A UNIFIED RECOGNITION AND STEREO VISION SYSTEM
FOR
SIZE ASSESSMENT

By
Andrew Naiberg
B.Sc. (Physics), Queen's University

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in
THE FACULTY OF GRADUATE STUDIES
COMPUTER SCIENCE

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

July 1994
© Andrew Naiberg, 1994
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Computer Science

The University of British Columbia
Vancouver, Canada

Date July 18, 1994
ABSTRACT

This paper presents a unified recognition and stereo vision system which locates objects and determines their distances and sizes from stereo image pairs. Unlike other such systems, stereo information is not the input to the recognition stage. Instead, recognition is performed first and its output forms the input to stereo processing. This permits successful analysis of images captured in poor conditions (murky, specularities, poor camera alignment) and reduces time requirements.

Model-based recognition is accomplished in two stages. The first stage seeks feature matches by comparing the absolute orientation, relative orientation and relative length of each image segment to those of the model segments in order to find chains of adjacent segments in the image which match those of the models. The absolute orientation constraint can be suppressed to permit identification of randomly-oriented objects.

The second stage verifies candidate matches by comparing the relative locations of matched image features to the relative locations of the corresponding model features. The models themselves are generated semi-automatically from images of the desired objects.

In addition to providing distance estimates, feature-based stereo information is used to disambiguate any multiple or questionable matches.

Although the system and issues presented herein are quite general, the discussion and testing are primarily related to the motivating task of non-invasively assessing the size of sea-cage salmon.
Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

Acknowledgements vi

1 Introduction 1
  1.1 The NIFE System: Non-Invasive Fish Enumeration 2
  1.2 The Scope of this Thesis 4
  1.3 Thesis Overview 7

2 Previous Work 8
  2.1 Previous Systems Related to Aquaculture 8
  2.2 Edge Detection 10
  2.3 Recognition 12
  2.4 Stereo 18
     2.4.1 Correlation 19
     2.4.2 Feature-Based Stereo 21

3 The Manual Stereo System 22
  3.1 Acquiring Stereo Images 22
  3.2 Camera Correction and Calibration 23
  3.3 Size Assessment 27
  3.4 Results 28
## The Automatic System: Design and Implementation Details

### 4.1 Design Decisions
- 4.1.1 Model-Based Recognition
- 4.1.2 Stereo

### 4.2 Implementation Details
- 4.2.1 Pre-Processing
- 4.2.2 Model Creation and Use
- 4.2.3 Matching
- 4.2.4 Verification
- 4.2.5 Stereo

### 5 Results
- 5.1 Example #1
- 5.2 Example #2
- 5.3 Example #3
- 5.4 Example #4
- 5.5 General Observations
- 5.6 Complexity

### 6 Conclusions & Future Work
- 6.1 Conclusions
- 6.2 Future Work

## Appendices

### A Binocular Stereo Geometry

### B Imaging Geometry

## Bibliography
## List of Figures

1.1 The stages of processing ........................................ 5
2.1 The need for super-segments .................................... 15
2.2 A super-segment for Structural Indexing ...................... 16
3.1 Apparatus for gathering images ................................. 24
3.2 Correction of lens distortion .................................... 25
4.1 Modified super-segment .......................................... 34
4.2 Polygonal approximations ....................................... 35
4.3 Schematic of procedure .......................................... 39
4.4 Segmentation of a curve ......................................... 40
4.5 Sample segment information ..................................... 42
4.6 Handling discontinuous contours ............................... 50
4.7 Models with shifted centreponts ................................. 58
5.1 Images for first example .......................................... 66
5.2 Images for second example ....................................... 69
5.3 Images for third example ......................................... 71
5.4 False matches due to similar shapes ............................ 72
5.5 Images for fourth example ........................................ 73
6.1 Missing segments during matching ............................... 81
A.1 Stereo imaging geometry ......................................... 85
B.1 Imaging Geometry .................................................. 87
I was a less than serious student in college. If I had it to do over again, I would be far more serious. I did play a lot of golf. But I don’t think that’s any reflection on my ability to lead this nation.

- U.S. Vice President Dan Quayle

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Jim Little, for his continued support, guidance and easy-going demeanour despite my frequent deviation from the straight and narrow. I also express my gratitude to the second reader, Dr. David Lowe, who first stimulated my interest in computer vision and provided sound advice throughout.

A chorus of thank-you’s to all who made it fun: Bill Gates, Michele & Sara McIntosh, Larry Palazzi, Lynette Alderson, Andrew Beatty, the Subaru & the mother of all tapes, Andrew Robinson, Hubert Lai, Rob Burton and the 14, Roz Shafran, the Jericho Laser fleet, Dan Kovacs, Jen Butler, Homer Simpson, Colin Savage, Trev Neufeld, Ted Charlton, Whistler/Blackcomb, the Blackcomb Ski School desk (especially John Riedl and Susie Price), rejects & rejectors too numerous to mention, Mom, Dad & the family, Gilli and all those who helped along the way.

Finally, I would like to thank the Department of Computer Science and the British Columbia Science Council, whose support allowed me to maintain my opulent lifestyle of the oppressed and downtrodden.
CHAPTER 1

INTRODUCTION

Last year I went fishing with Salvador Dali. He was using a dotted line. He caught every other fish.

- Steven Wright

Vision provides humans (and most animals) with a staggering amount of information about their surroundings. A passing glance at a scene can provide an observer with a detailed account of the objects in the scene, their locations, their states of motion with respect to the observer and the spatial relationships between them. Furthermore, all this can be accomplished even in the presence of unfamiliar objects, random orientations and configurations, unfavourable lighting, varying degrees of overlap and similar hindrances.

Biological vision systems derive this information from many sources. Objects may be recognized based on their shape, colour or context while distances may be calculated from stereoscopic information or inferred using such clues as occlusion, relative sizes and familiarity. Still more information is available from lighting and shadows, motion and the scale of visible detail. Despite the amount of information available and the number of sources, biological vision systems interpret scenes rapidly and effortlessly.

The development of successful computer-based vision systems, in contrast, has been neither rapid nor effortless. In fact, continued research has underscored the complexity of the world and the attendant intricacies of trying to interpret general visual information. In an unconstrained environment, many factors such as uneven lighting, coincidental alignments and marked changes in the appearances of objects as they are rotated can give rise to confusing or misleading information; the correct interpretation often requires the assimilation of several sources of information at various scales. In short, the visual world is often an inhospitable place.
Nonetheless, guided by both continued study of biological vision systems and consistent thought, researchers in computer science have devised many techniques which accomplish important steps in the vision process. Also, although no existing computer-based vision system even approaches the generality, speed and robustness of human vision, we do have useful computer systems that tackle various aspects of the vision process and/or derive specific information from images under restricted circumstances.

This thesis describes the development of a non-invasive size assessment system using model-based recognition and stereoscopic vision. The term “non-invasive” means that no specialized handling or presentation of the objects is required for size assessment; the objects are simply videotaped as they would typically be found and size estimates are obtained from the information on the resulting videotape alone.¹

Although the system was designed to be as general as possible and is hence suitable for generic size-assessment applications, it was written as part of a project to assess the size of net-pen salmon non-invasively. For this reason, the system is described and tested in the context of this application.

1.1 THE NIFE SYSTEM: NON-INVASIVE FISH ENUMERATION

Fish size is an important factor affecting the economy of any aquaculture enterprise. Size information can be used to monitor the growth of the fish, determine the amounts of feed and medication required and would aid in deciding when to grade, sort or harvest the fish. However, despite its economic importance, fish size in tanks and sea cages is measured infrequently because most available methods of measurement are time-consuming, labour-intensive, insufficiently accurate and harmful to the fish.

Most current measurement techniques involve measuring a sample of approximately 200 fish from the population, either by capturing the fish and measuring them above water or by videotaping the fish as they are shuttled through a narrow portal and taking measurements from

¹This, however, does not preclude either strategic placement of the cameras to capture the best possible images or the inclusion of a custom background to enhance image contrast or clarity.
the videotape. Many farmers have reported error on the order of 15-25% using these techniques ([Klo93, PMO+86]) where they would like size information within 5% of actual values ([Klo93]). The large error has been attributed to several factors: the sample may be too small to provide an accurate representation of the entire population, the sample may be skewed because some fish are more easily captured than others and the measurements may reflect inaccuracy resulting from human error.

In addition, these methods cause the fish considerable stress and damage ([Klo93, PMO+86]). In a population of salmon, stress results in lower disease tolerance and an increase in mortalities ([SS78, Sch82]). In addition to stress, methods requiring capture or that the fish be shuttled through a narrow portal cause bruising and scale loss as a result of collisions and handling, thereby reducing the value of the fish.

Finally, as a result of these problems and the required labour, size assessments are performed infrequently. Whereas farmers would like to monitor growth on a weekly basis, practicality dictates that assessments be performed when the fish are being handled for other reasons, such as when harvesting, changing nets, grading or sorting.

In response to these problems, the Department of Bio-Resource Engineering at the University of British Columbia set out to develop a system capable of providing reliable size distribution information without harming the fish or, indeed, encroaching upon them in any way. Proposed by Trev Neufeld in [Neu83], the goal of the system was to provide Non-Invasive Fish Enumeration, thus it became known as the NIFE system.

The proposed system was to accomplish its goal by videotaping the fish as they swim freely in their enclosure and calculating the size of individual fish which appear in the images. The size of each fish assessed would then be added to a database whence a statistical size distribution would be derived after enough data had been collected. Because all necessary information would be obtained from the videotape, the fish would not have to be manipulated in any way. Furthermore, if the system were fully automated, fish farmers could update their information inexpensively and frequently without risking harm to the fish.

Conceptually, the automated system operates by locating in stereo image pairs the salient features which delimit the desired dimensions of the fish. To determine the length of a fish,
these features are the head and tail; the breadth of a fish may be delimited by its dorsal and ventral fins. Once the apparent dimensions of the fish have been obtained from these features, the distance to the fish is calculated from the stereoscopic information. Given the apparent dimensions of an object and the distance to that object, its actual dimensions follow straightforwardly. It is expected that a relationship can be found between the exterior dimensions of a fish and its biomass so that the system can provide a biomass distribution rather than a length distribution. The stages of processing are shown in Figure 1.1.

The system was developed in stages. The first stage was to design and assemble the necessary video equipment and examine precisely what information can be derived from still images of fish. The results of this first stage indicated that the system would require stereo video in order to deliver the desired information. Thus the second stage was to develop a manually-operated prototype stereo system in order to illustrate the technique and test the results.

The ultimate goal of the NIFE project is a fully automated system which would capture stereo images from videotape, locate as many fish as possible in the images, determine their lengths and biomass and enter the information in a database for statistical analysis, all without external assistance. This would allow a fish farmer to videotape the fish during the day, initiate the computerized evaluation process in the evening and view the data the next morning. The system would stop analyzing image pairs automatically once a predetermined number of fish or images had been analyzed, as determined by statistical requirements.

1.2 The Scope of this Thesis

As was mentioned earlier, vision systems typically have great difficulty operating in unconstrained environments. The specific requirements of the NIFE system do little to simplify the situation. The first complication is the marine environment: images captured underwater frequently exhibit either low contrast in murky conditions or pronounced back-scattering and uneven illumination in brighter conditions. In addition, there is always the possibility of a physical contaminant such as silt or algae further obscuring the images.

Fish are themselves not an easy object to recognize despite having distinct features. First-off, fish generally school in dense groups, thus images of fish are often complicated by overlap,
Figure 1.1: The stages of image processing: (a) The edge detector has traced the outline of one fish in each image and some noise. (b) All contours returned by the edge detector have been approximated using linear segments. (c) The recognition system has identified potential heads and tails in both images and omitted all other contours. (d) The verifier has eliminated the falsely-matched features and the stereo procedure has calculated the disparity $d$ of both the head and tail and two estimates of the apparent length $l$, one from each image.
hidden features and incomplete outlines (see Figure 5.5a). Secondly, the surface of fish can give rise to specular reflections that complicate the edge-detection process.

However, even if complete outlines were guaranteed, there are still the problems of size and shape. Because fish can be found at any stage of growth, size is an unknown which complicates the determination of distance or scale rather than a constant from which distance or scale can be calculated. Finally, recognition is hindered by the fact that not all fish look alike: even within a single species, biological variation ensures that different fish often have slightly different shapes.

As a result of these complications, this thesis does not attempt to solve the general problem of assessing the size of random three-dimensional objects in arbitrary orientations. Instead, this thesis takes the more successful approach outlined earlier, that of extracting specific information under restricted circumstances.

The first of these restrictions specifies a requirement of the images themselves. It is assumed that the images being analyzed are of sufficient quality that a good edge detector can extract highly-connected contours; the recognition system relies upon extended features derived from these contours. It is also implicitly assumed that images are easily collected, thereby allowing problematic images to be discarded without fear of being left with too few.

The remaining assumptions specify the nature of the objects being analyzed. It is assumed that the objects being measured are fairly rigid and are being imaged from restricted viewpoints, generally such that the desired measurements are roughly parallel to the image plane, making the object effectively two-dimensional. Although the inclusion of more models could accommodate deformable objects and several viewpoints, this would become very computationally expensive. This restriction is intended to specify that the system does not deform its models in an effort to achieve a match (i.e., the system models are assumed rigid) nor does it solve for arbitrary perspective projection.2 Finally, the features being recognized are assumed to be parts of larger objects as the system is designed to calculate dimensions as the distances between pairs of features found on a single larger object.

The system returns the apparent dimensions and disparity of individual objects and features.

2Although fish deform as they swim, they are videotaped in side profile, thus the deformations are in a direction parallel to the optical axis and appear as relatively small changes in length.
With correct calibration, this information can provide actual dimensions.\textsuperscript{3} The conversion from linear dimensions to biomass and the calculation of size distributions from biomass estimates are left to further work in biology and statistics.

1.3 THESIS OVERVIEW

This thesis is organized in six chapters. Chapter 2 reviews some of the related work in both computer vision and size assessment of fish. Chapter 3 introduces the prototype manual system and explains how images are obtained and prepared for both the manual and automatic systems. Chapter 4 explains the design and operation of the automatic system. Chapter 5 presents the results of performing size assessment on several stereo image pairs. Chapter 6 concludes the thesis by evaluating the success of the system and suggesting some possible directions for future work.

\textsuperscript{3}The manual system described in Chapter 3 was fully calibrated and the calibration process is explained in Section 3.2.
CHAPTER 2
PREVIOUS WORK

The simplest schoolboy is now familiar with facts for which Archimedes would have sacrificed his life.

- Ernest Renan

One unfortunate result of the degree of specialization of current research is that it leads researchers to focus only on their particular problem, assuming that the related problems either have been or will be solved for them. Thus many feature detection algorithms rely upon receiving clear, connected segments and, in turn, stereo algorithms rely on good feature detection and localization.

An interesting aspect of the NIFE system is that, being primarily an engineering rather than an academic project, no part of the system could be taken for granted and furthermore, no restrictions were placed on how the system achieve its results. For these reasons, almost every step in the vision process required decisions.

This chapter describes some of the previous work related to the current system in order to outline the evolution of ideas and explain the rationale behind some of these decisions.

2.1 PREVIOUS SYSTEMS RELATED TO AQUACULTURE

With hopes of keeping the system simple and inexpensive, preliminary investigation sought a dimensionless number, some ratio of measurements on the fish that varies appreciably with length or biomass. If found, such a value could provide the length or biomass of the fish from a single image without knowing the distance of the fish from the camera. After approximately a year of study indicated that no such ratio exists, this programme was abandoned and it was acknowledged that the system would require some means of determining the distance to features
in the field of view. Without this ability, it would be impossible to differentiate between a large fish located far from the camera and a small fish close to the camera. Stereo vision was the obvious choice because it is entirely non-invasive, requiring no handling of the fish or special lighting.

While stereo is a well-known technique for determining distances and has been studied by computer scientists, cartographers and others, some more closely-related work has been done to determine both the three-dimensional positions of individual fish and three-dimensional structure of fish schools.

In their research into the structure of fish schools, Cullen et al [CSB65] state that prior work in that area was based on two-dimensional photographs of fish schools taken from above and did not take into account the vertical distance between fish. In an effort to correct this oversight, their system used a single still camera fitted with a stereo-prism attachment to produce stereo images on a single negative. The camera was mounted above the tank and white styrene sheets were place on the bottom of the tank. Disparity was measured directly from enlarged images using a ruler. The same group devised a second method to determine the distance to the fish using the parallax between a fish and its shadow on the white bottom-sheet given the projection of a 10cm rod placed vertically on the bottom-sheet. In both systems, the position of a fish in the plane parallel to the image plane was determined from its position in the images relative to a grid marked on the bottom-sheet.

Van Sciver [VS72] describes a similar method for determining the size of objects underwater. His system uses a single underwater camera and requires that the objects remain stationary while the camera is physically moved sideways between exposures to obtain a stereo pair. Disparity was once again measured from enlargements of equal magnification.

Van Long et al [VLAI85] describes a more advanced system for determining the location and size of marine organisms in situ using underwater stereo cameras and a stroboscope to illuminate the cameras' field of view. In order to ensure simultaneous exposure of the stereo photographs, thereby permitting motion of the organisms, the cameras had no shutters and films were simply advanced after each flash of the strobe. Disparity was again measured manually from enlargements of equal magnification. In a test of the system, Long et al took many
Chapter 2: Previous Work

10

stereo photographs of both a test pattern and a scale bar in a 10m diameter aquarium. Using the photographs of the test pattern along with actual measurements of the three-dimensional locations of points, they calculated correction factors to convert from the calculated locations of points to actual locations in all three dimensions. They quote mean absolute errors of 0.77% in estimating distance to the scale bar and 0.75% in estimating its length. Van Long and Agayama [VLA85] describes what appears to be the same system with the addition of a computer and graphics tablet to aid in calculating distances, lengths and bearing angles.

Pitcher and Lawrence [PL84] describe a stereo system which implements video equipment very much like that used by the NIFE project. Pitcher explains that although video equipment offers low running costs, the ability to re-record and special effects such as slow-motion and time-lapse, its major drawback is its relatively low resolution compared to photographic film. This drawback was amplified in stereo video, which at the time was accomplished using a video mixer that displayed both images simultaneously in "split-screen" format, thereby halving the already inadequate resolution. To overcome this problem, Pitcher proposed a rapid switch circuit to route still images alternately from two video cameras to a single video-cassette recorder (VCR). The apparatus was calibrated with a grid in the field of view and was used to measure the three-dimensional positions of individual fish in schools.

Naiberg et al [NPSN93] describes a computerized stereo system for estimating the distance, size and biomass of individual fish. With this system, the operator uses a mouse to mark the head and tail of a single fish in both images of a stereo pair. The system then uses these points to calculate the distance to and length of that fish. This manually-operated system was used as a prototype to demonstrate the stereo concept and was generally well-received by local salmon farmers, motivating the automated system described herein. The prototype system is discussed in chapter 3.

2.2 Edge Detection

Because it is intended that the NIFE system recognize and locate fish automatically in images, reliable detection of edges in the images is clearly crucial to its success.

The central assumption of edge detection is that brightness discontinuities in an image
correspond to relevant scene features such as illumination or reflectance boundaries or, ideally, the edges of objects. It is the job of an edge detector to accurately locate these brightness discontinuities by performing numerical differentiation on the intensity values in an image. Unfortunately, no method has been found that accurately locates all edges with no erroneous responses under all circumstances, thus different edge detectors adhere to different criteria and offer different trade-offs between competing goals. Colin Savage, one of the original members of the NIFE project, has tested various edge detectors extensively for his Masters thesis. Savage concluded that the Canny edge detector [Can86] performs better than the alternatives on images of fish in sea cages, which tend to exhibit low contrast and high noise levels.

The two primary design criteria for the Canny edge detector are that it offer good detection, meaning it detects real edges well and rarely marks edges falsely, and good localization, meaning the points it marks as edges are very close to the actual edges. A third criterion was that a single edge give rise to a single response. Canny points out that the probability of failing to mark real edges and the probability of marking edges falsely both decrease with increasing signal-to-noise ratio, thus good detection corresponds to maximizing signal-to-noise ratio. Also, because the Canny edge detector is a gradient operator, meaning it responds with a peak to mark the location of an edge, good localization corresponds to narrowing its response peak.

In working through the mathematical formulation of the above criteria as applied to step edges, Canny found that there is a theoretical trade-off between the primary goals of detection and localization: those steps which increase signal-to-noise ratio to improve detection broaden the response peak and those which narrow the response peak to improve localization decrease signal-to-noise ratio. With this trade-off, it is possible to derive a single optimal operator.

Because the Canny edge detector responds with a peak to mark edges, its output must be thresholded to discriminate edges from noise. This step often introduces excessive streaking, the breaking up of edge contours caused by fluctuation of the detector output above and below a single threshold. To minimize streaking, the Canny detector implements hysteresis with two thresholds: once a pixel with intensity above a pre-set high threshold is found and marked, all connected pixels are marked provided their intensity is above a pre-set lower threshold.
2.3 Recognition

Simply stated, object recognition consists of determining what objects are present in an image and where they are located. Specifically, a model-based object recognition system,\(^1\) given a collection of object descriptions ("models") and an image containing one or more of these objects, attempts to match the object descriptions to objects in the image and determine precisely where the objects are located in the image. These tasks are often referred to as "labeling" and "pose estimation" respectively.

There have been many approaches to object recognition over the years and it remains a vibrant field in computer vision because there is no consensus on how to represent objects or search for them in the presence of noise, occlusion, translation, rotation and changes of scale. Clearly there exists at least one combination of object representation and matching strategy which yields excellent performance, for humans are able to recognize almost instantly a vast array of objects from almost any viewpoint even when faced with occlusion, clutter, poor lighting and any of a number of other hindrances. Furthermore, humans are able to correctly classify objects they have never seen before, such as the first time one sees a mountain bike or hand-held video-cassette recorder, and objects that appear in different forms and sizes, such as cars, chairs and dogs.

Computer vision often derives its shape representations from the boundary contours of objects in images. These contours can generally be detected under a wide range of imaging conditions without prior processing or knowledge of the image content. A number of strategies have been devised to describe contours and to compare image contours to stored descriptions of model contours.

One of the earliest attempts at shape representation is chain coding, described in [FF65]. Essentially, chain coding is performed by numbering one through eight the eight pixels surrounding the central pixel in a three-by-three pixel box. Any arbitrary line is then described by beginning at any pixel and recording the numerical code of the next pixel comprising the line, then the next and so on. This yields a chain of numbers representing the path followed by

\(^1\)As opposed to, for example, a neural network or statistical pattern classifier.
Chapter 2: Previous Work

the line.

Chain-code correlation is the associated method of shape recognition. Line segments in an image are coded as described above and correlated against stored "model" chains such that a perfect match link for link results in a score of one and any differences between the links in the image chain and those in the model chain reduce this score. A match between an image chain and a model chain is declared whenever this score is above a threshold. Davis [DR76] explains that chain-code correlation is not rotation-invariant and is very sensitive to both small changes in shape and global changes in scale, concluding that, "it cannot be considered a generally useful tool for shape matching" ([DR76], p.61).

Chain-code correlation is one example of a number of correlation-based algorithms which perform template matching. Most of these algorithms suffer from the same drawbacks and are therefore useful only when dealing with non-occluded, non-rotated objects of fixed scale ([Ste92], p.11).

Since this early attempt, much work has been devoted to developing more general and robust recognition systems. Davis [DR76] describes a large class of shape recognition systems which use sets of numerical values as shape descriptions and match these sets of values using techniques from statistical pattern recognition. The values used include various moments, Fourier shape descriptors and measures of compactness and elongatedness. However, Davis explains, these systems cannot be used to match a part of one shape to part of a larger shape because the description of any part of a shape is generally not related to the description of the entire shape in any simple way. This is a serious drawback from both practical and theoretical points of view. In practice, we would like a system to recognize partially occluded or incomplete objects for enhanced robustness, particularly because it is likely that the system will receive incomplete edges. From a theoretical point of view, we would like artificial intelligence to explain human capabilities, thus any system which cannot recognize incomplete objects is unsatisfying for it cannot adequately explain human visual processes. For these reasons, most recent research in object recognition is concerned with the general case of identifying objects from arbitrary viewpoints allowing for rotation, scale changes, partial occlusion, moderate distortion and noisy images.
Davis [DR76] presents a two-stage shape matching procedure in which shapes are represented as sequences of angles and their positions. This representation has the advantage that it is invariant to both rotation and a wide range of scale changes. Furthermore, it facilitates detection of partial matches as a series of matches between angles can begin at any point in the sequence.

A model of the polygonal (i.e., piecewise-linear) representation of a curve $M = M_1, M_2, \ldots, M_n$ where each $M_i = <x_i, y_i, \text{mag_i}>$ with $x$ and $y$ providing the location of the angle and $\text{mag}$ the magnitude of the angle. The polygonal representation of a curve in the image is similarly represented as $I = I_1, I_2, \ldots, I_m$ where $m < n$. A match of $I$ against $M$ is comprised of (1) a mapping associating the elements of $I$ with the appropriate elements of $M$ and (2) a coordinate transformation from model coordinates to image coordinates to provide spatial registration.

The first stage of the process searches for potential matches. The matching function regards the models as being spring-loaded templates, thus any model can be made to match any image curve if one is willing to expend the energy required to deform the springs as required. A perfect match occurs when the model curve lies perfectly over the image curve with no energy required to deform the springs. The match evaluation function sums the spring deformation energy required to obtain a match and some function of the difference between the relative distance between image segments and the relative distance between the corresponding model segments, adding a further penalty for missing angles, to assess the “cost” of any given association. The task of matching image curves to model curves is now the dynamic programming problem of finding the association functions $F_i$ which minimize cost. To avoid the computational requirements of dynamic programming, Davis uses a parallel, iterative procedure with local evaluation functions.

Matched segments are entered in an “association graph” in which each association of an image curve with a model curve corresponds to a node in the graph with a weight equal to the cost of that association. Two nodes are considered connected if the two associations they represent can reasonably occur simultaneously. This decision is made by determining whether the spring deformation energy required to connect the two nodes is below a threshold.

The second stage of the process attempts to eliminate incorrect matches from the association graph by means of relaxation techniques. Relaxation is an iterated parallel operation which
discards a node from the association graph if it is not sufficiently consistent with enough of its neighbour nodes.

A relaxation procedure similar to Davis's is explained concisely by Rosenfeld in [RHZ76]. Rosenfeld's algorithm begins with the set of all possible labels for all objects and, at each iteration, discards from each feature all those labelings which would result in a related feature having no consistent label. In this way it ensures that each node is labeled so as to be consistent with all of its neighbour nodes. These iterations proceed until no further labels can be discarded or, ideally, each feature has exactly one consistent label, yielding a consistent match. Davis's algorithm was able to discard all but the correct match in 47 of 50 tests matching portions of digitized coastlines.

Figure 2.1: This segmented model of a fish illustrates how individual segments are too local and provide little information. Those segments with corresponding numbers would easily be confused if recognition were based on the length and orientation of individual segments. Grouping adjacent segments provides more information and alleviates the potential confusion.

While the recognition system used in NIFE, like Davis's, operates in two stages and uses the locations and magnitudes of angles in its representation of shapes, it is more like two-dimensional Structural Indexing presented in [SM92, Ste92]. For recognition of two-dimensional objects, the outlines of the objects provide a simple representation. Structural Indexing is based on the observation that the system must operate with only parts of outlines if it is to successfully handle occlusion and noise, yet individual segments are too local to be used as matching primitives.

Two features are related if they mutually constrain each other's labeling.
because they provide very little information (see Figure 2.1). Thus the basic features in this method, called super-segments, are formed by grouping a fixed number of adjacent segments. Super-segments are described using the following terms (see Figure 2.2):

**Cardinality** The cardinality of a super-segment is the number of segments of which it is comprised.

**Arclength** The arclength of a super-segment is the sum of the lengths of the individual segments. Arclength is not scale-invariant.

**Angle** Angles are measured between successive segments. Thus a super-segment with \( n \) segments has \( n - 1 \) angles.

**Location** For super-segments of even cardinality, the location of the super-segment is defined to be that of the middle vertex; for super-segments of odd cardinality, the location is the
midpoint of the middle segment.

**Orientation** The orientation of a super-segment is the direction of the vector between the predecessor and successor of its middle vertex.

**Eccentricity** The eccentricity of a super-segment is computed from its second moment of inertia. It is essentially a measure of the aspect ratio of the super-segment and is used because, without length information, two super-segments can have very different shapes despite having identical angles.

Because the polygonal approximation of a contour depends on its size in the image and the resolution of the segmentation, Structural Indexing uses polygonal approximations of various resolutions simultaneously.

To generate models for Structural Indexing, the Canny edge detector is first applied to the scene and the resulting contours are segmented to compute polygonal approximations. Super-segments are formed by grouping a fixed number of connected linear segments. These super-segments are stored as entries in a hash table. Each super segment is assigned a code based on the angles between its composite segments and its eccentricity. This code serves as the key for the hash table, providing fast access.

In the hypothesis-generation stage of recognition, the image is processed as described above and the angles and eccentricities of the resulting super-segments are compared to the angles and eccentricities of the super-segments in the candidate models. Hypothesized matches between the image and the models are retrieved from the hash table using the encoded keys.

Once all possible matches between image super-segments and model super-segments have been computed, they are passed to a verification procedure. This procedure begins by dividing the hypotheses by model and storing them in a table with the models as keys and hypotheses as entries. These hypotheses are then grouped into consistent clusters.

A match is declared if three consistent hypotheses are found to support it. Consistency is determined using the constraints suggested by Grimson and Lozano-Perez in [GLP84]:

**Distance** The distance between two super-segments in the image must be close to the distance
between the corresponding super-segments in the model. Differences of scale are accommodated by factoring in the ratio of arclengths between the super-segment in the image and that in the model.

**Angle** The angle between the orientation vectors of two image super-segments must be close to the angle between orientation vectors of the corresponding model super-segments.

**Direction** The range of components of a vector spanning the two image super-segments in the direction of their normal vectors must be close to the corresponding range for the model super-segments.

The final step is to apply two iterations of least-squares fitting to compute the transformation from model coordinates to scene coordinates. The second iteration is generally necessary as the first one provides only a rough estimate of the transformation, not a perfect match.

Stein states that Structural Indexing performed well on both real and synthetic data exhibiting translation, rotation, scaling, occlusion and noise. It does, however, require dense image data and fails to detect objects if the edges exhibit severe discontinuity. Its efficiency is given to be $O(n) \leq O_{\text{recognition}} \leq O(n^2m^3)$ where $n$ is the number of features in the scene and $m$ is the number of candidate models.

### 2.4 Stereo

The use of stereopsis to determine distances is based on the principle that, given two images of a scene taken simultaneously from slightly different vantage points, a given point in the scene will be imaged to a different point in each of the two images unless steps are taken to compensate. The human stereopsis system compensates by adjusting the vergence angle of the eyes to make certain image points coincide, then uses this vergence angle to determine the distance from the viewer to those points. The farther the point is from the observer, the less crossed the eyes must be; if the point were infinitely distant, both eyes would be looking straight ahead. This system works well because the brain can correctly adjust the vergence angle of the eyes very rapidly and because at any given time, a human can focus on only a fairly restricted region in
the field of view, thus the vergence angle can be adjusted to obtain stereo information only for that region.

Instead of adjusting the vergence angle of the cameras, most computer-based stereo imaging systems use fixed cameras, resulting in the aforementioned relative shift in the location of a given scene point between the two stereo images. The distance to a point in the scene can then be determined from this shift because its magnitude is inversely proportional to the distance of the point from the cameras. Thus, points near the cameras may appear in opposite corners of the two images while points infinitely far from the cameras will appear at the same location in both images. A derivation of this relationship is provided in Appendix A.

In order to determine the distance from the cameras to some object visible in both images, one must locate the projection of at least one point on that object in one of the images, then locate the projection of the same point in the other image and determine the magnitude of the relative shift in position. Given this relative shift in position, called disparity, and a few parameters of the imaging hardware, the distance to the point can be calculated quite simply. In practice, the second step in this process, finding the point in the second image which corresponds to a given point in the first image, is the difficult step. This is known as the correspondence problem and there are two general approaches to solving it.

2.4.1 Correlation

The correlation-based approach is based on the assumption that corresponding points in the two images will lie in regions with similar brightness patterns. To search for the point \( P_R \) in the right image which corresponds to a given point \( P_L \) in the left image, a correlation algorithm compares the brightness of each pixel in an \( m \times n \) (usually square) "window" centred on \( P_L \) to each pixel in identical windows centred on each possible \( P_R \) in the right image. The correlation function \( f \otimes g \) is generally such that a close match between brightness patterns in the two regions yields a high correlation score while large differences between the two brightness patterns result in a low score. In other words, correlation performs essentially by laying a template window from one image over candidate windows in the second image and scoring the closeness of the matches. Whichever window in the right image produces the highest correlation score is the
best match and, provided that score is above some threshold, its centre pixel is considered $P_R$.

[Fua91] presents such a correlation algorithm with an extra verification step: the two images are
used symmetrically so that if window $a_R$ yields the closest match to $b_L$ of all possible windows
in the right image, $b_L$ must provide the closest match to $a_R$ of all possible windows in the left
image in order to declare a correct match. That is, the two windows must "choose" each other as the best possible match.

Correlation has the advantages that is a simple, general algorithm requiring no prior knowledge of the application and has the capability to produce very dense disparity maps because it attempts to determine disparity at each point in an image. Furthermore, the correlation need not be performed on raw intensity values; frequently derivatives of intensity provide more stable values on which to perform correlation.

One problem with correlation is that it may not operate well when confronted by highly specular surfaces because the shift in viewpoint between the two images may give rise to corresponding regions with very different brightness patterns. Also, in order to match windows reliably, correlation requires intensity patterns that vary sufficiently over the area of the correlation window being used. This could be a problem with murky or low-contrast images. Possibly the biggest theoretical problem with correlation is that it fails at object boundaries: where a correlation window overlaps a depth discontinuity, covering both foreground and background, there will be no matching window in the second stereo image. Most mature systems analyze boundaries to address this issue.

Perhaps the major practical drawback of correlation is that it is a very computationally intensive procedure because it operates at every pixel in the images. To minimize computation, stereo cameras are typically aligned with their optical axes parallel so that corresponding points are constrained to lie at the same vertical level in the two images. This simplification, called the epipolar constraint, reduces the search space from two dimensions to one.\(^3\) Furthermore, only a certain range of disparities is examined to further reduce the one-dimensional search. However, even with these reductions, correlation must perform a computation for each pair of windows being compared; with each computation being a compound step such as adding the

\(^3\)With the camera axes aligned parallel, epipolar lines are parallel and horizontal; if the cameras are not parallel, the epipolar lines diverge.
square of a difference between differences, correlation is not for the computationally faint of heart.

2.4.2 Feature-Based Stereo

The second approach to stereo matching seeks to identify a salient feature in one image and a matching feature located plausibly in the second image. For example, if a corner is identified in the left image at some point P, the algorithm searches the region in the right image to the left of P along the epipolar line for a matching corner. Marr and Poggio [MP79] proposed such a feature-based procedure as a model of human stereopsis.

Feature-based algorithms are not as computationally intensive as correlation-based algorithms because they operate only on salient features rather than on every pixel in the images. They often perform well provided the images do not exhibit many instances of similar features on the same epipolar lines.

Assuming the required salient features exist in the images, the drawback of feature-based stereo is that it requires reliable identification and location of these features, thus it is only as good as the feature detection system providing this information. Depending on which features are examined, different problems must be addressed. If general features such as zero-crossing are examined, the system may have difficulty because there will typically be many features to interpret; if higher-level semantic features are examined, the system will typically require prior knowledge of what features are likely to be in the images and from what viewpoints. Regardless of the approach, feature-based stereo provides relatively sparse disparity maps for it can determine disparity only at recognized features rather than at every pixel.

Finally, studies of stereopsis have revealed several properties which can be used to improve the reliability of stereo algorithms. Pollard, Mayhew and Frisby [PMF85] demonstrate that placing a limit on disparity gradients provides a means of resolving ambiguities without severely restricting the generality of the stereo system. Mohan, Medioni and Nevatia [MMN89] present a method to detect and correct disparity errors by exploiting figural continuity, the property that disparity changes smoothly along a contour.
CHAPTER 3
THE MANUAL STEREO SYSTEM

"He's suffering from Politicians' Logic."
"Something must be done, this is something, therefore we must do it."
- Jonathan Lynn & Antony Jay, "Yes, Prime Minister"

In order to familiarize ourselves with the use of stereopsis to determine length and to gauge reaction to the idea, a prototype system was developed and demonstrated to salmon farmers, equipment suppliers and government representatives. The prototype was also presented at an aquaculture conference [NPSN93]. This method uses the same equipment and procedure to gather stereo images as the automatic system described in Chapter 4. However, this system requires the user to manually locate and mark the heads and tails of fish in the stereo images, thereby performing the recognition and correspondence steps for the system. The computer then calculates and displays the disparity, distance, length and estimated biomass of the fish being assessed.

3.1 ACQUIRING STEREO IMAGES
The cameras used are Panasonic WBVD400 black and white video cameras. Designed for surveillance, they are well-suited to low-light applications, offering sensitivity down to 0.5 lux. They are fitted with Cosmicar 4.8mm wide-angle lenses and placed in custom-built underwater housings with lexan dome-port lenses. Application of Tsai’s camera calibration procedure [Tsa87] estimated the effective focal length of the lens/dome-port combination to be 5.4mm. These wide-angle lenses were chosen to maximize the field of view. The housings are bolted to

---

1This work was supported by a research grant from the British Columbia Science Council.
2Thanks to Rod Barman of the UBC Laboratory for Computational Intelligence for programming and performing the Tsai calibration.
an adjustable aluminum platform at a predetermined separation with alignment pins to keep the camera axes parallel. The platform and cameras are lowered into the tank or sea-cage on an aluminum column and the entire apparatus is painted black to minimize disturbance to the fish. Power to and communication with the cameras are through waterproof umbilical cables to the surface.

The video signals from the cameras are routed through a Panasonic WJFS10 digital frame switcher and recorded on a Panasonic AG-1960 S-VHS video-cassette recorder. The digital frame switcher routes still images to the VCR from the two cameras alternately at 1/15 second intervals. This permits the recording of images from both cameras on a single videotape and greatly simplifies the pairing of stereo images: each left image is followed by the corresponding right image. Unfortunately, the 1/15 second interval between the corresponding images of a stereo pair can cause significant error in disparity if the fish are swimming quickly. To overcome this problem, the signal from the second camera in the stereo sequence should be routed to the digital frame switcher through a 1/15 second frame-delay module.\textsuperscript{3} The equipment is shown in Figure 3.1.

To facilitate both the location of points on the fish and accurate length estimates, we would like the fish to be videotaped with their centrelines parallel to the image plane of the cameras. Fortunately, pacific salmon tend to circulate in an annular region in the sea-cage, hence placing the cameras on the outside of this annulus pointing radially inward captures many fish this way. Video footage was gathered with the cameras at several different locations in the sea-cages; more research is required into the behaviour of fish in sea-cages to determine how many and which locations are required to obtain accurate information about the fish.

\section*{3.2 Camera Correction and Calibration}

The ideal camera separation is determined by the estimated range of distances to the targets: we wish to maximize disparity by increasing camera separation while maintaining adequate overlap between the two cameras to obtain stereo information over a reasonable area. If the targets

\textsuperscript{3}The frame delay was not used in the prototype system because testing was carried out almost exclusively with stationary targets.
are expected to be distant, the cameras must generally be farther apart to compensate for the
decrease in disparity with increased distance. Fortunately, this is feasible because the overlap
region increases with distance. Once the estimated range of distances and a few constants of
the imaging hardware are known, the appropriate camera separation can be calculated from
the equations of Appendix A.

Before accurate measurements can be made, various lens distortions must be removed from
the images. These distortions include barrel distortion introduced by the wide-angle lenses and
ripples introduced by the hand-polished lexan dome-port lenses on the waterproof camera housings.

The distortions are corrected by capturing an image of a plane square grid placed parallel
to the image plane of the camera. This image typically shows the lines of the grid to be

---

Footnotes:

4Barrel distortion causes horizontal and vertical lines to bow increasingly outward toward the edges of the image.
straight in the centre of the image and increasingly curved and rippled toward the edges. By examining the central, essentially undistorted lines, one can determine the number of pixels which should separate the lines in an ideal, undistorted image. By comparing the locations of the grid points in the actual image to where they would be in an undistorted image, one can determine the correction values for each intersection point, the number of pixels each point must be moved both horizontally and vertically to correct the lens distortions (see Fig. 3.2).

A program was written which requires the user to mark the grid intersection points using a mouse-operated cross-hair, then calculates the horizontal and vertical correction values and writes the intersection points and their associated correction values to a correction file on disk.

Figure 3.2: A simulation of a distorted grid (dashed line) over an ideal grid. The magnified view shows the horizontal and vertical correction values, marked $dx$ and $dy$ respectively.

A second program reads in the correction file and uses the locations of the intersection points and their associated correction values to move each pixel in the image as necessary to remove distortions. This program uses linear extrapolation and cubic spline interpolation to generate rows of smoothly-varying correction values based on the known correction values at intersection points. Given a point in a distorted image, its correction values are determined by finding the closest known correction value(s) and either linearly extrapolating or linearly...
Chapter 3: The Manual Stereo System

or bilinearly interpolating as necessary, depending on the location of the point relative to the known intersection points.\(^5\) The accuracy of the correction can be tested by correcting the original image of the grid and examining how close the corrected image is to a square grid.

A similar procedure was written to align the epipolar lines in the two images of a stereo pair. After capturing a stereo pair of the calibration grid, the user marks pairs of corresponding points in the two images; the procedure leaves one image untouched and aligns the epipolars by calculating the vertical shifts required in the second image to make corresponding points lie at the same vertical level. These methods of correcting distortion were chosen because the hand-polishing of the dome-ports gives rise to random, non-uniform distortions which cannot be described by an analytical model of distortion.

During size assessment, distortion correction is performed in real time only on the points marked by the user. Because very few points are used during size assessment,\(^6\) performing correction on the entire image in advance requires inordinate amounts of processing time and storage space. A second reason is that during correction, many points in a distorted image are relocated beyond the borders of the image, hence these points would be lost. By performing correction internally after a point is marked, the entire original image is usable.

Once the lens distortions have been corrected, the final calibration step is to calculate the constants required to convert measurements of disparity, length in pixels and height in pixels to measurements of distance, actual length and actual height respectively. This is performed by obtaining a stereo pair of images of an object of known size placed a known distance in front of the cameras. The calibration grid used earlier serves well because it provides many “objects” of known size in many orientations throughout the image. Knowing both the actual distance to the object and its resulting disparity, the constant converting disparity to distance is easily calculated for the camera separation used. Similarly, knowing the apparent length and height of the object (in pixels) and its distance, it is straightforward to calculate the constants converting corrected length and height (in pixels) to actual length and height for the imaging hardware used. Separate constants are required for the horizontal and vertical

\(^5\) Although more sophisticated extrapolation methods are available, linear extrapolation was found to produce the best results.

\(^6\) In the demonstration system, the user clicks on only four pixels per fish in a 512 × 480 pixel image.
directions because the aspect ratio of pixels is typically not 1:1. These values follow from the equations of appendix A.

3.3 Size Assessment

Fish size is determined from pairs of uncorrected stereo images of the fish in side profile. The coordinates are corrected in real time as outlined above. The programs were written in ANSI C on an Intel 486 machine equipped with an ITEX Overlay Frame Grabber card for acquiring and processing images. The ITEX card provides only frame RAM; all image processing routines are implemented serially in software. This equipment was chosen for its availability and relatively low cost in the hope of making the system accessible to local salmon farmers.

Using a mouse-operated cross-hair, the user must mark the nose, tail-fork, dorsal and ventral sides (at the widest point) of a flat, non-occluded fish in both the left and right images. The program automatically displays the second image of the stereo pair once the user has marked all four points in the first image.

The disparity of the nose of the fish is calculated by subtracting the horizontal coordinate of the nose in the right image from that in the left image. Because distance varies inversely with disparity, the distance to the nose is calculated by dividing the previously-determined distance constant by the disparity. The same procedure provides the distance to the tail-fork.

The length of the fish is calculated using the Pythagorean theorem in three dimensions to obtain accurate estimates regardless of the orientation of the fish in the image. The horizontal and vertical components of length are calculated by subtracting the x and y coordinates of the head and tail and converting these to metres using the distance and the constants determined earlier. These components are determined from both images and the larger values chosen because contortions may make the fish appear shorter in one image but never longer. The third component of length, parallel to the optical axis, is calculated by subtracting the distance to the tail from the distance to the head. Clearly if the fish is oriented parallel to the image

---

7 Locations on the dorsal and ventral sides are required because the most accurate formula found relating exterior dimensions to biomass involved the “height” of the fish as well as length.
8 Future versions of the system could analyze more points to check for consistency.
9 Recall the two images are actually separated by 1/15 second, thus regular swimming motions may cause the fish to appear shorter in one image than the other.
plane, this value is zero. The height of the fish (from dorsal side to ventral side) is calculated analogously except the estimated distance to the centre of the fish is based on the distances to the nose and tail because there are no features which provide reliable disparity values in the centre of the fish.

Once the length and height of the fish have been calculated, the biomass is determined from a formula which takes into account condition factors for the particular fish in question. The program can easily be tailored to suit different conditions or groups of fish by providing different biomass formulae as necessary. For each fish, the program displays the length and disparity in pixels, the distance, length and height in centimetres and the biomass in grams.

### 3.4 Results

Feedback from fish farmers and industry representatives was generally very positive. In fact, representatives from an aquaculture equipment distributor offered a development grant in exchange for distribution rights. Most users felt comfortable with the system and could assess fish quite quickly with very little practice. All agreed that the technique is much easier than existing methods. As will be discussed shortly, some users reported difficulty in determining precisely which points to mark under certain circumstances.

The accuracy of the system was tested by estimating the distance and dimensions of both artificial targets and dead or anesthetized fish submerged in test tanks.

The necessary constants for each test were determined using the method described above and averaging several readings of disparity and apparent dimensions taken randomly throughout the images. The distortion correction files for individual cameras\(^\text{10}\) were known in advance. The stereo pair used for calibration was analyzed to ensure the accuracy of the calibration but was not used during testing.

The artificial targets included the calibration grid and several white plastic shapes. The grid is an excellent target because the distance between any two intersection points is easily calculated, thus the grid offers many targets in various orientations throughout the images. The

\(^\text{10}\) For the correction files, the term "camera" includes the lens and housing as the distortion is affected by the dome port and the precise position of the camera within the housing.
targets were placed between 30 and 130 centimetres from the focal plane of the cameras and were not necessarily parallel to the image plane.

In repeated tests on these well-defined targets, the estimates for distance, length and height were always within 4% of the correct values when the targets were parallel to the image plane. When the targets were not parallel to the image plane, tests showed average errors of 5.6% for length and 4.3% for height. The orientation of the object (horizontal or diagonal) in the plane parallel to the image plane had no effect on the error.

In limited testing conducted by suspending dead fish in front of the cameras, average errors were 2.7% for length estimates and 4.3% for height estimates. The maximum errors were 4.7% for length and 8.3% for height.

A more thorough evaluation of the system was performed by Andrew Ohara as an undergraduate thesis project [Oh93]. Testing was carried out on three dead and 42 anesthetized chinook salmon\(^\text{11}\) by measuring them with calipers and then holding them underwater in front of the cameras in various orientations. Once the anesthetized fish regained consciousness, they were videotaped swimming freely in the tank together; still frames were captured later.

Ohara’s analysis of these images indicates that the system provides accurate length estimates at shorter distances (50 cm) but that the accuracy of the length estimates decreases slightly with increased distance (up to 110 cm). The system was very accurate with the fish parallel to the image plane but tended to underestimate lengths increasingly as the fish were rotated further out of the image plane. The width estimates were not as accurate and showed no obvious relationships to distance or orientation.

The fact that the system provided very accurate estimates with well-defined targets but could not equal this accuracy with fish indicates that it may be difficult to mark points correctly on ill-defined targets. Some users reported this problem. This is further supported by the increase in error with distance because as distance increases, the outline of the targets becomes more difficult to locate precisely. Furthermore, errors introduced by increased distance are compounded because as distance increases, each pixel represents a larger increment of length or height, thus any error is also magnified.

\[^{11}\]The fish were anesthetized with 2.5 grams of tricaine-methane-sulfonate.
Incorrect marking of points could lead to a compound error because the same points are used to determine both apparent length and disparity. If the user erroneously marks a point in one image which makes the fish appear larger to the system, this exaggerated length will be chosen as explained in section 3.3 above. If this erroneous point also causes an apparent decrease in disparity, the distance estimate will be too high and the exaggerated apparent length will be further magnified as a result of the exaggerated distance. If the manual system were to be used commercially, it would benefit from some form of image processing such as edge detection to facilitate accurate location of points.

Another complication in determining the length of fish is that the swimming motion involves swinging the tail from side to side, causing the fish to appear shorter when viewed from the side. This motion is often modeled by a sine wave traveling along the body of the fish. The precise shape of this wave varies with each species but for fin fish the amplitude of the wave is typically negligible over the front 2/3 of the body. Accepting this model, swinging the aft third of the body 35° from the centreline results in an apparent length approximately 6% shorter than the correct length. There is no obvious solution to this problem when assessing individual fish because it is often impossible to determine from an image if the fish is contorted or flat. When assessing a large group of fish, the solution may be to adjust the system constants such that the length of flat fish is slightly overestimated and that of contorted fish is slightly underestimated. This may result in accurate statistical predictions for the group as a whole.

The one simple conclusion from this testing was that the cameras must be securely locked into position in their housings. Before this was done, removing the cameras from their housings and replacing them was found to necessitate changes in the constants of up to 4% to account for small shifts in the position of the camera within the housing. Furthermore, although the two cameras should both function equally well in either the left or right position, interchanging them resulted in errors of up to 20%, indicating less than perfect alignment within the housings.

12 Once again, it is for this reason the larger estimate from the two images is chosen by the system.
4.1 Design Decisions

We wish to identify fish in images and determine automatically their length from head to tail-fork. As mentioned in Section 2.1, the first method proposed to estimate size sought to discover some non-uniformity in the growth of a fish which gives rise to a ratio of measurements that changes appreciably as a fish grows. The value of this ratio for a particular fish would then accurately reflect the stage of growth and size of the fish. The chief advantage of this ratio is that, being a ratio, any effects of scale would divide out, thus the size of the fish could be estimated from a single image without knowledge of distance or scale. Unfortunately, fish were found to grow uniformly, hence all such ratios remain constant (to within biological diversity) throughout the life of the fish and provide no useful information.

Given that no ratio can be used to determine the size of a fish, any measurement taken from an image will clearly reflect the distance of the fish from the camera as much as it does the actual size of the fish because apparent length varies directly with actual length and inversely with distance (see Appendix B). Thus the system must be able to determine the distance to the fish in order to eliminate this unknown.

Sonar ranging is one method capable of furnishing distance information and has been used in
aquaculture research to provide information about fish population and density. Sonar equipment works by emitting a sound wave which is reflected back to a detector by the entire school of fish. Based on the strength of the return signal and the time required for it to reach the detector, the density of and distance to the fish population can be calculated. However, because the emitted wave front spreads as the wave travels, resulting in a cone of energy, sonar equipment cannot be aimed so as to give the distance to a particular fish. In fact, no single-camera method exists which can provide accurate distance information without specialized lighting, background or the inclusion of markers in the image. Binocular stereopsis was chosen because it is entirely non-invasive, requiring no handling of the fish or specialized background or lighting.

The model-based recognition/feature-based stereo approach is not the only way to solve the problem at hand. Many recognition/stereo systems begin by performing correlation on a pair of stereo images to obtain the disparity at each point, then locate objects by searching for regions of approximately constant or smoothly-varying disparity. This approach was rejected because correlation had difficulty dealing with the poor contrast and specularities found in real images and, moreover, the approach misdirects effort, generating more information than can be used and not making optimal use of that which it does generate (see “Problems with Correlation”). As will be discussed presently, (Section 4.1.1), the model-based recognition/feature-based stereo approach taken, on the other hand, handles difficult images well, provides only the information required and makes very good use of this information.

### 4.1.1 Model-Based Recognition

A recognition system adequate to the task at hand must be able to locate the heads and tails of fish in images. This task is complicated by the poor contrast often exhibited in underwater images, which makes it difficult to extract complete edges, and by subtle variations in shape between fish, which make it more difficult to obtain a precise match. Matching leniently and using multiple models and segmentations should help overcome these problems.

Fortunately, the tail of a fish is a distinct shape characterized by sharp angles. This makes segmentation and matching more reliable because a sharp angle signifies clearly where one linear segment should end and the next begin, thereby facilitating accurate determination of the
lengths of segments and the angles between them. However, the heads of certain fish are much more rounded (i.e., no sharp angles), hence the dividing points between segments become rather arbitrary and the lengths of and angles between segments become correspondingly unreliable.

It was decided that a two-dimensional recognition system would be sufficient because, as was mentioned in Section 3.1, the cameras can generally be positioned so as to provide images of the fish in side profile, thereby simplifying the problem. Furthermore, tests on the manual system showed measurements to be more accurate when the fish are oriented roughly parallel to the image plane. A two-dimensional system should also be much simpler and faster than a general three-dimensional system, which must calculate projections to solve for arbitrary orientation.

Given that we wish to identify two-dimensional objects in two-dimensional images, it is natural to use the boundary contours of objects as features. With this in mind, the system implemented uses a shape representation similar to that used in Stein’s two-dimensional Structural Indexing described in Section 2.3 and [SM92, Ste92]. Recall that this method seeks to match shapes based on descriptions of the piecewise-linear segmentations of their outlines.

The basic feature used herein is Stein’s super-segment with a change in the way lengths are used, the addition of orientation angles and removal of the eccentricity (see Figure 4.1). The orientation angle, which describes the absolute orientation of a segment, was included to exploit any regularity of orientation exhibited by the objects being recognized, in this case the tendency of Pacific salmon to swim horizontally. Angles between successive segments are also used and are referred to as “relative angles.”

The length information introduced problems because the system must be scale invariant. Nonetheless, inclusion of length information proved beneficial once the length of each segment was divided by the length of its predecessor to normalize the measurements. The eccentricity is redundant once length is included because the lengths and angles completely specify the shape of a super-segment. The orientation as Stein defines it is unnecessary in this application.

Because a given shape can have several polygonal approximations depending on the resolution of the segmentation or the size of the shape at a fixed resolution, many recognition systems permit the use of multiple models and/or multiple image segmentations to make recognition

---

1It should be pointed out that orientation angles cannot be used for all objects or even all fish. Atlantic salmon, for example, swim randomly in all directions and thus have no preferred orientation.
Figure 4.1: A super-segment modified to include orientation angles $\theta_i$. Relative angles $\alpha_i$ are similar to those used by Stein. Segment $S_0$ is not part of the super-segment but is included to show the relative angle $\alpha_1$. The relative length of segment $S_n$ is calculated by dividing the length of $S_n$ by the length of $S_{n-1}$. The location of this super-segment is marked $L$.

More robust. These capabilities are particularly important to the NIFE system because, in addition to segmentation and scale differences, recognition must also contend with shape differences arising naturally as a result of biological diversity. Using multiple image segmentations means that the image being processed is approximated at several resolutions\(^2\) and each approximation is compared to the models. This increases robustness by increasing the likelihood of a segmented feature in the image matching one of the models (see Figure 4.2).

Multiple models may be either multiple segmentations of a single instance of a feature at different resolutions (to accommodate differences of scale and detail) or different instances of a given feature (to better accommodate biological diversity). To accommodate multiple models of a single feature, each model in the NIFE system can be given both a name and a feature identification number. The name of each model is intended to be unique in order to discriminate between specific models while the identification number is intended to be the same.

\(^2\)The resolution is determined by the deviation permitted between a contour and its linear approximation.
Figure 4.2: (a) The linked outline of a fish. (b) and (c) Segmentations (polygonal approximations) at lower and higher resolutions respectively.

for all models of a given feature so as to reflect the equivalence of multiple models of a single feature. Thus, all models of a fish tail would be given the same identification number and it is this identification number that is referenced during verification or when determining disparities and lengths. This ensures that the verification and stereo procedures handle equivalent features uniformly regardless of which particular model was matched.

Finally, because the system is designed to determine the dimensions of objects as delimited by specific features, the models are generally of the specific features involved in measurements rather than of entire objects. For example, the length of a fish is measured from nose to tail-fork, thus separate models are provided for the nose and the tail-fork rather than a single model of an entire fish. This way, the system is able to isolate the constituent features of an object individually and a desired dimension is calculated as the distance between two delimiting features. The user specifies these features by their identification numbers.\(^3\)

The general recognition strategy is to match features leniently and then perform strict verification. Although strict matching would decrease susceptibility to false matches by admitting only very close matches, it may do so at the expense of failing to match desired features. However, verification provides a second chance to eliminate falsely matched features while failure to

\(^3\)The recognition system can make use of models of entire objects but it cannot use these matches to determine the dimensions of a single object; to do this it must isolate at least two features of the object individually.
match a feature results in its being lost for good. Thus, because the NIFE system has both a verification step and a stereo step offering ample opportunity and information to discriminate correct matches from false matches, all reasonable matches are admitted and false matches are eliminated later in the process.

Matching is performed by comparing the angle and length information between the image and the models segment-by-segment, searching for super-segments over a fixed range of cardinalities with average error below a predetermined threshold. All plausible matches (those with average error below the threshold) are passed to the verifier.

The verification step checks whether candidate feature matches are in plausible locations relative to one another by comparing the distance and angle between pairs of different features in the image to the distance and angle between the corresponding features in the model. To accommodate differences of scale, the distance between features is scaled by the ratio of arclengths between the image super-segment and the model super-segment. The verifier assumes that each object has only one instance of any given feature. This is a valid assumption in the case of fish heads and tails but does not work for bicycle wheels, airplane wings or chair legs.

4.1.2 Stereo

Problems with Correlation-Based Stereo

Although correlation is a straightforward, off-the-shelf technique for stereo correspondence matching, it has a number of drawbacks which led to its rejection.

Because the size assessment system requires distance estimates only at the nose and tail of each fish to calculate its most reliable length estimate (i.e., to employ the Pythagorean theorem in three dimensions), the dense disparity maps produced by performing correlation at every pixel in an image provide much more information and require more computation than is necessary. Tests of a PC-based implementation of Fua's algorithm [Fua91] indicate that correlation would require roughly 24 minutes per image pair. Although the program could be further optimized, it does not appear that this time requirement can be substantially decreased without an expensive upgrade such as a card which implements parallel convolution in hardware. Because the goal of NIFE is a system which analyzes many images and runs on inexpensive
PC's, correlation is likely simply too computationally intensive a process. Furthermore, the disparity maps produced by correlation are only a first step; they must be interpreted in order to locate the required features.

Even if the time requirement were acceptable, limited testing indicates that the specular surface of the fish combined with the difficult lighting conditions encountered underwater, either very low light and poor contrast or pronounced backscattering, make it very difficult for correlation to function effectively. The correlation program was unable to provide accurate disparity measurements even where edge detectors had no trouble recovering edge detail.

Finally, correlation requires near-perfect epipolar alignment of the two images in order to perform well because it attempts to match regions pixel-by-pixel. If the epipolar lines in the two images are not aligned properly, a candidate window scanned along a horizontal line in one image is never properly aligned with the corresponding window in the other image, resulting in inaccurate correlation scores (see Section 2.4). Unfortunately, because of non-uniform ripples introduced by the dome-ports and the fact that the cameras are moved around and removed from their housings often, epipolar alignment has been less than perfect and cannot be sufficiently rectified by the software or a known geometric correction.

**Feature-Based Stereo**

Feature-based stereo offers an excellent solution to these problems. First, feature-based stereo uses the features returned by the recognition system as the basis for its disparity calculations. Because the recognition system retains only those image features necessary to provide size estimates, feature-based stereo calculates disparity estimates only for those points at which disparity is required. This results in much lower computational requirements than correlation-based stereo because the number of features in an image is typically much smaller than the number of pixels. Furthermore, the feature-based verification step associates heads and tails thought to belong to the same fish, thereby identifying ambiguous matches and providing the apparent length of the fish in addition to the disparity. Correlation cannot provide such associations because it operates on syntactic primitives (pixels) rather than meaningful (semantic) features.
Chapter 4: The Automatic System: Design and Implementation Details

Secondly, feature-based stereo is less susceptible than correlation to the specific problems associated with images of fish mentioned above. Because it does not endeavour to match brightness patterns directly, feature-based matching is not as sensitive to specular reflections or poor contrast. Provided the edge detector locates the outline of the same identifiable feature in both images, the stereo algorithm should be able to identify the features as a correspondence match. As mentioned above, the Canny edge detector was able to recover the necessary edge information in places where the correlation algorithm failed.

Finally, because feature-based stereo does not endeavour to match regions pixel-by-pixel, it does not require perfect epipolar alignment. Instead, features are sought independently in the two images and two features are declared epipolar if their vertical separation is below a predetermined threshold.

Determination of the distance between two given features in a pair of stereo images requires that (1) both features be identified in at least one image to provide the apparent distance between them and (2) at least one and ideally both features be identified in the second image in order to provide the distance to one or both features. With this in mind, the stereo matcher was designed to accept features from both images and attempt to interpret ambiguous matches based on the disparity estimates in addition to making simple correspondence matches.

4.2 Implementation Details

The system was implemented entirely in ANSI C using the UBC Vista library. Vista is a collection of programs, subroutines and data structures developed by UBC's Laboratory for Computational Intelligence for use in image processing and computer vision. It was adopted because it provides representations for images and edge sets as well as a number of related tools such as the edge detector, linker and command-line parser.

A schematic diagram of the entire process is provided in Figure 4.3.

4.2.1 Pre-Processing

Because objects and models are represented by their outlines, edges are first extracted from images using an edge detector. As mentioned earlier, the Canny edge detector (described in
Chapter 4: The Automatic System: Design and Implementation Details

Create Models

Images of isolated objects
Gray-scale image

Canny Edge Detection
Gradient map
Link Edges
Contour map

Segment Edges
at multiple resolutions

Piecewise-linear segmentation
Clean Image
Join & order segments
Enhanced segmentation
Create & edit models

Model file (ASCII)

Match image segments to model segments to identify features
Lists of candidate features

Verify using relative positions of features; group features that are thought to belong to the same object
Lists of verified features

Locate all epipolar features
Candidate corresponding features

Disambiguate any multiple matches if possible
Corresponding feature pairs

Determine disparity and length
Distance and Length Information

Analyze Stereo Images

Stereo pair containing objects
Gray-scale Image pair

Canny Edge Detection
Gradient maps
Link Edges
Contour maps

Segment Edges
at multiple resolutions

Piecewise-linear segmentation

Match image segments to model segments to identify features
Lists of candidate features

Verify using relative positions of features; group features that are thought to belong to the same object
Lists of verified features

Locate all epipolar features
Candidate corresponding features

Disambiguate any multiple matches if possible
Corresponding feature pairs

Determine disparity and length
Distance and Length Information

Figure 4.3: A schematic diagram of the entire procedure showing inputs and outputs.
Section 2.2) provided the best results. The Canny edge detector\textsuperscript{4} first smooths the image intensities with a Gaussian filter and then calculates the gradient of the intensity at each point in the resulting image. Local maxima in the gradient are marked with a value proportional to the magnitude of the gradient at that point; pixels that are not local maxima are set to zero. The edge detector returns the resulting gradient map.

The linker accepts these gradient maps and joins connected non-zero pixels to form edges. Because the gradient magnitudes supplied by the edge detector represent the "strength" of each edge, these values should be thresholded to discriminate true edges from noise. Like the Canny edge detector, the Vista linker\textsuperscript{5} performs Canny hysteresis using two thresholds to minimize streaking (see Section 2.2). The linker returns an edge set in which each contour is numbered and the coordinates of each pixel in each contour are specified. These contours may be of arbitrary length and shape as each contour follows the intensity gradient contour traced by the edge detector and a new contour begins only where there is a discontinuity in the current gradient contour.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.4.png}
\caption{(a) The polygonal approximation of contour $AB$ is computed by joining its endpoints and finding the point of maximum perpendicular deviation $C$. (b) If this deviation $d$ is above a pre-set threshold, the contour $AB$ is replaced with two straight segments $AC$ and $CB$. (c) These two new segments are themselves subdivided if necessary.}
\end{figure}

These contours are next broken into piecewise-linear segments to generate the polygonal approximations used in matching. This segmentation is accomplished by joining the endpoints of each contour with a straight line and locating the point on the contour of maximum perpendicular deviation from this straight line. If this deviation is below a predetermined threshold, the original contour is replaced by the straight line joining its endpoints and the next contour

\textsuperscript{4}The Vista Canny edge detector was written by Richard Pollock, Daniel Ko and Art Pope.

\textsuperscript{5}The Vista linker was written by David Lowe.
is examined. If the deviation exceeds the threshold, the contour is divided at this point of maximum deviation and the two resulting partial contours are passed back to the procedure recursively for further subdivision. This process ensures that each contour is approximated by a set of straight segments such that no segment deviates from the original contour by more than the allowed distance (see Figure 4.4). When the recursive subdivision is complete, the entire set of straight segments is written to a new edge set. This new edge set is a more compact representation than the input edge set because all the segments are linear, thus only the coordinates of the endpoints are retained. By varying the deviation allowed between a contour and its linear approximation, the original contours can be approximated at different resolutions.

Both the linker and the segmentation program can help remove noise from the images by retaining only those contours which exceed a pre-set minimum length (in pixels). Elimination of short contours is a valuable time-saving step because most of the contours resulting from noise are very short. If the edge detector were guaranteed to extract complete outlines of objects, this threshold could be set quite high to remove virtually all contours except these outlines. Unfortunately, because discontinuities often cause a long contour to be broken into a number of shorter contours, setting this threshold too high will likely remove valuable information from the image.

The data structure used to store Vista edge sets (such as those created during segmentation) permits access to information about individual segments only in numerical order. In order to compute and match super-segments efficiently, we must have random access to information about the segments, including knowledge of which segments are connected. Random access is required because connected segments are not necessarily numbered sequentially throughout an image: sequentially-numbered segments may lie at opposite corners of an image while any segment may be connected to several segments numbered seemingly randomly.

To accommodate this, each segment in the edge set is entered into an array with its identification number serving as the array index. Each entry in the array provides the length, endpoints and orientation angle of the segment and a pointer to a list of all the connected segments (see Figure 4.5). This array provides fast access to all the information about any segment without having to calculate values repeatedly. The relative angle of a segment is calculated from the
Figure 4.5: The segment information calculated for segment 12. Segments 11, 17 and 33 are considered connected to segment 12 because each has an endpoint within the prescribed radius of an endpoint of segment 12 (dashed circle); segment 37 is not connected. The orientation angle is in radians from the horizontal. The connect field describes what type of connection exists between the specified segments.

Two segments are considered connected if either end of one segment lies within a predetermined radius of either end of the other segment. Using the start-points and endpoints assigned to each segment by the segmenter, the connections are labeled by the type of connection that exists between them. If a single contour is divided into segments, the endpoint of one segment will meet the start-point of the next. However, discontinuities in the contours and contours resulting from noise and nearby objects often lead to segments joined start-to-start, end-to-end or start-to-end.

4.2.2 Model Creation and Use

Models are created from single images of isolated objects. Because only the shape of the outline is required, a clean image of the object to be modeled can be created by cropping the original image, enhancing contrast or including only those contours longer than some minimum length (in pixels). In fact, a model can be created from an image of a black paper cutout of the object against a white background.

The pre-processing follows the same course described above with one additional step performed after segmentation. This step, called “cleaning” the image, reorders the segments so
that connected segments are numbered consecutively and joins those segments which are likely part of the same closed shape.

Irregular ordering often results from discontinuities in the edges traced by the edge detector. Such discontinuities can cause the linker to declare two separate contours where there should be only one. During segmentation, these two separate contours may not be be processed consecutively, in which case the segments approximating the second contour will not numerically follow those approximating the first contour. Cleaning the image after segmentation reorders the segments such that each one is followed numerically by the closest remaining segment which has not already been numbered. This method is based on the notion that contours which were separated by erroneous discontinuities will still lie close to one another and will thus be re-numbered consecutively.

After reordering, each segment is extended in both directions to determine if its extension crosses within a predetermined radius of either endpoint of the next segment, in which case it is extended to meet the next segment. The overall effect of the cleaning process is thus to repair breaks caused by imperfect edge detection and order segments such that their numbers increase sequentially around the perimeter of a closed figure. This step is helpful when creating models because it makes the segment lengths more accurate and ensures that information about connected segments is stored sequentially in the model file. Cleaning is performed only on the images used for model creation because they are assumed to have relatively few segments outlining only the one object being modeled while real images upon which recognition is performed are often littered with too many spurious and disconnected segments to be cleaned effectively.

A model file is simply a text file which provides the length and angle information for each segment in the model. It is created semi-automatically. A model is itself a super-segment describing a single feature, either a head or a tail, at a given resolution. Heads and tails are connected only during verification and each model includes information specifying by which models it can be verified. Each model file has a header line which specifies the unique name of the model, the number of segments in the model, a flag indicating whether to compare

6Recall that these images can be cropped and contrast-enhanced.
Table 4.1: A sample model file. The '1' on the first line indicates that orientation angles are to be examined. The last line is a note to the user. Note that although the first line states that the model consists of four segments, there are actually five segments in the file. The fifth segment, despite not being part of the model proper, describes the next segment in the polygonal approximation and provides relative angle and length information when examined in conjunction with the previous segment.

BecausethemodelsarestoredasASCIIfiles,theycanbeeditedeasily. The user typically creates a model of a specific feature by generating a model of the entire object and deleting from this model all the segment information lines except those which describe the desired feature. By doing this repeatedly, separate models for all the features of a given object can be created from a single model of the entire object.

For reasons that will be explained when verification is discussed (Section 4.2.4), all features of a single object must be extracted from images which show the object at the same scale and in the same location in the image. In the development and testing, all models were taken from different segmentations of the same image of a fish.

Most applications of the system require several models. To simplify use, the user may create a file providing the number of models being used, their names, the feature identification number

7Orientation angles need not be compared if the object being recognized exhibits no regularity of orientation. Furthermore, orientation-dependent models can be used together with orientation-independent models if some of the objects to be recognized have expected orientations and others do not.
associated with each one and a list of those models which can verify each model. Specifying the system models, feature identification numbers and verifying models in this separate file allows a single collection of models to be used for several different recognition tasks.

4.2.3 MATCHING

After segmenting the current image at the current resolution, the program examines each image segment and attempts to match a super-segment beginning with the current image segment against every possible super-segment in each candidate model. A super-segment may begin on any segment of either the image or any model provided a super-segment of the specified minimum cardinality is attainable: it is pointless to examine a match beginning on the third segment of a four-segment model or image feature if the minimum cardinality is three.

Matching is performed by comparing the angle and length information between the image and the models segment-by-segment, searching for super-segments over a fixed range of cardinalities with average error below a predetermined threshold. The three values compared are as follows:

Orientation Angle The absolute orientation of the segments, measured relative to the horizontal, is examined to exploit any regularity of orientation exhibited by the objects, in this case the tendency of Pacific Salmon to swim horizontally. This value can be ignored to identify randomly-oriented objects.

Relative Angle The relative angle is the angle between a segment and the previous segment in the super-segment. This value is invariant to changes in scale and orientation.

Relative Length The length of a segment is divided by the length of the previous segment in order to accommodate differences of scale.

Matching is accomplished by summing the squares of the discrepancies in these three values between the current image segment and model segment. Prior to squaring, each discrepancy is divided by a pre-set constant which specifies a "pivot point," the value which separates

---

8 This information can also be entered from the keyboard during execution but this is very slow.
an acceptable discrepancy which decreases when squared from a significant discrepancy which increases when squared. Division by a constant also permits specification of an equivalence between errors in relative length and errors in angle. Setting the angle constant to 0.15 and the length constant to 0.1, for example, specifies that an angular discrepancy of 0.15 radian is equivalent to a 10% error in relative length. Thus the total error for a super-segment is given by:

$$\text{error} = \frac{1}{n} \sum_{j=1}^{n} \left( \frac{\alpha_m - \alpha_i}{c_{\text{angle}}} \right)^2 + \left( \frac{\theta_m - \theta_i}{c_{\text{angle}}} \right)^2 + \left( \frac{l_{m_j} - l_{i_j}}{l_{m_j-1}} \right)^2$$

where $\alpha$ is the relative angle, $\theta$ the orientation angle, $l$ the length of the segment, subscripts $m_j$ and $i_j$ refer to the $j^{th}$ model segment and $j^{th}$ image segment of the current super-segment respectively and $c_{\text{angle}}$ and $c_{\text{length}}$ are the constants explained above. The cumulative error is divided by $n$, the number of segments matched, so that each segment may contribute a small error without the average error exceeding the allowed maximum. Without this division, every long super-segment (high cardinality) would likely exceed the error limit even if each of its constituent image segments matched the corresponding model segment quite closely. The total error could also be divided by some function of $n$ so as to encourage longer or shorter matches as desired. Any super-segment having average error below the pre-set limit is considered a match; lowering this limit makes the matching more strict.

Orientation angles are restricted to the interval $-\pi < \theta \leq \pi$. Restricting the orientation angles to this interval creates an unwanted discontinuity at $\pm\pi$. Switching to degrees for simplicity, a segment oriented at $183^\circ$ to the horizontal is recorded as lying at $-177^\circ$, thus the discrepancy in orientation angles between a segment oriented at $178^\circ$ and one oriented at $-177^\circ$ is $178 - (-177) = 355^\circ$ where it should be $5^\circ$. To compensate for this discontinuity, any orientation angle within a predetermined $\epsilon$ of $\pm\pi$ is examined both as is and after adding or subtracting $2\pi$. In the previous example, the system would realize that adding $2\pi$ to $-177^\circ$ results in the correct discrepancy of $\|178 - ((-177) + 360)\| = 5^\circ$.

The angle information is useful and easy to work with because angles are invariant to scale changes. The length information poses a problem in this respect because segment lengths are largely determined by the scale of an object in the image. This makes it difficult to compare
Chapter 4: The Automatic System: Design and Implementation Details

image segment lengths to model segment lengths when the scale difference between the image and model objects is unknown. To compensate for this, the length of each segment is divided by the length of the previous connected segment to provide a scale-invariant ratio (see “Travel, Traversal & Reversal”). Unfortunately, this ratio is very sensitive to the precise locations of the breaks between segments because the division introduces a compounding effect: if a relatively straight contour is divided into a three-unit segment followed by a four-unit segment, the ratio is $\frac{3}{4}$; if the order of these two segments is reversed, the ratio is only $\frac{3}{4}$. However, such discrepancies generally occur only when the contour is fairly straight, resulting in somewhat arbitrary breakpoints between segments; the method is reliable whenever sharp angles between segments provide clear breakpoints.\(^9\)

The matching process is simplified by the fact that model segments are stored sequentially, thus there is never any doubt regarding which model segment should follow the current one or which is the previous segment. However, this is not the case with image segments: any segment may be connected to any number of other segments.

To accommodate this, all matching after the first segment of a super-segment is carried out by a procedure which uses the segment connection lists (see Figure 4.5) to traverse all possible paths through connected segments depth-first. Using recursion, this procedure keeps track of the current path and the cumulative error to this point, thus it can always back up to a previous juncture to try another path without repeating the work performed to reach that juncture. The procedure always tries to construct the longest match possible, stopping only when the accumulated error exceeds the threshold or the cardinality of the match reaches the specified upper limit. Although this recursive procedure introduces combinatorial complexity, it is the only way to ensure that all possible super-segments are examined. This is necessary because noise and nearby objects in the images often give rise to unwanted segments which appear to be connected to other segments, requiring that several paths along the connected segments be examined in order to find a proper match.

---

\(^9\)Using the angle between segments to gauge reliability is discussed in Section 6.2.
Chapter 4: The Automatic System: Design and Implementation Details

Matching the First Segment

Evaluating the first segment of a potential match calls for special care because the previous connected segment, required in order to calculate the relative length and relative angle of the current segment, is not always obvious. Often noise and nearby objects give rise to more than one connected segment in the image and make the resulting relative length and angle values correspondingly unreliable.

The system offers a choice of two methods to deal with this problem. The first method examines each connected segment and calculates the relative value errors only if exactly one segment bears the appropriate connection to the current segment. This solution is based on the notion that if only one previous segment is suitable, it is likely the correct one. If the system locates either fewer or more than one suitable segment, no relative value errors are added to the accumulated error, thereby tending to favour a match.

The second method examines each suitable connected segment and chooses the one which minimizes the accumulated error in relative angle and length, thereby also tending to favour a match. Although this second method would seem to find the correct matches while admitting fewer false matches than the first method, it was found to offer no advantages to justify its slightly increased computation. Nonetheless, the second method was left in the system and can be selected upon execution; the system defaults to the first method for all orientation-dependent matching.

Also assessed during examination of the first segment are the likely directions of traversal and travel of the current object, discussed presently.

Travel, Traversal & Reversal

The linker and segmenter typically traverse boundaries clockwise, meaning that the segment identification numbers increase sequentially as one travels clockwise around the perimeter of an object. Assuming the model was traversed clockwise, matching is most straightforward when the perimeter of the object in the image was also traversed clockwise. In this situation, the first image segment of the object would match one segment in the appropriate model without adjusting the angles or relative lengths, the next image segment would match the next model
segment and so forth until the match is complete.

Under certain circumstances, however, some object boundaries may be traversed counterclockwise in an image. In order to detect matches when this has occurred, the program is able to reverse the direction in which the model is traversed. As mentioned earlier, the program attempts to begin a match on every potentially-successful segment of each model. If a reverse match beginning with the current model segment is possible (i.e., there are enough previous segments in the current model to achieve the minimum cardinality), the error in orientation angle is calculated both normally and after reversing the traversal direction of the current model segment. If the error for the reversed segment is lower than for the forward segment, the system reverses the direction in which the model is traversed. This reversal entails moving backwards through the segments (i.e., examining segment \( n - 1 \) after segment \( n \)), reversing the direction of each individual segment in order to adjust its orientation angle and calculating the relative angles and lengths relative to the segment which follows rather than precedes the current segment in the model (i.e., the relative length is given by \( \frac{l_n}{l_{n+1}} \) rather than \( \frac{l_n}{l_{n-1}} \)). This reverse traversal is also attempted whenever only a reverse match is possible from the current model segment (i.e., there are enough previous segments in the model but not enough subsequent segments to achieve the minimum cardinality).

If a discontinuity divides the outline of a shape into two contours, it is possible for the two contours to be traversed separately from their opposite ends such that the resulting two sets of segments meet at the discontinuity. This situation can be detected because the segments where the two traversals meet will be joined endpoint-to-endpoint rather than endpoint-to-startpoint as with a single continuous traversal. When this is detected, the system continues traversing forward through the model segments but reverses the direction of the individual image segment in order to adjust the angles. It then examines segments connected to the startpoint of the current segment. If the current segment has been reversed, the start-point of the current segment should meet the endpoint of the next segment, thus the system will generally remain in "reverse" mode until the match is completed. If, however, the start-point of the current segment meets the start-point of the next segment, another reversal has occurred and the system returns to "forward" mode. By doing this, the system can handle any number of
reversals (see Figure 4.6).

Figure 4.6: Segments 6 and 7 were derived from a single contour thus they meet endpoint-to-startpoint. However, segment 7 meets segment 23 endpoint-to-endpoint, thus the system moves into "reverse" mode. The system then finds segment 22 by examining the start-point of segment 23. Because segment 23 meets segment 22 startpoint-to-endpoint, the system remains in "reverse" mode. However, segment 37 meets segment 22 startpoint-to-startpoint, thus the system returns to "forward" operation.

Another possible reversal is in the direction of travel of the object in the image: even if orientation angles are being used to specify that the fish swim horizontally, they may swim left-to-right or right-to-left in the images. This corresponds to a perfectly-legitimate reflection of the object (and each of its constituent features and segments) about a vertical axis. The decision of whether the object in the image is traveling parallel or anti-parallel to the object in the model\textsuperscript{10} is also made during examination of the first segment of the match. Once again, the decision is made by comparing the error in orientation angle for both possible orientations of the model segment; the system selects whichever direction of travel minimizes the error.

Orientation angles are used to determine the directions of traversal and travel because the orientation angle is the only matching criterion that is intrinsic to a single segment; all other criteria require the identification of an adjacent connected segment and thereby introduce a possible source of error.

**Matching Randomly-Oriented Objects**

Because the system was designed to identify objects which are generally found in a particular orientation, it would be foolish to neglect this source of information when matching features. However, the system is also able to ignore the absolute orientation of segments in order to identify objects that do not exhibit any preferred orientation. The system can be instructed to

\textsuperscript{10}If orientation angles are being used, the model must describe the object in (one of) its expected orientation(s).
do this either upon execution, in which case no orientation angles are used, or within individual model files, which allows the user to specify that orientation is to be examined for certain objects being identified and not for others.

When matching randomly-oriented objects, the relative lengths of the segments and the relative angles between the segments are the only criteria available to the system. In order to locate the previous connected segment when computing these values for the first segment of a match, the system examines each suitable connected segment and chooses the one which minimizes the accumulated error in relative angle and length, thereby tending to favour a match. This is the second method described in “Matching the First Segment.”

The matching process is slightly simplified when orientation is not being examined because the system need not ascertain the direction of travel: only the orientation angles change if travel is reversed; relative angles and lengths remain the same. However, the system still must determine the direction in which to traverse the model. This is accomplished by computing the relative angle and length for both traversal directions of the current model segment, the traversal direction which minimizes the resulting accumulated error is chosen.

Once the first segment has been matched and a traversal direction has been selected, matching the remainder of the super-segment proceeds precisely as with orientation-dependent matching except that discrepancies in orientation angle between image and model are ignored.

As each segment of a super-segment is matched, the difference in orientation angle between the image segment the corresponding model segment is added to a running total. Upon completion of the super-segment, this total is divided by the number of segments in the super-segment in order to provide an estimate of how the image feature is rotated relative to the model.

Eliminating Redundant Matches

One difficulty of using multiple models and segmentations is that the same image feature may be matched more than once. That is, a single feature in an image may be matched to two different models of that feature or at two different resolutions. Although these superfluous

\[11\text{Recall that reversing the traversal direction also changes the end to which the previous segment must be connected (see "Travel, Traversal & Reversal").}\]
matches can cause confusion during the stereo processing stage, a match cannot be discarded simply because the same image feature was matched to an equivalent model earlier because the new match may be more precise or complete, thereby providing a more accurate location of the feature. Furthermore, it is difficult to determine if a feature has already been matched because different segmentations result in different endpoints and numberings for the segments, thus one cannot simply check whether the current segment has already been matched.

However, because the segmentation process operates repeatedly on a single set of contours, it is virtually guaranteed that the sets of segments comprising different segmentations of a given contour will share at least one common endpoint.\textsuperscript{12} Exploiting this property to eliminate redundant matches, the system examines the list of matches for the current image each time a match is found; if a previous equivalent match shares a segment endpoint with the current match, only the match with the longer arclength is retained.

Any of several values could have been used to compare matches but arclength was found to be the most satisfactory. Choosing the match made at the higher resolution could be misleading because a match made at a lower resolution may represent a larger portion of the feature and hence more accurately reflects the location of the feature. Similarly, comparison based on the number of segments comprising a match is unsatisfactory because a higher-resolution match could represent a small portion of the feature despite having more segments than a lower-resolution match of the entire feature. Selecting the match with lower cumulative error could also result in a shorter match being chosen over a more complete match. Although the proximity of two matches may indicate whether the two are redundant, this test is not as reliable as comparing segment endpoints and the locations cannot indicate which match is preferable.

Comparing arclengths takes care of a number of these problems. First, because both matches are of the same feature derived from a single set of contours, the comparison will not be influenced by differences of scale. Secondly, the "completeness" problem is solved automatically: if a low-resolution match comprises significantly more of the feature than a higher-resolution match, it should have a longer arclength and will thus be chosen. If, however, two matches comprise the same "amount" of a feature, the higher-resolution match will show greater detail.

\textsuperscript{12} The only time this will not occur is if one segmentation is very low-resolution, the other very high-resolution and the contour exhibits high curvature.
and will thus have a longer arclength, causing it to be chosen as we would like.

All super-segments that match a model (i.e., have normalized cumulative error below the allowed threshold) and are not redundant are entered in a linked list of matches. Each entry in this list contains all the information about the match including the feature identification number and the particular model matched, the cardinality of the match, the starting segment, location and arclength of the super-segment in both the model and the image, the relative rotation between the image feature and the corresponding model feature, the hypothesized directions of travel and traversal, the resolution at which the feature was matched and a pointer to a list of supporting features which will be constructed during verification.

Both images of the stereo pair are passed to the matching procedure separately, resulting in two linked lists of features.

4.2.4 Verification

Candidate feature matches are verified by testing whether they are located plausibly relative to one another. The verification procedure examines the matched features in one image at a time, comparing the distance and angle between pairs of different but related image features to the distance and angle between the corresponding model features. These consistency criteria, also used by Stein, are those introduced by Grimson and Lozano-Perez ([GLP84]) and are calculated as follows:\(^\text{13}\)

**Distance** The distance between features in two dimensions. To compensate for differences of scale between the model and the image, the distance between the two features in the model is scaled by the ratio of super-segment arclengths between the image and the model.

**Angle** The angle between features is measured relative to the horizontal. For objects of known orientation, only direction reversal is allowed without penalty; for randomly-oriented objects, arbitrary rotation is permitted.

Related features are simply features which belong to the same object and are thus eligible to verify one another. The user specifies the related features for each model; this helps prevent

\(^{13}\)Grimon and Lozano-Perez (and Stein) also compare the ranges of components of vectors spanning the two scene super-segments and the two model super-segments. This criterion was not used in the current system.
false verifications when features belonging to more than one object are being identified. For example, if two different types of fish are being identified, the user can specify that a head and tail must belong to the same type of fish in order to verify each other, thereby preventing a head from fish A from falsely verifying a tail from fish B.

The method by which the consistency of a pair of matches is determined is very similar to the method used to find matches: the discrepancies in distance and angle between the pair of image features and the corresponding model features are squared, summed and compared to a predetermined threshold. As was done when matching, the discrepancies are divided by predetermined constants prior to being squared, once again to provide a "pivot point" and to define an equivalence between discrepancies in length and in angle. Thus the verification error is given by:

$$\text{error} = \left( \frac{\theta_i - \theta_m}{c_{\text{angle}}} \right)^2 + \left( \frac{d_i - (\frac{a_i}{a_m})d_m}{c_{\text{distance}}} \right)^2$$

where $\theta$ and $d$ refer to the angle and distance between the two features, $a$ refers to the arclength of the matched super-segment, $c_{\text{angle}}$ and $c_{\text{distance}}$ are predetermined constants and the subscripts $i$ and $m$ refer to values in the image and model respectively. Two features are considered consistent with one another if the cumulative error in their relative positions is below a predetermined threshold; lowering this threshold makes the verification more strict.

To expedite verification, the list of matches is first sorted so that all instances of a particular feature are grouped consecutively, followed by all instances of the next feature and so forth. Given $t$ different feature identification numbers, verification now proceeds by checking all instances of feature $m$ against all instances of related feature $n$ for $1 \leq m < t$ and $m < n \leq t$. In English, each instance of each feature is compared to each instance of every related feature bearing a higher feature identification number.\(^{14}\) Because features are sorted by increasing identification number, the comparisons begin with the first feature in the list bearing a higher identification number than the current feature and continue to the end of the list of features. The reflexive nature of verification (i.e., A verifies B $\Rightarrow$ B verifies A) ensures that checking a feature against features with lower identification numbers would be redundant.

\(^{14}\)These identification numbers actually refer to the position of a group of equivalent features within the sorted list and need not correspond to the assigned feature identification numbers.
The angle between features is handled differently depending upon whether the objects being identified have definite or random orientation. If the objects have definite orientation, the angle between the image features must match either the angle between the corresponding model features (if the image object was identified traveling parallel to the model object) or the reflection of this angle about the vertical axis (if the image object was identified traveling anti-parallel to the model object) (see “Travel, Traversal & Reversal”). Two orientation-definite features can verify each other only if they were both identified traveling in the same direction.

If the objects exhibit no preferred orientation, the verifier must determine where two image features should be located relative to each other in order to determine whether or not they are consistent. To do this, the verifier uses the previously-calculated estimate of the relative rotation between the current image feature and the corresponding model feature (see “Matching Randomly-Oriented Objects”). The angle between the two model features is rotated by this angle before calculating the discrepancy in orientation. This effectively aligns the model with the first image feature and then determines whether the second image feature being considered is in the appropriate location. The system cannot check whether the two features were traveling in the same direction because no direction of travel is calculated for randomly-oriented objects.

The user specifies the number of consistent features required to constitute verification. In the NIFE system, only one consistent feature is necessary for verification as only heads and tails are available to verify each other; if another application were to identify several features on each object, the user could demand more consistent matches to make verification more rigorous.

Each feature maintains both a counter and a list to keep track of how many and which features have been found to verify it. Whenever a consistent pair of features is found, both features have their associated counters incremented to reflect this new support and a pointer to the appropriate supporting feature is appended to each support list. Thus if feature B is found to be consistent with feature A, both A and B will have their support counters incremented and a pointer to B will be added to the support list for A while a pointer to A will be added to the support list for B. This once again highlights the reflexivity of verification.

By keeping track of supporting features in this way, the verification process not only helps eliminate false matches, those that are not consistent with any others (as judged by location),
but also serves to associate features that likely belong to the same object. With the NIFE system, for example, if a head is found to be consistent with a tail, they are probably parts of the same fish and the apparent length of that fish is simply the distance between them.

The verification step relies upon one of the assumptions made during the design of the system, the assumption that each object has at most one of each feature. Because the verification procedure checks each feature against only features with different identification numbers, two equivalent features\(^{15}\) would not be checked against one another. Thus the two wheels of a bicycle would not be examined to verify one another. If this restriction were removed, steps would have to be taken to prevent a feature from verifying itself.

Finally, because the verification procedure calculates distances and angles between features in the models, all models of features belonging to single object must be acquired from images which show the features in the same locations and at the same scale. If the model of a head were taken from the top left corner of the image and the model of a tail from the bottom left, the calculated distance between the head and tail would be arbitrary and the calculated angle between them would be 90° (vertical) rather than 0° (horizontal). Of course, this could all be accounted for by scaling and translating features or models as necessary, but it has proven easier to acquire models from similar images. All models of a single fish used in testing were acquired from different segmentations of the same image.

4.2.5 Stereo

Once the features have been verified and associated with their supporting features, the stereo procedure attempts to ascertain the distances to individual fish by identifying pairs of corresponding features in the two images of a stereo pair and examining the disparity between them (see Appendix A). The stereo procedure also examines the locations of features in both images to interpret ambiguous matches, those in which one head, for example, is consistent with two tails. The system favours accuracy of results over quantity, thus any matches which cannot be interpreted reliably are discarded. This policy should not result in a shortage of usable data.

\(^{15}\)Recall that equivalent features are features that have been matched against models bearing the same feature identification number; multiple models of a tail acquired at different resolutions would all have the same identification number and would be equivalent models.
because images are easily collected: at 7.5 stereo pairs per second, a one-hour videotape of the fish circulating in a net pen provides 27000 image pairs, each of which may show several fish.

In order to determine the actual length of a fish, the system requires the apparent length of the fish (in pixels) as an indication of size and the distance to the fish to establish scale. The apparent length can be calculated provided the system can locate both the head and tail of a single fish in one image. The distance to a fish is calculated from the disparity exhibited by a particular feature of the fish, requiring that that feature be identified in both images of a stereo pair. Hence in order to determine the actual size of a fish from a stereo pair, the system must be able locate both the head and tail of a single fish in one image and either the head or tail of the same fish in the other image. Of course, more information can be put to use: if both the head and tail are present in both images, the system can double-check or average two apparent length measurements and can obtain estimates of the distances to the head and tail independently.

Dimensions are calculated as distances between pairs of features, thus the stereo procedure is provided with the identification numbers of those features which delimit the particular dimension desired. For example, if the heads tails, dorsal fins and ventral fins have been located and are identified by the numbers 1, 2, 3 and 4 respectively, the length of the fish is determined by having the stereo procedure find the distance between features 1 and 2 and the width of the fish can be determined from the same lists of matches by having the stereo procedure calculate the distance between features 3 and 4. By doing this, the system can perform all the matching and verification at once, regardless of how many dimensions are being calculated, and avoids redundancy, particularly when a single feature delimits more than one dimension.

The stereo procedure begins by searching for an instance of a head in the left image and, if one is found, searches for other heads on the same epipolar line of the same image. Two features are considered epipolar if their vertical separation (in pixels) is below a pre-set limit or if this separation would be below the limit had the two features been matched to the same

---

16Standard video captures one image every 1/15 second.

17This explanation will use the concrete example of estimating the length of fish by identifying heads and tails for ease of understanding; the system will of course calculate the dimension between any two arbitrary features.
model. This second test is required because one instance of a feature may, as a result of different segmentation, have its centrepoint (which is considered its location) at a slightly different level than some other instance of the same feature. If these two features were matched to two models with similarly-displaced centrepoints, the system would scale the vertical displacement between the centrepoints of the models by the ratio of the image arclength to the model arclength and examine the locations again with this offset taken into account (see Figure 4.7).

![Diagram showing two instances of the same feature at different resolutions. The right-hand instance is located above the left-hand instance even though both features occupy the same region of the image.](Figure 4.7: Two instances of the same feature at different resolutions. The right-hand instance is located above the left-hand instance even though both features occupy the same region of the image.18 Provided these two instances matched two models with similarly-displaced centrepoints, the offset \(d\) would be taken into account and the two would be considered epipolar.

If more than one verified head is found on the current epipolar of this image, the system ignores this epipolar and searches for another head elsewhere in the image. It does this because if multiple heads were verified on the same epipolar line, the information on that epipolar will likely be difficult to interpret reliably; although the system does attempt disambiguation, it is preferable to begin with at least one unambiguous feature. Furthermore, as was mentioned above, there should be enough images available that there is no need to go to great trouble or computational expense to interpret particularly difficult data, especially when such data is less likely to provide reliable results.19 Note that the system accepts unverified features because they often supply useful information provided the missing information is available elsewhere (as will be explained shortly).

Assuming only the one verified head is found on the current epipolar, the system examines its list of supporting features to determine if it was verified by one or more tails. The right

---

18 Recall that for super-segments of even cardinality, the location of the super-segment is that of the middle vertex; for super-segments of odd cardinality, the location is the midpoint of the middle segment.

19 With improved disambiguation and record-keeping, this shortcoming could be addressed in future work.
image is then searched for corresponding heads and tails.

Of course, the ideal situation would be to locate a single head supported by a single tail in the left image and the corresponding head and tail in the right image. If this happens, the system obtains two independent values for the disparity, one from each pair of corresponding features, and two independent estimates of the apparent length, one from each image. The difference between the two disparities and the percent difference between the two length estimates must be below pre-set limits to help detect false matches. Assuming the values are acceptable, the system creates a new entry in a linked list of "objects" and records the two disparities, the average of the two apparent lengths and the features involved. The two apparent lengths are averaged because the system has no way to determine which one is more accurate; if it is found to be advantageous, the system can be made to choose the larger length on the assumption that contortions may make the fish look shorter than it really is but never longer. Both disparities are retained because the two features could legitimately be at different distances from the cameras.

Unfortunately, factors such as multiple objects, occlusion, noise, poor contrast and the like often result in more or fewer features being detected. If fewer features are detected, the system attempts to make use of the information that is available. Recall that in order to calculate an actual length, the system must locate both the head and tail of a single fish in one image (to obtain apparent length) and either the head or tail of the same fish in the other image (to determine distance). With this in mind, if the system locates a head with no corresponding tail (i.e., an unverified head) in the left image, it examines the epipolar line in the right image for a corresponding head which has been associated (verified) with at least one tail. If it finds such a head, it can attempt to ascertain the apparent length of the fish from the head and tail in the right image and the disparity from the two corresponding heads. Similarly, if the head located in the left image was verified by a tail, the system can calculate the apparent length from these features and the disparity given only an unverified head or tail in the right image. Of course, when one of the features is missing, the system has no way to double-check the accuracy of either the length or the disparity estimate, nor can it determine whether both the head and tail are the same distance from the cameras. Nonetheless, the ability to make use of all available information is important: even in the absence of noise and occlusion, frequently
the disparity between a pair of stereo images will alone result in a feature being absent from one image while the entire object is visible in the other image. It is for this reason that the stereo system examines unverified features.

The opposite situation occurs when the recognition system locates more features than are expected, as would happen if one fish were partially hidden behind another fish such that only part of it were visible. If this occurs, the system has three procedures by which it identifies the features that most likely offer the correct information. These procedures are based on the following criteria:

**Disparity Range** Because disparity is a function of distance, the range of acceptable disparities specifies the range of distances within which the objects are either confined or likely to be found.

**Discrepancy Limit** This limit specifies the maximum allowable difference between the disparity of a pair of heads and that of the corresponding pair of tails. Limiting this discrepancy helps eliminate false matches and gives the user some control over the range of orientations that will be accepted.

The first procedure is used when at least one head is found in one image and more than one epipolar head is found in the other image. The disparity is calculated for each possible pair of heads in the two images; if only one pair yields a disparity within the allowed range, those two heads are considered correct. This procedure is particularly valuable if the objects are confined to a region such that it is impossible (rather than just unlikely) for the disparity to be outside the specified range.

The second procedure is called upon when an uncontested disparity has been obtained for a pair of heads but one or both of the images has more than one tail associated with the head. This situation could easily occur if one fish is partially occluded by another fish such that only its tail is visible, resulting in the head and tail of the closer fish appearing in both images and the tail of the more distant fish appearing in at least one image (see Section 5.3). If this occurs, the disparity between the two more distant tails should be significantly smaller than that of the heads simply because the tails are more distant than the heads. Similarly, the “disparity”
between a closer tail and a more distant tail should also differ significantly from the disparity of the heads because it reflects a shift in the absolute location of one of the features in addition to a shift in camera position. Thus only the disparity between the two correct (closer) tails should be sufficiently close to the the disparity between the heads. To identify these two tails, the disparity is calculated for every contending pair of tails and if the disparity of exactly one pair of tails matches the disparity of the heads to within the allowed discrepancy, that pair of tails is chosen.

The final procedure is used if the recognition system has located one head in the left image, one or more epipolar heads in the right image and one or more corresponding tails in both images. This procedure mates the head in the left image with each head in the right image, calculates the resulting disparity, then tests every possible pair of tails to determine which head (in the right image) and pair of tails provide a disparity match within the allowed disparity range and discrepancy. Once again, if exactly one pair of heads and one pair of tails are found to satisfy these criteria, these pairs are chosen. This same procedure could in principle be used in the event that multiple heads and multiple tails were found in both images, but this situation cannot arise because the system discards any occurrence of multiple epipolar verified heads in the left image. As was mentioned earlier, it is unlikely to interpret such a collection of features reliably if every feature is open to multiple interpretations.

In keeping with the reliability requirement, each of these procedures succeeds only if exactly one allowable match is found: if more than one is found, the system cannot reliably select any one and discards the features.

The latter two of these disambiguation techniques rely on the disparity between tails equaling the disparity between heads to within the allowed discrepancy. This implies that the heads and tails are expected to be found at roughly the same distance from the cameras, meaning the fish are roughly parallel to the image plane. While this situation can generally be arranged when videotaping fish for the NIFE system, it may be difficult or impossible to arrange with other objects in other applications. In such applications, the allowed discrepancy may have to be be increased in order for any feature pairs to be accepted. If the objects are free to assume

---

20: The routine which compares the disparity between heads to that between tails when exactly one pair of each was found makes the same assumption.
any orientation, the only solution may be to set the limit to the maximum possible discrepancy, presumably allowing a difference in distance almost equal to the length of the objects. Although this would appear to defeat the purpose of having such a limit, the limit may still serve to eliminate preposterous pairings between falsely-matched features and would also give the user some control over the range of orientations which will be accepted.

Finally, there is an undesired side-effect to these arrangements: the choices of which feature and image are examined first can affect which particular fish are measured. Assume the recognition system has located two verified heads and one tail on a given epipolar in the left image and one head and one tail on the same epipolar in the right image. If the stereo system is instructed to begin by searching the left image for heads, it will find two on the same epipolar and will discard this epipolar because it is too difficult to interpret reliably. However, if the system begins by searching the left image for tails, it will find the one tail and will attempt to assess the size of that fish. If the right image is examined first, it makes no difference which feature is sought first because the stereo system will find one of either feature and will attempt to determine the size of the fish. Similarly, had no head been found in the left image, a search for heads in this image would yield nothing while a search for tails or a search of the right image would likely result in the fish being evaluated. Nonetheless, the choices of which feature and image to examine first are arbitrary: the goal of the system is to provide a statistical estimate of the size distribution and it is assumed that any effects of these decisions will vanish in any statistically significant sample.

The information returned to the user consists of a list of those feature pairs which the system considers to comprise objects, the apparent distance between these pairs of features (in pixels) and either one or two disparity values (in pixels), depending on which features were located. Once the system has been calibrated, the disparity values can be converted into distance estimates (using the equations of Appendix A) and, once this is done, the apparent lengths can be converted into actual lengths (using the equations of Appendix B). For development and testing purposes, the current system also provides both lists and images showing the candidate features identified in each image and those candidate features that were successfully verified.
CHAPTER 5
RESULTS

[I]t is a natural instinct to be charmed by one's own productions. That's why raven chicks are such a delight to their parents and mother apes find their babies exquisitely beautiful.

- Thomas More, "Utopia"

This section provides some examples which illustrate the successes and shortcomings of the system. Model and test images for the first three examples were created using a set of black paper cutouts of an American Shad and a Requiem Shark.\(^1\) These were placed on a sheet of textured Styrofoam and imaged from above. All models were created from different segmentations of two images, one of the shad and one of the shark. The models show the fish traveling from right to left and boundaries are traversed clockwise. The images used to generate models were not used in subsequent testing.

The images for the fourth example were captured in a commercial sea cage using the NIFE apparatus (described in Section 3.1). These images show Pacific salmon; models were generated from other images captured in the same tank.

5.1 EXAMPLE #1

The first stereo pair was created from a single image using a graphics editor. The original image was produced by photocopying the American Shad at three different magnifications, placing the three resulting cutouts on a sheet of white Styrofoam and capturing an image with the Styrofoam sheet roughly parallel to the image plane. This image was copied and, using a graphics editor, the three fish in the copied image were shifted to the right as they would have

\(^1\)These outlines were photocopied from [Sau92].
been had a second image been captured simultaneously by a camera located slightly to the left of the camera used. Thus a simulated stereo pair was created. To show that the system is tolerant to imprecise camera calibration, the fish in one image were shifted vertically by small amounts. One of the fish was also slightly shortened in one image. The graphics editor was then used to clutter both images with various shapes, some random and some resembling parts of fish. The resulting images are shown in Figure 5.1a.

The program was supplied with two models of the shad head and one model of the tail. The system segmented the images at two resolutions and sought all super-segments consisting of between three and five segments.

The results of matching stage are displayed in Figure 5.1b and detailed (for the right image) in Table 5.1, both generated by the matching procedure. The images show that the matching procedure successfully located all the correct features plus a few of the added features. In the right image, one of the added features resembling a head and one resembling a tail were falsely matched (second head and tail from the bottom).

These images illustrate the value of multiple models: the head of the middle fish (see left image) does not show any mouth detail while the upper and lower heads show the mouth clearly. Correspondingly, the middle head (feature #5 in Table 5.1) was matched to model #0 while the other two heads (features #1 and #4) were matched to model #1.² Because they are equivalent features, all heads have feature ID #1 (column #4).

Figure 5.1c and Table 5.2 show those matches that were successfully verified. All false matches were correctly identified and removed.

Table 5.3 displays the final output of the system: the apparent lengths and associated disparities of the three objects identified in the images. Note that the disparities of the head and tail of the second fish differ because this fish was shortened slightly in one image only.

The locations of the six verified features indicate that the epipolar alignment was off by between 7 and 11 pixels between the two images, a misalignment that would have hampered the performance of correlation-based stereo.

By segmenting only those contours over 24 pixels in length,³ the number of contours was

²The false head in the right image (feature #3), showing no mouth detail, was also matched to model #0.
³The minimum length is set upon execution.
Chapter 5: Results

Table 5.1: Candidate matches: The left # column provides reference numbers for the features, the Edges column refers to edge numbers in the image of the matches, model gives the number of the model matched, ID shows the feature identification number of the match, seg gives the number of the starting model segment of the match, res tells at which resolution the feature was matched, nsegs gives the cardinality of the match and score provides the cumulative discrepancy. The two right-hand columns give the location of the feature match in the image and the model.

<table>
<thead>
<tr>
<th>#</th>
<th>Edges</th>
<th>Model</th>
<th>ID</th>
<th>Seg</th>
<th>Res</th>
<th>nsegs</th>
<th>Score</th>
<th>image-centre</th>
<th>model-centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1 2</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>(6.5, 466.0)</td>
<td>(32.5, 305.5)</td>
</tr>
<tr>
<td>2</td>
<td>3 4 5</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>15.37</td>
<td>(226.5, 293.0)</td>
<td>(34.5, 309.5)</td>
</tr>
<tr>
<td>3</td>
<td>6 7 8</td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>17.45</td>
<td>(71.5, 199.0)</td>
<td>(34.5, 309.5)</td>
</tr>
<tr>
<td>4</td>
<td>9 10 11</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>8.18</td>
<td>(133.5, 146.0)</td>
<td>(32.5, 305.5)</td>
</tr>
<tr>
<td>5</td>
<td>12 13 14</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>21.84</td>
<td>(84.0, 325.5)</td>
<td>(34.5, 309.5)</td>
</tr>
<tr>
<td>6</td>
<td>15 16 17</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>12.81</td>
<td>(29.5, 427.0)</td>
<td>(433.5, 273.0)</td>
</tr>
<tr>
<td>7</td>
<td>18 19 20</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>15.68</td>
<td>(144.0, 383.0)</td>
<td>(425.0, 208.5)</td>
</tr>
<tr>
<td>8</td>
<td>21 22 23</td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>11.20</td>
<td>(378.5, 362.5)</td>
<td>(433.5, 273.0)</td>
</tr>
<tr>
<td>9</td>
<td>24 25 26</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>9.71</td>
<td>(311.5, 219.5)</td>
<td>(433.5, 273.0)</td>
</tr>
<tr>
<td>10</td>
<td>27 28 29 30</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>19.03</td>
<td>(180.5, 453.5)</td>
<td>(409.5, 242.5)</td>
</tr>
<tr>
<td>11</td>
<td>31 32 33 34</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>(331.5, 283.5)</td>
<td>(409.5, 242.5)</td>
</tr>
<tr>
<td>12</td>
<td>35 36 37 38</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td>(333.5, 152.5)</td>
<td>(409.5, 242.5)</td>
</tr>
</tbody>
</table>

Table 5.2: Verified matches: This table shows the details of each feature successfully verified. The edges column gives edge numbers in the image of verified features while the # column corresponds to the list of candidate matches (Table 5.1).
Table 5.3: Measured objects: The apparent length is given in pixels, the two disparities (in pixels) are of the two pairs of features (heads & tails) and the remaining four columns indicate which features were used to obtain these measurements (feature numbers refer to Table 5.1).

<table>
<thead>
<tr>
<th>App.Length</th>
<th>disp1</th>
<th>disp2</th>
<th>f1l</th>
<th>f2l</th>
<th>f1r</th>
<th>f2r</th>
</tr>
</thead>
<tbody>
<tr>
<td>174.4</td>
<td>16.0</td>
<td>16.0</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>201.7</td>
<td>70.0</td>
<td>73.0</td>
<td>3</td>
<td>11</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>251.0</td>
<td>64.0</td>
<td>64.0</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>11</td>
</tr>
</tbody>
</table>

reduced from 621 to 115 in the left image and from 760 to 183 in the right. Running on a Sparc2 workstation, the system required 22.8 seconds including the time to generate and store the diagnostic images and information. Adding an extra head model increased this time requirement by only 0.7 seconds but the extra model resulted in an false match which could not be reliably disambiguated, thus only two of the three fish were successfully measured.

Figure 5.1a: The left and right image contours for example #1.

Because linked contour images form the input to the system, times given in this paper do not include the time required for edge detection or linking. These typically required 18 and 0.5 seconds respectively.
Figure 5.1b: Candidate matches from the left and right images.

Figure 5.1c: Verified matches from the left and right images.

5.2 Example #2

The images in the second example contain both a shad and a shark and were created using a pair of stereo cameras. The images are shown in Figure 5.2a.

Six models were provided: three shad heads, one shark head and one of each tail. The system segmented the image at two resolutions and sought all super-segments with cardinality between three and five.
Chapter 5: Results

68

<table>
<thead>
<tr>
<th>#</th>
<th>Feature</th>
<th>Model</th>
<th>Edges</th>
<th>image_centre</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>shktail1</td>
<td>3</td>
<td>0 1 2 3 4</td>
<td>(438.0, 362.5)</td>
</tr>
<tr>
<td>2</td>
<td>shkhead1</td>
<td>2</td>
<td>5 6 7</td>
<td>(253.0, 373.5)</td>
</tr>
<tr>
<td>3</td>
<td>tail</td>
<td>1</td>
<td>8 9 10 11</td>
<td>(454.5, 183.5)</td>
</tr>
<tr>
<td>4</td>
<td>head2</td>
<td>0</td>
<td>12 13 14</td>
<td>(298.5, 205.0)</td>
</tr>
</tbody>
</table>

Table 5.4: The four verified matches in the left image.

<table>
<thead>
<tr>
<th>App_Length</th>
<th>disp1</th>
<th>disp2</th>
<th>f1l</th>
<th>f2l</th>
<th>f1r</th>
<th>f2r</th>
</tr>
</thead>
<tbody>
<tr>
<td>161.4</td>
<td>241.5</td>
<td>233.0</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>185.7</td>
<td>225.0</td>
<td>224.0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.5: The two objects identified in the images.

Figure 5.2b shows the candidate feature matches. Only the correct image features were identified during the matching stage, hence the images of verified features are identical. Examination of the polygonal approximations reveals that the shad outline was traversed counter-clockwise in the right image, requiring the system to reverse its traversal of the models to locate this fish. The lists of verified matches (see Table 5.4) shows that the four required features were all correctly identified. The resulting measurements are shown in Table 5.5.

Once again to show that the system is tolerant to imprecise camera calibration, the cameras were crudely adjusted by hand while viewing their output on a monitor. The locations of the four verified features show epipolar misalignment of up to 8.5 pixels.

Segmenting only contours over 24 pixels in length reduced the number of contours from 238 to 47 in the left image and from 325 to 68 in the right, resulting in an average time requirement of 3.0 seconds. Diagnostic information indicated that only four of the six models were used; eliminating the two unused models reduced the time requirement to an average of 2.7 seconds.
Chapter 5: Results

5.3 Example #3

This image pair was created by placing the shad horizontally and the shark diagonally on the Styrofoam. The shark was also raised slightly to give it a different disparity. To demonstrate stereo disambiguation, an extra shad tail was placed behind the shad, thereby simulating a partially occluded shad. The Styrofoam sheet was then cluttered with random laboratory junk and the scene was captured with a pair of stereo cameras. The images are shown in Figure 5.3a.

The system was provided with two models of a shad head, one model of a shark head and one model of each tail. The shad models were declared orientation-dependent, exploiting the horizontal orientation of the shad, while the shark models were declared orientation-independent to identify the shark in its diagonal orientation. Once again, images were segmented at two
resolutions and all super-segments with cardinality between three and five were sought.

The images of candidate matches (see Figure 5.3b) show both the correct matches (all correctly identified) and many false matches. The false matches are caused partially by the many random objects, some of which resemble fish parts when segmented, and partially by the many models and the arbitrary orientation permitted for the shark models. For example, if the model of the shark tail is rotated roughly 150°, three of its segments match the dorsal fin of a shad very closely (see Figure 5.4). This accounts for the matched shad dorsal fin in the left image.

Inspection of the polygonal approximations reveals that the shark was traversed counterclockwise in the right image and the segments constituting the shark head were numbered erratically in both images. The shad in this image pair is reflected about the vertical axis relative to the shad models. The system handled all these abnormalities without difficulty.

The verifier managed to eliminate most of the false matches from the left image but several remained in the right image because of coincidental image features (see Figure 5.3c). However, these false matches do not affect the final size information; all are eliminated during stereo processing either through disambiguation or because there are no corresponding features in the left image. The resulting size information is shown in Table 5.6.

Examination of the diagnostic information reveals that the stereo disambiguation procedure selected the correct pair of tails by comparing the disparities of all possible pairs of shad tails to the disparity of the heads and finding only the correct pair of tails within the allowed disparity discrepancy.

These images contained 521 (left) and 570 contours (reduced to 113 and 146 respectively)

\[\begin{array}{ccccccc}
\text{App.Length} & \text{disp1} & \text{disp2} & \text{f1l} & \text{f2l} & \text{f1r} & \text{f2r} \\
166.4 & 192.5 & 198.0 & 1 & 15 & 1 & 56 \\
305.8 & 254.5 & 252.0 & 38 & 18 & 45 & 24 \\
\end{array}\]

Table 5.6: The size and disparity estimates.

\(^5\)Recall that, to accommodate fish crossing the cameras in both directions, this is the only orientation difference allowed without penalty in orientation-dependent matching.
Chapter 5: Results

Figure 5.3a: The left and right image contours for example #3.

Figure 5.3b: Candidate matches from the left and right images.

Figure 5.3c: Verified matches from the left and right images.
Figure 5.4: (a) A portion of shark tail rotated 150°. (b) Shark head. (c) Shad dorsal fin. Segments with corresponding numbers could be confused, thereby causing false matches even when several adjacent segments are examined.

and required 11.2 seconds to analyze. Supplying three extra models increased the time requirement by only 0.8 seconds but once again yielded a false match that could not be disambiguated, resulting in measurements for only one of the two fish.

5.4 EXAMPLE #4

The images for the final example were captured at a salmon farm using the NIFE equipment. The fish are Pacific Salmon and the only change to their environment was the addition of a light-coloured backdrop which the farmers said would not disturb the fish at all. The images are shown in Figure 5.5a and the resulting contours in Figure 5.5b.

Two models, one head and one tail, were generated from a fish found in another image. This fish was chosen for the models because it appeared similar to the fish in the test image but the resulting models were not altered in any way to aid the system.

The system parameters were set as for previous examples except the matching threshold had to be increased by roughly a factor of two in order to identify necessary features.
Chapter 5: Results

The images of candidate matches (Figure 5.5c) show that the head and tail of the uppermost fish were correctly identified in both images as were two other tails in the left image and one head in the right image; the remaining matches are incorrect. The the large fish (centre of left image) was not measured because its tail is partially occluded and its head is missing in the right image. Although the head and tail likely could be identified in the left image if more models were provided, this fish cannot be evaluated because neither its head nor tail is identifiable in the right image.

The verification stage removed half of the incorrect matches in both images (see Figure 5.5d). The false matches do not lead to incorrect measurements because for each false match in one image, there is no corresponding feature in the other. The system provides the correct values for the one fish correctly identified in both images:

<table>
<thead>
<tr>
<th>App.Length</th>
<th>disp1</th>
<th>disp2</th>
<th>f1l</th>
<th>f2l</th>
<th>f1r</th>
<th>f2r</th>
</tr>
</thead>
<tbody>
<tr>
<td>176.6</td>
<td>106.5</td>
<td>95.5</td>
<td>1</td>
<td>14</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

The system analyzed 217 of 775 segments in the left image and 235 of 1156 in the right image; the time required was 15.3 seconds including the time to generate and store the diagnostic information (approximately 0.9s).

Figure 5.5a: The left and right images for example #4.
Chapter 5: Results

Figure 5.5b: The left and right image contours.

Figure 5.5c: Candidate matches from the left and right images.

Figure 5.5d: Verified matches from the left and right images.
5.5 General Observations

A number of general trends became apparent while working with the system:

- To minimize false matches, the shortest acceptable match (lowest cardinality) should generally be three segments. It is unlikely for three segments to join in a configuration similar enough to a model to produce a false match. However, when the minimum required cardinality was reduced to two, the number of false matches resulting from coincidental arrangements increased sharply. Although this does not prevent the system from identifying the correct features, the system may fail to measure identified objects if the false matches cannot be successfully eliminated or disambiguated.

- Setting the matching threshold too high or providing extra models likely increases the number of false matches. Once again, even if the desired features are correctly matched, the false matches may result in a loss of measurement information if they cannot be successfully eliminated or disambiguated.

- The system locates matches much more successfully when there are sharp angles between successive segments in the model shapes. Such angles provide well-defined breakpoints between segments, resulting in reliable segment lengths and angles. During testing, the system located the highly-angular shad tails and shark tails very reliably despite having only one model of each. The system had difficulty identifying the much more rounded shad head even with multiple models because the lack of sharp angles leads to somewhat arbitrary breakpoints between segments, hence lengths and angles are less reliable.

- The shark head illustrates a difficulty with the system. Because the two long segments running from the top of the shark ventral fin to the front of the head (see Figure 5.4b, segments 2 and 3) form an almost-straight line, they are often approximated as a single straight segment. When this happens, the shark head becomes a two-segment shape resembling a “V” on its side and, consisting of only two segments, is ignored when the minimum required cardinality is three. The simple solution is to add a two-segment model of the head and reduce the minimum required cardinality to two. Unfortunately,
Table 5.7: Time requirements for the examples above. Note that examples 1 and 3 have similar numbers of segments but example 1, even with half as many models, requires roughly twice the time.

as was mentioned above, lowering the minimum cardinality to two will likely increase the incidence of false matches.

- The majority of unexplained false matches\(^6\) are to small features in the image; rarely are features that appear large in the image falsely matched without explanation.

- Although the system did sometimes fail to measure objects in images, never did features in an image pair conspire to yield measurements for a non-existent object.

### 5.6 Complexity

The time complexity is difficult to calculate because it depends not only on the number of image segments but also on their configuration and shape. It is clear, however, that the most time-consuming operations are, in order, matching and segmentation, which process image segments and contours respectively. The verification and stereo procedures, which examine matched features (hence less information), complete their tasks almost instantly. The execution times for the examples above are given in Table 5.7.

Because the recursive matching procedure traverses all possible paths through connected segments, its time requirement is higher if many of the segment endpoints lie close to one another.
another, resulting in many possible paths and longer paths. On a random image, the time required to locate candidate matches should increase with the number of image segments, itself a function of the segmentation resolution and the number of resolutions analyzed, and the number of model segments.

The time requirement of the segmentation procedure, which increases with the number of contours in the image, is reduced by segmenting only those contours exceeding a pre-set minimum length. However, it also depends on the length and shape of the contours. The time required to locate the point of maximum deviation on a contour increases with the number of pixels in the contour, thus shorter contours are processed more quickly. Also, straighter contours are generally divided into fewer straight segments and are hence segmented more quickly than contours with many curves or high curvature.

Interestingly, the number of models has only a very small effect on the time requirement. Even if the segmentation time and I/O time are subtracted from the total to leave only the matching time, the time difference between executions with different numbers of models is not explained by the number of models. This result indicates that several models can be included to increases robustness without causing unreasonable execution times. A number of system parameters can also affect the time requirement:

- Higher cardinality matches generally take longer to construct and involve the examination of more paths. However, shorter matches cause the system to attempt more matches because fewer model segments are required to achieve the minimum cardinality.

- Increasing the segmentation resolution increases the number of segments involved in matching. Of course, each segmentation analyzed involves matching two new images (left and right).

- Increasing the radius within which two segment endpoints are considered connected increases the number of connected segments and hence the number of paths that must be attempted.

- Reducing the minimum required contour length results in more contours being segmented and analyzed.
Don't let it end like this. Tell them I said something.

- Last words of Pancho Villa

6.1 Conclusions

I have presented a unified system to identify known objects in images and, using stereo information, determine the distance to these objects and their dimensions. Whereas many such systems perform stereo analysis first and segment the stereo output to identify objects, this system locates features first. This permits successful analysis of images where poor imaging conditions such as low contrast, pronounced backscattering, specular reflection and imperfect epipolar alignment would make standard stereo analysis difficult. The approach also reduces computation by calculating disparity only where it is required. The computational requirements of the system are reasonable so that the system can be used on inexpensive hardware by small enterprises.

The basic feature used in recognition is obtained by grouping adjacent segments; a shape is described by the lengths of the segments comprising it and the angles between them. In addition to this information, which is invariant to rotation, the system is able to examine the absolute orientation of segments in order to exploit any regularity of orientation exhibited by the objects being recognized.

The feature-matching routine performs well on highly-angular shapes where individual segments are well-defined; the shape representation is particularly well-suited to such shapes. Performance is not as good when the shapes being recognized are smooth. Overall recognition performance is improved by the use of multiple models and multiple image segmentations. The
verification stage both eliminates false matches and identifies those features which belong to a single object. Specified dimensions are then calculated as the distance between these features.

The stereo procedure is equipped with techniques to interpret ambiguous matches resulting from occlusion and imperfect recognition. The information returned is easily converted to distance and size information using the equations of stereopsis and optics.

With many features and promising performance, the current system provides an excellent framework for research and development in recognition and stereo.

6.2 Future Work

You never really finish a master's thesis, you just decide to stop working on it.

- Scott Flinn

The first step in improving the system would be to make the recognition stage more robust. A number of steps exist which could give the matcher more power and increase the amount of information brought to bear upon matching within the existing framework.

- The system has difficulty calculating lengths and angles reliably without fairly sharp angles between segments. This difficulty arises because, as the angle between two segments approaches 180°, meaning the two segments form a fairly straight line, the location of the breakpoint between them becomes somewhat arbitrary. To remedy this, the angle between segments could be used to estimate the accuracy of calculated values and the resulting discrepancies could be weighted accordingly. Thus, as the angle between two segments approaches 180°, the system could reduce the effect of the resulting discrepancies on the cumulative error of the match to reflect the fact that the values are not as reliable. Conversely, if two segments met at a relatively sharp angle such as 90°, the effect of the resulting discrepancies could be magnified. This weighting could be built into the models, where the angles between segments remain constant, or calculated for each pair of image segments.
The segment-by-segment matching strategy likely fails if a segment is missing from the image or model (see Figure 6.1a). Although this situation could be accommodated by providing more models, it could be handled better by allowing the matcher to "look ahead" one segment if a promising match, perhaps two or more segments long, encounters an image segment that does not match the current model segment. If the following image segment matched the model segment, the system could assume that a segment was missing and continue the match. Ideally the matcher could skip either an image or a model segment because the segment could be missing from a or an image.

A problem similar to that described above arises if two segments in the model appear as one segment in the image or vice-versa, as could happen if the two segments meet at close to 180° (see Figure 6.1b). Rather than provide another model or reduce the minimum cardinality, the matcher could combine two such segments and compare the resulting single segment to the corresponding image or model segment.

Having two distinct modes for recognition is somewhat restrictive and may neglect valuable information about objects. During orientation-dependent matching, penalizing a potential match for any deviation from the model orientation may be too strict in all but the most controlled environments. Conversely, during orientation-independent matching, the freedom to rotate models arbitrarily is unnecessarily permissive and discards valuable information: Atlantic salmon may swim randomly in all directions but they don't swim upside-down. Furthermore, this arbitrary rotation can cause false matches (see Figure 5.4). Instead of these two distinct modes, each model could specify an allowable range of orientations for that object; this range could be as restrictive or permissive as the objects and environment dictate. If this were done, the system should also have a least-squares line-fitting stage or some similar means of accurately determining the orientation of individual features to perform reliable verification.

Because most false matches involve very small image features, a minimum "size" for matches could be specified either for the entire matching process or within each model.

\footnote{Of course, the matching threshold can be raised to admit less precise matches.}
This limit would reflect both the size of the feature and the fact that, as an object moves away from the cameras and gets smaller, the likelihood correctly identifying corresponding features and deriving accurate information about it decreases.

Figure 6.1: (a) This shaded head illustrates the problem of missing segments: if the marked segment in the left-hand shape were "skipped," the two shapes would match very closely. (b) This problem with the shark head was mentioned in Section 5.5: if the two marked segments in the left shape were combined, the resulting single segment would yield a perfect match between the two shapes.

In addition to these relatively simple modifications, there are changes that would mandate substantial changes to the recognition system.

- The efficiency of matching could be increased by indexing the models and candidate matches. Stein ([Ste92]) uses such indexing for fast retrieval of models and to group clusters of related or consistent matches.

- With an appropriate indexing scheme, matching could proceed by examining each image segment or group of segments and retrieving only similar models which are likely to yield a match. This would be much more efficient and elegant than the current brute-force method of comparing each group of image segments to each model. In addition, this
would be similar to the human vision system, which can recognize similarities between objects and can provide at least a range of possible explanations given restricted knowledge or a partial view of an object.

- Instead of searching for features as though each one stands alone in the image, recognition could be guided by features already identified. If a fish tail is located in an image, the system can estimate where the corresponding head and, if desired, dorsal and ventral fins should be. These estimates could be used to influence which models are examined or which feature is chosen in the event of multiple matches.

- Using previous matches to guide recognition (discussed above) could be extended to a general model-fitting approach. This would also provide a more accurate means of verification. However, model-fitting would be difficult because of the shape differences between fish.

- Saund ([Sau92]) describes a shape representation designed to capture subtleties between similar shapes by detailing many geometric properties and spatial relationships. The representation is a hierarchical system in which low-level features are combined to create a vocabulary of abstract features. Features are indexed by location and scale on a “Scale-Space Blackboard” to expedite matching. Saund’s sample shape domain is the dorsal fins of different species of fish. Incorporation of such a representation would be necessary if the system were to distinguish between similar objects. The NIFE system could use it to discriminate grilse, salmon undergoing precocious maturation, from fish maturing normally; the ratio of grilse to “normal” fish is an important factor affecting harvest dates.

- Because the video apparatus provides continuous stereo footage, the system could incorporate motion sequences in its analysis. This could aid in feature recognition because real features would tend to retain their state of motion from image to image while spurious features and coincidental alignments would vanish over time. Analysis of motion sequences could also help interpret partially occluded objects. In addition, if the objects being recognized travel at known speeds, the camera-to-object distance can be calculated from the distance traveled between images given the apparent distance traveled between images (in
pixels). This would provide distance information from single images (i.e., without stereo). Unfortunately, because fish swim at different speeds, the distance to a fish would still be required to calculate the actual distance traveled between images and vice-versa.

A few final ideas relate to other aspects of the system.

- The accuracy of size measurements and disparities could be increased by integrating a better way to specify the locations of features. Currently, the location of a feature is determined by the centre of its super-segment. This does not necessarily correspond to the point on the feature which correctly delimits desired dimensions. Furthermore, if the feature is only partially matched, its centrepoint is shifted accordingly. A better method would be to specify the location of a feature in its model such that the system could properly locate it even in the event of a partial match. While on the subject of dimensions, it might also be advantageous to provide maximum and minimum dimensions of objects as yet another means of eliminating false matches.

- Several aspects of the entire process merit further study. First-off, we require more information about where the cameras should be positioned in the sea-cages in order to collect images which accurately reflect the fish population. During pre-processing, both the standard deviation ($\sigma$) of the Gaussian with which the image is first smoothed and the minimum contour length could have significant effects on the reliability and efficiency of the system. Many more parameters such as the matching and verification thresholds, the radius within which segments are considered connected and the permissible disparity range and discrepancy should also be examined to yield optimum results under different conditions. Most of these can be set upon execution.

- The stereo disambiguation stage could be expanded to become a powerful addition to the system. In addition, images and features could be used symmetrically during stereo analysis to improve reliability and ensure identical results regardless of the order in which images or features are examined. These modifications would likely necessitate more thorough examination of local features and better record-keeping of which features are available and which have been used.
• One last idea that may be somewhat far-fetched is the incorporation of intelligence in the linking stage. If it were possible to favour elongated contours or to guide the linking of intensity contours with descriptions of desired shapes, the linker may be able to extract more complete outlines or mark potential matches, thereby making subsequent recognition easier or faster.

• Finally, if the system were to be used commercially, the current user-hostile interface should be replaced with a snazzy graphical user interface.
Figure A.1: The geometry for binocular stereo imaging using two parallel cameras of focal length \( f \) separated horizontally by distance \( s \). We wish to determine the distance \( d \) from the focal plane to arbitrary point \( P \).

The diagram above shows the geometry used to determine distance from a stereo pair of images. The cameras have focal length \( f \) and are mounted on a horizontal plane with their optical axes parallel and separated by a known distance \( s \). With the cameras arranged thus, a pair of images taken simultaneously by the left and right cameras will have substantial overlap in the centre with points in the right image displaced to the left relative to corresponding points in the left image. This relative displacement is called disparity and it varies inversely
with desired distance \( d \), as shown below.

\[
x_l = f \cdot \tan(\alpha) \quad x_r = f \cdot \tan(\beta)
\]
\[
\tan(\alpha) = \frac{x + s/2}{d} \quad \tan(\beta) = \frac{x - s/2}{d}
\]  \hspace{1cm} \text{(A.1)}

\[
\text{disparity} = |x_l - x_r| = f \cdot |\tan(\alpha) - \tan(\beta)|
\]
\[
= \frac{f}{d} \left( x + \frac{s}{2} - (x - \frac{s}{2}) \right) \quad \text{(A.3)}
\]
\[
= \frac{f \cdot s}{d} \quad \text{(A.4)}
\]

Thus the distance \( d \) is given by

\[
d = \frac{f \cdot s}{|x_l - x_r|} \quad \text{(A.6)}
\]
\[
= \frac{f \cdot s}{\text{disparity}} \quad \text{(A.7)}
\]

This derivation applies when disparity is continuous, as when it is measured from stereo photographs. Because disparity takes on discrete values in digitized images, another constant is required which relates distance to the corresponding number of pixels. Thus, for digitized images, the distance \( d \) is given by

\[
d = \frac{c \cdot f \cdot s}{\text{disparity}}
\]

where \( c \) reflects the spatial resolution of the imaging hardware.

Note that:

- Disparity varies inversely with distance.

- This result is independent of the location of the point in the image.
Figure B.1: Geometry relating the actual length $l$ of parallel projection $AB$ to its apparent length $l_i$ in the image.

Diagram B.1 shows the projection of object $PB$ onto the image plane through a perfect lens of focal length $f$. Because $PB$ is oblique to the image plane, the lens images $AB$, the component of $PB$ lying parallel to the focal plane a distance $d$ in front of it.
From similar triangles,

\[
\frac{l}{l_i} = \frac{AB}{A'B'} = \frac{BC}{B'C} = \frac{d}{f} = \frac{d}{f} = \frac{d}{f}
\]

Thus, the actual length of the object in a continuous image is given by

\[
l = \frac{d \cdot l_i}{f}
\]

In a digitized image the length \( l_i \) is given as a discrete number of pixels, thus another constant is required relating length to number of pixels, yielding

\[
l = \frac{c \cdot d \cdot l_i}{f}
\]

for some constant \( c \) which is a parameter of the imaging hardware.

Note that:

- Apparent length in the image is directly proportional to the actual length of the parallel projection of the object.
- Apparent length is inversely proportional to the distance of the object from the camera.
- The focal length and pixel density are constant, leaving distance as the only unknown in the right hand side of the equation.
- Rather than a single constant \( c \), one will likely require a \( c_x \) for the horizontal direction and a \( c_y \) for the vertical direction because the aspect ratio of a pixel is seldom 1:1.
BIBLIOGRAPHY


