

# **Digital Enhancement Techniques for Underwater Video Image Sequences**

by

**Tristan John Lambert, BComp**

A dissertation submitted to the

School of Computing

in partial fulfilment of the requirements for the degree of

**Bachelor of Computing with Honours**



UNIVERSITY  
OF TASMANIA

**November, 2005**



---

## **Declaration**

I, Tristan Lambert, declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution, and that, to my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text.

Tristan Lambert

---

## **Abstract**

Due to concern about the current state of the world's oceans, several large scale scientific projects have begun to investigate the condition of our oceans. These projects are making use of underwater video sequences to monitor marine species. The move to using underwater video monitoring introduces labour intensive manual processing techniques. This leads to the need for an automated system capable of processing the data at a much greater speed. This project investigated whether the development of suitable image processing techniques could be used for pre-processing underwater images from a fish farm and locating fish within these images using computer vision techniques.

Using underwater images leads to some serious problems when compared to images from a clearer environment. Visibility in an underwater environment is poor, even when using state of the art equipment. After reviewing the broad field of computer vision and current underwater projects, an image pre-processing system was developed in MATLAB using suitable image processing and analysis techniques.

The application developed was able to successfully locate an acceptable number of fish within the underwater images. The project demonstrated that automated analysis of underwater video images is needed and is possible. Automatic processing of large quantities of video image sequences will be of great benefit in the future. It will allow scientific researchers to study the ocean environment and its species more effectively. Pre-processing is an essential component of the overall process that will lead to automation of underwater video data analysis for marine science applications.

---

## Acknowledgments

Firstly I would like to thank my supervisor, Dr Ray Williams. The input and support received from Ray this year was excellent. The time taken by Ray to provide feedback is very much appreciated. I would also like to thank AQ1 systems for providing the images used in this project. The project has proved to be extremely interesting.

Special thanks go to all of the Hobart honours crew. In particular I would like to thank David, Michael, Anthony, Ivan, Hallzy, Dima, Tristan (Ling), Emma, Simon, Duncan, Hamid (aka Warhamma), Bruno and Steve. The atmosphere in the honours rooms this year was great, with loads of fun and enjoyment for everyone (unless there was work to be done). Thanks to everyone for regular trips to the oval to kick the footy and get some fresh air, it really helped to keep the stress levels down. Also, thanks to all my mates who helped throughout the year (either by playing games or drinking beer).

Big thanks to my family for providing support throughout the year. This would not have been possible without you and any support has been greatly appreciated. Mum, thanks for providing a roof over my head (even though you may not have seen much of me this year).

Finally, I would like to thank my girlfriend Belinda, not just for this year, but for the past three years. Your support has been extremely helpful and I am a better person because of it. Your constant drive to do well set a great example, helping me to be where I am today.

---

# Contents

<b>DECLARATION.....</b>	<b>I</b>
<b>ABSTRACT.....</b>	<b>II</b>
<b>ACKNOWLEDGMENTS .....</b>	<b>III</b>
<b>CONTENTS.....</b>	<b>IV</b>
<b>LIST OF FIGURES.....</b>	<b>VI</b>
<b>LIST OF TABLES.....</b>	<b>VII</b>
<b>INTRODUCTION.....</b>	<b>1</b>
<b>COMPUTER VISION TECHNIQUES.....</b>	<b>3</b>
2.1 <b>CONTRAST ENHANCEMENT.....</b>	<b>3</b>
2.1.1 <i>Histogram Equalisation.....</i>	<i>4</i>
2.2 <b>SPATIAL IMAGE SMOOTHING.....</b>	<b>5</b>
2.2.1 <i>Linear Filters.....</i>	<i>5</i>
2.2.2 <i>Non Linear Filters.....</i>	<i>6</i>
2.3 <b>BACKGROUND REMOVAL.....</b>	<b>6</b>
2.4 <b>TEXTURE.....</b>	<b>7</b>
2.5 <b>IMAGE SEGMENTATION .....</b>	<b>7</b>
2.6 <b>EDGE DETECTION.....</b>	<b>8</b>
2.7 <b>IMAGE ANALYSIS.....</b>	<b>9</b>
2.8 <b>UNDERWATER VISION PROJECTS.....</b>	<b>10</b>
2.8.1 <i>The Monterey Bay Project.....</i>	<i>10</i>
2.8.2 <i>Fish Species Identification.....</i>	<i>11</i>
2.8.3 <i>Clear Underwater Vision.....</i>	<i>13</i>
<b>METHODOLOGY.....</b>	<b>14</b>
3.1 <b>INITIAL CONSIDERATIONS.....</b>	<b>14</b>
3.1.1 <i>Aim and Requirements.....</i>	<i>14</i>
3.1.2 <i>Platform.....</i>	<i>15</i>
3.1.3 <i>Input/Output.....</i>	<i>16</i>

---

3.2	PRE-PROCESSING.....	17
3.2.1	<i>Contrast Enhancement</i> .....	17
3.2.2	<i>Noise Removal</i> .....	18
3.2.3	<i>Background Detection</i> .....	19
3.2.4	<i>Initial Segmentation</i> .....	21
3.3	ANALYSIS.....	24
3.3.1	<i>Ellipse Fitting</i> .....	24
3.3.2	<i>Confidence Factor</i> .....	26
3.3.3	<i>Further Segmentation</i> .....	30
3.4	EVALUATION.....	32
<b>RESULTS AND DISCUSSION.....</b>		<b>34</b>
4.1	ACCEPTANCE THRESHOLDS FOR THE ELLIPSE ANALYSIS STAGE.....	34
4.1.1	<i>Estimating Thresholds from Ideal Fish Models</i> .....	35
4.1.2	<i>Estimating Thresholds from Real Fish Images</i> .....	36
4.1.3	<i>Comparison of Threshold Ranges</i> .....	36
4.2	RESULTS BASED ON THE INITIAL SEGMENTATION STAGE.....	38
4.3	RESULTS BASED ON THE INITIAL ANALYSIS STAGE.....	39
4.4	RESULTS BASED ON THE SUBSEQUENT ANALYSIS STAGE.....	41
4.5	RESULTS BASED ON COMPLETE PROCESS.....	43
<b>CONCLUSION AND FUTURE DEVELOPMENT.....</b>		<b>45</b>
5.1	CONCLUSION.....	45
5.2	FUTURE DEVELOPMENT.....	47
<b>REFERENCES.....</b>		<b>49</b>
<b>APPLICATION DETAILS.....</b>		<b>51</b>
A.1	APPLICATION CD.....	51
A.2.1	<i>RESULTSCRIPT.M</i> .....	52
A.2.2	<i>SEGFISH.M</i> .....	52
A.2.3	<i>STATFISH.M</i> .....	52
A.2.4	<i>ERODEFISH.M</i> .....	52
A.2.5	<i>STATFISH2.M</i> .....	52
A.2.6	<i>DRAWELLIPSE.M</i> .....	52
A.2.7	<i>POINTS.M</i> .....	53
A.2.8	<i>DRAWFISH.M</i> .....	53

---

---

## List of Figures

FIGURE 2.2.1 HISTOGRAM EQUALISATION.....	5
FIGURE 2.2.2 SOBEL EDGE DETECTOR.....	9
FIGURE 2.2.3 BACKGROUND SUBTRACTION.....	12
FIGURE 3.3.1 HISTOGRAM EQUALISATION.....	18
FIGURE 3.3.2 NOISE REMOVAL .....	19
FIGURE 3.3.3 SOBEL EDGE DETECTION.....	21
FIGURE 3.3.4 INITIAL SEGMENTATION PROCESS.....	22
FIGURE 3.3.5 INITIAL SEGMENTATION OUTPUT.....	23
FIGURE 3.3.6 ELLIPSES PLOTTED OVER SEGMENTED IMAGE.....	26
FIGURE 3.3.7 ELLIPSES PLOTTED WITH INITIAL CONFIDENCE VALUE.....	27
FIGURE 3.3.8 ELLIPSES PLOTTED WITH SECOND CONFIDENCE VALUE.....	28
FIGURE 3.3.9 PROCESS OF CALCULATING CONFIDENCE VALUE.....	29
FIGURE 3.3.10 FURTHER SEGMENTATION.....	31
FIGURE 3.3.11 THE APPLICATION PROCESS.....	32
FIGURE 4.4.1 DISTRIBUTION OF RANGES FOR THE XOR CONFIDENCE VALUE.....	37
FIGURE 4.4.2 DISTRIBUTION OF RANGES FOR THE AREA CONFIDENCE VALUE.....	37
FIGURE 4.4.3 INITIAL SEGMENTATION TO FINAL OUTPUT.....	43

---

## List of Tables

TABLE 4.4.1 ELLIPSE MATCHING ON IDEAL FISH MODELS.....	35
TABLE 4.4.2 INITIAL SEGMENTATION RESULTS.....	38
TABLE 4.4.3 IDEAL AND ACTUAL THRESHOLD RANGES.....	40
TABLE 4.4.4 FURTHER SEGMENTATION RESULTS.....	42



## Chapter 1

### Introduction

Due to concern about the current state of the world's oceans, several large scale scientific projects have begun to investigate the condition of our oceans. These projects are making use of underwater video sequences to monitor marine species. This leads to the problem of using visible light imaging in an underwater environment. Underwater vision is plagued by poor visibility conditions producing images with poor contrast and colour variation. In the past other techniques such as sonar ranging have been used due to the limitation of visible light imaging in an underwater environment. Scientists are now starting to use visible imaging for close range studies due to the fact that alternative techniques produce low resolution images that are difficult to interpret (Wilson 2003).

The move to using underwater video monitoring introduces a huge bottle-neck for any research that relies on this data. There are now huge amounts of video data available for researchers to use. Many current techniques for processing this data are labour intensive and tedious, usually requiring a highly trained scientist to view the videos, make annotations of the animals, then enter the annotations into a database. This creates serious limitations in the amount of underwater data that can be processed. Manual processing of such large amounts of data has become impossible, which leads to the need of an automated system capable of processing the data at a much greater speed. There are already projects that are using automated systems to detect objects in underwater video, such as detecting objects that are of interest to human researchers (Rife & Rock 2001).

This project involves the development of suitable image processing techniques for pre-processing underwater images from a fish farm and locating fish within these

---

images using computer vision techniques. Using underwater images leads to some serious problems when compared to images from a clearer environment. Visibility in an underwater environment is poor, even when using state of the art equipment (Jaffe 1990). Also, light is strongly attenuated in water, producing images of low contrast and little colour variation.

This task may be fairly easy for a human, even if they have never seen the particular fish before, or know what a fish farm environment is like. These issues will pose little problem in identifying several fish within the image (given that there is a reasonable amount of contrast within the image). Assuming that the viewer has some idea about the shape of the fish, they should be able to separate the background from the fish. They will also be able to perceive each fish as a separate object from the others, possibly even counting the number of fish within the image. This may not seem like a difficult task to perform, even with the low image quality due to the underwater environment, but suppose there were hundreds of images and video sequences that needed to be analysed. This task then becomes difficult to perform due to the huge amount of time required. It is possible that this task could be solved using computer vision to create an automated system.

To give this task to a computer, it will need to have some knowledge of the shape of the particular fish, and be able to match this shape to fish within the image. Unlike humans, computers cannot easily separate objects within an image, or perceive objects as separate from the background. To make the task of identifying the fish easier for the computer to handle, image pre processing techniques can be performed. Through the use of specific image enhancement techniques, the images may be taken to a stage where a computer can start to identify objects within that image. At the very least these image enhancements would be useful within a manual system, possibly getting to a point where these images can be used in an automated system. This process of applying pre-processing techniques, then applying techniques to identify objects within the image, describes a typical computer vision system.

## Chapter 2

### Computer Vision Techniques

Computer vision is the task of automating and integrating a wide range of processes and representations used for visual perception (Ballard & Brown). Through the use of digital hardware, computer vision is used to reinvent the most basic abilities of a biological visual system. Computer Vision, as a process includes many stages and techniques that are useful by themselves, such as image capture, image processing and image analysis.

The focus of this project lies mainly in the area of image processing, while also performing some initial analysis on the processed images. This may be considered a computer vision system by itself; however, the focus is on the pre-processing of video images providing output which can be used in a further analysis application. Much of the work done in the past on image pre-processing has had very little focus on underwater images. The use of imaging techniques used for sequences of images rather than just focusing on a single image may provide further improvements to underwater image quality.

#### 2.1 Contrast Enhancement

Image contrast enhancement is a technique used to increase the visibility of images. Many methods have been proposed which emphasize different properties or components of an image. Most methods for contrast enhancement can be placed into one of two main categories (Zhu, Chan & Lam 1999): intensity based methods and feature based methods. Intensity based methods are of the form:

---

$$I_o(x, y) = f(I(x, y))$$

**Equation 2.2.1 Intensity based contrast enhancement**

Where the original image is  $I(x, y)$ , the output image is  $I_o(x, y)$  after contrast enhancement, and  $f$  is the transformation function. Intensity based methods transform the grey levels over the whole image. Therefore pixels with equal grey levels throughout the image will still have equal grey levels after the transformation. Contrast stretching is a widely used technique which fits into this category, (Fahnestock & Schowengerdt 1983). Another very popular method is histogram equalisation (Gonzalez & Woods 2001). Any intensity based method is actually a gray level transformation, it does not consider any other components of the image except for its grey level distribution. Feature based methods are of the form:

$$I_o(x, y) = L_i(x, y) + G(x, y)H_i(x, y)$$

**Equation 2.2.2 Feature based contrast enhancement**

where  $L_i$  represents the low frequency components which keep the basic appearance of the original image,  $H_i$  represents the high frequency components which contain the concerned features, and  $G$  is the enhancement gain. Feature based methods are used to enhance the smaller components of an image, usually representing specific features of an image. Feature based contrast enhancements can be done in the frequency domain or spatial domain, (Gonzalez & Woods 2001).

### **2.1.1 Histogram Equalisation**

A grey level histogram of an image is the frequency of each grey level occurring in the image (Figure 2.2.1). Where the grey levels in an image range from 0 – n, the value of the histogram at a particular grey level p, is the number of pixels in that image with that grey level.

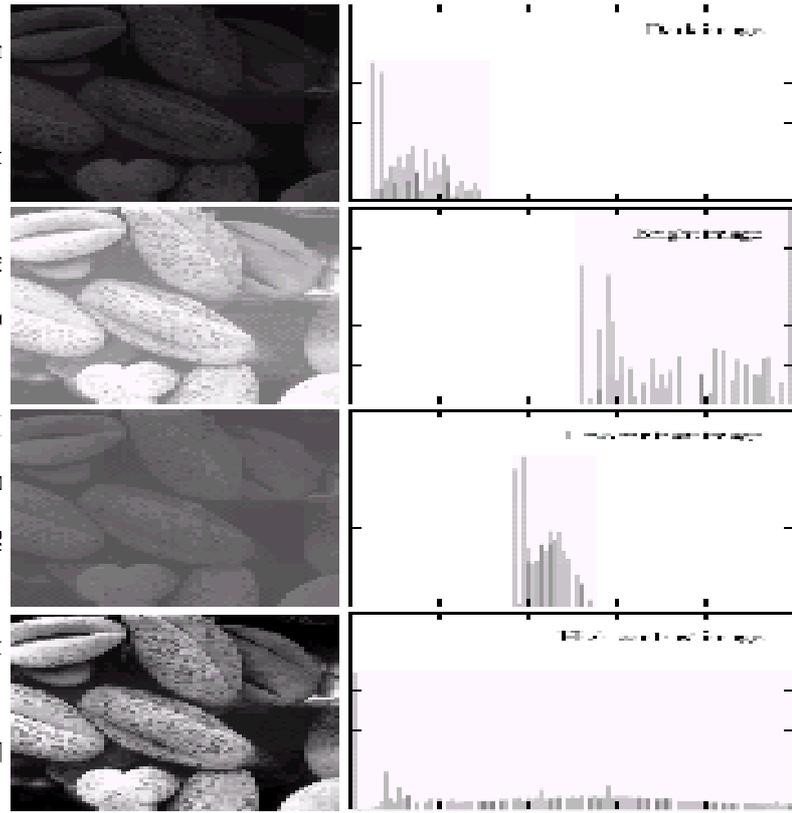
**Figure 2.2.1 Histogram**

Four basic image types:  
their corresponding histo

A histogram of an image can be  
transformations to achieve a spe  
earlier, one of the more popular  
Histogram equalisation maps grey  
in such a way that grey levels  $q$  a  
(contrast) that are near the histog  
levels that are near the histogram  
expanded for most pixels, improv

Using histogram equalisation can l  
is desired, although there are still

uniform histogram may not be the best approach. In these situations, histogram equalisation effects may be too severe. So other histogram techniques may need to be used, such as adaptive histogram equalisation (Stark 2000).



## 2.2 Spatial Image Smoothing

Image smoothing can be used to blur an image and to remove noise from an image. Blurring is generally used in the pre-processing stage, to remove small details from an image before attempting to extract larger objects. Blurring will also bridge small gaps in lines and curves. Noise reduction is achieved by smoothing with a linear filter and also by non-linear filtering.

### 2.2.1 Linear Filters

The output of a smoothing linear spatial filter is the average of the pixels within the neighbourhood of the filter mask. Therefore these filters are also known as averaging filters. By replacing the value of every pixel in the image with the average of the grey levels in the area defined by the filter mask, the resulting image will contain less sharp transitions in grey levels. Random noise in an image will typically be sharp

transitions in grey levels, therefore image smoothing is a very popular method of noise reduction. However, a problem with using this technique is that edges are also typically defined by sharp transitions in grey levels. So using this technique to remove noise may also lead to a blurring of edges, which are extremely important in automated systems that require edge detection. Although edge detection and smoothing are regarded as two different image processing tasks, there have been some recent advances leading to methods which are able to perform both together (Chen, Barcelos & Mair 2001).

### **2.2.2 Non Linear Filters**

The best known filter in this category is the median filter, which is an order-statistic filter. Order-statistic filters are nonlinear spatial filters whose output is based on ordering the pixels contained in the neighbourhood of the filter, then replacing the centre pixel with the value determined by the ordering result. The median filter replaces the centre pixel with the median of the grey levels within the mask. Median filters can be very effective for removing particular types of random noise with less blurring than linear smoothing filters of similar size.

## **2.3 Background Removal**

Background removal is another important step in pre processing. The subtraction of the background from a video sequence is often one of the most important parts of an automated vision system (KaewTraKulPong & Bowden 2001). Usually the output of background subtraction will be used by a higher process, such as object tracking and detection. A typical form of background subtraction for video images is to calculate a reference image, subtract each new frame from the reference image and threshold the result. The may be used to produce a segmentation of moving objects within the image. One of the simplest forms of reference image is a time averaged background image. One of the problems that this technique encounters is the requirement for the reference image to be calculated with no foreground objects present. This technique cannot handle changing illumination or moving backgrounds correctly. An attempt can be made to overcome these problems by constantly re-estimating the reference image. Background removal in real world scenes can be very difficult due to

---

illumination changes, as well as swaying vegetation, rippling water and many other factors. Shadows can also cause problems. There have been models created that attempt to adaptively overcome these problems, such as texture based background removal (Heikkila, Pietikainen & Heikkila 2004).

## **2.4 Texture**

There is no set description for texture, most textures can be defined as “something composed of closely interwoven elements” (Ballard & Brown 1982). This definition relates to texture resolution, which can be thought of as the number of pixels in each texture element. If the texture resolution is high, it is possible to describe the individual texture elements in detail. However, as the texture resolution becomes lower, it becomes difficult to describe the texture elements individually. A texture can be thought of as being made up of repeated primitive elements, with these elements possibly displaying some kind of variation. These elements may be obvious, making modelling the texture relatively easy, whereas with textures such as those of wire braid or netting, it may be better to try to model the relationship of a weave in the image. Two broad techniques are used with the aim of recognising or classifying a texture. The structural model regards the primitives as forming a repeating pattern, these patterns are then described in terms of rules for generating them. This model is usually best for describing textures where there is a high level of regularity for the placement of the individual texture elements. The statistical model describes texture by rules that describe the distribution and relation of grey levels. This model works well for textures that have no recognisable primitive elements.

## **2.5 Image segmentation**

Segmentation subdivides an image into different regions which may represent objects or parts of objects, which exist in the scene being viewed. The level of subdivision is dependent upon the problem being solved. Ideally, segmentation should stop when the required objects in an application have been found. For example, with the automated detection of fish in an image taken from a fish farm, the segmentation process should stop when the fish objects have been found. There is no need to continue subdivision past the level of detail required to identify the fish. In a

---

computer vision system, segmentation accuracy determines the success or failure of the procedure. Image segmentation algorithms are usually based on one of these two properties of intensity values: similarity and discontinuity (Gonzalez & Woods 2001). Similarity approaches attempt to partition an image into regions that are similar according to a set of predefined criteria. Some examples of methods that fit into this category are thresholding, region growing or region splitting and merging. Discontinuity approaches attempt to partition an image based on sudden changes in intensity, such as edges in an image.

There are three main types of grey level discontinuities in images: points, lines and edges. Although point and line detection are important for segmentation, the most common technique for detecting discontinuities in grey levels is edge detection.

## **2.6 Edge Detection**

An edge in an image can be thought of as a set of connected pixels that lie on the boundary between two regions (Gonzalez & Woods 2001). While a boundary between two regions can be thought of as an edge, it is not necessarily the case that an edge represents a boundary. That is, a boundary represents more of a global concept made up of connected edges, where an edge is a local concept that can create a boundary. A more accurate description of an edge in an image is a sharp transition in gray levels. However, edges in images can not be expected to have this ideal sharp transition in gray levels. Because of image acquisition imperfections, edges are usually subject to a level of blurring, with this level being determined by the quality of the image, the lighting conditions and resolution of the image. As a result of this blurring, edges in images are more closely represented by a ramp like transition in gray levels rather than a sharp transition. By calculating the first and second derivatives of a gray level profile, useful information can be obtained that will indicate whether an edge exists within this profile (Gonzalez, Woods & Eddins 2004). The magnitude of the first derivative is positive at the points into and out of the ramping transition in gray levels. Therefore, using the magnitude of the first derivative can be used to indicate an edge at a point in an image. The second derivative is zero along the ramping transition in gray levels, however it is positive at the transition with the dark side of the edge and negative at the transition of the light

---

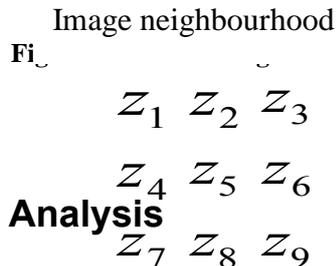
side of the edge. A line crossing between the positive and negative points of the second derivative can be used to indicate the mid point of a thick edge in an image.

There are several edge detection algorithms that provide different methods for digitally estimating the derivatives of transitions in gray levels. One example of an edge detection algorithm is the Sobel Edge detector. The Sobel edge detector uses the masks in Figure 2.2.2 to digitally estimate the first derivatives  $G_x$  and  $G_y$ . For example, the gradient at the centre pixel in the neighbourhood is calculated by the Sobel edge detector as follows:

$$g = [G_x^2 + G_y^2]^{1/2}$$

Equation 2.2.3 Sobel gradient

The level at which this value indicates an edge in the image is quite varied depending on the image and the type of edges being detected. Therefore, edge detection algorithms such as the Sobel edge detector make use of a threshold value to determine if the transition in gray level is a meaningful edge point. If the first derivative is greater than the specified threshold, then the point is defined as an edge point in the image.



## 2.7 Image Analysis

While image enhancement and segmentation form the largest part of an image pre-processing system, image analysis also has an important role to perform (Gauch 1999). Analysis of an image is the process of identifying what objects in the image are present. Analysis of segmented images is used to obtain information such as the number, size, orientation and position of each object. Analysis may also provide an indication of how important the object is to the system. This

$$G_x = \frac{1}{2}(z_7 + 2z_8 + z_9) - \frac{1}{2}(z_1 + 2z_2 + z_3) \quad G_y = \frac{1}{2}(z_3 + 2z_8 + z_9) - \frac{1}{2}(z_1 + 2z_4 + z_7)$$

measure may then be used to discard objects that are irrelevant to the system. Also, by performing segmentation together with analysis, the information obtained may be used in higher level vision tasks such as shape matching and object recognition.

The techniques chosen for image analysis are dependent upon the overall task of the system. A fruit sorting system with objects that are ellipsoidal in shape may use ellipse fitting to provide analysis of the objects (Wijewickrema & Paplinski 2004). Ellipses are also useful in systems that perform face recognition, where an ellipse can be used to estimate the overall area of the face (Kim et al. 2000). Other techniques may take into account features such as colour, texture etc. to perform analysis on feature spaces (Semani et al. 2002). If the information obtained from performing segmentation and analysis is to be used in a higher level vision task, the analysis technique chosen must be able to accurately provide this information. While it is important that analysis techniques are chosen based on the objects being segmented, it is equally important to consider the output needed for higher level vision tasks.

## **2.8 Underwater Vision Projects**

### **2.8.1 The Monterey Bay Project**

One of the major underwater projects currently in progress involves visual tracking of gelatinous animals and is being carried out by the Monterey Bay Aquarium Research Institute. They have developed an automated system that detects and tracks objects that are of potential interest for human video annotators (Walther, Edgington & Koch). Several techniques have been developed that overcome some of the problems of an underwater environment. These include poor contrast, variable lighting conditions, and the presence of high contrast noise from organic debris (“marine snow”). The system consists of several sub-components which interact with each other.

The first step for all video frames is background removal. This ensures the removal of artifacts such as lens glare, parts of the camera housing, or parts of the ROV instrumentation. Also, non-uniform lighting conditions create luminance gradients which can interfere with the contrast based detection algorithm. Because these effects

---

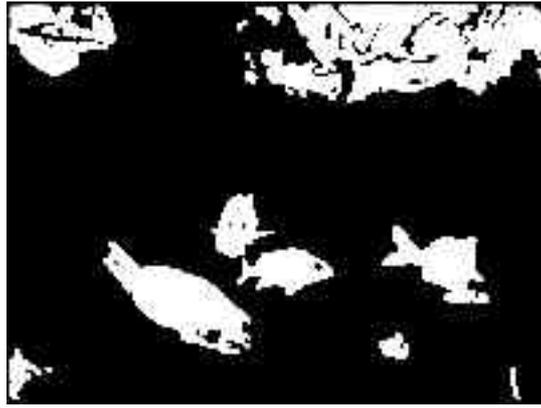
are constant over a medium or long period of time, they can be removed by background subtraction without removing the faster moving objects in the water.

To detect new objects in the image once background subtraction has been applied, they use the saliency-based bottom-up attention system developed by Itti, Koch & Niebur (1998). The frames are broken down into seven channels at six spatial scales, giving 42 “feature maps”. After iterative spatial competition for salience in each map they are combined into a “saliency map”. This saliency map is then scanned to find the most salient locations. Objects are segmented from the image at these locations and tracking is then initiated. The tracking system assumes that the camera is moving at a constant speed through stationary objects.

### **2.8.2 Fish Species Identification**

Another interesting project that makes use of underwater vision techniques is presented in Semani et al. (2002). This article presents a method for characterising fish species within a basin, based on the analysis of video sequences obtained from a fixed camera.





**Figure 2.2.3 Background Subtraction**

(a) The original frame. (b) The background image (c) Resulting segmentation. (Semani et al. 2002)

The first stage of the process involved segmenting the scene. The assumptions made during the segmentation stage are that there is constant illumination and the camera is fixed, however the fish (the objects of interest) are moving. Due to illumination changes introduced by the aquatic environment as well as electronic noise from the camera, the standard background subtraction technique was not robust enough over time. This problem was solved by dynamically updating the background image. This background image was initialised offline from another video sequence to obtain a suitable reference image (Figure 2.2.3b). This background reference image can then be subtracted from the current image of the video sequence (Figure 2.2.3a) to produce a binary difference image using a threshold value for the difference (Figure 2.2.3c). Because colour images were available from the video sequences, three binary difference images could be produced (representing red, green and blue). These three binary differences images were then combined using an OR operation to produce the final segmented image.

Once the segmentation process had been performed, feature extraction and selection was performed on each region within the segmented images. This included features such as colour, moments, texture etc. These features were then selected to form pattern vectors for each region. Once the pre-processing stage had been completed, the higher level task of species characterisation was performed.

### **2.8.3 Clear Underwater Vision**

While most underwater vision research focuses on processing images once they have been captured, there has also been research undertaken on improving the quality of underwater images by using new image acquisition techniques. Traditional image enhancement techniques such as histogram equalisation (section 2.1.1), are spatially invariant. Therefore, enhancements are made equally over the whole image without considering the distance of objects in the image. This causes problems when these techniques are applied in an underwater environment where the visibility of objects is largely dependent on the distance from the camera.

Schechner & Karpel (2004) used a physics based approach is developed for visibility recovery in underwater environments. Their approach relied on raw images taken through different states of a polarizing filter. These raw images had slight differences which were used to reduce the poor visibility cause by turbulence in the water. The assumption made in this approach is that due to natural illumination, underwater polarization is the leading cause of visibility disturbance (labelled as backscatter). By taking advantage of the effects of polarization due to underwater scattering of light, clearer images were able to be recovered that show an increase in colour variation as well as contrast.

## **Chapter 3**

### **Methodology**

This chapter shall discuss the tasks undertaken in this project. These tasks are divided into four main sections for discussion. The first of these covers the issues considered before designing the application, including the requirements of the project while also taking into account platform issues and input/output of the application. The second section discusses the pre-processing stage of the application. The techniques discussed in this section play a critical role in determining the overall success of the application. Another major element of the application is the analysis stage. This process, combined with segmentation, provides the application with the ability to provide output for use in a higher level application. The final part of this chapter discusses the processes used to perform evaluation on the application.

#### **3.1 Initial Considerations**

This section discusses the aims of the application and the issues considered before designing the application. Due to the nature of the project, a critical factor in deciding on an application platform was the decision not to implement image processing techniques from scratch using C++ or Java, but rather use existing techniques already implemented in a development platform and design the application on top of these techniques. Therefore, the aims of the application will be discussed, followed by the choice of development platform.

##### **3.1.1 Aim and Requirements**

The aim of this project has been to test the hypothesis that:

---

*Suitable image processing techniques can be used to process underwater video images, allowing the detection of objects within these images for use with further scientific analysis applications.*

The underwater video images used in this project are obtained from a local fish farm and are in the TIFF image format. These images are produced by an aquaculture monitoring system placed within the salmon tanks, capturing images of the salmon as they swim past. Currently these images are used to measure fish size using a manual analysis process. This process involves human input to locate a fish within the image and select points at key locations on the fish from which the measurement of fish size can be made.

The requirements for this project can be described as the bottom up approach to autonomous detection of a location of fish within underwater images. This bottom up approach requires the use of image processing techniques to enhance and segment the images, so that the locations of individual fish can be determined. Once these locations have been found, attributes such as estimates of the centroid, orientation, length and height of the segment representing the fish can potentially be used in a higher level approach to accurately measure the fish.

### **3.1.2 Platform**

Two development packages were investigated for the development of the research application. One was MATLAB which is commercially available, and the other a package called Octave, which is open source and reasonably compatible with MATLAB. Both of the above packages are scientific computational packages, which can be run on various different platforms. Due to the fact that both packages can be run on Microsoft Windows as well as MacOS X, the operating system choice was not a major factor in the choice of computation package.

The main advantage offered by Octave was the fact that it is a free open source package. There is no problem obtaining a licence. Therefore availability of the application for use in the project was not an issue. However, the Octave package is open source, which can bring problems along with it. Initial investigations showed

that even the simple step of reading an image file and displaying it in Octave was not straight forward, and was a different process on Microsoft Windows and MacOS X. The MATLAB package offered a significant advantage over Octave, because it also provided access to the Image Processing Toolbox. The Image Processing Toolbox offers a comprehensive suite of standard algorithms and graphical tools for image processing, analysis and visualisation.

If the application was to be developed using Octave, there would be no initial access to standard image processing algorithms. This would require implementations to be found or written for Octave, reducing the amount of time that could be spent on developing the application. Also, there is no guarantee that these implementations will function correctly if ported to another Operating System. However, using the Image Processing Toolbox in MATLAB would enable instant access to standard image processing algorithms, allowing more time to be spent in developing the application. Using the MATLAB language would allow the application to be easily ported to any environment that is supported by MATLAB.

After some consideration and the initial investigation into using Octave, MATLAB was chosen as the computational package that would be used to develop the application. The operating system used was MacOS X, because a MacOS X version of MATLAB and the Image Processing Toolbox was available for this project.

### **3.1.3 Input/Output**

The input to the application comprises of images in the Tagged Image File Format (TIFF). The images have a size of 640\*480 pixels and are 8-bit gray scale images. While the input images will be enhanced for the purposes of locating fish, the modified images will not be used as the output of the application. The output will consist of a data file containing the locations of fish within the images. Each line in the data file will contain five values separated by the character '|', represented as follows:

FileName|CentroidX|CentroidY|MajorAxisLength|Orientation

---

where FileName is the name of the image, CentroidX and CentroidY are the estimated locations of the centre of the fish, MajorAxisLength is the estimated length of the fish and Orientation represents the estimated orientation of the fish. This output can then provide estimates of the location, scale and orientation of the fish to be used as input to a higher level application. The benefits of providing output in this format are that the amount of data being exchanged from a bottom up system such as this to a top down system for subsequent data analysis is kept to a minimum. Also, providing output in plain text also enables the two different approaches to be developed on different platforms independently of each other, providing more flexibility for the final applications.

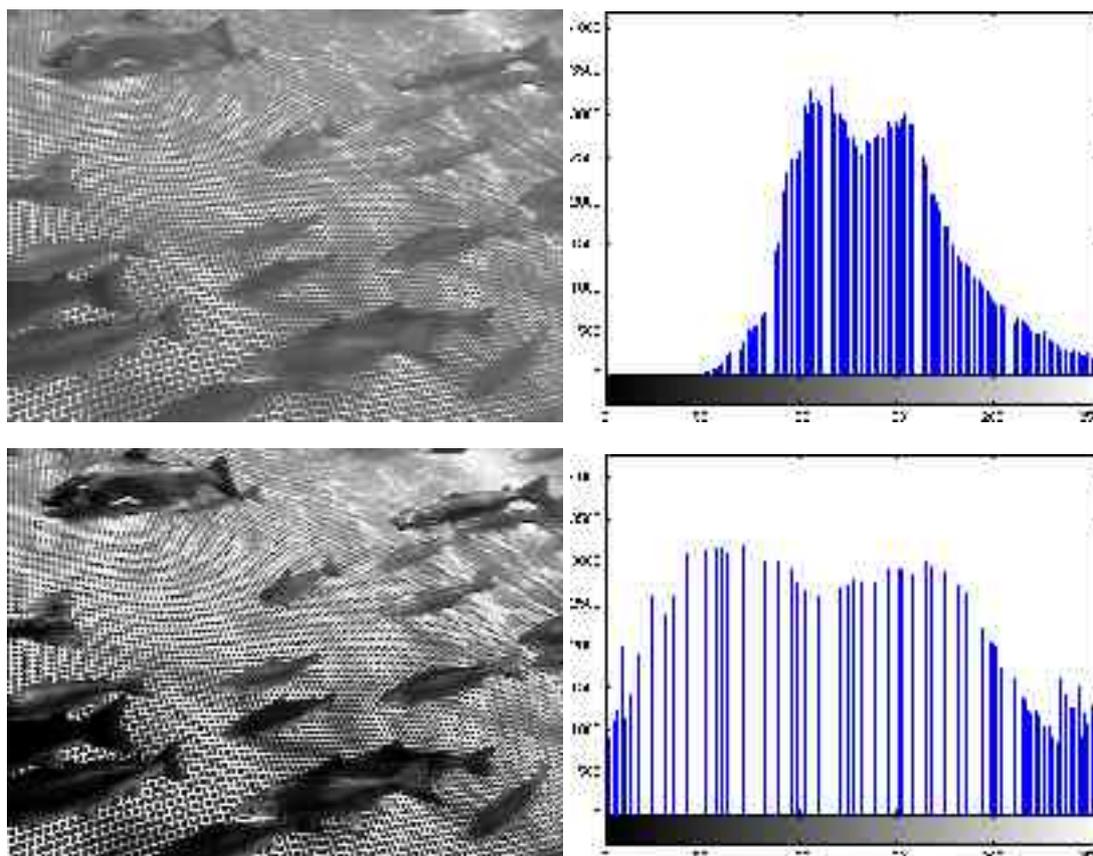
## **3.2 Pre-Processing**

### **3.2.1 Contrast Enhancement**

As described in section 2.1, contrast enhancement is a technique used to improve the visibility of images. Due to the nature of the input data, consisting of low contrast underwater images, contrast enhancement is the first process to be performed on the images. An important fact to consider while choosing a contrast enhancement technique is the need for automation. The contrast enhancement technique used must be able to perform in an automated application without requiring any user input. Histogram equalisation was chosen because it offers the benefit of automatic enhancement, also producing a higher contrast image.

In this application, histogram equalisation was performed over all 256 gray levels of the image. The images before histogram equalisation was applied contained mainly high pixels with high value gray levels, resulting in a low contrasting, bright image (Figure 3.3.1a - Figure 3.3.1b). After histogram equalisation was performed, these high value gray level pixels were expanded to more evenly cover the whole range of 256 gray levels in the image (Figure 3.3.1c - Figure 3.3.1d). The resulting images were of higher contrast containing more clearly defined fish. While histogram equalisation clearly improves contrast and visibility in the underwater images for human viewers, it also offers improvements for other processing techniques discussed in further sections.

---

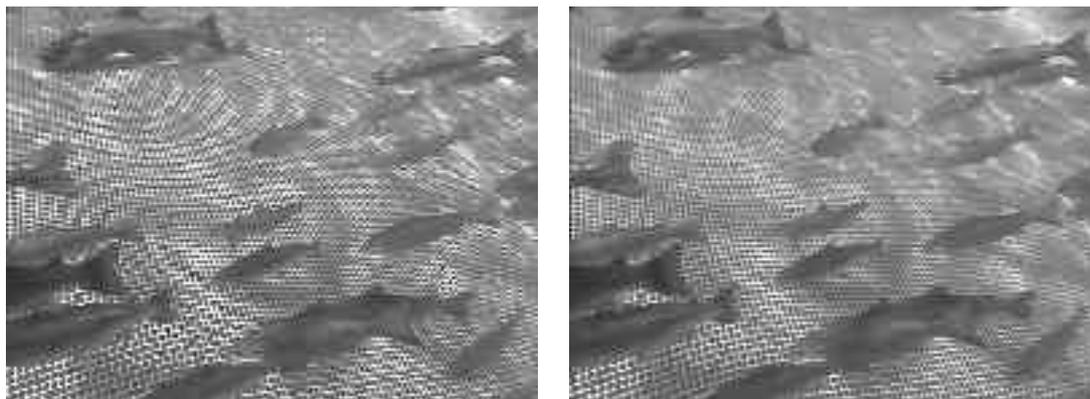


**Figure 3.3.1 Histogram Equalisation**

(a) Original image (courtesy of AQ1 Systems). (b) Histogram of original image. (c) Image after histogram equalisation. (d) Histogram of (c).

### 3.2.2 Noise Removal

Noise removal, also known as image smoothing, can play an important role in image enhancement. However, noise removal can also lead to the blurring of edges, which are extremely important in an automated system such as one required for this project. A median filter of size  $3 \times 3$  was tested on several images. Such a filter should generally be effective at removing noise while also not blurring the image as much as a linear filter. However, the resulting images were not improved significantly because they did not contain much random noise initially (Figure 3.3.2). Also, smoothing had a negative impact on the output of edge detection. Therefore, noise removal was not used as a technique in the final application.



**Figure 3.3.2 Noise Removal**

**(a) Original image (courtesy of AQ1 Systems). (b) Image after smoothing using a median filter**

### **3.2.3 Background Detection**

Background removal is also an important step in pre-processing. The subtraction of the background from video sequences is often the most important stage of an automated system. This stays true for this project, with the key step being that of separating the fish from the background. One of the problems with the images is that there is no way of obtaining a background reference image to work with, because every image contains fish within it. Therefore there is no way to use the typical background removal technique of subtracting a reference image from the current image to obtain a segmentation of moving objects. This technique would also experience problems caused by varying illumination and shadows.

A major feature of the underwater images in this project is the netting used to enclose the tank. This netting basically represents the background of the image, with the fish being the foreground objects. Therefore, background removal within these images can be thought of as being able to detect the netting and remove it, resulting in a binary segmentation of the fish from the background. The netting is a texture within the image, and the next stage of the project was to define this texture and apply image processing techniques to detect and remove it from the image.

The texture can basically be defined as a series of vertical and horizontal lines with high value gray levels. In between these lines, a square is formed with low value gray

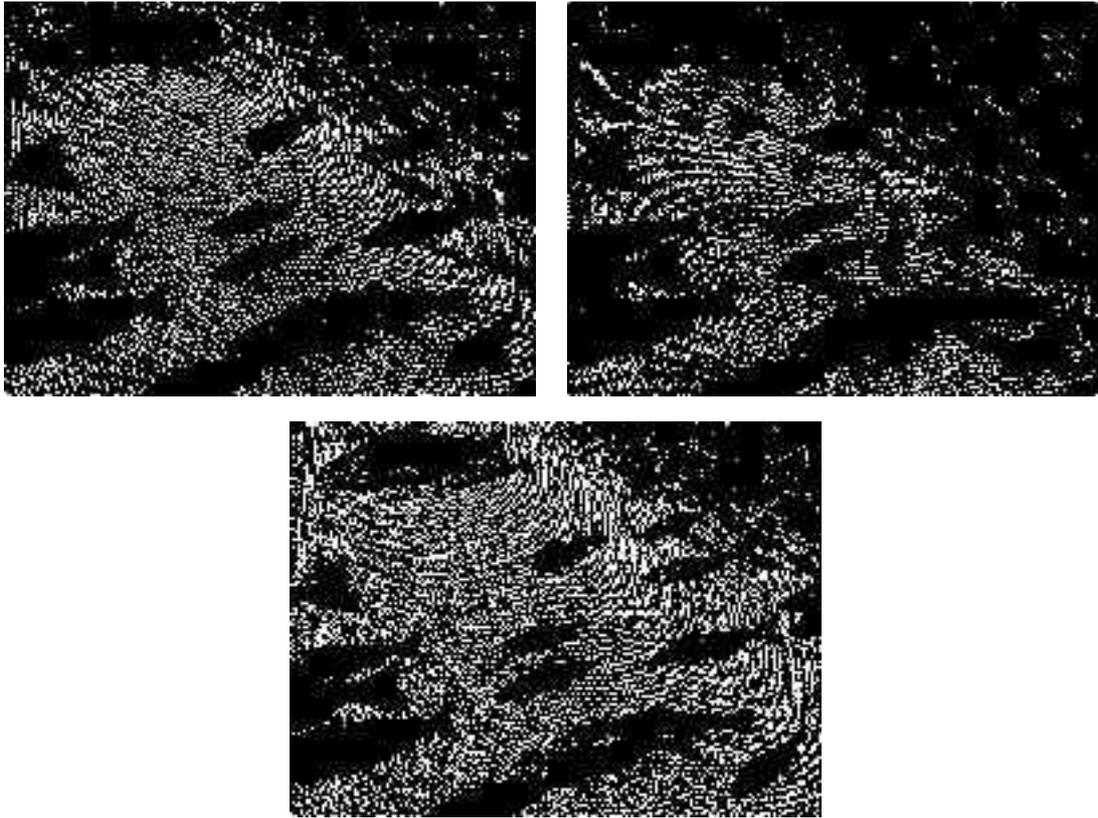
levels. Because edge detection operates by locating sharp transitions in gray levels, and this texture is a repeating pattern of transitions in gray levels, edge detection was used to detect the lines in the background netting.

There are many edge detection algorithms available, with some able to determine the direction of an edge, as well as weak and strong edges. The edge detection algorithm chosen for this project was the Sobel edge detector. This edge detector offered the ability to choose the direction of edges detected; vertical, horizontal or both. Also, some fish within the images contained some texture and with the correct gradient threshold value the Sobel edge detector did not overly detect this texture.

The threshold value that was chosen for use with the sobel edge detector was calculated using Otsu's method (Otsu 1979). Otsu's method chooses the threshold that minimizes the interclass variance of the black and white pixels. This threshold value was then multiplied by .1 to scale the threshold to a usable value with the edge detection algorithm being applied. Use of this threshold and the Sobel edge detector provided accurate separation of the background texture from the fish within the image.

The next step in the edge detection process was to attempt to reduce the amount of texture that was detected on the fish. While the steps so far have all been focusing on accurate detection of the background texture while avoiding detection of the fish texture, there was still another step that could be taken to reduce this further. One of the features of the Sobel edge detector is that it is able to detect the direction of an edge, whether it is horizontal or vertical. Most of the fish texture consisted of horizontal edges being detected. Therefore, only vertical edge detection was used with the Sobel edge detector. This reduced the amount of texture being detected on the fish, while still providing accurate detection of the background texture (Figure 3.3.3c). As mentioned in section 3.2.2, noise removal had a negative impact on the output of edge detection (Figure 3.3.3b), resulting in very poor separation between the background and the fish within the image. The benefit of histogram equalisation can also be seen in Figure 3.3.3, with shadows greatly effecting the output of edge detection in the original image without histogram equalisation (Figure 3.3.3a)

---



**Figure 3.3.3 Sobel edge detection**

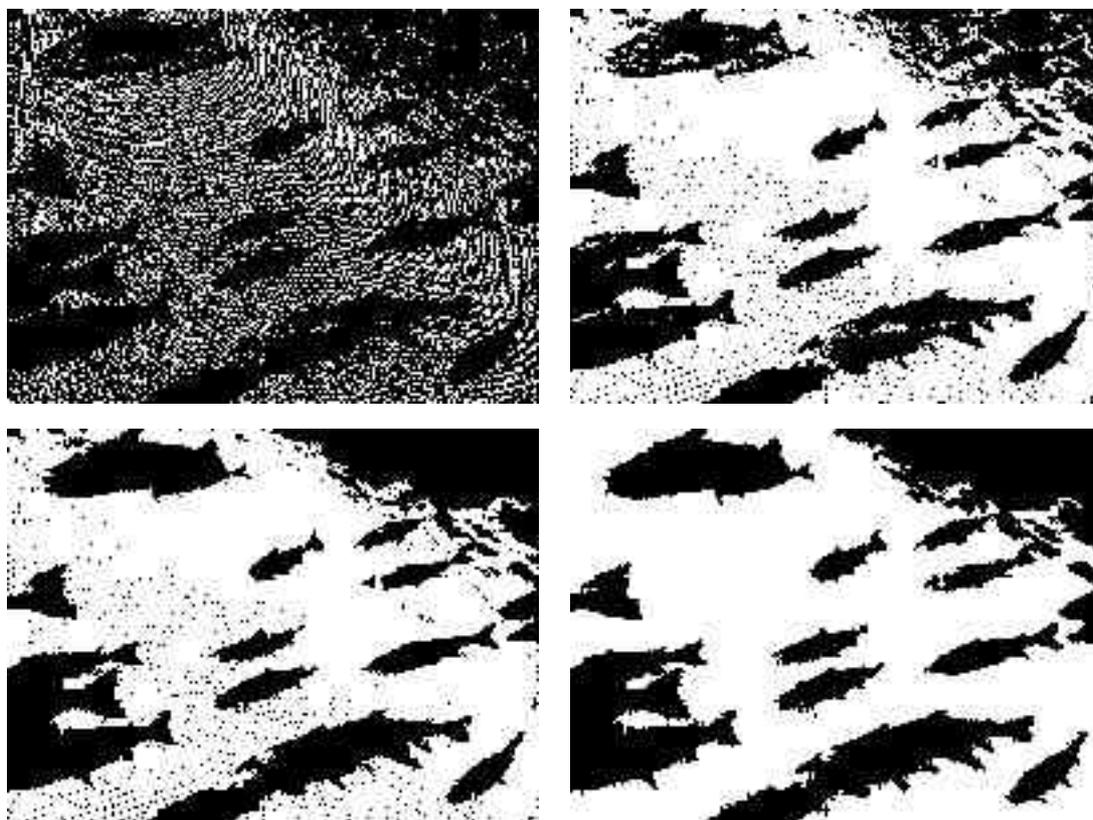
(a) Sobel edge detection on original image. (b) Sobel edge detection after image smoothing using a median filter. (c) Sobel edge detection after histogram equalisation has been performed

### 3.2.4 Initial Segmentation

Once the texture had been detected (Figure 3.3.4a), the next stage of the project was to remove the texture from the image. This background removal stage is what provided the binary segmentation of fish objects from the background. At this stage, the image consisted of a series of lines which mostly represented the background netting. The segmentation process needed to fill in the area that was the texture, while not filling in the areas that represented the fish objects within the image.

Firstly, the lines within the Sobel image were dilated using two linear structuring elements. The first structuring element represented a horizontal line with a length of three, the second represented a vertical line of length three. The dilated Sobel image now contains a filled in background texture (Figure 3.3.4b). However there are still

small gaps within this texture that will need to be filled. Also, another problem with the dilation of the Sobel image is that any texture that is detected on the fish also becomes dilated, increasing the chances that sections of fish may also be removed as part of the background.



**Figure 3.3.4 Initial Segmentation Process**

- (a) Initial Sobel image. (b) Image once edges have been dilated.
- (c) Image after attempted removal of texture within fish segments.
- (d) Removal of holes within the background netting.

The second stage of this process was to attempt to remove textures from within the segments representing the fish, while also filling in the small gaps within the background segment. Small, unconnected segments that were smaller than three hundred pixels in size were merged with the foreground segment (Figure 3.3.4c). At this stage, these small segments generally represented texture on the fish. Removing them was an attempt to reduce the amount of fish texture that was included as the background. The small gaps within the background segment were then filled, which

---

involved a process of taking the negative of the image, then again, removing any segments that were smaller than three hundred pixels (Figure 3.3.4d).

The next stage of the segmentation process involved eroding the background segmentation. This process provided a more accurate and smooth segmentation between the background and the fish. The structuring element used for the erosion process was a 3\*3 diamond, which provided a more natural segmentation between the fish and the background.



**Figure 3.3.5 Initial Segmentation Output**

Final segmentation, any region connected to the edge has been removed.

The next stage of the segmentation process was to remove the background as well as any segments that were connected to the border of the image (Figure 3.3.5). It was important to remove any segments connected to the border because these segments would not represent whole fish within the image. Performing this process also has the possibility of removing segments representing whole fish that are connected to the

---

border through other segments representing occluded fish. However, the removal of these segments was not seen as a significant problem. It is likely that they would be rejected before the final output phase anyway, if they could not be segmented to represent a single fish.

The final stage of the segmentation process was to label and obtain statistics for each separate segment. The labelling process involves determining separate segments within the binary image and labelling the segment accordingly. The labelled image contains integer values of greater than or equal to 0. Any background pixels are labelled as 0, any pixels that are part of the first object are labelled as 1, and pixels that are part of the second object are labelled as 2 and so forth.

The label image obtained was then used to obtain statistics about each object in the segmented image. The statistics used in this application were:

Area: The number of pixels in the objects region

Centroid: The x, y, coordinates of the centre of mass of the object

MajorAxisLength: The major axis length of the object

MinorAxisLength: The minor axis length of the object

Orientation: The angle in degrees between the x axis and the major axis of the object

PixelList: The coordinates of the pixels in the region

### **3.3 Analysis**

#### **3.3.1 Ellipse Fitting**

Once the segmentation process has been performed, the segments and any information obtained about each segment are usually passed on to a higher level process. This process might then use this information for tasks such as object tracking, shape matching and measuring. One of the problems with the pre-processing stage is that all segments are not guaranteed to represent individual fish. That is, some segments may represent the more than one fish.

The first step to solving this problem is to perform further analysis on the segments. The analysis in this application would need to determine whether a segment represents a single fish, multiple fish or other artifacts that have occurred during the initial segmentation process. The method chosen for this project was to perform ellipse matching with each segment. The basic shape of a salmon can be fairly accurately described by an ellipse. Therefore an ellipse offered the best potential for determining how likely it was that a segment represented a single fish.

The ellipse plotting needed to be performed for each segment in the image. Using the major and minor axis length values of the segment, an ellipse could be plotted. Using the parametric equation of an ellipse, an ellipse was plotted around the origin, such that:

$$a = \text{majorAxisLength}/2$$

$$b = \text{minorAxisLength}/2$$

The ellipse was plotted using 36 different points, which, with the available dimensions within the image, provided sufficient accuracy. These points represented an ellipse around the origin (coordinates 0, 0) using the dimensions of the segment. The next step was to apply a rotation transform to each point, using the orientation angle obtained from the segment statistics found earlier. The ellipse now correctly matched the orientation of the segment that it was plotted for. The final stage of the ellipse plotting process was to translate each point so that the points correctly matched the location of the segment. This was performed by adding the x and y coordinates of the segment centroid to each point, translating the ellipse centre to the centre of the segment.

The initial plotting of ellipses for a segmented image involved a repetitive process which would carry out the following steps for each separate segment in the image.

- (a) Obtain statistics for current segment.
  - (b) Calculate 36 points on the ellipse.
  - (c) Rotate and translate the points.
  - (d) Plot the current ellipse over the segmented image.
-

The example in Figure 3.3.6 shows the initial plotting of ellipses on a segmented image. A small square identifies the calculated centre of the segment. This example shows that ellipse fitting provides an accurate match for the location, orientation and size of each segment.

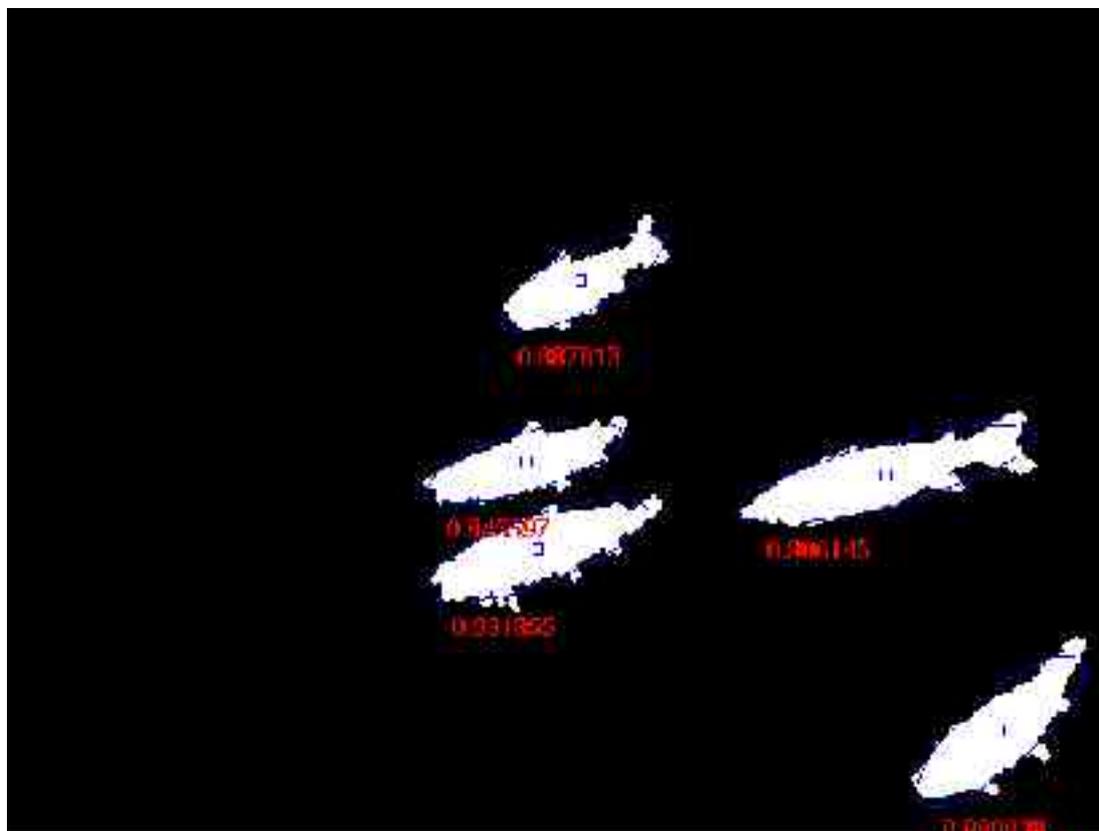


**Figure 3.3.6** Ellipses plotted over segmented image

### **3.3.2 Confidence Factor**

The previous example shows that the statistics for each segment provide a fairly accurate representation of the size, location and orientation of the segment. However, there is still no guarantee that the segment is actually representing a fish within the image. The benefit of using ellipse fitting on the segmented objects is that tests may be performed that will show the accuracy of the match. Because the basic shape of the fish relates to an ellipse quite well, segments that do not show a reasonable match

can be considered as not representing a single fish within the image and will not be included in the final output of the system.



**Figure 3.3.7 Ellipses plotted with initial confidence value**

Values from top to bottom, left to right: 0.887813, 0.946597,  
0.906145, 0.931855, 0.900038

The first approach taken to calculate a confidence factor for each plotted ellipse was to calculate the ratio between the area of the segment (the number of pixels in that region), and the area of the plotted ellipse. The area of the plotted ellipse was calculated using the function:

$$\text{ellipseArea} = \pi * a * b$$

The final ratio was calculated by dividing the area of the segment by the area of the ellipse (Figure 3.3.7).

After initial investigation into this method it was discovered that just comparing the overall areas of the segment and ellipse did not capture the key features in the ellipse matching process. In some situations there were large portions of the segment that were located outside the ellipse, while there were also ‘empty’ sections inside the ellipse. Using the above method, these mismatches with the ellipse would not be detected because the areas of the segment outside the ellipse would effectively cancel out any areas inside the ellipse that were empty.

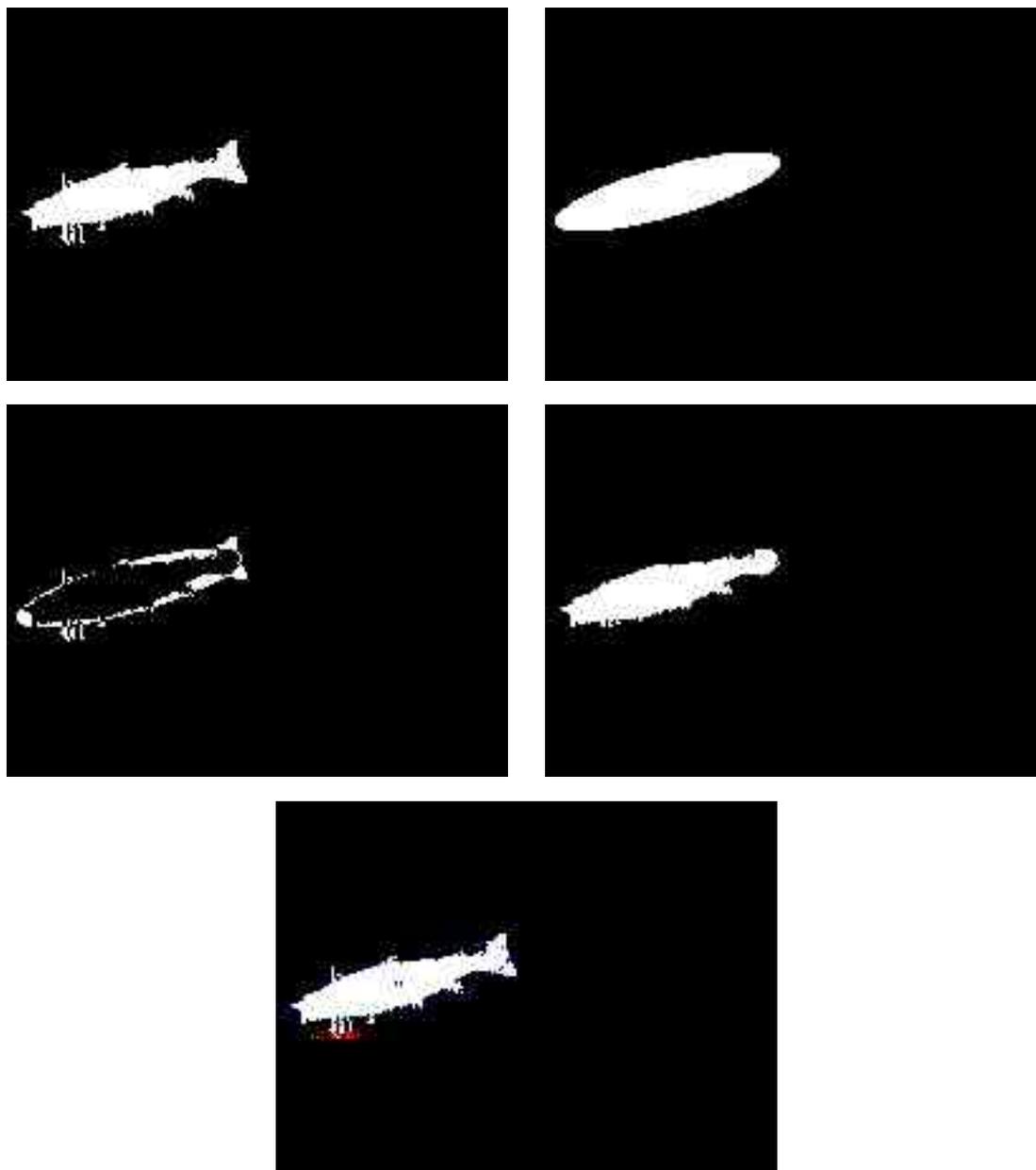


**Figure 3.3.8 Ellipses plotted with second confidence value**

Values from top to bottom, left to right: 0.559396, 0.348385

The second approach taken to calculate a confidence factor needed to address the problems with the first approach. Firstly, this approach involved calculating the total number of pixels outside the ellipse that belonged to the segment, as well as the number of pixels inside the ellipse that did not belong to the segment. The second step was to calculate the total number of pixels that belonged to the segment that were located inside the plotted ellipse. Dividing the first value by the second value

would provide a ratio that represented how well the ellipse matched the segment, while also considering the area that was not matched (Figure 3.3.8).



**Figure 3.3.9 Process of calculating confidence value**

(a) Image containing a single segment. (b) Filled ellipse calculated from a. (c) XOR result of a and b. (d) AND result of a and b. (e) Resulting confidence value: 0.352609

The first calculation consisted of an XOR operation on the segment and the filled plotted ellipse. This involved creating two separate images for each segment. One image contained the current segment without any other segments in the image (Figure 3.3.9a). The second image contained the filled ellipse for the current segment

(Figure 3.3.9b). The image containing the filled ellipse was created by using Bresenham's line drawing algorithm (Bresenham 1965) to draw lines between each of the 36 points making up the ellipse, then filling the ellipse.

An XOR image was created by taking the XOR of these two new images (Figure 3.3.9c). The number of white pixels in this image gave the number of pixels that the ellipse did not match. An AND image was created by performing an AND operation on the two images (Figure 3.3.9d). The number of white pixels in this image gave the number of pixels that the ellipse did match.

The final ratio (Figure 3.3.9e) was calculated by dividing the sum of the XOR image by the sum of the AND image. As with the first confidence factor calculated, features such as fins and the shape of the tail will usually be located outside the ellipse (Figure 3.3.9c). These features would never allow the ellipse to have a perfect match.

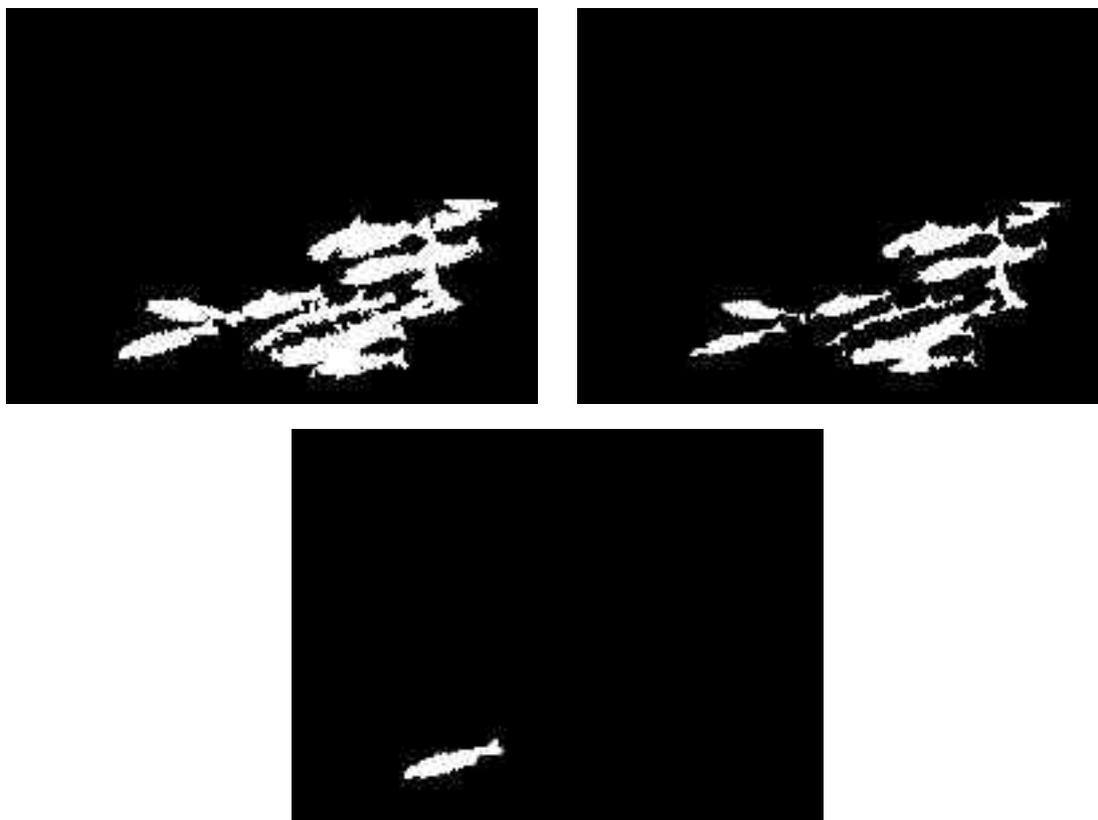
### **3.3.3 Further Segmentation**

Using the confidence value to determine if a segment is likely to be a single fish increases the accuracy of the output of this application. However, rather than throwing away segments that do not fit in this category, it may be beneficial to perform further segmentation. In some cases these segments represent multiple fish that are only slightly occluded by each other (Figure 3.3.10a). It is quite likely that the segments that are not classed as individual fish may actually be separated into single fish segments, increasing the ability of the application to locate the position of single fish within the image.

The focus of the further segmentation procedure was the separation of segments that are only connected by a small number of pixels. The method used to separate these segments was to erode the edge pixels of the segment, attempting to separate slightly joined segments. The erosion was performed three times as it allowed slightly joined segments to be separated while also maintaining the overall shape of the segment (Figure 3.3.10b). Once the erosion had been performed, each segment was placed into a separate image. Any small junk segments created due to the erosion process were removed at this stage. All other segments were thickened three times. This

---

process allowed each segment to be grown to its original size without reconnecting it to other segments (Figure 3.3.10c).



**Figure 3.3.10 Further segmentation**

(a) Image showing a segment representing multiple fish within the image. (b) Image once erosion has been performed 3 times. (c) Image showing separated segment thickened 3 times.

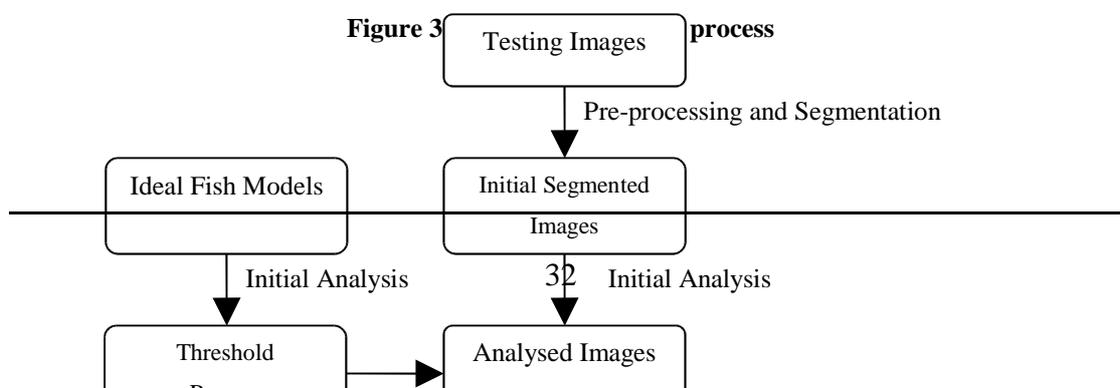
All segments that were not accepted as representing single fish during the initial analysis stage of the application were passed onto the further segmentation process. Once this process had been performed, the segments were then passed back into the analysis process. Any segments that were now accepted during the second analysis stage were included in the output of the application. Any segments that were not accepted during the second analysis stage were discarded completely.

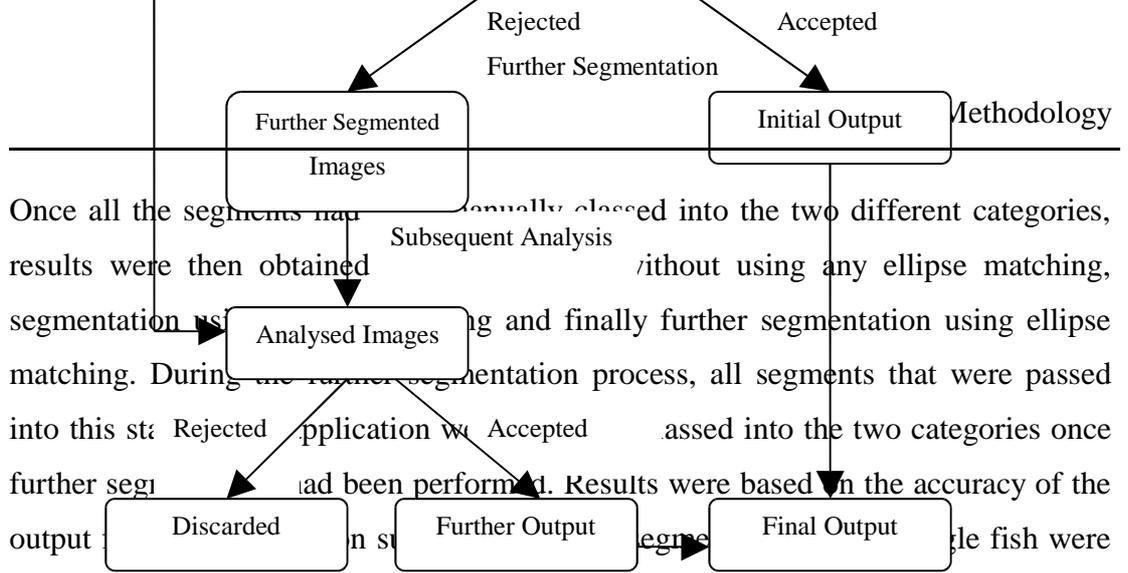
### 3.4 Evaluation

The available images were divided into three classes: good, medium and bad. The images were classed based on criteria such as number of non occluded fish, contrast, and number of fish. By dividing the images into three classes, a selection of images can be made that gives a reasonable representation of real world performance from the application. A testing set of 60 images was chosen. This testing set included an equal number of images from each class. From each class of image, 20 images were selected randomly to be included in the testing set.

The thresholds used for the analysis stage of the application were obtained by performing ellipse matching on ideal fish models obtained from another related research project involving fish matching using adaptive shape models (Kelsall 2005). Because the final aim of the project is to pass on the output of the application to other applications such as the top-down shape modelling system, obtaining thresholds this way gave an indication of what fitness range a segment representing a fish should be within.

The testing process involved firstly running the application using just the initial segmentation process on the testing set of images. Ellipses were plotted around the segments during this initial testing stage. Each segment was then manually classed as either being a segment representing a single fish, or a segment that did not represent a single fish. Segments were classed as representing a single fish within the image based on how well the plotted ellipse matched a single fish in the original image. If the ellipse was reasonably close to matching the width and height of the single fish, the segment was classed as representing a single fish. The segment would be classed as not representing a single fish if the ellipse did not match the width and height of a single fish, matched more than one fish or did not match a fish at all (an example of this is the ellipse matching a shadow).





Once all the segments had been manually classed into the two different categories, results were then obtained without using any ellipse matching, segmentation and finally further segmentation using ellipse matching. During the further segmentation process, all segments that were passed into this stage had been performed. Results were based on the accuracy of the output. Discarded segments were found and how many segments had been incorrectly identified by the system as containing single fish. The overall process of the system is illustrated in Figure 3.3.11, with results being obtained at four stages of the process. These include the initial segmented images, initial output (once initial analysis has been performed), further output (once subsequent analysis has been performed) and the final output (combined initial output and further output).

## Chapter 4

### Results and Discussion

This chapter presents five main sections for the evaluation of the final application. The first of these discusses the technique used to determine the threshold ranges for the ellipse analysis stage. The following sections discuss the results obtained during all stages of the application and evaluation. Results are shown from each stage of the application to allow careful evaluation when considering the impact each stage has on the final output of the application. Results are also shown for the two different ellipse fitness measures which allow for the two techniques to be compared in a real testing environment.

#### 4.1 Acceptance Thresholds for the Ellipse Analysis Stage

Whether a segment was sufficiently close to being elliptical in shape was decided by comparing those pixels contained within the fitted ellipse with those contained in the segment. If the shapes were similar one would expect their areas to be similar. One measure of similarity used was the Area Fitness. As mentioned in section 3.3.2, this was the ratio of the number of pixels within the ellipse compared to the number of pixels within the segment. The other measure used was the XOR Fitness, also mentioned in section 3.3.2. This involved counting the number of pixels inside the ellipse that were not part of the segment, and adding to that the number of pixels that were part of the segment but outside the ellipse. This result was then divided by the total number of pixels that were inside the ellipse and part of the segment. In one set of trials, if the Area Fitness was within a specified range, the segment was regarded as being sufficiently elliptical in shape to be accepted as representing a single fish. In

another set of trials, if the XOR Fitness was within a specified range, the segment was regarded as being sufficiently elliptical to be accepted.

The difficulty here was deciding on the thresholds to use in each case since it was not anticipated that any segments depicting single fish would be exactly elliptical in shape, because the tail and fins would naturally fall outside an elliptical outline.

#### 4.1.1 Estimating Thresholds from Ideal Fish Models

The results shown in Table 4.4.1 were gathered from performing ellipse matching on 21 ideal fish model segments. These results were then used to obtain the actual thresholds for the final output of the application.

Number Of Models	<b>21</b>
Average XOR Fitness	<b>0.253</b>
Average Area Fitness	<b>0.920</b>
Standard Deviation XOR Fitness	<b>0.044</b>
Standard Deviation Area Fitness	<b>0.021</b>
Lowest XOR Fitness	<b>0.154</b>
Lowest Area Fitness	<b>0.877</b>
Highest XOR Fitness	<b>0.342</b>
Highest Area Fitness	<b>0.966</b>

**Table 4.4.1 Ellipse matching on ideal fish models**

The technique used to estimate appropriate thresholds employed a set of 21 ideal fish models, created from real images by Kelsall (2005) using a program called the ASM Toolkit. These ideal models were subjected to the same ellipse fitting process and the range of Area and XOR Fitness values were determined for the 21 ideal shapes. Statistics were recorded for both measures (Table 4.4.1). The threshold range for both the Area and XOR Fitness measures were chosen to be from the mean value minus two standard deviations, to the mean value plus two standard deviations. This range was chosen because, assuming a normal distribution, 95% of ideal fish should fall within this range. The decision was made to obtain a threshold from segments created from ideal fish models rather than running tests on a separate set of training

images because this should allow the output to mesh with the shape modelling application more smoothly.

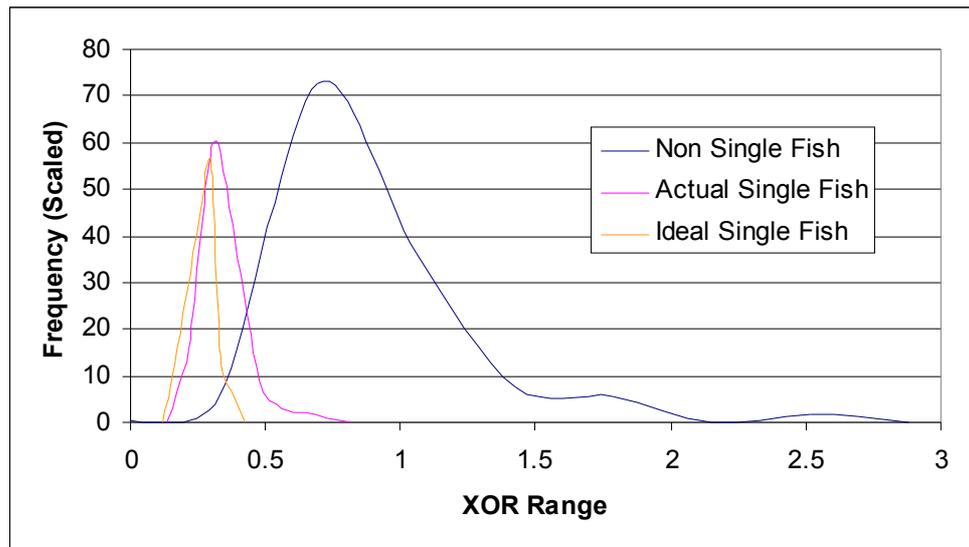
#### **4.1.2 Estimating Thresholds from Real Fish Images**

An alternative technique would be to use a set of real image segments taken from a set of training or development set of images to estimate appropriate thresholds for the two fitness measures. Because the aim of this project is to provide output for the top down shape modelling system, this technique was not used in favour of the technique discussed in section 4.1.1. However, to provide comparative results for the thresholds that were used, actual thresholds were obtained from the set of real image segments taken from the training set. As well as obtaining actual thresholds from the training set of images, statistics were also gathered on the Area and XOR Fitness measures for segments that did not represent a single fish. Obtaining results on these segments allowed Fitness ranges to be derived for segments that should be rejected by the system.

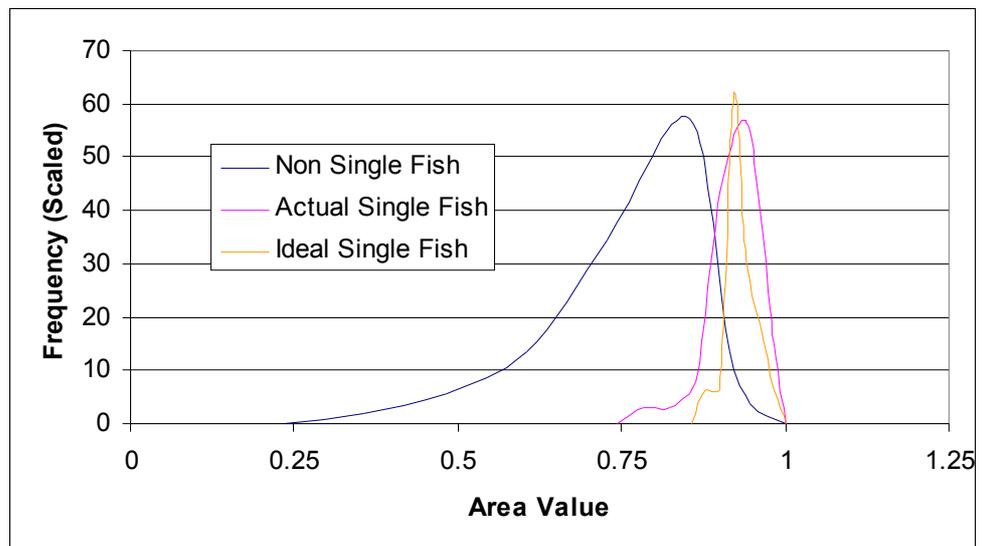
#### **4.1.3 Comparison of Threshold Ranges**

Figure 4.4.1 shows the distribution of XOR fitness values for three different classes of segments in the application. These values were obtained from the results of the testing images and ideal fish models. The sample number for ideal fish segments and non single fish segments has been scaled up to match the sample size of the actual fish segments.

Figure 4.4.2 shows the distribution of Area fitness values for the three classes of segments in the application. As with Figure 4.4.1, the values were obtained from the results of the testing images and the ideal fish models. Also, the sample number for ideal fish and non single fish has been scaled to match actual single fish.



**Figure 4.4.1** Distribution of ranges for the XOR confidence value



**Figure 4.4.2** Distribution of ranges for the Area confidence value

As can be seen from Figure 4.4.1 and Figure 4.4.2, most segments within threshold ranges for both the Area and XOR fitness for ideal fish fall outside the range for segments that should be rejected by the system.

## 4.2 Results based on the Initial Segmentation Stage

The results shown in Table 4.4.2 were obtained by performing the initial segmentation process on the 60 testing images. These results also show the actual fitness values for the segments in the final application. These results are useful because they can be used as a comparison for the actual output of the application using thresholds found from Table 4.4.1.

Testing Images	<b>60</b>
Total Segments found	<b>198</b>
Average Segments per Image	<b>3.3</b>
Total Single fish found	<b>130</b>
Accuracy	<b>65.66%</b>
Average XOR Fitness for single fish	<b>0.305</b>
Average Area Fitness for single fish	<b>0.903</b>
Average XOR Fitness for non single fish	<b>0.691</b>
Average Area Fitness for non single fish	<b>0.729</b>
Standard Deviation XOR single fish	<b>0.085</b>
Standard Deviation Area single fish	<b>0.040</b>
Standard Deviation XOR non single fish	<b>0.365</b>
Standard Deviation Area non single fish	<b>0.124</b>
Highest XOR Fitness for single fish	<b>0.666</b>
Highest Area Fitness for single fish	<b>0.971</b>
Highest XOR Fitness for non single fish	<b>2.416</b>
Highest Area Fitness for non single fish	<b>0.944</b>
Lowest XOR Fitness for single fish	<b>0.156</b>
Lowest Area Fitness for single fish	<b>0.746</b>
Lowest XOR Fitness for non single fish	<b>0.196</b>
Lowest Area Fitness for non single fish	<b>0.350</b>

**Table 4.4.2 Initial segmentation results**

The results obtained from performing the initial segmentation process on the 60 testing images in Table 4.4.2 show that 130 segments representing single fish were found. While this figure may be quite low compared to the actual number of fish within the images, in an automated system this figure may actually provide enough data due to the high number of images that will be processed. However, the initial segmentation process also found 68 segments in the 60 testing images that did not represent a single fish. This result confirms the need for the analysis stage of the application to attempt to remove segments from the final output that are unlikely to represent a single fish within the image. If these initial segments were to be used as the output of the application, there is about a 66% chance that the segment is going to

represent a single fish. This result may also be too low to be used in a higher level process such as the shape matching that will be using the output of this application. If 34% of segments passed on to the shape matching application were not single fish segments, it is possible that the accuracy and efficiency of the shape matching process will be lowered substantially.

These results also provide some interesting information on the variation of the segments that were obtained. While the means and standard deviations of both thresholds used provide a range that a particular class of segment (single fish or non single fish) is likely to be, the minimum and maximum of these values found for both classes of segment also provide an interesting result. An example in Table 4.4.2 can be taken from the XOR fitness measure results, where the mean for single fish is close to .3 and the mean for non single fish is close to .7. From these results, it would be expected that segments that have an XOR fitness value closer to .3 would indicate that the segment represents a single fish within the image, and an XOR fitness value closer to .7 would indicate that the segment represented a non single fish within the image. This may be the case for most segments; however, there are some segments where this is not the case. One example is the minimum XOR fitness value for a non single fish, which is close to .2, which would then be expected to be a single fish segment, which would be incorrect. The same is true for the highest XOR fitness value for a single fish segment, which is close to .67, which would then be expected to be a non single fish segment, which would also be incorrect. These extreme values make it difficult to achieve total accuracy when classifying segments. These values occur for a number of reasons, including: inaccurate segments, segments representing fish swimming towards or away from the camera and segmented shadows resembling the shape of a fish. Also, another problem is introduced when segments representing two or more fish are shaped in such a way that it appears as though it is representing one fish. However, substantial improvements can still be made by using the fitness measures to filter the output of the initial segmentation.

### **4.3 Results Based on the Initial Analysis Stage**

Table 4.4.3 compares the results from using ideal and actual threshold ranges to determine if a segment represents a single fish. The thresholds used were a range

---

using mean  $\pm$  2 standard deviations. The second column in Table 4.4.3 (Actual) uses thresholds obtained from the mean and standard deviation of single fish segments from Table 4.4.2. The third column in Table 4.4.3 (Ideal) uses thresholds obtained from the mean and standard deviation of the ideal fish segments in Table 4.4.1. Both the XOR and Area fitness values are shown here.

<b>Using Fitness Measure</b>	<b>Actual</b>	<b>Ideal</b>
XOR Threshold Range	<b>0.134 - 0.475</b>	<b>0.165 - 0.342</b>
XOR Total segments found	<b>142</b>	<b>95</b>
XOR Total single fish found	<b>124</b>	<b>91</b>
Area Threshold Range	<b>0.824 - 0.982</b>	<b>0.878 - 0.962</b>
Area Total segments found	<b>142</b>	<b>104</b>
Area Total single fish found	<b>124</b>	<b>98</b>
XOR Fitness Accuracy	<b>87.32%</b>	<b>95.79%</b>
Area Fitness Accuracy	<b>87.32%</b>	<b>94.23%</b>

**Table 4.4.3 Ideal and Actual threshold ranges**

As discussed in the previous section, there was definitely a need to perform analysis on the segments in an attempt to remove the segments that were not representing the location of a single fish within the image. The actual output of the application used the XOR threshold obtained from the results in Table 4.4.1. As discussed in chapter 3, the XOR fitness measure should provide a clearer indication of how well the ellipse is matched to the segment. While the XOR fitness measure was used for the actual output of the program, results were also gathered using the Area fitness measure for the Ideal fish models as well as XOR and Area fitness measures for the testing images. These results were then compared to allow the performance of the XOR fitness measure to be rated against the Area fitness measure.

As can be seen in Table 4.4.3, the XOR threshold obtained from the ideal fish models performed the best when measuring the probability that a fish segment is actually representing a single fish within the image, with the Area threshold performing very slightly lower. However, the results shown from using thresholds obtained from the actual testing images show that the XOR and Area fitness measures perform equally well. This shows that the Area fitness measure may actually be just as effective at

determining if a segment represents a single fish within the image. However, this application is not necessarily trying to output every segment representing a single fish. The purpose of this application is to output segments that are more likely to be matched by the shape modelling application, which is where the XOR fitness measure slightly outperforms the Area fitness measure.

Figures 4.1 and 4.2 illustrate the reason why using the XOR fitness measure provides better accuracy than the Area fitness measure. In Figure 4.4.1, the distribution of the XOR values for the ideal fish models is fairly clearly separated from the distribution for non single fish segments. In Figure 4.4.2, the distribution of the Area values for the ideal fish models seems to intersect the distribution for the non single fish segments at a higher point than seen in Figure 4.4.1 for the XOR values. However, the distribution of XOR values for ideal fish models are also further separated from the distribution of XOR values for actual single fish segments than the Area values in Figure 4.4.2. While this shows why using the XOR fitness measure was more likely to output segments representing single fish, it also had the lowest output of these segments. Although using the XOR fitness measure throws away slightly more segments that do represent single fish than the Area fitness measure, the aim of the application is to output only segments that represent single fish. It is also quite possible that many of the segments that get rejected that did represent a single fish would not have been matched by the shape modelling application. Without further testing between the two applications this balance may be hard to obtain.

#### **4.4 Results Based on the Subsequent Analysis Stage**

The results shown in Table 4.4.4 were obtained by performing further segmentation on the 60 testing images. The results include the total number of segments and single fish segments found, the results after the XOR threshold was applied, and the results of the final total output of the application.

Total Segments Used from initial segmentation	<b>103</b>
Total Segments Found	<b>99</b>

Total Single Fish Found	<b>49</b>
XOR Total segments found	<b>35</b>
XOR Total single fish found	<b>30</b>
Accuracy	<b>85.71%</b>
Total Including initial Segmentation	
XOR Total Segments	<b>130</b>
XOR Total single fish found	<b>121</b>
Accuracy	<b>93.08%</b>

**Table 4.4.4 Further segmentation results**

Results gathered from the final process of the application using the XOR fitness measure show the effect of performing further segmentation. From the data in Table 4.4.4, it can be seen that 103 segments were passed onto the further segmentation stage. Of these 103 segments, 39 were classed as representing a single fish within the image. It may be expected that the further segmentation process would result in a larger number of segments found than the number of segments used as input. However, as can be seen in Table 4.4.4, this is not the case. The total number of segments produced from the further segmentation process was 99, which has decreased from the 103 segments that were used as input. Also, the number of segments representing a single fish has increased from 39 to 49, showing that there has been an increase in the number of separate segments in the images. The explanation of why the total number of segments decreased when the further segmentation process should be producing more segments is briefly mentioned in section 3.3.3. Due to the erosion process used, some segments may become too small and be discarded completely before the second analysis stage. Therefore, any segments that are small to begin with that are then passed onto the further segmentation stage, will become too small and be discarded during the erosion process.

The increase in the number of segments found that represent single fish through the use of the further segmentation process shows the usefulness that this process has to offer. Although the increase in the number segments found that represent a single fish is 10, it is possible that the number of these segments found due to the further segmentation process is higher than 10. The reason for this has been discussed above, with the erosion process potentially removing existing segments if they become too

small. The actual output of the further segmentation process that is also shown in Table 4.4.4 using the XOR fitness measure shows that 35 segments were added to the final output of the program, with 30 of those segments representing a single fish within the images. This shows that the accuracy of the output remains high while also adding a substantial number of single fish locations to the output. While this result may be positive, it may be the case that the segments included in the output after further segmentation may be less ideal than the initial output. Because of the further segmentation process, the segments become more smoothed than in the initial segmentation. This smoothing may lead to further inaccuracies in detecting an ideal match based on the thresholds obtained from Table 4.4.1.

## 4.5 Results Based on Complete Process

Figure 4.4.3 shows the segments that would be included in the output at various stages of the segmentation process. Initial segmentation results below show the type of segments found before any analysis has been performed. The second set of results below show the output from the initial segmentation process once analysis has been performed using the XOR fitness thresholds found from Table 4.4.1. The last set of results show the final output of the application after further segmentation has been performed.

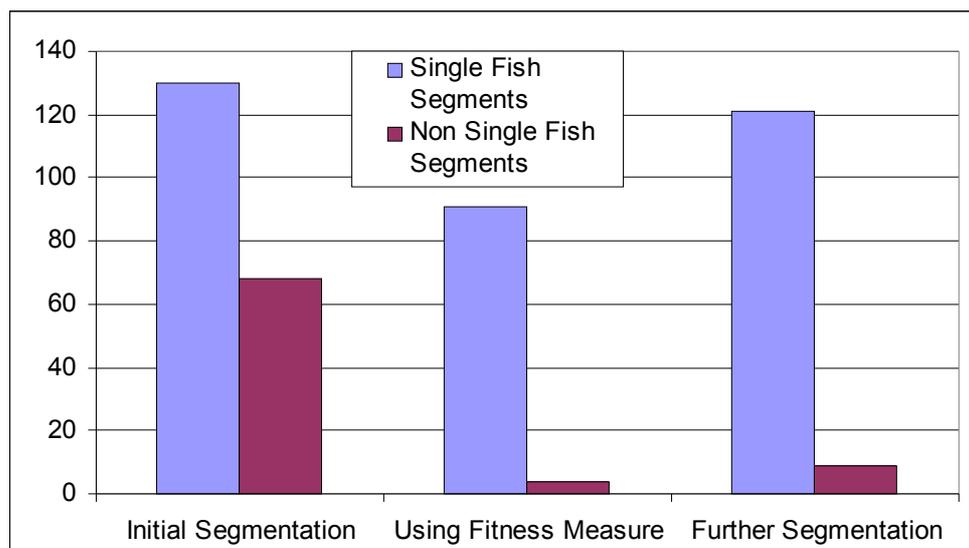


Figure 4.4.3 Initial segmentation to final output

The results from the various stages of the application have been discussed so far, showing positive results. However, the most significant result can be seen by looking at the final output of the application and comparing it to the output at the two previous stages. Figure 4.4.3 compares the output at each stage in the application, showing that each stage adds a further improvement to the previous stage. The output from the initial segmentation includes too many segments representing non single fish locations (false positives) within the images to be useful in a further analysis application. To improve this, the segment analysis stage of the application was necessary. This was successful in achieving an acceptable level of false positives in the output. However, the number of actual single fish locations included in the output was also reduced. The last stage of the application offers an increase in this number, while also keeping the number of false positives to an acceptable level.

## Chapter 5

### Conclusion and Future Development

#### 5.1 Conclusion

This project sought to determine whether suitable image processing techniques can be used to process underwater video images, allowing the detection of objects within these images for use with further scientific analysis applications.

With the increase in the use of under water video images comes an increase the time it takes to manually process these images. For this reason, automation is needed in the processing of underwater videos. For this project, video image sequences were obtained from a local fish farm, on which an application was designed that would allow the hypothesis to be tested in a real underwater environment. This required research into a variety of image processing techniques and current underwater vision projects. Chapter 2 discussed these areas, with the focus on techniques thought suitable for the image sequences available to the project. An application was developed that combined the most suitable techniques with the aim of being able to detect the locations of fish within the image sequences.

The development of the application provided an initial confirmation of the hypothesis - that image processing techniques can be used to process underwater image sequences. However, it is important to realise that these image processing techniques are also heavily affected by conditions caused by the underwater environment such as, poor contrast, varying illumination, and little colour variation. Because of this an image processing system in an underwater environment must take advantage of any features in the image that can be used to help overcome these problems. An example in this project is the texture of the netting in the background, which could be detected even with varying illumination. Although the conclusion is that image processing

---

techniques can be used to process underwater images, they still suffer significant limitations reducing the effectiveness of such techniques.

The results reported here also confirm the hypothesis – that image processing techniques do allow the detection of objects in underwater images. The results from just performing pre-processing and segmentation show that an acceptable number of single fish locations were found, while using subsequent analysis to reject unsuitable segments significantly reduces the number of incorrectly identified objects. This number is seen as acceptable because in a real application, there would be large numbers of images which analysed providing statistically significant results even when only some of the fish in a specific image have been identified.

The analysis stage of the application also provides another answer to the hypothesis – that the objects found can then be used in further scientific analysis applications. Although results were not included in this paper from any further analysis applications, significant progress has been made towards integrating this system with further analysis applications. Output from this application has been tested with an adaptive shape modelling application (Kelsall 2005). Assuming that the adaptive shape modelling system could not easily distinguish between a relevant object and one that is not, the low number of false positives in the output provides efficient output data for use with testing. The analysis stage of the application provided the information needed to estimate the location of the fish within the image for the adaptive shape modelling application. Results show that there is great potential for the two projects to be used together.

The final conclusion from this project is that automation of underwater video images is needed and is possible. Automatic processing of large numbers of video image sequences will be of great benefit in the future. It will allow scientific researchers to study the ocean environment and its species more effectively. Pre-processing is an essential component of the overall process that will lead to automation of underwater video data analysis for marine science applications.

## 5.2 Future Development

While a significant amount of progress has been made towards the use of this application with further analysis applications, such as the top-down shape modelling system, to achieve automated image analysis for fish sizing, there has been little collaboration between the two projects so far. Integration between the two projects in the future will be another important step towards achieving automated (or semi-automated) image analysis for fish sizing. This integration would allow further feedback to be obtained. This feedback could provide insight into some extra segment analysis techniques. Currently, the output is determined by attempting to determine how likely it is that a segment represents the location of a single fish in the image. However, integration with the top-down shape modelling system will provide more specific details, such as the range of orientation, the ratio of length compared to height and other information that may indicate the direction in which the fish is swimming. This extra feedback may help to discard segments that represent fish that are facing towards or away from the camera, or other segments that will not provide good input to the shape modelling system.

An assumption has been made that it is better to only include segments in the output of this application that are likely to provide an ideal match with the shape modelling system. However, this assumption is based on the fact that currently the higher level shape modelling system does not determine whether it has correctly matched a fish or not. If the higher level analysis system was able to determine this, it may actually prove to be beneficial to include a higher number of segments in the output. This would include segments representing the location of single fish as well as those that did not. However, as already mentioned, if the higher level analysis system could distinguish between a match and a non match this may not be a problem.

It has been stated that the number of single fish locations found should provide enough statistical data over a large quantity of images. However, to truly test this significance, further testing will be needed on a large number of images. The number of images tested in this project was only 60. While this number was enough to test the performance of the techniques used in the system, testing on a much larger set of images will provide conclusive results on the ability of an automated system to

---

provide enough statistical data for meaningful measurements to occur. However, at this stage it will be more beneficial to focus on the integration of the two projects and creating accurate output before large testing sets should be used.

For the duration of this project, there has been little input into the system from the actual manual process of analysing the images. While this has not created a problem so far, due to this project focusing on the pre-processing of these images, as the application is integrated with the top-down shape modelling system to form a complete analysis system in the future, more input will be needed. It may also be interesting to know how flexible camera positions can be. A change in camera position may provide further benefit for the image processing techniques used, which would in turn provide a benefit to the system as a whole. In the current images, the distance between the background netting and the camera increases towards the top right corner (Figure 3.3.2). This increase in distance leads to a lower performance of the image processing techniques used in this project (Figure 3.3.3). This problem may be solved by a change in camera position.

## References

- Ballard & Brown 1982, *Computer Vision*, Prentice Hall.
- Bresenham, JE 1965, 'Algorithm for computer control of a digital plotter', *IBM Syst.*, vol. 1, pp. 25-30.
- Chen, Y, Barcelos, CAZ & Mair, BA 2001, 'Smoothing and Edge Detection by Time-Varying Coupled Nonlinear Diffusion Equations', *Computer Vision and Image Understanding*, vol. 82, no. 2, pp. 85-100.
- Fahnestock, JD & Schowengerdt, RA 1983, 'Spatially variant contrast enhancement using local range modification', *Optical Engineering*, vol. 22, pp. 378-81.
- Gauch, JM 1999, 'Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies', *IEEE Transactions on Image Processing*, vol. 8, no.1.
- Gonzalez, RC & Woods, RE 2001, *Digital Image Processing (2nd Edition)*, Pearson Prentice Hall.
- Gonzalez, RC, Woods, RE & Eddins, SL 2004, *Digital Image Processing using MATLAB*, Pearson Prentice Hall.
- Heikkila, M, Pietikainen, M & Heikkila, J 2004, 'A Texture-based Method for Detecting Moving Objects', *British Machine Vision Conference*, pp. 187-96.
- Itti, L, Koch, C & Niebur, E 1998, 'A model of saliency-based visual attention for rapid scene analysis', *IEEE PAMI*, vol. 20, pp. 1254-69.
- Jaffe, JS 1990, 'Computer Modeling and the Design of Optimal Underwater Imaging Systems', *Ieee Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101-11.
- KaewTraKulPong, P & Bowden, R 2001, 'An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection', *Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems*.
- Kelsall, A 2005, 'Flexible Shape Models for Marine Animal Detection in Underwater Images', Honours thesis, University of Tasmania.
- Kim, H-S, Kang, W-S, Shin, J-I & S-H., P 2000, 'Face detection using template matching and ellipse fitting.' *IEICE Transaction on Information and System*, vol. E83-D, no. 11, p. 2008-11.
-

- Otsu, N 1979, 'A Threshold Selection Method from Gray-Level Histograms', *IEEE Trans. Systems, Man, and Cybenetics*, vol. SMC-9, no. 1, pp. 62-6.
- Rife, J & Rock, SM 2001, 'Visual Tracking of Jellyfish in Situ', *Proc. of Int. Conf. of Image Processing*, vol. 1, pp. 289-92.
- Schechner, Y & Karpel, N 2004, 'Clear Underwater Vision', *Computer Vision & Pattern Recognition*, vol. 1, pp. 536-43.
- Semani, D, Saint-Jean, C, Frelicot, C, Bouwmans, T & Courtellemont, P 2002, 'Alive Fishes Species Characterization from Video Sequences', *Lecture Notes in Computer Science*, vol. 2396, p. 689.
- Stark, AJ 2000, 'Adaptive Image Contrast Enhancement Using Generalizations of Histogram Equalization', *IEEE Transactions on Image Processing*, vol. 9, no. 5.
- Walther, D, Edgington, DR & Koch, C 2004, 'Detection and Tracking of Objects in Underwater Video', *IEEE International Conference on Computer Vision and Pattern Recognition*.
- Wijewickrema, SNR & Paplinski, AP 2004, *Principal component analysis for the approximation of an image as an ellipse*, Technical Report, Computer Science and Software Engineering, Monash University, Australia.
- Wilson, A 2003, *First steps towards autonomous recognition of Monterey Bay's most common mid-water organisms: Mining the ROV video database on behalf of the Automated Visual Event Detection (AVED) system.*, Technical Report, MBARI.
- Zhu, H, Chan, F & Lam, FK 1999, 'Image Contrast Enhancement by Constrained Local Histogram Equalization', *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 281-90.

## **Appendix A**

### **Application Details**

#### **A.1 Application CD**

There are five directories available on the sample CD: 'Images', 'Initial', 'Further', 'Source' and 'Data'.

The 'Images' folder contains the original images obtained from AQ1 Systems.

The 'Initial' folder contains output images from the testing set after the initial analysis stage.

The 'Further' folder contains output images from the testing set after the subsequent analysis stage.

The 'Source' folder contains the MATLAB source code for the application.

The 'Data' folder contains all result data and output of the application.

#### **A.2 Application File List**

The description of each MATLAB source file for the application is listed here for easy reference. Further detail can be found from the source code on the CD-ROM supplied with this document.

### **A.2.1 RESULTSCRIPT.M**

Contains a list of all the testing images. Each image is read in and passed on to the segmentation stage of the application. Once this has been performed, the resulting image is then passed on to the analysis stage of the application.

### **A.2.2 SEGFISH.M**

Performs the initial segmentation process of the application. The output from this process is a labelled segmented image.

### **A.2.3 STATFISH.M**

Performs the initial analysis stage of the application. All properties of each segment are obtained and an ellipse is plotted for each segment. If the segment is not accepted, it is passed on to the further segmentation stage. If the segment is accepted, output is written.

### **A.2.4 ERODEFISH.M**

Performs the further segmentation stage of the application. Once this has been performed the segments are then passed onto the subsequent analysis stage.

### **A.2.5 STATFISH2.M**

Performs the subsequent analysis stage of the application. If the segment is accepted, output is written. If the segment is rejected, no further processing is performed on that segment.

### **A.2.6 DRAWELLIPSE.M**

Given a list of coordinates plotting an ellipse, draws an ellipse by plotting lines from each point then filling the ellipse.

**A.2.7 POINTS.M**

Contains a list of points representing the 21 ideal shape models used to calculate the threshold values.

**A.2.8 DRAWFISH.M**

Uses the points from POINTS.M to draw a filled ideal fish model on an image and passes the image to the analysis stage of the application. Used to obtain the thresholds from the ideal fish models.