Automatic Video Analysis for Fisheries Survey Systems

Meng-Che Chuang

A dissertation

submitted in partial fulfillment of the

requirements for the degree of:

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Jenq-Neng Hwang, Chair

Linda G. Shapiro

Kresimir Williams

Program Authorized to Offer Degree:

Department of Electrical Engineering

University of Washington

Abstract

Automatic Video Analysis for Fisheries Survey Systems

Meng-Che Chuang

Chairperson of the Supervisory Committee:
Professor Jenq-Neng Hwang
Department of Electrical Engineering

Fisheries survey with the use of cameras has drawn increasing attention since it enables a non-lethal and high-resolution sampling of the fish stocks. This dissertation strives to build an automatic video analysis system that provides an effective solution to video-based fisheries surveys. The proposed system includes segmentation and tracking with controlled background, segmentation and tracking with uncontrolled background, body length estimation, and species recognition.

In segmentation and tracking with controlled background, a reliable algorithm for low-contrast and low-frame-rate (LFR) stereo videos is proposed for trawl-based camera systems. The automatic segmentation integrates the histogram backprojection to double local thresholding to produce accurate shape boundary. Built upon a feature-based temporal matching method, the multiple-target tracking algorithm via Viterbi data association is proposed to overcome the abrupt motion and frequent entrance/exit of targets.

In segmentation and tracking with uncontrolled background, a novel tracking algorithm based on deformable multiple kernels (DMK) is proposed. In light of the deformable part model (DPM), a number of kernels are defined to represent the holistic body and parts of the object. Kernel motion is efficiently estimated by the mean-shift algorithm on color and texture features. The DPM deformation costs are adopted as soft constraints on kernel positions to maintain the part configuration when matching HOG

features. After tracking, an active contour algorithm is applied around each target to realize target segmentation.

In body length estimation, an efficient block-matching approach performs successful stereo matching. It also enables an automatic fish-body tail compensation to reduce segmentation error and allows for an accurate fish length measurement after disparity refinement and stereo triangulation.

In species recognition, an innovative object recognition framework is proposed. To extract discriminative fish species features, both the supervised feature descriptors and the unsupervised feature learning method via the novel non-rigid part model are presented. For the classifier, unsupervised clustering generates a binary class hierarchy, where each node is a classifier. Partial classification is introduced to assign coarse labels to ambiguous data by systematically optimizing the “benefit” of classifier indecision. Several experiments show the effectiveness of the proposed system against public datasets and practical scenarios.

Table of Contents

Chapter 1 – Introduction 1
1.1 Fisheries Surveys 1
1.2 Challenges to Underwater Video Analysis 2
1.3 Contributions 4
1.4 Dissertation Organization 5

Chapter 2 – Background and Related Work 7
2.1 Cam-trawl System 7
2.2 Object Segmentation 8
2.2.a Segmentation in Underwater Video 8
2.2.b Uncontrolled Background 9
2.3 Object Tracking 10
2.3.a Controlled Background 10
2.3.b Uncontrolled Background 11
2.4 Body Length Estimation 13
2.5 Species Recognition 14
2.5.a Automatic Live Fish Classification 14
2.5.b Classification with Data Uncertainty 15

Chapter 3 – Segmentation and Tracking: Controlled Background 17
3.1 Overview 17
3.2 Automatic Segmentation 18
3.2.a Double Local Threshold 18
3.2.b Histogram Backprojection 20
3.2.c Threshold by Area and Variance 21
3.2.d Post-processing 22
3.3 Tracking from Low-Frame-Rate Video 23
3.3.a Stereo Matching with Object-Height Blocks 23
3.3.b Feature-based Temporal Matching 25
3.3.c Viterbi Data Association 27
3.4 Experimental Results 30
3.4.a Implementation Details 30
3.4.b Results of Segmentation 31
3.4.c Results of Tracking 33
3.5 Discussion 35
3.5.a Histogram Backprojection 35
3.5.b Features for Temporal Object Matching 38
3.5.c Target Occlusion 40

Chapter 4 – Segmentation and Tracking: Uncontrolled Background 42
4.1 Review of Basic Concepts 42
4.1.a Kernel-Based Tracking 42
4.1.b Deformable Part Model 43
4.2 Deformable Multiple-Kernel Tracking 45
4.2.a Models 46
4.2.b Tracking by Color and Texture 47
4.2.c Deformable HOG Mean-Shift 47
4.2.d Kernel Aggregation 49
4.2.e Scale Adjustment 50
4.3 Target Segmentation 50
4.4 Experimental Results 52
4.4.a Implementation Details 52
4.4.b Tracking Accuracy 53
4.4.c Multiple-Target Tracking and Segmentation 56
4.5 Discussion 59

Chapter 5 – Stereo Body Length Estimation 61
5.1 Overview 61
5.2 Length Estimation in 3-D Space 61
5.2.a Fish-Tail End Compensation 61
5.2.b Body Length Estimates 63
5.3 Experimental Results 64
5.4 Discussion 66

Chapter 6 – Species Recognition 67
6.1 Overview 67
6.2 Part-Aware Fish Feature Descriptors 68
6.2.a Head/Tail Localization 68
6.2.b Feature Descriptors: 69
6.3 Non-Rigid Part Model 70
6.3.a Fitness 71
6.3.b Separation 72
6.3.c Discrimination 72
6.3.d Objective Function 72
6.4 Unsupervised Part Model Discovery 73
6.4.a Part Initialization 73
6.4.b Labeling-based Part Association 74
6.4.c Unsupervised Part Model Discovery 76
6.5 Hierarchical Partial Classification 79
6.5.a Unsupervised Construction of Class Hierarchy 80
6.5.b Benefit-based Partial Classification 81

6.6 Probabilistic Outputs for Support Vector Machines 83
6.7 Experimental Results 84
6.7.a Datasets 84
6.7.b Implementation Details 85
6.7.c Feature Learning 87
6.7.d Hierarchical Partial Classification 88
6.7.e Running Time 90
6.8 Discussion 90
6.8.a Model Design 90
6.8.b Partial Classification 91

Chapter 7 – Conclusion and Future Work 93
7.1 Conclusion 93
7.2 Future Work 94

References 96

List of Figures

Figure 1.1: Fisheries survey. (a) Trawl net with fish; (b) pollock echo sign by acoustic surveys. (Source: (a) http://commons.wikimedia.org/wiki/File:Fish_on_Trawler.jpg#/media/File:Fish_on_Trawler.jpg); (b) http://www.afsc.noaa.gov/RACE/midwater/AFSC%20AT%20Survey%20Protocols_Feb%202013.pdf)..... 1
Figure 2.1: Cam-trawl underwater imaging system. (a) Illustration of the system overview; (b) underwater video captured at 5 frames per second, showing the abrupt motion and frequent entrance/exit of fish. 8
Figure 2.2: Comparison of stationary and moving cameras. (a) A stationary camera covers a fixed field of view. Targets are located and tracked by maintaining a background model. (b) A moving camera has ego-motion and its covered field of view changes constantly. Background modeling approaches are no longer effective. Camera ego-motion also introduces error when estimating target motion. 12
Figure 3.1: Upright (red) and oriented (green) bounding box of a target 17
Figure 3.2: Flow chart of the proposed fish segmentation algorithm 18
Figure 3.3: Basic concept of histogram backprojection..... 21
Figure 3.4: Example objects that are discarded due to (a) small area, (b) large area and (c) small variance of pixel values; (d) an object that are preserved after thresholding by area and variance. Note that (a) and (b) present the entire video frame, while (c) and (d) are zoomed-in regions. 22
Figure 3.5: Flow chart of the proposed multiple fish tracking system. 23
Figure 3.6: Object-height blocks (yellow) on the stereo-rectified left and right image. There are 4 candidates from each of the right target's upright bounding box for the minimum-SAD scheme..... 24
Figure 3.7: Multiple-target Viterbi data association. (a) Each target maintains a separate trellis during its lifespan with its own starting node (triangles) and ending node (squares). The optimal path in each trellis is labeled by colored arrows. (b) An overall trellis showing several paths from targets in (a). 29
Figure 3.8: Tracking multiple fish in an underwater video clip. Targets are labeled by numbers and oriented bounding boxes with different colors. (a) Using the proposed algorithm with matching cost plus Viterbi data association. (b) Using matching cost only. (c) Using primitive nearest neighbor data association. 35
Figure 3.9: Sensitivity test of histogram backprojection. Each color shows the error rates by using different numbers of histogram bins. Each labeled point shows the error rate by setting different values for the ratio histogram threshold. 37
Figure 3.10: An example target and its histograms generated by the proposed double local thresholding: (a) Low mask of the target; (b) high mask of the target; (c) two histogram from low and high mask, respectively; (d) ratio histogram given by (2).

One can see that most ratio histogram bins are close to either 0 or 1. As a result, the performance of segmentation is less sensitive to the selection of ratio histogram threshold. 38
Figure 3.11: Tracking success rate vs. unselected cues for temporal matching. Each color bar represents one cue that is not used when matching objects across video frames. 40
Figure 4.1: Flow chart of the deformable multiple-kernel (DMK) tracking algorithm. ... 45
Figure 4.2: Deformable multiple-kernel model (a) the fish DPM with 3 components, each of which is presented with a specific aspect ratio of fish body to capture different viewing perspective. One DPM component consists of a root filter and 8 part filters (sampled at twice the resolution of root filter); (b) a target object in the video; (c) root kernel corresponding to the root filter; (d) part kernels corresponding to the part filters and located according to the deformation cost map. 46
Figure 4.3: The root kernel (red) and part kernels (other colors): (a) from the previous frame; (b) after color mean-shift; (c) after texture mean-shift; (d) after HOG mean-shift, which considers the deformation costs. Note the restoration of part configuration according to the DPM (such as the cyan and white kernels on the bottom left corners). 48
Figure 4.4: Example of kernel aggregation: (a) the deformable multiple-kernel model after tracking; (b) the root kernel (red) and two part kernels (cyan and magenta), whose anchor vectors and projected object center given by (9) are illustrated; (c) final bounding box. 50
Figure 4.5: Errors of target center in pixels vs. frame number of tested datasets. (a) ROVVTS1_1; (b) AFSCUW_2; (c) HAUL1010_1; (d) HAUL1010_2..... 55
Figure 4.6: Results of a specific target using the proposed DMK tracking algorithm in the tested datasets. (a) ROVVTS1_1; (b) AFSCUW_2; (c) HAUL1010_1; (d) HAUL1010_2. Partial occlusion of the target occurs in (a), while abrupt change in speed and sharp turns of target motion can be observed in (c) and (d). 56
Figure 4.7: Average tracking error of multiple-target tracking in tested datasets. 57
Figure 4.8: Results of tracking multiple targets simultaneously. Each target is labeled by different colors and numbers. (a) ASFCUW_1; (b) ROVVTS_3..... 57
Figure 4.9: Results of target segmentation. (a) Target images; (b) segmentation masks; (c) contours. 58
Figure 4.10: Average tracking error by the proposed method vs. kernel aggregation weight. 60
Figure 5.1: End square regions (orange) of a fish. In the figure w_O and h_O denote respectively the width and height of the oriented bounding box (blue). 62
Figure 5.2: Illustrative example of fish-tail end compensation: (a) stereo-rectified image; (b) object mask before end compensation; (c) object mask from (b) and result of snake algorithm; (d) new object mask after morphological closing operation; (e) upright bounding box (green), oriented bounding box (blue) and end points (yellow)

before end compensation; (f) upright bounding box, oriented bounding box and end points after end compensation, where the less-reflective part around tail fin is recovered. 63
Figure 5.3: Length frequency of fish targets. 65
Figure 6.1: Fish binary masks (top row) and their vertical projections (bottom row). The boundary positions for the tail and head region are labeled separately by red and turquoise lines and dots. 69
Figure 6.2: Overview of the proposed unsupervised non-rigid part learning algorithm. . 70
Figure 6.3: Part initialization based on PFT saliency. (a) Input images; (b) PFT results of (a); (c) points with top saliency values inside the segmentation masks; (d) initial parts. 74
Figure 6.4: An example of hierarchical partial classifier. 80
Figure 6.5: Partial classification for an SVM. Ambiguous data have small absolute decision values, so they fall in the indecision domain (ID) and are not assigned to either of the classes. 81
Figure 6.6: Visualization of (a) Equation (6.29) and (b) Equation (6.30) with $a = 0.5$ ... 83
Figure 6.7: Decision rate and exponential benefit vs. decision threshold. 83
Figure 6.8: Fish species in Fish4Knowledge dataset. Two largest species (left) and two smallest species (right) are shown to demonstrate the imbalance in class distribution. 85
Figure 6.9: Fish species in NOAA Fisheries dataset sorted in the descending order of the number of samples. 85
Figure 6.10: Examples of parts detected by the proposed unsupervised feature learning algorithm. Each column shows the parts found by one part model. Meaningful fish body parts are successfully learned regardless changes in size, such as head, caudal fin and pectoral fin. 88
Figure 6.11: Precision, recall and F_1-score of each species from NOAA Fisheries dataset. 90
Figure 6.12: Performance vs. number of parts in the non-rigid part model. 91

List of Tables

Table 3.1: Values of parameters used in the experiments 31
Table 3.2: Precision and recall of fish segmentation 32
Table 3.3: Mean absolute percentage error of large target length 32
Table 3.4: Performance of fish detection over video clips 34
Table 3.5: Tracking success rate vs. data association methods 34
Table 3.6: Performance vs. occlusion handling on video 3 41
Table 4.1: Average tracking error (pixels) 54
Table 4.2: Computational consumption 59
Table 6.1: List of descriptors in part-aware fish features 69
Table 6.2: Performance of feature learning methods on NOAA Fisheries dataset 88
Table 6.3: Performance of top 15 species of F4K dataset 89
Table 6.4: Performance of NOAA Fisheries dataset 90
Table 6.5: Performance of full vs. partial classification algorithms 92

Acknowledgements

It is still hard to believe that the 4-year-and-9-month journey of studying abroad has come to an end. I can't express how grateful I was at the moment when committee members shook my hand and said to me, "Congratulations, now you're Dr. Chuang." I know from deep in my heart that I would not be able to accomplish this without help from many people.

I would like to first express my sincere gratitude to my adviser, Prof. Jenq-Neng Hwang, for his guidance throughout my PhD study. He is always supportive and encouraging to help me pursue the goals and overcome the difficulties in research. From him I have learned a lot about how a successful scholar tackles open problems and gives good presentations.

I am grateful to my committee members, Prof. Mark Ganter, Prof. Linda Shapiro, Prof. Steve Tanimoto and Dr. Kresimir Williams, for the valuable suggestions they have given since my General Exam.

Special thanks to Dr. Fang-Fei Kuo and Prof. Man-Kwan Shan for countless insightful discussions and giving me advices on paper writing during their visit here at the University of Washington.

I am thankful to the alumni I have met at the Information Processing Lab (IPL), Chun-Te Chu, Shian-Ru Ke, Po-Han Wu, Pei-An Lee, Youngdae Lee and Ruizhi Sun, for their help and discussions during my first couple years in this group.

I would like to thank the current colleagues at IPL: Xiang Chen, Kuan-Hui Lee, Younggun Lee, Jounsup Park, Zheng Tang, Renshu Gu, Qiuyu Chen, Arash Tarkhan and Tsung-Wei Huang. They make this research group an excellence environment to work. Best wishes to all of them on their graduate study and future career.

I also thank all the friends I have met in Seattle. You have not only given me support but also enriched my life of studying abroad in the beautiful Emerald City. I will remember the laughter every time we got together.

With deepest love, I would like to thank my soul mate and friend for life, Dr. Chieh-Li Chen, for her insightful suggestions and warm encouragement. I am lucky to have such a great partner on our own adventure to PhD. I am even luckier to have this partner's company for the very last mile. I cherish the coast-to-coast phone calls every day and night, the experience she shared in being a graduate student, and the wonderful memories we have made all around the world. I look forward to our next adventure.

Last but not least, I am grateful to my dearest family: my father, my mother, my sister and my grandmother, for their endless love, support and encouragement for everything I do. They are the ones who make me strong and confident to conquer the hurdles set in front of me. This dissertation is dedicated to them.

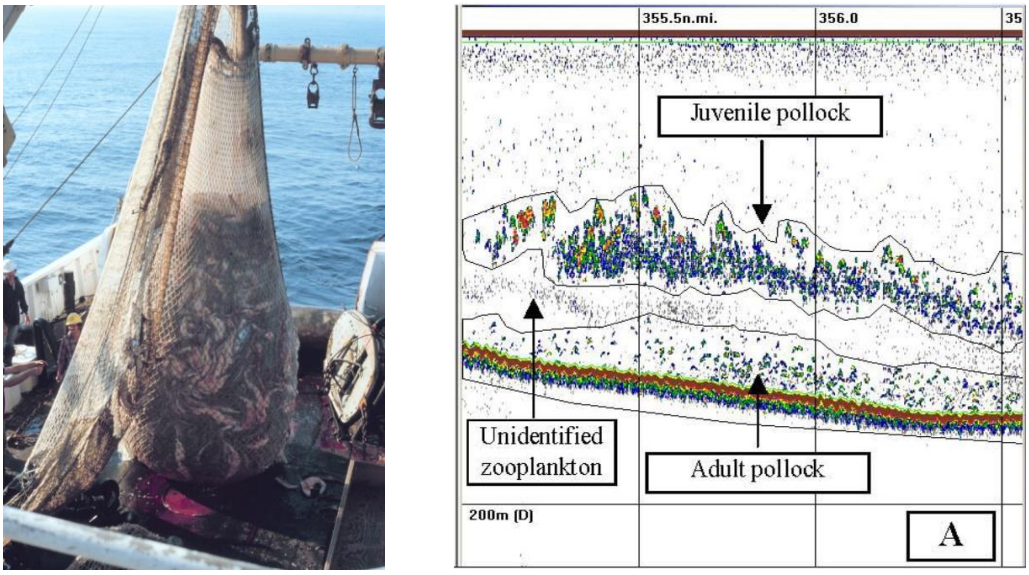
Dedication

To my family

Chapter 1 – Introduction

1.1 Fisheries Surveys

One of the most critically required tasks for the conservation and management of fish stocks, especially scientifically or commercially important resources, is conducting fisheries surveys [44]. Fisheries scientists often call for the use of bottom and mid-water trawls to estimate fish abundance, size structures and species composition. There are several methodologies for fisheries surveys. Two examples are shown in Figure 1.1. The traditional trawl survey records the information about retained fish using manual labor. Facilitated by signal processing techniques, some studies presented the acoustic-trawl survey [32, 50, 64] where trawl catches provide information that allows human to convert acoustic data into estimates of abundance and size structure.



(a) (b)

Figure 1.1: Fisheries survey. (a) Trawl net with fish; (b) pollock echo sign by acoustic surveys. (Source: (a) http://commons.wikimedia.org/wiki/File:Fish_on_Trawler.jpg#/media/File:Fish_on_Trawler.jpg); (b) http://www.afsc.noaa.gov/RACE/midwater/AFSC%20AT%20Survey%20Protocols_Feb%202013.pdf)

There are, however, several drawbacks to both types of surveys. The catch represents a relatively long space- and time-integrated sample, so high-resolution information along

the trawl path is inaccessible. Many smaller animals travel through the trawl and are not retained in the catch, and information about these components of the ecosystem require other specialized equipment. Furthermore, fish caught by trawls often do not survive, and thus trawl survey methods are inappropriate in some areas where fish stocks are severely depleted.

To address these needs, optical instruments such as underwater cameras are rapidly becoming a standard methodology for surveying fishery resources and studying fish behavior in a nonextractive manner. For instance, the Cam-trawl [87, 88] is a self-contained combination of trawl and stereo cameras system. It enables a non-lethal and high-resolution sampling, since fish are allowed to pass unharmed to the environment after being sampled. The collected image or video data provide much of the information that is typically collected from fish that are retained by traditional trawl methods.

The main issue in conducting video-based fisheries surveys is the vast amounts of data rapidly collected during the survey. The challenges to analyzing the data can be reduced by using video processing techniques for automatic detection, segmentation, tracking, size measurement and species recognition. A successful development of these algorithms will greatly ease one of the most onerous steps in video-based sampling. With automation of these tasks, monitoring the amount and species composition of fish schools becomes possible and thus provides a mean of assessing the status of fish stocks and the ecosystem.

1.2 Challenges to Underwater Video Analysis

Video analysis techniques for live fish detection, tracking and counting using monocular or stereo cameras have been investigated in the video processing and computer vision community [12, 29, 46, 54, 62, 80, 84]. There are, however, still several challenges for estimating the size and abundance of fish by using automatic processing. First, the fast absorbance of light and non-uniformity of artificial illumination results in relatively low contrast of the foreground targets. Fish with similar ranges from the

cameras can have significantly different intensity because of the differences in angle of incidence as well as reflectivity of fish bodies among species. These factors make segmentation of fish difficult. In addition, the ubiquitous noise is created by non-fish objects such as bubbles, organic debris and invertebrates can easily be mistaken for real fish. Due to the limitation of data transmission in some cases, images are captured at a low frame rate (LFR), which results in poor motion continuity and frequent target entrance/exit of the field of view, traditional multiple-target tracking algorithms [12, 84] are infeasible. In some situations, such as conducting surveys in coral or rocky habitats, underwater video is captured by towed cameras or remotely-operated underwater vehicles (ROV's). Object segmentation and tracking becomes more challenging since the complex background and camera motion make established techniques of background modeling [99] inapplicable.

On the other hand, while object recognition in various contexts has been well investigated, additional challenges are posted to identifying fish in an unconstrained natural habitat. For freely-swimming fish, there is a high uncertainty existing in many of the data because of poor image quality, non-lateral fish views or curved body shapes. Critical information about species in these data may be lost or comes with large measurement error. Even without uncertainty, fish often share a strong visual similarity among species. Common features for image classification are hence not discriminative, since they represent merely the global appearance of an object.

Motivated by all the issues above, this dissertation strives to build an automatic and consistent video analysis system for underwater cameras that facilitates video-based fisheries survey systems. The segmentation and tracking algorithms for controlled background, such as the trawl-based cameras, focus on mitigating the challenges posted by low contrast, low frame rate and ubiquitous noise. For moving cameras where the background is uncontrolled, the aim is to reduce the computational cost required by the existing tracking-by-detection paradigm. The size of each segmented target is estimated accurately via stereo vision. Finally, in species recognition, the aim is to reduce the

human intervention as much as possible while handling practical issues such as class imbalance and data uncertainty. To this end, unsupervised learning is applied to learn the *non-rigid part model*. The model provides discriminative features to the hierarchical partial classifier that allows for coarse-to-fine categorization for both certain and uncertain data. The proposed system shows its effectiveness against public datasets and practical scenarios in several experiments and comparative studies.

1.3 Contributions

The contributions of this dissertation are summarized as follows.

- The proposed automatic and adaptive fish segmentation algorithm overcomes the challenges posted by underwater imagery, such as low contrast, uneven illumination, and ubiquitous noise.
- The novel multiple-target tracking algorithm based on feature-based temporal matching and Viterbi data association is proposed to track stereo-paired fish with abrupt motion and limited duration of presence due to the low frame rate.
- For video with an uncontrolled background, the proposed deformable multiple-kernel object tracking algorithm provides an effective and efficient solution.
- With the use of gradient and texture features, the accuracy of kernel-based tracking is enhanced in underwater scenes where the color features are often distorted.
- By applying kernel-based tracking, the computational requirement to track objects from a moving camera is significantly reduced compared to existing tracking-by-detection paradigm.
- With stereo vision available, a fast and effective stereo matching method followed by a self-compensation scheme is proposed to accomplish target pairing and thus allows for accurate target length measurement.
- For species recognition, part-aware features provide simple but discriminative feature descriptors for fish species recognition.

- A novel non-rigid part model that represents local attributes fish body is proposed for species recognition. Such a model can be trained via a fully unsupervised learning algorithm.
- The proposed hierarchical partial classification handles data uncertainty and class imbalance in practical object recognition applications without discarding samples. In addition, finding the decision criteria of partial classifiers is formally formulated as a convex optimization problem.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows.

Chapter 2: the Cam-trawl system is briefly overviewed, followed by a review of related work on segmentation and tracking for controlled and uncontrolled background, stereo size estimation, and object recognition.

Chapter 3: the proposed segmentation and tracking framework for controlled background is described. First, the automatic segmentation algorithm is introduced. This method handles low contrast and ubiquitous noise issues in underwater imagery by integrating a novel double local thresholding scheme with histogram backprojection. Thresholding by target area and variance further discards irrelevant foreground objects. Afterward, the multiple-kernel tracking framework for low-frame-rate video is presented. A feature-based temporal matching method is performed frame-by-frame followed by the Viterbi data association for global optimization. Extensive experiments and analysis of the proposed framework are given at the end.

Chapter 4: the proposed segmentation and tracking framework for uncontrolled background is presented. First, the basic concepts of kernel-based tracking and deformable part model (DPM) detector are reviewed. The proposed deformable multiple-kernel (DMK) model is then introduced. Kernel-based tracking using color, texture and HOG features are applied to the DMK model with constraints given by the DPM

deformation costs. Based on tracking results, segmentation is applied locally by using the active contour approach. Experimental results and analyses are also reported.

Chapter 5: the stereo body length measurement algorithm is described. A self-compensation scheme is applied to a pair of targets matched across the left and right cameras to reduce the error in length estimation. After reprojecting targets via stereo triangulation, their body lengths are estimated in the 3-D space and are consistent with the physical measurements.

Chapter 6: the proposed fish recognition system is introduced. A supervised Part-Aware Fish Feature Descriptors provides simple but discriminative feature descriptors to categorize fine-grained fish species. To further automate feature extraction, the non-rigid part model is proposed and a fully unsupervised learning algorithm is presented to discover discriminative parts. Next, the hierarchical partial classification framework is proposed to handle the class imbalance and data uncertainty. Finding the optimal decision criteria of partial classifiers is systematically formulated as maximizing the benefit, a quantity that evaluates the effectiveness of partial classification. Extensive experiments on public and self-collected datasets along with analysis of the proposed system are provided.

Chapter 7: a conclusion of this dissertation is given by summarizing the main contributions and discussing some extensions to this work that lead to potential research topics in the future.

Chapter 2 – Background and Related Work

2.1 Cam-trawl System

The Cam-trawl [87, 88] represents a new class of mid-water imaging sampler to study the marine environment. With ongoing development, however, the Cam-trawl is poised to become a standard marine surveying tool to provide a more holistic view of the marine environment, and improve the management of our marine resources.

As shown in Figure 2.1(a), the Cam-trawl consists of two high-resolution machine vision cameras, a series of LED strobes, a computer, microcontroller, sensors, and battery power supply. The cameras and battery pack are housed in separate 4-inch diameter titanium pressure housings, and the computer, microcontroller and sensors are placed in a single 6-inch diameter aluminum housing. This self-contained stereo-camera system is fitted to the aft end of a trawl, which is attached to a moving boat, in place of the codend (i.e., capture bag) for video sequence capture. The absence of the codend allows fish to pass unharmed to the environment after being sampled (video captured).

The high-resolution high-sensitivity cameras are capable of capturing 4-megapixel images. The cameras are connected via a gigabit Ethernet to a Core 2 Duo PC with software to control the camera's operation and to store the video data to a solid state hard disk drive. Due to the limited bandwidth of Ethernet data transmission and storage in an earlier hardware design of the Cam-trawl, the capturing rate of the cameras is at most 10 frames per second (fps). Considering the tradeoff between the image quality and data transmission speed, the capturing rate is set to 5 fps. This allows the cameras to collect high-definition video data that are favorable for accurate segmentation and tracking. At this capturing rate, targets move abruptly from one frame to another and enter/exit the field of view (FOV) frequently (4.3 frames of target lifespan on average). This makes conventional tracking methods infeasible for this task. To illustrate this scenario, six consecutive frames captured by the Cam-trawl are shown in Figure 2.1(b). A full-featured

software development kit (SDK) supports the core acquisition and control routines. The PC runs a customized Linux operating system, which allows precise control over what software and services are started depending on how the system is being used.

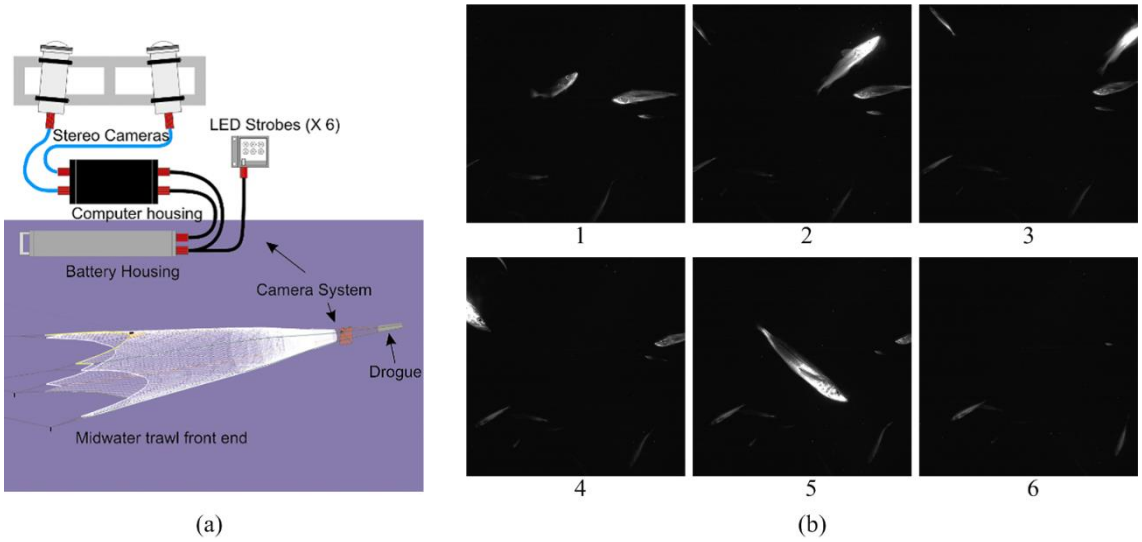


Figure 2.1: Cam-trawl underwater imaging system. (a) Illustration of the system overview; (b) underwater video captured at 5 frames per second, showing the abrupt motion and frequent entrance/exit of fish.

2.2 Object Segmentation

2.2.a Segmentation in Underwater Video

In general cases of controlled background with spatial homogeneity, as shown in Figure 2.1(b), thresholding can be an effective tool to extract the foreground objects. The main challenge to segmentation in underwater video, however, is the loss of color and contrast due to fast light absorbance by the water. Using an artificial light source such as LED strobes leads to non-uniform scene illumination both spatially and temporally. As a result, many foreground targets have rather low contrast to the background and uneven illumination on their bodies since the reflective light intensity is much stronger than the diffusive and ambient light. When the artificial light is not collimated, targets at similar distance from the camera but different locations in the image can appear with significantly different luminance. Furthermore, the ubiquitous organic debris introduces

noise in the images that degrades the accuracy in finding the target boundary. These factors hinder finding a threshold that applies reliably for the entire image or video frame. Wang et al. [85] imposed the notion of multiple thresholds and enforced the result by a boundary refinement technique. However, it requires intensive computational power and fails to detect small holes within silhouettes. Spampinato et al. [80] combined two models, namely the moving average and the Gaussian mixture model (GMM), to separate objects from the background in shallow underwater scenes with natural light. Uneven intensity on the same target body, which is common when using artificial light sources, is unable to be handled by this approach. Walther et al. [84] proposed an object detection approach based on saliency maps. However, the boundary of a target is unable to be tracked very accurately and the thresholded object may not exactly correspond to the true shape of the tested target. To this end, a local and adaptive algorithm is proposed to address these issues and produce reliable segmentation results. By integrating a novel double local thresholding technique to histogram backprojection, each target is processed separately based on the intensity around its local region. Histogram backprojection combines two different masks given by double thresholding and successfully refines the target boundary. Moreover, area and grayscale variance serves as reliable cues to preserve only targets of interest.

2.2.b Uncontrolled Background

Identifying targets where image backgrounds are uncontrolled and non-uniform is challenging because it requires video object segmentation approaches that are able to separate targets from the background with complex textures. Automated target segmentation from uncontrolled backgrounds becomes even more challenging when the camera platform is in motion since the scene is continuously changing and established techniques of background subtraction are no longer applicable.

Existing work on segmentation that exploits temporal relations of target instances is based on target tracking. Most of them employ a mechanism that projects the previous

regions to the current frame and adjust the projected regions in the current frame [42, 65]. Since this approach tends to be robust to sequences with moving background, it is now becoming commonly accepted although some form of automatic object segmentation for the initial frame and the subsequent key frames is required. The drawback of these methods mainly results from the difficulty of either adjusting the projected regions in the current frame or coping with non-rigid motions. Usually, segmentation based on color or grayscale information can be used to correct motion-projected regions, but this process requires a high computational cost.

Other challenges in segmentation include the boundary ambiguity due to the diversity in complex background colors and frequently changing intensity in the underwater environment. There has been a large body of work on active contour techniques, which are based on the "snake" algorithm [51] and have been widely applied in medical image processing studies. Active contour methods defines an initial target contour, which could be in some generic shape, and iteratively deforms the contour to minimize an energy functional imposed by the image. A number of existing active contour methods have been shown the reliability in challenging cases [13, 14]. Most of these approaches involves numerical analysis and requires extremely heavy computational cost. There exists work in the literature that accelerates these approaches based on image morphologies [63].

2.3 Object Tracking

2.3.a Controlled Background

Object tracking has been one of the most seminal topics in video content analyses for decades because of its importance in a wide range of applications [91]. For fisheries applications, there exist publications in the literature on tracking multiple fish in underwater scenes [84]. The main challenge to tracking from underwater cameras includes unstable illumination as well as the ubiquitous noise. Walther et al. [84]

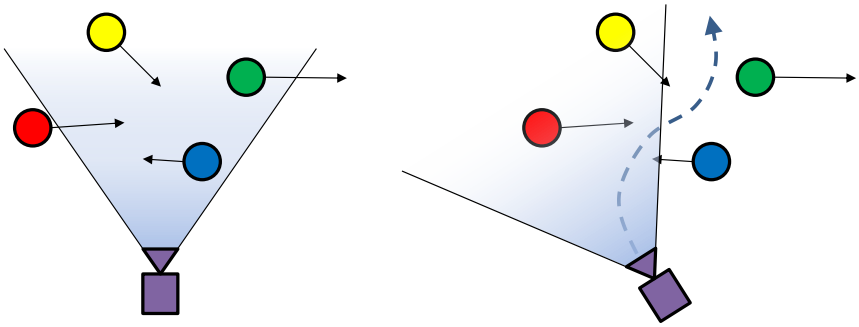
performs multiple-target tracking on underwater videos using the conventional Kalman filter. Butail et al. [12] takes a nonlinear estimation approach by using a sampling importance resampling particle filter to track multiple fish in a school.

Most traditional tracking techniques are developed based on the assumption of motion continuity. For low-frame-rate (LFR) video, however, abrupt motion is observed for almost every target. Due to the poor motion continuity and short target lifespan in the field of view, most existing tracking methods are unable to be readily applied to the LFR scenario. A number of approaches has been proposed to address this issue by using detection. For instance, Porikli et al. [73] extended the kernel-based technique by placing multiple kernels around motion areas given by background modeling and optimizing around these kernels. On the other hand, Li et al. [60] proposed a cascaded particle filter with a number of discriminative observers trained from different range of video frames. These methods relax the dependence of smooth target motion, but they require high computational cost. In addition, none of the methods take into account the global optimality while extending to multiple-target tracking circumstance. To this end, a matching-based tracking algorithm for LFR video is proposed, which not only requires less computations but also optimize the solutions both spatially and temporally.

2.3.b Uncontrolled Background

Most traditional video tracking applications, such as surveillance and human-computer interaction, involve a number of stationary cameras that are set up with fixed location and orientation. Background subtraction is commonly applied in these cases and provides reliable measurements for target location and shape based on a Gaussian mixture model (GMM) for background intensity [99]. Recently, an emerging body of work employs one or multiple moving cameras for video capturing [17, 58]. A moving camera is usually mounted on a mobile vehicle-like platform, such as autonomous car or robot vehicle. There also exist applications in fisheries studies that utilize cameras mounted on a remotely-operated underwater vehicle (ROV) or the end of a trawl [19, 21,

23-25]. A comparison between these two scenarios are illustrated in Figure 2.2. The main challenge to tracking objects from these cameras is that background modeling techniques are not applicable since the field of view changes continuously. As a result, it becomes very difficult to locate objects with satisfactory accuracy. Moreover, object perspective and scale may change rapidly across time in the video, i.e., primitive size and shape information of a tracked object is not invariant.



(a) (b)

Figure 2.2: Comparison of stationary and moving cameras. (a) A stationary camera covers a fixed field of view. Targets are located and tracked by maintaining a background model. (b) A moving camera has ego-motion and its covered field of view changes constantly. Background modeling approaches are no longer effective. Camera ego-motion also introduces error when estimating target motion.

To address these challenges, tracking based on object detection results throughout the video has been investigated and is referred to as the tracking-by-detection paradigm [4, 5, 11, 92-94]. For all video frames, object detection gives the patches of object's presence, which are then associated through time to form target tracks via local or global optimization schemes. There is a large body of work on generic object detection algorithms such as the implicit shape model (ISM) [59], C^4 [89], Regionlets [86] and deformable part model (DPM) [34]. Among these techniques, the DPM has shown its effectiveness in challenging cases and simplicity of the discriminative model comparing to later work based on deep structures [39, 82]. Based on a variant of the histogram of oriented gradients (HOG) features [30], the DPM depicts the object category by the pictorial structure framework [35], which comprises the holistic appearance and a

collection of parts arranged in a deformable configuration. The detection-based paradigm has the advantage of requiring no knowledge about background appearance or target motion, and hence it is suitable for moving cameras. However, the computational cost is extremely heavy since it scans each video frame via a window with dozens of different scales, which leads to an exponential growth in the number of object proposals.

Another type of approach that can deal with camera motion is kernel-based tracking [18, 27, 28, 33, 43]. This type of method builds a target model in terms of a color histogram where each pixel is weighted by its spatial distance to the object center. The mean-shift algorithm is then employed to efficiently find the local maximum of the feature similarity function [28]. To handle partial occlusions, a multiple-kernel approach is introduced that represents a target by more than one kernel [18]. The projected gradient method is used to optimize the kernel locations under some given equality constraints. However, kernel-based tracking methods fails easily when there is a high similarity in color between the target and background or among several nearby targets. To this end, by taking advantage of both the merits of DPM detection and multiple-kernel tracking, a novel tracking algorithm based on deformable multiple kernels is proposed to address the issues introduced by camera motion.

2.4 Body Length Estimation

There are some studies in the fisheries literature about fish size estimation by using stereo cameras [29, 46, 54, 62]. Video objects are matched between a pair of calibrated and rectified images, and reprojected to the 3-D space (world coordinates) through stereo triangulation. Lines et al. [62] adopted a classification-based segmentation algorithm that integrates stereo matching given the epipolar geometry of two cameras. Some of these studies matched corresponding points in stereo images by manual processing and corrected the results by some additional processing. For instance, Harvey et al. [46] repeated the measurement from one fish over sequential images and introduced simple statistical analyses to avoid the error due to sinusoidal changes in body shape during fast

swimming. Costa et al. [29] trained a neural network based on the backpropagation algorithm to reduce the error in 3-D length measurement.

Although automatic stereo matching has been one of the most heavily investigated topics in computer vision [9, 77, 98], most state-of-the-art stereo matching approaches suffer seriously from their intensive computations, making them rather infeasible in many applications. In the proposed stereo fish body length estimation method, a fast but effective block-matching algorithm is based on the knowledge of segmentation masks in dual cameras. The algorithm is followed by a self-compensation scheme to recover less reflective areas of targets such as fins. After refining the disparity and reprojecting the 2-D object to the 3-D space, an accurate estimate of body length is obtained for each target.

2.5 Species Recognition

2.5.a Automatic Live Fish Classification

Live fish recognition is one of the most crucial elements in camera-based fisheries survey systems [49, 55, 57, 81]. Similar to most recognition frameworks, the successful extraction of informative features is the key to enhancing fish recognition performance. Existing feature extraction techniques are divided into two categories, namely the supervised and unsupervised methods. Supervised methods represent a fish by pre-specified features that adopt common low-level image descriptors such as contour shape. For instance, Lee et al. [55] used a curvature analysis approach to locate critical landmark points. The contour segments of interest were extracted based on these landmark points to achieve satisfactory shape-based species classification results. In their subsequent work [56], the features were further extended to include several shape descriptors such as Fourier descriptors, polygon approximation and line segments. A power cepstrum technique was developed in order to improve the categorization speed using contours represented in tangent space with normalized length. Spampinato et al. [81] proposed fish descriptors that consider appearance attributes such as texture in addition to contour

shape. With the huge diversity of fish, however, one set of features designed for some species is not guaranteed to be discriminative for other species. Moreover, manually selected features may lead to suboptimal recognition performance even based on domain knowledge.

On the other hand, unsupervised methods learn informative features directly from the images. Some approaches in this category can be found in the literature of fine-grained object recognition [37, 38, 41, 90, 95] or discriminative mid-level image patch discovery for scene recognition [31, 79]. There are also other branches of unsupervised methods based on, for example, traditional feature selection theories [67]. We have conducted a systematic comparison between supervised and unsupervised approaches and shown that the unsupervised approaches in general lead to better recognition performance [21]. Based on this, we extend the unsupervised feature learning algorithm in this paper by proposing a novel non-rigid part model, which considers part geometry and can be learned in a fully unsupervised manner.

2.5.b Classification with Data Uncertainty

In statistics, traditional strategies for handling uncertain data include discarding outlier data or performing imputation, where estimates are used to fill in the missing values. Some work integrated the classification formulation with the concept of robust statistics by assuming different noise distributions [1, 3]. Huang et al. [49] used a hierarchical classifier constructed by heuristics to control the error accumulation. However, the error propagates to the leaf layer of the tree once a wrong decision occurs.

One strategy to handle uncertainty is partial classification, i.e., allowing indecision by the classifier in certain regions in the data space. Partial classification has shown its effectiveness in various practical applications [1, 3, 6]. Ali et al. [1] proposed to determine whether to make a decision based on data mining techniques. Baram [3] introduced a benefit function for evaluating the deferred decisions and search for the optimal decision criterion exhaustively. However, the importance of rejected instances is

removed since no information about the data is retrieved. Besides, there are not yet any systematic methods proposed to determine the criteria of decision making.

Since objects can be naturally categorized into higher groupings of classes based on either domain knowledge or visual similarity, the recognition algorithm would be favorable by obtaining a hierarchical relation among classes automatically and then providing a coarse-to-fine categorization that can retrieve partial information from those uncertain data. To this end, we generalized the definition of benefit function, and propose a novel formulation based on exponential functions that systematically help select the decision criteria for a partial classifier [19].

Chapter 3 – Segmentation and Tracking: Controlled Background

3.1 Overview

In this chapter, a multiple-fish segmentation and tracking framework is proposed to provide an automatic solution to fish target length estimation and counting for trawl-based underwater camera systems. The proposed framework includes an adaptive object segmentation algorithm that overcomes the challenges imposed by low contrast and uneven illumination by modifying Otsu thresholding [69] and histogram backprojection procedure [53], and a novel multiple-target tracking algorithm to track fish with abrupt movement due to low frame rate by developing a feature-based temporal matching approach and extending the Viterbi data association used in single-target tracking.

For the convenience in describing each target and its neighborhood, two types of object bounding boxes are introduced for use in the rest of this chapter: 1) upright bounding box and 2) oriented bounding box. An upright bounding box is an axis-parallel rectangle that encloses the object. It is commonly used to indicate the region where an object appears in an image or video frame. On the other hand, an oriented bounding box is a rotated rectangle that encloses the object. Its width, height and orientation are determined by the principal component analysis (PCA) of the binary object mask. More specifically, the width is the object size measured along the direction of the first principal component, the height is that measured along the second principal component, and the orientation of the rotated box is parallel to the first principal component. Examples of upright and oriented bounding boxes are illustrated in Figure 3.1.

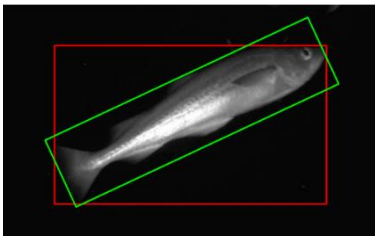


Figure 3.1: Upright (red) and oriented (green) bounding box of a target

3.2 Automatic Segmentation

The proposed algorithm for automatic fish segmentation is divided into four steps, as shown in Figure 3.2. The first step is double local thresholding. Next, binary object masks generated by two different thresholds are effectively integrated using a method based on histogram backprojection. After that, thresholding by area and variance removes noise and unwanted objects. Finally, a post-processing step is applied in order to refine the segmented object boundaries. Double local thresholding is designed to resolve the problem of non-uniform illumination over the video frame by focusing only on the vicinity of each target. On the other hand, histogram backprojection effectively merges two segmentation masks intelligently according to their difference in intensity distribution and refines the mask boundary for low-contrast imaging at the same time.

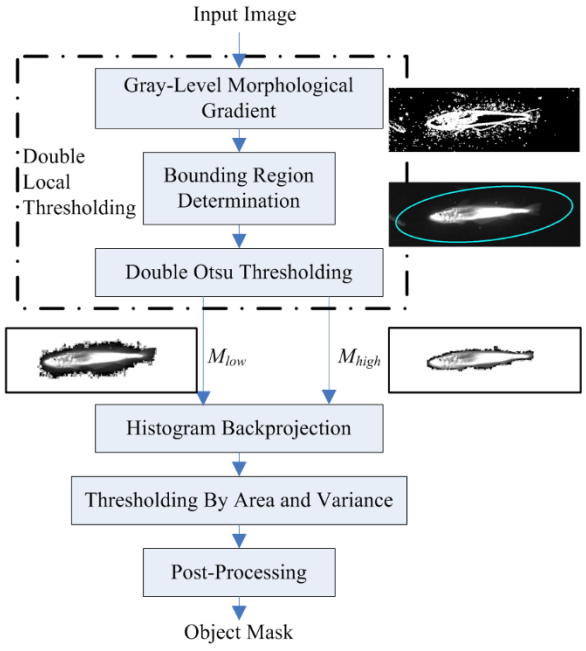


Figure 3.2: Flow chart of the proposed fish segmentation algorithm

3.2.a Double Local Threshold

One approach to object segmentation for video with a simple background, as shown in Figure 2.1(b), is thresholding, which binarizes the video frame by setting a threshold

on pixel intensity. Otsu's method [69] is widely used to find the optimal threshold, which separates the histogram into two classes so that the combined intra-class variance is minimal. Using only one threshold, however, introduces defects in object contours when the contrast between foreground and background is low. Also, thresholding over the entire video frame usually fails to segment objects if the illumination is uneven across the frame. To overcome these challenges, the double local thresholding algorithm is proposed to find two thresholds, i.e., generate two different binary masks, within each object's neighborhood. These two binary masks will be merged and refined by the histogram backprojection scheme described in Section 3.2.b.

When using double local thresholding for fish segmentation, what is needed first is to find a rough position and size of the fish. A gray-level morphological gradient operation [78] is performed on the input video frame to roughly locate the fish object in the input video frame. Next, the local region around the detected objects has to be determined. The classic connected components algorithm [45] is applied to mark the isolated local region in the object mask. Each region is then described by an inscribed ellipse of the oriented bounding box. The length of major and minor axis of the ellipse is the width and height of oriented bounding box, respectively. The orientation of ellipse is the same as the oriented rectangular box. Finally, the local region is determined by enlarging the oriented ellipse by a factor of 1.5 in both the major and minor axes.

With these elliptic local regions, the target is ready for the double local thresholding method. For each region, adaptive thresholds are selected using our proposed variant of the Otsu's method. To preserve some dim targets, which have intensity values close to the background, the threshold is given by

$$\tau_x = \tau - p(\tau - \mu_L(\tau)), \tag{3.1}$$

where τ is the threshold given by Otsu's method, $\mu_L(\tau)$ is the mean of lower class separated by τ in the histogram and p is a scalar shifting factor. Using (3.1), two thresholds are obtained by setting different values for p, i.e., $\tau_{low} = \tau - p_{low}(\tau - \mu_L(\tau))$ as

the low threshold and $\tau_{high} = \tau - p_{high}(\tau - \mu_L(\tau))$ as the high threshold. Applying these two thresholds to the local region in the video frame results in two corresponding object masks M_{low} and M_{high}, as shown in Figure 3.2. A 3×3 median filter is applied to both binary object masks as in [15, 96] to reduce the impulsive noise before being merged by histogram backprojection described in the next subsection.

3.2.b Histogram Backprojection

In the underwater scenario, unstable lighting condition causes the low contrast in captured video data. As a result, the segmentation of fish is usually defective, especially around the boundary. It is thus desirable to refine the boundary of segmentation by comparing and aggregating two object masks according to their distributions of pixel values since they cover different amounts of background pixels [52].

To check whether a specific pixel $I(x, y)$ within the bounding region of an object candidate belongs to the foreground or the background, a histogram backprojection scheme is adopted as follows. First, two gray-level histograms $H_{low}(r)$ and $H_{high}(r)$ are computed according to the object masks M_{low} and M_{high}, respectively. A ratio histogram of any gray-level value r is defined as

$$H_R(r) = \min\left(\frac{H_{high}(r)}{H_{low}(r)}, 1\right). \tag{3.2}$$

Next, the ratio histogram is backprojected to the video frame domain, i.e., $BP(x, y) = H_R(I(x, y)),\ 1 \le x \le W,\ 1 \le y \le H$, where $I(x, y)$ denotes the pixel value at (x, y). A thresholding process is then applied to the backprojection of the ratio histogram $H_R(r)$, and the final binary segmentation mask $B(x, y)$ is given by

$$B(x, y) = \begin{cases} 1 & \text{if } BP(x, y) > \theta_{bp} \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$

where θ_{bp} denotes a threshold between 0 and 1. An illustrative example for the basic concept of histogram backprojection is shown in Figure 3.3.

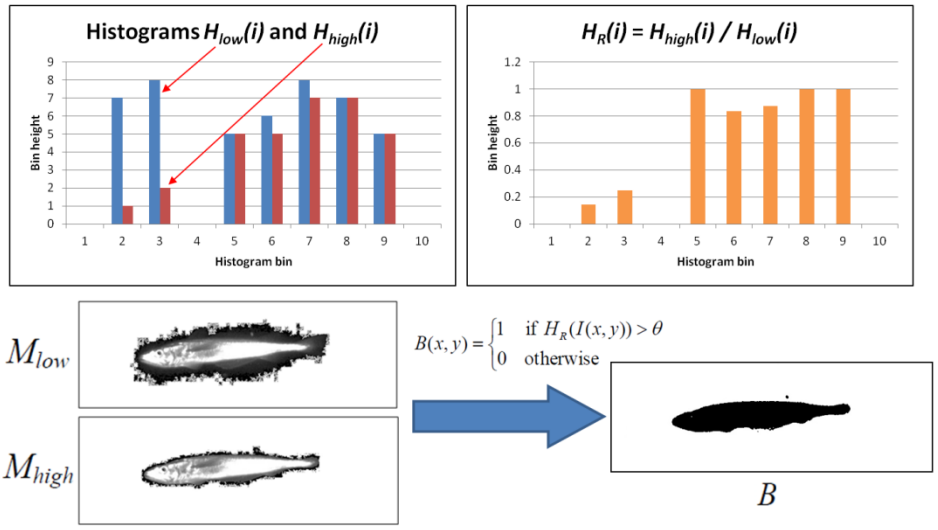


Figure 3.3: Basic concept of histogram backprojection

3.2.c Threshold by Area and Variance

In addition to using histograms to refine the segmentation masks, the proposed algorithm also takes into account the area of an object and variance of pixel values within an object. The connected components algorithm is applied to determine each isolated blob with its area. Those objects whose areas are greater than an upper threshold (corresponding to targets which are too close to the cameras with partial fish body capturing) or less than a lower threshold (corresponding to noise or very far away fish which cannot be reliably measured) will be rejected. Specifically, for each pixel (x, y) within the k-th segmented object O_k, its corresponding pixel on the foreground mask is modified by

$$B(x,y)=\begin{cases}1 & \text{if } \theta_A^L \le A(O_k) \le \theta_A^U \\ 0 & \text{otherwise}\end{cases},\ (x,y)\in O_k, \tag{3.4}$$

where $A(\cdot)$ gives the area of an object, and (θ_A^L, θ_A^U) are the lower and upper bound of the area to preserve.

Object candidates are also examined by calculating the variance of pixels within each segmented objects. Since foreground objects (fish) are inclined to be more textured than the background or unwanted objects, the variance of the segmented object is likely to be larger. The variance of pixels for an object is given by

$$\sigma_k^2=\frac{1}{A(O_k)-1}\sum_{(x,y)\in O_k}\left(I(x,y)-\bar{I}_k\right)^2,\ (x,y)\in O_k, \tag{3.5}$$

where $\bar{I}_k$ denotes the mean of pixel values of the *k*-th object. Given the variance, the foreground mask for this object is then thresholded by

$$B(x,y)=\begin{cases}1 & \text{if } \sigma_k^2 \ge \theta_V \\ 0 & \text{otherwise}\end{cases}. \tag{3.6}$$

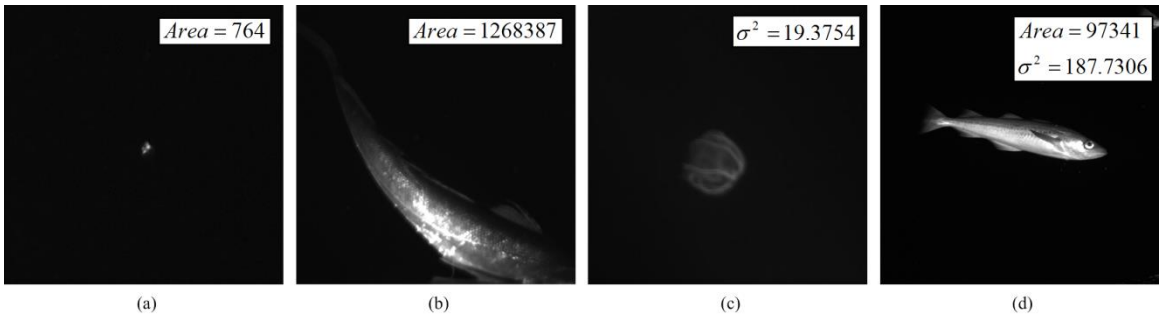


Figure 3.4: Example objects that are discarded due to (a) small area, (b) large area and (c) small variance of pixel values; (d) an object that are preserved after thresholding by area and variance. Note that (a) and (b) present the entire video frame, while (c) and (d) are zoomed-in regions.

3.2.d Post-processing

There may still exist some errors, such as gulfs or peninsulas, created at the boundaries of histogram backprojection refined objects. A cascade of morphological

operations can be adopted to further refine the segmentation boundaries. More specifically, a closing followed by an opening morphological operation with a disk structuring element is applied to the object mask. In the experiments, the size of the structuring element is empirically set as 7×7 pixels. In this way, the object boundaries are smoothed without affecting the details of the shape information.

3.3 Tracking from Low-Frame-Rate Video

Most moving objects in the low-frame-rate video are much more difficult to track due to their poor motion continuity and frequent entrance/exit. This makes most standard video object tracking methods as [12, 84] infeasible. An overview of the proposed multiple fish tracking algorithm is shown in Figure 3.5. The fish segmentation described in Section 3.2 is performed and followed by the proposed fast stereo matching method to match objects in two cameras. With the result of segmentation and stereo matching, fish are tracked by the feature-based temporal matching and the multi-target Viterbi data association. The proposed tracking algorithm exploits the temporal relationships throughout the target lifespan instead of only the current and previous frames. As a result, fish targets can be tracked in a more robust way, even when the motion is abrupt under the low frame rate.

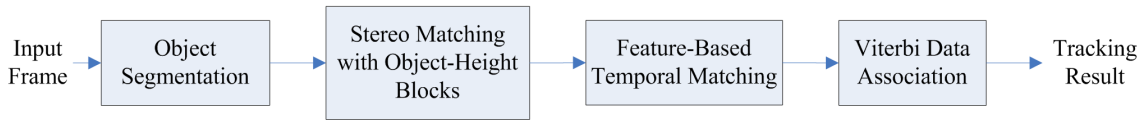


Figure 3.5: Flow chart of the proposed multiple fish tracking system.

3.3.a Stereo Matching with Object-Height Blocks

Stereo imaging provides target depth information and validate the result of object tracking. However, one major drawback of traditional dense stereo matching techniques is the intensive computations. When tracking fish using the proposed approach in the following subsections, stereo vision techniques are utilized for pairing the left and right targets. Therefore, with knowledge of the fish target location available in the

segmentation stage, an efficient block matching approach is proposed to find the stereo match for each target while reducing much of the computation.

Before stereo matching, preprocessing such as stereo calibration and stereo rectification are applied to each pair of input video frame. Through these steps, the camera views are undistorted and transformed so that each pair of corresponding epipolar lines in both cameras are exactly horizontal and align with each other in terms of vertical location. Stereo matching is hence reduced to a 1-D search problem along a horizontal line. It also ensures that the height of an object is equal in two camera views. In light of this, the notion of object-height blocks is introduced to speed up the matching process.

Given a segmented object in the left video frame, its upright bounding box is equally divided to 4 non-overlapping blocks horizontally, as shown in Figure 3.6. These blocks are referred to as *object-height blocks*. For each object-height block in the left video frame, the best match in the right video frame is determined by a simple block-matching algorithm based on the minimum sum of absolute difference (SAD) criterion. Note that an object-height block only searches along the horizontal line, and takes as candidates only those blocks within the upright bounding box of the object. That is, each target appearing on the horizontal line of the right video frame provides 4 candidates for a block from the left video frame to match. This results in great saving of computations with little loss of the accuracy, which is tolerable for the purpose of pairing objects in the left and right cameras. The object in the right view that has the minimum of the sum of 4 object-height blocks' SADs is then selected as the corresponding object of the left target.

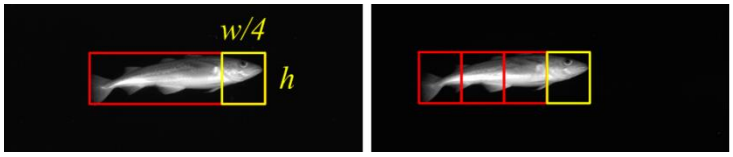


Figure 3.6: Object-height blocks (yellow) on the stereo-rectified left and right image. There are 4 candidates from each of the right target's upright bounding box for the minimum-SAD scheme.

3.3.b Feature-based Temporal Matching

The ubiquitous noise introduced by organic debris and abrupt movement of targets are the major difficulties for tracking in a low-frame-rate underwater video. An object matching approach is therefore developed for associating observations and targets. Various useful features are considered for measuring the similarity between objects. Given objects O_t^j in frame t and O_{t-1}^i in frame $(t-1)$, four cues are investigated as follows.

1) Vicinity cue

The Euclidean distance is given by $\left\|\mathbf{x}_t^j - \hat{\mathbf{x}}_t^i\right\|$, where $\mathbf{x}_t^j$ and $\hat{\mathbf{x}}_t^i$ denote the center point coordinates of the observation j and the prediction of target i at frame t, respectively. Details about target prediction obtained by a motion projection scheme are described in Section 3.3.c.

2) Area cue

In order to remove the factor of frame-by-frame discrepancy of target distance from cameras when calculating the area difference between objects, the object depth is calculated by stereo triangulation, and the target area is normalized accordingly as if all targets are placed at the same distance from the stereo cameras. The difference of area between the associated objects in two consecutive frames is supposed to be small. The object area, denoted as $A(\cdot)$, is calculated by the connected components algorithm. The difference of area between the object O_t^j in frame t and the object O_{t-1}^i in frame $(t-1)$ is then given by $\left|A(O_t^j) - A(O_{t-1}^i)\right|$.

3) Motion direction cue

Given two objects O_t^j and O_{t-1}^i to be matched, let the corresponding motion vector be $\mathbf{v}_t^{i,j} = \mathbf{x}_t^j - \mathbf{x}_{t-1}^i$. The direction of motion is then represented by the angle $\theta(\mathbf{v}_t^{i,j}, \mathbf{v}_{ref})$ between $\mathbf{v}_t^{i,j}$ and a predefined reference vector $\mathbf{v}_{ref}$, given by

$$\theta(\mathbf{v}_t^{i,j}, \mathbf{v}_{ref}) = \cos^{-1} \frac{\mathbf{v}_t^{i,j} \cdot \mathbf{v}_{ref}}{\left\|\mathbf{v}_t^{i,j}\right\| \left\|\mathbf{v}_{ref}\right\|}. \tag{3.7}$$

The predefined reference vector can be chosen according to the motion trend of fish schools or the movement of cameras.

4) Histogram distance

In addition to geometric features, the appearance (pixel intensity) also plays an important role. To exploit the dissimilarity of intensity distribution between two objects, the earth mover's distance (EMD) [76] is computed as the distance metric between 16-bin gray-level histograms.

Combining all the four cues above, the likelihood for object temporal matching between O_t^j and O_{t-1}^i is given by

$$\begin{aligned} P(O_{t-1}^i \left| O_t^j \right.) &= \exp\left(-\frac{\left\|\mathbf{x}_t^j - \hat{\mathbf{x}}_t^i\right\|^2}{{\sigma_v}^2}\right) \cdot \exp\left(-\frac{(A(O_t^j) - A(O_{t-1}^i))^2}{{\sigma_a}^2}\right) \\ &\quad \cdot \exp\left(-\frac{(\theta(\mathbf{v}_t^{i,j}, \mathbf{v}_{ref}))^2}{{\sigma_m}^2}\right) \cdot \exp\left(-\frac{(EMD(O_t^j, O_{t-1}^i))^2}{{\sigma_h}^2}\right) \\ &= \exp(-z_v) \cdot \exp(-z_a) \cdot \exp(-z_m) \cdot \exp(-z_h) \\ &= \exp(-(z_v + z_a + z_m + z_h)), \end{aligned} \tag{3.8}$$

and the "matching cost" is defined as

$$c_{ij}(t) = -\ln P(O_{t-1}^i \mid O_t^j) = z_v + z_a + z_m + z_h. \tag{3.9}$$

The $\{\sigma\}$ values in (3.9) denote the features' standard deviations. These standard deviations are calculated systematically by collecting the feature values for all temporal matching candidates in each frame of our video data. This cost is assigned to the edge between O_t^j and O_{t-1}^i in the trellis for Viterbi data association.

3.3.c Viterbi Data Association

In the proposed data association system, the stereo information is utilized by regarding a pair of stereo-matched object, i.e., the same object in the left and right video frame, as one observation for tracking. The matching cost of temporal matching is then given by the sum of the costs from two cameras, i.e., $c_{ij}^{stereo}(t) = c_{ij}^{L}(t) + c_{ij}^{R}(t)$. The advantage of combining a stereo pair of objects can be seen from two aspects. First, the stereo-matched objects are ensured to be bound together during tracking, so the mismatch between tracking and stereo vision is greatly reduced. Also, an object pair is considered the optimal candidate for a tracked target only if this object pair matches well in both the left and right images. In other words, pairing of objects with stereo cameras enables an implicit tracking validation, which is not available in the single camera scenario. This is especially helpful for improving tracking performance in our scenario, where targets to be tracked can be visually similar.

To exploit temporal correlations of objects across several past frames, a multiple-target Viterbi data association algorithm is introduced. Algorithm 3.1 summarizes the procedure for each video frame.

Algorithm 3.1: Viterbi data association iteration at each frame

1. **input:** time t , objects $\{O_t^j\}_{j=1}^{|N(t)|}$, trellises $\{\mathbf{T}^k\}_{k=1}^{m}$
2. **output:** target paths $P(t)$ ending at time t
3. initialize the object selection set $S \leftarrow \varnothing$
4. **for** $k = 1$ to m **do**
5. estimate $c_{ij}(t)$ for all i, j based on (3.9)
6. update $\mathbf{T}^k$ with $\pi_j(t), C_j(t)$ based on (3.11) and (3.12)
7. $S \leftarrow S \cup \arg\min_j C_j^{stereo}(t)$
8. **if** target k exits the FOV **then**
9. $P(t) \leftarrow P(t) \cup$ GetPathByBacktracking($\mathbf{T}^k$)
10. **end if**
11. **end for**
12. create a new trellis $\mathbf{T}^j$ for each $O_t^j \notin S$

1) Basic idea

The Viterbi data association [2, 71, 74] is performed based on a trellis of the observations in each frame. As illustrated in Figure 3.7, a trellis is a type of directed graph in which nodes are partitioned into ordered subsets $N(t) = \{n_j(t) \mid j = 1, 2, ..., |N(t)|\}$ for $t = 1, 2, ..., T$, and edges $a_{ij}(t)$ lie between any pair of node in adjacent subsets $\{n_i(t-1), n_j(t)\}$. Nodes in a subset represent objects in one frame, and each edge is assigned a cost $c_{ij}(t)$. The total cost of a path (a sequence of edges) is then given by

$$C(P) = \sum_{t=2}^{T} c_{p_{t-1}p_t}(t),$$

$$\text{where } P = \{a_{p_1p_2}(2), a_{p_2p_3}(3), ..., a_{p_{T-1}p_T}(T)\}. \tag{3.10}$$

The Viterbi algorithm [36] is applied here to find the minimum-cost path during single-target tracking. For every observation a node is initialized with zero cost and a null predecessor. In each iteration, the matching cost for every node $n_j(t), j = 1, 2, ..., |N(t)|$ is given by (3.9). Then the predecessor and accumulated cost are assigned to node $n_j(t)$:

$$\pi_j(t) = \arg \min_{1 \le i \le |N(t-1)|} C_i(t-1) + c_{ij}(t), \tag{3.11}$$

$$C_j(t) = C_{\pi_j(t)}(t-1) + c_{\pi_j(t)j}(t). \tag{3.12}$$

Once the target leaves the FOV (as established by distance from video frame borders, see below), i.e., the final stage of trellis is reached, a backtracking step is performed. Starting from the minimum-cost node in the final stage, the optimal path $P^* = \arg\min_P C(P)$ is recovered by traversing backward to the first stage according to the predecessors stored at each stage.

2) Multiple-target case

A multiple-target Viterbi data association algorithm is proposed for our case of low-frame-rate fish tracking [24]. Since the starting frame may differ among targets, the

predecessor and minimum cost at each node may also differ. Therefore, a separate trellis is created for every target to track. Data association is then performed separately by following (3.11) and (3.12) for each target with all observations, as shown in Figure 3.7. Observations that are not associated with any target corresponds to new targets or false alarms. New targets tend to appear close to the video frame border in their first frame. Also, most false alarms here are generated during the post-processing of segmentation, so their areas are usually small. An examination on position and area of the observations is thus performed to distinguish new targets from false alarms. As for exiting targets, the abrupt movement and short lifespan make the criteria of track creation and deletion less reliable. Therefore, a track is restricted to end only when its predicted position is close to the frame border. This also prevents targets from being deleted if they are temporarily occluded. If a track is lost before approaching the frame boundary, the predicted position is used as the actual position and the velocity remains. A new prediction is then made for the next frame.

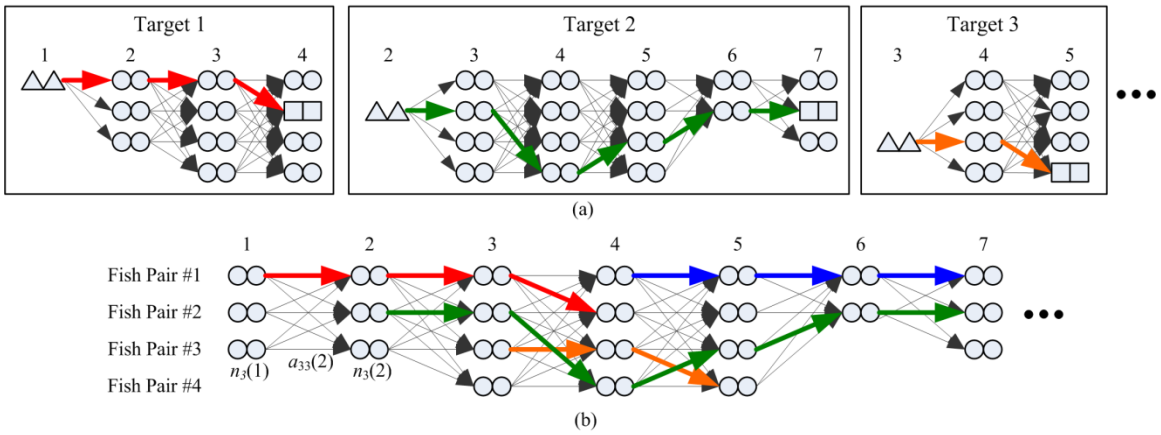


Figure 3.7: Multiple-target Viterbi data association. (a) Each target maintains a separate trellis during its lifespan with its own starting node (triangles) and ending node (squares). The optimal path in each trellis is labeled by colored arrows. (b) An overall trellis showing several paths from targets in (a).

Note that occlusions are inherently handled by our method since paths in different trellises may share the same nodes. In terms of the overall trellis in Figure 3.7(b), nodes

in any stage in the diagram are allowed to be included by more than one path, which means the object in this frame is occluded by others.

In each frame, a motion projection mechanism is utilized to estimate and update the position of the tracked target. Given the position $\mathbf{x}_{t-1}^{k}$ and velocity $\mathbf{v}_{t-1}^{k}$ of the *k*-th tracked target from frame $(t-1)$, the predicted position at the current frame is given by $\hat{\mathbf{x}}_{t}^{k} = \mathbf{x}_{t-1}^{k} + \mathbf{v}_{t-1}^{k}$. After the data association, the observation node with the minimum cost is chosen to update the position and velocity by

$$\mathbf{x}_{t}^{k} = \mathbf{x}_{t}^{j^*}, \tag{3.13}$$

$$\mathbf{v}_{t}^{k} = \alpha \mathbf{v}_{t}^{j^*} + (1-\alpha)\mathbf{v}_{t-1}^{k}. \tag{3.14}$$

where $\mathbf{x}_{t}^{j^*}$ and $\mathbf{v}_{t}^{j^*}$ denote separately the position and velocity of the minimum-cost observation and α is the update rate.

Rather than using a batch scheme as in [74], which accumulates a fixed number of frames and perform backtracking periodically, in the proposed scheme backtracking is performed and the optimal sequence of observations is recovered once there is a target leaving the FOV. The proposed system is thus able to not only perform online tracking but also exempt from potential failures due to the gaps between batches.

3.4 Experimental Results

3.4.a Implementation Details

Simulations of the proposed system are carried out on several 8-bit grayscale video clips recorded underwater by the stereo cameras on Cam-trawl. The frame size is 2048×2048 pixels, and the frame rate is 5 frames per second (fps). Before the object segmentation stage, a pre-processing is performed to eliminate the trawl web in video frames. The trawl web behind the targets appears as white diagonal grids spreading over the top and bottom regions in the stereo camera views (see Figure 3.8). Its drift with the

current makes background modeling approaches ineffective in this scenario. As a preprocessing, the web pattern is successfully removed by applying morphological opening operations with main diagonal (from top-left to bottom-right) and anti-diagonal (from top-right to bottom-left) structuring elements with length of 7 pixels. For the rest of the proposed algorithm, the structuring element for all morphological operations is empirically determined as a disk with size 7×7 pixels. Values of parameters that are determined empirically in our experiments are provided in Table 3.1. The whole process is fully automatic and requires no manual intervention. Tracked fish are labeled with numbers and bounding boxes with different colors in order to make them differentiable.

Table 3.1: Values of parameters used in the experiments

Symbol	Value	Description
Automatic Fish Segmentation		
p_{low}	1	Shifting factor of low threshold
p_{high}	0.7	Shifting factor of high threshold
θ_{bp}	0.3	Threshold of histogram backprojection
θ_A^L	2×10^3	Lower limit of object area
θ_A^U	10^6	Upper limit of object area
θ_V	30	Threshold of object variance
Multiple-Fish Tracking		
$\mathbf{v}_{ref}$	$(-1,0)$	Reference motion vector
α	0.3	Update rate for target velocity
M	100	Margin from frame border for target entrance/exit

3.4.b Results of Segmentation

The proposed segmentation algorithm is tested with three sample video sequences consisting of 74 frames in total to evaluate its performance. According to the hand-labeled ground truth, there are 514 fish in total to be segmented. The fish length is

defined as the Euclidean distance between head and tail. Note that fish lengths are estimated only from a single camera in this experiment. In Chapter 5, fish lengths measured from stereo cameras will be discussed and compared with the single-camera case.

The performance of fish segmentation is measured in terms of precision and recall accuracy as well as the mean absolute percentage error (MAPE) of the measured length of "large targets", which are the ones with length greater than 100 pixels, since they have more reliable ground truth. There are 189 large targets out of 514 targets in the testing set. In this experiment, the length of a fish object is estimated by finding its oriented bounding box. The measured fish length equals to the maximum between the width and height of the oriented bounding box.

As shown in Table 3.2, the proposed algorithm achieves a 74.6% precision and a 78.4% recall under very low-contrast underwater videos. The MAPE of measured length out of 189 large targets is 10.7%, as shown in Table 3.3. This shows that the proposed segmentation algorithm gives quite accurate information of fish silhouette for the use of stereo matching as well as target tracking. A major source of error is the fish cropping that happens often because of the low reflectivity of caudal fins. Based on this, the performance of length estimation and the object detection is further enhanced by the stage of stereo matching and multiple-fish tracking, respectively, as described in the following subsections.

Table 3.2: Precision and recall of fish segmentation

Num. of Targets	Precision	Recall
514	0.746	0.784

Table 3.3: Mean absolute percentage error of large target length

Num. of Large Targets	MAPE of Length
189	10.7

3.4.c Results of Tracking

The proposed system is used to track multiple fish targets simultaneously in several sample video clips. All these video clips are grayscale and recorded underwater by the stereo cameras on Cam-trawl. Some tracking statistics of the proposed system comparing with other data association methods are listed in this subsection.

From Table 3.4, the precision and recall of fish detection in sample video clips containing 62 distinct fish targets are 98% and 94%, which shows that the proposed system enhances greatly the accuracy of underwater fish detection by utilizing temporal information. In Table 3.5, the performance of the proposed system is evaluated and compared with other data association algorithms. Here, the tracking success rate is defined as the ratio of correctly tracked targets to correctly detected targets, i.e.,

$$\text{tracking success rate} = \frac{\text{\# of targets correctly labeled}}{\text{\# of targets correctly detected}}. \tag{3.15}$$

One can see that the proposed system, i.e., matching cost defined in (3.9) plus Viterbi data association (MC+VDA) outperforms other data association methods, including the state of the art proposed by Spampinato et al. [80]. The conventional nearest neighbor (NN) suffers from poor motion continuity and short lifespan of targets, and thus tracks fish in a low success rate. The shape-based method (Shape) finds the candidate with the most similar shape to the target in each frame. In the experiment this method is based on the Hu's invariant moments [48], which are known for their robustness against image translation, rotation and scaling. Shape-based method thus performs slightly better than the nearest neighbor approach. However, unlike pedestrians or vehicles, fish in an unconstrained environment are considered deformable bodies since their poses change abruptly during common actions such as swimming and turning. This is obvious especially when the video is captured at low frame rate, where the time interval between two consecutive frames is long (0.2 seconds in our data). For this reason, shape does not serve as a good feature to track fish in this case.

The other compared method proposed in [80] exploits several features, including position, motion vector, area and orientation, which are similar to those used in the proposed method. However, same as conventional tracking algorithms, the temporal correlation of targets is only considered over two frames. This makes it less successful in the low-frame-rate case, where the motion can be very abrupt. Note that the proposed matching cost (MC) is also tested alone to show the effectiveness of matching objects based on various types of features. Without the Viterbi data association, the object matching scheme itself gives a tracking success rate close to Spampinato's method [80]. By incorporating the Viterbi data association (VDA), our proposed method shows a much improved success rate for multiple target tracking.

The qualitative result in three consecutive frames with tracked fish labeled on both sides of stereo video frames is shown in Figure 3.8, where comparative results from two other data association methods are also displayed. It clearly demonstrates the robustness of the proposed system. The stereo videos and our tracking results based on the oriented rectangular bounding boxes along with the ground truth are available online[1], so that people can further develop useful algorithms to improve the performance.

Table 3.4: Performance of fish detection over video clips

Num. of Targets	Precision	Recall
62	0.98	0.94

Table 3.5: Tracking success rate vs. data association methods

Clip	NN	Shape	[80]	MC	MC+VDA
1	0.38	0.31	0.66	0.47	**0.94**
2	0.42	0.50	0.58	0.58	**0.83**
3	0.29	0.44	0.44	0.50	**0.86**
Avg.	0.36	0.41	0.56	0.54	**0.88**

[1] URL: http://allison.ee.washington.edu/mengchec/camtrawl

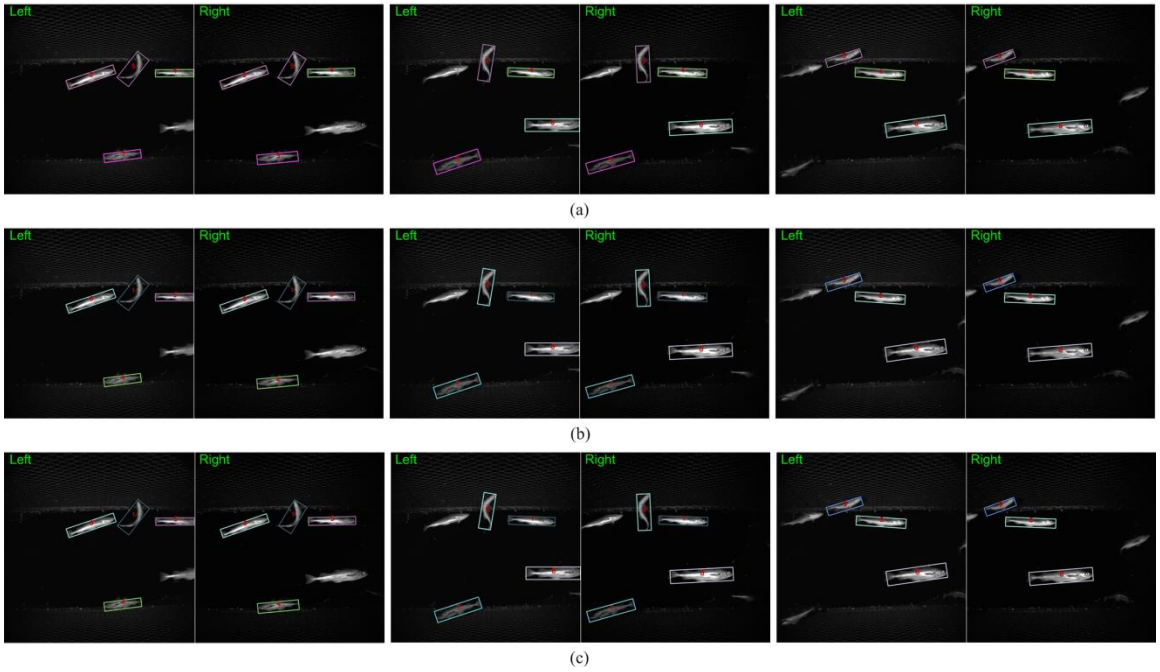


Figure 3.8: Tracking multiple fish in an underwater video clip. Targets are labeled by numbers and oriented bounding boxes with different colors. (a) Using the proposed algorithm with matching cost plus Viterbi data association. (b) Using matching cost only. (c) Using primitive nearest neighbor data association.

3.5 Discussion

3.5.a Histogram Backprojection

As discussed in Section 3.2.b, the contrast between the objects and the background is very low in underwater video data due to unstable lighting condition. This results in defective fish segmentation around the boundary. The boundary of the segmentation is successfully refined by utilizing the statistical distributions of pixel intensities in the neighborhood of an object. From the proposed double local thresholding, the low and high object masks have different sizes. Specifically, the low-threshold mask has a larger segmentation area, so it covers the foreground pixels better near the object boundary but meanwhile includes more background pixels. The high-threshold mask has a smaller segmentation area than the low mask, so it avoids covering background pixels but might

lose a small portion of the foreground. Such differences in foreground and background pixel coverage are reflected in the mask histograms $H_{low}(r)$ and $H_{high}(r)$, where $H_{low}(r)$ consists of larger histogram values in the small gray-level bins (corresponding to background pixels) while $H_{high}(r)$ consists of lower histogram values in the small gray-level bins, as shown in Figure 3.3 and Figure 3.10(c). The background pixels take a greater proportion in the low mask, so the bin height ratio $H_{high}(r)/H_{low}(r)$ is small for bins representing the background pixels. On the other hand, the foreground pixels take similar or equal proportion in two masks, so the bin height ratio $H_{high}(r)/H_{low}(r)$ is close to 1 for bins representing the foreground. Therefore, the ratio histogram $H_R(r)$ defined in (3.2) can be viewed as a confidence level of the pixel belongs to the foreground, and its backprojection provides a convenient way to examine each pixel around the boundary of the segmented object.

The effectiveness of segmentation boundary refinement based on histogram backprojection depends on the number of bins in each histogram as well as the thresholding value θ_{bp} for the ratio histogram in (3.3). To better understand the impacts of these parameters, the segmentation performance is compared by using 4, 8, 16 and 32 bins for each histogram with threshold values as 0.1, 0.3, 0.5, 0.7 and 0.9. Following the experiments in Section 3.4.b, the performance is measured by the mean absolute percentage error (MAPE) of length obtained by segmentation for 189 large fish targets.

A sensitivity test on histogram bins and thresholds is reported in Figure 3.9. One can see that using 16-bin histograms gives the lowest error in fish length estimation. The main reason is that a discriminative and robust representation for object appearance is crucial in low contrast and noisy underwater imaging. A histogram with fewer bins (e.g. 8 bins) mixes pixel values together and reduces the discrimination; a histogram with more bins (e.g. 32 bins) enhances the accuracy but become highly sensitive to noise. The

16-bin histogram groups similar pixel values appropriately, so it represents the object in a way which is not only discriminative but also robust against noise.

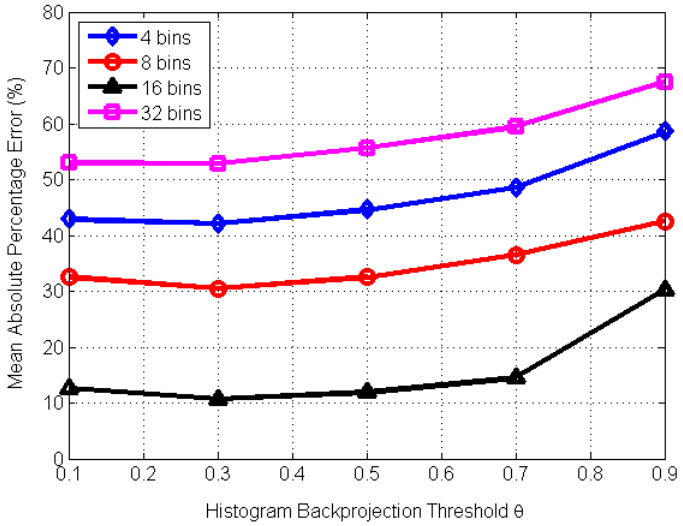


Figure 3.9: Sensitivity test of histogram backprojection. Each color shows the error rates by using different numbers of histogram bins. Each labeled point shows the error rate by setting different values for the ratio histogram threshold.

Moreover, one can see from Figure 3.9 that the segmentation performance is rather insensitive to the threshold value θ_{bp} of ratio histogram. The reason can be seen from an example using 16 bins shown in Figure 3.10. After our double local thresholding approach, almost all pixels covered by the high mask are also covered by the low mask. This gives the corresponding ratio histogram bins with heights being close to 1. On the other hand, the low mask covers much more dark background pixels than the high mask does (see the two leftmost bins in Figure 3.10(c)), so the heights of corresponding ratio histogram bins are close to 0. As a result, there is little difference in the performance as long as the threshold value is around the middle of the interval [0, 1].

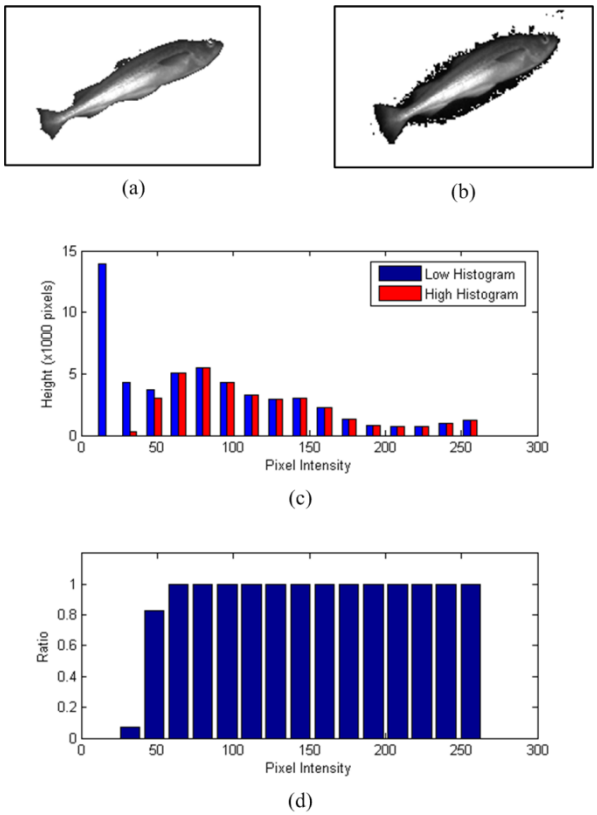


Figure 3.10: An example target and its histograms generated by the proposed double local thresholding: (a) Low mask of the target; (b) high mask of the target; (c) two histogram from low and high mask, respectively; (d) ratio histogram given by (2). One can see that most ratio histogram bins are close to either 0 or 1. As a result, the performance of segmentation is less sensitive to the selection of ratio histogram threshold.

3.5.b Features for Temporal Object Matching

The major challenge to tracking in low frame rate video is abrupt motion and short lifespan of targets. Traditional object tracking approaches such as particle filtering and kernel tracking fail to track in this case since they rely on the assumption of smoothness in motion trajectory, i.e., small incremental motion per frame. As a result, an approach related to object matching across frames is developed.

In the proposed object temporal matching method, four different features are considered for measuring the similarity between objects in the current and previous frame. In order to further investigate the impact on tracking performance from each feature, a sensitivity test on these cues is carried out as follows. Using the same video data for

evaluating tracking performance, one cue is removed from the matching cost function defined in (3.9) each time. The results of temporal object matching using partial cues are used by the subsequent multiple-target Viterbi data association. Finally, tracking success rate given in (3.15) for each video clip is calculated as a performance metric of multiple-target tracking. In this evaluation, it is expected that matching objects with all cues gives the best performance, and discarding one of the cues results in a decreased performance. The importance of each removed cue is then reflected by the amount of drop in tracking success rate in the low frame rate video sequences.

Figure 3.11 reports the tracking performance with different removed cues in the matching cost function. An average success rate for each unselected cue is also given in addition to that for three testing video clips. As shown in Figure 3.11, the vicinity cue has the greatest impact on tracking performance, followed by the motion direction cue. This implies that the characteristics of motion, in spite of the poor continuity, is still informative to associate objects across frames. The concept of motion direction cue represents the directional property of not only the motion vector between two consecutive frames, but also the overall trajectory of a target throughout its lifespan. This is fully exploited by the Viterbi data association method since the dynamic programming approach provides a global optimization along the time horizon.

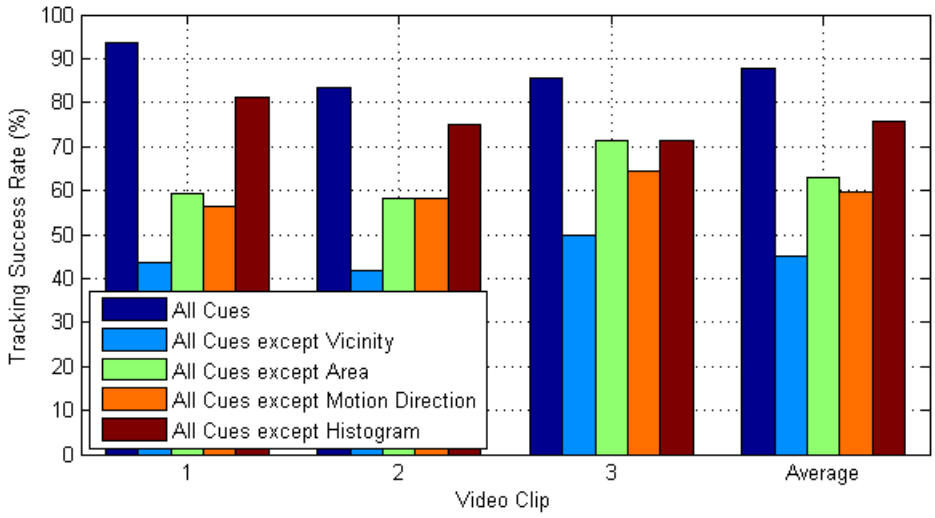


Figure 3.11: Tracking success rate vs. unselected cues for temporal matching. Each color bar represents one cue that is not used when matching objects across video frames.

3.5.c Target Occlusion

Fish swimming through the nearly-unconstrained trawl often occlude each other in the underwater video. The low contrast imaging leads to fuzzy edges between two fish bodies when one fish overlaps the other. It is therefore difficult to deal with occlusions in fish segmentation. As discussed in Section 3.3.c, occlusions are inherently handled by our method by allowing paths in different trellises to share the same nodes. However, overlapping fish are viewed as one objects instead of separated targets during data association. One may thus be interested in whether the performance can be improved if occlusions are handled according to object depth before tracking.

To investigate into this, the disparity refinement used for fish length measurement before tracking is performed. Based on the results of object-height block matching, the disparity is refined by using a dense grid of blocks with size 8×8 pixels. Block matching approach with maximum-SAD criterion is employed, same as in coarse matching. A simple Canny edge detection is then applied to the Gaussian-smoothed fine disparity map to separate objects when they are occluded.

For simplicity, video clip 3 is tested to see the influence of target occlusions. Results on fish tracking and length measurement are shown in Table 3.6. The success rate in tracking is the same with or without performing occlusion handling. This is expected since the proposed Viterbi data association allows paths from different trellises to intersect at some frame and even overlap for several frames. These correspond to one-frame and multiple-frame occlusion, respectively. As for length measurement, the MAPE is even higher after introducing occlusion handling. Despite the errors in measurements, this is caused by the fact that occluded fish are seriously underestimated in length after separation, even when the lengths of those fish which occlude others are measured more accurately.

Table 3.6: Performance vs. occlusion handling on video 3

Method	Tracking Success Rate	MAPE of Length
w/o occlusion handling	0.94	0.06
w/ occlusion handling	0.94	0.07

Chapter 4 – Segmentation and Tracking: Uncontrolled Background

In this chapter, a novel tracking and segmentation algorithm is proposed based on deformable multiple kernels (DMK) and active contours to address the issues introduced by camera motion.

4.1 Review of Basic Concepts

Before introducing the proposed method, a recap is given in this section for the basic concepts of two related techniques, namely kernel-based tracking and deformable part model (DPM) object detection.

4.1.a Kernel-Based Tracking

As one of the major categories of object tracking techniques, the kernel-based method has shown its advantages in relatively low computation and robustness against non-rigid deformation [28]. The basic concept of kernel-based tracking is iteratively computing the kernel motion so that the feature distribution between the candidate and the target is best matched. This can be formulated as a search for the position that maximizes a density estimator

$$f(\mathbf{x};h)=\frac{\sum_{i=1}^{N_h} w_i k(\left\|\frac{\mathbf{x}-\mathbf{z}_i}{h}\right\|^2)}{\sum_{i=1}^{N_h} k(\left\|\frac{\mathbf{x}-\mathbf{z}_i}{h}\right\|^2)}, \tag{4.1}$$

where $\mathbf{x}$ is the kernel center, $\mathbf{z}_i$ is the i-th pixel position within the kernel, w_i is the sample weight of $\mathbf{z}_i$, $k(\cdot)$ is the kernel function with a bandwidth h and N_h is the total number of pixels within the kernel.

In general, the target and the candidate model are represented as the probability density functions of features, i.e., a color histogram where the contribution of each pixel

is spatially weighted by the kernel function $k(\cdot)$. The optimal kernel position is efficiently found by applying the mean-shift algorithm [27]. More specifically, given the current kernel position $\mathbf{x}$, the new kernel position is computed by $\mathbf{x}' = \mathbf{x} + \mathbf{m}(\mathbf{x})$, where

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{N_h} w_i k'(\left\| \frac{\mathbf{x}-\mathbf{z}_i}{h} \right\|^2)(\mathbf{z}_i - \mathbf{x})}{\sum_{i=1}^{N_h} w_i k'(\left\| \frac{\mathbf{x}-\mathbf{z}_i}{h} \right\|^2)} \in \mathbf{R}^2, \tag{4.2}$$

is referred to as the *mean-shift vector*.

One major challenge to conventional kernel-based tracking is partial occlusion, i.e., a part of the target is behind some obstacles or other targets. The occluded part is not visible and thus introduces error when computing histogram similarity. Chu et al. [18] proposed the constrained multiple-kernel (CMK) tracking method to address the issue. In the CMK formulation, a target is represented by a number of kernels that comply with predefined spatial constraints. By associating adaptive weights to the kernels, the adverse effect introduced by partial occlusion can be compensated. The kernel motion is given by minimizing the sum of cost functions $J_i(\mathbf{x})$ subject to the constraints, i.e.,

$$\min_{\mathbf{x}} \sum_{i=1}^{n} w_i J_i(\mathbf{x}), \tag{4.3}$$

$$\text{s.t. } C(\mathbf{x}) = 0 \tag{4.4}$$

where w_i is the weight for the *i*-th kernel. A projected gradient approach is applied to solve the above equality-constrained optimization problem.

4.1.b Deformable Part Model

The deformable part model (DPM) [34] has been regarded as one of the most generic and powerful object detectors even for challenging scenarios by capturing significant intra-class variations. The DPM uses discriminative training and learns object appearance based on the pictorial structure [35], which represents an object with a collection of parts in a deformable configuration.

The DPM for an object category consists of a root filter $\mathbf{F}_0$, n part models $P_1,...,P_n$ sampled at twice the resolution of the root filter, and a real-valued bias term b. The i-th part model P_i is defined by a 3-tuple $(\mathbf{F}_i, \mathbf{v}_i, \mathbf{d}_i)$, where $\mathbf{F}_i$ is the part filter, $\mathbf{v}_i \in \mathbf{R}^2$ specifies the anchor position for the part relative to the root position, and $\mathbf{d}_i \in \mathbf{R}^4$ denotes coefficients of a quadratic deformation cost function of part displacement.

Detecting objects in an image is done by applying linear filtering to a feature pyramid, which consists of dense feature maps computed from each layer of a standard image pyramid. A variation of the histogram of oriented gradients (HOG) [30] is used as the feature vector. Let H be a feature pyramid and $\mathbf{p} = (\mathbf{x}, l) = (x, y, l)$ denote a specific position and level in the pyramid. Define an object hypothesis as the location of each filter in the feature pyramid, $\mathbf{z} = (\mathbf{p}_0,...,\mathbf{p}_n)$, with the constraint that all parts are placed at twice the resolution of the root. The detection score of a hypothesis $s(\mathbf{p}_0,...,\mathbf{p}_n)$ is then given by the sum of all filter responses at their respective position subtracted by the deformation cost of each part, and plus the bias term, i.e.,

$$s(\mathbf{p}_0,...,\mathbf{p}_n) = \sum_{i=0}^{n} \mathbf{F}_i \cdot \phi(H, \mathbf{p}_i) - \sum_{i=1}^{n} \mathbf{d}_i \cdot \phi_d(d\mathbf{x}_i) + b, \tag{4.5}$$

where $\phi(H, \mathbf{p}_i)$ denotes the feature vector obtained from the location $\mathbf{p}_i$ in the feature pyramid, $d\mathbf{x}_i = (dx_i, dy_i) = \mathbf{x}_i - (2\mathbf{x}_0 + \mathbf{v}_i)$ denotes the displacement of the i-th part with respect to its anchor position (note the factor 2 is due to double-resolution sampling of the part models), and $\phi_d(d\mathbf{x}) = (dx^2, dx, dy^2, dy)$ is the deformation features that express the displacement in the quadratic form. The bias b is introduced to accommodate the constant term when training the final binary classifier for $s(\mathbf{p}_0,...,\mathbf{p}_n)$. A latent SVM formulation is proposed to train the DPM parameters $\{\mathbf{F}_0, P_1,..., P_n, b\}$ for the object category of interest.

4.2 Deformable Multiple-Kernel Tracking

Inspired by the notion of the multiple-kernel representation [18], the proposed DMK tracking regards each part model as a kernel. Like the predefined constraints that bind the kernels in the CMK formulation, the DMK algorithm adopts the deformation costs to restrict the displacement of kernels during tracking. For each video frame, the proposed algorithm iteratively shift the kernels based on weighted color histogram, texture histogram and HOG. As a result, the DMK approach takes advantage of not only low computational cost from the kernel-based tracking but also the robustness of object localization from the DPM detection.

A flow chart of the proposed deformable multiple-kernel tracking algorithm is shown in Figure 4.1. Given the detected object, a multiple-kernel target model is initialized such that the kernels correspond to root and part filters from the deformable part model (DPM). The location and size of each kernel are updated via the mean-shift algorithm for spatially-weighted color histogram, texture histogram and HOG features. The deformation cost is imposed in the HOG mean-shift stage to maintain the part configuration. If the matching score in the HOG mean-shift stage is low, the target model is reset. Finally, the tracking result (in terms of bounding box location and size) is produced by a kernel aggregation method.

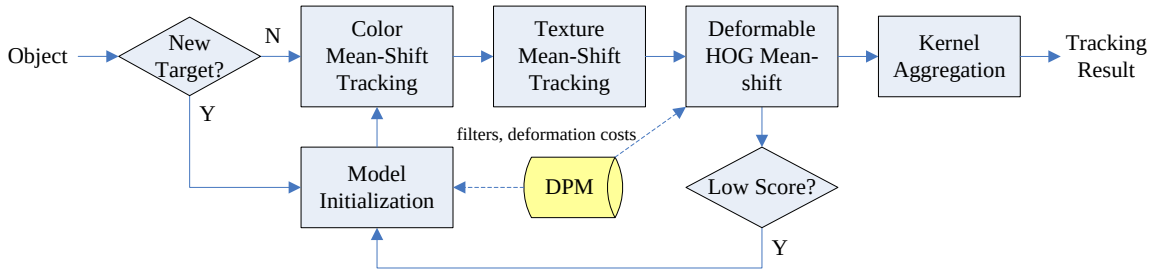


Figure 4.1: Flow chart of the deformable multiple-kernel (DMK) tracking algorithm.

4.2.a Models

In practice, the DPM represents an object category by a mixture of star models, which consists of a number of components [34]. Each component is trained by a group of positive samples with similar bounding box aspect ratio, which serves as a simple indicator of intra-class variation in perspective. An example of the mixture DPM for fish is shown in Figure 4.2(a). As a result, the deformable multiple-kernel (DMK) model is initialized by using the most suitable component as follows.

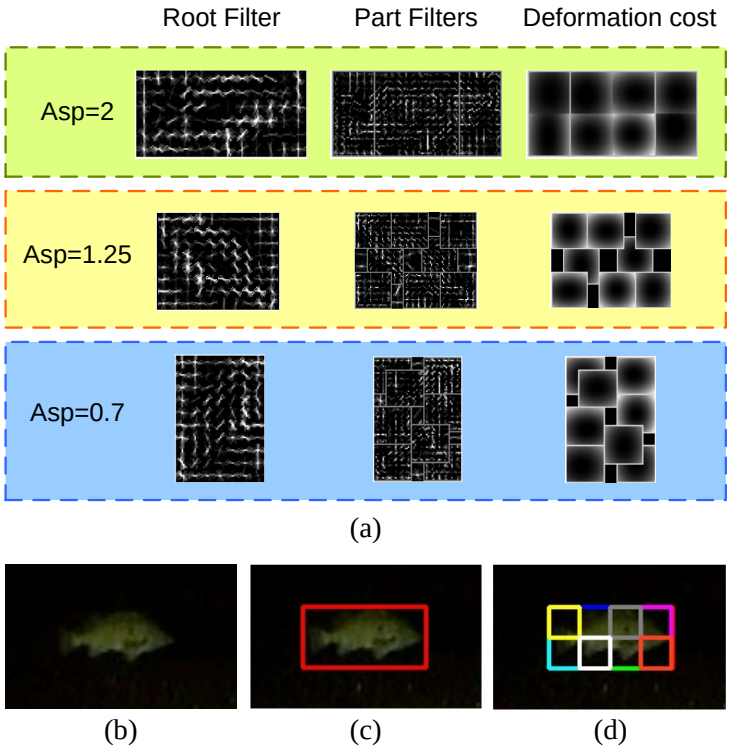


Figure 4.2: Deformable multiple-kernel model (a) the fish DPM with 3 components, each of which is presented with a specific aspect ratio of fish body to capture different viewing perspective. One DPM component consists of a root filter and 8 part filters (sampled at twice the resolution of root filter); (b) a target object in the video; (c) root kernel corresponding to the root filter; (d) part kernels corresponding to the part filters and located according to the deformation cost map.

The DMK model represents a target by $n+1$ kernels. First, the root kernel is set identically to the object bounding box. Let R_i and C_i be the number of cell rows and

columns for the *i*-th filter. The DPM component with the root filter aspect ratio C_0 / R_0 most similar to the bounding box aspect ratio w/h is selected. Based on the selection, each part kernel is then placed at its anchor position. The size of each part kernel is scaled by the size ratio between the part and root filter such that $h_i / h_0 = R_i / R_0$ and $w_i / w_0 = C_i / C_0$ for each part kernel size (w_i, h_i). An example of multiple kernels for an object is shown in Figure 4.2(b)–(d). For kernel features, we use the color histogram, texture histogram and HOG, which are described in the following subsections.

4.2.b Tracking by Color and Texture

In kernel-based tracking [28], a target is represented by a color histogram where each pixel's contribution is spatially weighted by a kernel function. The kernel motion is efficiently computed by maximizing the histogram similarity between a candidate model and the given target model via the mean-shift algorithm.

The weakness of this approach is that the similarity metric is not reliable when the target color is not distinct. To improve this, the texture attribute is also utilized as an additional feature. We use the rotation-invariant uniform type of local binary patterns (LBP) histogram proposed in [68]. This type of pattern is derived such that the feature descriptor is not only invariant to rotation but also robust to noisy high-frequency patterns. Similar to color, an LBP histogram is constructed with the spatial weight given by the kernel function. Each kernel is then moved to its optimal location by applying the mean-shift algorithm separately. The mean-shift algorithm is iterated until the maximum iteration (T_{color} for color and T_{texture} for texture) is reached.

4.2.c Deformable HOG Mean-Shift

After the color and texture kernel tracking, part kernels are likely to move away from their anchor positions. To restore the object part configuration, these part kernels are further shifted to optimize the HOG similarity while being bound with the root kernel by the deformation cost functions, which are similar to the imposed constraints used in the

CMK tracking. The effect of restoring part configuration by the proposed deformable HOG mean-shift algorithm is demonstrated in Figure 4.3.

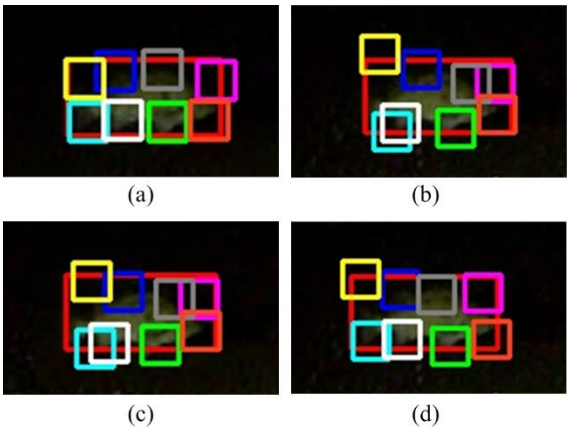


Figure 4.3: The root kernel (red) and part kernels (other colors): (a) from the previous frame; (b) after color mean-shift; (c) after texture mean-shift; (d) after HOG mean-shift, which considers the deformation costs. Note the restoration of part configuration according to the DPM (such as the cyan and white kernels on the bottom left corners).

To ensure the accuracy of HOG feature matching, we first calculate the object scale, i.e., the feature pyramid level where the object is detected. Following the notation in Section 4.2.a, given a root filter with R_0 rows and C_0 columns of cells, the image pyramid level of the object's presence is

$$l_0 = \lfloor \lambda \cdot \min\{\log_2(w_0/kC_0), \log_2(h_0/kR_0)\} \rfloor, \tag{4.6}$$

where (w_0, h_0) is the detected width and height of the object and k denotes the HOG square cell size in pixels. Following the standard setting in [30] the cell size is set as $k = 8$ pixels. In practice, there are $\lambda = 10$ levels per octave in the image pyramid during object detection. For part kernels, features are extracted at twice the resolution of the root kernel level, i.e., $l_i = l_0 - \lambda$ for $i = 1, \cdots, n$.

Below we following the notations in Section 4.1.b. For the *i*-th kernel ($i = 0$ for root and $i = 1, ..., n$ for parts), let $\mathbf{p}_i = (\mathbf{x}_i, l_i) = (x_i, y_i, l_i)$ specify a position and pyramid level, $\phi(H, \mathbf{p}_i)$ denote the corresponding HOG features, $d\mathbf{x}_i = (dx_i, dy_i)$ denote the kernel

displacement from its anchor position, and $\phi_d(d\mathbf{x}_i) = (dx_i^2, dx_i, dy_i^2, dy_i)$. Given the DPM filter vector $\mathbf{F}_i$ and deformation coefficients $\mathbf{d}_i$, the HOG similarity for the *i*-th kernel is given by

$$s(\mathbf{p}_i) = \mathbf{F}_i^* \cdot \phi^*(H, \mathbf{p}_i) - (1-\delta_i)(\mathbf{d}_i^* \cdot \phi_d^*(d\mathbf{x}_i)), \tag{4.7}$$

where δ_i denotes the Kronecker delta function at $i = 0$, and $(\cdot)^*$ denote L^2-normalized vectors. Vector normalization is imposed to eliminate magnitude the difference between two terms. The mean-shift algorithm is applied to update the location of each kernel based on the HOG similarity $s(\mathbf{p}_i)$. The iteration continues until the maximum iteration T_{HOG} is reached.

4.2.d Kernel Aggregation

The object bounding box is determined by the root and part kernels after the kernel-based tracking and deformable HOG mean-shift algorithms. We integrate the root kernel and the part kernels to obtain the final position of the bounding box as follows. Let $\mathbf{c} \in \mathbf{R}^2$ denote the original bounding box center. Define the projected object center from the *i*-th part kernel by its current position $\mathbf{x}_i$ and anchor vector $\mathbf{v}_i$, i.e., $\mathbf{c}_i = (\mathbf{x}_i - \mathbf{v}_i)/2$. The new center position is then given by

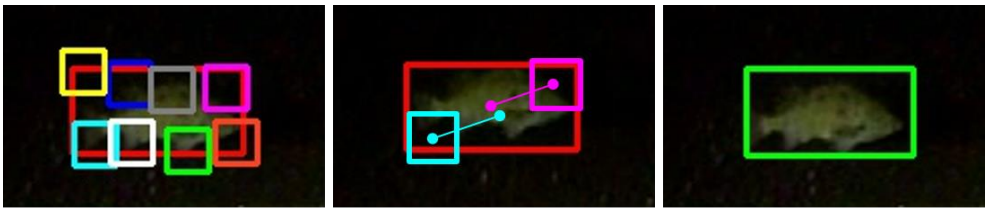
$$\mathbf{c}' = \alpha \mathbf{c}_{\text{root}} + (1-\alpha)\mathbf{c}_{\text{part}}, \tag{4.8}$$

where $\alpha \in [0,1]$, $\mathbf{c}_{\text{root}} = \mathbf{x}_0$ is the root kernel center, and

$$\mathbf{c}_{\text{part}} = \mathbf{c} + \frac{\sum_{i=1}^{n} s(\mathbf{p}_i)(\mathbf{c}_i - \mathbf{c})}{\sum_{i=1}^{n} |s(\mathbf{p}_i)|} \tag{4.9}$$

is a weighted average of projected centers. The weight $s(\mathbf{p}_i)$ is directly adopted from (4.7) as an indication of how well the candidate match the target. A higher weight value corresponds to higher confidence of this part kernel. The absolute value in the denominator ensures the capability of handling negative similarity values. In this way, the

final object position is robust against partial occlusions or low discrimination of the feature around the neighborhood. An example of computing the bounding box of the target by the kernel aggregation approach is illustrated in Figure 4.4.



(a) (b) (c)

Figure 4.4: Example of kernel aggregation: (a) the deformable multiple-kernel model after tracking; (b) the root kernel (red) and two part kernels (cyan and magenta), whose anchor vectors and projected object center given by (9) are illustrated; (c) final bounding box.

4.2.e Scale Adjustment

Targets captured by a moving camera are likely to undergo rapid change in scale throughout the tracking lifespan. The proposed algorithm updates the scale of targets by adopting the mechanism based on the derivative of density estimate f with respect to the kernel bandwidth h [18]. Here the target scale is represented by the size of the target bounding box. Let $s_k \in \mathbf{R}^2$ be the target scale at frame k, and β denote the update step size. The target scale at frame $k+1$ is then updated by

$$s_{k+1} = (1+\Delta s)s_k = \left(1+\beta\frac{\nabla f(h)}{f(h)}\right)s_k \tag{4.10}$$

4.3 Target Segmentation

Based on the DMK tracking technique presented in Section 4.2, one obtains reliable knowledge of the location and size for each target even when there is complex background or camera motion. Facilitated by this, a local approach based on the active contour algorithm is applied as follows.

For each tracked target, a "neighborhood region" is defined based the target bounding box. The upright rectangular bounding box is given by object detection and tracking in Section 4.2. Each bounding box is enlarged in order to cover the possible target body even when the target cropping occurs in the tracking step. The grayscale centroid is computed by using the grayscale intensity of all pixels within the neighborhood region. An initial contour, where the active contour algorithm starts, is then placed centered at this centroid location.

The Active Contours without Edges (ACWE) algorithm [14] is adopted to update the contour. Rather than searching for edges as the traditional snake algorithm [51] and the Geodesic Active Contours (GAC) [13] approach, ACWE updates the contour points by maximizing the intensity uniformity within and out of the contour, i.e., minimizing the energy function

$$\begin{aligned} F(c_1, c_2, C) = \mu \cdot \text{len}(C) + \nu \int_{\text{int}(C)} d\mathbf{x} \\ + \lambda_1 \int_{\text{int}(C)} \left\| I(\mathbf{x}) - c_1 \right\| d\mathbf{x} + \lambda_2 \int_{\text{ext}(C)} \left\| I(\mathbf{x}) - c_2 \right\| d\mathbf{x} \end{aligned} \tag{4.11}$$

where c_1 and c_2 are the expected pixel intensity for interior and exterior of the contour, respectively, and $\mu, \nu, \lambda_1, \lambda_2$ are scalar parameters that control the weight of each factor. After iterations the deformed contour produces a binary mask for the silhouette of the target. The histogram backprojection scheme introduced in Section 3.2 further refines the segmentation boundary. Rather than two object masks given by different thresholds, the histogram backprojection merges the mask produced by active contour, and an inflated version of the former one by morphological dilation as in [20]. Finally the post-processing described in Section 3.2 is applied.

4.4 Experimental Results

4.4.a Implementation Details

The public source code of DPM from the original authors [40] is used as the object detector. The number of parts in the DPM is determined automatically by the algorithm during the training stage. The mixture DPM is employed to enable capturing targets viewed in different perspectives throughout the videos. The number of perspective components in the mixture DPM is set as $M = 3$, which is determined empirically based on the trade-off between robustness against intra-class variations and simplicity of trained models.

The color histogram is built in the HSV color space to reduce the influence of brightness discrepancy between the target and candidates. The hue channel is divided equally into 15 bins. The saturation and value channels are divided equally into 8 bins respectively. For texture features, we follow the settings from [68] and generate a 10-bin histogram of the RIU-LBP for each kernel. The HOG features in each cell is represented by a 9-bin histogram. For mean-shift procedures, the maximum iterations are set as $T_{\text{color}} = 5$, $T_{\text{texture}} = 5$ and $T_{\text{HOG}} = 3$. The K-L distance is used as the similarity metric for color and texture histograms. In kernel aggregation, the root center weight is set as $\alpha = 0.5$ empirically (more discussion presented in Section 4.5). The proposed algorithm is not very dependent on the value of scale update step size β. A high value such as 10,000 should be enough for most cases. We set the criteria for reinitializing the DMK model as either the total HOG matching score of all kernels is lower than 0.2 or the scores are decreased for 3 consecutive frames. Finally, a linear Kalman filter is applied in every video frame to smooth the target trajectory.

For segmentation, the upright bounding box from tracking is enlarged by a factor of 1.7, which is determined empirically. An active contour is initialized to be centered at the centroid of grayscale values. The contour radius r is set according to the diagonal of the

enlarged bounding box, i.e., $r = \sqrt{w^2 + h^2}/15$. The Active Contour Without Edges (ACWE) technique is applied to the contour to achieve object segmentation. In practice, the fast morphological algorithm [63] for ACWE is adopted to accelerate the processing to an acceptable speed for practical applications. The parameters in ACWE are set respectively to $\mu = 1, \nu = 0$, $\lambda_1 = 1$ and $\lambda_2 = 4$. The maximum number of iterations is set to 70.

4.4.b Tracking Accuracy

The proposed method is evaluated by its tracking accuracy on several underwater mobile video datasets. The cameras were either mounted on a remotely operated underwater vehicle (ROV) or passively towed by a vessel. In these datasets targets are live fish, which are considered challenging cases due to the frequent variation in perspective and high body deformation during swimming. For each video, the target of interest is located manually by the bounding box in the first video frame. The performance is evaluated by the tracking error, i.e., the spatial distance measured in pixels between the center of target bounding box and the manually-labeled ground truth. The result videos associated with the simulations reported in this section can be watched online.[2]

The proposed method belongs to the kernel-based type in object tracking techniques. As a result, it is compared with the traditional mean-shift tracking (MS) [28] and constrained multiple-kernel tracking (CMK) method [18]. In CMK realization we represent a target by two kernels of identical size and are aligned horizontally. The spatial distance between two kernels is regarded as the constraint. To demonstrate the effectiveness of adopting texture features, we also report the results of the proposed method without taking into account the LBP histograms.

[2] URL: http://allison.ee.washington.edu/mengchec/dmkt

Table 4.1 reports the average tracking error of these techniques throughout the video frames on each of the tested datasets. The proposed DMK tracking method significantly outperforms others in all of the video datasets. By taking texture attributes into account, the proposed method successfully tracks the targets with little to no discriminative color distributions comparing to its vicinity. The deformation cost functions, which effectively retains the part configuration, also help determine the object position with better accuracy. On the contrary, the MS method uses only one kernel for the target, and hence suffers from drifting when occlusion occurs. The CMK method handles occlusion better than the MS method does, but still gives results with large error due to the inflexible spatial restrictions and dependency on only color histogram features.

Table 4.1: Average tracking error (pixels)

Dataset	MS [28]	CMK [18]	DMK-clr-hog	DMK-clr-tex-hog
ROVVTS_1	101.45	48.51	27.31	**8.61**
ROVVTS_2	57.80	39.16	22.45	**8.11**
AFSCUW_1	89.12	18.24	9.84	**7.29**
AFSCUW_2	27.52	56.18	6.91	**5.56**
HAUL1010_1	59.91	28.50	37.00	**7.38**
HAUL1010_2	200.37	76.67	24.14	**11.67**

Time plots of tracking error (i.e. error versus frame number) by using these techniques are also shown in Figure 4.5. From these time plots one can observe the robustness of the proposed method in extreme scenarios. For instance, in Figure 4.5(a) the target is partially occluded from frame #340 to #440, which can be seen in Figure 4.6(a). The proposed DMK method successfully recovers the target's trajectory, because the kernel aggregation scheme suppresses those part kernels with lower tracking confidence. Another major challenge to tracking live fish is the abrupt changes in velocity. As shown in Figure 4.6(c) and (d), the proposed method demonstrates its capability of tracking targets with sudden acceleration and sharp turns by adopting texture and HOG features, which are more reliable in the noisy underwater environment.

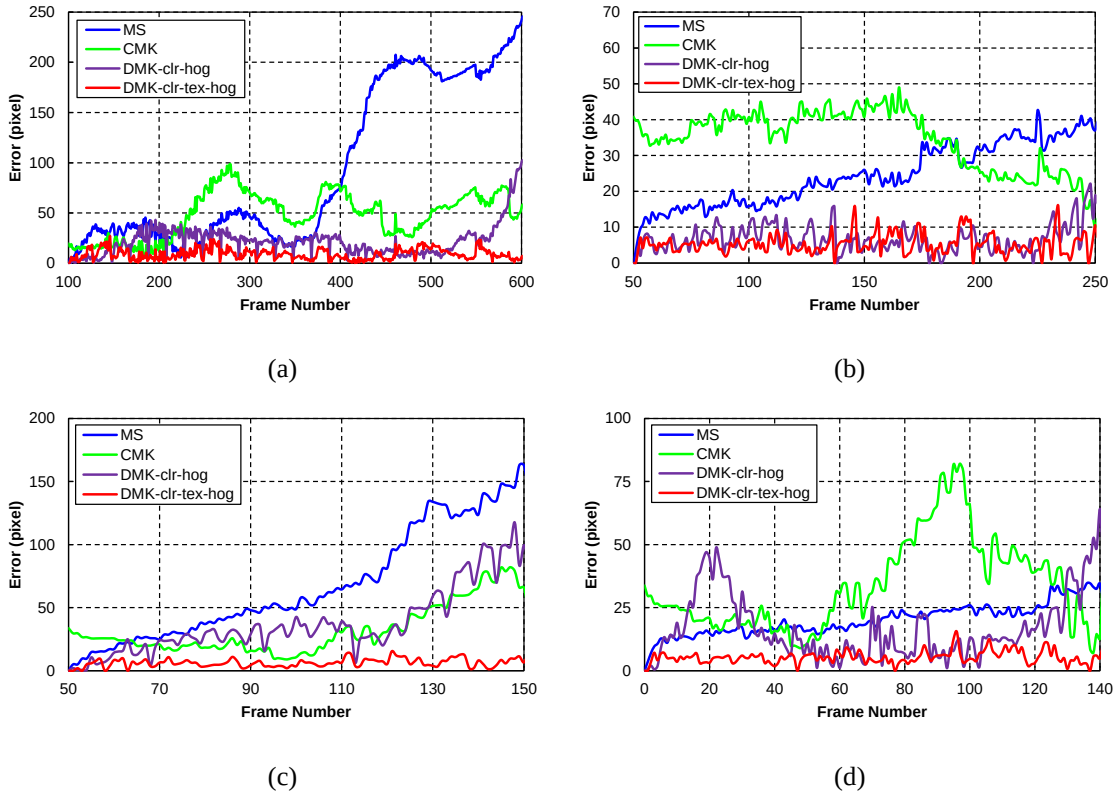
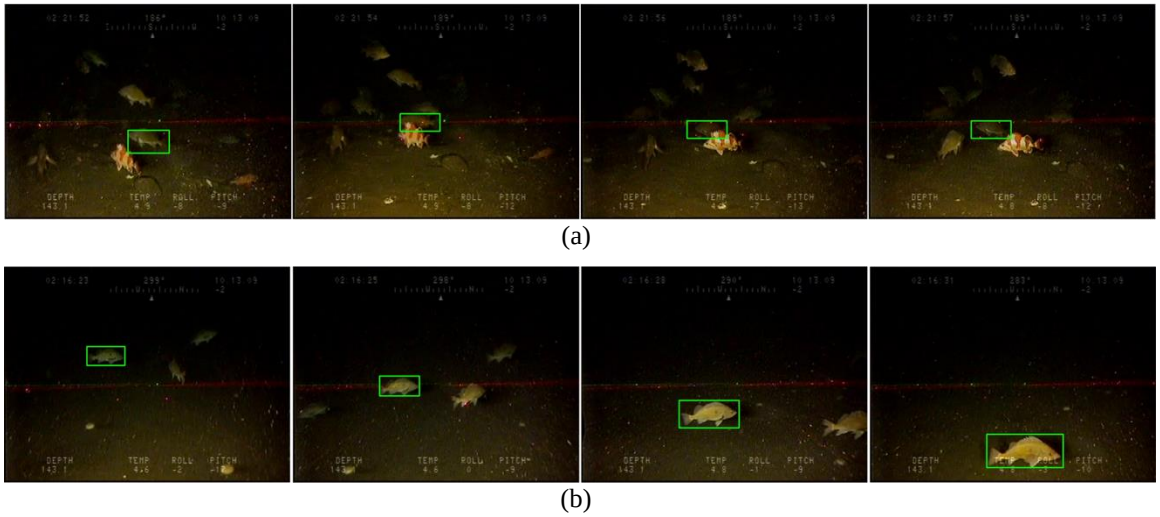


Figure 4.5: Errors of target center in pixels vs. frame number of tested datasets. (a) ROVVTS1_1; (b) AFSCUW_2; (c) HAUL1010_1; (d) HAUL1010_2.



(c)

(d)

Figure 4.6: Results of a specific target using the proposed DMK tracking algorithm in the tested datasets. (a) ROVVTS1_1; (b) AFSCUW_2; (c) HAUL1010_1; (d) HAUL1010_2. Partial occlusion of the target occurs in (a), while abrupt change in speed and sharp turns of target motion can be observed in (c) and (d).

4.4.c Multiple-Target Tracking and Segmentation

The proposed method is further tested for multiple-target tracking, i.e., all targets visible in the video are tracked simultaneously. Here the DPM object detection is performed to locate the targets, since background modeling techniques are not applicable to moving cameras. For each object, an independent DMK model is initialized and updated via the proposed method as what is applied in single-target tracking. The results of multiple-target tracking using the proposed DMK method are shown in Fig. 8. The proposed method successfully tracks all objects that are visible in the video despite the color ambiguity and occlusion against some of the objects. In addition to the MS and CMK methods from previous experiment, the proposed method is also compared with Berclaz et al. [5], which is the state-of-the-art of tracking-by-detection methods. The average tracking error for all targets are reported in Figure 4.7. The proposed method still outperforms the MS and CMK methods. However, the performance is slightly worse than the detection-based method in 4 out of 6 datasets. This is not surprising since detection-based methods use the object detection results directly and thus gives the most accurate results as long as object detection is reliable. The exception can be found in datasets

ROVVTS_1 and AFSCUW_1, where frequent partial occlusion leads to degraded accuracy in object detection. Some representative results of multiple-target tracking are shown in Figure 4.8 and target segmentation in Figure 4.9.

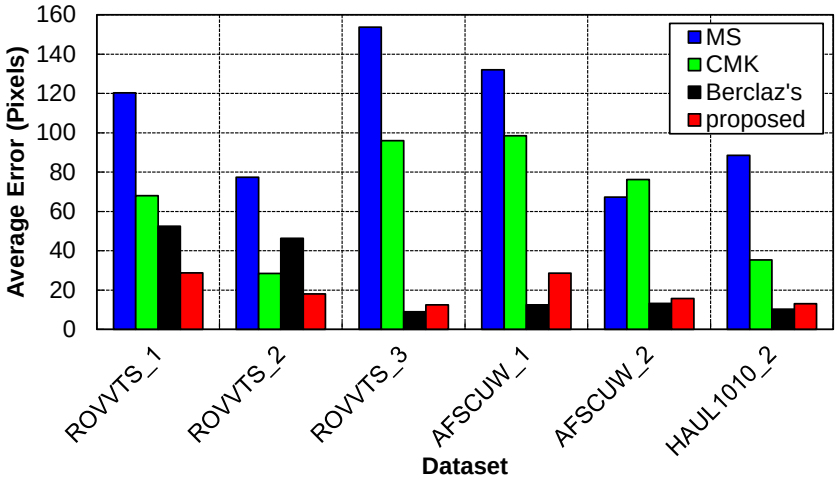
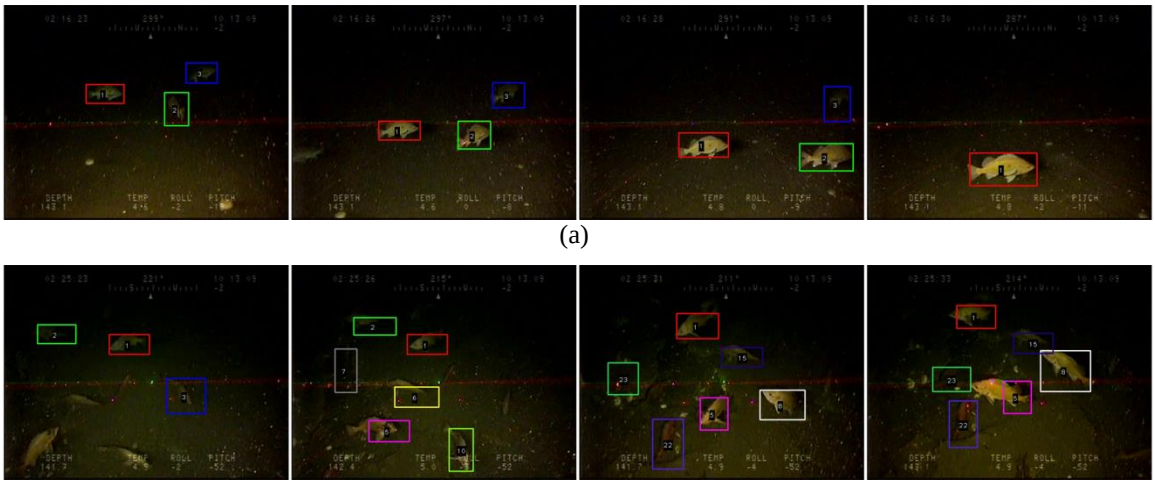


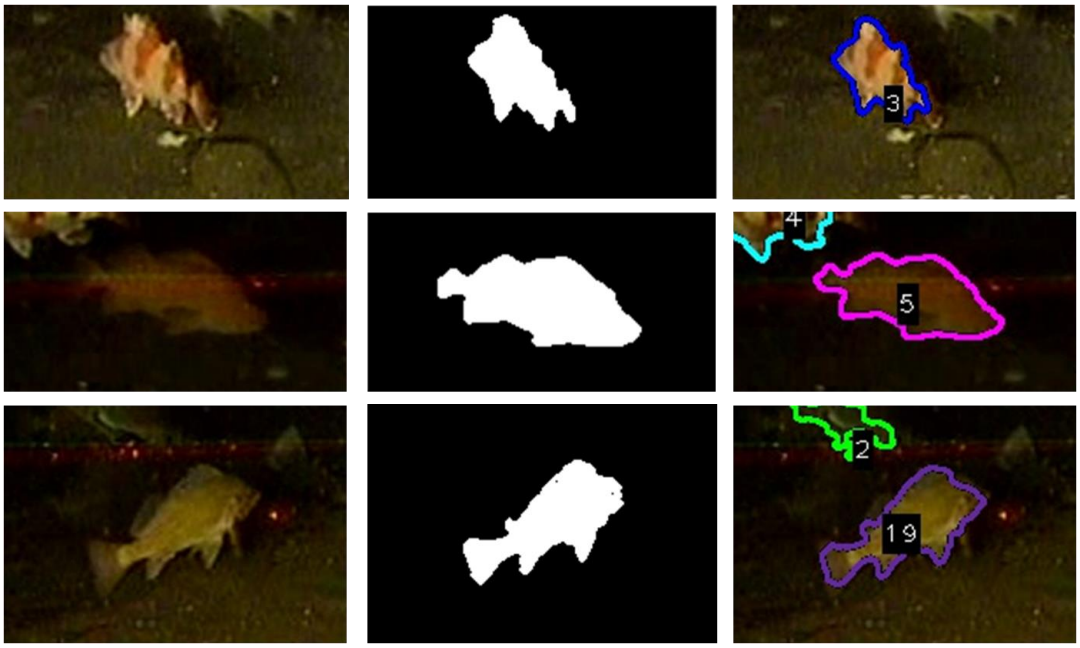
Figure 4.7: Average tracking error of multiple-target tracking in tested datasets.



(a)

(b)

Figure 4.8: Results of tracking multiple targets simultaneously. Each target is labeled by different colors and numbers. (a) ASFCUW_1; (b) ROVVTS_3.



(a) (b) (c)

Figure 4.9: Results of target segmentation. (a) Target images; (b) segmentation masks; (c) contours.

The advantage of the proposed method against the tracking-by-detection methods, however, is the computational requirement. This is due to the high efficiency of kernel-based tracking techniques. To verify this, we compare the CPU time consumption for updating the location of the target for one video frame. Table 4.2 reports the average computation time for tracking one target using several techniques, including the tracking-by-detection approaches as well as the kernel-based approaches. As shown in Table 4.2, the approach by Berclaz et al. takes the highest computation due to the multi-scale scanning by the object detector. The CMK uses the efficient kernel-based tracking algorithm and takes the least computation time. However, it leads to worse tracking accuracy in complex moving-camera cases as demonstrated in Section 4.4.b. The proposed DMK tracking technique, which combines the advantages from object detector and kernel-based tracking, achieves high performance in challenging moving camera scenarios while requires low computational cost by using the efficient mean-shift algorithm.

Table 4.2: Computational consumption

Methods	MS [28]	CMK [18]	Berclaz [5]	Proposed
Avg. time per frame (sec)	0.235	0.263	7.834	1.069

4.5 Discussion

One major factor to the tracking accuracy in the proposed DMK tracking algorithm is the weight during kernel aggregation introduced in (4.8). The effect of the kernel aggregation weight is balancing the influence on the target position by the root kernel and part kernels after a series of deformable part shifting. When the target is well visible in the field of view, part kernels are mostly well aligned with the expected position given by the anchors. In this case, either the root kernel position or the position projected by part kernels serves as a good estimate of the object position. If partial occlusion occurs, the root kernel position is likely off the real target position since there exists significant error when calculating the histogram similarity. Moreover, some of the part kernels are unable to reach the positions with high matching score since the corresponding parts are invisible or the positions are constrained by the deformation cost functions. As a result, performing kernel aggregation with a higher weight value leads to a target position contributed less by the part kernels and more robust against deformation. On the other hand, a lower weight gives a more accurate estimate of target position when the object is highly deformed, but very sensitive if any part is occluded.

To visualize the effect of the kernel aggregation weight, we tested the proposed algorithm with different weight values with the underwater video datasets. Single target tracking is performed with the target manually localized in the first video frame. Other simulation settings are the same as what is described in Section 4.4.b. The tracking performance with $\alpha = 0$ and $\alpha = 1$ are also reported as references to demonstrate the extreme cases, i.e., estimating the target location by only the part kernels or only the root kernel.

The results are illustrated in Figure 4.10. One can see that the tracking performance in two of the datasets are actually not very sensitive to the aggregation weight between the root and part kernels. The range of error in two datasets are within 4 pixels, which is only 5% of the diagonal length for a typical target bounding box. This implies that the root kernel center and the "expected" center by part kernels are very close to each other as well as the ground truth. In dataset ROVVTS_1, a serious partial occlusion of the target exists for approximately 100 frames. By the proposed kernel aggregation scheme, the HOG matching score is taken into account for each part kernel. The matching scores from occluded parts are much lower than the visible ones. As a result, the estimate of target center is robust against partial occlusion by suppressing the parts with lower confidence in matching. Dataset HAUL1010_2 presents another challenging scenario, namely tracking a target with abrupt motion. High deformation is often observed when the target is making sharp turns or change of speed. In this case the part kernels may not give a consistent estimate of the target center. This explains the lower performance achieved by using a small value for α, where the part kernel locations dominate the final target center in (4.8).

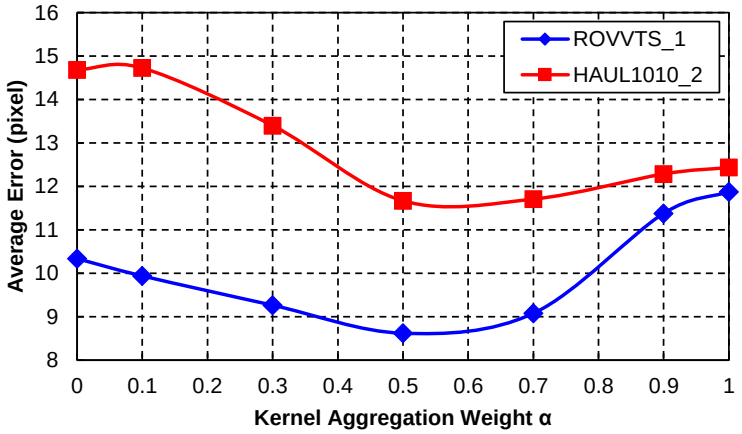


Figure 4.10: Average tracking error by the proposed method vs. kernel aggregation weight.

Chapter 5 – Stereo Body Length Estimation

5.1 Overview

In addition to tracking and counting, additional beneficial information for the fish abundance survey comes from knowing the length estimate of each fish in the video. By capturing fish video with the Cam-trawl, live fish length measurement can be done by employing stereo imaging techniques. Stereo matching guided by the object-height blocks, as described in Section 3.3.a, matches fish targets between the left and the right video frames and generates a coarse disparity map. To enhance the accuracy of fish length measurement, a fish-tail end compensation step is further applied to each pair of stereo-corresponding targets. Disparity refinement followed by stereo triangulation is performed to estimate the 3-D length of fish body.

5.2 Length Estimation in 3-D Space

5.2.a Fish-Tail End Compensation

A common failure case in the fish segmentation algorithm described in Section 3.2 is fish cropping caused by the lower reflectivity of the caudal fin (tail). This introduces considerable error in fish length measurement. To overcome this issue, a fish-tail end compensation technique using the result of stereo matching is performed.

To obtain a region for the possible presence of the fish tail, we utilize the oriented bounding box of the fish introduced in Section 3.1. The end region of a fish is defined as a square with size $h_0 \times h_0$, where h_0 is the height of the oriented bounding box. An example of the end square region is illustrated in Figure 5.1. For a pair of stereo-matched objects, the two ends of the objects are validated by computing the sum of absolute difference (SAD) of two end regions between the left and the right video frames. A

mismatch is detected via thresholding the ratio of the computed SAD value to the average minimum SAD of the same target. More specifically,

$$mismatch = \begin{cases} 1 & \text{if } SAD/\mu_{SAD} > \theta_{SAD} \\ 0 & \text{if } SAD/\mu_{SAD} \le \theta_{SAD} \end{cases}, \tag{5.1}$$

where μ_{SAD} is the average of the minimum SAD values of the four object-height blocks associated with the same target, and θ_{SAD} is an empirically defined threshold value.

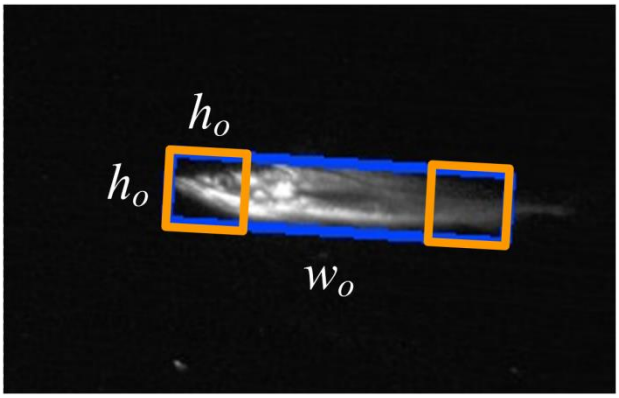


Figure 5.1: End square regions (orange) of a fish. In the figure w_O and h_O denote respectively the width and height of the oriented bounding box (blue).

End compensation method is applied once a mismatch is detected. The classic snake (deformable contour) algorithm [51] is used to extract the missing tail. We use a circle with a diameter of $h_0/2$ as the initial snake and place it next to the mismatch end along the principal component of the target. After the iterations, the extracted region is connected with the body by using a morphological closing operation. The effect of the fish-tail end compensation procedure is demonstrated in Figure 5.2.

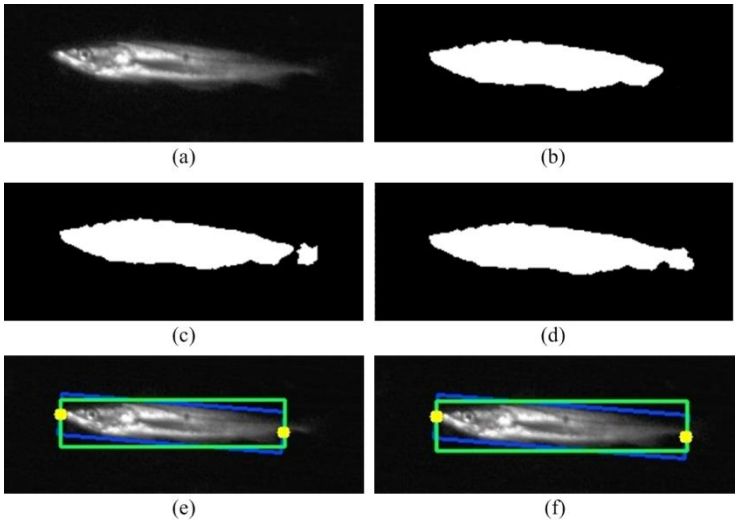


Figure 5.2: Illustrative example of fish-tail end compensation: (a) stereo-rectified image; (b) object mask before end compensation; (c) object mask from (b) and result of snake algorithm; (d) new object mask after morphological closing operation; (e) upright bounding box (green), oriented bounding box (blue) and end points (yellow) before end compensation; (f) upright bounding box, oriented bounding box and end points after end compensation, where the less-reflective part around tail fin is recovered.

5.2.b Body Length Estimates

Measuring the target length requires more accurate information of depth. A disparity refinement based on coarse disparity map is therefore applied. As in Section 3.3.a, a block-matching approach with a SAD criterion is used. To measure the fine disparity, we use dense grid blocks for matching and set the search range as $[d_o - r, d_o + r]$, where d_o is the coarse disparity at that position as derived from stereo matching based on object-height blocks, and r denotes the search range size.

The technique of stereo vision leads the way to absolute length measurement of targets by locating the targets back in the 3-D space. Intrinsic parameters of the cameras have been obtained in prior by stereo calibration. Given a point on the 2-D image represented by its screen coordinates (x, y) and disparity d, it can be projected into 3-D space by using stereo triangulation [10]. That is, a point in the 2-D space is reprojected

into the 3-D space given its image coordinates (x, y) and associated disparity d through the linear transformation

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \mathbf{Q} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{c_x - c'_x}{T_x} \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix}, \tag{5.2}$$

where (c_x, c_y) and f denote respectively the principal point and the focal length of the left camera, c'_x denotes the principal point x-coordinate of the right camera and T_x denotes the x-coordinate of between two cameras. These values can be easily estimated, if not provided by the camera manufacturer, through standard stereo camera calibration in the form of intrinsic and extrinsic parameters. Based on the homogenous coordinates, the reprojected 3-D location of the point is given by $(X/W, Y/W, Z/W)$. The length of the *i*-th fish is thus given by $L(O_i) = \left\| \mathbf{x}_i^H - \mathbf{x}_i^T \right\|_2$, i.e., the Euclidean distance between the head point and tail point of the fish body in the 3-D space.

5.3 Experimental Results

The accuracy of fish body length measurement by 3-D estimation is evaluated and compared with fish segmentation results described in Section 3.4. The testing set consists of 7,120 video frames from a single haul of fish. According to manually-measured ground truth, there are 316 targets in total captured in this haul by the Cam-trawl system. For disparity refinement, we use a dense block grid with size 8×8 pixels. The threshold of SAD for body end mismatch is set to $\theta_{SAD} = 16$, while the search range in disparity refinement is set to $r = 16$.

Thanks to the stereo matching and fish-tail end compensation, we are able to achieve 6.0% of mean absolute percentage error (MAPE) in fish length measurement, which greatly improves the previous measurements done with a single camera as in Table 3.3.

To further demonstrate the performance of the proposed algorithm, the length distribution of targets is also generated from the proposed algorithm and compared with two sources of ground truth. One is the physical measurements of the fish onboard, i.e., manual measurement of each individual fish with a ruler. The other is the measurements using a software program requiring manual selection of head and tail points for stereo-matched fish targets in the left and the right video frames.

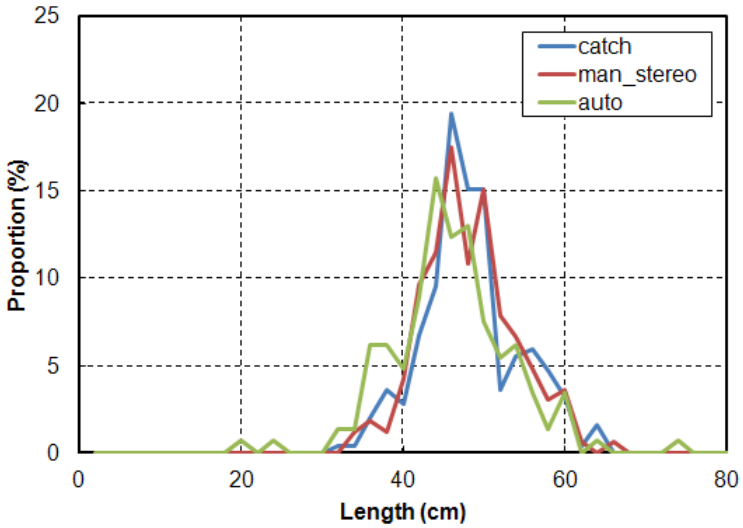


Figure 5.3: Length frequency of fish targets.

The results in terms of length frequency distribution are shown Figure 5.3. There are 316 fish targets for physical measurements (catch) and 208 fish targets according to measurements using manual selection of end points (man_stereo) and using the proposed stereo length measurement algorithm (auto). The decrease in target number from manual count to image-based count is due to the low frame rate of video capturing, so that a number of targets never appear completely inside the field of view in any frame. One can observe that the proposed algorithm generates a good match with the two sources of ground truth in terms of the distribution of fish length frequency.

5.4 Discussion

As shown in Figure 5.3, the proposed algorithm gives slightly underestimated measurements. There are two main reasons of this phenomenon. One is the systematic difference in fish length measures. In physical procedures we measure the *fork length*, which refers to the length measured from the snout tip to the end of the middle caudal fin rays. On the other hand, the caudal fin is often poorly illuminated and may not be fully recovered by the fish-tail compensation scheme. The automatic image-based procedure using the rotated bounding box results in the *standard length*, which totally excludes the caudal fin, and therefore appears shorter. The other reason for the negative bias is that many fish in the video present a certain extent of body bending, which makes them appear shorter.

One way to compensate for this defect, especially for the systematic difference in measures, is a rescaling or regression technique on the data. A more precise approach, which will be one objective of future work, is to establish a 3-D model for the fish body from the stereo cameras. By calculating the arc length of model midline, the length of the fish body can be estimated with higher accuracy.

Chapter 6 – Species Recognition

In this chapter, a species recognition framework is proposed for live fish images captured in the unrestricted natural environment.

6.1 Overview

The proposed fish species recognition framework consists of feature extraction and classification [19, 21, 22]. There are two approaches to fish species feature extraction presented in this chapter, i.e., the supervised method and unsupervised method. The supervised method extracts part-aware features that are predefined based on the domain knowledge in fisheries and biology. These features reflect some specific local details as well as the holistic appearance of the fish body such as the tail shape, eye size, aspect ratio, etc. The advantage of the supervised features includes model simplicity and consistency with fisheries knowledge. However, it does not guarantee the optimality of selected features and is unable to handle the huge diversity of fish species. On the other hand, the main advantage of the proposed feature learning algorithm is that it uses fully unsupervised algorithms to learn the features and class correlation, and thus provides an automatic solution for practical recognition systems without any requirement for human interference. A systematic comparison has also shown that the unsupervised approaches in general lead to better recognition performance [21].

As for classification, the proposed hierarchical partial classifier is able to handle uncertain or missing data, which is a common and challenging issue in practical applications of object recognition. For example, capturing images for freely-swimming fish in an open aquatic habitat usually introduces a high uncertainty in many of the data due to poor image quality and non-lateral fish. Experimental results and analyses show that the proposed system achieves a favorable recognition accuracy on underwater fish images with high uncertainty and class imbalance.

6.2 Part-Aware Fish Feature Descriptors

The major visual differences among fish species often lie in some external anatomical parts, such as the head, the caudal (tail) fin and eyes location and shapes. These body parts are therefore emphasized by the proposed part-aware feature extraction to identify the fish species [19].

6.2.a Head/Tail Localization

To determine the head/tail side of a fish body, we utilize the concept of 1-D image projection. For the *i*-th fish binary mask $B_i(u,v)$, with image size $W_i \times H_i$, the vertical projection is given by

$$p_i^{B,vert}(u) = \left(\sum_{v=1}^{H_i} B_i(u,v) \right) * G_\sigma(u), \ \forall u = 1,...,W_i, \tag{6.1}$$

where $G_\sigma(u)$ denotes the 1-D Gaussian filter with standard deviation σ. Using the vertical projection, positions of tail and head are estimated by

$$u_i^t = \arg\min_{u \in \Omega_i^t} p_i^{B,vert}(u), \tag{6.2}$$

$$u_i^h = \arg\min_{u \in \Omega_i^h} \frac{d}{du} p_i^{B,vert}(u), \tag{6.3}$$

where Ω_i^t and Ω_i^h are the tail and head search ranges, respectively. Formulations in (6.2) and (6.3) are given by knowledge of fish anatomy that in general the boundary between tail and body is thinnest in width, while the boundary between head and body is where the body width tends to stop increasing. Examples of head and tail localization are shown in Figure 6.1.

In addition to head and tail, the eye is also an important key to distinguishing fish species. An object detector based on the Viola-Jones cascade classifier using Haar-like features [83] is trained and employed inside the head region to locate the eyes. The object detector is trained by a hand-labeled image set consisting of 1240 eye samples and 1680 non-eye samples at resolution of 8×8 pixels. A 15-stage cascade classifier is learned over these samples and used to locate the eye regions in test images.

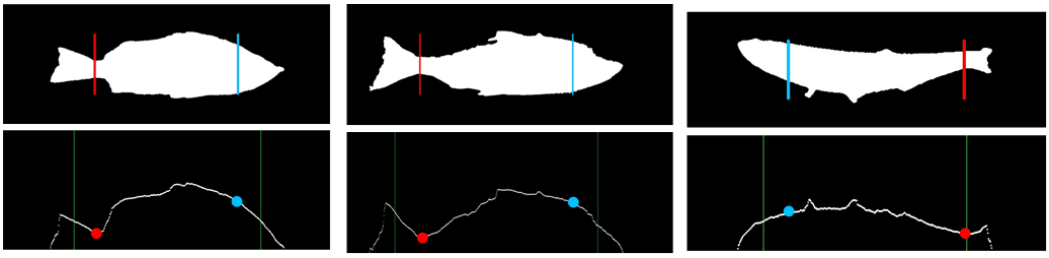


Figure 6.1: Fish binary masks (top row) and their vertical projections (bottom row). The boundary positions for the tail and head region are labeled separately by red and turquoise lines and dots.

6.2.b Feature Descriptors:

The size, shape and texture attributes of fish body parts mentioned above are extracted as species features. The size attributes are estimated by the part area calculated by the connected components algorithm and normalized by the whole body area. Shape is represented by the Fourier descriptor of contour points normalized by its DC component so that the descriptor is scale-invariant. Texture properties are represented by the histogram of local binary patterns (LBP) within the detected body parts [68]. Moreover, by leveraging fisheries knowledge, several global attributes of the fish body are also added to the features. These includes the aspect ratio, the projection ratio and the length ratio of fish body. The species features, with a total dimensionality of 75, are summarized in Table 6.1.

Table 6.1: List of descriptors in part-aware fish features

Feature	Dim.	Description
Aspect ratio	1	Body length / body height
Length ratio	1	Length from mouth to tail / total length
Projection ratio	1	$\max_u p_i^{B,vert}(u) / \min_u p_i^{B,vert}(u)$
Tail size	1	Tail area divided by body area
Tail shape	19	Fourier descriptor of tail contour
Tail texture	16	LBP histogram of tail
Head size	1	Head area divided by body area
Head shape	19	Fourier descriptor of head contour
Eye texture	16	LBP histogram of eye

6.3 Non-Rigid Part Model

An overview of the proposed unsupervised non-rigid part learning algorithm is shown in Figure 6.2. In the training stage, object parts are initialized and associated across training image. The non-rigid part model is then learned from images via an unsupervised algorithm. In the testing stage, the model locates informative object parts in each testing image. The location, size and appearance of each part are extracted as features. Finally, these features are used to train a fine-grained object classifier.

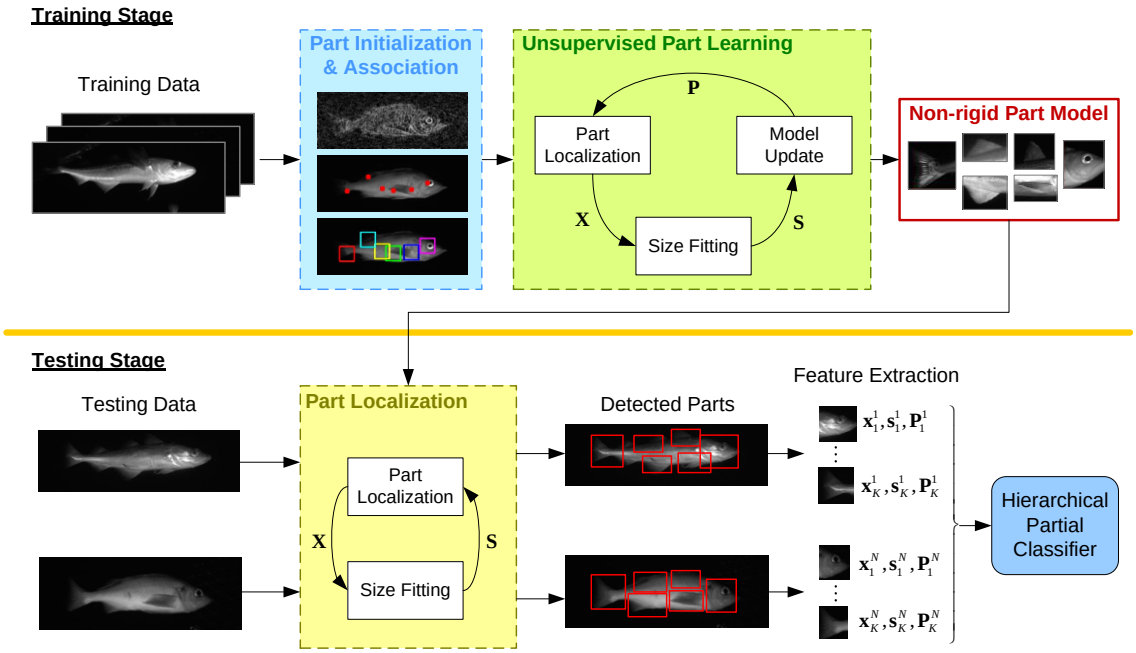


Figure 6.2: Overview of the proposed unsupervised non-rigid part learning algorithm.

Given a set of training images $\mathbf{I}=\{I^1,...,I^N\}$, the goal is to discover discriminative features for the objects in terms of their subordinate categories. Let $\mathcal{M}=\{\mathbf{P},\mathbf{X},\mathbf{S}\}$ denote a model that consists of part appearance $\mathbf{P}=\{\mathbf{P}_1,...,\mathbf{P}_K\}$, part center locations $\mathbf{X}=\{\mathbf{X}^1,...,\mathbf{X}^N\}$ and part sizes $\mathbf{S}=\{\mathbf{S}^1,...,\mathbf{S}^N\}$, where K is the number of object parts. For each image I^m we denote $\mathbf{X}^m=\{\mathbf{x}_1^m,...,\mathbf{x}_K^m\}$ and $\mathbf{S}^m=\{\mathbf{s}_1^m,...,\mathbf{s}_K^m\}$ accordingly. The location and size of each part is normalized with respect to the image size, i.e., $x_i, s_i \in [0,1]\times[0,1]$. The model $\mathcal{M}$ is referred to as a *non-rigid part model* since it

describes common parts and allows for deformation in both position and scale. Based on this, the problem of learning such a model can be written as a constrained minimization programming problem:

$$(\mathbf{P}^*, \mathbf{X}^*, \mathbf{S}^*) = \arg\min J(\mathbf{P}, \mathbf{X}, \mathbf{S}), \tag{6.4}$$

$$\text{s.t. } 0 \le \mathbf{s}_i^m \le 1, \quad i = 1, ..., K, \quad m = 1, ..., N \tag{6.5}$$

$$0 \le \mathbf{x}_i^m \pm (1/2)\mathbf{s}_i^m \le 1, \quad i = 1, ..., K, \quad m = 1, ..., N \tag{6.6}$$

where $J(\mathbf{P}, \mathbf{X}, \mathbf{S})$ is the objective function of the model. Note that (6.5) and (6.6) denote entry-wise inequalities for $\mathbf{x}_i^m$ and $\mathbf{s}_i^m$.

To model both the appearance and geometry of object parts, the objective function $J(\mathbf{P}, \mathbf{X}, \mathbf{S})$ takes three factors into account: 1) *fitness*, which computes the appearance similarity between the model and an image region; 2) *separation*, which guides the detected parts to match as disjoint as possible regions instead of concentrating on a few regions of the image; 3) *discrimination*, which encourages the selected parts to have distinct appearance from each other in order to capture as many aspects of the object appearance.

6.3.a Fitness

Image regions corresponding to a certain object part have high appearance similarity with the model. The fitness cost is thus calculated by the distance between the part appearance $\mathbf{P}_i$ and the appearance of a rectangular region in image I^m defined by the center location $\mathbf{x}_i^m$ and size $\mathbf{s}_i^m$. The fitness cost associated with this region is given by

$$J_{fitness} = \sum_{m=1}^{N} \sum_{i=1}^{K} d(\mathbf{P}_i, \phi(I(\mathbf{x}_i^m, \mathbf{s}_i^m))), \tag{6.7}$$

where $\phi(\cdot)$ denotes the feature descriptor for an image region, and $d(\mathbf{P}, \mathbf{Q})$ is the distance between two appearance feature vectors $\mathbf{P}$ and $\mathbf{Q}$. For the following part of this paper, we denote $I(\mathbf{x}_i^m, \mathbf{s}_i^m)$ by I_i^m for convenience.

6.3.b Separation

The separation cost enforces the parts to cover the maximum area of the whole object. This is achieved by minimizing the total overlapping rate of the image regions defined by the location-size tuples $(\mathbf{x}_i^m, \mathbf{s}_i^m)$ and $(\mathbf{x}_j^m, \mathbf{s}_j^m)$ for $i \neq j$:

$$J_{separation} = \sum_{m=1}^{N} \sum_{i=1}^{K} \sum_{j \neq i} v(I_i^m, I_j^m), \tag{6.8}$$

The overlapping rate is defined as the area ratio between the intersection and union of two rectangles, i.e.,

$$v(I_i^m, I_j^m) = \frac{\left| I_i^m \cap I_j^m \right|}{\left| I_i^m \cup I_j^m \right|}. \tag{6.9}$$

We denote $v(I_i^m, I_j^m)$ by $v_{i,j}^m$ for the following part of this paper for convenience.

6.3.c Discrimination

It is desired that the non-rigid part model covers every representative part of the object. As a result, the discrimination cost is introduced to encourage the maximization of the distance between each pair of part features $\mathbf{P}_i$ and $\mathbf{P}_j$, i.e.,

$$J_{discrimination} = -\sum_{i=1}^{K} \sum_{j=1}^{K} d(\mathbf{P}_i, \mathbf{P}_j), \tag{6.10}$$

where d is the same distance metric as the one in (6.7).

6.3.d Objective Function

Having the above cost functions, the final objective function is given by

$$J(\mathbf{P}, \mathbf{X}, \mathbf{S}) = J_{fitness} + J_{separation} + J_{discrimination}, \tag{6.11}$$

The non-rigid part model, i.e., part features, locations and sizes are trained by minimizing (6.11) over the given training set $\mathbf{I}$ using the proposed unsupervised learning algorithm described in the following section.

6.4 Unsupervised Part Model Discovery

Now we have a non-rigid part model – features, locations and size of each part – that represents the object local appearance and configuration, as well as an objective function (6.11) to be minimized to find the model. An EM-like algorithm, i.e., alternating optimization, enables an unsupervised approach to learning such a model from the training images. With a systematic initialization technique, no human annotation for parts is required for learning the final feature descriptors.

6.4.a Part Initialization

The effectiveness of alternating optimization guarantees only the convergence to local optima. To ensure a good solution can be obtained, we propose a systematic approach to initialize the part model. Note that most details that distinguish fine-grained categories match those parts which are prominent to humans' perception, such as beak of a bird, pedal of a flower, or tail fin of a fish. Saliency operators works perfectly for this purpose.

There have been a variety of techniques investigated for estimating image saliency. For the efficiency in dealing with vast data amount, we adopt the phase Fourier transform (PFT) approach described in [70]. Given an image, we calculate its 2-D discrete Fourier transform, which can be expressed by the magnitude term $M(u,v)$ and phase term $\Phi(u,v)$, i.e.,

$$F(I(x,y)) = M(u,v)e^{j\Phi(u,v)}, \tag{6.12}$$

The saliency is obtained by taking the inverse Fourier transform of only the phase term:

$$s(x,y) = G_\sigma(x,y) * F^{-1}(e^{j\Phi(u,v)}), \tag{6.13}$$

where $G_\sigma(x,y)$ is a 2-D Gaussian filter with a standard deviation σ. Non-maximal suppression is applied to extract local maxima from the saliency map. Here we use the object segmentation mask to discard salient points in the background. Note that using the given segmentation does not make our learning method supervised, since the masks can be easily generated by existing techniques such as the GrabCut segmentation [75].

Finally, we choose the top K local maxima locations, each of which serves as an initial part. Examples of part initialization based on PFT saliency are shown in Figure 6.3.

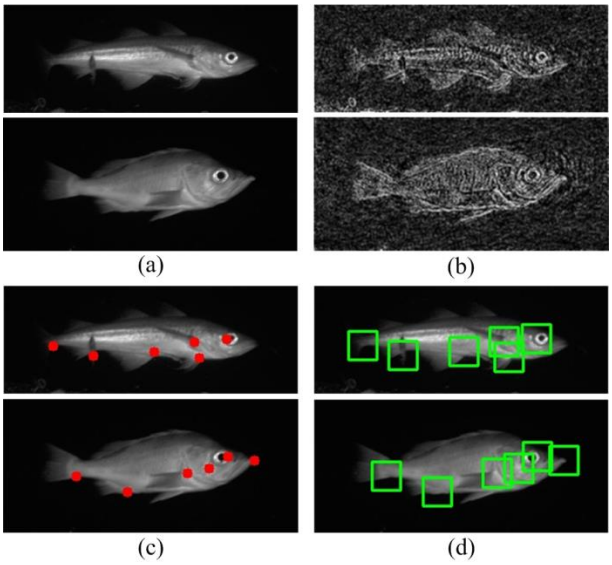


Figure 6.3: Part initialization based on PFT saliency. (a) Input images; (b) PFT results of (a); (c) points with top saliency values inside the segmentation masks; (d) initial parts.

6.4.b Labeling-based Part Association

Due to the pose variation, one object part may appear in different locations in two images. To ensure the correctness of learning, it is important to align the extracted points from one image to another. In the proposed method, we formulate part identification as a one-to-one association problem and apply the relaxation labeling process as follows.

Suppose a reference set contains K part locations, denoted by $\mathbf{Y}=\{\mathbf{y}_1,...,\mathbf{y}_K\}$, and a candidate set also contains K part locations $\mathbf{X}=\{\mathbf{x}_1,...,\mathbf{x}_K\}$. The goal of part identification is to find an optimal association from candidate parts to reference parts, which is similar to the matching problem between two sets of 2-D points that undergo some non-rigid deformation [57, 97]. The association can be expressed by a $(K+1)\times(K+1)$ binary matrix $\mathbf{\Pi}$, where $\pi_{ij}=1$ when $\mathbf{x}_i$ is associated with $\mathbf{y}_j$ and $\pi_{ij}=0$ otherwise. Each row of $\mathbf{\Pi}$ corresponds to a candidate part, and each column corresponds to a reference part. The augmented row and column denote the "outliers,"

i.e., no match for a certain candidate or reference part. Introducing the outlier notion brings in two advantages. Firstly, it allows for handling substantial pose variations or partial occlusions. Moreover, it facilitates the imposition of one-to-one match constraint between two part location sets.

In relaxation labeling, the binary constraint $\pi_{ij} \in \{0,1\}$ is relaxed to $\pi_{ij} \in [0,1]$. It has been proved that π_{ij} approaches to either 0 or 1 upon convergence [57]. During the iterations, each entry π_{ij} is updated by exploiting the contextual information, which is represented by the compatibility coefficient $r(i, j, k, l) \in [0,1]$. A high value of $r(i, j, k, l)$ corresponds to high association likelihood between $(\mathbf{x}_i, \mathbf{y}_j)$ and $(\mathbf{x}_k, \mathbf{y}_l)$. When the part configurations are non-rigid, the compatibility coefficient considers only local neighbors of both $\mathbf{x}_i$ and $\mathbf{y}_j$ [57, 97]. Here two parts $\mathbf{x}_i$ and $\mathbf{x}_k$ are defined neighbors of each other only if $\mathbf{x}_i$ is one of the *k*-nearest parts of $\mathbf{x}_k$ and vice versa. In the experiments we set $k = 3$, which gives promising results. For each part $\mathbf{x}_i$ the indices of its neighbors are denoted by a set N_i .

To this end, a novel compatibility coefficient that imposes pairwise geometric constraints between two neighboring parts is defined as follows. To handle pose variation among part sets, we first project all part coordinates to the basis formed by the object's two principal components. Denote the projected coordinates of candidate and reference parts by $\mathbf{u}_i$ and $\mathbf{v}_j$, respectively. The associations $(\mathbf{u}_i, \mathbf{v}_j)$ and $(\mathbf{u}_k, \mathbf{v}_l)$ are more compatible if their distance between $\mathbf{u}_i$ and $\mathbf{u}_k$ is similar to the distance between $\mathbf{v}_j$ and $\mathbf{v}_l$. Also, if the vector direction from $\mathbf{u}_i$ to $\mathbf{u}_k$ is similar to the one from $\mathbf{v}_j$ to $\mathbf{v}_l$, they are more compatible. The distance and angle disparity is thus given by

$$a(i, j, k, l) = (\|\mathbf{u}_{i,1} - \mathbf{u}_{k,1}\| - \|\mathbf{v}_{j,1} - \mathbf{v}_{l,1}\|)^2, \tag{6.14}$$

$$b(i, j, k, l) = (\cos\theta_{ik} - \cos\theta_{jl})^2. \tag{6.15}$$

Note here only the first principal component coordinates are considered to handle the horizontal flipping of images. The final compatibility coefficient between $(\mathbf{u}_i, \mathbf{v}_j)$ and $(\mathbf{u}_k, \mathbf{v}_l)$ is defined as $r(i,j,k,l) = \exp(-(a(i,j,k,l) + b(i,j,k,l)))$. Based on this, the support function q_{ij} in each iteration is given by

$$q_{ij} = \sum_{l \in N_j} \sum_{k \in N_i} \pi_{kl} r(i,j,k,l) . \tag{6.16}$$

Each entry π_{ij} is then updated by

$$\pi_{ij} \leftarrow \pi_{ij} q_{ij} \Big/ \sum_{k=1}^{K} \pi_{ik} q_{ik} . \tag{6.17}$$

Alternated row-column normalization is performed after each relaxation update. It has been shown that this normalization process ensures that each row and column of $\mathbf{\Pi}$ sum to one according to Sinkhorn's theorem [57]. The final association for each target part is chosen upon convergence by the maximum π_{ij} with respect to j.

6.4.c Unsupervised Part Model Discovery

The non-rigid object part model is learned by solving the optimization problem (6.4)–(6.6), where the objective function $J(\mathbf{P}, \mathbf{X}, \mathbf{S})$ is given by (6.11). In order to effectively update the variables $\mathbf{P}, \mathbf{X}, \mathbf{S}$ without human assistance in the loop, we adopt the alternating optimization that is similar to the expectation-maximization (EM) algorithm or the coordinate descent approach. In the proposed unsupervised part discovery algorithm, each iteration consists of three steps. The algorithm first updates the locations $\mathbf{X}$ (part localization) with the remaining variables fixed, then updates the sizes $\mathbf{S}$ (part size fitting) with the remaining variables fixed, and finally updates the part features $\mathbf{P}$ (part model learning) with the remaining variables fixed. For each training image the part locations and sizes are initialized based on the saliency detection and relaxation labeling procedure described in Section 6.4.a and 6.4.b. The appearance for each part is initialized by the average value of the corresponding block over the training set. The learning procedure of non-rigid part model is summarized in Algorithm 6.1.

Algorithm 6.1: Non-rigid Part Model Learning

1. **input:** images $\mathbf{I} = \{I_1, ..., I_N\}$, maximum iteration *maxIter*
2. **output:** part features $\mathbf{P}$, locations $\mathbf{X}$, sizes $\mathbf{S}$
3. initialize $\mathbf{P}, \mathbf{X}, \mathbf{S}$ by the PFT saliency
4. associate parts with reference by relaxation labeling
5. **for** $t = 1$ to *maxIter* **do**
6. update $\mathbf{X}$ for each image based on (6.18)–(6.19)
7. update $\mathbf{S}$ for each image based on (6.22)–(6.24)
8. update $\mathbf{P}$ based on (6.26)
9. **if** *converged* **then**
10. break
11. **end if**
12. **end for**

1) Part Localization

In this step, the part features $\mathbf{P}$ and sizes $\mathbf{S}$ are given. By updating $\mathbf{X}$, we localize the sub-region that corresponds to each part in each image. The discrimination cost term in (6.10) becomes a constant since $\mathbf{P}$ is fixed for now. Hence the optimization problem (6.4)–(6.6) becomes

$$\min_{\mathbf{X}} \sum_{m=1}^{N} \left(\sum_{i=1}^{K} d(\mathbf{P}_i, \phi(I_i^m)) + \sum_{i=1}^{K} \sum_{j \neq i} v_{i,j}^m \right) \tag{6.18}$$

$$\text{s.t.} \quad 0 \leq \mathbf{x}_i \pm (1/2)\mathbf{s}_i \leq 1, \quad i = 1, ..., K. \tag{6.19}$$

Here the mean-shift algorithm [27] is adopted to solve for $\mathbf{X}$ as follows. In the *m*-th image, we start from the initial location $\mathbf{x}_i^m(0)$ for the *i*-th object part. The typical mean-shift algorithm is a gradient ascent optimization procedure, so the negative of objective function (6.18) is maximized instead. The gradient estimate is iteratively calculated by using pixels within the located sub-region. Given the gradient estimate, the part location is updated by

$$\mathbf{x}_i^m(t+1) = \frac{\sum_{j=1}^{n_p} w_j k(\mathbf{z}_j - \mathbf{x}_i^m(t))(\mathbf{z}_j - \mathbf{x}_i^m(t))}{\sum_{j=1}^{n_p} w_j k(\mathbf{z}_j - \mathbf{x}_i^m(t))}, \tag{6.20}$$

where $k(\cdot)$ is the kernel function, n_p is the number of pixels in the part and w_j is the sample weight at pixel $\mathbf{z}_j$:

$$w_j = -\frac{\partial}{\partial \mathbf{x}_i^m}[d(\mathbf{P}_i, \phi(I_i^m)) + \sum_{j \neq i} v_{i,j}^m]\Big|_{\mathbf{x}_i^m = \mathbf{z}_a}. \tag{6.21}$$

The iteration stops when the mean-shift vector norm $\left\|\mathbf{x}_i^m(t+1) - \mathbf{x}_i^m(t)\right\|$ is small enough.

2) Part Size Fitting

The goal is to optimize the part size while fixing the appearance $\mathbf{P}$ and location $\mathbf{X}$. Same as the part localization step, the discrimination cost term from (6.10) is held constant. Problem (6.4)–(6.6) can hence be written as

$$\min_{\mathbf{S}} \sum_{m=1}^{N} \left(\sum_{i=1}^{K} d(\mathbf{P}_i, \phi(I_i^m)) + \sum_{i=1}^{K} \sum_{j=1}^{K} v_{i,j}^m \right) \tag{6.22}$$

$$\text{s.t. } 0 \leq \mathbf{s}_i \leq 1, \quad i = 1, ..., K \tag{6.23}$$

$$0 \leq \mathbf{x}_i \pm (1/2)\mathbf{s}_i \leq 1, \quad i = 1, ..., K. \tag{6.24}$$

Here we solve for $\mathbf{S}$ by adopting the scale-space mean-shift algorithm [26]. In the formulation, given the coordinate base $b > 1$, the part size is iteratively updated by

$$\mathbf{s}_i^m(t+1) = \mathbf{s}_i^m(t) b^{r'}, \; r' = \frac{\sum_{r \in \Omega} \sum_{j=1}^{n_p} H(\mathbf{z}_j, r) w(\mathbf{z}_j) r}{\sum_{r \in \Omega} \sum_{j=1}^{n_p} H(\mathbf{z}_j, r) w(\mathbf{z}_j)}, \tag{6.25}$$

where Ω is the search range in the scale space centered at the current part size $\mathbf{s}_i^m(t)$, H is the scale kernel, n_p is the number of pixels inside the current part size, and $w(\mathbf{z}_a)$ is the sample weight defined in (6.21). The iteration stops when $|r'|$ is small enough.

3) Part Model Learning

The goal here is to find the optimal part appearance $\mathbf{P}$ without changing location $\mathbf{X}$ and size $\mathbf{S}$. When optimizing (6.11), the separation cost term from (6.8) is constant with fixed $\mathbf{X}$ and $\mathbf{S}$. Therefore the part appearance model $\mathbf{P}_i$ can be found by

$$\min_{\mathbf{P}_i} \sum_{m=1}^{N} d(\mathbf{P}_i, \phi(I_i^m)) - \sum_{j=1}^{K} d(\mathbf{P}_i, \mathbf{P}_j), \tag{6.26}$$

Note that if the discrimination cost term from (6.10) were ignored, the objective function would be $\sum_{m=1}^{N} d(\mathbf{P}_i, \phi(I_i^m))$, which only considers the similarity between the model and all image regions. The solution for $\mathbf{P}_i$ would then be given by the average of $\phi(I_i^m), m = 1, ..., N$ when L^2-distance is used.

6.5 Hierarchical Partial Classification

To exploit the information from uncertain data without introducing misclassification, a novel classifier that learns a hierarchical structure for the classes and allows for indecision for ambiguous data is proposed [19]. A class hierarchy, i.e., a binary decision tree with one classifier at each non-leaf node, is generated to determine the grouping of classes in higher levels. The grouping labels can serve as coarse categorization results when the exact class label cannot be identified. In the testing phase, the input data instance is examined by layers of classifiers, each of which gives a prediction label. If the instance falls in the indecision range at any layer, the classification procedure stops and returns an incomplete sequence of class labels. In this way, misclassifications are avoided without losing the entire information provided by uncertain data. The concept of hierarchical partial classifier is illustrated in Figure 6.4. The class hierarchy is learned from the training data via an unsupervised clustering procedure. The fully-classified instance (blue) reaches the leaf layer and receives a complete label sequence *A*-*a*-*C1*, while the ambiguous instances (green) stop at middle layers and receive incomplete sequences *A*-*a* and *B*.

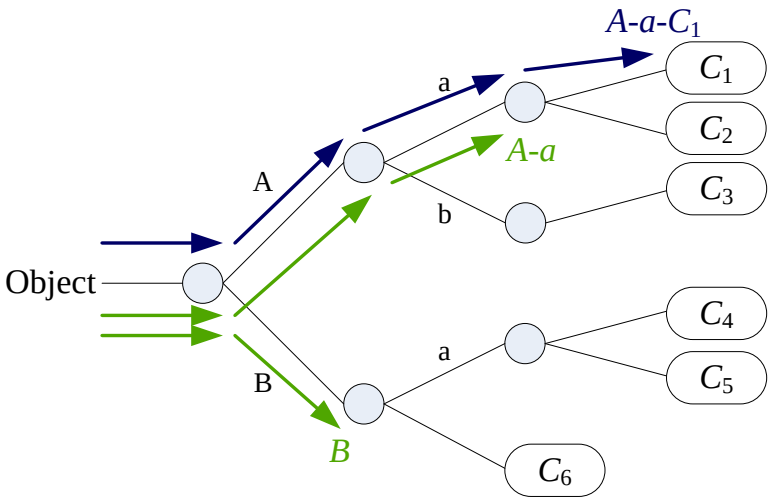


Figure 6.4: An example of hierarchical partial classifier.

6.5.a Unsupervised Construction of Class Hierarchy

The class hierarchy follows a binary tree structure, i.e., each node separates data into two categories. The arrangement of class grouping is learned by an unsupervised recursive clustering procedure as follows. The EM algorithm for mixture of Gaussians (MoG) is applied to separate all data into two clusters, which can be viewed as "positive" and "negative" data respectively. For each species, data are relabeled based on which cluster the majority of this species belongs. A radial bases function (RBF) kernel support vector machine (SVM) is trained with these two super-classes. The above steps are then repeated separately within each cluster until there is only one species in each cluster.

To handle the class imbalance issue, which is caused by the dominance of one or more species in the sampled habitats, a biased-penalty approach is adopted during the SVM training procedure [66]. Rather than using only a single penalty parameter, the penalty parameters for positive and negative classes are set differently to $C_+ = C \cdot N_- / N_{total}$ and $C_- = C \cdot N_+ / N_{total}$, where C is the original penalty parameter, N_+ and N_- denote separately the number of positive and negative training samples, and $N_{total} = N_+ + N_-$.

6.5.b Benefit-based Partial Classification

After the SVM classifier is trained, one needs to define its indecision criterion in order to enable partial classification. In light of evaluating deferred decisions, our task is formulated as an optimization problem as follows. Given the sample data $(\mathbf{x}_i, y_i)$, $i = 1,...,N$, and an SVM decision function $f : \mathbf{R}^d \to \mathbf{R}$ trained by these data, the generalized benefit function of partial classification is defined as

$$B(D) = s_c(\mathbf{x})P(D, \hat{y} = y) - s_w(\mathbf{x})P(D, \hat{y} \neq y), \tag{6.27}$$

where $s_c(\mathbf{x})$ and $s_w(\mathbf{x})$ are score functions for correct and wrong decisions, respectively, and D denotes the event of decisions being made. One can interpret (6.27) as the expected value of total reward for classification, where one earns $s_c(\mathbf{x}_i)$ points for being correct, loses $s_w(\mathbf{x}_i)$ points for being wrong, and gets zero points for indecision with the i-th data point. The score functions can be any nonnegative functions that decrease monotonically with respect to $|f(\mathbf{x})|$ so that a greater importance is added to ambiguous data. We hence choose $s_c(\mathbf{x}) = s_w(\mathbf{x}) = \exp(-|f(\mathbf{x})|)$.

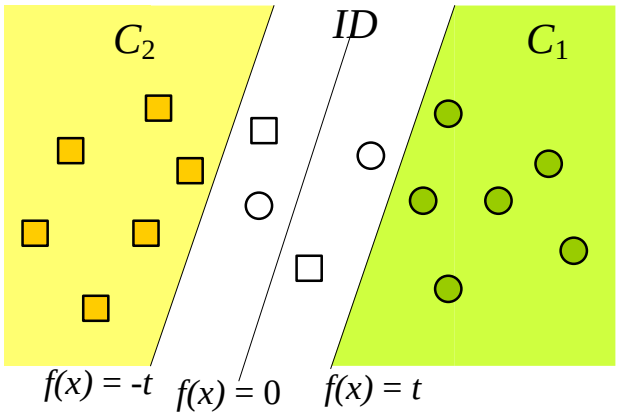


Figure 6.5: Partial classification for an SVM. Ambiguous data have small absolute decision values, so they fall in the indecision domain (ID) and are not assigned to either of the classes.

The goal is to find D that maximizes (6.27). First note that $y \in \{-1,1\}$. Also, a correct decision implies $yf(\mathbf{x}) > 0$ and a wrong decision implies $yf(\mathbf{x}) < 0$. As

illustrated in Figure 6.5, a partial SVM classifier makes a decision only if $|f(\mathbf{x})|$ is greater than a threshold $t > 0$. Therefore (6.27) can be written as

$$\begin{aligned} B(t) &= e^{-yf(\mathbf{x})}P(yf(\mathbf{x}) \geq t) - e^{yf(\mathbf{x})}P(yf(\mathbf{x}) \leq -t) \\ &= \frac{1}{N}\left(\sum_{i=1}^{N} e^{-a_i}\mathbf{1}[t \leq a_i] - \sum_{i=1}^{N} e^{a_i}\mathbf{1}[t \leq -a_i]\right), \end{aligned} \tag{6.28}$$

where $a_i = y_i f(\mathbf{x}_i)$ and $\mathbf{1}[\cdot]$ denotes the 0-1 indicator function. It can be easily verified, as shown in Figure 6.6, that indicator functions are bounded below by exponential functions, i.e.,

$$\sum_{i=1}^{N} \mathbf{1}[t \leq a_i] \geq \sum_{i=1}^{N} (1 - e^{t-a_i}) \tag{6.29}$$

$$-\sum_{i=1}^{N} \mathbf{1}[t \leq -a_i] \geq -\sum_{i=1}^{N} e^{-t-a_i} \tag{6.30}$$

Using (6.29) and (6.30), define an exponential benefit function, $B_{\exp}(t)$, which serves as a lower bound of $B(t)$:

$$\begin{aligned} B_{\exp}(t) &= (1/N)\left(\sum_{i=1}^{N} e^{-a_i}(1 - e^{t-a_i}) - \sum_{i=1}^{N} e^{-a_i}e^{-t-a_i}\right) \\ &= (1/N)\left(\sum_{i=1}^{N} e^{-a_i} - e^{t}\sum_{i=1}^{N} e^{-2a_i}\right) - e^{-t}. \end{aligned} \tag{6.31}$$

An example of the exponential benefit function with respect to decision threshold is visualized in Figure 6.7. Based on this, selecting the decision threshold can be written as an inequality constrained minimization problem:

$$\min_{t} \quad e^{t}\sum_{i=1}^{N} e^{-2a_i} + Ne^{-t} \tag{6.32}$$

$$\text{s.t.} \quad f_{\min} \leq t \leq f_{\max},\ B_{\exp}(t) \geq B_{\exp}(0) \tag{6.33}$$

where $f_{\min} = \min_{i=1,\ldots,N} |f(\mathbf{x}_i)|$ and $f_{\max} = \max_{i=1,\ldots,N} |f(\mathbf{x}_i)|$. Constraints in (6.33) ensure not only feasible solutions but also a gain in the exponential benefit function comparing to full classification. The problem defined in (6.32)–(6.33) is solved by applying the barrier method [8]. The optimal threshold t^* found by solving (6.32)–(6.33) is then used in the testing phase.

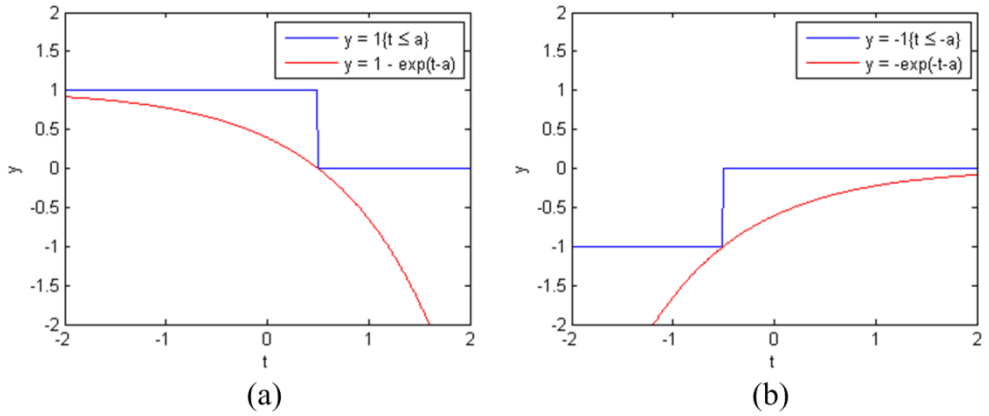


Figure 6.6: Visualization of (a) Equation (6.29) and (b) Equation (6.30) with $a = 0.5$.

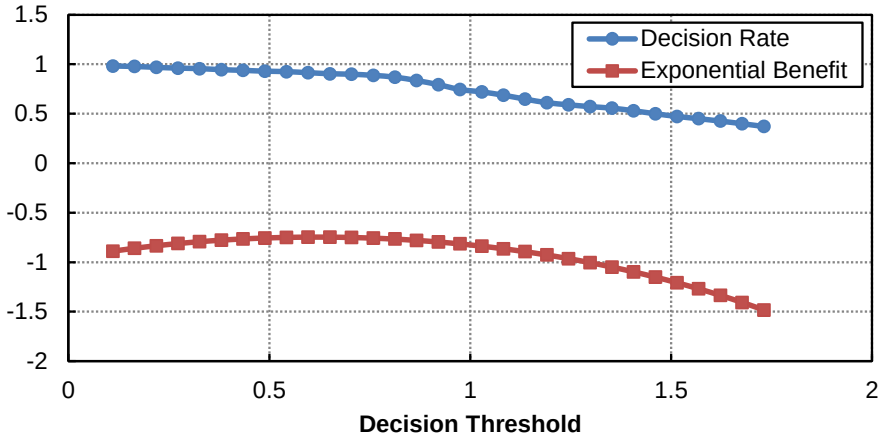


Figure 6.7: Decision rate and exponential benefit vs. decision threshold.

6.6 Probabilistic Outputs for Support Vector Machines

In many cases of object recognition it is desirable for the classifier to produce a probabilistic value for each testing instance being classified. This value, which can be some type of posterior probability or likelihood, reflects the degree of belief by the classifier in the output given to this instance. The support vector machine (SVM), although being powerful in practical applications, produces a decision value for each instance calculated by the separation function. The decision value is proportional to the distance from the separation hyperplane in the feature space. Using the decision value makes it difficult to compare the degree of belief among SVM classifiers since its

magnitude depends on the separation function coefficients and is not calibrated in any way. The hinge loss function in traditional SVM training formulation focuses on only the accuracy, and thus there are no ways one can obtain probabilistic values directly from the SVM function.

To obtain probabilistic SVM outputs, a function fitting approach proposed by Platt [72], which was later improved by Lin et al. [61], is adopted to each node in the proposed hierarchical partial classifier. For a trained SVM with its separation function $f(\mathbf{x})$, the posterior positive class probability is approximated by a sigmoid function

$$P(y=1|\mathbf{x}) = \frac{1}{1+\exp(Af(\mathbf{x})+B)}. \tag{6.34}$$

where $A, B \in \mathbf{R}$ are parameters for the sigmoid function. The parameters A, B are learned based on a regularized maximum likelihood problem, i.e., fitting the sigmoid function to the same set of training data that were used to train the SVM function. The problem is solved by Newton's method and backtracking [61]. For the negative class, the posterior probability can be simply obtained by $P(y=-1|\mathbf{x}) = 1 - P(y=1|\mathbf{x})$.

6.7 Experimental Results

6.7.a Datasets

The proposed method is evaluated on the Fish4Knowledge (F4K) recognition dataset [7]. For comparison purpose, we followed the settings in [49] and conducted the experiments on the top 15 species, which consists of 26,418 fish images. As shown in Figure 6.8, the F4K dataset is highly imbalanced where the most frequent species size is approximately 500 times larger than the least frequent one. Object segmentation mask for each fish image is provided along with the dataset. It worth emphasizing again that the proposed feature learning method remains fully unsupervised even by adopting segmentation, and existing unsupervised techniques such as the GrabCut [75] can be applied to generate segmentation masks on the fly.

Extending the work on video-based fisheries surveys [25], the proposed method is also evaluated on NOAA Fisheries dataset, a self-collected underwater fish dataset as

shown in Figure 6.9. The images are captured by the Cam-trawl system [88] from a mid-water trawl with only simple web patterns in the background and being illuminated by artificial LED lighting. The dataset consists of 2,195 grayscale fish images from 7 species. All fish images are manually labeled by following instructions from fisheries scientists. Compared to F4K dataset, NOAA Fisheries dataset collects images at higher resolution, but discard color information due to the color distortion in deeper water. We apply our automatic fish segmentation algorithm [23] to generate reliable object bounding boxes and segmentation masks.

Figure 6.8: Fish species in Fish4Knowledge dataset. Two largest species (left) and two smallest species (right) are shown to demonstrate the imbalance in class distribution.

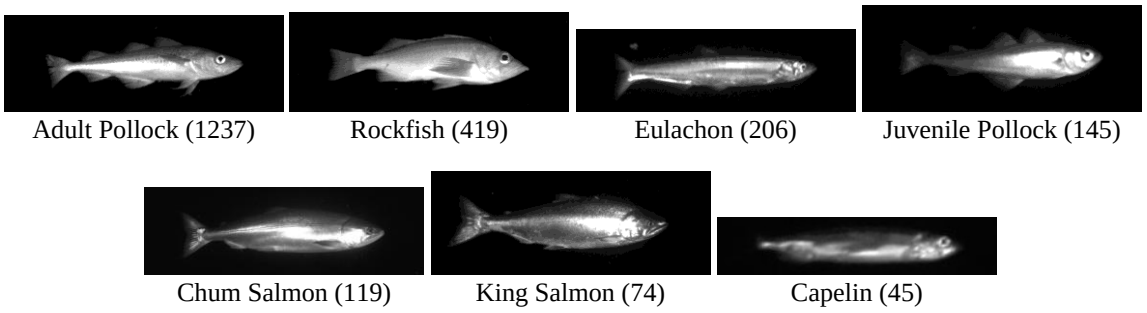


Figure 6.9: Fish species in NOAA Fisheries dataset sorted in the descending order of the number of samples.

6.7.b Implementation Details

Before the analysis, each image is scaled so that the bounding box is no larger than 200×200 pixels with its aspect ratio preserved. The number of parts is empirically determined as $K = 6$ in the experiments (more discussion in Section 6.8.a). Each part is

initialized with a size of 48×48 pixels in the rescaled images. For part features $\mathbf{P}$, the SIFT descriptors and weighted color histogram is used. SIFT descriptors are sampled densely every 4 pixels within the part region. After extraction, the dimensionality is reduced to 128 by applying the principal component analysis (PCA). The weighted histogram is the HSV color histogram weighted by an isotropic kernel that is maximized at the center [18]. Geometric information of parts is also encoded by the location and size of each part in the normalized coordinates $\mathbf{x}_i^m, \mathbf{s}_i^m \in [0,1]\times[0,1]$. In addition to localized features, global features are also taken into account during classifier training. The same SIFT descriptors and weighted color histogram are extracted from the entire bounding box. The global features and local features from each part are concatenated to form the feature vector for one image.

Parameters in the unsupervised non-rigid part model learning algorithm are set empirically as follows. For the stopping criteria in Algorithm 6.1, the convergence of object part models is set as $\sum_{i=1}^{K}\|\Delta\mathbf{P}_i\| \le 0.5$. The maximum iteration is set to 15 as we observed that the algorithm converges within 10 iterations in most cases. The part initialization algorithm is not very dependent on the Gaussian standard deviation σ. A typical value such as 0.5 or 1 (i.e., a 3×3 or 5×5 filter window) works in most cases. The convergence criteria for mean-shift update in part localization is set to $\varepsilon_{\mathbf{x}} = 0.5$. As for part size fitting, the coordinate base is set to $b = \sqrt[5]{2}$ and the search space as $-2 \le \omega \le 2$.

The method for solving (6.26) in the part model learning step depends on the distance metric chosen for features. In our experiments, we measure the distance metric between feature descriptors $\mathbf{P}$ and $\mathbf{Q}$ by the normalized correlation function, i.e., $d(\mathbf{P},\mathbf{Q}) = 1 - \mathbf{P}^* \cdot \mathbf{Q}^*$, where $(\cdot)^*$ denote normalized vectors. Hence (6.26) can be minimized by solving a standard linear programming problem:

$$\min_{\mathbf{P}_i^*, c} \ \sum_{j=1}^{K} \mathbf{P}_i^* \cdot \mathbf{P}_j^* . \tag{6.35}$$

$$\text{s.t.} \ \ \mathbf{P}_i^* \cdot \phi(I_i^m)^* \ge c, m = 1,...,N . \tag{6.36}$$

We use the LIBSVM [16] to train the multi-class classifier, which is implemented by the one-versus-one strategy. A 10-fold cross-validation is applied, and the final classifier is used for the testing set.

6.7.c Feature Learning

In this experiment, the proposed non-rigid part model learning algorithm is compared with several feature descriptors or learning methods. The template model [90] and alignment method [38] are unsupervised techniques, while the part-aware fish features (PAFF) described in Section 6.2 is a supervised technique. Fish images are classified by a flat multiclass SVM, which simultaneously classifies 7 species from the NOAA Fisheries dataset.

The accuracy rates are shown in Table 6.2. The proposed method, which is based on saliency and relaxation labeling, outperforms the fine-grained object recognition methods as well as the supervised PAFF method. Also, the methods based on simple grid-based part initialization and spatial ordering for part matching are compared. For grid initialization we used an 8-part model as in [21], which consists of one block for the head, one block for the tail and a 3×2 grid for the torso. Spatial ordering matches the parts purely based on the part locations. These two approaches are followed by the proposed unsupervised part model learning algorithm proposed in this paper. As shown in Table I, these approaches give lower accuracy rates than the proposed method for not taking into account the potential rotation or deformation of fish body, which are common in images captured from unconstrained natural habitats.

A visualization of some fish parts discovered by the proposed algorithm are shown in Figure 6.10. In addition to the head and the caudal fin (tail), which are most seminal fish parts, the non-rigid part model locates the pectoral fin and classify the fish based on its characteristics. In particular, the pectoral fin part is systematically discovered by the unsupervised non-rigid part model learning. If we remove the corresponding features deliberately, the recognition performance is decreased, as shown in the last row of Table 6.2. This serves as a justification for the high impact of salient features systematically discovered by the proposed non-rigid part model.

Table 6.2: Performance of feature learning methods on NOAA Fisheries dataset

Method	Accuracy (%)
Template [90]	88.4
Alignment [38]	91.6
PAFF [19]	90.1
Proposed (grid, spatial-order)	86.9
Proposed (saliency, spatial-order)	89.5
Proposed (saliency, relax-label)	**93.8**
Proposed (removing pectoral fins)	87.3

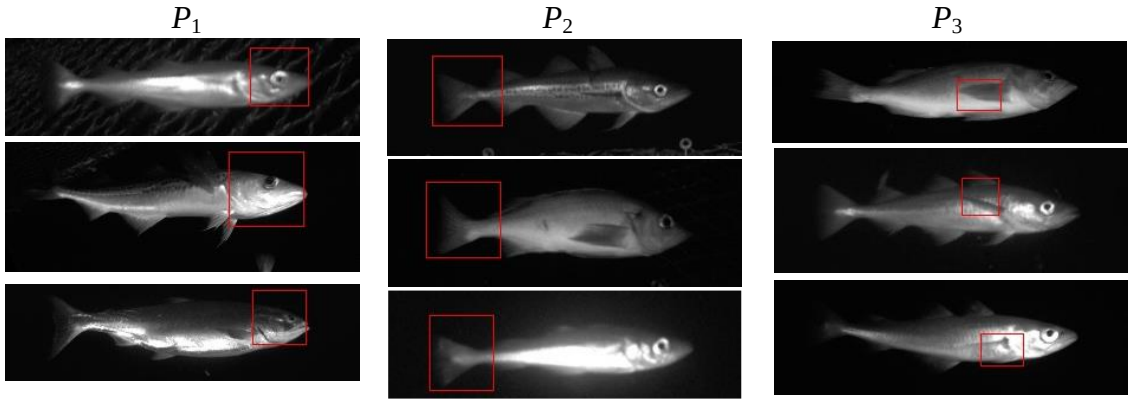


Figure 6.10: Examples of parts detected by the proposed unsupervised feature learning algorithm. Each column shows the parts found by one part model. Meaningful fish body parts are successfully learned regardless changes in size, such as head, caudal fin and pectoral fin.

6.7.d Hierarchical Partial Classification

In this experiment, the proposed hierarchical partial classifier algorithm is evaluated with both NOAA Fisheries and F4K datasets. The proposed algorithm is compared with several baseline algorithms including Flat SVM, principal component analysis (PCA-Flat SVM), taxonomy tree, BEOTR [49] and the classification and regression tree (CART) [47]. A 10-fold cross-validation procedure is applied to train each classifiers. The trajectory voting scheme from [49] is applied to the prediction labels so that results in the same trajectory remain consistent.

Results of F4K and NOAA Fisheries dataset are shown respectively in Table 6.3 and Table 6.4. The proposed hierarchical partial classifier performs the best in terms of precision, recall as well as accuracy. The recognition performance of each algorithm is evaluated through the average precision (AP), average recall (AR) in addition to the accuracy (AC). The metrics AP and AR are defined as follows:

$$AP = \frac{1}{C}\sum_{i=1}^{C} precision(i) = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i}. \tag{6.37}$$

$$AR = \frac{1}{C}\sum_{i=1}^{C} recall(i) = \frac{1}{C}\sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}. \tag{6.38}$$

where TP_i, FP_i and FN_i denote the i-th class true positive, false positive and false negative, respectively. Precision reflects the percentage of correct data in a certain recognized class. Recall indicates the percentage of data from a certain class being correctly recognized. In addition, the precision, recall and F_1-score for each species is reported in Figure 6.11. The F_1-score, which provides an integrated measure combining the precision and recall, is given by

$$F_1\text{-}score = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{6.39}$$

The proposed algorithm achieves an average F_1-score of 93.4% and demonstrates consistency among fish species, as shown in Figure 6.11.

Table 6.3: Performance of top 15 species of F4K dataset

Method	**AP (%)**	**AR (%)**	**AC (%)**
Flat SVM	88.5	76.9	95.7
PCA-Flat SVM	88.9	77.7	95.4
CART [47]	52.9	53.6	87.0
Taxonomy	87.2	76.1	95.3
BEOTR [49]	91.4	84.8	97.5
Proposed	**92.1**	**91.6**	**97.7**

Table 6.4: Performance of NOAA Fisheries dataset

Method	AP (%)	AR (%)	AC (%)
Flat SVM	87.9	83.1	86.6
Proposed	**97.1**	**98.9**	**98.4**

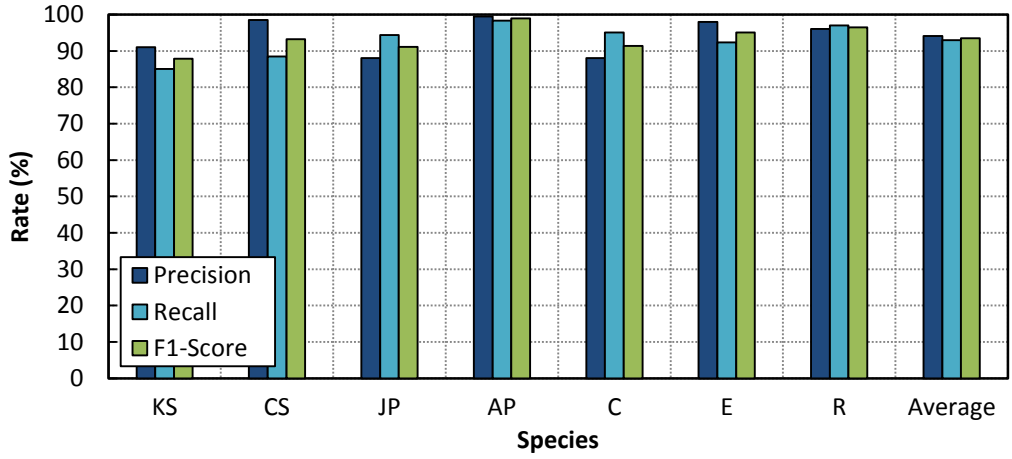


Figure 6.11: Precision, recall and F_1-score of each species from NOAA Fisheries dataset.

6.7.e Running Time

The proposed algorithm is efficient in both training and testing stage. On a laptop PC powered by an Intel Core-i5 520M CPU and 4-GB RAM, each iteration in the unsupervised model learning algorithm takes less than 3 minutes to update. In the testing stage, the process of scale-invariant part detection and feature extraction takes less than 5 seconds for each image.

6.8 Discussion

6.8.a Model Design

In many cases, the performance of unsupervised learning algorithms depends highly on how well the variables are initialized. For the proposed non-rigid part model, one can decide the number of parts to be learned by the part model. This factor not only affects

the power of discrimination but also gives different dimensionality of feature descriptors that represent fish species characteristics.

To investigate into this, the proposed algorithm is tested with different numbers of parts. As shown in Figure 6.12, the best performance is achieved by using 6 parts. It successfully learns rather subtle but meaningful parts of fish body such as the pectoral fins, which are consistent with the domain knowledge in fisheries studies for recognizing a fish species. Also noted that the performance is decreased by using 8-part and 10-part model. The reason is that these models tend to divide some large seminal parts (e.g., head and tail) into small pieces, which leads to worse performance when locating parts. Moreover, the models with more parts are likely to incur overlapping of parts and thus the discrimination cost is greatly increased.

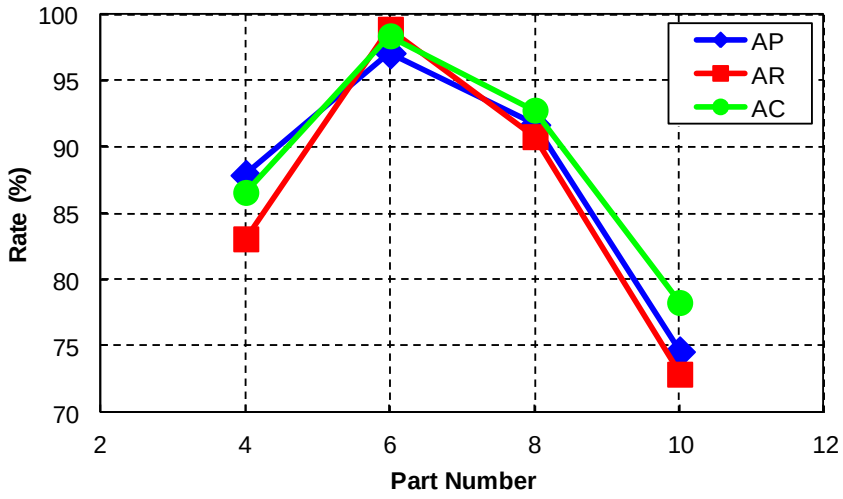


Figure 6.12: Performance vs. number of parts in the non-rigid part model.

6.8.b Partial Classification

To see the effectiveness of handling uncertain data by partial classification, we compare the performance using a flat classifier, a hierarchical full classifier and the proposed method. The flat classifier is a multi-class one-against-all SVM, which classifies objects to all classes simultaneously. The hierarchical full classifier follows the proposed method from learning the class hierarchy to training every SVM, except the decision threshold is set to zero so that all instances are classified down to the lowest

layer. The accuracy and partial decision rate (percentage of data not being classified to the lowest level) of the proposed algorithm are shown in Table 6.5. The flat classifier performs the worst due to the misclassification for those images with high uncertainty. Besides, the subtle visual difference between some species is difficult to learn by a single classifier. The hierarchical classifier learns the relations directly from the data, and thus performs better when the inter-class variation is small. By allowing partial classification using the optimal decision criterion, the accuracy is further increased by 4% while less than 5% of data receive incomplete categorizations.

Table 6.5: Performance of full vs. partial classification algorithms

Classification	**AC (%)**	**PD (%)**
Flat SVM	93.8	-
Hierarchical Full SVM	94.3	0.00
Hierarchical Partial SVM	98.4	4.92

Chapter 7 – Conclusion and Future Work

7.1 Conclusion

In this dissertation, an automatic video analysis system is proposed for camera-based fisheries surveys. There are four topics in underwater video analysis being investigated, including segmentation and tracking with controlled background, segmentation and tracking with uncontrolled background, body length estimation and species recognition.

A novel multiple fish tracking system is proposed for low-contrast and low-frame-rate underwater stereo cameras. Double local thresholding is developed to overcome the challenges posed by unstable illumination and ubiquitous noise in underwater imaging. Using histogram backprojection, a reliable fish segmentation in shape boundary is successfully generated under very low contrast. For low-frame-rate tracking, exploiting various appearance features, the cost function for feature-based object matching acts as an effective metric to find the temporal relationship of targets in the noisy underwater environment. Multiple-target Viterbi data association exploits multiple video frames from the past and takes advantage of dynamic programming to overcome the difficulties of abrupt target motion and frequent entrance/exit due to the low frame rate. Experimental result shows that the proposed system gives a success rate at 88% in terms of fish tracking for low-contrast and low-frame-rate underwater stereo videos.

For more general cases, an innovative object tracking technique for moving cameras based on deformable multiple kernels is proposed. A set of kernels is employed to represent not only the holistic object but also its local parts in terms of color histogram, texture histogram and HOG features. Integrating the deformable part model (DPM) widely used for object detection to the mean-shift optimization algorithm, our proposed method successfully combines the advantages from both techniques: the DPM introduces gradient features and part deformation costs to facilitate multiple-kernel tracking, while the mean-shift algorithm significantly reduces the computations required by typical object detection methods. Experimental results show the proposed method outperforms

the state-of-the-art tracking-by-detection and kernel-based tracking approaches for tracking live fish in challenging underwater videos.

Based on the results of object-height block-based stereo matching, the proposed fish-body tail compensation successfully recovers the transparent caudal fins that are frequently excluded in segmentation stage. By refining the disparity map, each target is reprojected via stereo triangulation and the head-to-tail distance is measured in the 3-D space for body length estimation. Experimental results show that the proposed method gives 6% of mean absolute percentage error in fish length measurement under the low-contrast environment.

Finally, a novel framework comprising feature learning and object recognition for live fish categorization is proposed. The proposed framework is facilitated by unsupervised learning algorithms and thus reduces the requirement of human interference comparing to existing approaches. The non-rigid part model effectively discovers discriminative parts by adopting saliency and relaxation labeling. Fitness, separation and discrimination of parts are considered for finding meaningful representations of fish body parts in a fully unsupervised fashion. On the other hand, data uncertainty and class imbalance are two of the most common issues in practical classification applications. The proposed hierarchical partial classification successfully handled these issues by enabling coarse-to-find categorization and thus retrieving partial information from those ambiguous data which are possibly misclassified or rejected by other algorithms. We further develop a systematic optimization approach to selecting decision criteria for partial classifiers by introducing the exponential benefit function. Experimental results show a favorable performance of fish recognition on both large-scale public dataset and practical highly-uncertain dataset of live fish.

7.2 Future Work

One extension to this work in the future is the capability of learning new classes in species recognition. Like typical machine learning algorithms, the proposed fish recognition system handles only the species included in the training set. In practical testing stage of fisheries surveys, however, it is common to observe species of fish or

invertebrates that were not collected as the training data. It is possible to incorporate an additional class named “unknown” for all those instances that does not belong to any of the classes. Clustering procedures such as the Expectation-Maximization algorithm applied within this unknown class and check if there is any groups with sufficient instances. A potential class is established based on these data, and the unsupervised learning algorithm described in Section 6.4 can be readily applied to build the corresponding non-rigid part model. In this way, the proposed system enables an adaptive species identification function while keeping itself unsupervised.

In addition, after effective fish tracking, size estimation and species recognition, it is a natural progression to understand the actions and behaviors of fish. One approach to this is analyzing the motion trajectories obtained from tracking and the correlation with species. For example, the fish trajectory dynamics is analyzed based on the clustering for each fish species [81] in order to understand fish behavior and detect rare events, which may present the cue of interest for fisheries scientists. Action analyses would be further facilitated by incorporating the use of 3-D models of generic fish shapes. Precise pose estimation can be achieved by fitted such 3-D models to the targets. When combined with species prediction, automated behavior analysis would greatly reduce the effort required for scientists to analyze the correlation between species and specific reactions and schooling behavior. This allow substantial gains to be made in topics based on animal behaviors, such as fisheries bycatch reduction and ecosystem assessment.

References

[1] K. Ali, S. Manganaris, and R. Srikant, "Partial Classification Using Association Rules," in *KDD*, 1997, pp. p115-118.

[2] A. Azim and O. Aycard, "Multiple pedestrian tracking using Viterbi data association," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 706-711.

[3] Y. Baram, "Partial classification: The benefit of deferred decision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 20, pp. 769-776, 1998.

[4] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 36, pp. 1614-1627, 2014.

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 33, pp. 1806-1819, 2011.

[6] J. M. Bloemer, T. Brijs, K. Vanhoof, and G. Swinnen, "Comparing complete and partial classification for identifying customers at risk," *International journal of research in marketing,* vol. 20, pp. 117-131, 2003.

[7] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher, "Supporting ground-truth annotation of image datasets using clustering," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 1542-1545.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge University Press, 2009.

[9] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 23, pp. 1222-1239, 2001.

[10] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*: " O'Reilly Media, Inc.", 2008.

[11] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 33, pp. 1820-1833, 2011.

[12] S. Butail and D. A. Paley, "Three-dimensional reconstruction of the fast-start swimming kinematics of densely schooling fish," *J R Soc Interface,* vol. 9, pp. 77-88, Jan 7 2012.

[13] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision,* vol. 22, pp. 61-79, 1997.

[14] T. F. Chan and L. A. Vese, "Active contours without edges," *Image processing, IEEE transactions on,* vol. 10, pp. 266-277, 2001.

[15] E. Chandra and K. Kanagalakshmi, "Noise Suppression Scheme using Median Filer in Gray and Binary Images," *International Journal of Computer Applications,* vol. 26, pp. 49-57, 2011.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

[17] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 35, pp. 1577-1591, 2013.

[18] C.-T. Chu, J.-N. Hwang, H.-I. Pai, and K.-M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *Multimedia, IEEE Transactions on,* vol. 15, pp. 1602-1615, 2013.

[19] M.-C. Chuang, J.-N. Hwang, F.-F. Kuo, M.-K. Shan, and K. Williams, "Recognizing Live Fish Species by Hierarchical Partial Classification based on the Exponential Benefit," in *Image Processing (ICIP), 2014 21th IEEE International Conference on,* 2014.

[20] M.-C. Chuang, J.-N. Hwang, and C. S. Rose, "Aggregated segmentation of fish from conveyor belt videos," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 1807-1811.

[21] M.-C. Chuang, J.-N. Hwang, and K. Williams, "Supervised and Unsupervised Feature Extraction Methods for Underwater Fish Species Recognition," in *Computer Vision for Analysis of Underwater Imagery (CVAUI), 2014 ICPR Workshop on*, 2014, pp. 33-40.

[22] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A Feature Learning and Object Recognition Framework for Underwater Fish Images," *Image Processing, IEEE Transactions on,* (submitted) 2015.

[23] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Automatic fish segmentation via double local thresholding for trawl-based underwater camera systems," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 3145-3148.

[24] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Multiple fish tracking via Viterbi data association for low-frame-rate underwater camera systems," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, 2013, pp. 2400-2403.

[25] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Tracking Live Fish from Low-Contrast and Low-Frame-Rate Stereo Videos," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 25, pp. 167-179, Jan. 2015.

[26] R. T. Collins, "Mean-shift blob tracking through scale space," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. II-234-40 vol. 2.

[27] D. Comaniciu and P. Meer, "Mean-shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 603-619, 2002.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, pp. 564-577, 2003.

[29] C. Costa, A. Loy, S. Cataudella, D. Davis, and M. Scardi, "Extracting fish size using dual underwater cameras," *Aquacult Eng*, vol. 35, pp. 218-227, 2006.

[30] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 886-893.

[31] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, 2013, pp. 494-502.

[32] I. Everson, M. Bravington, and C. Goss, "A combined acoustic and trawl survey for efficiently estimating fish abundance," *Fisheries research*, vol. 26, pp. 75-91, 1996.

[33] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 502-509.

[34] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, pp. 1627-1645, 2010.

[35] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 22, pp. 67-92, 1973.

[36] G. D. Forney Jr, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268-278, 1973.

[37] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning Category-Specific Dictionary and Shared Dictionary for Fine-Grained Image Categorization," *Image Processing, IEEE Transactions on*, vol. 23, pp. 623-634, 2014.

[38] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-Grained Categorization by Alignments," presented at the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 580-587.

[40] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. (2012). *Discriminatively trained deformable part models, release 5*. Available: http://people.cs.uchicago.edu/~rbg/latent-release5/

[41] C. Goering, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," presented at the Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2014.

[42] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 572-584, 1998.

[43] G. D. Hager, M. Dewan, and C. V. Stewart, "Multiple kernel tracking with SSD," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I-790-I-797 Vol. 1.

[44] D. G. Hankin and G. H. Reeves, "Estimating total fish abundance and total habitat area in small streams based on visual estimation methods," *Canadian journal of fisheries and aquatic sciences*, vol. 45, pp. 834-844, 1988.

[45] R. M. Haralick and L. G. Shapiro, *Computer and robot vision*: Addison-Wesley Longman Publishing Co., Inc., 1991.

[46] E. Harvey, M. Cappo, M. Shortis, S. Robson, J. Buchanan, and P. Speare, "The accuracy and precision of underwater measurements of length and maximum body depth of southern bluefin tuna (Thunnus maccoyii) with a stereo–video camera system," *Fisheries Research*, vol. 63, pp. 315-326, 2003.

[47] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning* vol. 2: Springer, 2009.

[48] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on,* vol. 8, pp. 179-187, 1962.

[49] P. X. Huang, B. J. Boom, and R. B. Fisher, "Hierarchical classification with reject option for live fish recognition," *Machine Vision and Applications,* vol. 26, pp. 89-102, 2014.

[50] J. M. Jech, C. N. Rooper, G. R. Hoff, and A. De Robertis, "Assessing habitat utilization and rockfish (Sebastes spp.) biomass on an isolated rocky ridge using acoustics and stereo image analysis," *Canadian Journal of Fisheries and Aquatic Sciences,* vol. 67, pp. 1658-1670, 2010.

[51] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision,* vol. 1, pp. 321-331, 1988.

[52] C. Kim and J.-N. Hwang, "Video object extraction for object-oriented applications," *Journal of VLSI signal processing systems for signal, image and video technology,* vol. 29, pp. 7-21, 2001.

[53] C. Kim and J.-N. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 12, pp. 122-129, 2002.

[54] A. Klimley and S. Brown, "Stereophotography for the field biologist: measurement of lengths and three-dimensional positions of free-swimming sharks," *Mar Biol,* vol. 74, pp. 175-185, 1983.

[55] D.-J. Lee, S. Redd, R. Schoenberger, X. Xu, and P. Zhan, "An automated fish species classification and migration monitoring system," in *Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE*, 2003, pp. 1080-1085.

[56] D.-J. Lee, R. B. Schoenberger, D. Shiozawa, X. Xu, and P. Zhan, "Contour matching for a fish recognition and migration-monitoring system," in *Optics East*, 2004, pp. 37-48.

[57] J.-H. Lee and C.-H. Won, "Topology preserving relaxation labeling for nonrigid point matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 33, pp. 427-432, 2011.

[58] K.-H. Lee, J.-N. Hwang, G. Okapal, and J. Pitton, "Driving recorder based on-road pedestrian tracking using visual SLAM and Constrained Multiple-Kernel," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, 2014, pp. 2629-2635.

[59] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International journal of computer vision,* vol. 77, pp. 259-289, 2008.

[60] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 30, pp. 1728-1740, 2008.

[61] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine learning,* vol. 68, pp. 267-276, 2007.

[62] J. Lines, R. Tillett, L. Ross, D. Chan, S. Hockaday, and N. McFarlane, "An automatic image-based system for estimating the mass of free-swimming fish," *Computers and Electronics in Agriculture,* vol. 31, pp. 151-168, 2001.

[63] P. Marquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 36, pp. 2-17, 2014.

[64] I. H. McQuinn, Y. Simard, T. W. Stroud, J.-L. Beaulieu, and S. J. Walsh, "An adaptive, integrated "acoustic-trawl" survey design for Atlantic cod (Gadus morhua) with estimation of the acoustic and trawl dead zones," *ICES Journal of Marine Science: Journal du Conseil,* vol. 62, pp. 93-106, 2005.

[65] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 8, pp. 525-538, 1998.

[66] K. Morik, P. Brockhausen, and T. Joachims, "Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring," Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund1999.

[67] M. Nery, A. Machado, M. F. M. Campos, F. L. Pádua, R. Carceroni, and J. P. Queiroz-Neto, "Determining the appropriate feature set for fish classification tasks," in *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*, 2005, pp. 173-180.

[68] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 971-987, 2002.

[69] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica,* vol. 11, pp. 23-27, 1975.

[70] D. J. Park and C. Kim, "A Hybrid Bags-of-Feature model for Sports Scene Classification," *Journal of Signal Processing Systems*, 2014.

[71] F. Pitié, S.-A. Berrani, A. Kokaram, and R. Dahyot, "Off-line multiple object tracking using candidate selection and the viterbi algorithm," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2005, pp. III-109-12.

[72] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers,* vol. 10, pp. 61-74, 1999.

[73] F. Porikli and O. Tuzel, "Multi-kernel object tracking," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005, pp. 1234-1237.

[74] G. W. Pulford, "Multi-target viterbi data association," in *Information Fusion, 2006 9th International Conference on*, 2006, pp. 1-8.

[75] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," presented at the ACM Transactions on Graphics (TOG), 2004.

[76] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, pp. 99-121, 2000.

[77] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, pp. 7-42, 2002.

[78] J. Serra, *Image Analysis and Mathematical Morphology*. London, UK: Academic Press, 1982.

[79] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised Discovery of Mid-Level Discriminative Patches," presented at the European Conference on Computer Vision (ECCV), 2012.

[80] C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. B. Fisher, "Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos," *VISAPP (2)*, vol. 2008, pp. 514-519, 2008.

[81] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan, "Automatic fish classification for underwater species behavior understanding," in *1st ACM International Workshop on ARTEMIS*, 2010, pp. 45-50.

[82] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553-2561.

[83] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. I-511-I-518 vol. 1.

[84] D. Walther, D. R. Edgington, and C. Koch, "Detection and tracking of objects in underwater video," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I-544-I-549 Vol. 1.

[85] L. Wang and N. H. Yung, "Extraction of moving objects from their background based on multiple adaptive thresholds and boundary evaluation," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 11, pp. 40-51, 2010.

[86] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for Generic Object Detection," in *International Conference on Computer Vision*, 2013.

[87] K. Williams, R. Towler, M.-C. Chuang, and J.-N. Hwang, "Camtrawl: a Combined Camera-Trawl System for High Resolution Non-Lethal Sampling of Marine Environments," in *American Fisheries Society 141th Annual Meeting*, Seattle, WA, USA, 2011.

[88] K. Williams, R. Towler, and C. Wilson, "Cam-trawl: a combination trawl and stereo-camera system," *Sea Technology,* vol. 51, pp. 45-50, 2010.

[89] J. Wu, N. Liu, C. Geyer, and J. M. Rehg, "C4: A Real-Time Object Detection Framework," *Image Processing, IEEE Transactions on,* vol. 22, pp. 4096-4107, 2013.

[90] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *Advances in Neural Information Processing Systems*, 2012, pp. 3122-3130.

[91] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys (CSUR),* vol. 38, p. 13, 2006.

[92] H. Zhang, S. Cai, and L. Quan, "Real-Time Object Tracking with Generalized Part-Based Appearance Model and Structure-Constrained Motion Model," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 1224-1229.

[93] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.

[94] L. Zhang and L. van der Maaten, "Preserving structure in model-free tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 36, pp. 756-769, 2014.

[95] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2013, pp. 729-736.

[96] Y. Zhao and G. Taubin, "Real-time median filtering for embedded smart cameras," in *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, 2006, pp. 55-55.

[97] Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 643-649, 2006.

[98] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 22, pp. 675-684, 2000.

[99] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, pp. 28-31.

Vita

Meng-Che Chuang was born and raised in Taipei, Taiwan. He received a Bachelor of Science degree from the Department of Electrical Engineering, National Taiwan University in 2008. In 2010, he joined Information Processing Laboratory at the University of Washington and pursued his Ph.D. degree. He received a Master of Science degree from the Department of Electrical Engineering, University of Washington in 2012. In 2015, he received a Doctor of Philosophy degree from the Department of Electrical Engineering, University of Washington. His research interest includes computer vision, digital image/video processing and machine learning.