is $O(n^2)$ per rotation, for the QL algorithm the computational complexity is $O(n)$ per iteration when the input is a tridiagonal matrix.

In the next section, some numerical examples are provided in order to evaluate the performances of the Jacobi reduction and the combination Householder reduction – QL algorithm.

E.5 The algorithms performance and evaluation

The purpose of this section is to investigate the performances of two popular methods employed for computing the eigenvalues associated with a real symmetric matrix. This includes a set of numerical examples utilised not only to evaluate the precision in extraction of the eigenvalues but also to assess the processing time required by both methods. In order to give relevant results, the precision is evaluated using matrices with small dimensions, a situation that allows the possibility to compute the ideal eigenvalues by hand. Then, the processing time is measured by using a set of test matrices with their dimensions incrementally increased. Table E.1 illustrates the precision in the calculation of the eigenvalues when 2 x 2, 3 x 3 and 4 x 4[25] matrices are considered as input.

Matrix	**Ideal eigenvalues**	**Jacobi reduction**	**Householder reduction + QL**
$\begin{bmatrix} 2 & 1.5 \\ 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 2.8027 \\ -0.8027 \end{bmatrix}$	$\begin{bmatrix} 3.2360 \\ 1.2360 \end{bmatrix}$
$\begin{bmatrix} 3 & 0 & -2 \\ 0 & 2 & 0 \\ -2 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 4.0055 \\ 1.9944 \\ -1 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 2 \\ -1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} \frac{3+\sqrt{5}}{2} \cong 2.618033 \\ 2 \\ \frac{3-\sqrt{5}}{2} \cong 0.381966 \\ -3 \end{bmatrix}$	$\begin{bmatrix} 2.610045 \\ 2.012730 \\ 0.375822 \\ -3.006547 \end{bmatrix}$	$\begin{bmatrix} 2.618034 \\ 2 \\ 0.381966 \\ -3 \end{bmatrix}$

Table E.1 An evaluation of the algorithms based on their results.

[25] The characteristic polynomials for 2 x 2 and 3 x 3 matrices were obtained by directly computing the determinant illustrated in Equation E.1. For 4 x 4 case the determinant was expanded into minors which were computed individually.

The results depicted in Table E.1 indicate that the precision offered by the combination Householder reduction - QL algorithm is superior to the precision associated with Jacobi reduction. In addition, it is worth mentioning that the results returned by Jacobi reduction are unreliable when it is applied to matrices with a dimension greater that 20. Fortunately, the results returned by the Householder – QL method proved to be reliable in all cases when the dimension was gradually increased to 100.

As mentioned earlier, another problem of interest consists of evaluating the computational efficiency between these two methods. To carry out this comparison, a set of symmetric matrices with a dimension ranging from 5 to 100 was utilised and the numerical results are illustrated in Figure E1.

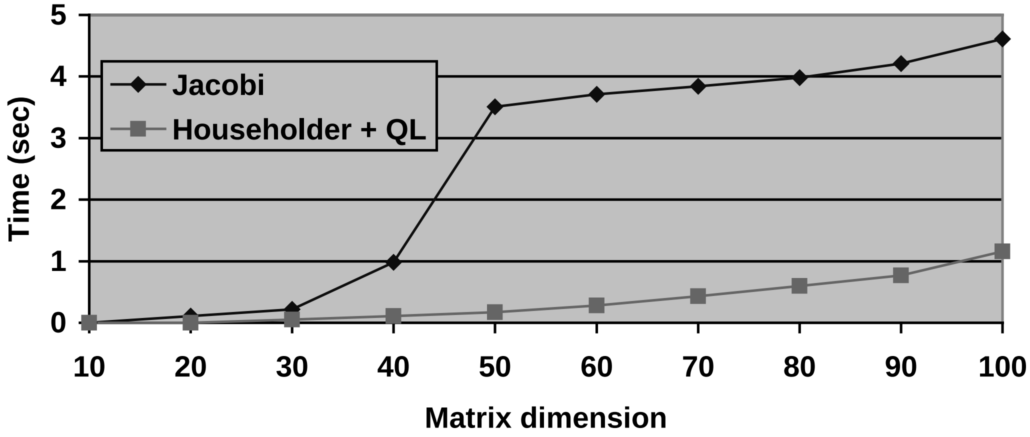


Figure E.1. A comparison of the processing times associated with the algorithms that are investigated.[26]

As expected the Jacobi reduction is computationally intensive when it is applied to large matrices. Therefore, this method is appropriate when dealing with matrices of moderate order where the computational overhead is not a major problem. Moreover, the precision associated with this method is drastically reduced when large matrices are considered, thus in these cases the use of the Householder – QL algorithm is recommended.

[26] These measurements were performed on a Pentium 133 MHz, 32 MB and running Windows 98.

Publications resulting from this research

Ghita O. and Whelan P.F. (1997), - "Object recognition using eigenvectors", *Proceedings of SPIE, vol. 3208, pp. 85-91.*

Ghita O. and Whelan P.F. (1998a), - "Eigenimage analysis for object recognition", *Proceedings of the Optical Engineers Society of Ireland and the Irish Machine Vision and Image Processing Joint Conference, National University of Ireland, Maynooth.*

Ghita O. and Whelan P.F. (1998b), - "Robust robotic manipulation", *Proceedings of Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, SPIE, Boston, Massachusetts, USA.*

Ghita O. and Whelan P.F. (1999a), - "A bifocal range sensor based on depth from defocus", *Proceedings of Image and Vision Computing '99, University of Canterbury, Christchurch, New Zealand.*

Ghita O. and Whelan P.F. (1999b), - "Real time 3-D estimation using depth from defocus", *Proceedings of Irish Machine Vision and Image Processing (IMVIP), Dublin City University, Dublin, Ireland.*

Ghita O. and Whelan P.F. (2000), - "Real time 3D estimation using depth from defocus", *Vision, vol. 16, no. 3, Third Quarter 2000.*

Ghita O., Whelan P.F. and Drimbarean A. (2000), - "An efficient method for edge thinning and linking using endpoints", *Image and Vision Computing, in press.*

Ghita O. and Whelan P.F. (2000), - "A video-rate range sensor based on depth from defocus", *Optics & Laser Technology, in press.*

Ghita O. and Whelan P.F. (2000), - “A bin picking system based on depth from defocus”, *Machine Vision & Applications, in press.*