Identification and Tracking of Maritime Objects for Collision Risk Estimation

Andrew Alexander Walker Smith

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of Doctor of Philosophy

November 2004

Identification & Tracking of Maritime Objects for Collision Risk Estimation

Andrew A.W. Smith

Abstract

With the advent of modern high-speed passenger ferries and the general increase in maritime traffic, both commercial and recreational, marine safety is becoming an increasingly important issue. From lightweight catamarans and fishing trawlers to container ships and cruise liners one question remains the same. Is anything in the way?

This question is addressed in this thesis. Through the use of image processing techniques applied to video sequences of maritime scenes the images are segmented into two regions, sea and object. This is achieved using statistical measures taken from the histogram data of the images. Each segmented object has a feature vector built containing information including its size and previous centroid positions. The feature vectors are used to track the identified objects across many frames. With information recorded about an object's previous motion its future motion is predicted using a least squares method. Finally a high-level rule-based algorithm is applied in order to estimate the collision risk posed by each object present in the image. The result is an image with the objects identified by the placing of a white box around them. The predicted motion is shown and the estimated collision risk posed by that object is displayed.

The algorithms developed in this work have been evaluated using two previously unseen maritime image sequences. These show that the algorithms developed here can be used to estimate the collision risk posed by maritime objects.

Acknowledgements:

The author would like to acknowledge a great number of people whose help, support and encouragement have led to the completion of this work. There are, however, a few people the author would like to pay special tribute to, as without them this research would not have been completed.

Firstly, I would like to thank my supervisor Dr Martin Teal whose support, advice, comments, criticism, and unique wit have all contributed to the development of the research presented here.

I would also like to thank the staff of the department of Design, Engineering and Computing at Bournemouth University, in particular Mr Jim Roach, Prof Denzil Claremont, Mrs Jacqui Holmes, and Mr Petr Voles all of who have had faith in my ability.

To my parents I owe immense thanks for their unerring commitment and belief in me and finally to my wife Ann, I am indebted to you for the support you have given me through the highs and lows of this work.

Contents:

Figures:

Tables:

Chapter 1: Introduction

1.1 Maritime Safety

With the advent of modern high-speed passenger ferries and the general increase in maritime traffic, both commercial and recreational, marine safety is becoming an increasingly important issue. In addition to high-speed, lightweight catamaran ferries capable of travelling at high speeds, container ships and cruise liners are becoming larger and heavier making them less manoeuvrable. An important question for all of these craft remains the same. Is anything in the way?

A collision of any nature involving a marine craft can have a devastating effect. History shows that a collision between a cruise liner and an iceberg can lead not only to the loss of the ship but also to a massive loss of life. In more recent times November 1999 saw the banana boat 'Dole America' collide with the Nab Tower, a lighthouse off the coast of the Isle of Wight (Fleet, 1999). In December 2002, a cargo vessel carrying nearly 2900 luxury cars collided with a container ship off the coast of Dunkirk (McBeth, 2002). Although no lives were lost in these recent cases, massive damage occurred to both the vessels involved and to the environment through the release of pollutants into the sea and atmosphere.

At present, maritime vessels employ two different forms of observation. The first is radar. Radar images show the energy that has been reflected back from objects in the full 360 degrees surrounding the vessel. Whilst being an extremely useful source of information the results are blobs on a screen that can be difficult to interpret. A large craft, heading directly towards a vessel could show up on radar as a relatively small reflection that could be misinterpreted as a small object presenting only a minor risk. Skolnik (2002) states that radar also suffers from the multipath effect where energy returning from an object then reflects off another surface (the sea) causing further returns to be received. This could result in multiple objects being displayed where there may only be one.

The second form of observation is the human eye. Either by looking directly out of the bridge window or by viewing a video monitor linked to forward-looking cameras. If an object appears to be in the path of the vessel the operator raises the alarm and avoiding action is taken. This solution suffers from the major drawback of operator fatigue. Staring at a monitor or out to sea for long periods of time on the off chance that something might appear in the path of the vessel becomes very tiresome, very quickly. Lan et al (2002) includes tedious work as a key factor in their model of human fatigue. Makris and Ellis (2003) support this by saying that the inspection of a visual surveillance system by human operators is tedious work and highly susceptible to error. They suggest that the automatic analysis of such an activity is of great benefit.

1.2 The Maritime Environment

In figure 1.1 we see a typical maritime scene as viewed over the bow of a vessel leaving port. It is a very busy scene with several maritime objects present. There are two objects toward the bottom of the image, a rigid inflatable boat (RIB) on the left and a sailing dinghy toward the right. At the

top of the image there are numerous other maritime objects, in these cases boats tied to mooring buoys. If the question 'Is anything in the way?' is asked of this image the simple answer is yes. However, if another question 'which object presents the greatest risk of collision?' were asked the answer is not obvious. Further questions have to be asked about the image:

1. Which object is closest to the vessel?
2. What is the point of collision with that object?
3. Are any of the objects moving?
4. Is the vessel closing on the objects?

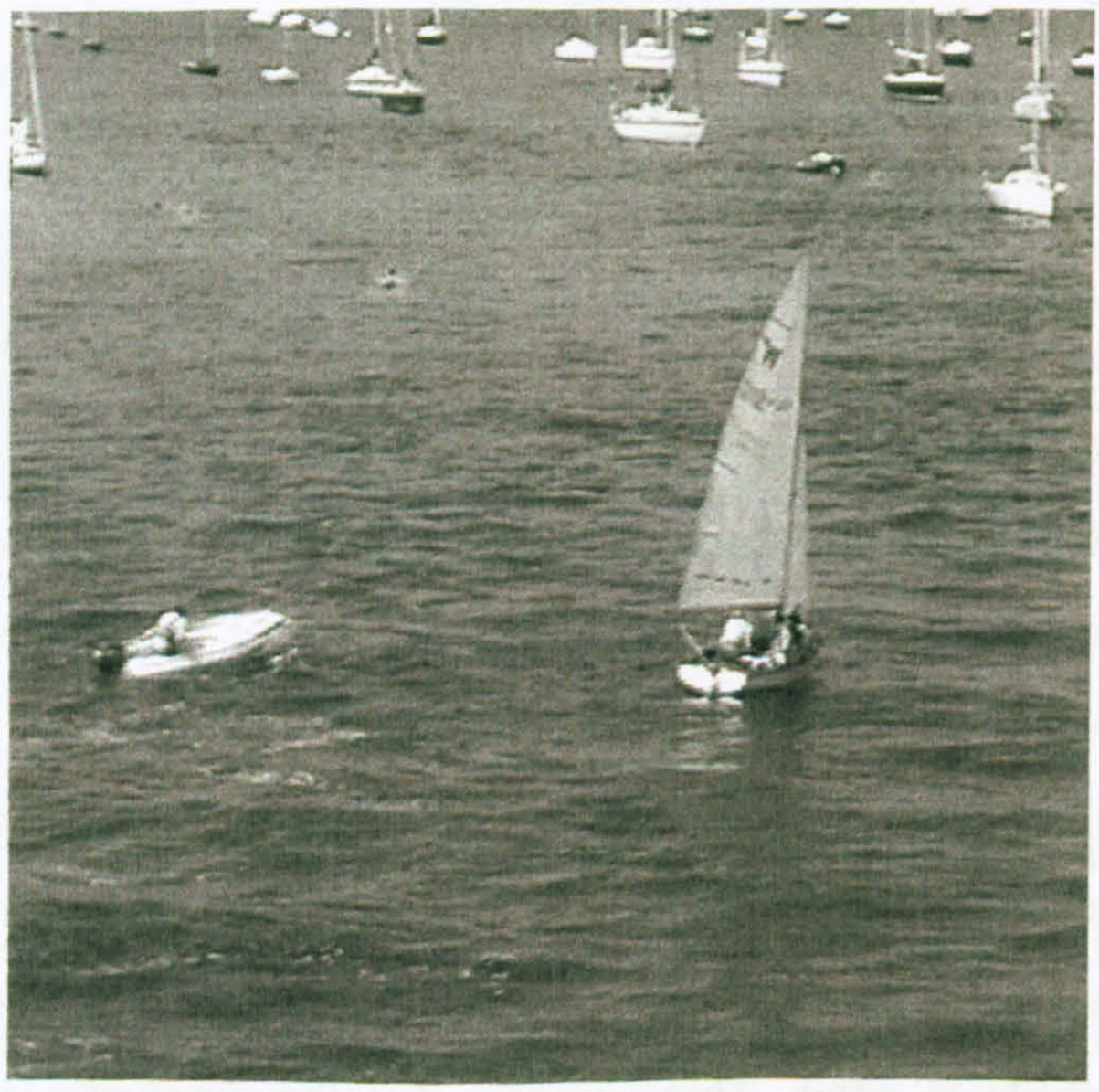

Figure 1.1 – A Maritime Scene

Given the single image above only the first two of these questions can be answered but even then certain assumptions have to be made:

- The surface of the sea can be considered to be the ground plane as no object below the surface can be observed.

- If the camera is viewing an area in front of the vessel then it can be assumed that the effect of perspective is being seen. The ground plane, and hence the objects, at the top of the image are a greater distance from the vessel than the ground plane at the bottom of the image.
- The bow of most vessels approaches a point and it can be considered that the centre bottom of the image represents this point.

Given these assumptions the object closest to the vessel is the object that is nearest the bottom centre of the image. In figure 1.1 the dinghy toward the bottom of the image is closer to the bottom centre of the image (the vessel's bow point) than the RIB and the objects at the top of the image are some distance further away.

In order to answer the final two questions a further image is required taken some time after the first to show the dynamic behaviour of the objects with respect to the vessel. Figure 1.2 shows the same scene some time after the first. In answer to question 3 it can be seen that the RIB has moved to the right but without moving any further up the image in the direction it is pointing. This is due to its own movement in a left to right direction and due to the movement of the vessel straight ahead. The dinghy has also carried out the same movement. As all the other objects are in the same relative positions it can be considered that they are stationary.

With a single uncalibrated camera and without knowing the speed of the vessel it is not possible to accurately determine the speed of the objects in the image. This is not necessary to answer the fourth question, however, as the only requirement is to determine whether the distance between the two is increasing or decreasing. If the distance decreases then the vessel is travelling faster than the object and the risk of a collision increases whereas if the distance increases then the vessel is travelling slower than the object and the risk of collision decreases. As the distance between the objects and

the bottom of the image has remained constant from figure 1.1 to 1.2 the speeds of the objects and the vessel can be considered comparable and therefore the vessel is not closing on the objects.

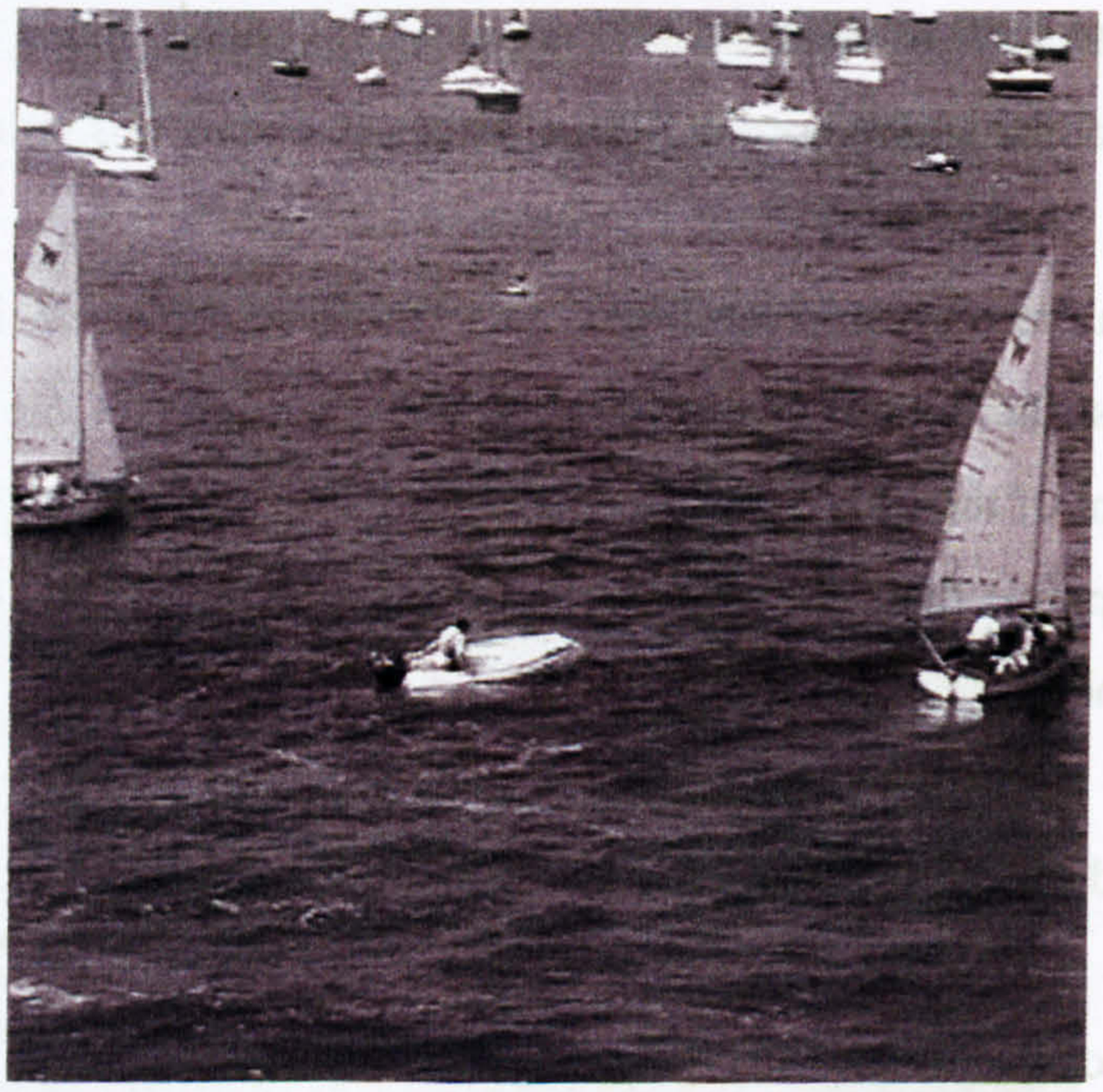

Figure 1.2 – A Maritime Scene, some time later

Revisiting the question 'which object presents the greatest risk of collision?' though gives a different answer. The collision risk from figure 1.1, the dinghy, has now moved toward the right edge of the image and is looking set to continue in that direction. It can be considered that the risk of colliding with this object is decreasing. The RIB, however, is moving further toward the centre of the image and is now the object posing the highest risk of collision. It should also be noted that a further dinghy has entered the scene from the left. Whilst not currently a major threat the risk of collision could increase if it continues toward the centre of the image.

Whilst it is extremely unlikely that none of the crew would be monitoring the sea in front of the vessel in such a crowded location once out in open water such as that shown in figure 1.3 the situation may be quite different.

This view may continue for many hours at which point human fatigue will become a real issue.

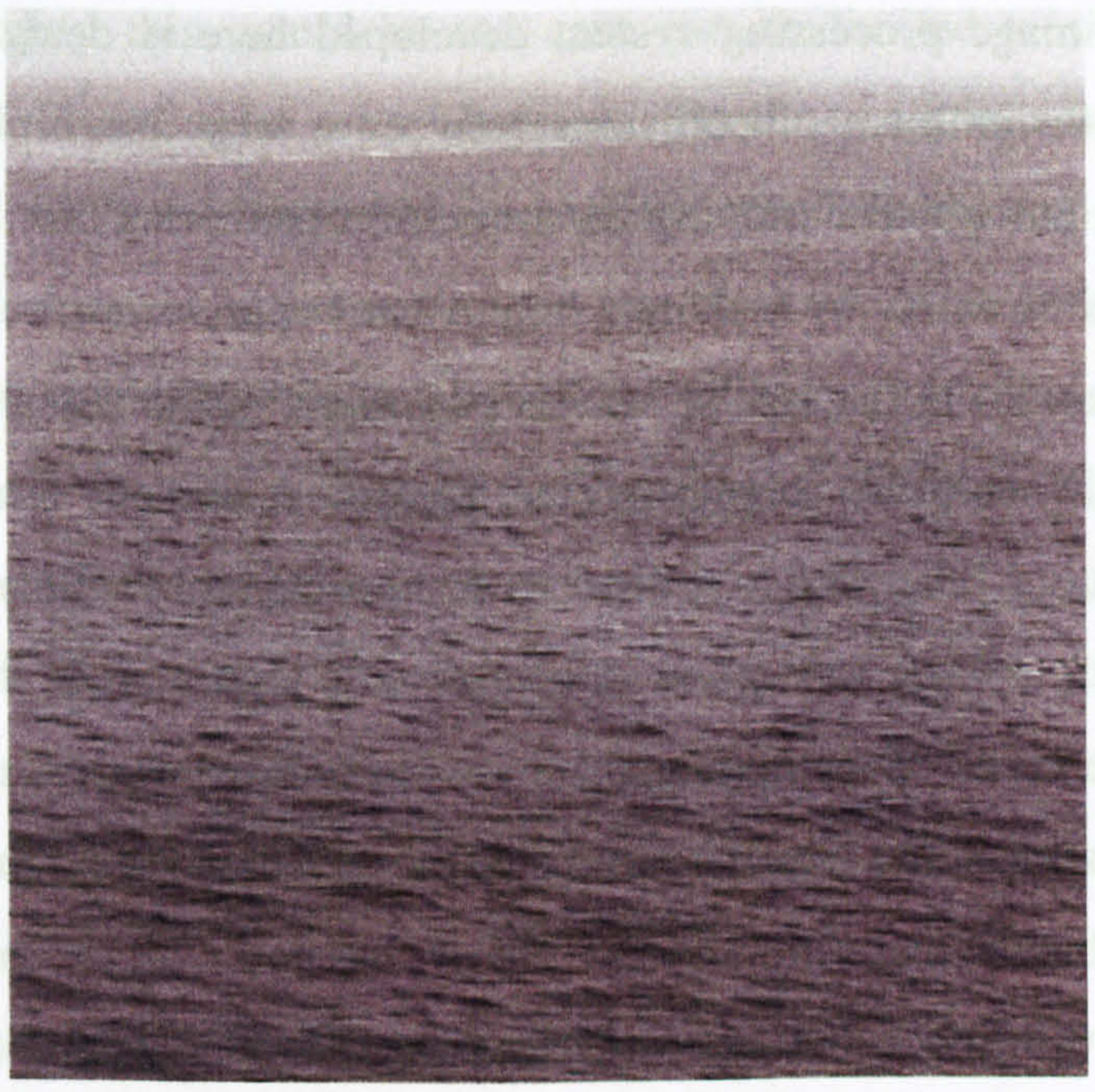

Figure 1.3 – Open Water

1.3 Machine Vision System

In answering the questions posed above several processes have been gone through, each of which can be realised using image processing techniques leading to the possibility that a complete machine vision system could be developed for the tedious task of monitoring the sea in front of the vessel. The first process was to identify the objects present in the scene. This can be realised using some form of image segmentation where regions of interest (objects) are distinguished from the rest of the image. Having identified the objects the next task is to follow their movement over time. In an image processing system this is performed by a tracking algorithm to match the regions of interest between frames. The final process is to use the knowledge of the movement of the objects to decide what collision risk the objects are posing a vessel at the camera position. That part of an image

processing system dealing with knowledge based decision making is known as high-level vision.

The image processing system developed here is designed to address the specific problem of collisions between a large maritime vessel and other maritime objects moving in areas of open sea. The aim is to increase maritime safety by making a large vessel operator aware of any objects that are present in the area of sea directly ahead of the vessel and estimating the risk of collision posed by the identified objects. This is achieved by capturing long, image sequences from a camera mounted on board a large vessel, looking forward at an area directly ahead of the vessel from the bow of the vessel to the horizon. The image processing system then identifies and tracks maritime objects present in the scene; predicts the future motion of the objects; and estimates the risk of collision with another maritime object. This information is presented to the operator as visual cues superimposed onto the original captured image.

Figure 1.4 shows the context of the system developed in this thesis. The system is shown with an object, a camera, and an observation monitor. The input is an image sequence captured from a camera and the output from the system is the input image with any maritime objects identified highlighted to the operator. In addition the predicted future motion of each object is displayed along with the estimation of the collision risk the object poses to the vessel the camera is mounted on.

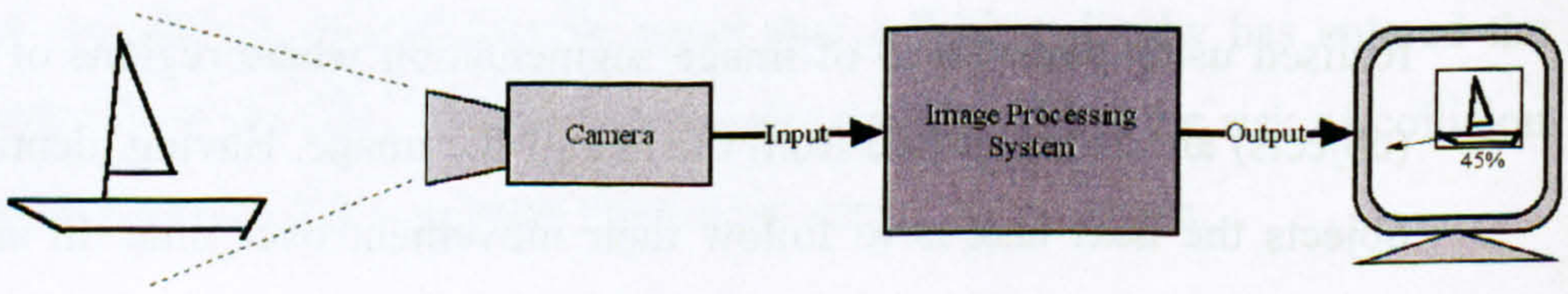

Figure 1.4 – System Context diagram

Certain assumptions have been made about the working environment in order to bound the research presented here. As the system is intended to be of greatest use when the level of observation is at its lowest the weather conditions when this would be the case can be limited to conditions of daylight and relative calm seas. These are also the conditions when moving objects will be travelling at their highest speed increasing the risk of collision. As a vessel enters and leaves port the level of observation will be at its highest due to the increased risk of collision with the multitude of other maritime objects present in the confines of the port. Therefore this system is considered to be used away from port, out in the open sea where the level of observation will be much lower and where there will be considerably fewer objects present. It can therefore be assumed that there will only be a small number of objects present in any one image.

As the system is not attempting to calculate the distance of any object from the camera a stereo camera is not required. The only criteria regarding an object's distance from the camera is whether the distance between the two is increasing or decreasing and this can be ascertained by observing the number of pixels between the bottom of the object and the bottom of the image. Therefore a single camera recording monocular image sequences has been used throughout.

A frame rate of 10 frames per second has been used here as this is considered a high enough rate to capture object motion between frames. A higher frame rate would require more data to be processed without adding a significant amount of accuracy to the results and a lower frame rate would compromise the tracking algorithm due to the greater movement of objects between frames.

Out in the open sea where this system is to be used, maritime objects, be they the camera platform or an object in the scene, are likely to be

commercial vessels and therefore moving in the most efficient manner. This is likely to be on a direct bearing between two points at a constant velocity. Because of this it can be assumed, for the purposes of predicting future object motion that object motion will be uniform across the image.

1.3.1 System Overview

The image processing system developed here is subdivided into four accepted machine vision steps as shown in figure 1.5. They are: Image Smoothing; Segmentation; Object Tracking; and High Level Vision.

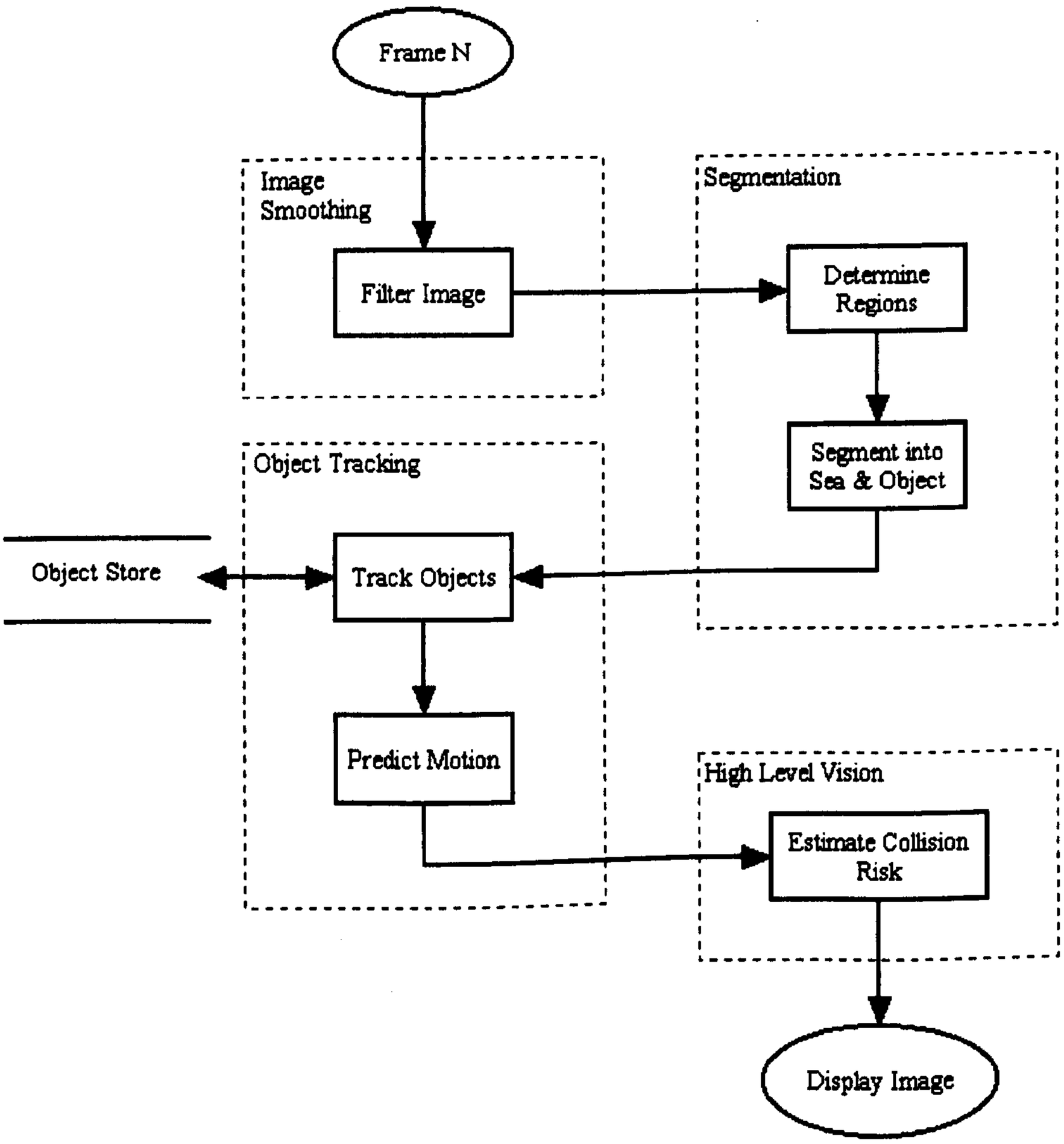

Figure 1.5 – System block diagram

Image Smoothing

The image acquisition process introduces noise to the images from a number of sources including the image sensor, recording medium, and through image digitisation. In order for further processing steps to make full use of information present in the images it is necessary to reduce the undesirable effects of noise. Gaussian noise contains variations in intensity that are drawn from a Gaussian or normal distribution and is a very good model for many kinds of sensor noise (Jain et al, 1995). In addition, of particular concern to the maritime images used here, Voles et al (1999) note that the waves of the sea also cause noise. They go on to say that this wave noise does not have a gaussian distribution and that traditional (linear) methods of filtering are ineffective This is also implied by Pozzer and Pellegrino (2001) who add noise to their synthesised sea images.

Segmentation

Maritime images are, by their very nature, open world scenes. These, as opposed to synthetic images or those from a highly constrained environment, introduce a range of complexities into the analysis of the image. Teal (1997) names these as uncontrolled and variable light conditions; different object scales; orientation of the object with respect to the camera; and partial or full occlusion of the object. Pozzer and Pellegrino (2001) note that the behaviour of the sea will vary also as a result of external factors such as wind and temperature.

Segmentation is concerned with splitting an image up into regions (also called segments or areas) that each hold some property distinct from their neighbour (Low, 1991). This part of the system is concerned with separating the maritime image into two distinct regions, sea and object. Segmentation algorithms generally fall into a small number of categories. Bassmann and Besslich (1995) detail region-oriented and contour-oriented segmentation

techniques although there are also methods involving the use of texture analysis and motion detection.

Object Tracking

The third stage considers the matching of those segmented objects across multiple frames. The tracking of objects has been a subject of considerable interest to researchers from many fields in the machine vision world. Work has been carried out into tracking small objects, in particular aircraft in low contrast images (Davies et al, 1998), objects from sporting events such as players and balls (Agbinya and Rees, 1999), vehicles on a road from both a remote viewpoint (Teal, 1997), (Enkelmann, 1989), (Koller and Nagel, 1993) and from another vehicle (Smith and Brady, 1995). Here it is the tracking of maritime objects from another vessel.

Techniques for tracking objects through a sequence of images include the use of optical flow (Smith and Brady, 1995), model-based tracking (Ferryman et al, 1998), measurements of object parameters (Ellis et al, 1991), and the Kalman filter (Kalman, 1960).

High Level Vision

High level vision is concerned with using previously known knowledge about the goal or application of the system to make decisions about the information presented in the image. The final stage in this system uses a high-level vision technique to estimate a measure of risk that the objects will collide with the vessel the camera is mounted on.

1.4 Image Acquisition

The image processing system described in this thesis has been developed using three test sequences: Poole; Dover; and Portsmouth. These contain different maritime objects and sea conditions that are representative of a typical scene the system is designed for. In order to obtain the best possible images with which to test the algorithms these sequences were taken from a static camera position. The developed system has been evaluated using two previously unseen sequences taken from a moving camera platform. The environment of these evaluation sequences is considered to be typical of the type the system would be expected to operate in and they contain a variety of maritime objects.

A camcorder was mounted looking forward over an area of sea. Image sequences with an original frame size of 768 x 576 pixels were captured onto video film and later digitised using a PC containing a video capture card. The images were digitised at a rate of 10 frames per second, cropped to 512 x 512 pixels, and stored in the loss-less GIF format. The camcorder was set to record image sequences at a focal length of 100mm, giving a field of view of approximately 23 degrees. This focal length was chosen as from a typical camera height of 15 metres this gives a cropped image stretching from the bow of the vessel to the horizon. The range of the objects captured in the images was typically 50 metres to 1000 metres.

1.4.1 Algorithm Development Test Sequences

Three image sequences have been used to develop the algorithms presented here. Sequence 1 was taken looking out over Poole Harbour on a very dull overcast day with very little wind. The sequence is 60 frames in length and shows a dark stationary dredging boat in the centre of the image and a

launch with a dark hull and light cabin entering the scene and travelling from right to left. The sea state is calm and the shoreline occurs at a level of 127 pixels down from the top of the image throughout the sequence. Figure 1.6 shows two frames from this sequence.

Sequence 2 is of Dover Harbour on a very bright day with a moderate breeze. The sequence is 270 frames long and shows the harbour wall in the top right of the images and a buoy in the centre. The sea conditions are a slight swell with considerable illumination differences due to this. There is no shoreline in this sequence so the level has been manually set to 0 pixels down from the top. Figure 1.7 shows two frames from this sequence.

Sequence 3 was taken in Portsmouth on a calm overcast day. This sequence is 150 frames in length and shows a number of small craft moving from the right towards the upper left of the image. As with the Dover sequence there is no shoreline in this sequence. Figure 3.4 shows two frames from this sequence.

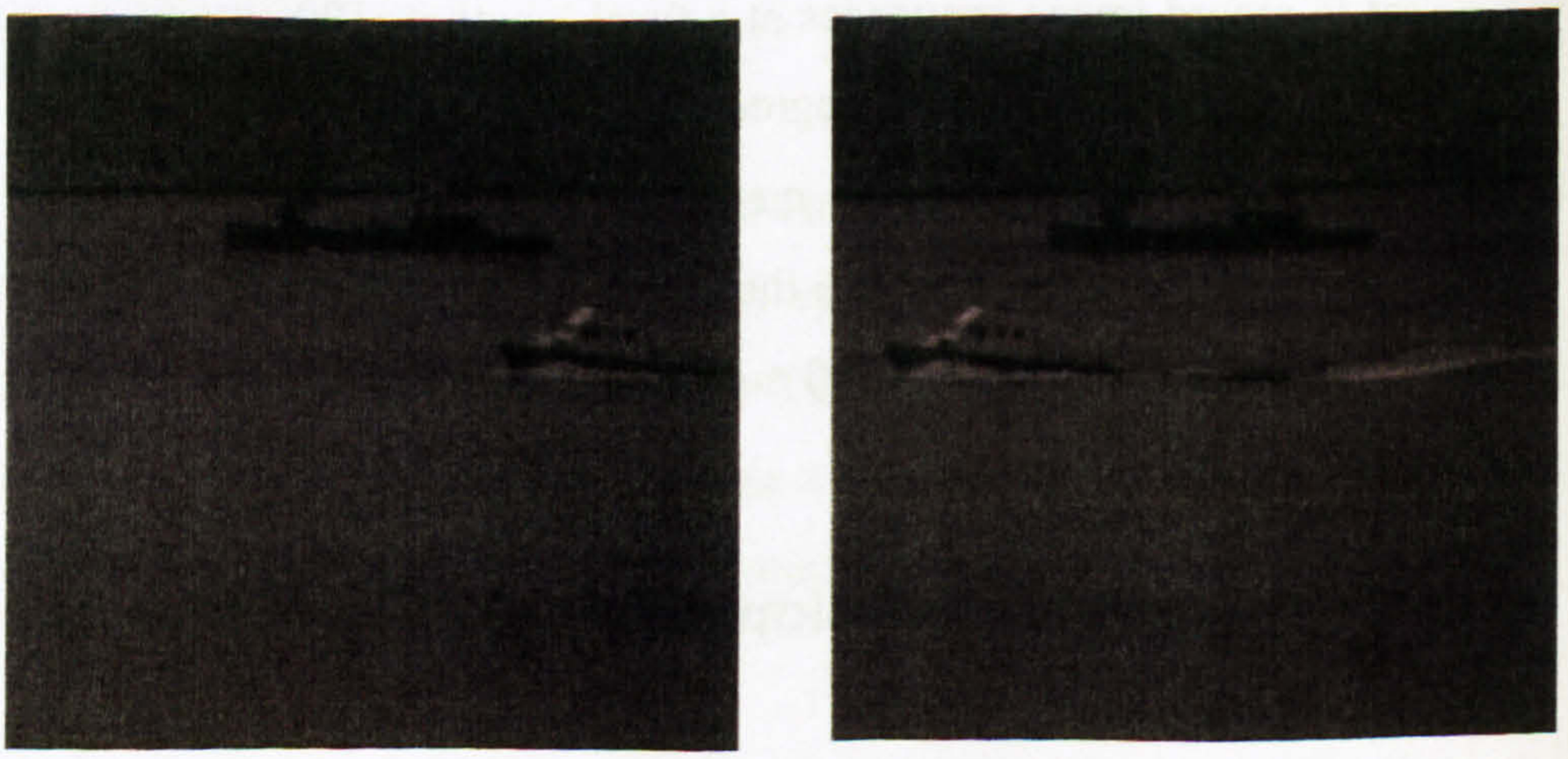

Figure 1.6 – Two frames from the Poole test sequence

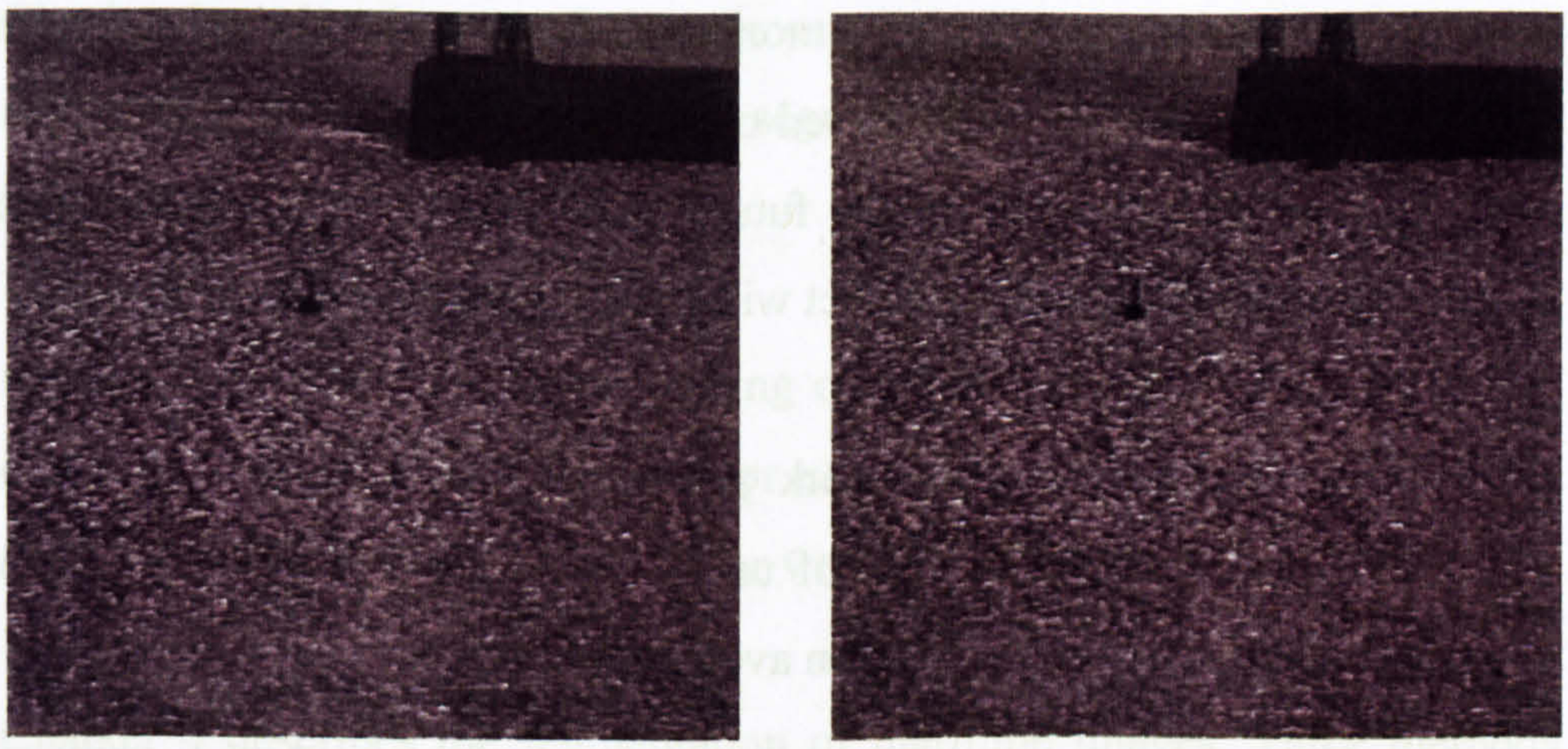

Figure 1.7 – Two frames from the Dover test sequence

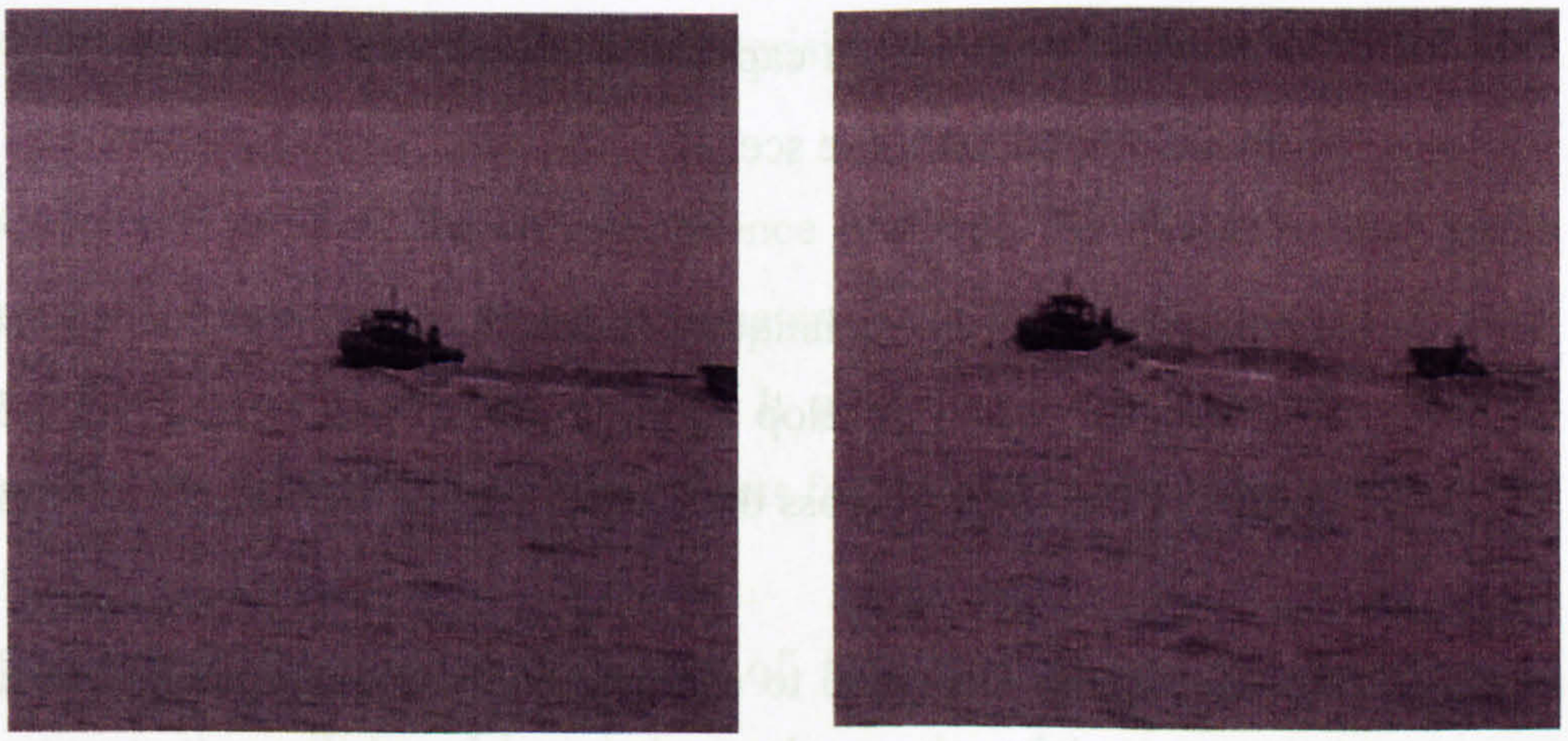

Figure 1.8 – Two frames from the Portsmouth test sequence

1.5 Research Aims

The specific problem being addressed in this work is that of maritime collision avoidance. The principal aim of this research is to develop a series of image processing algorithms that could be incorporated into a machine vision system designed to aid the safe passage of vessels travelling in a maritime environment. Any system would then be mounted on board a ship and would monitor an area of the sea directly ahead of the vessel. The key

tasks of the image processing algorithms are to identify and segment maritime objects from long, monocular maritime image sequences and track those objects over time. Based on the object motion, to predict and display to the operator the likely future path of the object and estimate the percentage risk that the object will collide with the operator's vessel.

The contribution of the work presented here is a set of algorithms that demonstrate the feasibility of using a machine vision system to solve the problem of maritime collision avoidance.

This research has three main objectives:

- To develop an algorithm capable of identifying and segmenting objects from a cluttered maritime scene.

- To employ tracking techniques to match these maritime objects from frame to frame and develop an algorithm to predict and plot the future course of the objects across the scene based on the objects prior motion.

- To develop an algorithm to estimate and show the percentage risk of a collision between the object highlighted and the vessel the camera is mounted on.

1.6 Thesis Outline

Chapter 1 introduces the project, it discusses the background and reasons why there is a need for a machine vision solution in marine safety. It gives an overview of the system, gives details of the image capture process and the test sequences used in the development of the system and finally outlines the organisation of this thesis.

Chapter 2 reviews some of the relevant literature covering the key areas addressed by this thesis, filtering, segmentation, tracking, and high-level reasoning.

Chapter 3 presents the pre-processing of the images. The FWV filter was developed as part of this research for use with maritime images and this is discussed and compared against other well-known filters.

Chapter 4 describes the segmentation of maritime images. Three methods have been applied to the test images, results shown and conclusions drawn. A thresholding algorithm, a method of segmentation using frame differencing and a method of characterising the sea are all implemented.

Chapter 5 presents the correspondence problem. The Kalman filter and a method of measuring object parameters have been implemented to track maritime objects between frames. It also details the algorithm used to predict the future motion of the objects for the object parameter method.

Chapter 6 describes the method of high-level reasoning employed to estimate the collision risk and shows the results of the system using the test sequences utilised throughout the development process.

Chapter 7 evaluates the performance of the system. Two previously unused sequences are used to test the system without any alteration to the parameters. A critical analysis of the results is also shown in this chapter.

Chapter 8 draws conclusions and discussions on all of the developments made throughout the research. The current limitations are discussed and recommendations for future work are also made.

Chapter 2: Literature Review

2.1 Introduction

In the machine vision system being developed here the goal is to take a sequence of images, process the thousands of pixels that make up each image to identify the objects present in the image and robustly track those objects over time. A decision then needs to be taken for each identified object as to its estimated risk of collision with the camera platform. Applications such as this require image-processing techniques that span low, intermediate, and high level vision.

Low level vision techniques are concerned with the manipulation of the image data at the pixel level. The aim is typically to reduce the effects of noise and to enhance data relating to useful features such as edges. Intermediate level processing takes the enhanced image data and tries to identify features or regions of interest within the image, usually called objects. A task performed under this heading could include segmentation. High level vision by contrast attempts to apply some predefined reasoning to the identified objects. This could be to track the objects over multiple frames for surveillance tasks such as intruder detection or it could be for product inspection where the aim is to identify those objects that do not adhere to a set criteria. In this application it is to track objects present in the image and estimates the collision risk posed by each object.

Each of the vision levels has seen considerable research carried out and many results reported. In the following sections techniques covering the four stages identified for this system will be reviewed to give direction to the research presented in this thesis.

2.2 Image Smoothing

Image smoothing is the set of local processing methods which has the aim of suppressing image noise (Sonka et al, 1993). These methods are categorised as linear and non-linear. Linear filters are good at removing Gaussian noise and are implemented using the weighted sum of the pixels in successive windows. A typical approach to this is to use a convolution mask (Jain et al, 1995). The term non-linear filter is given to any filter that is not a weighted sum of pixels. Both linear and non-linear filters can be either spatially variant or invariant (Jain et al, 1995). Noise in maritime images is considered to come from the image capture process and from the waves of the sea.

Bassmann and Besslich (1995) describe the linear mean filter using a convolution mask size of 3 by 3. The mask is placed over the 9 pixels in the top left corner of the input image. The central pixel under the mask is replaced with the average of the pixel values in the whole of the mask. Therefore in this case the greylevels of the pixels in the mask would be summed and divided by 9. The mask is then repeatedly shifted 1 pixel to the right and the calculation repeated. At the end of the top row the mask is moved down by one pixel and the process repeated. Bassmann and Besslich (1995) say that whilst the result of the mean filter brings the greylevels of noisy pixels closer to those of uncorrupted pixels any formerly steep greylevel step between regions (edges) will be flattened (blurred). Gonzalez and Woods (1992) say that one of the principal difficulties with linear filtering is that it blurs edges and other sharp details.

There are variations of linear filter using different weighting values in the mask. Jain et al (1995) say that when designing linear smoothing filters the filter weights should be chosen so that the filter has a single peak (main lobe) and symmetry in the vertical and horizontal directions.

Many non-linear filters have been developed with one of the most popular being the median filter. In its simplest form the greylevel of each pixel is replaced by the median of the greylevels in a neighbourhood of that pixel and is particularly effective when the noise pattern consists of strong, spikelike components and the characteristic to be preserved is edge sharpness (Gonzalez and Woods, 1992).

Sonka et al (1993) state that the main disadvantage of median filtering in the usual rectangular neighbourhood is a damaging of thick lines and sharp corners in the image. They go on say that this can be avoided by using a different neighbourhood shape. If horizontal and vertical lines need preserving then they suggest the neighbourhood shown in figure 2.1..

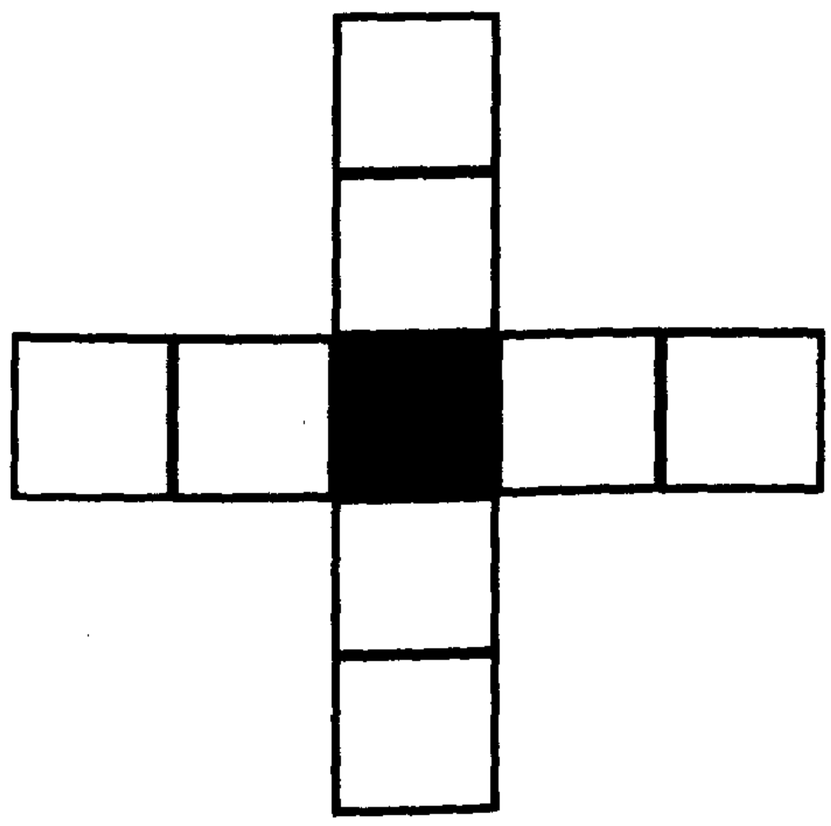

Figure 2.1 – Horizontal and Vertical Line Preserving Median Neighbourhood

Sonka et al (1993) go on to say that median filtering belongs to a category of techniques known as rank filtering which is described as the ordering of pixels in some neighbourhood followed by the calculation of some statistics over this sequence. The *k*-closest averaging method is another rank filter. Low (1991) says the aim of this filter is to preserve, to some extent, the actual values of the pixels without letting noise get through to the final image. To perform the operation Low (1991) says all the pixels in the window (neighbourhood) are sorted and the *k* pixels values closest in value to the target pixel – usually the centre of the window – are averaged. *k* is a selected constant value less than the area of the window.

Kuwahara et al (1976) takes the principle of the rank filter and applies it across a number of neighbourhoods in their work on RI-angiocardiographic images. Instead of a single neighbourhood Kuwahara et al (1976) use four neighbourhoods of size W x W around the target pixel, i,j, as shown in figure 2.2.

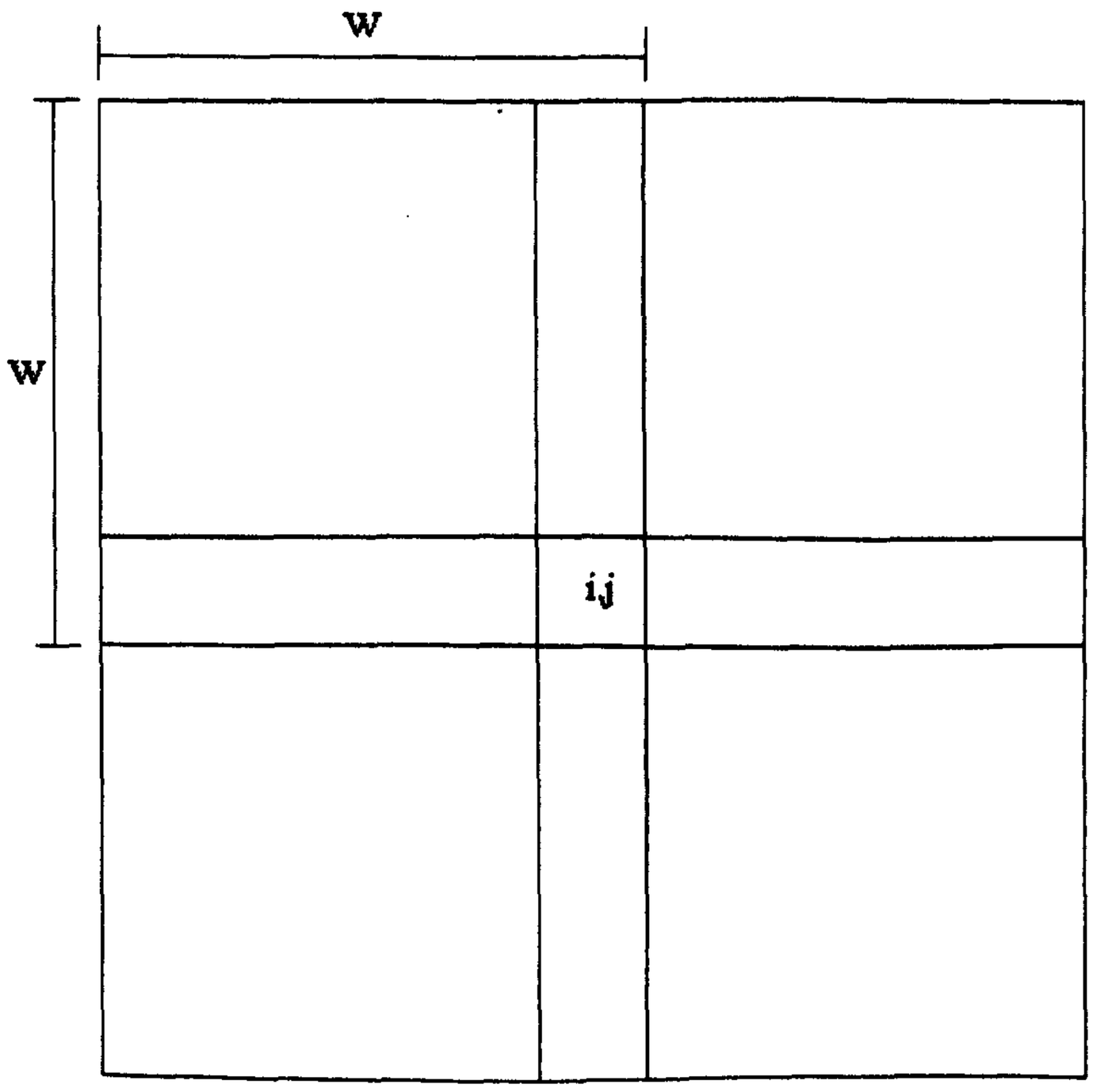

Figure 2.2 – Kuwahara filter neighbourhoods

For each of the neighbourhoods the mean and variance are calculated and then sorted by variance. The neighbourhood with the lowest variance is considered to be that to which the target pixel belongs and it is replaced with the mean value of that neighbourhood. Kuwahara et al (1976) say that the lowest variance is chosen as this region will not contain a boundary (edge) because the existence of a boundary makes the variance increase. They show the result of this through a simulation study on 1-dimensional data. The results of this can be seen in figure 2.3.

The original data, figure 2.3a, are normal random numbers having different mean values on either side of a control boundary. While, in the mean filter, figure 2.3b, blur near the boundary is predominant, in the non-linear Kuwahara filter, figure 2.3c, the result has a sharp transition at the boundary.

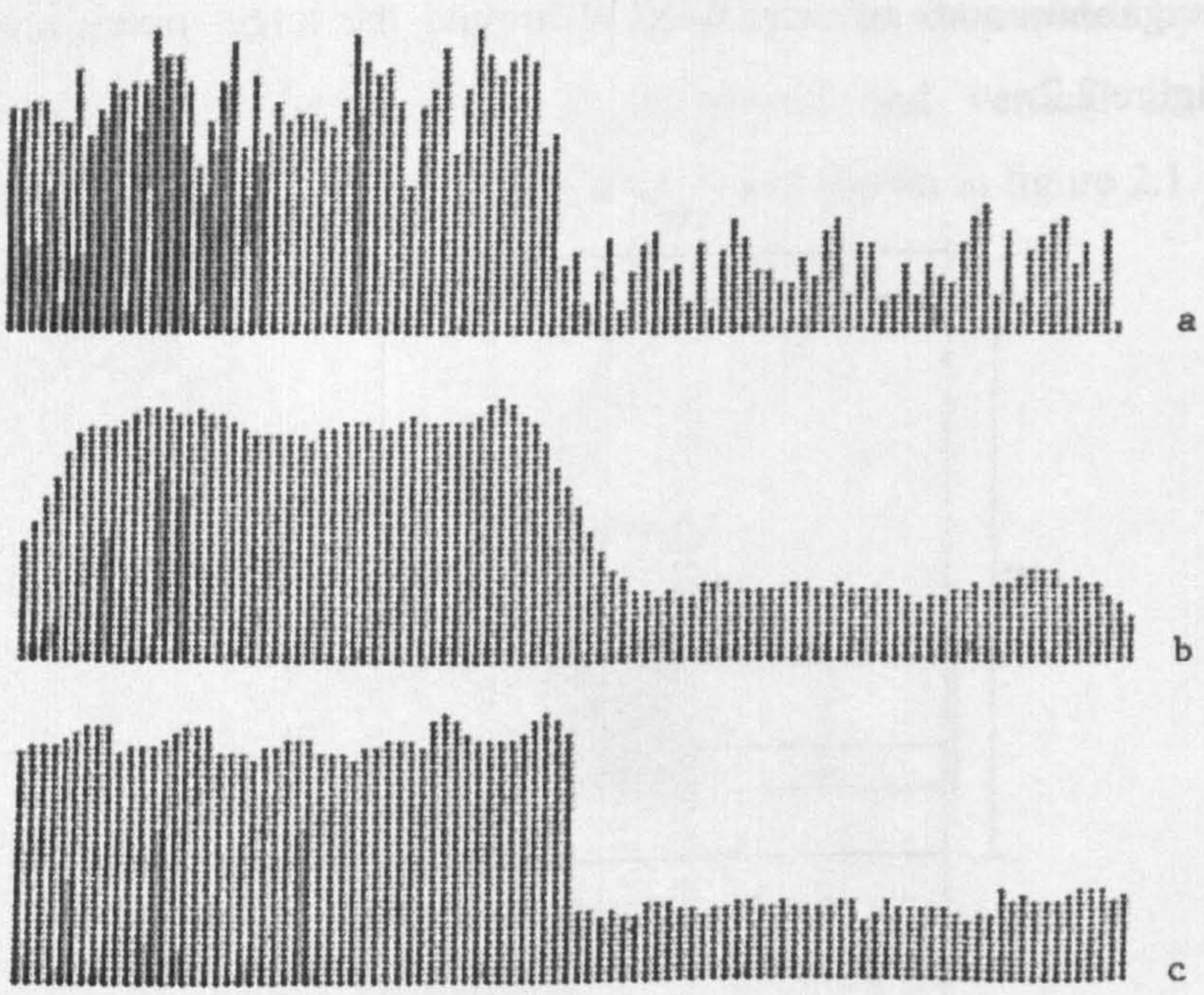

Figure 2.3 – Simulation result on 1-dimensional data: a) original data, b) mean filter (window size 10), c) Kuwahara filter (window size 10) (Kuwahara, 1976)

In an extension to the Kuwahara filter, Minato et al (1987) repeatedly apply the Kuwahara filter to the same image until no further pixel value changes occur. This results in a fixed-point image. The fixed-point set of a filter is an array (image) which is unchanged under the operation of filtering. Figure 2.4 shows an encouraging result of the repeated use of the Kuwahara filter. Figure 2.4a shows the original 1-dimensional data set. Figure 2.4b after the Kuwahara filter has been applied once and figure 2.4c the fixed-point image, which in this case occurred after 10 iterations of the filter. This shows edges not only preserved but also enhanced. With regard to maritime images though this could lead to boundaries between waves also being enhanced potentially leading to them being segmented as false objects.

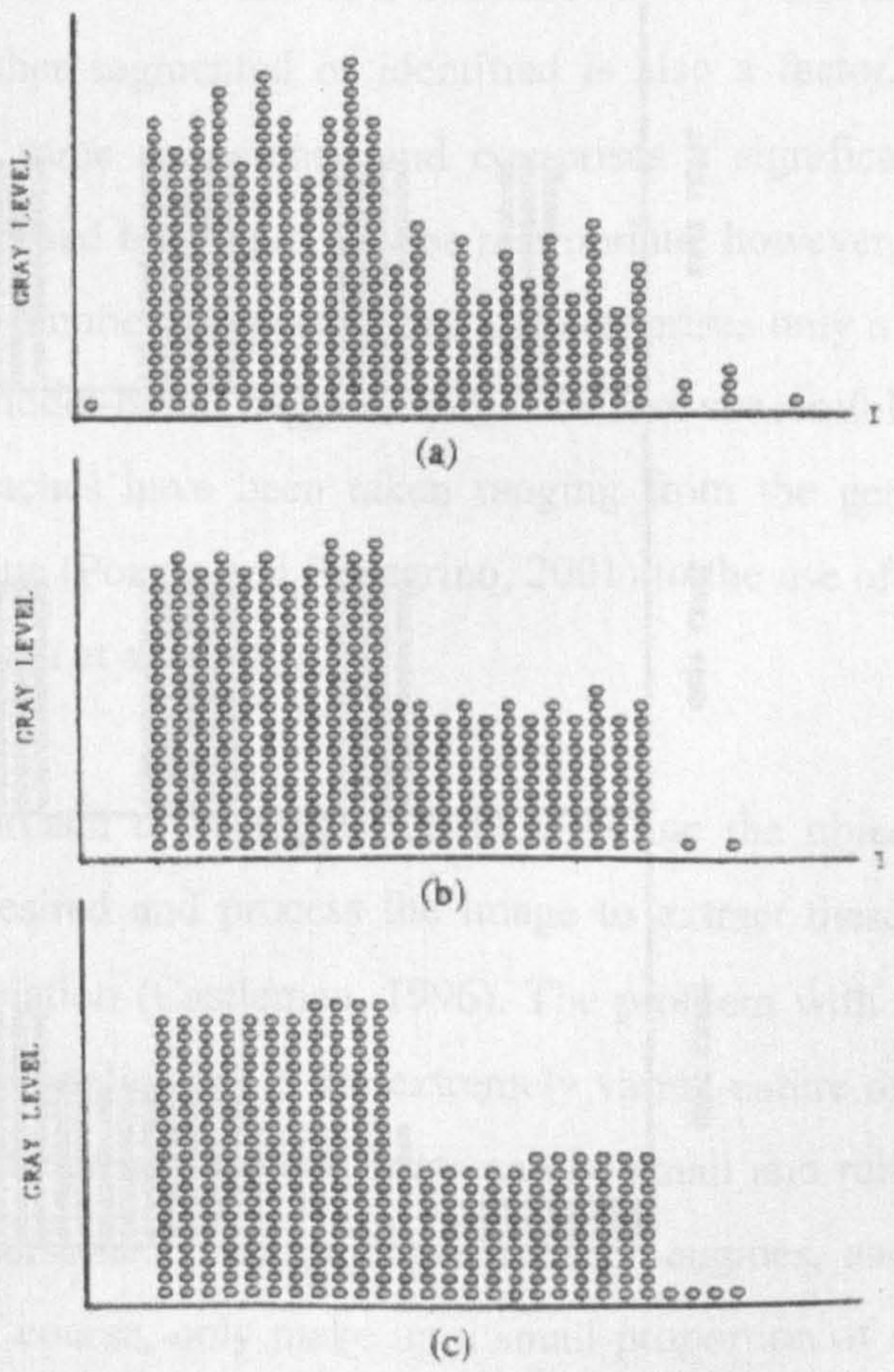

Figure 2.4 – Fixed-Point Kuwahara Filter: a) original image, b) Kuwahara applied once, c) Fixed-point image (Minato et al, 1987)

Less encouraging for this method is the second example Minato et al (1987) show that is repeated here in figure 2.5. This shows the sensitivity of the window size when using this method. Figure 2.5a shows the original image that consists of four patches 3, 4, 5, and 6 pixels wide each separated by null regions 4 pixels wide. Figure 2.5b shows the fixed-point image with a window size of 3. Patches of less than 4 pixels have been eliminated. Figure 2.5c shows the fixed-point image with a window size of 4 where only the patch with 6 pixels has survived unchanged and all the others have been averaged. Although this method initially looked promising it suffers from being very sensitive to the size of the window and the number of iterations required to arrive at a fixed-point image.

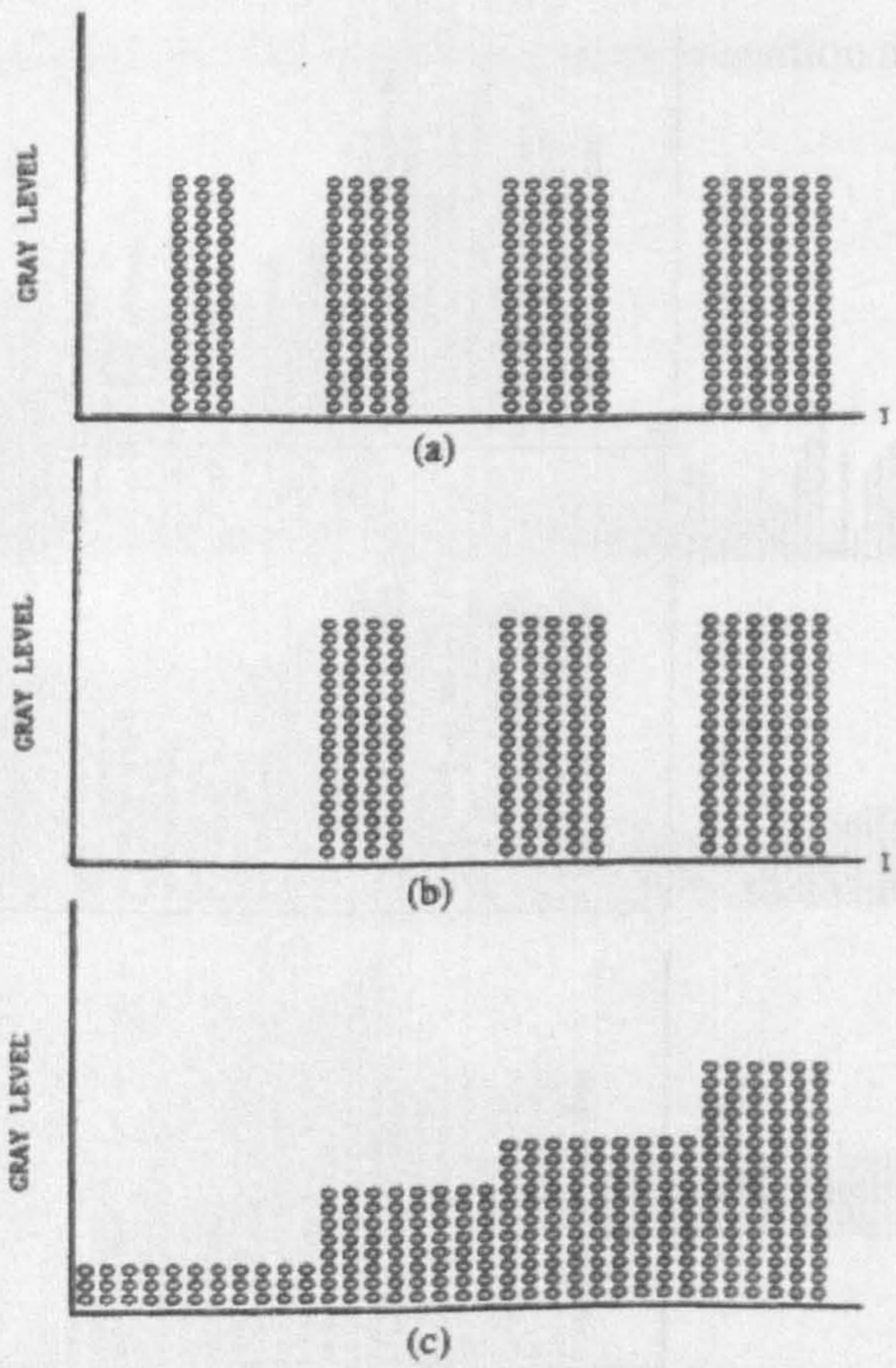

Figure 2.5 – Fixed-Point Kuwahara: a) original image, b) Fixed-point image window=3, c) Fixed-point image window=4 (Minato et al, 1987)

2.3 Segmentation

Image segmentation is an important and challenging problem in image analysis (Galic and Loncaric, 2000) (Tu and Zhu, 2002) and many methods have been developed to carry out this task. The goal of segmentation has variously been described as the identification of groups of similar image primitives (Puzicha et al, 1999), the parsing of an image into its constituent parts (Tu and Zhu, 2002), and with particular reference to texture segmentation as the accurate partitioning of an image into different (textured) regions (Dunn and Higgins, 1995). The approach to segmenting an image can depend on a number of factors such as the context of the image, is it a real-world scene or a constrained environment? The type of object being either segmented or identified is also a factor. If the object always has the same appearance and comprises a significant number of pixels a model-based technique may be appropriate, however, if the objects can take on any number of appearances and comprises only a small number of pixels then model-based methods might be less successful (Teal, 1997). Different approaches have been taken ranging from the generation of an artificial sea scene (Pozzer and Pellegrino, 2001) to the use of the frequency domain (Sanderson et al, 1999).

A common approach to segmentation is to define the objects or type of region that is desired and process the image to extract these. This is also called object isolation (Castleman, 1996). The problem with following this approach for this application is the extremely varied nature of the maritime objects that are to be segmented. Boats can be small and relatively flat, or large and cumbersome. They can have sails or engines, and be black or white. Boats, of course, only make up a small proportion of the number of objects that could be encountered on the water. There are also channel marking buoys, mooring buoys, harbour walls, piers, pontoons, debris floating in the sea, and land.

2.3.1 Region-Oriented Segmentation

Thresholding is a widely employed technique for the segmentation of grey level images. This approach is based on the assumption that the image has a uniform and stationary distribution of intensities over the object and over the background (Zhang and Desai, 2001) or that objects can be distinguished from the background by their intensity values (Cheng et al, 1998). Jain et al (1995) say that any thresholding method should automatically determine the threshold value to make the segmentation more robust. They go on to say that automatic thresholding analyses the grey level distribution in an image, usually by using a histogram of the grey levels, and applies some knowledge about the application to select the most appropriate threshold value. Boyle and Thomas (1988) describe a technique for segmentation by histogram for scenes that consist of an object on a uniform or slowly varying background. They expect such a scene to have a histogram with two peaks, one corresponding to the background and one to the object. This is shown in figure 2.6. It is important to note that Boyle and Thomas (1988) are not saying that the intensity is the same throughout the object or background region, just that they are distinguishable from each other and the trough between them in the histogram is easy to detect. Once the threshold has been determined any pixel whose intensity is below it is set to 0 and any pixel above the threshold is set to 1.

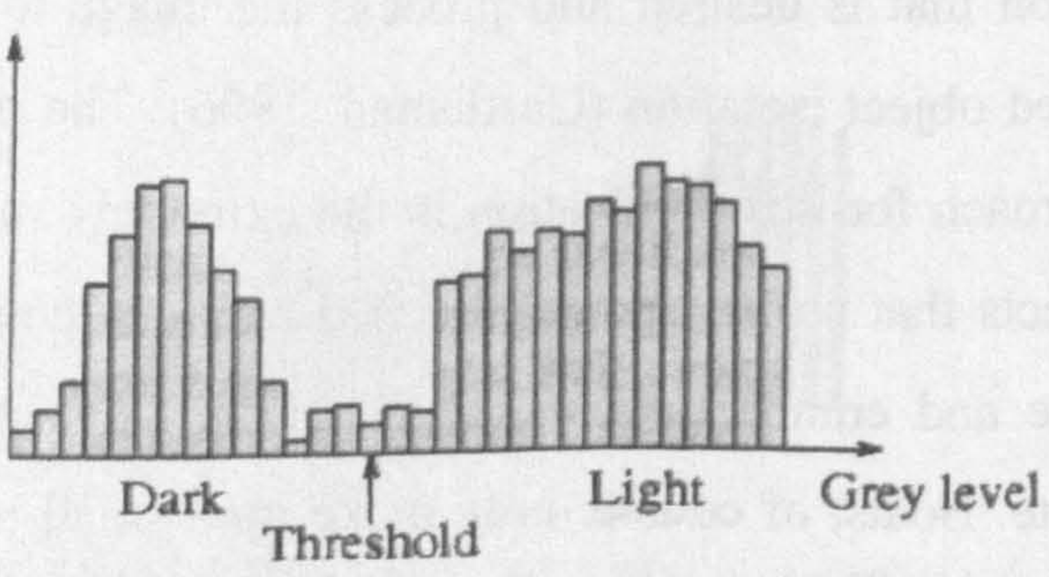

Figure 2.6 – Bimodal Histogram

Cheng et al (1998) believe that image processing bears some fuzziness in nature due to information loss while mapping 3D objects into 2D images, ambiguity and vagueness in edge, boundary, and region definitions, and in interpreting low level processing results. This belief is followed and uses the maximum entropy principle and the concept of a fuzzy *c*-partition to select the threshold values in grey-level images. In many real-world cases the boundaries between subsets are usually not clear. For this reason the hard *c*-partition is extended to the fuzzy *c*-partition which allows objects in the set to partially belong to different subsets. As there is no sharp boundary between two fuzzy sets any partition calculated would have been determined subjectively and there may exist some other partition that can produce better results. In order to automatically find the partition that best partitions the image Cheng et al (1998) use a simulated annealing approach and the fuzzy *c*-partition entropy as the criterion to measure the fitness of a fuzzy partition. This method has been applied to several real-world images including that of a coastal scene including a lighthouse and sea. The original image can be seen in figure 2.7. The fuzzy 2-partition threshold value for this image is 115 and the result is shown in figure 2.8. By way of comparison Cheng et al (1998) applied Kapur's method (Kapur et al, 1985) of 2-level thresholding which also uses the entropy of the grey level image histogram. The result of this is shown in figure 2.9. They conclude that their approach has outperformed existing methods as it has segmented the main components of the image well. In contrast by using the Kapur method (Kapur et al, 1985) the contours of the objects have disappeared and some objects such as the upper part of the tower and part of the house have vanished.

Tu and Zhu (2002) state that image segmentation is a long-standing problem in computer vision and it is found difficult and challenging for two main reasons. First, is the fundamental complexity of modelling the large amount

Figure 2.7 – Original coastal image comparison (Cheng et al, 1998)

Figure 2.8 – Fuzzy segmented image image

(Cheng et al, 1998)

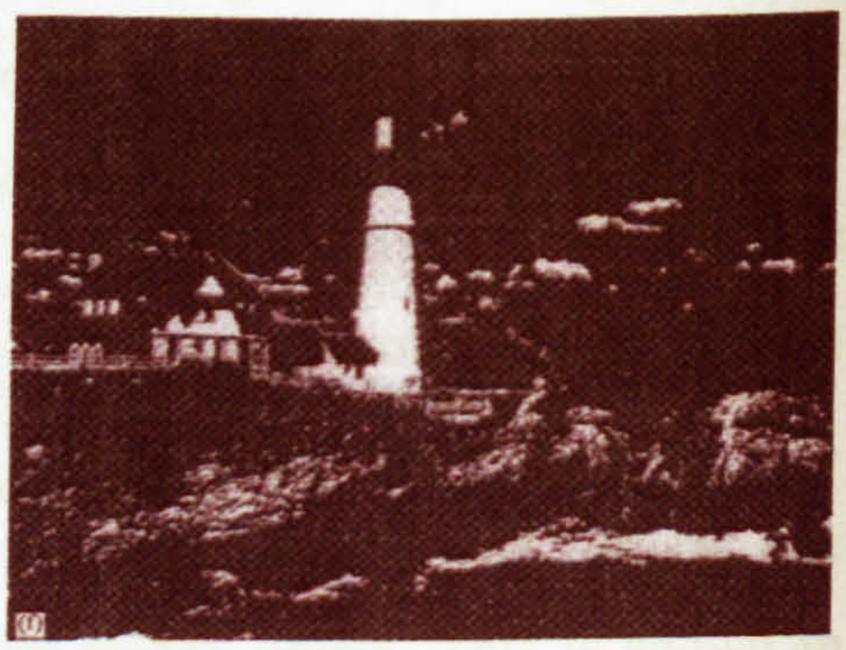

Figure 2.9 – Kapur segmented image

(Cheng et al, 1998)

of visual patterns and secondly are the intrinsic ambiguities in image perception. Real world images are fundamentally ambiguous with human perception of the image changing over time. Hence the more you look, the more you see. They go on to say that it would be wrong to think that a segmentation algorithm should output only one result.

According to Ishii and Kyuma (2001) image segmentation is a mixture of complex sub-problems and they highlight perceptual organisation as a concept to describe that it consists of many stages and that different types and levels of image features are involved. They propose a segmentation method at the primitive level that consists of processes to produce a representation whose dimensionality is lower than real objects.

Since real world images are the subject of their work, Ishii and Kyuma (2001) say it is not possible to strictly define connected areas or continuous image features. Even though there may be real segments in 3D physical space, the optical mapping to 2D image space includes very complex noise and existing boundaries are mapped incompletely. They go on to categorise the incompleteness into optical, structural, and higher level types.

They assume the input to their segment scanning network is a smoothness value of some locally determined image feature. $\{e(x)\}$ is the map of image feature continuity where x denotes spatial location and e is signal level information. Given $\{e(x)\}$, Ishii and Kyuma (2001) intuitively define that two neighbouring locations should be in the same segment when the absolute values in the continuity map are both very small. This means that $\{e(x)\}$ represents the boundary of segments. This rule of neighbours may be applied to the next neighbours until all the locations in a segment are bound. A scanning wave is used to solve the incompleteness problem where the wave should not pass through small holes in the boundary.

Huang et al (2000) state that methods of motion segmentation can be grouped into two broad classes. One solves the problem by letting multiple models simultaneously compete for the description of the individual motion measurements, whilst the other excavates out the multiple models sequentially by solving for a dominant model. Both encounter difficulties, the first at the determination of the number of models or uncertainty of mixture models and the latter in the case of absence of dominant motion.

Huang et al (2000) use the dominant motion model described by Black (1996) to calculate both foreground and background motion and then use the watershed algorithm to segment the foreground from the background.

The dominant motion method begins by constructing a three level gaussian pyramid with the coarse level initially set to zero. Huang et al (2000) chose 10 iterations and the estimated parameters for interframe motion, affine flow model, and scale are used to warp the first image to the second image. The simultaneous-over-relaxation method lowers the scale parameter in each iteration (within bounds) leading to an estimation of the dominant motion.

The watershed technique of static segmentation considers the gradient magnitude image as a landscape where brightness levels correspond to the elevation. Areas where a raindrop would drain to the same minimum are considered to be catchment areas and the separations between catchments are the watersheds. As this technique can lead to many small partitioned subregions the catchment areas are iteratively merged based on thresholding the difference of two adjacent subregions mean values.

Another approach to the segmentation problem is described by Voles et al (2000) where maritime vessels and other static nautical objects are extracted from the sea to aid the recognition and tracking process. Three algorithms are introduced to perform variable size image window analysis, statistical analysis by reclustering, and region segmentation. The first determines a set of overlapping image windows and calculates four statistical measures for each window. The measures used are more often seen in texture analysis having first generated a grey level co-occurrence matrix for the image. Voles et al (2000) do not calculate the matrix and apply the energy, entropy, homogeneity, and contrast measures directly to the image pixels.

The statistical analyser uses a method of iterative reclustering of the feature space to determine the centroid of vectors that represent the sea, the main feature in the scene. The region segmentor then calculates the Mahalanobis distance between the values of the feature centroid in each window to identify outliers from the mean. The outliers are potentially regions

containing inhomogeneities that may indicate the presence of a rigid object. These extracted regions effectively form the segmented regions of interest in the image, which are then identified to the user.

Figure 2.10 shows the variable sized image windows overlaid onto the image of the sea. Figure 2.11 shows the Mahalanobis distance transformed back onto the same image and figures 2.12 and 2.13 show the result of the segmentation process on two different maritime scenes. The method correctly segmented the yachts in figures 2.12 and 2.13 95% of the time and the buoy in figure 2.13 85% of the time. Voles et al (2000) conclude by saying that one advantage of this method is that it does not rely on any change between consecutive frames, it uses only the current image to perform the segmentation. The main drawback of this process is the requirement for the threshold value for separating the main feature from outliers in the Mahalanobis distance calculation to be set manually prior to segmentation.

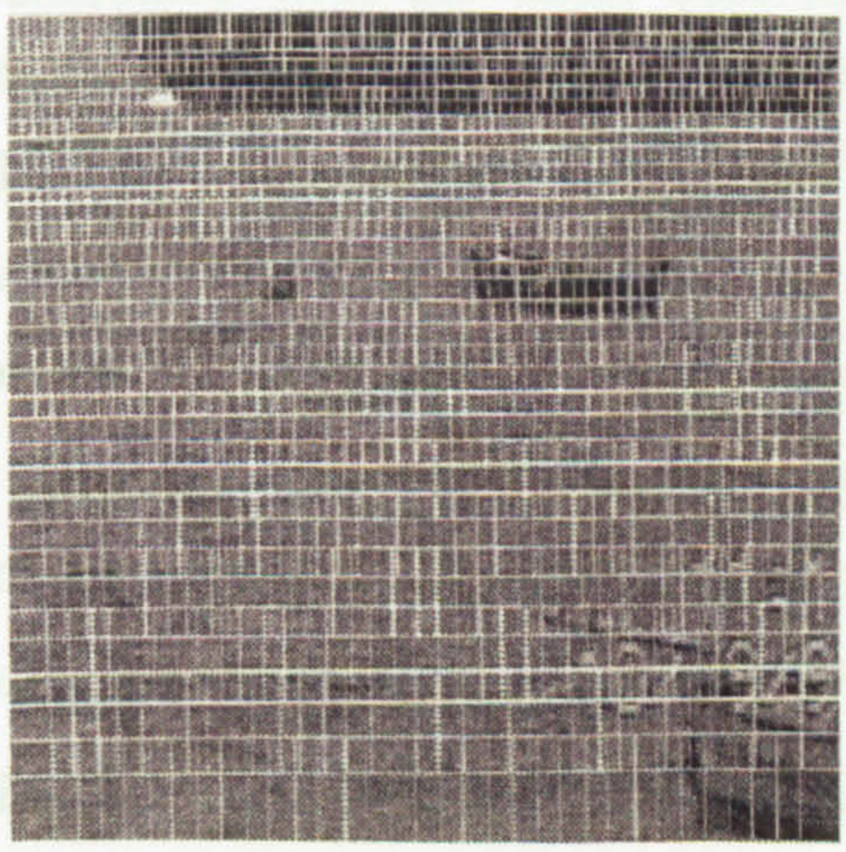

Figure 2.10 – Variable sized windows (Voles et al, 2000)

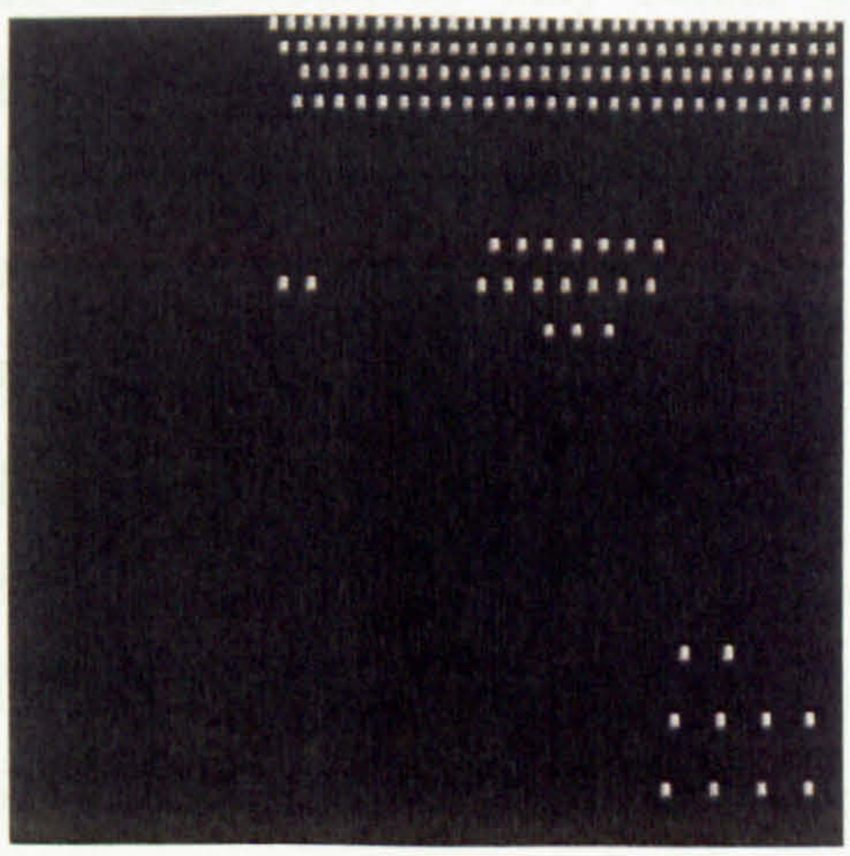

Figure 2.11 – Mahalanobis distance (Voles et al, 2000)

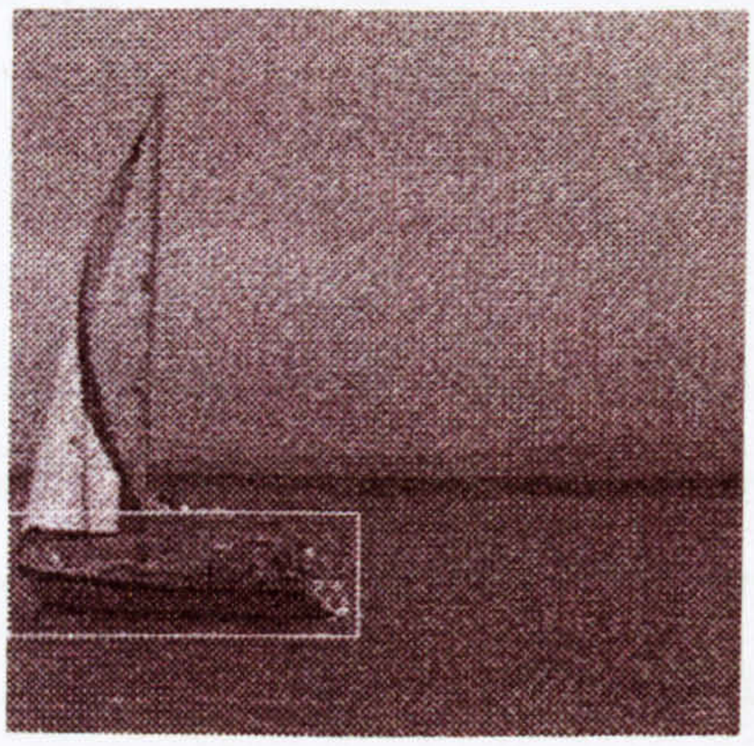

Figure 2.12 – Segmentation output 1 (Voles et al, 2000)

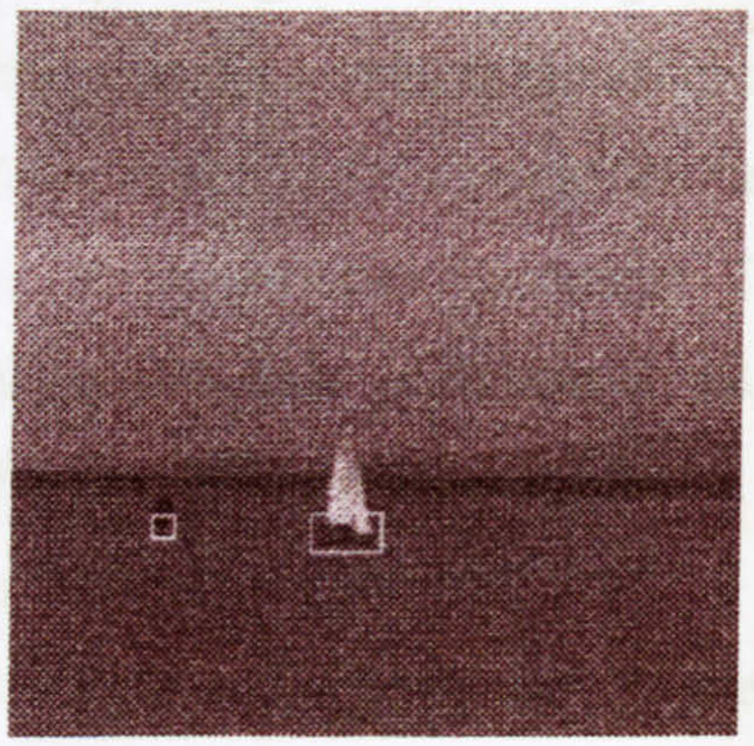

Figure 2.13 – Segmentation output 2 (Voles et al, 2000)

A feature interval graph is used by Mills and Novins (2000) for segmentation in long image sequences. Their representation shares many properties with the Kalman filter (Kalman, 1960). It is recursively computed, relying on only the current and previous frames; it is efficient to compute; and it is robust with respect to measurement uncertainty.

The feature interval graph is constructed from three-dimensional features identified in each frame. These features may be located and tracked using established techniques such as the Harris feature detector, stereo, and motion correspondence. In particular, here the Harris feature detector and stereo and motion correspondences were established using bipartite graph matching techniques. The feature interval graph is initialised with the first observation. A vertex is added to the graph for each feature in the scene, and an edge links each pair of distinct features. Associated with each edge is a measurement of the three-dimensional distance between the features, computed using interval arithmetic. This particular approach to calculating the location of features and the distances between features has been used to account for any uncertainty in the measurement of the feature locations. Each subsequent frame of the image sequence gives another set of observations and distance measurements.

There are two main tasks to be carried out – combine information from multiple observations, and account for missing observations. At each frame a new measurement may be made for the distance between each pair of points. These distances are represented as intervals and are combined using an intersection operator. The new distance stored in the graph is the intersection of the old distance and the latest observation.

The task of accounting for points that appear, disappear, and reappear in the scene is dealt with by adding a new vertex to the graph when a new feature appears in the scene. This new vertex is linked to all other visible features vertices and the edge distances are initialised with the current distance measurement. When a feature disappears the portion of the graph relating to it is frozen. The location of the feature is estimated but the edges incident on the corresponding vertex are not changed. Features that disappear are not immediately removed from the graph as they may reappear. In the case of an occluded feature reappearing, information gathered prior to the occlusion is combined with the latest observation. Edges observed only before the occlusion retain their distance information, and those observed only after the occlusion use the new information.

To segment the scene Mills and Novins (2000) propose an object definition based on triangles in the feature interval graph. Triangles correspond to small rigid substructures in the graph and lead to a segmentation method, which is robust with respect to a small number of missing or spurious edges in the graph. To construct a triangle-based segmentation an auxiliary graph is constructed from the feature interval graph. The vertices of the triangle graph correspond to triangles in the feature interval graph and two vertices are linked by an edge if the corresponding triangles share an edge in the feature interval graph. Component analysis is then applied to segment the triangle graph and the segmentation transferred back to the vertices of the feature interval graph.

The feature interval graph currently takes several frames to converge. Although not necessarily a problem from a computational point of view, practically this could be a concern. Mills and Novins (2000) suggest that the convergence of the feature interval graph could be accelerated by applying heuristics to remove edges from the graph, or to reduce features' initial connectivity.

2.3.2 Texture Segmentation

Texture has been used to successfully segment images for many applications. It is of interest to this work because the sea, as viewed by human observers, has a perceived texture. If this texture could be described it could be segmented from the remaining objects in the image. Many types of texture features exist including those based on grey level co-occurrence statistics such as homogeneity, contrast, entropy, and energy (Bassman and Besslich, 1995), fractal dimension (Chaudhuri and Sarkar, 1995), Gabor filters (Dunn and Higgins, 1995) and more recently, association rules (Rushing et al, 2002).

For Chaudhuri and Sarkar (1995) the fractal dimension (FD) of a bounded set A is a real number used to characterise the geometrical complexity of A which can be estimated using the concept of self-similarity. The bounded set A is considered to be self-similar if A is the union of N non-overlapping copies of itself scaled up or down by a ratio r. The fractal dimension is then given by the relation specified by Mandelbrot (1982) in equation 2.1:

$$FD = \log(N) / \log(1/r) \tag{2.1}$$

Chaudhuri and Sarker (1995) go on to use the FD for six variants of the image to calculate six features to discriminate the dominant orientation and degree of anisotropy of textures present in the image. This is done because

the combined effect of these two aspects may be the cause of different textures having the same FD.

The six features are then smoothed to reduce the misclassification that may occur in the inner regions and at region boundaries. A *K*-means clustering algorithm is then used to give the final segmentation of the image. This method has been tested on texture images from the Brodatz album (Brodatz, 1966) and also on coarse grain cemented sandstone as shown in figure 2.14. The problem was to separate the quartz particles from the iron cementing materials present in the sandstone. Compared with human judgement the FD technique correctly segmented 91.7% of areas as shown in figure 2.15.

Figure 2.14 – Original Sandstone (Chaudhuri and Sarker, 1995)

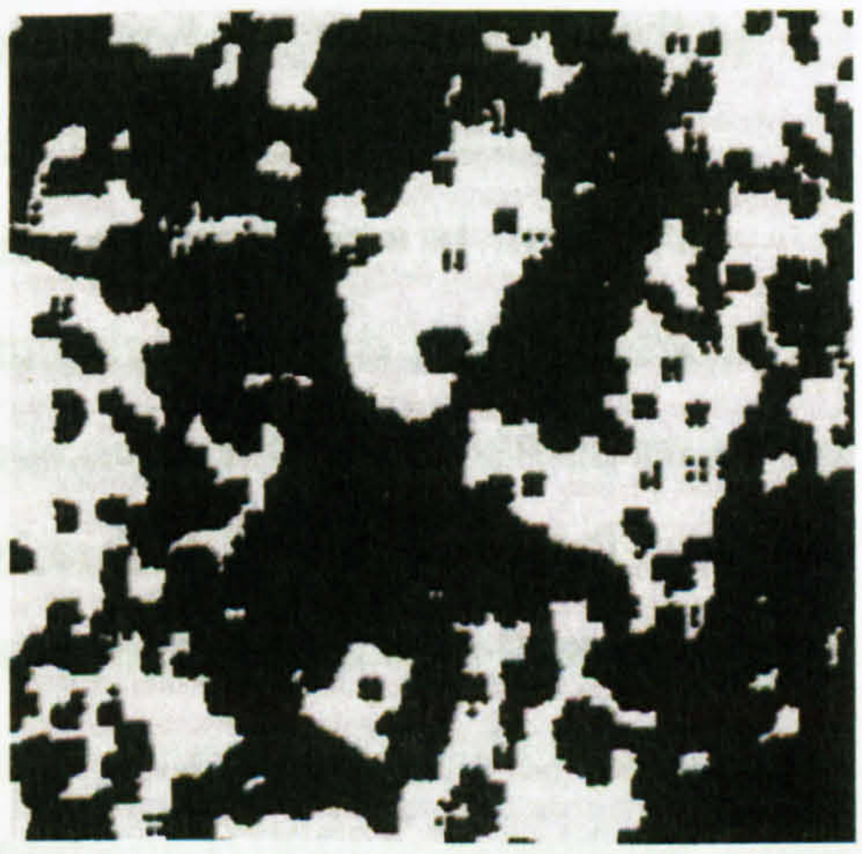

Figure 2.15 – Segmented Sandstone (Chaudhuri and Sarker, 1995)

Gabor filters are considered by Dunn and Higgins (1995) for the segmentation of textured images. The principle behind Gabor filters is to use a number of different linear filters operating in parallel, in a combination of the spatial and frequency domain, to segment the image. No effective systematic method existed for designing Gabor filters for the segmentation of natural textures, which have wide variability, so Dunn and Higgins (1995) considered this problem. They developed a method based on a decision theoretic formulation using representative samples of the textures

of interest to detect the discontinuities at texture boundaries to segment the image. At present the filter-selection algorithm is limited to bipartite textures. Using a single filter to segment more than two textures might not be as effective as integrating the outputs from multiple filters.

Rushing et al (2002) have developed a texture feature based on association rules to segment clouds in satellite images. They define texture as a surface property of an object that is reflected in a digital image as a local pattern of intensity variation. An association rule captures information about items that are frequently associated with each other. In texture analysis the aim is to capture information about how pixels in a given neighbourhood are related to each other and in order to use association rules to capture this information the image needs to be modelled in terms of items and transactions.

An item is a pixel in the neighbourhood of a given root pixel (the central pixel of an n x n neighbourhood). The item is specified in terms of its mapping from the root pixel and is given by (X, Y, I). A transaction is a set of items associated with a root pixel. A root pixel can have exactly one transaction and this transaction describes the texture feature.

Rushing et al (2002) go on to describe methods of mining these association rules from the images and how to use the rules to segment the image. They identify problems in capturing local image structures in grey level images using association rules. Instances of a texture pattern that might appear similar are rarely identical, there are usually small variations in the structure and intensity values of the corresponding pixels. This leads to common structures present in images not being recognised by the mining algorithm. This problem is alleviated by a method of quantization (reducing the number of grey levels) so that the structure is preserved but small variations in intensity are ignored.

This method is shown to work well for supervised segmentation applications where sample images are available in advance for each texture class to allow for the determination of the association rules. Rushing et al (2002) conclude that the segmentation problem can be viewed as a series of classification problems followed by some post-processing.

Puzicha et al (1999) consider two problems in the unsupervised segmentation process:

- The modelling problem – The requirement of a precise mathematical notion of homogeneity or similarity between image primitives in order to formalise the clustering problem.
- The computational problem – The derivation of an efficient clustering algorithm. The selection of a suitable method is tightly coupled to the chosen similarity measure and its underlying data representation.

Their goal is to group pixels or small image patches such that segments of identical texture are obtained. They use a general method for grouping feature distributions that involves performing grouping directly on the histogram data, which they also call distributional data, as opposed to other classical clustering approaches such as a *k*-means algorithm on vectors generated from local features. The histogram represents features occurring in an image neighbourhood around each pixel. A class model of textures is then generated characterised by the specific distribution of features found in each neighbourhood. The optimal class assignment of a pixel does not depend on the location of the pixel but on the locally measured features.

Puzicha et al (1999) go on to say that for most applications there is prior knowledge about inadmissible or unlikely texture configurations. As a quality criterion they propose to count for each pixel how many pixels of the same texture class are found in a 7x7 neighbourhood. When the number falls below a threshold the texture label configuration is considered to be

less likely. The results of this method for highly textured real-world images of wildlife are shown in figures 2.16 and 2.17.

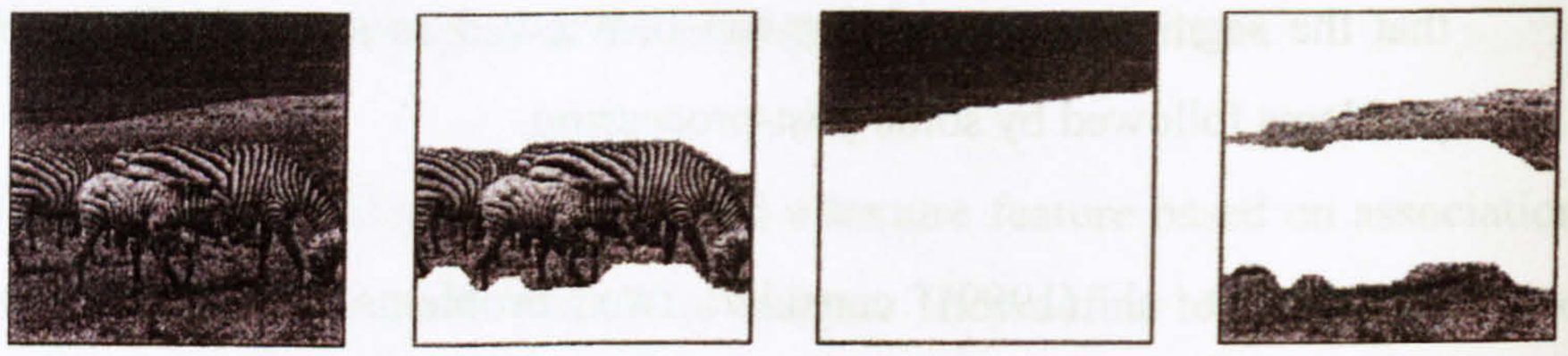

Figure 2.16 – Segmentation of Zebra into 3 texture classes
(Puzicha et al, 1999)

Figure 2.17 – Segmentation of Leopard into 2 texture classes
(Puzicha et al, 1999)

2.3.3 Motion Detection

Motion is described by Aach et al (2001) as the temporal changes, which are introduced by true scene changes, and as temporal changes in image intensities by Techmer (2001). Motion is more specifically defined by Strehl and Aggarwal (2000) as arising from moving objects viewed by a fixed camera against a static background, or by a moving camera depicting a still background, or by multiple motions, the combination of a moving camera and moving objects. Wang et al (2000) also observes that the differences between consecutive frames are usually created by a combination of camera motion and the movement of objects. Aach et al (2001) qualify their definition by adding that true scene changes does not include varying

illumination or noise (Techmer, 2001) saying that intensity changes can also be caused by changes in perspective and deformation.

Estimating motion has been approached in many ways with frame differencing (Franchi et al, 1996) (Xu and Ellis, 2002) being one of the most widely used. Despite its popularity frame differencing is not without its drawbacks. Lee et al (2001) state that frame differencing has the disadvantage that it can generally only be used with a static camera. Owens et al (2002) say that it is common for an object segmented by frame differencing to fragment due to parts of the object matching the greyscale of the background. And Ellis et al (1991) say that image differencing places considerable reliance on maintaining a reliable reference image. Voles et al (1999) apply frame differencing to maritime images the result of which shows that much of the sea is in motion in addition to any moving objects. Figure 2.18 shows an original image from the sequence and figure 2.19 shows the differenced image. Determining a suitable threshold value for the differenced image however can eliminate much of the motion of the sea and successfully segment the image.

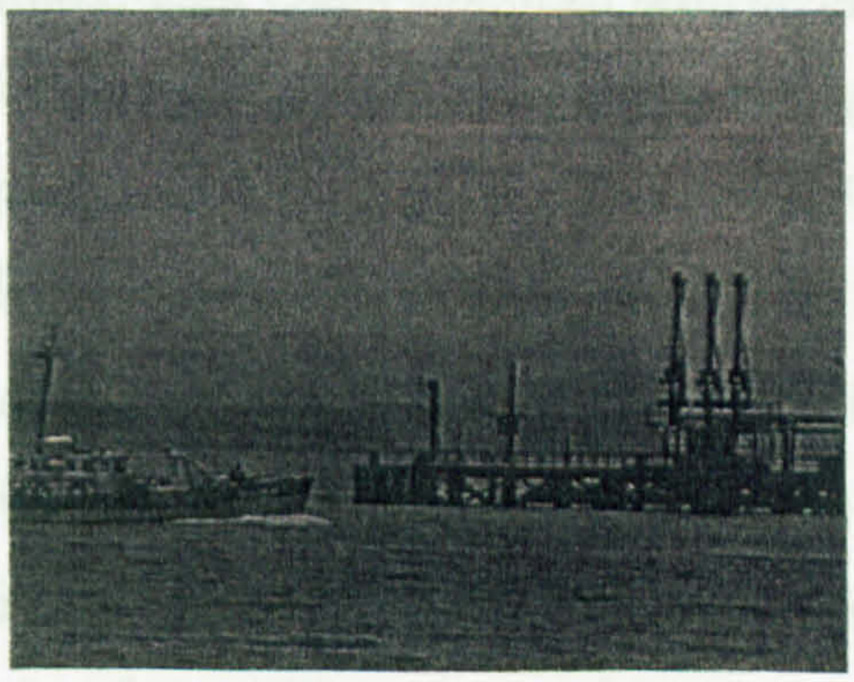

Figure 2.18 – Boat and pontoon (Voles, 1999)

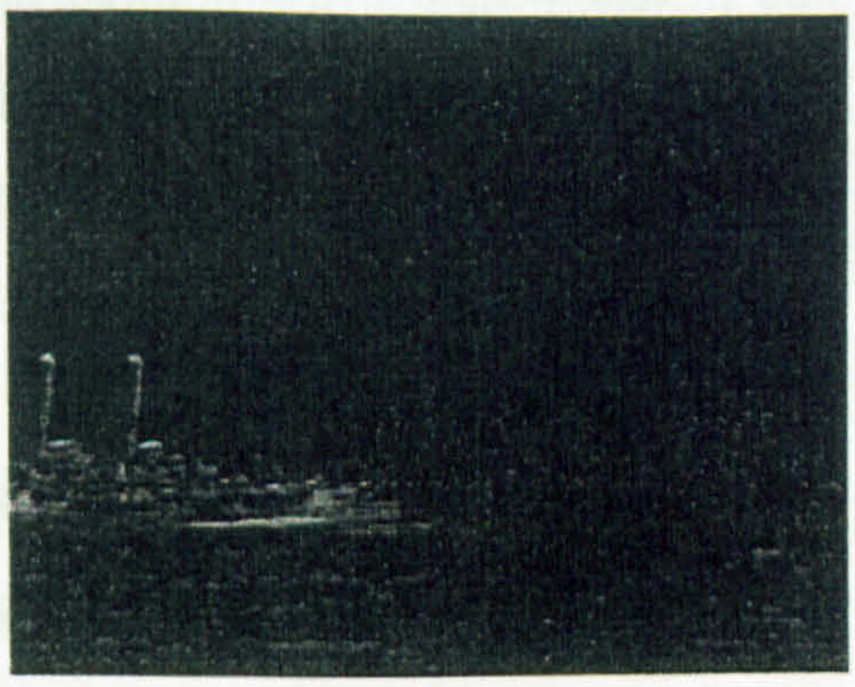

Figure 2.19 – Differenced image (Voles, 1999)

2.4 Object Tracking

Tracking an entity, feature or object over a sequence of frames requires the use of constraints based on the nature of the objects and their motion (Jain et al, 1995). Of the many methods developed for achieving this optical flow, implementations of the kalman filter, and the use of measurements of object parameters are discussed here.

Optical flow is described by Horn and Schunck (1981) as the distribution of apparent velocities of movement of brightness patterns in an image. There are a number of ways in which to calculate optical flow including feature-based estimation as used by Smith and Brady (1995) to determine objects in a sequence where the camera is looking at a vehicle moving along a road. The features used in this case are 2 dimensional features often referred to as corners. Guerrero and Sagues (1999) use a combination of optical flow and geometric features (straight lines) to determine camera motion whereas Mallot et al (1991) use a biologically motivated method of computing optical flow by combining correlation type motion detectors and local voting over the outputs of the detectors. Barron et al (1994) has evaluated the performance of a number of optical flow techniques and they report results for differential, matching, energy-based, and phase-based methods.

Stiller and Suntrup (1992) use a generalised parametric motion model where the motion of each object is described by a set of parameters and fixed transform. These parameters are estimated using two strategies, a slightly modified gradient-based algorithm and a region matching evolution technique. Raw sparse optical flow is used by Strehl and Aggarwal (2000) in a new probabilistic relaxation framework to iteratively perform robust multiple motion estimation while Techmer (2001) presents a contour-based approach of matching spatial contour points between frames for motion-based vehicle detection.

Marcenaro et al (2002) describes a tracking algorithm able to solve occlusion situations among moving objects in a scene (vehicles moving in a built up area). Their technique is based on a shape-matching algorithm initialised by using a linear Kalman filter. Using this technique it is possible to preserve object identities when they are partially occluded by a static or moving object.

They start by performing frame to frame differencing and then perform blob tracking based on the spatial relations between blobs in subsequent frames obtained by comparing each extracted region of interest and looking for those that overlap. Dependant on their positions in the image objects (vehicles) are labelled as NEW, OLD, or MERGED. The last case represents a warning that a dynamic occlusion has occurred between two or more moving objects in the scene. By labelling the region of interest as MERGED the information about the position of each single object is lost. The tracking algorithm does not take into account the object's movements and if the frame rate is too low objects in adjacent frames will not be overlapped and so will not be matched. Marcenaro et al (2002) say that in general this condition is not satisfied in outdoor video-surveillance systems because they track high speed moving vehicles far from the camera. The problem can however be solved by using an estimation technique for the prediction of the position of the region of interest in the next frame and for this they use the linear Kalman filter (Kalman, 1960).

During blob tracking a new Kalman filter is instantiated for each blob labelled as NEW. The filter is used to predict the position of the blob in the next frame. The extracted list of blobs in the next frame is compared with the predicted list of blobs from the previous frame. If a MERGED blob is detected, the system is not able to retrieve a new observation vector and the Kalman filter is updated only by using the previous state vector. This approach is only correct if the acceleration of the considered object is

constant. If the speed is not constant the prediction error of the Kalman filter increases and the system will fail to track the object.

To counter this problem Marcenaro et al (2002) propose a shape matching algorithm. The shape of an isolated object can be defined as the subpart of the change detection image within the associated bounding box. This binary shape image is stored for each moving object labelled OLD. Whenever an occlusion is detected the prediction is computed by using the Kalman estimator for each blob in the merged event and a correlation function is minimised. The output of the procedure is, for each blob, a vector that maximises the correlation between the stored shape and the merged change detection image. The output vector is then passed to the Kalman filter as the new measurement vector for the blob in the next frame. Marcenaro et al (2002) show results with the Kalman filter only failing to track objects that have non constant acceleration and succeeding in tracking when the shape matching algorithm is introduced. They claim an average object misdetection rate of 2% on the tested sequence.

Huang et al (2002) says that tracking objects in image sequences is an important task for vision-based control, human computer interaction, content based video indexing and structure from motion. They go on to say that visual tracking algorithms can be classified into two categories – feature-based methods and region-based methods and that the latter category can be subdivided into the view-based and parametric methods. The view-based method finds the best match of a region in a search area with a reference template whilst the parametric method assumes a parametric model of changes in the target image and computes optimal fitting of the model to pixel data in a region.

Huang et al (2002) uses the parametric method in their work on segmentation-based object tracking. They utilise a least squares method to

determine motion parameters and a Kalman filter for motion prediction. A method of image warping is them employed followed by region analysis to track the object between frames. They show some encouraging results for head and hand tracking but point out some disadvantages of their method. It assumes the object has been reliably detected in the first frame and they also found that the Kalman filtering does not always provide a correct prediction. This method also only works for moving objects.

Since Wang et al (2000) are interested in tracking moving objects they use the projective/bilinear camera model (Mann and Picard, 1997) to remove those differences caused by the camera as much as possible. They then go on to track objects with a rule-based algorithm using four sets of variables - the object trajectories, sizes, greyscale distribution, and textures.

The variables for object trajectory are the object centroid co-ordinates and its predicted future position, which is calculated using the centroid from the previous three frames and measures of initial speed and acceleration. By comparing the predicted position and real position it is possible to achieve trajectory-based tracking. Assumptions made by Wang et al (2000) here are that the object trajectories are close to straight lines in a few adjacent frames and that object acceleration rate is constant in these frames.

A further assumption is that for a given frame rate the size of the objects should not vary dramatically between frames. The dispersion variable is used to track objects based on size. The dispersion is calculated for each object using equation 2.2 (Wang et al, 2000) where (c_x, c_y) is the object centroid, O is the set of co-ordinates of the object area, and $p_{i,j}$ is the value of the edge image at position (i,j). The objects can then be tracked by comparing the dispersion variables.

$$disp = \left(\sum_{(i,j)\in O} \sqrt{(i-c_x)^2 + (j-c_y)^2} \cdot p_{i,j} \right) / \left(\sum_{(i,j)\in O} p_{i,j} \right) \tag{2.2}$$

They go on to say the greyscale distribution of an object does not usually change considerably given that the lighting condition stays relatively constant between frames. The variables they use are the mean of the whole greyscale range, the mean of the 10% pixels of largest greyscale range and the mean of the 10% pixels of smallest greyscale range. Greyscale based tracking is then achieved by matching the variables of the object in consecutive frames.

Wang et al (2000) notes that the surfaces of objects are usually not homogeneous with the greyscale variations on an object varying from object to object. These variations are reflected in the wavelet transform coefficients. The mean of the 10% of pixels with the largest values in the constructed 'edge' image roughly indicates the texture property of the object. The variable can be used in the same way as the others to track objects between frames.

Four sets of variables have been used as it was noted that there are extremes that violate the assumptions that had been made such that none of the sets of variables individually would accurately track the objects over time. To reliably track the objects each set of variables is calculated for an object, the differences found between the object in frame k and frame k-1 and those differences thresholded. These results are entered into a matrix of size m x n where m is the number of objects in frame k and n is the number of matched objects in frame k-1. To arrive at the best object match a number of rules are applied. First, at least three of the variable sets must be less than the threshold. If there is only one element in its matrix row and column that is greater than or equal to three then that object is matched and that row and column eliminated from the matrix. Secondly, is $e_{i,j}$ is the only eligible element in row i but if all other eligible elements in column j are not the only element in their corresponding rows j is chosen as the best match and that row and column eliminated. If, after applying these two rules, any

elements remain unmatched then a weighted sum is computed. Then, for each column, the row with the lowest weighted sum value is chosen as the best match. After this process there can still be some objects left unmatched. If this is the case they are either labelled as new tracks or ceased tracks.

Wang et al (2000) shows results for a toy vehicle sequence. Figure 2.20 shows frames 1 to 9 from this sequence. Figures 2.21 and 2.22 show the detected object areas for frames 4 and 5 and table 2.1 shows the tracking results. Objects tracked are given the same ID number so it can be seen that the car midway down on the left of the image has been tracked in the first five frames. The second vehicle, entering from the right, suffers occlusion in frame 1 – 3 and fails to be tracked but is successfully tracked in frames 4 and 5. Neither object matches the combined object in frame 6 however this new object is tracked through frame 6 – 8. In frame 9 the two objects are detected separately however they have been given new ID numbers so they have not been recognised as the same objects as tracked in frame 5. They conclude that, whilst their method works effectively, their goal was to identify and track objects quickly and the projective model for camera motion is somewhat computationally intensive and it cannot really deal with occlusion.

ASSET-2, a real-time image processing system developed by Smith and Brady (1995) takes a real life situation, a vehicle travelling along a road and monitors what is in front of it. If other vehicles can be seen they must be segmented from the background so their motion can be estimated and the necessary action taken.

(a) frame 1 (b) frame 2 (c) frame 3

(d) frame 4 (e) frame 5 (f) frame 6

(g) frame 7 (h) frame 8 (i) frame 9

Figure 2.20 – Toy vehicle sequence (Wang et al, 2000)

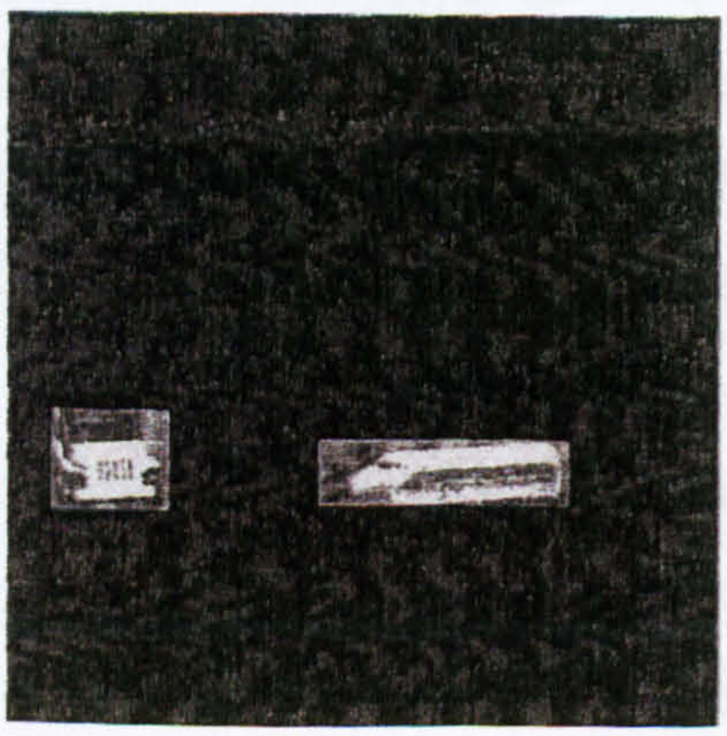

Figure 2.21 – Detected objects frame 4 (Wang et al, 2000)

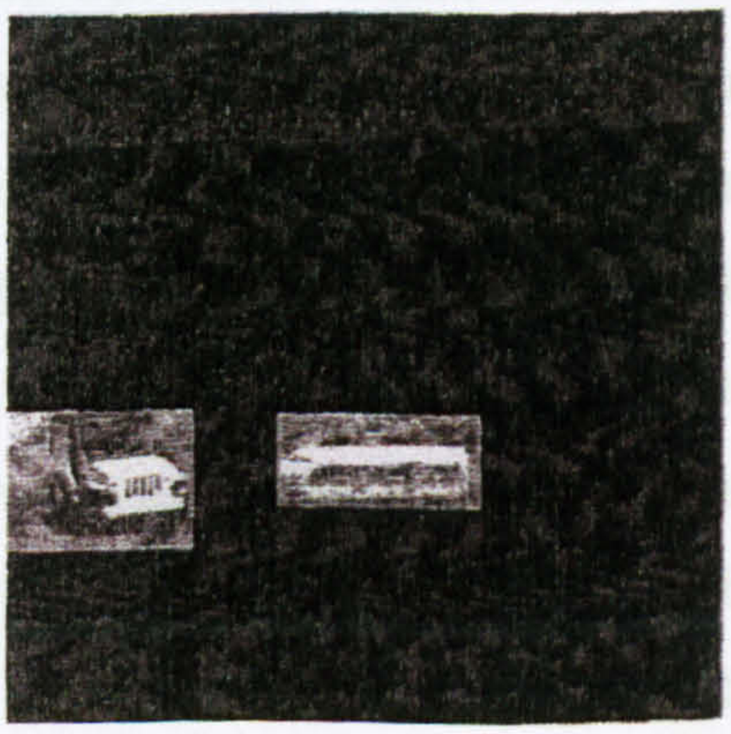

Figure 2.22 – Detected objects frame 5 (Wang et al, 2000)

Table 2.1 – Toy vehicle tracking results (Wang et al, 2000)

Frame No.	Obj. ID	Centroid	Obj. ID	Centroid
1	1	(53.2, 295.4)	-	-
2	1	(51.8, 303.4)	-	-
3	1	(53.2, 310.4)	2	(377.1, 336.9)
4	1	(66.5, 317.6)	3	(325.4, 338.6)
5	1	(79.7, 326.5)	3	(257.0, 338.9)
6	4	(135.5, 332.1)	-	-
7	4	(135.1, 334.8)	-	-
8	4	(144.2, 336.6)	-	-
9	5	(96.9, 313.5)	6	(283.3, 353.5)

The ASSET-2 system is built around feature-based image motion estimation. The features used are primarily two-dimensional 'corners', detected using either the SUSAN (Smith, 1992) or the Harris (Harris and Stephens, 1988) corner detectors, whilst edges are used to refine the results. These 'corner' features are used to find the optic flow in the scene. Smith and Brady (1995) state that corners were chosen to determine optic flow because they should contain as much information about the scene motion as is available at the places in the image where the process of flow recovery is most well conditioned and where the information is most relevant.

As soon as a corner has been matched once, the velocity estimate obtained allows a large reduction in the necessary search space for future matches. The motion model update filter used is a simplified two dimensional Kalman filter in which the search space is reduced over time and the model estimates are given increasing importance over time. The next step after flow estimation is to segment the flow list into clusters, which represent different objects in the world. The flow within each cluster is fitted to a linear space dependant model giving a total of six parameters for each cluster using a least squares fit. Newly found clusters are now matched with the clusters in the filtered list using time-symmetric matching. Both

elements of a potential matched pair must prefer their proposed partner more than any other in the partner's list for the match to be valid. The attributes used for the forward matching are the cluster's motion model and the shape of the cluster.

Figure 2.23 shows 15 frames from the ambulance sequence used in the work by Smith and Brady (1995). Figure 2.24 shows the flow vectors for one of the frames and figure 2.25 shows the clusters formed from the flow vectors. Figure 2.26 contains 12 frames from the final output of the ASSET-2 system showing how it successfully tracks objects before, during, and after occlusion.

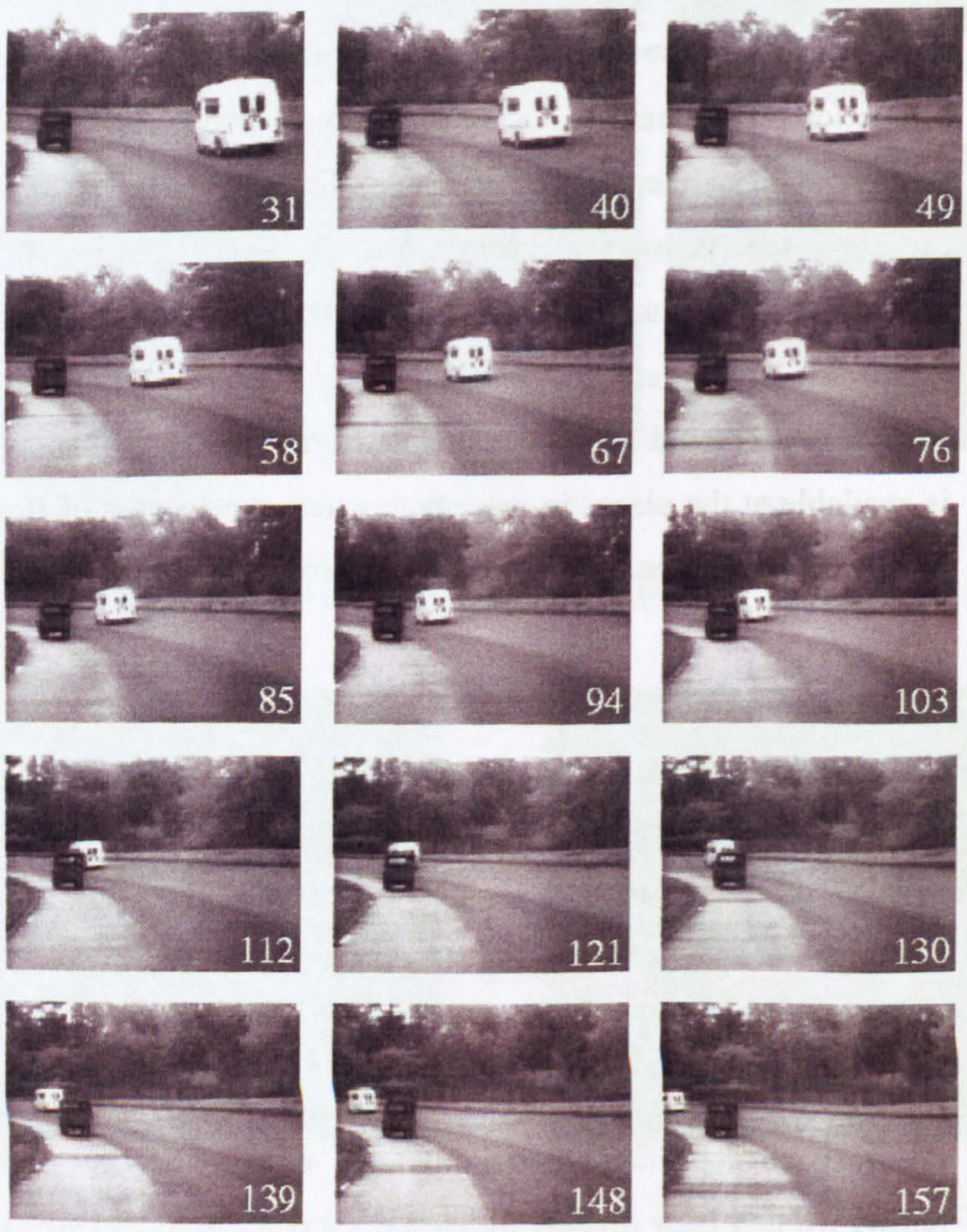

Figure 2.23 – Ambulance sequence (Smith and Brady, 1995)

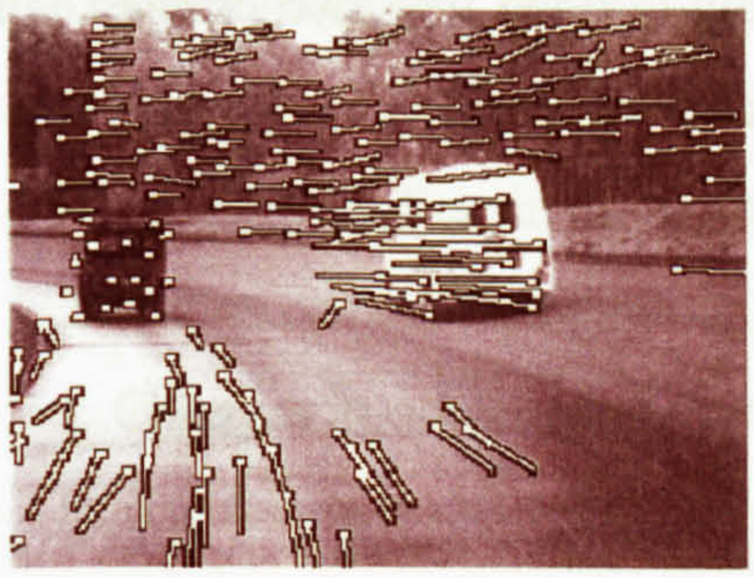

Figure 2.24 – Flow vectors (Smith and Brady, 1995)

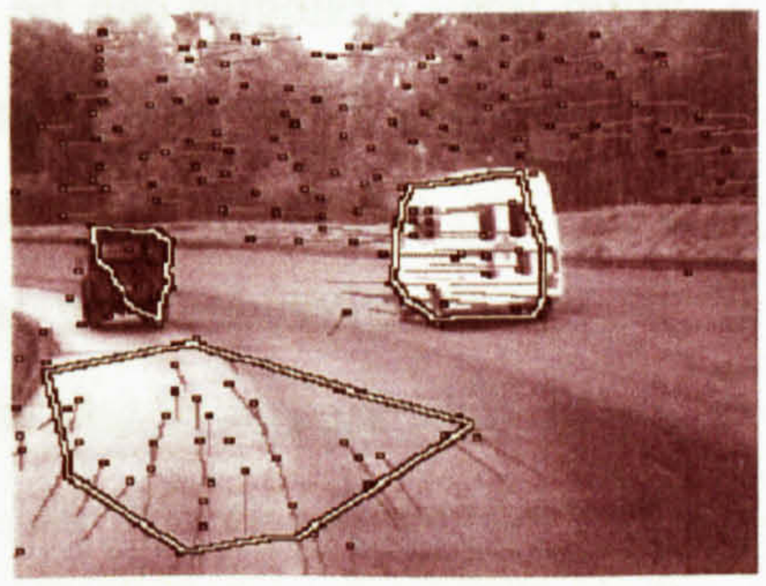

Figure 2.25 – Segmented clusters (Smith and Brady, 1995)

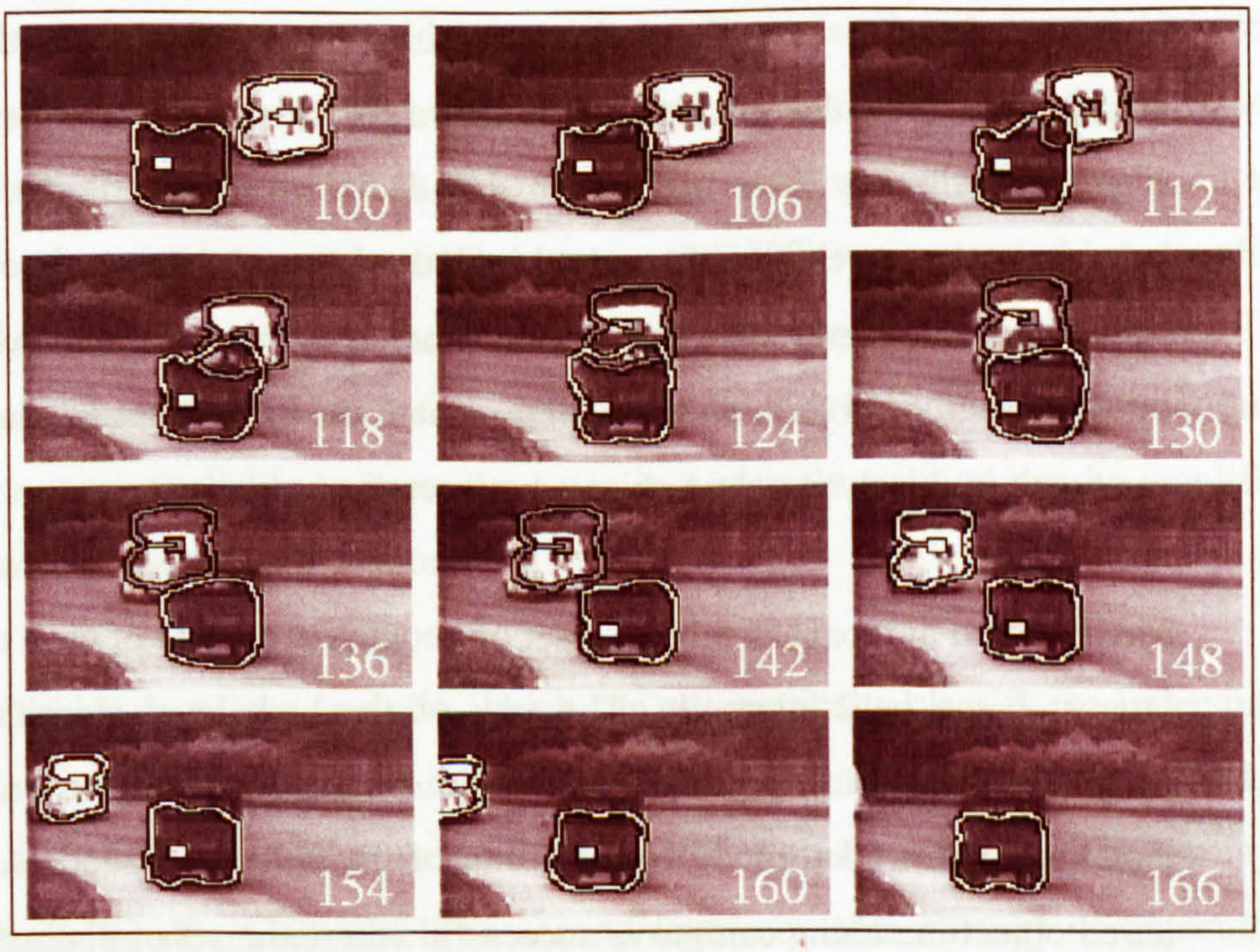

Figure 2.26 – ASSET-2 tracking output (Smith and Brady, 1995)

2.5 High-Level Vision

High-level vision is that part of any vision system that is concerned with recognition and interpretation (Gonzales and Woods, 1992). It takes information about the image from previous processing steps and also some knowledge provided *a priori* regarding the scene being analysed.

High level strategies can be broadly divided into two classes depending on the way the system represents the outside world. The classes are:

1 – Knowledge-based

2 – Model-based

Knowledge-based systems use information about the objects and their relationships between each other and the scene domain. Information is gathered about the objects during the intermediate processing stage from the identified features or regions. Predetermined knowledge about the scene domain and the rules governing the interactions between objects and the scene domain are then used to make decisions regarding the objects behaviour. Approaches to this might include the use of Semantic or Neural Networks. Knowledge-based systems can suffer in situations where the scene domain or objects do not exactly match the criteria set for them. This can result in falsely matched objects between frames.

Model-based approaches rely on a set of geometric models of each object of interest being held internally. The task of the high level algorithm is to extract such information from the image data as necessary to enable the matching of the identified object to one of the 2D or 3D models. Model-based object matching can prove successful where the set of distinct objects of interest is small and the objects occupy a large proportion of the image as in the vehicle tracking systems developed by Koller and Nagel (1993) and Ferryman et al (1995). These perform less well where sufficient information to enable matching to the models cannot be extracted (Ellis et al, 1991).

Several factors can impact both knowledge and model-based approaches to high level vision by changing the appearance of objects between frames.

These factors include:

1 Changes in illumination both inter- and intra-frame

2 Object occlusion including partial, full, and self occlusion

3 Changes in object orientation effecting its size

In 'Semantic Interpretation of Object Activities in a Surveillance System', Lou et al (2002) describe a framework for semantic interpretation of vehicle and pedestrian behaviours in visual traffic surveillance. They say the main problem in image or video semantic interpretation is the construction of a mapping from images or video into the human's conceptual space. Trajectories are often used in semantic interpretation of dynamic image sequences, and in the work by Lou et al (2002) are generated by recording the position, speed, and direction of the target in each frame. The acceleration of the targets is not recorded, as they believe this information to be very unreliable because the noise in a target's position can be amplified in acceleration information.

Trajectory pattern analysis, which can automatically classify the trajectories into several patterns, is an important method for activity interpretation. Lou et al (2002) designed a classification tree with three layers. The top layer contains all trajectories. The second layer contains clusters of trajectories based on spatial information and the third layer contains classes of trajectories based on dynamic information.

A distance formulation similar to the Hausdorff distance is used to measure the spatial similarity between trajectories before a variation on a C-mean clustering method is used. For the mid layer spatial clustering Lou et al (2002) select a threshold ρ. Then, a subset of the trajectories x_k is chosen as the initial centres in such a way that every two trajectories x_i and x_j in the subset satisfy $D_c(x_i,x_j)\geq\rho$ where D_c is the spatial distance between x_i and x_j.

For clustering into classes using dynamic information at the bottom layer of the tree the distribution density of every trajectory sample is evaluated by counting the number of trajectories around it while the trajectories which have heavy densities are selected to form the initial centres. All trajectory samples are classified to the class whose centre is the nearest one in all classes.

With the classification tree built action analysis takes place. To analyse the actions Lou et al (2002) use a trajectory segment analysis method based on a hidden Markov model similar to that used by Fraile and Maybank (1998). Each trajectory is divided into several segments and one of four actions is assigned to the tracked target in each segment. The actions are: Move Forward, Turn Right, Turn Left, Stop. These actions are assigned based on a curvature value for the segment $\kappa = \omega / \nu$ where ω is the mean of angular speed and ν is the mean of translation speed. A simple thresholding operation is then used to assign the actions as shown in table 2.2.

Table 2.2 – Thresholding operation (Lou et al, 2002)

Action	Curvature value
Move Forward	$-0.1 < \kappa < 0.1$ and $\nu > 0.5$
Turn Right	$\kappa > 0.1$ and $\nu > 0.5$
Turn Left	$\kappa < -0.1$ and $\nu > 0.5$
Stop	$\nu < 0.5$

Finally Lou et al (2002) introduce a simple grammar to generate natural language descriptions based on the activity patterns obtained. They assume that in most surveillance scenarios the questions most often asked are "Who does what at where? And How?" and so use only one grammar rule:

(The Obj) (Action) in (The place name) [at (high/middle/low) speed]

The contents of the square brackets are optional and the contents in parenthesis should be substituted with information provided by the classification tree. This then yields a natural language description such as:
'Vehicle 1 parked in parking lot'

The system does not output the semantic description every frame, only when one of the following conditions is satisfied:

1) A new action is happening
2) The target is entering a new region
3) An abnormal event is happening

The methods described by Lou et al in (2002) have been demonstrated on the real world scene shown in figure 2.27. When a car enters the view (identified in red) and then is parked the system gives the natural language description. The results show the system to be successful at automatically learning activity patterns and generating semantic interpretations for tracked targets. However, the system does suffer from the very limited grammar rule database used.

Semantic networks are also used by Shih and Huang (2003) for the interpretation of baseball video. They develop several semantic networks to model the different semantic events in baseball video such as view, field, zooming, regular-panning, and fast-panning. A Bayesian belief network training procedure is applied to the semantic networks, which are then used to interpret the semantic meanings of the different events in the video.

The overall structure consists of three layers: the category layer at the top, the mid level semantic layer, and the low level feature layer. The lowest level of the framework consists of an object analyser that finds the existence of the main object, the objects sizes and number and a texture analyser that describes the background in terms of the entropy and edge histogram. Based on this low level information the semantic network 'view' describes

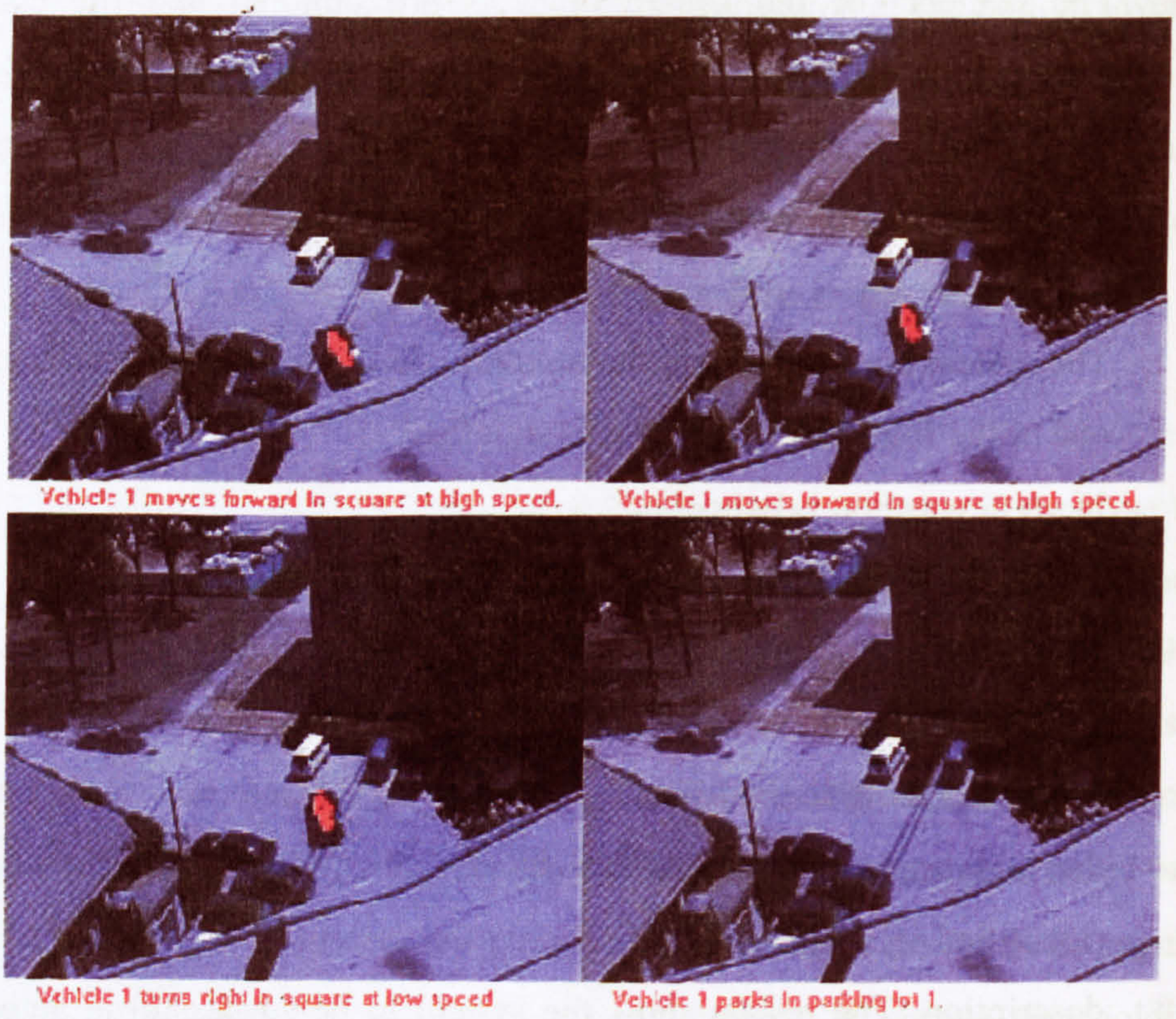

Figure 2.27 – Natural language descriptions (Lou et al, 2002)

the video as 'distant view' or 'close-up view', whereas the semantic network 'field' describes the video as 'infield' or 'outfield'. Figure 2.28 shows the semantic network for 'view'.

The mid level of the structure, which relates to motion activity, generates a displacement vector that can be used by the mid level semantic network to demonstrate the certainty of three different mid level semantics: fast-panning, regular-panning, and zooming. At the top of the multi-level hierarchy are the root nodes representing the certainty of the six different categories: Event Occur, Overview, Runner, Defence, Pitching, and Batter. The results of the mid-level semantic feature are shown in table 2.3 and for the categorisation level in table 2.4. It should be noted that each feature required many thousands of frames of training data to give these results.

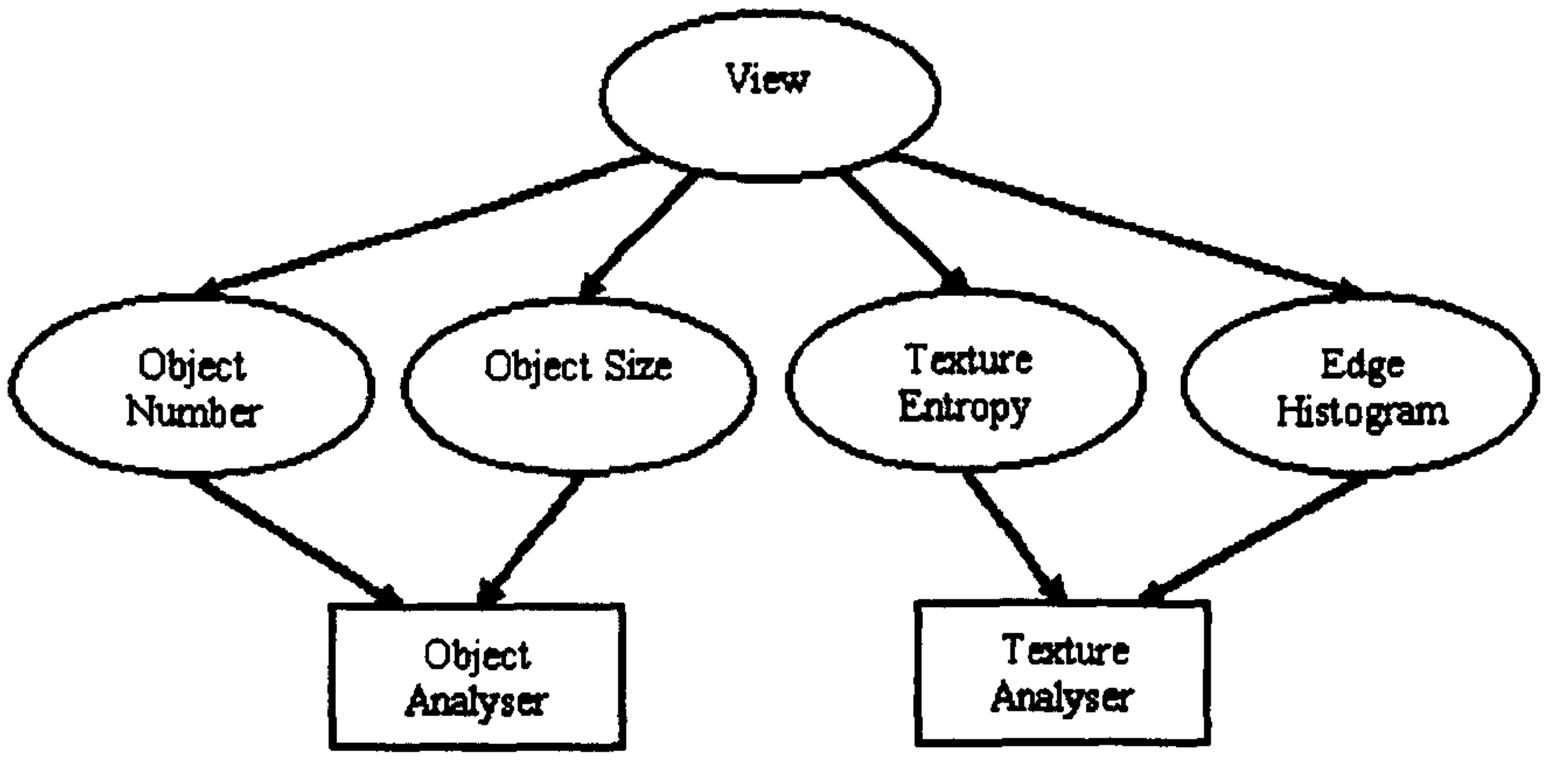

Figure 2.28 – Semantic Network for 'View' (Shih and Huang, 2003)

Table 2.3 – Mid-level semantic feature (Shih and Huang, 2003)

Mid-level Semantic feature	Training Data (frames)	Testing Data (frames)	Success rate (%)
Field	12,819	4,161	85.77%
View	17,995	7,777	80.01%
Zooming	11,106	30,238	82.20%
Fast Panning	10,429	27,592	79.24%
Regular Panning	9,724	31,183	74.82%

Table 2.4 – Result for category level (Shih and Huang, 2003)

Video Class	Detection Accuracy	False Alarm
Overview	96.23%	26.79%
Runner	97.20%	41.49%
Defence	96.55%	28.07%
Pitching	98.70%	38.92%
Batter	98.70%	43.60%

Semantic networks have also been used for different applications. Maderlechner and Mayer (1994) use them to collect geographical information from scanned maps whilst Teal (1997) uses a semantic network to interpret road traffic scenes in order to build up a map of the scene.

Di Bona et al (2001) present an approach for the classification of brain tissue density based on a hierarchical neural network applied to three dimensional brain volumes. Classification is obtained with a multi-step approach. In the first step, the region of interest is extracted from the background of each tissue slice in order to improve the efficiency of the approach. All voxels (three co-ordinates identifying a pixel by an x,y co-ordinate and a z co-ordinate identifying the tissue slice number) inside the region of interest are processed to compute a set of features which will be used to obtain the classification of the voxels themselves.

The classification consists of two-levels. The lower level performs the classification of the single features extracted from each voxel whilst the higher level takes these results and performs a final classification. The lower level classification is performed by a number of modules equal to the number of features extracted. Each module is a one-layer neural network where every output neuron is connected to every input neuron and every output neuron is also connected to other output neurons that are spatially close. Each network is trained for the classification of a specific feature. Weights are defined between input and output neurons and can change during the training phase. The weights are modified so that the network is able to recognise the input features and thus specialise it for a particular feature's input.

The number of neurons in each network is initially greater than the number of density classes and is then optimised by applying a clustering algorithm

based on a distance criterion. The output of each network is the classification of the voxel, depending on the specialised feature.

The classifications from these lower level neural networks are the input to the higher level classifier, which is performed by an Error Back-propogation algorithm. This algorithm has the following parameters: The network used is a feed-forward back-propogation network; The training function updates weights according to the back-propogation algorithm; The network is used with no hidden layer; The number of iterations is fixed to 1000. The format of the input is now an array containing, for each voxel, the classification given to it by each lower level classifier. During the training phase of the back-propagation the format is changed to that of a matrix where every row is a voxel array. A value is added at the end of each array representing the actual classification of the voxel, the target.

The supervised algorithm will be able to train the network propagating backward the resulting error. This error is calculated, starting from output neurons in the network, as a function of the neurons output and the difference between actual output and target. The output neurons propagate this error backward to each hidden neuron connected in the lower level network. The error is used to update the weights between neurons.

Di Bona et al (2001) state that the low number of inputs to the networks (four) allow the network to be trainable in a short time implying that if the number of inputs was high the training time would also be very high. Messer and Kittler (1998) reinforce this by saying that a 25 input, 25 unit hidden layer and 2 unit output layer neural network has 702 free parameters. They then say that it is very important that enough training examples are used to estimate these free parameters reliably. They state the generally accepted guideline is to have at least five to ten times the number of training patterns as free parameters. The example they give would therefore require a

minimum of 3510 training samples. Failure to have an adequate number of training samples can lead to the network learning the training set rather than building a statistical model of the process generating the data. Ultimately the network would be unsuccessful in classifying previously unseen patterns.

Messer and Kittler (1998) present an input selection algorithm that reduces the network complexity for searching through a large image database. An input saliency measure is defined which is the sum of the magnitude of the trained weights connecting that input to the hidden layer. The higher the sum the more important the feature input. The algorithm, which uses this measure to prune the network, requires a training dataset and a verification dataset. The user also needs to specify an acceptable error level.

The algorithm is then defined as follows:

- Step 1 - train the network on all the features in the training dataset
- Step 2 - calculates the saliency for each input node
- Step 3 - find the index of the feature returning the minimum saliency and remove this feature from the training dataset
- Step 4 - retrain the network using the reduced feature sub-set and calculate the error level on the verification dataset
- IF calculated error level <= acceptable error level then goto Step 2 ELSE add the removed feature back into the training dataset, retrain and stop.

The results presented were from the performance of the algorithm on a database of seismic images. The algorithm reduced the network size such that it had only 20 free parameters instead of 702. Messer and Kittler (1998) conclude by saying the reduction in network size also dramatically reduces the amount of time required to classify unseen patterns, which is important in their application.

Matesin et al (2001) uses a different approach to the high-level analysis of stroke lesions in CT brain images. Their rule-based expert system uses high-level knowledge about the anatomical structure and organisation of CT brain images containing stroke lesions together with various properties of the segmented image. The process used first determines the symmetry axis of the brain. Next the image is segmented into regions having uniform brightness. These regions are then labelled using the rule-based expert system.

The expert system works on facts based on rules, which are in the 'IF-THEN' form. If the left-hand-side of the rule is satisfied by the 'activation' of some facts the rule 'fires' and executes rules on the right-hand-side. There are several stages of labelling. In each stage only selected subsets of all facts can be activated or fired and this is controlled by a kind of priority in execution of rules. When a rule fires the stack that holds information about which rules can fire is updated with new activations. As long as there are rules with a higher priority, those rules with a lower priority cannot fire. The system presented by Matesin et al (2001) has 5 levels in the hierarchy as shown in figure 2.29.

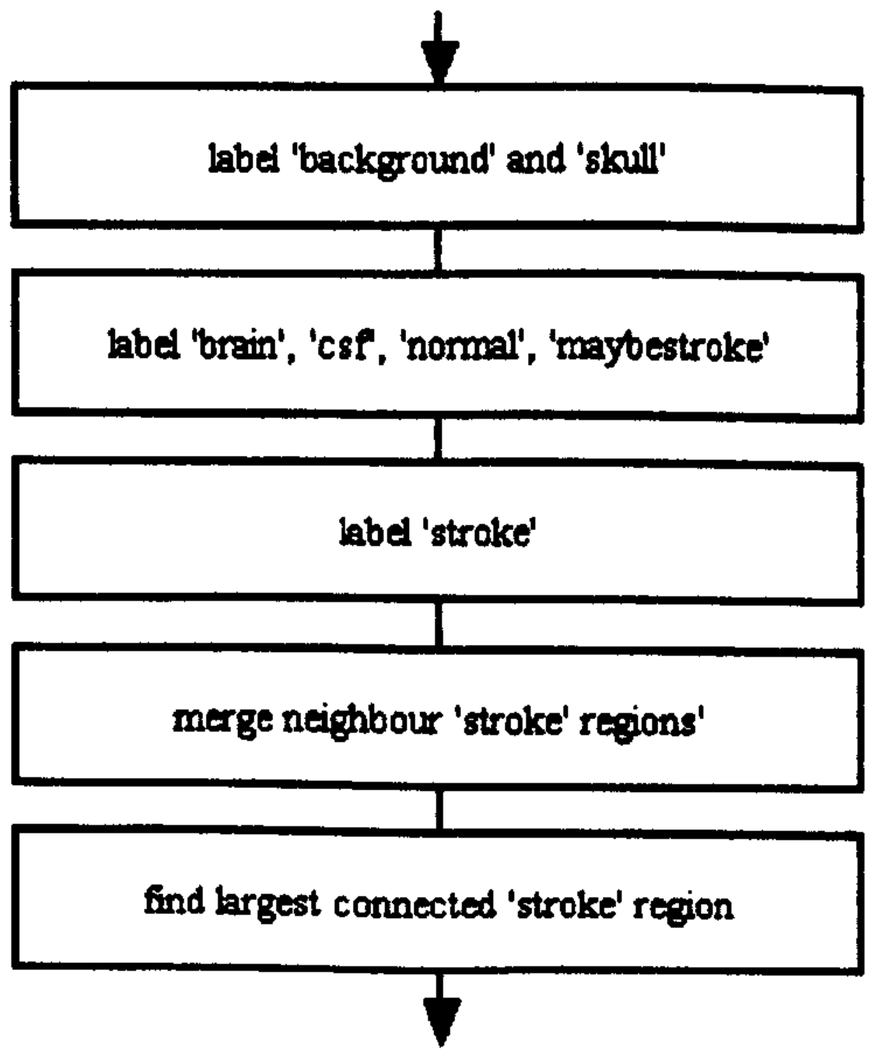

Figure 2.29 – Stages in the rule-based labelling Matesin et al, 2001)

Some of the rules from the first and third stages are:

- Rules for 'background'
  - Largest 'verydark' region is 'background'
  - 'verydark' or 'dark' region that has 'background' region for a neighbour is 'background' itself.
  - Region that has only one neighbour, and that neighbour is 'background', is 'background' itself.
- Rules for 'skull'
  - Largest 'verybright' region is 'skull'.
- Rules for 'stroke'
  - 'maybestroke' region that has no symmetric 'maybestroke' region is 'stroke'.

This rule-based approach provides flexibility in the experimentation of rules as it can be easily adapted by adding new rules or removing irrelevant ones without having to go through the lengthy process of retraining.

2.6 Discussion and Summary

It is generally accepted that the image capture process introduces noise to the image and that this noise approximates a gaussian distribution. In addition, and particular to maritime images, Voles et al (1999) tell us that waves cause noise that does not have a gaussian distribution. To reduce the effects of noise images are filtered using either a linear or non-linear filter. Linear filters use a weighted sum of pixels within a neighbourhood to calculate filtered pixel values such as the Mean filter described by Bassmann and Besslich (1995). Whilst these methods are good at removing gaussian noise they have the undesirable effect of blurring the image (Gonzales and Woods, 1992).

Non-linear filters attempt to remove noisy pixels whilst keeping edges intact. The Median filter orders the pixels within a window and selects the median of the ordered values as the new pixel value. Noise usually consists of spurious high and low intensity values and these are removed effectively by the Median filter. Senel et al (2002) point out the limitations of the Median filter saying that it can introduce spurious artefacts into the transformed image. They go on to say that the filter removes or attenuates image features that are smaller than the filter window and that in the presence of significant noise step edges can be blurred by the median filter contrary to its edge-preserving nature. Despite the reservations of Senel et al (2002) if the median smoothes the image whilst retaining edge detail it could be suitable for use with maritime images.

The Kuwahara filter (Kuwahara, 1976) appears to be a very effective filter for both noise reduction and edge preservation by making use of the advantages of both linear and non-linear filters. This is due to its non-linear selection of the window with the lowest variance from which it takes the linear mean value for the filtered pixel value. This gives excellent smoothing whilst retaining edge detail. This method would appear to be suitable for use with maritime images due to its smoothing abilities for the noise and the waves whilst retaining the edge detail of any objects present. Minato et al (1987) applies the Kuwahara filter repeatedly to the same image to obtain a fixed-point image. Their results are initially encouraging as they show very good edge retention and smoothing. However, they also show that the method is very sensitive to the size of filter window chosen. A further disadvantage is the increased computation time taken to apply the method many times to achieve the fixed-point image. Considering maritime images Minato et al's method could result in weak edges between areas of sea, such as waves, being strengthened and enhanced. This would make any segmentation process more likely to falsely segment areas of sea as object.

To the human observer the surface of the sea appears to have a texture. The use of a texture technique for the segmentation of maritime images should therefore be appropriate. Results shown by Puzicha et al (1999) and Rushing et al (2002) are encouraging as they use images of the natural world. However, for texture techniques to be successful the structure of the texture (the spatial relationship between texture elements) (Sonka et al, 1993) needs to be modelled. This is unlikely to be possible for maritime images as the sea can take on a wide range of appearances. Ellis et al (1991) support this as they say explicit models do not adequately describe most natural objects such as landscape because of the wide variety of form and shape. This is further reinforced by Pozzer et al (2001) in their work on the artificial generation of the surface of the sea. They state that modelling natural phenomena is one the most complex tasks in image synthesis. In order to achieve a high degree of realism they use a noise function that generates values with no correlation to any previously generated value. The random nature of this noise goes a long way to eliminating any structure the texture may have had.

The assumption stated by Cheng et al (1998) in their work on thresholding that an object can be distinguished from the background by their intensity values is shown to be true in figure 1.1 of a typical maritime scene. The objects present in the scene are noticeably darker or lighter than the sea. Therefore it should be possible to segment maritime images using the histogram thresholding technique described by Boyle and Thomas (1988) where separate peaks are expected for the background (sea) and objects. Huang et al (2000) uses calculations of foreground and background motion together with a watershed algorithm to segment the foreground from the background. This approach is considered inappropriate for maritime images, as an object such as a buoy could be moving as a result of the motion of the sea. This would not be segmented using this method. Other maritime objects could be moving more than the sea such as a motor boat and other less than

the sea such as a harbour wall. These differences in motion are considered too great for this method to be effective.

Voles et al (1999) uses frame differencing to identify maritime objects that are moving. The frame differenced image in figure 2.13 also shows areas of the sea that are moving and when considered with the original image in figure 2.12 it can be seen that the pontoon object has not been segmented. This is due to the static nature of the camera and as the camera will be mounted on a moving platform in this application static objects not being segmented by Voles et al (1999) should not be an issue. Frame differencing is considered to be a method worth investigating for maritime objects as Voles et al (1999) have shown it to be successful.

In later work Voles et al (2000) use statistical measures more commonly used in texture analysis to determine a threshold for the segmentation of maritime objects. Whilst this technique is shown to be successful it does suffer from the drawback of requiring the manual selection of the threshold value. With the addition of a method of threshold determination a statistical approach is considered a suitable approach for the segmentation of maritime images.

Following segmentation objects are often identified as blobs on a background. Several techniques for tracking these blobs through an image sequence have been reviewed including that by Marcenaro et al (2002) who look for regions of interest that overlap in adjacent frames. If the frame capture rate is high enough or the objects are not moving at high velocities then this approach is valid, as shown in figure 2.14. The approach taken by Ellis et al (1991) is to record measurements of object size, centroid, minimum bounding rectangle, location, and sequence number. Wang et al (2000) also uses measurements of object parameters to aid the tracking process. In this case the measurements taken are object trajectory (centroid position), size, greyscale distribution, and texture. Wang et al (2000) also

use the assumptions that object trajectories are close to a straight line over a few adjacent frames and that object acceleration is constant in these frames to help in the tracking of objects. Teal (1997) too uses object parameter measurements to track objects through image sequences. In this case it is vehicles travelling along a road. The technique is shown to be successful where objects are some distance from the camera and occupy a small number of pixels. As the objects of interest in maritime scenes could also occupy small numbers of pixels this approach could be appropriate for use in this application.

As maritime objects could have a wide variety of appearances the collection of measurements about each object could aid the tracking process. If the assumptions made by Wang et al (2000) were shown to be true of objects in maritime images this approach could well be appropriate for the prediction of future object motion in place of more complex techniques such as the Kalman filter (Kalman, 1960). However, the Kalman filter is arguably the most popular method for tracking objects and should be investigated for this application.

Model-based approaches to high level vision such as that used by Koller and Nagel (1993) and Ferryman et al (1995) rely on geometric models and have been shown to be successful. Ellis et al (1991) agrees that explicit models are appropriate for describing a wide range of man made objects. However, they go on to say that geometric models do not adequately describe most natural objects such as landscape. They also point out that small objects a large distance from the camera suffer from poor spatial resolution precluding the use of structural models to represent objects.

The essence of neural network techniques is the use of a multitude of elemental non-linear computing elements organised as networks reminiscent of the way neurons are interconnected in the human brain (Gonzales and

Woods, 1992). Di Bona et al (2001) and Messer and Kittler (1998) both use neural networks in their work and it becomes clear from both that the design of the network is complex and that the amount of training required for the network to interpret unseen images correctly is vast. Matesin et al (2001) uses a different approach to high-level vision. Their knowledge-based expert system uses previously known information and a set of rules of the form 'IF…THEN'. This allows the flexibility of adding or subtracting rules from the system easily and does not require any prior training before use with new image sequences. This type of approach is considered to be appropriate for use with maritime images as information can be ascertained through observation and experimentation that can be used to form rules which can then be used to develop a method of estimating the collision risk of objects identified in the scene.

Chapter 3: Image Smoothing

3.1 Introduction

The first task of the image processing system is the acquisition of suitable images from a sensor. The images for this system are those in the visual spectrum taken from a camcorder facing forward over an area of sea. The capture and digitisation process introduces unwanted noise, random variations in intensity values. It is considered to introduce noise from the optics, the recording onto tape, the playback of the tape, and the digitisation of the images. These types of noise can be considered Gaussian as Jain et al (1995) state that Gaussian noise contains variations in intensity that are drawn from a Gaussian or normal distribution and is a very good model for many forms of sensor noise.

Filtering is the generic name for techniques of changing the grey levels in an image to enhance the appearance of objects according to Ballard and Brown (1982). These techniques can equally be used to suppress elements within an image that are not desired with noise being a good example of this. Sonka et al (1993) state the aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing.

The first stage of this system, noise reduction and smoothing, is performed to achieve the following criteria:

- To reduce the Gaussian noise typically introduced by the acquisition and digitisation process.
- To enhance the edge detail of any objects in the image to aid the segmentation of the objects.

Jain et al (1995) note that whilst linear smoothing filters are good filters for removing gaussian noise and, in most cases, other types of noise as well, the application of linear noise-cleaning filters has the effect of defocusing the image. Edges become blurred (Haralick and Shapiro, 1992).

3.2 Image Filtering

Three non-linear filters have been considered for use in this application, the Median, Kuwahara, and FWV. Each is described in the following sections after which results are shown.

3.2.1 Median Filter

The median filter is a non-linear filter that replaces the current pixel in the image with the median of the intensities in its neighbourhood. In its usual form a square neighbourhood is used that may be typically 3x3 or 5x5 in size. Figure 3.1 shows a 3x3 neighbourhood. The mask is placed over the 9 pixels in the top left corner of the image, the intensity value of each pixel is sorted and the target pixel (shown as a black square in figure 3.1) replaced with the median value of the sorted intensities. The mask is then shifted by

one pixel and the median calculated. This process is repeated until the mask has been placed over all combinations of 9 pixels.

Although the median filter and its modifications have shown good efficiency in suppressing impulse noise Chen and Wu (2001) state that the approach is location-invariant in nature and tends to alter the pixels not disturbed by noise. Senel et al (2002) note also that the median can introduce spurious artefacts into the transformed image and that it removes or attenuates image features that are smaller than the filter window.

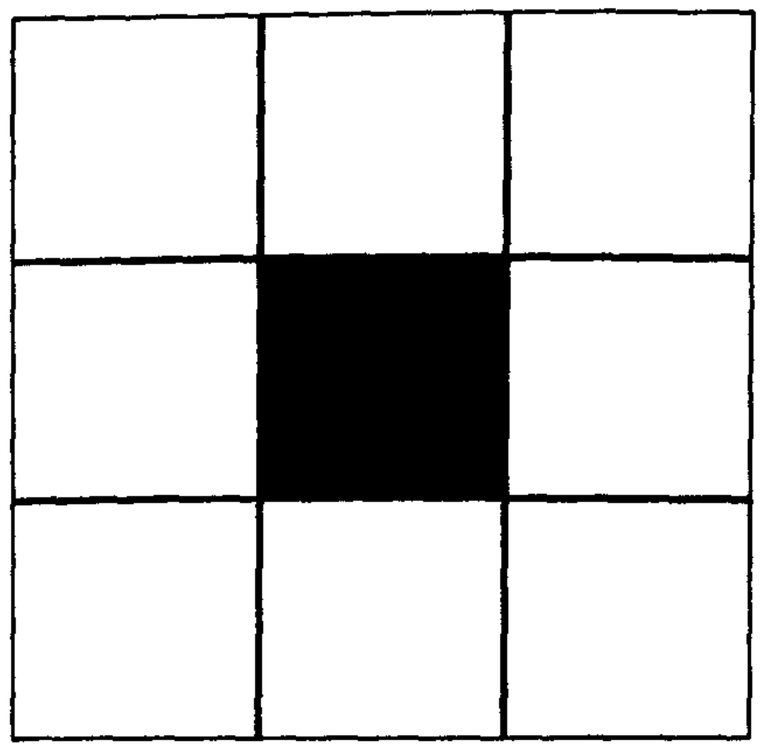

Figure 3.1 – 3x3 Median Neighbourhood

3.2.2 Kuwahara Filter

This was developed by Kuwahara et al. (1976) to facilitate the extraction of the left ventricle in medical images. It divided a square symmetric neighbourhood into four slightly overlapping windows, each containing a central pixel. The mean and variance are calculated for each window. The point of observation, the target pixel, is then replaced with the mean of the window with the lowest variance. Figure 3.2 shows the four neighbourhoods with W being the size of the window in pixels and i,j being the target pixel. Bakker et al (1999) state that the combination of filtering and homogeneity avoids filtering across edges in the image.

The Kuwahara filter has been applied repeatedly to the same image by Minato et al (1987) to arrive at a fixed-point image that is no longer changed by filtering. This repeated application of the filter has the effect of reducing the variance in the windows each iteration until such time as the variance is zero.

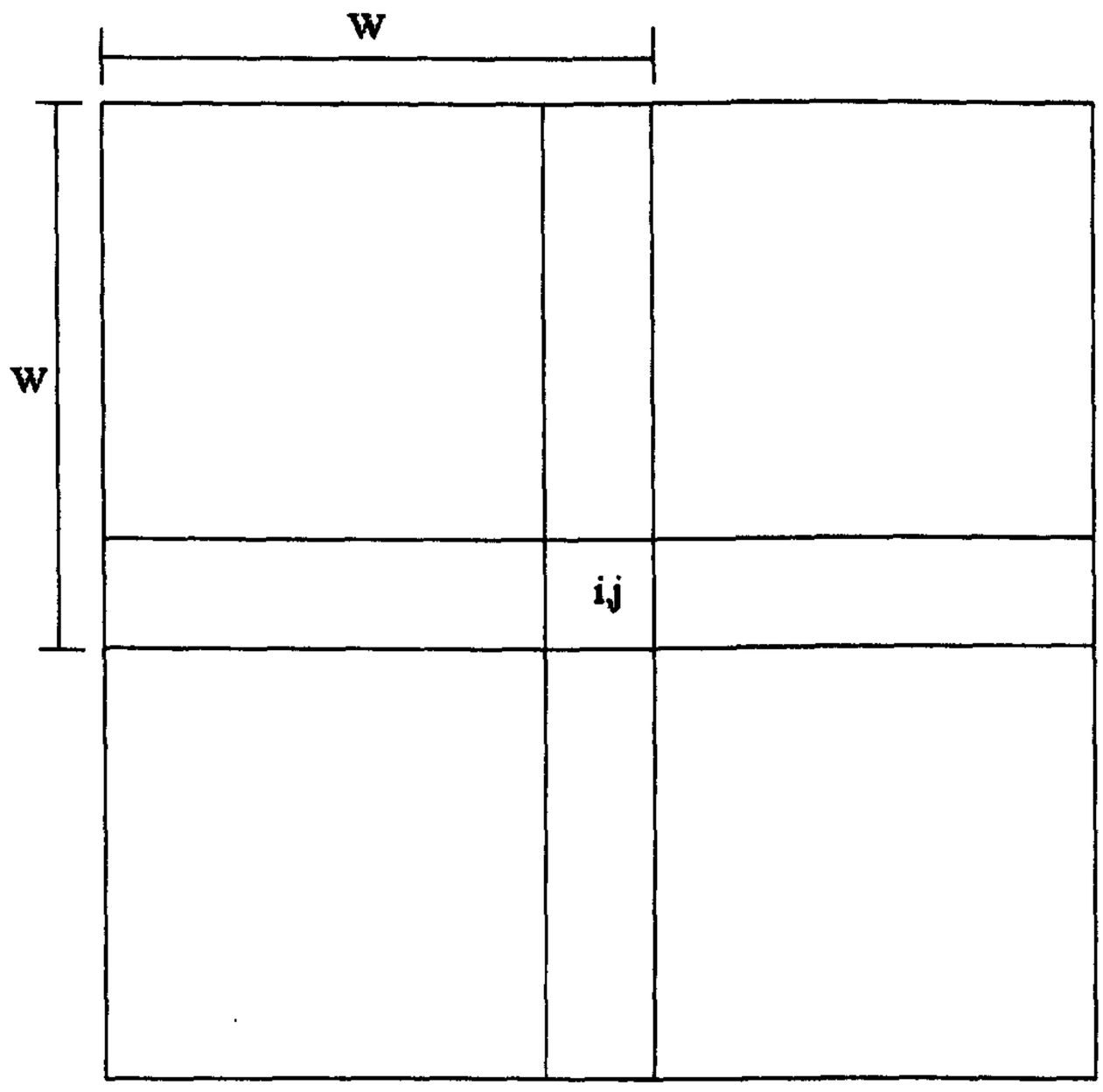

Figure 3.2 – Kuwahara filter neighbourhoods (Kuwahara et al, 1976)

3.2.3 FWV Filter

The FWV filter is an expansion of the Kuwahara filter and was developed by the author specifically for use with maritime images. Where the Kuwahara uses four neighbourhoods the FWV uses nine. A 3x3 convolution mask is placed over the image with the target pixel placed in the bottom right corner of the mask as in figure 3.3.

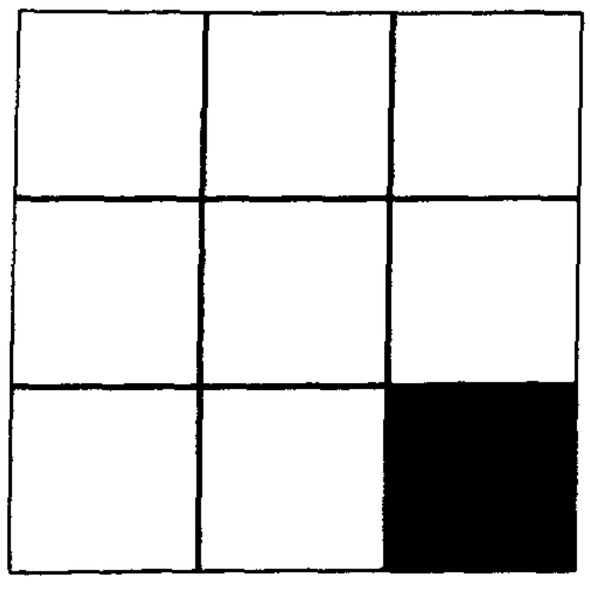

Figure 3.3 – Target Pixel in Mask

The Mean μ and Variance σ^2 are calculated for the 9 pixels using:

$$\mu = \frac{1}{9} \times \sum_{x=1}^{3} \sum_{y=1}^{3} P(x,y) \tag{3.1}$$

$$\sigma^2 = \frac{\sum_{x=1}^{3} \sum_{y=1}^{3} (P(x,y) - \mu)^2}{9} \tag{3.2}$$

where $P(x,y)$ is the grey level pixel value at position x,y

The mask is then shifted such that the target pixel is placed in the bottom centre of the mask and the mean and variance calculated. The mask is shifted a further seven times such that the target pixel has been placed in each possible place in the mask as figure 3.4 shows. The mean and variance are calculated each time using equations 3.1 and 3.2.

The new, filtered value of the target pixel is set to the mean of the window with the lowest variance. This gives the smoothing of a mean filter as the target pixel is ultimately replaced with the mean value of the pixels around it whilst retaining the edge detail as any windows with an edge present will result in a higher variance figure.

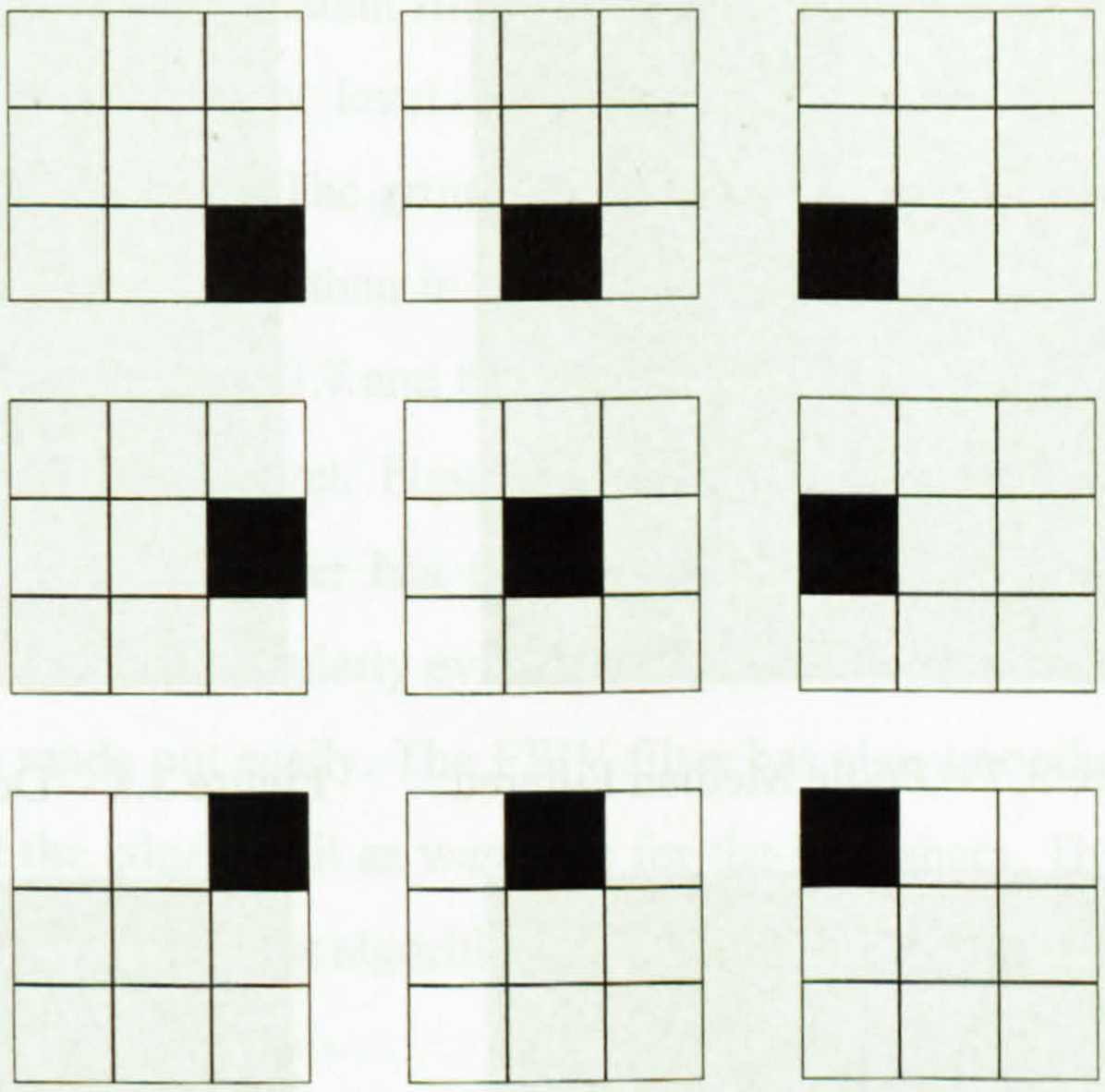

Figure 3.4 – Target Pixel Positions

3.3 Results

Shown here are the results of the Median, Kuwahara, and FWV filters. Figures 3.5 and 3.6 shows the original test images and figures 3.7 to 3.12 show an enlarged section of those images with the three filters applied.

Figure 3.5 – Poole Sequence

Figure 3.6 – Dover Sequence

Figure 3.7 – Poole Median Filtered

Figure 3.8 – Dover Median Filtered

Figure 3.9 – Poole Kuwahara Filtered

Figure 3.10 – Dover Kuwahara Filtered

Figure 3.11 – Poole FWV Filtered

Figure 3.12 – Dover FWV Filtered

It can be seen that the Median filter has tended to group pixels together into blocks of the same grey level. It gives the appearance of reducing the resolution of the image. The grouping has left collections of pixels near the top of figure 3.8 brighter than in the original image and has meant that the hull of the boat in figure 3.7 and the body of the buoy in figure 3.8 no longer contain a single grey level. Figures 3.9 and 3.10 show the results of the Kuwahara filter. This filter has smoothed the images whilst retaining the edge detail. This is particularly evident in figure 3.10 where the shape of the buoy can be made out easily. The FWV filter has also smoothed the images and retained the edge detail as was seen for the Kuwahara. This is expected as both filters use a similar algorithm.

3.4 Discussion and Summary

Although linear filters such as a low-pass filter are optimal when the noise is additive and gaussian (Yin et al, 1993) and can improve the quality of the image by filtering the noise, image details are also filtered and therefore smoothed with undesirable consequences (Perez-Luque et al, 1990). Non-linear filters have been successfully substituted for linear filters because of their better edge preservation and impulse noise rejection capabilities (Chen and Wu, 2001).

Anisotropic diffusion is an extension to isotropic diffusion equation and was introduced by Perona and Malik (1990). It is a non-linear filter that encourages smoothing within a region in preference to smoothing across the boundaries thus preserving edges. This was achieved by incorporating an 'edge-stopping' function to the equation (Black et al, 1998). Perona and Malik (1990) show that in addition to smoothing within a region, by choosing the conduction coefficient part of the diffusion equation locally as a function of the magnitude of the gradient of the brightness function not

only preserves, but actually sharpens brightness edges. This means that the more defined the region boundary the less smoothing will occur across it. They say that by using this approach to choosing the conduction coefficient, thinning and linking of edges is unnecessary and that edge junctions are preserved. They do point out however that in images where the brightness gradient generated by noise is greater than that of the edges, this scheme for anisotropic diffusion proves insufficient to obtain a correct segmentation. Voles et al (1999) highlight that anisotropic diffusion is an iterative and time consuming method.

Three non-linear filters have been tested with maritime images, the median, FWV, and Kuwahara filters. The median has preserved the edge detail although this has been through blocking grey-level areas together giving an image that has been smoothed by a lesser amount than the other filters. This can be seen clearly when the enlarged sections of the test images are considered.

The FWV and Kuwahara filters have both preserved edge detail and smoothed the gaussian noise and the wave noise present in the image. Visually there is little difference between them. As both filters use a similar algorithm this similarity in results is to be expected. The main difference is the FWV's use of nine to the Kuwahara's four neighbourhoods. This in theory should lead to a more appropriate filtered value for the target pixel. This is because the FWV considers a greater number of neighbourhoods than the Kuwahara and, as Minato et al (1987) point out, "among the neighbourhoods, the one which has the smallest variance is considered as the one to which the pixel should belong". In practice however, the increased number of neighbourhoods has made no difference to the final filtered result, a comparison of the filtered pixels between the Kuwahara and FWV results shows the pixel intensities to be the same.

The real difference between these two filters is the computation time. With the same calculations being made and the same conclusion being reached as to the filtered value for the target pixel the difference in computation comes down to the number of neighbourhoods being considered. With the FWV having nine and the Kuwahara four, the FWV takes over twice as long to filter each pixel.

The FWV filter was developed by the author (Smith and Teal, 1999) specifically for use with the maritime images used here and whilst successful was shown to be no more effective than the Kuwahara filter (Kuwahara et al, 1976) but had a higher computational cost. Consequently the Kuwahara filter was chosen for the image acquisition and smoothing sub-system.

Chapter 4: Maritime Image Segmentation

4.1 Introduction

A maritime scene is an extremely complex image to analyse (Sanderson et al, 1997), due to the fact that large areas of the image are moving with the motion of the sea. To this problem is added the fact that the objects to be tracked have a wide variety of appearances. The segmentation of an image is a very useful technique for reducing the amount of data further processing steps deal with and a great many methods have been developed to do this. According to Low (1991), segmentation can be approached from two points of view: by identifying edges that run through an image or by identifying regions within an image. Region-oriented segmentation techniques range from histogram thresholding (Bassmann and Besslich, 1995) to Huang et al (2000) who use the dominant motion model (Black, 1996) to determine both foreground and background motion and then apply a watershed algorithm to segment the image. Contour-oriented techniques include the Canny edge detector (Canny, 1986) and the scanning wave used by Ishii and Kyuma (2001). An approach more commonly used for object tracking is frame differencing. Voles et al (1999) use this approach to successfully segment maritime objects from the background but note that the motion of the sea is also captured in a frame differenced image. Because of this the differenced image is processed further to segment the objects from the sea.

Presented in this chapter are three methods of segmenting maritime images into areas of 'sea' and areas of potential maritime objects. The first is a region-oriented thresholding algorithm (Bassmann and Besslich, 1995) based on the histogram data taken from the whole image. The second is the method of frame differencing used by Voles et al (1999). The third is an algorithm developed by the author (Smith et al, 2003) based on histogram data taken from various regions of the image that calculates a statistical range of grey-levels that represent the 'sea' present in the image. This grey-level range is used to segment the image into two regions, labelled sea and object.

Each of the methods presented here have been evaluated against a method of manually segmentation to measure the success of each approach with maritime images and conclude which is the most appropriate algorithm for use in the system being developed.

4.2 Thresholding

This common segmentation technique, described by Bassmann and Besslich (1995), is based on grey level differences in the source image. In figure 4.1, a frame from the Dover test sequence, it can be seen that the two objects are darker than the sea. The histogram of the image is shown in figure 4.2 and the two regions are clearly distinguishable. The small peak to the left represents the objects and the large peak to the right represents the sea. Placing a threshold in the valley between the two peaks of the histogram allows the segmentation of the image into the two regions. The sea region is assigned grey level 0 and the object region grey level 1.

In order to determine the correct threshold value the histogram values are smoothed to remove any small valleys and peaks to leave only the

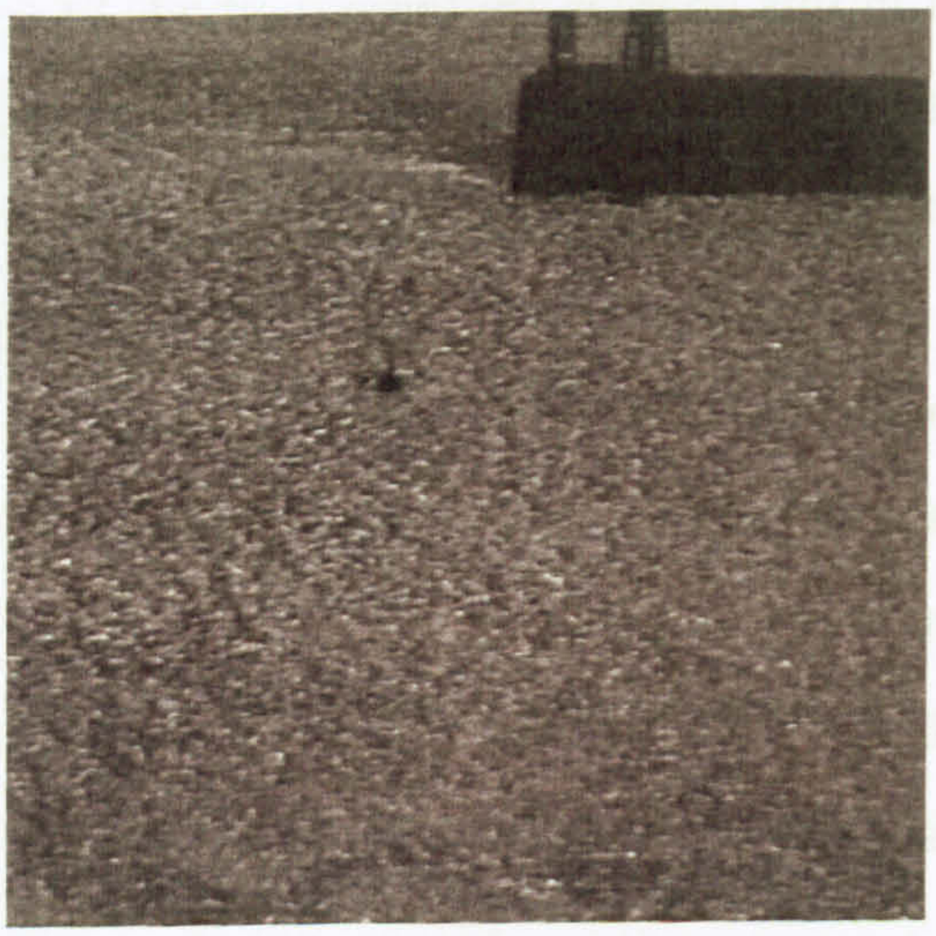

Figure 4.1 – An image from the Dover test sequence

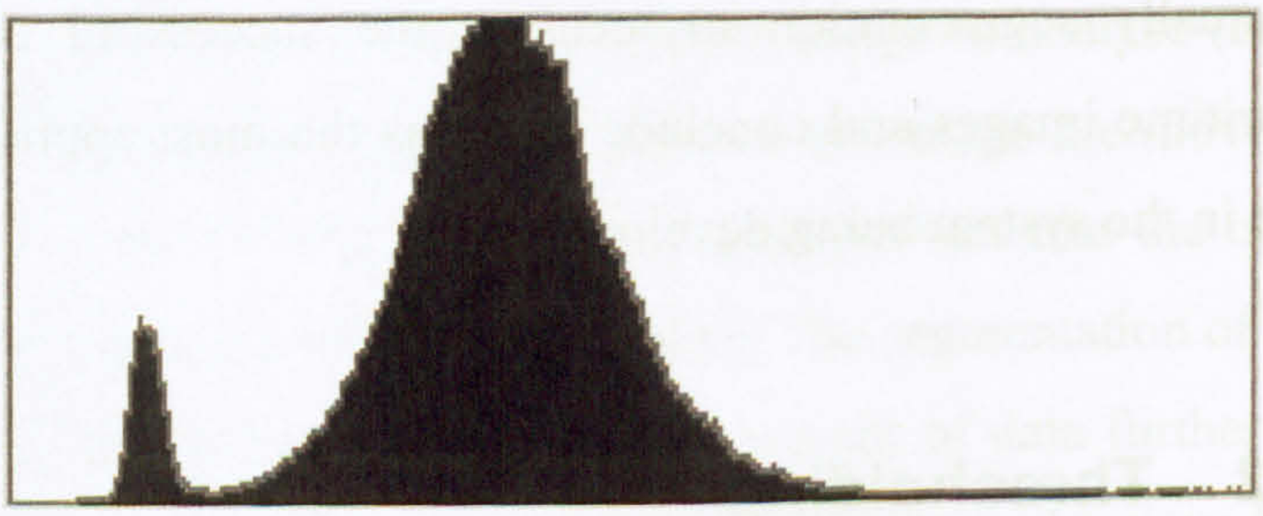

Figure 4.2 – Histogram of Dover image

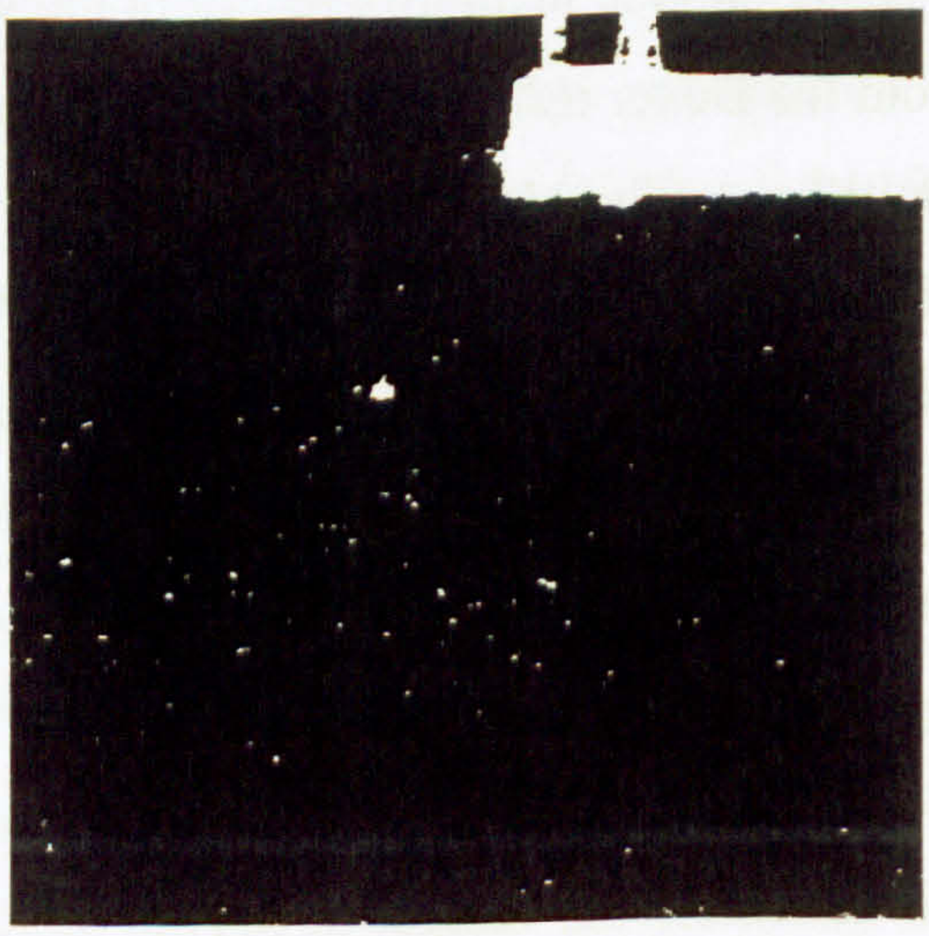

Figure 4.3 – Dover image Segmented by Thresholding

significant peaks and valleys. The thresholding algorithm searches the smoothed histogram data and allocates a threshold for every significant valley as maritime objects can be brighter as well as darker than the sea. Figure 4.3 shows the result of thresholding the frame from the Dover sequence.

4.3 Frame Differencing

Voles et al (1999) begin their segmentation by creating a difference image by subtracting each pixel value in the current frame from those in the previous frame in the sequence. To avoid potential implementation problems the absolute value of the pixel subtraction is considered. This results in an image with stationary areas represented as dark areas and areas where motion is present are bright. Figure 4.4 shows a current frame from the Poole test sequence and figure 4.5 shows the difference image.

Figure 4.4 – Current frame

Figure 4.5 – Difference image

To determine the co-ordinates of the bright areas of motion the difference image is projected onto its horizontal and vertical axes from which two projection histograms are obtained. Figure 4.6 shows the vertical projection histogram of the difference image. Between frame width 211 and 271 there is a large peak representing the moving object. As maritime scenes contain

large amounts of unpredictable wave motion it is necessary to threshold the projection histograms to eliminate the sea motion. The threshold value was set to the RMS value as Voles (1997) concluded that the best empirical results were obtained using this threshold method. A binary histogram is then created by setting every value below the threshold to 0 and all those above the threshold to 1. Many narrow peaks still remain and these are either combined with broader peaks if they are close to them or they can be deleted. Voles et al (1999) use a minimum peak width of 5 pixels for peak combination or deletion. The resulting projection histogram can be seen in figure 4.7.

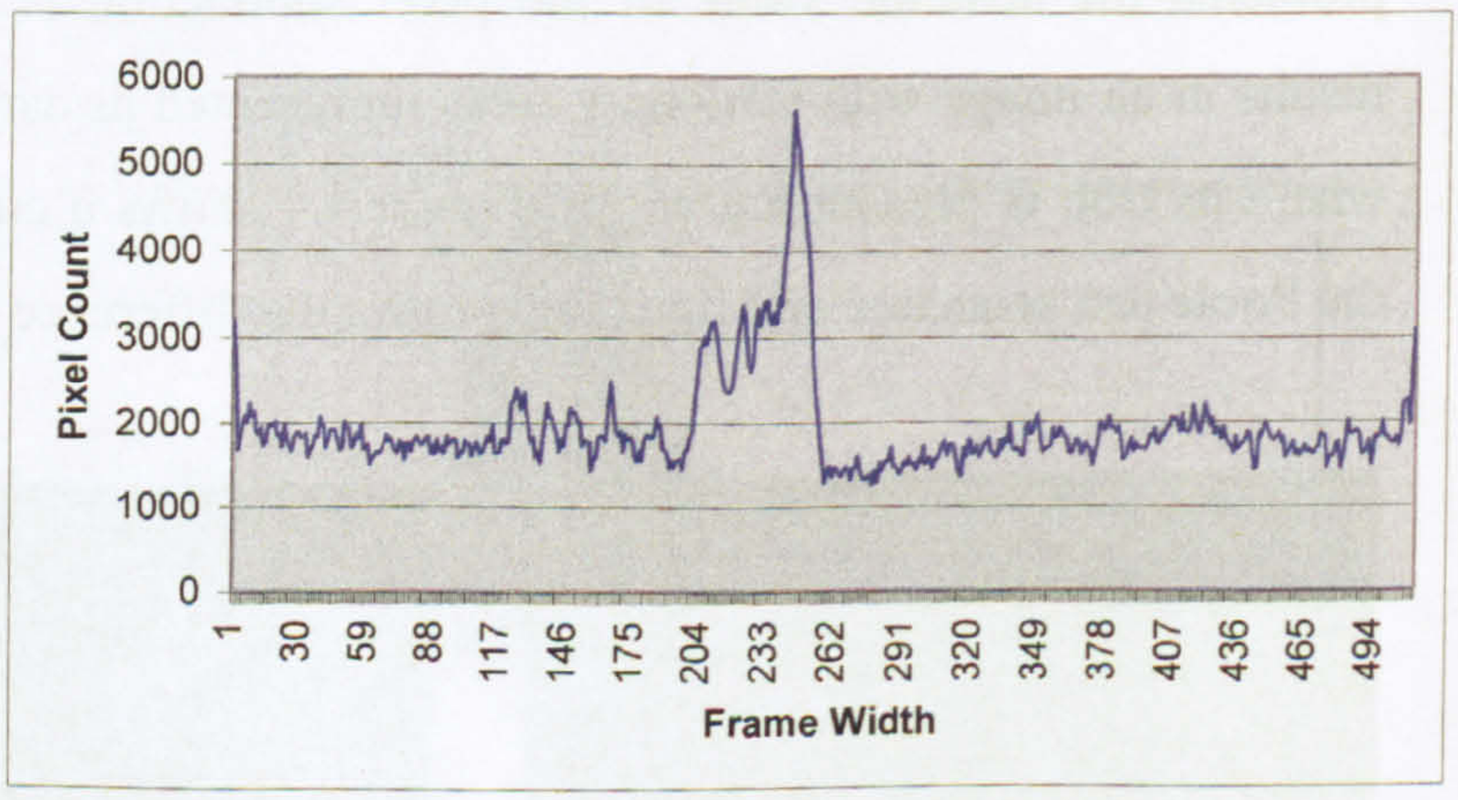

Figure 4.6 – Vertical Projection Histogram

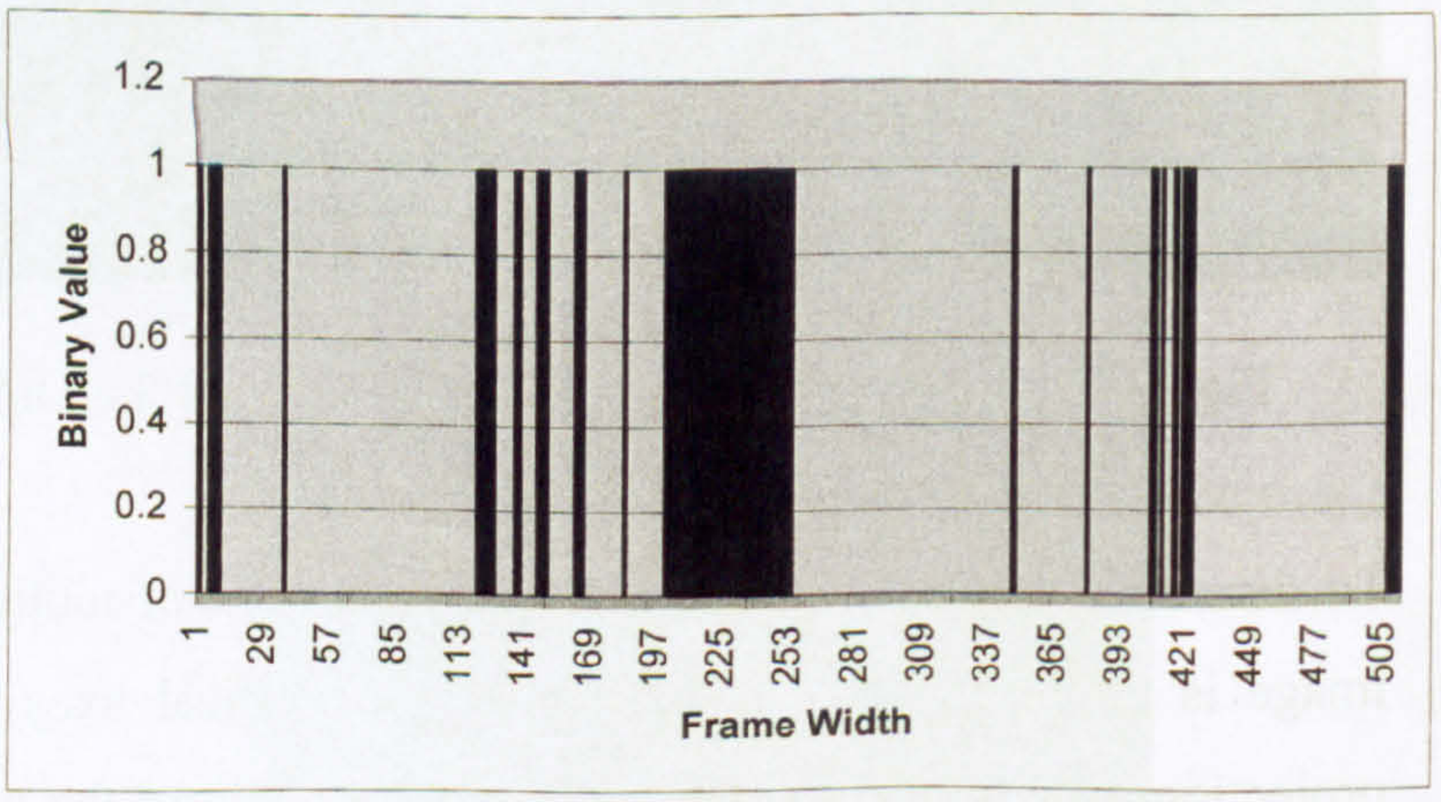

Figure 4.7 – Binary Vertical Projection Histogram

Finally, the resulting histograms are scanned to determine the horizontal and vertical positions of the remaining regions. These regions are identified as a binary image with white (motion) rectangles on a black (stationary) background. Figure 4.8 shows the segmented image. Although the stationary object has not been segmented this is due to the static nature of the camera. It is considered that with a camera mounted on board ship the motion of the platform would enable the stationary object to be identified and segmented.

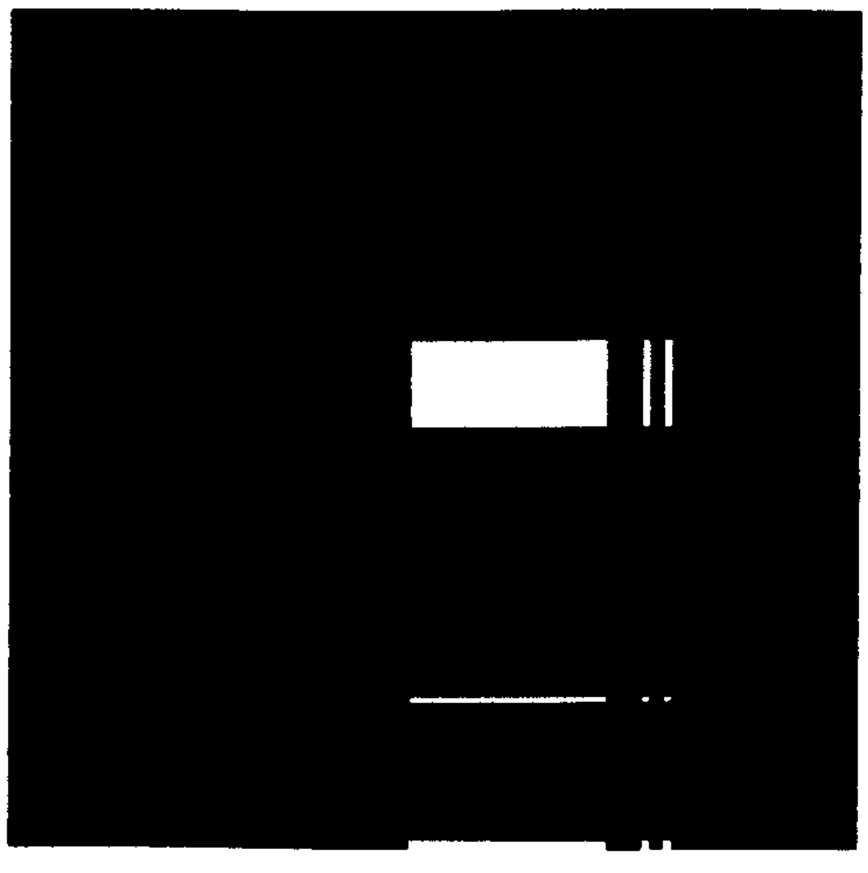

Figure 4.8 – Image Segmented by Frame Differencing

4.4 Sea Characterisation and Segmentation

It can be observed in histograms from each test sequence that objects occupy a different range of grey levels from the sea. Furthermore it is noted that the sea generally occupies more central grey levels whilst objects occupy grey levels either side of the grey level range of the sea. Figures 4.9 – 4.11 show histograms from each of the test sequences. The separate peaks of object and sea are particularly strong in the histogram from the Dover sequence, figure 4.10, although they can be picked out in all three histograms. It can further be considered that the distribution of sea pixel grey levels approximate a normal distribution. This method aims to determine the characteristic grey level range of the sea and use this to segment the image into regions of sea and object.

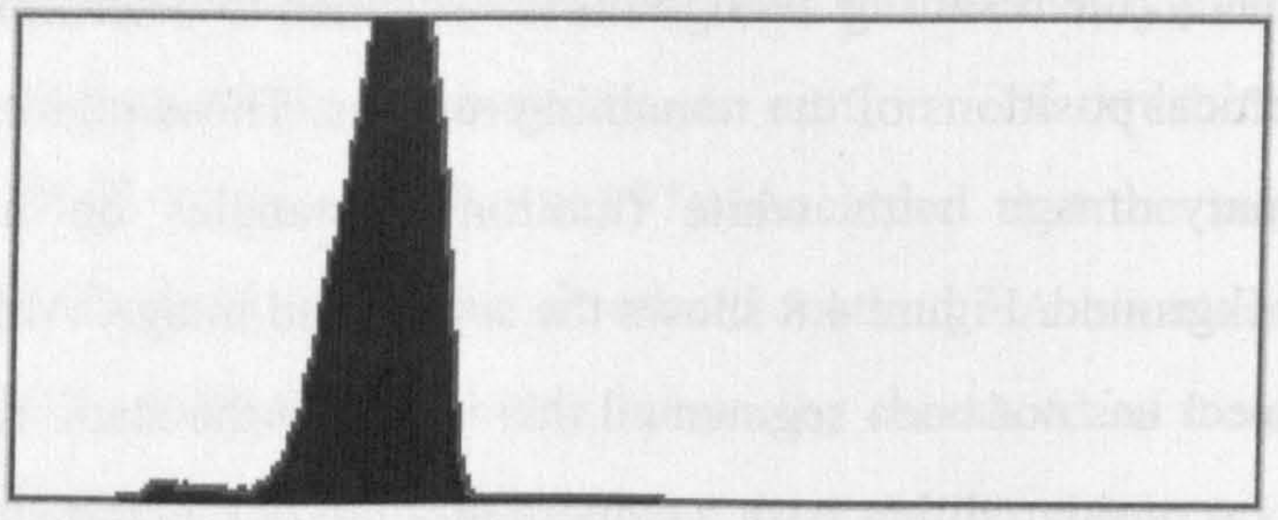

Figure 4.9 – Histogram from the Poole Sequence

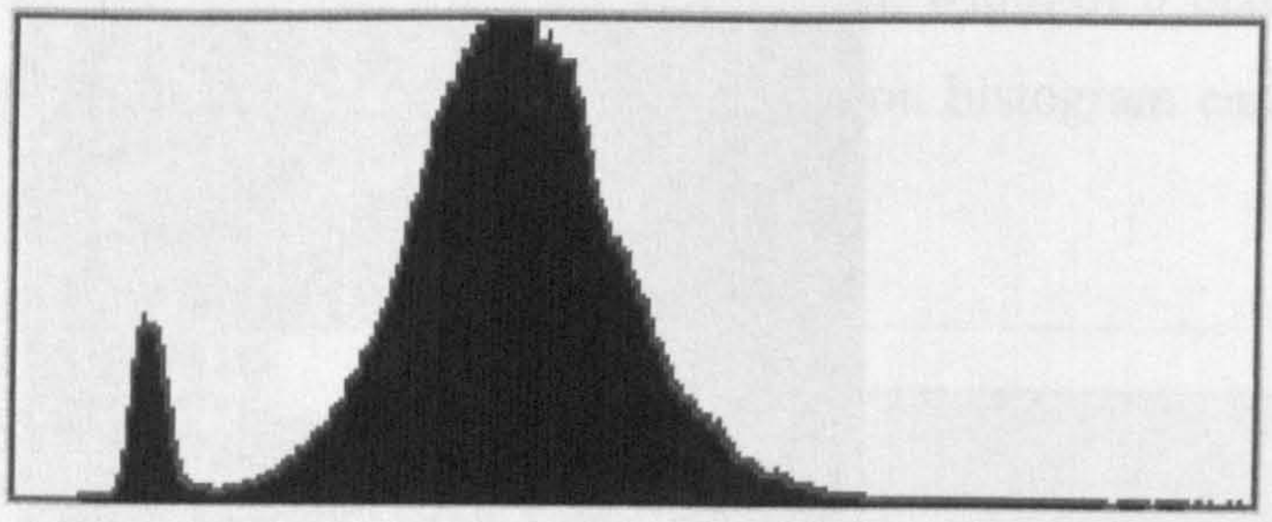

Figure 4.10 – Histogram from the Dover Sequence

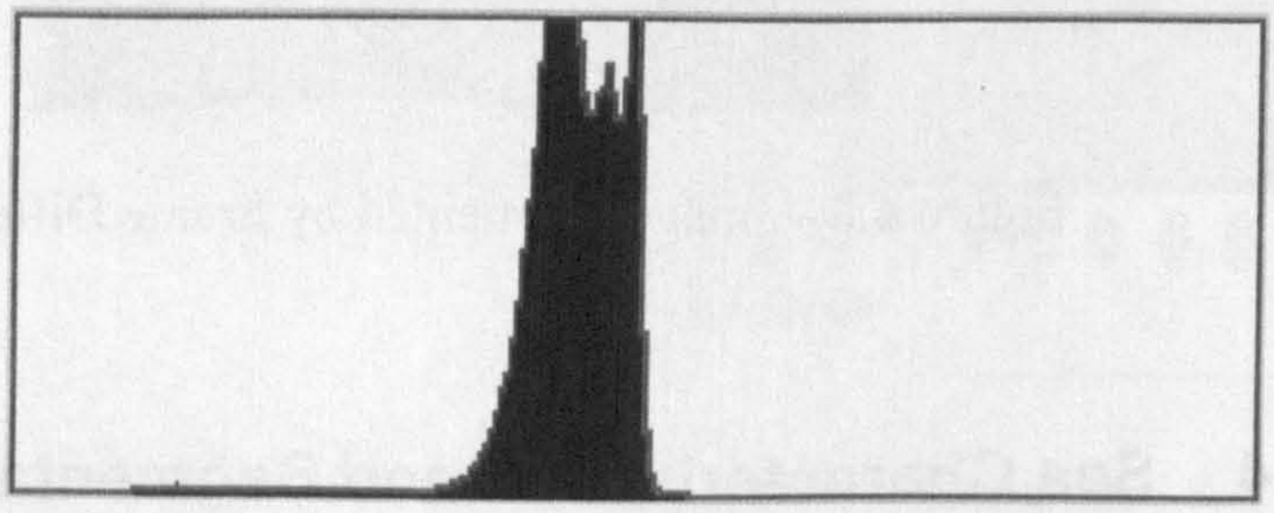

Figure 4.11 – Histogram from the Portsmouth Sequence

Figure 4.12 shows a flow chart of the Sea Characterisation & Segmentation process. The top half of the figure breaks the Sea Characterisation process into its individual processing blocks. The output of this process is four grey-level ranges, one for each of the four windowpanes the image has been divided into during the characterisation. The grey-level ranges are passed to the Segmentation process shown in the lower half of figure 4.12. Here the grey-level ranges are used to determine the segmentation of pixels into either sea or object regions.

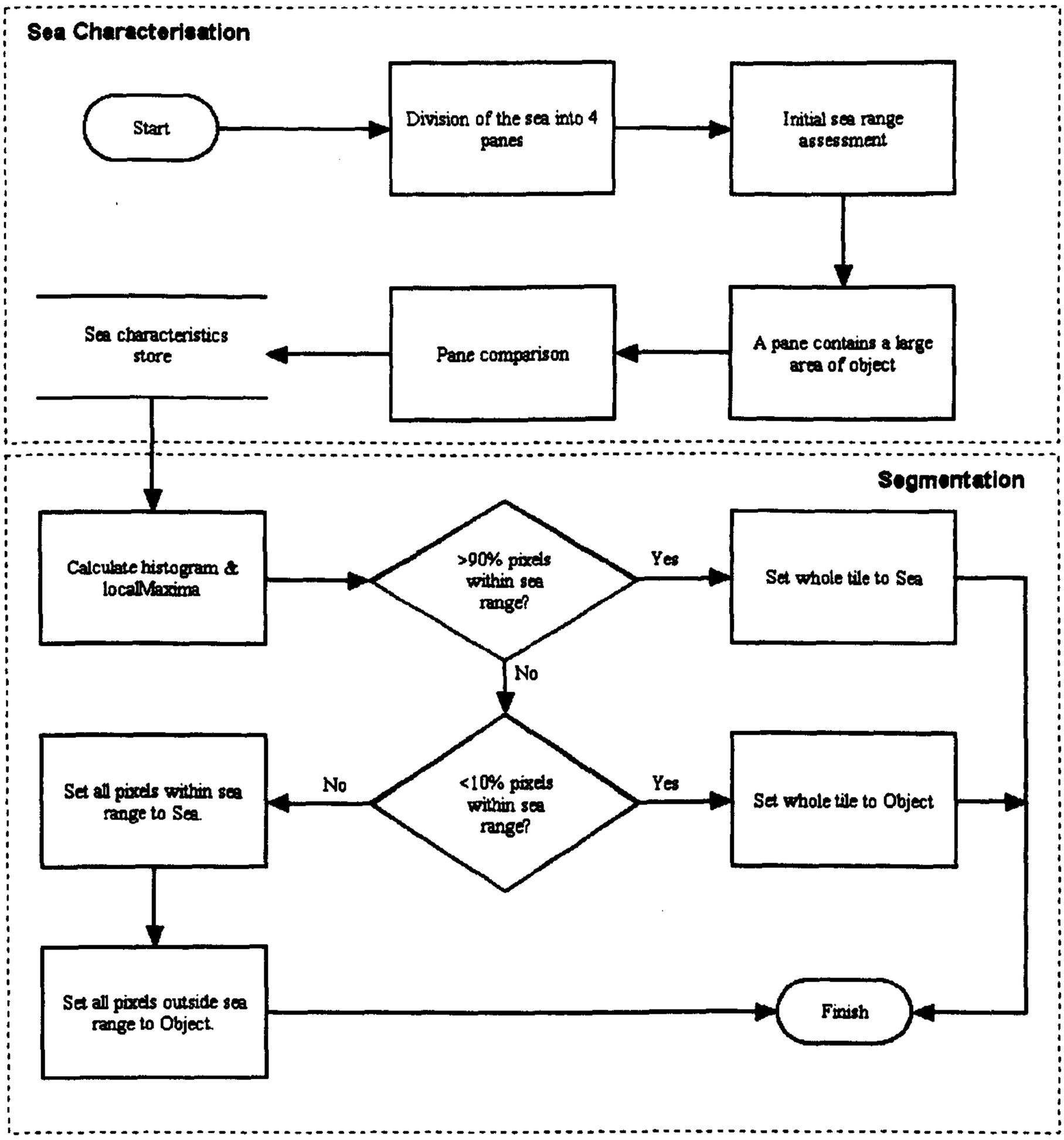

Figure 4.12 – Sea Characterisation & Segmentation Flow Chart

4.4.1 Sea Characterisation

The aim of the Sea Characterisation process is to arrive at a grey-level range that is representative of those in the sea portion of the image. Characterising the range comprises several smaller operations. First, that part of the image containing the sea, the sea window, is divided into four equally sized areas known as windowpanes. Each pane is then addressed in turn and a grey-level range determined for the sea pixels present in that part of the image using histogram statistics calculated for a set of tiles placed within the pane.

The results of the four panes are then compared to ensure a consistent result is obtained across the image.

Preparation of the Image

The first task is to determine that part of the image that contains sea pixels. The requirement for this is to determine the shoreline (that point in the vertical plane where the sea ends and the sky or land begins). The observation of video sequences taken from a moving vessel in calm conditions has shown that the shoreline does not pitch up or down, nor roll from side to side by more than 3 pixels. A fixed value for the shoreline can be set without compromising the outcome of the algorithms. It is necessary for the user to specify a value, which is treated as a horizontal line, for the level of the shoreline. The area under that shoreline is considered to be a sea 'window' as shown in figure 4.13.

The open world nature of maritime images gives rise to other issues that need addressing before any characterisation can be done. The state or condition of the water changes with the strength and direction of the wind between a flat calm when the water gives the appearance of glass, or a mirror, to storm force conditions where swell and white horses (the frothy white crests) are prevalent. In addition to the sea condition there is the issue

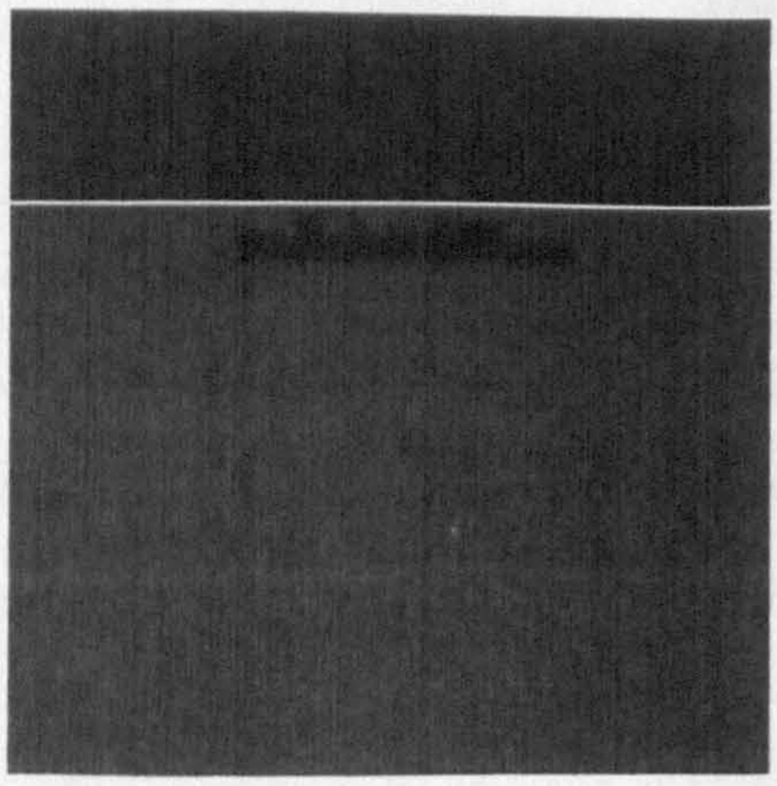

Figure 4.13 – Shoreline

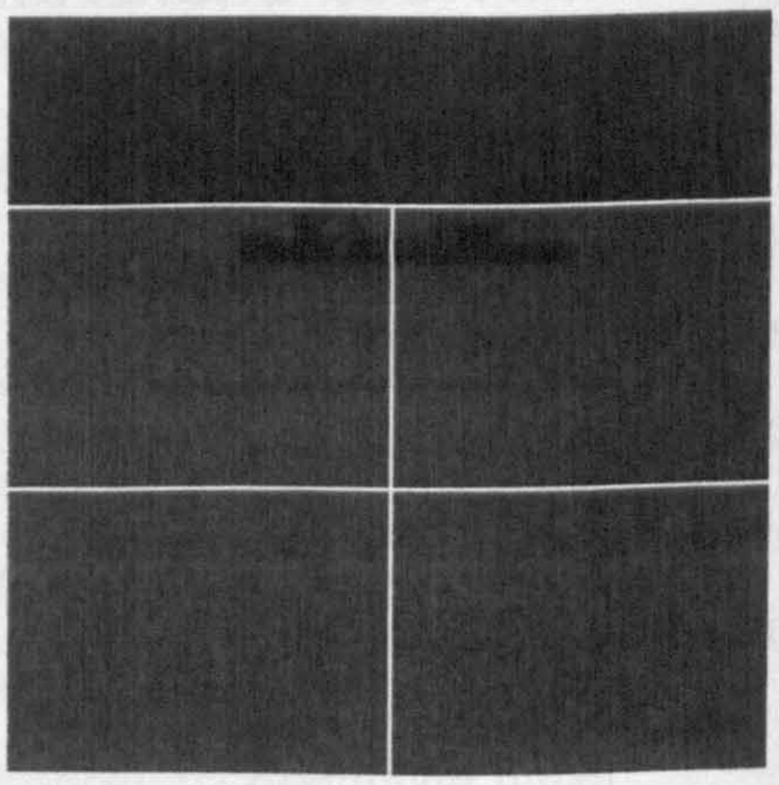

Figure 4.14 – Shoreline & Panes

of scene illumination. In any weather conditions the illumination will change but this is most noticeable in bright, sunny conditions. If a cloud were to pass in front of the sun a large shadow will be cast across the scene. These two aspects mean that the sea itself can contain a different range of grey-levels from one side of the image to the other. This could lead to false segmentation if the sea were characterised as a whole. To reduce the effects of sea state and differences in illumination across the image the sea 'window' is divided into four equal quarters called 'panes'. Figure 4.14 shows the image complete with the panes.

Initial assessment of each pane

Taking each pane in turn an initial assessment is made as to the characteristic grey level range for the sea in the pane. If the grey level range was calculated using every pixel in the pane then the pixels from any object present in the pane would be included giving a false characterisation. Therefore a number of smaller pixel regions called tiles are used and the initial grey level range calculated using the pixels in these tiles.

In order to achieve an accurate characterisation of the pane all grey-levels from every part of the pane are included in the calculations. The random placement of tiles within the pane could lead to the tiles all being placed in one small area of the pane or even overlapping each other. This is undesirable, as grey-levels present in other areas of the pane would not be represented in the characterisation. More importantly, if the tiles were randomly placed over an area of object then the characteristic grey-level range would be false. Fixed tile positions were used to ensure the consideration of grey-levels from all parts of the pane. A later stage in the characterisation process would address the issue of a tile being placed over an area of object.

The positioning of the tiles is shown in figure 4.15. This arrangement of tiles gives an even spacing between a tile and either the edge of the pane or another tile thereby minimising the possibility of a large area of grey-levels being underrepresented in the characterisation process.

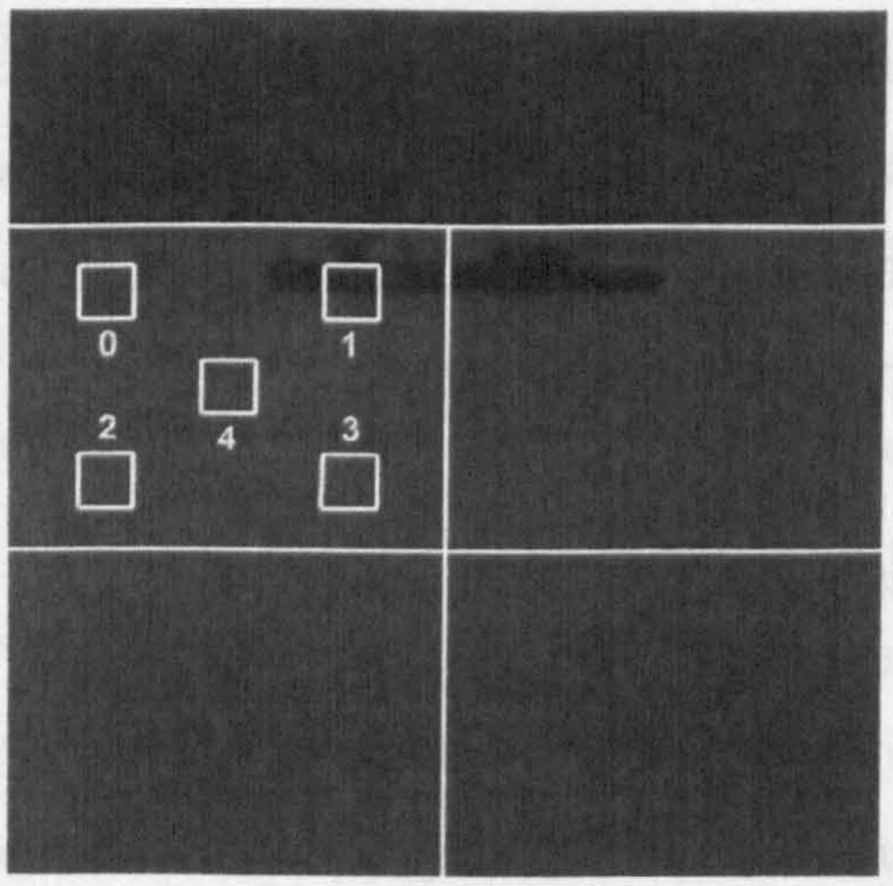

Figure 4.15 – Panes & Tiles

Calculation of Tile Statistics

The characterisation process revolves around histograms being calculated for each of the tiles. Figure 4.16 shows the histograms for the five tiles shown in figure 4.15. It is noticeable from figure 4.16b, the histogram for tile 1, that this tile has the object in it. Moreover, it is observable that there is a distinctive range of grey level's that covers those tiles containing only the sea. As with the sea peak in the histograms of each test sequence in figures 4.9-4.11 this range can be considered to approximate a normal distribution.

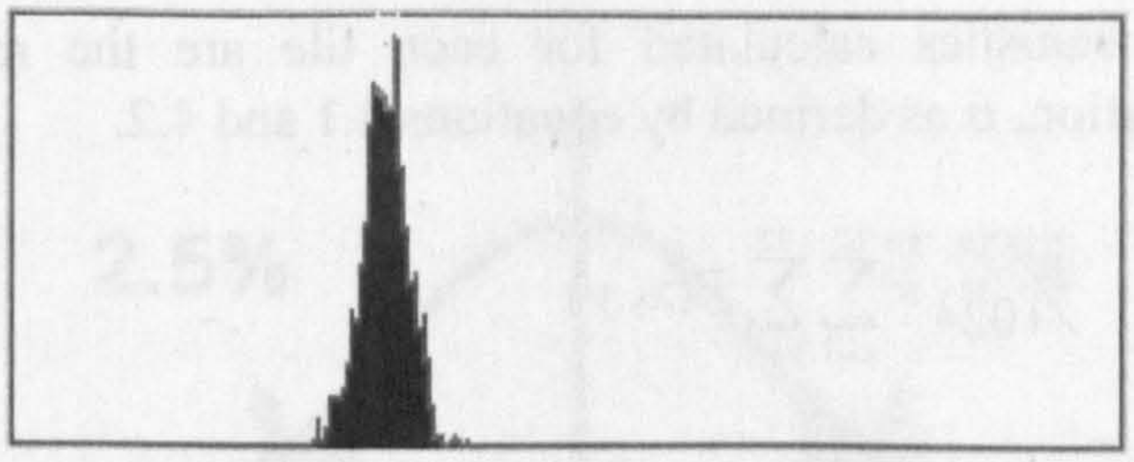

Figure 4.16a - Tile 0

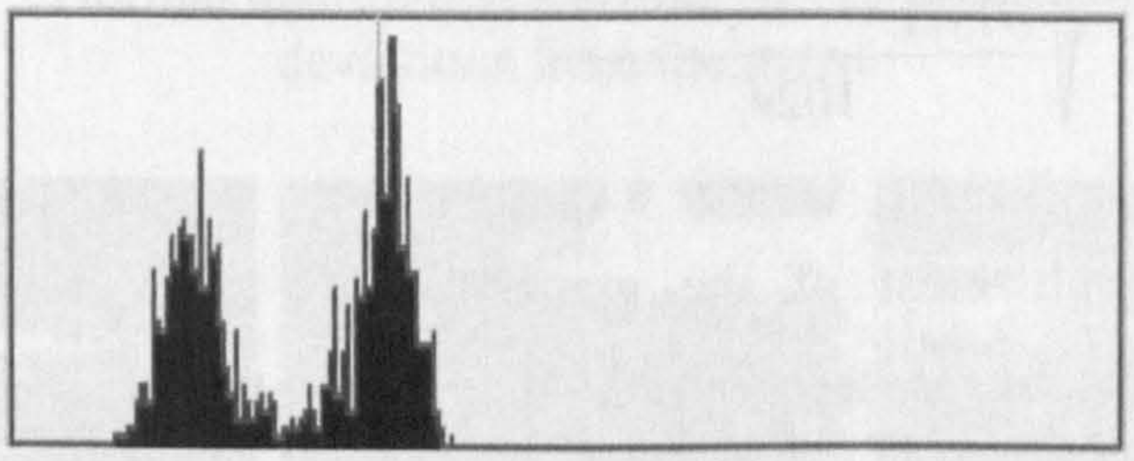

Figure 4.16b - Tile 1

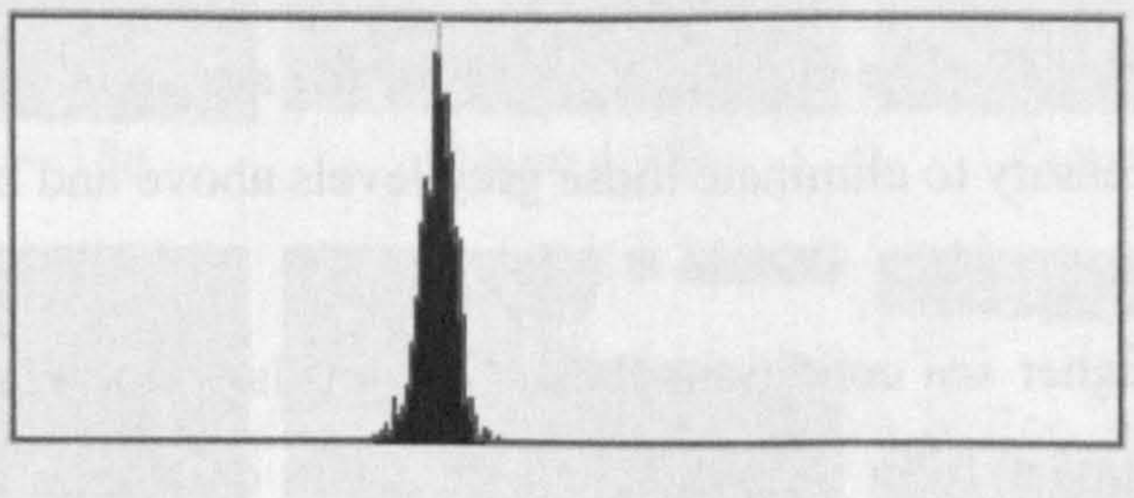

Figure 4.16c - Tile 2

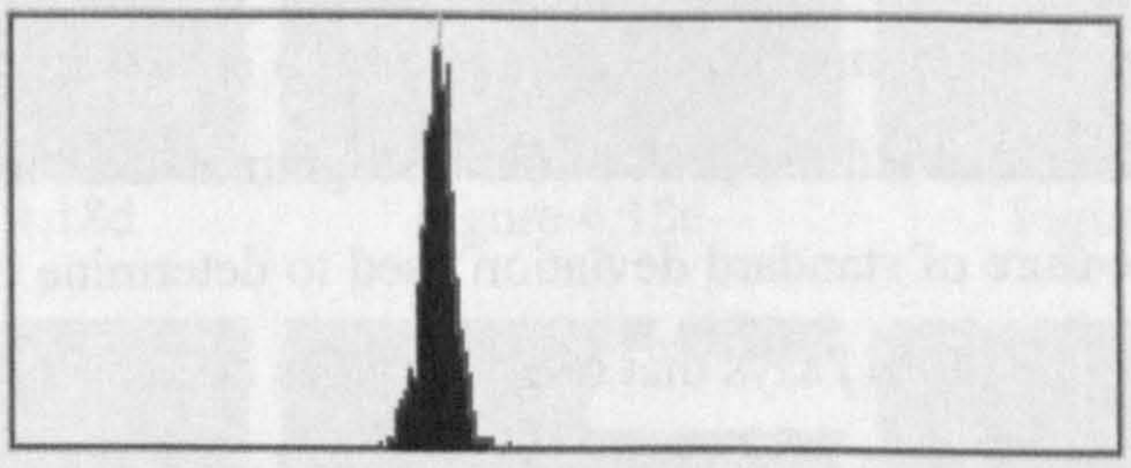

Figure 4.16e - Tile 3

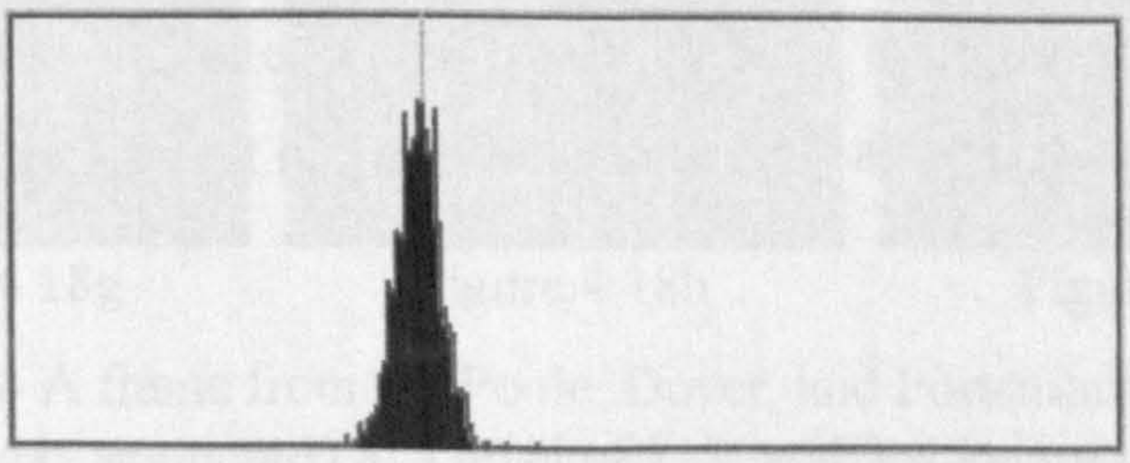

Figure 4.16e - Tile 4
Figure 4.16 – Tile histograms

The statistics calculated for each tile are the mean, μ, and standard deviation, σ as defined by equations 4.1 and 4.2.

$$\mu = \frac{1}{1024} \times \sum_{x=1}^{32} \sum_{y=1}^{32} P(x, y) \tag{4.1}$$

$$\sigma = \sqrt{\frac{\sum_{x=1}^{32} \sum_{y=1}^{32} (P(x, y) - \mu)^2}{1024}} \tag{4.2}$$

Although most of the graphs are of calm sea conditions the same distribution is seen in the histogram of the rougher sea in the Dover sequence, figure 4.10. Also noticeable from the tiles in figure 4.16 is that the sea range occupies a relatively central position in the grey level spectrum. To determine the grey-level range for the sea component of the histogram it is necessary to eliminate those grey levels above and below the sea range.

In rougher sea conditions these grey levels would contain a high grey level component from the white horses and a dark grey level component from wave troughs. As these values lie outside the characteristic grey-level range for the sea they need to be eliminated. To ensure the characteristic range of the sea is retained the peak of the histogram is used as the starting point and the measure of standard deviation used to determine the initial range for the sea. Stroud (2001) says that 68% of values occur within 1 standard deviation of the mean, 95% of values within 2 standard deviations, as shown in figure 4.17, and 99.7% of values within 3 standard deviations. To determine the most appropriate standard deviation value to use each has been applied to the test sequences and the results shown in figure 4.18.

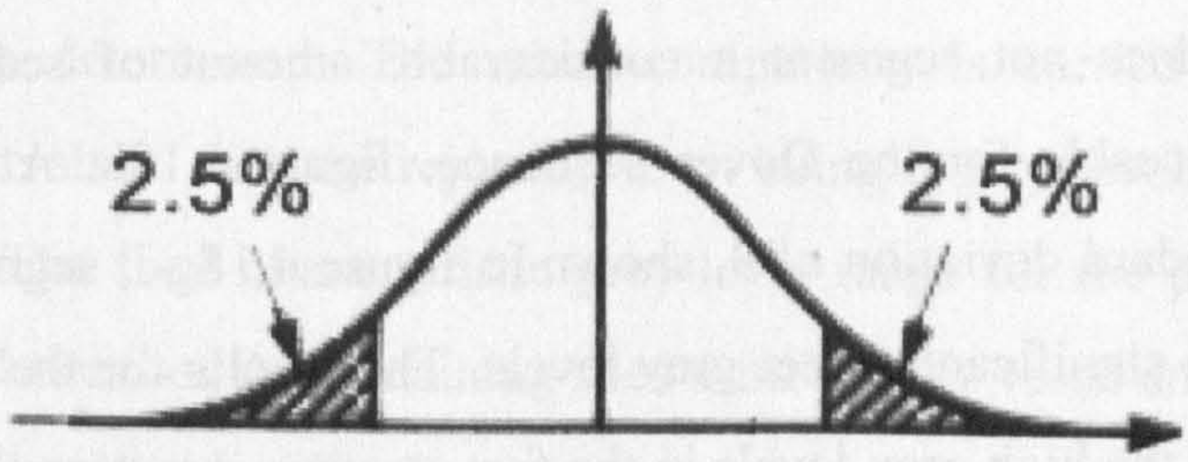

Figure 4.17 – Normal distribution showing the values outside 2 standard deviations from the mean

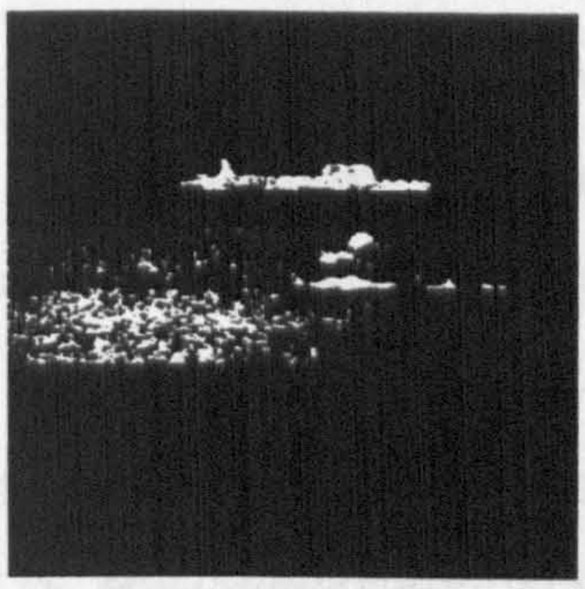

Figure 4.18a

Figure 4.18b

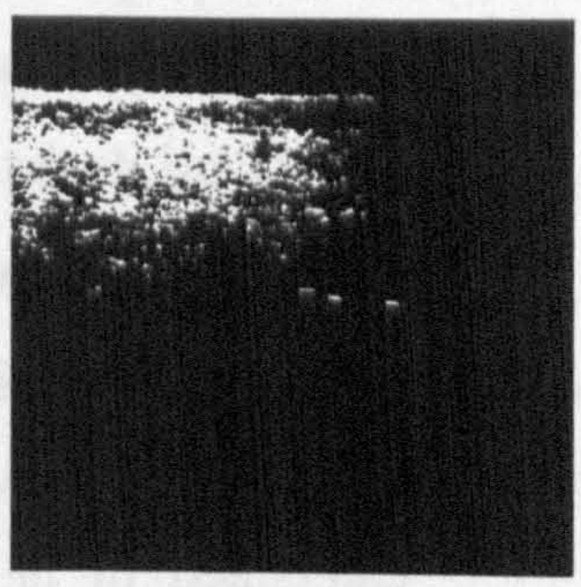

Figure 4.18c

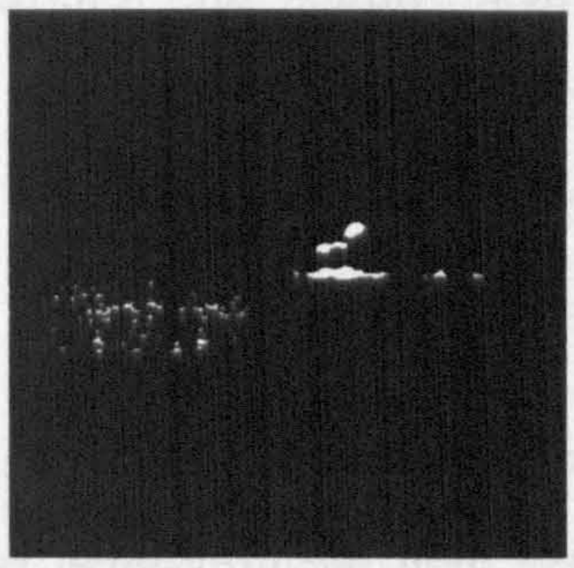

Figure 4.18d

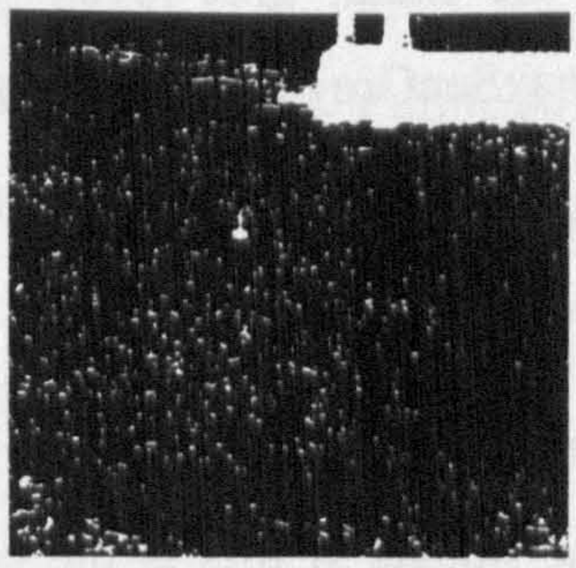

Figure 4.18e

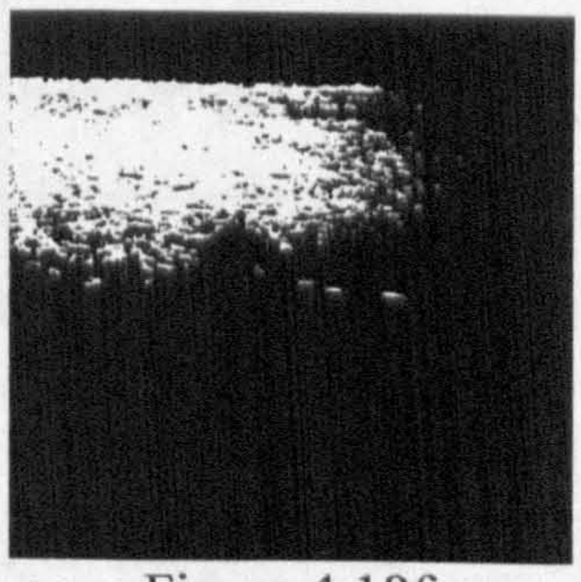

Figure 4.18f

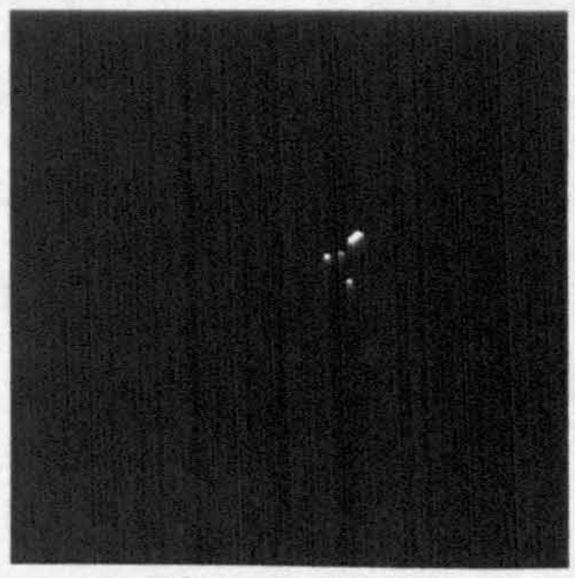

Figure 4.18g

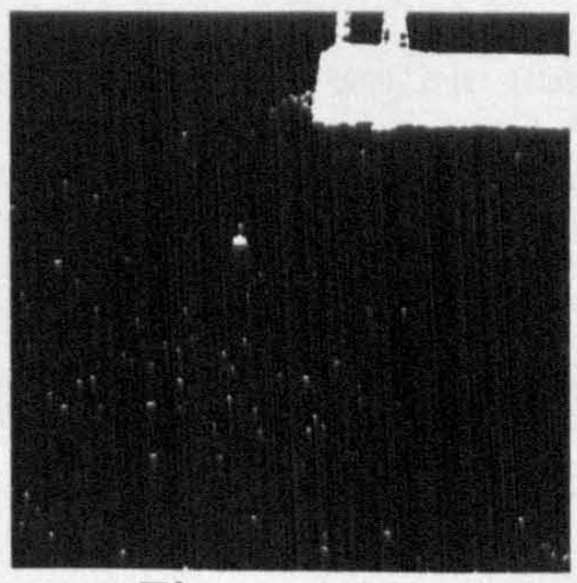

Figure 4.18h

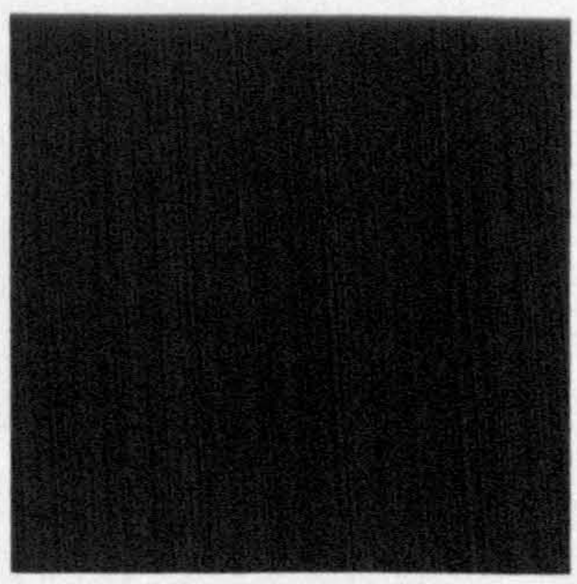

Figure 4.18i

Figure 4.18- A frame from the Poole, Dover, and Portsmouth sequences respectively segmented at a level of 1, 2, and 3 standard deviations respectively above and below the mean sea grey level

The standard deviation results show that 1 standard deviation, figure 4.18a-c, does not segment a considerable amount of sea. This is particularly noticeable for the Dover sequence, figure 4.18b. At the other extreme, a standard deviation of 3, shown in figure 4.18g-i, segments not only sea but also significant object grey levels. The results for the Poole sequence show that the high grey-levels in the fast moving object partly segmented in figure 4.18a but hardly at all in 4.18g. Although this is only an initial characterisation of the sea it is felt that both 1 and 3 standard deviations eliminate too few and too many grey levels respectively and so the value of 2 standard deviations is chosen for the initial characterisation. This measure allows the central 95% of the values to be retained thereby eliminating the lowest 2.5% of grey-levels values and the highest 2.5% of values (Stroud, 2001) as shown in figure 4.17. The initial sea range for each tile is calculated to be between 2 standard deviations below, and 2 standard deviations above the mean grey-level. These values are referred to as minSeaGrey and maxSeaGrey respectively.

Tile Rejection

It is possible that the location of a tile coincides with the position of one or more objects in the image. If this is the case then the grey-level range for that tile will not correctly represent that of the sea, it will represent the object, as can be seen in figure 4.16b. If this false grey-level range were used in the segmentation of the image then there is the potential for objects to be classed as sea and for area of sea to be classed as object. To counter this situation it is necessary to compare the initial grey-level ranges in each of the tiles.

The minSeaGrey value for each of the five tiles is compared and the median value calculated. The median is also calculated for the maxSeaGrey value for each of the five tiles. If any of the minSeaGrey values is greater than θ grey levels lower than the median then that value is rejected. The values

from two tiles can be rejected and a characteristic grey-level range for the pane still be determined. The same comparison is carried out for the maxSeaGrey values with the rejection level being θ grey levels higher than the median. The final characteristic grey-level range for the pane is set to between the lowest of the remaining tiles minSeaGrey value and the highest of the remaining tiles maxSeaGrey value.

The value of θ has been determined by recording the grey-level ranges for panes in two frames from the Poole and Dover test sequences. The grey-level ranges were calculated both manually using the histogram tool in the Adobe Photoshop software and using the algorithm developed here with various threshold values. Figures 4.19 shows the average error for each pane in frames from the Dover and Poole sequences for threshold values 15, 20, 25, 30, 35, and 40.

θ is set to the threshold giving the minimum overall percentage error between the actual (manual) and calculated (algorithm) grey-level ranges. From the results shown in figure 4.19, θ was set to 25. The data used to obtain this result can be found in Appendix A.

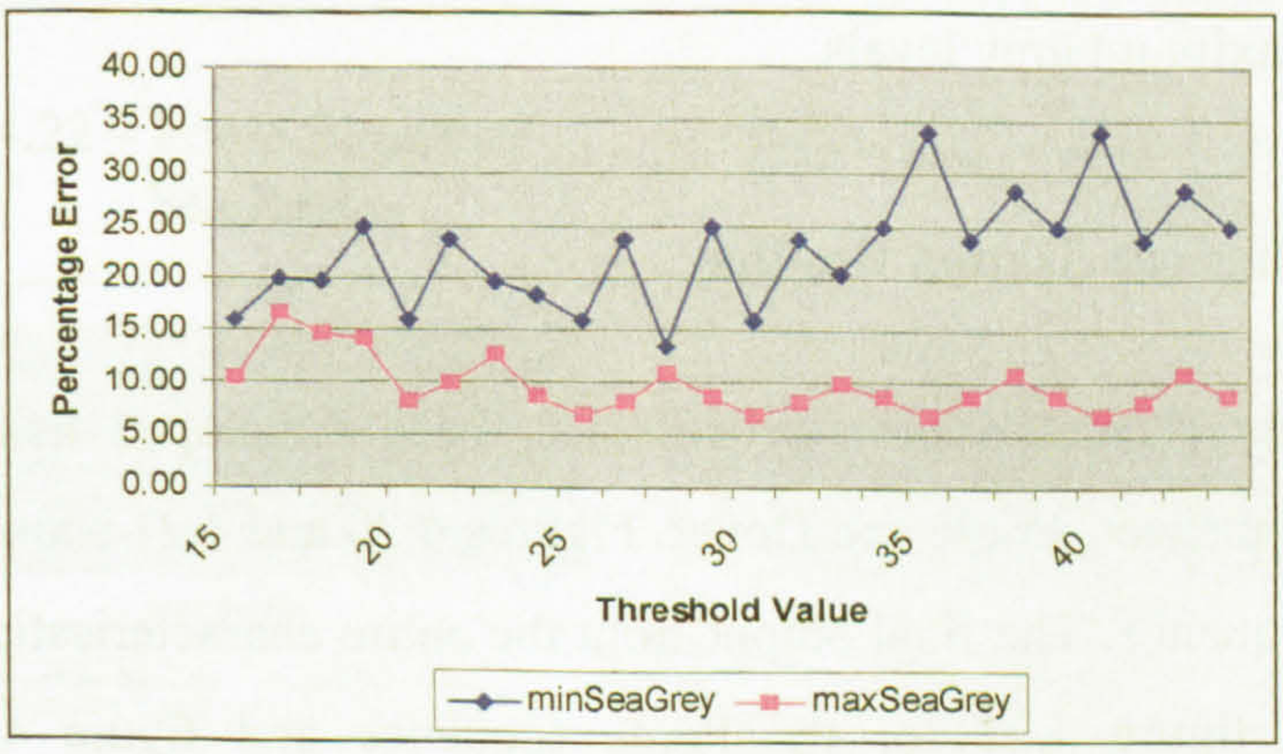

Figure 4.19 – Percentage error between actual and calculated sea grey level for threshold values 15, 20, 25, 30, 35, and 40

Pane Comparison

For the same reason that a tile coincides with the position of an object in the image, it is possible that a large object could have coincided with the majority of the tiles in a pane. This would result in the grey-level range for the pane being set to that of the object, which would be an incorrect characterisation. It is therefore necessary to compare the grey-level ranges across the four panes.

The panes are compared in the same way as the tiles. The minimum grey levels from the four panes are sorted into ascending order, as are the maximum grey levels. Adjacent ordered minimum values are then compared against the same value of θ. Although the panes are expected to have different grey-level ranges because of changes in sea state and illumination this difference is expected to be small, typically less than 20 grey levels. If the difference between two values is greater than θ then it is considered that the grey-level range for the pane has become corrupted, possibly by an object. The minimum grey-level for this pane is then set to be the same as that for the pane one higher in the ascending order. The same comparison is then carried out for the maximum grey level except that if the difference is greater than θ then the value will be set to one lower in the order of the maximum grey levels.

Characterisation Results

The characterisation process has been developed using two of the test sequences, Poole and Dover. Figures 4.20 and 4.21 show a image from each sequence. The final output from the entire characterisation process is shown in figure 4.22 for the Poole sequence and figure 4.23 for the Dover sequence. The Calculated Sea Ranges are those arrived at after the initial sea ranges for the tiles within each pane have been compared and following the comparison and adjustment of values in the four panes. The Actual Sea

Range figures for each pane have been arrived at manually by using the histogram tool in the Adobe Photoshop software and the Calculated Sea Range for each pane is the final result of the Sea Characterisation algorithm. The results shown in figures 4.22 and 4.23 were arrived at by averaging the values taken from every fifth frame from a sixty frame sequence.

Figure 4.20 – Poole Sequence

Figure 4.21 – Dover Sequence

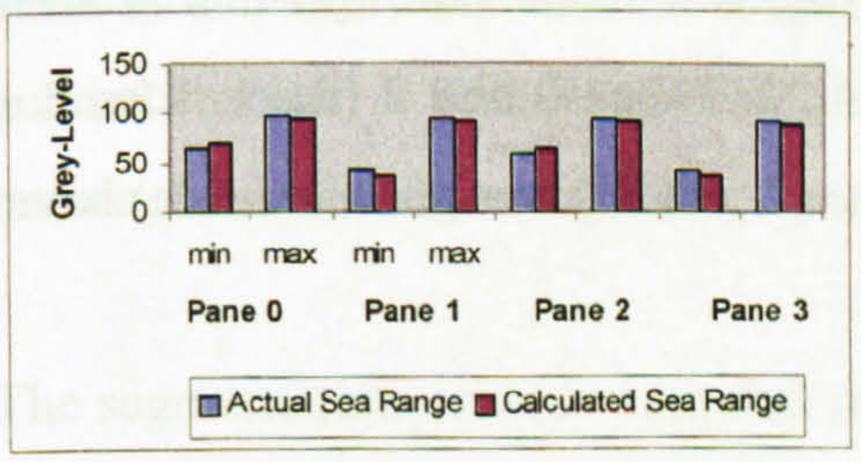

Figure 4.22 – Poole Sequence Sea Range

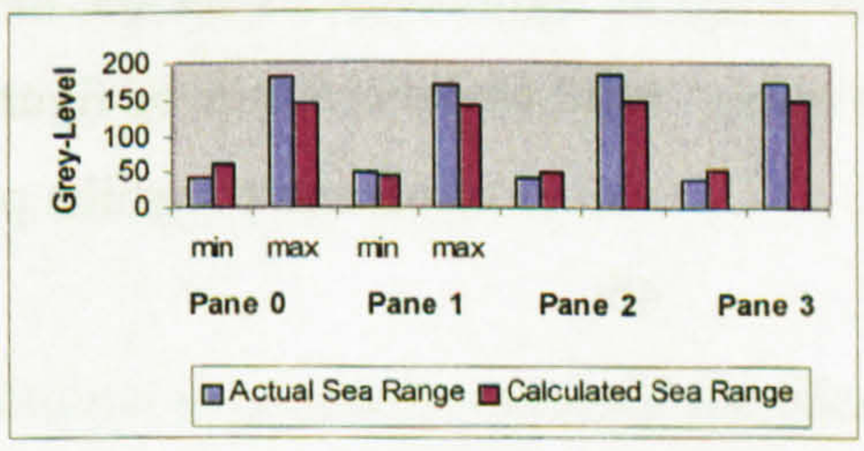

Figure 4.23 – Dover Sequence Sea Range

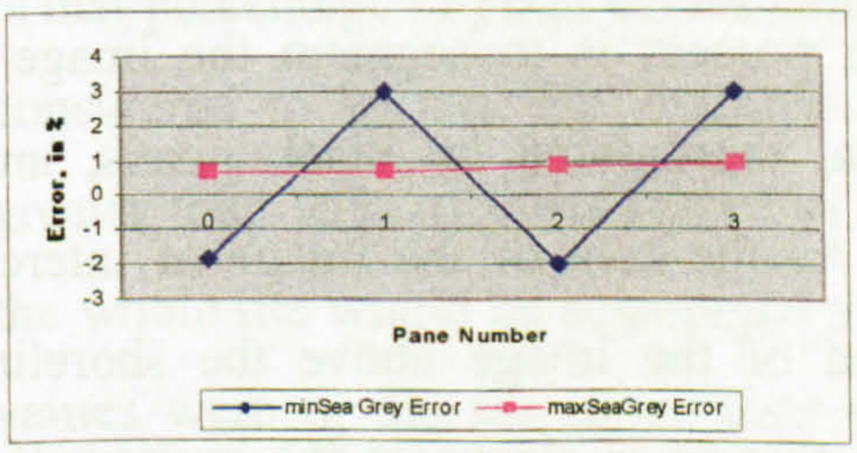

Figure 4.24 – Poole Sequence Characterisation Error

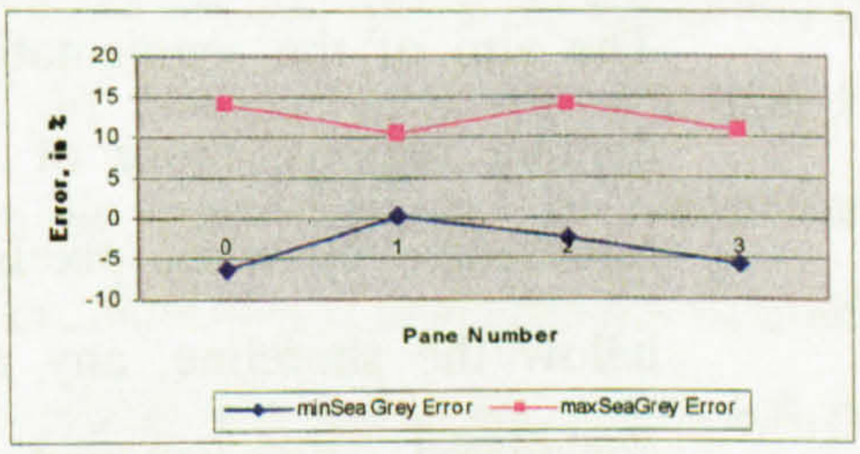

Figure 4.25 – Dover Sequence Characterisation Error

Figures 4.24 and 4.25 plot the average percentage error between the actual and calculated sea ranges for each of the panes in the image. It can be seen that the error for the Poole sequence is lower than that for the Dover sequence, particularly the maxSeaGrey error which shows a 14% error in panes 0 and 2 of the image from the Dover sequence. This is to be expected as the sea in the Dover sequence contains a wider range of grey-levels than the Poole sequence. The 2 standard deviation point used to determine the minSeaGrey and maxSeaGrey values will leave a greater number of grey-levels out of the characterisation than is true of the Poole sequence.

It should be noted that pane 1 of the Dover image only has errors of 0% and 10.5% respectively for the minSeaGrey and maxSeaGrey values. Given that this pane contains the large object in the top right of the image this shows the success of the fixed tile pattern and of the tile rejection stage of the process. Figure 4.22 also shows quite clearly the difference in illumination across the image. The right hand side of the image from the Poole sequence in figure 4.20 is darker than the left hand side and this is shown in the minSeaGrey values of figure 4.22. Panes 0 and 2 (the left hand side) have values of 65 and 60 whilst panes 1 and 3 (the right hand side) have the value 45.

4.4.2 Segmentation

The aim of the segmentation process is to segment the image into two distinct regions, those of sea, represented by black pixels, and object, represented by white pixels. As the area of the image of interest is that below the shoreline, any area of the image above the shoreline is not considered. The segmentation process is shown in the lower half of figure 4.12.

Each of the four panes is segmented in turn using the characteristic grey-level range determined by the Sea Characterisation process. It is assumed that a large proportion of the image will contain sea and therefore to segment the image pixel by pixel would be unnecessarily computationally intensive. Therefore the pane is further split into tiles of a fixed size, as shown in figure 4.26, before the segmentation process begins. The use of tiles enables a far less intensive decision to be made on the tile as a whole thus segmenting larger areas of the image at a time.

Although the images used in this system are real world images the three dimensional perspective is ignored. If a texture technique were being used to segment the images then perspective would have to be considered, as the texture element would need to be scaled according to the depth in the image. However, as the method used here is based on the statistical characterisation of the sea this is not necessary. Providing the size of the tiles used is sufficient to include representative grey levels for the sea the process will be successful. Any difference in grey level intensity between the foreground and background is addressed through the division of the image into quarters before the sea is characterised.

The segmentation process uses two additional parameters, a size for the tiles that the pane will be segmented into, and a percentage value for the decision level used to decide whether a tile is an object or sea tile. The decision level is that percentage of pixel values within a tile beyond which the whole tile is considered to be sea. For example, if the decision level was 5% then if greater than 95% of pixel values in the tile were within the sea range then the whole tile would be segmented as sea. Similarly if less than 5% of pixel values were in the sea range then the whole tile would be segmented as object. If neither of these cases is true for the tile then a decision on the whole tile cannot be made and segmentation of the tile is then carried out on a pixel by pixel basis. Those pixels whose grey-level value is within the sea

range for the pane are segmented to sea pixels and those pixels whose grey-level value is outside the sea range are segmented to object.

Segmentation Parameters

To arrive at the optimal segmentation for the tile the values for the tile size and decision level parameters have been derived as a pair. Both the Poole and Dover test sequences were used to ensure the consideration of different sea states.

As the aim of this process is to segment the sea region from any object region that may be present in the image the tile needed to be of sufficient size to contain a true representation of the sea. If the tile size were too small there would be an increased possibility that the low number of pixels in the tile lay outside the characteristic sea range for the pane. This would result in the false segmentation of the tile. This is particularly true of sea states such as that in figure 4.26 from the Dover sequence that contain a wide variety of grey-levels. If the tile size were too big then small objects of only a few pixels could be lost as the tile was segmented to the sea region.

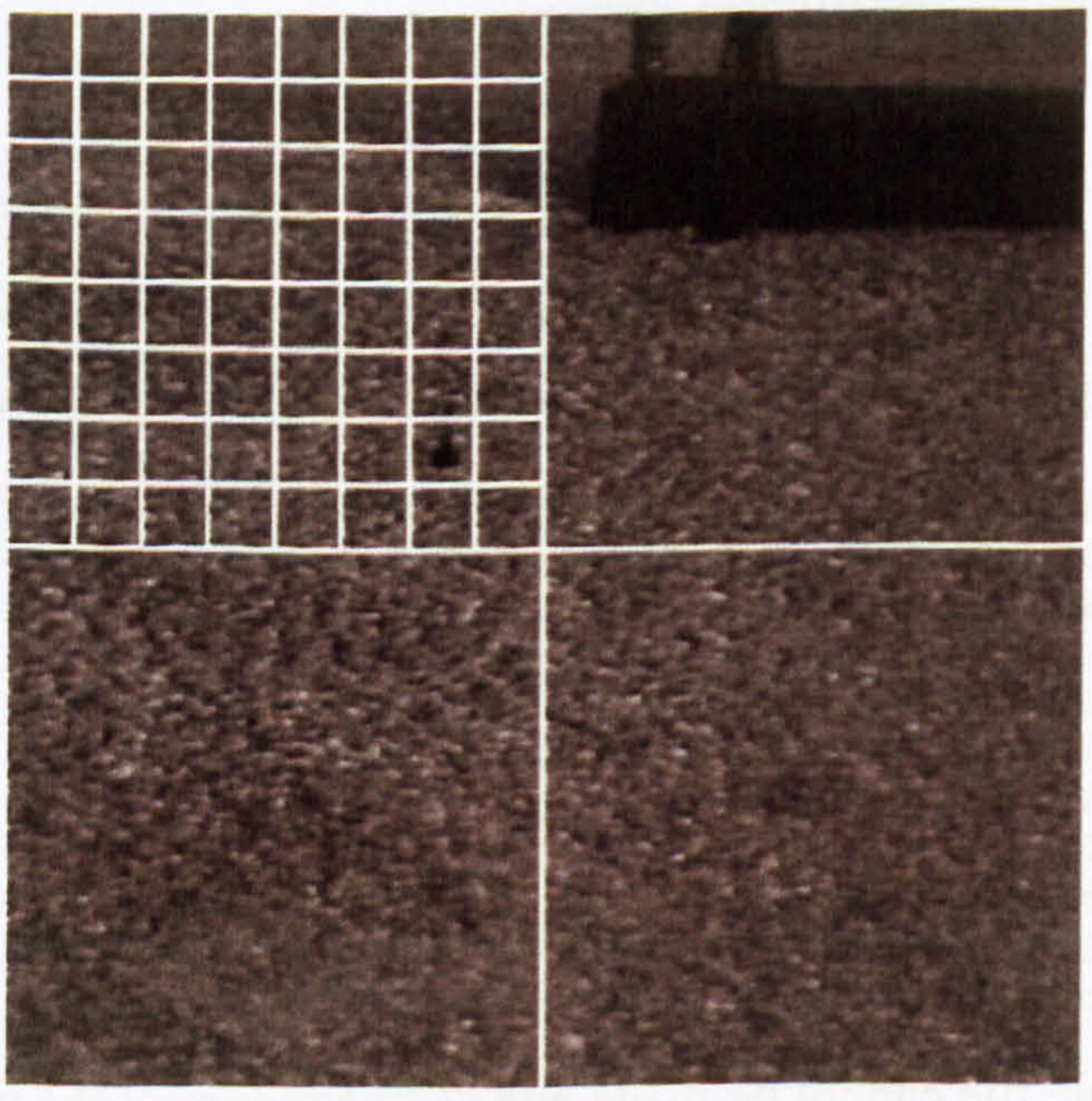

Figure 4.26 – 32x32 Pixel Tiles in Pane ready for Segmentation

The tile sizes considered were 16x16, 32x32, and 64x64 and the decision levels considered were 5%, 10%, 15%, and 20%. The following method was used to determine the values for tile size and decision level and the results for all the values considered can be found in Appendix B.

Every fifth image in a filtered 100 frame sequence had a grid of the tile size being considered laid over it. Each tile was then manually observed and recorded as being in one of the following three categories:

- Object Tile
- Sea Tile
- Undecided

This was repeated for the 100 frame sequences after they had been segmented by the algorithm presented here. The results were plotted and the percentage of correctly segmented tiles calculated for each combination of tile size and decision level. These results are shown in figures 4.27 for the Dover sequence and 4.28 for the Poole sequence. The data for all the results can be found in Appendix B. Each of the three categories has been calculated separately with the manually recorded number of tiles per category being 100%. As the number of Object and Undecided tiles are relatively small, typically between 12 and 20 Object tiles in the Poole sequence, the error in segmentation appears to be very high. This is unrepresentative of the operation of the algorithm but allows for a clearer differentiation to be seen between the parameter values being evaluated.

The parameter values to be used are those that give the highest percentage of correctly segmented tiles averaged across both test sequences. If a single sequence were used or the values yielding the highest percentage of correctly segmented tiles then the algorithm would be tailored to a specific image sequence. This would be wrong as the algorithms developed here are

to be used in a variety of sea conditions. This can be seen in figure 4.27e showing the result for a tile size of 16x16 pixels and a decision level of 15%. This combination of parameters for the Dover sequence shows the percentage of correctly segmented object tiles only drops below 80% in one frame and the undecided tiles in four frames. However, if the equivalent results from the Poole sequence are observed, figure 4.28e, it can be seen that the percentage of correctly segmented object tiles drops below 80% in fourteen frames and the undecided tiles in twelve frames.

The combination of parameter values giving the highest percentage of correctly segmented tiles in all categories over both test sequences is a tile size of 32x32 pixels and a decision level of 10%. Figure 4.29 shows the average results for these parameter values for both sequences. This figure shows the percentage of correctly segmented sea tiles is over 97% in all frames, the figure for object tiles is over 89% in all frames, and over 70% in all except one frame for the undecided tiles. The data and graphs for the averages of all combinations can also be found in Appendix B.

The figure of only 70% of undecided tiles being correctly segmented is lower than expected and it is considered that further processing is necessary to increase this result. A method of region growing will address this and a further consequence of segmentation.

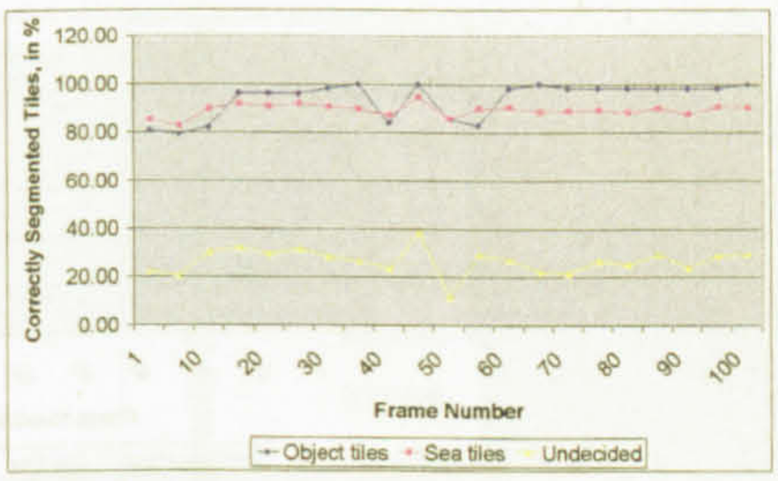

Figure 4.27a -Tile Size 16x16 Decision Level 5%

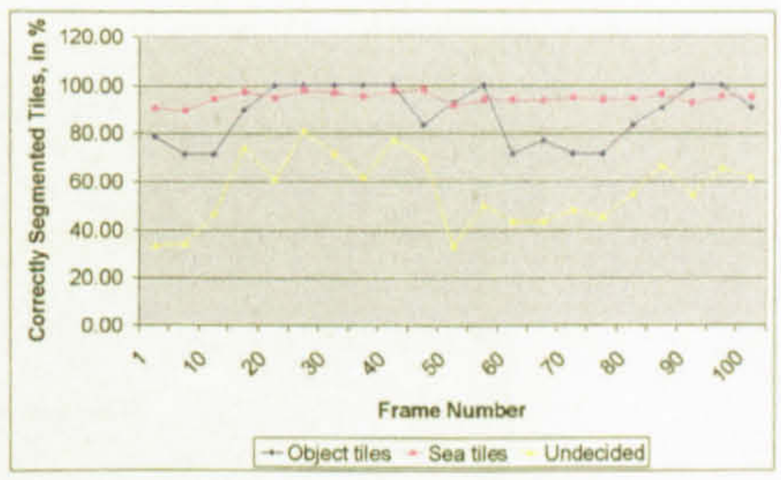

Figure 4.27b - Tile Size 32x32 Decision Level 5%

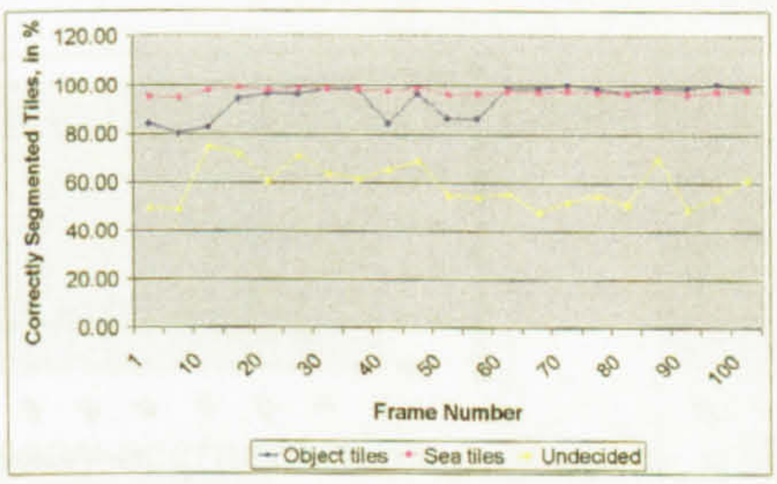

Figure 4.27c - Tile Size 16x16 Decision Level 10%

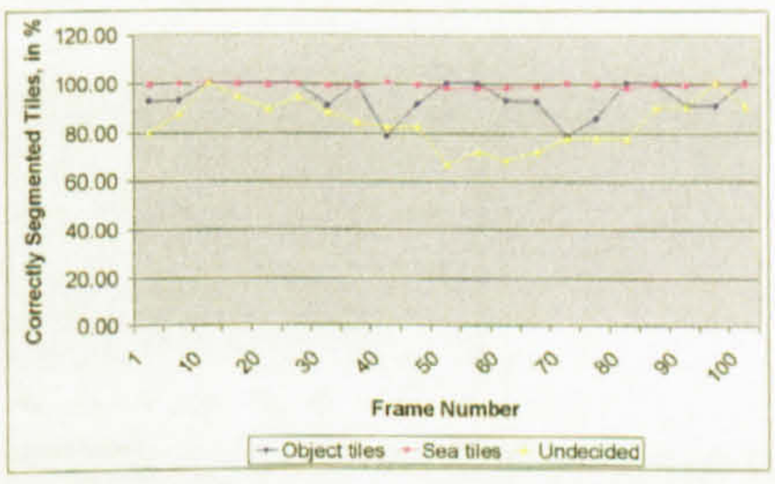

Figure 4.27d - Tile Size 32x32 Decision Level 10%

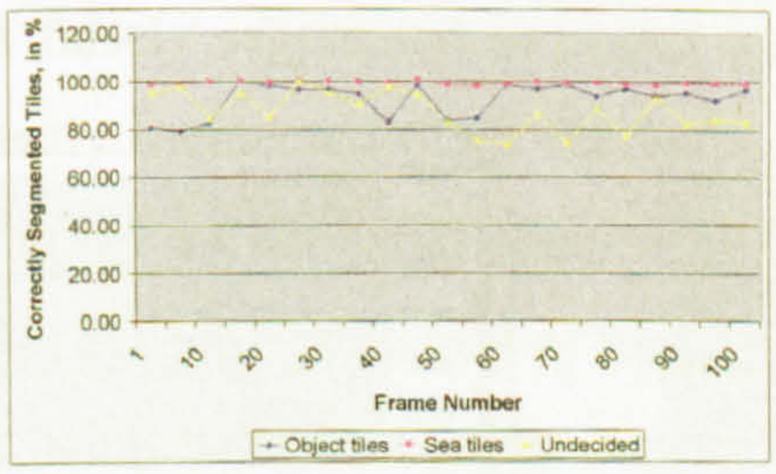

Figure 4.27e - Tile Size 16x16 Decision Level 15%

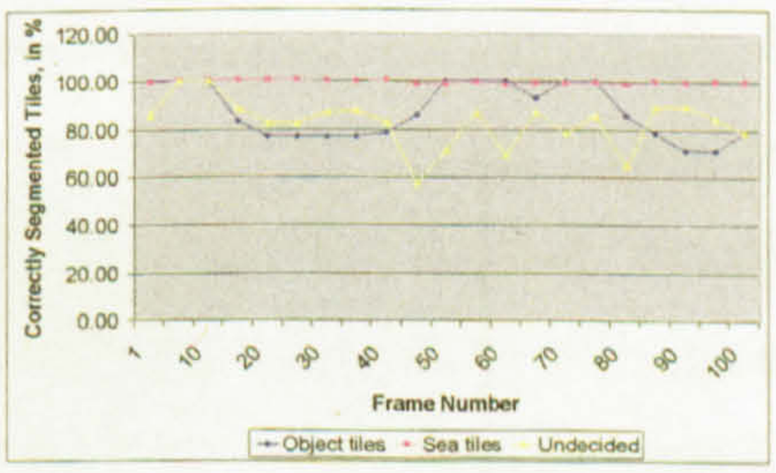

Figure 4.27f - Tile Size 32x32 Decision Level 15%

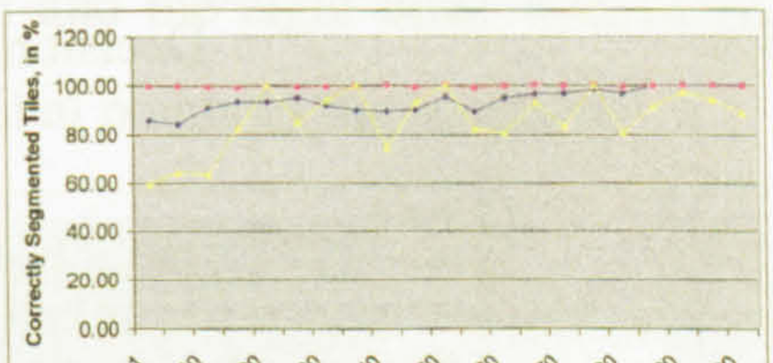

Figure 4.27g - Tile Size 16x16 Decision Level 20%

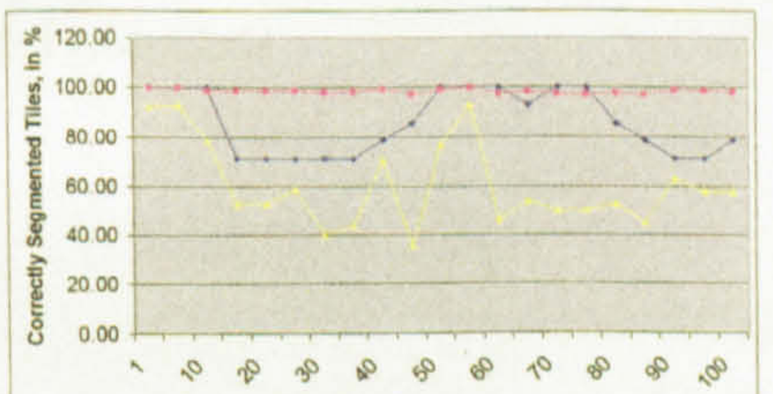

Figure 4.27h - Tile Size 32x32 Decision Level 20%

Figure 4.27 – Dover Sequence Tile Size and Decision Level Results

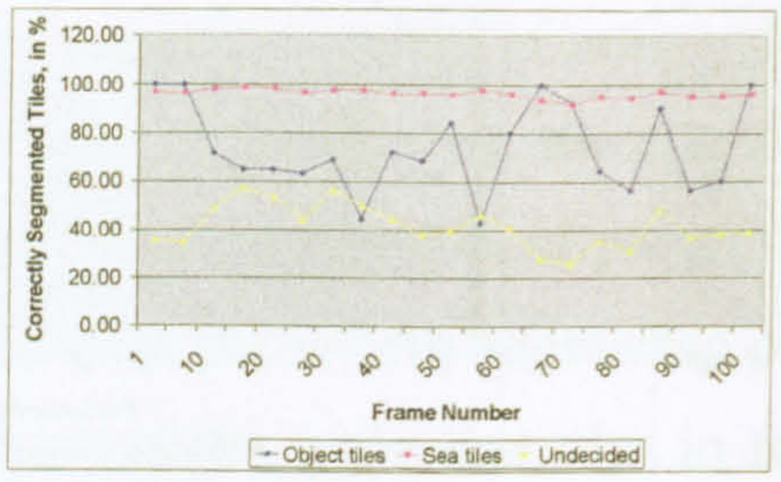

Figure 4.28a - Tile Size 16x16
Decision Level 5%

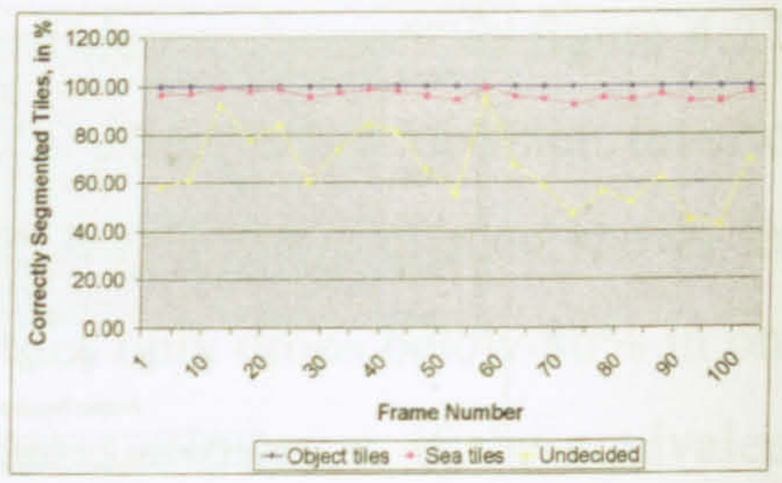

Figure 4.28b - Tile Size 32x32
Decision Level 5%

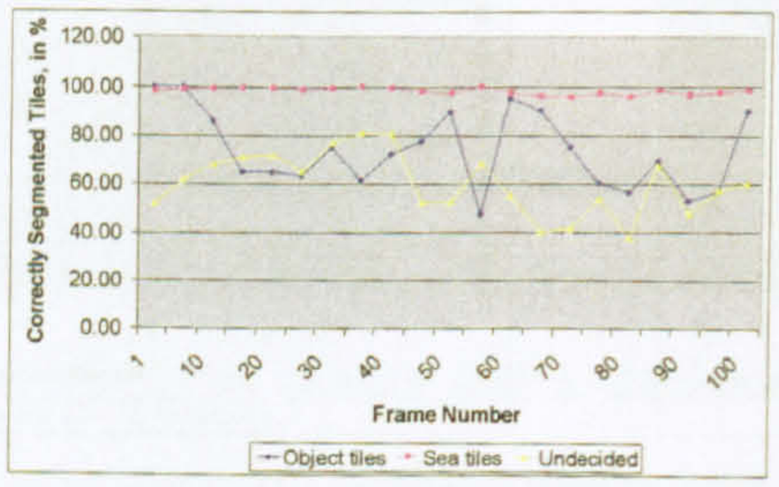

Figure 4.28c - Tile Size 16x16
Decision Level 10%

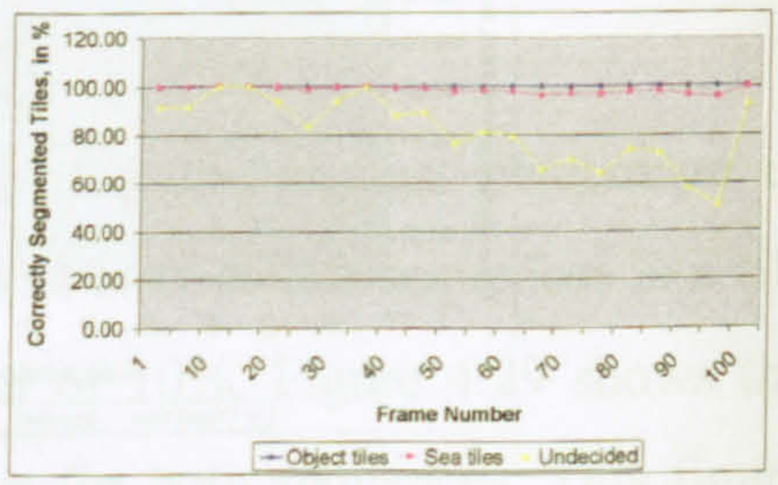

Figure 4.28d - Tile Size 32x32
Decision Level 10%

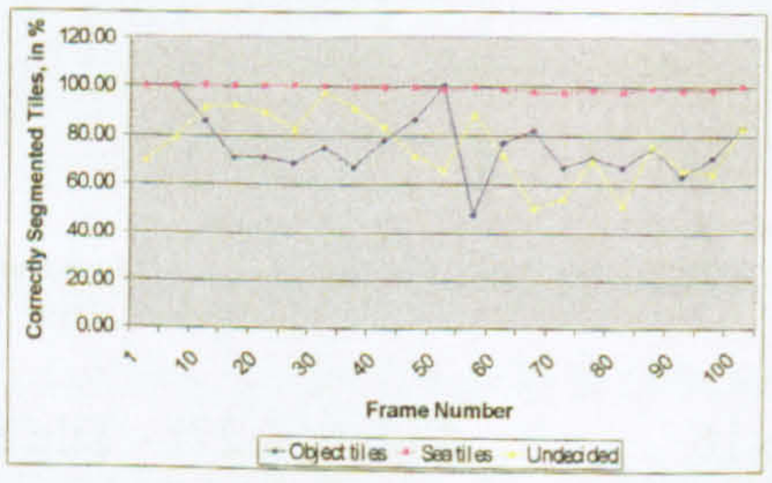

Figure 4.28e - Tile Size 16x16
Decision Level 15%

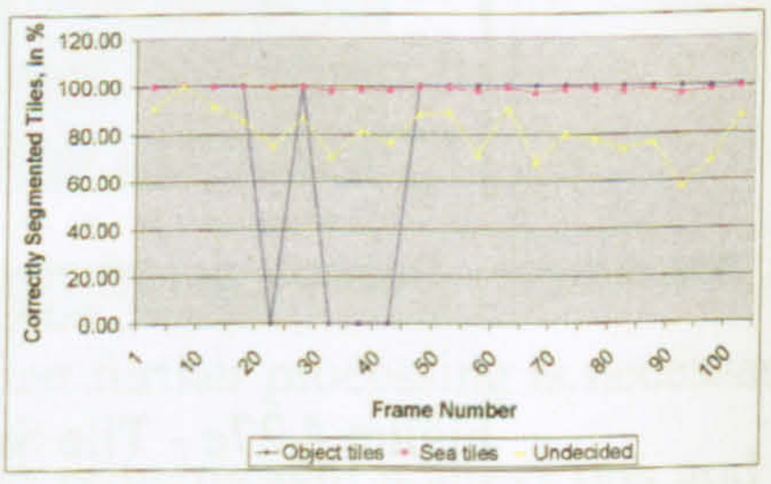

Figure 4.28f - Tile Size 32x32
Decision Level 15%

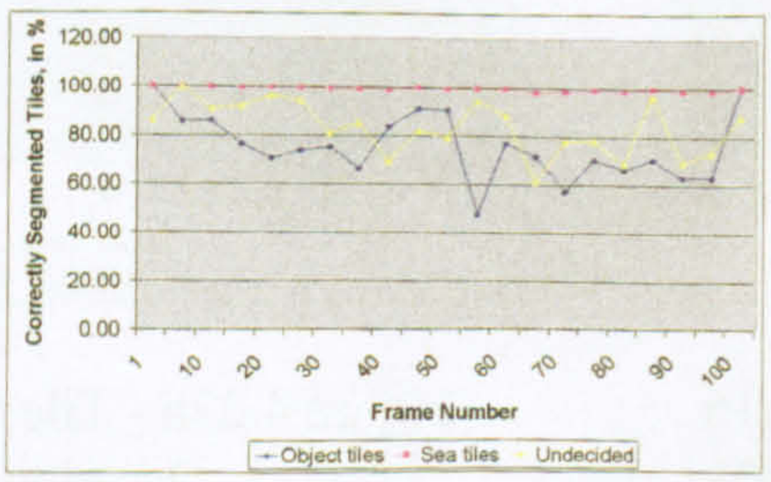

Figure 4.28g - Tile Size 16x16
Decision Level 20%

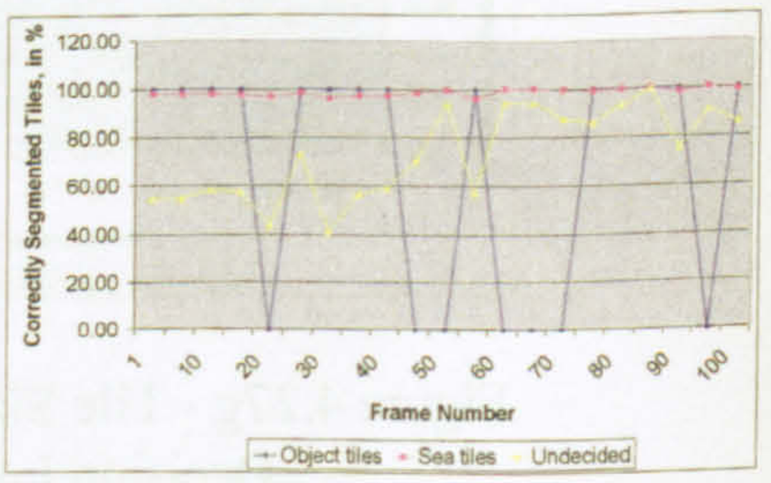

Figure 4.28h - Tile Size 32x32
Decision Level 20%

Figure 4.28 – Poole Sequence Tile Size and Decision Level Results

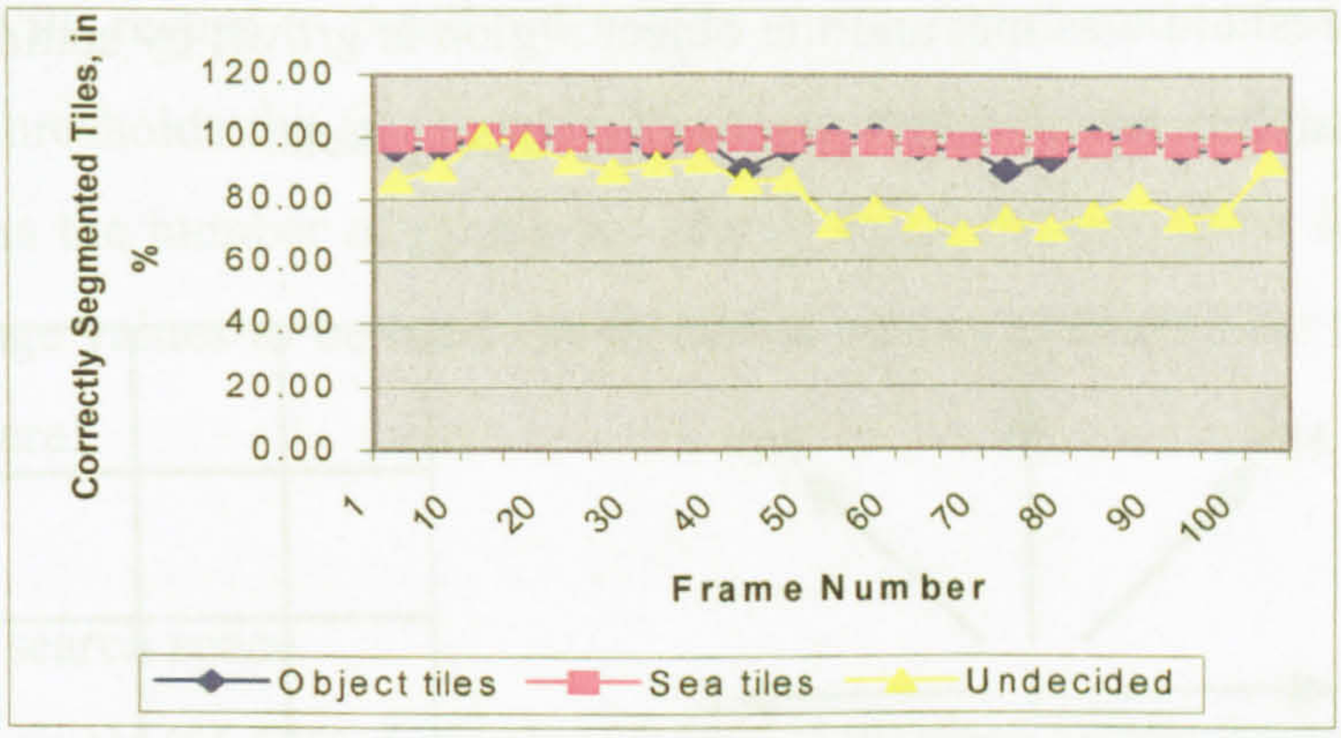

Figure 4.29 – Average Results for 32x32 Tile Size and 10% Decision Level

Region Growing

In many segmentation processes, objects can become fragmented (Owens et al, 2002) and some form of region growing is required to rejoin the objects. Region growing can also be used to remove small areas of sea that have been falsely segmented as object and therefore increase the percentage of correctly segmented tiles. The method of region growing used is similar to that of an 8-directional chain code as described by Gonzalez and Woods (1992) except that instead of using it to describe the boundary of the region it is being used to confirm a pixel's segmentation in a particular region. Nine directions have been used in this method as shown in figure 4.30, where C is a central direction and considers the pixels surrounding the target pixel and the eight other directions are labelled as the cardinal and inter-cardinal points of the compass.

The method involves, for each object pixel in the segmented image, a count of the number of object pixels in each of the nine directional search spaces. Figure 4.31 shows the search space for the SW direction. If a threshold value for the number of object pixels found is not met in any direction then the object pixel under test is considered to be a falsely segmented pixel and it is reclassified as a sea pixel. If the threshold is met then the pixel is considered to be a part of an object in all the directions in which the

threshold was met and the object region is grown by setting all the pixels in that directional search space to be object pixels.

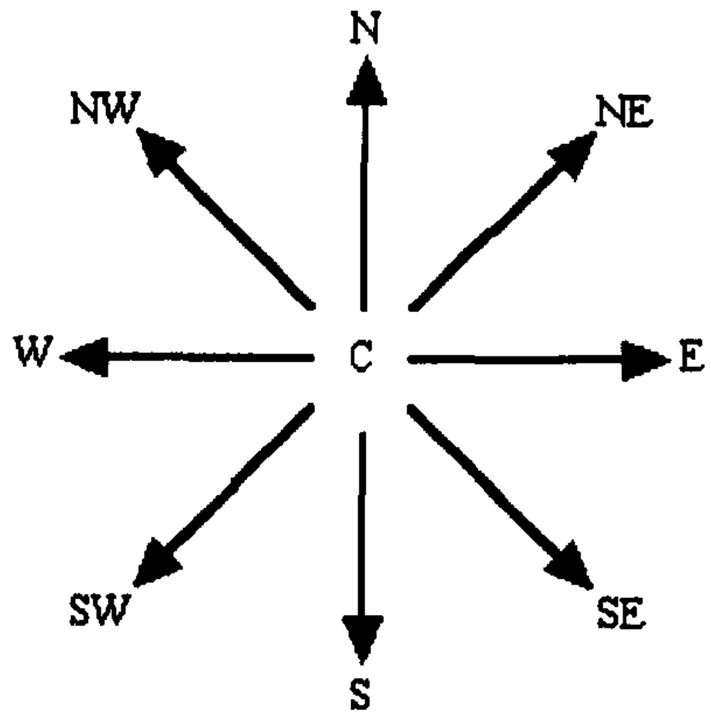

Figure 4.30 – 9-Way Directional Chain Code

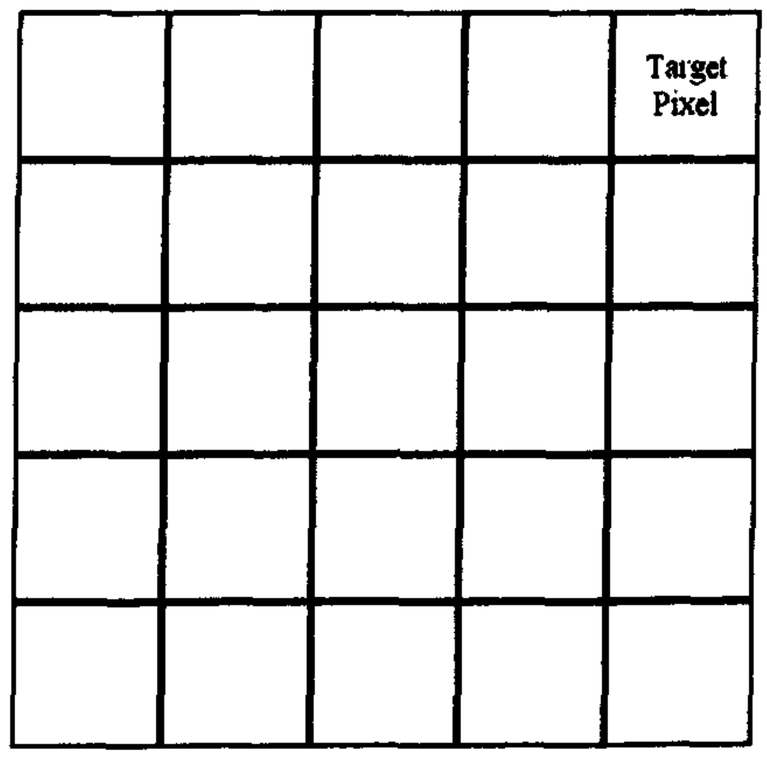

Figure 4.31 – SW Search Space

Region Growing Parameters

Two parameters are required for the region growing process. The size of the search space and the threshold value. These have been derived as a pair, in the same way as with the parameters for the segmentation method, to give the minimum number of falsely segmented object tiles where an object is a collection of white pixels completely surrounded by sea pixels.

If the size of search space is large then multiple objects could be merged into a single object and if the search space is small then correctly segmented object fragments could be falsely reclassified as sea pixels. Even numbered search spaces such as 4x4 or 6x6 were not considered as it would not be possible to locate the existing object pixel in the centre of the search space when considering the Central direction in figure 4.30. The search space sizes evaluated were 3x3, 5x5, and 7x7 pixels.

As the resolution of the region growing is at the pixel level the threshold value was determined to be a specific number of pixels within the search

space. With regard to threshold values it was found that above and below certain thresholds the false reclassification of pixels was 100% as in figure 4.32d. As the number of pixels per search space did not allow for the same percentage values to be used the threshold values evaluated for each search space were:

- 3x3 search space

  5 pixels (55.5%), 6 pixels (66.6%), 7 pixels (77.7%), 8 pixels (88.8%)

- 5x5 search space

  18 pixels (72%), 20 pixels (80%), 22 pixels (88%), 23 pixels (92%)

- 7x7 search space

  35 pixels (72%), 39 pixels (80%), 43 pixels (88%), 45 pixels (92%)

Both the Poole and Dover test sequences were used in the derivation of the final values and the results of the experiments can be seen in figure 4.32 and figure 4.33.

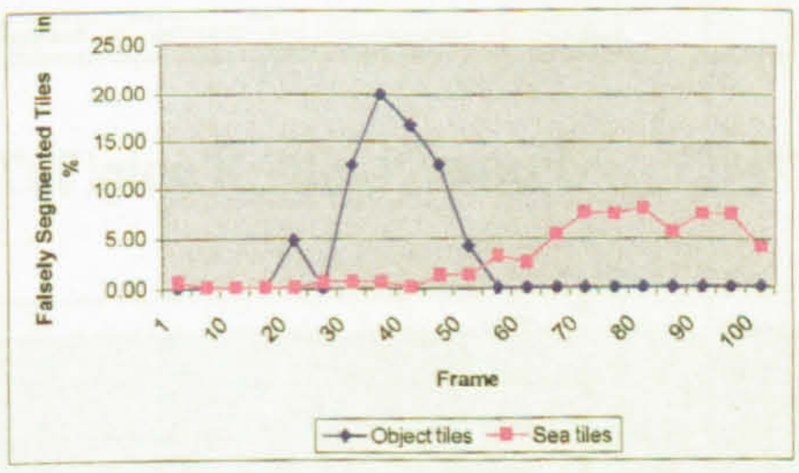

Figure 4.32a – Poole 3x3 72%

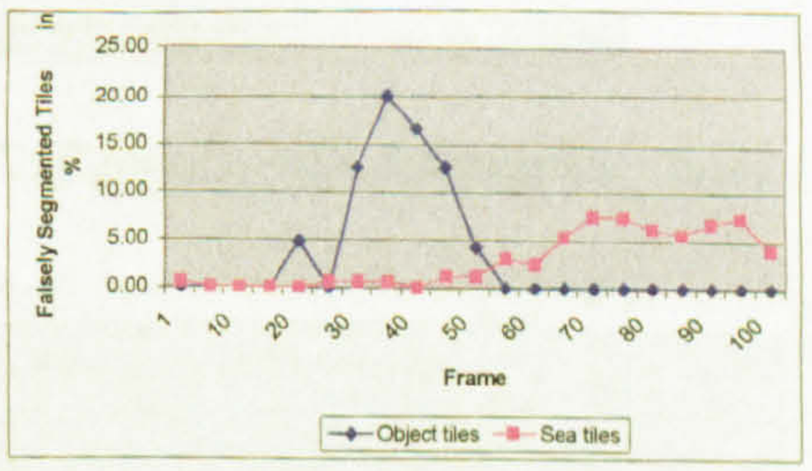

Figure 4.32b – Poole 3x3 80%

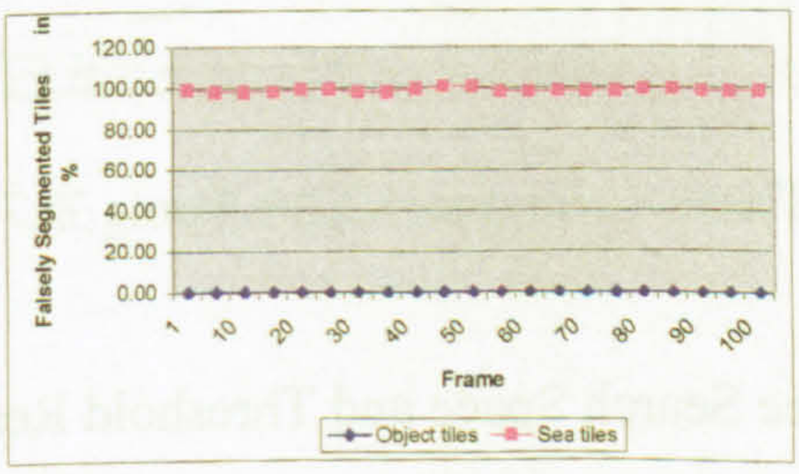

Figure 4.32c – Poole 3x3 88%

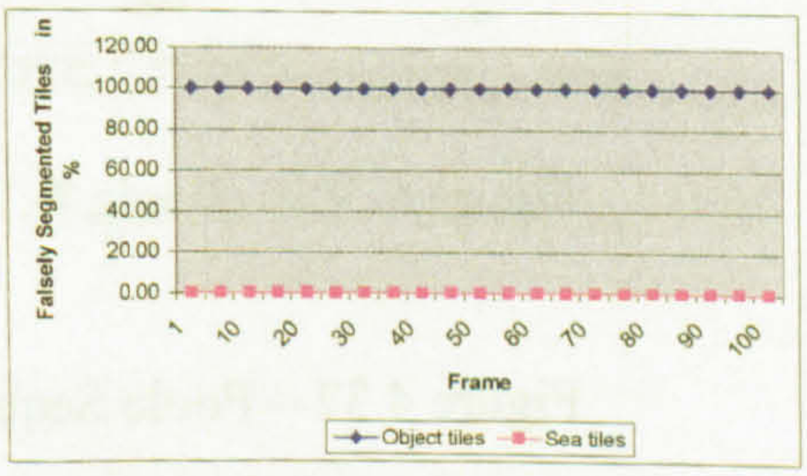

Figure 4.32d – Poole 3x3 92%

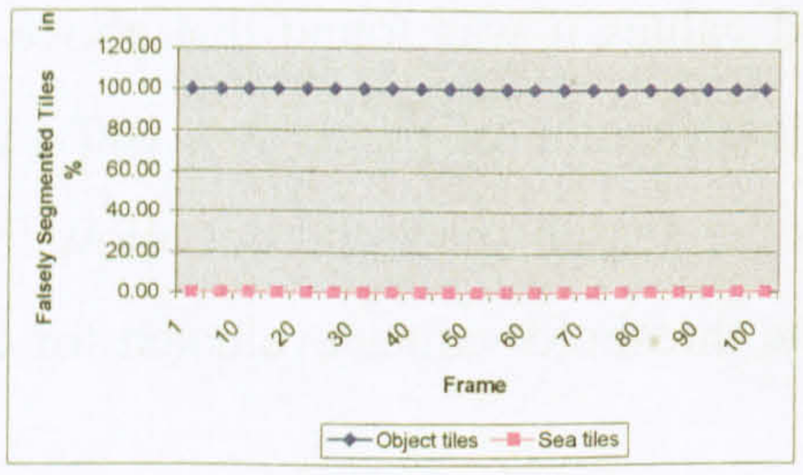

Figure 4.32e – Poole 5x5 72%

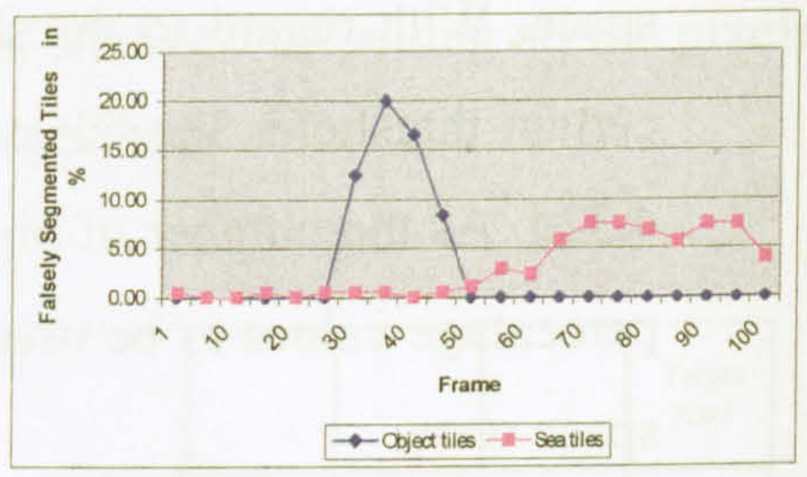

Figure 4.32f – Poole 5x5 80%

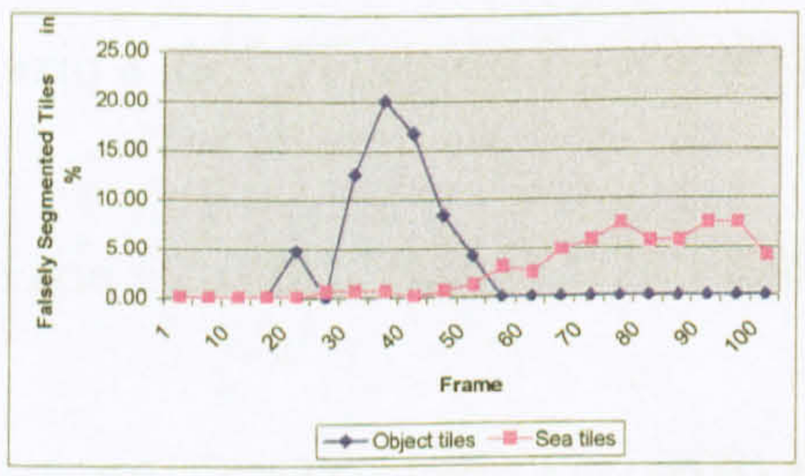

Figure 4.32g – Poole 5x5 88%

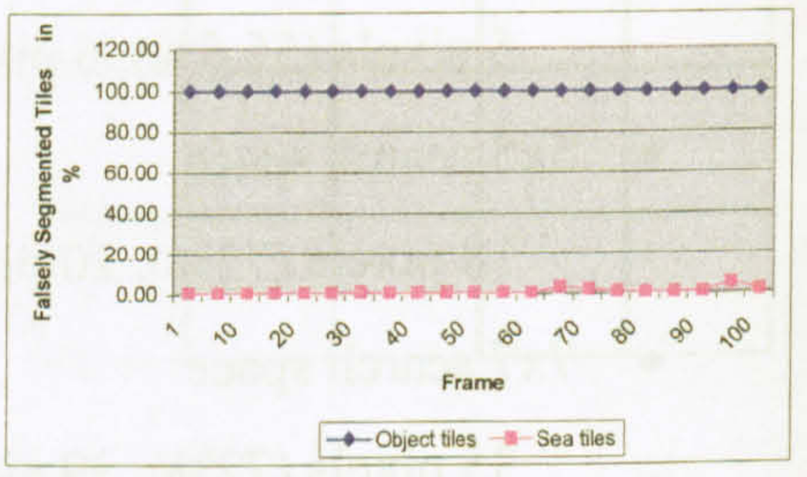

Figure 4.32h – Poole 5x5 92%

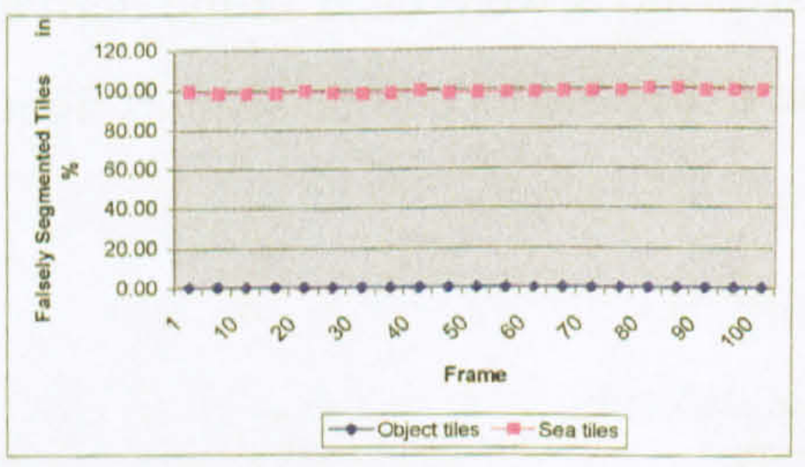

Figure 4.32i – Poole 7x7 72%

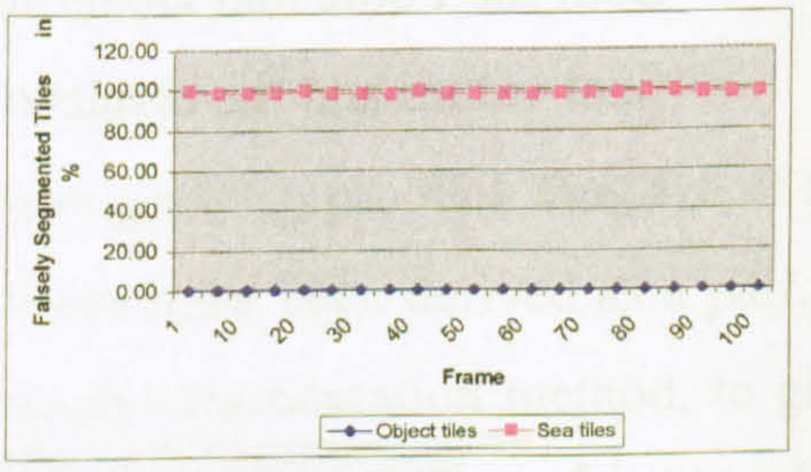

Figure 4.32j – Poole 7x7 80%

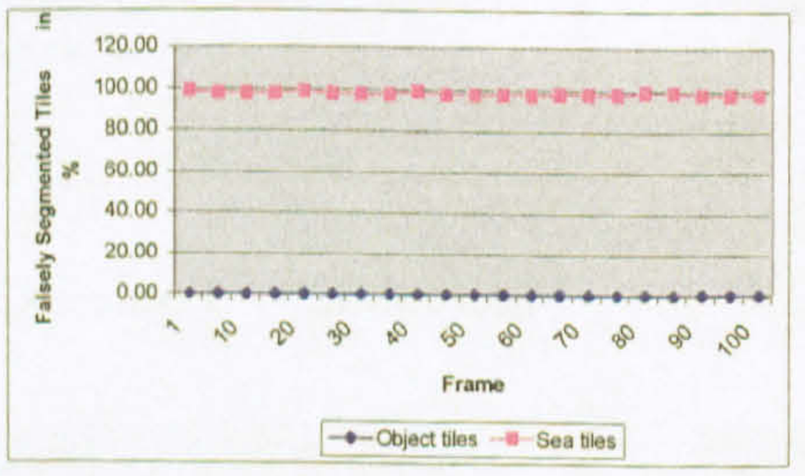

Figure 4.32k – Poole 7x7 88%

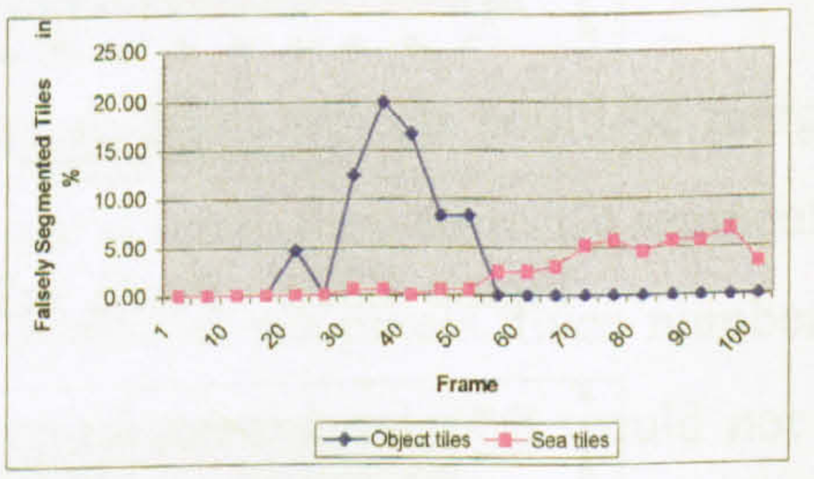

Figure 4.32l – Poole 7x7 92%

Figure 4.32 – Poole Sequence Search Space and Threshold Results

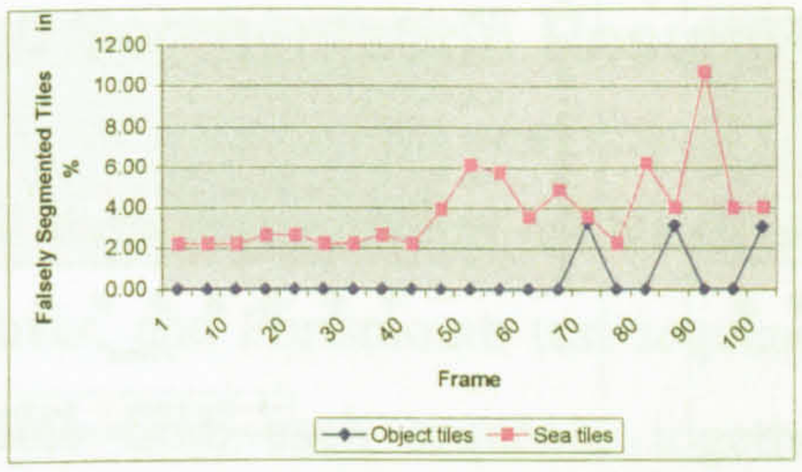

Figure 4.33a – Dover 3x3 72%

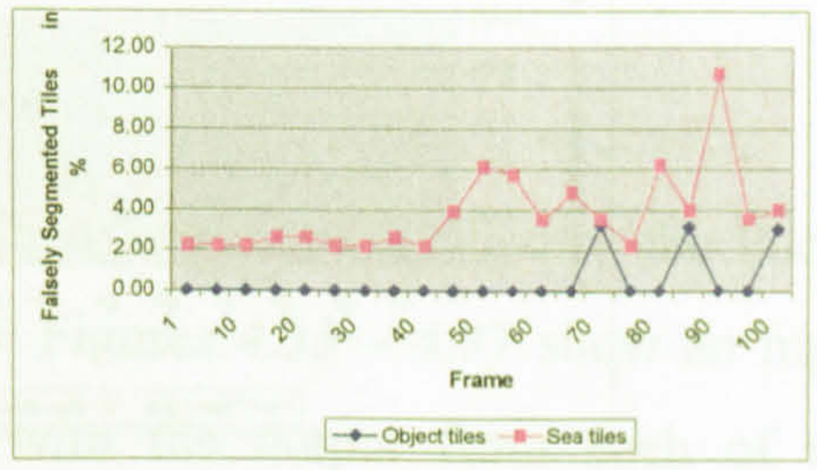

Figure 4.33b – Dover 3x3 80%

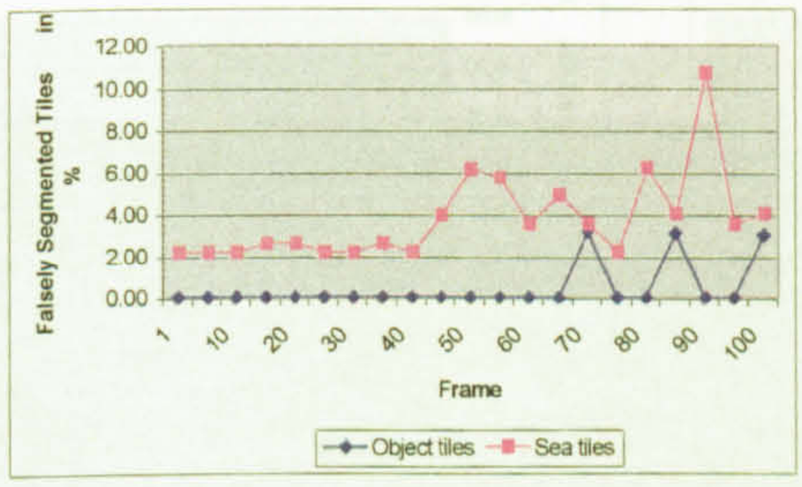

Figure 4.33c – Dover 3x3 88%

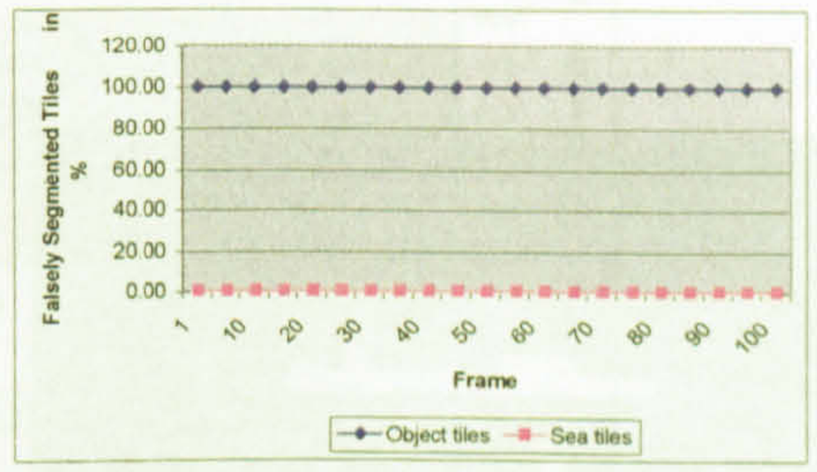

Figure 4.33d – Dover 3x3 92%

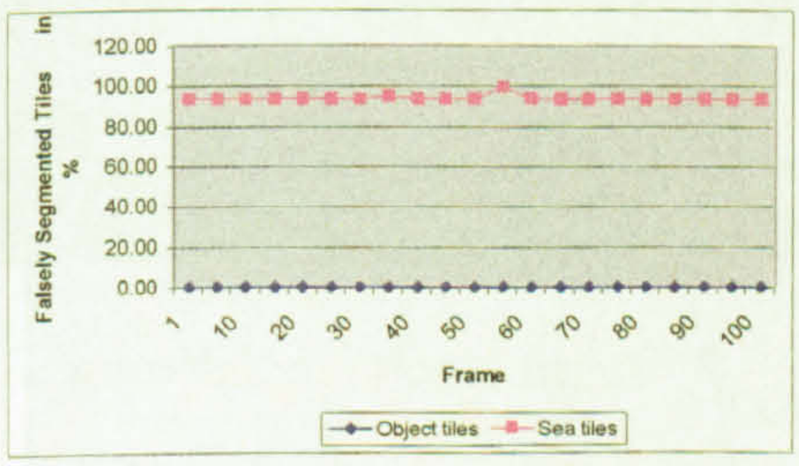

Figure 4.33e – Dover 5x5 72%

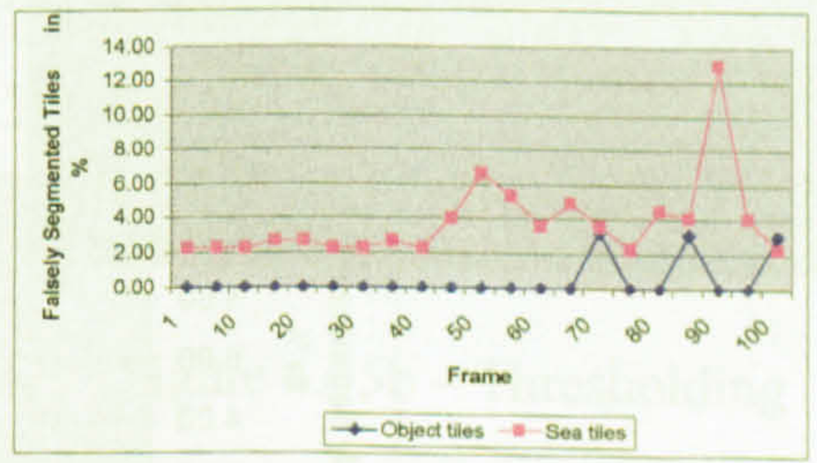

Figure 4.33f – Dover 5x5 80%

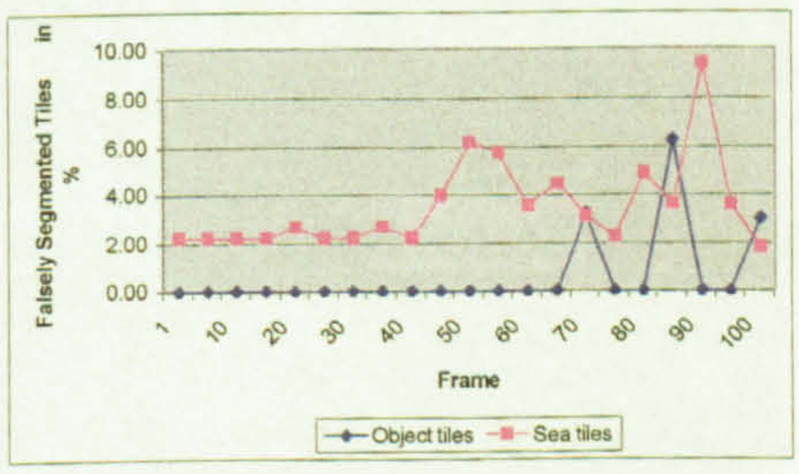

Figure 4.33g – Dover 5x5 88%

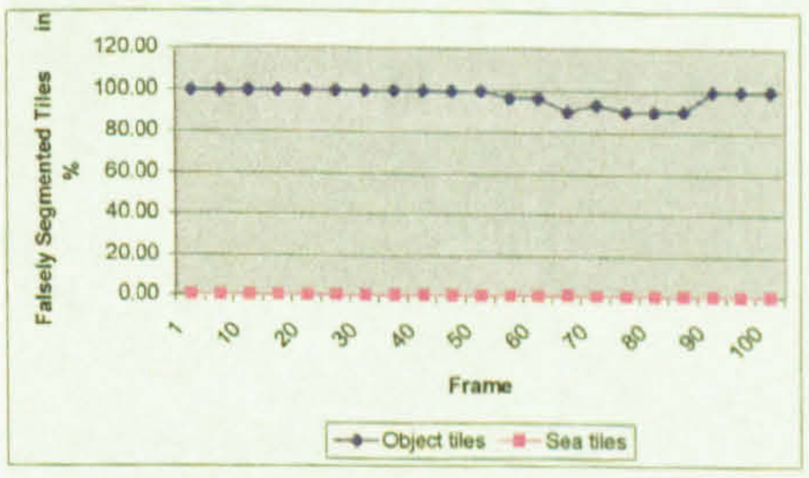

Figure 4.33h – Dover 5x5 92%

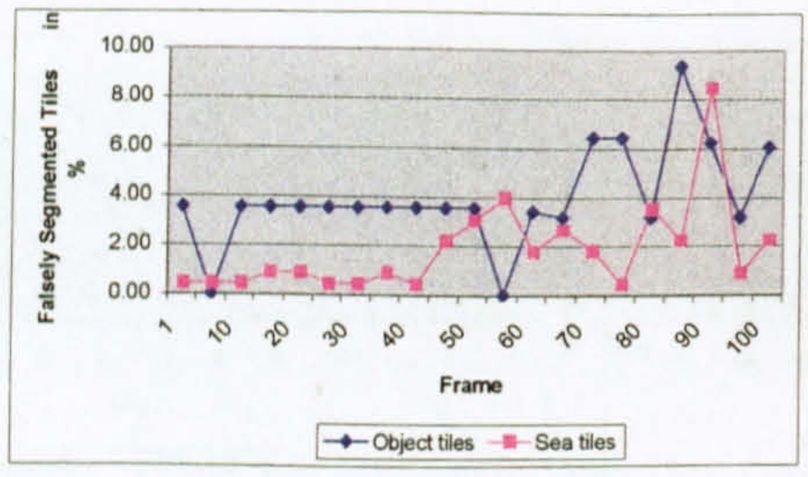

Figure 4.33i – Dover 7x7 72%

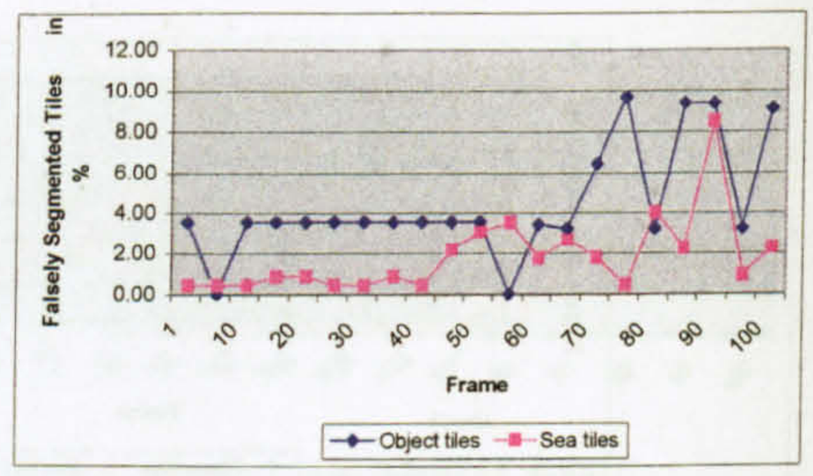

Figure 4.33j – Dover 7x7 80%

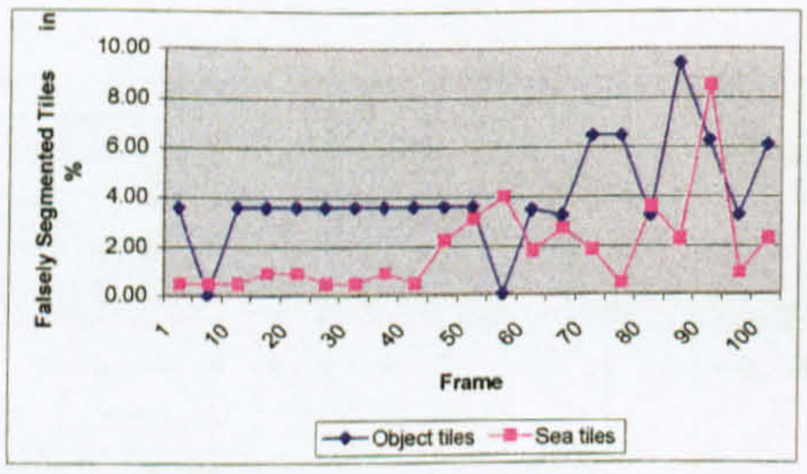

Figure 4.33k – Dover 7x7 88%

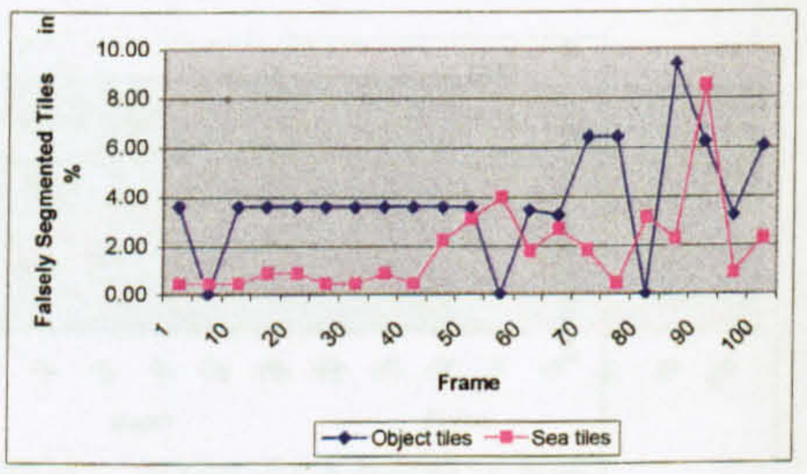

Figure 4.33l – Dover 7x7 92%

Figure 4.33 – Dover Sequence Search Space and Threshold Results

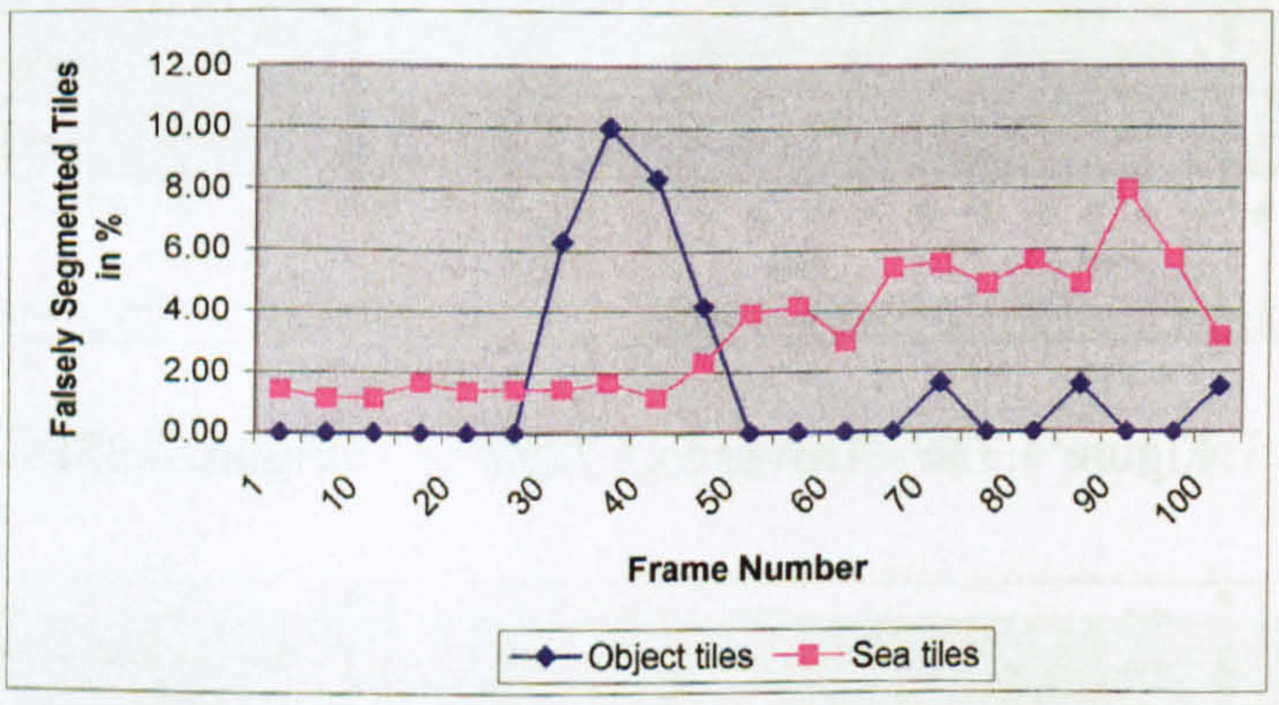

Figure 4.34 – Average Results for 5x5 Search Space and 80% Threshold

The average of the results for the two test sequences, Poole and Dover, used in the evaluation were calculated and the data and graphs for all of these can be found in Appendix B. The search space and threshold combination giving the lowest average false segmentation rate chosen for use in this algorithm is the 5x5 search space and the 80% threshold. This average result is shown in figure 4.34. The percentage of falsely segmented tiles is 10% or less in all frames for object tiles and less than 8% for sea tiles.

4.5 Segmentation Results

All three segmentation methods described have been applied to the Poole, Dover, and Portsmouth test sequences. Figures 4.35 – 4.37 show an input frame from each sequence together with the output from each of the thresholding, frame differencing, and sea characterisation segmentation methods for that frame.

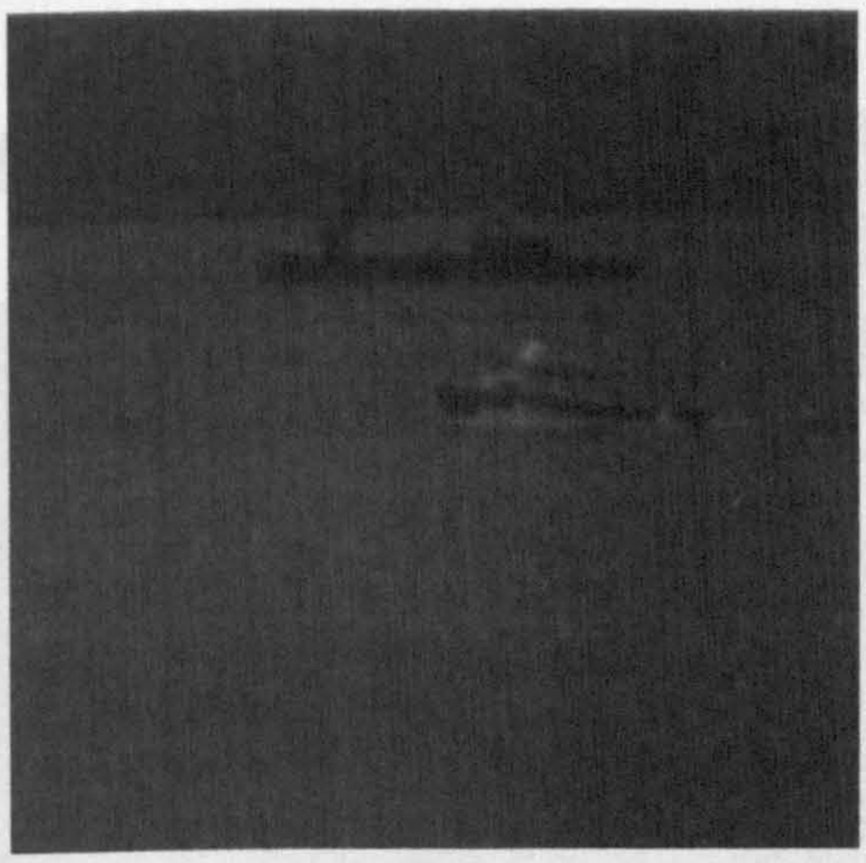

Figure 4.35a – Poole frame

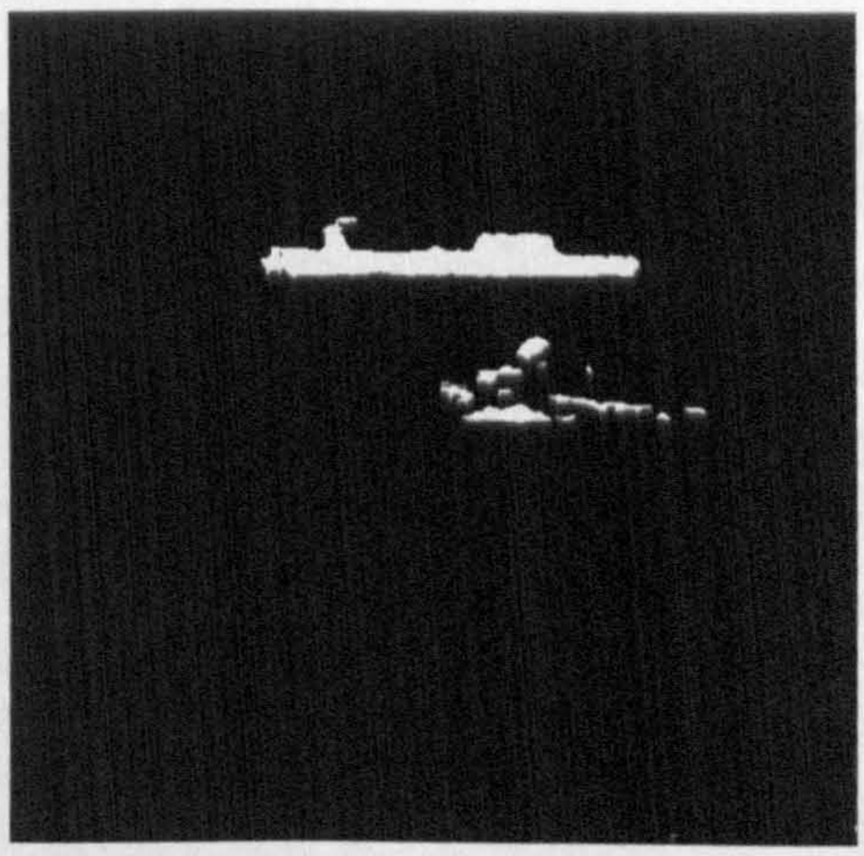

Figure 4.35b – Thresholding

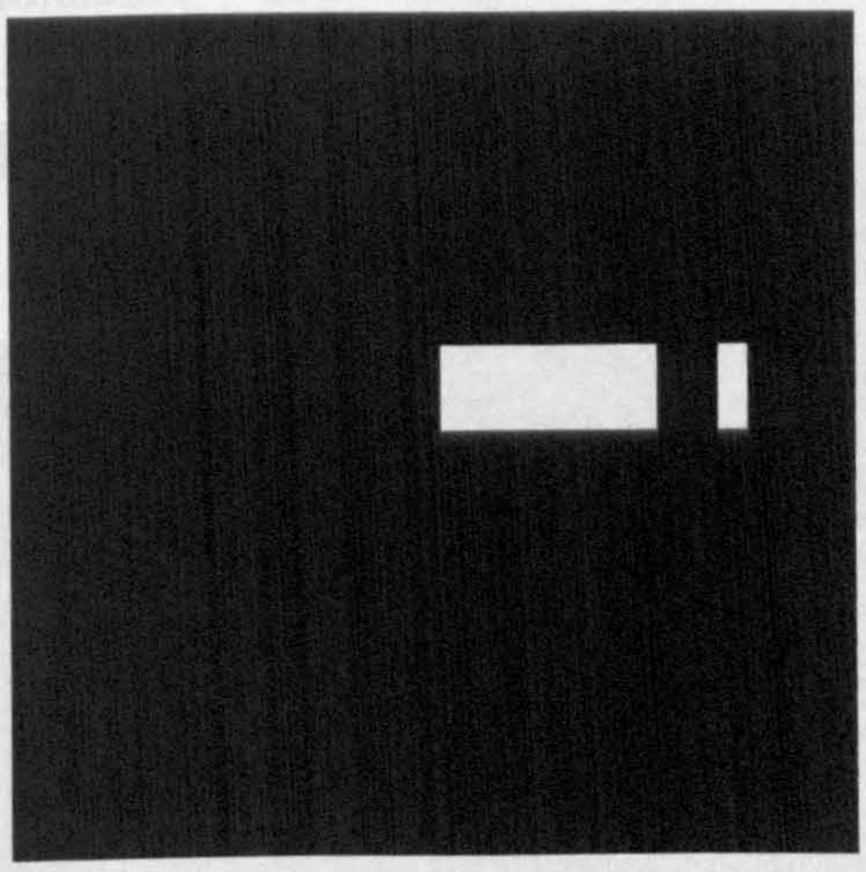

Figure 4.35c – Frame Differencing

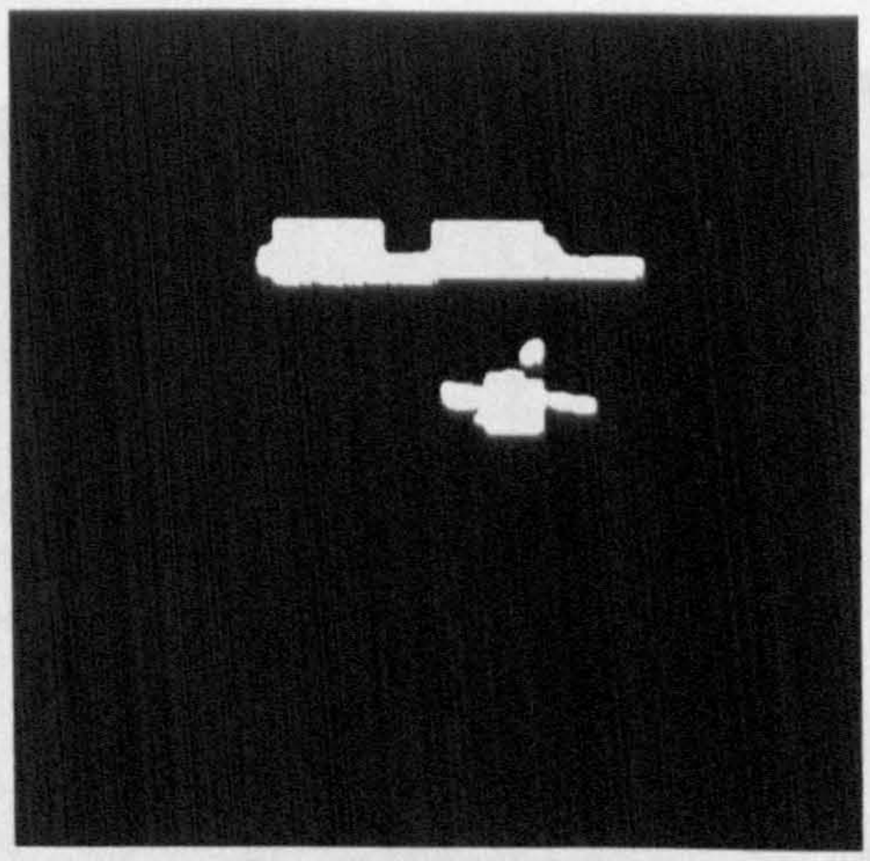

Figure 4.35d – Sea Characterisation

Figure 4.35 – Poole Sequence Segmentation Results

For the Poole sequence the thresholding method (figure 4.35b) has successfully segmented the objects and the sea although one of the objects has been heavily fragmented. This fragmentation has also occurred in the sea characterisation method (figure 4.35d) but to a much lesser extent. Due to the static camera platform the frame differencing method has as expected, not segmented the stationary object. The moving object has been segmented but the use of projection histograms has meant the segmentation will always be in the form of rectangles. A false object is also present.

Figure 4.36a – Dover frame

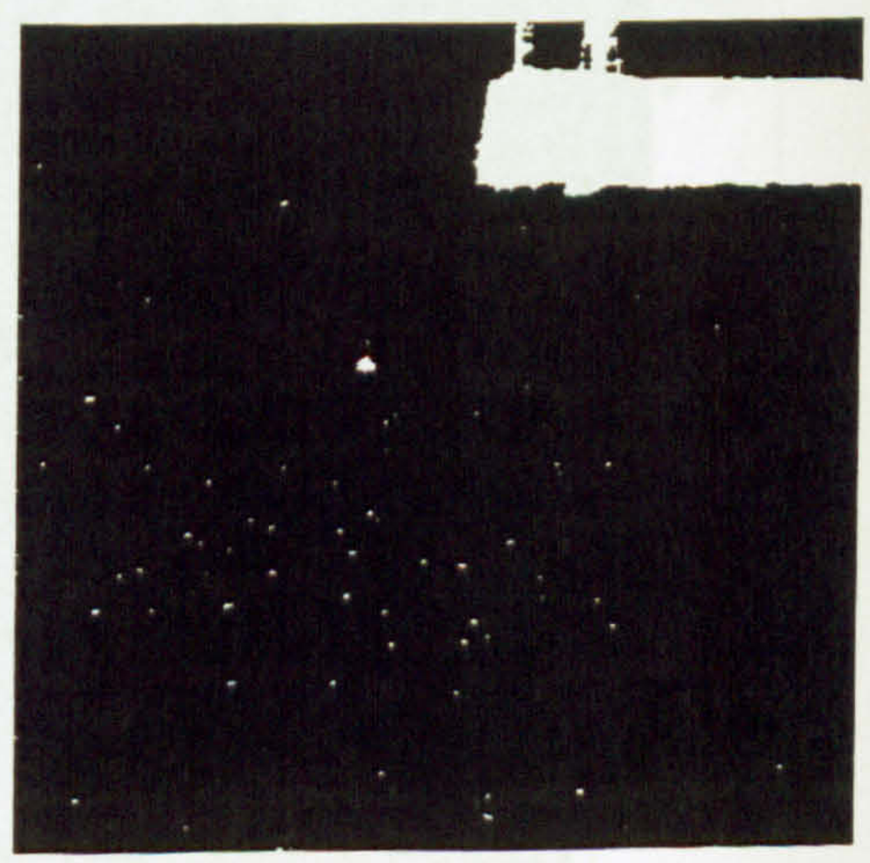

Figure 4.36b – Thresholding

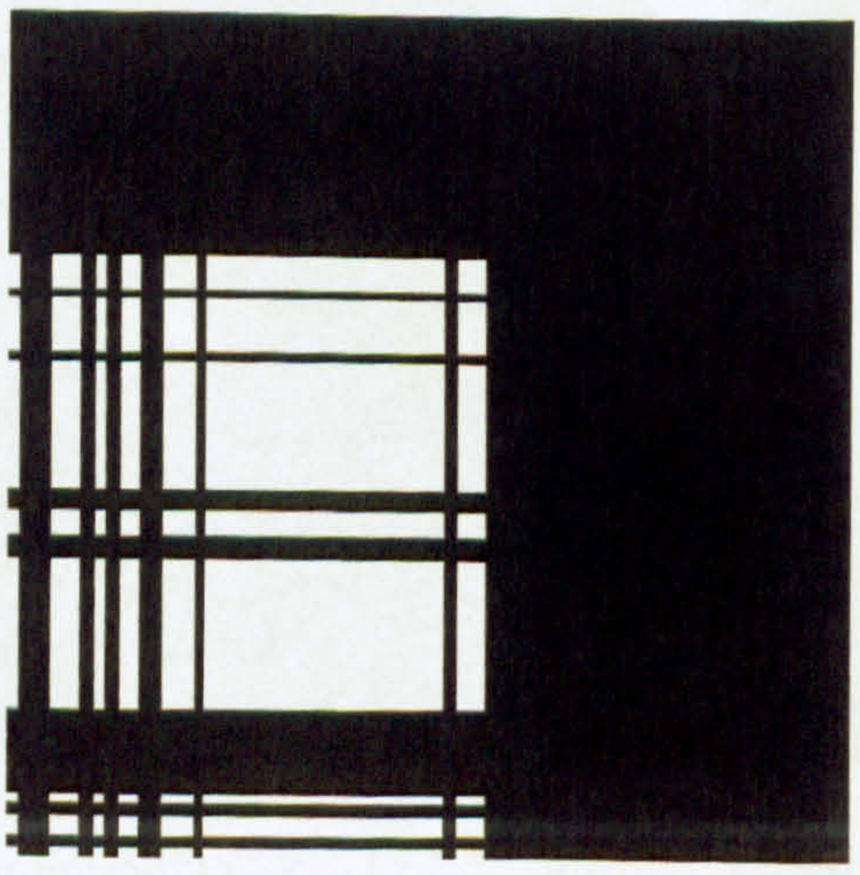

Figure 4.36c – Frame Differencing

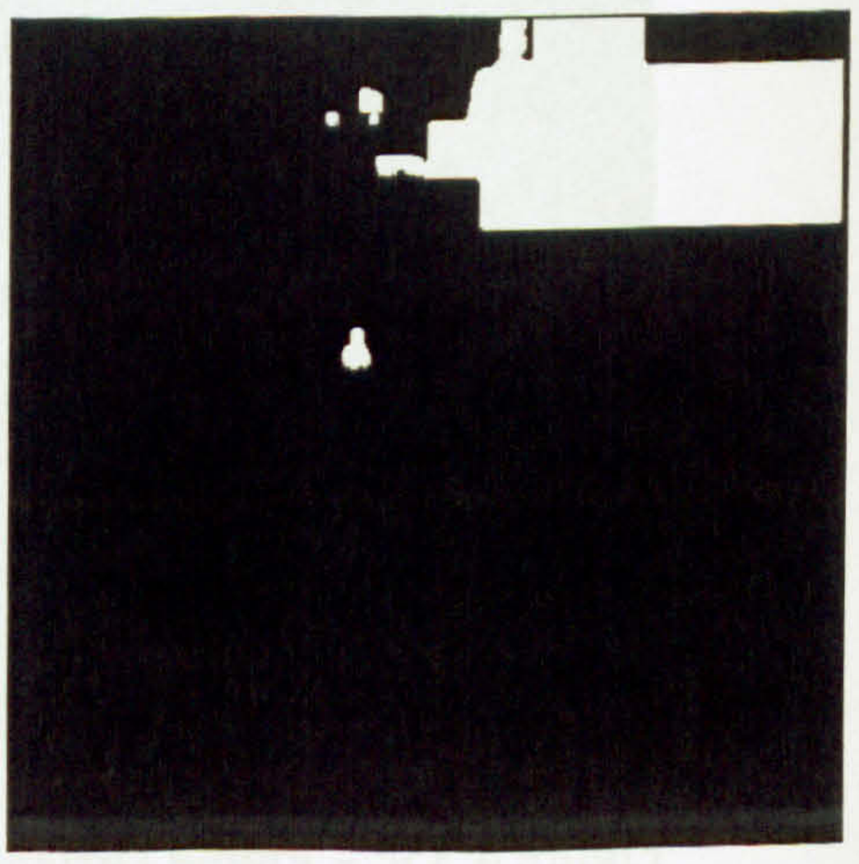

Figure 4.36d – Sea Characterisation

Figure 4.36 – Dover Sequence Segmentation Results

The thresholding method (figure 4.36b) has segmented the objects present in the Dover sequence but because of the wide range of grey levels in the sea many small areas of sea have also been segmented. The motion of the sea has caused the frame differencing method (figure 4.36c) to segment large areas of sea as objects. The stationary objects have not been segmented. The sea characterisation method has falsely segmented three small areas of sea. One of these is the dark shadow area to the left of the harbour wall and has been segmented as part of the wall object.

Figure 4.37a – Portsmouth frame

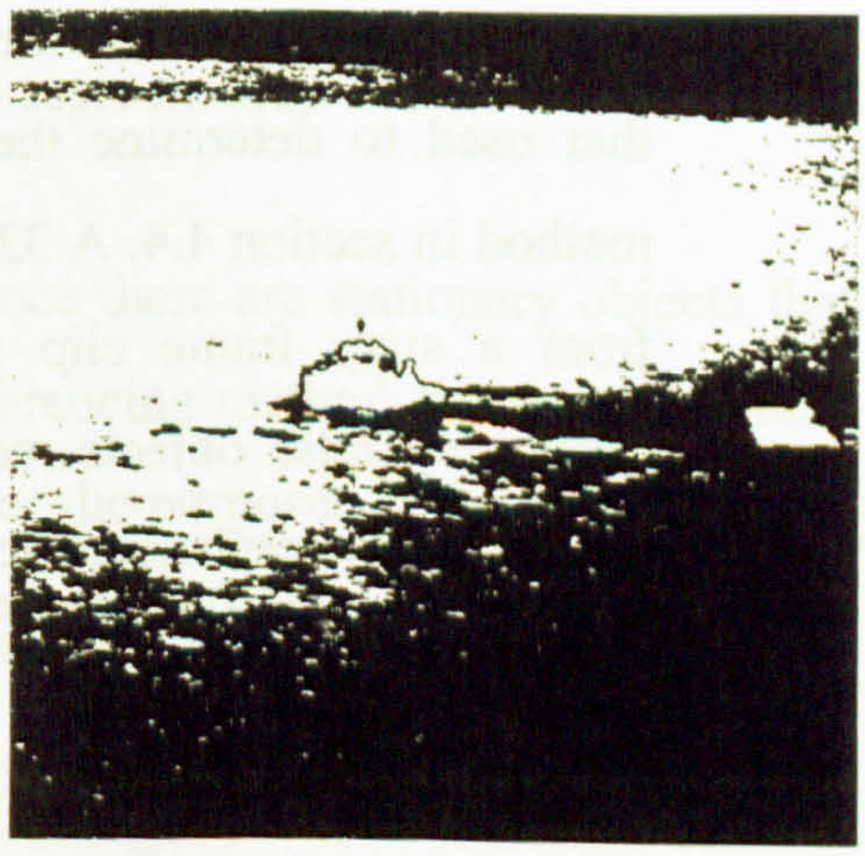

Figure 4.37b – Thresholding

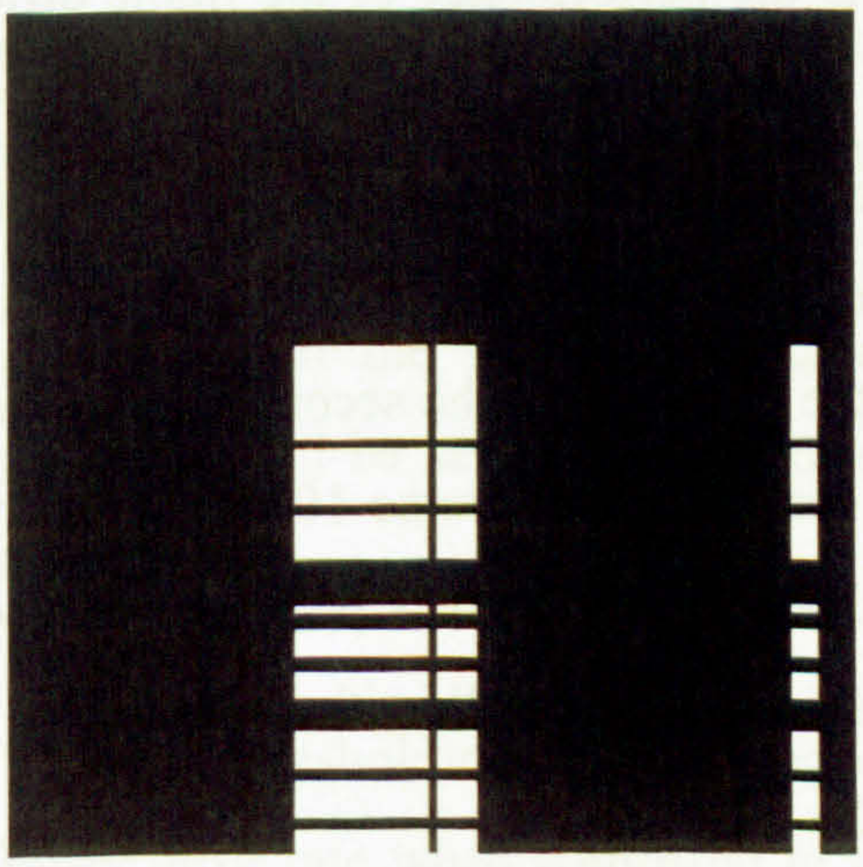

Figure 4.37c – Frame Differencing

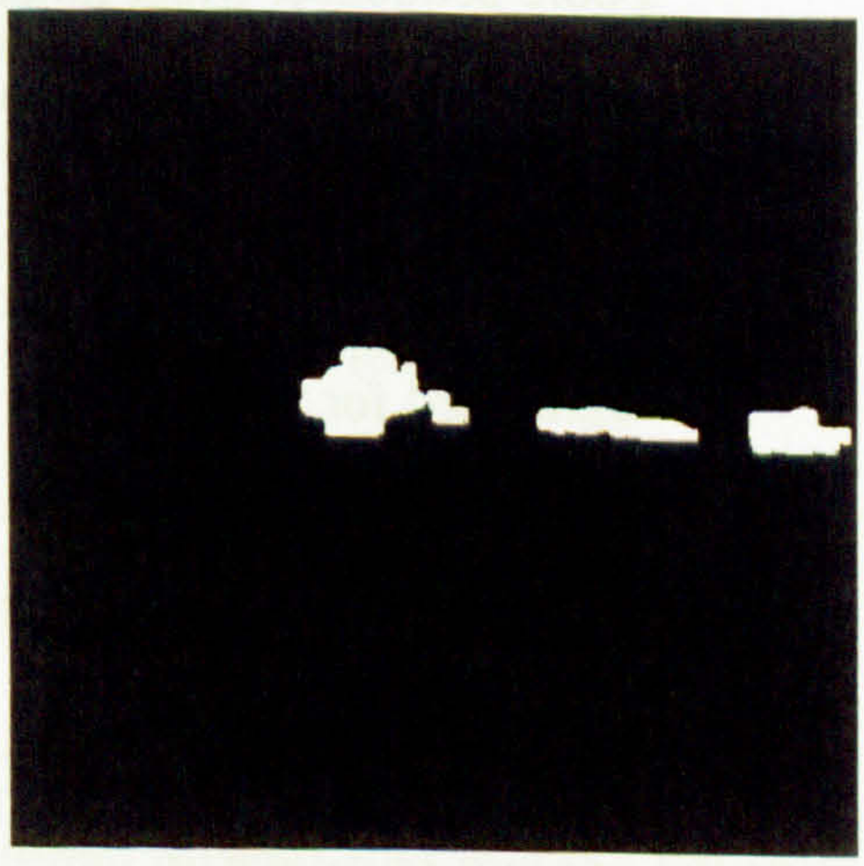

Figure 4.37d – Sea Characterisation

Figure 4.37 – Portsmouth Sequence Segmentation Results

In figure 4.37b the thresholding method has falsely segmented the lighter area of sea in addition to the objects from the darker area of sea in the Portsmouth sequence. The frame differencing method (figure 4.37c) has successfully segmented the larger object and part of the smaller object. However, many false objects have also been segmented. Figure 4.37d shows the result of the sea characterisation method. This has successfully segmented the two objects from the sea but has also falsely segmented the darker area of wake caused by the larger object.

The three segmentation methods have been evaluated in the same way as that used to determine the parameter values for the sea characterisation method in section 4.4. A 32x32 pixel grid was placed over every fifth frame from a sixty frame clip from each sequence and the number of tiles containing false objects recorded. The average number of tiles with false objects across each sequence, as a percentage of the total number of tiles, is shown in figures 4.38 – 4.40. Appendix C contains all the data used to generate these graphs.

Figure 4.38 shows the evaluation of the three segmentation methods on the Poole sequence. Most notable is the performance of the frame differencing method. The centrally located dark object is stationary and as there is no motion between frames the object is not segmented. As this is to be expected the tiles relating to this object have been considered to be correctly segmented for the purposes of the evaluation. The second, moving, object in this sequence enters the scene completely by frame 10 and exists by frame 30. It can be seen than while the moving object is present in the scene the frame differencing method performs quite well with a minimum error of 0.78% of tiles segmented falsely.

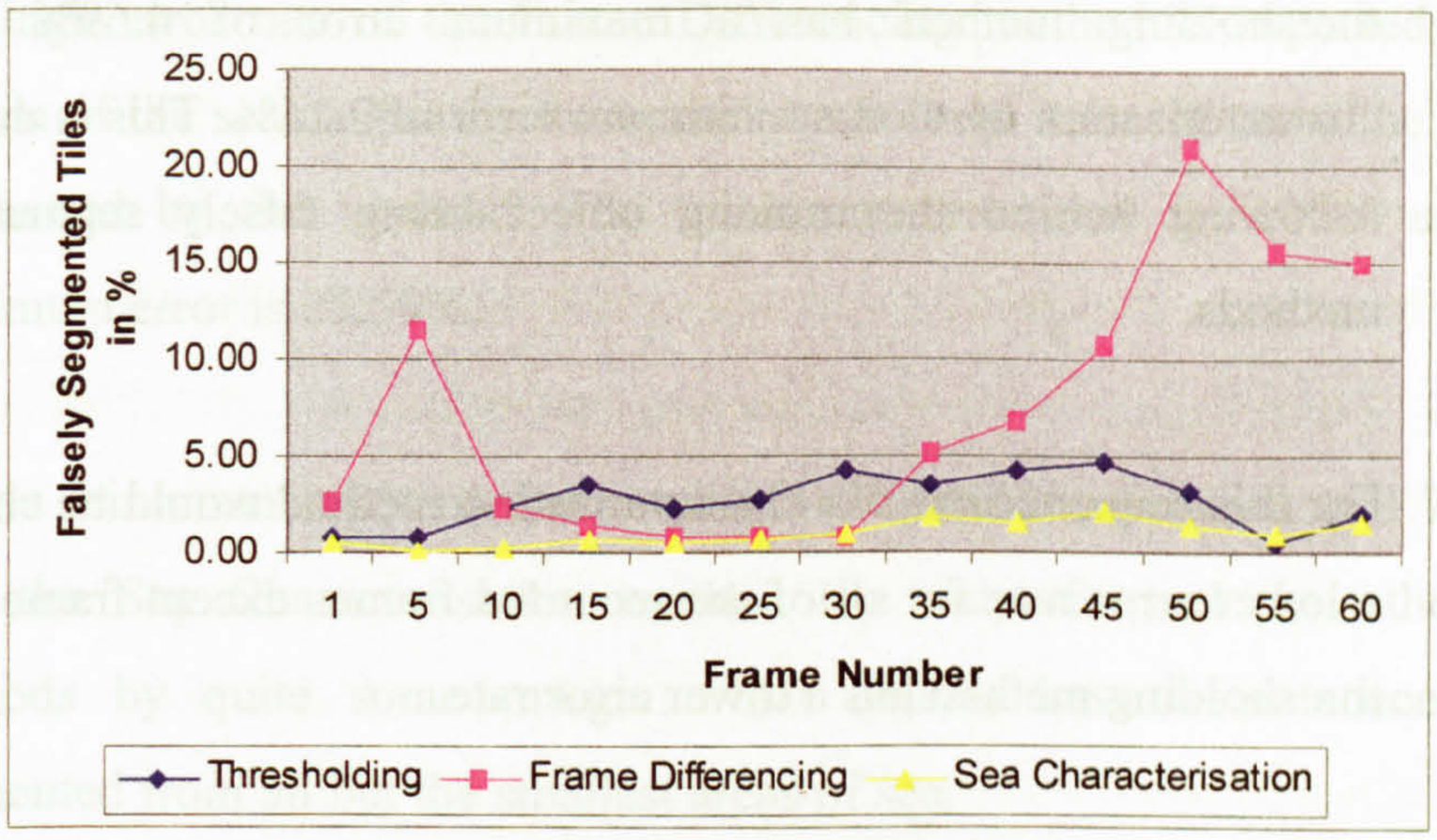

Figure 4.38 – Poole sequence segmentation evaluation

In both the Poole and Dover test sequence there are stationary objects that will not be detected with the frame differencing method owing to the static camera used to capture the sequences. For the purposes of the evaluation the tiles in which these objects should have appeared have not been marked as falsely segmented.

Both prior to and after the moving object's presence in the scene the error rate is considerably higher with a maximum error of 21.09% of tiles falsely segmented. This is due to the RMS method of threshold determination being used in this implementation of frame differencing. While an object with a high element of motion is present in the scene the threshold will be set above that of the sea motion but once that object leaves the scene the threshold will be set at a much lower level resulting in areas of sea being segmented.

Both the thresholding and sea characterisation methods have a consistently lower error rate when compared to the frame differencing method however when the moving object is present the thresholding method has a higher error rate. It can be seen that both the thresholding and sea characterisation methods have an increased error rate between frames 30 and 50. The

thresholding method has a maximum error of 4.69% and the sea characterisation method a maximum error of 2.15%. This is due to the wake following behind the moving object being falsely segmented by both methods.

For this sequence the sea characterisation method would be chosen as it has a lower error rate for all of the recorded frames except frame 55 where the thresholding method has a lower error rate.

Figure 4.39 shows the same evaluation for the Dover test sequence. This sequence contains a stationary harbour wall object and a floating buoy object. In this sequence the Frame Differencing method does not perform as well as the Thresholding method with the exception of 3 frames. The high error rate shown by the Thresholding method (maximum error of 67.58% and minimum error of 45.31%) is due to the wide range of grey levels present in the sea. The slightly choppy sea conditions have led to some of the very dark and very light grey levels of the sea to be segmented as object in addition to successfully segmenting the real objects.

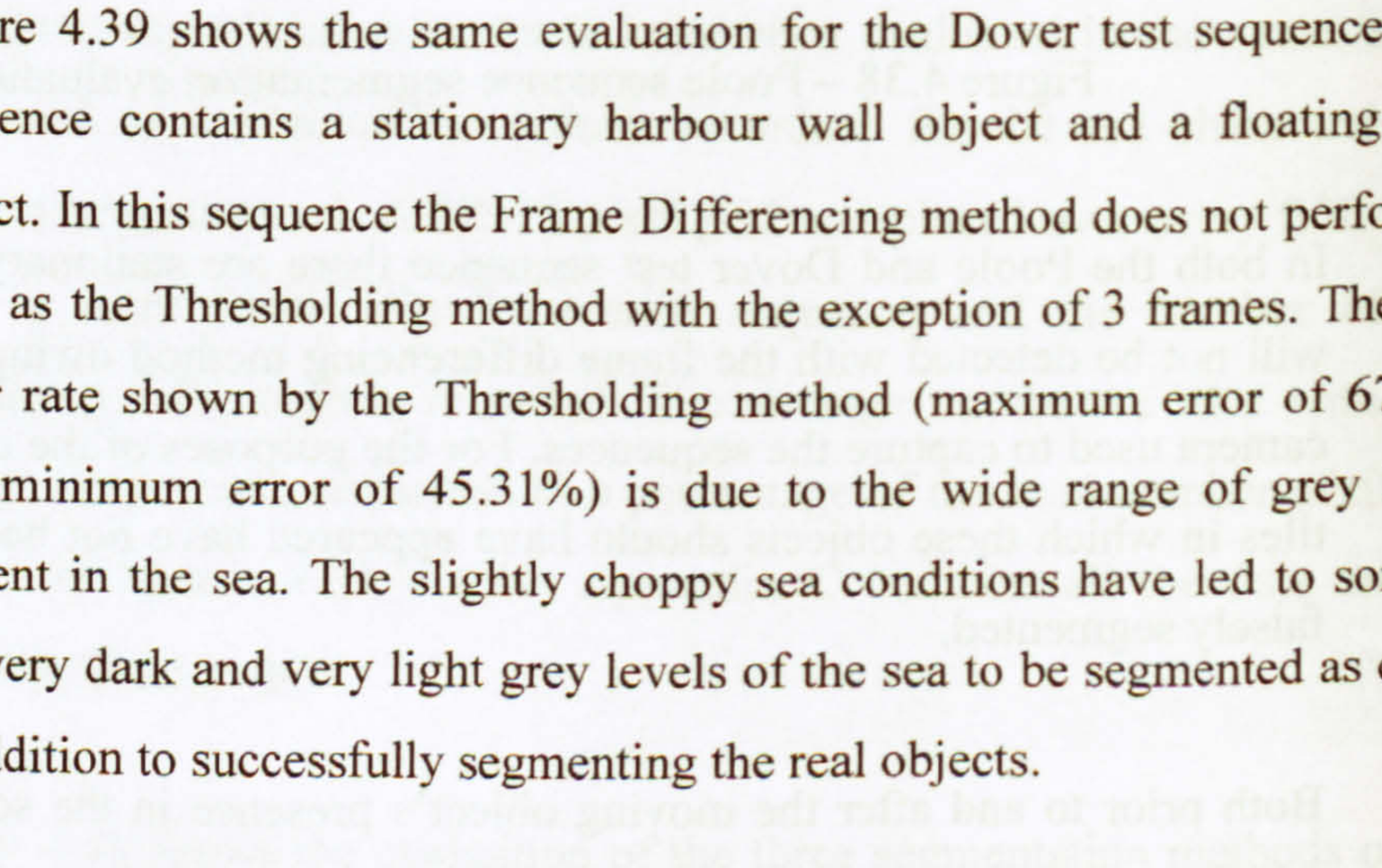

Figure 4.39 – Dover sequence segmentation evaluation

The high error rate of the Frame Differencing method is expected as the motion of the sea caused by the wind causes large areas of sea to be falsely segmented by this method. The minimum error rate is 9.77% and the maximum error is 25.59%.

With a minimum segmentation error of 0.2% and a maximum of 1.76% of tiles the Sea Characterisation method has outperformed the other two methods by quite some margin. All objects have been successfully segmented from all but the smallest areas of sea.

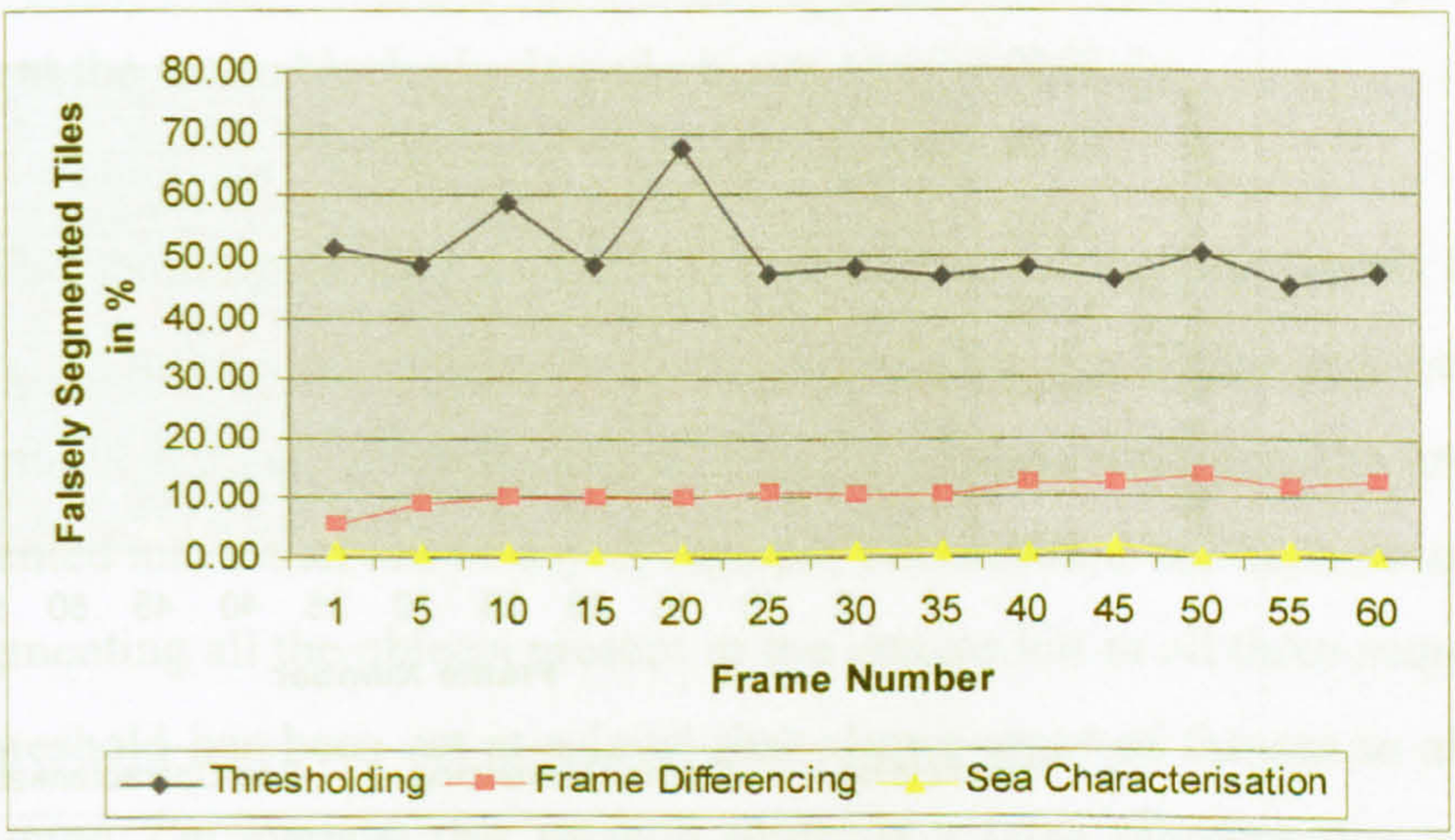

Figure 4.40 – Portsmouth sequence segmentation evaluation

Figure 4.40 shows the results for the Portsmouth sequence. The difference in performance between the three methods is clearer to see with the Sea Characterisation method showing the lowest level of false segmentation and the Thresholding method the highest level of false segmentation. The Thresholding method has suffered as the sea toward the top left of the image is lighter than in the rest of the scene and the threshold has been falsely segmented as large areas of object in many of the frames.

The low level of sea motion and the motion of objects in this sequence suggests that the Frame Differencing method should return a low error rate

as the RMS method of threshold selection would eliminate most of the sea motion. This has not been the case with this method showing a minimum error of 5.47% and a maximum of 13.48%.

The Sea Characterisation method has performed well with a minimum error rate of 0.2% and maximum error rate of 1.76%. The objects have been successfully segmented in all the tested frames and the Sea Characterisation method would be chosen for this sequence as it returned a lower error rate in every frame.

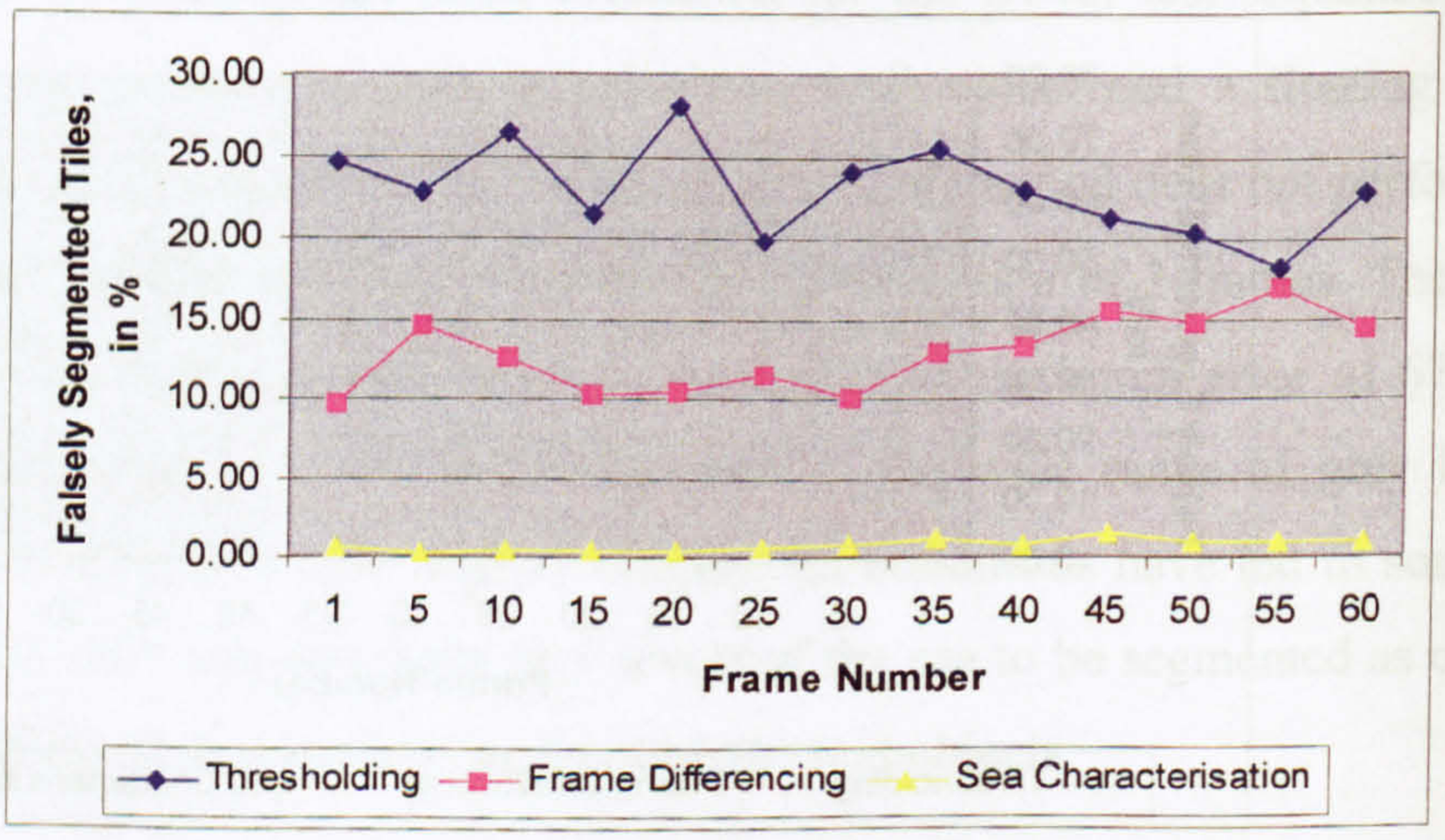

Figure 4.41 – Average Segmentation Error

In figure 4.41 is shown the percentage of tiles falsely segmented by each method when averaged across the three test sequences. This shows gives a more realistic result for the methods being evaluated here as it takes into account a variety of sea conditions and objects of differing sizes and grey level intensity. Despite the results being only 1.37% different for frame 55 it is clear that the Thresholding method performed the worst of the three with the Frame Differencing method returning on average 10% fewer falsely segmented tiles. However, the best performer was clearly the Sea Characterisation method. It successfully segmented objects in every frame and over 98% of the sea. This method returned a minimum average error rate of 0.2% and a maximum average of 1.5%.

4.6 Discussion and Summary

Three segmentation algorithms have been introduced in this chapter. First, a grey level thresholding method that searches the histogram of grey levels for valleys between distinct peaks representing sea and objects. Secondly the method of segmentation by frame differencing that Voles et al (1999) used on maritime images. Finally a method of segmentation developed for use with these images by statistically characterising the grey level of the sea and using this to segment the image into the same two regions as the other methods, namely sea and object. Implementations of all three algorithms segment the sea to black pixels and objects to white pixels.

The Thresholding method constructs a histogram of the grey levels in the image, smoothes the histogram then searches for the peaks and troughs. Thresholds are set in each trough and the regions between the troughs segmented into either sea or object regions. This method has been successful in segmenting all the objects present in the images but in all three sequences the threshold has been set at a level that allows areas of the sea to also be segmented. On average this method returned a false segmentation rate of between 18.23% and 27.99%.

The method of Frame Differencing implemented begins by subtracting the current frame from the previous frame. The resulting image is then projected onto its horizontal and vertical axes and a projection histogram for each obtained. These histograms are then thresholded using the RMS value of the histogram entries to eliminate the low levels of motion caused by the sea. Values above the threshold are then set to object and those below to sea. The co-ordinates of the objects are then determined by scanning the histograms. The result is a binary image with white rectangles representing areas of motion (object) and the background (sea) is black. It is noticed in the results for the Poole sequence that while a fast moving object is present in the scene this method works well with a minimum error rate of 0.78%. However,

because of the RMS method of threshold selection, when no object is present the threshold is set low enough for areas of sea motion to be segmented. This raises the error rate to a maximum of 25.59% in the Dover sequence. This method also suffers from the method of projecting the objects identified in the histograms back into co-ordinate space. If only a single object is found in each of the horizontal and vertical directions then a single object results. However, if multiple objects are found in each direction then an object results at each location where the co-ordinates cross. This is not necessarily correct and has led to a number of false objects to be segmented, particularly in the Dover sequence. Further processing steps would be required to eliminate these false objects.

The Sea Characterisation process uses a preset 'shoreline' value to mark the upper limit of the sea in the image. This is used to ensure that only valid sea pixels and those of any maritime object present in the scene are considered in the statistical characterisation of the sea. Statistical analysis based on the histograms of the pixels in the image then follows. This uses predetermined values for the tile size and decision level parameters to arrive at a characteristic range of grey-levels for the sea in the image. The tile size and decision level parameters were derived through extensive experimentation using differing values for the parameters. The characterisation process showed an error of 3% for the Poole sequence and 14% for the Dover sequence

The segmentation process uses the characteristic grey-level range determined for the image to segment the image into two regions, sea represented by black pixels and object, shown by white pixels. The tile size determined in the characterisation process is used again here. A grid of these tiles is placed over the image and a decision about the segmentation of the entire tile is attempted. This approach was taken as it is considered that the majority of the image will be sea therefore segmenting a whole tile at a time

would reduce the number of false objects segmented due to highlights and shadows present on the sea. If it is not possible to segment a whole tile then the tile is segmented on a pixel by pixel basis to be either sea or object. The success of the division of the image into quarters and the use of 32x32 tiles to segment the image shows that the decision to ignore scene perspective has not compromised the segmentation process.

A 9-way connectivity measure is used as a method of growing the object regions that have been segmented. This has been done as an object can be fragmented during the segmentation process as several smaller distinct objects. The region growing algorithm grows and joins these areas together. Nine directions were used in the connectivity measure using a predetermined search space size and threshold value.

The Sea Characterisation method has been shown to be the method of choice for each of the three test sequences with the average error across all the sequences ranging from 0.2% to 1.5%. The Sea Characterisation method has, therefore, been chosen as the method of segmentation for the system developed in this thesis.

Chapter 5: Object Tracking and Motion Prediction

5.1 Introduction

In image motion analysis, tracking is the process of identifying the same object in each frame of an image sequence. It plays a fundamental role in many applications including encoding of visual data, man-machine communication, and surveillance (Mitiche et al, 2002).

Tracking man-made objects moving in open world scenes is a complex task (Teal, 1997) and has received considerable attention in the literature. The tracking process has the primary goal of solving the correspondence problem (Jain, 1984). The correspondence problem can be solved using a number of techniques such as feature based geometric model matching (Tan et al, 1994). Worrall et al (1993) shows model based techniques to be successful where the objects to be tracked occupy a significant proportion of the image.

If the object to be identified and tracked is a long distance from the camera and only occupies a small proportion of the image pixels Rosin and Ellis (1991) and Teal and Ellis (1995) found that the matching of crude object descriptors is more robust. A similar approach is that of Wang et al (2000) who use a rule-based algorithm using the variables calculated from information about the object to perform the tracking of objects.

Maritime objects can take many forms, from a racing dinghy to a ferry and from a harbour wall to a buoy. Each of these objects could have a different velocity and manoeuvrability but each presents a potential collision threat for another maritime craft and needs to be tracked through the image sequence. In this chapter the correspondence problem is addressed to track maritime objects over many frames and the future motion of the objects tracked is then predicted. Two methods have been evaluated for use with maritime images. The first method uses a tracking method based on object parameter measurements similar to those used by Wang et al (2000) and Ellis et al (1991), a set of motion constraints (Teal and Smith, 1999) and a motion model developed for use with maritime images. Motion prediction is then carried out using a method of least squares (Stroud, 2001). The Kalman filter (Kalman, 1960) is the second method and is arguably the most popular technique for tracking objects for its ability to predict the future position of the object.

The two algorithms described here are evaluated against the manually derived centroid position of each object. Results for each algorithm are shown and a conclusion reached for which is the most appropriate method for use in the system being developed. The input to the tracking and prediction algorithms is a segmented image sequence where the maritime images have been segmented into sea and objects. Each segmented object has had its minimum and maximum x and y co-ordinates and centroid location calculated.

5.2 Object Parameter and Motion Model Tracking

Each segmented object input to the tracker has a statistical feature vector built. This is used together with a set of motion constraints (Teal and Smith, 1999) in the form of a motion model developed for maritime objects, similar in nature to the features used by Franchi et al (1996), to decide if an object identified in two adjacent frames is the same. Falsely segmented blobs are minimised through the use of the motion model. This object matching continues over N frames with key motion parameters being recorded about the object in each frame. The output from the tracker is a set of matched object descriptors. In the output image the tracked objects are highlighted by visual cues. A black box is drawn around the area of each blob that has been matched to show it is being tracked. Once a blob has been confirmed as a maritime object a white box is drawn. A flow chart showing the tracking and prediction process is shown in figure 5.1.

A number of methods have been employed in the prediction of object motion. Kalman filtering (Kalman, 1960) described in section 5.3, and curve-fitting such as the Linear Regression (Achelis, 2004) method. These methods can yield accurate results for the prediction of object motion. Wang et al (2000) show that their assumption that object trajectories are close to straight lines in a few adjacent frames was successful. This assumption was also shown to be true of maritime objects and is the basis for the choice of prediction algorithm used here. As object trajectories are assumed to approximate a straight line a less intensive Least Squares algorithm (Stroud, 2001) has been used for calculating the future position of an object in the scene. In frame N+1, the future motion of the object is predicted. A line of best fit is calculated and drawn from the recorded object data. Straight-line segments are then drawn in the image showing the estimated future motion of the identified objects.

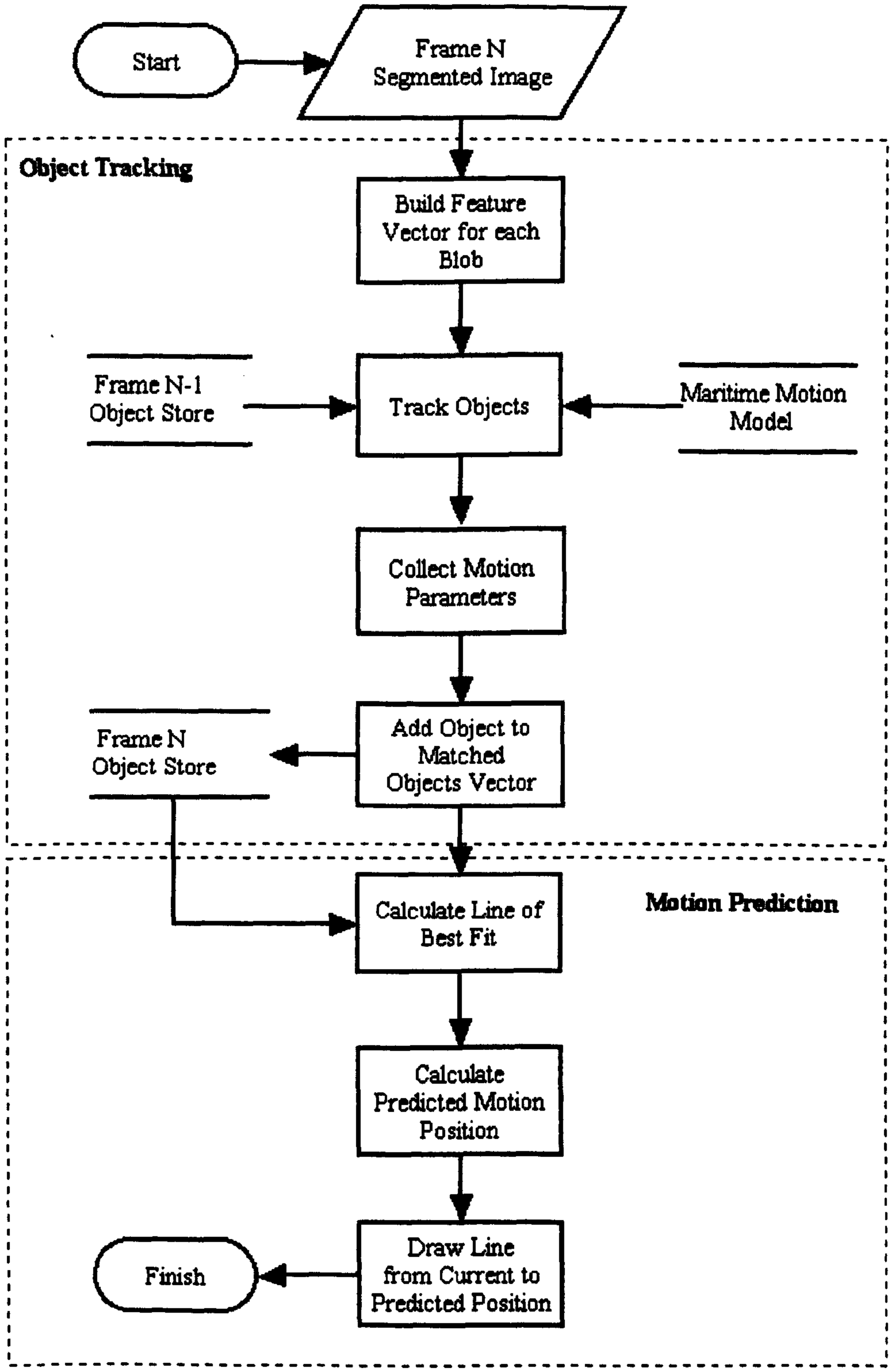

Figure 5.1 – Object Tracking & Motion Prediction Flow Chart

5.2.1 Maritime Object Motion Model

A model has been developed of the typical motion of maritime objects. This model will be used to solve the correspondence problem by determining whether an object segmented in successive frames is the same object. The model includes motion constraints such as the maximum speed of an object and the rate of change in orientation of a valid maritime object.

5.2.2 Motion

Through the observation of the maritime environment the total motion of a maritime object can be considered to contain three elements:

- Principal Motion
- Incidental Motion
- Platform Motion

Principal motion is defined as that motion introduced by the objects own power source to drive the object in an intended direction. For a boat the power source would be the engine and for a yacht it would be the wind. As this source of motion is applied intentionally it is assumed to be predictable allowing certain parameters such as orientation (direction of travel) to be taken into account when tracking the object.

Other maritime objects, such as a mooring buoy, have no such power source but observations show that there will still be some variable amount of motion. This is defined as Incidental motion. This motion can be due to the movement of the sea and can be caused by a number of variables, namely:

- Wind
- Tidal flow
- Wake from passing craft

The levels of these variables and the direction they come from will all vary the amount of motion applied to the maritime object. As the level of these forces is unknown this motion is assumed to be unpredictable, removing the use of the orientation parameter in the tracking process.

Platform motion is that motion introduced by the movement of the vessel the camera is mounted on. As no information is known about the movement of the sensor platform the motion from this source is also considered to be unpredictable. Objects present in the image are considered to have the combined motion of the object and the sensor platform.

5.2.2 Motion Model Parameters

The motion model contains the following parameters: Object Characterisation; Object Speed; and Object Orientation. The values of these parameters dictate how the tracking algorithm will approach the matching of each object in the image.

Object Characterisation

An object generating sufficient power for principal motion to be the higher proportion of total motion allows the possibility of using the predictability of the principal motion to aid the tracking of the object. If the reverse were true then the unpredictability of the incidental motion component would require a different approach to tracking the object. For this reason objects are categorised. The motion of the sensor platform will also affect the observed velocity of any object present in the image.

A maritime object such as a harbour wall is a solid object fixed to the seabed. It is not floating so we assume it is stationary. However, as the vessel on which the sensor is mounted could well be moving the harbour wall object could appear to have motion. If the sensor platform had a low speed and therefore had incidental motion as its major source of motion then the harbour wall object would also appear to have incidental motion as its major source of motion. If the sensor platform were travelling at a high speed then the harbour wall object would also appear to have a high speed and principal motion as its major source of motion.

An object that is floating in the water, either tethered in some way such as a buoy, a channel marker, or a boat on a mooring will 'bob' around with the movement of the water. These objects have no element of principal motion, any motion being the result of incidental motion. As with stationary objects any movement of the sensor platform will effect the motion of the object in the image. Objects moving with principal motion will have the effect of this motion changed by that of the sensor platform. If the platform were moving in the same direction as the object at a similar speed the object would appear to have a low speed.

Objects present in the image are categorised as either Slow Moving or Fast Moving. If the sensor platform were not moving the major source of motion alone could be used to categorise an object, slow moving for those with unpredictable incidental motion as the major source of motion and fast moving for objects with predictable principal motion as the major source. However, as the sensor platform could well be moving it is the combined total motion that is used to categorise the objects. To take account of the unpredictable incidental motion, a threshold has been determined below which all objects are categorised as Slow Moving and above which all objects are categorised as Fast Moving.

Speed

The definition of a slow moving object is that its motion will be made up of primarily incidental motion with principal motion being secondary. This is due to that motion caused by the sea being equal to or in excess of that provided by the object's own power source.

If we consider that a speed of 2 knots is equivalent to 3.7 kilometres per hour, or a steady walking pace, it is not unreasonable to assume this to be a slow moving maritime object and a speed at which incidental motion is the greater component of total motion.

Using the basic trigonometry function for a right angle triangle in equation 5.1, equations 5.2 and 5.3 and the image capture parameters it is possible to estimate the number of pixels an object moving at 2 knots would cover between frames.

$$O = \tan n * A \tag{5.1}$$

$$t = P / s \tag{5.2}$$

$$p = (w / t) / f \tag{5.3}$$

Where:

A = minimum distance from sensor to object in metres

n = field of view / 2

O = distance from edge to centre of the image

P = O * 2

s = speed in metres per second

t = time in seconds for object to cross the image

f = number of frames per second

w = width of image in pixels

p = number of pixels moved per frame

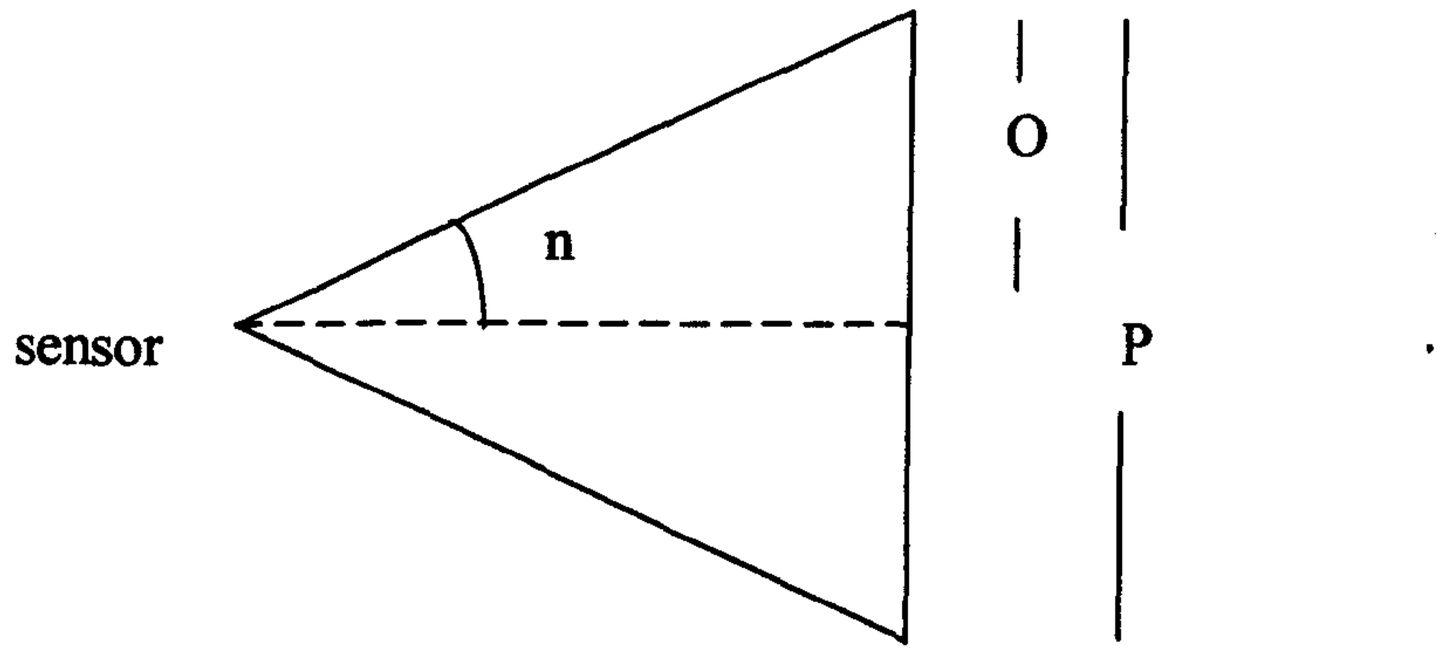

Figure 5.2 - Calculating the length of the opposite side of a triangle

We know from the image capture parameters that:

- the angle n will be half of the field of view of 23°
- the width of the captured image is 768 pixels
- the capture rate f is 10 frames per second

We consider the minimum distance, A, between the sensor and the object to be 50 metres as the length of a typical ferry is 82.4 metres (Whyte, 1998) and it is assumed that the camera will be mounted on the bridge, approximately half way along the vessel's length.

At a speed of 2 knots and a distance of 50 metres, the centroid of an object will move approximately 4 pixels in a horizontal direction between frames. As this movement can be in any direction this gives an area of 8x8 pixels within which a slow moving object may be found.

Fast moving objects are defined as those objects whose motion is comprised primarily of principal motion with incidental motion secondary. This is due to the object's own power being in excess of that provided by external

forces. The motion model has to allow for the movement of these fast moving objects up to the maximum speed of a typical high-speed vessel.

A typical fast maritime object, the Red Jet catamaran passenger ferry, has a maximum speed of 34 knots (Whyte, 1999). Given the formulas (5.1-5.3) and figure 5.2 the maximum number of pixels covered by an object travelling at 34 knots and a distance of 50 metres can be estimated as 33 pixels. Any object in frame N+1 that has a centroid greater than 33 pixels away from an object in frame N would not be considered the same, as the object could not have moved that far between frames. Maritime objects that are moving tend to do so at a consistent rate, it can be considered that they do not accelerate or decelerate between frames so it is assumed that object motion will be uniform across the image (Wang et al, 2000).

Orientation

Figure 5.3 shows the possible directions an object can have. Slow moving objects can have a high degree of unpredictable incidental motion. As the orientation change is unpredictable the motion model disregards the orientation of slow moving objects by setting the orientation parameter to zero. In the first frame in which a new object is identified the orientation of the new object will be unknown. New objects are therefore also given the orientation zero.

Franchi et al (1996) state that a very reasonable assumption is that the motion of an object does not change abruptly between consecutive frames. An object travelling at higher velocities will therefore not be capable of a dramatic change in orientation between frames. It is assumed that a fast moving object heading East in frame N cannot be moving West in frame N+1. However, because a fast moving object may change orientation between frames the motion model allows a fast moving object in frame N+1

to have moved up to one direction each side of its orientation in frame N. For example, with reference to figure 5.3, if an object in frame N had an orientation of 5 then in frame N+1 the object could have orientation 4, 5, or 6. Direction zero is the 8x8 pixel area within which slow moving objects may be found.

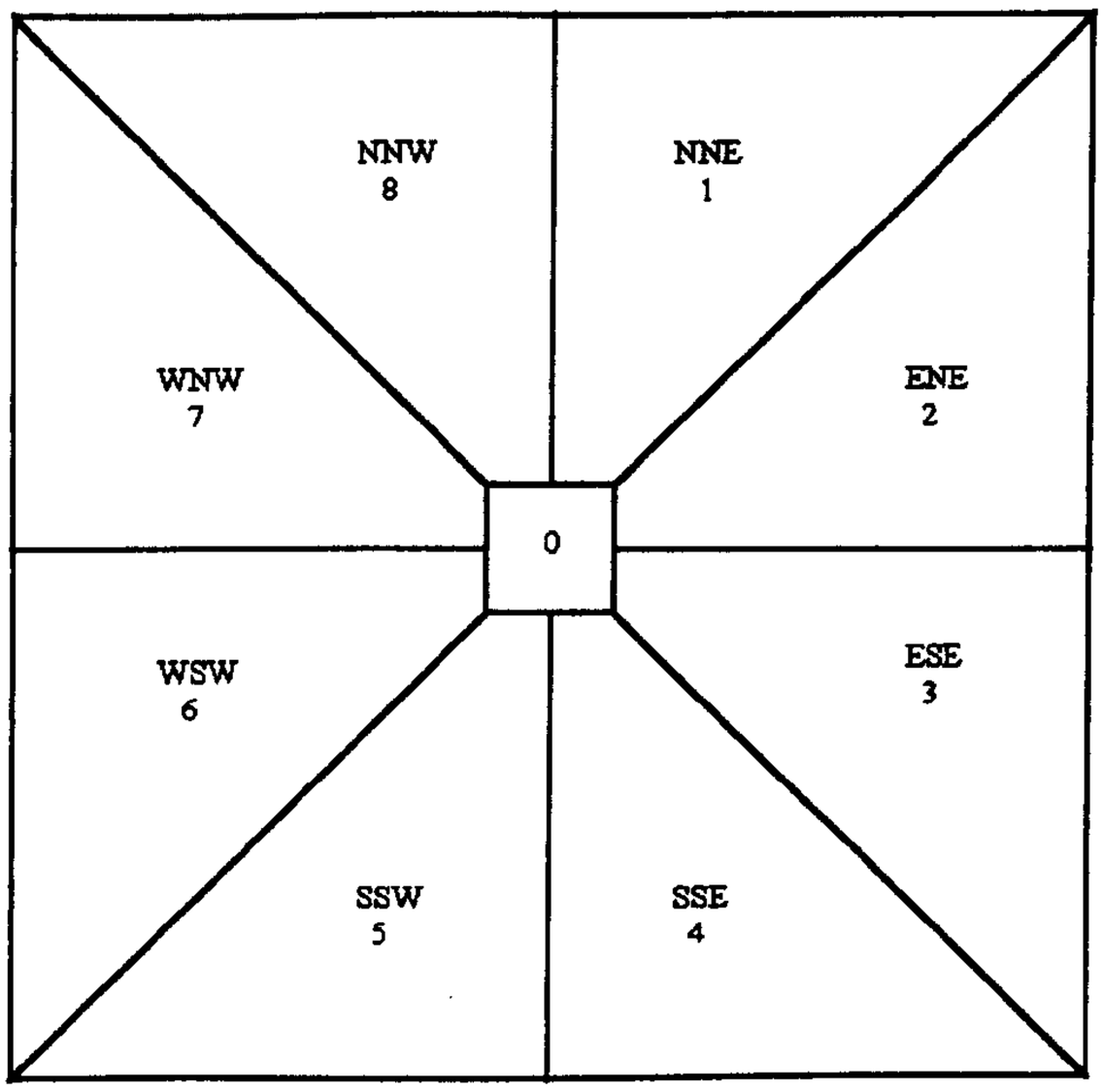

Figure 5.3 – Directions of Travel. The number associated with each direction is the value stored in the object feature vector. There is also a direction 0 for new objects and stationery or slow moving objects.

5.2.3 The Motion Model

From the points made in sections 5.2.1 and 5.2.2 the motion model developed contains the following motion constraints:

1. Objects may only move between their current orientation and either of the adjacent orientations shown in figure 5.3. The orientation may also be zero. An object moving in one direction in frame N cannot move in the opposite direction in frame N+1.

2. Objects moving at 2 knots or less and stationary objects are considered to have an unpredictable direction of travel therefore a direction zero has been established.
3. Objects can have a maximum estimated acceleration between frames of 33 pixels. Based on a typical high-speed ferry having a maximum speed of 34 knots (Whyte, 1999).
4. An object's motion is assumed to be uniform across the image as it is assumed that an object does not accelerate or decelerate between frames.
5. All maritime objects are assumed to be rigid.
6. All new objects will have an initial direction of zero.

5.2.4 Tracking Process

The tracking process is shown in flow diagram form in figure 5.4. Before the object matching begins a two-pass connected component analysis algorithm (Sonka et al, 1993) is applied to the segmented image to define the boundary for each region of interest (object). Each object is then labelled with a unique integer identifier (object number) and has a feature vector similar to that used by Teal and Smith (1999) built as shown in table 5.1. The min and max X,Y co-ordinates are determined by scanning the image in a raster scan order and noting the co-ordinates for each labelled object.

The direction is set to zero for each new object. Once an object has been matched between frames the direction is calculated from the centroid co-ordinates of the object in frame N+1 and the centroid co-ordinates of the object in frame N.

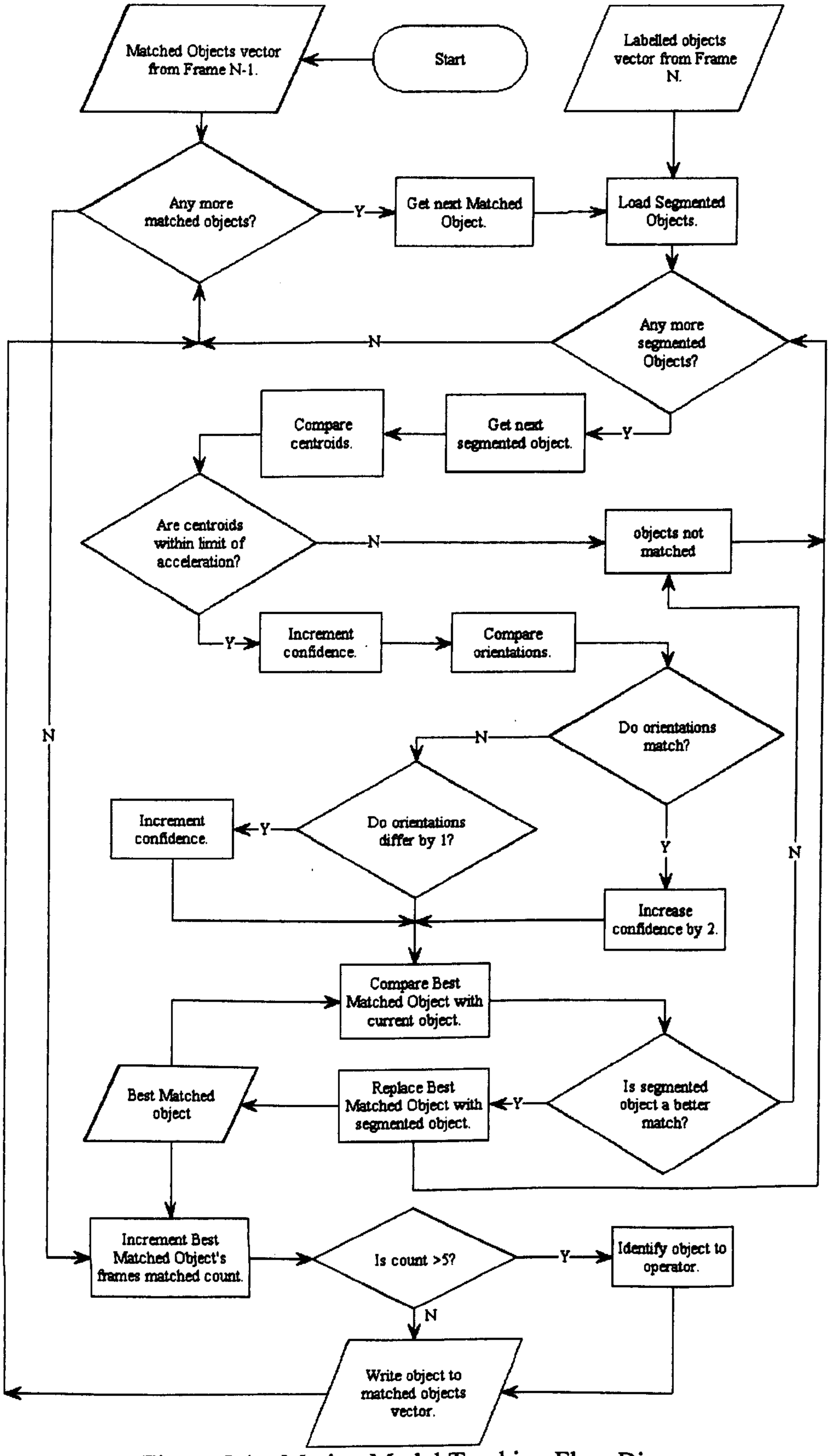

Figure 5.4 – Motion Model Tracking Flow Diagram

Table 5.1 - Feature vector built for each object in the image

Object Number
Min X Coordinate
Max X Coordinate
Min Y Coordinate
Max Y Coordinate
X Centroid
Y Centroid
Area
Direction
Number of Frames Matched
Centroid Data Array

As the feature vector is being built from the segmented image the zeroth order moment will give the area of the object (the total number of pixels in the object). Equation 5.4 defines the zeroth order moment (Glasbey and Horgan, 1995).

$$M_{00} = \sum_{(i,j) \in A} \sum i^0 j^0 \qquad (5.4)$$

where i , j are object pixel co-ordinates

A is the object of interest

The Number of Frames Matched is incremented every frame if the object is matched and is used by the motion prediction algorithm.

The Centroid Data Array is a 2-dimensional array containing the centroid co-ordinates of the object for a number of frames and is used by the motion prediction algorithm. The array is initialised to contain all zeros.

Segmentation algorithms that have no information regarding the objects of interest, such as that used here, can lead to objects being fragmented. This is a situation in which an object is segmented into many smaller parts. To reduce the effect of object fragmentation any objects whose boundaries overlap are combined into a single object. Although this can lead to objects that may be occluding each other being combined as a single object this is acceptable, as the collision risk for the combined object will be that of the object posing the greater risk. Once the occlusion has ceased the objects will be segmented individually and their collision risk estimated separately.

The problem of object correspondence between frames is minimised by applying the motion constraints from the motion model and maximising a confidence measure to find the best match for the object under consideration, as shown in figure 5.4.

Each object in frame N is compared with each object in frame N-1 using the constraints in the maritime motion model and a measure of the confidence of a match. The first constraint to be used is the acceleration. If the centroid positions of the objects being compared are within the limit of acceleration contained in the motion model then these could be the same object. The measure of confidence in the match is incremented. If this constraint is not met then the object in frame N cannot be a match and the object is rejected as a possible match for the object in frame N-1.

If the acceleration constraint is met then the orientation constraint is considered. The motion model states that a fast moving object travelling East in frame N-1 cannot be moving West in frame N. If the orientations of the two objects match then the confidence measure is increased by 2. If the orientations vary by a single direction then the confidence measure is incremented and if the orientations do not match then the confidence measure is left unaltered.

At this point the object from frame N will have a confidence level of 1, 2, or 3. The higher the number, the greater the confidence that this is the same object as that in frame N-1. If this is the first object from frame N to be considered for a match then this object becomes the Best Matched Object. If this is a further object being considered then the confidence level of the Best Matched Object and object currently under consideration are compared. The object with the higher confidence level becomes the Best Matched Object and the other is rejected. Once all objects in frame N have been compared against the object in frame N-1 the Best Matched Object has its feature vector updated with the object number from the object in frame N-1 and has its Number of Frames Matched value incremented. The object is then copied into a Matched Objects vector. In this frame, as in each frame that an object is matched, the X and Y centroid values for the object are stored in the Centroid Data Array within the feature vector for that object.

As it is possible for falsely segmented objects to be matched between frames each object that has been matched across two frames has a black box drawn around it and is termed a possible object. The black box shows that the object has been matched but that is has not been confirmed as a maritime object. The number of frames an object needs to be matched across before it is confirmed as a maritime object has been determined from the Poole and Portsmouth test sequences. Figures 5.5 and 5.6 show the number of frames false objects are matched across for 60 frames of the Poole and Portsmouth sequences. In both sequences false objects have been matched over several frames but in most cases these objects ceased to be matched after two, three, or four frames. Given the results from these sequences it has been decided that objects must be matched a minimum of five times before it is confirmed as a maritime object. Confirmed objects are displayed with a white box around them.

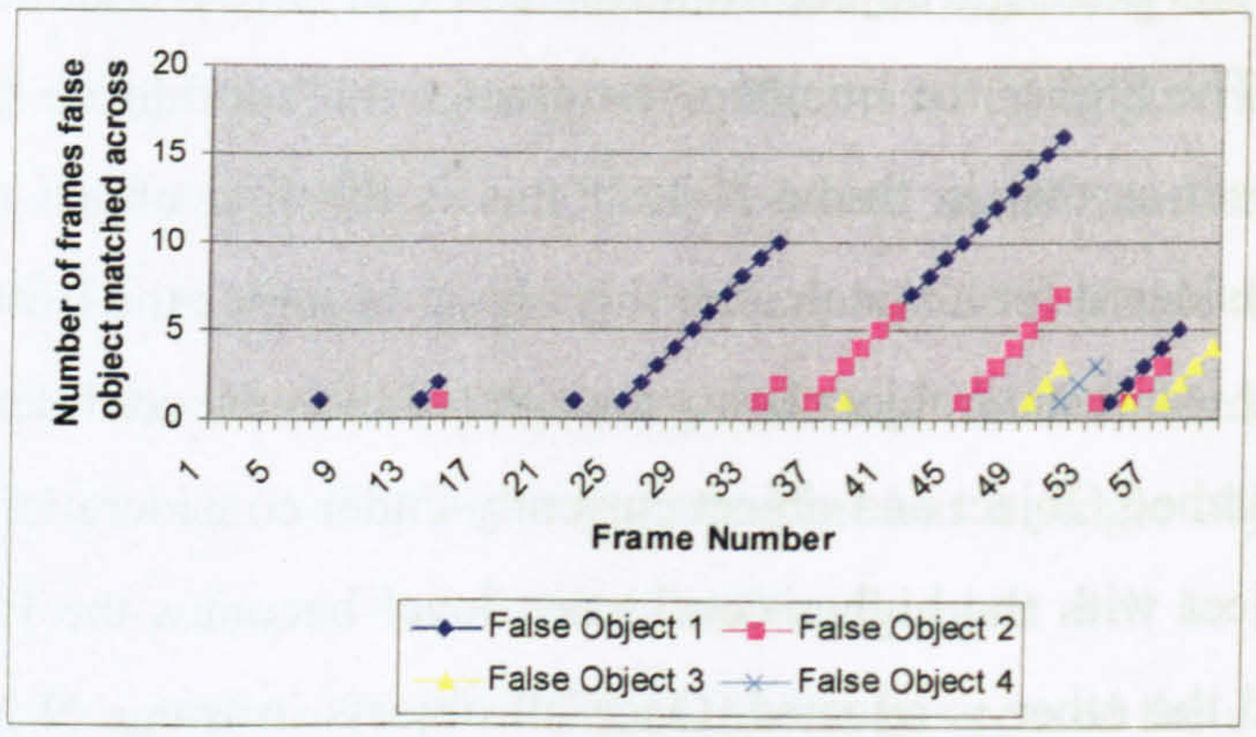

Figure 5.5 – The number of frames false objects are matched across in the Poole sequence

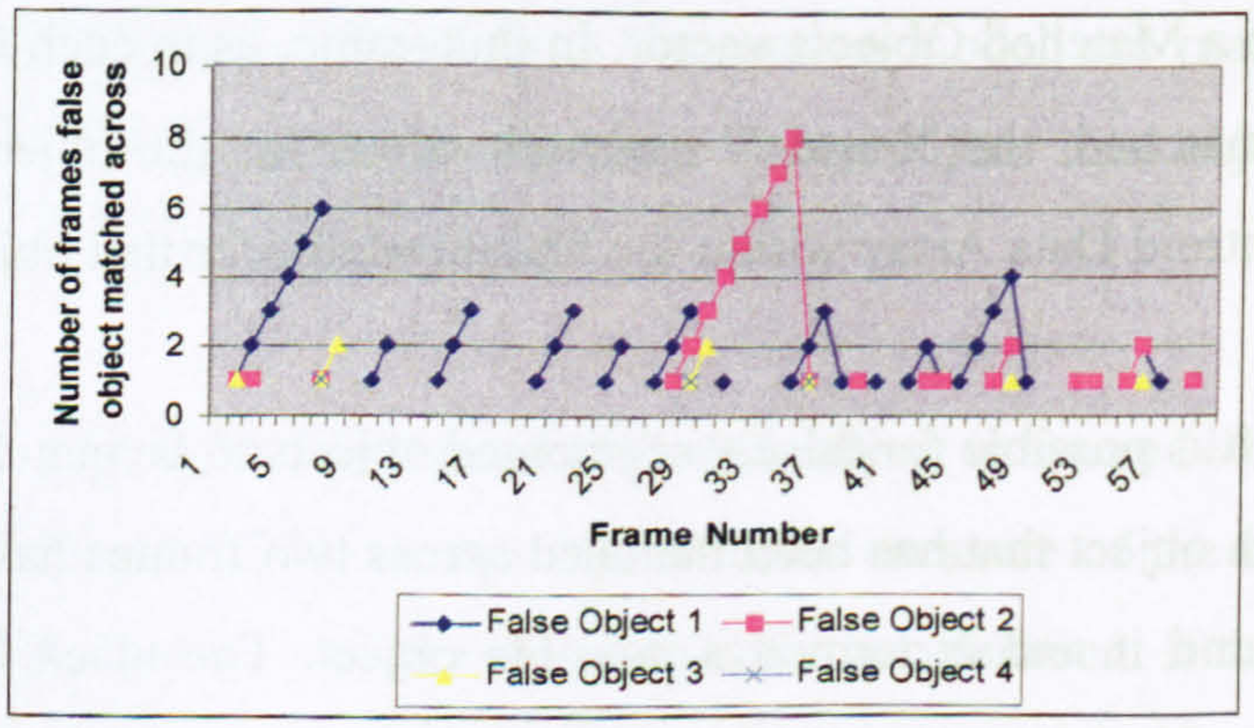

Figure 5.6 – The number of frames false objects are matched across in the Portsmouth sequence

5.2.5 Motion Prediction

The assumption by Wang et al that in a few adjacent frames object trajectories are close to a straight line (Wang et al, 2000) has been tested on two maritime test sequences. The plotting of centroid positions for moving objects in both the Poole and Dover test sequences, figures 5.7 and 5.8, showed that the assumption was true for the test sequences. In figure 5.8 the assumption is shown to be true over 20 frames however in figure 5.7 the object shows some movement away from a straight line after 15 frames.

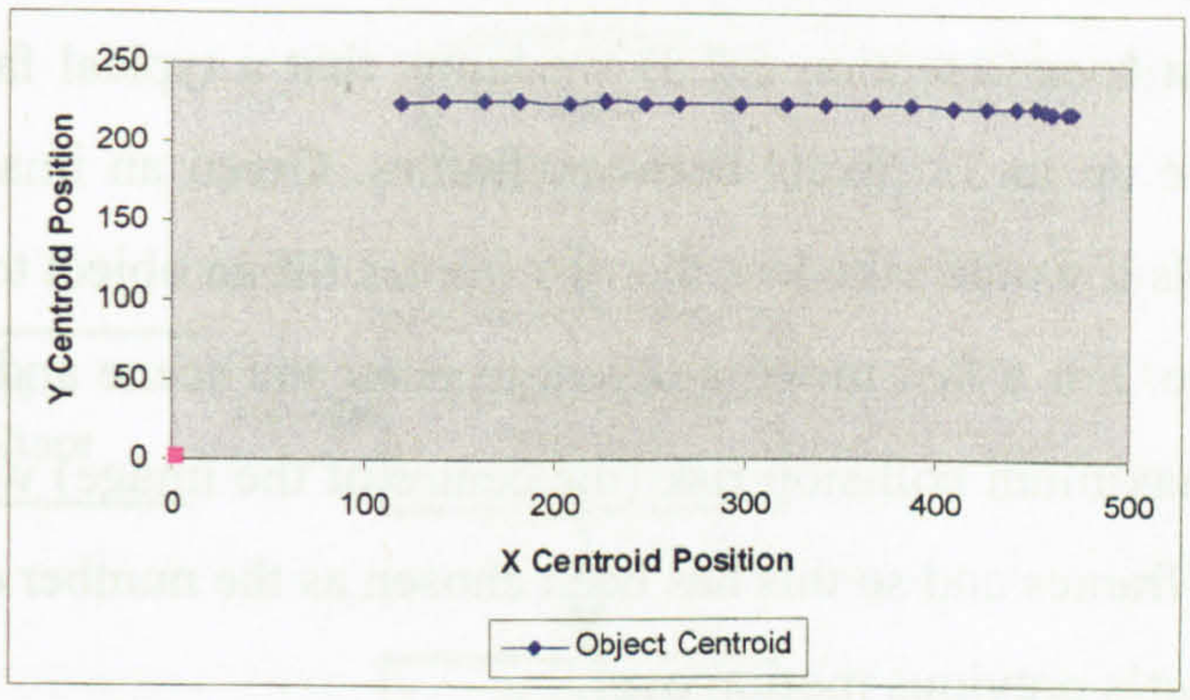

Figure 5.7 – The centroid position of the fast moving object in the Poole sequence for frames 10-30

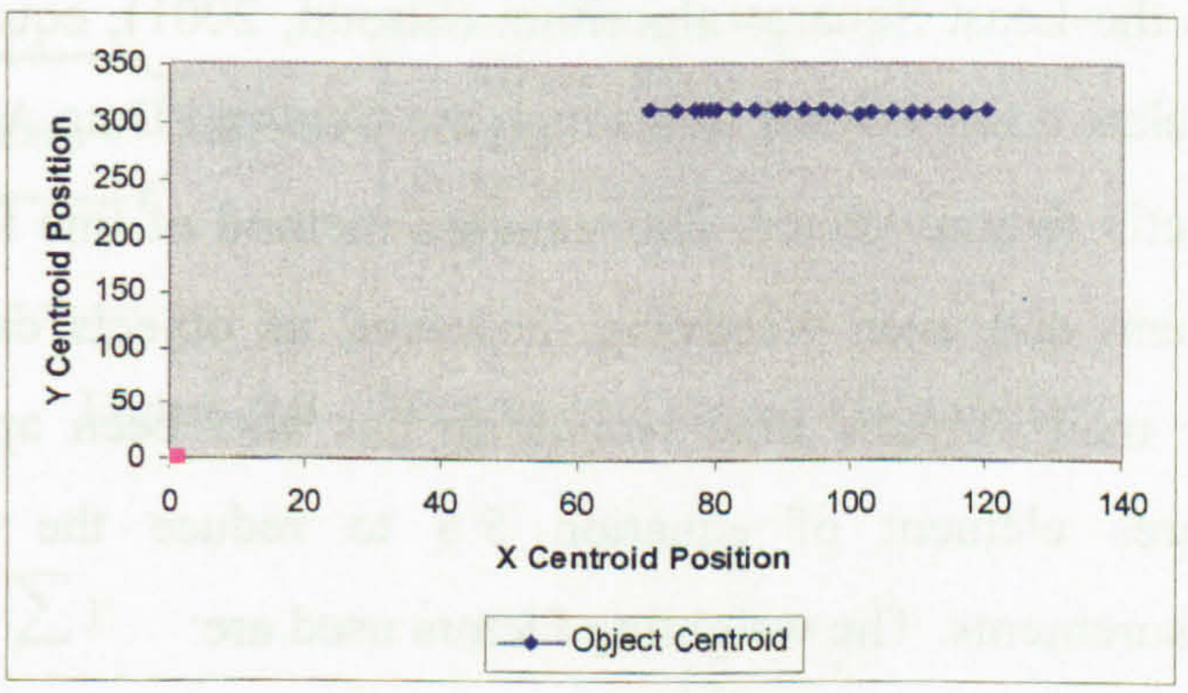

Figure 5.8 – The centroid position of the fast moving object in the Dover sequence for frames 90-110

From the result shown in figure 5.7 it was considered that a fast moving object would show a noticeable change in orientation after twenty frames so this was chosen as the period that the prediction of future motion would be calculated for.

Sumpter and Bulpitt (1998) say that the future behaviour of an object, in terms of its motion and appearance, can be implied from previous behaviour. Before any prediction can be made about an object's future motion then a certain number of frames worth of an object's previous motion data is required. If the number of frames chosen is too high then the object could have moved across the image and be leaving the scene before

the prediction is made. From the discussion on the maritime motion model about speed (section 5.2.3) we know that a typical fast moving object can move up to 33 pixels between frames. Given an image size of 512 x 512 pixels it would take less than 16 frames for an object to completely cross the scene. For a fast moving object to enter the scene and to arrive at the point of maximum collision risk (the centre of the image) would take a minimum of 8 frames and so this has been chosen as the number of frames to record an object's previous motion over.

The motion prediction process is shown in the flow chart in figure 5.9. It uses the Least Squares algorithm (Stroud, 2001), equations 5.5 and 5.6, to calculate a line of best fit through the Centroid Data Array, an element in an object's feature vector. The standard method of line fitting gives each data element and even weighting, however, as objects can change orientation over many frames time weighting has also been applied to the sum of squares element of equation 5.6 to reduce the confidence in older measurements. The weighting factors used are:

Centroid Data Array location:	1	2	3	4	5	6	7	8
Weighting Factor:	1.3	1.2	1.1	1.0	1.0	0.9	0.8	0.7

It then uses the equation for a straight line, equation 5.7, and the distance travelled over the 8 frames worth of entries in the Centroid Data Array to predict the likely centroid position of the object in a further 20 frames time. A line is then drawn from the current centroid position to the predicted centroid position. As the algorithm uses the centroid position of the object in each of the last 8 frames, the predicted future position takes into account the speed the object is moving at. The faster the movement of the object, the longer the line of predicted motion. Using the motion model assumption that a maritime objects motion is uniform across the image the predicted position in 20 frames time will be the average of the distance travelled between each of the Centroid Data Array entries.

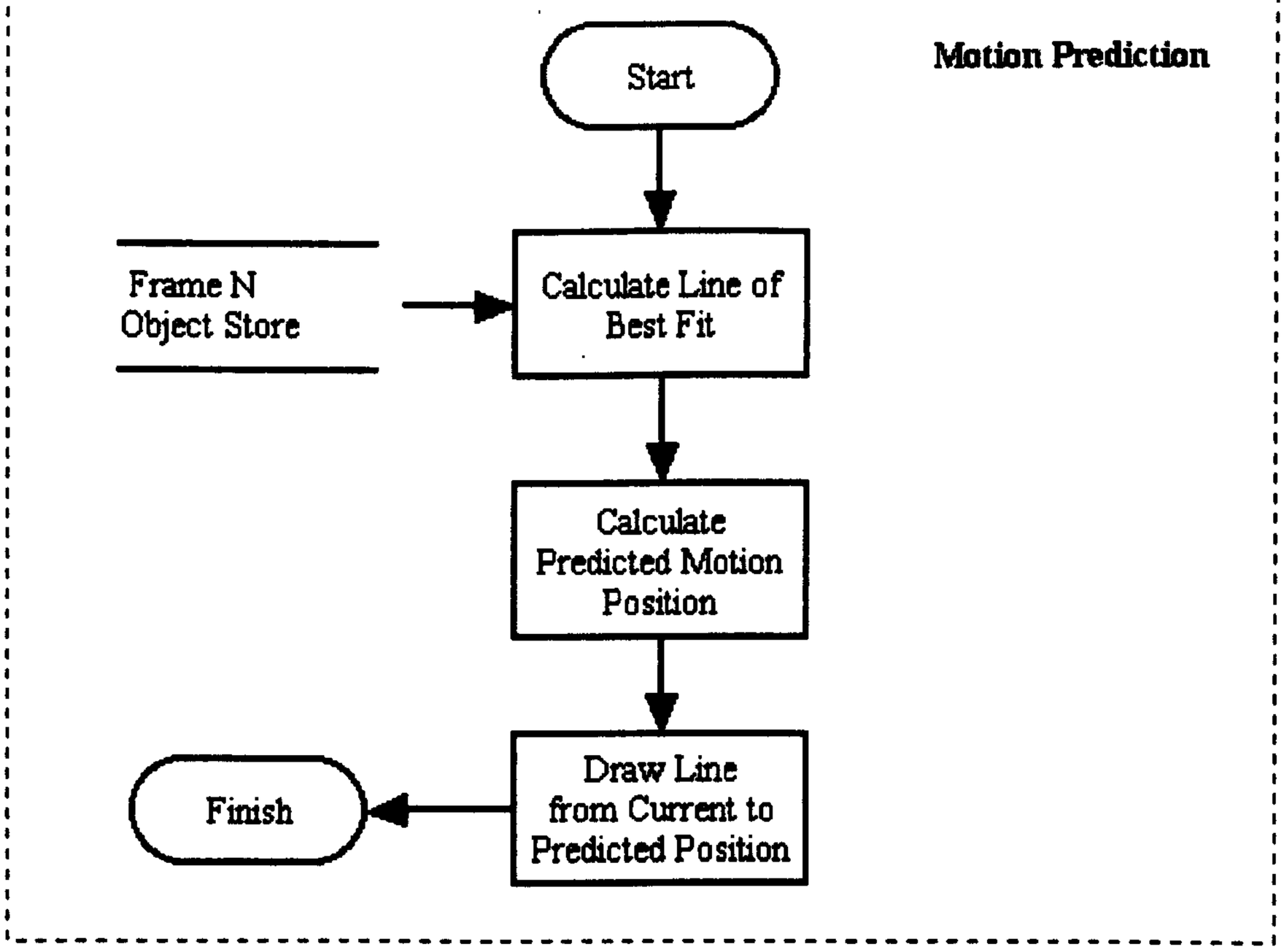

Figure 5.9 – Motion Prediction Flow Chart

$$an + b\sum x = \sum y \tag{5.5}$$

$$a\sum x + b\sum x^2 = \sum xy \tag{5.6}$$

$$\dot{y} = \dot{a} + bx \tag{5.7}$$

where:

x = object centroid value in the x direction

y = object centroid value in the y direction

n = the number of x,y centroid pairs

a and b = variables determined by dividing through equations 5.5 and 5.6 by the coefficient of a

5.3 Kalman Filter

The Kalman filter is an efficient recursive filter, which estimates the state of a dynamic system from a series of incomplete and noisy measurements. It was developed by Rudolph E. Kalman and first published in 1960 (Kalman, 1960). Being a recursive filter only the estimated state of the system from the previous time step and the current measurement are needed to compute the estimate for the next state. This is in contrast to batch estimation techniques where a history of observations and estimates are required.

Welch and Bishop (2004) give a practical introduction to the discrete Kalman filter and consider it to be very powerful as it supports estimations of past, present and future states of a system, even when the precise nature of the modelled system is unknown. The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements (Welch and Bishop, 2004). As such, the equations for the Kalman filter fall into two groups. Reid (2001) calls them the prediction step and the update step.

Prediction, also known as the time-update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. Update, also known as the corrector or measurement update equations are responsible for the feedback, for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. Welch and Bishop (2004) describe the two steps in simpler terms by saying the time update projects the current state estimate ahead in time and the measurement update adjusts the projected estimate by an actual measurement at that time. This ongoing discrete Kalman filter cycle is shown in figure 5.10.

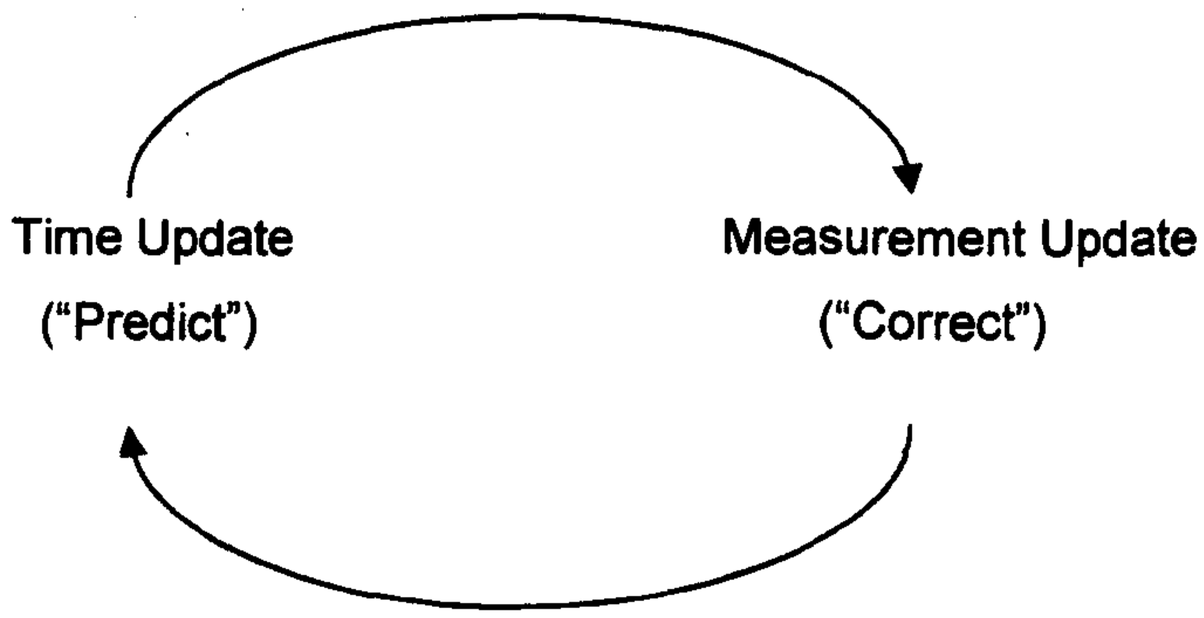

Figure 5.10 – The ongoing discrete Kalman filter cycle.

There are many versions of the discrete Kalman filter algorithm (Welch and Bishop, 2004). The implementation used here is that described by Ian Reid in his lecture notes on the subject (Reid, 2001). It uses interim equations for innovation and innovation update in the determination of the *a posteriori* estimate as they are considered to be more stable than the original formulas. Innovation is defined as the difference between the observation z_{k+1} and its prediction $z^{\wedge}_{k+1|k}$ made using the information available at time k. It is a measure of the new information provided by adding another measurement to the estimation process. Reid (2001) goes on to say this is important because, whereas z_{k+1} is in general statistically correlated to the set of observations up to and including time k, the innovation I_{k+1} is uncorrelated to the set of innovations up to and including time k and so effectively provides new information or 'innovation'.

The Kalman filter is not itself a tracking algorithm, it is a tool used to increase the robustness of a tracker by smoothing measurements taken in previous frames and hence increasing the confidence in these measurements. However, the filter can be used to predict the future position of an object being tracked.

It is useful to define the notation and assumptions used before describing the equations themselves.

z_k	-	Observation vector at time k
x_k	-	System state vector at time k
$x\hat{}_k$	-	Estimation of x at time k
P_k	-	Covariance matrix
F_k	-	State transition matrix
H_k	-	Output transition matrix
Q_k	-	Process noise covariance matrix
R_k	-	Measurement noise covariance matrix
K_k	-	Kalman gain matrix
v_k	-	Measurement noise
I_k	-	Innovation at time k
S_k	-	Innovation update at time k

The first assumption is that the system concerned can be modelled by the state transition equation,

$$x_{k+1} = F_k x_k \tag{5.8}$$

It is also assumed that the observations of the state are made through a measurement system represented by a linear equation of the form,

$$z_k = K_k x_k + v_k \tag{5.9}$$

Further assumptions are:

- The measurement noise v_k is uncorrelated, zero-mean white noise with known covariance matrix.
- Q_k and R_k are symmetric positive semi-definite matrices.
- The initial system state has a known mean and covariance matrix.

Given the above assumptions the task is to determine, given a set of observations $z_1,\ldots,z_{k+1}$, an optimal estimate of the state x_{k+1} denoted by $\hat{x}_{k+1}$.

The time update equations used to predict the state and variance at time $k+1$ dependent on information at time k are:

$$\hat{x}_{k+1|k} = F_k \hat{x}_k \tag{5.10}$$

$$P_{k+1|k} = F_k \hat{P}_{k|k} F_k^T + Q_k \tag{5.11}$$

The measurement update equations used to update the state and variance using a combination of the predicted state and the observation z_{k+1} are:

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} I_{k+1} \tag{5.12}$$

$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T \tag{5.13}$$

where:

$$I_{k+1} = z_{k+1} - H_{k+1} \hat{x}_{k+1|k} \tag{5.14}$$

$$S_{k+1} = R_{k+1} + H_{k+1} P_{k+1|k} H_{k+1}^T \tag{5.15}$$

$$K_{k+1} = P_{k+1|k} H_{k+1}^T S^{-1} \tag{5.16}$$

The complete picture of the Kalman filter operation, taken from Welch and Bishop (2004), is shown in figure 5.11.

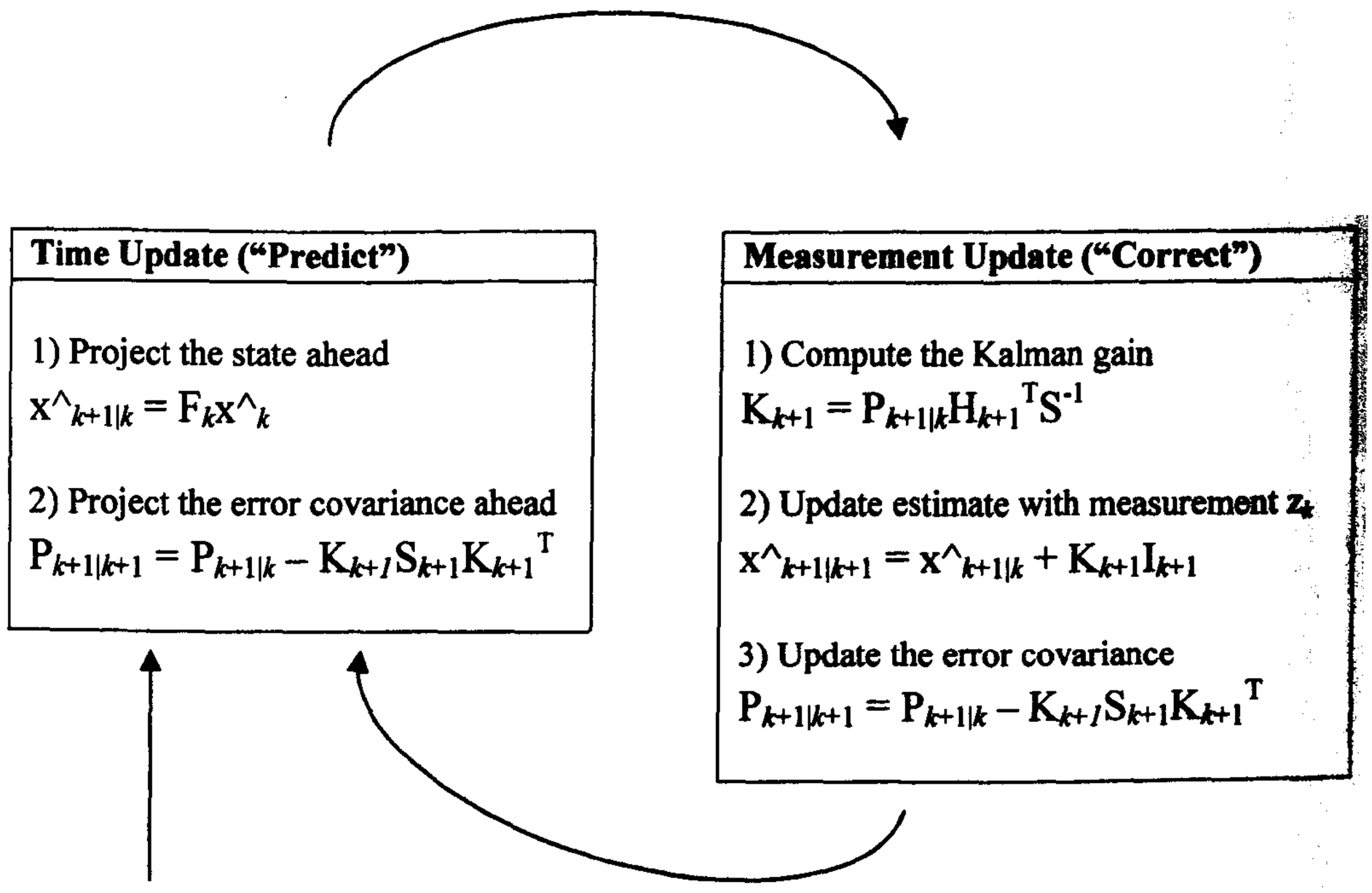

Figure 5.11 – The complete Kalman filter operation

5.3.1 Object Tracking Using The Kalman Filter

Figure 5.10 shows a flow chart of the tracking process utilising the Kalman filter. As the Kalman filter only performs the task of predicting the position of the objects in the next frame the other tracking elements such as determining the centroid and co-ordinates of objects and identifying the objects to the user are the same as those used in the motion model method.

In the first frame, N, of a sequence the centroid, X and Y co-ordinates, and area of all segmented objects are recorded. In frame N+1 the object with the closest area to one found in the first frame is considered a match. From the centroid of the matched object in frames N and N+1 the velocity is calculated using equations 5.1-5.3 and this, together with the position are

used to initialise a Kalman filter. This initialisation is then used to predict the position of the object in the following frame, N+2. The position variance defines the search area around the predicted position in which the object is looked for in the next frame. The variance is calculated from the differences between the manually derived object centroid locations and the centroid locations of the objects segmented in chapter 4.

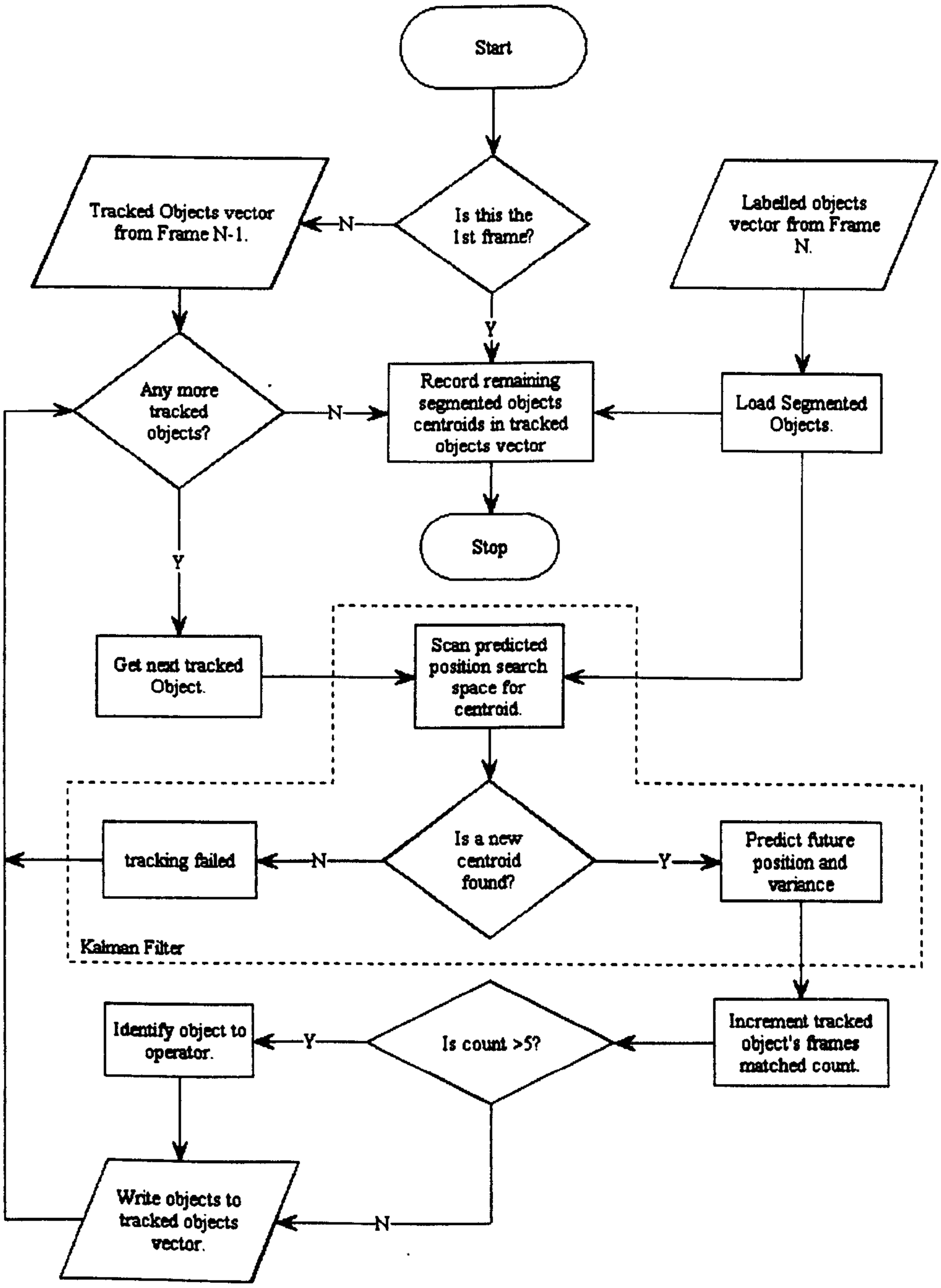

Figure 5.12 – Kalman filter tracking flow chart

If an object centroid is found in this search area a match is considered to have occurred and this new centroid becomes the measurement incorporated in the Kalman filter to predict the position of the object in the next frame. In addition to x_0 and P_0 the Kalman filter also requires initial values for the F, H, Q, and R matrices.

x_k

x_k is a state vector containing the actual centroid position from frame N+1 and the velocity along the line calculated from the object data in frames N and N+1. The format of x_k is:

$$x_0 = \begin{bmatrix} position(k) \\ velocity(k) \end{bmatrix} \tag{5.17}$$

P_0

P_0 is a diagonal matrix and is the covariance matrix of the Kalman filter output. This matrix is updated each iteration but requires initial values to be set. If the initial position and velocity are known perfectly this matrix should be initialised with zeros. However, if the initial position and velocity are not perfectly known the covariance matrix should be initialised with a suitably large number on its diagonal. The filter will then prefer the information from the first measurements over the information already in the model.

Welch and Bishop (2004) say that as it turns out, the alternative choice is not critical. Almost any initial value could be chosen so long as $P_0 \neq 0$ and the filter would *eventually* converge. They start their filter with $P_0 = 1$. As the initial position and velocity are not perfectly known in this case and the initial value is not critical, Welch and Bishop's lead has been followed and 1 used to initialise this matrix.

$$P_0 = \begin{bmatrix} location\ variance & 0 \\ 0 & velocity\ variance \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.18)$$

F_k

Given that measurements in this system are taken at discrete time intervals, ΔT, (1/frame rate of 10 frames per second) and assuming a constant velocity then:

$$F_k = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \quad (5.19)$$

The matrix F works in the following manner:

position (k+1) = position (k) + velocity (k) * ΔT

velocity (k+1) = velocity (k)

H

H contains a correction factor between the measurement and the actual position. The segmentation results show a maximum average error of 1.5% for the Sea Characterisation method of choice so the correction factor for has been set to 1.015.

$$H = [\,1.015 \quad 0\,] \quad (5.20)$$

Q_k

The system noise covariance matrix Q is another diagonal matrix containing the location and velocity variances as in P. The values contained here, however, are set to how much the position and velocity can change. As the system assumes a constant object velocity the values for the variances can also be assumed to remain constant. Therefore, they are set to unity:

$$Q = \begin{bmatrix} location\ variance & 0 \\ 0 & velocity\ variance \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.21)$$

R_k

R is a matrix of the same dimensions as the measurement vector z and contains the value of the measurement position variance. Here R is 1x1 as the measurement will be the centroid position of an object in one direction. This parameter represents the measured position variance and is the same as H. Therefore:

$$R_k = [\ 1.015\ ] \quad (5.22)$$

5.3.2 Predictive Tracking

The Kalman filter uses the series of object position and velocity measurements up to and including that taken in the current frame to predict the position of the object in the next frame. In order to predict the position any further forward in time than the next frame it is necessary to reapply the filter for each time interval. As the prediction in this system is to be made for the object position in frame N+20 the filter has to be applied twenty times per frame to achieve the desired predicted position. As measurements cannot be taken to update the predicted value, the prediction alone is used to project further into the future.

5.4 Tracking Results

The Poole and Portsmouth test sequences have been used in the development of the two tracking algorithms. The Poole sequence includes a

fast moving object travelling from right to left and a moored craft in the centre of the image. The Portsmouth sequence contains slow moving objects moving from right to left. The results are presented for each of the two sequences used.

5.4.1 Poole Sequence Tracking Results

Figure 5.13 shows the output of frames 6, 16, and 26 from the Poole test sequence after the motion model tracking algorithm has been applied. A white box is shown around the objects that have been matched across a minimum of five frames and a black box around that object matched in between two and five frames. In figure 5.14a, the actual and calculated centroid positions of object 1, the large dark stationary boat, are plotted for every fifth frame through a sixty frame sequence. Figure 5.14b shows the actual and calculated centroid positions for object 2, the fast moving object entering from the right. The actual centroid positions have been determined manually and the calculated centroid positions determined by the algorithm.

The graph in figure 5.15 shows the percentage error between the centroid values shown in figures 5.14a and 5.14b. The calculated centroid error for object 1 is less than 1% in all except the X centroid in frame 1 where the error is 3.7%. The centroid error for object 2 is higher at typically 2% but in frames 15 and 20 the X centroid error rises to 7 and 6.8% respectively. These higher percentages are due to the segmentation of the object. This is shown in figure 5.14b where the rear of the object has been segmented as sea owing to the grey levels present being within the grey level range for the sea in that part of the image.

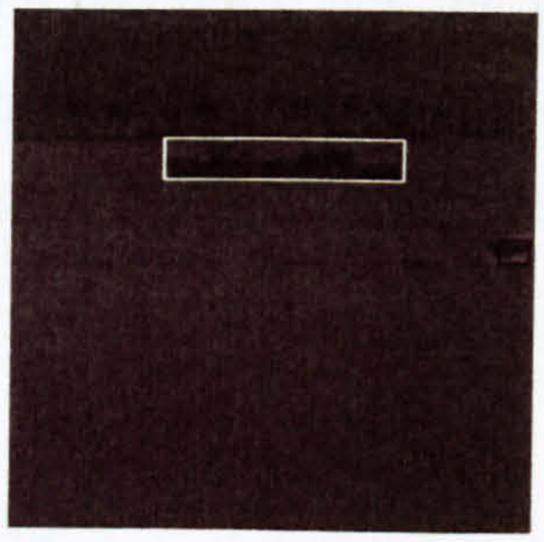

Figure 5.13a

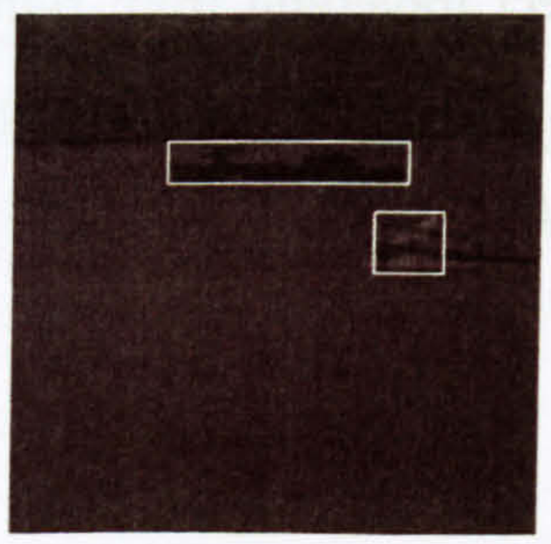

Figure 5.13b

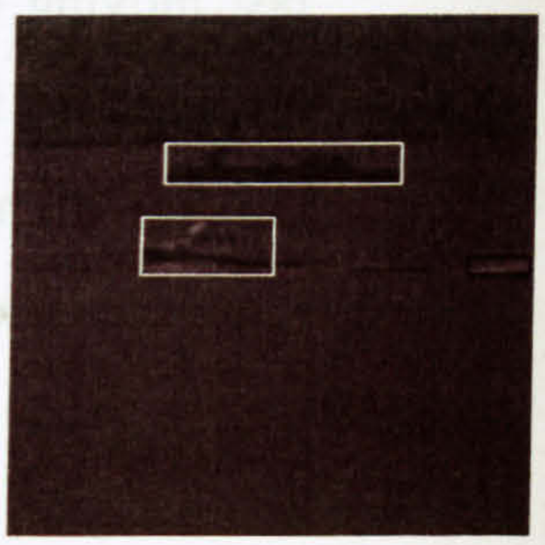

Figure 5.13c

Figure 5.13 – 3 tracked frames from the Poole sequence

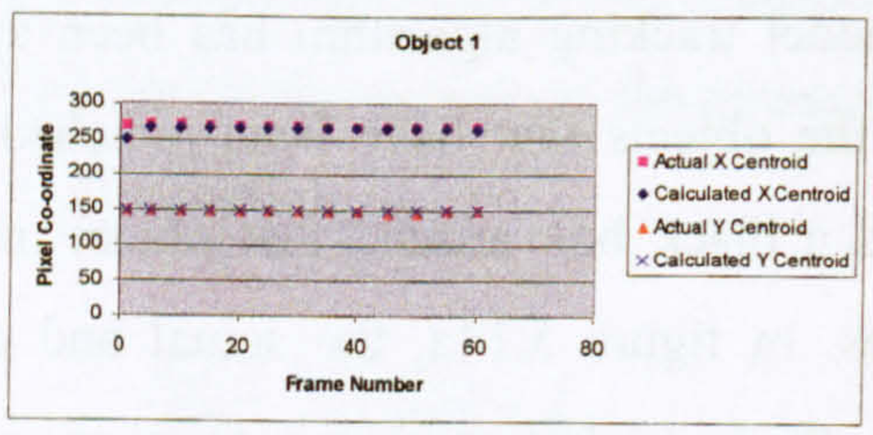

Figure 5.14a – Object 1

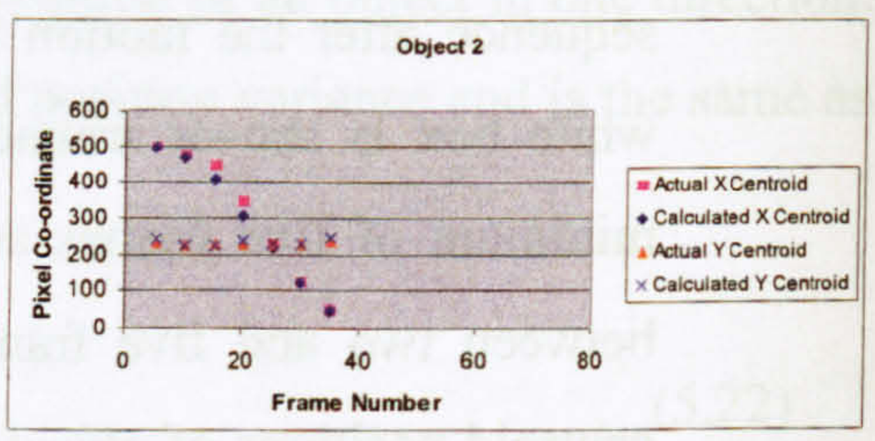

Figure 5.14b – Object 2

Figure 5.14 – Actual and Tracked Centroid Comparison by Pixel Position for the Poole sequence

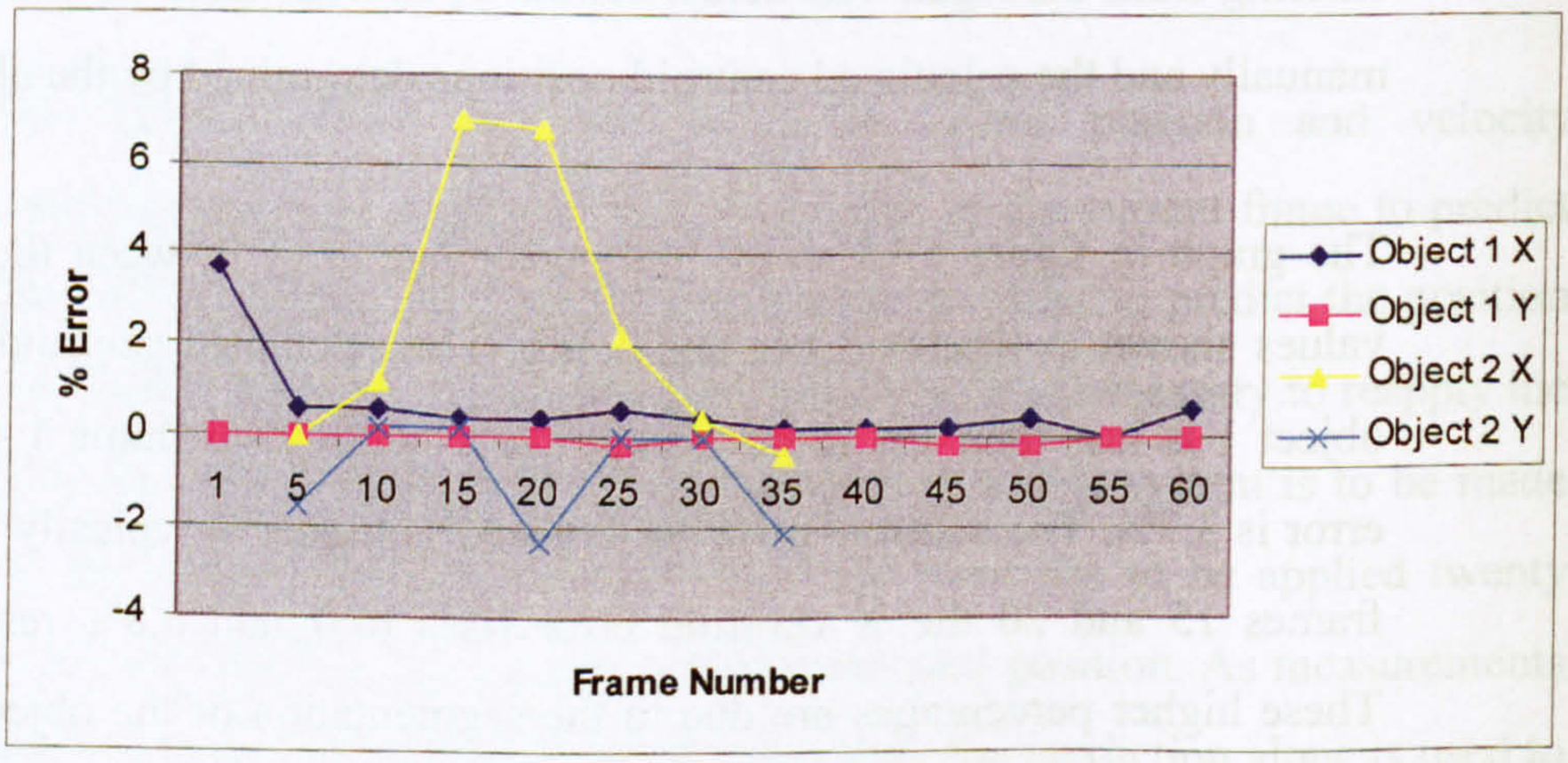

Figure 5.15 – Actual and Tracked Centroid Comparison by % Error for the Poole sequence

The Kalman filter also successfully tracked the real objects present in the Poole sequence. As the determination of object centroid and co-ordinates and the identification of the objects to the user are the same for both trackers

the results for the Kalman filter tracker are the same as those in figures 5.13 – 5.15.

The difference between the two trackers becomes evident when the number of false objects tracked is considered. As the Kalman filter searches an omni-directional space around the current centroid it matches false objects in a greater number of frames. Figure 5.16 shows a frame from the Poole sequence with a false object identified as a maritime object. The false object is the wake caused by the fast moving object. Although not a maritime object the wake does contain grey levels that are distinctly different from those of the sea and this is why it has been segmented, matched and tracked. The number of false objects tracked in the Poole sequence by the Motion Model Tracker is shown in figure 5.17 and figure 5.18 shows the same result for the Kalman filter tracker. It can be seen that the Motion model tracker has a maximum of two false objects tracked in seven frames and one false object in a total of sixteen frames. The Kalman filter tracker however, has a maximum of three falsely tracked objects in four frames. This method also tracks two false objects in eleven frames and one object in twenty frames.

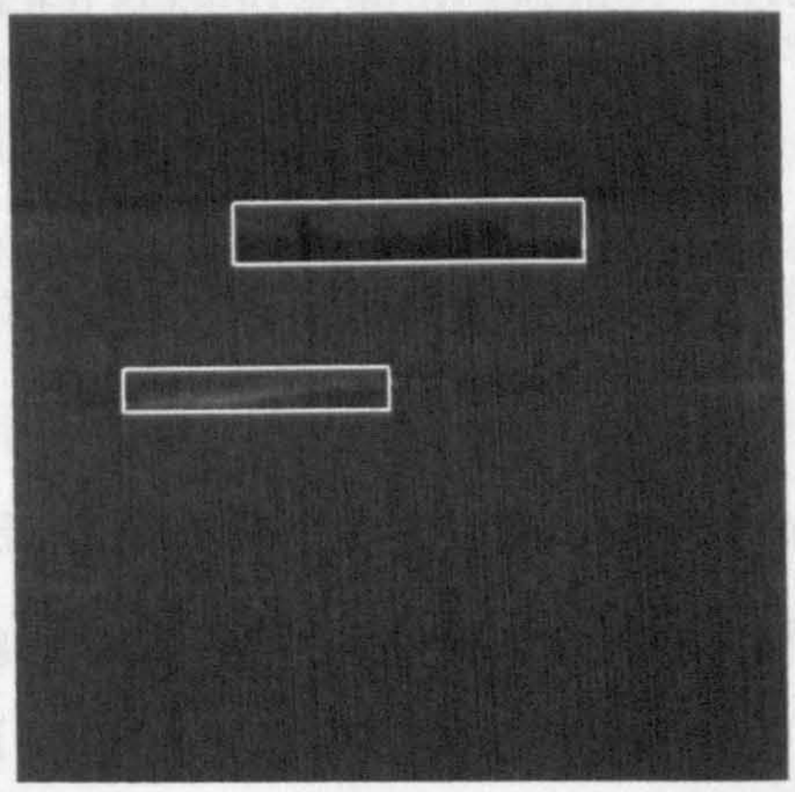

Figure 5.16 – False tracked object from Poole sequence, frame 42

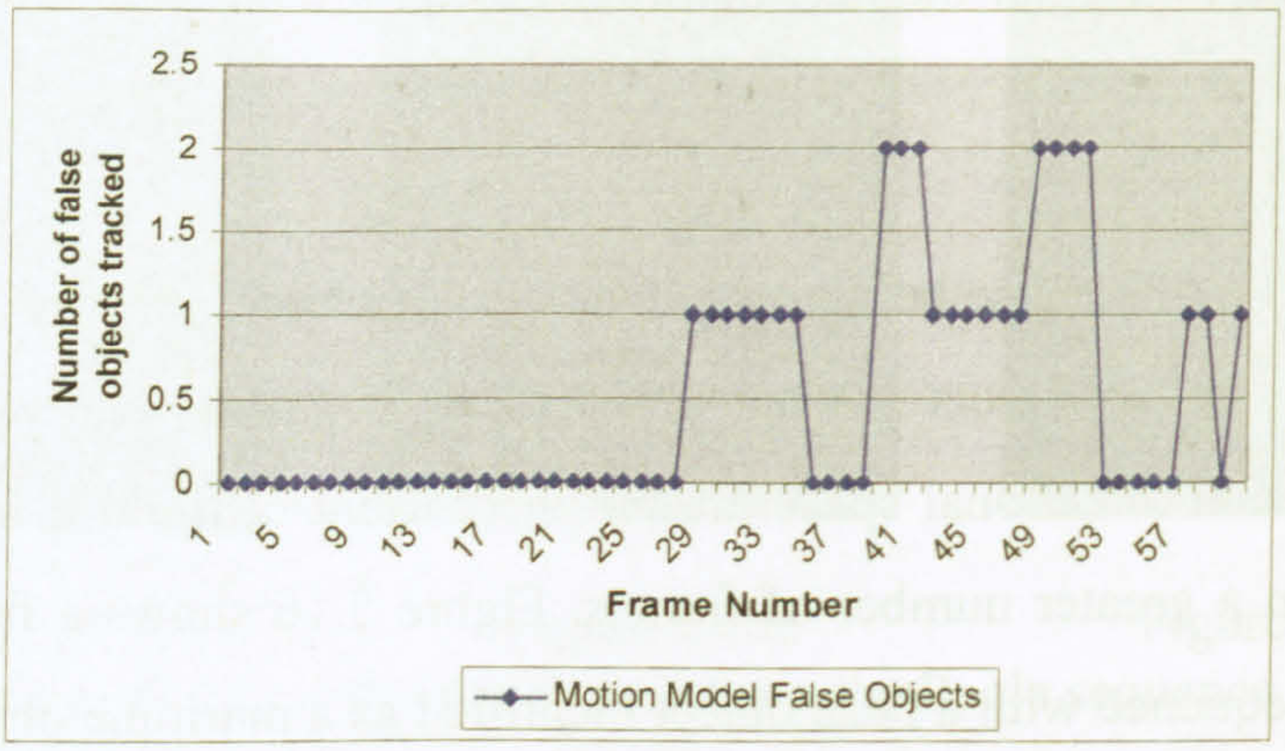

Figure 5.17 - Number of false objects tracked in Poole sequence by the Motion Model Tracker

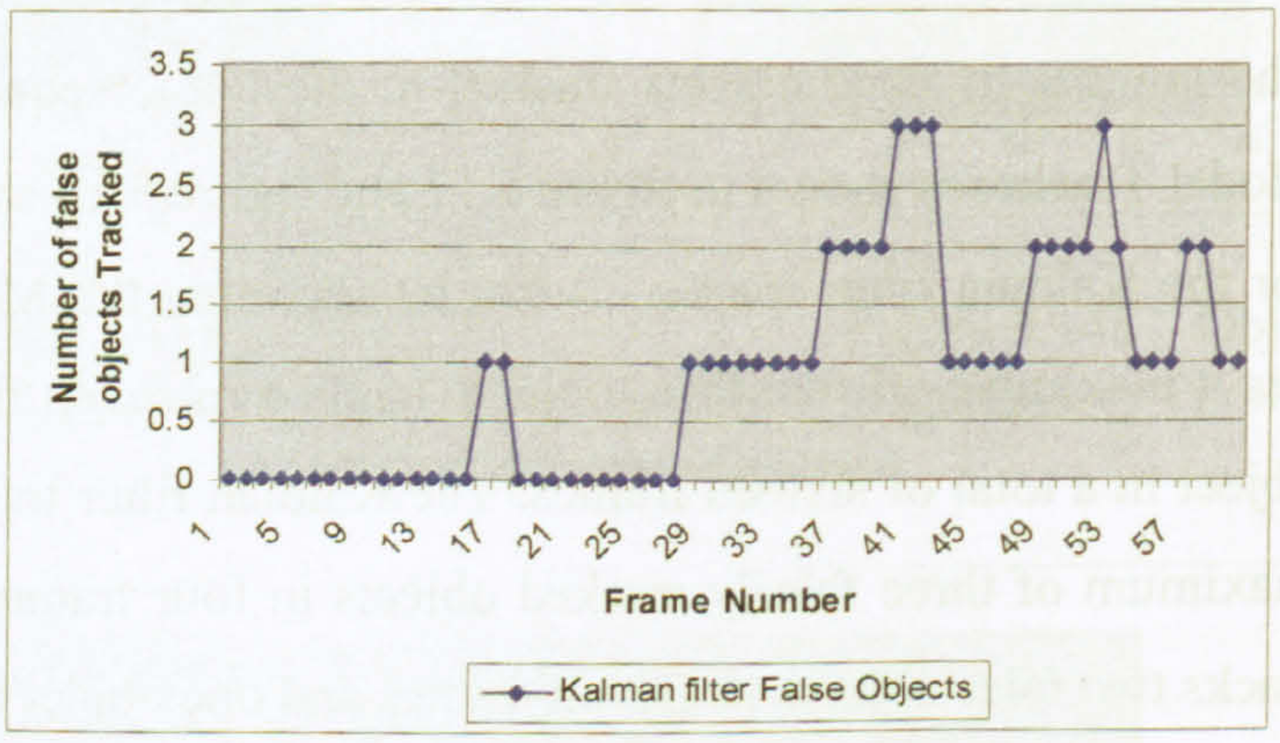

Figure 5.18 – Number of false objects tracked in the Poole sequence by the Kalman filter tracker

5.4.2 Portsmouth Sequence Tracking Results

Figure 5.19 shows the output from three frames from the Portsmouth test sequence after the Motion Model tracking algorithm has been applied. A white box is shown around the objects that have been matched across a minimum of five frames. In figure 5.19a this also includes a false object. This false object has occurred due to the narrow band of grey levels comprising the sea. This has left the area of wake behind the left most object several grey levels darker than the grey level range for that part of the image and it has therefore been segmented as an object. The false object is

eliminated later in the sequence, as darker grey levels are included in the characterisation of the sea. In figure 5.20a, the actual and calculated centroid positions of object 1, the centrally located moving boat, are plotted for every fifth frame through a sixty frame sequence. Figure 5.20b shows the actual and calculated centroid positions for object 2, the small moving boat entering from the right. As with the Poole sequence the actual centroid positions have been determined manually and the calculated centroid positions determined by the tracking algorithm.

Figure 5.21 shows the percentage error in the actual and calculated centroid for the objects in the Portsmouth sequence. The highest error in this sequence is 2.5% and all except a single result are below 2%. The low error rate is considered to be due to the increased contrast between the sea and objects in the Portsmouth sequence over that in the Poole sequence.

As with the Poole sequence the Kalman filter tracker successfully tracked all real objects and also the false object shown in figure 5.19a. The false object is, as with the Poole sequence, the wake caused by the central object containing grey levels significantly different from those of the sea. The false objects matched and tracked again provide the difference between the two tracking algorithms. Figures 5.22 and 5.23 show the number of false objects tracked as maritime objects (over more than 5 frames) for both trackers.

The Motion model algorithm tracks a maximum of one false object and it does this in seven frames. The Kalman filter algorithm tracks a maximum of two false objects and these are tracked in a total of five frames. One false object is also tracked in a further nineteen frames.

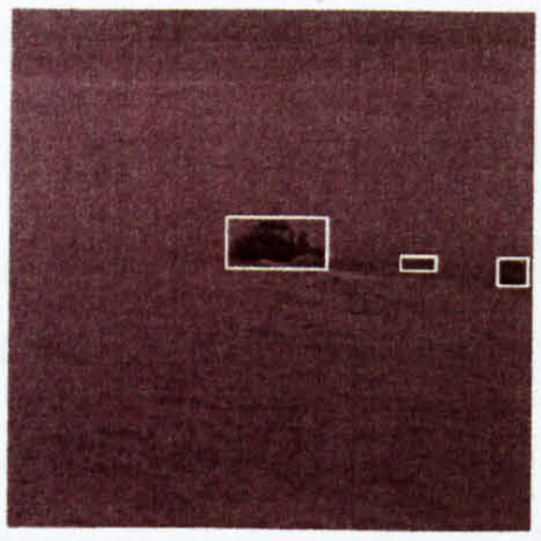

Figure 5.19a

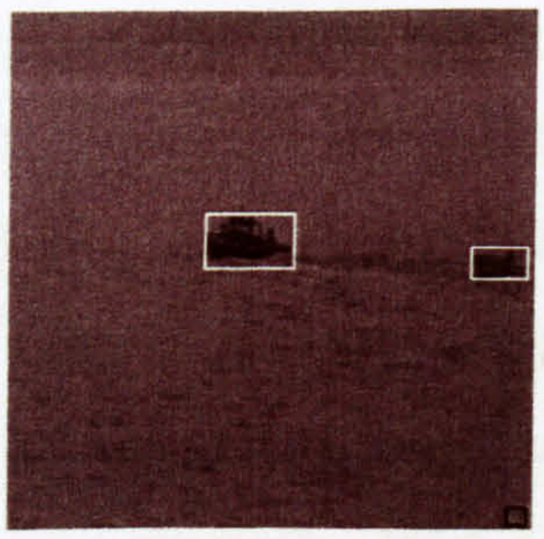

Figure 5.19b

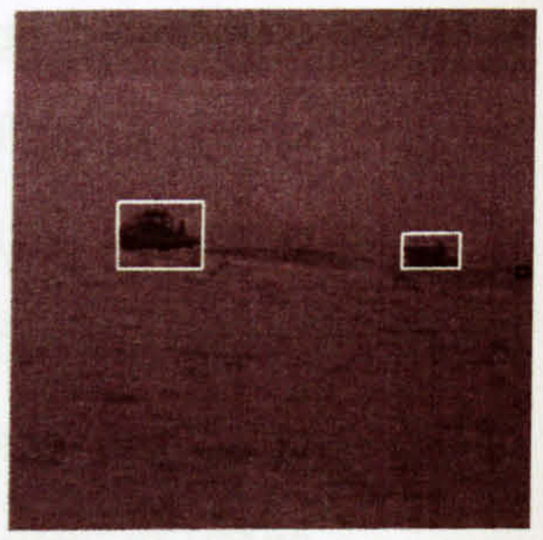

Figure 5.19c

Figure 5.19 – 3 tracked frames from the Portsmouth sequence

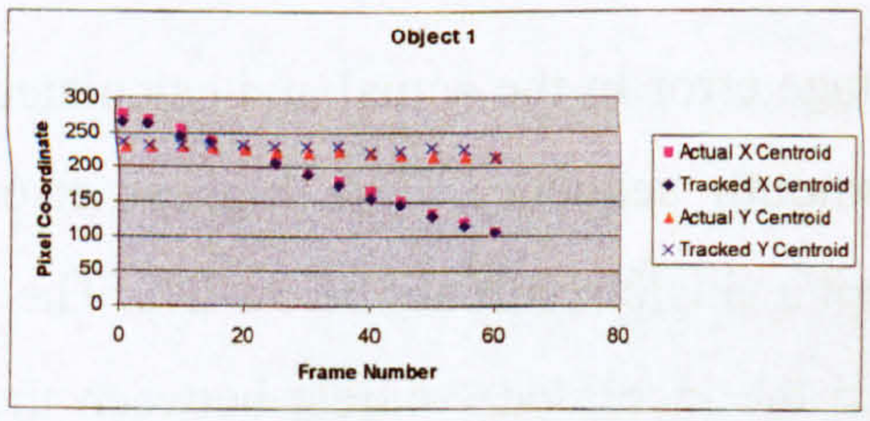

Figure 5.20a – Object 1

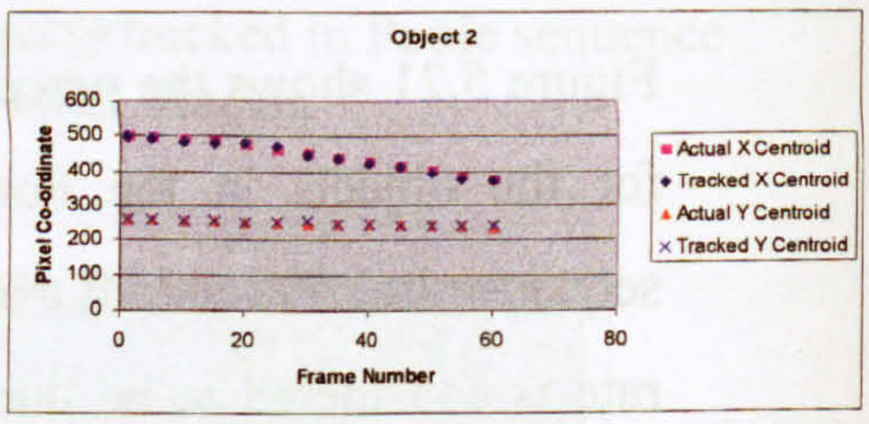

Figure 5.20b – Object 2

Figure 5.20 - Actual and Tracked Centroid Comparison by Pixel Position for the Portsmouth sequence

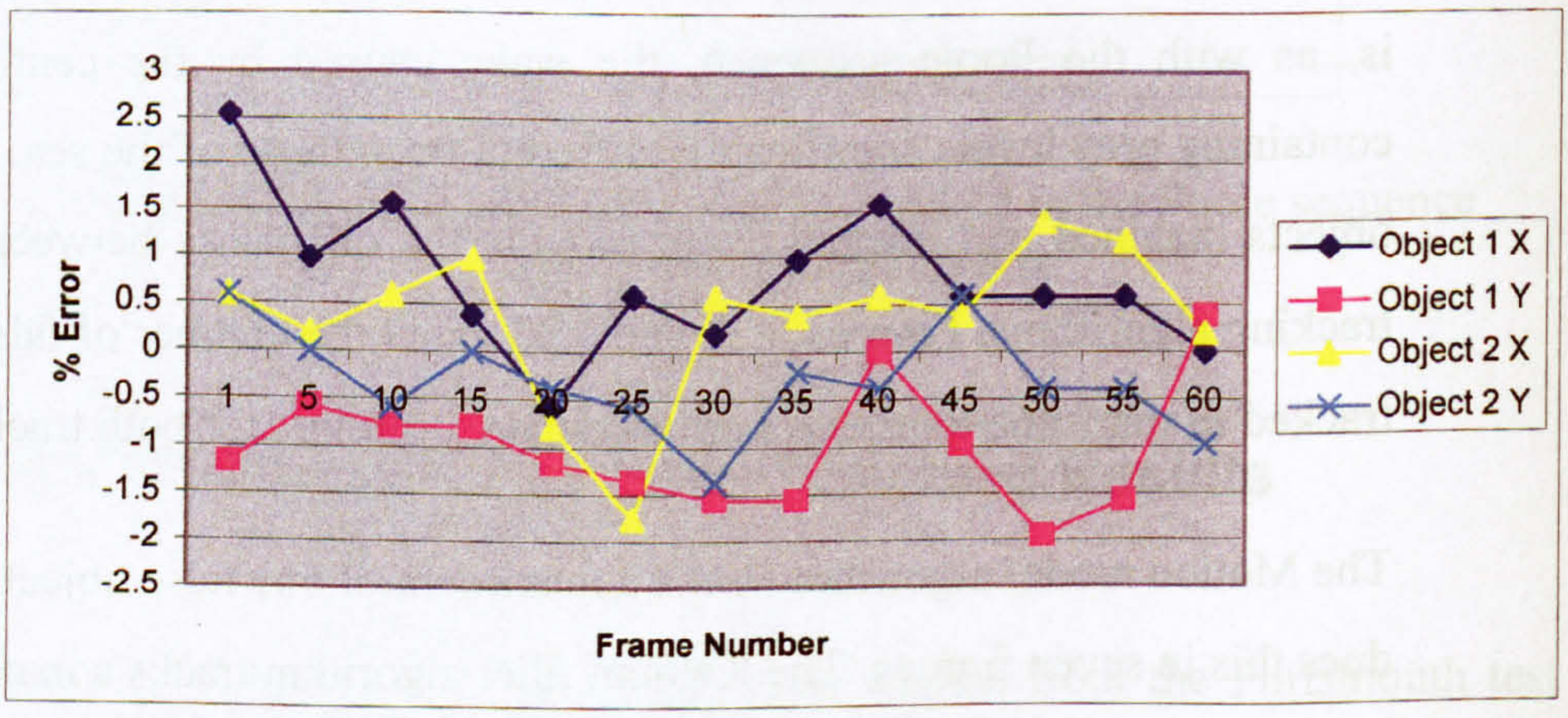

Figure 5.21 – Actual and Tracked Centroid Comparison by % Error for the Portsmouth sequence

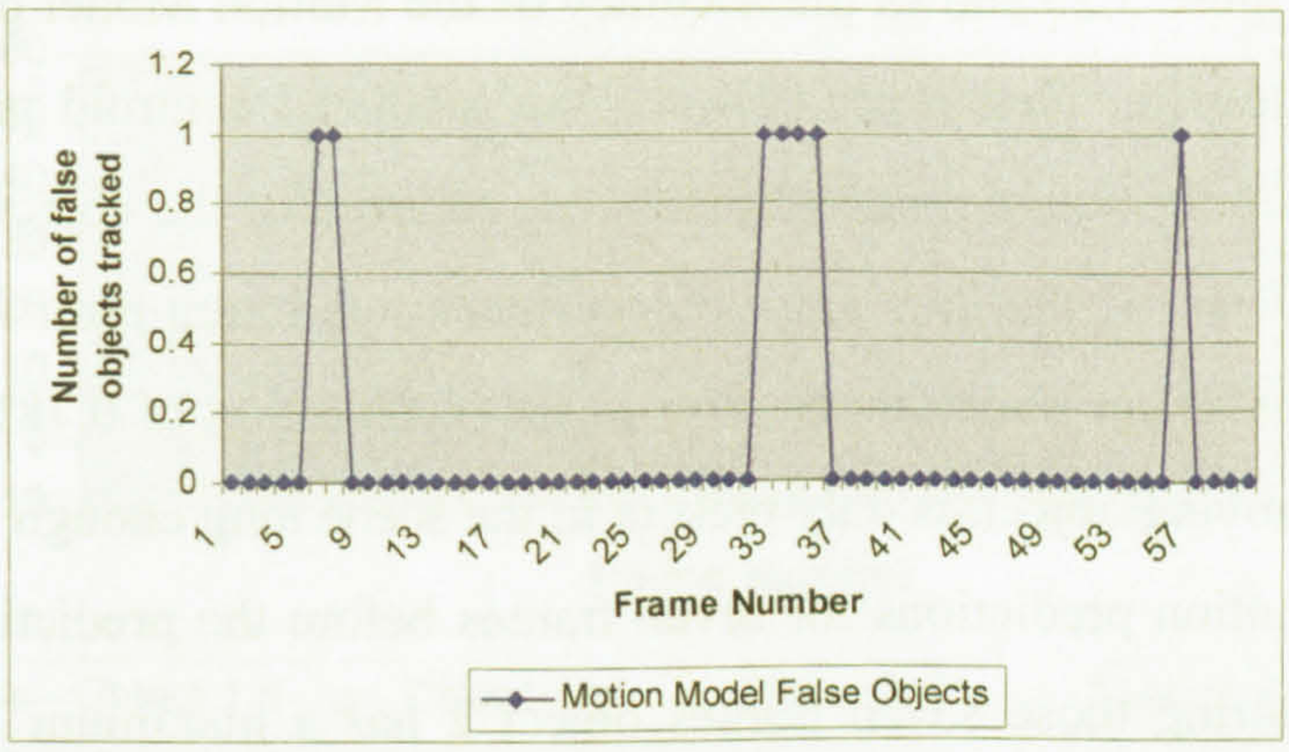

Figure 5.22 – Number of false objects tracked in the Portsmouth sequence by the Motion Model tracker

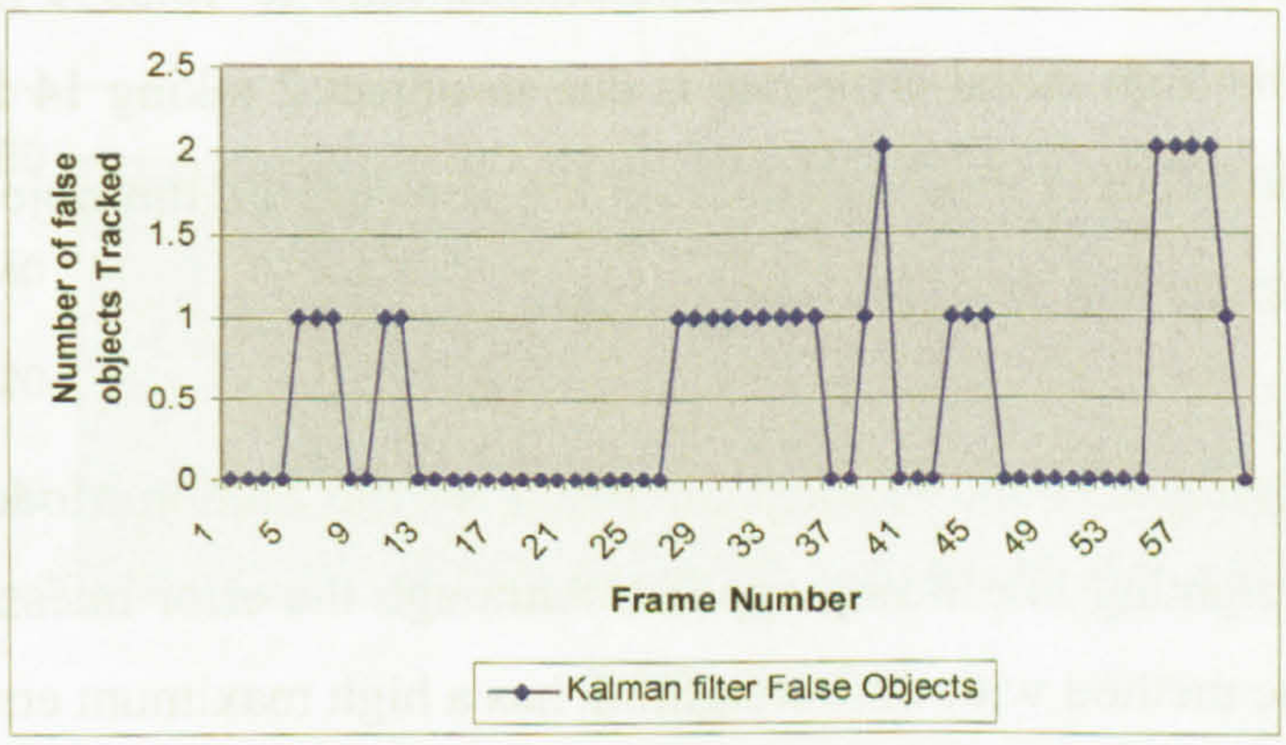

Figure 5.23 – Number of false objects tracked in the Portsmouth sequence by the Kalman filter tracker

5.4.3 Motion Prediction Results, Poole Sequence

Shown in figures 5.24 and 5.25 are the results of the Motion Model predictive tracking algorithm on frames 16 and 26 from the Poole sequence. It can be seen in the object positions that the lower object is moving much faster than the large dark object in the centre of the image. This also shows the effect of the predicted motion being based on object velocity, as the line of predicted motion is significantly longer for the lower object than the large dark object.

Figure 5.26 shows the accuracy of the Motion Model prediction by plotting the percentage error between the predicted centroid position of the object and the actual (manually derived) centroid of the object in the future frame. Object 1, the large dark object, has a maximum error in the X direction of 7.8% and a maximum error in the Y direction of 0.78%. Object 2, the fast moving object, is only present in the scene long enough to provide the future motion predictions for seven frames before the prediction leaves the scene. During those seven frames object 2 has a maximum X direction error of 54.23% and a maximum Y direction error of 4.45%. Although the error rate of 56.05% for the object 2 X direction is extremely high it should be observed that this rate drops dramatically to 10.39% in just seven frames. The high initial error rate is due to object 2 taking 14 frames to fully enter the scene during which time the centroid of the object will not be truly representative of the whole object.

Figure 5.27 shows the same result for the Motion Model method with time weighting line fitting applied. Although the error rates appear to be similar the method with time weighting has a high maximum error in all four plots.

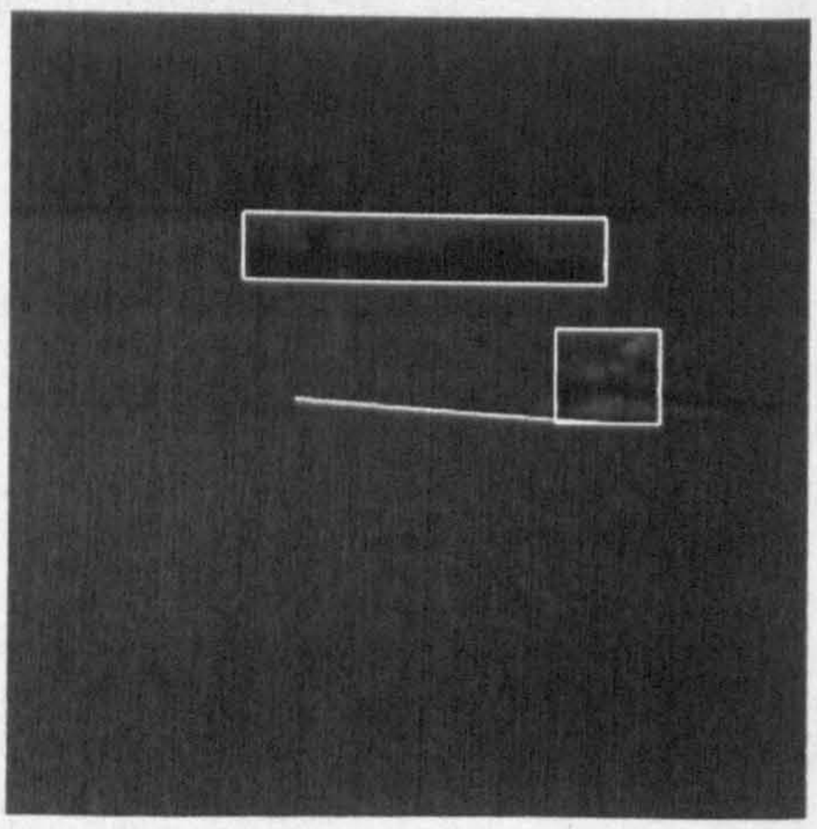

Figure 5.24 – Poole Sequence, frame 16

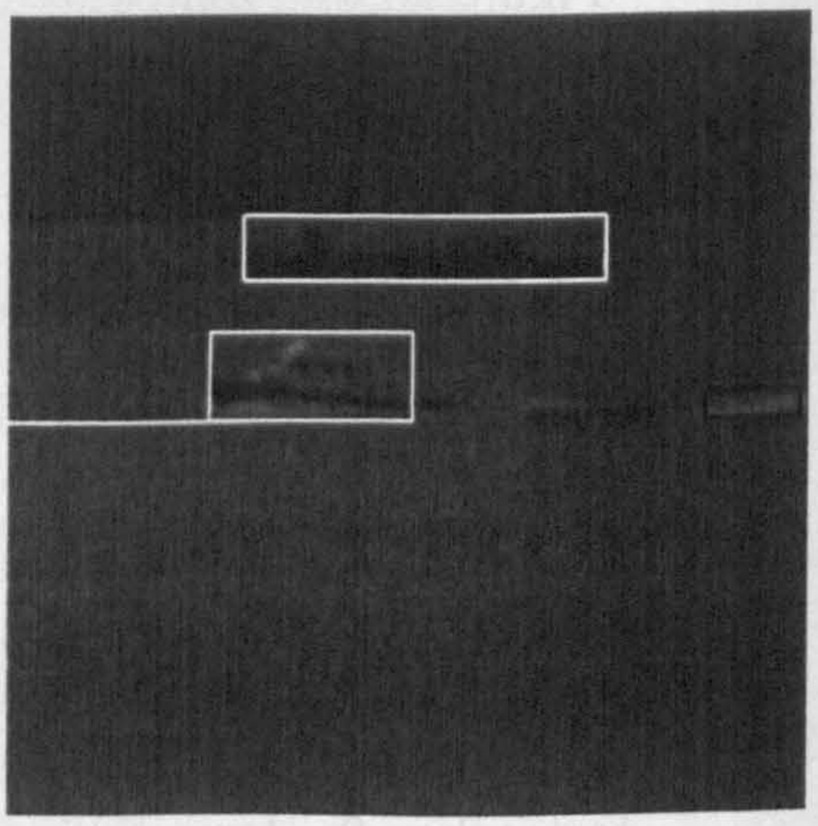

Figure 5.25 – Poole Sequence, frame 26

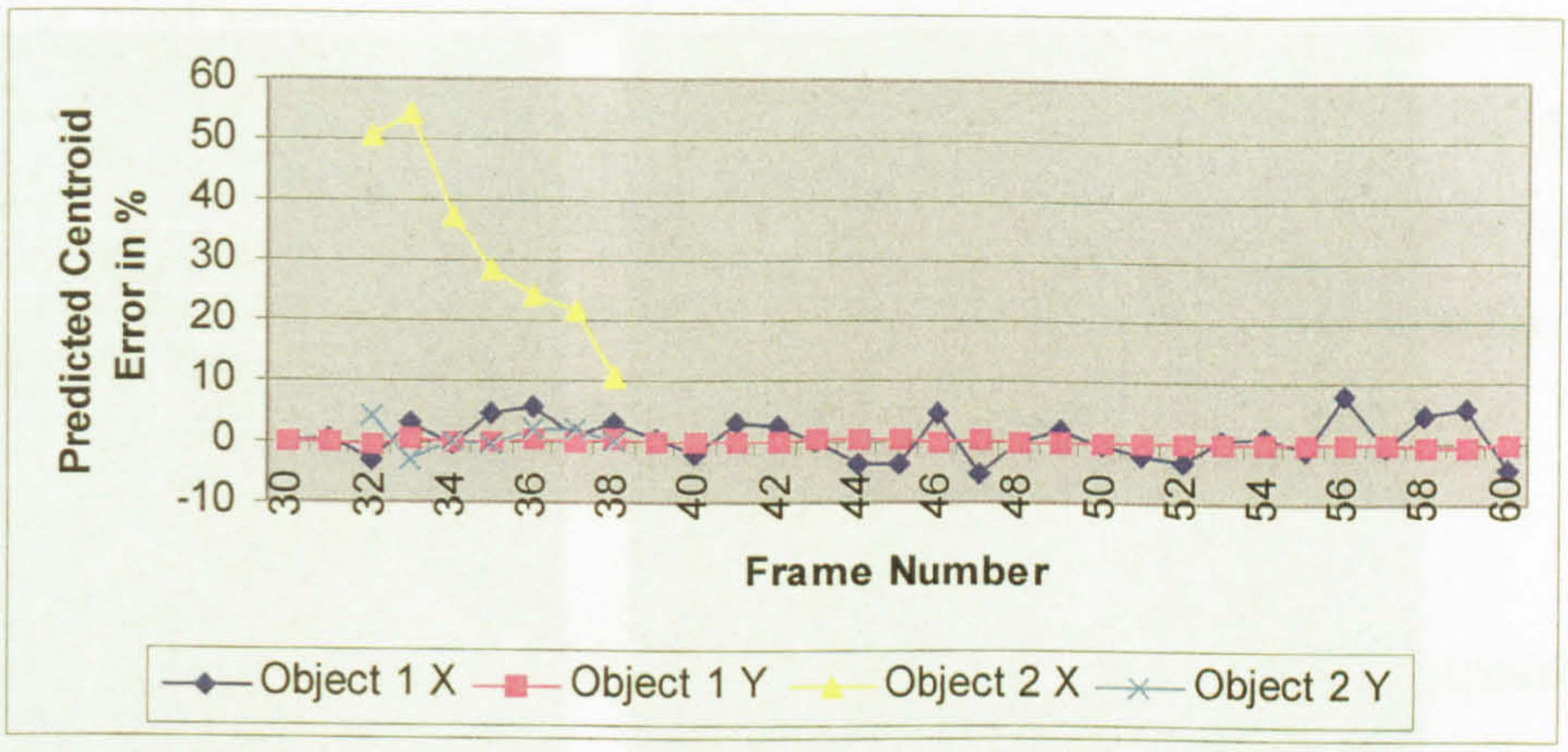

Figure 5.26 – Percentage error between predicted and actual centroid positions for objects in the Poole sequence using the Motion Model method with even weighted line fitting

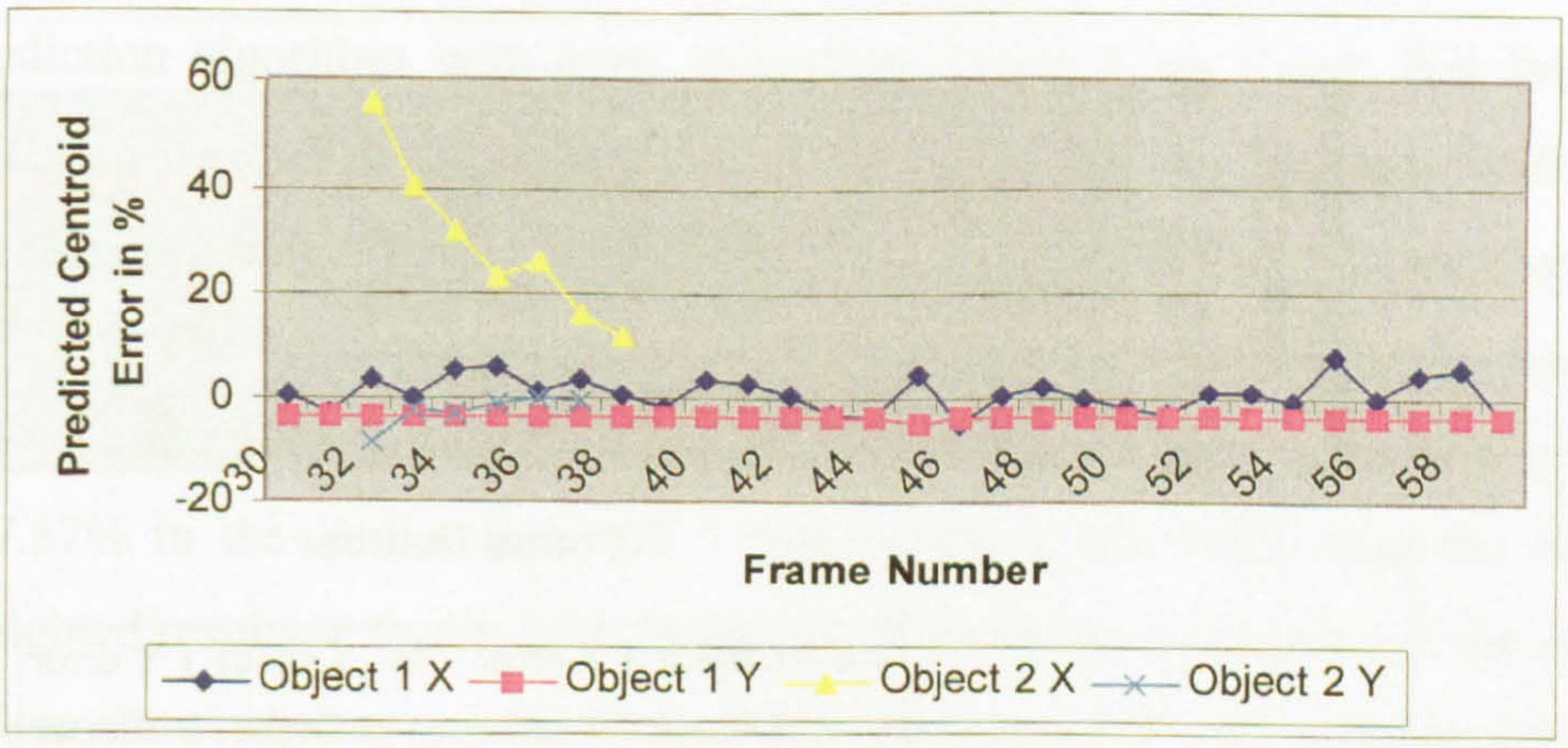

Figure 5.27 – Percentage error between predicted and actual centroid positions for objects in the Poole sequence using the Motion Model method with time weighting line fitting

The result of the Kalman filter predicted tracking is shown, for frames 16 and 26 of the Poole sequence, in figures 5.28 and 5.29. As with the Motion Model method it has correctly predicted that the fast moving object will move much further compared to the large dark object. However, the predicted centroid position is noticeably different to that of the Motion Model method in figures 5.26 and 5.27. In frame 16 the Kalman filter method has predicted that the object will move a greater distance in the x direction and further down in the y direction in frame 26.

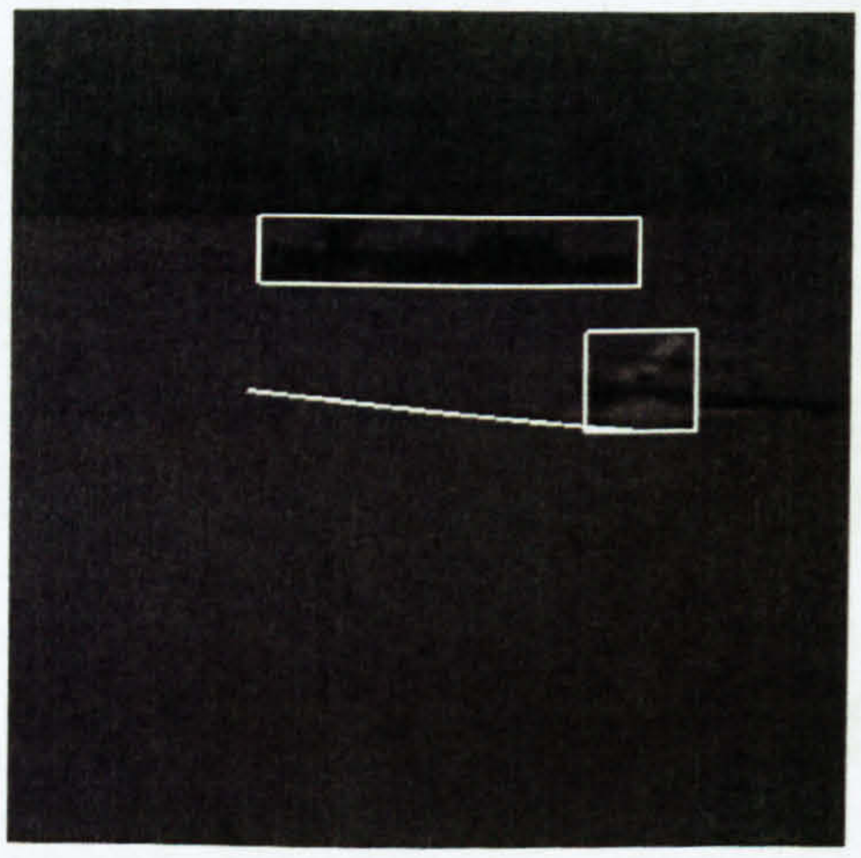

Figure 5.28 – Kalman filter predicted tracking, Poole frame 16

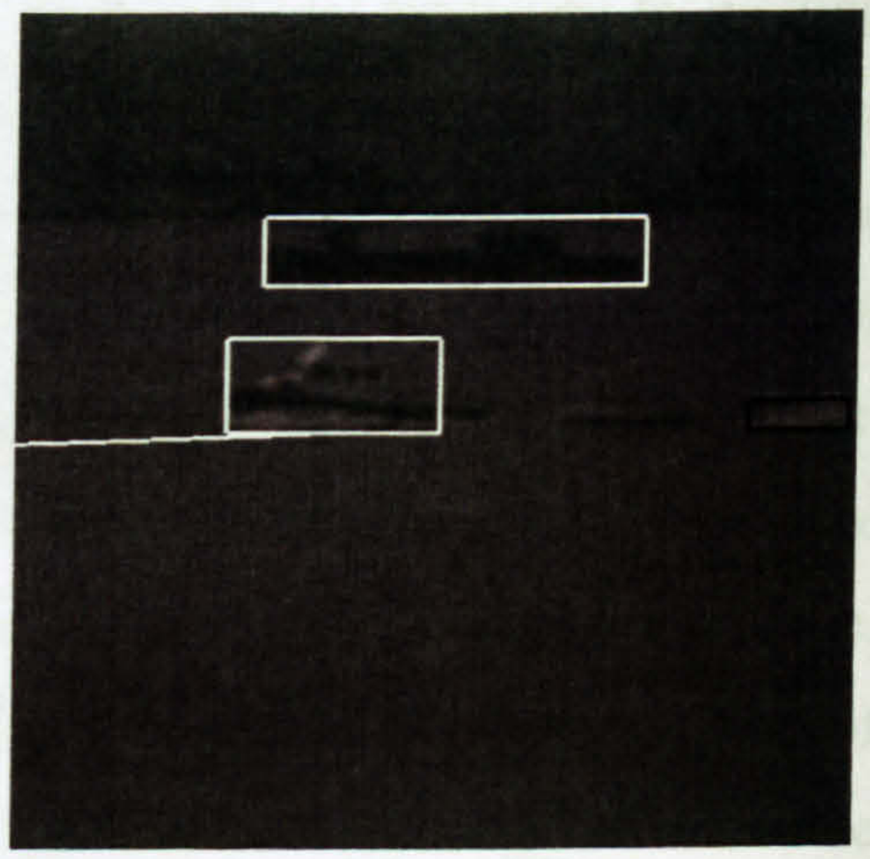

Figure 5.29 – Kalman filter predicted tracking, Poole frame 26

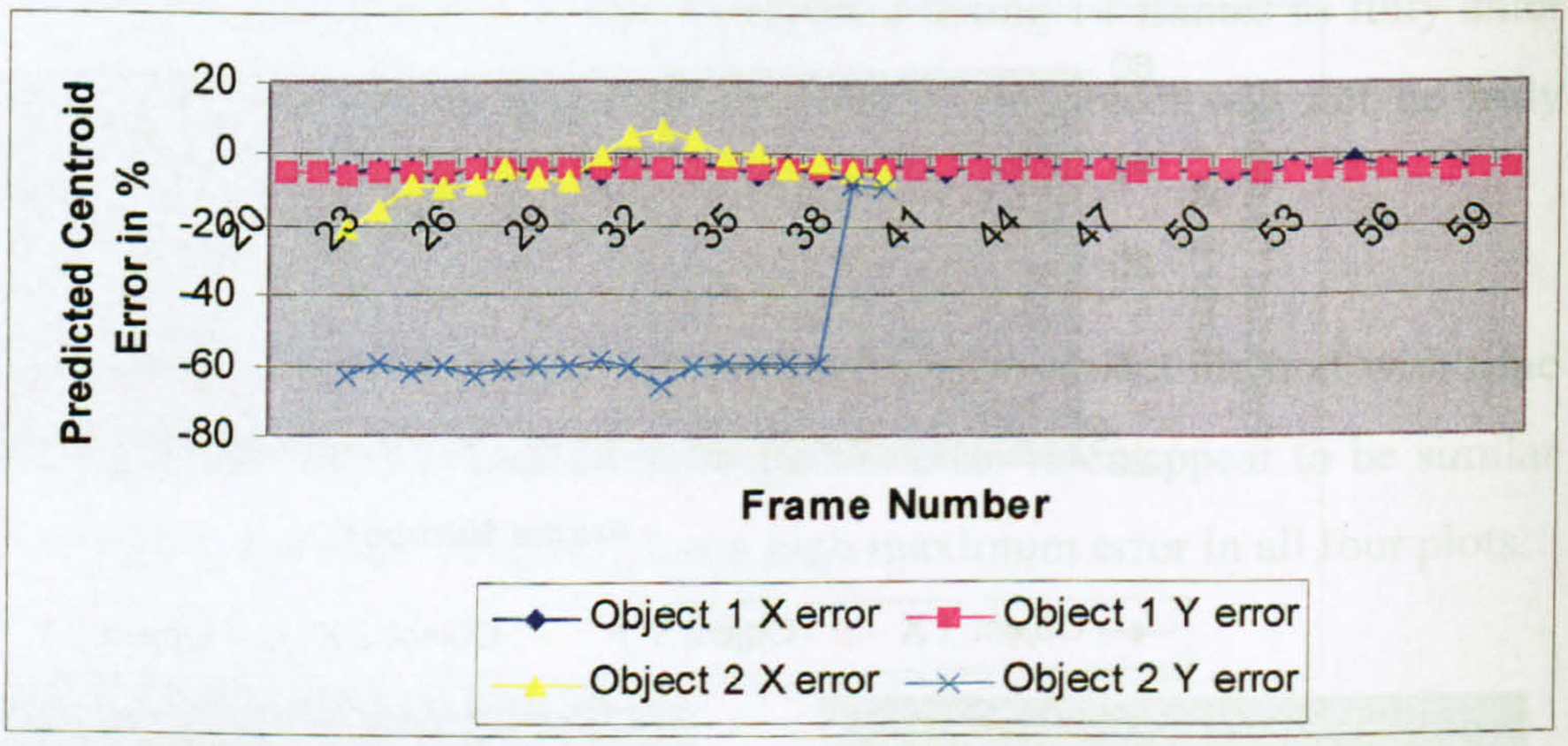

Figure 5.30 – Percentage error between predicted and actual centroid positions for objects in the Poole sequence using the Kalman filter method

Plotting the percentage error between the predicted centroid position of the object and the manually derived actual centroid of the object in the future frame, figure 5.30 and can be directly compared with figure 5.26 as it shows the accuracy of the Kalman filter prediction method. The large dark object has a maximum X direction error of 7.2% and a maximum error of 5.7% in the Y direction. The fast moving object has a maximum error of 21.1% in the X direction and 65.6% in the Y direction. The error rate for both X and Y direction of the fast moving object, object 2, is higher than that for the large dark object, object 1. This is due to the prediction between two frames

being used repeatedly to predict the position twenty frames later. In the absence of measurements taken from the interim frames to update the filter any error in the initial prediction is compounded leading to the higher error rate. Whilst this has also taken place for object 1 the effect is greater for object 2 as the velocity of this object is higher.

5.4.4 Motion Prediction Results, Portsmouth Sequence

The predictive tracking results for the Portsmouth sequence are shown for the Motion Model method in figures 5.31 and 5.32 and for the Kalman filter method in figures 5.35 and 5.36. The accuracy of the Motion Model prediction algorithm with even weighting, figure 5.33, shows that the Y direction prediction is more successful than the X direction with a percentage error of 3.3% or less while the X direction has a maximum error of 7.8%. The Motion Model method with time weighted line fitting, figure 5.31, shows both directions for object 1 having a higher maximum error (17.57% in the X direction and 5.86% in the Y direction) than the even weighted results in figure 5.33. In fact the Y direction for object 2 is the only one to show a lower maximum error rate. In figure 5.37, the accuracy of the Kalman filter method shows a similar error rate for all four parameters. The maximum error is the X direction for object 2 at 5.2% whilst the minimum error is that of the X direction of object 1 at 4.2%.

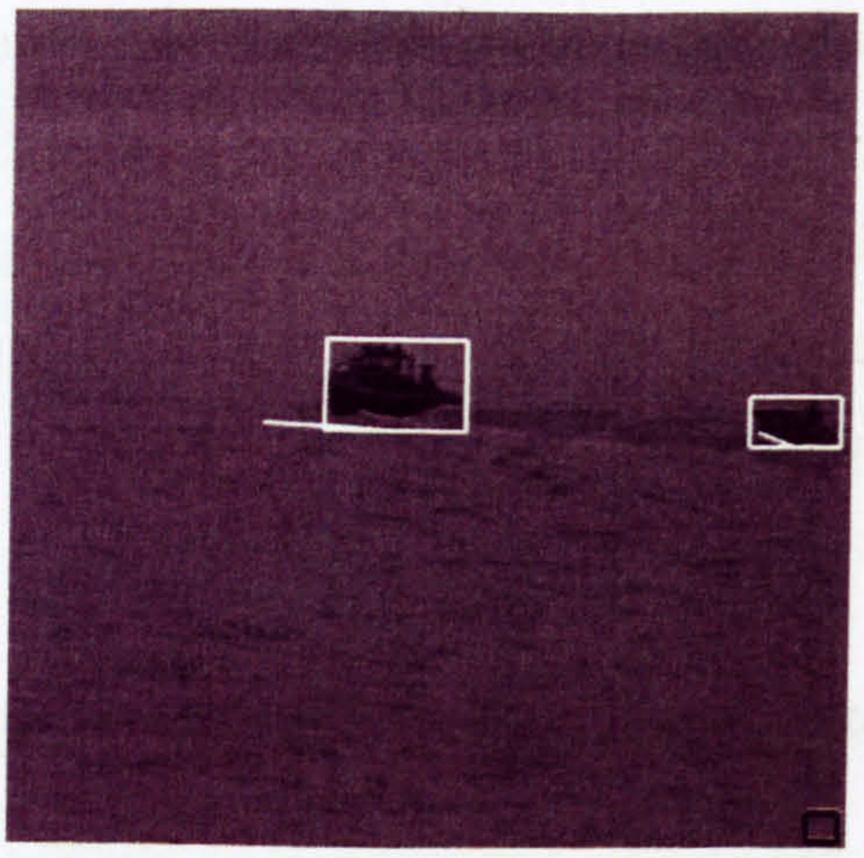

Figure 5.31 – Motion Model method
Portsmouth Sequence, frame 16

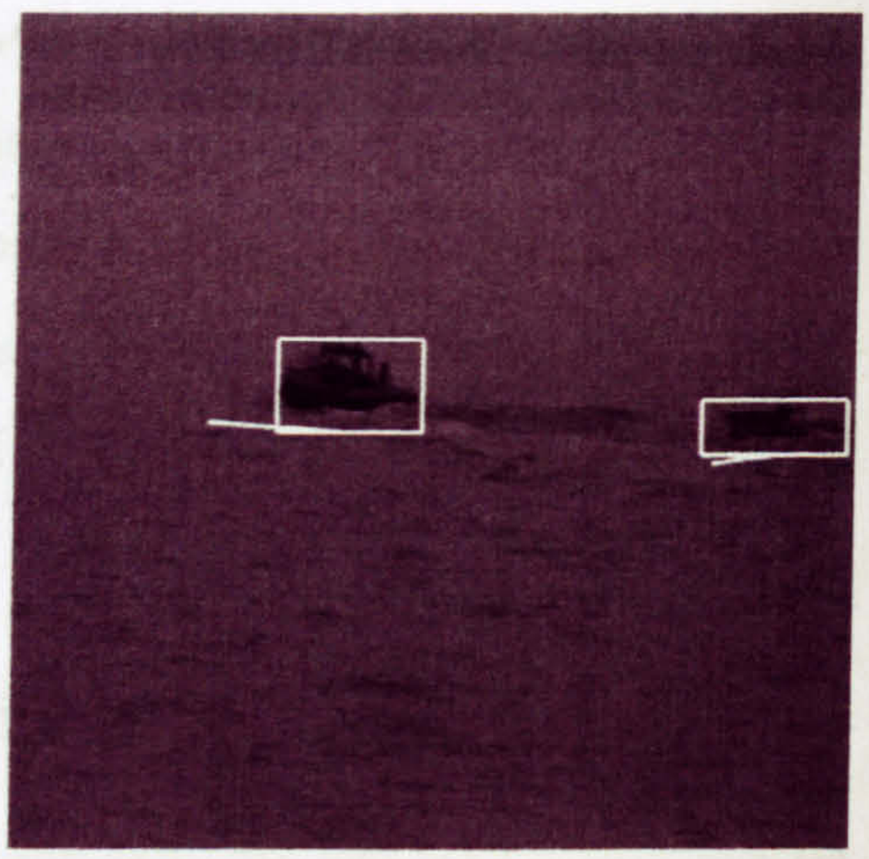

Figure 5.32 – Motion Model method
Portsmouth sequence, frame 26

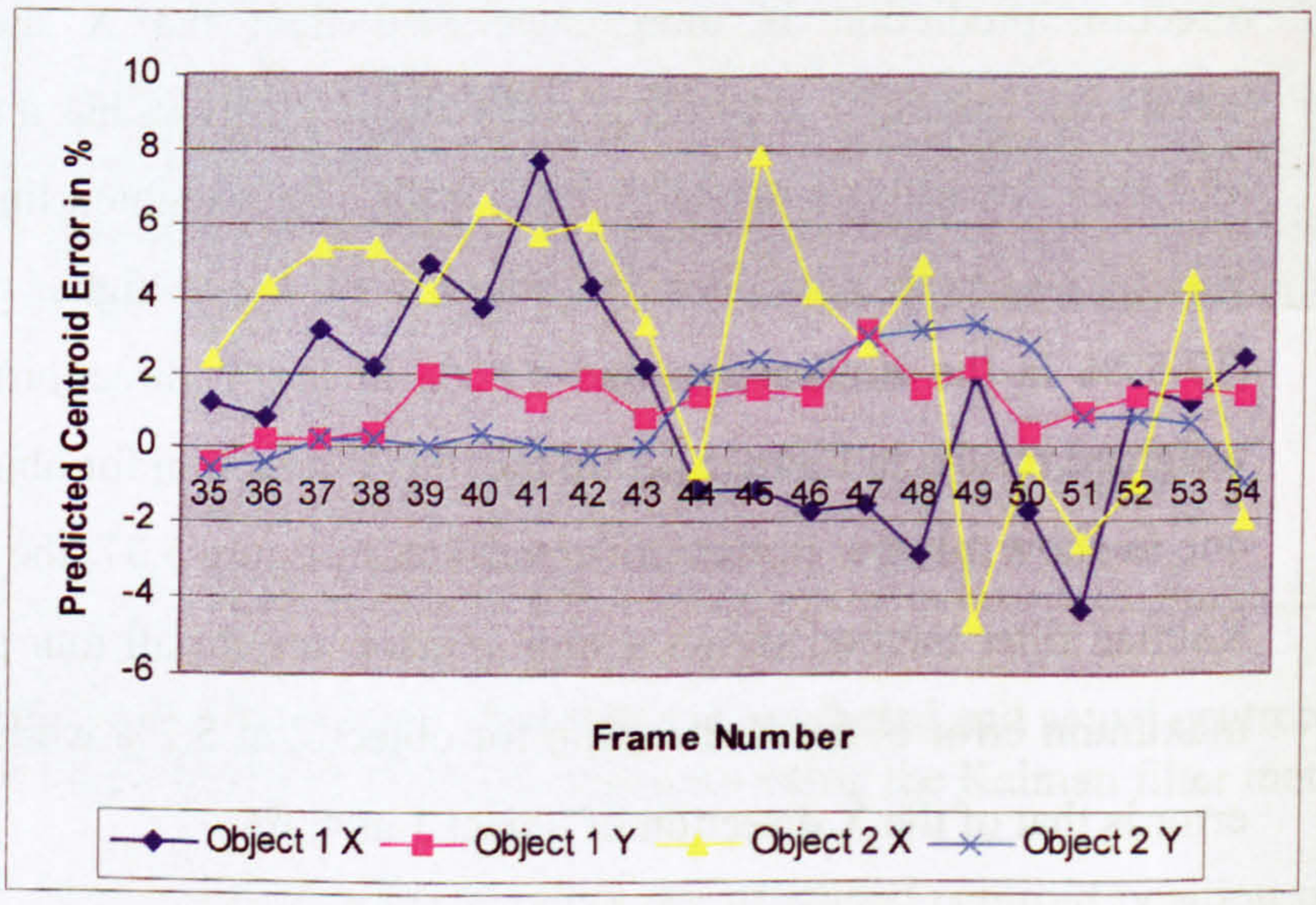

Figure 5.33 – Percentage error between predicted and actual centroid positions for objects in the Portsmouth sequence using the Motion Model method with even weighted line fitting

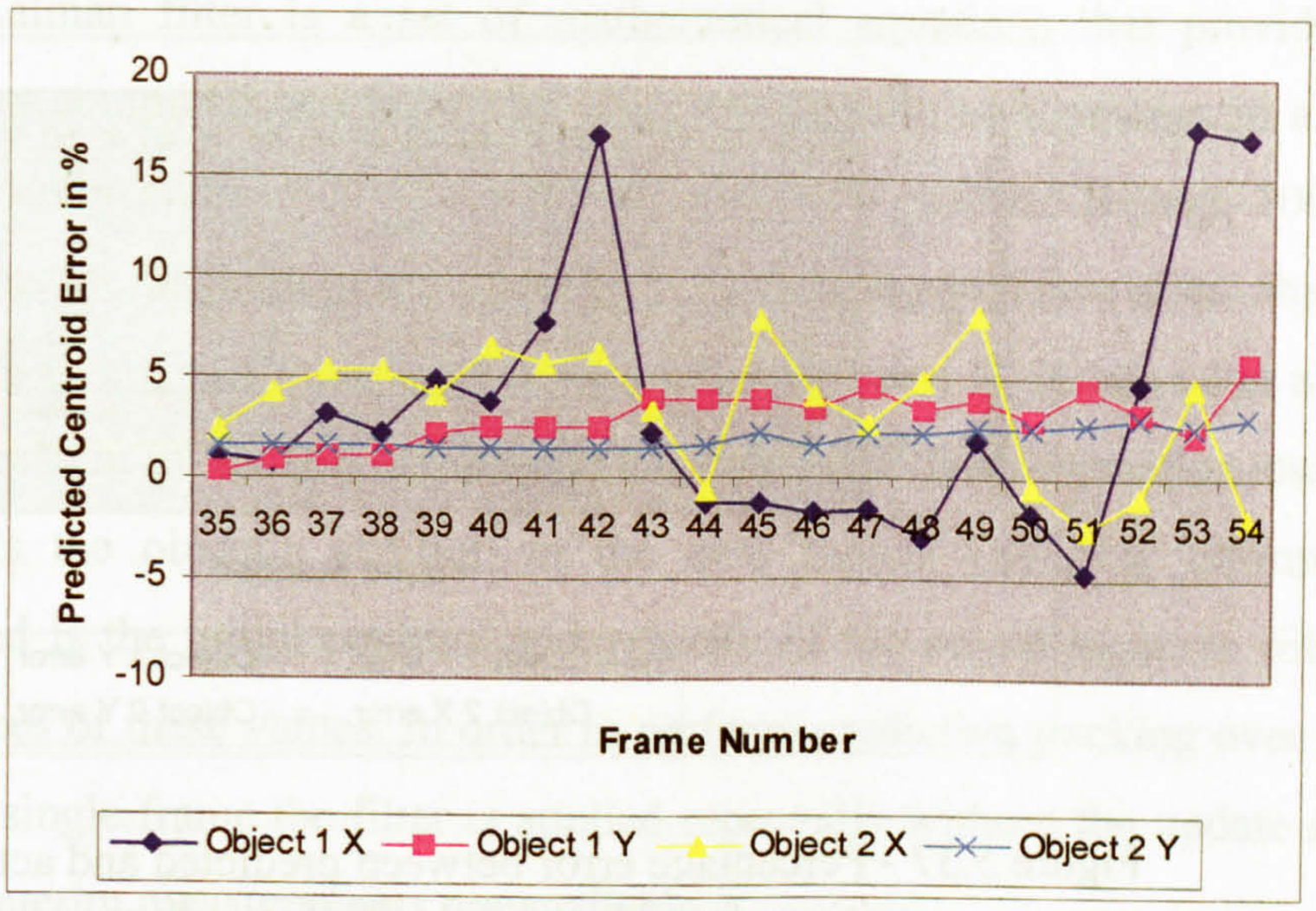

Figure 5.34 – Percentage error between predicted and actual centroid positions for objects in the Portsmouth sequence using the Motion Model method with time weighted line fitting

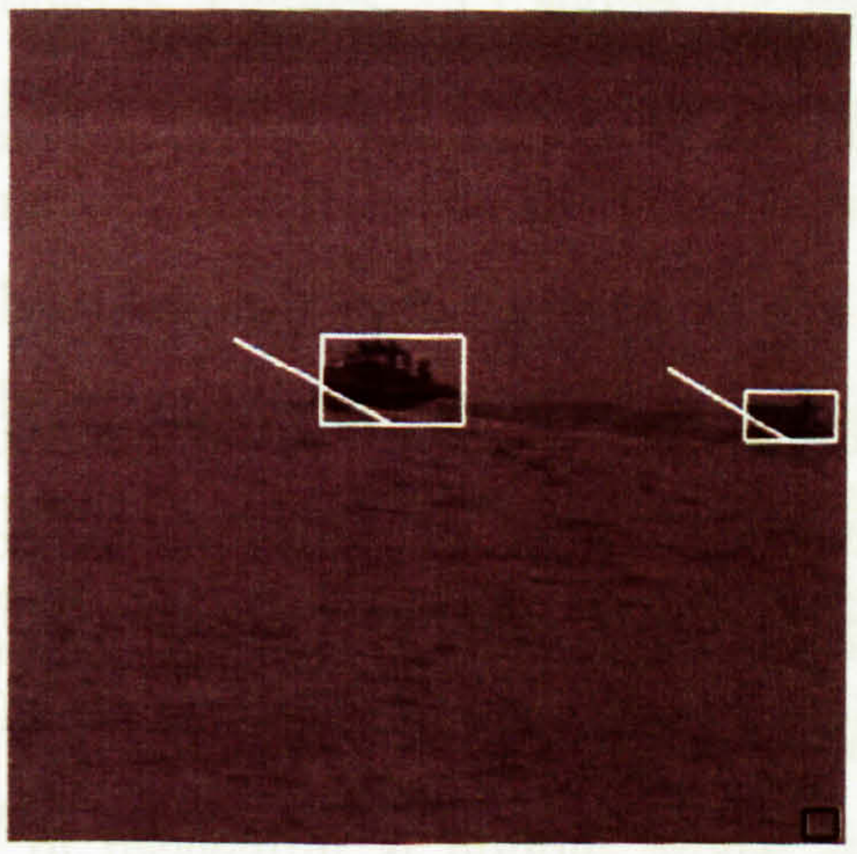

Figure 5.35 – Kalman filter method Portsmouth sequence, frame 16

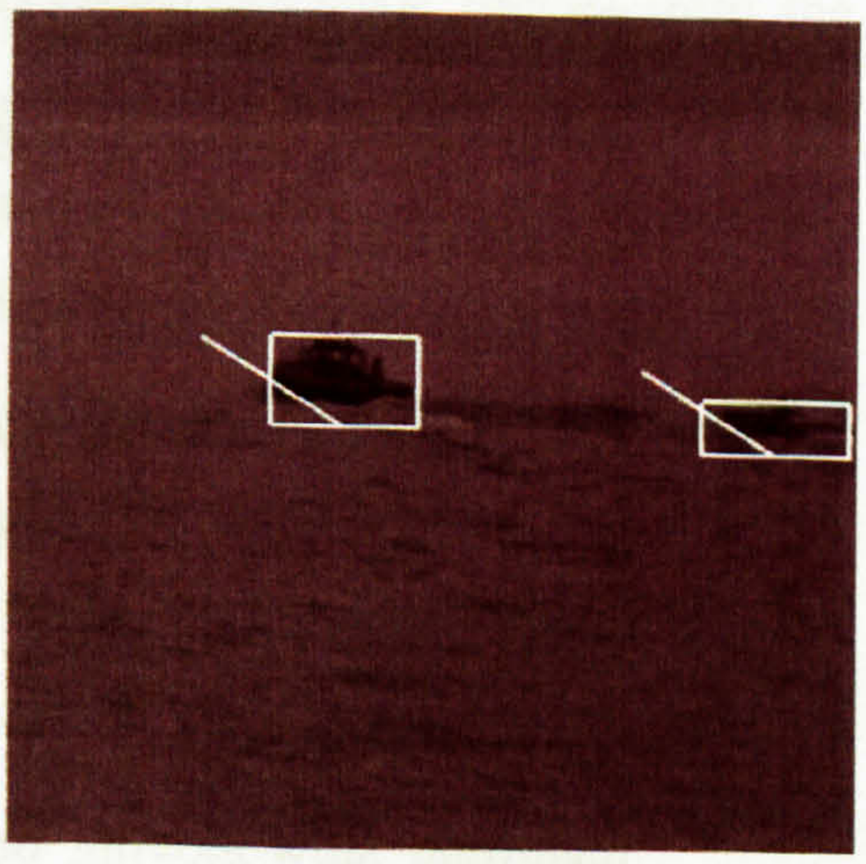

Figure 5.36 – Kalman filter method Portsmouth sequence, frame 26

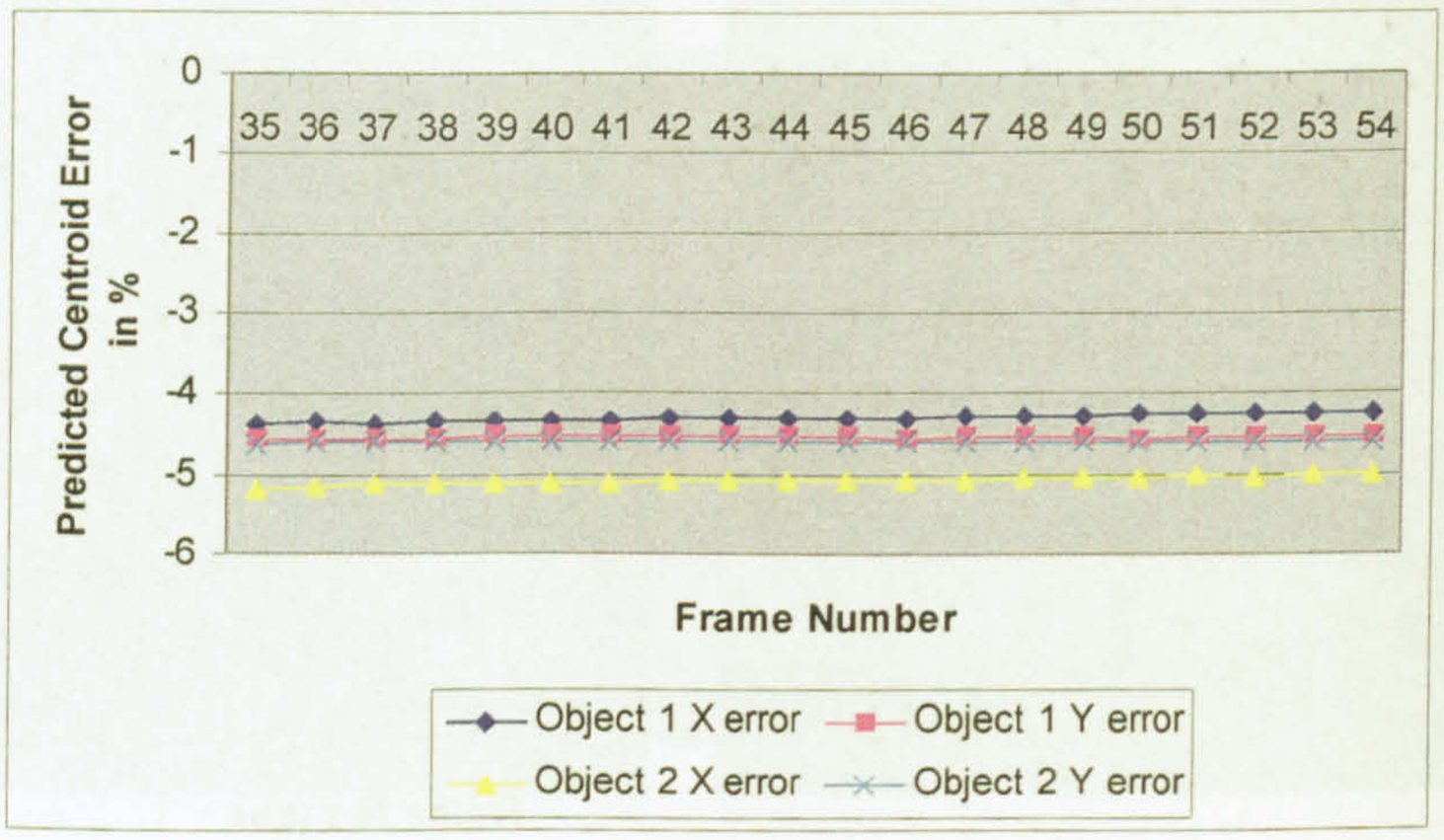

Figure 5.37 - Percentage error between predicted and actual centroid positions for objects in the Portsmouth sequence using the Kalman filter method

5.5 Discussion and Summary

Two methods of tracking have been evaluated for use with the maritime images being used in this system, the Motion Model method and the Kalman filter method.

The maritime motion model contains a set of rules determining the allowable motion of a valid maritime object. The rules include constraints on the speed of an object, which is assumed to be constant, the orientation change an object can go through between frames, and the specific rules regarding slow-moving objects whose motion is influenced to a greater extent by the movement of the sea than by its own movement.

A measure of confidence is used to determine which of the segmented objects in the current frame is the best match for an object in the previous frame. The object with the highest level of confidence is considered to be the same object and the feature vector for that object is updated and stored in the object vector that is carried forward to the next frame.

The Kalman filter is a set of mathematical equations that provides an efficient computational means to estimate the state of a process in a form that resembles a predictor-corrector algorithm (Welch and Bishop, 2004). It performs the tracking of an object by predicting the position of an object in frame N+1 given the measured position of it in frame N. It then takes a fresh measurement in frame N+1, updates the filter with this information and then predicts the object's position in the next frame. The only information required is the initial position and velocity of the object together with the variances of these values. In order to perform predictive tracking over more than a single frame the filter is applied repeatedly without the update stage, as no interim measurements are available.

Two sequences were used to test the algorithms developed here. The first of these, the Poole sequence, contained a fast moving object and a stationary object. The second, the Portsmouth sequence, contains two slow moving objects. Figures 5.13 shows the tracked objects in three frames from the Poole sequence. Figure 5.13a shows the results at the point when the fast moving object enters the scene from the right. At this point the segmented blob has only been matched on two occasions so the system considers it to be a potential object and has identified it with a black box. In the second frame the object has been confirmed by the placing of a white box but only half the object has been enclosed. This is due to the similarity of the grey levels contained in the rear of the object to those in the characterisation of the sea.

The segmentation process has allowed the front of the object to be classified as object and the rear of the object has as sea. This has been shown not to be restrictive in the performance of the tracking process as the third frame, figure 5.13c, shows that the object has continued to be tracked further in the sequence. The graphs in figure 5.14 show the accuracy of the tracking process in matching objects between frames by plotting the manually

recorded 'actual' centroid values against the centroid of the matched objects. Figures 5.15 shows these results as a percentage error. The error rate of 7% for frames 15 and 20 in figure 5.15 from the Poole test sequence are due to the error in the segmentation of the fast moving object due to the similarity of grey levels already mentioned.

The results outlined above are true for both the Motion Model and Kalman filter tracking methods. This is because the two methods are only used to match objects between frames. The determination of object parameters (centroid, x and y co-ordinates, area), and the drawing of black or white identification boxes is the same for each method. In order to distinguish between the two methods the number of false objects identified and tracked has to be considered. Figure 5.17 shows the number of false objects tracked by the Motion Model method and figure 5.18 the false objects tracked by the Kalman filter method. The Kalman filter method tracks (in 5 or more frames) a total of 54 false objects compared to a total of 30 false objects with the Motion Model method. When considering the number of frames in which false objects were tracked the Kalman filter method tracked false objects in 35 out of the 60 frames whilst the Motion Model method tracked false objects in 23 out of the 60 frames.

The results for the Portsmouth sequence are the same as those for the Poole sequence, the two methods yield the same successful tracking of real objects as shown in figures 5.19 – 5.21. It can be seen from each of the frames from the Portsmouth test sequence, figure 5.19, that the objects have been tracked successfully by the placing around them of a white box. This indicates that the object has been matched over a minimum of five frames to confirm it as a valid object. The graphs in figure 5.20 show the accuracy of the tracking process for this sequence by plotting the manually recorded 'actual' centroid values against the centroid of the matched objects. Figures 5.21 show these

results as a percentage error. The maximum error in the Portsmouth test sequence is shown to be 2.5%.

Again it is the number of false objects tracked that distinguishes the two methods apart. Figures 5.22 and 5.23 show the number of false objects tracked by the two methods through the Portsmouth sequence. In this sequence the Motion Model method tracks a total of 6 false objects over 6 of the 60 frames whereas the Kalman filter method tracks a total of 29 false objects over 24 of the 60 frames. The Motion Model method is the preferred tracker having shown to identify and track less false objects than the Kalman filter method whilst successfully tracking all the real objects present in the image sequence.

Figures 5.7 and 5.8 show the assumption made by Wang et al (2000) that object trajectories over a few frames approximate a straight line to be true for both the Poole and Portsmouth test sequences. For each matched object a centroid data array is filled with the centroid data from each matched frame. The data held in this array is used to predict the future motion of the object using the least squares method of calculating the line of best fit through the set of data. Two variants of this method have been applied, the first with even weighting and the second with time weighting to decrease the confidence in the old measurements in the data array.

This method is shown to be successful in figures 5.33 and 5.34 where the error between the predicted centroid positions and the actual centroid position of the objects in the Portsmouth test sequence is plotted. The maximum error of 8% for the method with even weighting is encouraging and shows the validity of the approach. The maximum error for the time weighted method is 17.57%. The same result for the Poole sequence is shown in figure 5.26. The stationary object has a maximum error of 7.8% in the X direction and 0.78% in the Y direction. The fast moving object,

however, shows a maximum error in the X direction of 54% and 4.4% in the Y direction. Whilst the error of 54% is initially disappointing it should be noted that within seven frames this error rate has dropped to 10.3%. The error is due to the object entering the scene. This has resulted in the centroid of the object moving very slowly leading to a prediction that the centroid will move slowly.

The results of repeatedly applying the Kalman filter to obtain the predicted position of an object in twenty frames time is shown in figures 5.28 – 5.30 for the Poole sequence and figures 5.35 – 5.37 for the Portsmouth sequence. The maximum error from the Portsmouth sequence was 5.2% and the minimum error was 4.2%. For the Poole sequence the maximum error was 65.6% for the Y direction of the fast moving object, object 2, and 21.1% for the X direction. For object 1, the maximum error in the X direction was 7.2% and 5.7% in the Y direction. The large error for object 2 is due to a combination of object 2 being fast moving and the repeated application of the Kalman filter without having the benefit of any measurements to update the filter with. As the object has a large velocity any error in the initial prediction will be accumulated over the twenty frame prediction. This is particularly evident in the Y direction with an error of over 60%. The X direction has suffered too, although to a much lesser extent with a maximum error of 21.1%. The slow moving object and both of the slow moving objects in the Portsmouth sequence show how the error is much reduced due to the lower speed of the objects. The highest error of the three slow moving objects is just 7.2%.

Given that the Motion Model method performed better in frame to frame tracking and that the Kalman filter is more computationally intensive, particularly in the predictive tracking element where the filter has to be applied twenty times per object, the Motion Model method of tracking and prediction using even weighting is chosen for use with this system.

Chapter 6: Collision Risk Estimation

6.1 Introduction

Image understanding is the highest level in the machine vision processing classification and one of the most complex challenges in artificial intelligence (Sonka et al, 1993). Approaches used in the literature include the use of Semantic Networks (Lou et al, 2002), (Fraile and Maybank, 1998), (Shih and Huang, 2003), and Neural Networks (Di Bona et al, 2001) , (Messer and Kittler, 1998). The main problem encountered with semantic interpretation according to Lou et al (2002) is constructing a mapping from images into the human conceptual space. Shih and Huang (2003) note that the exploitation of semantic information is difficult because of the large difference in representations and levels of knowledge present in images.

Neural network techniques require a learning process to be gone through before any target data is processed. A typical feed-forward network usually requiring many thousands of iterations through a training set (Sonka et al, 1993). Castleman (1996) lists the disadvantages of neural networks when compared to statistical approaches as 1, the extensive amount of training required, 2, slower operation when implemented as a simulation on a conventional computer and 3, the unavailability of a detailed understanding of the decision-making process that is being used. Another approach to image understanding is used by Matesin et al (2001), that of a rule-based expert system, which uses a combination of known information regarding

the subject matter and information, gained from processing the images. Gonzalez and Woods (1992) describe expert systems as offering flexibility of response through the use of a knowledge base containing procedural, visual, and world knowledge and a series of rules typically of the **if** (conditions) **then** (actions) form.

In this collision risk estimation chapter the question posed in chapter 1 is addressed. Is anything in the way? The feature vectors from tracked objects are used together with a high-level reasoning algorithm based on a set of rules to estimate the likelihood of each object colliding with the vessel the camera is mounted on. The high-level reasoning algorithm has been developed based on rules regarding the vessel the camera is mounted on, the maritime environment and the motion model developed previously to estimate the threat an object is posing to the vessel the camera is mounted on. An estimation algorithm has been developed whereby the threat posed by each object is estimated to be a function of the distance the centroid of the object is with respect to the point of collision. The threat is displayed to the user as a percentage risk of collision. The higher the percentage, the higher the risk of collision and the lower the value, the lower the risk of collision. The predicted future position of the object also has its collision risk estimated and displayed to the user.

6.2 Reasoning Strategy

The set of rules and reasoning strategy have been developed using the information known or that can be inferred about:

- The vessel the camera is mounted on and the mounting of the camera allowing inferences to be made about the camera platform and where the highest risk of a collision is likely to be.

- The environment in which the scene is taking place used to reinforce the inferences of where the highest risk of collision is likely to be found.
- The maritime objects using the motion model developed in chapter 5 providing information regarding the behaviour of the objects expected to be found in the images.

6.2.1 The Vessel and Camera

The primary research aim of this work is that of maritime collision avoidance. This is achieved through the monitoring of an area of sea directly ahead of the vessel. This aim leads to the inference that the camera will be mounted on a vessel such that it is looking forward out in front of the vessel.

For efficiency and smoothness of movement maritime vessels are designed so that the bow of the craft approaches a point as shown in figure 6.1. It can be assumed that the first point of collision when the vessel is travelling forward and therefore the point presenting the greatest risk of a collision will be the bow of the vessel. If the camera is mounted to point directly out in front of the vessel but does not include any of that vessel in the image, that initial point of impact will be the centre bottom of the camera image.

Figure 6.1 – QE2 showing pointed bow

It is assumed that the camera will be mounted upright such that if the camera platform were stationary an object moving from left to right across the image at the same height in the image remains the same distance from the camera.

6.2.2 The Maritime Environment

The maritime environment is one that is, due to the nature of water, essentially flat. Given that the camera is forward looking, perspective projection tells us that the higher the object appears in the image, the greater the distance between the camera and the object. If an object is further away from the camera there is more time for either the object or the vessel the camera is mounted on to move and therefore the risk of collision is less.

6.2.3 The Maritime Objects

Knowing that the point of contact is the centre bottom of the image and that height in the image represents depth and therefore distance away from the camera the following can be stated:

- Objects moving down the image (approaching the camera) will pose an increasing risk of collision.

- Objects moving up the image (away from the camera) will pose a reducing risk of collision

- Objects moving across the scene toward the centre of the image will pose an increasing risk of collision.

- Objects moving across the scene away from the centre of the image will pose a reducing risk of collision.

- All confirmed maritime objects could cause a collision.

- Although an object may pose little threat a collision is always possible.

6.2.4 High-Level Reasoning Rules

Given the known and inferred information in the previous sub-sections the following rules for the collision risk estimation can be stated:

1. the point of impact will be at the centre bottom of the image
2. the centre bottom of the image poses the greatest risk of a collision
3. the risk of collision will reduce the further up the image the object is located
4. the risk of collision will reduce the further each side of the centre of the image the object is located
5. there is always a risk of collision so the minimum threat cannot be 0%
6. there is always a chance that collision can be avoided so the maximum threat cannot be 100%

6.3 Implementation of Reasoning Strategy

This collision risk estimation is carried out for the current centroid position of each object and also for its predicted future centroid position and these are displayed to the user. In this way the user can see whether an object's risk is likely to increase or decrease.

The implementation of the high-level reasoning strategy requires the calculation of the risk an object poses to be based on its position in the image. The centroid location in the X direction is used and the lowest point in the Y direction. This point is used as the point of the object lowest in the image poses the greatest risk. From rules 1 and 2 we know the highest risk will be at the centre bottom of the image. Rules 3 and 4 tell us that the risk diminishes omni-directionally away from the centre bottom to the outside edge of the image. They also tell us that the lowest risk will be around the very edge of the image.

The percentage risk of collision based on the distance from the point of collision to the centroid position of the object is carried out mathematically using the following algorithm:

```
if (Xsquares <= totalYsquares) then
    if (Xsquares >= Ysquares) then
        %risk = Ysquares
    else if (Xsquares < Ysquares) then
        %risk = Xsquares
else if (Xsquares > totalTsquares) then
    %risk = Xsquares – 99
```

where

Xsquares = X centroid / 2.6 (pixels per 1% resolution)

Ysquares = Y lowest point / 5 (pixels per 1% resolution)

totalYsquares = total number of 5 pixel blocks in the Y dimension

6.4 Results

Figures 6.2 and 6.4 show four frames from each of the Portsmouth test sequence and the Dover test sequence after the collision risk estimation has taken place. The percentage risk of collision is written directly under the centroid and predicted future centroid position for each confirmed target in the scene. The percentage risk is taken from the centroid point on the X axis and the point on the Y axis that is nearest the bottom of the image as this is the point closest to the camera platform and therefore presenting the greatest risk of collision.

The collision risk for the objects in the Portsmouth sequence have been manually calculated and plotted against the collision risk estimated by the algorithm developed here. Figure 6.3 shows the percentage error between the manually calculated risk and the risk estimated by the developed algorithm. This shows the maximum error for object 1 is 4% and the maximum error for object 2 is 5%. The average error for object 1 is 2.34% and the average error for object 2 is 1.6%.

Figure 6.5 shows a graph plotting the percentage error between the manually calculated collision risk and the algorithm estimated collision risk for 60 frames from the Dover test sequence. The maximum errors for the three objects present in the sequence are 5% for the harbour wall, 1% for the buoy, and 4% for the boat. The average errors for the objects are 4.34%, 0.18%, and 1.3% respectively.

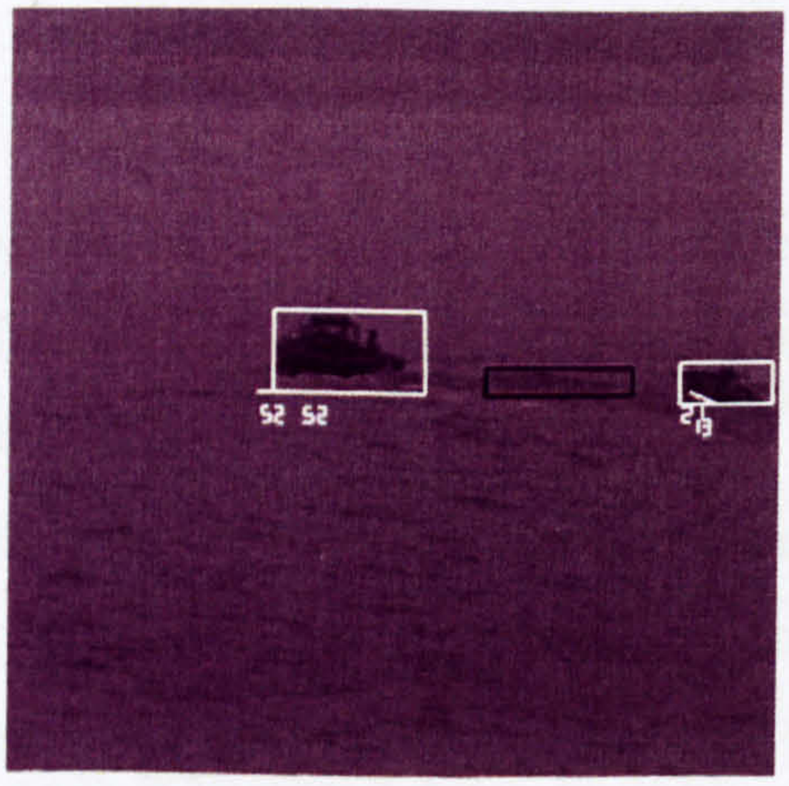

Figure 6.2a – Frame 20 collision risk

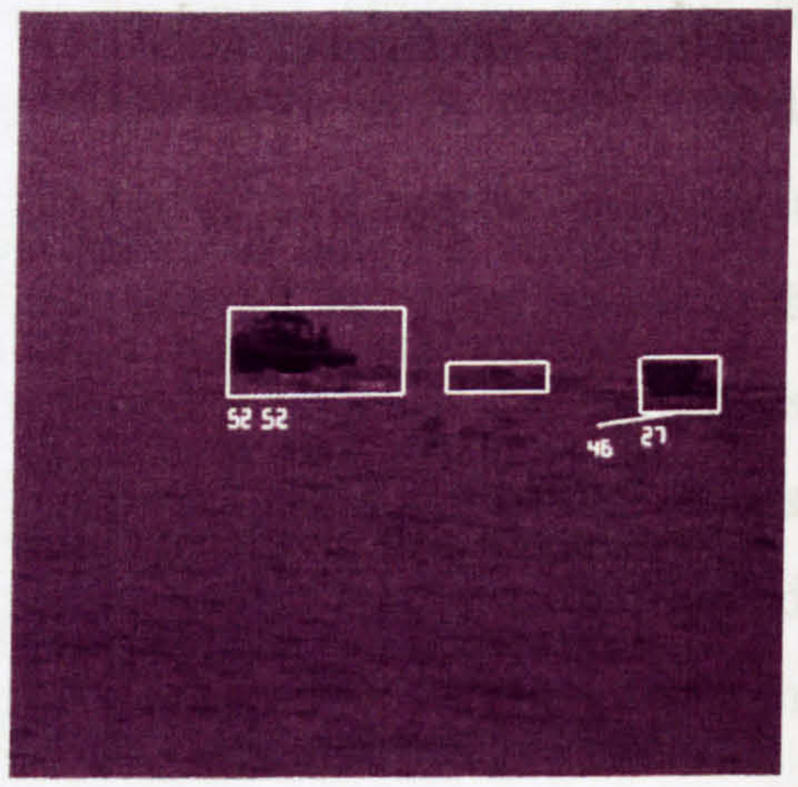

Figure 6.2b – Frame 30 collision risk

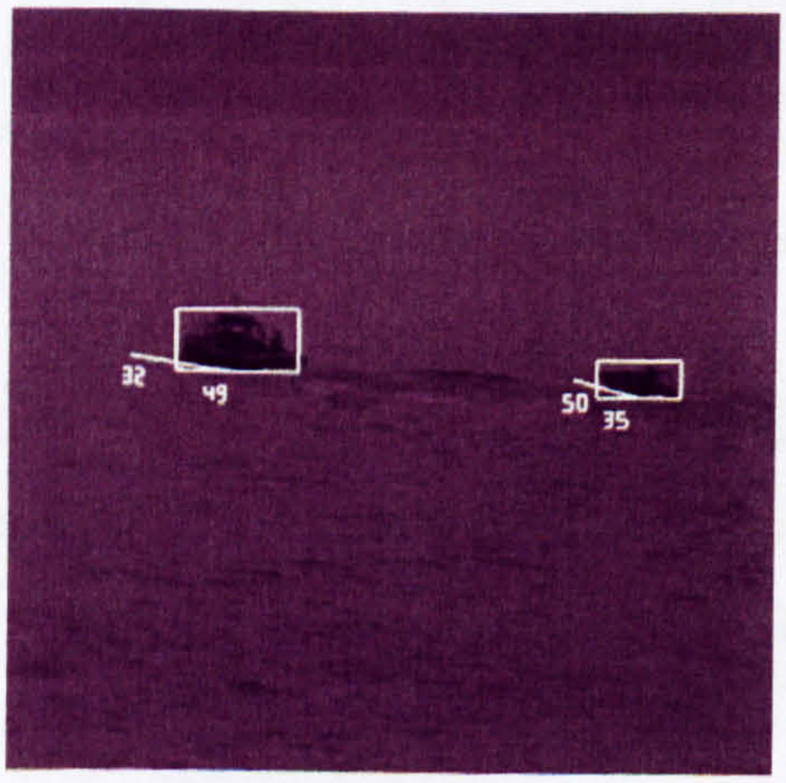

Figure 6.2c – Frame 40 collision risk

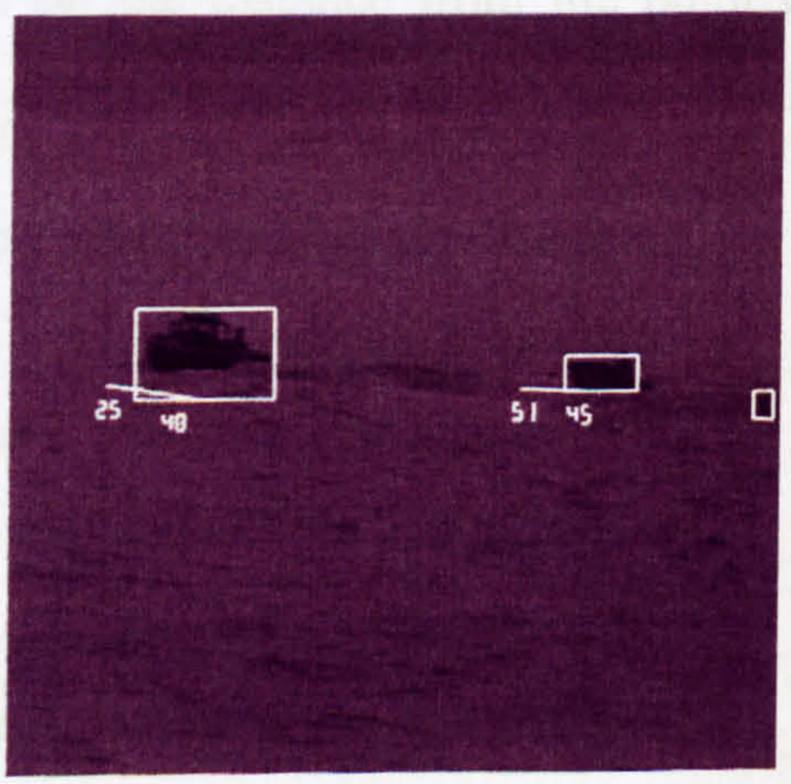

Figure 6.2d – Frame 50 collision risk

Figure 6.2 – 4 frames from the Portsmouth Sequence showing the collision risk posed by each object

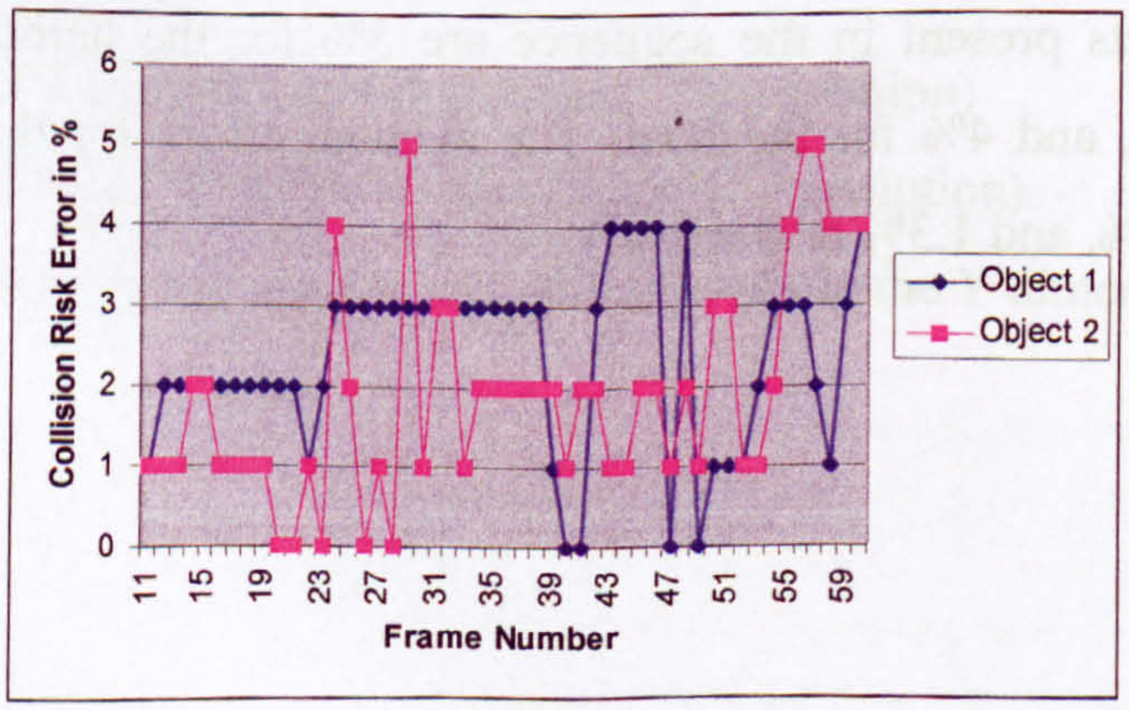

Figure 6.3 – % Error between manual and calculated collision risk for the Portsmouth sequence

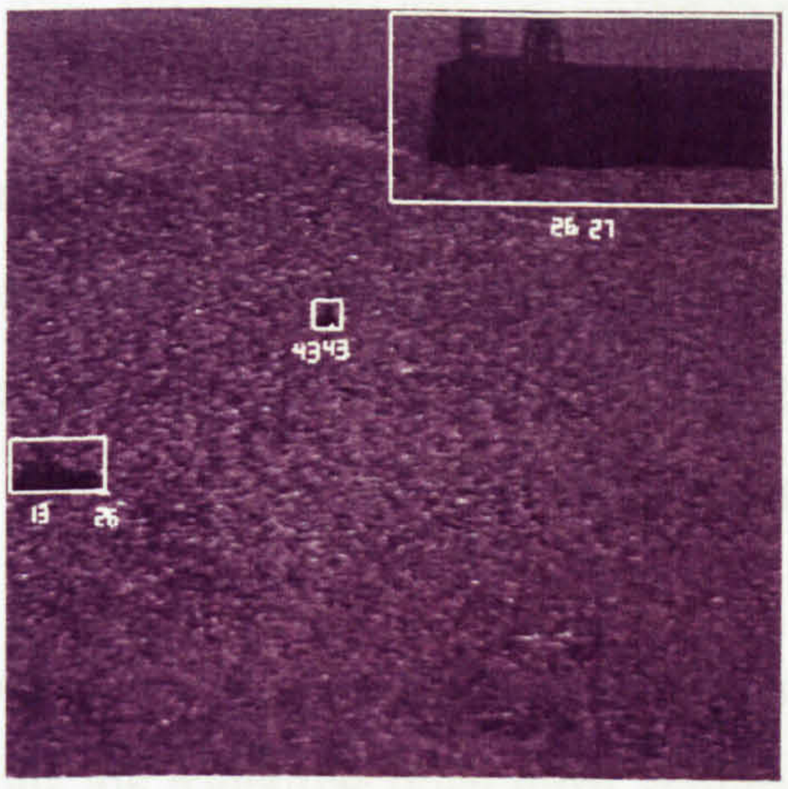

Figure 6.4a – Frame 70 collision risk

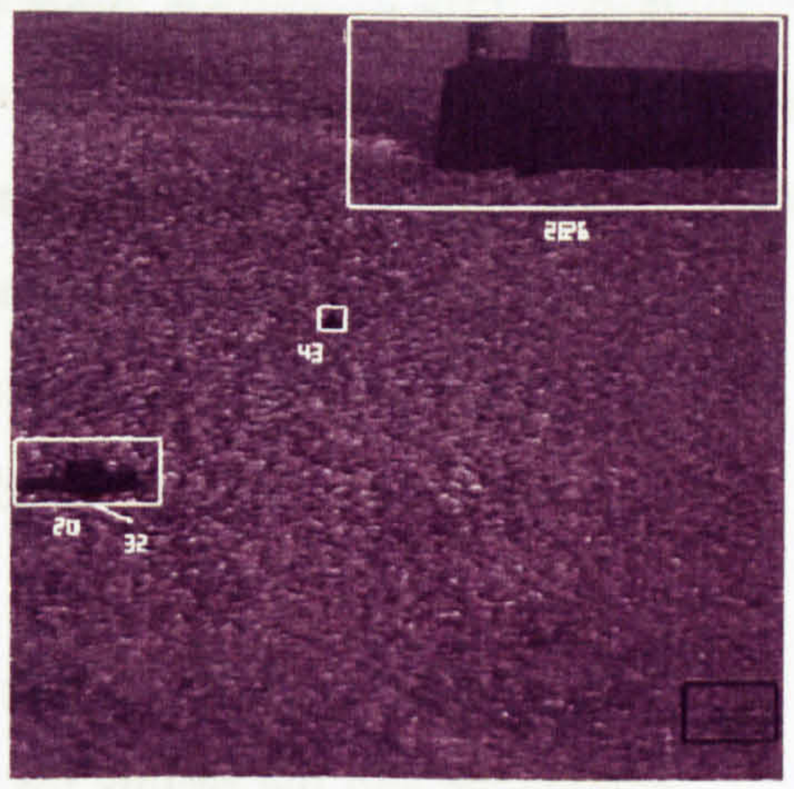

Figure 6.4b – Frame 80 collision risk

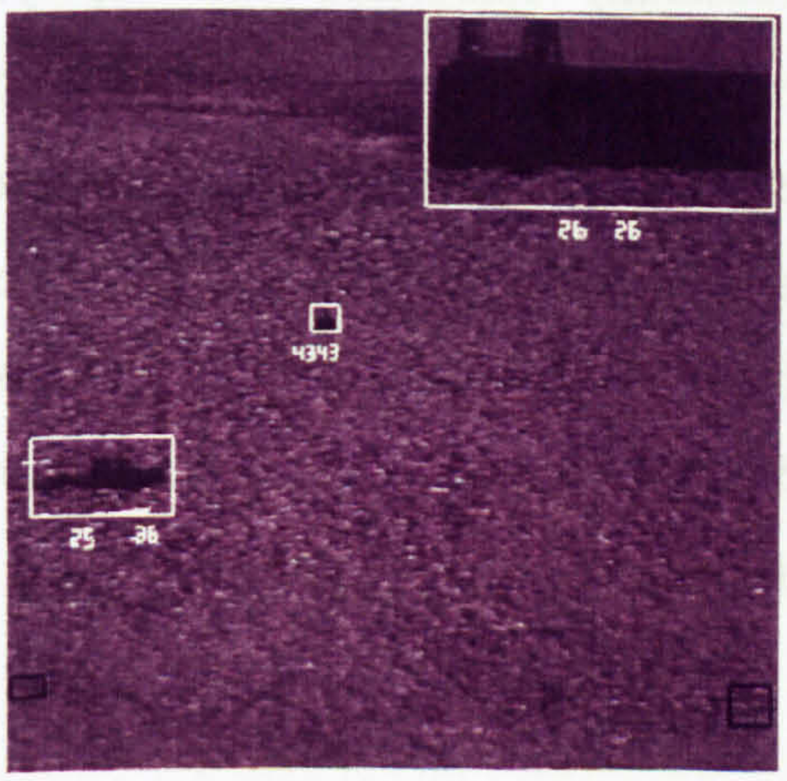

Figure 6.4c – Frame 90 collision risk

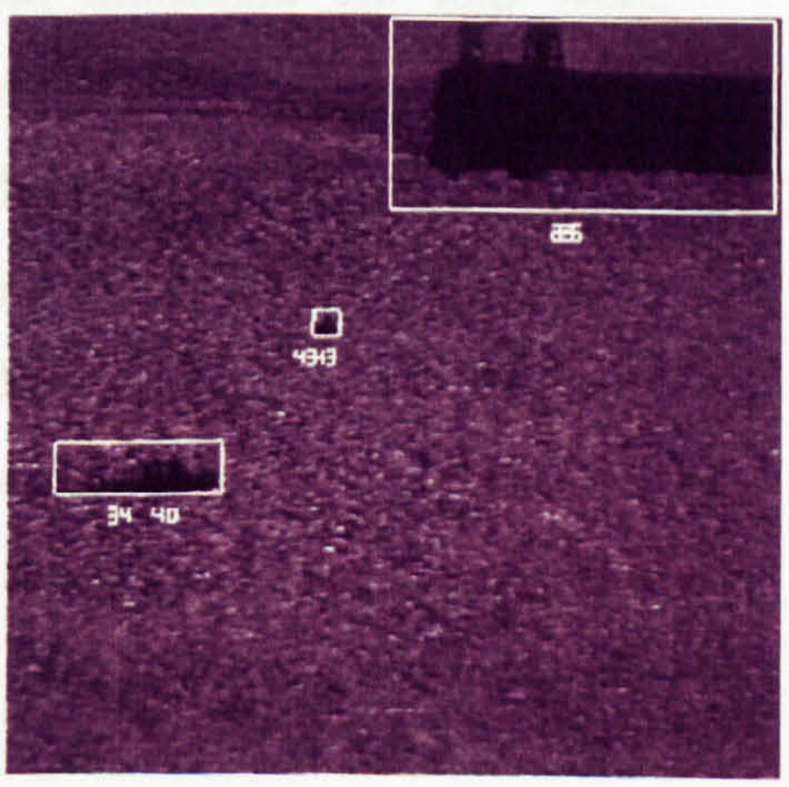

Figure 6.4d – Frame 100 collision risk

Figure 6.4 – 4 frames from the Dover sequence showing the collision risk posed by each object

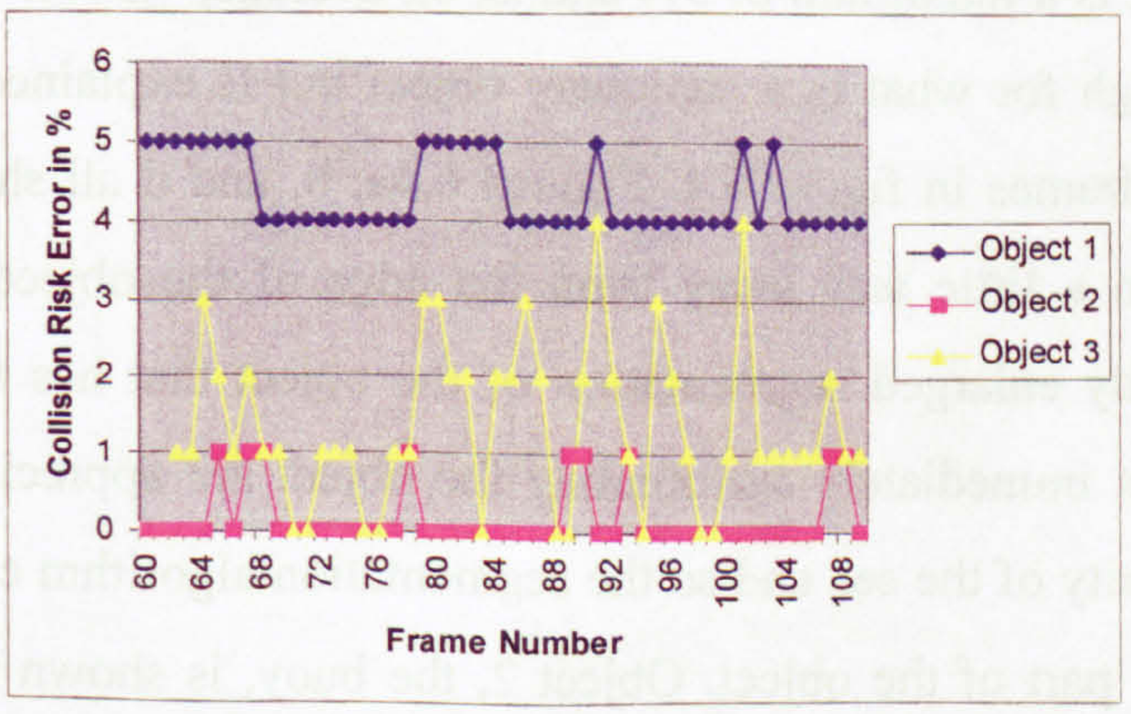

Figure 6.5 - % error between manual and calculated collision risk for the Dover sequence

6.5 Discussion and Summary

A high-level reasoning algorithm has been developed for estimating the percentage risk of collision posed by each object present in the scene. The collision risk has been calculated for both an object's current centroid position and also the predicted future centroid position.

The algorithm has been developed using the Portsmouth and Dover test sequences and the results for four frames from each sequence are shown in figures 6.2 and 6.4. The results from the Portsmouth sequence, figure 6.2, show that the algorithm has successfully calculated and displayed the estimated collision risk for the current and predicted future position for both objects present in the scene. Figure 6.3 shows the absolute percentage error between the collision risk calculated by the algorithm and that calculated manually. The average error for Object 1 is 2.34% and for Object 2 is 1.6% and the maximum error is 4% and 5% respectively. This error rate shows the algorithm has performed well in estimating the risk of collision based on the reasoning strategy developed.

Figure 6.4 shows four frames from the Dover sequence and figure 6.5 shows the percentage error between the manually calculated collision risk and that estimated by the reasoning algorithm. The error for Object 1, the Harbour Wall, is a maximum of 5% and is, on average, 4.34%. This is considered to be high for what is a stationary object but is explained by considering the four frames in figure 6.4. Figures 6.4a, b, and d all show the white box is drawn a little way away from the edge of the object. This is due to the slightly enlarged segmentation of the object that has taken place. The sea pixels immediately surrounding the object are appreciably darker than the majority of the sea and so the segmentation algorithm considers these pixels to be part of the object. Object 2, the buoy, is shown in all four frames of figure 6.4 to have been segmented accurately and this has led to the maximum error of 1%. Object 3, the boat, moves into the scene from the

left. This motion has led to some frames being segmented with the inclusion of sea pixels as in figure 6.4c. The amount of this object enlargement varies between frames and it is this variation in the segmentation of the object that has led to the differences shown in the collision risk results.

Chapter 7: System Evaluation

7.1 Introduction

The test sequences used in the development of the algorithms used in this system were all taken from a static camera position. The aim of this work has been to develop a system to be used on maritime vessels so the main objective of this evaluation is to test the system with sequences taken from a moving vessel.

The sequences used in this evaluation have been taken using a digital camcorder looking forward over the bow of a vehicle ferry travelling between Southampton and Cowes on the Isle of Wight. The image capture parameters are the same as those used for the test sequences (see chapter 1). The key difference between the test and evaluation sequences is the introduction of motion to the camera platform. This is considered to come from three sources: The motion of the vessel; the wind; and the vibration from the vessel's engines.

Both stationary and moving objects are included to evaluate the motion model because of the different types of motion the objects are under. The course of these objects through the scene will test the tracking, motion prediction, and collision risk algorithms. The objects in the evaluation sequences approach the camera, enter the scene and head toward the centre of the image, and head away from the centre of the image in order to fully test these algorithms and the reasoning behind them.

7.2 Evaluation Sequences

This system has been developed using three maritime test sequences, each showing object's in different maritime environments. It is important to show that the system is not scene dependent and that the parameter values and thresholds chosen are not specific to the test sequences but have been arrived at by logical and experimental means to cover the maritime objects and environments that are the subject of this work.

To show this, the system has been evaluated using two previously unseen image sequences. These sequences show maritime objects moving in different maritime environments to those in the test sequences. The objects present in the evaluation sequences are of differing size, shape, and intensity to show that small and light coloured objects can be identified and tracked as well as larger and darker objects.

7.2.1 Sequence 1

In this 300 frame sequence the camera is mounted on a vehicle ferry that is in dock being loaded. Although the ferry is not under way the engines are running and the camera is being subjected to both wind and vessel motion from its raised and exposed position. A motor boat approximately 25 feet in length can be seen moving from the top left of the image to the bottom right across in front of the camera. The illumination in this sequence is very bright causing some highlights to appear on the water. The object is light coloured and varies in size from 50 x 40 pixels at the beginning of the sequence to 140 x 80 pixels at the point it leaves the scene.

This sequence evaluates the performance of the sea characterisation algorithm on a brightly lit sea with highlights present and evaluates the identification and tracking algorithms when presented with a light coloured

large object. It also tests the collision risk estimation as the object approaches the centre of the image as shown in figures 7.1a and b, crosses it, and heads away from the centre, figure 7.1c, before leaving the scene, figure 7.1d. At the very top of the image throughout the sequence are a number of dark stationary objects.

Figure 7.1a –Sequence 1, Frame 10

Figure 7.1b – Sequence 1, Frame 100

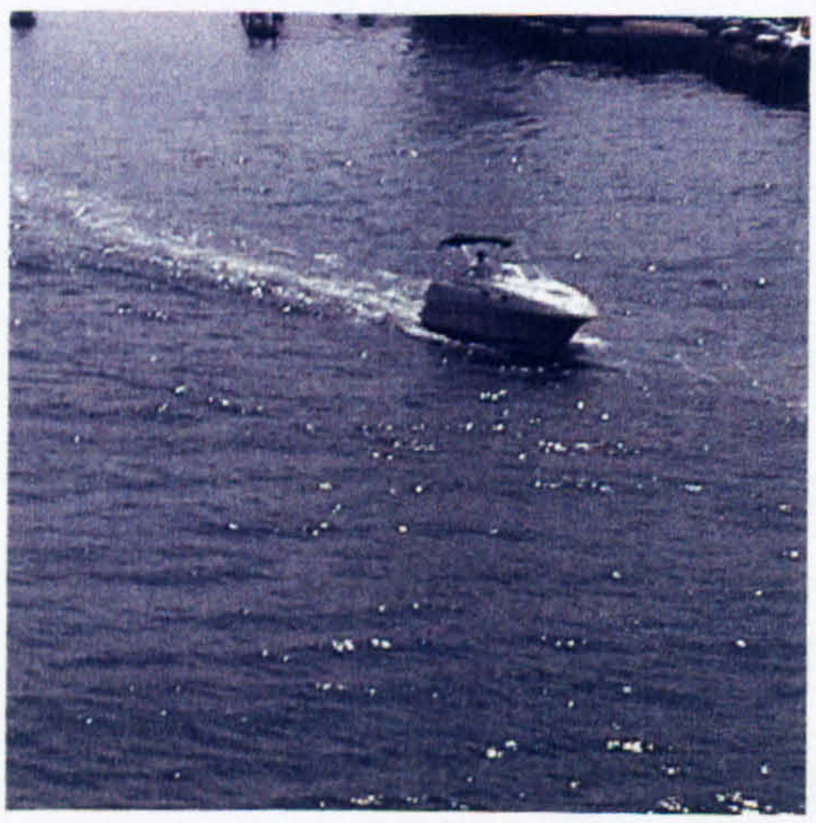

Figure 7.1c – Sequence 1, Frame 200

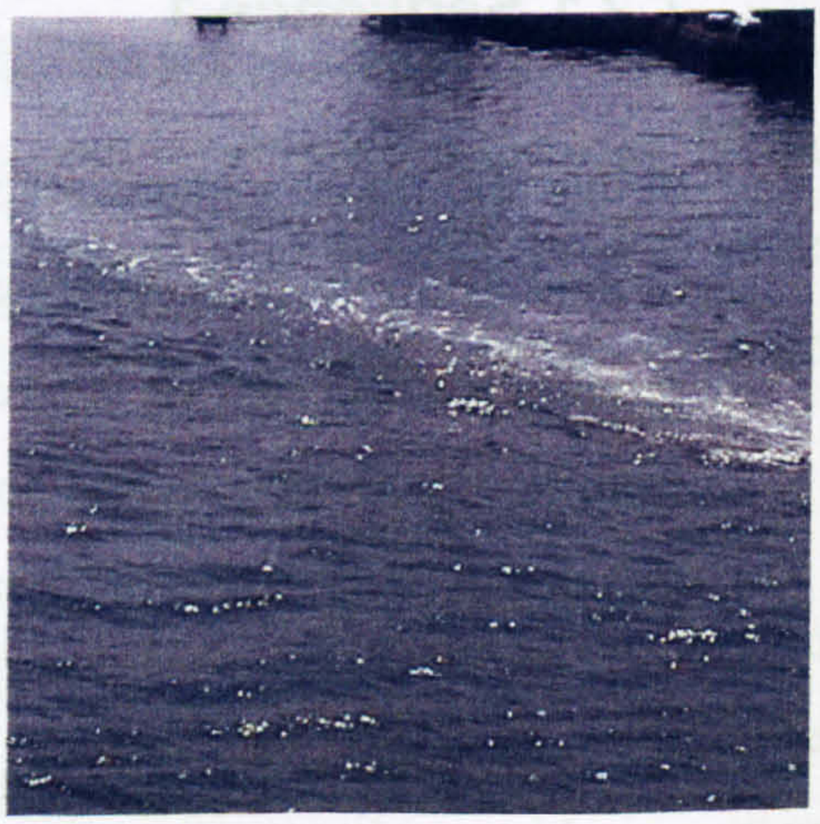

Figure 7.1d – Sequence 1, Frame 300

Figure 7.1 – Four frames from evaluation sequence 1

7.2.2 Sequence 2

This sequence shows the camera platform moving forward along a shipping lane. Figures 7.2a to 7.2d, taken from the sequence, show a channel marker buoy on the right of the image moving down and out of the image as the vessel moves towards and past the buoy. There is also a yacht in the top left of the image. As the sequence progresses this yacht moves across the image to the right and down the image as the camera platform approaches the yacht. The illumination of the scene is very bright, there is a light wind present contributing to camera motion, and the sea is calm.

Sequence 2 tests the system's tracking and prediction algorithms with an object undergoing only incidental motion in the form of the marker buoy and a slow moving object in the yacht. The motion of the camera platform adds to the motion of both objects. If a fast moving object travelling at 34 knots were approaching the camera platform then the closing velocity would be faster than that set in the motion model. This is not considered to lead to the tracking algorithm failing to match the object between frames as the perspective present in the image will have the effect of reducing the closing velocity.

The collision risk estimation is tested as both objects move down in the image as the camera platform approaches them. In addition to moving down in the image, the yacht also moves from left to right across the path of the camera platform. From the reasoning strategy it is expected that the collision risk of the yacht will increase as it moves toward the centre of the image and then decrease as it moves away to the right. It is also expected that the collision risk will increase as the yacht moves down in the image. This sequence also tests the performance of the algorithms with small objects as both the marker buoy and the yacht begin some distance from the camera. In

figure 7.2a the yacht is just 16 x 53 pixels, including the tall thin mast, and the buoy 19 x 35 pixels.

The sequence is 800 frames in length and both objects start within the scene. The marker buoy leaves the scene at approximately frame 200 and the yacht leaves the scene at frame 760. The shoreline has been set to the level of the horizon, 21 pixels down from the top of the image. The very small dark object at the top centre of the scene will not be identified as it resides on the horizon. The top of the yacht's mast will also not be identified as this also resides above the level of the horizon.

Figure 7.2a – Sequence 2, Frame 10

Figure 7.2b – Sequence 2, Frame 200

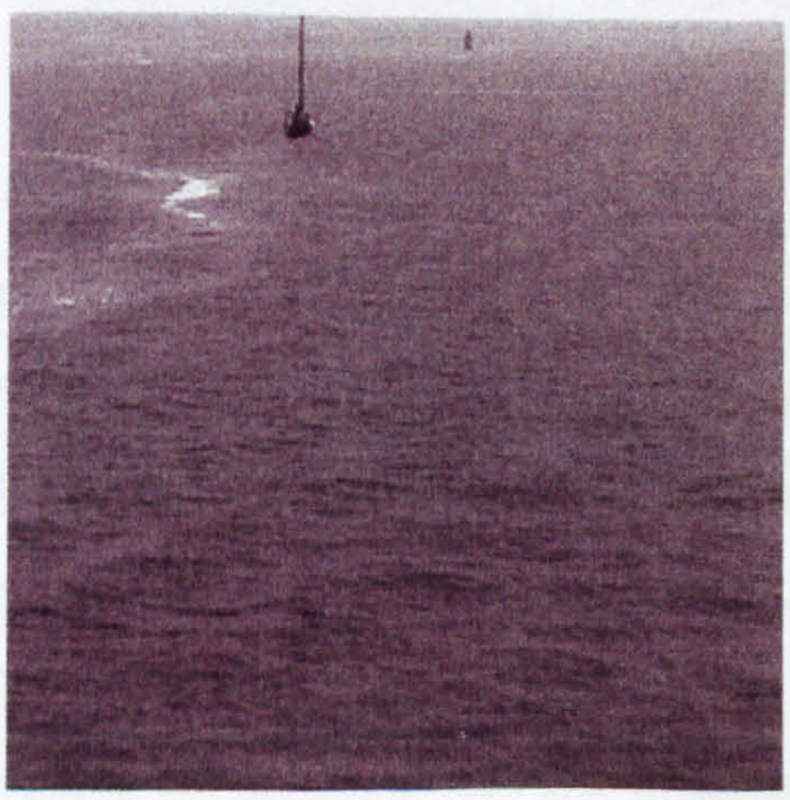

Figure 7.2c – Sequence 2, Frame 400

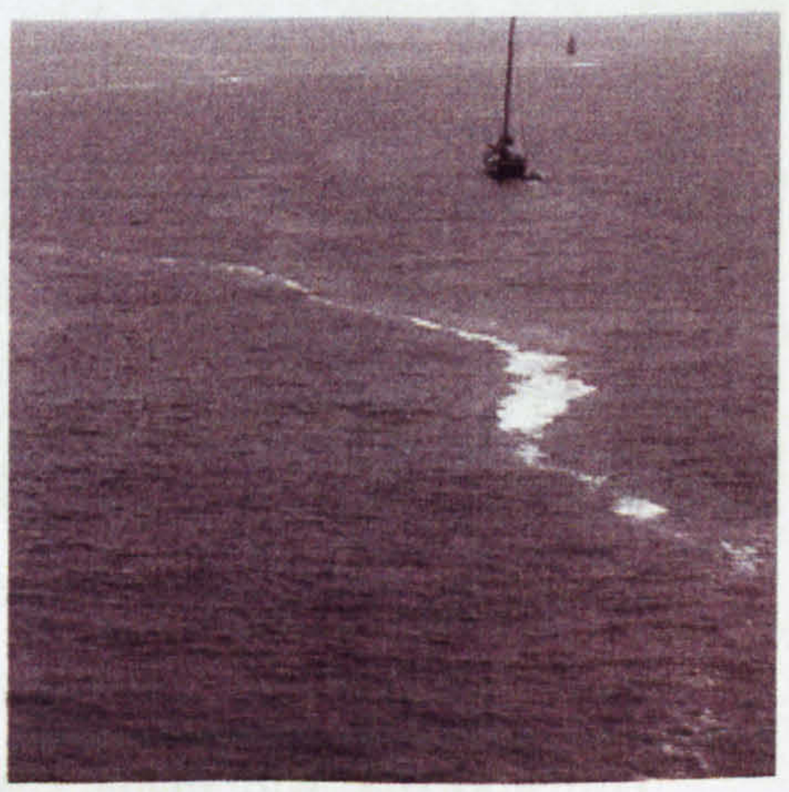

Figure 7.2d – Sequence 2, Frame 600

Figure 7.2 – Four frames from evaluation sequence 2

7.3 Sequence 1 Results

The output from the system for this sequence is shown in figure 7.3. Figure 7.3a shows the output from frame 10. The motor boat has been identified as a confirmed object (matched over more than 5 frames) as have two of the stationary objects at the top of the image. The area of dockside in the top right corner of the image has only been identified as an object (matched in less than 5 frames). This is due to the illumination of the scene. The shadow cast onto the water at the left end of the dockside varies from frame to frame resulting in it being segmented as a separate (false) object in some frames, as shown here, and as a single object in others. The motion prediction and collision risk estimation have been calculated for the motor boat and show its current collision risk to be 25%.

In figure 7.3b, showing the result from frame 100, it can be seen that the fast moving object has continued to be matched and had the future motion predicted and a collision risk estimated for it.

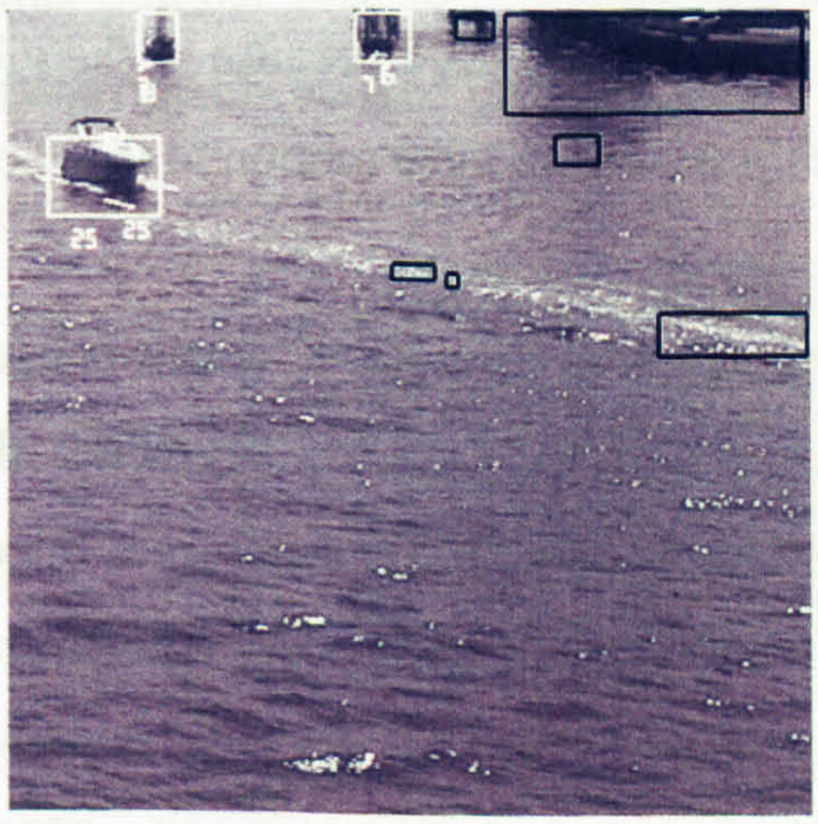

Figure 7.3a – Sequence 1, Processed Frame 10

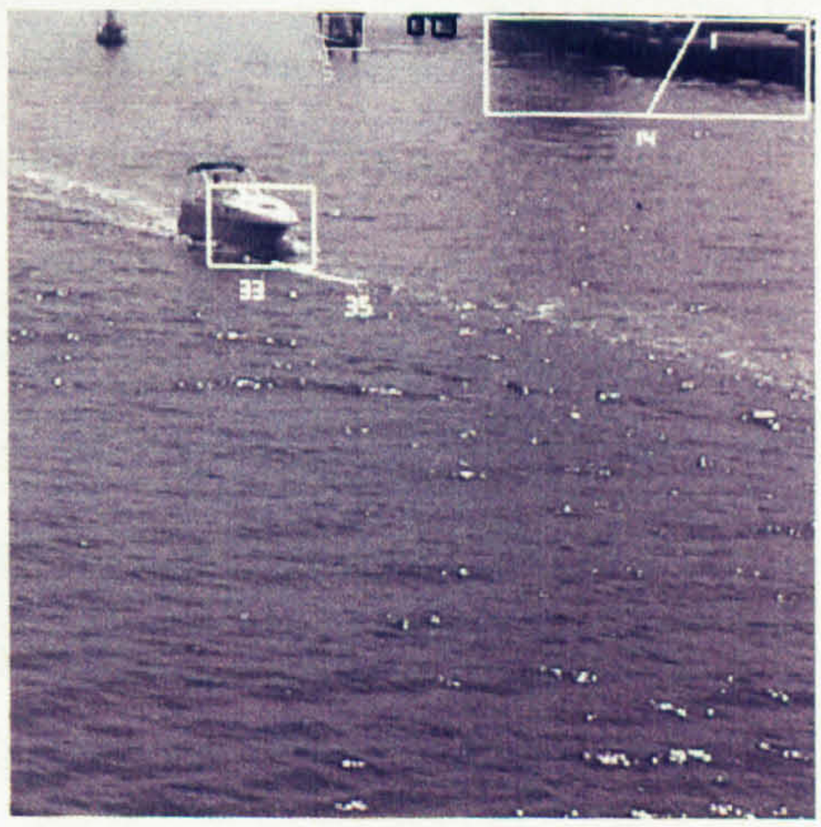

Figure 7.3b – Sequence 1, Processed Frame 100

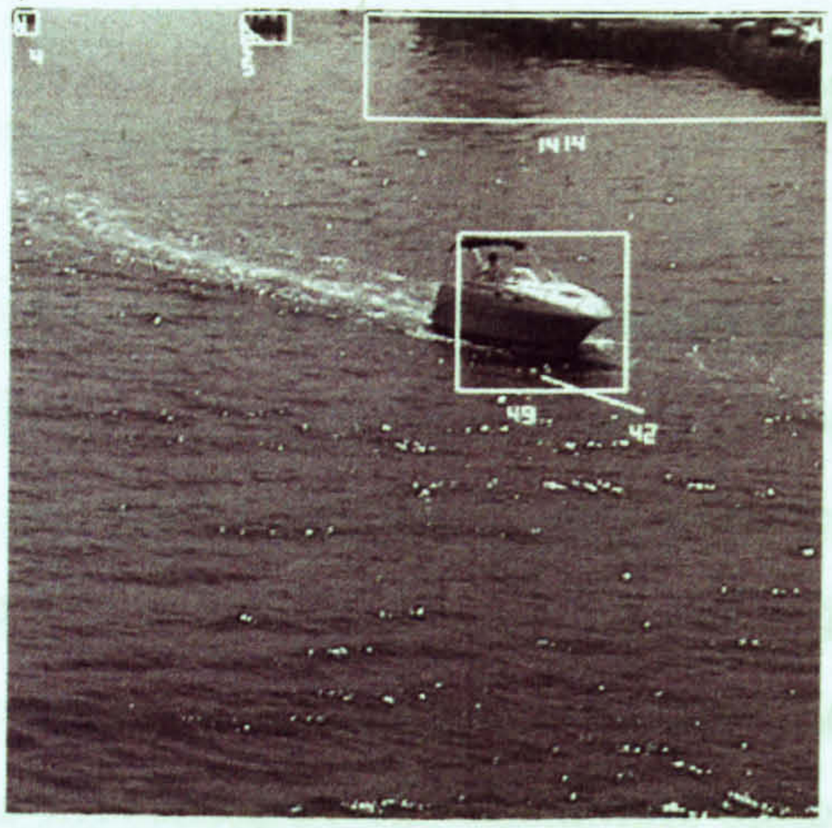

Figure 7.3c – Sequence 1, Processed Frame 200

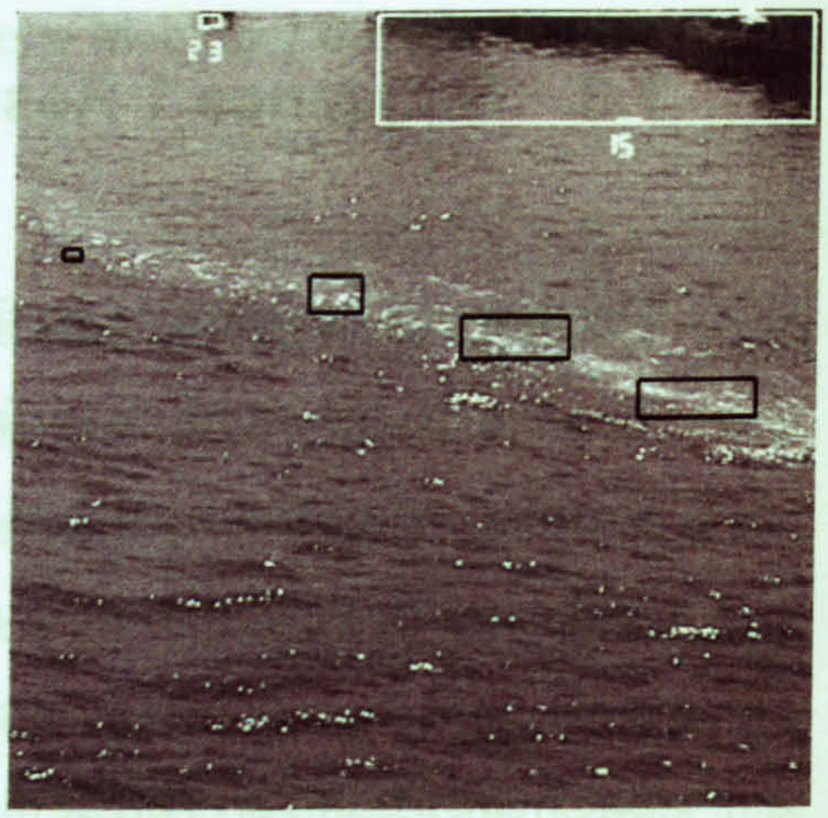

Figure 7.3d – Sequence 1, Processed Frame 300

Figure 7.3 – Evaluation sequence 1 results

As the object is moving towards the centre of the image the high-level reasoning strategy states that the risk of a collision should increase. The risk in figure 7.3a was 25%, in this frame the current risk is given as 33% and the future risk as 34% so the reasoning strategy is correct, the risk is increasing. The other objects in the scene have also continued to be identified as targets, including the dockside. Although stationary, the two objects in the top centre and left of the image are shown with a predicted future motion moving to the left. This has occurred because of the vibration and motion of the camera platform introducing false motion to those objects.

A collision risk has also been assigned to these objects, which is useful for the further testing of the reasoning strategy. The strategy states that the further up the image the object is located, the further away from the camera platform the object is in the real world and therefore poses a lesser threat of collision. It can be seen that the current collision risk for these two objects is 4% and 5% respectively. These are considerably lower than the 33% of the motor boat object that is lower in the image and as such, closer to the camera so the strategy is shown to be correct.

In figure 7.3c the fast moving object is beginning to move from the centre of the image toward the right of the image. The fan structure shows that the risk of collision should decrease as the object moves away from the centre of the image. This figure shows the strategy be to correct with a current collision risk of 48% and the predicted future collision risk of 25%.

Figure 7.3d shows the wake of the motor boat as a number of false objects identified with black boxes. This false object changes appearance between frames however it does get confirmed as a maritime object in the later frames of the sequence.

Figure 7.4 shows the number of false objects identified in the scene and the number of false objects confirmed and tracked through the sequence. The numbers of false objects identified are those that are matched in less than 5 frames. Those confirmed and tracked are those matched in more than 5 frames and are displayed to the operator by the placing of a white box around them. An average of 3.4 false objects were identified across the sequence with a maximum number of 10 false objects identified in frame 250. The number of false objects tracked was 0 or 1 until frame 250 when this number rose to 3. Across the whole sequence the average number of false objects tracked is 0.5.

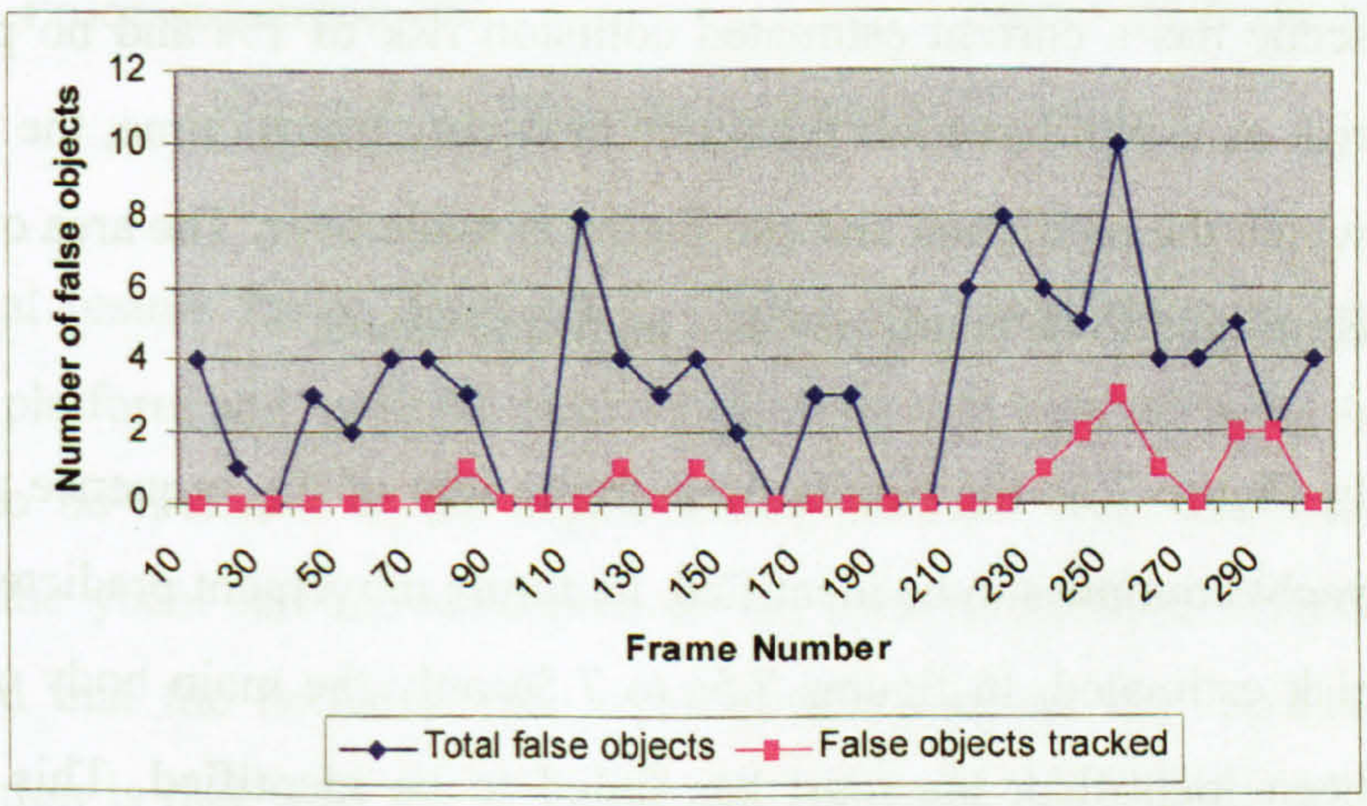

Figure 7.4 - Total number of false objects and false objects tracked

7.4 Sequence 2 Results

Figure 7.5 shows the results of four frames from sequence 2. In 7.5a it can be seen that the two objects present, a yacht and a channel marker buoy, have both been confirmed as maritime objects by the white boxes drawn around them. In addition the trailing wake of a previous maritime object has also been confirmed as an object. This is a false object. It consists of a group of pixel intensities that are distinctly different from those of the sea, which is why it has been identified. If figures 7.5b to 7.5d are also considered it can be seen that this area has been identified throughout the sequence. This is due to the non-dispersal of the wake owing to the calm sea conditions of this sequence. Object motion prediction and collision estimation has taken place in figure 7.5a. Also present in figure 7.5a are two areas at the bottom of the image that have been matched across less than five frames. These are false objects that have been matched as the pixel intensities are outside the characteristic grey level range for the sea.

In figure 7.5b it can be seen that after 200 frames the confirmed maritime objects present in the scene have continued to be identified as objects and have future movement prediction and collision estimation calculated for them. The channel marker buoy, on the right of the image and leaving the scene has a current estimated collision risk of 1% and no predicted future risk as it will have left the scene in twenty frames time, the time scale over which the prediction and estimation is made over. The area of wake has also continued to be confirmed as a maritime object.

In Figure 7.5c the results from frame 400 of the sequence are shown. The yacht continues to be identified, its future movement predicted, and collision risk estimated. In figures 7.5a to 7.5c only the main body of the yacht has been identified, the mast has failed to be identified. This is because the number of pixels the mast occupies in the tiles used for segmentation is small enough for the whole tile to be segmented as sea.

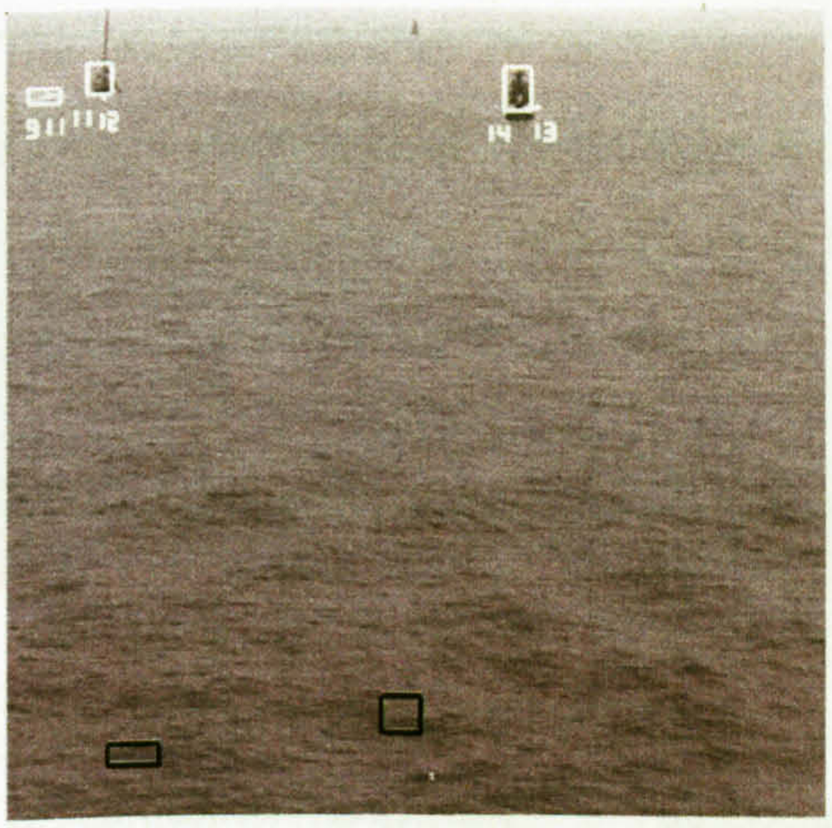

Figure 7.5a – Sequence 2,
Processed frame 10

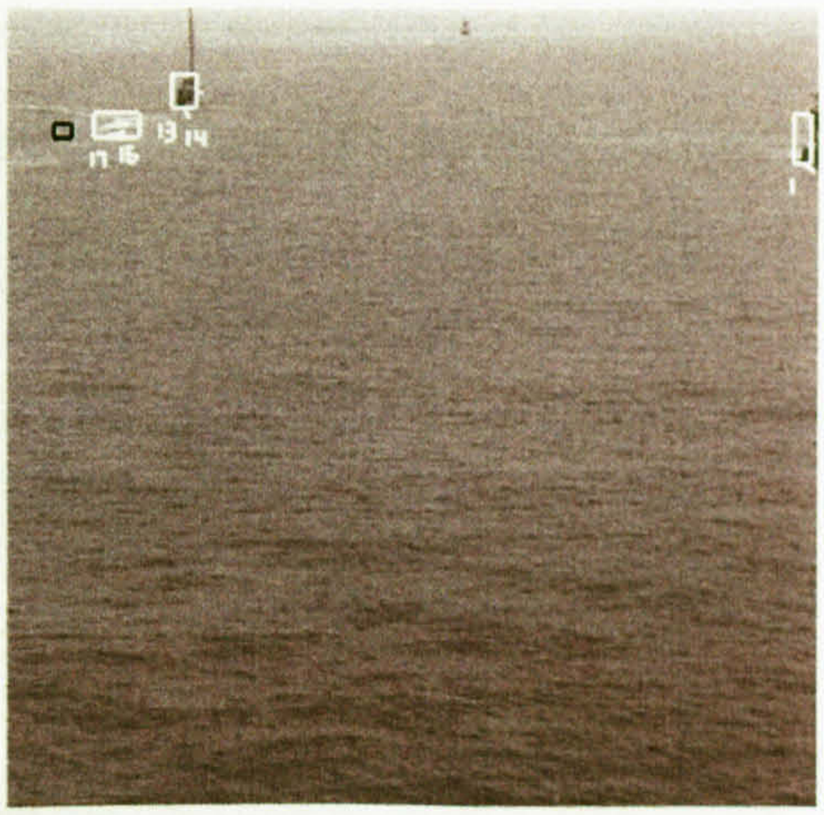

Figure 7.5b – Sequence 2,
Processed frame 200

Figure 7.5c – Sequence 2,
Processed frame 400

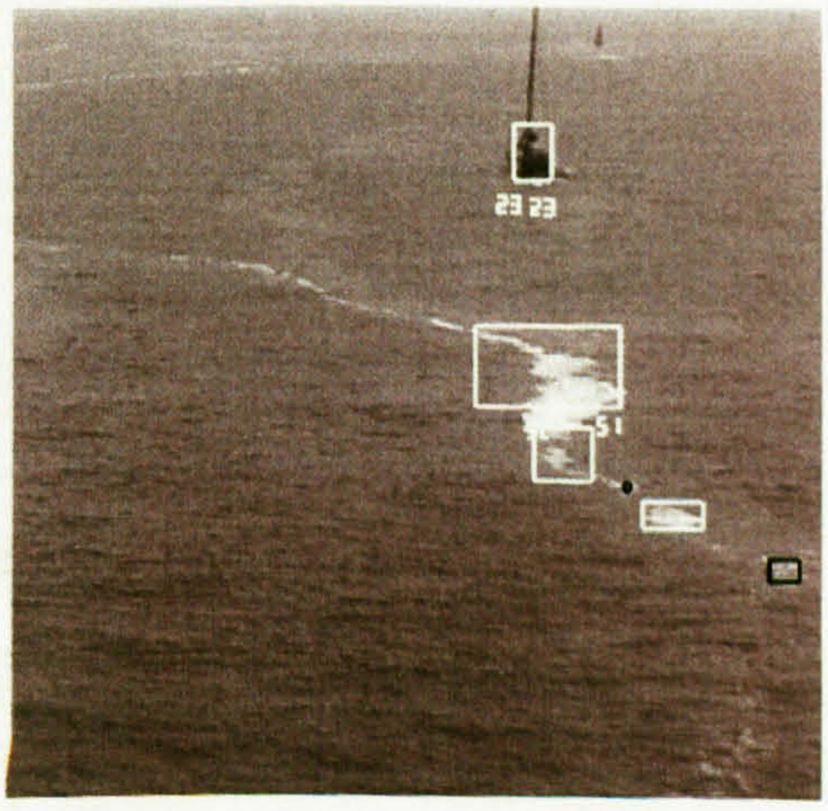

Figure 7.5d – Sequence 2,
Processed frame 600

Figure 7.5 – Evaluation sequence 2 results

The final results frame, figure 7.5d, shows the yacht much closer to the camera platform and includes much of the mast as it now occupies sufficient pixels to be included in the segmentation process. Over the four frames shown the yacht has moved closer to the camera platform therefore it is expected that the collision risk will increase through the sequence. This expectation is met with the collision risk estimates being 11%, 13%, 17%, and 23% for frames 10, 200, 400, and 600 respectively. The area of wake has also continued to be identified and has closed on the camera platform.

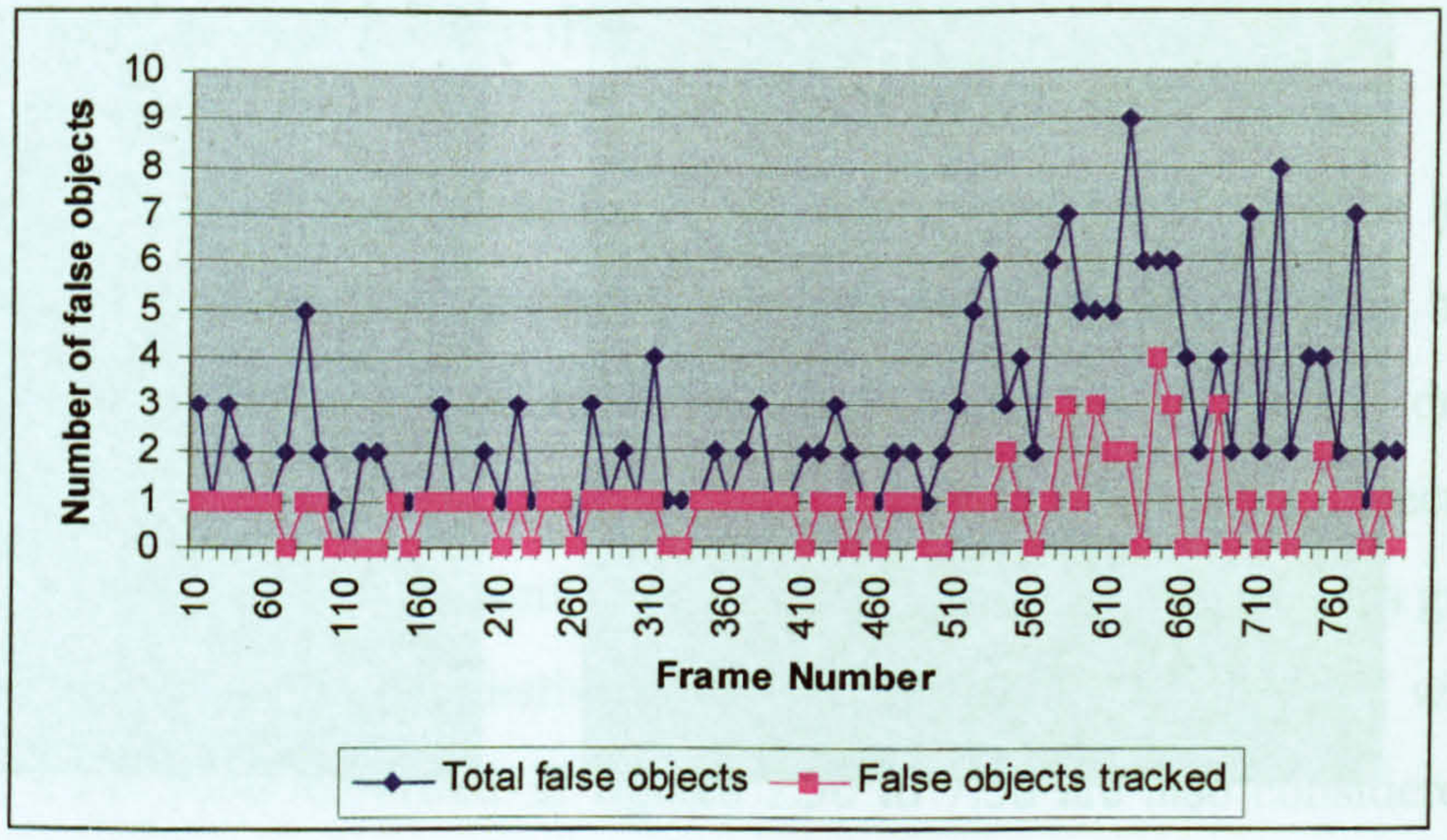

Figure 7.6 – Total number of false objects and false objects tracked

The high level reasoning strategy states that this should result in an increase in the collision risk, especially as it is central in the image. This is shown as the risk is given as 52% for its current position. A small number of areas of wake have been identified as objects with a black box indicating they have been matched in less than 5 frames. The vibration of the camera platform is evident in this frame with the image being less sharp than the others. This has not had an effect on the performance of the algorithms developed here showing a level of robustness and scene independence of the algorithms.

The number of false objects identified in the scene and the number of false objects confirmed and tracked through the sequence are shown in figure 7.6. The numbers of false objects identified are those that are matched in less than 5 frames. Those confirmed and tracked are those matched in more than 5 frames and are displayed to the operator by the placing of a white box around them. Figure 7.6 shows that the number of false objects rose as the sequence progressed. This is because the area of wake approaches the camera platform and is falsely segmented into a greater number of objects. Despite the total number of false objects rising to a maximum of 9 the number of false objects confirmed and tracked only reaches a maximum of 4. Across the whole sequence the average number of false objects is 2.7 and

the average number of false objects tracked is 0.8. The lower numbers of false objects tracked shows the success of the algorithm requiring an object to be matched across 5 frames before it is confirmed as a maritime object.

7.5 Discussion and Summary

Two previously unseen sequences taken from a maritime vessel have been presented to the software system developed in this thesis in order to evaluate the system for scene dependency and to test the high-level reasoning strategy with different maritime scenarios. In the first sequence the performance of the system is tested on a highly illuminated sea with many highlights present. This sequence also tests the performance of the system with a large light coloured object close to the camera. The camera platform undergoes motion introduced by engine vibration, and wind and vessel motion. The second sequence shows the vessel moving forward along a shipping lane. It tests the performance of the system with slow moving (or stationary) objects that are a long distance away from the camera. All of these objects move within the image through the sequence as a result of the motion of the camera platform.

A fast moving motor boat object crosses sequence 1 from high on the left to low on the right and leaves the scene. The high-level reasoning strategy for this scenario would be for the collision risk this object poses to increase as it moves down and toward the centre of the image and for the collision risk to decrease as the object moves away from the centre of the image. The results shown give the collision risk as 25% in figure 7.3a, 33% increasing to 35% as the object moves toward the centre of the image in figure 7.3b, and 49% decreasing to 42% as it moves away from the centre of the image in figure 7.3c.

The number of false objects identified through sequence 1 is shown in figure 7.4. This average number of false objects identified in the sequence was 3.4 although only an average of 0.5 of these went on to be tracked as confirmed maritime objects. This lower number of false objects tracked shows the success of the algorithm requiring an object to be matched across 5 frames before it is confirmed as a maritime object.

The objects present in the second sequence are a channel marker buoy that moves from high on the right of the scene down the right side of the image and leaves the scene as the vessel approaches and passes the object. The other object present is a yacht heading away from the vessel. It moves from the top left of the scene, down and across the centre of the scene finally leaving it on the right side. A third small dark object at the very top of the scene is above the horizon and therefore above the shoreline set for this sequence. As any area above the horizon does not contain grey levels from the sea it is in an area not considered by this system and so the small object will not be detected.

This second sequence shows how the system handles slow moving (and stationary) objects from a moving camera platform. The high-level reasoning strategy should identify the buoy as being toward the top of the image and as such pose a small risk of collision. As the sequence develops the buoy moves down in the image and to the right side of the scene. The risk of collision should increase as the object moves down in the image then decrease as the buoy moves toward the edge of the scene. Results show the collision risk presented by the buoy to be 14% decreasing to 13% in frame 8.5a, increase to 17% in frame 100 (not shown) and decrease to 3% in frame 200 as it leaves the scene. These results show the high-level reasoning strategy to be correct.

The yacht also present in sequence 2 is expected to present a low collision risk at the outset of this sequence as it is situated very high in the scene representing some distance from the camera and hence the vessel. As the sequence continues the collision risk is expected to rise slightly as the vessel approaches the yacht moving it down in the scene. This should be followed by a decrease in collision risk as the yacht moves to the right of the scene and finally leaves the scene. Figure 7.5a shows the yacht with a collision risk of 11% increasing to 12%, frame 200 (figure 7.5b) shows the yacht with a 13% risk increasing to 14%, frame 400 (figure 7.5c) as 17%, and frame 600 (figure 7.5d) as 23%. At this point the yacht is still moving down the scene as so the high-level reasoning strategy has performed correctly by estimating an increasing risk. In frame 750 (not shown) the yacht is situated on the far right of the image and has a collision risk of 13% with a predicted risk decreasing to 2%.

This second sequence does contain a false object that is identified as a target that continues to be identified as a target throughout the sequence. This is an area of wake from some maritime object that has left the scene and has occurred because of the calm sea conditions. The area of wake has not dispersed as would usually be the case resulting in the area being matched over multiple frames and considered a valid target. Figure 7.6 shows the number of false objects identified through sequence 2. The average number of false objects identified in the sequence was 2.7 of which an average of 0.8 went on to be tracked as confirmed maritime objects.

Chapter 8: Discussion and Conclusions

8.1 Discussion

The research presented here has been concerned with the complex task of identifying and tracking maritime objects in open world maritime scenes. This work has been further concerned with arriving at an estimate of the risk of collision between an identified maritime object and the camera platform. Of particular interest here is the ability of the system to identify any maritime object – described as any object that protrudes above the surface of the water – at any distance from the front of the camera platform (approximately 50 metres from the camera) to the horizon. This had led to the direction the development of some of the algorithms has taken.

The system has been broken down into four component parts: filtering of the input images; segmentation of the images; tracking of segmented objects; and finally the prediction of future object motion and collision risk estimation.

The first stage of the system is the enhancement of the input images. This is carried out to remove that noise introduced by the capture and digitisation process. The noise introduced into the system has been assumed to approximate a gaussian distribution. In addition to removing noise the image

enhancement had to enhance edge detail and smooth the grey-level distribution in the image. Results show the problems caused by noise were reduced by the application of the Kuwahara filter (Kuwahara et al, 1976) to maritime images. A fixed point image, arrived at by the repeated use of the Kuwahara filter on the same image, was found to be successful by Minato et al (1987) for contour detection. When applied to maritime images it was found that in addition to enhancing the intensity difference between valid maritime objects and the sea, differences between areas of sea were also enhanced. Consequently a single application of the Kuwahara filter was used.

The segmentation stage involves the identification of the maritime objects present in the scene. Detecting the motion of maritime objects with frame differencing methods would lead to a great number of false motion cues as almost every part of the image is moving. The sea is constantly moving, the objects are moving, and the camera platform is moving. Model-based techniques such as the geometric models used by Ferryman et al (1998) and Koller and Nagel (1993), or the feature model used by Luo and Etz (2002) require the reliable extraction of features that can be fitted to models. The wide variety of object shapes and the distances the objects could be away from the camera would yield unreliable features for these techniques to be successful. The method used was to calculate the characteristic grey-level range for the sea and to segment the image into two regions, those that contained the grey-level range, the sea, and those that did not, the maritime objects. This led some objects to be fragmented and so an 8-way connectivity method of region growing was used to join fragmented objects.

A feature vector is built for each object segmented in the image containing details including its co-ordinates and area. This vector is used by the tracking algorithm together with a maritime motion model to match objects between frames. The system considers objects to belong in one of two

categories, fast moving or slow moving. The distinction is made as the maritime motion model allows for different motion characteristics depending on the category of the object. Any object that is matched over at least two frames is identified by a black box drawn around the object. If an object is matched over five or more frames then it is confirmed as a maritime object and a white box drawn. This allows for the number of false objects to be reduced.

In each frame that an object or target is matched data regarding its current position is recorded in the feature vector. Targets that have been matched over more than 8 frames have a least-squares algorithm applied to their position data to arrive at a line of best fit through the data and to predict the future motion of the target.

Finally a high-level reasoning algorithm based on a rule-based strategy is used to estimate the risk of a collision that each target poses to the vessel the camera is mounted on. The pattern of risk suggested a structure that can be visualised as a fan shape with the highest risk at the bottom centre of the image and the lowest risk at the outer edge of the image. The percentage risk presented by the bottom centre of each target is obtained and displayed. The risk posed by the predicted future position of the target is also calculated and displayed.

8.2 Conclusions

The aim of this research has been to develop a machine vision system that would highlight objects of interest (maritime objects), track these objects over time, indicate the future course of the object, and provide an estimate of the likelihood of a collision occurring with that object.

It was found that the relatively straightforward techniques of statistical analysis of image histograms and the application of a set of motion constraints based on knowledge of the problem domain can be used to identify and track maritime objects.

Following confirmation in chapter 5 that the assumption made by Wang et al (2000) - object motion approximates a straight line over a few frames – held true for maritime objects the previous motion of identified objects was used to calculate a line of best fit through the motion data. This line was extended in the object's direction of travel to give an indication of future motion.

A knowledge-based reasoning strategy utilising known information about the maritime environment and the motion of maritime objects was used to estimate the risk of collision an identified and tracked object posed to the camera platform.

It has been demonstrated through the use of two image sequences representative of the situation the system is likely to be used in that without modification the developed algorithms could identify, track, and estimate the collision risk posed by maritime objects present in the scene

Little previous work has been carried out toward using image processing in the maritime environment and less still on the specific problem addressed here of collision avoidance. The originality and contribution of the work presented in this thesis is to bring together new and existing image processing techniques and apply them to a specific problem in a new application area, that of safety in the maritime environment. Further it has been shown that an image processing system can successfully address a problem traditionally solved by radar and human observation.

In each of the four stages of this system existing appropriate techniques were evaluated against methods developed specifically for use with maritime images. In the case of the image filtering the existing Kuwahara filter was chosen as the most appropriate. However in the other three stages newly developed algorithms outperformed existing techniques such as thresholding and frame differencing for the segmentation of the images and the Kalman filter for tracking the objects over time.

8.3 Limitations and Future Work

In order to show that applying image processing techniques to the maritime environment could be successful the sequences used here both to develop the algorithms and to evaluate the system have been taken in generally good weather conditions. The Poole test sequence was taken in dull, overcast conditions with a calm sea. The Dover sequence contained bright sunlight and a slight swell, as did the first evaluation sequence. The Portsmouth sequence was bright but calm, as was the second evaluation sequence.

The favourable weather conditions were used as this is considered to be the conditions when the level of observation on board ship will be at its lowest. These are also the conditions when objects will be moving at their highest speeds. The calm sea conditions used have also allowed the shoreline parameter to be a fixed value. Both the weather conditions and the fixed shoreline are limitations of the current system and should be addressed in future work. Whilst thick fog and very dark conditions are beyond the capability of cameras operating solely in the visual spectrum and therefore could not be considered the effects of rain or light mist or strong winds on the system are not. These environmental conditions might be addressed through the extraction of different features for different conditions as used by Cucchiara and Piccardi (1999). The fixed shoreline has been shown not

to restrict the performance of the system, as in the calm sequences used the pitch of the camera platform altered by just a small number of pixels.

Another limitation of the current system is the treatment of occluded objects. This is addressed by the targets being combined and treated as a single target for the duration of the occlusion. This does not effect the performance of the system, as the collision risk will be that of the closer target. However, when the targets cease to occlude each other it currently takes a number of frames for the targets to fully separate during which time the collision estimate is being made on the point midway between the two targets. A second issue regarding occlusion is that once the occlusion ceases the occluded object is considered to be a new object and will not be identified as a maritime object for a further five frames. Future work in this area would be for each target to be treated entirely independently leading to the prediction of occlusion duration and the correct collision risk estimation of the closer maritime target. This would also lead to the occluded object being identified as a maritime object as soon as the occlusion ceases.

A further limitation of this system is the environment in which it is designed to be used. As discussed in chapter 1, section 3, the system has been currently designed for open sea where object motion is assumed to be uniform and where only a few objects will be present. Future work in this area would involve allowing other environments such as entering and leaving a harbour. This would require updating the motion model to deal with a wider range of objects such as fast manoeuvring jet ski's and also to deal with objects whose velocity changes over a few frames.

Finally, future consideration could be given to the use of calibrated stereo cameras to make more accurate calculations regarding the distance of objects from the camera leading to a greater accuracy in the collision risk estimation. This may also help with the issue of object occlusion.

References:

AACH, T., DUMBGEN, L., MESTER R., AND TOTH, D., 2001. Bayesian Illumination-Invariant Motion Detection. *In IEEE International Conference on Image Processing*. 7-10 October 2001 Thessaloniki, Greece. Texas, USA: IEEE. 640-643.

ACHELIS, S.B. (2004) *Technical Analysis from A-Z* [online]. Salt Lake City, USA, Reuters. Available from: www.equis.com/Education/taaz/?page=65 [Accessed 3 August 2004].

AGBINYA, J.I. AND REES, D., 1999. Multi-object tracking in video. *Real Time Imaging*, 5(5), 295-304.

ARROWSMITH, M.J., VARLEY, M.R., PICTON, P.D. AND HEYS, J.D., 1999. Hybrid Neural Network System for Texture Analysis. *In 7th IEE International Conference on Image Processing and its Applications*, 13-15 July 1999 Manchester, UK. London: IEE. 339-343.

BAKKER, P., VAN VLIET, L.J., AND VERBEEK, P.W., 1999. Edge Preserving Orientation Adaptive Filtering. *In IEE Conference on Computer Vision and Pattern Recognition*, 23-25 June 1999 Fort Collins, USA. Los Alamitos, USA: IEEE. 535-540.

BALLARD, D.H. AND BROWN, C.M., 1982. *Computer Vision.* New Jersey, USA: Prentice Hall.

BARRON, J.L., FLEET, D.J. AND BEAUCHEMIN, S.S., 1994. Peformance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1), 43-77.

BASSMANN, H. AND BESSLICH, P.W., 1995. *Ad Oculos - Digital Image Processing*. 2nd ed. Oxford: International Thomson Publishing.

BAUMELA, L. AND MARAYALL, D., 1995. Real-Time Target Tracking. *IEEE Aerospace & Electronic Systems Magazine*, 10(7), 4-7.

BLACK, M.J., 1996 The Robust Estimation of Multiple Motions: Parametric and Piece-Smooth Flow Fields. *Computer Vision & Image Processing*, 63(part), 75-104.

BLACK, M.J., SAPIRO, G., MARIMONT, D.H. AND HEEGER, D., 1998. Robust Anisotropic Diffusion. *IEEE Transactions on Image Processing, 7(3), 421-432.*

BRODATZ, P., 1966. *Texture: A Photographic Album for Artists and Designers*. New York: Dover Publications.

CANNY, J.F., 1986. A Computational Approach to Edge Detection. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.

CASTLEMAN, K.R., 1996. *Digital Image Processing*. New Jersey, USA: Prentice Hall.

CHAUDHURI, B.B. AND SARKAR, N., 1995. Texture Segmentation Using Fractal Dimension. *IEEE transactions on Pattern Analysis & Machine Intelligence*, 17(1), 72-77.

CHEN, T. AND WU, H.R., 2001. Application of Partition-Based Median Type Filters for Suppressing Noise in Images. *IEEE Transactions on Image Processing*, 10(6), 829-836.

CHENG, H.D., CHEN, J-R. AND LI J., 1998. Threshold Selection Based on Fuzzy c-Partition Entropy Approach. *Pattern Recognition*, 31(7), 857-870.

CRISTOBAL, G. AND HORMIGO, J., 1999. Texture segmentation through eigen-analysis of the Pseudo-Wigner distribution. *Pattern Recognition Letters*, 20(3), 337-345.

CUCCHIARA, R. AND PICCARDI, M., 1999. Vehicle Detection under Day and Night Illumination. *In International ICSC Symposium on Intelligent Industrial Automation*, 1-4 June 1999 Geneva, Italy. Geneva, Italy: NAISO. 789-794.

DAVIES, D., PALMER, P. AND MIRMEHDI, M., 1998. Detection and Tracking of Very Small Low Contrast Objects. *In 9th British Machine Vision Conference*, 14-17 September 1998 Southampton, UK. Southampton: BMVA. 599-608.

DI BONA, S., PIERI, G. AND SALVETTI, O., 2001. A Multilevel Neural Network Model for Density Volumes Classification. *In 2nd International Symposium on Image and Signal Processing and Analysis*, 19-21 June 2001 Pula, Croatia. Pula, Croatia: IEEE. 213-218.

DI MAURO, E.C., COOTES, T.F., TAYLOR, C.J. AND LANITIS, A., 1996. Active Shape Model Search using Pairwise Geometric Histograms. *In British Machine Vision Conference*, 9-12 September 1996 Edinburgh, UK. Edinburgh: BMVA. 353-362.

DUNN, D. AND HIGGINS, W.E., 1995. Optimal Gabor Filters for Texture Segmentation. *IEEE transactions on Image Processing*, 4(7), 947-964.

ELLIS, T.J., ROSIN, P.L. AND GOLTON, P., 1991. Model-Based Vision for Automatic Alarm Interpretation. *IEEE Aerospace and Electronic Systems Magazine*, 6(3), pp. 14-20.

ENKELMANN, W., 1990. Obstacle detection by evaluation of optical flow fields from image sequences. *In 1st European Conference on Computer Vision*, 23-27 April 1990 Antibes, France. Antibes, France: Springer. 134-138.

EVANS, R., 1990 Kalman Filtering of Pose Estimates in Applications of the RAPID Video Rate Tracker. *In 1st British Machine Vision Conference*, 24-27 September 1990 Oxford UK. Oxford: BMVA. 79-84.

FERRYMAN, J.M., WORRALL, A.D., SULLIVAN, G.D. AND BAKER, K.D., 1995. A Generic Deformable Model for Vehicle Recognition. *In British Machine Vision Conference*. 11-14 September 1995 Birmingham, UK. Birmingham: BMVA. 127-136.

FERRYMAN, J.M., WORRALL, A.D. AND MAYBANK, S.J., 1998. Learning Enhanced 3D Models for Vehicle Tracking. *In BMVC '98*. 14-17 September 1998 Southampton, UK. Southampton: BMVA. 873-882.

FLEET, M. (1999) *Banana boat crashes into flashing lighthouse*. [online]. London, UK, Daily Telegraph. Available from: www.telegraph.co.uk/htmlcontent.jhtml;sessionid=FEZYIDDX1DQFIQMFSM5OAVCBQOJVC?html=/archive/1999/11/08/nboa08.html [Accessed 11 February 2002].

FRAILE, R. AND MAYBANK, S.J., 1998. Vehicle Trajectory Approximation and Classification. *In 9th British Machine Vision Conference*, 14-17 September 1998 Southampton, UK. Southampton: BMVA. 832-840.

FRANCHI, G., GAMBA, P., MARAZI, A. AND MECOCCI, A., 1996. Object Tracking in Complex Scenes by means of the Correspondence Based Method. *In MELECON'96*, 13-16 May 1996 Bari, Italy. Bari, Italy: IEEE. 1093-1096.

GALIC, S. AND LONCARIC, S., 2000. Spatio-Temporal Image Segmentation Using Optical Flow and Clustering Algorithm. *In 1st International Workshop on Image and Signal Processing and Analysis*. 14-15 June 2000 Pula, Croatia. Pula, Croatia: Sveucilisni Racunski. 63-68.

GLASBEY, C, A. AND HORGAN, G, W., 1995. *Image Analysis for the Biological Sciences*. New York, USA: Wiley.

GONZALEZ, R.C. AND WOODS, R.E., 1992. *Digital Image Processing*. Massachusetts, USA: Addison-Wesley.

GREWAL, M.S. AND ANDREWS, A.P., 1993. *Kalman Filtering, Theory and Practice*. New Jersey, USA: Prentice-Hall.

GUERRERO, J.J. AND SAGUES, C., 1999. Camera Motion from Brightness on Lines. Combination of Features and Normal Flow. *Pattern Recognition*, 32(2), 203-216.

HARALICK, R.M. AND SHAPIRO, L.G., 1992. *Computer and Robot Vision, vol. 1*. Massachusetts, USA: Addison Wesley.

HARRIS, C.G. AND STEPHENS, M., 1988. A Combined Corner and Edge Detector. *In 4th Alvey Vision Conference*, 31 August-2 September 1988 Manchester, UK. Sheffield: University of Sheffield Printing Unit. 147-151.

HORN, B.K.P. AND SCHUNCK, B.G., 1981. Determining Optical Flow. *Artificial Intelligence*, 16(1-3), 185-203.

HOSIE, R. (1998) *Kalman Filtering* [online]. Perth, Australia, Curtin University of Technology. Available from: www.cs.curtin.edu.au/~robin/thesis/node15.html [Accessed 22 October 2000].

HUANG, Y., PAULUS, D., AND NIEMANN, H., 2000. Background-Foreground Segmentation Based on Dominant Motion Estimation and Static Segmentation. *In First Int. Workshop on Image and Signal Processing and Analysis*. 14-15 June 2000 Pula, Croatia. Pula, Croatia: Sveucilisni Racunski. 69-74

HUANG, Y., HUANG, T.S. AND NIEMANN, H., 2002. Segmentation-based Object Tracking Using Image Warping and Kalman Filtering. *In International Conference on Image Processing*, 22-25 September 2002 Rochester, USA. New York, USA: IEEE. 601-604.

ISHII, T. AND KYUMA, K., 2001. An Image Segmentation and Tracking Method by Segment Scanning Wave Propagation Network. *In IEEE International Conference on Image Processing*. 7-10 October 2001 Thessaloniki, Greece. Texas, USA: IEEE. 774-777.

JAIN, R., 1984. Difference and Accumulative Difference Pictures in Dynamic Scene Analysis. *Image and Vision Computing*, 2(2), 99-108.

JAIN, R., KASTURI, R. AND SCHUNCK, B.G., 1995. *Machine Vision.* New York, USA: McGraw Hill.

KALMAN, R.E., 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME--Journal of Basic Engineering*, 82(D), 35-45.

KAPUR, J.N., SAHOO, P.K. AND WONG, A.K.C., 1985. A New Method for Grey-Level Picture Thresholding using the Entropy of the Histogram. *Computer Vision, Graphics, and Image Processing*, 29(3), 273-285.

KOLLER, D. AND NAGEL, H.H., 1993. Model-based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. *International Journal of Computer Vision*, 10(3), 257-281.

KUWAHARA, M., HACHIMURA, K., EIHO, S. AND KINOSHITA, M., 1976. Processing of RI-angiocardiographic images. *In Digital Processing of Biomedical Images*, 1976, 187-202.

LAM, C.L. AND YUEN, S.Y., 1998. An unbiased active contour algorithm for object tracking. *Pattern Recognition Letters*, 19(5-6), 491-498.

LAN, P., JI, Q. and LOONEY, C.G., 2002. Information Fusion with Bayesian Networks for Monitoring Human Fatigue. *In 5th International Conference on Information Fusion*. 7-11 July 2002 Maryland, USA. Maryland, USA: IEEE. 535-542.

LAWS, K.I., 1980. *Textured Image Segmentation*. Thesis (PhD) University of Southern California.

LEE, J-S., RHEE, K-Y. AND KIM S-D., 2001. Moving Target Tracking Algorithm Based on the Confidence Measure of Motion Vectors. *In International Conference on Image Processing*, 7-10 October 2001 Thessaloniki, Greece. Texas, USA: IEEE. 369-372.

LINNETT, L.M. AND CLARKE, S.J., 1997. The generation of a time correlated 2D random process for ocean wave motion. *In 6th IEE International Conference on Image Processing and its Applications*, 14-17 July 1997 Dublin, Ireland. London: IEE. 623-626.

LOU, J., LIU, Q., TAN, T. AND HU, W., 2002. Semantic Interpretation of Object Activities in a Surveillance System. *In 16th International Conference on Pattern Recognition*, 11-15 August 2002 Quebec, Canada. Quebec, Canada: IEEE. 30777-30780.

LOW, A., 1991. *Introductory Computer Vision and Image Processing*. London: McGraw-Hill.

LUO, J. AND ETZ, S.P., 2002. A Physical Model-Based Approach to Detecting Sky in Photographic Images. *IEEE Transactions on Image Processing*, 11(3), 201-212.

MALLOT, H.A., BULTHOFF, H.H., LITTLE, J.J. AND BOHRER, S., 1991. Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection. *Biological Cybernetics*, 64(3), 177-185.

MANDELBROT, B.B., 1982. *Fractal Geometry of Nature*. San Francisco: Freeman Press.

MANN, S. AND PICARD, R.W., 1997. Video Orbits of The Projective Group: A Simple Approach to Featureless Estimation of Parameters. *IEEE transactions on Image Processing*, 6(9), 1281-1295.

MADERLECHNER, G. AND MAYER, H., 1994. Automated Acquisition of Geographic Information from Scanned Maps for GIS using Frames and Semantic Networks. *In 12th International Conference on Pattern Recognition*, 9-13 October 1994 Jerusalem, Israel. Tel Aviv, Israel: IEEE. 361-363.

MAKRIS, D. AND ELLIS, T.J., 2003. Automatic Learning of an Activity-Based Semantic Scene Model. *In IEEE Conference on Advanced Video and Signal Based Surveillance*, 21-22 July 2003 Miami, USA. Florida, USA: IEEE. 183-188.

MARCENARO, L., FERRARI, M., MARCHESOTTI, L. AND REGAZZONI, C.S., 2002. Multiple Object Tracking Under Heavy Occlusions By Using Kalman Filters Based On Shape Matching. *In International Conference on Image Processing*, 22-25 September 2002 Rochester, USA. New York, USA: IEEE. 341-344.

MATESIN, M., LONCARIC, S. AND PETRAVIC, D., 2001. A Rule-Based Approach to Stroke Lesion Analysis from CT Brain Images. *In 2nd International Symposium on Image and Signal Processing and Analysis*, 19-21 June 2001 Pula, Croatia. Pula, Croatia: IEEE. 219-223.

MAYBECK, P.S., 1979. *Stochastic models, estimation, and control, volume 1*. New York, USA: Academic Press.

McBETH, J. (2002). *English channel ferry warning as ships collide* [online]. Scotland, Scotsman. Available from: http://news.scotsman.com/topics.cfm?tid=703&id=1402922002 [Accessed 3 June 2003].

MESSER, K. AND KITTLER, J., 1998. Choosing an Optimal Neural Network Size to Aid a Search through a Large Image Database. *In 9th British Machine Vision Conference*, 14-17 September 1998 Southampton, UK. Southampton: BMVA. 235-244.

MESSER, K., DE RIDDLER, D. AND KITTLER, J., 1999. Adaptive Texture Representation Methods for Automatic Target Recognition. *In 10th British Machine Vision Conference*, 13-16 September 1999 Nottingham, UK. Nottingham: BMVA. 443-452.

MILLS, S. AND NOVINS, K., 2000. Motion Segmentation in Long Image Sequences. *In British Machine Vision Conference*. 11-14 September 2000 Bristol, UK. Bristol, UK: BMVA. 162-171.

MINATO, K., TANG, Y-N., BENNETT, G.W. AND BRILL, B., 1987. Automatic Contour Detection Using a 'Fixed-Point Hachimura-Kuwahara Filter' For SPECT Attenuation Correction. *IEEE Transactions on Medical Imaging*, 6(2), 126-133.

MITICHE, A., FEGHALI, R. AND MANSOURI, A., 2002. Tracking Moving Objects as Spatio-Temporal Boundary Detection. *In 5th Southwest Symposium on Image Analysis and Interpretation*, 7-9 April 2002 Santa Fe, USA. New Mexico, USA: IEEE. 106-110.

OWENS, J., HUNTER, A. AND FLETCHER, E., 2002. *A Fast Model-Free Morphology-Based Object Tracking Algorithm* [online]. Cardiff, UK, BMVA. Available from: www.bmva.ac.uk/bmvc/2002/papers/99/full_99.pdf [Accessed 17 March 2003].

PELTONEN, S., GABBOUJ, M. AND ASTOLA, J., 2001. Nonlinear Filter Design: Methodologies and Challenges. *In 2nd International Symposium on Image and Signal Processing and Analysis*, 19-21 June 2001 Pula, Croatia. Pula, Croatia: IEEE. 102-107.

PEREZ-LUQUE, M.J., MUNOZ, C. AND GARCIA, N., 1990. Non Linear Spatial Filtering of FLIR Images. *SPIE*, 1360(0), 188-201.

PERONA, P. AND MALIK, J., 1990. Scale-Space and Edge Detection using Anisotropic Diffusion. *IEEE transaction on Pattern Analysis and Machine Intelligence*, 12(7), 629-639.

POZZER, C.T. AND PELLEGRINO, S.R.M., 2001. Procedural Models on Image Synthesis for Ocean Animation. *In 14th Brazilian Symposium on Computer Graphics and Image Processing*, 15-18 October 2001 Florianopolis, Brazil. Florianopolis, Brazil: IEEE. 130-137.

PUZICHA, J., HOFFMANN, T. AND BUHMANN, J.M., 1999. Histogram Clustering for Unsupervised Segmentation and Image Retrieval. *Pattern Recognition Letters*, 20(9), 899-909.

REID, I. (2001) *Estimation II* [online]. Oxford, UK. University of Oxford. Available from: www.robots.ox.ac.uk/~ian/Tracking/Estimation/LectureNotes2.ps.gz [Accessed 10 July 2004].

ROSIN, P. AND ELLIS, T.J., 1991. Detecting and Classifying Intruders in Image Sequences. *In 2nd British Machine Vision Conference*, 24-26 September 1991 Glasgow, UK. London: BMVA. 293-300.

RUSHING, J.A., RANGANATH, H., HINKE, T.H. AND GRAVES, S.J., 2002. Image Segmentation Using Association Rule Features. *IEEE transactions on Image Processing*, 11(5), 558-567.

SANDERSON, J.G., TEAL, M.K. AND ELLIS, T.J., 1997. Identification and Tracking in Maritime Scenes. *In 6th International Conference on Image Processing and its Applications*, 14-17 July 1997 Dublin, Ireland. London: IEE. 463-467.

SANDERSON, J.G., TEAL, M.K. AND ELLIS, T.J., 1999. Characterisation of a Complex Maritime Scene using Fourier Space Analysis to Identify Small Craft. *In 7th IEE International Conference on Image Processing and its Applications*, 13-15 July 1999 Manchester, UK. London: IEE. 803-807.

SENEL, H.G., PETERS, R.A. AND DAWANT, B., 2002. Topological Median Filters. *IEEE Transactions on Image Processing*, 11(2), 89-104.

SHIH, H-C. AND HUANG, C-L., 2003. A Semantic Network Modelling for Understanding Baseball Video. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, 6-10 April 2003 Hong Kong. Hong Kong: IEEE. 820-823.

SKOLNIK, M.I., 2002. *Introduction to Radar Systems*. New York, USA: McGraw-Hill.

SMITH, A.A.W. AND TEAL, M.K., 1999. Identification and Tracking of Maritime Objects in Near-Infrared Image Sequences for Collision Avoidance. *In 7th IEE International Conference on Image Processing and its Applications*, 13-15 July 1999 Manchester, UK. London: IEE. 250-254.

SMITH, A.A.W., TEAL, M.K. AND VOLES, P., 2003. The Statistical Characterisation of the Sea for the Segmentation of Maritime Images. *In 4th EURASIP Conference on Video and Image Processing and Multimedia Communications*, 2-5 July 2003 Zagreb, Croatia. Zagreb, Croatia: IEEE. 489-494.

SMITH, S.M., 1992. A New Class of Corner Finder. *In 3rd British Machine Vision Conference*, 22-24 September 1992 Leeds, UK. London: BMVA. 139-148.

SMITH, S.M. AND BRADY, J.M., 1995. ASSET-2: Real-Time Motion Segmentation and Shape Tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8), 814-820.

SONKA, M., HLAVAC, V. AND BOYLE, R., 1993. *Image Processing, Analysis & Machine Vision*. Cambridge, UK: Thomson Computer Press.

STILER, C. AND SUNTRUP, R., 1992. Parametric Object Motion Estimation. *In Communications on the Move*, 16-20 November 1992 Singapore. Singapore: IEEE. 633-637.

STREHL, A. AND AGGARWAL, J.K., 2000. A New Bayesian Relaxation Framework for the Estimation and Segmentation of Multiple Motions. *In 4th IEEE South West Symposium on Image Analysis and Interpretation*. 2-4 April 2000 Austin, USA. Texas, USA: IEEE. 21-25.

STROUD, K.A., 2001. *Engineering Mathematics*, (5th edition) London: Palgrave Macmillan.

SUMPTER, N. AND BULPITT, A.J., 1998. Learning Spatio-Temporal Patterns for Predicting Object Behaviour. *In 9th British Machine Vision Conference*, 14-17 September 1998 Southampton, UK. Southampton: BMVA. 649-658.

TAN, T.N., SULLIVAN, G.D. AND BAKER, K.D., 1994. Fast Vehicle Localisation and Recognition without Line Extraction and Matching. *In 5th British Machine Vision Conference*, 13-16 September 1994 York, UK. York: BMVA. 85-94.

TEAL, M.K. AND ELLIS, T.J., 1995. Target tracking in open world scenes using motion cues and target dynamics. *In 5th International Conference on Image Processing and its Applications*. 4-6 July 1995 place. London: IEE. 276-280.

TEAL, M. K., 1997. *Target Tracking and Image Interpretation in Natural Open World Scenes*. Thesis (PhD) City University.

TEAL, M.K. AND SMITH, A.A.W., 1999. Learning Structural Change for Identification and Tracking of Vehicles Moving in Open World Scenes. *In 7th International Conference on Image Processing and its Applications*, 13-15 July 1999 Manchester, UK.

TECHMER, A., 2001. Contour-Based Motion Estimation and Object Tracking for Real-Time Applications. *In IEEE International Conference on Image Processing*. 7-10 October 2001 Thessaloniki, Greece. Texas, USA: IEEE. 648-651.

TU Z. AND ZHU S-C., 2002. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE transactions on Pattern Analysis & Machine Intelligence*, 24(5), 657-673.

VASCONCELOS, G.C., ADEODATO, P.J.L., CARVALHO FILHO, E.C.B., SODRE EDE, A. AND CAVALCANTI, G.D., 1999. Neural Network Based Automatic Target Processing and Recognition. *In 7th IEE International Conference on Image Processing and its Applications*, 13-15 July 1999 Manchester, UK. London: IEE. 280-284.

VOLES, P., TEAL, M.K. AND SANDERSON, J., 1999. Target Identification in a Complex Maritime Scene. *In IEE Colloquium on Motion Analysis and Tracking*, 10 May 1999 London, UK. London: IEE. 15/1-15/4.

VOLES, P., SMITH, A.A.W. AND TEAL, M.K., 2000. Nautical Scene Segmentation using Variable Size Image Windows and Feature Space Reclustering. *In 6th European Conference on Computer Vision*. 26 June-1 July 2000 Dublin, Ireland. Dublin: Springer. 324-335.

WANG, Y, DOHERTY, J.F. AND VAN DYCK, R.E., 2000. Moving Object Tracking in Video. *In 29th Applied Imagery & Pattern Recognition Workshop*. 16-18 October 2000 Washington DC, USA. Washington DC, USA: IEEE. 95-101.

WELCH, G. AND BISHOP, G., 2001. An introduction to the Kalman Filter [online]. University of North Carolina. Available from: www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf [Accessed 2 August 2005].

WELCH, G. AND BISHOP, G., 2004. An introduction to the Kalman Filter [online]. University of North Carolina. Available from: www.cs.unc.edu/~welch/kalman/kalman_filter/kalman.html [Accessed 10 July 2004].

WORRALL, A. D., MARSLIN, R.F. AND SULLIVAN, G.D., 1991. Model-based Tracking. *In 2nd British Machine Vision Conference*, 24-26 September 1991 Glasgow, UK. London: BMVA. 310-318.

WORRALL, A.D., SULLIVAN, G.D. AND BAKER, K.D., 1993. Advances in Model-Based Traffic Vision. *In 4th British Machine Vision Conference*, 21-23 September 1993 Surrey, UK. Surrey: BMVA. 559-568.

WHYTE, A. (1998) *Vessel Factsheets* [online]. Southampton, Red Funnel Group. Available from: www.redfunnelgroup.com/redfunnel/travel/pdfs/VehicleFerries.pdf [Accessed 12 September 2001].

WHYTE, A. (1999) *RedJet Factsheet* [online]. Southampton, Red Funnel Group. Available from: www.redfunnelgroup.com/redfunnel/travel/pdfs/RedJet.pdf [Accessed 12 September 2001].

XU, M. AND ELLIS, T.J., 2002. Partial Observation vs. Blind Tracking through Occlusion. *In 13th British Machine Vision Conference*, 2-5 September 2002 Cardiff UK. Cardiff: BMVA. 777-786.

YIN, L., ASTOLA, J. AND NEUVO, Y., 1993. A New Class of Nonlinear Filters - Neural Filters. *IEEE Transactions on Signal Processing*, 41(3), 1201-1222.

YOSHIMURA, M. AND OE S., 1999. Evolutionary segmentation of texture image using genetic algorithms towards automatic decision of optimum number of segmentation areas. *Pattern Recognition*, 32(12), 2041-2054.

ZHANG, X-P. AND DESAI, M.D., 2001. Segmentation of Bright Targets Using Wavelets and Adaptive Thresholding. *IEEE transactions on Image Processing*, 10(7), 1020-1030.

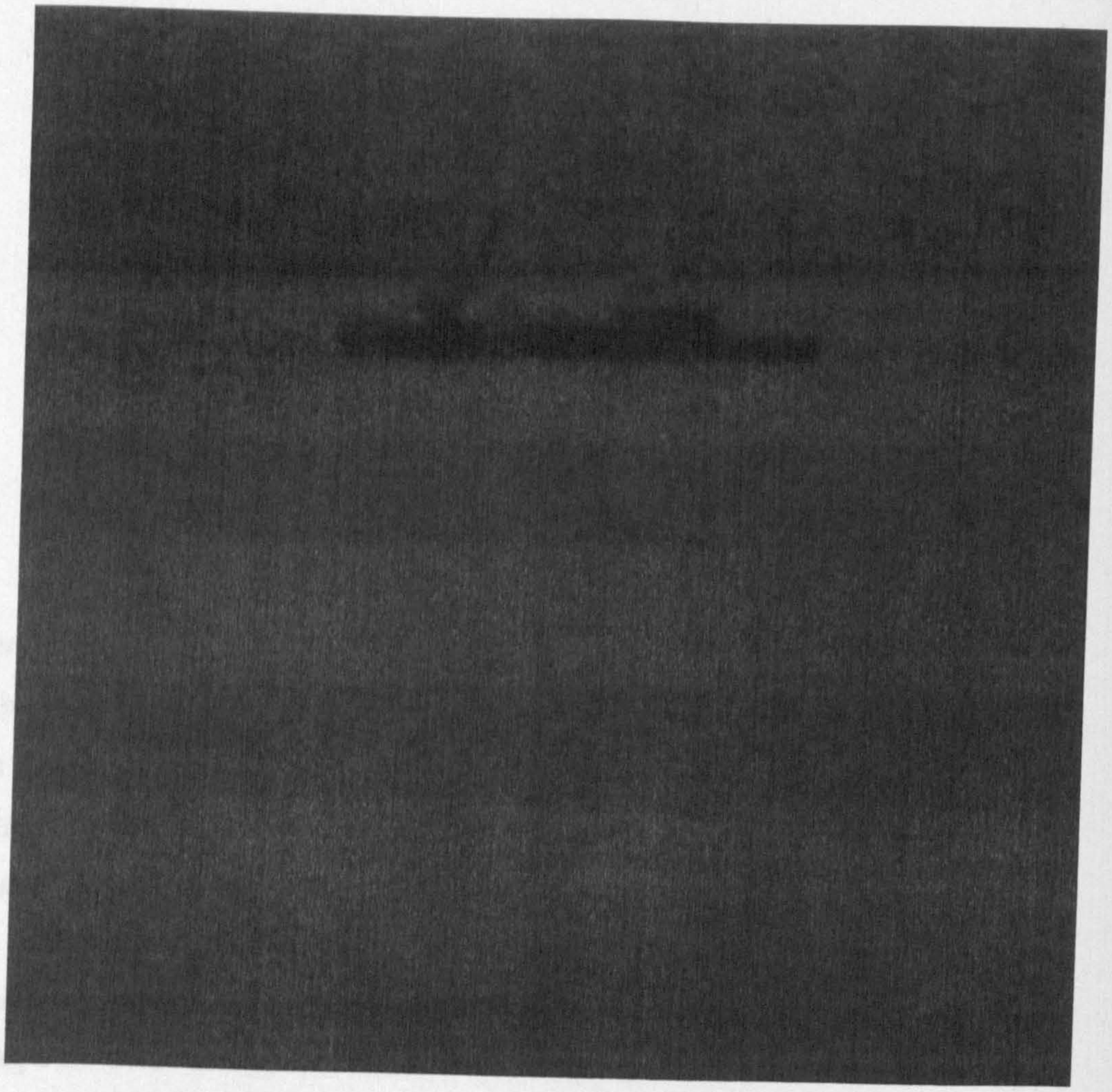

Figure A1 – Poole sequence, frame 1

Table A1 - Tile rejection value evaluation for Poole sequence, frame 1

Threshold Value	Pane	Actual Sea Range		Calculated Sea Range		%Error	
		min	max	min	max	min	max
15	0	65	97	70	95	7.692308	2.061856
	1	45	95	52	81	15.555556	14.736842
	2	60	95	66	91	10.000000	4.210526
	3	45	93	50	82	11.111111	11.827957
20	0	65	97	70	95	7.692308	2.061856
	1	45	95	37	93	17.777778	2.105263
	2	60	95	66	91	10.000000	4.210526
	3	45	93	50	91	11.111111	2.150538
25	0	65	97	70	95	7.692308	2.061856
	1	45	95	37	93	17.777778	2.105263
	2	60	95	66	91	10.000000	4.210526
	3	45	93	37	91	17.777778	2.150538
30	0	65	97	70	95	7.692308	2.061856
	1	45	95	37	93	17.777778	2.105263
	2	60	95	37	94	38.333333	1.052632
	3	45	93	37	91	17.777778	2.150538
35	0	65	97	37	95	43.076923	2.061856
	1	45	95	37	91	17.777778	4.210526
	2	60	95	37	91	38.333333	4.210526
	3	45	93	37	91	17.777778	2.150538
40	0	65	97	37	95	43.076923	2.061856
	1	45	95	37	93	17.777778	2.105263
	2	60	95	37	91	38.333333	4.210526
	3	45	93	37	91	17.777778	2.150538

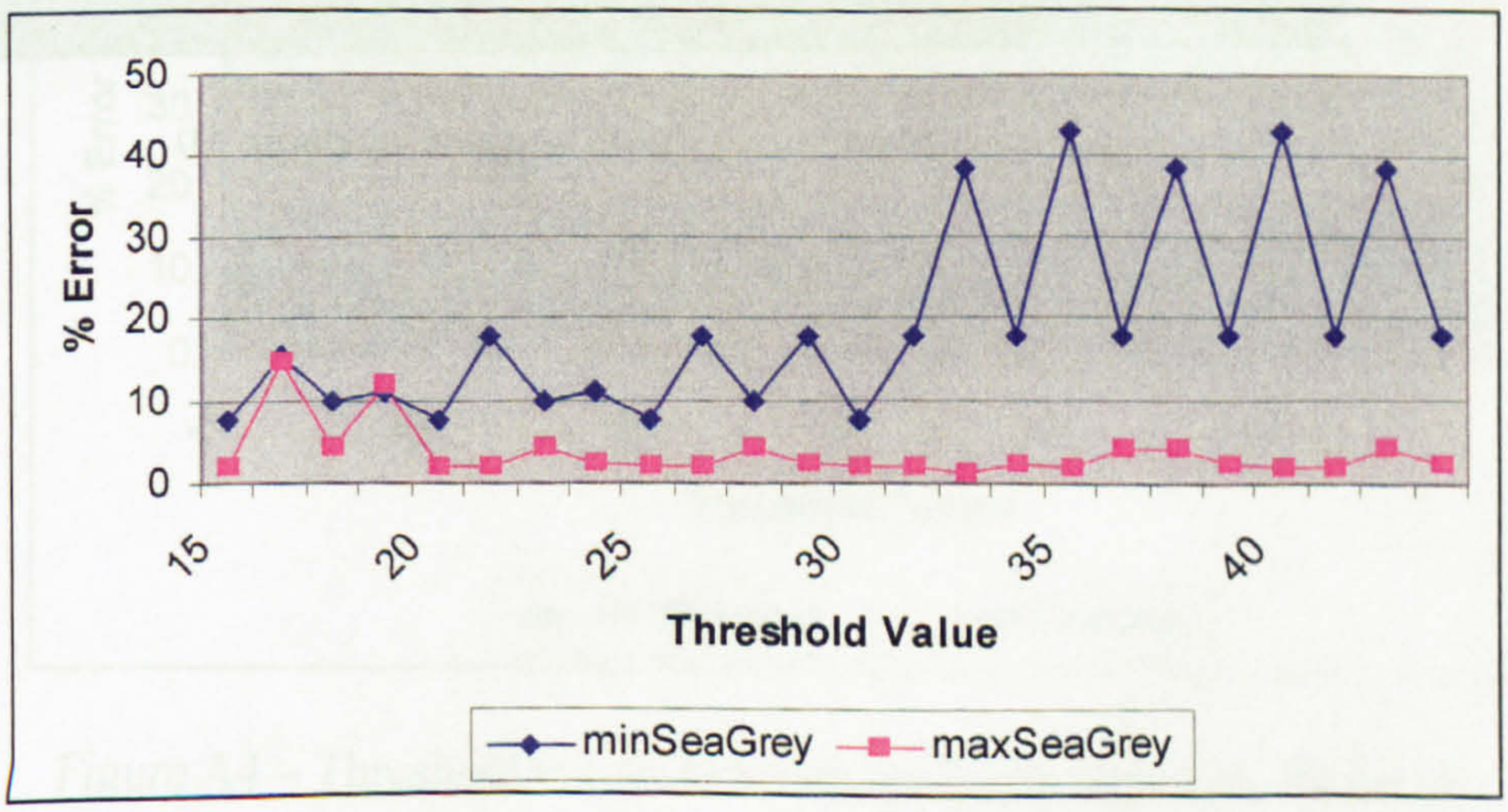

Figure A2 – Error rate for each pane at each threshold value

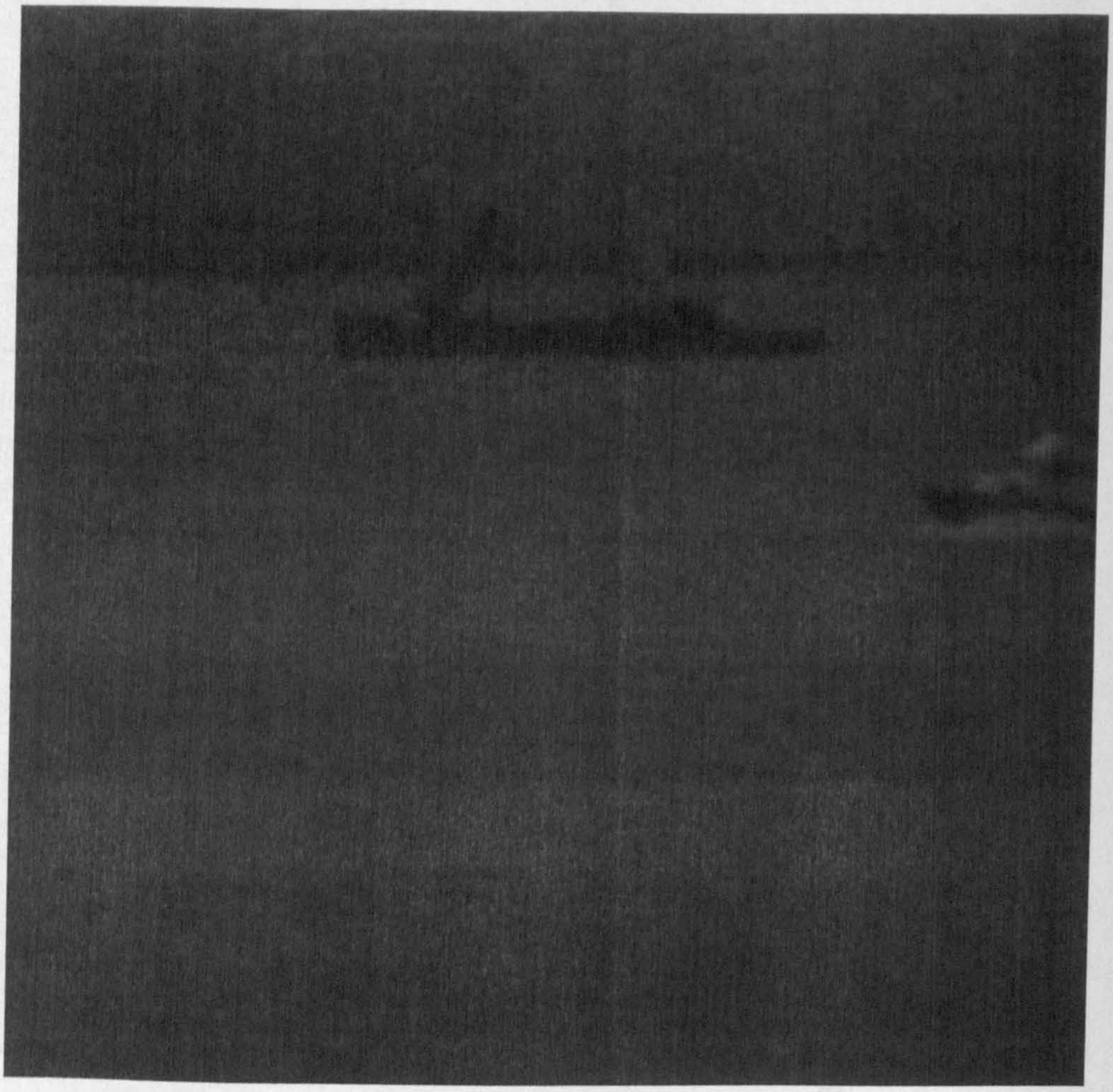

Figure A3 – Poole sequence, frame 10

Table A2 – Tile rejection value evaluation for Poole sequence, frame 10

Threshold Value	Pane	Actual Sea Range		Calculated Sea Range		%Error	
		min	max	min	max	min	max
15	0	64	96	69	97	7.812500	1.041667
	1	47	96	52	84	10.638298	12.500000
	2	60	94	65	93	8.333333	1.063830
	3	47	91	49	79	4.255319	13.186813
20	0	64	96	69	97	7.812500	1.041667
	1	47	96	36	91	23.404255	5.208333
	2	60	94	65	93	8.333333	1.063830
	3	47	91	49	90	4.255319	1.098901
25	0	64	96	69	97	7.812500	1.041667
	1	47	96	36	91	23.404255	5.208333
	2	60	94	65	93	8.333333	1.063830
	3	47	91	36	90	23.404255	1.098901
30	0	64	96	69	97	7.812500	1.041667
	1	47	96	36	91	23.404255	5.208333
	2	60	94	65	93	8.333333	1.063830
	3	47	91	36	90	23.404255	1.098901
35	0	64	96	36	97	43.750000	1.041667
	1	47	96	36	91	23.404255	5.208333
	2	60	94	36	93	40.000000	1.063830
	3	47	91	36	90	23.404255	1.098901
40	0	64	96	36	97	43.750000	1.041667
	1	47	96	36	91	23.404255	5.208333
	2	60	94	36	93	40.000000	1.063830
	3	47	91	36	90	23.404255	1.098901

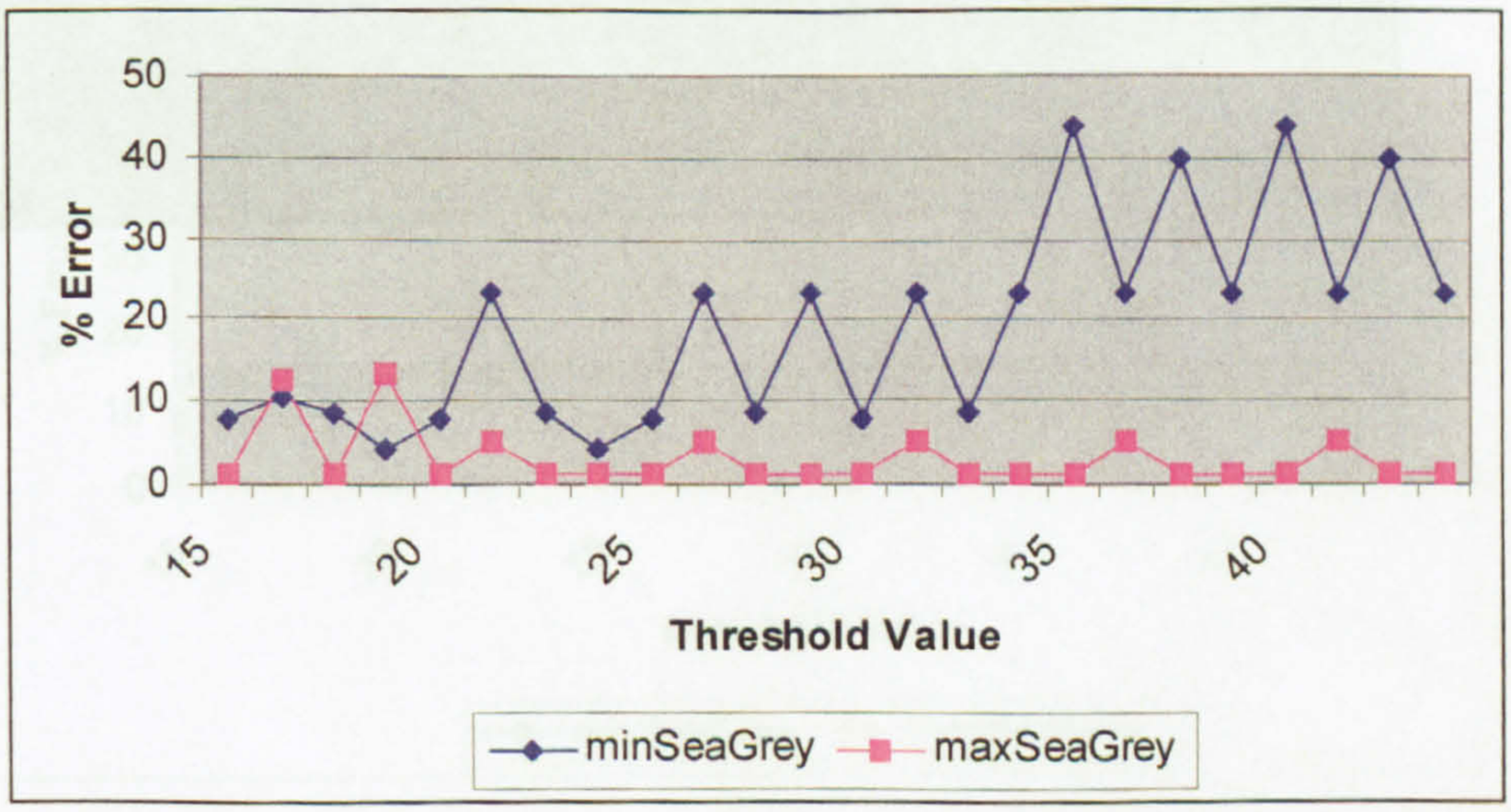

Figure A4 – Threshold value error rate for Poole sequence, frame 10

Figure A5 – Dover sequence, frame 1

Table A3 – Tile rejection threshold value evaluation for Dover sequence, frame 1

Threshold Value	Pane	Actual Sea Range		Calculated Sea Range		%Error	
		min	max	min	max	min	max
15	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	139	26.000000	18.235294
	2	45	185	53	140	17.777778	24.324324
	3	40	175	58	143	45.000000	18.285714
20	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	139	26.000000	18.235294
	2	45	185	53	140	17.777778	24.324324
	3	40	175	58	143	45.000000	18.285714
25	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	151	26.000000	11.176471
	2	45	185	53	153	17.777778	17.297297
	3	40	175	58	143	45.000000	18.285714
30	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	151	26.000000	11.176471
	2	45	185	53	153	17.777778	17.297297
	3	40	175	58	143	45.000000	18.285714
35	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	151	26.000000	11.176471
	2	45	185	53	153	17.777778	17.297297
	3	40	175	58	143	45.000000	18.285714
40	0	45	180	55	149	22.222222	17.222222
	1	50	170	63	151	26.000000	11.176471
	2	45	185	53	153	17.777778	17.297297
	3	40	175	58	143	45.000000	18.285714

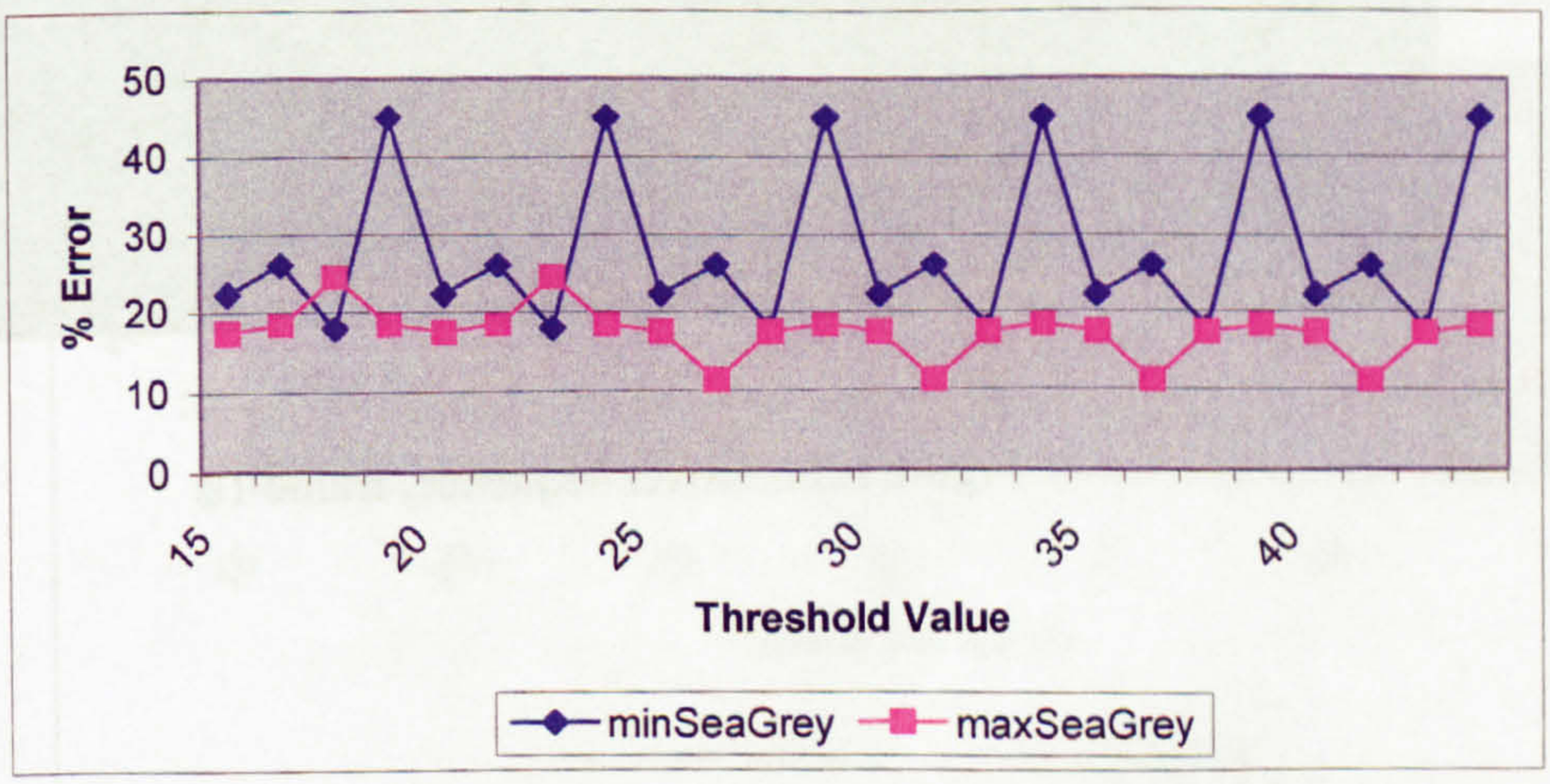

Figure A6 – Threshold value error rate for Dover sequence, frame 1

Figure A7 – Dover sequence, frame 10

Table A4 - Tile rejection threshold value evaluation for Dover sequence, frame 10

Threshold Value	Pane	Actual Sea Range		Calculated Sea Range		%Error	
		min	max	min	max	min	max
15	0	45	170	57	135	26.666667	20.588235
	1	50	170	64	135	28.000000	20.588235
	2	40	190	57	136	42.500000	28.421053
	3	45	170	63	148	40.000000	12.941176
20	0	45	170	57	150	26.666667	11.764706
	1	50	170	64	147	28.000000	13.529412
	2	40	190	57	152	42.500000	20.000000
	3	45	170	51	148	13.333333	12.941176
25	0	45	170	57	158	26.666667	7.058824
	1	50	170	64	147	28.000000	13.529412
	2	40	190	47	152	17.500000	20.000000
	3	45	170	51	148	13.333333	12.941176
30	0	45	170	57	158	26.666667	7.058824
	1	50	170	64	147	28.000000	13.529412
	2	40	190	47	152	17.500000	20.000000
	3	45	170	51	148	13.333333	12.941176
35	0	45	170	57	158	26.666667	7.058824
	1	50	170	64	147	28.000000	13.529412
	2	40	190	47	152	17.500000	20.000000
	3	45	170	51	148	13.333333	12.941176
40	0	45	170	57	158	26.666667	7.058824
	1	50	170	64	147	28.000000	13.529412
	2	40	190	47	152	17.500000	20.000000
	3	45	170	51	148	13.333333	12.941176

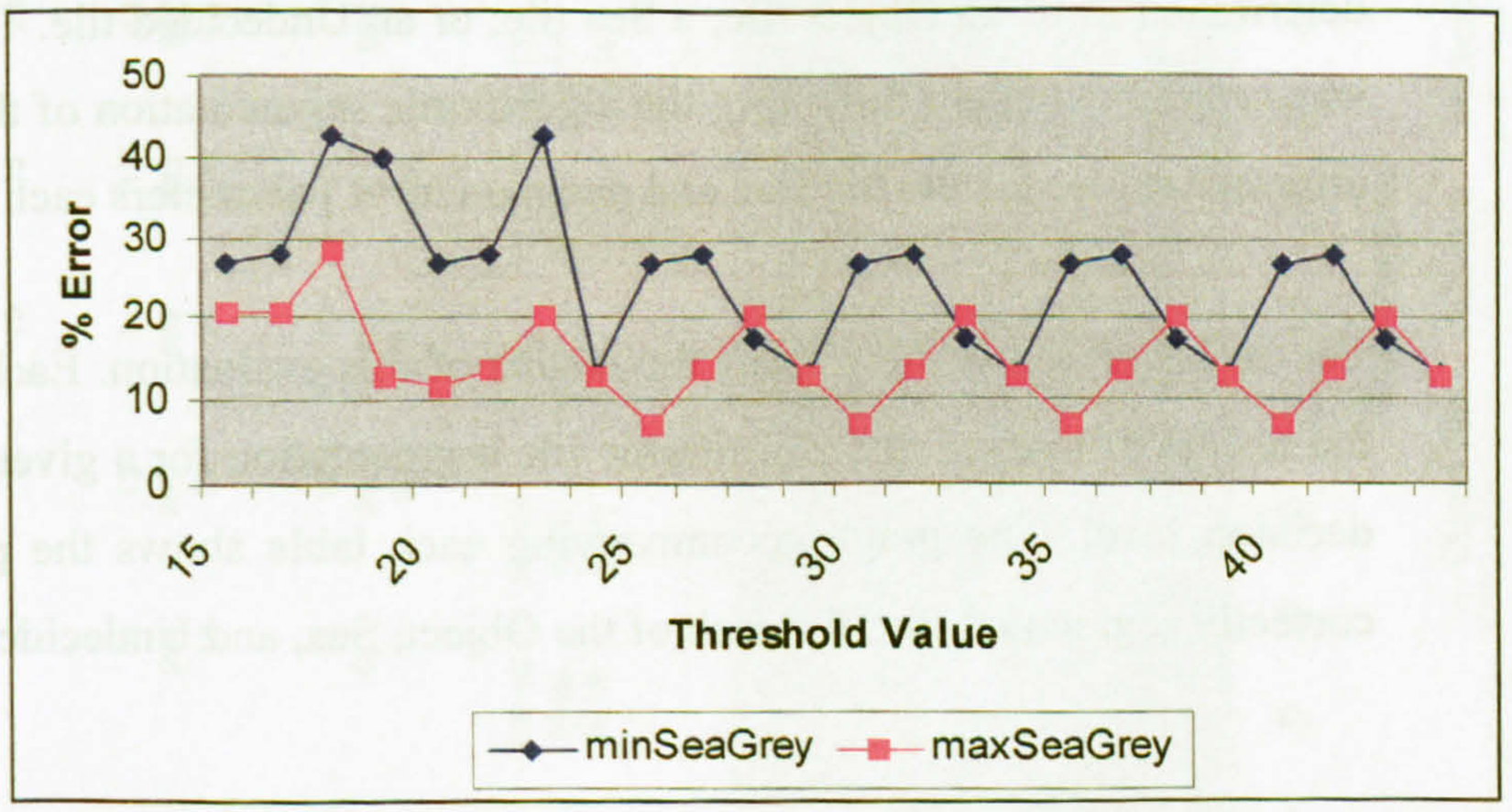

Figure A8 – Threshold value error rate for Dover sequence, frame 10

Appendix B: Segmentation Tile Size and Decision Level

This appendix shows the results of the determination of the tile size and decision level used in the segmentation process. For every fifth frame of a 100 frame sequence from the Poole and Dover test sequences, a grid of tile size being considered was placed over it. Each tile in the grid was then manually determined to be an Object tile; a Sea tile; or an Undecided tile. This procedure was carried out again following the algorithmic segmentation of the image with different values for the tile size and decision level parameters each time.

The tables presented here show the results of this evaluation. Each table shows the results of manual and algorithmic tile segmentation for a given tile size and decision level. The graph accompanying each table shows the percentage of correctly segmented tiles for each of the Object, Sea, and Undecided categories.

Table B1 - Poole sequence, tile size 16x16, decision level 5%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	12	12	14	17	17	19	16	18	18	22	19	21	20	18
	Sea tiles	994	993	989	983	982	972	972	973	970	975	975	973	974	981
	Undecided	18	19	21	24	25	33	36	33	36	27	30	30	30	25
Segmented Image Description	Object tiles	12	12	10	11	11	12	11	8	13	15	16	9	16	18
	Sea tiles	961	957	971	971	966	937	949	950	930	938	932	950	934	915
	Undecided	51	55	43	42	47	75	64	66	81	71	76	65	74	91
Results in %															
	Object tiles	100.00	100.00	71.43	64.71	64.71	63.16	68.75	44.44	72.22	68.18	84.21	42.86	80.00	100.00
	Sea tiles	96.68	96.37	98.18	98.78	98.37	96.40	97.63	97.64	95.88	96.21	95.59	97.64	95.89	93.27
	Undecided	35.29	34.55	48.84	57.14	53.19	44.00	56.25	50.00	44.44	38.03	39.47	46.15	40.54	27.47

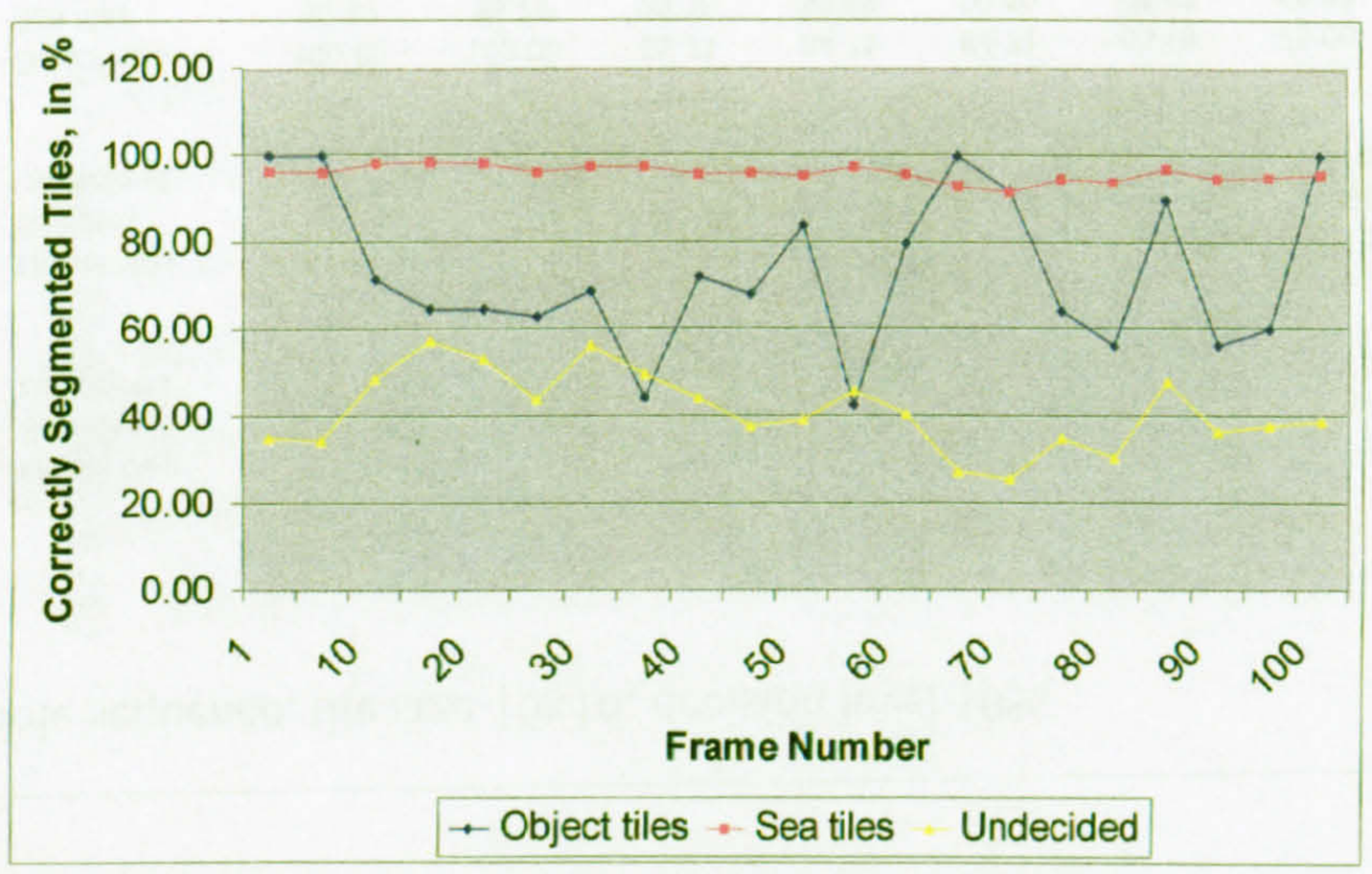

Figure B1 – Poole sequence, tile size 16x16, decision level 5%

Table B2 - Poole sequence, tile size 16x16, decision level 10%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	12	12	14	17	17	19	16	18	18	22	19	21	20	18
	Sea tiles	994	993	989	983	982	972	972	973	970	975	975	973	974	981
	Undecided	18	19	21	24	25	33	36	33	36	27	30	30	30	25
Segmented Image Description	Object tiles	12	12	12	11	11	12	12	11	13	17	17	10	21	20
	Sea tiles	977	981	981	979	978	961	965	972	966	955	949	970	948	941
	Undecided	35	31	31	34	35	51	47	41	45	52	58	44	55	63
Results in %															
	Object tiles	100.00	100.00	85.71	64.71	64.71	63.16	75.00	61.11	72.22	77.27	89.47	47.62	95.24	90.00
	Sea tiles	98.29	98.79	99.19	99.59	99.59	98.87	99.28	99.90	99.59	97.95	97.33	99.69	97.33	95.92
	Undecided	51.43	61.29	67.74	70.59	71.43	64.71	76.60	80.49	80.00	51.92	51.72	68.18	54.55	39.68

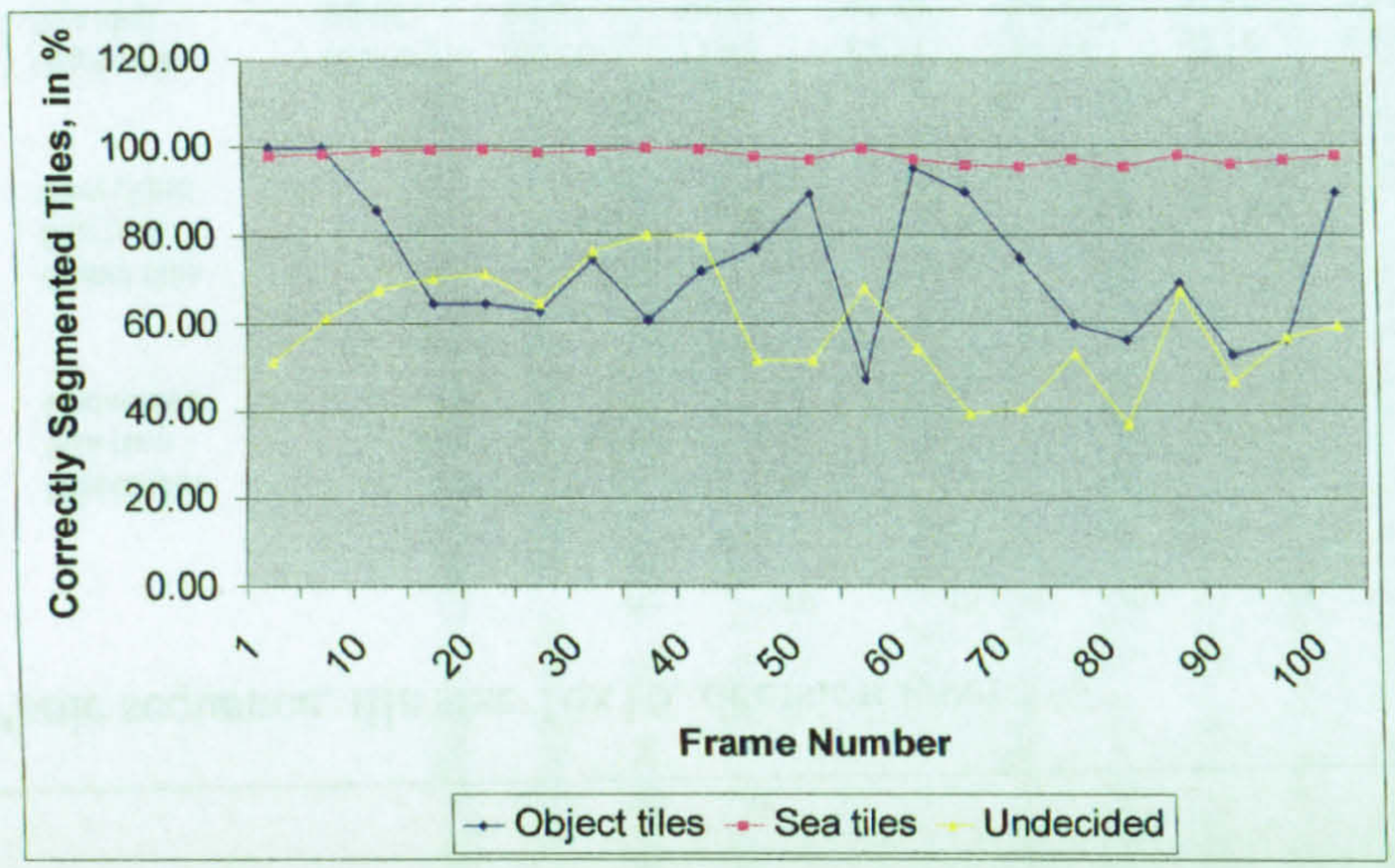

Figure B2 – Poole sequence, tile size 16x16, decision level 10%

Table B3 - Poole sequence, tile size 16x16, decision level 15%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	12	12	14	17	17	19	16	18	18	22	19	21	20	18
	Sea tiles	994	993	989	983	982	972	972	973	970	975	975	973	974	981
	Undecided	18	19	21	24	25	33	36	33	36	27	30	30	30	25
Segmented Image Description	Object tiles	12	12	12	12	12	13	12	12	14	19	19	10	26	22
	Sea tiles	986	988	989	986	984	971	977	982	980	967	959	980	956	952
	Undecided	26	24	23	26	28	40	35	30	30	38	46	34	42	50
Results in %															
	Object tiles	100.00	100.00	85.71	70.59	70.59	68.42	75.00	66.67	77.78	86.36	100.00	47.62	76.92	81.82
	Sea tiles	99.20	99.50	100.00	99.70	99.80	99.90	99.49	99.08	98.98	99.18	98.36	99.29	98.15	97.04
	Undecided	69.23	79.17	91.30	92.31	89.29	82.50	97.22	90.91	83.33	71.05	65.22	88.24	71.43	50.00

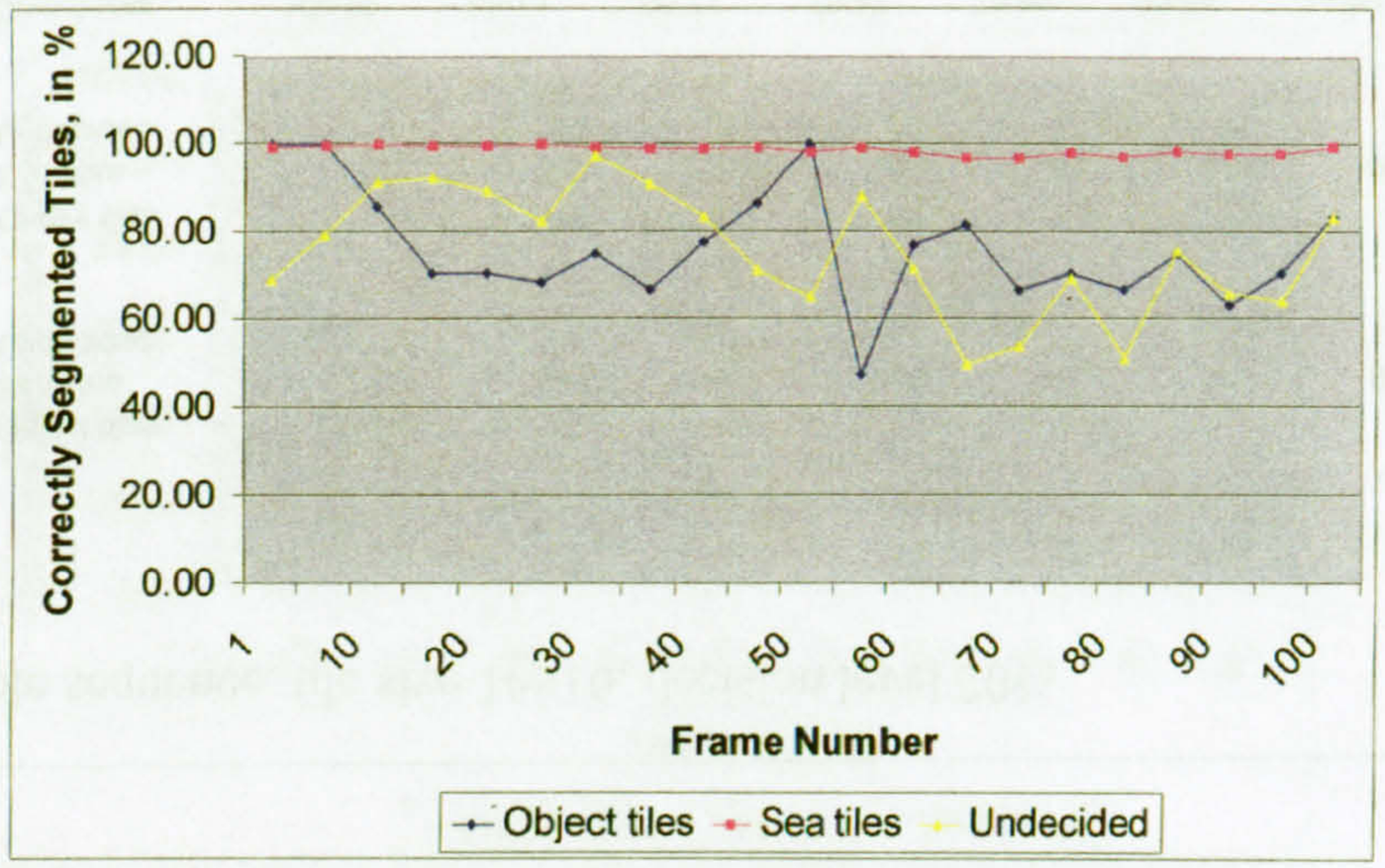

Figure B3 – Poole sequence, tile size 16x16, decision level 15%

Table B4 - Poole sequence, tile size 16x16, decision level 20%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	12	12	14	17	17	19	16	18	18	22	19	21	20	18
	Sea tiles	994	993	989	983	982	972	972	973	970	975	975	973	974	981
	Undecided	18	19	21	24	25	33	36	33	36	27	30	30	30	25
Segmented Image Description	Object tiles	12	14	12	13	12	14	12	12	15	20	21	10	26	25
	Sea tiles	991	991	993	989	988	979	983	984	984	971	965	982	964	958
	Undecided	21	19	19	22	24	31	29	28	25	33	38	32	34	41
Results in %															
	Object tiles	100.00	85.71	85.71	76.47	70.59	73.68	75.00	66.67	83.33	90.91	90.48	47.62	76.92	72.00
	Sea tiles	99.70	99.80	99.60	99.39	99.39	99.28	98.88	98.88	98.58	99.59	98.97	99.08	98.97	97.66
	Undecided	85.71	100.00	90.48	91.67	96.00	93.94	80.56	84.85	69.44	81.82	78.95	93.75	88.24	60.98

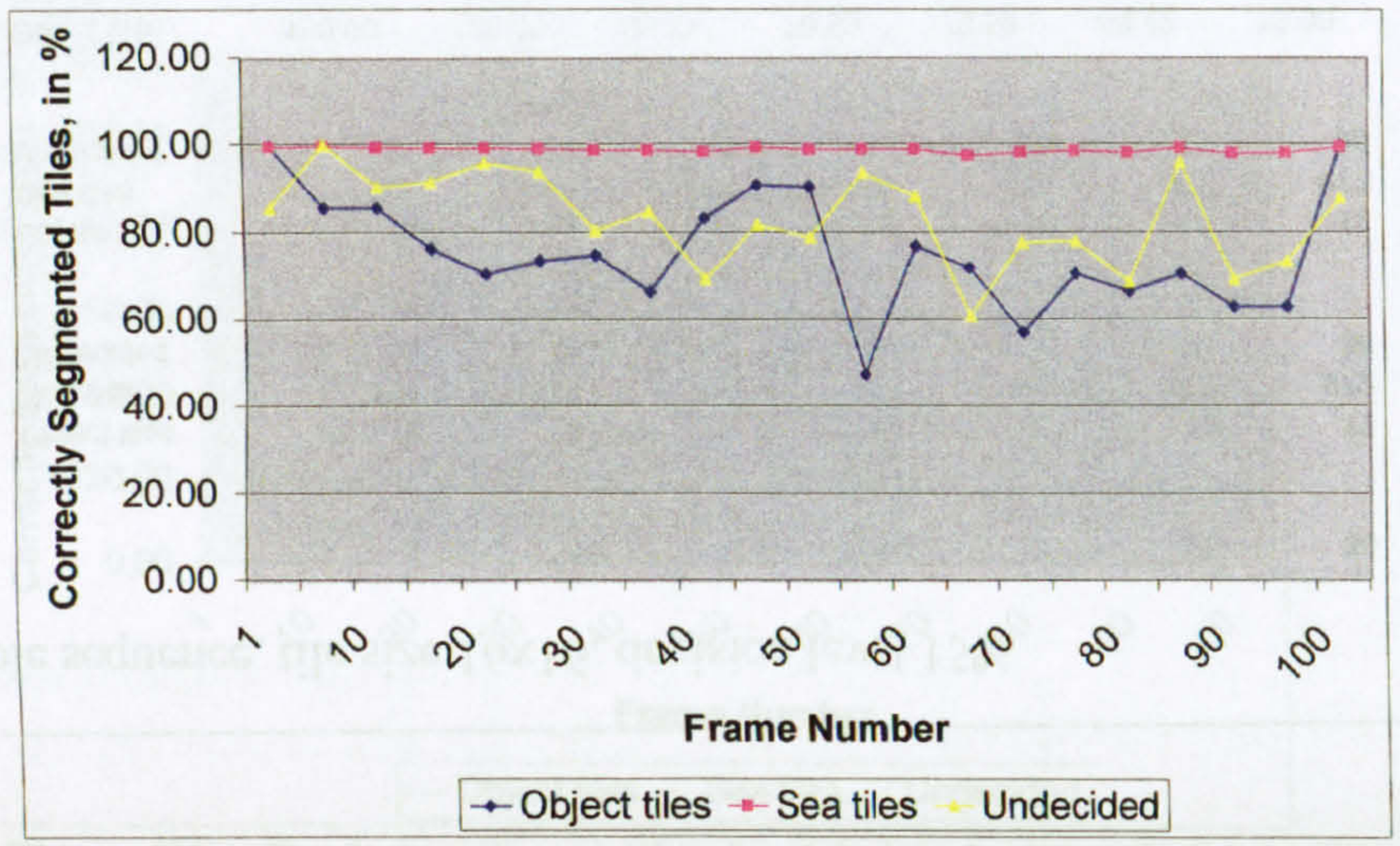

Figure B4 – Poole sequence, tile size 16x16, decision level 20%

Table B5 - Dover sequence, tile size 16x16, decision level 5%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	50	49	50	50	50	50	52	50	51	52	53	52	55	56
	Sea tiles	939	938	938	938	938	939	938	940	940	941	955	938	937	939
	Undecided	35	37	36	36	36	35	34	34	33	31	16	34	32	29
Segmented Image Description	Object tiles	62	62	61	48	48	48	51	50	61	52	62	63	56	56
	Sea tiles	801	775	843	863	853	864	852	844	819	891	817	844	848	833
	Undecided	161	187	120	113	123	112	121	130	144	81	145	117	120	135
Results in %															
	Object tiles	80.65	79.03	81.97	96.00	96.00	96.00	98.08	100.00	83.61	100.00	85.48	82.54	98.21	100.00
	Sea tiles	85.30	82.62	89.87	92.00	90.94	92.01	90.83	89.79	87.13	94.69	85.55	89.98	90.50	88.71
	Undecided	21.74	19.79	30.00	31.86	29.27	31.25	28.10	26.15	22.92	38.27	11.03	29.06	26.67	21.48

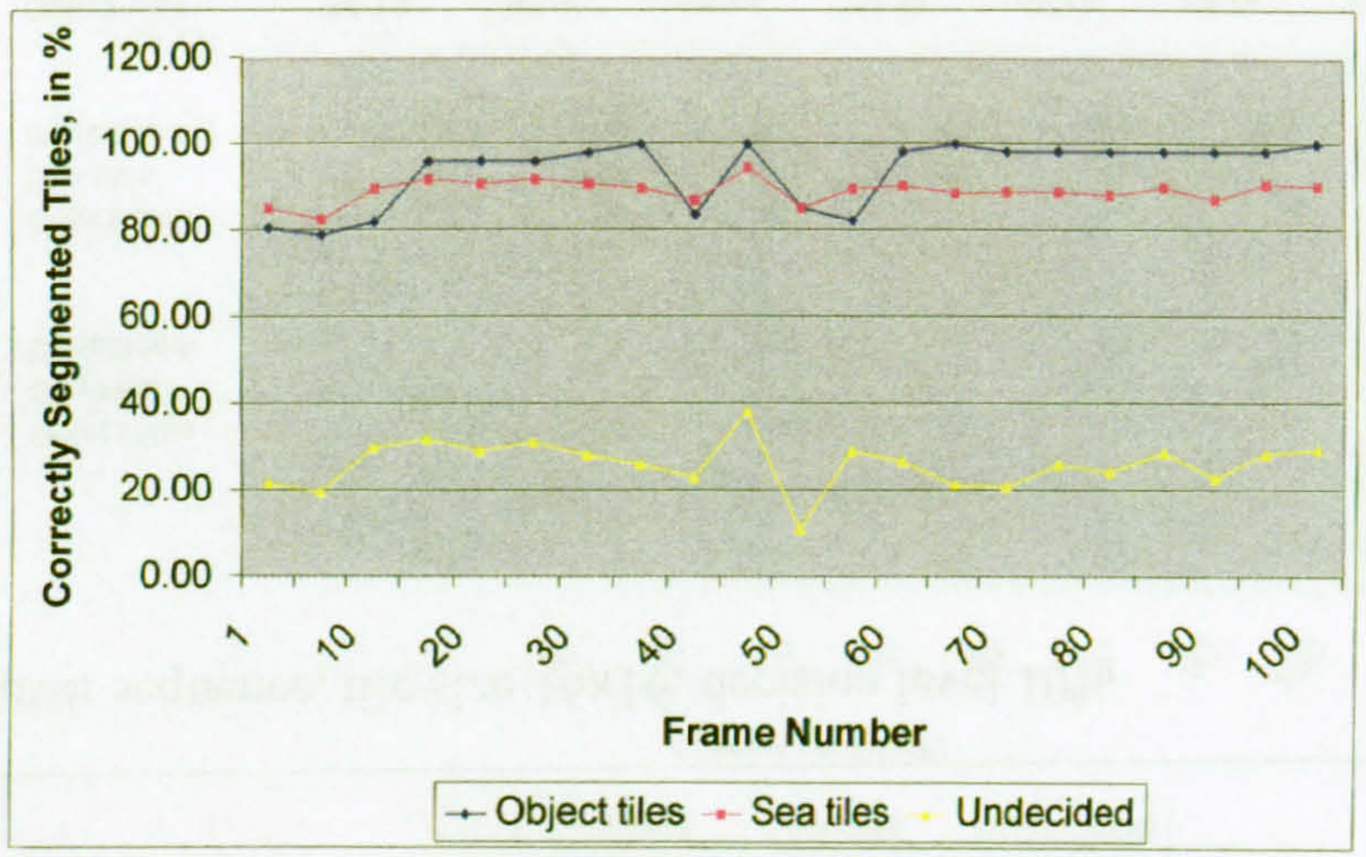

Figure B5 –Dover sequence, tile size 16x16, decision level 5%

Table B6 - Dover sequence, tile size 16x16, decision level 10%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	53	52	52	52	52	52	53	52	53	54	55	55	57	58
	Sea tiles	934	933	934	934	934	935	935	936	936	937	931	933	934	935
	Undecided	37	39	38	38	38	37	36	36	35	33	38	36	33	31
Segmented Image Description	Object tiles	63	65	63	49	50	50	52	51	63	52	64	64	58	57
	Sea tiles	886	879	910	922	911	922	915	914	907	924	890	893	906	902
	Undecided	75	80	51	53	63	52	57	59	54	48	70	67	60	65
Results in %															
	Object tiles	84.13	80.00	82.54	94.23	96.15	96.15	98.11	98.08	84.13	96.30	85.94	85.94	98.28	98.28
	Sea tiles	94.86	94.21	97.43	98.72	97.54	98.61	97.86	97.65	96.90	98.61	95.60	95.71	97.00	96.47
	Undecided	49.33	48.75	74.51	71.70	60.32	71.15	63.16	61.02	64.81	68.75	54.29	53.73	55.00	47.69

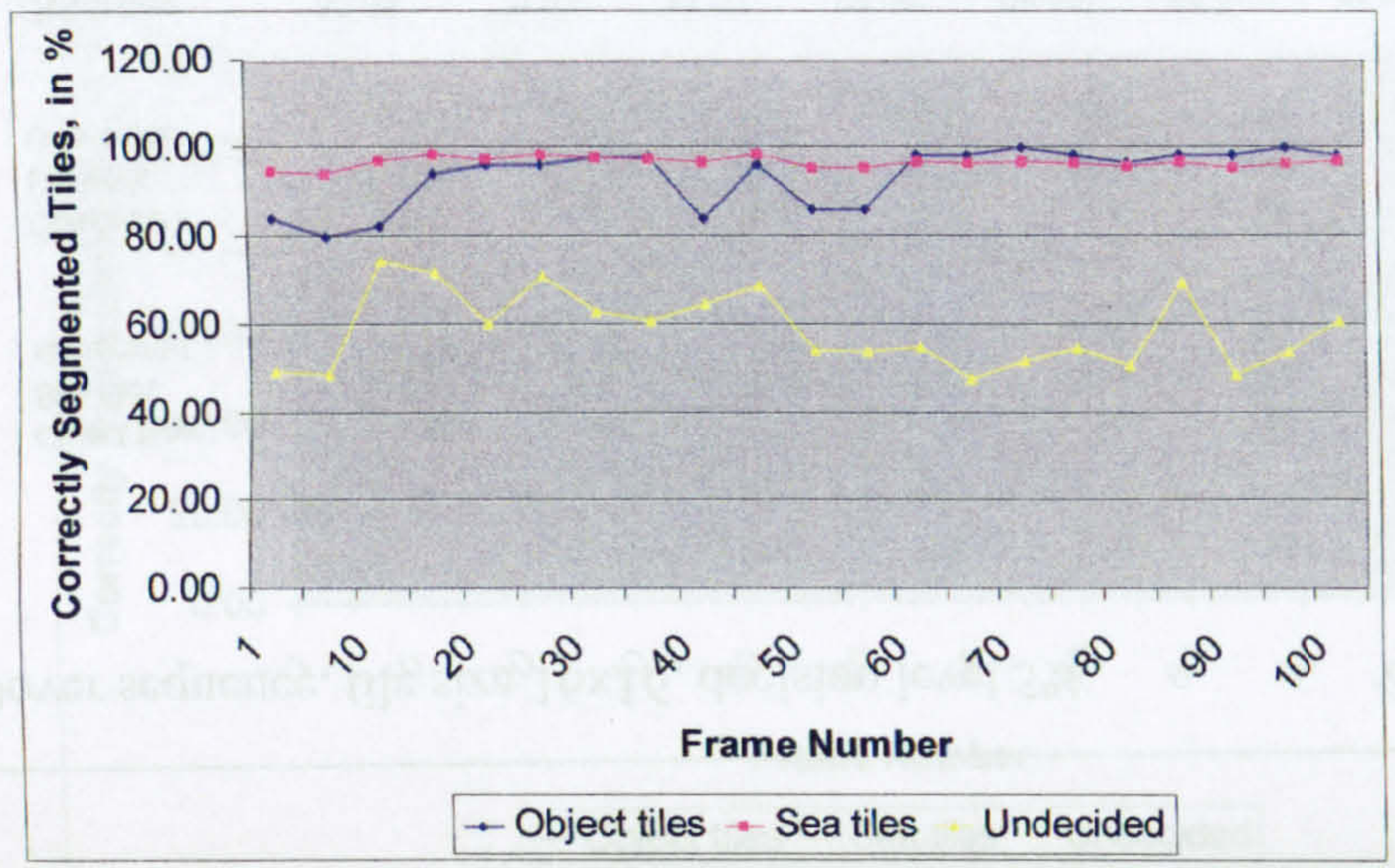

Figure B6 – Dover sequence, tile size 16x16, decision level 10%

Table B7 - Dover sequence, tile size 16x16, decision level 15%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	53	52	52	52	52	52	53	52	53	54	55	55	57	58
	Sea tiles	934	933	934	934	934	935	935	936	936	937	931	933	934	935
	Undecided	37	39	38	38	38	37	36	36	35	33	38	36	33	31
Segmented Image Description	Object tiles	66	66	63	52	53	54	55	55	64	55	66	65	58	60
	Sea tiles	919	918	929	936	926	933	931	929	926	938	912	911	921	928
	Undecided	39	40	32	36	45	37	38	40	34	31	46	48	45	36
Results in %															
	Object tiles	80.30	78.79	82.54	100.00	98.11	96.30	96.36	94.55	82.81	98.18	83.33	84.62	98.28	96.67
	Sea tiles	98.39	98.39	99.46	99.79	99.14	99.79	99.57	99.25	98.93	99.89	97.96	97.64	98.61	99.25
	Undecided	94.87	97.50	84.21	94.74	84.44	100.00	94.74	90.00	97.14	93.94	82.61	75.00	73.33	86.11

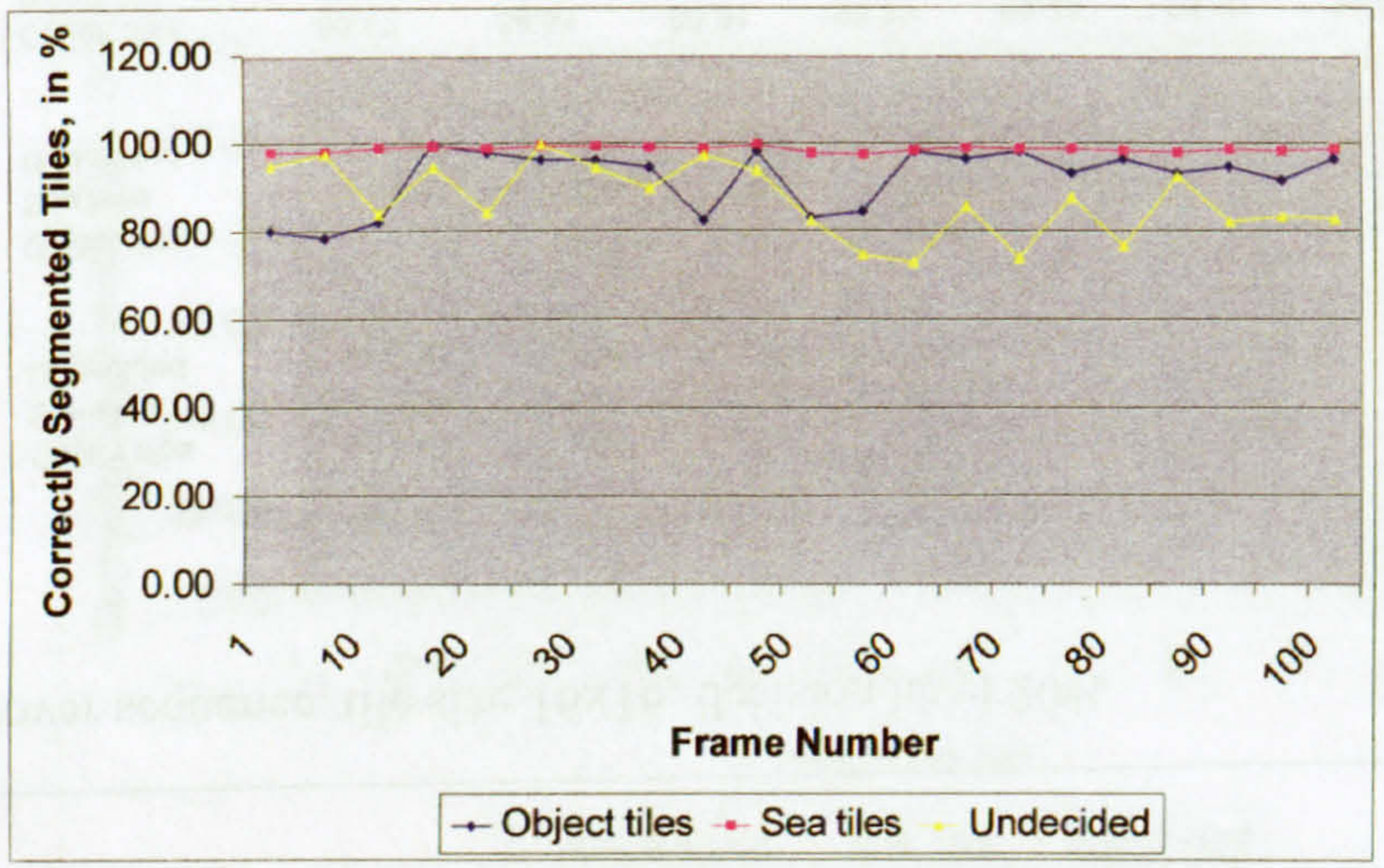

Figure B7 – Dover sequence, tile size 16x16, decision level 15%

Table B8 - Dover sequence, tile size 16x16, decision level 20%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	61	60	60	60	60	60	61	61	62	62	63	60	61	63
	Sea tiles	926	925	926	930	930	931	931	931	931	933	927	932	934	934
	Undecided	37	39	38	34	34	33	32	32	31	29	34	32	29	27
Segmented Image Description	Object tiles	71	71	66	56	56	57	56	55	69	56	66	67	58	61
	Sea tiles	931	928	934	940	934	939	938	937	932	941	924	920	930	934
	Undecided	22	25	24	28	34	28	30	32	23	27	34	39	36	29
Results in %															
	Object tiles	85.92	84.51	90.91	93.33	93.33	95.00	91.80	90.16	89.86	90.32	95.45	89.55	95.08	96.83
	Sea tiles	99.46	99.68	99.14	98.94	99.57	99.15	99.25	99.36	99.89	99.15	99.68	98.71	99.57	100.00
	Undecided	59.46	64.10	63.16	82.35	100.00	84.85	93.75	100.00	74.19	93.10	100.00	82.05	80.56	93.10

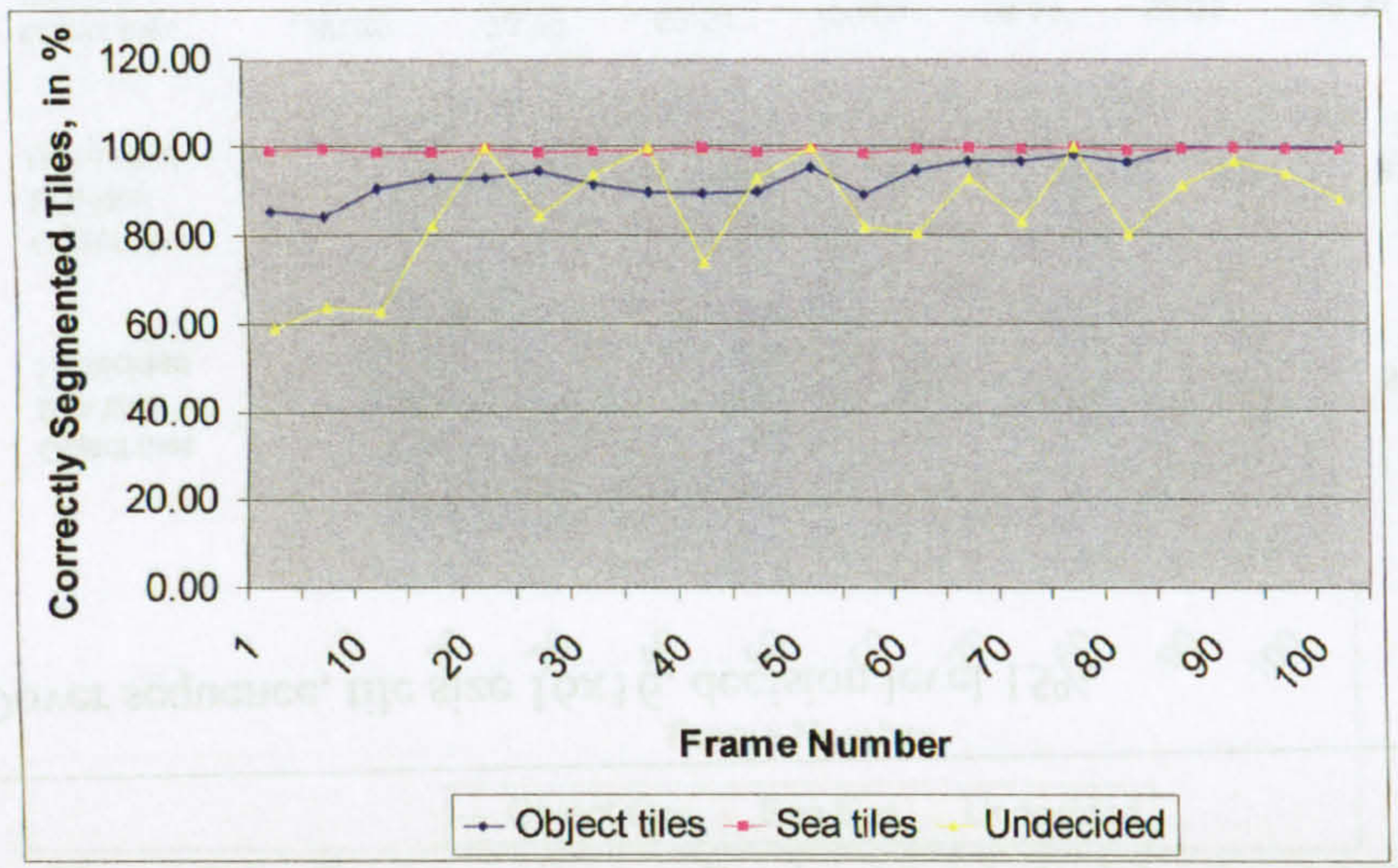

Figure B8 – Dover sequence, tile size 16x16, decision level 20%

Table B9 - Poole sequence, tile size 32x32, decision level 5%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sea tiles	245	245	244	242	240	241	239	240	239	239	240	235	237	239
	Undecided	11	11	12	14	16	15	17	16	17	17	16	21	19	17
Segmented Image Description	Object tiles	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sea tiles	237	238	243	238	237	231	233	237	235	230	227	234	228	227
	Undecided	19	18	13	18	19	25	23	19	21	26	29	22	28	29
Results in %															
	Object tiles	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Sea tiles	96.73	97.14	99.59	98.35	98.75	95.85	97.49	98.75	98.33	96.23	94.58	99.57	96.20	94.98
	Undecided	57.89	61.11	92.31	77.78	84.21	60.00	73.91	84.21	80.95	65.38	55.17	95.45	67.86	58.62

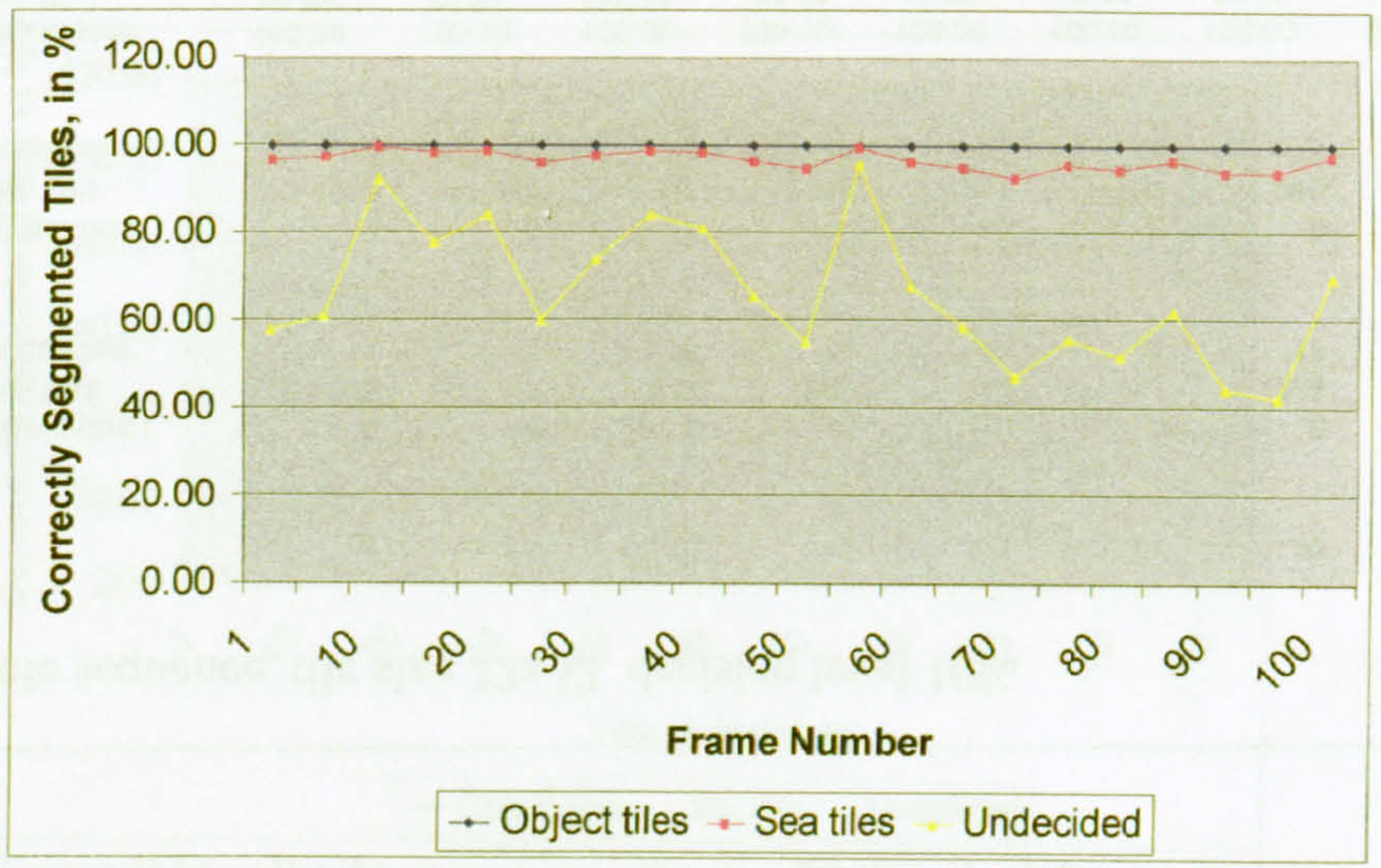

Figure B9 – Poole sequence, tile size 32x32, decision level 5%

Table B10 - Poole sequence, tile size 32x32, decision level 10%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sea tiles	245	245	244	242	240	241	239	240	239	239	240	235	237	239
	Undecided	11	11	12	14	16	15	17	16	17	17	16	21	19	17
Segmented Image Description	Object tiles	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Sea tiles	244	244	244	242	241	238	240	240	241	237	235	239	232	230
	Undecided	12	12	12	14	15	18	16	16	15	19	21	17	24	26
Results in %															
	Object tiles	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Sea tiles	99.59	99.59	100.00	100.00	99.59	98.76	99.58	100.00	99.17	99.16	97.92	98.33	97.89	96.23
	Undecided	91.67	91.67	100.00	100.00	93.75	83.33	94.12	100.00	88.24	89.47	76.19	80.95	79.17	65.38

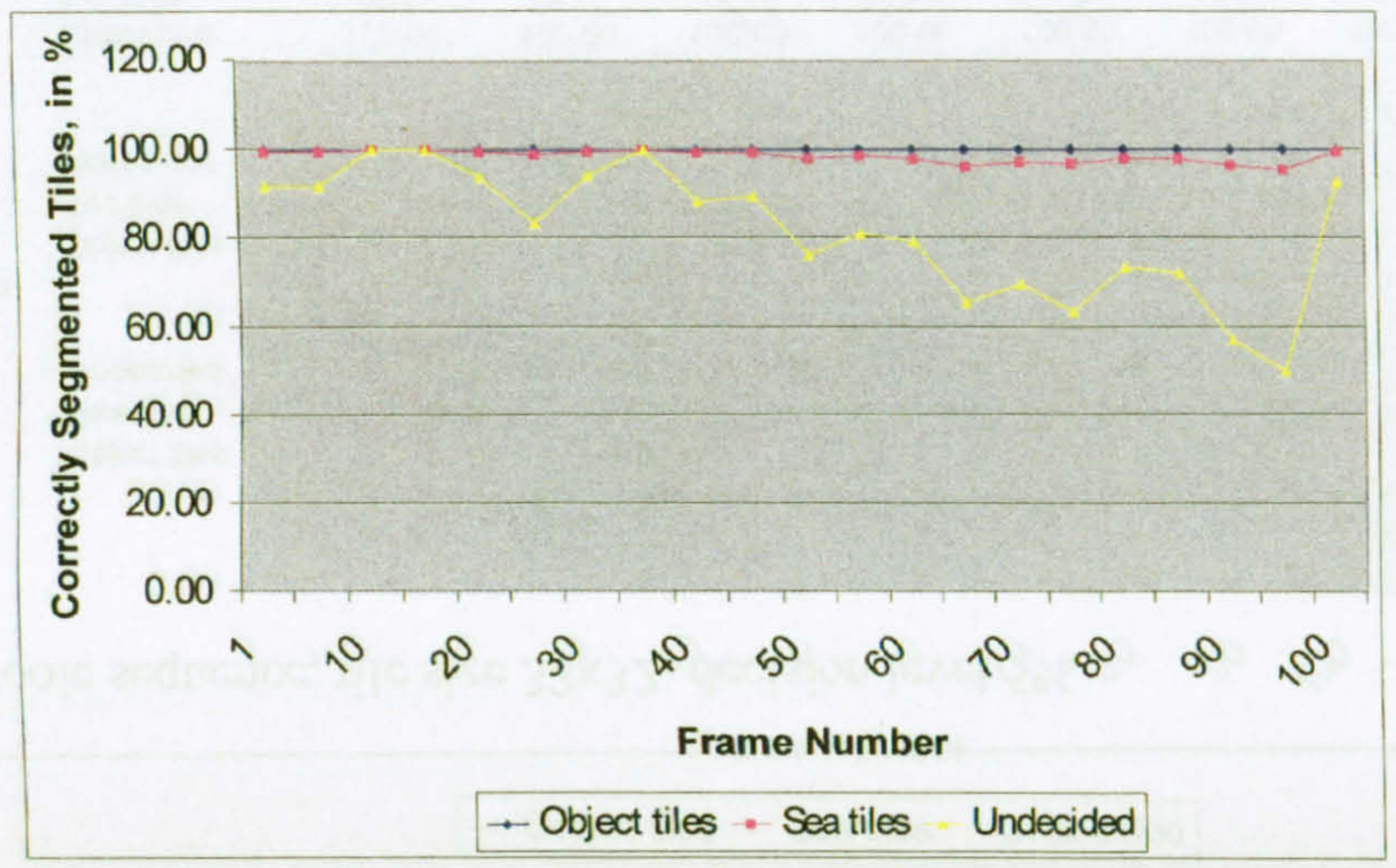

Figure B10 – Poole sequence, tile size 32x32, decision level 10%

Table B11 - Poole sequence, tile size 32x32, decision level 15%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	1	1	1	1	0	1	1	1	1	0	0	0	0	0
	Sea tiles	244	244	243	241	240	240	238	239	238	239	240	235	237	239
	Undecided	11	11	12	14	16	15	17	16	17	17	16	21	19	17
Segmented Image Description	Object tiles	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	Sea tiles	245	244	244	243	243	242	244	243	243	241	238	241	235	231
	Undecided	10	11	11	12	12	13	12	13	13	15	18	15	21	25
Results in %															
	Object tiles	100.00	100.00	100.00	100.00	0.00	100.00	0.00	0.00	0.00	100.00	100.00	100.00	100.00	100.00
	Sea tiles	99.59	100.00	99.59	99.18	98.77	99.17	97.54	98.35	97.94	99.17	99.17	97.51	99.16	96.65
	Undecided	90.91	100.00	91.67	85.71	75.00	86.67	70.59	81.25	76.47	88.24	88.89	71.43	90.48	68.00

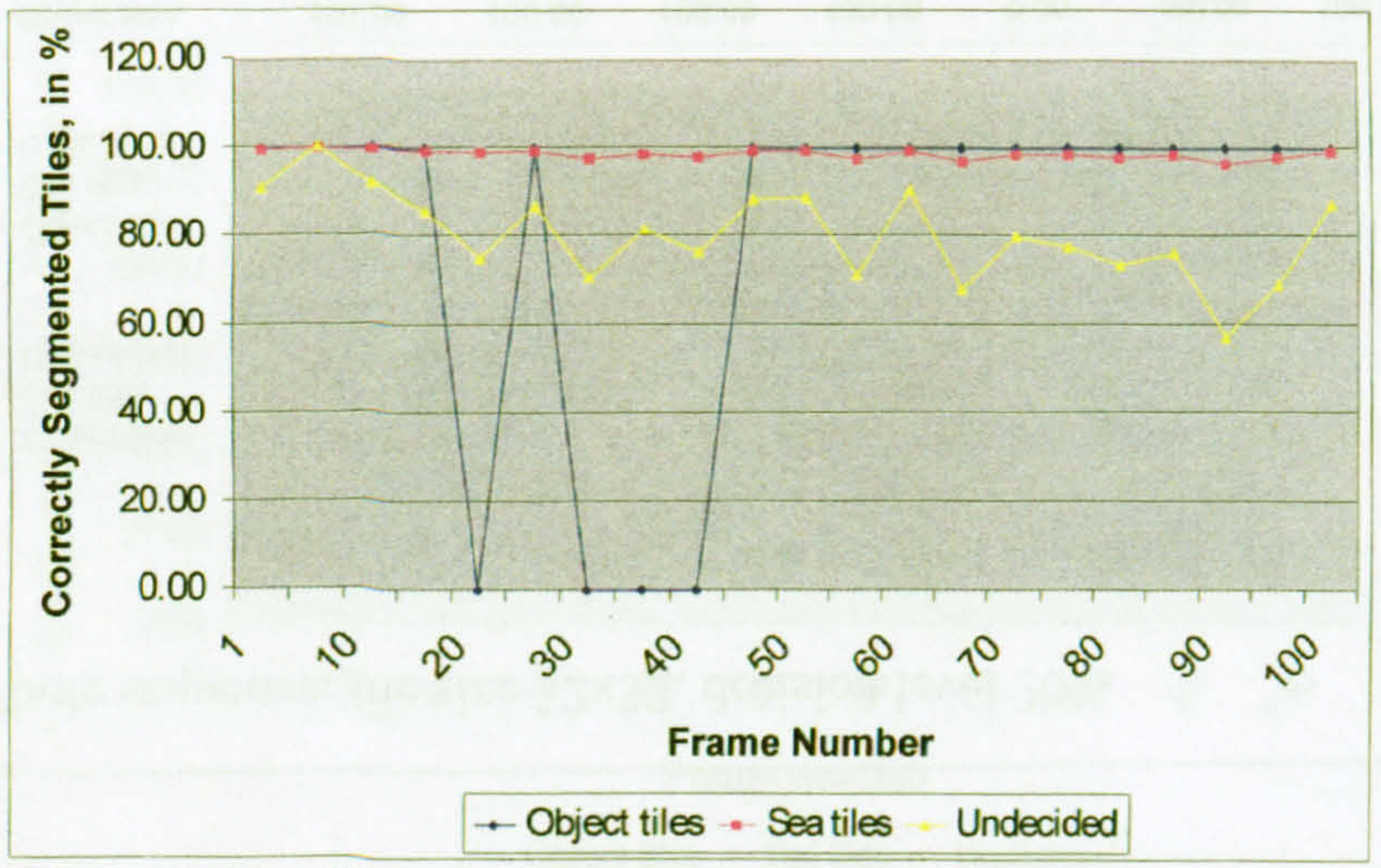

Figure B11 – Poole sequence, tile size 32x32, decision level 15%

Table B12 - Poole sequence, tile size 32x32, decision level 20%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	1	1	1	1	0	1	1	1	1	0	0	0	0	0
	Sea tiles	244	244	243	241	240	240	238	239	238	239	240	235	237	239
	Undecided	11	11	12	14	16	15	17	16	17	17	16	21	19	17
Segmented Image Description	Object tiles	1	1	1	1	1	1	1	1	1	1	2	0	1	1
	Sea tiles	249	249	248	247	248	244	248	246	245	243	239	244	237	239
	Undecided	6	6	7	8	7	11	7	9	10	12	15	12	18	16
Results in %															
	Object tiles	100.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	100.00	0.00	0.00	100.00	0.00	0.00
	Sea tiles	97.99	97.99	97.98	97.57	96.77	98.36	95.97	97.15	97.14	98.35	99.58	96.31	100.00	100.00
	Undecided	54.55	54.55	58.33	57.14	43.75	73.33	41.18	56.25	58.82	70.59	93.75	57.14	94.74	94.12

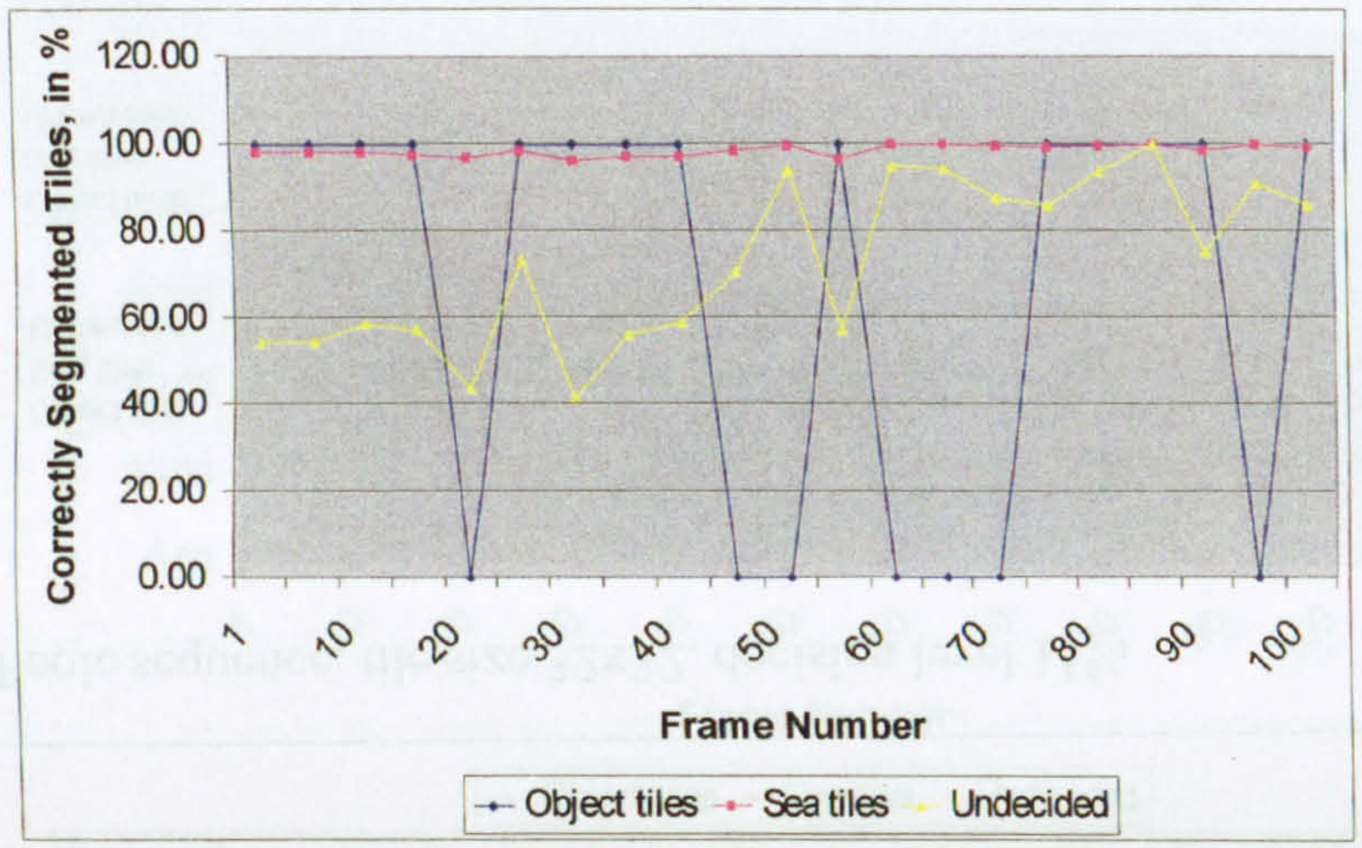

Figure B12 – Poole sequence, tile size 32x32, decision level 20%

Table B13 - Dover sequence, tile size 32x32, decision level 5%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	14	14	14	10	10	10	10	10	11	12	14	14	14	13
	Sea tiles	230	228	228	229	229	229	231	230	228	230	232	229	229	230
	Undecided	12	14	14	17	17	17	15	16	17	14	10	13	13	13
Segmented Image Description	Object tiles	11	10	10	9	10	10	10	10	11	10	13	14	10	10
	Sea tiles	209	205	216	224	218	225	225	220	223	226	213	216	216	216
	Undecided	36	41	30	23	28	21	21	26	22	20	30	26	30	30
Results in %															
	Object tiles	78.57	71.43	71.43	90.00	100.00	100.00	100.00	100.00	100.00	83.33	92.86	100.00	71.43	76.92
	Sea tiles	90.87	89.91	94.74	97.82	95.20	98.25	97.40	95.65	97.81	98.26	91.81	94.32	94.32	93.91
	Undecided	33.33	34.15	46.67	73.91	60.71	80.95	71.43	61.54	77.27	70.00	33.33	50.00	43.33	43.33

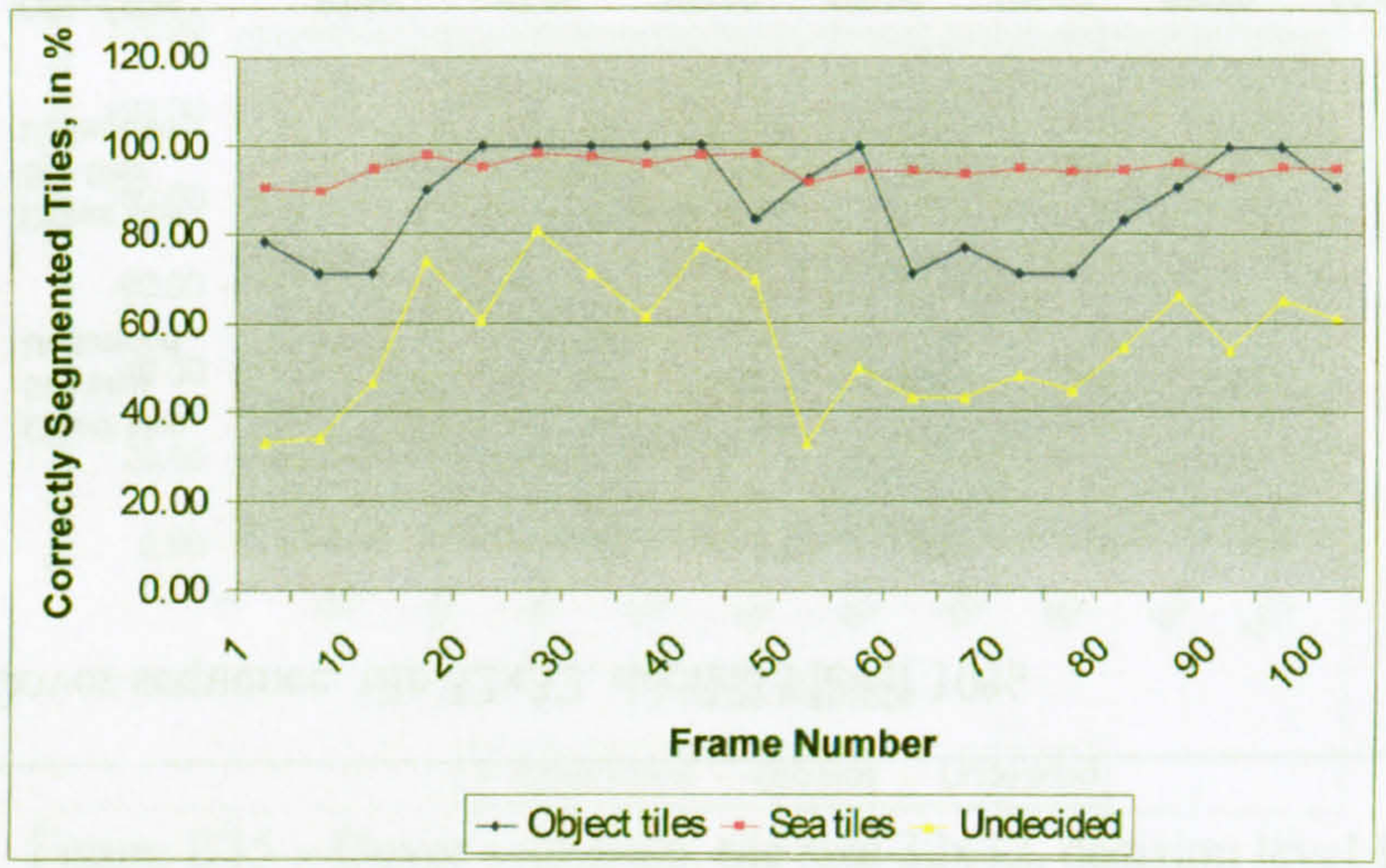

Figure B13 – Dover sequence, tile size 32x32, decision level 5%

Table B14 - Dover sequence, tile 32x32, decision level 10%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	14	14	14	10	10	10	10	10	11	12	14	14	14	13
	Sea tiles	230	228	228	229	229	229	231	230	228	230	232	229	229	230
	Undecided	12	14	14	17	17	17	15	16	17	14	10	13	13	13
Segmented Image Description	Object tiles	13	13	14	10	10	10	11	10	14	11	14	14	13	12
	Sea tiles	228	227	228	228	227	228	228	227	228	228	227	224	224	226
	Undecided	15	16	14	18	19	18	17	19	14	17	15	18	19	18
Results in %															
	Object tiles	92.86	92.86	100.00	100.00	100.00	100.00	90.91	100.00	78.57	91.67	100.00	100.00	92.86	92.31
	Sea tiles	99.13	99.56	100.00	99.56	99.13	99.56	98.70	98.70	100.00	99.13	97.84	97.82	97.82	98.26
	Undecided	80.00	87.50	100.00	94.44	89.47	94.44	88.24	84.21	82.35	82.35	66.67	72.22	68.42	72.22

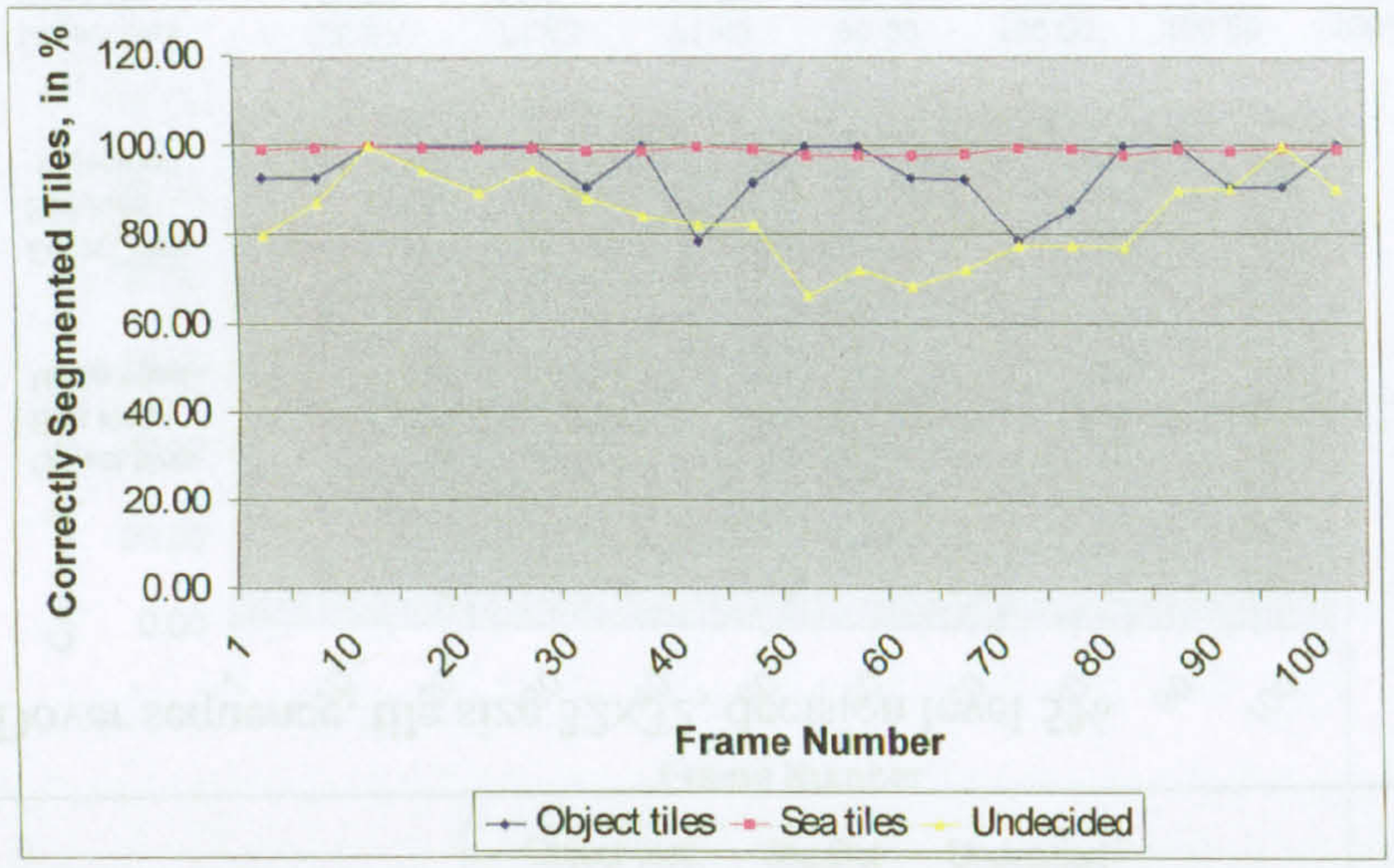

Figure B14 – Dover sequence, tile size 32x32, decision level 10%

Table B15 - Dover sequence, tile size 32x32, decision level 15%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	14	14	14	10	10	10	10	10	11	12	14	14	14	13
	Sea tiles	230	228	228	229	229	229	231	230	228	230	232	229	229	230
	Undecided	12	14	14	17	17	17	15	16	17	14	10	13	13	13
Segmented Image Description	Object tiles	14	14	14	12	13	13	13	13	14	14	14	14	14	14
	Sea tiles	228	228	228	229	229	229	230	229	228	234	228	227	233	227
	Undecided	14	14	14	15	14	14	13	14	14	8	14	15	9	15
Results in %															
	Object tiles	100.00	100.00	100.00	83.33	76.92	76.92	76.92	76.92	78.57	85.71	100.00	100.00	100.00	92.86
	Sea tiles	99.13	100.00	100.00	100.00	100.00	100.00	99.57	99.57	100.00	98.29	98.28	99.13	98.28	98.70
	Undecided	85.71	100.00	100.00	88.24	82.35	82.35	86.67	87.50	82.35	57.14	71.43	86.67	69.23	86.67

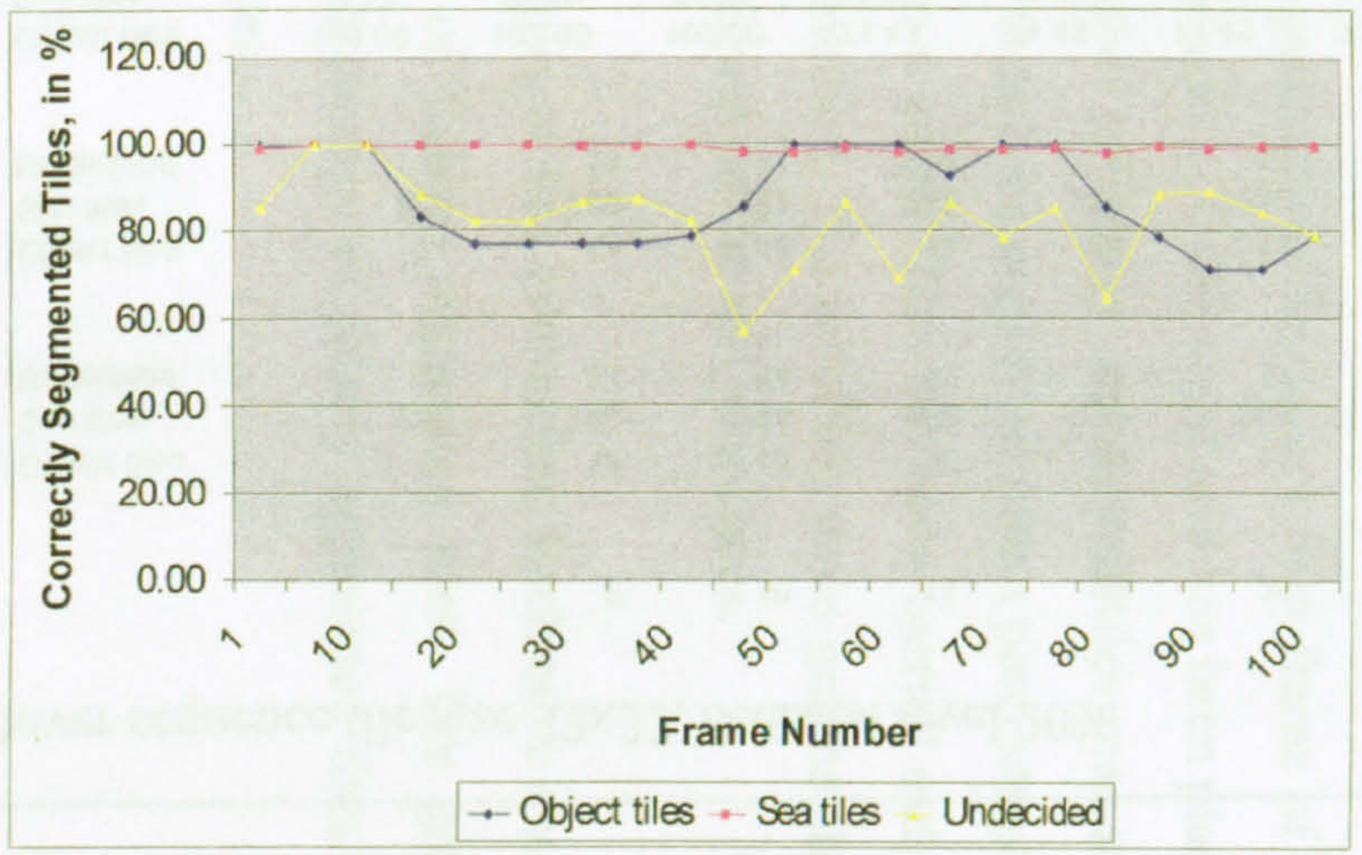

Figure B15 – Dover sequence, tile size 32x32, decision level 15%

Table B16 - Dover sequence tile size 32x32, decision level 20%

Frame		1	5	10	15	20	25	30	35	40	45	50	55	60	65
Original Image Description	Object tiles	14	14	14	10	10	10	10	10	11	12	14	14	14	13
	Sea tiles	230	228	228	229	229	229	231	230	228	230	232	229	229	230
	Undecided	12	14	14	17	17	17	15	16	17	14	10	13	13	13
Segmented Image Description	Object tiles	14	14	14	14	14	14	14	14	14	14	14	14	14	14
	Sea tiles	229	229	231	233	233	232	236	235	230	237	229	228	236	235
	Undecided	13	13	11	9	9	10	6	7	12	5	13	14	6	7
Results in %															
	Object tiles	100.00	100.00	100.00	71.43	71.43	71.43	71.43	71.43	78.57	85.71	100.00	100.00	100.00	92.86
	Sea tiles	99.57	99.56	98.70	98.28	98.28	98.71	97.88	97.87	99.13	97.05	98.71	99.56	97.03	97.87
	Undecided	92.31	92.86	78.57	52.94	52.94	58.82	40.00	43.75	70.59	35.71	76.92	92.86	46.15	53.85

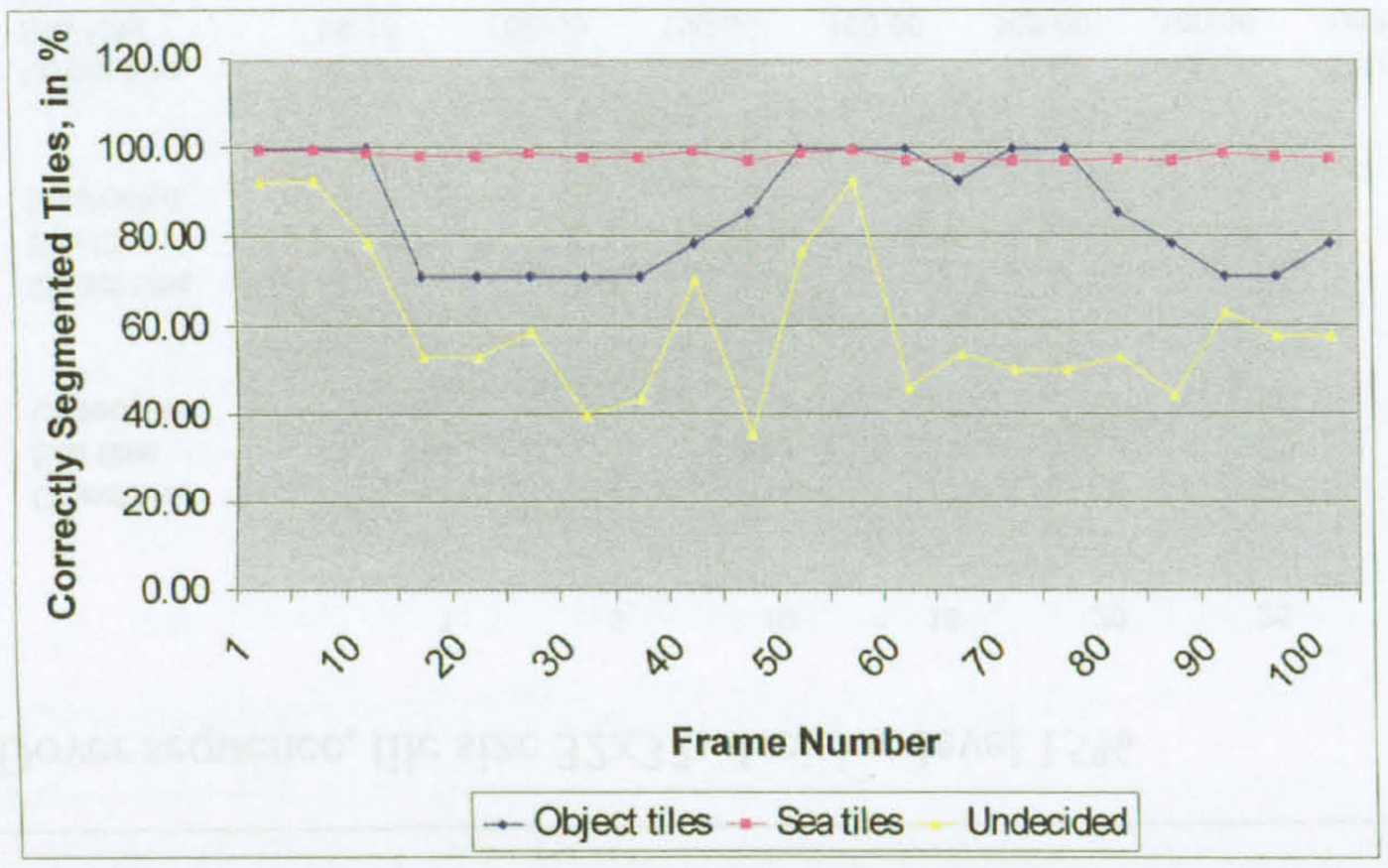

Figure B16 – Dover sequence, tile size 32x32, decision level 20%

Appendix C: Evaluation of Segmentation Techniques

This appendix shows the results of the evaluation of three segmentation methods. The methods used were Thresholding; Frame Differencing; Sea Characterisation and Motion Model Segmentation. For every fifth frame of the Poole, Dover, and Portsmouth test sequences a 32x32 sized grid has been placed over the segmented image and the number of falsely segmented tiles manually recorded. The number of false tiles has then been converted into a percentage of the total number of tiles and plotted on a graph. This has been carried out for each sequence and each segmentation method. The resulting three graphs show the direct comparison of the three techniques on the test sequences.

The tables presented here show the results of this evaluation. Each table shows the results of the manual recording of falsely segmented tiles. The graph accompanying each table shows the percentage of falsely segmented tiles for each of the three segmentation techniques.

Table C1 – Evaluation of the three segmentation methods on the Poole sequence

	Frame	1	5	10	15	20	25	30	35	40	45	50	55	60
		Thresholding												
Segmented Image Description														
FALSE	Object tiles	2	2	6	9	6	7	11	9	11	12	8	1	5
	Sea tiles	0	0	0	0	0	0	0	0	0	0	0	0	0
Results in%														
FALSE	Object tiles	0.78	0.78	2.34	3.52	2.34	2.73	4.30	3.52	4.30	4.69	3.13	0.39	1.95
	Sea tiles	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Frame Differencing												
Segmented Image Description														
FALSE	Object tiles	12	59	12	7	4	3	3	26	35	55	108	80	77
	Sea tiles	1	0	0	0	0	1	1	0	0	0	0	0	0
Results in%														
FALSE	Object tiles	4.69	23.05	4.69	2.73	1.56	1.17	1.17	10.16	13.67	21.48	42.19	31.25	30.08
	Sea tiles	0.39	0.00	0.00	0.00	0.00	0.39	0.39	0.00	0.00	0.00	0.00	0.00	0.00
		Sea Characterisation												
Segmented Image Description														
FALSE	Object tiles	1	0	0	1	0	2	5	9	8	11	7	5	8
	Sea tiles	1	0	1	2	2	1	0	0	0	0	0	0	0
Results in%														
FALSE	Object tiles	0.39	0.00	0.00	0.39	0.00	0.78	1.95	3.52	3.13	4.30	2.73	1.95	3.13
	Sea tiles	0.39	0.00	0.39	0.78	0.78	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average False tile segmentation														
Thresholding		0.78	0.78	2.34	3.52	2.34	2.73	4.30	3.52	4.30	4.69	3.13	0.39	1.95
Frame Differencing		2.54	11.52	2.34	1.37	0.78	0.78	0.78	5.08	6.84	10.74	21.09	15.63	15.04
Sea Characterisation		0.39	0.00	0.20	0.59	0.39	0.59	0.98	1.76	1.56	2.15	1.37	0.98	1.56

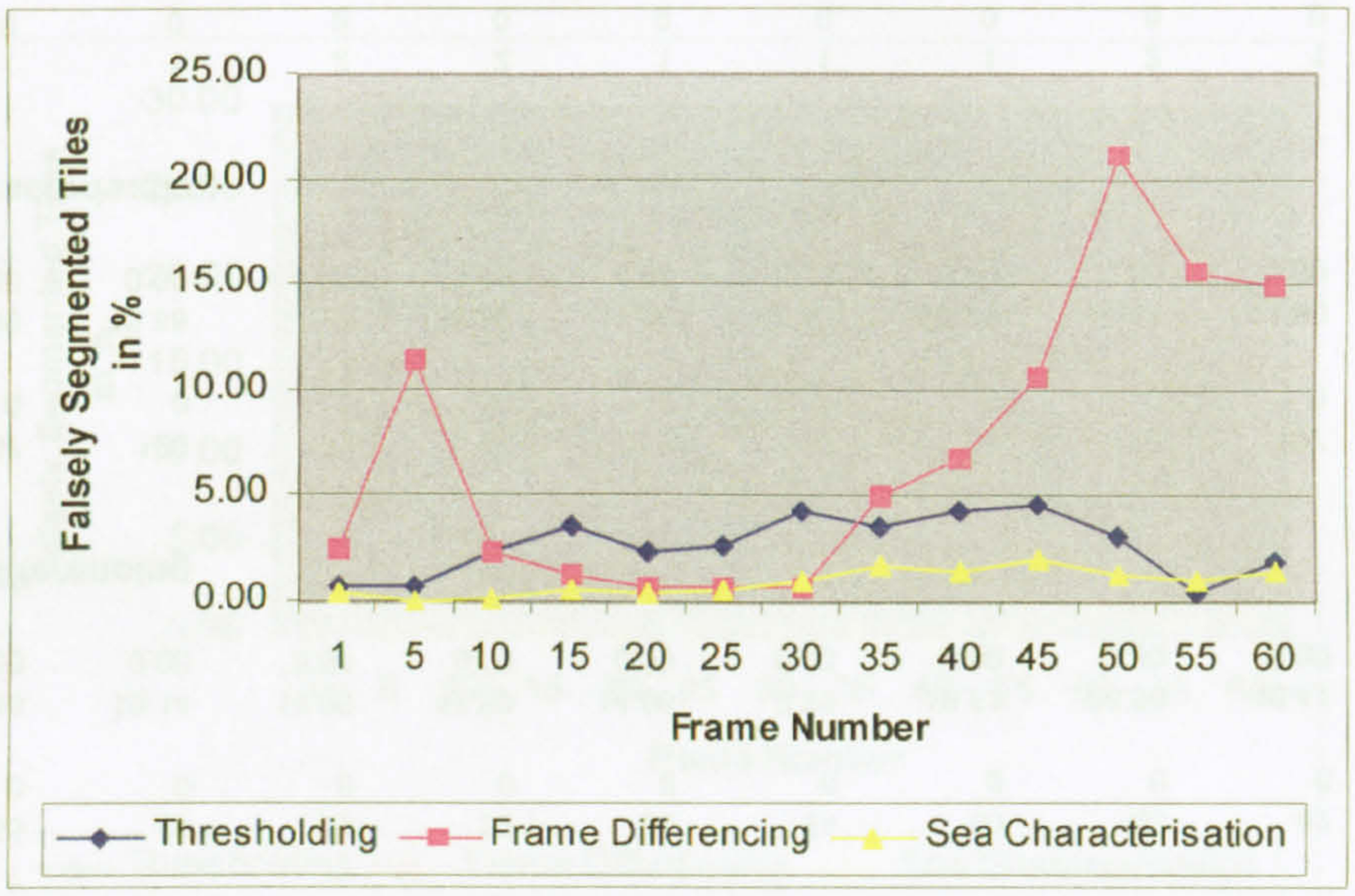

Figure C1 – Evaluation of the three segmentation methods on the Poole sequence

Table C2 – Evaluation of the three segmentation methods on the Dover sequence

		Frame	1	5	10	15	20	25	30	35	40	45	50	55	60
		Thresholding													
Segmented Image Description															
	FALSE	Object tiles	55	49	47	32	36	24	50	68	42	33	19	23	50
		Sea tiles	0	0	0	0	0	0	0	0	0	0	0	0	0
Results in%															
	FALSE	Object tiles	21.48	19.14	18.36	12.50	14.06	9.38	19.53	26.56	16.41	12.89	7.42	8.98	19.53
		Sea tiles	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Frame Differencing													
Segmented Image Description															
	FALSE	Object tiles	107	120	131	98	107	119	98	119	107	119	50	119	80
		Sea tiles	0	0	0	0	0	0	0	0	0	0	0	0	0
Results in%															
	FALSE	Object tiles	41.80	46.88	51.17	38.28	41.80	46.48	38.28	46.48	41.80	46.48	19.53	46.48	31.25
		Sea tiles	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Sea Characterisation													
Segmented Image Description															
	FALSE	Object tiles	1	1	2	2	1	1	1	2	1	3	7	5	6
		Sea tiles	0	0	0	0	0	0	0	0	0	0	0	0	0
Results in%															
	FALSE	Object tiles	0.39	0.39	0.78	0.78	0.39	0.39	0.39	0.78	0.39	1.17	2.73	1.95	2.34
		Sea tiles	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Average False tile segmentation

Thresholding	21.48	19.14	18.36	12.50	14.06	9.38	19.53	26.56	16.41	12.89	7.42	8.98	19.53
Frame Differencing	20.90	23.44	25.59	19.14	20.90	23.24	19.14	23.24	20.90	23.24	9.77	23.24	15.63
Sea Characterisation	0.20	0.20	0.39	0.39	0.20	0.20	0.20	0.39	0.20	0.59	1.37	0.98	1.17

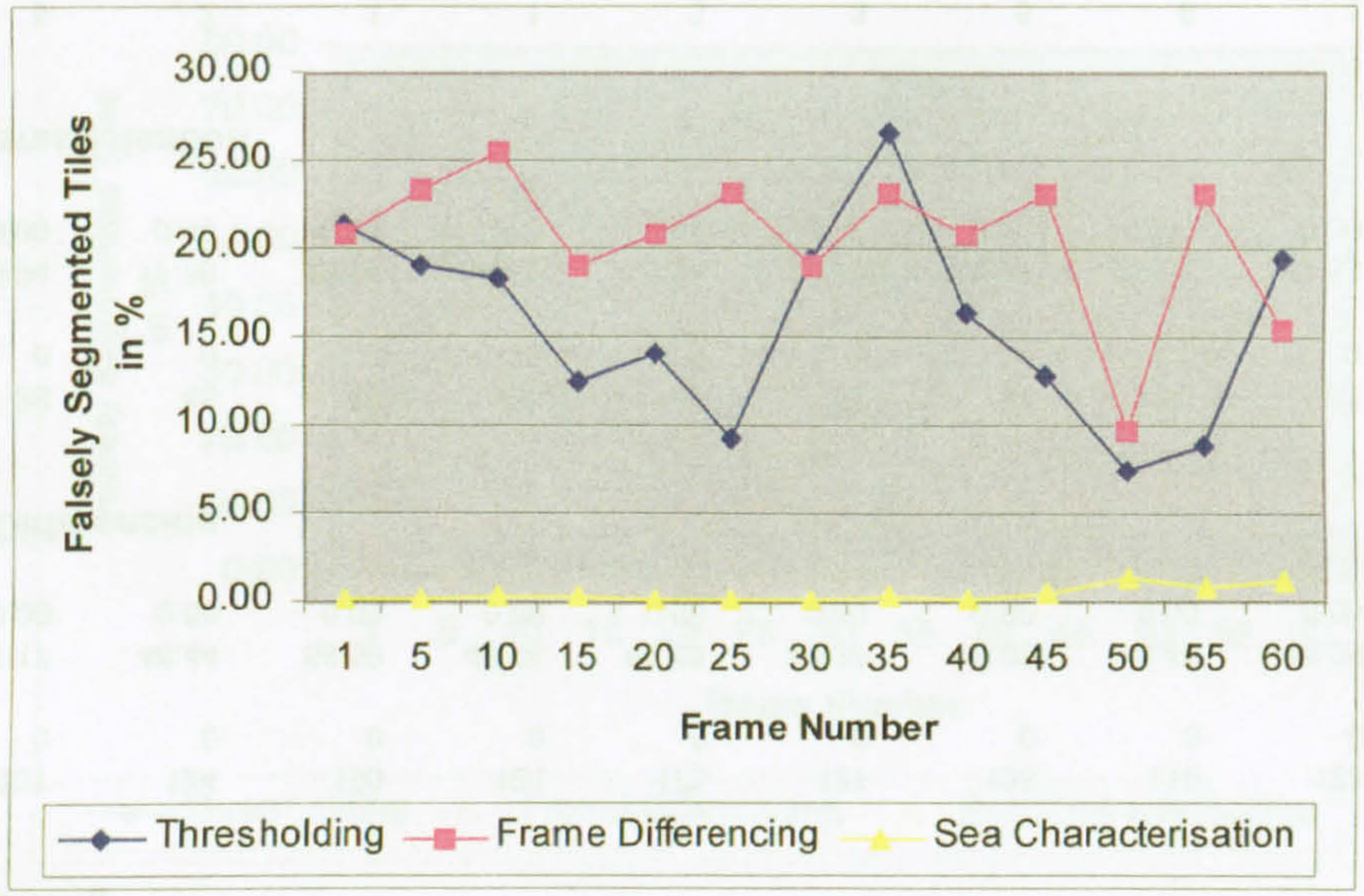

Figure C2 – Evaluation of the three segmentation methods on the Dover sequence

Table C3 – Evaluation of the three segmentation methods on the Portsmouth sequence

		Frame	1	5	10	15	20	25	30	35	40	45	50	55	60
		Thresholding													
Segmented Image Description															
	FALSE	Object tiles	131	124	150	124	173	121	123	119	123	118	129	116	120
		Sea tiles	0	0	0	0	0	0	0	0	0	0	0	0	0
Results in%															
	FALSE	Object tiles	51.17	48.44	58.59	48.44	67.58	47.27	48.05	46.48	48.05	46.09	50.39	45.31	46.88
		Sea tiles	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Frame Differencing													
Segmented Image Description															
	FALSE	Object tiles	28	45	49	50	49	53	51	52	61	62	69	60	64
		Sea tiles	0	0	0	0	0	1	0	0	0	0	0	0	1
Results in%															
	FALSE	Object tiles	10.94	17.58	19.14	19.53	19.14	20.70	19.92	20.31	23.83	24.22	26.95	23.44	25.00
		Sea tiles	0.00	0.00	0.00	0.00	0.00	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.39
		Sea Characterisation													
Segmented Image Description															
	FALSE	Object tiles	5	2	1	1	3	3	6	6	1	9	0	4	0
		Sea tiles	0	0	1	0	0	0	0	0	2	0	1	0	0
Results in%															
	FALSE	Object tiles	1.95	0.78	0.39	0.39	1.17	1.17	2.34	2.34	0.39	3.52	0.00	1.56	0.00
		Sea tiles	0.00	0.00	0.39	0.00	0.00	0.00	0.00	0.00	0.78	0.00	0.39	0.00	0.00
Average False tile segmentation															
		Thresholding	51.17	48.44	58.59	48.44	67.58	47.27	48.05	46.48	48.05	46.09	50.39	45.31	46.88
		Frame Differencing	5.47	8.79	9.57	9.77	9.57	10.55	9.96	10.16	11.91	12.11	13.48	11.72	12.70
		Sea Characterisation	0.98	0.39	0.39	0.20	0.59	0.59	1.17	1.17	0.59	1.76	0.20	0.78	0.00

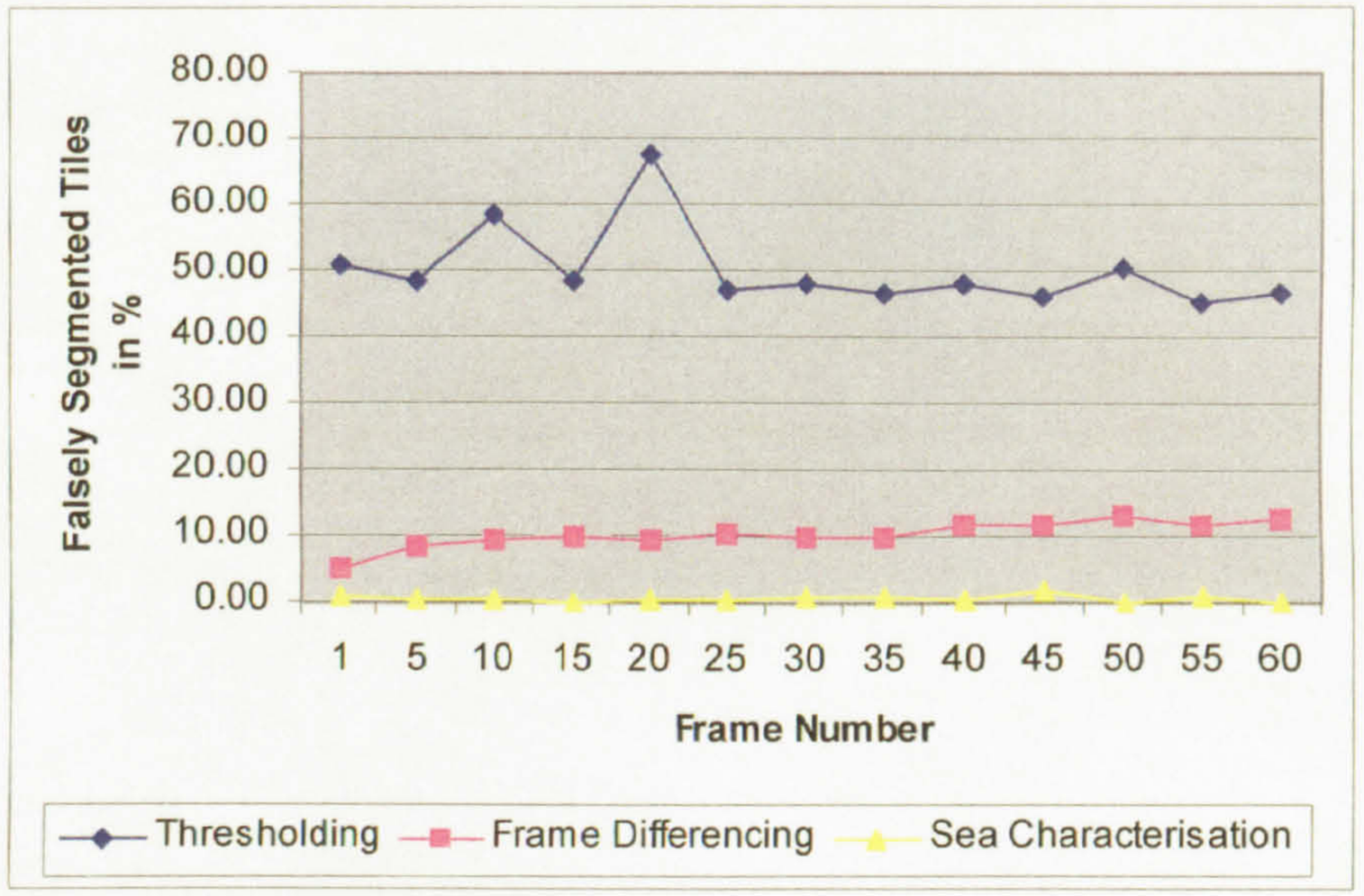

Figure C3 – Evaluation of the three segmentation methods on the Portsmouth sequence

Appendix A: Evaluation of Tile Rejection Threshold

This appendix shows the results of the determination of the tile rejection threshold value, θ. Two frames from each of two test sequences have been used. These are shown in figures A1 and A3 for the Poole sequence and A5 and A7 for the Dover sequence. The tables A1 to A4 give the manually calculated and the algorithmically calculated characteristic sea range for each of the four panes in the image. This has been carried out for six different threshold values and the error between the manual and algorithmic values calculated for each threshold. Figures A2 and A4 show the error rates for each threshold value on a graph for the two frames from the Poole sequence and figures A6 and A8 show the error rates for the two frames from the Dover sequence.