

**SYSTEM INTEGRATION AND IMAGE
PRE-PROCESSING FOR AN AUTOMATED,
REAL-TIME IDENTIFICATION AND MONITORING
SYSTEM FOR CORAL REEF FISH**

BY CHETAN TONDE

**A thesis submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Master of Science
Graduate Program in Electrical and Computer Engineering**

Written under the direction of

Dr. Joseph D. Wilder

and approved by

New Brunswick, New Jersey

October, 2010

ABSTRACT OF THE THESIS

System Integration and Image Pre-processing for an Automated, Real-Time Identification and Monitoring System for Coral Reef Fish

by Chetan Tonde

Thesis Director: Dr. Joseph D. Wilder

In this work we build an underwater vision system capable of monitoring the activities of fish found near coral reefs. We propose a unique hardware platform capable of monitoring a volume of water in a very efficient and cost effective way. We also develop algorithms required to take advantage of such a system. There are three main contributions of this work, which are; (1) using two right-angled camera's to capture underwater image sequences, (2) developing algorithms to track and pre-process images for recognition (3) and demonstrating that we can recognize fish families or in some cases exact fish species using fish shape (with size, color and pattern features to be added later). We conclude from this work that using just two cameras in a right-angled setup is a cheap and effective way of monitoring fish activities in general. It is cost effective when compared to using multiple cameras and also less computationally intensive. We developed and modified our approach based on observations we made while testing this setup and accommodated these modifications in our software. We installed this system at the artificial coral reef in the New York Aquarium and periodically collected image sequences for processing. We demonstrate our results on the collected sequences and show pre-processing results on them. We also demonstrate, using shape feature from a

fish sequence we collected at the aquarium (using cross-validation); that we can recognize fish families or in some cases exact species using those features.

Keywords: Background Modelling, Camera Calibration, Multi-Target Tracking, coral Reef, Fish Species, Shape Analysis, Recognition.

Acknowledgements

This project was funded by NSF grant NSF DBI-0649878.

Dedication

Dedicated to friends and family.

Table of Contents

Abstract	ii
Acknowledgements	iv
Dedication	v
List of Tables	ix
List of Figures	x
1. Introduction	1
1.1. Motivation	1
1.2. Goal	1
1.3. Problem Statement	2
1.4. Challenges	3
1.4.1. Target localisation	3
1.4.2. Non-rigidity of target (Self Occlusion)	4
1.4.3. Mutual occlusion	4
1.4.4. Tracking stability	4
1.4.5. Feature Selection	5
1.5. Thesis Outline	5
2. Previous Work	6
2.1. Small Aquarium Setup	6
2.2. Target localisation	6
2.3. Epipolar Geometry	7
3. System Instrumentation	9

3.1. System Design	9
3.2. Camera Calibration	10
4. Software Development	16
4.1. Software Design	16
4.2. Image Capture	16
4.3. Background Modelling	18
4.4. Tracking	21
4.4.1. Robust Visual Tracking	21
4.4.2. Updating tracking parameters online	23
4.4.3. Multi-Target Tracking	25
4.5. Pose Correction	26
4.6. Re-segmentation	28
4.7. Feature Selection and Recognition	29
4.7.1. Feature Selection	29
4.7.2. Recognition	30
4.7.3. PCA based shape descriptors	30
5. Results	38
5.1. Data Collection	38
5.2. Background Modelling	39
5.3. Tracking	43
5.4. Pose Correction and Scaling	44
5.5. Re-segmentation	45
5.6. Recognition	45
6. Conclusions and Future Work	78
6.1. Conclusions	78
6.2. Future Work	80
References	82

Appendix A. Lego Remote Focus Control System	85
Appendix B. Graphical User Interface-Screenshots	88
Appendix C. Fish Species Database	91
Appendix D. Confusion matrix for leave-one-out cross-validation	116

List of Tables

3.1. Specifications of Prosilica GigE camera.	13
3.2. Camera 1 intrinsic parameters	13
3.3. Camera 2 intrinsic parameters	13
4.1. Example of camera configuration shown for camera 2.	17
5.1. The parameter setting used for the sequences	44
5.2. Fish codes with common names and number of examples used for each species in the database	46
A.1. Two bit code used to send control information to the Lego system. . . .	85
D.1. Confusion matrix for best matched species (based on shape).	117

List of Figures

2.1. Small Aquarium with illumination at IMCS.	7
2.2. Images from camera 1 (a) and camera 2 (b).	8
3.1. Concept of the proposed setup at coral reef.	10
3.2. Proposed design of submersible system for viewing coral reef fish popu- lation.	11
3.3. Photo of the system installed at the New York Aquarium	12
3.4. Photo of the system installed at the New York Aquarium	12
3.5. Sample calibration images from camera 1 (3.5(a)-3.5(g)) and camera 2 (3.5(h)-3.5(n))	15
4.1. Overall software design with example intermediate results below every block.	32
4.2. Image capture system block diagram.	33
4.3. Graph constructed from image with source s and sink t . The cut (dotted line) corresponds to the maximum flow and minimum edge cut cost. . . .	33
4.4. Hidden Markov model with hidden states X_t and observation variables I_t	34
4.5. Flow chart showing management of trackers and how tracks are created, validated to display and then later deleted.	34
4.6. "Bodianus puchellus- Cuban hogfish" - Variations of color and patterns within species but having strong similarity in shape.	35
4.7. "Acanthurus coeruleus - Blue Tang" - Blue Tang with shape and color similarities with Doctor Fish (Figure 4.8).	36
4.8. "Acanthurus chirurgus - doctorfish" - Doctor Fish with shape and color similarities with Blue Tang (Figure 4.7).	37

5.1. Flowchart describing the "quick" background subtraction and sequence classification process.	40
5.2. An image pair from sequence 1 (05/25/2010 at 10:49:24 hrs) showing the "quick" background subtraction result.	41
5.3. An image pair from sequence 2 (05/24/2010 at 11:51:40 hrs) showing the "quick" background subtraction result with a whole fish in one view on the right and a partial segment of the same fish near the image border on left.	42
5.4. A "typical" image pair from sequence 1 (05/25/2010 at 10:49:24 hrs).	43
5.5. Background result for the "typical" image pair from sequence 1.	43
5.6. Background modelling result after post-processing for the "typical" image pair.	44
5.7. Background modelling results on sequence 1 (05/25/2010 at 10:49:24 hrs).	48
5.8. Background modelling results on sequence 1 (05/24/2010 at 11:51:40 hrs).	49
5.9. Tracking result on a "typical" scene.	50
5.10. Tracking results on sequence 1 (05/25/2010 at 10:49:24 hrs).	50
5.11. Tracking results on sequence 1 (05/24/2010 at 11:51:40 hrs).	51
5.12. Image pair with fish overlap.	51
5.13. Image pair after few frames without overlap.	52
5.14. Image pair with ellipses around each observed fish.	52
5.15. Matched fish in both views marked with a blue ellipse. Two horizontal lines across in the left image denote the search region.	52
5.16. Pose corrected and scaled fish.	53
5.17. Re-segmentation results on sequence 1 (05/25/2010 at 10:49:24 hrs).	53
5.18. Re-segmentation results on sequence 2 (05/24/2010 at 11:51:40 hrs).	54
5.19. Species grouped together which have strong similarity in shape.	54
5.20. Contour tracing on a example fish from the database. Figure 5.20(a) original, Figure 5.20(b) flipped left-right, 5.20(c) flipped up-down and 5.20(d) flipped left-right and up-down repectively.	55

5.21. Shapes of of species from 1-12 from the database.	56
5.22. Shapes of of species from 13-25 from the database.	57
5.23. Different Species found in top 5 matches using shape for each of the species 1-8.	68
5.24. Different Species found in top 5 matches using shape for each of the species 9-17.	69
5.25. Different Species found in top 5 matches using shape for each of the species 18-25.	70
5.26. Histogram of top 5 matches for each example of a species 1-8.	71
5.27. Histogram of top 5 matches for each example of a species 9-18.	72
5.28. Histogram of top 5 matches for each example of a species 19-25.	73
5.29. Similar shapes for an example fish of each species from the database (Species 1-12).	75
5.30. Similar shapes for an example fish of each species from the database (Species 13-25).	76
5.31. Test results for matching test fish image 1 with the database.	77
A.1. <u>Lego Remote Focus Control</u> block diagram.	86
A.2. <u>Control Input</u> block schematic.	87
B.1. <u>Screenshot-Camera Calibration</u>	88
B.2. <u>Screenshot-Image Capture</u>	89
B.3. <u>Screenshot-Background Modelling</u>	90
C.1. Species Code:1 Elacatinus randalli yellownose goby.	91
C.2. Species Code:2 Chaetodon ocellatus spotfin butterfly.	92
C.3. Species Code:3 Abudefduf saxatilis Sergeant major.	93
C.4. Species Code:4 Bodianus puchellus Cuban hogfish.	94
C.5. Species Code:5 Haemulon flavolineatum French grunt.	95
C.6. Species Code:6 Pseudupeneus maculatus spotted goatfish.	96
C.7. Species Code:7 Haemulon melanurum cottonwick grunt.	97
C.8. Species Code:8 Halichoeres bivittatus Slippery Dick.	98

C.9. Species Code:9	<i>Acanthurus chirurgus</i> doctorfish.	99
C.10.Species Code:10	<i>Acanthurus coeruleus</i> Blue Tang.	100
C.11.Species Code: 11	<i>Mulloidichthys martinicus</i> yellow goatfish.	101
C.12.Species Code:12	<i>Chaetodon capistratus</i> four-eyed butterfly.	102
C.13.Species Code:13	<i>Bodianus rufus</i> Spanish hogfish.	103
C.14.Species Code:14	<i>Anisotremus virginicus</i> porkfish.	104
C.15.Species Code:15	<i>Pomacanthus paru</i> French angelfish.	105
C.16.Species Code:16	<i>Holacanthus ciliaris</i> queen angel.	106
C.17.Species Code:17	<i>Haemulon sciurus</i> bluestriped grunt.	107
C.18.Species Code:18	<i>Selene vomer</i> lookdown.	108
C.19.Species Code:19	<i>Serranus tortugarum</i> chalk bass.	109
C.20.Species Code:20	<i>Chromis scotti</i> purple reefish.	110
C.21.Species Code:21	<i>Hypoplectrus indigo</i> indigo hamlet.	111
C.22.Species Code:22	<i>Chromis cyanea</i> blue chromis.	112
C.23.Species Code:23	<i>Chaetodon sedentarius</i> reef butterflyfish.	113
C.24.Species Code:24	<i>Chaetodon striatus</i> banded butterflyfish.	114
C.25.Species Code:25	<i>Halichoeres garnoti</i> yellowhead wrasse.	115

Chapter 1

Introduction

1.1 Motivation

The monitoring of fish populations is important for fish ecology studies and biological research. Coral reefs all over the world are dying and fish populations that live in and near them are threatened. Collecting information about the health of various species is important not only for knowing which species are in decline, but also as an indicator of the health of the coral reefs themselves and the extent to which environmental changes (eg. global warming and pollution) are affecting the entire marine environment. Traditional fish monitoring techniques rely on manual processing of raw videos gathered through devices called fish passes [8] constructed in rivers through which migratory fish pass. Where this is not possible, fish must be tagged or captured, measured and counted manually. This method is invasive and not preferable. Vision systems perform very well doing jobs involving non-contact based tracking, recognition and counting. The focus of this project sponsored by the National Science Foundation (NSF #*DBI* – 0649878), is to develop such a vision system for monitoring fish population.

1.2 Goal

The goal of this project is to develop an underwater multi-camera vision system to identify and count fish species found near coral reefs, observe them over time, and collect data leading to understanding of the effects of environmental changes on those populations.

Project Timeline:

1. Year 1 (Small aquarium): Develop and test system hardware and software in a custom built aquarium containing a variety of reef fish. The initial phase was carried out at the Rutgers Institute for Marine and Coastal Sciences(IMCS).
2. Year 2 (New York Aquarium): Develop and test a completely submersible version of the system, install it in the New York Aquarium, test the hardware and software setup and prepare for further experiments on coral reefs off the coast of Belize.
3. Year 3 (Coral Reef): Install and test the prototype system at the Glover's reef Marine Research Station in Belize. Data on fish species common to this reef will be collected and analysed to demonstrate the feasibility of using this type of system to monitor reef fish populations.

1.3 Problem Statement

The main tasks to be accomplished are:

- Develop a robust hardware platform with optics, illumination and image sensing appropriate to the underwater reef environment.
- Develop a background subtraction algorithm for a dynamic, underwater environment that will enable reliable segmentation of the fish from their background while distinguishing true fish images from floating debris.
- Using multiple cameras, correct for scale and pose variations for free swimming fish.
- Extract robust features from pose-corrected images of the fish, using size, shape, color and texture.
- Develop robust species classification algorithms combining models based on prior knowledge with in-situ training techniques.

The contributions to the project presented in this thesis are:

- Developing system hardware and software for underwater data capture.

- Developing software for analysis of captured sequences.
- Developing a robust tracking algorithm for tracking fish.
- Local re-segmentation to obtain fine details of fish shape.
- Feature selection.
- Classification of a few select fish species
- Developing a user-interface (GUI) for the system.

1.4 Challenges

Various techniques have been developed for robust detection, tracking and recognition of non-rigid objects. But most of the work done is with regards to tracking and recognition non-rigid objects like shoes [14], faces [19] or articulated objects like people [16] and hand gestures. Very few techniques have been developed before for detecting, tracking and recognizing non-rigid objects like fish which have appearance variation within the same species and extreme pose variations. Those which exist avoid these challenges by constraining the pose or appearance variations manually [8, 9]. We present in this section the following major challenges: target detection and representation, non-rigidity, mutual occlusion, tracking stability and feature selection.

1.4.1 Target localisation

Detecting the presence of objects has been an area of extensive research in computer vision. Vision systems have been developed to detect objects based on prior view based appearance models [13], or detecting object features like Shift-Invariant Feature Tracking (SIFT) [10], haar-like features [11] and so on. These techniques mostly rely on the different view-based appearance models of the object and are hard to generalize for detecting a particular class/species of objects, where there is considerable interclass variation. The matching search space grows exponentially with the number of views and articulations of the object/target. Techniques based on silhouettes or edge frames structures [12] are useful but not reliable as they consider only object structure which

varies greatly with the object appearance and pose. Detecting motion in a scene using grayscale/color changes or observing motion of pixels using optical flow and then eliminating non-target objects based on shape, size and appearance criteria is the best approach in such scenarios.

1.4.2 Non-rigidity of target (Self Occlusion)

Non-rigid objects are difficult to detect and track. Research has been done to detect them from a known background using background modeling [6, 2] and to track them as they move [3]. Non-rigid articulations make it difficult to track the target in a single view. Moreover, using multiple views requires complete 3D-reconstruction and 3D modeling to track them accurately and makes the problem computationally intensive and intractable for real-time applications. In our scenario where our goal is to just recognize the target species, it is good enough to not lose track of them as they move, and attempt to recognize them from the best possible view, thereby avoiding multiple counts.

1.4.3 Mutual occlusion

Tracking multiple fish simultaneously poses unique challenges of its own. Due to occlusion caused by multiple fish which could be of the same or different species, we could lose track of one of them or swap them. These problems can be avoided to some extent by having an appearance model of the fish and incorporating motion information simultaneously in the tracker [17]. This approach makes it robust to occlusions but makes the approach sensitive to appearance variations and constrained to only first order linear motion. The problem can be solved to a great extent by incorporating multiple view information into the system to determine occlusion based on location of same fish in other views which may not always be the case.

1.4.4 Tracking stability

Tracking fish species using rectangular bounding boxes is difficult, as it is hard to fit a bounding box accurately and exactly to the fish shape, which could be of any form.

Tracking using arbitrary active contours [4] has been used before to track non-rigid shape objects, but using this approach is computationally expensive and fails to handle occlusions (mutual and self). Having a bounding box-based tracker is fast but some of the background may be added to the appearance model, the box may drift relative to the shape over time. The problem remains even with the use of ellipse or any other simple shape as the fish shape could be arbitrary. Using a bounding box for tracking the object roughly tracking the object and scaling the bounding box bigger after each frame adaptively to account for drift is the best approach. Further, to obtain the exact object boundary a local, detailed segmentation within the bounding box can be done.

1.4.5 Feature Selection

Recognizing fish species from multiple view images is a challenging problem. Biologically, fish species are recognized by locating unique patterns, overall shape and color. Identifying and locating such patterns from 2D images is a challenging problem. Also, appearance and color variations within the same species makes it impossible to generalize such features. These also vary significantly as the fish grows from a juvenile to an adult.

1.5 Thesis Outline

The outline of the thesis is as follows. Chapter 1 is the introduction. chapter 2 describes the previous work done in the project. Chapters 3 and 4 describe the overall hardware system design and software algorithms respectively. Chapter 5 reports the system test results. Finally, chapter 6 presents the conclusion discussion and future work.

Chapter 2

Previous Work

In this chapter we describe previous work completed during the Year 1 phase of the project.

2.1 Small Aquarium Setup

Previous work done in Phase I of the project evaluated the feasibility of the proposed fish monitoring system on a small scale in a small fish tank as shown in figure 2.1. The work involved setting up two cameras at right angles from the front and right side of the tank, with the left and back side painted black, providing a uniform black background. The two right angled cameras were positioned at the same height with reference to ground level which captured a target fish at the same height in both cameras. Cameras were calibrated using a 2D checkerboard pattern and the MATLAB camera calibration toolbox [1]. To simulate sunlight conditions found during daylight hours on a coral reef four special fluorescent fixtures were added to approximate the color temperature of sunlight. This initial setup was used to capture image sequences with three different target species (Figure 2.2).

A flexible image capture system was developed for capturing high resolution (1024 x 1360) images simultaneously from both cameras for further processing.

2.2 Target localisation

Target detection and localization can be done by background modeling of the visual scene. Since we have an almost fixed background we train a background image using a fraction (50 %) of frames from a sequence of 300. Using a technique described in

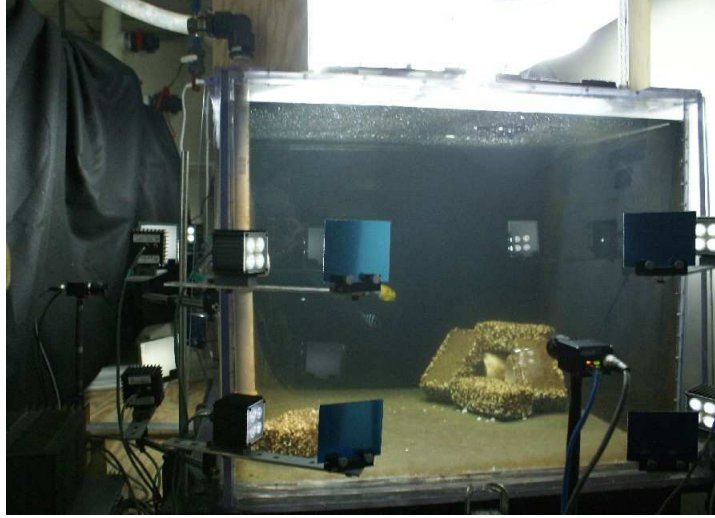


Figure 2.1: Small Aquarium with illumination at IMCS.

section 4.3 we estimate a background model for both cameras and foreground objects are detected based on values of pixel probability of being foreground.

2.3 Epipolar Geometry

Since, both cameras are right angled to each other we could use depth information to estimate fish 3D location and size. This information can be used to detect overlapping fish in one view and mark them for tracking and recognition analysis. Both the cameras are calibrated using 2D calibration described in [18]. Using epipolar geometry we can establish correspondences between target blobs from both views and perform pose correction and scaling. These techniques are described in section 4.5.



(a) Camera 1



(b) Camera 2

Figure 2.2: Images from camera 1 (a) and camera 2 (b).

Chapter 3

System Instrumentation

In this chapter we describe the system hardware design for the fully submersible system that has been deployed at the New York Aquarium and will be installed off the Glover’s reef research station in Belize. We also describe the system design issues and camera calibration.

3.1 System Design

The fully submersible system was designed to support two cameras underwater adjacent to a coral reef. The concept diagram of the system is shown in figure 3.1. The system mainly consists of two cameras in underwater housings at right angles to each other. The camera housings were provided by Ocean Presence Technologies , a company with extensive experience in supplying underwater imaging systems. The system was constructed by Piotra Nawrot of the Rutgers Institute of Marine and Coastal Sciences (IMCS). He also made significant contributions to the design of the system. The mechanical design of the system is shown in figure 3.2 and a photo of the system is shown on figures 3.3 and 3.4. The system mainly consists of four parts.

1. Two right angled cameras with underwater housing
2. Two black panels facing each camera to provide uniform background
3. Four led light arrays, one on each side of each camera.
4. A support structure that consists of a frame that holds all of the components and tilt bases for the camera housings.

The complete structure was made of light weight PVC material and pipes. The cameras are placed exactly at the center of adjacent faces so that the optical axes of the cameras are normal to their front facing panels, and intersect at the center of viewing volume. Adjustments are made using a laser-level in the final setup. The completed fully submersible system was tested in a large outdoor tank here at Rutgers before installing the system at the New York Aquarium. This enabled us to look for and correct any system issues before installing the system in a public facility.

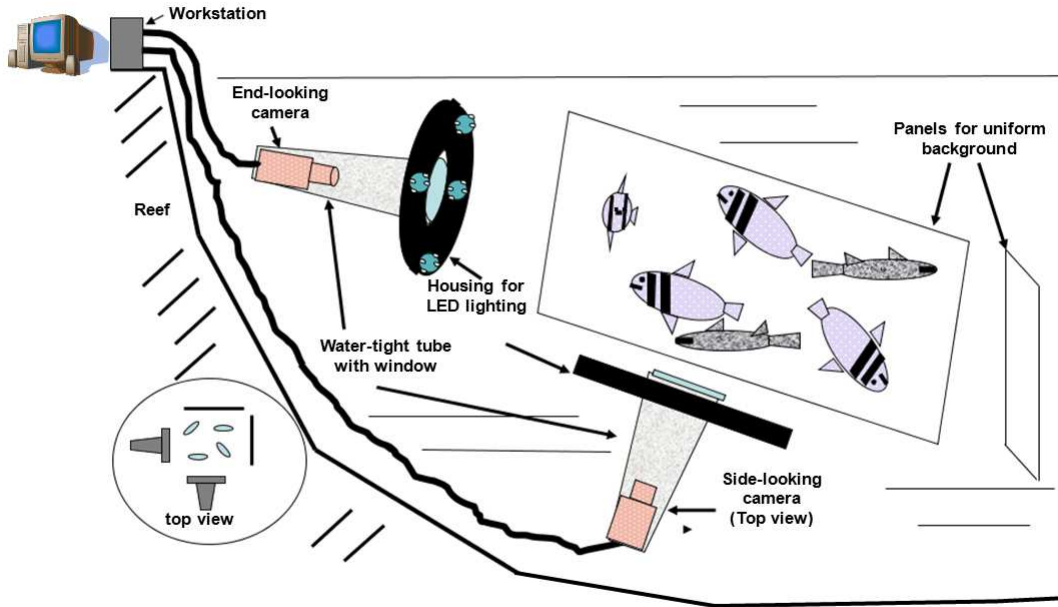


Figure 3.1: Concept of the proposed setup at coral reef.

3.2 Camera Calibration

Two prosilica GC series GigE cameras were used for image capture. These cameras are designed to transmit data over very large distances, have programmable frame rates upto 20 frames/sec, capture programmable regions of interest and have super pixel capabilities. Specifications are given in Table 3.1.

To know the complete geometry of the system the intrinsic and extrinsic parameters

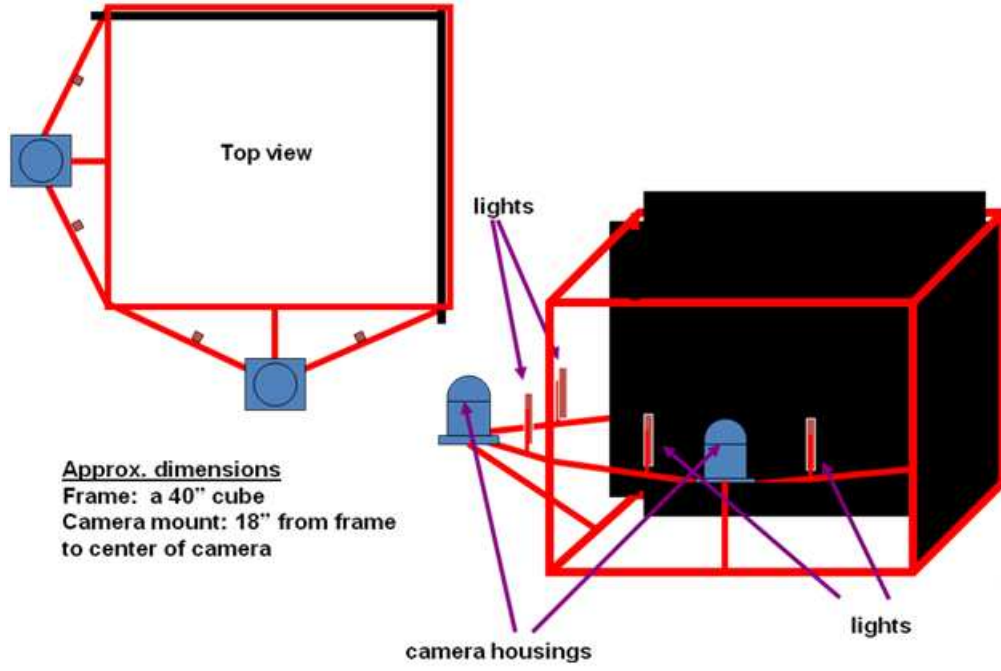


Figure 3.2: Proposed design of submersible system for viewing coral reef fish population.

of the cameras should be known. The intrinsic parameters of the camera are fixed and depend on the design of the camera sensor and focus system. The intrinsic parameters consist of the focal length¹ (f_x, f_y) , principal point (u_0, v_0) , skew co-efficient α_c ² and camera distortion (k_c) ³. They can be represented by a 3×3 matrix called the intrinsic matrix K shown below.

$$K = \begin{bmatrix} f_x & \alpha_c f_x & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The extrinsic parameters of the camera characterise the 3D location of the camera with respect to the world reference frame. The location of the camera with respect to some arbitrary fixed co-ordinate axis if specified by the camera rotation R and

¹The focal length f_x and f_y is in pixels and adjusted for the pixel size in x and y directions respectively.

² α_c is the skew angle between x and y directions of any pixel.

³The camera distortion is a 5×1 matrix and we currently ignore it in our setup, as we are mostly concerned with central part of the image, for which distortion is small



Figure 3.3: Photo of the system installed at the New York Aquarium

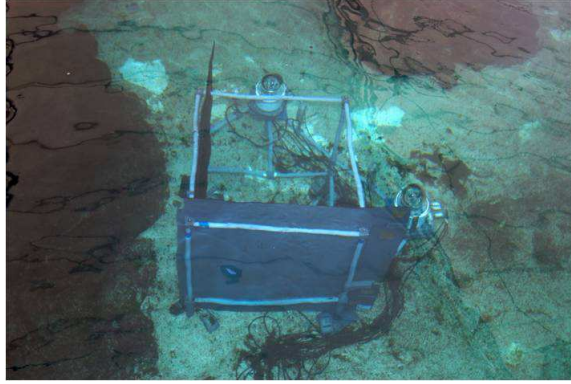


Figure 3.4: Photo of the system installed at the New York Aquarium

translation matrix t . These two matrices together form a 3×4 matrix $[R|t]$ called the extrinsic matrix. These two matrices together describe the complete image formation process from a 3D point in the world to a pixel location on the image. For our system the cameras were calibrated using checkerboard images taken underwater (See figure 3.5). Calibration data is given for both the cameras in Tables 3.2 ,3.3.

Intrinsic matrices of both the camera denoted by K_1 and K_2 are given in equations 3.1 and 3.2). Extrinsic matrices rotation R and translation t are given in equations 3.3 (Note: R and t are rotation and translation of camera 1 with respect to camera 2).

$$K_1 = \begin{bmatrix} 1517.8 & 0 & 704.0 \\ 0 & 1498.6 & 408.1 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Prosilica	GC1380
Resolution	1360 x 1024
Max frame rate at full resolution	20 fps
Type	CCD Progressive
Interface	IEEE 802.3 1000baseT
Output	8/12 bit
Sensor Size	Type 2/3
Sensor	Sony ICX285
Cell size	6.45 μ m
On-board FIFO	16 MB
Body Dimensions (L x W x H in mm)	$33 \times 46 \times 59$ including connectors, w/o tripod and lens

Table 3.1: Specifications of Prosilica GigE camera.

Table 3.2: Camera 1 intrinsic parameters

f_x	1517.8
f_y	1498.6
α_c (Skew)	0
k_c (Distortions)	$[-0.2643, -8.5052, 0.0084, -0.0047, 0]'$
u_0	704.0168
v_0	408.1469

Table 3.3: Camera 2 intrinsic parameters

f_x	1486.8
f_y	1472.7
α_c (Skew)	0
k_c (Distortions)	$[-0.3141, -0.8938, 0.0073, 0.0174, 0]'$
u_0	717.4240
v_0	390.6831

$$K_2 = \begin{bmatrix} 1486.8 & 0 & 717.4 \\ 0 & 1472.7 & 390.7 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$R = \begin{bmatrix} 0.7910 & -0.0652 & 0.6083 \\ 0.0731 & 0.9973 & 0.0118 \\ -0.6074 & 0.0351 & 0.7936 \end{bmatrix} \quad (3.3)$$

$$t = \begin{bmatrix} -496.0733 \\ 104.0548 \\ 526.4208 \end{bmatrix} \quad (3.4)$$

The cameras were underwater in a completely sealed enclosure with no controlled focus or aperture control. To adjust for sharp images with good contrast, both of these parameters needed to be fixed to gain best image quality. Changing the aperture and focus manually and testing it could have been a very tedious and challenging process. Hence, for this we developed a temporary focus control system using Lego-Mindstorms Robotics Invention System 2.0 [20]. The Lego[®] motors were interfaced with the cameras general purpose IO pins (GPIO) to send 2- bit code to turn the motors clockwise, anti-clockwise and stop. Using this tool focus of both the cameras was set appropriately, in a small underwater tank with the largest aperture. The aperture then was later adjusted manually for enough light (at f/2.5 f-stops) (See Appendix A for further details).

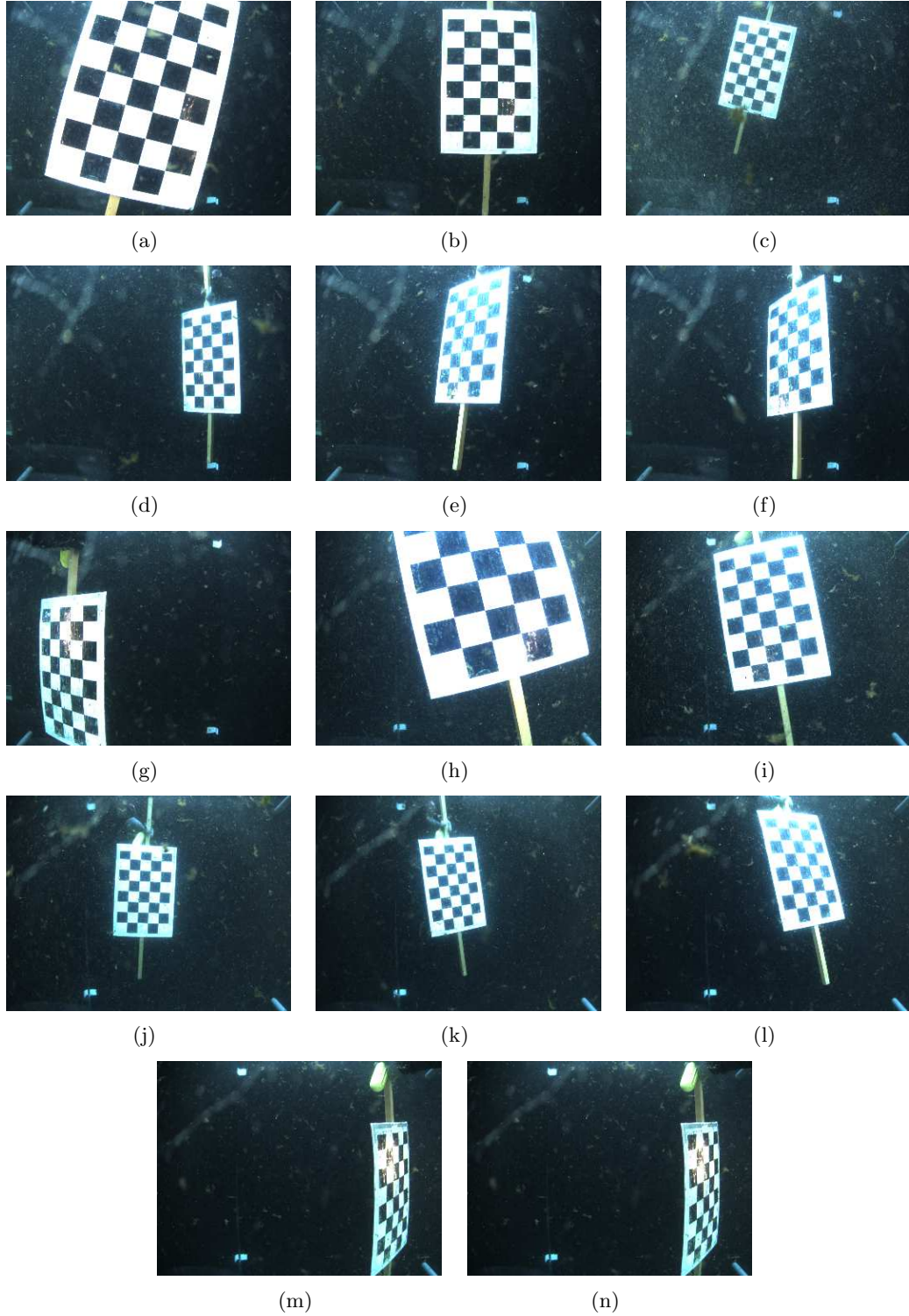


Figure 3.5: Sample calibration images from camera 1 (3.5(a)-3.5(g)) and camera 2 (3.5(h)-3.5(n)). These sample images of checkerboard at different distances and orientations were used in conjunction with the MATLAB Camera Calibration Toolbox [1], which estimates the camera parameters from the known size checkerboard squares and their projected images.

Chapter 4

Software Development

4.1 Software Design

Figure 4.1 shows the overall pre-processing flow chart beginning (Image Capture) till the end Recognition. The following paragraphs trace the sections of flowchart and describes them in detail.

4.2 Image Capture

A flexible image capture system was developed that captures simultaneous sequences of image pairs from the two cameras. We have two types of image sequences, i.e., those used for training images and those used for testing. The training sequences allow us to model the background from which we are extracting the fish images. These sequences are required on a regular basis, since illumination conditions and artefacts in the water are always changing. The testing sequences allow us to locate the fish in the 3D volume of water being imaged, correct their pose, extract features for classification and finally to classify and count the fish swimming through the volume under inspection. The length of the sequences N (e.g. 200 image pairs) and the frame rates (e.g. 20, 15, 10, 5 frames/second) can be chosen under computer control. Special attention was paid to maintaining the timing between corresponding members of each image pair throughout the duration of the image sequences. This flexibility of duration and frame rate is required for the following reasons.

- in order to model the background we need enough training image pairs taken over a long enough period of time to create a reliable model of the background while fish are swimming through the volume of water being imaged

Camera 2	169.254.200.2 (GC1380C)
AcquisitionMode	Continuous
FrameStartTriggerMode	FixedRate
GainMode	Manual
FrameRate	5.000000
SyncOutMode2	GPO

Table 4.1: Example of camera configuration shown for camera 2.

- having learned the background, we need a sufficient number of test image pairs to disambiguate overlapping fish, i.e. detect the presence of overlapping fish in one or both images of an image pair and continue tracking the fish until they are no longer overlapping.

As we learn more about the behaviour of the reef fish we are studying, we will be able to settle on the sequence lengths and frame rates we will need for both background modelling and fish classification. Some details of the image capture setup are given below.

The block diagram below (Figure 4.2) shows the flow involved in the capture of image sequences using Prosilica GigE interface cameras. Two independent threads configure each camera independently and simultaneously transfer images from the interface to internal RAM which subsequently is written to disk.

Prosilica cameras support fixed frame rate capture and both cameras are configured with fixed frame rate f ($= 5, 10, 15$ or 20 fps) and the capturing is triggered simultaneously. After the triggering, both cameras start capturing the frames independently at their fixed frame rates. Capture time, frame rate and image pair time delay are computed using the computer's internal system clock up to an accuracy of ± 1 ms. The captured frames from the cameras are transferred to the GigE interface which stacks up to 100 frames in a queue. Images from the queue are simultaneously transferred from the queue to internal RAM storage. All the captured images corresponding to each camera are then written to a secondary storage, i.e. hard disk, for training and testing purposes. A sample configuration of one of the cameras is given in the table 4.1.

4.3 Background Modelling

Target species form the foreground against a fixed background. By modelling this background we can extract target species which are used for further processing. The foreground represents the fish species we are interested in tracking and identifying.

We explored and tested three different approaches to background modelling. The first approach, was pixel-based, i.e. it only considered the history of individual pixels in the image without regard to what is happening to surrounding pixels. It modelled the individual pixels within an image as a weighted combination of a K (usually 3 to 5) mixture of Gaussian distributions, by Stauffer and Grimson [6]. This approach turned out to be sensitive to outliers from the recent past. It also had trouble distinguishing pixels on a fish with colors and intensities similar to that of the background, e.g. black vertical stripes on a fish could not be distinguished from the black background panels and, therefore, it was difficult to separate a complete fish image from its background.

The second approach follows a method proposed by Elgammal, et al [2], which use Kernel Density Estimation(KDE) over n recent values of a pixel. It uses normal kernel smoothing over the histogram of the n most recent values of a pixel along with a neighbourhood and a connected component based conditional probability model. The approach consists of three stages; pixel displacement probability computation, connected displacement probability computation, and shadow removal. Each of which is explained below,

- Pixel displacement probability: The pixel displacement probability is defined based on the following equation.

$$P_N = \max_{y \in N(x)} P(x_t | B_y) \quad (4.1)$$

It is the maximum probability that the observed value, x_t , belongs to the background distribution of some point in the neighbourhood $N(x)$ of x , where B_y is the background sample for pixel y . The probability estimation is carried out by using a normal kernel density estimation window given by,

$$P(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2} \frac{(x_i - x_j)^2}{\sigma_j^2}} \quad (4.2)$$

The kernel bandwidth chosen here is diagonal with each σ_j corresponding to independent color channel variance. The variance is chosen as $\sigma_j = \frac{m}{(0.68\sqrt{2})}$ where m = median of absolute difference values. This helps the model suppress small motions in the background but also causes more false detections when foreground colors are similar to the background.

- Connected displacement probability: To suppress false detection, an additional connected component-based conditional probability is computed. It is the probability that the pixel moved is from a component C from a nearby location given by,

$$P_C = \prod_{x \in C} P_N(x) \quad (4.3)$$

This probability will be small if the component displaced from the background is from a real target in the foreground.

Combining $P_N(x)$ and $P_C(x)$ we get the result that a pixel is background if and only if $(P_N(x) > th_1) \& (P_C(x) > th_2)$ is true. Where th_1 and th_2 are appropriately adjusted thresholds. The background model is then successively repeated either selectively, based on change if there are pixel values in the derived mask, or blindly. Long term or short term models are considered for model updating.

- Shadow removal: The probabilities based on $P_N(x)$ and $P_C(x)$ do not handle shadow removal very well and they need to be treated separately. For this purpose chromaticity co-ordinates are used.

$$\begin{aligned} r &= \frac{R}{(R+G+B)} \\ g &= \frac{G}{(R+G+B)} \\ s &= R + G + B \end{aligned}$$

Considering a subset B of $\langle r_i, g_i, b_i \rangle$ defined as $B = \{x_i \in A \text{ and } \alpha \leq \frac{s_i}{s_i} = \beta\}$,

density estimation over the (r, g) color space and simple thresholding removes shadows while still detecting the object.

Although this algorithm performed better than the Stauffer and Grimson [6] pixel based algorithm, it still had some difficulty in creating complete binary masks when portions of a fish image were similar in color and intensity to that of the background. The third approach involved use of the Graph Cut Algorithm [7]. It is an energy minimization method to solve Markov Random Field (MRF) problems in computer vision.

The figure 4.3 shows a graph whose nodes are connected by weighted edges, and also connected to the source s and the sink t . We are trying to find a minimum cut to divide the graph into source s and sink t . Ideally, a cut through the least weighted edges (as shown in the graph) gives the bipartition of the graph with minimum cut. In our application, weights between each pixel to the source are probabilities to be background, edge weights between pixels are given by color similarities.

The energy we are trying to minimize is in the form of

$$E(f) = \sum_p (D_p(f_p)) + \sum_{p,q \in N} V_{(p,q)}(f_p, f_q) \quad (4.4)$$

Where f_p is the labeling at a given node, $D_p(f_p)$ assigns a data term for the given labeling, $V_{(p,q)}(f_p, f_q)$ is the smoothness term between neighbouring pixels p and q .

The data term comes from the results from background subtraction using Kernel Density Estimation (KDE), and the smoothness term comes from the pixel similarities between neighbours.

The pixel-based background subtraction algorithms did not take the spatial relationship between neighbouring pixels into account. The graph cut algorithm with the smoothness energy solves this problem by assigning neighbouring pixels with similar colors to be the same (either foreground or background). Hence, we use this approach to detect foreground moving objects in the scene.

4.4 Tracking

To account for single a fish moving across multiple frames the fish needs to be tracked as long as it is in the field of view (FOV). For this we develop an algorithm which allows us to track fish across frames and also helps us detect overlap between multiple fish. For this we further use a technique developed by David Ross, et. al in [17] and extend it to detect and handle such scenarios.

4.4.1 Robust Visual Tracking

The approach is based on an appearance based tracker which uses an eigenbasis representation of the target image and tracks it in the affine space using a sequential inference model (particle filter) ¹

The approach as described in [17] is as follows:

1. Locate the target object in the first frame by using Background Modelling technique as described in section 4.3 and represent it as a single particle $X_t = [x_t, y_t, \theta_t, s_t, a, \phi_t]$ which are the affine parameters of location of the bounding box. These are defined as follows,

$$x_t = \text{x translation}$$

$$y_t = \text{y translation}$$

$$\theta_t = \text{rotation angle}$$

$$s_t = \text{scale}$$

$$\alpha_t = \text{aspect ratio}$$

$$\phi_t = \text{skew direction}$$

¹A particle filter also known as sequential Monte Carlo methods is a model estimation technique based on simulation. They perform model estimation by sampling particles i.e. sample points from the observation space. They are an alternative to Extended Kalman Filter(EKF) or Unscented Kalman Filter(UKF), with sufficient number of samples, they approach the Bayesian optimal estimate, and can be made more accurate than EKF or UKF.

2. Initialize the eigenbasis U to be empty, and mean μ to be the appearance model² of the target in the first frame. The observation number is set $n = 1$.
3. Move to the next frame and draw particles from the particle filter according to the dynamical model as below.

$$p(X_t|X_{t-1}) = N(X_t; X_{t-1}, \Psi)$$

Where, Ψ is the diagonal covariance matrix of affine parameters whose diagonal elements are the corresponding variances of the affine parameters i.e. $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$.

4. Then for each particle, extract the corresponding image patch (i.e. bounding box around the object) say I_t from the current frame and calculate its likelihood $p(I_t|X_t)$ under the current observation model, which is given by,

$$p(I_t|X_t) = p_{d_t}(I_t|X_t)p_{d_w}(I_t|X_t) \quad (4.5)$$

$$= N(I_t; \mu, UU^T + \epsilon I)N(I_t; \mu, U \sum -^2 U^T) \quad (4.6)$$

Here, μ corresponds to the center of subspace, ϵI is the additive Gaussian noise in the model and \sum corresponds to singular values of columns of U . The above equation can be interpreted as the probability of a sample being generated is proportional to the distance d from a point in the subspace to the reference point (i.e. μ) of the subspace, which is a product of distance-to-subspace, d_t , and distance within-subspace from a projected point to subspace center d_w .

5. Store the image patch (i.e. bounding box) corresponding to the most likely particle. Update the eigenbasis model when the desired number of new images (e. g. $m = 5$) are obtained with a forgetting factor $f = 0.3$. Updating the mean image is done using equation 4.7,

$$I_C = \frac{fn}{(fn + m)}I_A + \frac{m}{(fn + m)}I_B \quad (4.7)$$

²An appearance model of an object is a representation of an object either directly as an image or as mean image with alongwith the eigen representation using multiple images. Both of these models(image or mean image+eigen representation) are equivalent but the later one is more robust to noise and hence used in our case.

I_C , I_A , and I_B are the new, old and the incremental mean images respectively.

6. Go to step 3 and repeat.

The above approach is able to cope with partial occlusions, gradual pose variations and illumination changes.

In this approach the visual tracking problem is cast as an inference task on a Markov model with hidden state variable X_t , which describes the affine motion parameters of the target (See figure 4.4). The distribution of the target over an object's location is modelled by using a variant of the condensation algorithm³ [25].

Since, particle filters are used to approximate the posterior distribution of $p(X_t|X_{t-1})$, there is tradeoff between number of particles N needed to be drawn and accuracy of the modelled distribution. Using fewer particles reduces the computational load but degrades tracking performance and vice versa. The number of particles required to model the distribution need to be set adaptively, as motion of the target varies with time from frame to frame. Also, the variances of the affine parameters which are assumed to be constant need to be adjusted dynamically to account for variation in motion as the fish moves during an image sequence.

4.4.2 Updating tracking parameters online

To update the parameters online adaptively, we need a measure of tracking accuracy which can be computed online as the tracker moves. To have a measure of tracking accuracy, we define a quantity σ_{err} (over the past k frames) as below.

$$\sigma_{err} = med_i |e_i - mean_i(e_i)| \quad i = t, t-1, \dots, t-k-1 \quad (4.8)$$

Where $e_i = \sum_{\text{over all pixels } i} |I_C - I_t|$, is the residual error between the mean image I_C and observed image I_t at time t (e.g. $k=10$). Using the above measure for the accuracy of the tracker the tracking parameters are adjusted adaptively using the following

³Condensation algorithm is particle filter which uses sequential (i.e recursive) version of importance sampling (i.e. weighted sampling) to sample particles for model estimation. It has been successfully used to track contour of objects in clutter.

equations.

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_\theta \\ \sigma_s \\ \sigma_\alpha \\ \sigma_\phi \end{bmatrix}^{new} = k \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_\theta \\ \sigma_s \\ \sigma_\alpha \\ \sigma_\phi \end{bmatrix}^{old} \quad (4.9)$$

where, (4.10)

$$k = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.001 \\ 2 * 10^{-7} \\ 2 * 10^{-7} \\ 0 \end{bmatrix}^T \quad \text{a constant} \quad (4.11)$$

$$N^{new} = 400 + 1000\sigma_{err}. \quad (4.12)$$

In the equations above $\sigma_x, \sigma_y, \sigma_\theta, \sigma_s, \sigma_\alpha, \sigma_\phi$ are the affine variances of the tracker and left-hand side corresponds to the new values for the next frame while the right-hand side are the old values from the previous frame. N^{new} is the number of particles used to for the next frame in the tracker. In general we use 400 particles which gives decent tracking for normal fish motion, where $\sigma_{err} \approx 0$ and as the motion varies rapidly from normal more particles are used proportional to σ_{err} , we found 1000 to be an appropriate proportionality constant to account for these rapid changes.

We can see from the above equations that, as the tracker drifts the residual error e_i increases, successively causing increase in variances of the affine parameters thus increasing the sample space for the drawn particles and also increasing in the number of particles used, which then causes the tracker to latch back on to the object.

4.4.3 Multi-Target Tracking

In order to incorporate tracking multiple targets simultaneously, we need to run many such trackers simultaneously. For this we need to assign new blob observations to existing trackers and/or create new ones for newly appearing blobs. We pose this problem as a linear assignment problem and assign blobs to tracks based on a minimum cost assignment rule [26]. The cost of assignment of blob B to track T is defined as follows,

$$C(B \rightarrow T) = \begin{cases} (\Delta x^2 + \Delta y^2)/r + \text{corr}(\text{hist}(I), \text{hist}(T)) & \text{if } \Delta x < c_x \text{ and } \Delta y < c_y, \\ \infty & \text{otherwise,} \end{cases} \quad (4.13)$$

$\text{hist}(I)$ = normalised grayscale histogram of the image,

$\text{hist}(T)$ = normalised grayscale histogram of the mean image.

As we advance to the next frame we construct a matrix of number of tracks (n_t) as rows and number of blobs (n_b) as columns and compute the cost of each such assignment. We minimize the overall cost by trying every possible combination of assignments and then selecting the one with the least cost. For the newly detected blobs which are not assigned to any of the existing blobs new trackers are created for them.

To avoid false detections due to false positives from the Background Modelling stage we set a threshold of a minimum three successive detections for a track to be considered as a valid track. Once a track is deemed valid the tracker keeps track of the target but if, later on, a target is missing for more than 7 times in the past 10 frames it is declared as missing and the corresponding track is deleted.

Since we know the location and size of the trackers we can detect overlapping trackers in an image and hence avoid updating of their model during overlap.

A flowchart shown in figure 4.5 summarises this approach.

A similar tracking analysis is performed in the other view of the second camera and corresponding trackers are matched from both view by utilising camera geometry and then pose correction is performed on the target fish.

4.5 Pose Correction

⁴ Two cameras oriented perpendicular to each other will not provide true 3D stereo images of all fish, in at all possible orientations. They will, however, provide approximate location of the objects and provide enough information for pose correction as well as disambiguating overlapping fish. Our goal is to provide pose and scale corrected 2D images of the fish to be classified, matching their appearance with known side views in a database of image species. In order to begin segmentation and pose correction it is imperative to have a clean binary mask from the background subtraction step. Once an acceptable mask is acquired, it is then operated on by two simple sobel (3×3 size) convolution kernels to extract the edges of the masks. Since the data being operated on is a bitmask the edge map formed, E , will be an ideal single pixel outline of the area in question. A seed mask is then formed such that:

$$S_1(i, j) = i(\text{mod } 13).j(\text{mod } 13)$$

Where $i = [1 : R]$ and $j = [1 : C]$, R and C being the number of pixels in the Y and X direction with origin at top-left, respectively. For all $S_1 = 1$, the following criteria is then enforced using an 11×11 window for each matching pixel $S_1(j, k)$

$$S_1(j, k) = \begin{cases} 1 & \text{if } \sum_{n=-5}^5 \sum_{m=-5}^5 E(j+n, k+m) = 121, \\ 0 & \text{otherwise} \end{cases} \quad (4.14)$$

Since the edge map is singular in all areas except where edges are not present and -1 otherwise, the above restriction removes points in the S_1 that lay too close to an edge within an 11×11 window. This process forms n islands in S_1 , which are then labeled in a set N from $[1, n]$ corresponding to each individual fish present in the initial binary mask. For each label, the horizontal and vertical centroids of the blobs are calculated. Once again, because the masks are binary images, it becomes trivial to extract the edge points of the labeled bit masks, assuming B represents the labeled binary mask, finding $B \cdot E = n$, where $n \in N$ will create a set of points I_n on the edge

⁴This work was done by Lev Barinov and Jigesh Baxi in Phase II of the project

of the labeled segment for each segment of N . Knowing C_n as well as both the intrinsic and extrinsic parameters of the camera system, the corresponding epipolar line Δ_2^n in the second image can be drawn. Since the following is true for a camera system:

$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}^T F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0 \quad (4.15)$$

Where F is called the fundamental matrix of the camera system and (u_1, v_1) and (u_2, v_2) are the pixel co-ordinates of a point in the first and second image plane respectively. We can expand the formulation in terms of the parameters of the second plane:

$$(F_{11}u_1 + F_{12}v_2 + F_{13}) * u_2 + (F_{21}u_1 + F_{22}v_2 + F_{23}) * v_2 + (F_{31}u_1 + F_{32}v_2 + F_{33}) = 0 \quad (4.16)$$

Because of the imposed restriction of having a zero y translation between the two cameras (same vertical height), another set of horizontal lines can be drawn which pass through equal y values in both image planes at values $\pm w$ above and below the vertical centroid. They are defined as follows:

$$H_1^n(x) = C(n) * [0, 1]^T + w \quad (4.17)$$

$$H_2^n(x) = C(n) * [0, 1]^T - w \quad (4.18)$$

Including the image borders when necessary, a bounding box ρ can be formed between H_1^n , H_2^n and Δ_2^n which will contain the corresponding fish. Knowing the dimensions of ρ , the equivalent area of the second image plane's binary mask, E_2^ρ , is cropped out. A similar analysis is performed, isolating the edges and centroids of all segments present in this block (which in most cases is limited to 1) and the segment with the largest area is chosen as the corresponding labeled segment for the one initially chosen in the first image plane. Now both centroids of the corresponding fish are known as C_i^1 and C_i^2 . Pose correction can now be performed and is started with fitting contours I_n^1 and I_n^2 with ellipses using the standard least squares model as performed in [5].

In order to preserve directionality of the fish, the head and tail have to be marked consistently in image 1 and image 2. The orientation is chosen arbitrarily in image

1 and since the stereo system is setup with orthogonal cameras simple conditions are imposed based on the camera calibration's acquisition of the principal point P_1 and P_2 . Depending on which quadrant the fish was located in image 1, the intersections of the longitudinal (in most cases major) axis of the fish with the edge of the fitted ellipse will have a sign exchange. These two intersections will be called $A_{i,1}^1$, $A_{i,1}^2$ for image 1 and $A_{i,2}^1$, $A_{i,2}^2$ for image 2.

If the triangulation between two points P_1 and P_2 is denoted as $T_1(P_1, P_2)$ or $T_2(P_1, P_2)$, depending on which image is taken as reference, the 3D location of the head and tail can be accurately recovered as P_h and P_t . Once P_h and P_t are established, it is a simple matter to recover the pose of the fish by first ignoring the depth components of $(P_h - P_t)$ and solving for the planar angular change using simple trigonometry followed by solving the depth angular change which will yield the correct stretching parameter in order to properly display the fish's dimensions.

4.6 Re-segmentation

A detailed segmentation of the target needs to be done to extract accurate silhouette to further perform feature extraction. Since, we have a rough estimate of the target foreground and background pixels we can use this information to re-segment the target by training a new background and foreground model of the fish using just the local area in the bounding box established in the original segmentation. To do this we use the Graph Cut based approach given in [23],[24] which uses (Gaussian Mixture Models) GMM's to train a background and foreground model and then minimize an energy function E corresponding to good segmentation. The approach is as follows. An image is considered to consisting of pixels $z_n = z_1, z_2, z_3, z_4, \dots, z_N$ in RGB colour space with each pixel belonging to either foreground or background i.e $\alpha_n = 1$ or $\alpha_n = 0$ respectively. Using the initial segmentation obtained from background modelling we construct GMM's with k components and assign a component $k_n = k_1, k_2, k_3, \dots, k_N$ to each pixel from either the background or the foreground model. We construct a Gibb's

energy function E which is,

$$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z) \quad (4.19)$$

where $U(\alpha, k, \theta, z)$ is the data term for a pixel and $V(\alpha, z)$ is the smoothness prior imposed based on a 4-pixel or 8-pixel neighborhood of a pixel. The expressions for each of these terms are given below.

$$U(\alpha, k, \theta, z) = \sum_n D(\alpha_n, k_n, \theta, z_n) \quad (4.20)$$

where $D(\alpha_n, k_n, \theta, z_n)$ and k_n are obtained from the GMM model,

$$D(\alpha_n, k_n, \theta, z_n) = -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log(\det(\sum \alpha_n, k_n)) \quad (4.21)$$

$$+ \frac{1}{2} [z_n - \mu(\alpha_n, k_n)]^T \sum (\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)], \quad (4.22)$$

and $\theta = \{\pi(\alpha, k), \mu(\alpha, k), \sum(\alpha, k), \alpha = 0 \text{ or } 1, k = 1, 2, 3, \dots, N\}$ are the model, the parameters. π , μ , \sum are the weights, means and covariances of the $2 * K$ Gaussian components for background and foreground distributions. The smoothness prior $V(\alpha, z)$ is defined as,

$$V(\alpha, z) = \gamma \sum_{(m,n) \in N} [\alpha_n \neq \alpha_m] \exp - \beta |z_m - z_n|^2 \quad (4.23)$$

where γ is a constant which determines the degree of smoothness and β is a constant given by $\beta = (\langle (z_m - z_n)^2 \rangle)^{-1}$ ($\langle \rangle$ denotes the expectation).

The objective function to be minimized is,

$$\hat{\alpha} = \arg \min_{\alpha} E(\alpha, \theta). \quad (4.24)$$

This minimisation is performed by computing mincut/maxflow iteratively as the energy E converges monotonically to a local optimum. A detailed step by step flowchart of the algorithm can be found in [23].

4.7 Feature Selection and Recognition

4.7.1 Feature Selection

Size, shape, color and patterns are used as primary features for recognition purposes. Due to the large number of species and intra-species appearance variation, comparing

each test target exhaustively to all variations of each type of species would make it computationally intensive (See Appendix C). Hence, we took help from biologists to roughly categorize target species initially into groups based on shape. Figure 4.6 best illustrates this approach for a unique species and Figure 5.19 shows the different, but related species i.e families that have similar shapes.

In the Figure 4.6, we can see that for the same species classifying on features based on color and patterns alone wouldn't be reliable enough. Also, in Figures 4.7 and 4.8 we see examples of two different fish species which have similar shape and color. In this case unique patterns (White and Blue colored spots near the tail) on the fish help us to distinguish between them. Hence, we take the approach of narrowing down candidate species primarily by shape and then use size, color and patterns features to identify exact species. For further expanding the database to more species more specific features unique to those particular species need to be used.

4.7.2 Recognition

4.7.3 PCA based shape descriptors

As shape is a key feature for identifying fish species, we used PCA(Principal Component Analysis) based shape descriptors [21] to perform shape-based clustering. Shape descriptors of labelled fish were obtained from a coral reef database DVD [27] and also captured and manually segmented images from various other databases. The goal is to classify fish having similar shapes into the same class and then later use color and pattern features to further identify their exact species. To extract the contour of the fish exactly the same way for all images, corresponding landmarks on all image samples need to be matched. For this purpose a normalization procedure is followed on all shape samples. The procedure works as follows:

1. Translation: All shapes are first translated by placing their centroids on the origin point. The centroid of the shape is given by $x_c = \sum_{k=1}^n x_k$.

$$V = [x_1 - x_c, x_2 - x_c, \dots, x_n - x_c]$$

2. Rotation: The sample is rotated around the principal axis to make the shape horizontal. The rotation angle is determined by fitting an ellipse to the contour points and finding the orientation of the major axis.
3. Scaling: After translation and rotation, the size of the shape is normalized by scaling the shapes to a bounding box of the same area.
4. Re-sampling: $N = 40$ evenly space points are sampled along the contour, starting from the farthest intersection from the centroid of the contour and the x-axis.

Using the above sample points the shape is represented as a $2N$ dimensional vector $v = [x_1, \dots, x_N, y_1, \dots, y_N]$. To include possible flips (up-down, left-right) of the shape we also include the flipped versions of them in our analysis. Thus for each shape v_i^k , where $k = 1, 2, 3, 4$, we have the original, an up-down flipped, a left-right flipped and an up-down then left-right flipped normalized shape. Using these normalized shapes, features are extracted using Principal Component Analysis (PCA).

The mean shape of the database containing M shapes is given by,

$$\bar{v} = \frac{1}{4M} \sum_{i=1}^M \sum_{k=1}^4 v_i^k \quad (4.25)$$

, and the covariance matrix is,

$$C = \frac{1}{4M} \sum_{i=1}^M \sum_{k=1}^4 (v_i^k - \bar{v})(v_i^k - \bar{v})^T. \quad (4.26)$$

Selecting the t largest eigenvalues and corresponding eigenvectors $P = [p_1, p_2, p_3, \dots, p_t]$. We have the shape v_i^k represented as $v_i^k = \bar{v} + Pb_i^k + \epsilon$ where $b_i^k = [b_{i,1}^k, b_{i,2}^k, \dots, b_{i,t}^k]$ are the weights and ϵ is the error. This error decreases as the number of eigenvectors increases. So each shape is now represented by a set of feature vectors $b_i^1, b_i^2, b_i^3, b_i^4$. To match two shapes v_i and v_j 16 euclidean distances $d_{m,n} = (b_i^m - b_j^n)^T (b_i^m - b_j^n)$ corresponding to two sets of features vectors are calculated. The minimum euclidean distance between any two sets is taken as the true distance between the two shapes.

$$D(v_i, v_j) = \min_{m,n} (d_{m,n}) \quad (4.27)$$

Species having similar shapes are clustered using the above distance metric and are further classified using additional features.

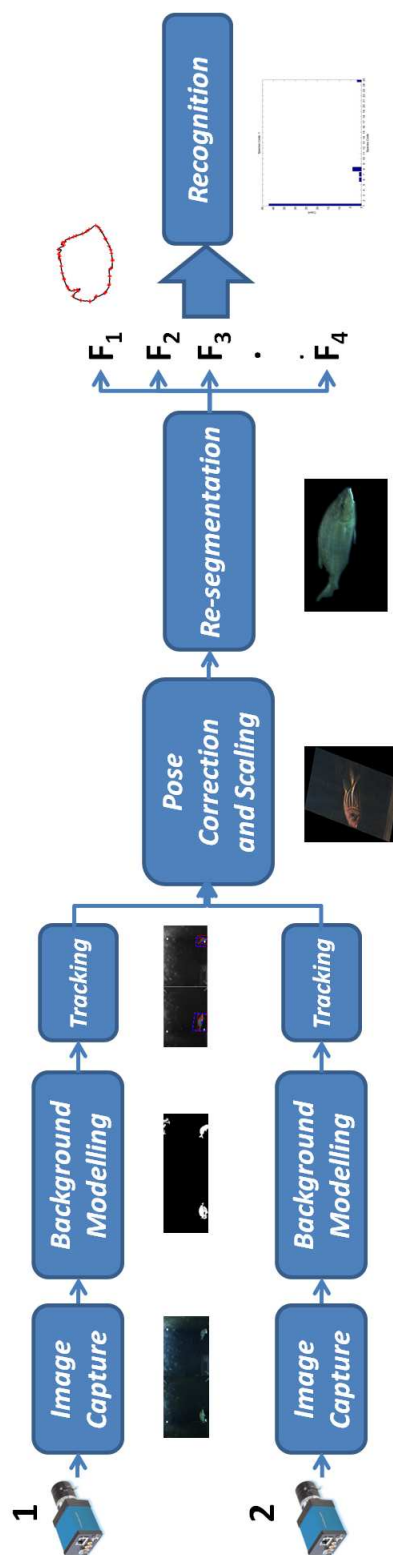


Figure 4.1: Overall software design with example intermediate results below every block.
The system begins with

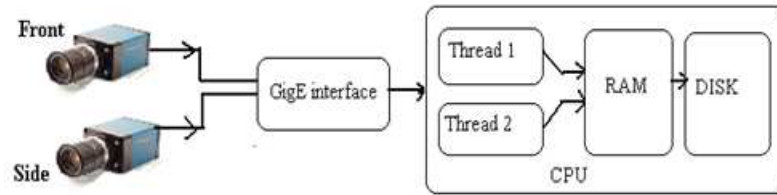


Figure 4.2: Image capture system block diagram.

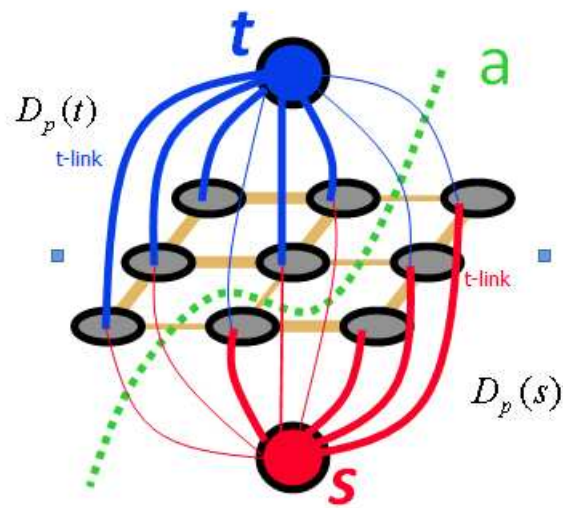


Figure 4.3: Graph constructed from image with source s and sink t . The cut (dotted line) corresponds to the maximum flow and minimum edge cut cost.

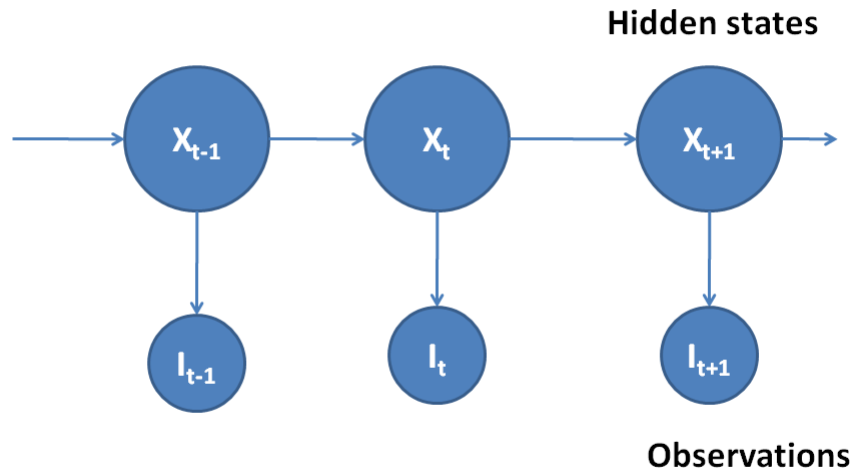


Figure 4.4: Hidden Markov model with hidden states X_t and observation variables I_t .

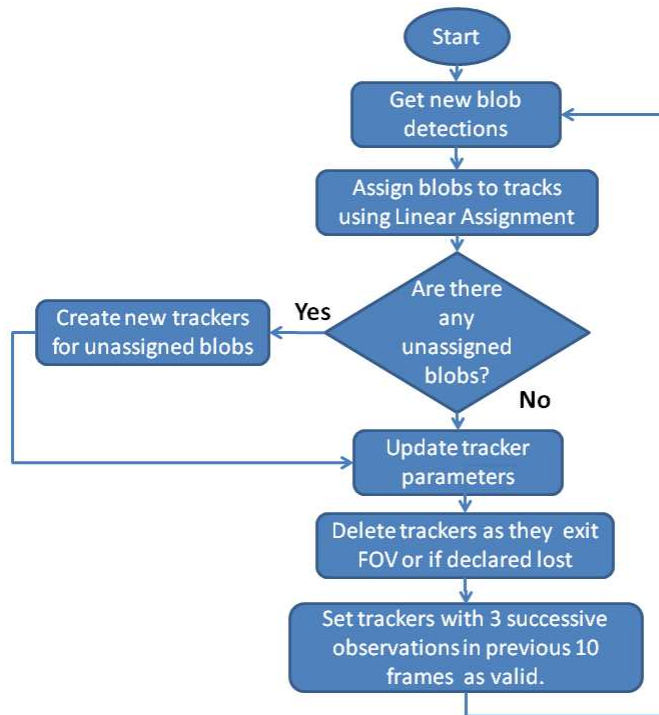


Figure 4.5: Flow chart showing management of trackers and how tracks are created, validated to display and then later deleted.

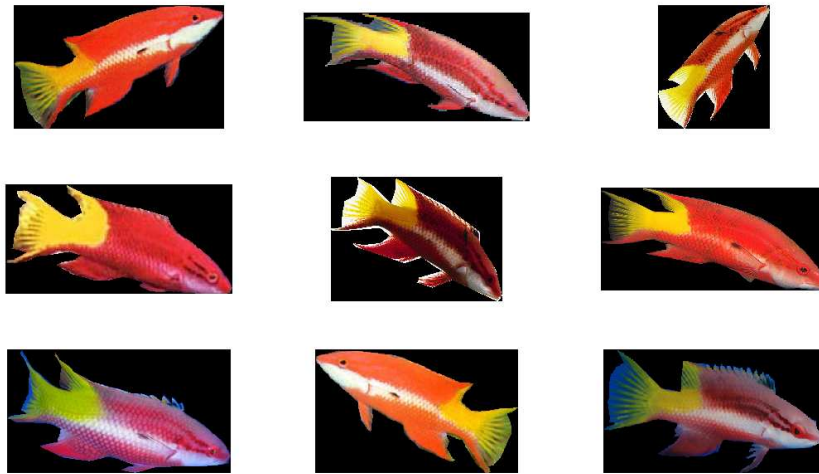


Figure 4.6: "Bodianus puchellus- Cuban hogfish" - Variations of color and patterns within species but having strong similarity in shape.

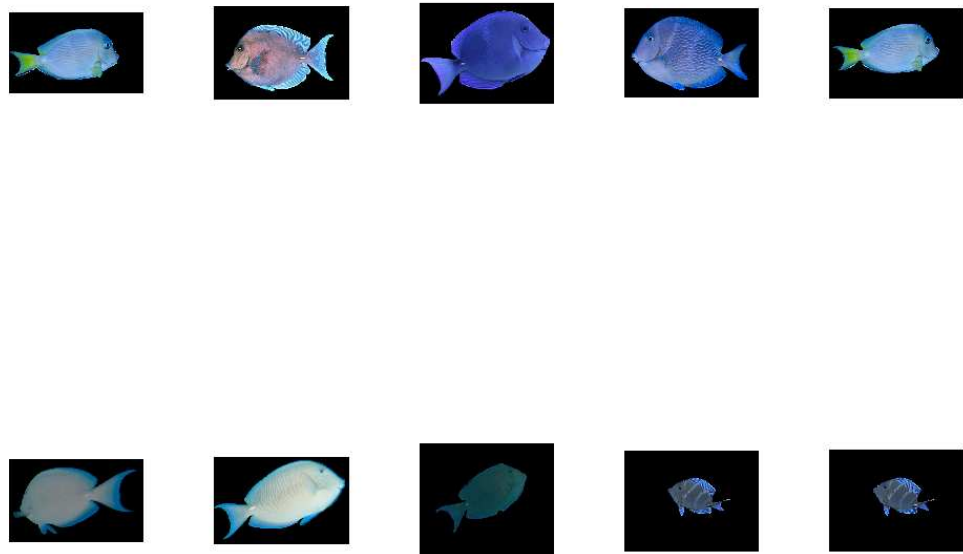


Figure 4.7: "Acanthurus coeruleus - Blue Tang" - Blue Tang with shape and color similarities with Doctor Fish (Figure 4.8).

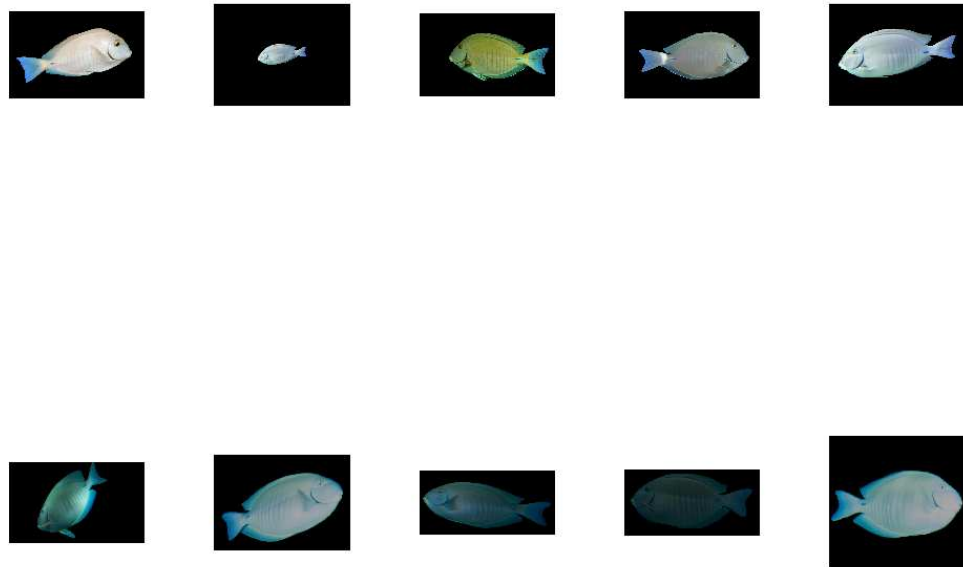


Figure 4.8: "Acanthurus chirurgus - doctorfish" - Doctor Fish with shape and color similarities with Blue Tang (Figure 4.7).

Chapter 5

Results

In this chapter we report the results of each stage of pre-processing. Each section below describes the pre-processing results and further processing requirements of the next stage.

5.1 Data Collection

Our current data collection software system was designed to periodically capture image sequences. Sequences of 300 image pairs were captured after every 12 minutes at 10frames/sec at a full resolution of 1024×1360 . The system was set to capture sequences daily during the daytime from morning 9:00 am to evening 7:00 pm four days a week excluding Friday, Saturday and Sunday. With one video sequence captured every 12 minutes, we captured 5 sequences per hour, which is 50 sequences per day, which is 200 sequences per week. These sequences were captured and then transferred to an external hard drive which was replaced with an empty one for the following week. We will make suitable changes to the software in the final version when the system is installed in Belize.

The algorithm for capturing sequences is given below:

Algorithm: CaptureSequence()

1. Set prevtime = current system time
2. Set curtime = current system time
3. If curtime == shutdowntime then stop capturing and shutdown the PC else goto next step

4. If curtime == (prevtime + 10 min) goto step 5 else goto step 2
5. Set prevtime = curtime
6. Start sequence capture
7. Wait for 6 min for the capture to finish
8. Goto step 2

With such a large amount of captured sequences, it was important, for initial data collection to quickly analyse, and find those with at least one fish and reject the others. For this purpose we developed a "quick" background subtraction technique to look for moving foreground blobs in the sequences. The flowchart for this "quick" background subtraction and sequence classification technique is given in figure 5.1.

Since the technique uses the median image as the background model for the sequences, this approach is prone to false detections caused by moving debris and changes in illumination. For this reason, some manual intervention is required to reject these sequences, which are falsely classified as with fish even though they do not contain any. Examples of an image pair found by the "quick" background subtraction technique are given in Figures 5.2 and 5.3.

In all, 403 image sequences were captured over a period of 18 weeks for processing. Out of these, 253 sequences were found to contain at least one image pair with a clear view of a fish in both views, 55 sequences contained image pairs with a clear view of a fish only in one of the view and remaining sequences contained no fish.

5.2 Background Modelling

The goal of Background Modeling is to detect moving objects which constitute the foreground of a scene. Figure 5.4 shows a "typical" image pair with a single fish totally in view in one camera and partially in view in other. The image pair consists of the following objects:

- Background objects

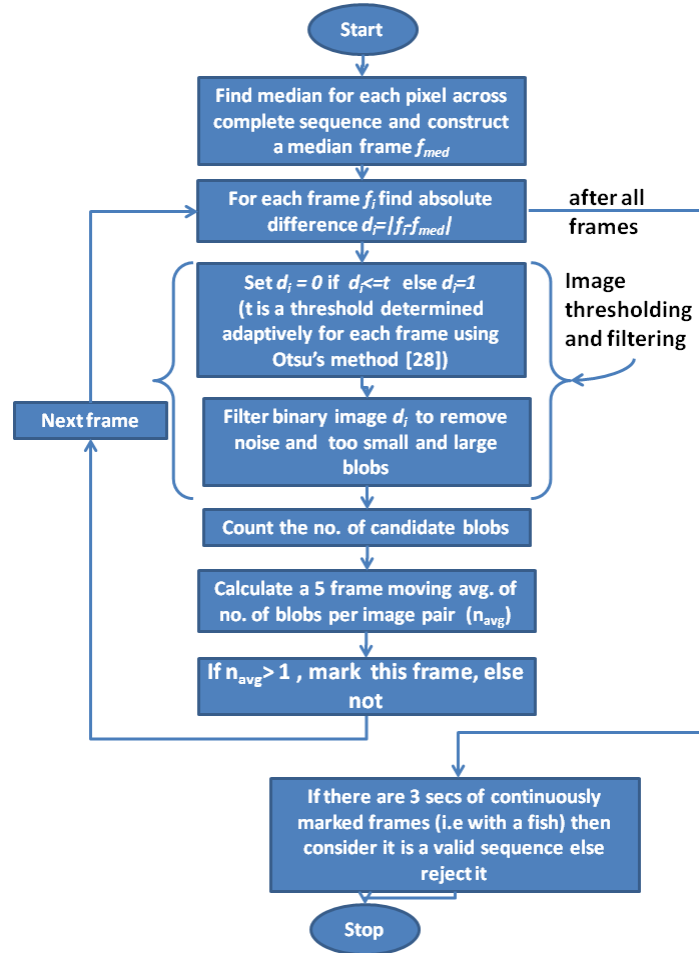


Figure 5.1: Flowchart describing the "quick" background subtraction and sequence classification process.

1. Two black background panels with white vertical alignment rectangles (used to correct, in software, for small height differences between the two camera platforms)
 2. Visible PVC structure.
- Foreground objects
 1. Target fish
 2. Floating debris

The Background Modelling technique is expected to primarily detect the fish, our object of interest, in the foreground, and reject everything else.

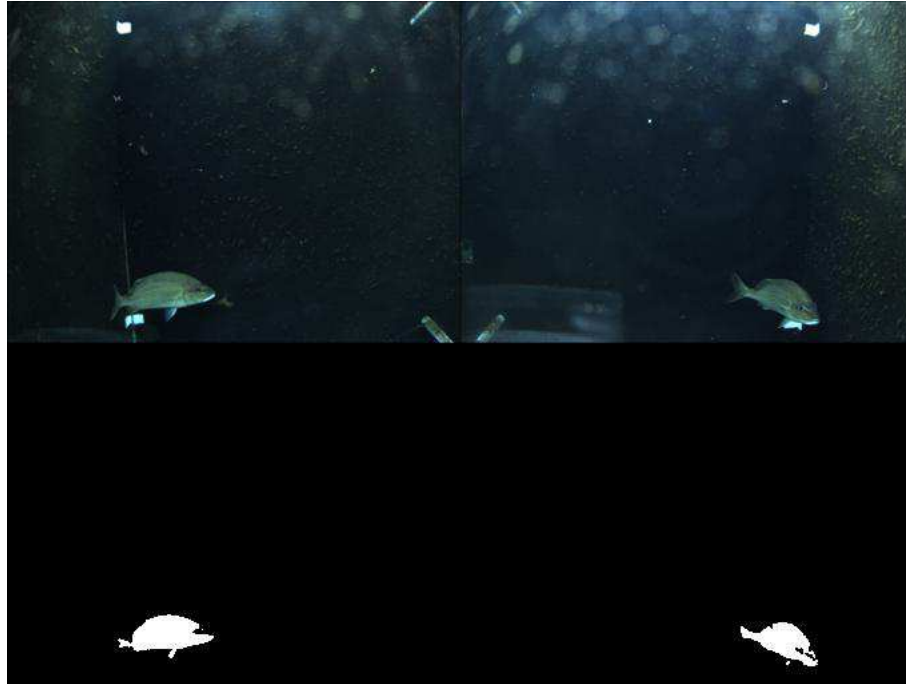


Figure 5.2: An image pair from sequence 1 (05/25/2010 at 10:49:24 hrs) showing the "quick" background subtraction result.

Background Modelling results for the "typical" image pair shown in Figure 5.4 are given in Figure 5.5. The foreground is indicated in white (1's) and the background in black (0's). We see that the target fish(true positive) is detected accurately along with some false positives. These false positives consists of; the floating debris in the scene and a part of the stationary background (right image-top right corner). The reason for these false positives is that the background modeling accounts for a stationary background with only a gradual variation in illumination. Since the floating debris are not modelled explicitly by this technique, they are detected as part of foreground. Also, small but sudden changes in illumination in the background, specular reflections, caused by varying sunlight and non-lambertian¹ surface reflectance of the housings, causes some part of the background to be detected as foreground.

Many of these false positives can be eliminated based on size and location but still a few remain (e.g. floating debris in left image and the previously mentioned part of

¹If a surface exhibits Lambertian reflectance, light falling on it is scattered such that the apparent brightness of the surface to an observer is the same regardless of the observer's angle of view.

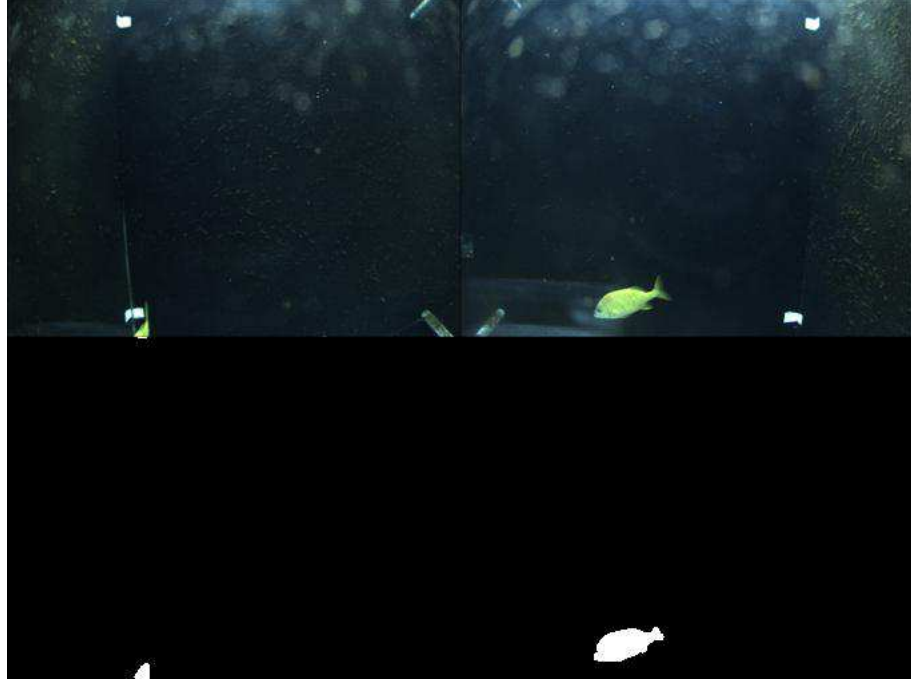


Figure 5.3: An image pair from sequence 2 (05/24/2010 at 11:51:40 hrs) showing the "quick" background subtraction result with a whole fish in one view on the right and a partial segment of the same fish near the image border on left.

the stationary background in the upper right part) which need to be handled explicitly. Figure 5.6 shows the post-processing result of an initial dilation with a disk kernel of 3×3 size and the elimination of blobs too small or too large ($< 0.005\%$ and $> 50\%$ of image area).

Figures 5.7 and 5.8 show every 5th image pair of a sequence with the original image pair on the top and the background subtracted and post-processed image below it. In the sequences of images, we see that we detect the foreground objects successfully with no misses(true negatives). The Background Subtraction also handles gradual change of illumination from sunlight and localized illumination variation which happens due to wave motion of water. The parameter settings used for the sequences are given in Table 5.1, where training and testing frames were selected without fish and with fish respectively. This requirement, although not strict, was only selected only for evaluation purposes. In real-life, the system could train itself on some fixed set of background images without fish initially during installation and then Background Modelling would adaptively adjust the model during the testing phase.

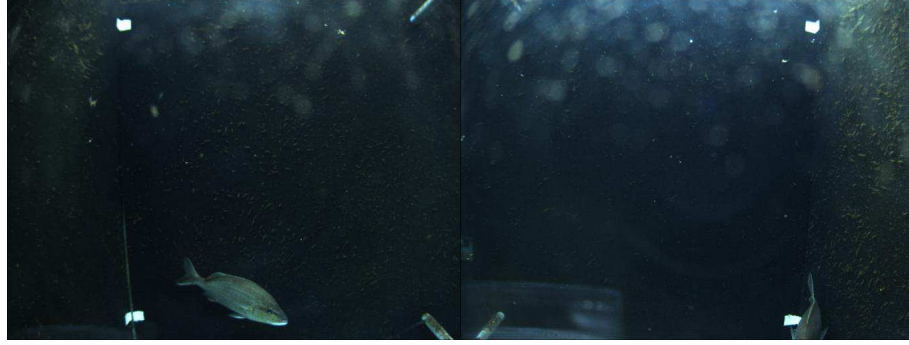


Figure 5.4: A "typical" image pair from sequence 1 (05/25/2010 at 10:49:24 hrs).



Figure 5.5: Background result for the "typical" image pair from sequence 1.

5.3 Tracking

The detection results obtained from Background Modeling now mainly contain, the target fish blobs, which need to be tracked and recognized and floating debris, which also need to be tracked carefully for elimination.

The tracker initializes with the first frame and then later validates the tracks as it tracks across the frames. We explain the marking convention using a single image pair shown in Figure 5.9 and later on present only tracking results on a complete sequence.

Figure 5.9 shows an image pair with different colored bounding boxes. The red box indicates foreground detections from Background Modelling, the dashed blue-red box the tracker and a red number indicates tracker id.

Figures 5.10 and 5.11 show tracking results for the two selectd sequences. We see that the tracker is able to handle pose variations.

In Figures 5.11(b)) and 5.11(e), we observe that that the tracker also eliminates



Figure 5.6: Background modelling result after post-processing for the "typical" image pair.

Parameter	Sequence 1	Sequence 2
Timestamp	05/25/2010 at 10:49:24 hrs	05/24/2010 at 11:51:40 hrs
Training frames	0-80	101 - 199
Testing frames	81-220	0-100

Table 5.1: The parameter setting used for the sequences

false positives which appear momentarily.

Using these tracks, we can now select the best view of a fish based on orientation of the fish and perform the post-processing operations of pose correction, scaling and re-segmentation for species recognition. The image pairs were selected based on maximum horizontal axis length(after rotating the blobs so that the major axis lies in the horizontal direction) but further investigation needs to be done for a better and robust criteria for the best view selection for the fish.

5.4 Pose Correction and Scaling

From the previous work done in this project, we demonstrate one example of an image pair on which we can perform accurate pose correction and scaling of a fish, which is visible in both views.² Figure 5.12 shows such an image pair with fish overlapping in one view but not in the other. Figure 5.13 shows the same fish in an image pair taken a few frames later, here the fish are no longer overlapping. Here, we show that we can handle partial occlusion and track multiple candidate objects effectively.

²This work was carried out by Lev Barinov and Jigesh Baxi, students who participated in the earlier phase of the project.

Using the technique described in section 4.5, we perform pose correction and scaling on the later image pair(See Figure 5.14). The binary image mask used as the input for the processing.

Figure 5.15 shows the intermediate matching result with the matched fish marked by blue ellipses in both views. The blue horizontal lines in the left image indicate the search region for the corresponding fish in the right image.

The scaled and pose corrected fish is shown in figure 5.16 and is now ready for further preprocessing, feature extraction and classification.

5.5 Re-segmentation

To segment fish shape accurately from its immediately surrounding background, we place a scaled (1.2 times) bounding box around the target and perform a detailed re-segmentation within the box as described in section 4.6. The final result of these detailed segmentations are shown in Figures 5.17 and 5.18.

We can see that by performing a detailed segmentation within the bounding box we obtain detailed shape features and also are able to get rid of some extraneous background regions which were detected as foreground and became a part of the fish.

The implementation for this part of the work was done using Interactive Segmentation Toolbox for MATLAB® [22].

5.6 Recognition

In this section we demonstrate our ability to use shape as a feature to segregate different families of species for classification. Our future work will focus on using other features such as size, color and patterns to further pinpoint exact species.

We use a coral reef species database (See Appendix C) compiled by two summer students using images from the Internet, during the summer of 2009. The database consists of 25 different fish species along with their manually extracted silhouettes. These species were selected, first, because they are found on the coral reef in Belize(and also in the Belize coral reef exhibit at the New York aquarium) and second, because

Code	Species	no. of examples
1	Elacatinus randalli-Yellownose goby	10
2	Chaetodon ocellatus-Spotfin butterfly	10
3	Abudefduf saxatilis-Sergeant major	10
4	Bodianus puchellus-Cuban hogfish	10
5	Haemulon flavolineatum-French grunt	10
6	Pseudupeneus maculatus-Spotted goatfish	10
7	Haemulon melanurum-Cottonwick grunt	10
8	Halichoeres bivittatus-Slippery Dick	10
9	Acanthurus chirurgus-Doctorfish	10
10	Acanthurus coeruleus-Blue Tang	10
11	Mulloidichthys Martinicus - Yellow goatfish	10
12	Chaetodon capistratus-Four-eyed Butterfly	10
13	Bodianus rufus-Spanish hogfish	10
14	Anisotremus virginicus-Porkfish	10
15	Pomacanthus paru-French Angelfish	10
16	Holacanthus ciliaris-Queen Angel	10
17	Haemulon sciurus-Bluestriped Grunt	10
18	Selene vomer-Lookdown	10
19	Serranus tortugarum-Chalk Bass	10
20	Chromis scotti purple-Reeffish	10
21	Hypoplectrus indigo-Indigo Hamlet	10
22	Chromis cyanea-Blue Chromis	10
23	Chaetodon sedentarius-Reef Butterflyfish	10
24	Chaetodon striatus-banded Butterflyfish	10
25	Halichoeres garnoti-Yellowhead Wrasse	10

Table 5.2: Fish codes with common names and number of examples used for each species in the database

they are small enough to be imaged in our prototype viewing system. We select 10 examples for each fish species and perform a leave-one-out cross validation to report our results. Table 5.2 shows the 25 species with their biological and common names. The Code column in the left indicates a unique number for each species; hereinafter we use these codes to indicate specific species.

We tested our shape classifier, which we described in section 4.7.3, to cluster shapes of the example species. We rely upon the fact that fish species belonging to the same common-family have similar shapes, and are grouped together to form a cluster within the same family. We use the minimum distance definition as described in section 4.7.3, and form groups of species by examining the top five matched species for each example of a species. We observe whether these top five matches for each of these example

species are from the same families and as we describe later, it turns out, for the most part, to be the case.

In Figures 5.21 and 5.22 we show the different fish species along with their extracted boundaries from silhouettes. The fish pose was corrected by using a simple technique of rotating the major axis of the fish to make it horizontal. The fish boundary is traced beginning from the extreme left, and the up-down, left-right and up-down-left-right flipped versions are also taken into account as described in the section 4.7.3. We select 50 evenly spaced points on the contour of the fish (See Figure 5.20). We compute the first 15 eigenvalues and eigenvectors of the covariance matrix. We ignore the first component of the eigenvalue while computing the minimum distance as it contains only the DC offset information.

In the leave-one-out cross validation we compute the top 5 matches for each example of a species. In all we have 50 matches for 10 examples and note down the different species which come up in these 50 matches. The tables below show these different species (right column) for each of the selected species(left column).

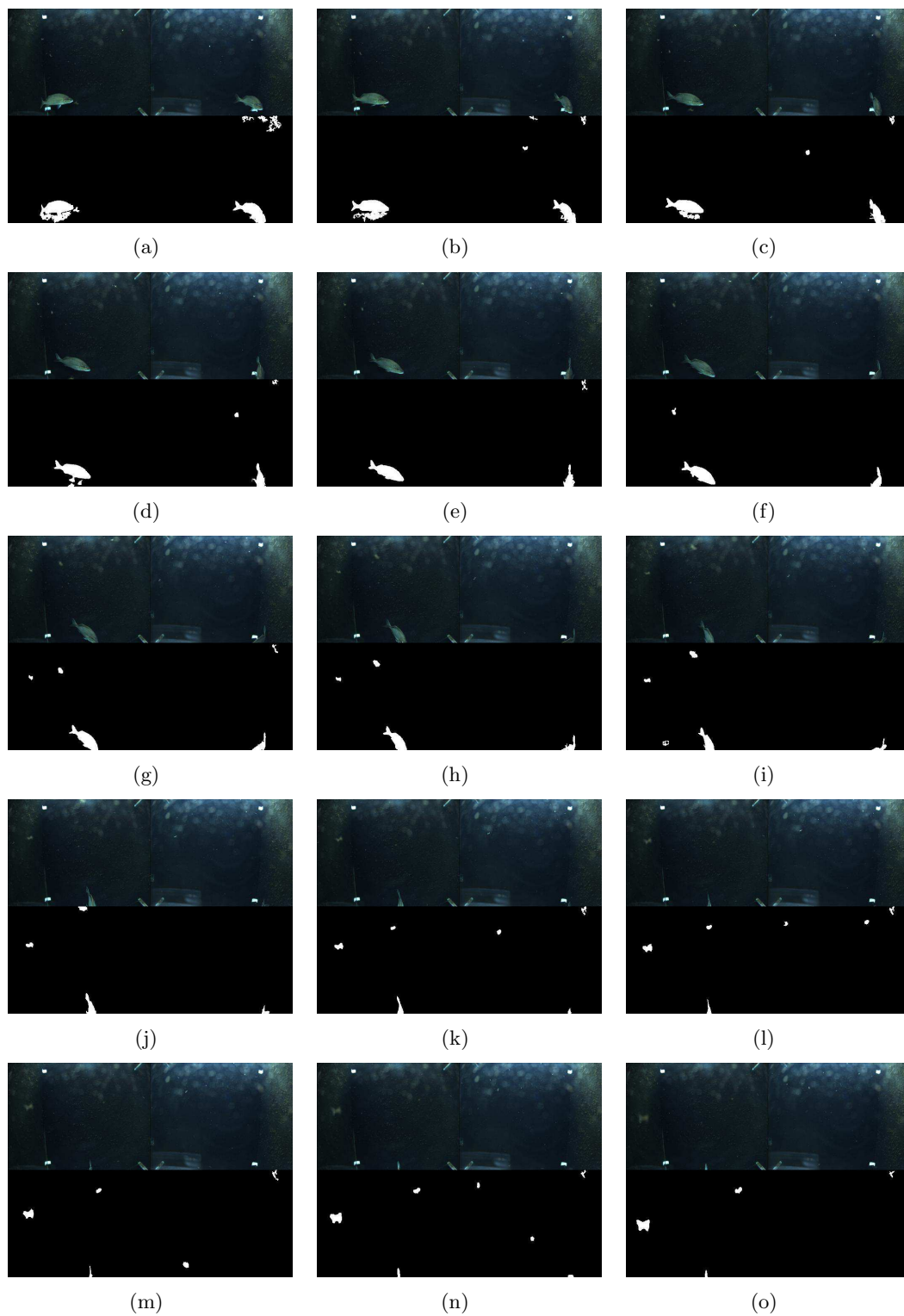


Figure 5.7: Background modelling results on sequence 1 (05/25/2010 at 10:49:24 hrs). Shows every 5th image pair from 0, 5, ...70

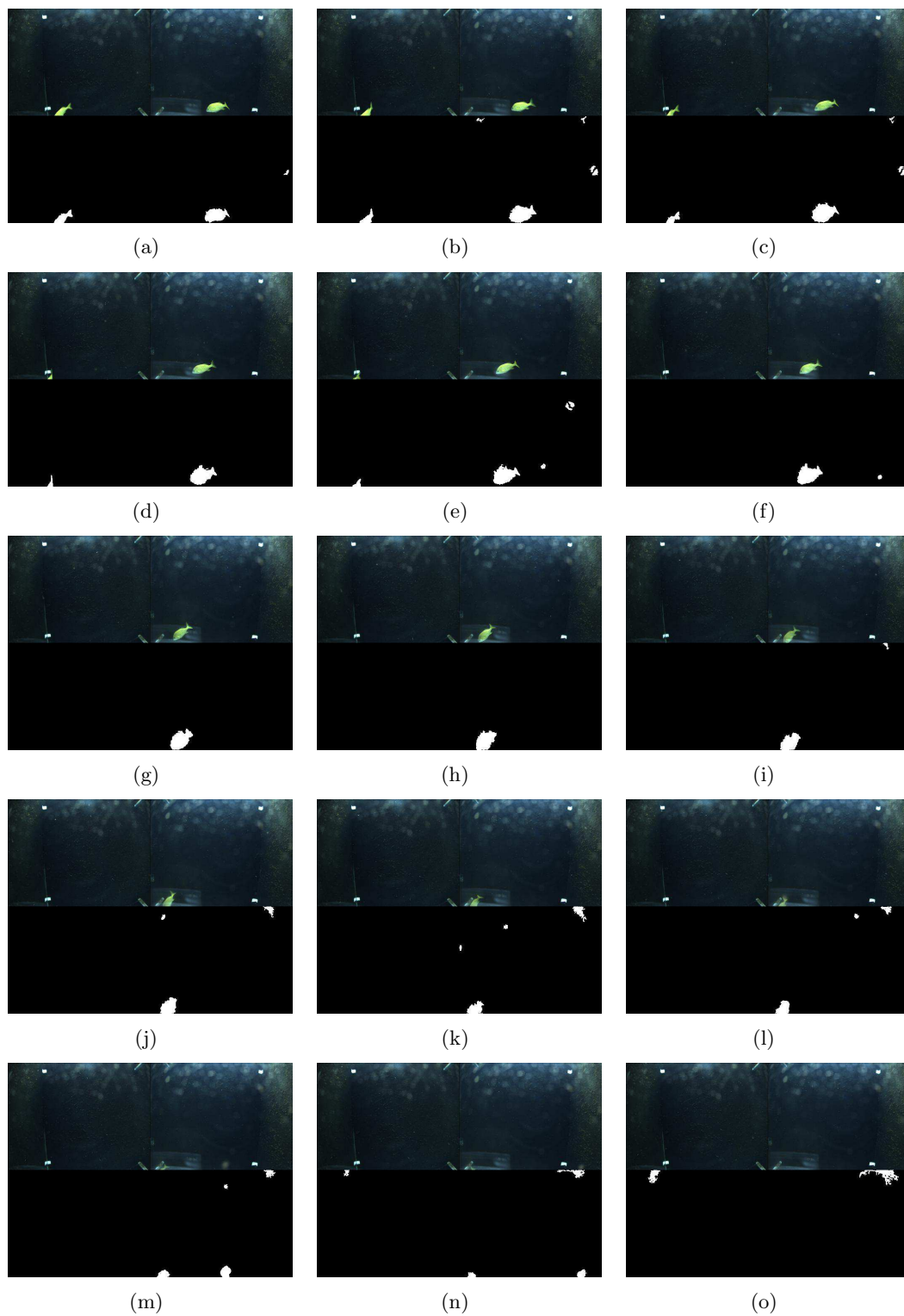


Figure 5.8: Background modelling results on sequence 2 (05/24/2010 at 11:51:40 hrs). Shows every 5th image pair from 135, 140, ...210

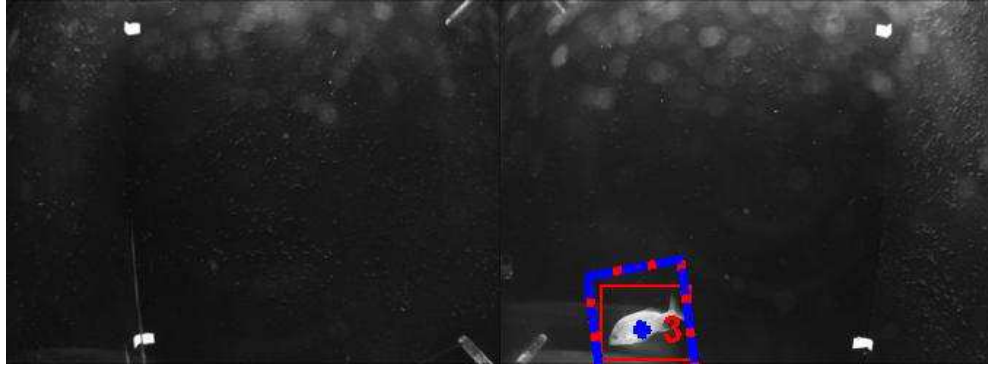


Figure 5.9: Tracking result on a "typical" scene. Red box indicates foreground detections from Background Modelling, dashed blue-red box is the tracker and a red number indicating tracker id.

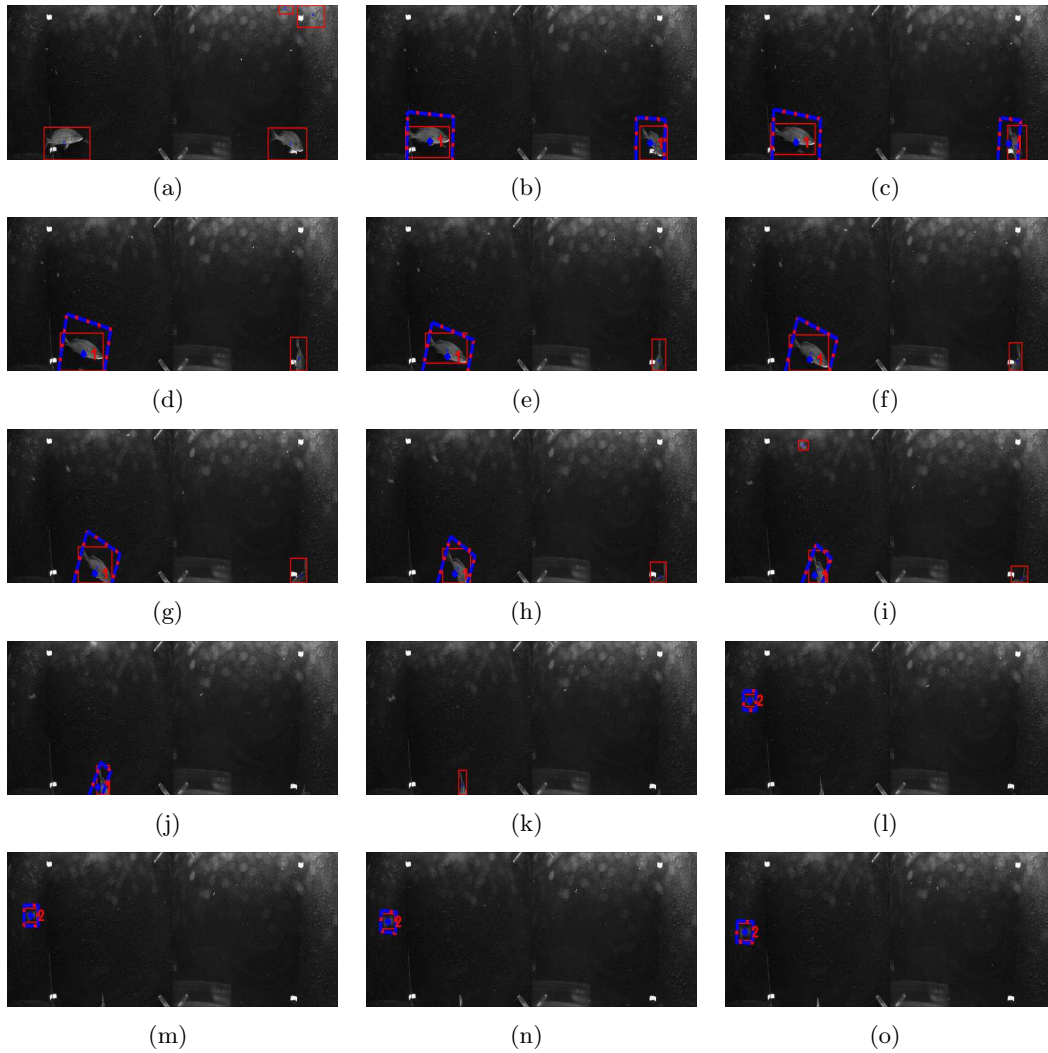


Figure 5.10: Tracking results on sequence 1 (05/25/2010 at 10:49:24 hrs). Shows every 5th image pair from 0, 5, ...70. Red box indicates foreground detections from Background Modelling, dashed blue-red box is the tracker and a red number indicating tracker id.

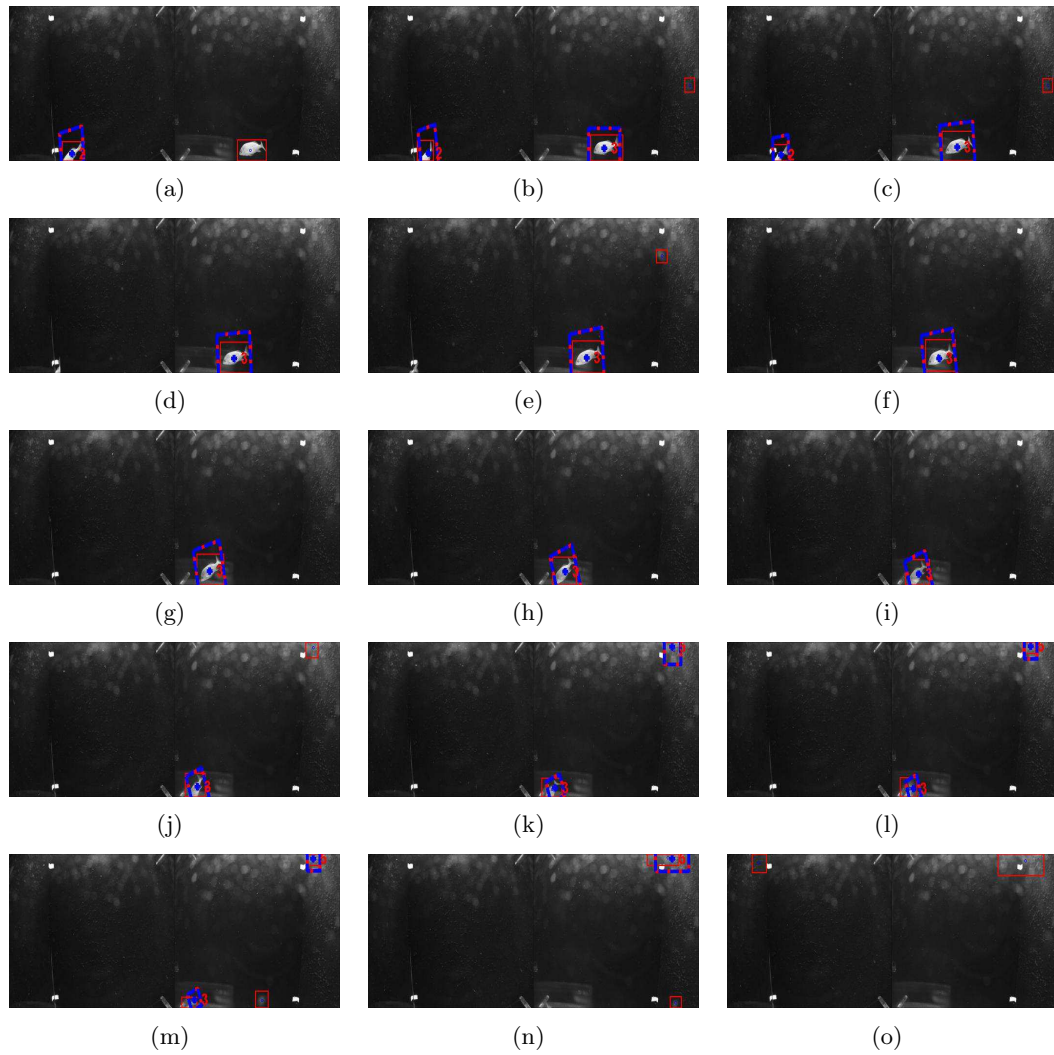


Figure 5.11: Tracking results on sequence 2 (05/24/2010 at 11:51:40 hrs). Shows every 5th image pair from 135, 140, ... 210. Red box indicates foreground detections from Background Modelling, dashed blue-red box is the tracker and a red number indicating tracker id.



Figure 5.12: Image pair with fish overlap.



Figure 5.13: Image pair after few frames without overlap.



Figure 5.14: Image pair with ellipses around each observed fish.



Figure 5.15: Matched fish in both views marked with a blue ellipse. Two horizontal lines across in the left image denote the search region.



Figure 5.16: Pose corrected and scaled fish.



(a) Initial binary mask



(b) Initial color mask



(c) Re-segmented binary mask



(d) Re-segmented color mask

Figure 5.17: Re-segmentation results on sequence 1 (05/25/2010 at 10:49:24 hrs). Figures 5.17(a) and 5.17(c) show the binary mask (black-background, white-foreground) before re-segmentation. Figures 5.17(b) and 5.17(d) show their corresponding color mask's.

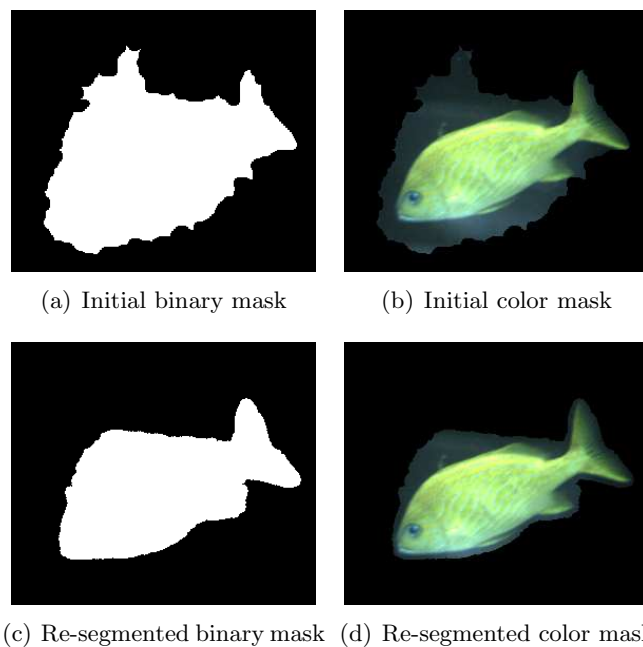


Figure 5.18: Re-segmentation results on sequence 2 (05/24/2010 at 11:51:40 hrs). Figures 5.18(a) and 5.18(c) show the binary mask (black-background, white-foreground) before re-segmentation. Figures 5.18(b) and 5.18(d) show their corresponding color mask's.

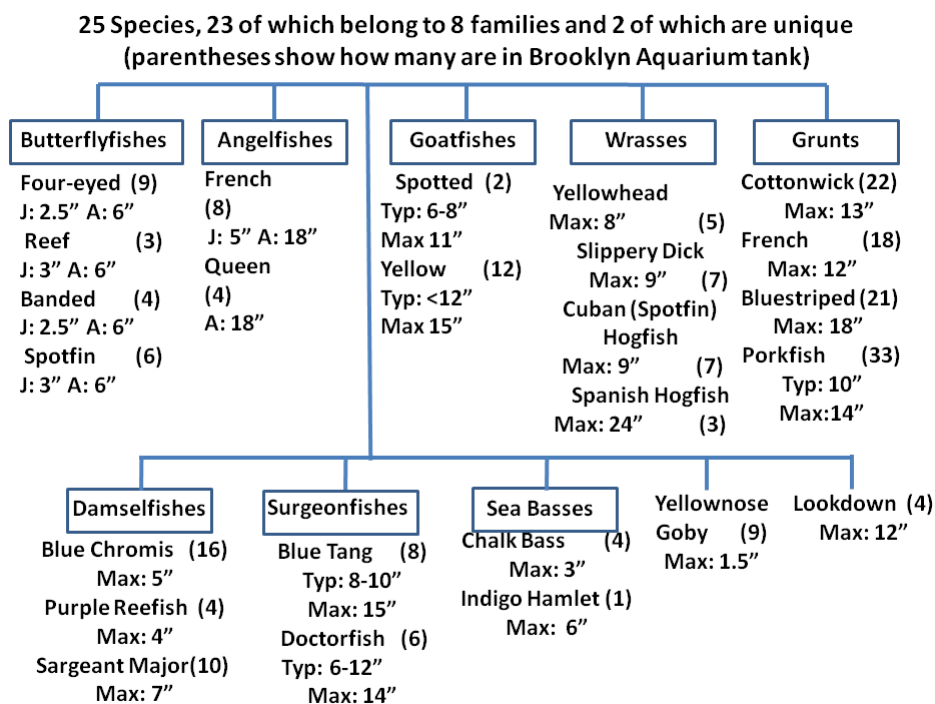


Figure 5.19: Species grouped together which have strong similarity in shape.

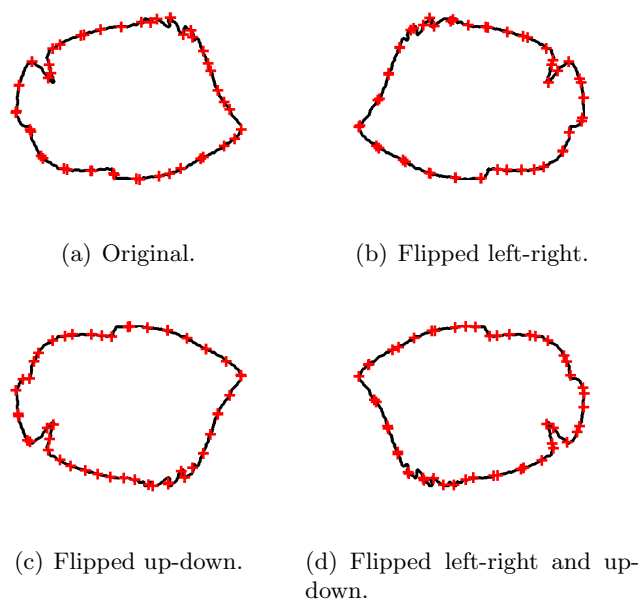


Figure 5.20: Contour tracing on a example fish from the database. Figure 5.20(a) original, Figure 5.20(b) flipped left-right, 5.20(c) flipped up-down and 5.20(d) flipped left-right and up-down respectively.

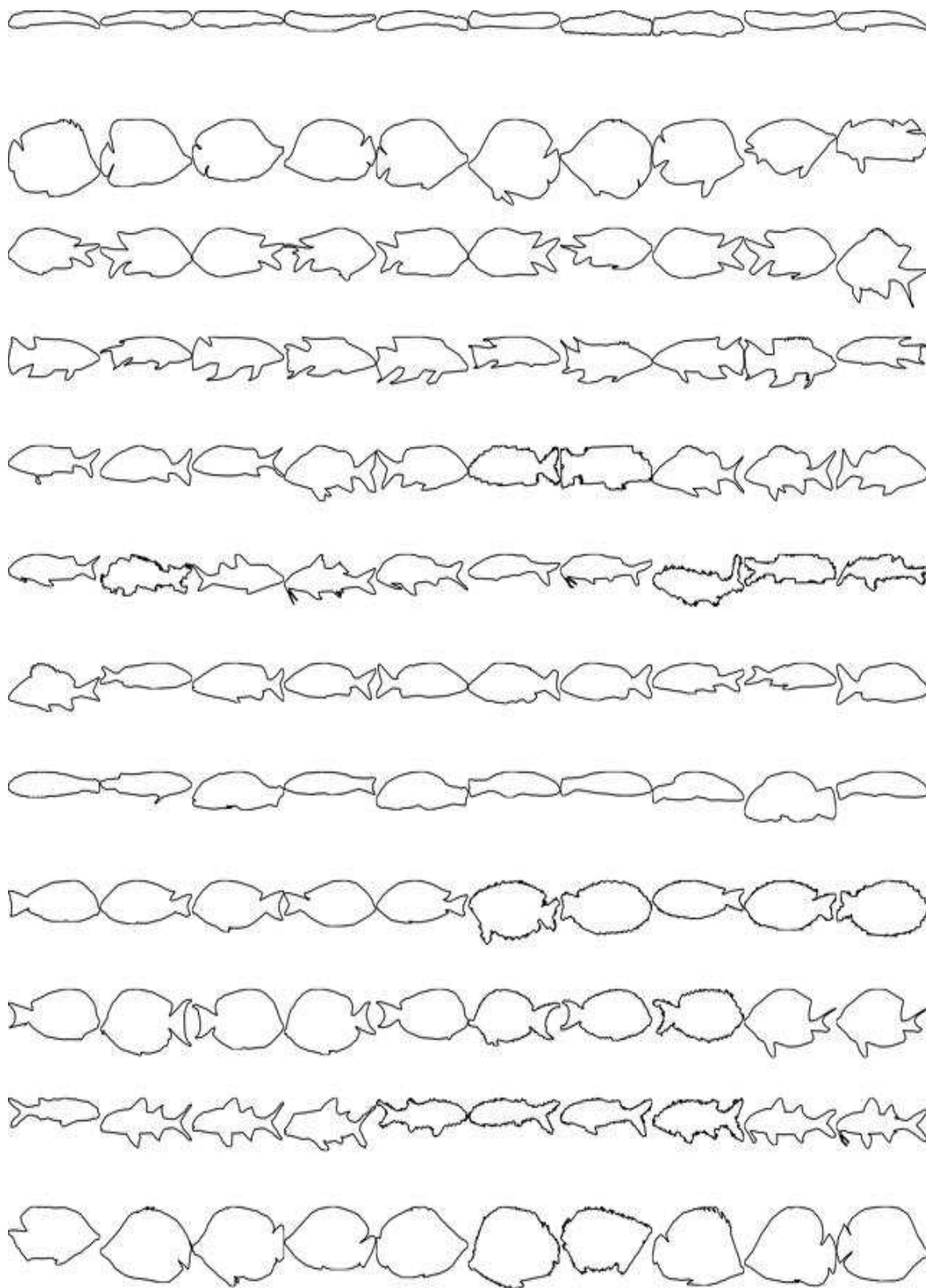


Figure 5.21: Shapes of of species from 1-12 from the database.

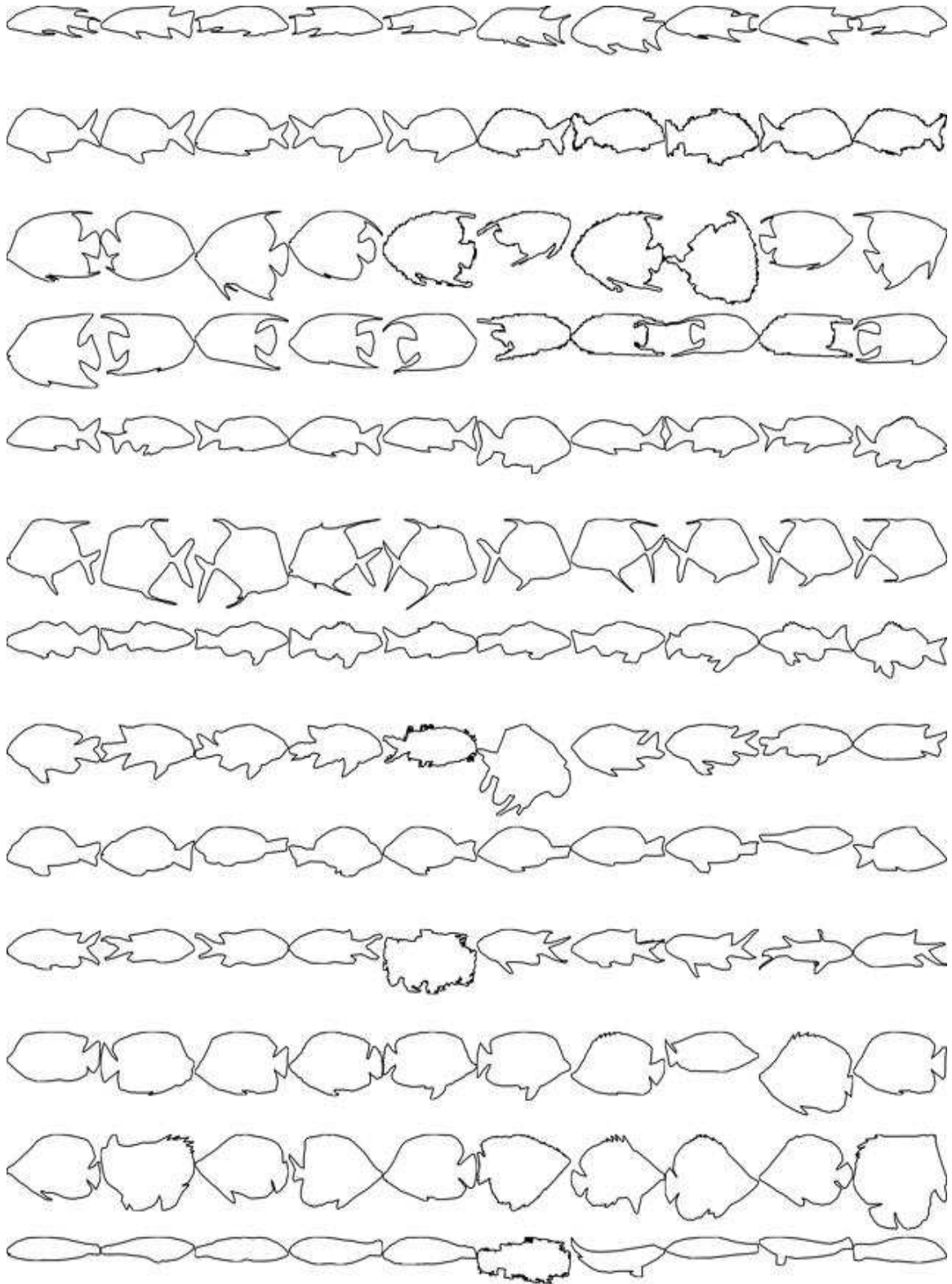


Figure 5.22: Shapes of of species from 13-25 from the database.

1-Elacatinus randalli yellownose goby	Code	Species
	1	Elacatinus randalli yellownose goby
	7	Haemulon melanurum cottonwick grunt
	8	Halichoeres bivittatus Slippery Dick
2- Chaetodon ocellatus spotfin butterfly	Code	Species
	2	Chaetodon ocellatus spotfin butterfly
	9	Acanthurus chirurgus doctorfish
	12	Chaetodon capistratus four-eyed butterfly
	14	Anisotremus virginicus porkfish
	19	Serranus tortugarum chalk bass
	20	Chromis scotti purple reeffish
	21	Hypoplectrus indigo indigo hamlet
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish
3- Abudefduf saxatilis Sergeant major	Code	Species
	3	Abudefduf saxatilis Sergeant major
	9	Acanthurus chirurgus doctorfish
	13	Bodianus rufus Spanish hogfish
	17	Haemulon sciurus bluestriped grunt
	18	Selene vomer lookdown
	20	Chromis scotti purple reeffish
	22	Chromis cyanea blue chromis
	24	Chaetodon striatus banded butterflyfish

4- Bodianus puchellus Cuban hogfish	Code	Species
	3	Abudefduf saxatilis Sergeant major
	4	Bodianus puchellus Cuban hogfish
	7	Haemulon melanurum cottonwick grunt
	9	Acanthurus chirurgus doctorfish
	13	Bodianus rufus Spanish hogfish
	14	Anisotremus virginicus porkfish
	17	Haemulon sciurus bluestriped grunt
	22	Chromis cyanea blue chromis
5- Haemulon flavolineatum French grunt	Code	Species
	5	Haemulon flavolineatum French grunt
	6	Pseudupeneus maculatus spotted goatfish
	7	Haemulon melanurum cottonwick grunt
	9	Acanthurus chirurgus doctorfish
	11	Mulloidichthys martinicus
	14	Anisotremus virginicus porkfish
	17	Haemulon sciurus bluestriped grunt
	19	Serranus tortugarum chalk bass
	20	Chromis scotti purple reeffish
	21	Hypoplectrus indigo indigo hamlet

6- Pseudupeneus maculatus spotted goatfish	Code	Species
	1	Elacatinus randalli yellownose goby
	5	Haemulon flavolineatum French grunt
	6	Pseudupeneus maculatus spotted goatfish
	7	Haemulon melanurum cottonwick grunt
	8	Halichoeres bivittatus Slippery Dick
	10	Acanthurus coeruleus Blue Tang
	11	Mulloidichthys martinicus
	13	Bodianus rufus Spanish hogfish
	17	Haemulon sciurus bluestriped grunt
	19	Serranus tortugarum chalk bass
7- Haemulon melanurum cottonwick grunt	Code	Species
	5	Haemulon flavolineatum French grunt
	7	Haemulon melanurum cottonwick grunt
	8	Halichoeres bivittatus Slippery Dick
	9	Acanthurus chirurgus doctorfish
	17	Haemulon sciurus bluestriped grunt
	19	Serranus tortugarum chalk bass
	21	Hypoplectrus indigo indigo hamlet
	25	Halichoeres garnoti yellowhead wrasse

8- Halichoeres bivittatus Slippery Dick	Code	Species
	1	Elacatinus randalli yellownose goby
	7	Haemulon melanurum cottonwick grunt
	8	Halichoeres bivittatus Slippery Dick
	14	Anisotremus virginicus porkfish
	17	Haemulon sciurus bluestriped grunt
	19	Serranus tortugarum chalk bass
	21	Hypoplectrus indigo indigo hamlet
	25	Halichoeres garnoti yellowhead wrasse

9- Acanthurus chirurgus doctorfish	Code	Species
	3	Abudefduf saxatilis Sergeant major
	4	Bodianus puchellus Cuban hogfish
	5	Haemulon flavolineatum French grunt
	7	Haemulon melanurum cottonwick grunt
	9	Acanthurus chirurgus doctorfish
	10	Acanthurus coeruleus Blue Tang
	11	Mulloidichthys martinicus
	14	Anisotremus virginicus porkfish
	16	Holacanthus ciliaris queen angel
	19	Serranus tortugarum chalk bass
	21	Hypoplectrus indigo indigo hamlet

10- Acanthurus coeruleus Blue Tang	Code	Species
	3	Abudefduf saxatilis Sergeant major
	9	Acanthurus chirurgus doctorfish
	10	Acanthurus coeruleus Blue Tang
	14	Anisotremus virginicus porkfish
	17	Haemulon sciurus bluestriped grunt
	20	Chromis scotti purple reeffish
	21	Hypoplectrus indigo indigo hamlet
	23	Chaetodon sedentarius reef butterflyfish
11- Mulloidichthys martinicus yellow goatfish	Code	Species
	3	Abudefduf saxatilis Sergeant major yellow goatfish
	5	Haemulon flavolineatum French grunt yellow goatfish
	6	Pseudupeneus maculatus spotted goatfish yellow goatfish
	7	Haemulon melanurum cottonwick grunt yellow goatfish
	9	Acanthurus chirurgus doctorfish yellow goatfish
	11	Mulloidichthys martinicus yellow goatfish
	14	Anisotremus virginicus porkfish yellow goatfish
	17	Haemulon sciurus bluestriped grunt yellow goatfish
	20	Chromis scotti purple reeffish yellow goatfish
	22	Chromis cyanea blue chromis yellow goatfish

12- <i>Chaetodon capistratus</i> four-eyed butterfly	Code	Species
	2	<i>Chaetodon ocellatus</i> spotfin butterfly
	12	<i>Chaetodon capistratus</i> four-eyed butterfly
	15	<i>Pomacanthus paru</i> French angelfish
	22	<i>Chromis cyanea</i> blue chromis
	23	<i>Chaetodon sedentarius</i> reef butterflyfish
	24	<i>Chaetodon striatus</i> banded butterflyfish
13- <i>Bodianus rufus</i> Spanish hogfish	Code	Species
	3	<i>Abudefduf saxatilis</i> Sergeant major
	4	<i>Bodianus puchellus</i> Cuban hogfish
	7	<i>Haemulon melanurum</i> cottonwick grunt
	13	<i>Bodianus rufus</i> Spanish hogfish
	17	<i>Haemulon sciurus</i> bluestriped grunt
	19	<i>Serranus tortugarum</i> chalk bass
	20	<i>Chromis scotti</i> purple reeffish
	22	<i>Chromis cyanea</i> blue chromis
14- <i>Anisotremus virginicus</i> porkfish	Code	Species
	4	<i>Bodianus puchellus</i> Cuban hogfish
	5	<i>Haemulon flavolineatum</i> French grunt
	9	<i>Acanthurus chirurgus</i> doctorfish
	10	<i>Acanthurus coeruleus</i> Blue Tang
	14	<i>Anisotremus virginicus</i> porkfish
	17	<i>Haemulon sciurus</i> bluestriped grunt
	20	<i>Chromis scotti</i> purple reeffish
	21	<i>Hypoplectrus indigo</i> indigo hamlet

15- Pomacanthus paru French angelfish	Code	Species
	2	Chaetodon ocellatus spotfin butterfly
	4	Bodianus puchellus Cuban hogfish
	6	Pseudupeneus maculatus spotted goatfish
	12	Chaetodon capistratus four-eyed butterfly
	15	Pomacanthus paru French angelfish
	16	Holacanthus ciliaris queen angel
	18	Selene vomer lookdown
	22	Chromis cyanea blue chromis
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish
16- Holacanthus ciliaris queen angel	Code	Species
	4	Bodianus puchellus Cuban hogfish
	5	Haemulon flavolineatum French grunt
	9	Acanthurus chirurgus doctorfish
	15	Pomacanthus paru French angelfish
	16	Holacanthus ciliaris queen angel
	19	Serranus tortugarum chalk bass
	22	Chromis cyanea blue chromis
	24	Chaetodon striatus banded butterflyfish

17- Haemulon sciurus bluestriped grunt	Code	Species
	3	Abudefduf saxatilis Sergeant major
	5	Haemulon flavolineatum French grunt
	7	Haemulon melanurum cottonwick grunt
	10	Acanthurus coeruleus Blue Tang
	11	Mulloidichthys martinicus
	13	Bodianus rufus Spanish hogfish
	14	Anisotremus virginicus porkfish
	17	Haemulon sciurus bluestriped grunt
	20	Chromis scotti purple reeffish

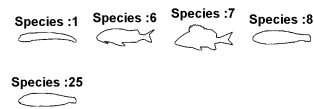
 19-Serranus tortugaru

18- Selene vomer lookdown	Code	Species
	12	Chaetodon capistratus four-eyed butterfly
	15	Pomacanthus paru French angelfish
	16	Holacanthus ciliaris queen angel
	18	Selene vomer lookdown

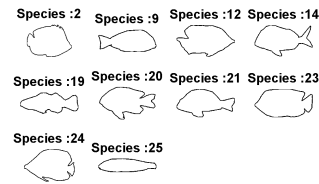
20-Chromis scotti purple reeffish	Code	Species
	3	Abudefduf saxatilis Sergeant major
	11	Mulloidichthys martinicus
	13	Bodianus rufus Spanish hogfish
	14	Anisotremus virginicus porkfish
	16	Holacanthus ciliaris queen angel
	17	Haemulon sciurus bluestriped grunt
	19	Serranus tortugarum chalk bass
	20	Chromis scotti purple reeffish
	22	Chromis cyanea blue chromis
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish
21-Hypoplectrus indigo indigo hamlet	Code	Species
	7	Haemulon melanurum cottonwick grunt
	9	Acanthurus chirurgus doctorfish
	10	Acanthurus coeruleus Blue Tang
	14	Anisotremus virginicus porkfish
	19	Serranus tortugarum chalk bass
	21	Hypoplectrus indigo indigo hamlet
	25	Halichoeres garnoti yellowhead wrasse
22-Chromis cyanea blue chromis	Code	Species
	7	Haemulon melanurum cottonwick grunt
	11	Mulloidichthys martinicus
	13	Bodianus rufus Spanish hogfish
	17	Haemulon sciurus bluestriped grunt
	22	Chromis cyanea blue chromis
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish

23-Chaetodon sedentarius reef butterflyfish	Code	Species
	2	Chaetodon ocellatus spotfin butterfly
	3	Abudefduf saxatilis Sergeant major
	7	Haemulon melanurum cottonwick grunt
	10	Acanthurus coeruleus Blue Tang
	12	Chaetodon capistratus four-eyed butterfly
	21	Hypoplectrus indigo indigo hamlet
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish
24-Chaetodon striatus banded butterflyfish	Code	Species
	2	Chaetodon ocellatus spotfin butterfly
	12	Chaetodon capistratus four-eyed butterfly
	15	Pomacanthus paru French angelfish
	23	Chaetodon sedentarius reef butterflyfish
	24	Chaetodon striatus banded butterflyfish
25-Halichoeres garnoti yellowhead wrasse	Code	Species
	2	Chaetodon ocellatus spotfin butterfly
	6	Pseudupeneus maculatus spotted goatfish
	7	Haemulon melanurum cottonwick grunt
	8	Halichoeres bivittatus Slippery Dick
	19	Serranus tortugarum chalk bass
	21	Hypoplectrus indigo indigo hamlet
	25	Halichoeres garnoti yellowhead wrasse

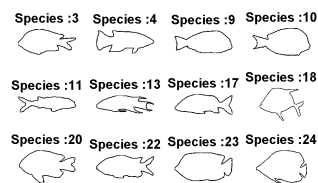
From the tables we observe that species of same family always appears in the top 50 matches indicating the strong connection between fish shape and fish family. To demonstrate this in a quantitative way we can plot a histogram of these 50 species which is shown in Figures 5.26, 5.27 and 5.28. In these Figures we can observe large



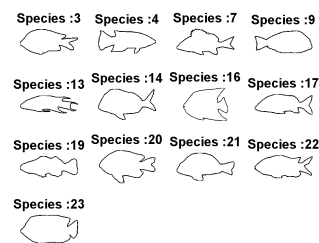
(a) Species Code: 1.



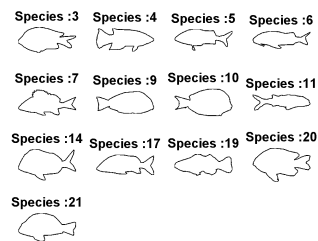
(b) Species Code: 2.



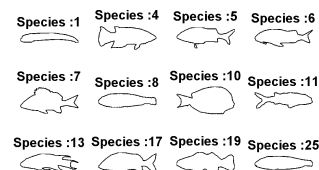
(c) Species Code: 3.



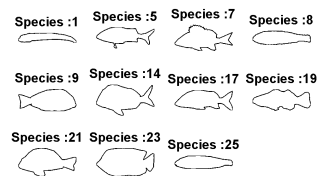
(d) Species Code: 4.



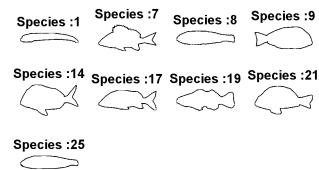
(e) Species Code: 5.



(f) Species Code: 6.

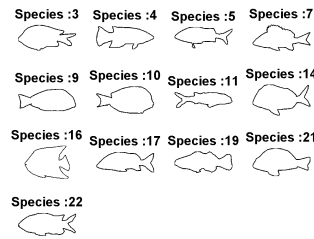


(g) Species Code: 7.

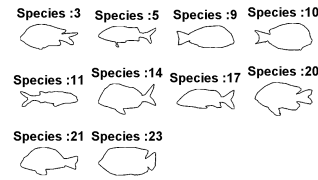


(h) Species Code: 8.

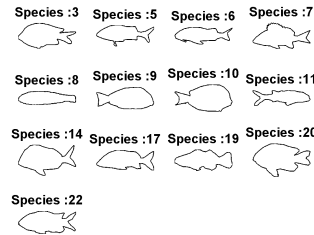
Figure 5.23: Different Species found in top 5 matches using shape for each of the species 1-8.



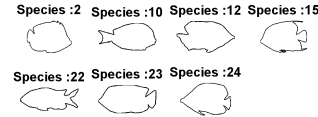
(a) Species Code: 9.



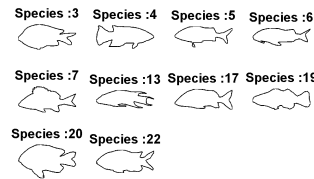
(b) Species Code: 10.



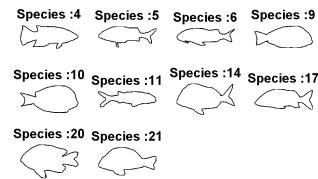
(c) Species Code: 11.



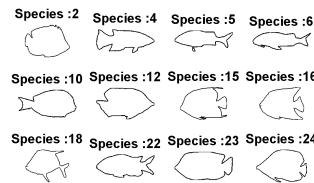
(d) Species Code: 12.



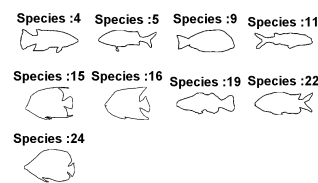
(e) Species Code: 13.



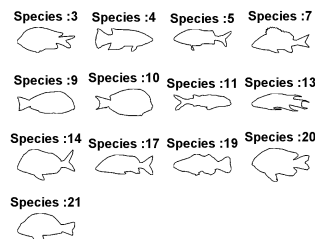
(f) Species Code: 14.



(g) Species Code: 15.

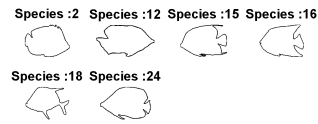


(h) Species Code: 16.

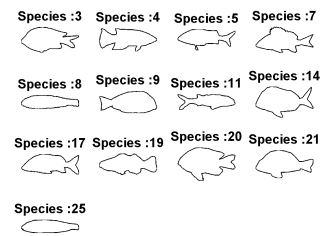


(i) Species Code: 17.

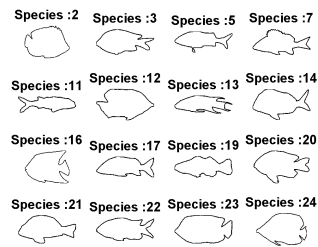
Figure 5.24: Different Species found in top 5 matches using shape for each of the species 9-17.



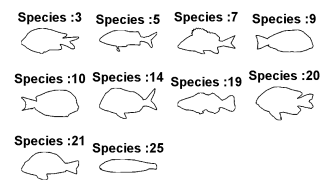
(a) Species Code: 18.



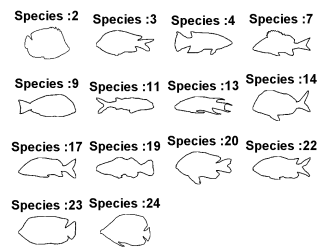
(b) Species Code: 19.



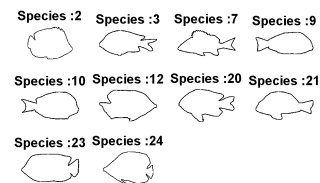
(c) Species Code: 20.



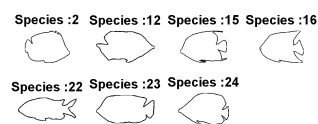
(d) Species Code: 21.



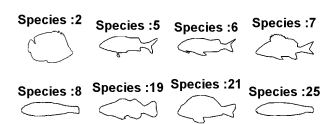
(e) Species Code: 22.



(f) Species Code: 23.

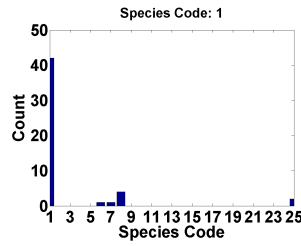


(g) Species Code: 24.

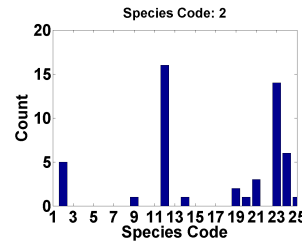


(h) Species Code: 25.

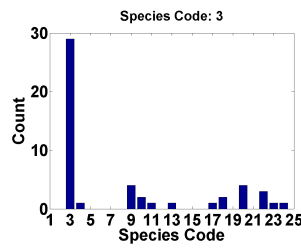
Figure 5.25: Different Species found in top 5 matches using shape for each of the species 18-25.



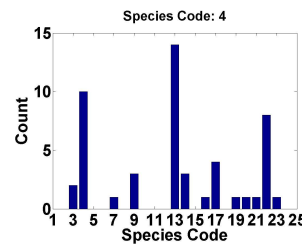
(a) Histogram for species code 1. $\hat{p}_C = 0.9$, $\hat{p}_F = 0$ and $\hat{p}_I = 0.1$.



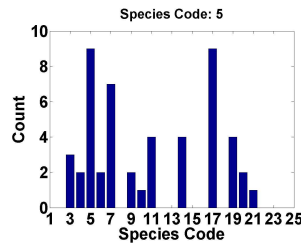
(b) Histogram for species code 2. $\hat{p}_C = 0.2$, $\hat{p}_F = 0.7$ and $\hat{p}_I = 0.1$.



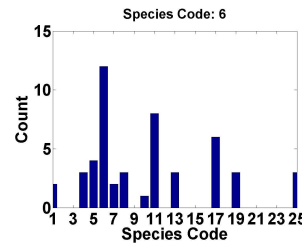
(c) Histogram for species code 3. $\hat{p}_C = 0.7$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.2$.



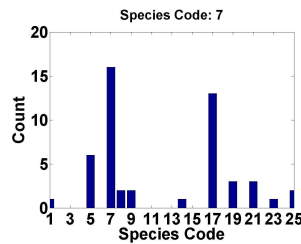
(d) Histogram for species code 4. $\hat{p}_C = 0.7$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.2$.



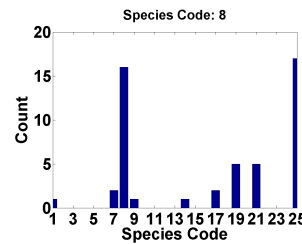
(e) Histogram for species code 5. $\hat{p}_C = 0.4$, $\hat{p}_F = 0.2$ and $\hat{p}_I = 0.4$.



(f) Histogram for species code 6. $\hat{p}_C = 0$, $\hat{p}_F = 0.4$ and $\hat{p}_I = 0.6$.

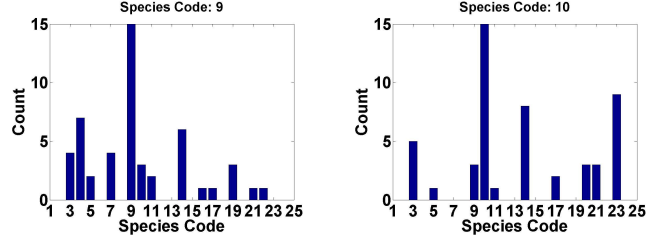


(g) Histogram for species code 7. $\hat{p}_C = 0.3$, $\hat{p}_F = 0.4$ and $\hat{p}_I = 0.3$.

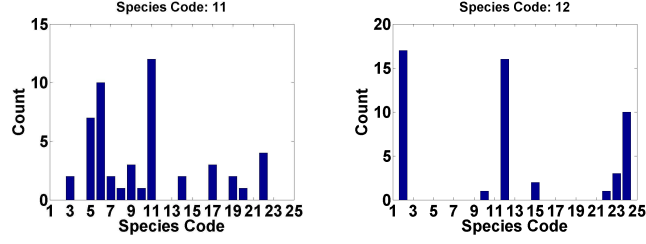


(h) Histogram for species code 8. $\hat{p}_C = 0.4$, $\hat{p}_F = 0.2$ and $\hat{p}_I = 0.4$.

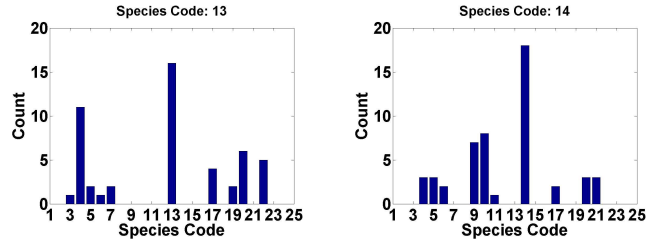
Figure 5.26: Histogram of top 5 matches for each example of a species 1-8.



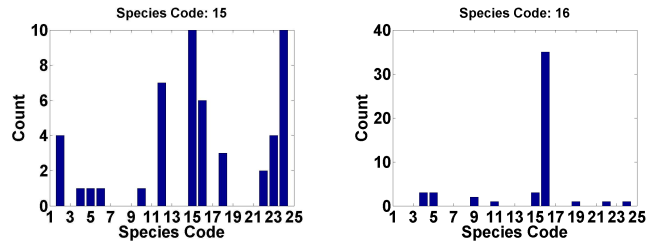
(a) Histogram for species code 9. $\hat{p}_C = 0.5$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.4$. (b) Histogram for species code 10. $\hat{p}_C = 0.5$, $\hat{p}_F = 0$ and $\hat{p}_I = 0.5$.



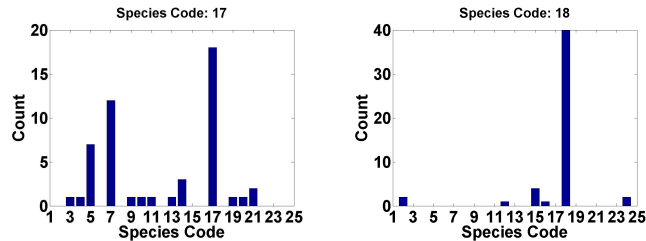
(c) Histogram for species code 11. $\hat{p}_C = 0.3$, $\hat{p}_F = 0.2$ and $\hat{p}_I = 0.5$. (d) Histogram for species code 12. $\hat{p}_C = 0.2$, $\hat{p}_F = 0.8$ and $\hat{p}_I = 0$.



(e) Histogram for species code 13. $\hat{p}_C = 0.6$, $\hat{p}_F = 0.2$ and $\hat{p}_I = 0.2$. (f) Histogram for species code 14. $\hat{p}_C = 0.5$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.4$.

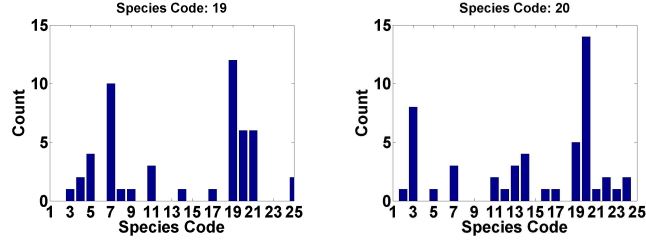


(g) Histogram for species code 15. $\hat{p}_C = 0.3$, $\hat{p}_F = 0$ and $\hat{p}_I = 0.7$. (h) Histogram for species code 16. $\hat{p}_C = 0.8$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.1$.

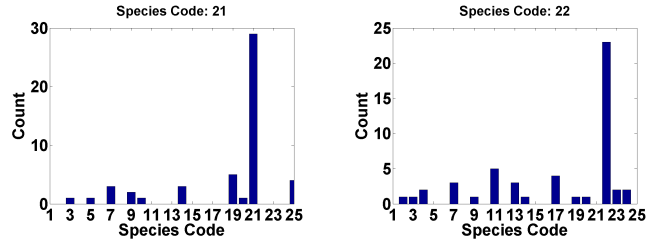


(i) Histogram for species code 17. $\hat{p}_C = 0.5$, $\hat{p}_F = 0.5$ and $\hat{p}_I = 0$. (j) Histogram for species code 18. $\hat{p}_C = 1$, $\hat{p}_F = 0$ and $\hat{p}_I = 0$.

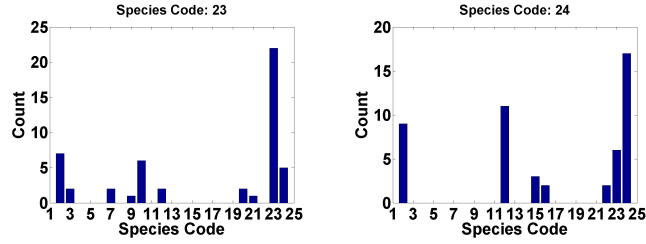
Figure 5.27: Histogram of top 5 matches for each example of a species 9-18.



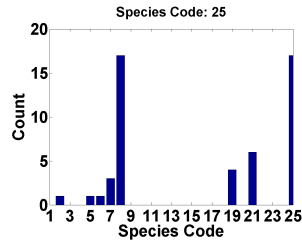
(a) Histogram for species code 19. $\hat{p}_C = 0.4$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.5$. (b) Histogram for species code 20. $\hat{p}_C = 0.4$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0.5$.



(c) Histogram for species code 21. $\hat{p}_C = 0.7$, $\hat{p}_F = 0$ and $\hat{p}_I = 0.3$. (d) Histogram for species code 22. $\hat{p}_C = 0.7$, $\hat{p}_F = 0$ and $\hat{p}_I = 0.3$.



(e) Histogram for species code 23. $\hat{p}_C = 0.6$, $\hat{p}_F = 0.2$ and $\hat{p}_I = 0.2$. (f) Histogram for species code 24. $\hat{p}_C = 0.6$, $\hat{p}_F = 0.4$ and $\hat{p}_I = 0$.



(g) Histogram for species code 25. $\hat{p}_C = 0.9$, $\hat{p}_F = 0.1$ and $\hat{p}_I = 0$.

Figure 5.28: Histogram of top 5 matches for each example of a species 19-25.

peaks for fish species belonging to the same family. \hat{p}_C , \hat{p}_F and \hat{p}_I are the estimated probabilities of the species belonging to the correct species, correct family only and incorrect family respectively. If a species has only a single large peak then we can reliably identify the exact species using shape for that particular species (e.g. See plots of species 1 and 3). In cases where there are two or three peaks, we see that these peaks frequently correspond, to those species which belong to the same fish family as the test species (e.g. See plots of species 2,5,6, 16 and 18). We can use this information to identify the common family of the fish. Figures 5.23, 5.24 and 5.25 show the shapes of different species which were observed in the 50 matches for the examples of each species.

To demonstrate the shape matching results we randomly pick one example for each species from the database, train on the rest and find out the top 9 best matches with the lowest score from the remaining 249 shapes. In all we choose 25 examples, one for each species. In Figures 5.29 and 5.30 for each row the leftmost shape is the picked shape and the following 9 shapes from left to right are the shapes with increasing matching score(smaller score indicates better matching). The similarity between the test shape and examples shapes is captured fairly using this technique³.

To further demonstrate our capability to analyze shapes, we test our shape analysis technique on a test example obtained from sequence 1(taken from the New York Aquarium), after post-processing. The Figure 5.31 shows our obtained results. Figure 5.31(a) shows the test image along with the top 9 lowest score matches, species code and score indicated on the top of every shape. Figure 5.31(b) is a histogram plot of the top 10 species obtained on the test image. We can clearly see that the highest peak corresponds the correct species (Species Code = 7, Cottonwick Grunt).

From the above results above we can fairly conclude that using shape is a strong cue for initial classification into similar shapes from the same family. In our future work we will focus on narrowing down these groups and further use size, color, shape and patterns wherever necessary to pinpoint the exact species.

³See Appendix D for the confusion matrix for best ONE match for 10 examples for each species

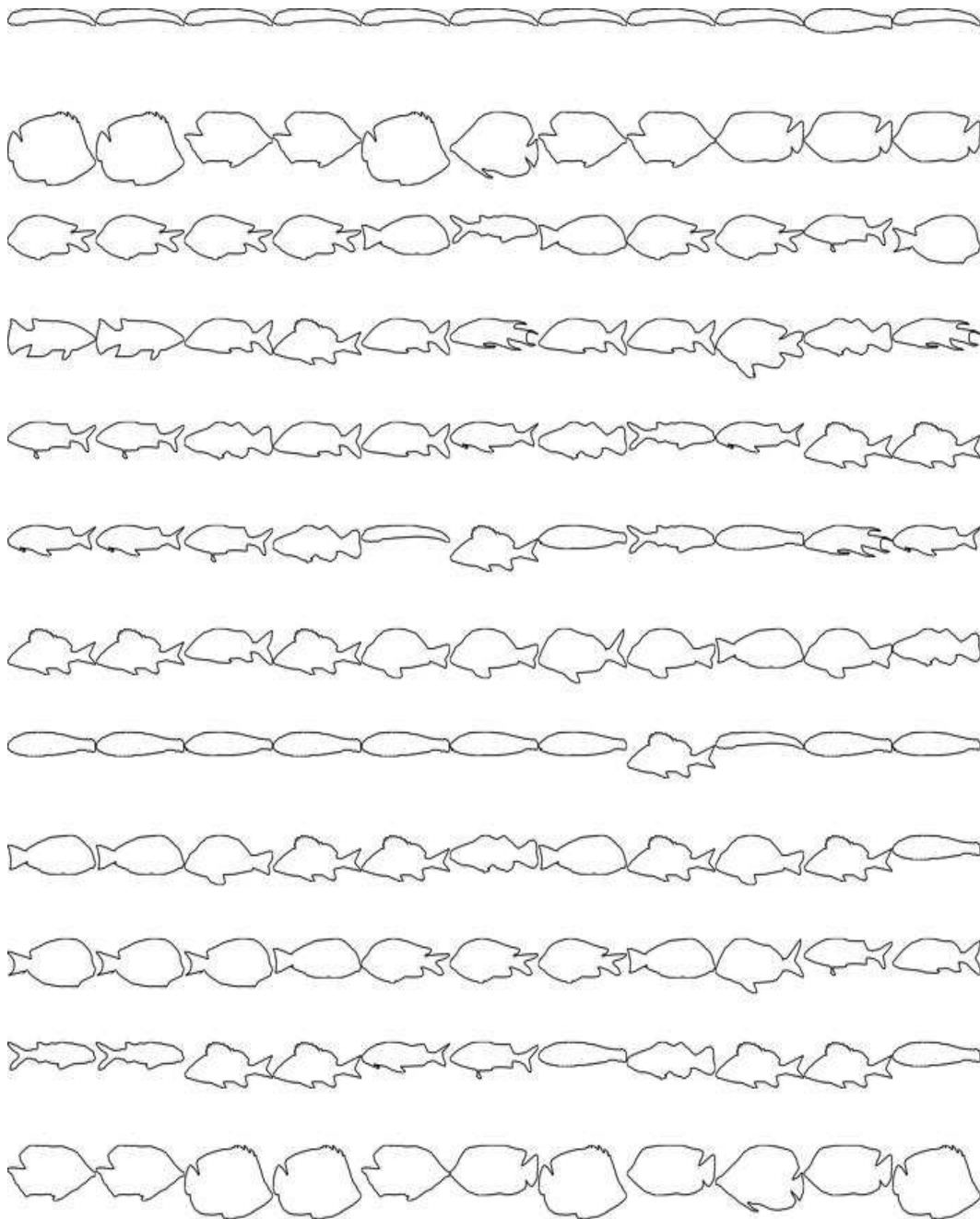


Figure 5.29: Similar shapes for an example fish of each species from the database (Species 13-25). The leftmost in every row is the test shape and the following 10 shapes in the same row are the matches with lowest scores.

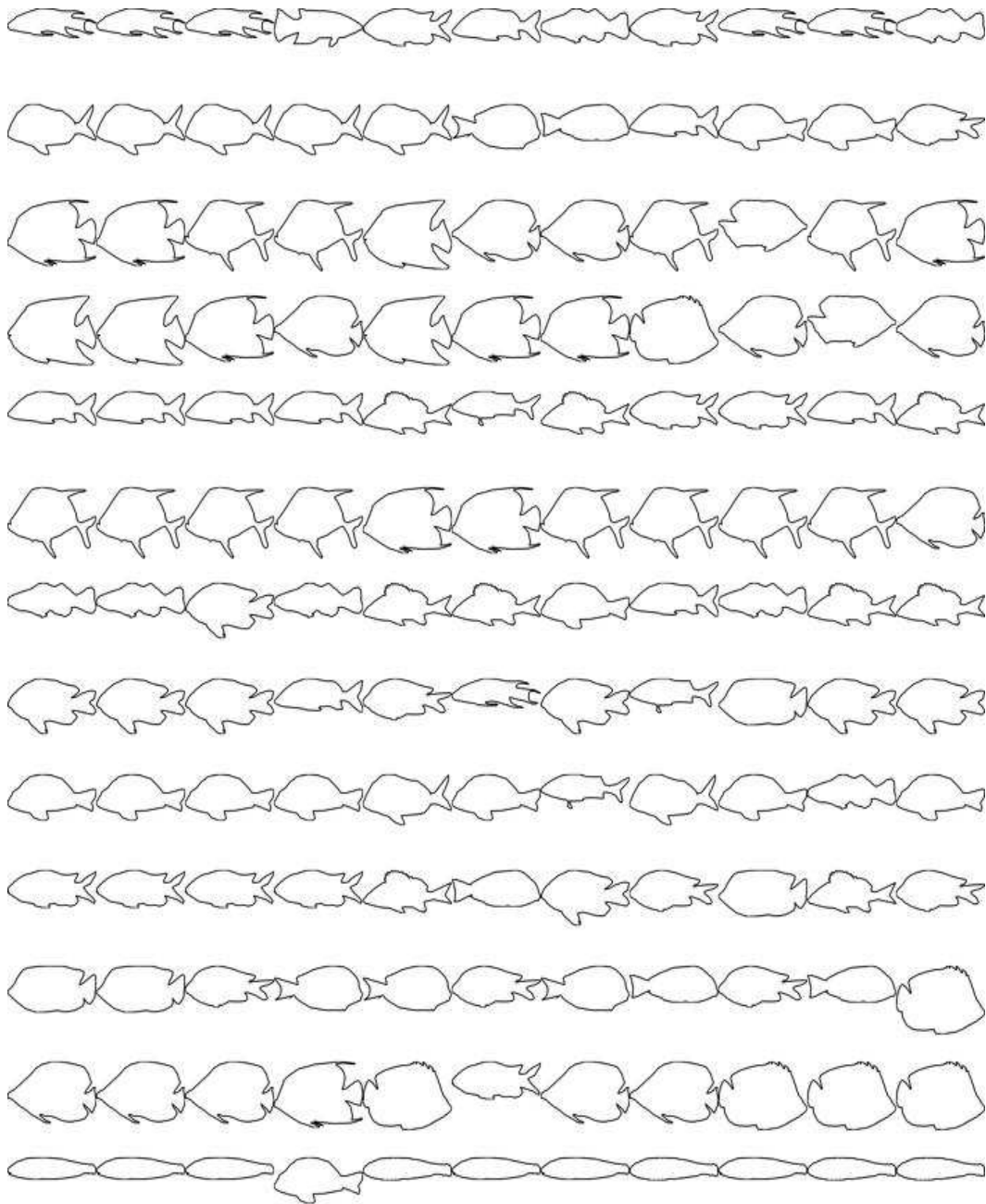
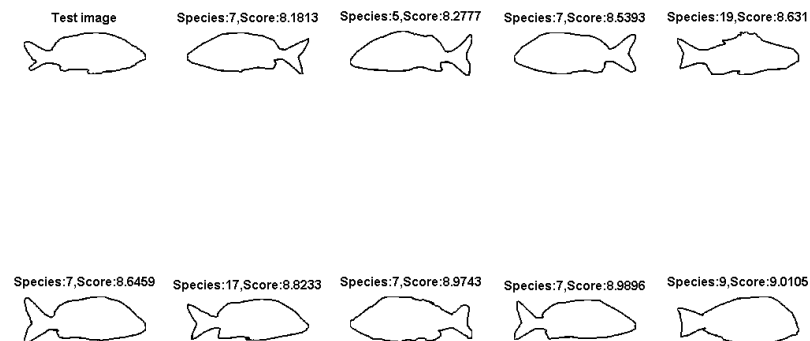
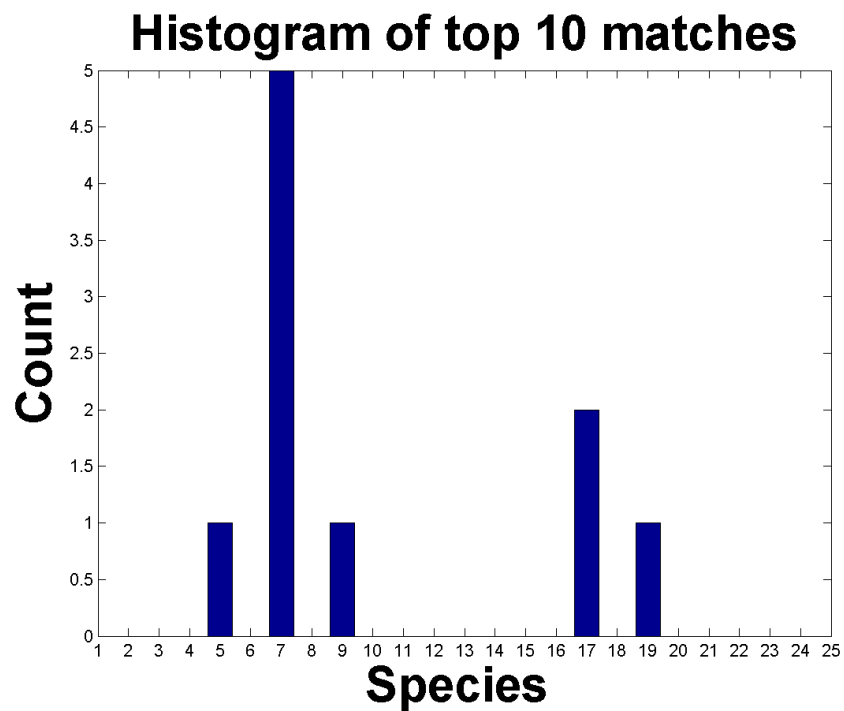


Figure 5.30: Similar shapes for an example fish of each species from the database (Species 13-25). The leftmost in every row is the test shape and the following 10 shapes in the same row are the matches with lowest scores.



(a) Best match species for test image 1(top-left) from from sequence 1.



(b) Histogram of top 10 best matches for test image 1. Ground truth species = 7.

Figure 5.31: Test results for matching test fish image 1 with the database.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The goal of our project is to build an automatic, real-time identification and monitoring system for reef fish. We have made significant progress since we started working on this project. We developed an image capture system that could capture high resolution (1360 x 1024 pixel) images from two cameras simultaneously, at configurable frame rates from 0 to 20 frames per sec. The capture system is optimized to have accurate frame rates and minimum inter image pair delay (± 1 ms). This was achieved using the computers internal timer clock.

We successfully calibrated the cameras to give better focus underwater by doing experiments in our lab with our custom built Remote Focus Control System. We obtained sharp camera images with good illumination in the New York Aquarium installation. In the New York aquarium, where the system is just few feet below the surface of the water, the LED lights in our prototype system made only a minor contribution to the overall illumination, as there was ample ambient sunlight entering from the outside in the camera field of view (FOV). We used a gain of 15 on both cameras. In the future, when we install and test the system at greater depths near the coral reef in Belize, we will need these LED lights and/or need to increase the gain (max. 27) of both cameras, if required, for larger dynamic intensity range and better contrast.

We developed a robust background modeling technique which can model the background and detect foreground moving objects (i.e. moving fish) in the scene and localize them accurately. The background modeling technique also detects floating debris which is eliminated based on size. The technique fails to eliminate debris which appears close to the camera because they appear larger and hence cannot be eliminated on size, but

elimination can be done in the recognition phase, based on shape and color information. This still remains to be done. Overall, the background modeling technique performed quite well and gave smooth and close to accurate shapes due to the added smoothness prior. For finer details like fins and tail of the fish, we performed local re-segmentation using local background information inside the bounding box and were able to obtain better, finer shape details for better shape features.

We developed an extension of a tracking algorithm to track foreground moving blobs (i.e moving fish) which could be used to track foreground objects with time and avoid multiple counts. The extensions of the tracking algorithm makes it adaptive, capable of handling multiple moving foreground blobs simultaneously and detect overlap among blobs. The selection of the best view for a fish from either of the views is done by selecting the larger horizontal major axis of the fish from both views. We still need to find out a better more robust criterion to do this which takes actual 3D pose of the fish into account.

Using the initial shape based analysis, we demonstrated that we can recognize fish species and/or fish families from shape features and use this as an initial stage for pinpointing the exact species. Further use of size, color and pattern features will do this. We are currently actively working on this task.

We have made significant progress since I started working on this project. The project began with mainly three modules which were Image Capture, Background Modelling and Pose Correction. These were developed by graduate student Ning Huang (Image Capture and Background Modelling) and three undergraduates Lev Barinov, Jigesh Baxi and James Bibby (Pose Correction and Epipolar Geometry). My main contributions to the project were Camera Calibration, System Integration, Tracking and shape based Recognition.

Below is a detailed list of my contributions to the project:

1. Built a remote focus control system for underwater camera calibration.
2. Performed tests for underwater camera calibration.
3. Helped in testing the prototype system at IMCS.

4. Helped in installing the prototype system at the Glovers Reef exhibit at the New York Aquarium.
5. Developed a software for data collection of image sequences at the New York Aquarium for pre-processing.
6. Worked on the Background Modeling software and made it faster by using lower resolution images for modelling
7. Worked on the Image Capture program to reduce timing delay between image pairs.
8. Developed a tracking algorithm to track observed fish over time. The algorithm is adaptive, capable of handling multiple targets simultaneously and detecting overlap between blobs.
9. Tried various approaches for fish shape analysis and applied a shape analysis technique (using PCA for the reduced representation of perimeter points) to classify fish species and/or fish families.
10. Built a Graphical User Interface (GUI) for the system and system integration (See Appendix B).

6.2 Future Work

The future work for the project involves:

- Expanding the database to include more fish species and using more features which will aid in uniquely classifying fish species.
- Developing a robust algorithm for best view selection to further improve fish species classification.
- Carrying out extensive testing of the system in New York Aquarium and incorporating fish behavior studies to further improve the system design.
- Putting the system to test at the coral reef in Belize and gathering feedback.

- Integrate the final different modules to make a completely automated near real-time system for fish monitoring and identification.
- Place many such systems at different locations along the coral reef to gather fish population data overtime.
- Design a communication system to gather all such data from these systems and use it to monitor reef health and fish populations.

References

- [1] Jean-Yves Bouguet, Camera Calibration Toolbox for Matlab, available at http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [2] A. Elgammal, D. Harwood, L. S. Davis, Non-parametric Model for Background Subtraction, IEEE ICCV99 Frame Rate Workshop. IEEE 7th International Conference on Computer Vision. Kerkyra, Greece, September 1999.
- [3] Q Wang, G Xu, H Ai, Learning object intrinsic structure for robust visual tracking - 2003 IEEE Computer Society Conference on Computer, 2003.
- [4] Delanges, P., Benois, J., Barba, D., Active Contours Approach to Object Tracking in Image Sequences With Complex Background, PRL(16), No. 2, February 1995, pp. 171-178. BibRef 9502.
- [5] R. Halif and J. Flusser, Numerically Stable Direct Least Squares Fitting of Ellipses, Department of Software Engineering, Charles University, Czech Republic, 2000.
- [6] C. Stauffer and W. Grimson, Learning Patterns of Activity using Real-time Tracking, TPAMI, 2000.
- [7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts vol. 11 p. 12221239, November 2001.
- [8] Castignolles Nathalie, Cattoen Michel, Larinier M., Identification and counting of live fish by imageanalysis, Proc. SPIE Vol. 2182, p. 200-209, Image and Video Processing II, Sarah A. Rajala; Robert L. Stevenson; Eds.
- [9] Martinez-de Dios J. R. and Serna C. and Ollero A. Computer Vision and robotics techniques in fish farms, Robotica Vol 21, p. 233-243, Cambridge University Press, New York, NY, USA.
- [10] David G. Lowe, Object recognition from local scale-invariant features, International Conference on Computer Vision, Corfu, Greece (September 1999), pp. 1150-1157.
- [11] Viola, P. and Jones, M.J., Robust Real Time Object Detection, SCTV01, 2001.
- [12] Wolfson, H.J. and Rigoutsos, I (1997). Geometric Hashing: An Overview, IEEE Computational Science and Engineering, vol. 4(4), p. 10-21.

- [13] Lee, C.S. and Elgammal, A.M., Modeling View and Posture Manifolds for Tracking, International Conference on Computer Vision, 2007,p. 1-8.
- [14] Torresani, L. and Yang, D.B. and Alexander, E.J. and Bregler, C., Tracking and Modeling Non-Rigid Objects with Rank Constraints, CVPR 2001,p. 493-500.
- [15] Comaniciu, D. and Ramesh, V. and Meer, P., Real-Time Tracking of Non-Rigid Objects using Mean Shift, CVPR, 2000, p. 142-149.
- [16] Andreas K., Sangkyu K, Joonki P., Bessa A., Mongi A., Color active shape models for tracking non-rigid objects, Pattern Recognition Letters, Vol. 24, Issue 11, Color Image Processing and Analysis. CGIV 2002, July 2003, p. 1751-1765.
- [17] Incremental Learning for Robust Visual Tracking, David Ross,Jongwoo L.,Ruei-Sung L.,Ming-Hsuan Y., In the International journal of Computer Vision, Special Issue: Learning for Vision, 2007.
- [18] Zhengyou Zhang, Flexible camera calibration by viewing a plane from unknown orientations, ICCV, 1999, p. 666–673
- [19] Kim, M.Y. and Kumar, S. and Pavlovic, V. and Rowley, H.A., Face tracking and recognition with visual constraints in real-world videos, CVPR 2008, p. 1-8,
- [20] The LEGO group, Website: <http://shop.lego.com/Product/?p=3804>
- [21] Wang B.H, Bangham J.A., PCA based shape descriptors for shape retrieval and the evaluations, Proceedings of 2006 International Conference on Computational Intelligence and Security, Volume 2, 2006, 1401-1406.
- [22] Mohit Gupta, Krihnan Ramnath Interactive Segmentation Toolbox, URL:<http://www.cs.cmu.edu/~mohitg/segmentation.htm>.
- [23] C. Rother, V. Kolmogorov, A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts, ACM Transactions on Graphics (SIGGRAPH'04), 2004
- [24] A. Blake, C. Rother, M. Brown, P. Perez,P. Torr. Interactive image segmentation using an adaptive GMMRF model, Proc. Eur. Conf. on Computer Vision, ECCV (2004).
- [25] M. Isard, A. Blake. Contour tracking by stochastic propagation of conditional density. Proceedings of the Fourth European Conference on Computer Vision, p. 343-356. Springer Verlag, 1996.
- [26] R. Jonker, A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, Computing, Vol. 38, pp. 325-340, 1987.
- [27] ReefNet Inc., Reef Fish Identification: Florida, Caribbean, Bahamas, Interactive 4th edition, copyright - 1998-2007. URL:www.reefnet.ca

- [28] Otsu N., A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

Appendix A

Lego Remote Focus Control System

To adjust the focus of the underwater cameras when the cameras are, indeed, underwater, a remote focus control system was built (there was not a commercially available lens with remote focussing for our wide angle lens). This system gave us the flexibility to remotely rotate the focus ring clockwise or anti-clockwise so as to achieve optimal focus manually for sharp images.

To send control signals underwater, inside the camera housing, we used the GPIO pins available on the camera. The control system needed three signals to function: rotate clockwise, rotate anti-clockwise and stop. The signals were send directly to the camera from the computer using the GigE connection which then controlled the output on the GPIO pins. A two bit code was used to send control information to the Lego system (See Table A.1).

To achieve slow and smooth control of the motor speed, on-off motor control technique was used. The pulse frequency was set to 500 Hz at 50% duty cycle (i.e. 0.25ms on-period). A block diagram is shown in Figure A.1.

The system has the following components:

- Control Input: To interface the Lego brick with the camera GPIO pins, output from the camera GPIO pins was translated from CMOS level logic to TTL level logic which the Lego system used. The schematic for this is shown in Figure

Rotation	code
clockwise	01
anti-clockwise	10
stop	00 or 11

Table A.1: Two bit code used to send control information to the Lego system.



Figure A.1: Lego Remote Focus Control block diagram.

A.2. These TTL output pins were input to Control System block which further controlled the motor.

- Control System: The control system consisted of the Lego RCX-2 brick which was programmed to take input from the Control Input system and generate the appropriate on-off signals to the motor.
- Actuators: The actuator was the Lego DC motor brick and a rubber belt drive both mounted on top of the camera. The belt transmission gripped the focus ring to rotate it.

Using the above system, experiments were conducted to calibrate the camera using a checkerboard pattern. The cameras, in their enclosures were placed, one at a time, in a six foot long fish tank along with the checkerboard pattern. Measurements were taken at three different positions corresponding to the near, middle and farthest distances found in our actual submersible system. The focus was set to be best in the approximate middle of our viewing volume.

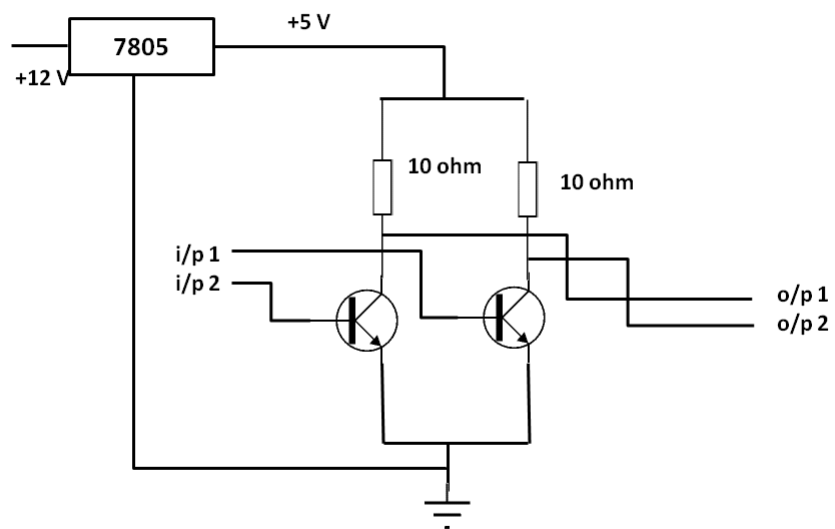


Figure A.2: Control Input block schematic.

Appendix B

Graphical User Interface-Screenshots

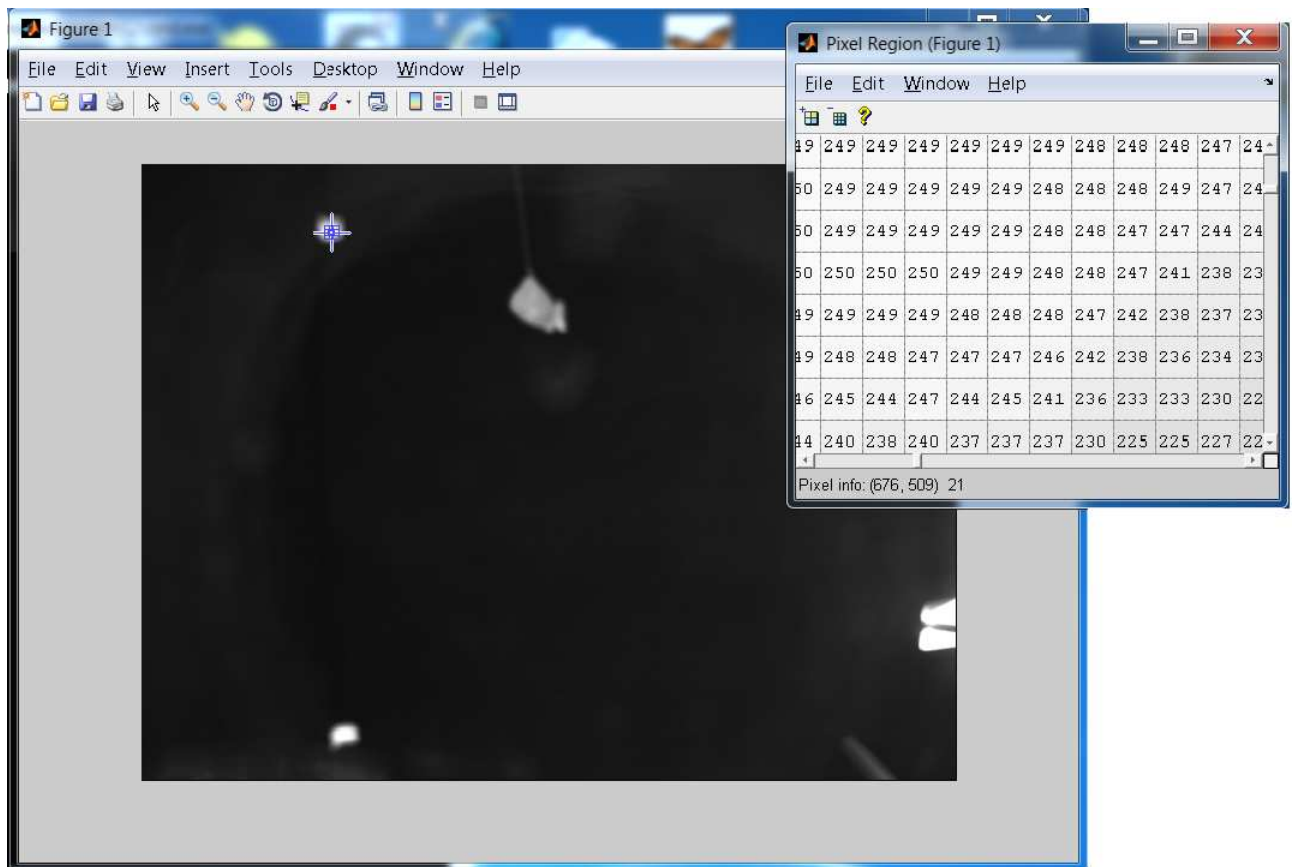


Figure B.1: Screenshot-Camera Calibration.

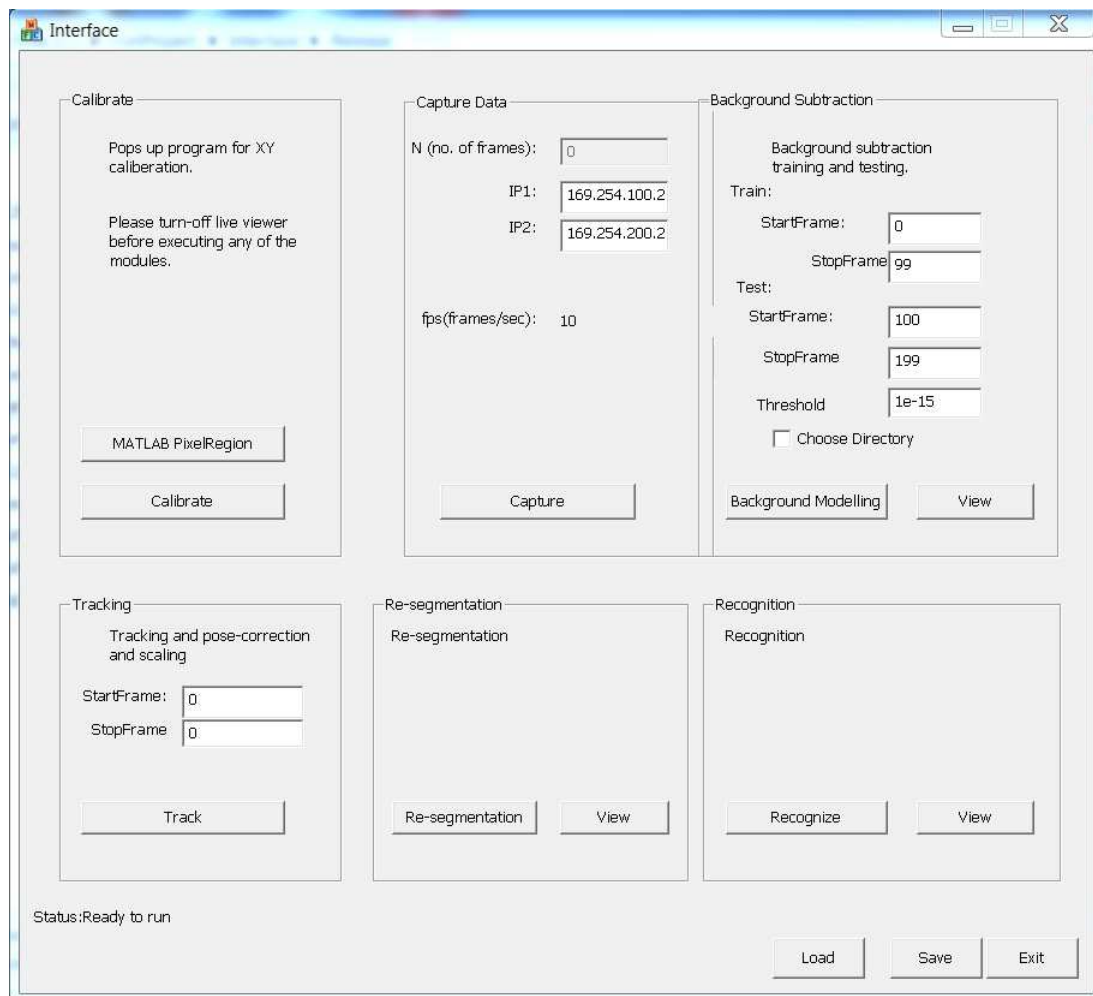


Figure B.2: Screenshot-Image Capture.

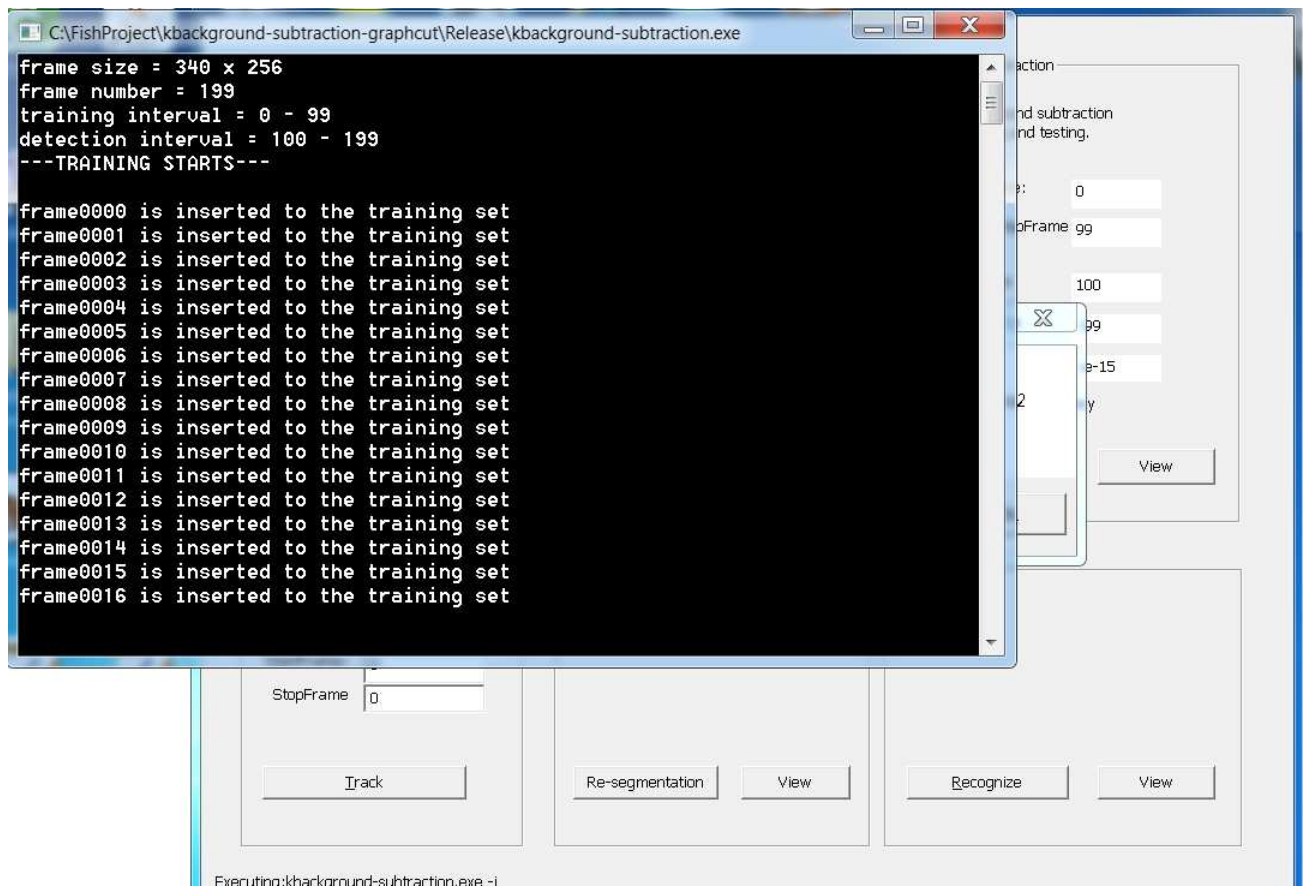


Figure B.3: Screenshot-Background Modelling.

Appendix C

Fish Species Database

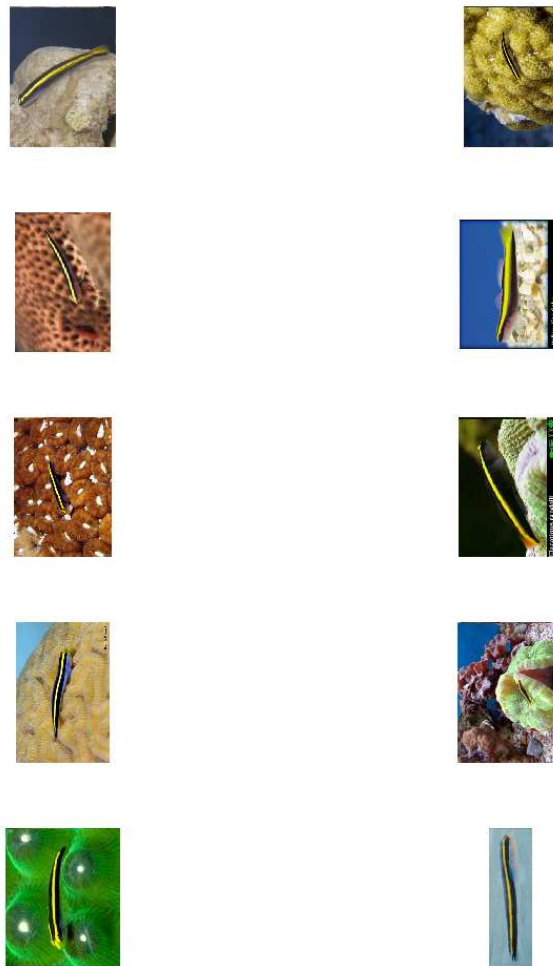


Figure C.1: Species Code:1 Elacatinus randalli yellownose goby.



Figure C.2: Species Code:2 Chaetodon ocellatus spotfin butterfly.



Figure C.3: Species Code:3 Abudefduf saxatilis Sergeant major.



Figure C.4: Species Code:4 Bodianus puchellus Cuban hogfish.

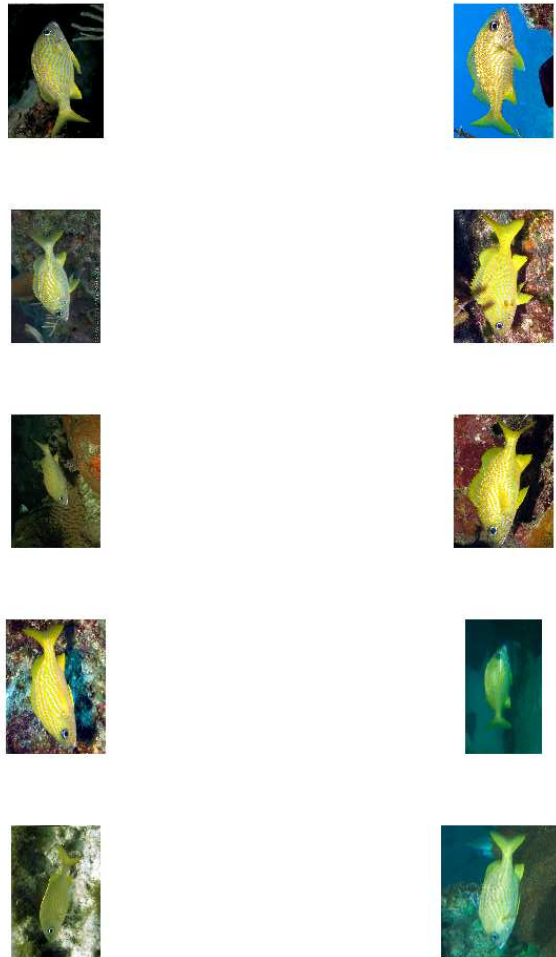


Figure C.5: Species Code:5 Haemulon flavolineatum French grunt.

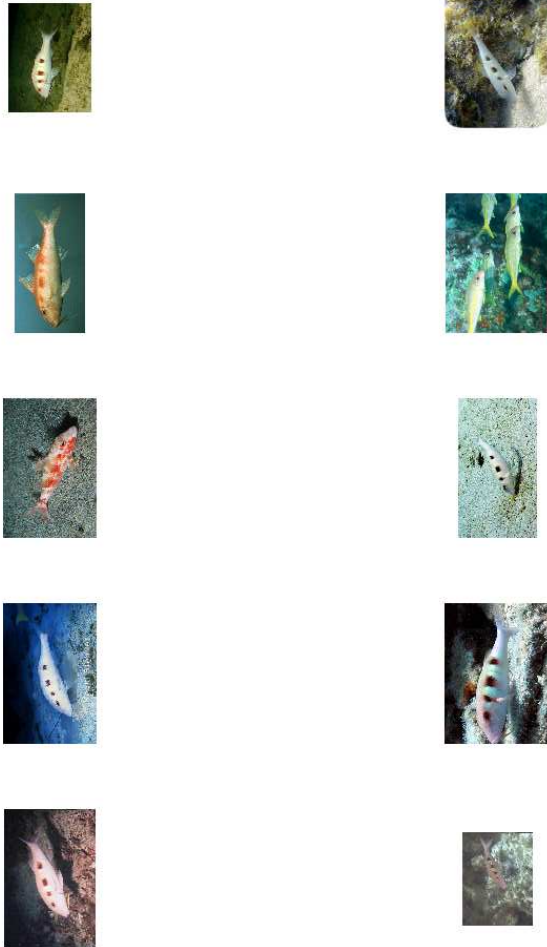


Figure C.6: Species Code:6 *Pseudupeneus maculatus* spotted goatfish.

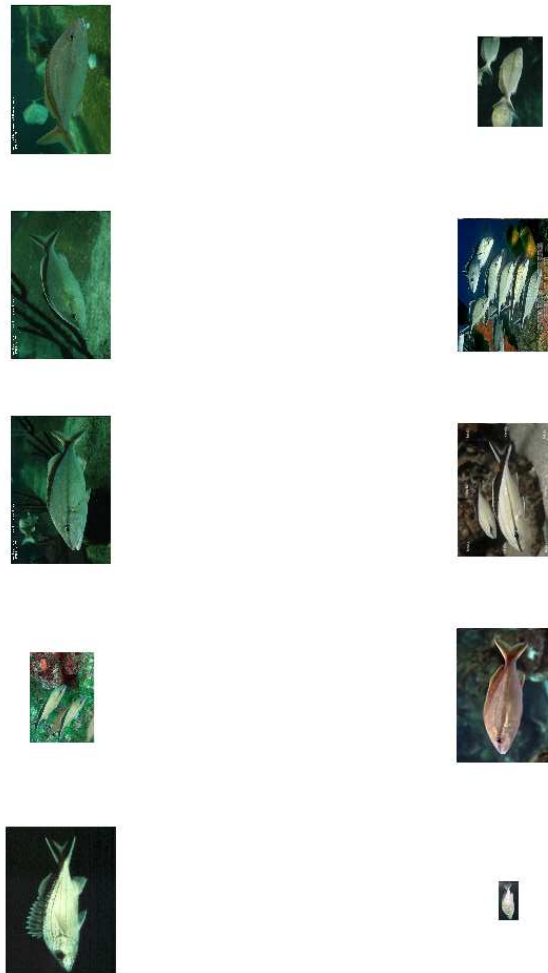


Figure C.7: Species Code:7 Haemulon melanurum cottonwick grunt.



Figure C.8: Species Code:8 *Halichoeres bivittatus* Slippery Dick.



Figure C.9: Species Code:9 Acanthurus chirurgus doctorfish.

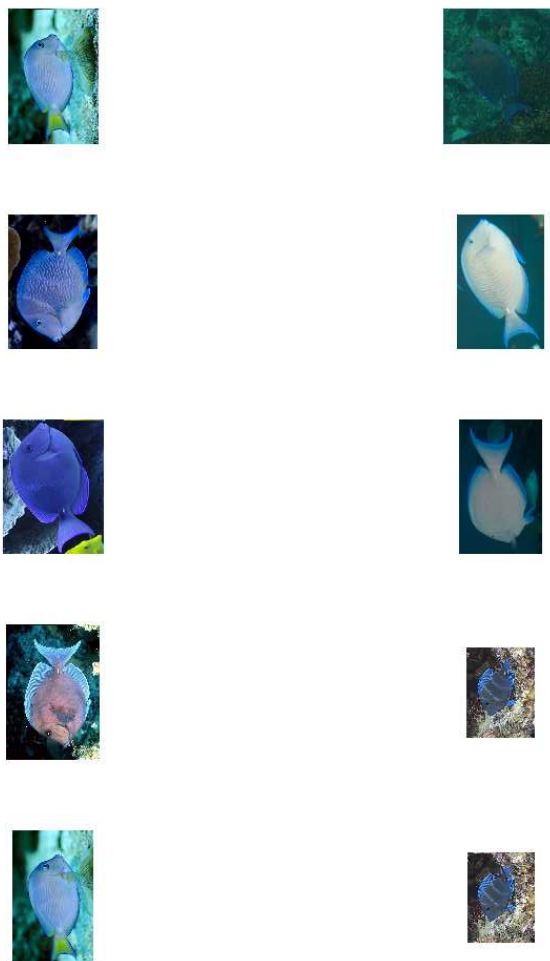


Figure C.10: Species Code:10 *Acanthurus coeruleus* Blue Tang.

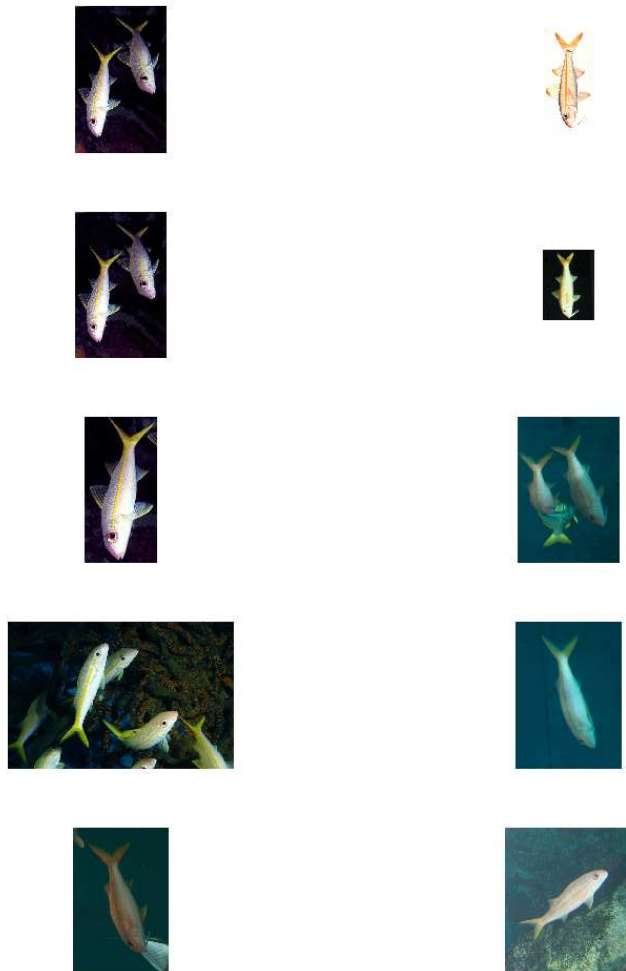


Figure C.11: Species Code: 11 Mulloidichthys martinicus yellow goatfish.

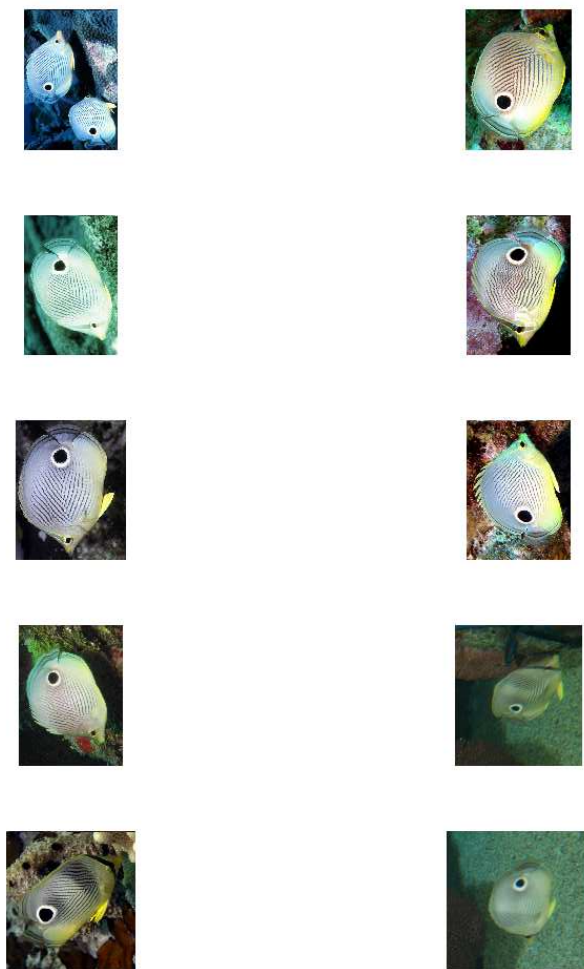


Figure C.12: Species Code:12 *Chaetodon capistratus* four-eyed butterfly.

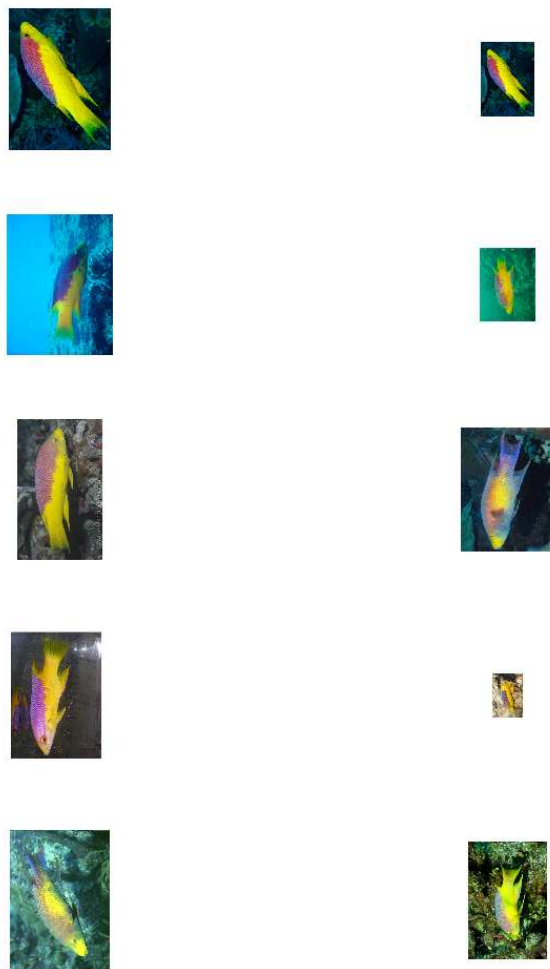


Figure C.13: Species Code:13 *Bodianus rufus* Spanish hogfish.



Figure C.14: Species Code:14 *Anisotremus virginicus* porkfish.

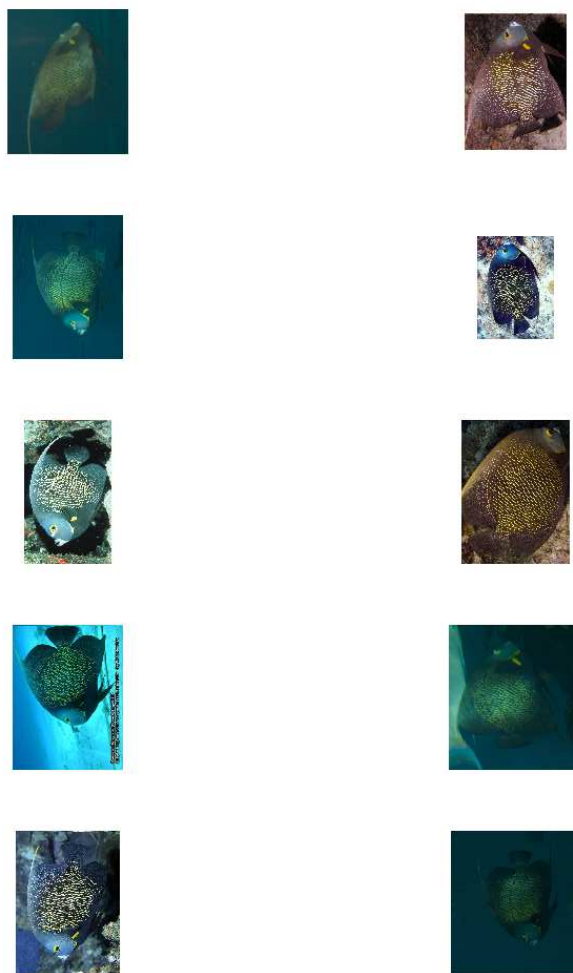


Figure C.15: Species Code:15 Pomacanthus paru French angelfish.

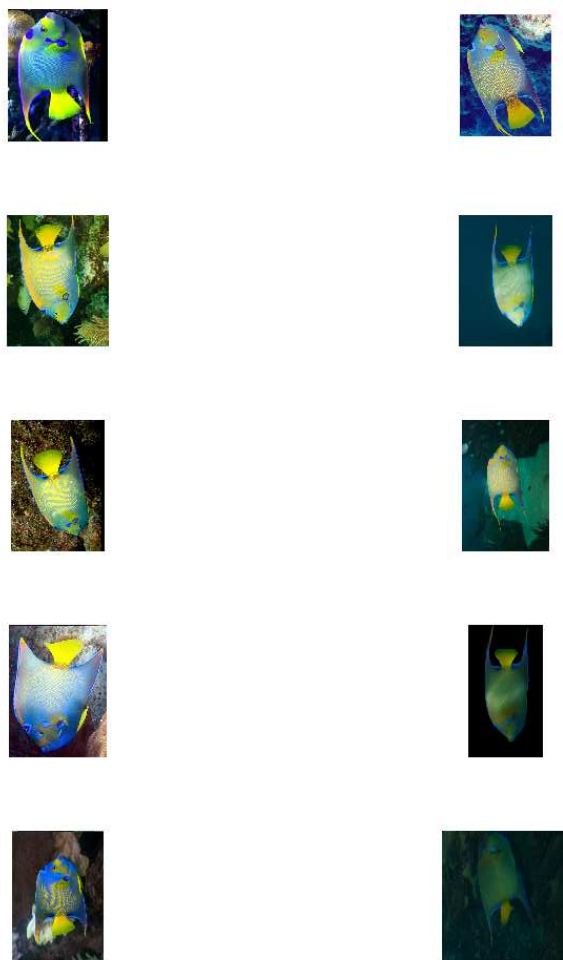


Figure C.16: Species Code:16 Holacanthus ciliaris queen angel.

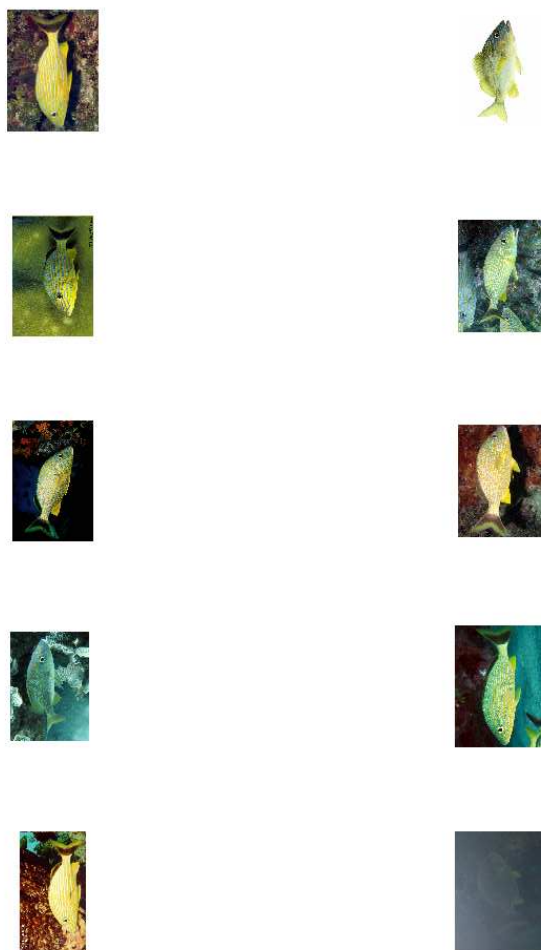


Figure C.17: Species Code:17 *Haemulon sciurus* bluestriped grunt.

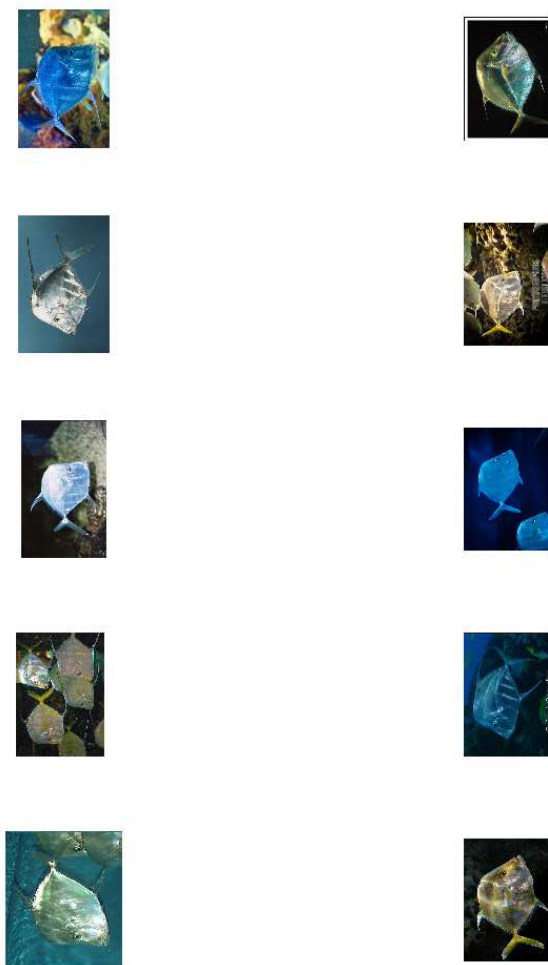


Figure C.18: Species Code:18 Selene vomer lockdown.

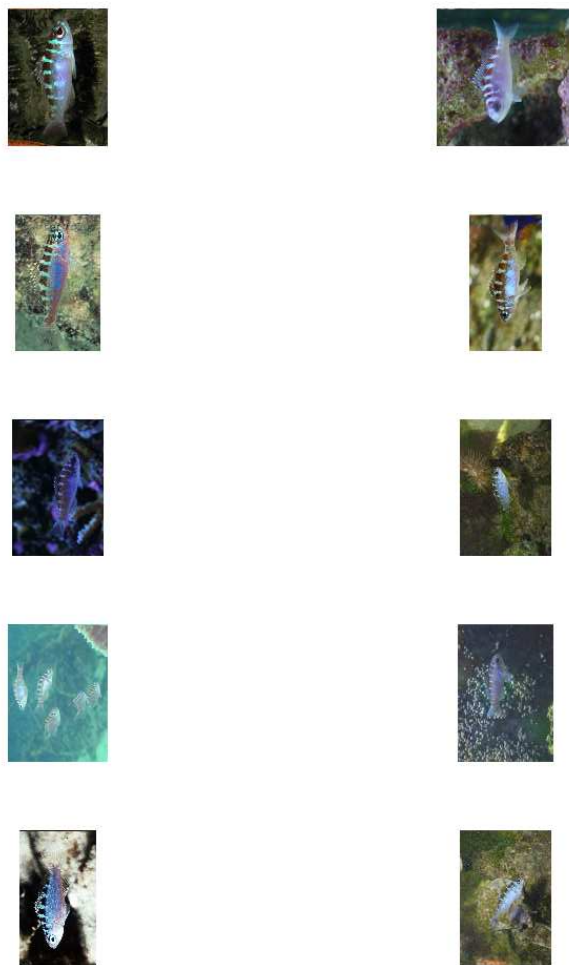


Figure C.19: Species Code:19 *Serranus tortugarum* chalk bass.

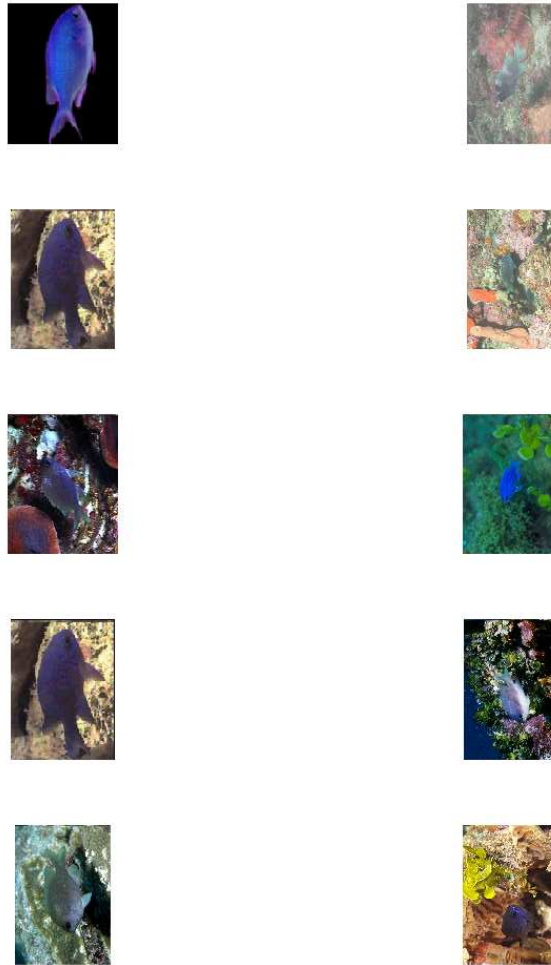


Figure C.20: Species Code:20 *Chromis scotti* purple reef fish.

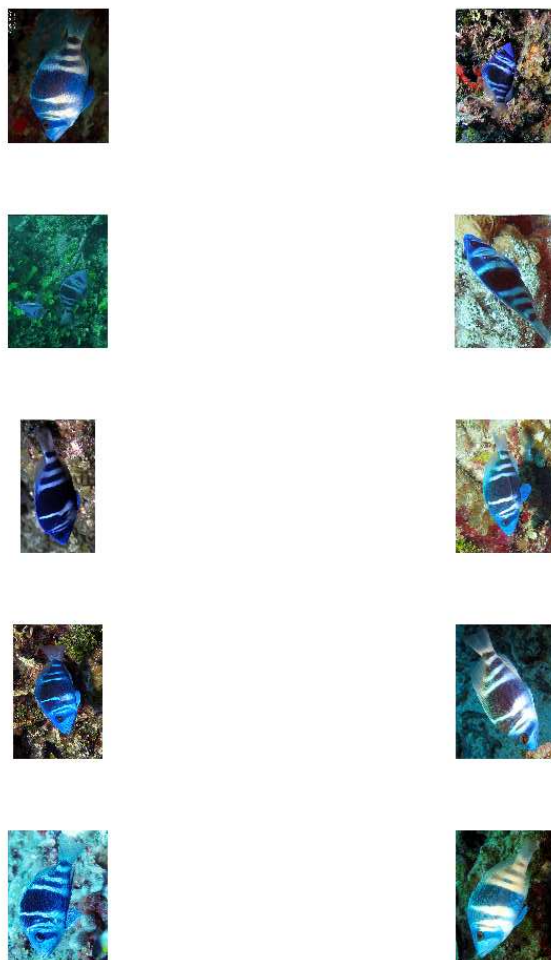


Figure C.21: Species Code:21 *Hypoplectrus indigo indigo* hamlet.

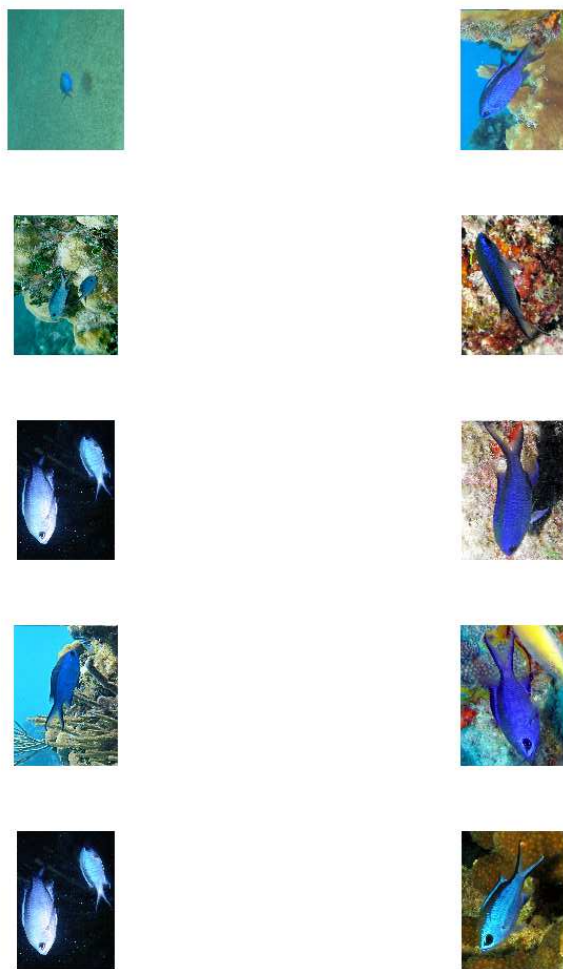


Figure C.22: Species Code:22 Chromis cyanea blue chromis.



Figure C.23: Species Code:23 Chaetodon sedentarius reef butterflyfish.

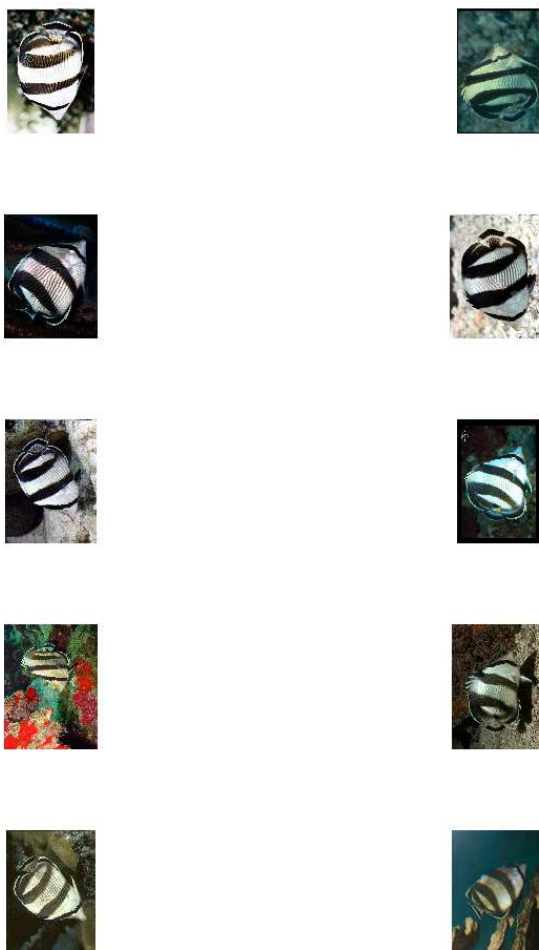


Figure C.24: Species Code:24 Chaetodon striatus banded butterflyfish.



Figure C.25: Species Code:25 Halichoeres garnoti yellowhead wrasse.

Appendix D

Confusion matrix for leave-one-out cross-validation

Continued next page ...

RECOGNIZED IDENTITY																													
T R U E I D E N T I T Y	Species	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	\hat{p}_C	\hat{p}_F	\hat{p}_I
	1	9							1																		0.9	0	0.1
	2		2										<u>6</u>		1										<u>1</u>		0.2	0.7	0.1
	3			7														1			<u>1</u>				<u>1</u>		0.7	0.1	0.2
	4				7									<u>1</u>				1					1				0.7	0.1	0.2
	5					4		<u>2</u>		1		1								1	1						0.4	0.2	0.4
	6	1				2	0	<u>1</u>				<u>4</u>		1				1									0	0.4	0.6
	7					<u>2</u>		3										<u>2</u>		1		1				1	0.3	0.4	0.3
	8							1	4						1							2				<u>2</u>	0.4	0.2	0.4
	9				1	1				5	<u>1</u>									1		1					0.5	0.1	0.4
	10										<u>5</u>				2			1				1		1			0.5	0	0.5
	11			1		1	<u>2</u>	1		1		3									1						0.3	0.2	0.5
	12		<u>5</u>										2												<u>3</u>		0.2	0.8	0
	13				<u>2</u>			1						6							1						0.6	0.2	0.2
	14									2	1				5			<u>1</u>			1						0.5	0.1	0.4
	15		1				1						1			3			1						3		0.3	0	0.7
	16				1											<u>1</u>	8										0.8	0.1	0.1
	17					<u>3</u>		<u>1</u>							<u>1</u>				5								0.5	0.5	0
	18																			10							1	0	0
	19					1				1											4	2	<u>1</u>			1	0.4	0.1	0.5
	20											1		1	1		1			1	4		<u>1</u>				0.4	0.1	0.5
	21							1		1													7			1	0.7	0	0.3
	22											2											7		1		0.7	0	0.3
	23		<u>2</u>	1				1																	6		0.6	0.2	0.2
	24		<u>1</u>										<u>1</u>											<u>2</u>	6		0.6	0.4	0
	25									<u>1</u>																9	0.9	0.1	0
	error	1	9	2	4	10	3	9	<u>2</u>	6	2	8	8	3	6	1	1	7	1	4	7	6	2	3	9	5			

Table D.1: Confusion matrix for best matched species (based on shape). Correct matches are diagonal entries. Incorrect matches that are in the same family are double underlined. All other entries are incorrect species and incorrect families. \hat{p}_C = Fraction of correct species, \hat{p}_F = Fraction of incorrect species but correct family, \hat{p}_I = Fraction of incorrect species and incorrect family. Ave. \hat{p}_C = 0.524, Ave \hat{p}_F = 0.2, Ave \hat{p}_I = 0.276. The sum of the errors in the bottom-most row is an indication of the uniqueness of the shape of the various species.