State Estimation Strategies for Autonomous Underwater Vehicle Fish Tracking Applications

by

Jun Zhou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Jun Zhou

Abstract

As the largest unexplored area on earth, the underwater world has unlimited attraction to marine scientists. Due to the complexity of the underwater environment and the limitations of human divers, underwater exploration has been facilitated by the use of submarines, Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs). In recent years, use of autonomous control systems being integrated with visual sensors has increased substantially, especially in marine applications involving guidance of AUVs.

In this work, autonomous fish-tracking via AUV with vision servoing control system is studied with the purpose of assisting marine biologists in gathering detailed information about the behaviors, habits, mobility, and local and global distributions of particular fish species.

The main goal of this work in this thesis is to develop an AUV sensing system, including both video and auxiliary sonar, which has the ability to carry out visually guided autonomous tracking of a particular species of fish, Large Mouth Bass. A key in enabling fish-tracking involves the development of a vision-processing algorithm to measure the position of the vehicle relative to the fish. It is challenging because of the complex nature of the underwater environment including dynamic and varied lighting conditions, turbulent water, suspended organic particles and various underwater plants and animals, and the deformable body of fish while swimming. These issues cause target fish identification by computer vision processing extremely difficult.

In automated fish-tracking work, we provide two valid and efficient segmentation and recognition vision algorithms to identify a fish from the natural underwater environment: one is a feature extraction algorithm based on Gabor filter texture segmentation and a new approach that we call projection curve recognition. It is able to extract the feature on the fish tail and body and successfully describe the fish as two straight line segments. The second algorithm is SIFT based fish recognition algorithm. The SIFT approach introduced by David Lowe in 1999 extracts distinctive invariant features to scaling, illumination, rotation or translation of the image. The reliable keypoints matching in the database of keypoints from target fish is implemented by Best-Bin-First (BBF) algorithm. Clustering keypoints that agree on the possible object with Hough transform are identified as the object fish, reliable recognition is possible with as few as 3 features. Finally, a dynamic recognition process was designed using continuously updated fish model to match and recognize the target fish from a series of video frames. The SIFT Based recognition algorithm is effective and efficient in identifying Large Mouth Bass in a natural cluttering underwater environment.

For a monocular camera system, the depth of field is extremely hard to obtain by vision processing. Hence, the system is augmented with a forward-looking digital image micro sonar. With the sonar image processing algorithm, the target fish is recognized. Sonar can not only provide the relative range between the fish and AUV, but also assist in identifying the target.

Finally, the relative position and orientation of the fish in the image plane is estimated using an image processing method, transforming the coordinates between camera, sonar and AUV, and applying the estimation algorithm. The results of off-line data processing taken from a natural Lake environment shows these computer vision algorithms for identifying fish and state estimation are efficient and successful. The proposed system has potential to enable a vision servo control system of AUV to reliably track a target fish in natural underwater environment.

Acknowledgements

I am truly indebted to many people who have contributed to this research. Without their support and assistance, this work would have been impossible.

First of all, I would express my deepest appreciation to my supervisor Professor Christopher M. Clark, who has been the most influential person throughout the whole process of the research. He has introduced me to this exciting field and has given me academic freedom; he has provid excellent guidance and valuable comments both on academics and otherwise. I really appreciate him not only for spending so much time helping me out with research problems, but also for his consideration and care for students, from the very beginning of defining the research topic, to the very end of editing the wording of the thesis. It has been such a pleasure to work with him and I have learned so much from him during my entire research project in the past two years.

I would also like to extend my endless thanks to my co-supervisor Professor Jan Paul Huissoon, who not only give me valuable advice on my research, but also spent considerable time and energy in my experiments and other academic activities.

I offer my thanks to Professor Kaan Erkorkmaz and Professor Mustafa Yavuz for agreeing to read my thesis. Many thanks also go to the group of LAIR who have assisted me in numerous ways and have made my time at the LAIR an enjoyment and beneficial experience.

Finally, I am very grateful to my parents and my brother for their unconditional support and understanding throughout my studying period. A special thank you must also go to my husband, whose love and encouragement have given me confidence and momentum, and have changed my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The purpose of our research is to develop a visual servoing system for underwater vehicle to assist marine biologists in gathering detailed information about the behaviors, habits, mobility, and local and global distributions of particular fish species.

More than seventy percent of the earth's surface is currently covered by ocean, river and lake, although as the largest area on the world, they contain abundant aquatic biology, huge power, and various mine and oil resource, and play a significant role on globe climate, shipping and feeding for human being, they still lack enough comprehension and adequate exploitation. Especially over the past century, notable changes have happened to underwater environment in virtue of industrialization and human activities, which have begun to effect on human's life profoundly [60].

In order to adequately understand, reasonably develop and appropriately protect this largest unexplored field, more and more scientists and engineers are being attractive to plunge into research and investigation in the mysterious underwater world. Generally, the changes of animal underwater behaviors, population density and migration patterns reflect the changes in the environment, climes and so on. By studying animals through direct and indirect observations, scientists can deeply grasp the knowledge and characters about not only the animals but also the underwater world.

Nature underwater fish unlike the terrestrial animals can be hardly observed the behavioral routines living in their native habitats. In addition, it is impossible to acquire the particular information about behaviors, habits, mobility, and local and global distributions of fish in nature underwater world through aquarium studies.

Due to the complexity of the underwater environment and the limitations of human divers, underwater exploration has been facilitated by the use of submarines. These submersible robots have opened a new window which exposes the lives of underwater animals and offer scientists the capability for both direct and indirect observations of the ocean's physical properties and its ecology.

Most present submersible robots technologies are adaptive to fixed target tracking, short period observations, or dynamic object following under human assistance. Therefore, there is an exigent requirement for fish study to develop a Vision based AUV tracking technology that has ability to identifying and following an interested target autonomously. This aim is to detect and track the target in long period, to observe and investigate subtle changes of growth, feeding, migration and aggregation.

1.2 Underwater Vehicles

Because of the danger of underwater environment and the restriction of human divers, the demand of Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) in the underwater exploration fields is increasing rapidly.

Traditionally, Remotely Operated Vehicles (ROVs) have been adopted as a primary tool for underwater scientific exploration, surveying, inspection, searching and salvage, and mine countermeasures [52].Although they have proven to be useful in aiding certain tasks, they still require human intervention for overall guidance of the vehicle. ROVs cannot yet recreate the full sensory presence available from within a manned research submarine. Safety and financial considerations, coupled with the desire to observe greater depths in short range for longer durations, has lead to a major increase into the research and development of submersibles robotic vehicles that introduce autonomous control technologies.

An Autonomous Underwater Vehicle (AUV) is a robotic device which travels through the water by a propulsion system, controlled and guided by an onboard computer using various surrounding environment information extracted by sensors. Thus, AUVs do not require any surface support and man assistance to make navigational decisions. Also, the clumsy tether which takes a risk of being entangled or broken during underwater task can be threw away. These make AUVs more flexible and mobile.

The potential of AUVs in furthering the exploration of the deep sea have been widely realized. Some scientists and organizations have applied AUVs to scientific research such as studying oceanography, geology, marine creatures, etc., and also to environmental monitoring such as waste disposal surveillance. Others have focused on the commercial implementation of AUVs in surveying the ocean floor for valuable minerals, oil and gas. In addition, AUVs have been identified as being especially useful in carrying out hazardous underwater tasks such as deep-sea object retrieval, military countermeasures and detection of undersea mines [59].

The applications of AUV's are still growing with the high technology development. As autonomous control and computer vision improve, there has been a definite trend toward more robust methods of autonomous navigation such as vision servo control, which enable underwater robots to autonomously search in regular patterns, follow along fixed natural or artificial features, and track behind dynamic targets. These capabilities are essential to tasks like exploring geologic features, cataloging reefs, and studying marine creatures, as well as inspecting pipes and cables, and assisting divers and ROV operators. In recent years, among these marine applications, target tracking with ROV and AUV using visual servoing is of particular interest, especially for enabling short-range applications such as jellyfish observing , fish tracking, cable following and docking etc. [43][44].

1.3 Vision sensor

In the field of autonomous robotics, the primary challenge of robot technology development is the sensing issues. Sensors like eyes of a robot enable the robot to perceive its relationship to the physical world, and to guide itself for moving or interacting, which means the robot has the capability for self-adaptation to a changing working environment. In an indoor environment, laser sensors or vision system are used to detect artificial cues for robots localization. In outside, floating and aerial vehicles may use the Global Position System (GPS) for navigation. But in underwater, GPS signals cannot be received, submersible vehicles have to adopt other types of sensor for navigation and relative position [60]. Current popular sensors used in underwater robots are Doppler velocity log, flow meters, compass and gyro. The major drawback of this method is that the quality of navigational decisions is highly dependant on having accurate sensors which are typically very expensive due to the errors accumulation rapidly increase with time. Also, it should be noted that these methods are used for estimating the vehicle position and cannot directly estimate the target position [29]. Although sonar technology might provide a non-intrusive method of tracking underwater objects, it is difficult to avoid problems due to the multiple path effect and acoustic shading [36]. Furthermore, it is also difficult to extract the target from the cluttered natural environment, usually depending on the assistance of the pilots' experience.

Within the last decade, systems being integrated with vision based control which primarily requires the use of vision sensing have drawn a lot of attention, especially in the field of underwater robots. Vision is a high-resolution sensing modality that provides information about the surrounding environment at high bandwidth. Vision sensors are relatively inexpensive and consume low power, and yet they are capable in capturing rich information from the environment such as color, texture, shape, dynamic properties and geometric properties [69]. The primary role of a vision system is to extract the information of interest from an image and use this information to guide a host system. As a computer's ability to process multiple frames increases, the goal of real time image processing is realized.

In our work, the major objective is develop computer vision algorithms providing the fish position for visual servo control system of AUV to tracking the target fish in natural underwater environment. Generally, image processing in the natural underwater environment is a very complex task. In underwater systems, turbulent water and dynamic lighting conditions greatly affect the images captured. Also, the clutter and disorder of surrounding cause the image background complicated for identify object. Therefore, the crucial core in the design of a vision sensor for fish-tracking involves the segmentation and recognition process.

1.4 Visual Servoing

The visual fish-tracking application falls within the broad research area known as position-based visual servoing. Visual servo can be considered the fusion of computer vision, robotics and control. The applications of computer vision are new tracking technology compared to the traditional radar or sonar scanning. Historically, image processing technology was used on the static image, such as analyzing could layer on the satellite maps for weather forecast or other meteorology application and applying pattern recognition method for products picking up, labeling and error detecting in the automation product line.

In the recent ten years, dynamic image processing that processes a series of video frames for object identifying and motion detection was extensively used in different fields, especially in visual servo control that is a relatively new topic. The most popular applications of vision systems in both land and marine tasks include 3D machine vision [34], obstacle avoidance [66], object recognition, maintenance and inspection [55] and object tracking [13].

There are some typical image approaches are applied into visual serving control robots for those man-mad objects that have distinct feature such as corners, lines and spots in the familiar ground scene, indoor environment. Jennings, Murray, and Little used correlation-based stereo vision to build the map of environment with occupancy grids to assist Robots to navigate and autonomously explore unknown, dynamic indoor environment [22]. Amidi et al. provided color segmentation or template-based detection to identify a ground target [11]. Papanikolopoulos , Khosla and Kanade, described the image Jacobian method compute the motion vector of discrete displacements each instant of time for real-time visual tracking of arbitrary 3D objects at unknown velocities [53]. Westmore and Wilson extracted features from an image to compute 3D reconstruction of the motion of a target object[25]. Batavia, Pomerleau, and Thorpe used an optical flow technology obstacle detection system to detect vehicles presenting to the blind spot of the car on local street or highway [17].

In the underwater application of visual servo technology, there is a major limiting factor that the image processing is difficultly to identify the target from natural underwater environment that is disordered and blurry lacking of artificial scene and obvious feature. Therefore, this requires more complicated image processing methods that enable identifying characteristics of the target to segment and recognize reliably for vision servo control system.

Although the application of visual servoing make little mention of underwater applications, companying with the fast development of computer vision, scientists and technicians have made substantial progress in the visual navigation of submersible vehicles relative to the ocean floor. Furthermore, recent publications have touched on the topic of automated object tracking in natural underwater environments, some of which have even implemented a real time tracking. The Robotic Systems Lab at Australian National University (ANU) is developing a visually-guided autonomous underwater vehicle, Kambara, (shown in Fig. 1.1), for underwater exploration and observation. A position-based visual servo control of fixed and slow moving targets using visual position feedback and sensor-based orientation feedback have been achieved and test in the deep ocean [15]. Rife et al. implemented real-time Gelatinous animals target following in the deep ocean by using the Monterey Bay Aquarium Research Institute (MBARI) ROV Ventana with a stereo video system. It has demonstrated fully autonomous tracking of a gelatinous animal in the waters of Monterey Bay, California, (shown in Fig. 1.2). The device, intended as an aid for human ROV pilots, will increase the feasible duration of observational experiments in the field of marine biology [60]. Minami, Agbanhan and Asakura who are all based at Fukui University Faculty of Engineering, Japan, present a model-based image recognition and 1-step-GA evolution to track a target fish in a small tray by the visual servoing of a manipulator in the lab, (shown in Fig. 1.3). GA-based visual servoing has been implemented in a real system for a sampling period of 120ms, and the results have shown its effectiveness for successfully tracking a moving target fish in tray under lab condition [48].

Currently, there have been many other visually-guided AUVs for ocean animals tracking deployed for research purposes that are still remain off-line processing or open servo loops. Fan and Balasuriya used the optical flow technique for 20Hz off-line fish tracking, video collected in the open ocean [29]. Kocak et al. applied vision techniques in off-line analysis and classification of marine zooplankton data [38]. Yuan et al. pursued a neural network for online learning and recognition to detect underwater targets [72].

1.5 Underwater Sonar

Since electromagnetic waves and light attenuate far more rapidly over very short distances underwater, the lack of long range vision is a great limitation in underwater working. To meet the need arising to track underwater objects where optical systems fail, engineers have turned from the visible light spectrum to another form of transmittable energy underwater: sound. Sound is also absorb in the dense water environment, but not over as short a distance as light. Although the resolution of acoustic imaging does not approach optics, it does provide a remarkable extension of our vision [9]. A sonar system consists of both a projector and hydrophone which is capable of transmitting and receiving acoustic signals. Examples of active sonars

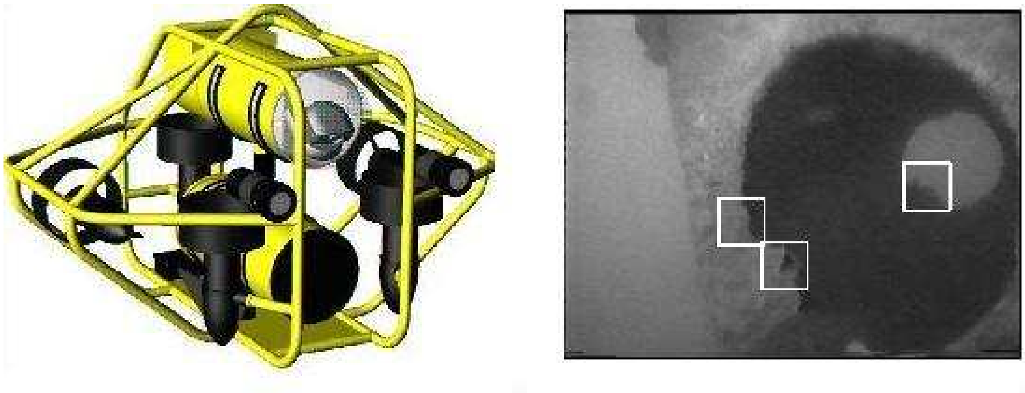


Figure 1.1: Kambara AUV and a followed underwater support pile picture

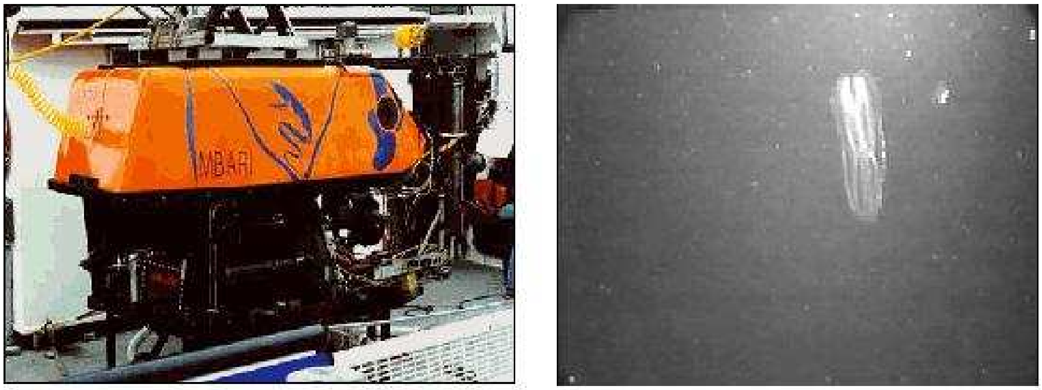


Figure 1.2: MBARI ROV Ventana and the tracked jellyfish picture

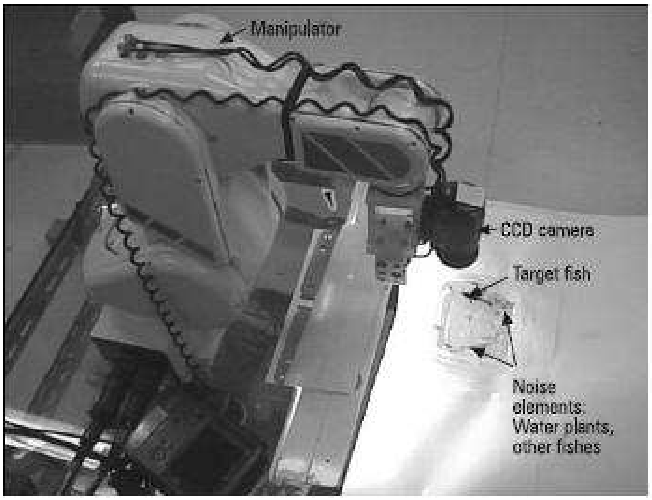


Figure 1.3: fish tracking in a small tray in lab on Fukui University

include echo sounders, commercial side scan sonar and many types of sub bottom profilers. The advanced sonar imaging service provides a window into that dark and murky environment. This rich underwater data provides new challenges and possibilities within the field of underwater visualization. Those working underwater, including oceanographers, marine geologists, ROV, and AUVs now depend heavily on imaging sonar to transform the things we cannot see underwater into pictures.

Forward looking scanned sonars are used for a varied number of underwater machine vision applications, such as obstacle avoidance, mid-water mine detection and surveillance. These sonars display plan position (distance and orientation) indicator (PPI) image of acoustic energy returns as brightness levels. Modern devices use an array of hydrophones which allows for much faster updates of images. They are preferred to order mechanically scanned devices taking several seconds for each scan [35].

The Ocean Systems Laboratory has been involved in the development of both automatic classifications of objects (divers, ROVs, etc.) [20] and obstacle avoidance systems using forward looking multibeam sonar images [58]. Essential to these applications are the detection, tracking and motion estimation of objects / obstacles in a sonar image sequence [32].

Usually sonar systems have a poor resolution as compared to video images. The resolution can be improved by an increase in the frequency of operation of the sonar and by a shorter pulse width. However, increased frequency implies an increase in transmission absorption which reduces the range. Moreover, it may not be possible to operate active sonar under certain underwater conditions. Despite these limitations, the development of high-speed, multi-frequency continuous scan sonar at has resulted in the acquisition of extremely accurate, high-resolution bathymetric data. This leads to production of a very large amount of data which is to be processed in a comparatively short time to ensure realtime detection of obstacles or objects for the AUV. This processing may invoke methods from adjacent research areas: low resolution sonar, medical ultrasound techniques, digital image processing, methods for object tracking,

1.6 Project Scope

1.6.1 The goal and challenge of project

As discussed in Section 1.1, with the rapid development of new technology, AUVs have a range of applications in both commercial and non commercial underwater exploration and research. Of particular interest in this project is the developing a visual system including both video and auxiliary sonar for AUV, which is able to autonomously track a particular species of fish, larger mouth bass, shown in the Fig.1.4, and monitor its behavior in short range (i.e.1-2m) in its natural underwater environment.

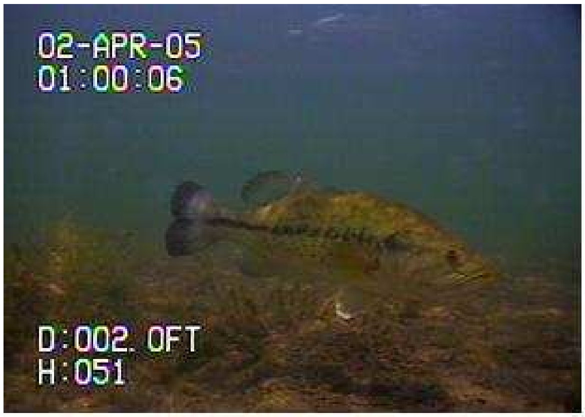


Figure 1.4: A large mouth bass picture taken from ROV within Paradise Lake, Ontario, Canada

Fish tracking is a fairly difficult task to achieve in the natural underwater environment due to the frequent presence of noise and other issues specific to the environment. For example, light attenuates exponentially with distance in water, which makes the quality of underwater images very poor. The fish do not appear as exclusive bright against dark backgrounds. Illumination backscatters to the camera, producing a relatively bright and non-uniform background image. Suspended organic particles and sporadic marine growth introduce continual small fluctuations to this background image.

In generally, the volume of a Micro ROV or AUV is limited, hence there is only one camera can be installed into tight pressure hull. For this monocular camera system, high-resolution imaging sonar is required to assist in depth of image measurement and also benefit to target identification.

Therefore, the main goal of this project is to create a system capable of carrying out visually guided tasks on an AUV. In current project, target tracking means estimating the fish state through the frames of image sequence produced by video and sonar. Before this goal can be achieved, the core tasks, image processing algorithms of both video and sonar for identifying the target fish must be implemented first. The vision system is then programmed to perform off-line fish tracking, which must address the issues of identify, location and orientation of the fish. Our proposal is verified through off-line data processing. Those data are collected by micro- ROV tracking the bass fish in the nature underwater environment under the manned control.

1.6.2 Visual servoing control system

Visual servoing control system Autonomous target tracking is commonly achieved (or partly achieved) by holding station close to the target object over some period of time. The navigation commands are determined based on the position of the AUV with respect to the position of the target. The control system to track a moving fish must be designed for “moving while looking”, which requires real-time recognition and motion control. In this system, a single camera with auxiliary sonar that plays the role of the manipulator’s eye is mounted in the AUV. When the manipulator is correctly tracking the target fish, it must always be seen in a vicinity of the image center position. Obviously, the system has to deal with the unpredictable position and orientation of the fish from one control loop time step to the next, and also, it will have to cope with the difficulty of handling possible variations of the fish’s shape.

Fig. 1.5 shows the structure of the visual serving control system used in this project. In this system block diagram, video stream taken from a CCD camera mounted on AUV are decomposed into a series of image frames, the intensity image $f^k(x, y)$ for the k^{th} frame of a video sequence is processed to extract the target’s position (u, v) corresponding to video frame coordinates. The sector scan sonar data are stored and produce the image frame sequences $I^k(x, y)$.Through image processing methods, the relative distance, d, and orientation,φ , between target and sonar can be obtained by reading the data of scanline. After that the state estimation algorithm combines these state data gained from both sensor to implement the AUV state estimation and input the system range ρ, bearing ψ, and depth Z in polar coordinates to controller which drive the AUV to hold the target in the center of the camera image and at some desired distance. Note that a vehicle equipped with a single camera cannot estimate the depth to targets.

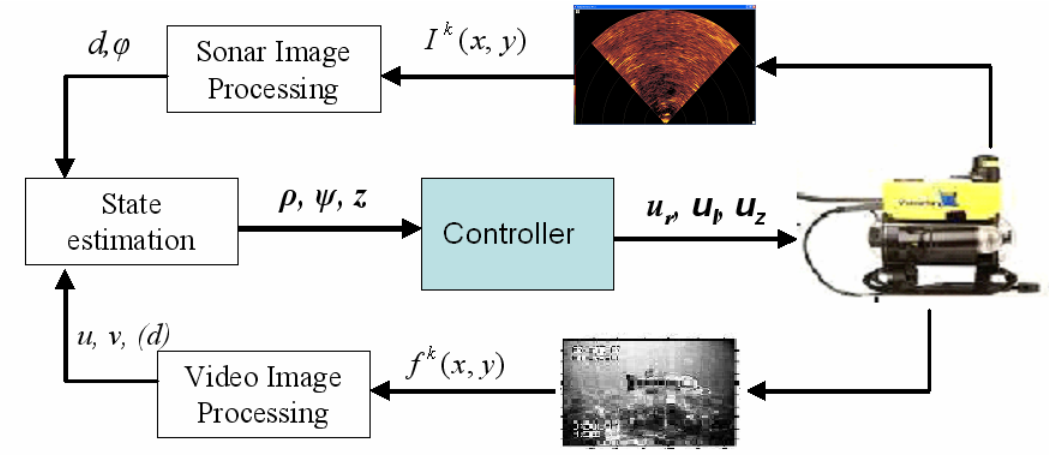


Figure 1.5: The visual serving control system

1.6.3 Experiment Platform

This thesis focuses on the development of a tracking system that employs both vision and sonar sensors for underwater fish tracking. To validate the proposed methods, real data was obtained using a VideoRay Pro III MicroROV deployed within Paradise Lake, Ontario, Canada. This data was then used in offline validation experiments.

The VideoRay Pro III is a system designed for intensive, underwater operations. It has an open architecture that accommodates a wide variety of tools and sensors. The VideoRay Pro III system consists of a control console, a submersible robot and a tether deploying mechanism. The control console has a video display, joystick controls for horizontal and vertical movement, and a computer control interface. The submersible robot has two horizontal thrusters and one vertical thruster for its motion control, a pressure sensor for measuring depth, and a compass for measuring orientation. The WDCC-6300 CCD camera is installed in the front of the pressure hull. This camera uses 1/3" color CCD format, low illumination, and electronic auto iris, 570 lines of resolution, internal synchronization, pinhole or standard lens, low power consumption. Images were of dimensions 240x320, and were grabbed at a frame rate 30Hz. The imaging sonar, SeaSprite DST Micron sonar, was mounted vertically on the front top of the VideoRay. The sonar scan range was set to its minimum value of 5 m with a scan area of 90 degrees. The time to capture a complete sonar scan is 4 s. It also has an accessory connector allowing for field integration of various instruments and sensors, as shown in Fig. 1.6.

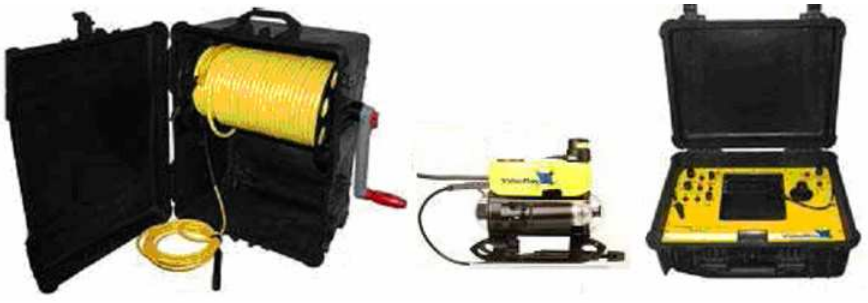


Figure 1.6: VideoRay Pro III system

1.7 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents the first of two image processing algorithms developed in this project to identify the target fish and extract its features. This algorithm uses a series of existing technology commonly found in the vision literature to pre-process the original fish images, before applying a Gabor filter to implement Texture segmentation and obtain obvious features. Finally, it uses a Projection Curve Segmentation Method to extract these features for identifying the target fish and estimating the relative position.

Chapter 3 describes the second image processing based approach to tracking. Using SIFT, the Scale Invariant Feature Transform, this approach has the ability to recognize fish from the cluttered underwater scene. SIFT was introduced by David Lowe in 1999 as a method for extracting distinctive invariant features to scaling, rotation or translation of the image. It is partially invariant to illumination changes and can be used to perform reliable matching between different images of the same object or scene.

Chapter 4 introduces the imaging sonar and a data processing method that can be used to identify basic objects. With this sensing system, both range and bearing to the fish can be estimated.

In Chapter 5, an approach to obtaining the relative position and orientation between fish and AUV is presented, followed by off-line results.

In Chapter 6, the conclusions and future work are summarized.

Chapter 2

Fish Feature Extraction Algorithm

2.1 Introduction

In many cases, including the fish-tracking application, a computer visual algorithm is demanded by visual servoing system for tracking targets. Visual tracking algorithms are developed to drive AUV in order to hold the moving objects in the centre of 2D image frames in all of video sequences, without any implication of closed-loop motion control of the imaging platform. Computer vision algorithms distinctly address two related imaging problems: segmentation and recognition. The segmentation subdivides a digital image into its constituent regions or objects (set of pixels) that are similar according to a set of predefined criteria or partitions an image based on abrupt changes in intensity, such as edges in an image. Segmentation simplifies and changes the representation of an image into something that is more meaningful and easier to analyze. These regions or boundary correspond to the tracked objects. The recognition is a process of representing the region in terms of its characteristics, describing the representation by extracted features, and identifying the best match to target by statistical (or decision theoretic), or syntactic (or structural) methods.

Vision literature suggests many segmentation techniques, but few perform well for fish tracking. There are some foreseen difficulties in the vision processing of underwater fish images. Since the contrast of an underwater image is fairly low, it makes image quality very poor. Image contrast is affected by depth (Light attenuates exponentially with distance in water), sediment in the water, and light diffraction. Illumination backscatters to the camera, producing a relatively bright and non-uniform background image. Suspended organic particles, known as marine snow, introduce continual small fluctuations to this background image. Also, the fish do not appear as exclusive bright against dark backgrounds [30]. The vast array of unknown objects in the environment can be misinterpreted of object of interest. Furthermore, finding gradients is also difficult with fish. Due to the difference of the light reflection ratio of fish scales, the intensity is uneven and the gradient distributions are scattered on the entire body, with some areas of strong intensity and others of weak intensity. Moreover, hotspots on the camera enclosure produce a strong gradient response. Lighting geometries that can result from these bright reflections are difficult to predict in advance.

There are many basic and efficient segmentation methods in the image processing library that fail in their application to fish images taken from the natural underwater environment. For example, Color segmentation has success in extracting the fish from the water background, but encounters difficulty in separating the fish from seaweed and the floor. Background Subtraction methods [30] based on a largely stable background image differences cause moving objects to stand out saliently in sequential images. In this case, this approach works poorly because the background typically changes over time when the ROV is moving, when the fish remains moderately still with respect to the ROV, or in the presence of currents. The active contour method [67] and snake method [61] fails in the various seaweeds and the very uneven intensity on the fish body. Intensity threshold routines, even adaptive ones, proved unreliable. Gradients segmentation [60] fails into the background image creating overlap between target and background intensity values. In these cases, no unique threshold level exists. Region-merging methods [63] also encounter difficulties that result from the similar seaweed and fish body. Expansive regions belonging to the background are often misclassified as target regions, and vice versa. Nor did watershed methods [27] give reliable results. When applied to the gradient image using bright intensity patches to form initial markers, different intensity gradients on the surface of the fish body created multiple watersheds for the same target. Attempts to merge these watersheds encountered difficulties similar to those observed for other region-merging methods

Since the extracting the target fish (Large Mouth Bass) from the natural underwater environments background is complex and hard, it is impossible to use one general technology from a vision library. Fish identification requires an efficient image processing algorithm in which several computer vision methods and a new approach called projection curve recognition are incorporated. The technique is simplest and successful in identifing the fish through its tail and body features in the typical images taken from a real underwater environment. These features are extracted to estimate the relative position between the target fish and the AUV. Fig. 2.1 shows the overview of this algorithm.

The implementation relies on three assumptions. First of all, it is assumed that only one fish exist in the frame, more fishes will invalidate the results. Secondly, the fish side faces the camera and the tail can be seen. When the fish swims directly toward the lens, it is rather difficult to find out the target. And lastly, the fish doses not swim among the seaweeds and floor, it should keep the little higher than the underwater plants. These assumptions will permit the texture and projection segmentation to produce a correct result.

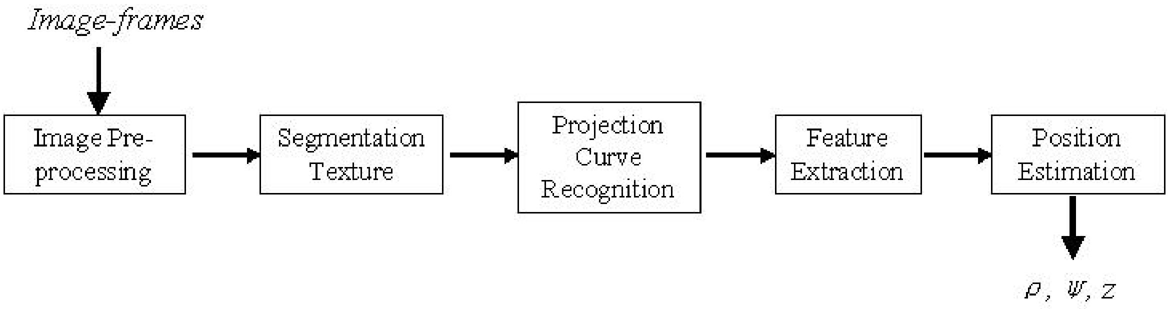


Figure 2.1: Feature Extraction Algorithm diagram

2.2 Image Pre-Processing

2.2.1 Image Conversion

The raw image of fish taken by the digital camera from a nature underwater environment is a color image. Although Color is a rich and complex experience, and the color of object seems to be useful cue in identifying them, it is currently difficult to segment fish from background The use of color in computer vision is surprisingly primitive. One difficult is some legitimate uncertainly about what it is good for. Especially, in fish tracking, the fish is nice match for background; thus its color is close to the color of seaweeds, sands or mud in underwater bottom. Furthermore, a color image is an $M \times N \times 3$ array of color pixels, where each color pixel is triplet corresponding to the red, green, and blue components of an RGB image at a specific spatial location. A 3-D matrix can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count, however, it causes exceedingly computationally expensive. A greyscale (or greylevel) image is simply one which the only colors are shades of grey. It is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image [30]. Since the color segmentation is unsuccessful to fish image and greyscale images are entirely sufficient for our project, in order to reduce the quantity and complexity of computation, the input original color image is converted to greyscale image is given as below, and the image result is shown in Fig. 2.2.

$$I(x,y) = T \cdot c(x,y) = \begin{bmatrix} 0.3 & 0.59 & 0.11 \end{bmatrix} \times \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} \tag{2.1}$$

$I(x,y)$- luminance function at spatial coordinates (x,y);

T- transformation factor;

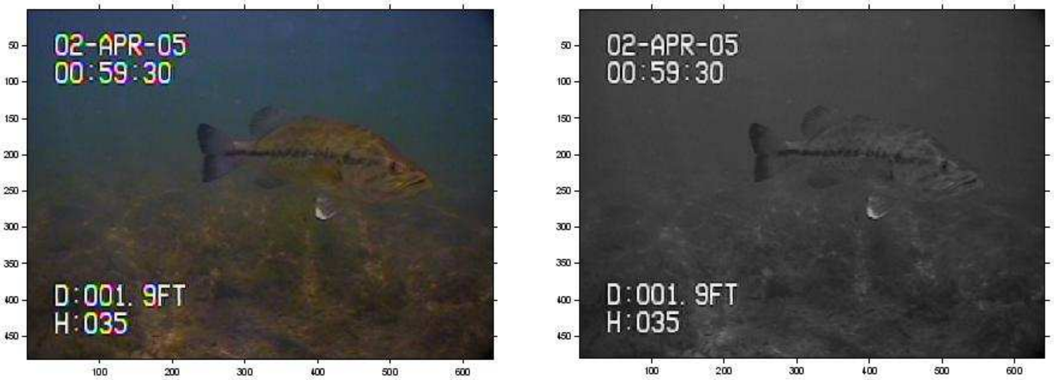


Figure 2.2: The fish color image and the converted greyscale image

$c(x, y)$-an arbitrary vector as the color components function at(x, y)in RGB color space.

At present, Greyscale images are very common, such that many image processing methods in vision library are greatly suitable for the greyscale image. Generally, the greyscale intensity is stored as an 8-bit integer giving 256 possible different shades of grey from black to white. To be convenient of image processing, in our work, the 256 8-bit integers are converted into double float numbers, and the values are limited in the interval $[0, 1]$.

2.2.2 Image Enhancement

Usually, the fish gray-scale image is underexposed and blurry, due to underwater light limitations, seeing an example image in Fig. 2.3 (a). The image is extremely dark; it lacks detail since the range of colors seems limited to low grey-levels. We can verify this by looking at the image's histogram, Fig. 2.3 (b).

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. The operation is very simple as follow [31]

$$h(r_k) = n_k \tag{2.2}$$

where, r_k is the kth intensity level in the interval $[0, G]$ and n_k is the number of pixels in the image whose intensity level is r_k. In our work, $G = 1$, the interval is [0, 1]. The image is scanned in a single pass and a running count of the number of pixels found at each intensity value is kept. The fish image intensity histogram, Fig. 2.3 (b), forms a tight, narrow peak cluster in the lower greylevel region between the greylevel intensity values of 0.1 to 0.3, which means the whole image is represented almost entirely by dark pixels. It is harder to enhance.

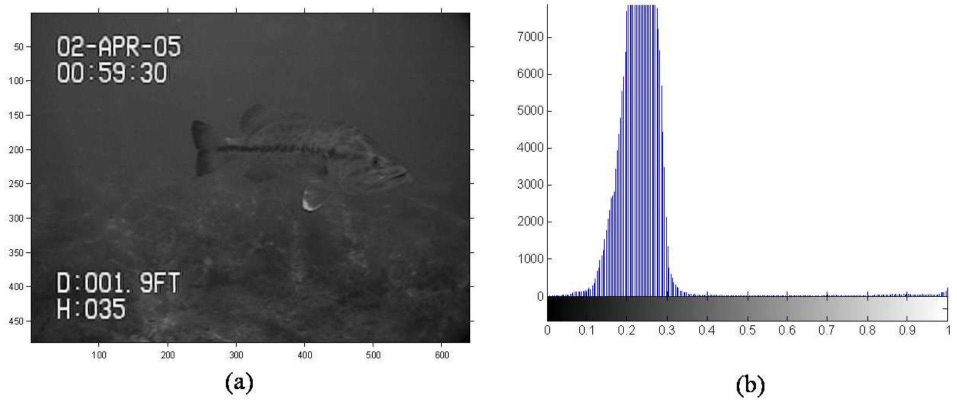


Figure 2.3: The gray-scale image and its related histogram

Enhancing the image can remarkably improve detail of an image; therefore it can provide a big assistance for other machine vision operations such as segmentation. For given data, the fish image is more difficult to enhance, so we respectively apply two basic and popular methods in image processing of contrast adjustment using the image's histogram: the first one is Contrast stretching and the other is called histogram equalization.Theses two methods is useful in images with backgrounds and foregrounds that are both bright or both dark, and lead to better detail in images that are over or under-exposed.

Contrast stretching

Contrast stretching is a basic IPT tool for intensity transformations of greyscale images that attempts to improve the contrast in an image by applying a function to stretch the range of intensity values input image contains to span a desired range of values Simply said, the function maps the intensity values in input image f to new values in g, such between low-in and high-in maps to value between low-out and high-out [31]. Here, we select a linear scaling function that maps the intensity values in f to create g :

$$g(x,y) = \frac{d-c}{b-a} \times (f-a) + c \tag{2.3}$$

where, c and d called the lower and the upper limit irrespectively, a and b is the lowest and highest pixel values currently present in the image.

In our fish greyscale image, to avoid unrepresentative scaling that caused by some outlying pixels with either a very high or very low value, the value of a and b are selected at 5th and 95th percentile in histogram. From the histogram shown above Fig. 2.3, the values of a and b are 0.1 and 0.3, and the values of c and d are 0 and 1 respectively. The Fig. 2.4 shows the result of this enhancement. Notice that, though the sum of the pixels is not changed, the values between different pixels are increase and most histogram pixels have been moved into the central greylevel region. The contrast has been significantly improved.

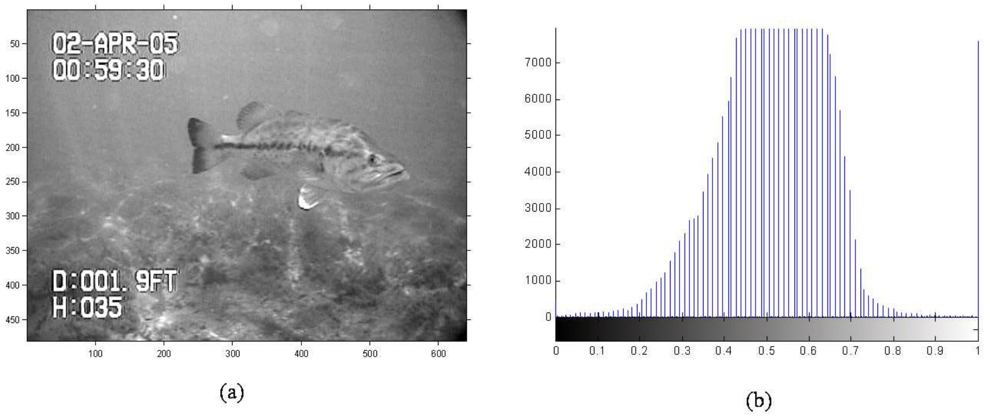


Figure 2.4: The image and its histogram after Contrast stretching

Histogram equalization

While the quality of the current image is much better than the original, the enhanced image itself still appears somewhat flat, and the histogram confirms what we can see by visual inspection: this image has poor dynamic rangeand it has not been stretched across the entire spectrum yet. Therefore, this image is not satisfied as the final idea one that will be used into image segmentation.

When the usable data of the image is represented by close contrast values, as the Fig. 2.4, we consider applying histogram equalization, another method in image processing of contrast adjustment using the image's histogram, to expand the grey-levels within the image to fill the entire spectrum. Though this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast without affecting the global contrast. If the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. The calculation is not computationally intensive. [31]

Assume that intensity levels are continuous quantities normalized to range [0,1], and let $P_r(r)$ denote the probability density function of the intensity levels in a given image. Suppose that we perform the following transformation on the input levels to obtain output intensity levels, s, [31]

$$s = T(r) = \int_0^r P_r(w)dw, 0 \leq T(r) \leq 1 \text{ for } 0 \leq r \leq 1. \tag{2.4}$$

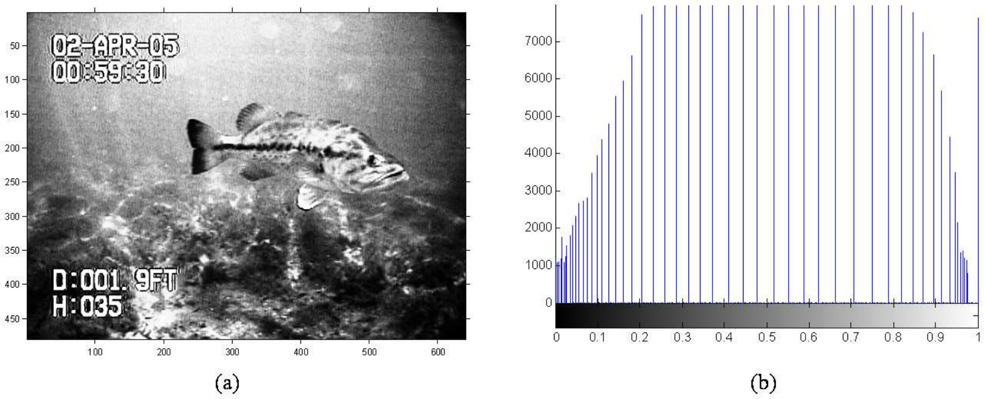
where w is a dummy variable of integration.

In general, the histogram of the processed image will not be uniform, due to the discrete nature of the variables. For discrete quantities we work with summations, and the equalization transformation becomes [31]:

$$s_k = T(r_k) = \sum_{i=0}^{k} \frac{n_i}{N} = \sum_{i=0}^{k} \; P_r(r_i) \tag{2.5}$$

for $k = 1, 2, , L$, where is the intensity value in the output image corresponding to value r_k in the input image.

The final result is shown in Fig. 2.5.Look how much clearer the image seems, details in the fish and the floors and seaweeds are a lot sharper. The equalized histogram has been "stretched" across the entire spectrum.



(a)

(b)

Figure 2.5: The image after adjusting contrast with its histogram

2.3 Texture Segmentation by Gabor Filter

Texture is a phenomenon that is widespread, easy to recognise and hard to define. Typically, whether an effect is referred to as texture or not depends on the scale at which it is viewed. Theoretically textures are visual patterns or spatial arrangements of pixels that cannot be completely described using regional intensity alone, they may have statistical properties, structural properties, or both. Texture segmentation is the problem of subdividing an image into differently regions where the texture is constant. This suggests representing image texture in terms of the response of a collection of filters. But what filters we should use are no canonical answer.

The Gabor Filters have received considerable attention in texture segmentation problems for computer vision and image processing since it was proposed by Gabor in 1980. The family 2-D Gabor filters was originally presented by Daugman as a framework for understanding the orientation-selective and spatial-frequency selective receptive field properties of neurons in the brains visual cortex, and then was further mathematically elaborated.

In this case, the fish has own notable texture different from background, especially, the target fish tail and body's central patterns consists of quite regular and obvious orientation stripes. To extract these features, a single oriented Gabor filter of both spatial and frequency domain spatial-frequency is applied. Although it is only single filter, Gabor filter is effective and efficient in texture segmentation: fish's tail and body central pattern are primly represented and most background can be subtracted. There does not seem to be much benefit in using more and complicated sets of filters than the Gabor filter, and yet using more filters leads to very expensive for convolving the image.

Generally, Gabor filters are defined by harmonic functions modulated by a Gaussian distribution. Where $s(x, y)$ is a complex sinusoidal, known as the carrier, and $w(x, y)$ is a 2-D Gaussian- shaped function, known as envelop [16]

$$\begin{aligned} g(x,y) &= s(x,y) \cdot w(x,y) \\ &= Kexp(-\pi(a^2(x-x_0)_r^2 + b^2(y-y_0)_r^2)) \\ &\quad exp(j(2\pi(u_0 x + v_0 y) + P)) \end{aligned} \tag{2.6}$$

Where (u_0, v_0) and P define the spatial frequency and the phase of the sinusoidal in Cartesian coordinates respectively. (x_0, y_0) is the peak of the Gaussian function, a and b are scaling parameters of the Gaussian, and the r subscript stands for a rotation operation. Each complex Gabor consists of two functions in quadrature (out of phase by 90 degrees), conveniently located in the real and imaginary parts of a complex function.

The 2-D Fourier transform of this Gabor is as follows:

$$\begin{aligned} \hat{g}(x,y) &= \frac{k}{ab} exp(j(-2\pi(x(u-u_0) + y(v-v_0)) + P) \\ &\quad exp(-\pi(\frac{(u-u_0)_r^2}{a^2} + \frac{(v-v_0)_r^2}{b^2})) \end{aligned} \tag{2.7}$$

The complex Gabor function is defined by the following nine parameters: [16], and shown in fig. refgaborkernel.

- K : Scales the magnitude of the Gaussian envelop
- (a, b) : Scale the two axis of the Gaussian envelop

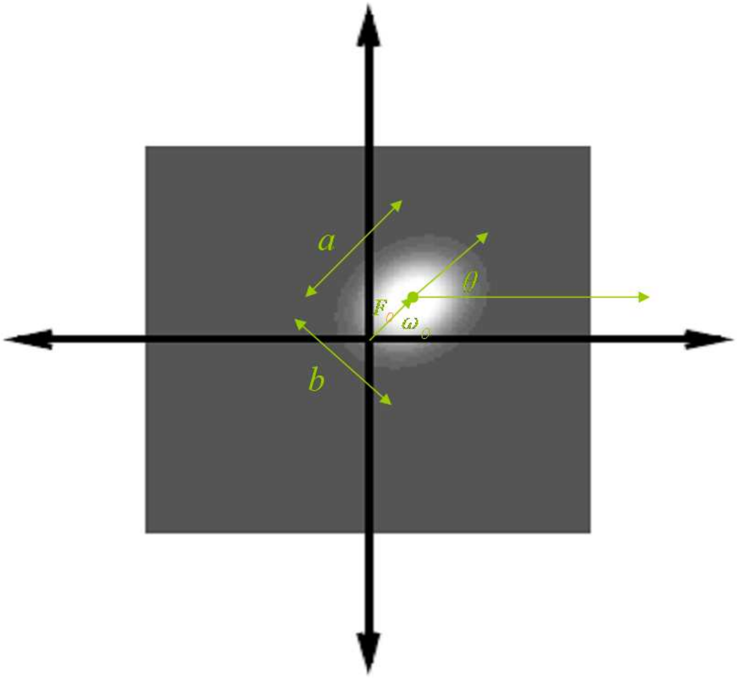


Figure 2.6: An example of Gabor kernel and reflected parameters

- θ : Rotation angle of the Gaussian envelop
- (x_0, y_0) : Location of the peak of the Gaussian envelop
- (u_0, v_0) : Spatial frequency of the sinusoidal carrier in Cartesian coordinate. It can also be expressed in polar coordinates as
- P : Phase of the sinusoidal carrier.

The simple Gabor function

Rewrite equation (2.7) with a spread of σ_x and σ_y in the x and y directions:

$$\begin{aligned} G(x,y) &= exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \\ &\quad exp\left(-2\pi \cdot i(u_0(x-x_0) + v_0(y-y_0))\right) \end{aligned} \tag{2.8}$$

$$G_{symmetric}(x,y) = cos(k_x x + k_y y) \cdot exp\left(\frac{(x-x_0}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \tag{2.9}$$

Derived from equation(2.8) by elaborately selecting above parameters, the even-symmetric real component of the original 2-D Gabor filer can be obtained. [12]:

$$h(x,y,T,\phi) = exp\left(-\frac{1}{2}\left[\frac{x_\phi^2}{\sigma_x^2}+\frac{y_\phi^2}{\sigma_y^2}\right]\right) \tag{2.10}$$

$$\begin{aligned} x_\phi &= xcos\phi + ysin\phi \\ y_\phi &= -xsin\phi + ycos\phi \end{aligned} \tag{2.11}$$

Where, $\phi$is the orientation of the derived Gabor filter, and T is the period of the sinusoidal plane wave. If we decompose equation (2.10) into two orthogonal parts, one parallel and the other perpendicular to the orientation ϕ, the following formula can be deduced:

$$\begin{aligned} h(x,y,T,\phi) &= h_x(x,T,\phi)\cdot h_y(y,\phi) \\ &= \left\{exp\left(-\frac{x_\phi^2}{2\sigma_x^2}\right)\cdot\cos\left(\frac{2\pi x_\phi}{T}\right)\right\}\cdot\left\{exp\left(\frac{y_\phi^2}{\sigma_y^2}\right)\right\} \end{aligned} \tag{2.12}$$

The first part h_x behaves as a 1-D Gabor function which is a band pass filter, and the second one h_y represents a Gaussian function which is a low pass filter. Therefore, a 2-D even-symmetric Gabor filter performs a low pass filtering along the orientation ϕ and a band pass filtering orthogonal to its orientation ϕ. It should be pointed out that h_x in equation (15) could be treated as a non-admissible mother wavelet (indicated by its Fourier representation $\hat{h_x}(0) \neq 0$). Its band pass property is related with the σ_x . If σ_x is too small, the band pass filter degenerates into a low pass function (indicated by its Fourier representation $\hat{h_x}(0) \gg 0$). On the other hand, if σ_x is appropriately large, h_x can be approximately regarded as an admissible mother wavelet (indicated by its Fourier representation $\hat{h_x}(0) \approx 0$) with good band pass property (fig.2.7).

The image with Gabor filter

Use such Gabor filter as kernel to convolute with the input the image. Fig.2.8 shows the different results by Gabor filter according to different parameters for a image. When Gabor filter parameters were selected as: $\sigma_x = 1$, $\sigma_y = 4$, $T = 1/8$, $\phi = \pi/3$, the effect is best satisfied. Obvious fish feature are remained and most background including the hotspots marine snow and some grass are successfully subtracted. Fig.2.9 shows the results of various image frames by Gabor filter. Thus the Gabor filter establishes a good basis for the following projection curve segmentation.

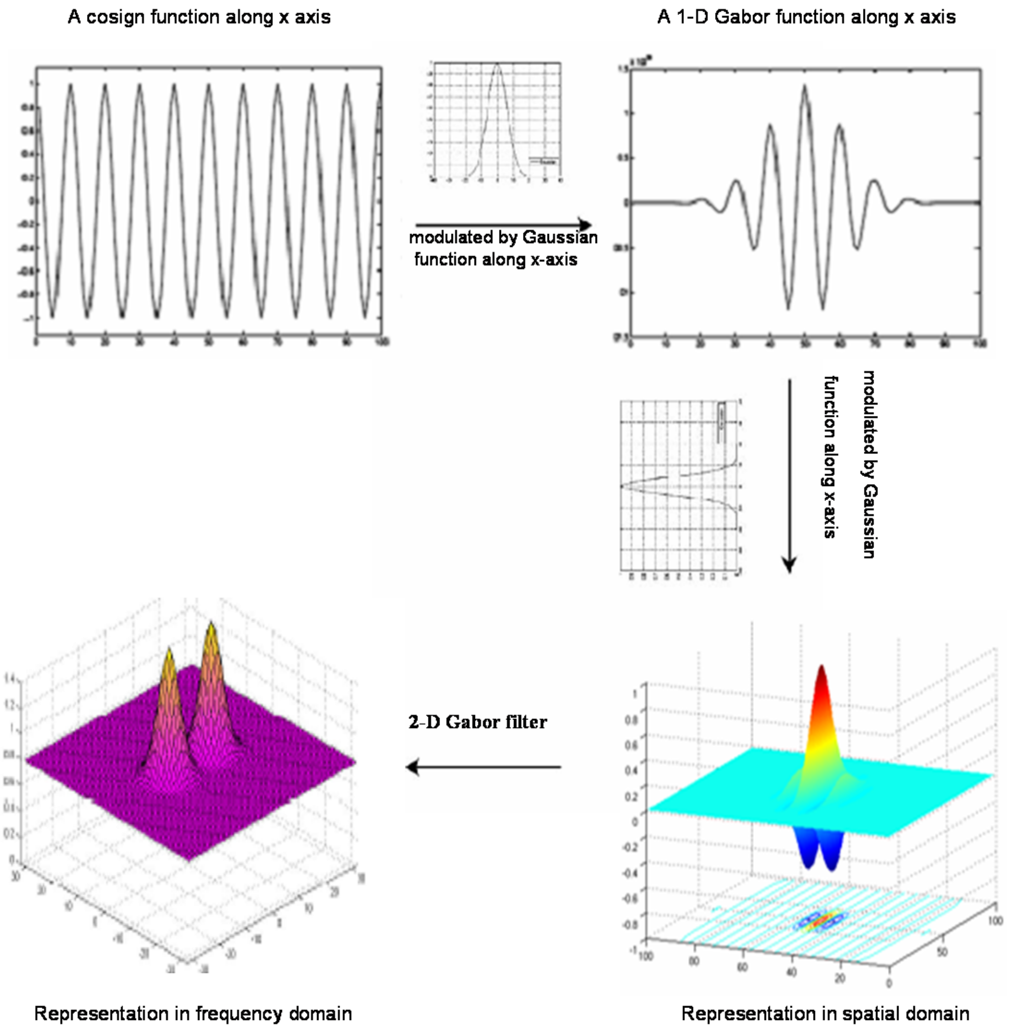
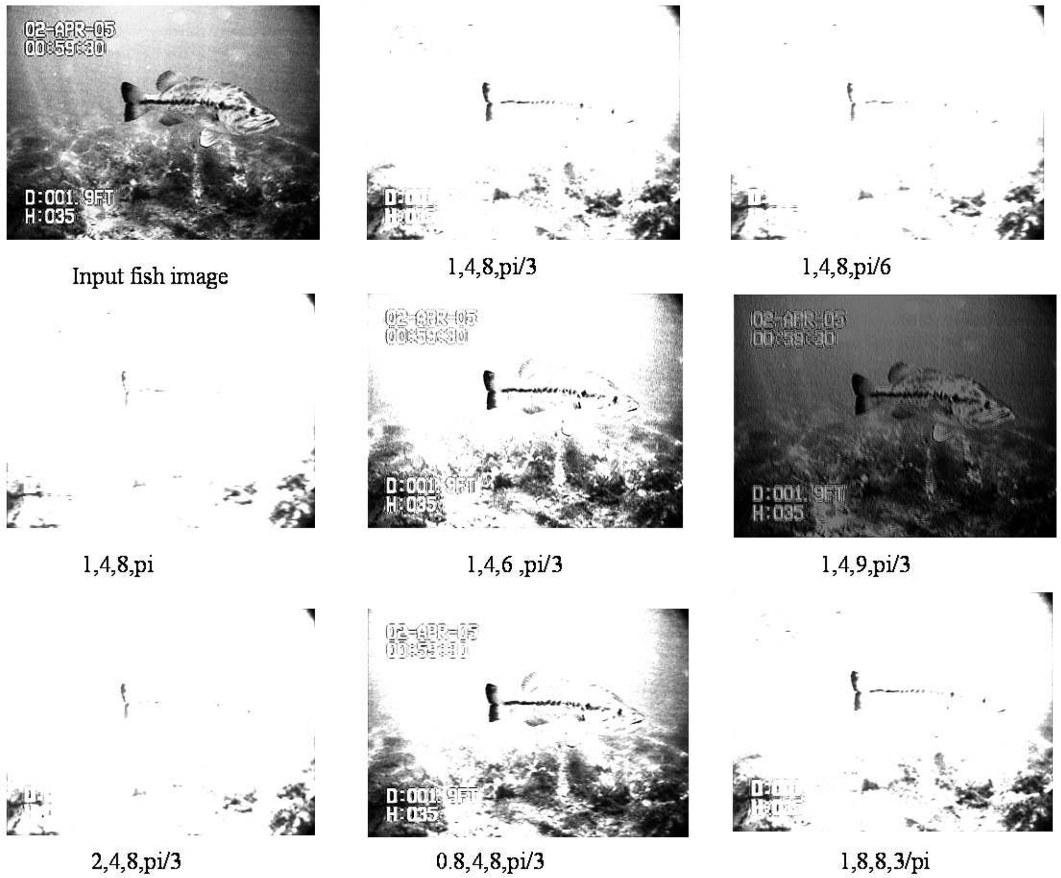


Figure 2.7: The Gabor filter and its response represented in spatial and frequency domain



Input fish image

1,4,8,pi/3

1,4,8,pi/6

1,4,8,pi

1,4,6 ,pi/3

1,4,9,pi/3

2,4,8,pi/3

0.8,4,8,pi/3

1,8,8,3/pi

Figure 2.8: The result of Gabor filter using different parameters

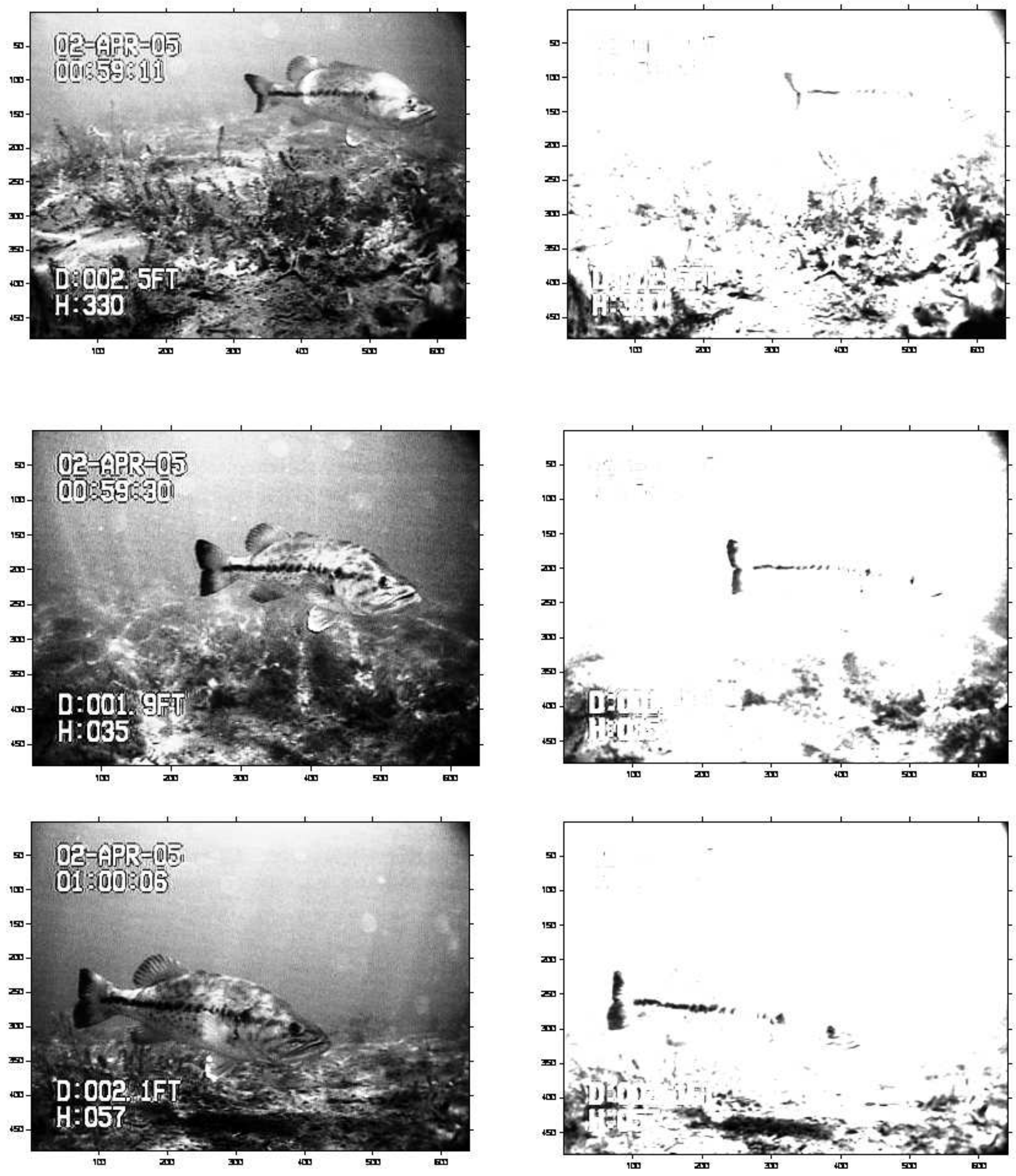


Figure 2.9: Gabor filter results with different images

2.4 Projection Curve Based Recognition

After the image is processed by the Gabor Filter, a threshold is applied to force pixels to take on values of 0 or 1, so that a binary image is obtained.

$$I(x,y) = \begin{cases} 1 & \text{if } I(x,y) > \text{Threshold}, \\ 0 & \text{if } I(x,y) \leq \text{Threshold}. \end{cases}$$

Analysis of binary images is one of the most simple and efficient ways to extract information from pictures. It is particularly useful when information needs to be acquired about an object's position and orientation within an image. In our project, this threshold is 0.8, the result image is shown in Fig.2.10 (a).

In observing the binary image, only the fish tail pattern, body center pattern, and some background patterns (i.e. underwater grass) remain. The fish patterns have limited overlap with the background. Projecting the threshold image into a vertical histogram Hv(y), i.e. Projecting the threshold image into a vertical histogram $H_v(y)$, i.e. summing the number of black pixels in each row of the image, results in two separate shapes. The first is the background curve with no defining shape. The second is a sharp and narrow spike protruding from a smooth and low curve. This second shape is a projection of the tail and body features, (Fig.2.10 (b)). With this histogram, a search for the tail and body patterns is conducted to produce an interval of rows in which the fish is located. If A is a predetermined threshold that characterizes the tail width, the tail interval is defined as rows belonging to $[y_{tailstart}, y_{tailstop}]$ such that a scan from the top of the image produces:

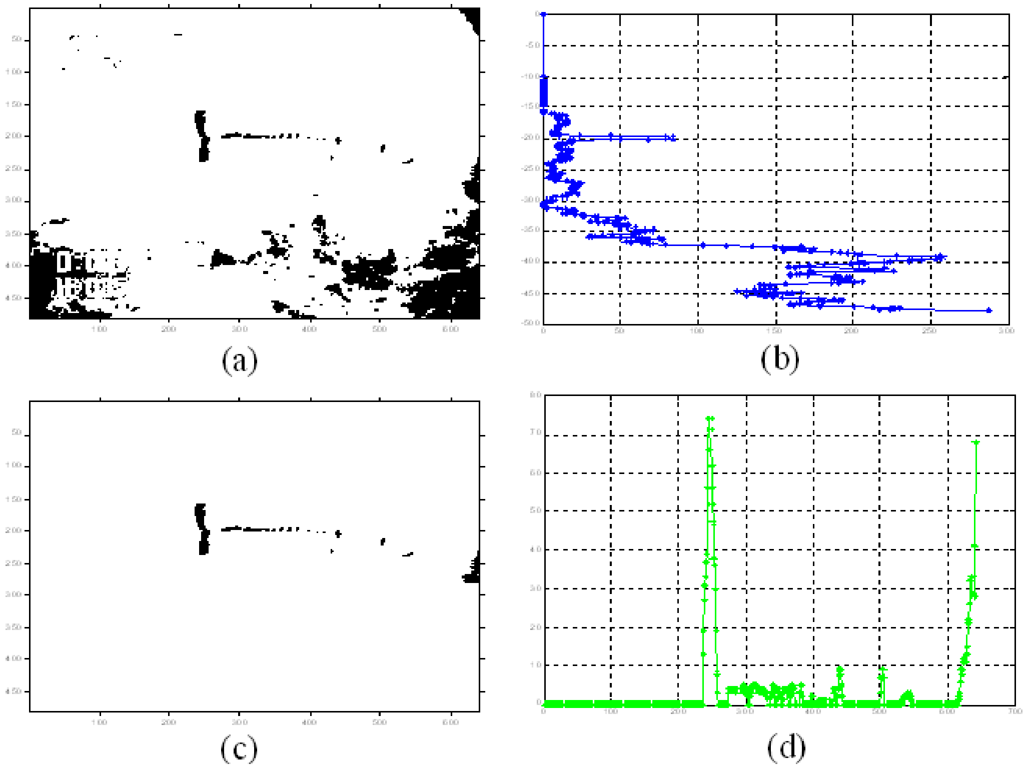
$$\begin{aligned} y_{tailstart} &= max(y|H_v(y) > A) \\ y_{tailstop} &= max(y|H_v(y) < A, y < y_{tailstart}) \end{aligned} \tag{2.13}$$

The peak within this interval is determined by:

$$y_{max} = max(y|y \in [y_{tailstart}, y_{tailstop}]) \tag{2.14}$$

If the slope of the histogram within intervals $[y_{max} - \delta, y_{max}]$ and $[y_{max}, y_{max} + \delta]$ have magnitudes less than m_{min}, it is determined that the fish tail feature is found. If the slope conditions are satisfied, rows outside the interval $[y_{tailstart}, y_{tailstop}]$ are subtracted from the image, effectively eliminating background in the top and bottom portions of the image, (see Fig. 2.10 (c)).

In a similar fashion, the image is projected into a horizontal histogram $H_h(x)$, i.e. summing the number of black pixels in each column of the image. The tail pattern dominates the histogram with an obvious spike. The body pattern is also evident as a region of constant amplitude adjacent to the tail spike. In this case, a



(a)

(b)

(c)

(d)

Figure 2.10: Projection Curve Segmentation: (a) Image after Gabor filter and threshold, (b) the vertical projection curve, (c) the image after subtracting top and bottom background, and (d) the horizontal projection curve.

search for these two features is conducted to define an interval of columns in which the fish resides. Columns outside this interval are subtracted to remove background on the two sides of the fish, (see Fig. 2.10(d)). What remains is an image with only the tail and body features.

2.5 Feature Extraction

After an image has been segmented into regions by above methods, the next step is to represent and describe the segmented aggregate. We can represent the region in terms of its shape or properties characteristics such as boundary, color, texture and so on, and then describe the representation by feature that should be insensitive as possible to variation in region size, translation, and rotation, such as its length and the number of concavities it contains. Seen from segmentation results shown in section 2.3, we should be capable of describing the fish tail and body as two straight lines that will be profitable to future state estimation by outline extracting and the least squares linear regression line fitting method.

2.5.1 Edge Detection

Edge detection is a most commonly used operations in image analysis to catch information from the frames as a precursor step to feature extraction or object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image. Among many popular edge detector, such as Sobel Detector, Prewitt Detector and Laplacian of a Gaussian (LoG) Detector, the Canny detector that was introduce by Canny in 1986 [19] is the most powerful edge detector. Canny detector finds edges by looking for local maxima of the gradient of image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges [5].

Canny edge detector algorithm can be summarized as follow:

1. The image is smoothed by a Gaussian kernel with a specified standard deviation, σ,

$$I_c = I \otimes G \tag{2.15}$$

2. Find the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal directions.

$$\nabla I_{c-x} = I_{c-x} \otimes \frac{\partial G}{\partial x} \tag{2.16}$$

$$\nabla I_{c-y} = I_{c-y} \otimes \frac{\partial G}{\partial x} \tag{2.17}$$

The local gradient at each point:

$$\nabla I_c(x,y) = \sqrt{\nabla I_{c-x}^2(x,y) + \nabla I_{c-y}^2(x,y)} \tag{2.18}$$

3. compute the direction of the edge at each point,

   $$\angle I_c(x,y) = tan^{-1}\left[\nabla I_{c-x}(x,y)/\nabla I_{c-y}(x,y)\right] \tag{2.19}$$

   An edge point is defined to be a point whose strength is locally maximum in the direction of the gradient.

4. The edge points determined in (2) give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in output, a process known as nonmaximal suppression.

5. The ridge pixels are then threshold using two thresholds, T_{low} and T_{high}, with $T_{low} < T_{high}$. Ridge pixels with values great than T2 are said to be strong edge pixels. Ridge pixels with values between T_{low} and T_{high}, are said to be weak edge pixels.

6. Finally, performs edge linking by incorporating the weak pixels that are 8-connected to the strong pixels.

Since the projection curve segmentation method have obtain the target fish region in the image frames, finding out the edges is limited a smaller area, compared to searching in whole original image, which will obviously reduce the computation. Canny detector produces clean edge maps that build the complete fish tail and fish body by extracting the principal edge feature of the input image. Some isolated point can be subtracted, which will be benefit to feature extraction. The image result with Canny edge detection is shown in Fig. 2.11(a).

2.5.2 Line Extraction

The Canny detector yields pixels lying only on edges. Despite of complete outline in most time, in some practice, the resulting pixels may not characterize an edge completely because of noise, breaks in the edge from nonuniform illumination, and other effects that introduce spurious intensity discontinuities. In order to make facile for calculating relative position and orientation of fish in the image plane in future, we consider using two lines respectively through the centre of fish tail and body to represent the fish. For the sake of reducing the effect of uncompleted outline, we applied a simple algorithm to fit the straight line.

1. Use all central points within outline instead of boundary points in tail and body interval decided by projection curve segmentation. It is very easy.

- find each leftmost point, $h_l(i)$, and rightmost point, $h_r(i)$ along each horizontal line, i, of image in tail interval . If lack of either left point or right point, this isolate point is rejected. In body interval, the same method, searching along the vertical lines of an image, obtain the $v_l(j)$ and $v_r(j)$ at the j^{th} line.

- The all central points are denoted respectively as below:

$$\begin{aligned} h_o(i) &= (h_l(i) + h_r(i))/2 \\ v_o(j) &= (v_l(j) + v_r(j))/2 \end{aligned} \tag{2.20}$$

Thus, the fish tail and body are replaced as these discrete central points.

2. Apply least squares linear regression to fit a straight line. The straight line fitting basically uses an equation $f(x) = a + bx$ which is a line graph and describes the trend of the discrete central points set $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. The n should be greater or equal to $2(n2)$in order to find the unknowns a and b.

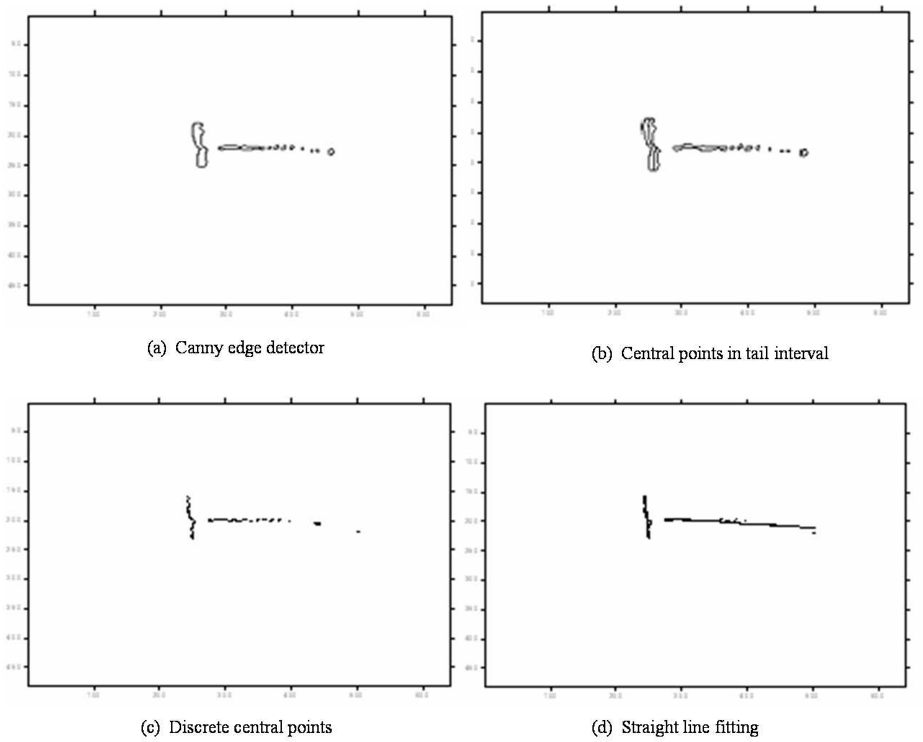
$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}^2 \times \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_i \end{bmatrix} \tag{2.21}$$

2.6 Results and Analysis

Video data images of a Large Mouth Bass were acquired using the WDCC-6300 CCD color camera installed on human driven VideoRay ROV. Images were of dimensions 480x640, and were grabbed at a frame rate 20Hz.

The implementation relies on three assumptions. First of all, it is assumed that only one fish exist in the frame, more fishes will invalidate the results. Secondly, the fish side faces the camera and the tail can be seen. When the fish swims directly toward the lens, it is rather difficult to find out the target. And lastly, the fish doses not swim among the seaweeds and floor, it should keep the little higher than the underwater plants. These assumptions will permit the texture and projection segmentation to produce a correct and perfect result.

The image processing algorithm was applied to each frame of each sequence. The series of filters including texture, projection curve recognition, and geometrical shape feature extraction proved simple, efficient and effective if several conditions



(a) Canny edge detector

(b) Central points in tail interval

(c) Discrete central points

(d) Straight line fitting

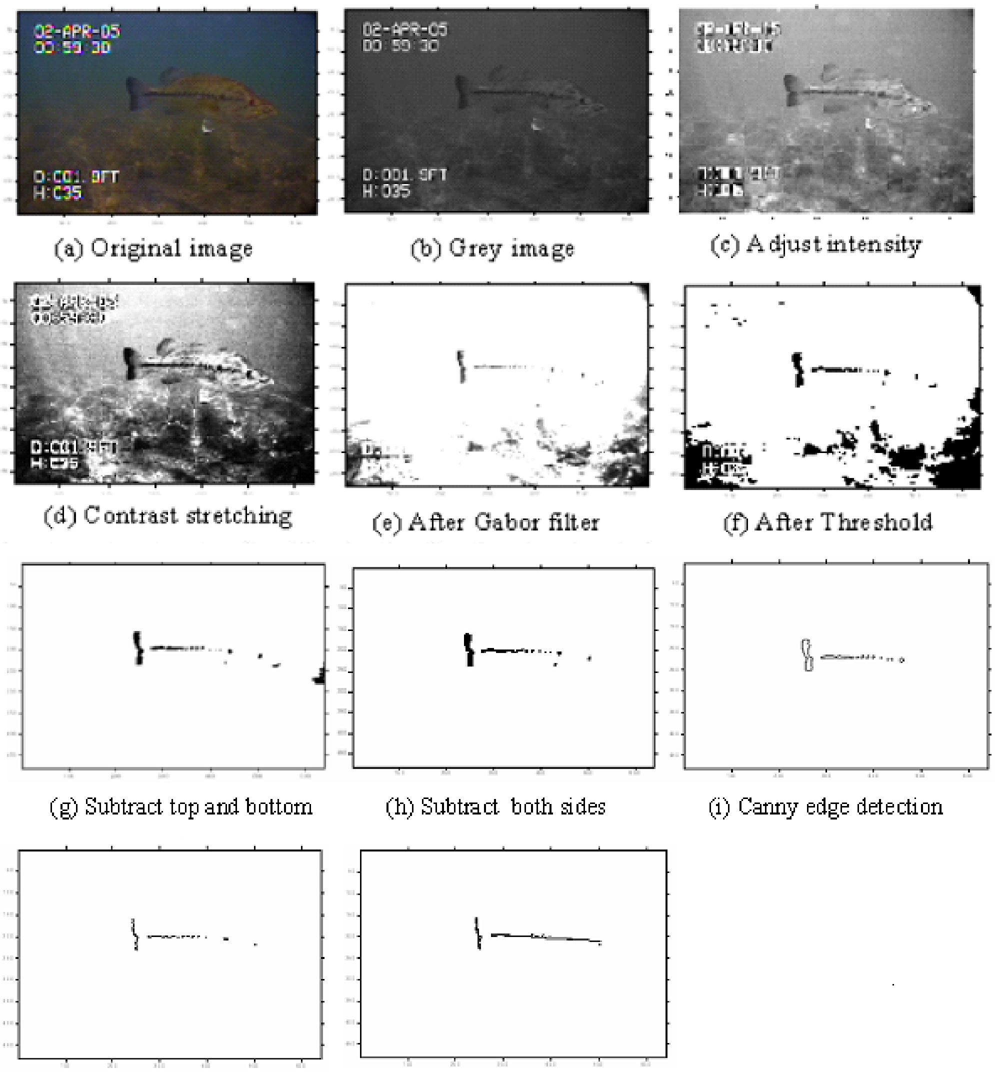
Figure 2.11: The results of Canny detector and feature extraction.

were met. There are three assumptions: first of all, it is assumed that only one fish exist in the frame, more fishes will invalidate the results. Secondly, the fish side faces the camera and the tail can be seen. When the fish swims directly toward the lens, it is rather difficult to find out the target. And lastly, the fish doses not swim among the seaweeds and floor, it should keep the little higher than the underwater plants. An example of a typical image being processed is shown in Fig. 2.12.

In Fig. 2.13, the fish motion is represented by the geometrical feature in the 9 sequential images taken from the ROV. The position of tail line can be calculated the relative position between fish and AUV. The slope of body line can be used to predict the possible orientation of the fish in next interval. The results indicate that the image segmentation, recognition and feature extraction method provide sufficient relative pose estimates for fish tracking.

2.7 Summary

In this chapter, a feature recognition and extraction algorithm is developed, which has ability to identify the Large Mouth Bass from underwater background, and to represent the target fish using the simple lines such that is convenient to estimate the relative position. This algorithm uses the Gabor filter to implement texture segmentation, and then applies a new method called projection curve recognition to identify fish feature from background. Finally, it use Canny edge detection and



(a) Original image

(b) Grey image

(c) Adjust intensity

(d) Contrast stretching

(e) After Gabor filter

(f) After Threshold

(g) Subtract top and bottom

(h) Subtract both sides

(i) Canny edge detection

(j) Draw the central line

(k) Straight line fitting

Figure 2.12: Examples of the result of image processing

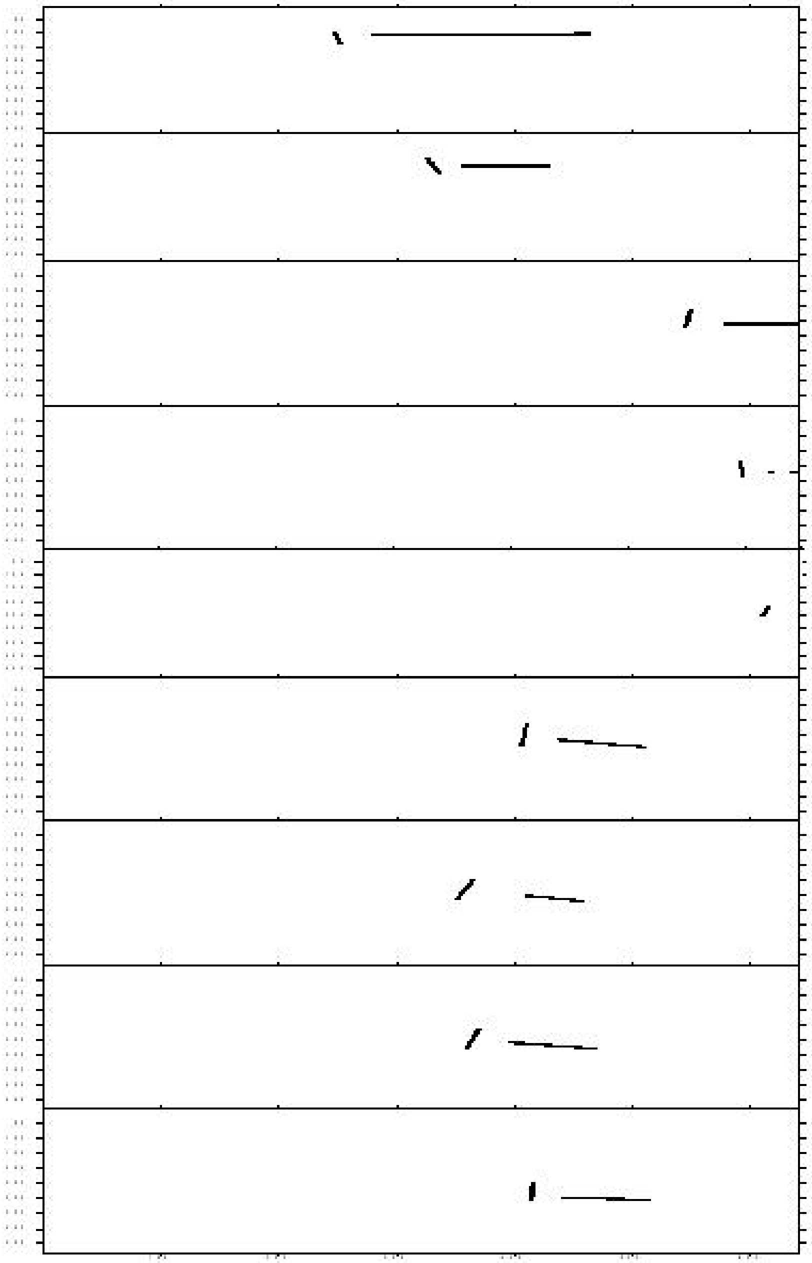


Figure 2.13: The extracted feature showing the motion of the fish in ten successive images taken by ROV in the lake in Waterloo, sampled at 0.2 second intervals.

linear regression to accomplish the fish representation. To validate the algorithm, off-line image processing was conducted on video footage obtained by piloting the ROV around Paradise Lake, Ontario, Canada. Despite the success in tracking fish over several images, this algorithm has several limitations and is valid under some conditions. First, it is assumed that only one fish can be present in each frame. Second, it is assumed that the fish swim perpendicular to the camera lens. Lastly, it is assumed that the fish cannot be occluded (e.g. by seaweed). In next chapter, the other recognition algorithm based on the SIFT approach will be able to solve most of these problems.

Chapter 3

SIFT Based Fish Recognition

3.1 Introduction

Segmenting and extracting a Large Mouth Bass from a natural underwater background is extremely difficult using current computer vision technology. As introduced in the previous chapter, although the Fish Feature Extraction Algorithm is valid and efficient under some ideal conditions, it will fail to identify the target fish in the more complex and disordered states, especially when the fish swims among seaweed, settles on the bottom, or encounters other animals in the neighborhood. All of these are common occurrences and cannot be easily avoided in a natural underwater environment. Therefore, other object recognition and tracking methods are investigated that do not use species-specific segmentation but instead find feature points on the fish and match them between frames during fish tracking. In such methods, there exist several points of interest on the object being tracked that can be extracted to provide a "feature" description of the object. In order to identify a specific fish from a natural underwater environment, local image features are required that are unaffected by nearby clutter or partial occlusion.

Object recognition, one of the most important tasks of computer vision, has found wide application in factory automation for parts inspection and identification as well as for manipulation. The main subject of these tasks is to deal with object recognition almost exclusively on correlation-based template matching in engineered environments where object pose and illumination are tightly controlled and the object is permitted to undergo translation, rotation and scaling. [71].

Recently, some approaches have been applied to extract features invariant to the image formation process. Delopoulos et al. use Triple-Correlation-Based Neural Networks to classify an image [26]. Patrick and Laine implemented the hand print identification by wavelet descriptor [56]. Hu et al. proposed an oriented-polar representation for 2-D shape recognition [51]. However, the vital drawback is that all of them require considerable computation. In addition, other methods were provided based on the feature types including line segments, groupings of edges, and regions, worked well for certain object classes [45]. However, they are often not detected frequently enough or with sufficient stability to form a basis for reliable recognition, especially for a moving fish.

Template-based methods exhibit excellent performance in the detection of a single object category, such as faces and pedestrians [49].One limitation of these rigid template based features is that they might not adequately capture variations in object appearance: they are very selective for target shape but lack invariance with respect to object transformations. Appearance-based recognition including eigenspace matching [39], color histograms [50], and receptive field histograms [40] have all been very robust on isolated objects or presegmented images, nevertheless they have been difficult to extend to cluttered and partially occluded images due to their dependency on global features. Although eigenspace matching is successful in cluttered images through many small local eigen-widows, it requires considerable computation for searching each window.[39].

Moravec first proposed using a set of local keypoints to develop image matching in 1981, and his detector was improved by Harris to increase repeatability under small variation or when close to edges [33]. Since Harris successfully demonstrated his detector is efficient for motion tracking and 3-D structure from motion recovery, the Harris corner detector has been widely applied to the image matching task. Zhang et al. matched Harris corners over a large image range by using a correlation window around each corner to select likely matches [73]. Schmid and Mohr made further progress in Harris detector application, using Harris corners to identify interest points, and then creating a rotationally invariant descriptor of the local image region rather than attempting to correlate regions. This allowed features to be matched under arbitrary changes in orientation and location. Impressive results were shown both for the speed of recognition in a large database and the ability to handle cluttered images.[23]. Although the Harris corner detector selects any image location that has large gradients in all directions at a particular scale, it fails in matching images of different sizes due to very sensitive to changes in image scale.

David Lowe first introduced the SIFT algorithm for scale-invariant feature detection and object recognition in 1999 and refined and expanded upon it in 2004 [45]. *SIFT*, Scale Invariant Feature Transform, is a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different images of the same object or scene. Mikolajczyk and Schmid have shown that of several currently used interest point descriptors, SIFT descriptors are the most effective [47]. Because of its computational efficiency and effectiveness in object recognition, the SIFT algorithm has led to significant advances in computer vision.

In this chapter, the SIFT approach is described in detail. It first takes an image and transforms it into a ”large collection of local feature vectors”. These feature vectors are also called SIFT keypoints, each of which is invariant to scaling, rotation or translation of the image, and partially invariant to illumination changes and affine or 3D projection. SIFT features are also resilient to the effects of ”noise” in the image. The recognition proceeds by matching individual features to a database of features from target fish using a fast nearest neighbor algorithm, followed by a Hough transform to identify potential model pose clusters belonging to a single object. Any 3 of these matched keys would be sufficient to accept the presence of the object. In the fish tracking project, there is no unique fish model that can be used to recognize the target fish from the underwater environment in all image frames, so an improved recognition method of fish in series of video frames is presented. Finally, the results of fish recognition based on SIFT approach will be analyzed and discussed to verify this algorithm is valid and efficient.

3.2 SIFT Approach

In a nutshell, the SIFT algorithm finds features stable over scale space by repeatedly smoothing and down sampling an input image and subtracting adjacent levels to create a pyramid of difference-of-Gaussian images. The features the SIFT algorithm detects represent minima and maxima in scale space of these difference-of-Gaussian images. At each of these minima and maxima, a detailed model is fit to determine location, scale and contrast, during which some features are discarded based on measures of their stability. Once a stable feature has been detected, its dominant gradient orientation is obtained, and a keypoint descriptor vector is formed from a grid of gradient histograms constructed from the gradients in the neighborhood of the feature.

Following are the four major stages of computation in the SIFT algorithm applied to extracting these features [45]. ***(Below some contents in this section paraphrased from (Lowe, 2004))***

3.2.1 Scale-Space Extreme Detection

The first stage of computation attempts to search all image locations to identify potential interest points that are invariant to scale change of the image. It is implemented efficiently by searching for stable features across all possible scales by using a continuous function of scale space.

Finding locations and scales which are identifiable from different views of the same object, can be efficiently achieved using a scale space function that is invariant to image translation, scaling, and rotation. It has been shown under a variety of reasonable assumptions that a scale-space kernel must be based on the Gaussian function [42]. To realize rotation invariance and maintain computation efficiency, the local maxima and minima of a difference of Gaussian function in scale space is selected.

We first define the scale space as a function $F(X, \sigma) \in \Re$ of a spatial coordinate $X \in \Re^2$ and of scale coordinate $\sigma \in \Re_+$. Since the scale space $F(X, \sigma)$represents the same information at different levels of scale (various scales σ), its domain is sampled in a particular way to reduce redundancy. The scale coordinate σ is discredited into logarithmic steps arranged in O octaves. Each octave is further subdivided in sub-levels, $S \in N$. Octaves and sub-levels are identified by a discrete octave index o and sub-level index s respectively. The octave index o and the sub-level index s are mapped to the corresponding scale σ by the formula: [42]

$$\sigma(o,s) = \sigma_0 \cdot 2^{o+s/S}, o \in o_{min} + [0, ..., O-1], s \in [0, ..., S-1] \tag{3.1}$$

where, O is the total number of octaves, s is the sub-level index (scale index), S is the sub-level resolution and $\sigma_0 \in \Re_+$ is the base scale offset.

Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space,$F(X,\sigma)$, introduced above. Koenderink and Lindeberg have shown the only possible scale-space kernel is the Gaussian function under a variety of reasonable assumptions. Therefore, the scale space of an image is defined as a function, $f(x,y,\sigma)$, that is produced from the convolution of a variable-scale Gaussian,$G(x,y,\sigma)$,with an input image, $I(x,y)$:

$$f(x,y,\sigma) = G(x,y,\sigma) \otimes I(x,y) \tag{3.2}$$

Where,

σ: The Scale: $\sigma(o,s) = \sigma_0 \cdot 2^{o+s/S}$ is sampled as explained above.

(x,y): The spatial image frame coordinate;

$\otimes$: Denoted as the convolution operation in x and y;

$I(x,y)$: Input image;

$G(x,y,\sigma)$: An isotropic Gaussian kernel:

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} \cdot exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.3}$$

To simplify the computation, consider that the 2D Gaussian function can be separated into two 1D Gaussian functions,

$$G(x,y,\sigma) = g(x,\sigma) \cdot g(y,\sigma) = \frac{1}{\sqrt{2}\sigma} exp\left(-\frac{x^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2}\sigma} exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{3.4}$$

Its convolution with the input image can be efficiently computed by applying two passes in both the horizontal and vertical direction; this is followed by finding the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal directions. This optimization is not limited to software implementation only, but applies to hardware implementation as well. The Gaussian pyramid is computed incrementally from the bottom by successive convolutions with small kernels.

$$\begin{aligned} f_x &= g(x,\sigma) \otimes I(x) \\ f_y &= g(y,\sigma) \otimes I(y) \end{aligned} \tag{3.5}$$

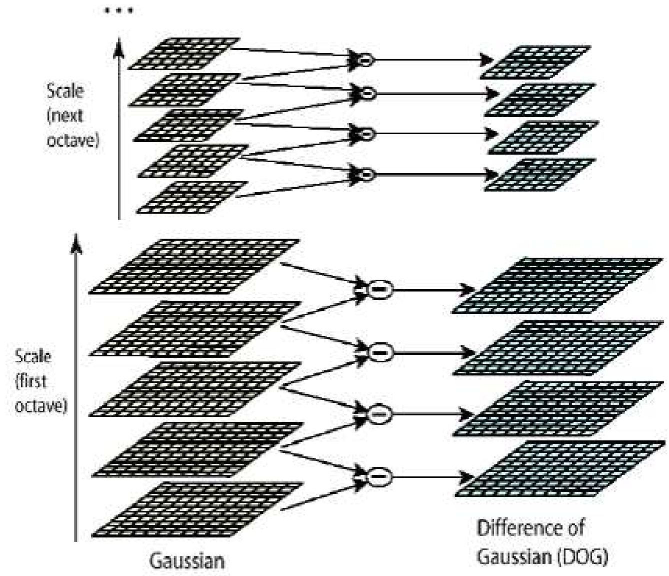


Figure 3.1: Diagram showing the computation of difference-of-Gaussian images Note: The input image is incrementally convolved with Gaussians to produce images separated by a constant factor k in scale space, shown stacked in the left column. Adjacent image scales are subtracted to produce the difference-of-Gaussian images shown on the right. Once a complete octave has been processed, we resample the Gaussian image that has twice the initial value of σ by taking every second pixel in each row and column. The accuracy of sampling relative to σ is no different than for the previous octave, while computation is greatly reduced. [45].

To efficiently achieve stable keypoint locations in scale space, scale-space extrema in the difference-of-Gaussian DoG function are convolved with the image, $D(x, y, \sigma)$, are proposed, which can be computed from the difference of two nearby separated scales. Fig.3.1 shows an efficient approach to construction of $D(x, y, \sigma)$.

$$\begin{aligned} D(x,y,\sigma(s+1,o)) &= f(x,y,\sigma(s+1,o)) - f(x,y,\sigma(s,o)) \\ &= [G(x,y,\sigma(s+1,o)) - G(x,y,\sigma(s,o))] \otimes I(x,y) \end{aligned} \tag{3.6}$$

To detect the local maxima and minima of D(x, y,), each sample point is checked to see whether it is larger or smaller than its neighbors, using the eight closest neighbors in image location and nine neighbors in the scale above and below. (see the Fig.3.2). First, a pixel is compared to its 8 neighbors at the same level of the pyramid. If it is a maximum or minimum at this level, then the closest pixel location is calculated at the next lowest level of the pyramid, taking account of the 1.5 times resampling. If the pixel remains higher (or lower) than this closest pixel and its 8 neighbors, then the test is repeated for the level above. Since most pixels will be eliminated within a few comparisons, the cost of this detection is small and much lower than that of building the pyramid. If the first level of the pyramid is sampled at the same rate as the input image, the highest spatial frequencies will be ignored. This is due to the initial smoothing, which is needed to provide separation of peaks for robust detection. Therefore, we expand the input image by a factor of 2, using bilinear interpolation, prior to building the pyramid.

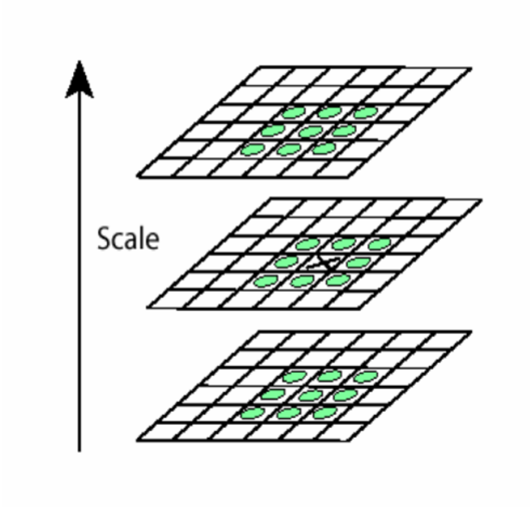


Figure 3.2: Local extrema detection, the pixel marked is compared against its 26 neighbors in a $3 \times 3 \times 3$ neighborhood that spans adjacent DoG images [45].

3.2.2 Accurate keypoints localization

This stage performs a detailed fit of keypoints to the nearby data to determine the keypoint's location, scale, and ratio of principal curvatures in order to eliminate keypoints with low contrast or situated on an edge.

Eliminating low contrast

To eliminate low contrast keypoints, a Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, is calculated. This function will represent the contrast local to the keypoint point at X.

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \ , \ X = (x, y, \sigma)^T \tag{3.7}$$

The extremum $\hat{x}$ is located by taking the derivative of this function with respect to X and setting it to zero, giving:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial X} \tag{3.8}$$

The function value at the extremum, $D(\hat{x})$, can be obtained by substituting equation 3.8 into 3.7, giving:

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X} \tag{3.9}$$

If the function value $|D(\hat{x})|$, used to measure contrast, is below a threshold T_1 value then this point is rejected. This removes extrema with low contrast. For the fish tracking experiments in our work, $|D(\hat{x})|$ less than 0.02 were discarded (image pixel values were in the range [0, 1]).

Eliminating edge responses

It is not sufficient to reject keypoints with low contrast for stability. The difference of Gaussian function will have a strong response along edges and the location along the edge is poorly determined and is unstable to small amounts of noise. We call this poor localization.

To eliminate extrema have poor localization, we use the fact that in such cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. The principal curvatures can be computed from a 2×2 Hessian matrix, H, at the location and scale of the i^{th} keypoints:

$$H^i = \begin{bmatrix} D^i_{xx} & D^i_{xy} \\ D^i_{xy} & D^i_{yy} \end{bmatrix} = \begin{bmatrix} D^i_x D^i_x & D^i_x D^i_y \\ D^i_x D^i_y & D^i_y D^i_y \end{bmatrix} \tag{3.10}$$

The derivatives are estimated by taking differences of neighboring sample points. Then, we can use the ratio of eigen values of H to infer if there exists large curvature in one direction and low curvature in the perpendicular direction.

To avoid explicitly computing the eigenvalues, only their ratios are considered. Let α be the eigenvalue with the largest magnitude and β be the smaller one. Then, compute the sum of the eigenvalues from the trace of H and their product from the determinant:

$$\begin{aligned} Tr(H^i) &= D^i_{xx} + D^i_{yy} = \alpha^i + \beta^i \\ Det(H^i) &= D^i_{xx} D^i_{yy} - (D^i_{xy})^2 = \alpha^i \beta^i \end{aligned} \tag{3.11}$$

In the unlikely event that the determinant is negative, the curvatures have different signs so that the point is discarded as not being an extremum. Therefore, we consider using ratio of sum and product of determinant:

$$\frac{Tr(H^i)^2}{Det(H^i)} = \frac{(\alpha^i + \beta^i)^2}{\alpha^i \beta^i} = R^i \tag{3.12}$$

Let a threshold defined as follow:

$$T_2 = \frac{(t+1)^2}{t}, \ t = \alpha/\beta \tag{3.13}$$

The value of $(t+1)^2/t$ is at a minimum when the two eigenvalues are equal and it increases with t.

Therefore, to check that the ratio of principal curvatures is below some threshold T, only need to check:

- If $R^i < T_2$, the current keypoints is remained
- If $R^i \geq T_2$, the keypoints is rejected;

We use a value of $t = 8$ as threshold in our work.

3.2.3 Orientation assignment

One or more orientations are assigned to each keypoint location based on local image gradient directions. This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor can then be represented relative to this orientation, achieving invariance to rotation.

The following approach is applied to find the stable orientation results. The scale of the keypoint is used to select the Gaussian smoothed image, f, described in section 3.2.1, so that all computations are performed in a scale-invariant manner. For each image sample, $f(x, y)$, at this scale, the gradient magnitude, $|\nabla f(x, y, \sigma)|$, and gradient orientation, $\angle \nabla f(x, y, \sigma)$, are precomputed using pixel differences:

$$\begin{aligned} M_{xy} &= |\nabla f(x, y, \sigma)| = \sqrt{(f_{x+1,y} - f_{x-1,y})^2 + (f_{x,y+1} - f_{x,y-1})^2} \\ \theta_{xy} &= \angle \nabla f(x, y, \sigma) = \arctan\left(\frac{f_{x+1,y} - f_{x-1,y}}{f_{x,y+1} - f_{x,y-1}}\right) \end{aligned} \tag{3.14}$$

An orientation histogram including 36 bins covering the 360 degree range of rotations is formed from the gradient orientations in a window around the keypoint. Each sample added to the histogram is weighted both by the magnitude of the gradient $|\nabla f(x, y, \sigma)|$ and a Gaussian window $W(x)$ centered on the keypoint and of deviation 3σ times that of the current smoothing scale. The orientation, $\vartheta(x, y)$, of a keypoint (x, y, σ) is obtained as the dominant orientation of the gradient that is the maximum of the histogram of the gradient orientations $\angle\nabla f(x, y\sigma)$within a window around the keypoint. In addition to the global maximum, any other local maximum that is within 80 percents of the highest peak is retained as well. Although a few points are assigned multiple orientations, they contribute significantly to the stability of matching. Finally, a parabola is fit to the 3 histogram values closest to each peak to interpolate the peak position for better accuracy. Thus each key location is assigned a canonical orientation so that the image descriptors are invariant to rotation.

3.2.4 Keypoint Descriptor

Given a stable location, scale, and orientation for each keypoint, it is now possible to describe the local image region in a manner invariant to these transformations. The SIFT descriptor of a keypoint (x, y, σ) is a local statistic of the orientations of the gradient of the scale space, $f(\cdot, \sigma)$.

One approach to this is suggested by the response properties of complex neurons in the visual cortex. In neurons, features are responded to as a gradient at a particular orientation, and their positions are allowed to shift over a small region instead of being fixed at an exact location, while orientation and spatial frequency specificity are maintained. Edelman et al. [28] have performed experiments that simulated the responses of complex neurons to different 3D views of computer graphic models, and found that the complex cell outputs provided much better discrimination than simple correlation-based matching. This robustness to local geometric distortion can be obtained by representing the local image region with multiple images representing each of a number of orientations (referred to as orientation planes). Each orientation plane contains only the gradients corresponding to that orientation, with linear interpolation used for intermediate orientations. Each orientation plane is blurred and resampled to allow for larger shifts in positions of the gradients.

The SIFT descriptor of a keypoint (x, y, σ) is a local statistic of the orientations of the gradient of the Gaussian scale space,$f(\cdot, \sigma)$. This approach can be efficiently implemented by using the same precomputed gradients, M, and orientations, θ that were used for orientation selection, (see section 3.2.3). For each keypoint, we use the pixel sampling from the pyramid level at which the key was detected. The pixels that fall in a circle of radius 8 pixels around the key location are inserted into the orientation histograms. The orientation is measured relative to the keypoint's orientation.

Descriptor vector setting: each keypoints vector includes $N = N_g \cdot N_d$ elements features. Here, N_g is 2-D square spatial grids with unitary edge. $N_d = 8$ is numbers of bins for the orientation histogram in each grid. The window $W(x)$ is Gaussian with deviation equal to half the extension of the spatial bin range that is $\sigma_w = N_g/2$. In our work, grids $N_g = 4 \times 4$, bins $N_d = 8$, the total number of vector element features is $N = N_g \cdot N_d = 128$. (These numbers can be changed by setting the appropriate parameters).

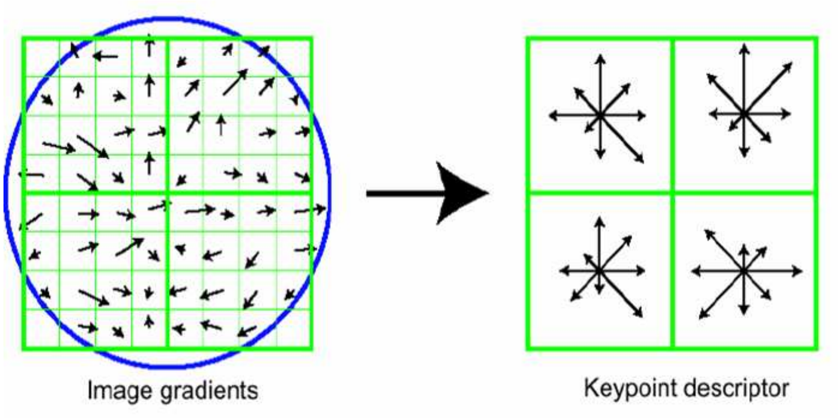


Figure 3.3: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point, as shown on the left. These are weighted by a Gaussian window, indicated by the overlayed circle. These samples are then accumulated into orientation histograms summarizing the contents over larger regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. To reduce clutter, this figure shows a 2x2 descriptor array computed from an 8x8 set of samples. [45]

Weighting: The samples surrounding keypoints are weighted by the gradient modulus and a Gaussian window. The purpose of the Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are far from the center of the descriptor, as these are most affected by misregistration errors.

Each sample $(x, y, \vartheta(x, y))$ is processed as follows:

- Weighted by the gradient magnitude ;

- Weighted by the Gaussian window W(x, w);

- Projected on the centers of the eight surrounding bins;

- Summed to each of this bins proportionally to its distance from the respective center.

Vector normalization: First, brightness changes the image contrast (effectively adding a constant to each image pixel), which hence affects the gradient values. To reduce these effects, the vector is normalized to unit length. Second, illumination changes may result in large changes in gradient magnitude, but are likely to have less influence on gradient orientation. To remedy this, we cancel those values in the unit feature vector no larger than some threshold (for example, T=0.3), and renormalize to unit length. Third, In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the keypoint orientation,(see Fig.3.4).

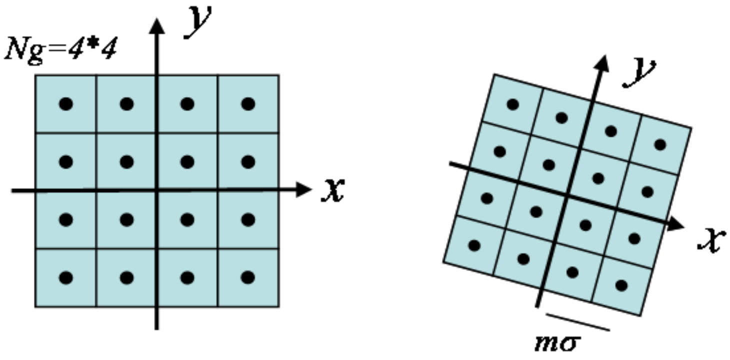


Figure 3.4: A sketch map of the result of descriptor vector normalization: at left is the $Ng = 4 * 4$ descriptor, and at right is the result that normalized vectors are rotated, so that the axis x is aligned to the direction θ of the keypoint.

3.3 Keypoints Matching

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector described above.

Our keypoint descriptor has a 128-dimensional feature vector. There is no algorithm known that can identify the exact nearest neighbor of a point in high dimensional spaces that has greater efficiency than an exhaustive search. At present, the best-known and most widely used of these is the k-d tree lookup of nearest neighbors, a complete binary tree having smaller bins in the higher-density regions of the space. In our work, we use a modification of the k-d tree algorithm called the best-bin-first search method provided by Bees and Lowe [18] that identify the nearest neighbors with high probability using only a limited amount of computation.

K-D Tree Search

The standard version of the k-d tree is built as follows. Beginning with a complete set of N points, the data space is split on the dimension i in which the data exhibits the greatest variance. A partition is made at the median value m of the data in that dimension, so that an equal number of points fall to left/right (or up/down) setseach with half the points of the parent node. These children are again partitioned into equal halves, using planes through a different dimension. Partitioning stops after log_2N levels, with each point in its own leaf cell. It is an iteration process with both halves of the data, and an internal node. This creates a balanced binary tree with depth $d = log_2N$, [46]. Fig.3.5 shows a kd-tree: The left one shows the split lines around medians, and the right figure shows the actual kd-tree.

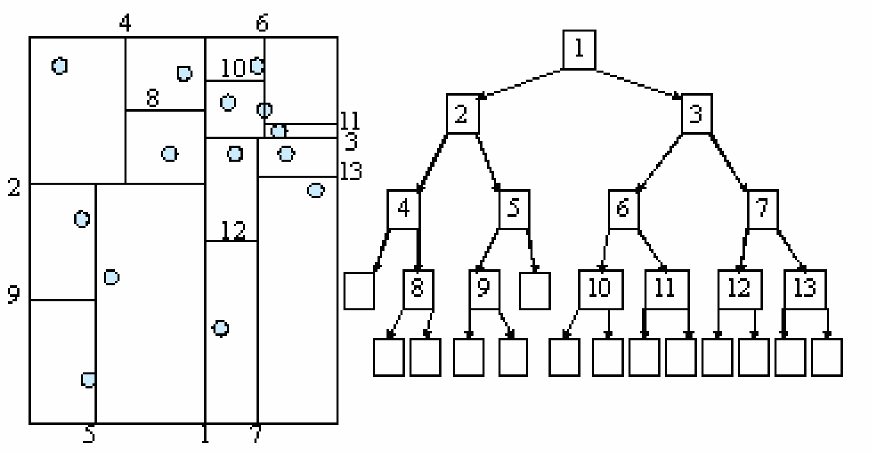


Figure 3.5: k-d tree

In the K-D Tree Build Algorithm, the median can be found by creating a list of sorted points for each dimension. The leaves of a k-d tree form a complete partition of the data space, with the interesting property that bins are smaller in higher-density regions and larger in lower density areas. This means that there is never an undue accumulation of points in any single bin, and that the nearest neighbor (NN) to any query should lie, with high probability, in the bin where the query falls, or in an adjacent bin.

The k-d tree nearest neighbor (NN) search algorithm, is used to find the NN to a given target point that is not a node in the tree, q. It uses a simple test to discard large portions of the tree. A backtracking, branch-and-bound search for the tree is used that iteratively refines the nearest distance.

The search starts from the root of the tree, and is recursive. At each point in time, the algorithm maintains the distance R from the point on the tree encountered that is closest to q. Initially, $R = \infty$.

At a leaf node (say point p) the algorithm checks if $\|q - p\| < R$. If so, R is set to $\|q - p\|$, and p is stored as the closest point candidate.

At an internal node, the algorithm proceeds as follows. A median value M is computed for each node, so that at least 50 percent of the points have their i^{th} coordinate greater-or-equal to M, while at least 50 percent of the points have their i^{th} coordinate smaller than or equal to M.) The algorithm checks if the i^{th} coordinate of q is smaller than or equal to M. If so, the algorithm recurses on the left node; otherwise it recurses on the right node.

After returning from the recursion, the algorithm performs the "bounds overlap ball" test: it checks whether a ball of radius R around q contains any point in $\Re^d$ whose i^{th} coordinate is on the opposite side of M with respect to q. If this is the case, the algorithm recurses on the yet-unexplored child of the current node. Otherwise, the recursive call is terminated. At the end, the algorithm reports the final closest-point candidate.

The k-d algorithm of NN search process is very effective in low-dimensional spaces, but in higher dimensions performance degrades rapidly because there are many more bins adjacent to the central one that must be checked. Checking a large number of bins is the cost that must be paid to guarantee that the nearest neighbor has been found. This is despite the fact that only a small fraction of their volume could possibly supply the nearest neighbour. If we are willing to sacrifice this guarantee, then the complexity can be reduced by interrupting the search before it terminates. To accomplish this we simply limit the numbers of leaf node to visit to some maximum, N_{max}.

In addition, the backtracking search described above is not efficient because the order of examining leaf nodes is determined according to the tree structure. This structure depends only on the stored points, and does not take into account the position of the query point. Therefore, it is desirable to order the search so that bins more likely to contain the nearest neighbor are examined early on. This suggests a variant of the standard k-d tree algorithm that visits the bins of the k-d tree in increasing order of distance from the query point. This requires the use of a heap based priority queue for efficient determination of search order.

Best Bin First (BBF) Search

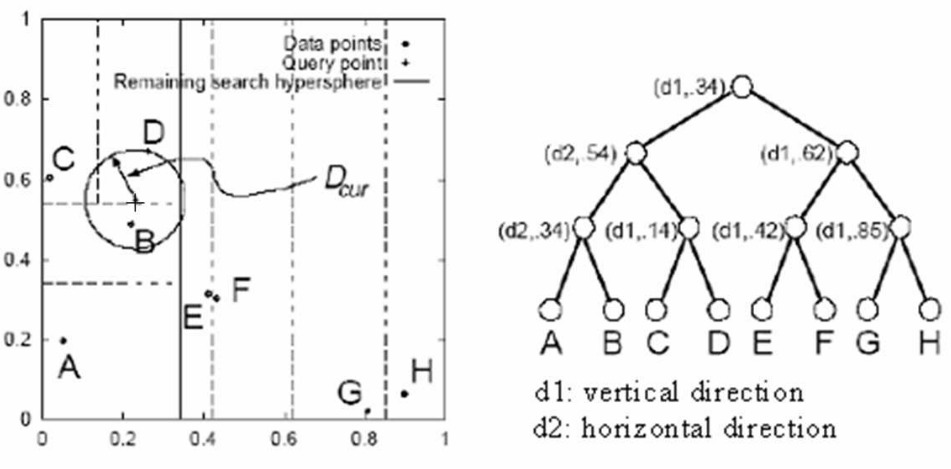


Figure 3.6: An example of Best Bin First (BBF) search.[46]

Fig.3.6 shows a k-d tree with 8 points, $k = 2$. In BBF search, the bins closest to the query point q are examined first. More than the standard search, this is likely to maximize the overlap of the hypersphere centered on q with radius D_{cur}, with the hyperrectangle of the bin to be searched. In the example shown, BBF would reduce the number of leaf nodes examined since the NN is in the closest adjacent bin in space (directly below q), rather than the closest bin in the tree structure (to the left of q).

The Best-Bin-First (BBF) algorithm maintains a priority queue of subtrees, where the priority of a subtree is inversely related to the distance between the query point and the bin corresponding to the subtree [14]. Initially, we insert the root of the k-d tree into the priority queue.

Then we repeatedly carry out the following procedure. First, we extract the highest priority node v from the queue (subtrees $v1$ through $v4$ are initially in the queue.) In this case the highest priority corresponds to that closest to the query point. For example, in Fig.3.7, the closest is $v1$. Then we descend the subtree rooted at this node in search of the leaf bin that is closest to the query point. As we descend the subtree, for each node u that we visit we insert u's sibling. For example, in Fig.3.7, nodes $u1$, $u2$, and $u3$ are inserted into the queue. These nodes will be considered at a later time in the search.

After a leaf node has been examined, the top entry in the priority queue is removed and used to continue the search at the branch containing the next closest bin. The BBF algorithm terminates when the priority queue is empty (meaning that the entire tree has been searched), or sooner, if the distance from the query point to the rectangle corresponding to the highest priority subtree is greater than the distance to the closest data point.

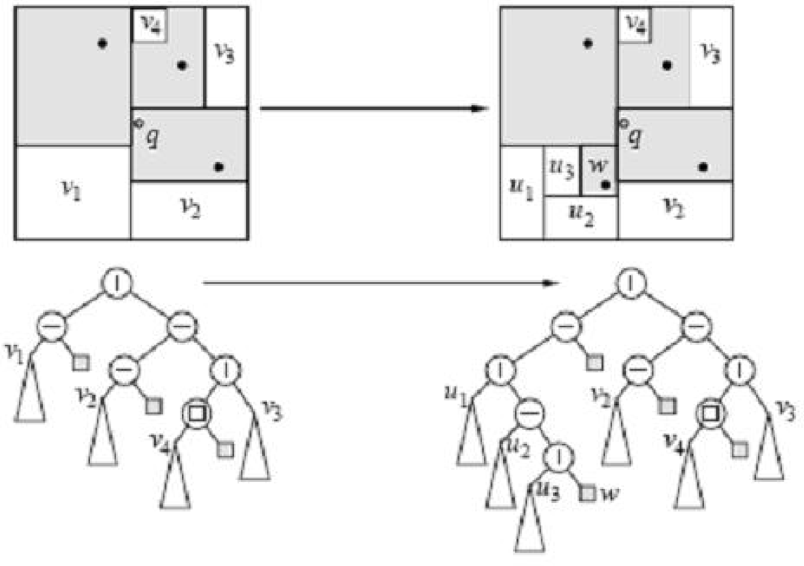


Figure 3.7: Priority k-d tree in BBF algorithm.[14]

3.4 Clustering with the Generalized Hough transform

In general, a typical image has more than two thousand keypoints that may belong to various objects and background. Although the BBF search based on the nearest neighbor permit rejecting lots of false matches coming from the background, it cannot eliminate those true matches occurring from other objects. To efficiently implement the object identification for some highly occluded or small objects, (i.e. if there are as few as 3 features), the recognition is must be valid and reliable.

There exist several robust fitting methods such as RANSAC or Least Median of Squares. These methods work well for finding a cluster of features in transformation space, but perform poorly when the percent of inliers falls much below 50 percent. Fortunately, the Generalized Hough Transform is a better tool that has ability to cluster features in pose space successfully [64].

The Generalized Hough Transform (GHT) is an extension of the standard Hough Transform that can be used not only to detect an object described with an analytic equation (e.g. line, circle, etc.), but also to detect any arbitrary objects described by a model in an image (such as Fig.3.8).In fact, all shapes aren't easily expressed using a small set of parameters. The solution proposed is then to create a table for storing all the edge pixels (feature points) of the target shape. For each pixel, we can store its position relative to a chosen reference point (x_c, y_c) of the shape. With this information, we can build a table that records for each point's edge tangent orientation ϕ, the direction β, and distance r to the reference point. Thus, when we find a point with edge tangent orientation , we have to vote only in the direction of r, β. Of course, depending on the complexity of the shape, there may be multiple such points with similar tangent orientation.

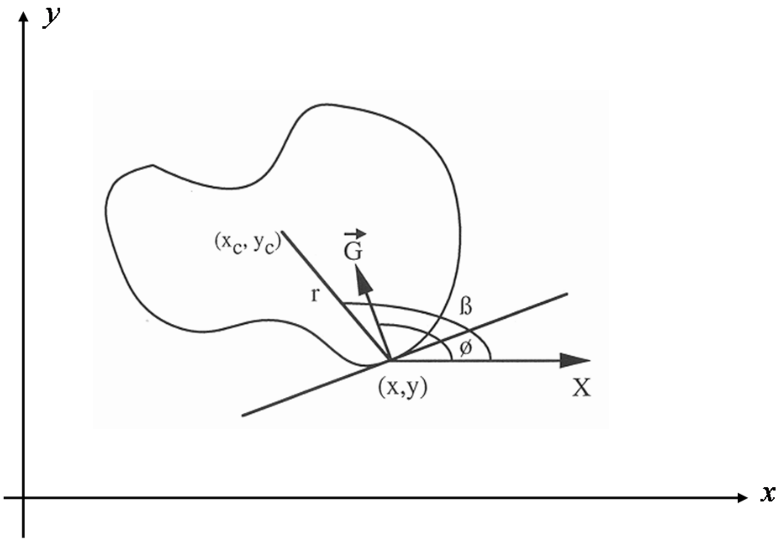


Figure 3.8: An arbitrary object described its model in an image.[64]

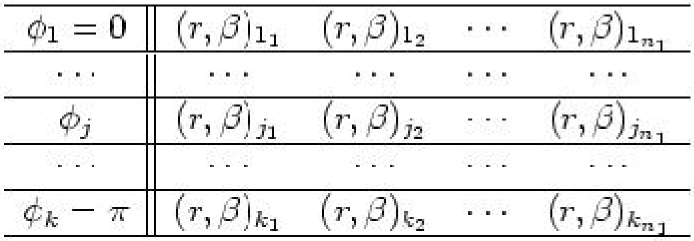
Define a reference point (x_c, y_c) somewhere inside the 2D shape (e.g., the gravitational center). For each point (x, y) on the boundary of the shape, find two parameters r, β and the gradient direction $\angle G$. Set a table with k entries each indexed by an angle $\phi_i (i = 1, ..., k)$ which increases from 0 to 180 degrees with increment $180/k$, where k is the resolution of the gradient orientation. Add the pair (r, β) to the table entry with its ϕ closest to $\angle G$.

For each image point (x, y) with $|G(x, y)| > T_s$, find the table entry with its corresponding angle ϕ_i closest to $\angle G(x, y)$. Then for each of the n_j pairs $(r, \beta)_i, (i = 1, ..., n_j)$ in this table entry, find

$$\begin{cases} x_c = x + y \cos \beta \\ y_c = y + r \sin \beta \end{cases} \tag{3.15}$$

and increment the corresponding element in the $H(x_c, y_c)$ array by 1, and then all elements in the H table satisfying $H(x_c, y_c) > T_h$ represent the locations of the shape in the image.

In our project, this Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much



$\phi_1 = 0$	$(r,\beta)_{1_1}$	$(r,\beta)_{1_2}$	$\cdots$	$(r,\beta)_{1_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
ϕ_j	$(r,\beta)_{j_1}$	$(r,\beta)_{j_2}$	$\cdots$	$(r,\beta)_{j_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\phi_k - \pi$	$(r,\beta)_{k_1}$	$(r,\beta)_{k_2}$	$\cdots$	$(r,\beta)_{k_{n_1}}$

Table 3.1: An example of R-table for GHT

higher than for any single feature. Each of our keypoints has 4 parameters: 2D location (x, y) , scale (σ), and orientation (θ), and each matched keypoint has a record of the keypoint's parameters relative to the training image in which it was found. Therefore, a Hough transform entry is created to predict the model location, orientation, and scale from the match hypothesis.

Now the Hough space becomes 4-dimensional $H(x_c, y_c, \sigma, \theta)$. For each image point $(x, y)$with $|G(x, y)| > T$, find the proper table entry with $\phi_j = \angle G(x, y)$. Then for each of the n_j pairs $(r, \beta)_i (i = 1, ..., n_j)$ in this table entry, do the following for all σ and θ, find

$$\begin{cases} x_c = x + r \cdot \sigma \cdot \cos\beta + \theta \\ y_c = y + r \cdot \sigma \cdot \sin\beta + \theta \end{cases} \tag{3.16}$$

and increment the corresponding element in the 4D H array by 1:

$$H(x_c, y_c, \sigma, \theta) = H(x_c, y_c, \sigma, \theta) + 1 \tag{3.17}$$

All elements in the H table satisfying $H(x_c, y_c, \sigma, \theta) > T_h$ represent the scaling factor , rotation angle θ of the shape, as well as its reference point location (x_c, y_c) in the image.

This prediction has large error bounds, as the similarity transform by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations. Therefore, use broad bin sizes of 30 degrees for orientation, one octave (factor of 2) for scale, and 0.25 times the maximum projected training image dimension (using the predicted scale) for location. If all clusters with fewer than 3 entries in a bin, then the match is rejected [45].

3.5 Dynamic fish recognition process

The fish is a non-rigid object and swims freely in the lake or sea at all time. The illumination underwater is unstable and blurry. There are different light reflections from the fish's scales and fish's shape changes over time. All of these factors cause features to vary considerably. Therefore there is no unique fish model can be used to recognize the target fish from the underwater environment in all image frames.

However, between two video frames in a sequence, the fish can be considered as a rigid object without moving in the ideal background environment, so that regarding the fish in the current frame as a model to identify the target fish in the next frame is feasible. Following this idea, we always use the fish in former frame as the model when trying to find the fish in the current frame. The boundaries around the matched keypoints in the former frame create a subsection of the current image within which keypoint matching between frames can occur. When the fish in the current frame can be found, this fish can be used as the model to recognize the fish in the next frame. This updating of the fish model using SIFT based recognition in a series of frames is implemented continuously during tracking. The algorithm is as follow:

1. Obtain a target fish model from the database or the result of our first feature extraction algorithm presented in chapter 3.
2. Find out the SIFT keypoints in the current frame, and match with the model:

   If there is no keypoints can be matched, then check out the next frame;

   If some keypoints are good match, which means the target fish is found, then
3. Calculate the location of central point among these matched keypoint in the current frame (f_i), which also represents the target fish position in the image.
4. Open a window around this central point, the size of windows according to the fish size and the distance gained from Sonar data.
5. Regard the fish in above window as a new model to match with SIFT keypoints in the next frame (f_{i+1}).
6. Return step 3 and repeat following steps.

3.6 Result and Analysis

The Large Mouth Bass video streams were taken by a CCD camera installed on a MicroROV in Paradise Lake, Ontario, Canada. There are two video data used in verifying the SIFT based object recognition algorithm. In the first video, images are of dimensions 480 × 640, in the second video, images are of dimensions 240 × 320 and the quality of image is poorer. Both of video streams are grabbed at a frame rate 20Hz. All of the images shown in below figure are the results that have been preprocessing introduced in section 3.2 of chapter 2.

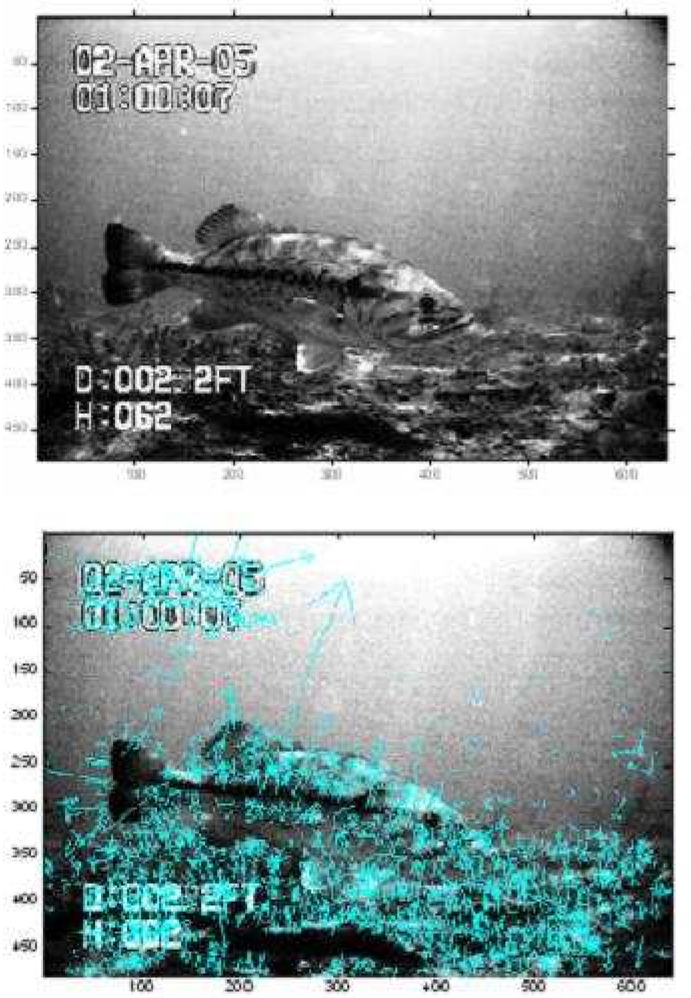


Figure 3.9: An example of SIFT keypoints in the fish image. The top one is the 480 × 640 pixel original image. The bottom one shows final 2904 SIFT keypoints.

Fig. 3.9 shows the final SIFT Keypoints on a nature fish image with a high-resolution 480 by 640 pixel. Keypoints are displayed as vectors indicating scale, orientation, and location. Since this image has good quality and abundant vision information in both of target fish and background, the SIFT features reach more than two thousand.

In Fig.3.10, we see that 6 points are matched between the fish model and the target fish in the natural underwater environment image. Although the fish settle

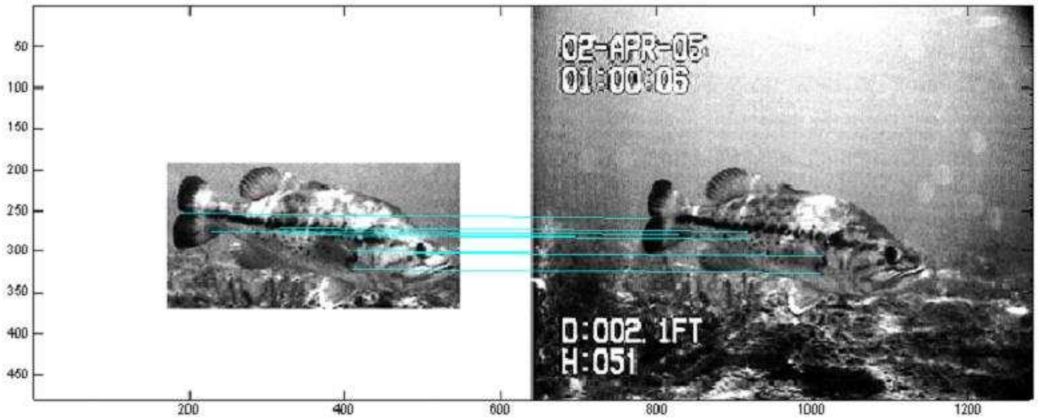


Figure 3.10: An example shows the result of matched keypoints.

near the floor of the lake, weeds make the background cluttered, the illumination has obvious changes caused by the unstable and poor underwater lighting and fish scale reflections, and fish orientation varies slightly from the model, we still obtain a good match. Hence, the SIFT based recognition algorithm is successful in identifying the target fish from the natural environment.

Fig.3.11 shows an improved recognition method and its results. In Fig.3.11(a), the left image is a model from the former video frame (f_{i-1}). The right image is the current frame (f_i). In (b), the left image shows the fish matched in frame (f_i) as the model. On the right is the image from the video frame (f_{i+1}). In (c), the new model comes from the matched fish in the (f_{i+1}) frame and is used to recognize the target fish in next frame (f_{i+2}). Finally in (d), the same procedure is used to apply the model in (f_{i+2}) to match the fish from the video frame (f_{i+3}). The time interval of each video frame is 0.5 second. The positive matching results indicate that the improved recognition method is valid and efficient, and it has ability to identify the object with these SIFT features from a series of video frames showing that it has definite potential for AUV tracking tasks.

Fig.3.12 displays the matching results for various states of the fish and environment. The images have low-resolution (240 × 320 pixels) and poor quality. In the top figure, the target fish is in the middle of the water column and the background is clear. Many keypoints are matched and fish recognition is easy and reliable. In the middle figure, part of the fish overlaps with the bottom grass but the target fish is still detected. In the bottom figure, the fish is swimming slowly among the water grass near the floor, and another fish is swimming along its right side. Despite the complexities of the underwater environment, the SIFT approach successfully identified the target fish from cluttered backgrounds.

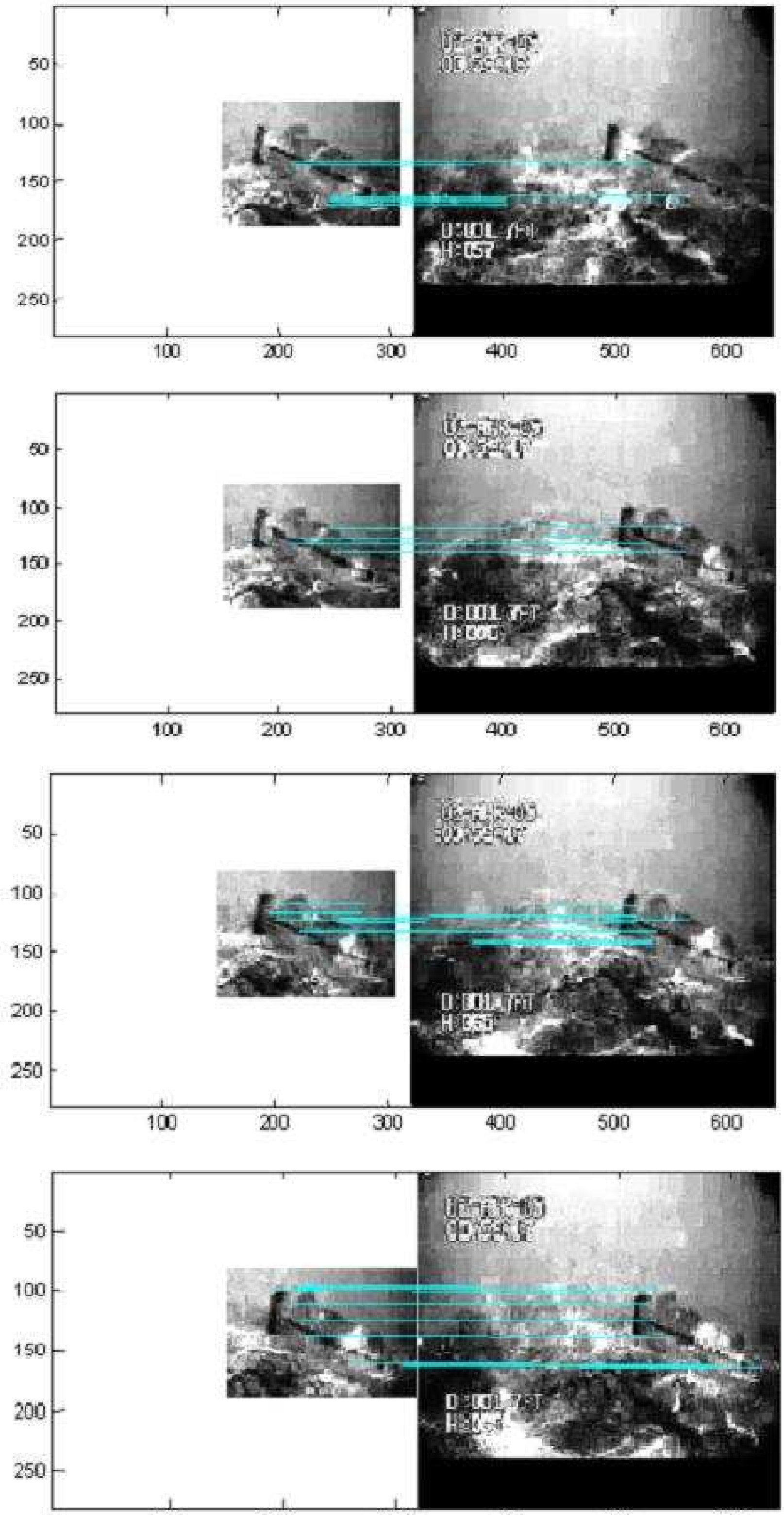


Figure 3.11: Fish tracking based on SIFT algorithm.

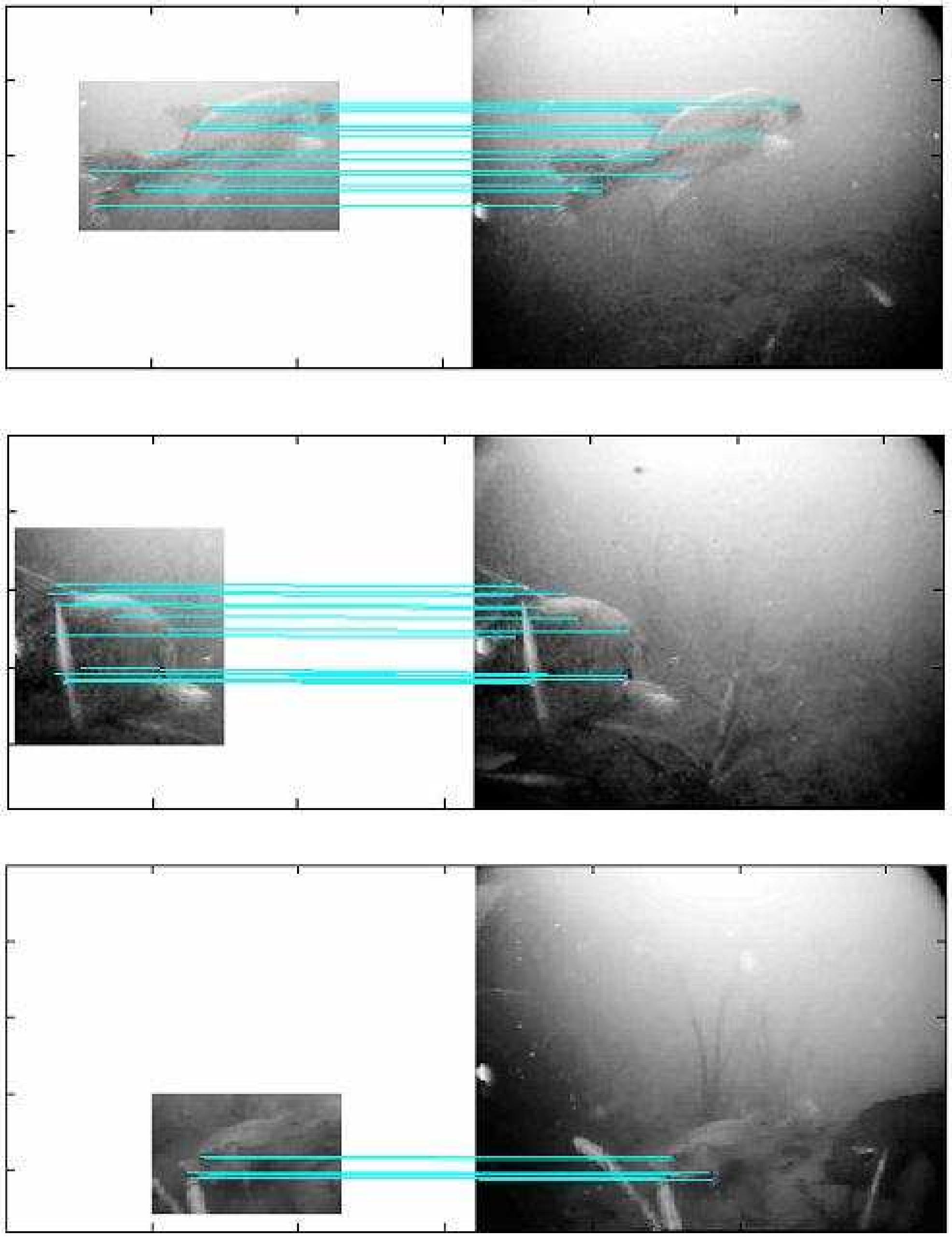


Figure 3.12: The results of fish recognition in different situation of the lake.

3.7 Summary

In this chapter, an object recognition algorithm based on SIFT approach was developed. The SIFT keypoints description vectors were obtained by implementing four major stages of SIFT approach. The best candidate match for each keypoint is found by identifying its nearest neighbor (the keypoint with minimum Euclidean distance for the invariant descriptor vector) in the database of keypoints from training images. An approximate algorithm, Best-Bin-First (BBF) algorithm, was applied to identify the closest neighbor instead of exact nearest neighbor with high probability. Clustering keypoints that agree on the possible object with Hough transform are identified as the object fish, reliable recognition is possible with as few as 3 features. For images of fish taken from the natural lake, a dynamic recognition process was designed using continually updated fish model to match and recognize the target fish from a series of video frames. The image processing results demonstrate the SIFT Based recognition algorithm is effective and efficient to identify Large Mouth Bass in natural cluttering underwater environment.

Chapter 4

Underwater Image Sonar

In the case of a MicroROV, i.e. the VideoRay Pro III, the room inside the small pressure hull where hardware is placed is very restricted. Furthermore, there is inadequate space for installing a second video camera. For a monocular camera system, the depth of field is very difficult to estimate accurately by way of vision processing. Along with the camera, the most important sensor for small, low cost ROVs is sonar. These sensors can locate and track objects reliably using real, multibeam sonar sequences, and take full advantage of dense spatial and temporal information to be fast enough to support real-time ROV or AUV operations. Therefore, a high-resolution forward looking imaging sonar that provides accurate images at high bandwidth can be particularly helpful in obtaining distance measurements to objects of interest, (e.g. fish).

4.1 Introduction

Underwater, Light and Electromagnetic waves attenuate rapidly and are absorbed over very short distances. Although lasers and radar are used extensively for certain applications, their working ranges are greatly reduced underwater, and are mostly used for ground or atmospheric observations. Sound can more easily penetrate the dense water environment than light and Electromagnetic waves. They are ideal for navigation and measurement under water. The resolution of acoustic imaging does not approach optics, yet it does provide a remarkable extension to our vision system.

Most underwater vehicles require forward looking sonar for navigation, obstacle avoidance and target recognition. These sonars display plan position indicator (PPI) image of acoustic energy returns as brightness levels. Modern devices use an array of hydrophones which allows for much accurate and faster updates of images, and are preferred to order scanned devices that take several seconds for each scan.[10].

4.1.1 Brief Overview of Subsea Sonar

Sonar (sound navigation and ranging) can be divided into two generic types, passive and active (echolocation). The passive sonar is a very sophisticated apparatus that detects noise radiated by targets. This noise can be a mixture of sounds generated by propellers and hull vibrations caused by motors, engines, pumps, and hydrodynamic forces. By analysis of the received signals the sonar often can identify the type of target. Passive sonars are mainly used in military applications to detect submarines. The two major weaknesses are that it cannot directly measure range to a target and most require large receiving transducer arrays.

Active sonars are used in a broader range of applications, and we shall concentrate on them. The active sonar emits pulses of acoustic energy, often called a "ping", and receives reflections of the pulse. It displays or records these received signals for the observation. To measure the distance to an object, one measures the time from emission of a pulse to reception. To measure the bearing, one uses several hydrophones, and measures the relative arrival time to each in a process called beamforming.[10].

The most common types of active sonars available include five kinds as follows. The first is the *Echo Sounder Sonar Transponder* is a marine instrument used primarily for determining the depth of water by means of an acoustic echo. It creates a pulse at a given frequency, and then receives the pulse reflected back from some surface. The distance between surface and sonar can be estimated by measuring the time interval between transmission and reception of the pulse. Normally, these sonars are used in boats or AUVs to estimate the vehicle's altitude from the seabed. [7].

The second common type of active sonar is the *Side-Scan Sonar*. This acoustic imaging device is often towed by a vessel or submarine, typically to form wide-area images of the seabed for surveying purposes. It emits fan-shaped pulses down toward the seafloor across a wide angle perpendicular to the sensor direction. The intensity of the acoustic reflections are recorded to generate the image of the seafloor. Side-scan sonars have been used as tools for mapping the seabed or detecting underwater objects in many commercial applications. In addition, Slant-range and geometric corrections of images can be used to work out the position of a vehicle, but this requires too many constraints, including the assumption of a flat seabed. [4].

The third common active sonar is the Bathymetric Sonar, also known as the multibeam echo sounder. It is similar in operation to the echo sounder sonar transponder, except that it uses an array of hydrophones (as opposed to a single one), which allows for multiple beams. This produces profile (cross-section) views of the seabed. Unlike standard Side-Scan Sonar, which produce a flat image with shadows denoting the shape of the seafloor, this type of acoustic technology produces a more fully rendered image that covers a wide swath of the benthos, providing a more detailed image, as well as more accurate measures of water depth.

This type of sonar is commonly used to build contour maps or investigate more about seafloor habitats in the hopes of conserving them.[8].

The fourth common active sonar is *Forward-Looking Scanned Sonar*. It is used for a diverse set of applications, such as obstacle avoidance, midwater mine detection, and surveillance. The sonar consists of a single hydrophone which is mechanically scanned along the horizontal axis, sweeping a so-called sector. The returns are then used to create an image. Most systems provide the user with the option of choosing the size of the sector to scan and with some degree of control on the resolution. In most cases higher resolution results in a slower refresh rate. Again a longer range will also result in a slower refresh rate. The major advantage of this type of sonar is its capability of detecting objects or seabed features. These can be observed in subsequent scans and tracked, providing the underwater vehicle is moving at a slow speed. Another advantage is its price, which is much less than many advanced sonar system.

In recent years the fifth common active sonar, *Forward-Looking Multibeam Sonar*, has also appeared. This type of sonar uses a fixed array of hydrophones, scanned electronically, which allows much faster updates of sectors. These sonars are more expensive than mechanical systems, but nevertheless their popularity in the underwater community has been growing in obstacle avoidance, motion estimation and image recognition.[57].

In conclusion, forward-looking sonars are the only sensors that satisfy the two main requirements for tracking: they can scan an area *in front of* the vehicle and at a *sufficiently high frame* rate to yield large overlaps between consecutive frames. This makes it possible to track targets using image processing techniques.

4.1.2 Forward-looking Image Sonar Principle

All sonar systems have the same basic principle of operation. The area or object to be identified is insonified by acoustic energy. Usually, forward-looking imaging sonars with the sonar's transducer at the top are installed vertically to the front of an underwater vehicle. The sonar beam can continuously scans a given area, maximum to full 360 degree sector, in either direction by rotating the transducer in a straight line in terms of a series of angular scan ping width that is a fixed step angle depending on the resolution setting of the sonar. Increasing the fixed step angle will increase scan speeds but will lower the resolution of the sonar image. Reducing the fixed step angle will improve image resolution but give slower overall scan time due to more data samples taken during the scan. The beam's movement through the water will reflect some of the energy back toward the sonar when it meets objects in the path. These returned signals are mapped against a color scale. The different colored points, representing the time (or slant range) of each echo return, can plot a line on a video display screen. The different colored lines consist of an image that depicts the various echo return strengths. The following characteristics are necessary to produce a visual or video image of the sonar image:

1) the angle through which the beam is moved is small, 2) the transmitted pulse is short, and 3) the echo return information is accurately treated. The visual images produced by the sonar provide the viewer with enough data to draw conclusions about the environment being scanned, and to recognize sizes, shapes and surface reflecting characteristics of the chosen target. The primary purpose of the imaging sonar is as a viewing tool. [10]

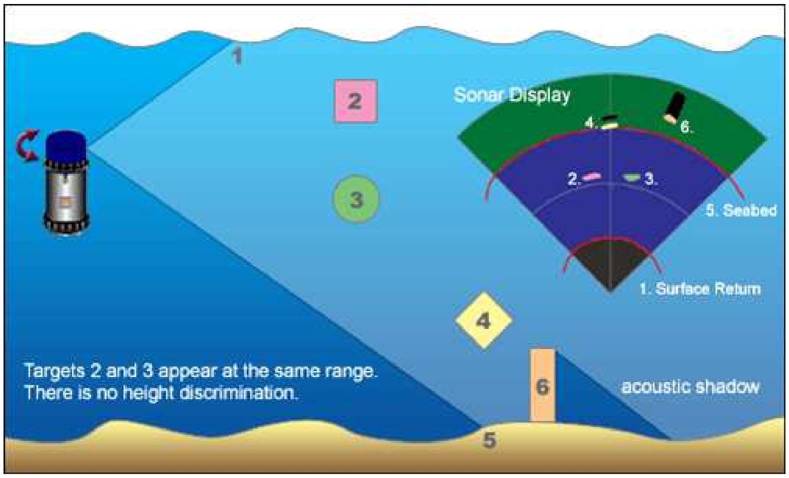

1
2
3
4
5
6
Sonar Display
4.
6.
2.
3.
5. Seabed
1. Surface Return
Targets 2 and 3 appear at the same range.
There is no height discrimination.
acoustic shadow

Figure 4.1: Forward-looking imaging sonar principle. [10]

4.1.3 Relative Work Overview

Essential to the development of both automatic classification of objects for divers, ROVs, etc. [21] and obstacle avoidance system using forward looking multibeam sonar image [70] are the detection, tracking and motion estimation of objects or obstacles in a sonar image sequence.

Much work has been done on segmenting side-scan sonar images, but not so much on forward looking sonar images. Unfortunately, the noisy nature of forward-looking sonar images makes it very difficult to compute meaningful segmentations from a single return.

The literature of subsea sonar image processing, to our knowledge, does not include many different approaches. Hallam [37] proposed an early approach that tackled the tracking and motion estimation. He demonstrated a system that tracks point-features by means of single Kalman filter. This system could deal with moving objects and could be used to find the portions of an underwater vehicle. The system was demonstrated with encouraging simulation results. Carpenter [62] presented an approach for tracking returns from a forward-looking sonar with the aim of performing concurrent mapping of environment and localization of an underwater vehicle. This approach uses decoupled extended Kalman filters associated to regions of strong bottom backscatter. It was tested in real conditions and showed promising results. Also of interest is the approach proposed by Moran et al. [15], who use a high-frequency profiling sonar to sense the environment. Tracking is then performed by means of a Multiple Hypothesis Tracking (MHT) algorithm. This approach is used to perform curved shape reconstruction of objects in a water tank. It is limited by the chosen sensor, as only a limited region can be sensed in each iteration. A novel method for tracking return from a multibeam forward-looking sonar was developed by Lane et al. [41]in the Ocean System Laboratory. The approach works by segmenting high intensity returns into regions. Optical flow calculations are then performed on these regions to obtain magnitude and direction motion estimates. Matches of these motion estimates are then used as a basis for identifying corresponding targets in adjacent scans.

4.2 Tritech Micron DST Image Sonar

For our MicroROV, the Tritech Micron DST (Digital Sonar Technology) Sonar shown in Fig.4.2, was selected to identify target fish and provide a distance measurement. It is a Forward-Looking 360 Degree Sonar and Sector Scan Sonar Modes have been designed as obstacle avoidance or object tracking. According to Tritech, it is currently the smallest digital CHIRP sonar in the world.[10]

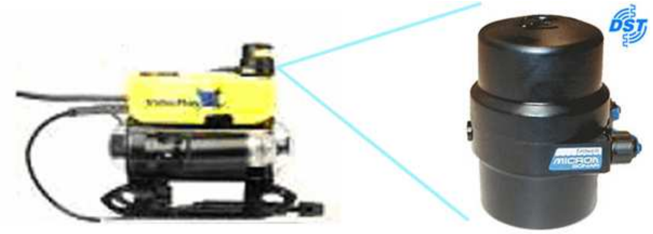
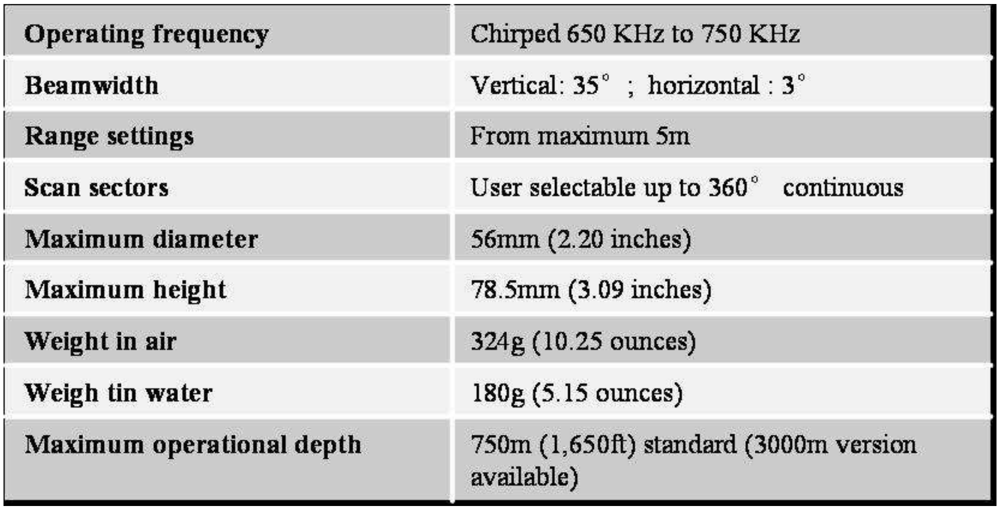


Figure 4.2: Tritech Micron DST Sonar. [10]

This Micron DST Sonar incorporates surface mounted digital electronics and many software features normally found only on full sized commercial sonar systems. It can be controlled by a customer supplied PC or laptop and it can be configured for either RS232 or RS485 protocols. The Micron Sonar also has a standard auxiliary port to allow it to interface with other Tritech sensors. Some major specifications are exhibited in table 4.1.

4.2.1 CHIRP Technology

The Tritech Micron DST sonar applies a novel CHIRP technique, where CHIRP stands for Compressed High Intensity Radar Pulse. The techniques dramatically



Operating frequency	Chirped 650 KHz to 750 KHz
Beamwidth	Vertical: 35° ; horizontal : 3°
Range settings	From maximum 5m
Scan sectors	User selectable up to 360° continuous
Maximum diameter	56mm (2.20 inches)
Maximum height	78.5mm (3.09 inches)
Weight in air	324g (10.25 ounces)
Weigh tin water	180g (5.15 ounces)
Maximum operational depth	750m (1,650ft) standard (3000m version available)

Table 4.1: Specification of Tritech Micron DST Sonar

increase the ability of discrimination between closely spaced targets by improving the range resolution of the fixed-frequency conventional sonars.[10].

In conventional monotonic techniques, an acoustic pulse consists of an on/off switch modulating the amplitude of a single carrier frequency, (as shown in Fig. 4.3 left). To get enough acoustic energy into the water for target identification and over a wide variety of ranges, the transmission pulse length has to be relatively long. The equation for determining the range resolution of a conventional monotonic acoustic system is given by:

$$Rangeresolution = (pulselength \times velocityofsound)/2 \tag{4.1}$$

When the smallest pulse length is 50 micro seconds and velocity of sound in water (VOS) 1500 meters/second (typical), the range resolution is 37.5mm. That is, if two targets are less than 37.5mm apart then they cannot be distinguished from each other.

CHIRP signal processing overcomes these limitations. Instead of using a burst of a single carrier frequency, the frequency within the burst is swept over a broad range throughout the duration of transmission pulse. This creates a 'signature' acoustic pulse; the sonar knows what was transmitted and when. Using 'pattern-matching' techniques, it can now look for its own unique signature being echoed back from targets, (see right side of Fig.4.3). In a CHIRP system, the critical factor determining range resolution is now the bandwidth of the CHIRP pulse. The range resolution is given by:

$$Rangeresolution = (velocityofsound)/(bandwidth \times 2) \tag{4.2}$$

The bandwidth of a typical Tritech CHIRP system is 100 kHz. With velocity of sound in water (VOS) 1500 meters/second, our new range resolution = 7.5mm. This time, when two acoustic echoes overlap, the signature CHIRP pulses do not merge into a single return. The frequency at each point of the pulse is different, and the sonar is able to resolve the two targets independently.

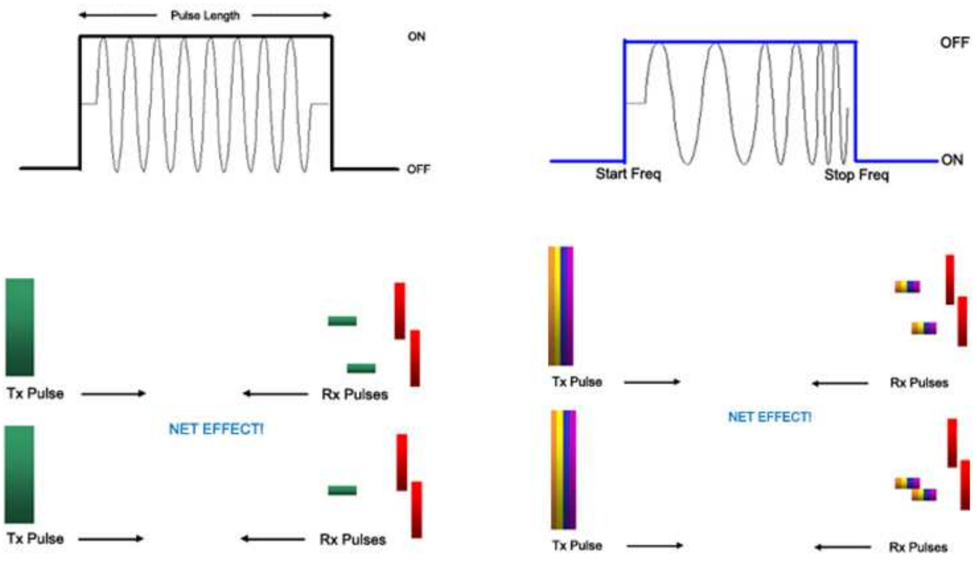


Figure 4.3: Acoustic pulse principle for sonar, at left is a monotonic technique, and at right is CHIRP techniques.[10]

4.3 Sonar Image Processing Algorithm

4.3.1 Sonar Image Analysis

The data returned from a sonar scanning is mapped against a color scale and plotted to create an image. These visual images provide us with sufficient data to identify and measure the position and orientation of a target when used in conjunction with image processing methods.

Since the forward-looking image sonar we selected uses CHIRP technology, the length of the acoustic pulse no longer affects the amplitude of the echo on the sonar display. This remarkably increase the range resolution making the information of object's shape and outline more abundant and obvious in the image frames. In addition, CHIRP offers improvements in background noise rejection, as the sonar is only looking for a swept frequency echo, and removes random noise or out-of-band noise. These advantages are of great benefit to sonar image processing for object identification. Like any acoustic sonar, this DST image Sonar reflections of isolated small objects give no indication of shape or attitude. When very close to large objects or in a depression in the seabed, the viewing range of the sonar may be severely limited. Very strong reflectors may give multiple echoes along a bearing line. Generally, interpretation of sonar data develops with experience. Scanning an unrecognizable target from one direction may be quite easily identified from another. Multiple echoes can be identified by being equispaced in range.

Underwater, sound transmission is limited. This is most notable in useable ranges. The usable range of high frequency sound energy is greatly reduced by seawater. Therefore, a tradeoff exists between higher resolution images produced by a high frequency sonar and the longer range provided by a low frequency sonar. In our work, as the visibility of the camera is just less than 2 meters in water, the minimum range (5m) with higher frequency of this sonar is enough for tracking task and produces the higher image quality and reliability.

The Imaging Sonar only produces a planar view that does not show how high an object is. This means that two objects which are shown in the same location on the horizontal plane could be at completely different heights in the vertical plane (seeing Fig.4.1). In our project, the integrated camera vision system provides the target height. Another drawback of the imaging sonar is that rough seabed textures can blot out smaller objects completely. Again the vision system can be used in lieu of the sonar for object recognition. Hence the vision system and image sonar sensor are complementary in identifying target and state estimation for the fish tracking.

4.3.2 Image Processing

High resolution sonar provides high-quality acoustic images allowing the classification of objects of the underwater environment. Multi-beam sonar images can be very noisy, due to reverberation from the seabed, surface or water column. A Median filtering is applied to reduce this affect. The objects which we wish to track have a higher reflectivity property than the surrounding environment, so measurements can be obtained by thresholding the sonar image intensity. The optimal filter designed for such an image does not look like a known low-pass filter but strongly depends on the principle of the sonar image formation in terms of size and coefficient values. Morphological filters not only produce snowless luminance image but also enlarge the strong refection region that can be target object. Several segmentation methods have been proposed [54]. A classical technique is based on thresholding [65], nevertheless as far as sonar images are concerned, simple threshold selection is not trivial. At last, an object recognition scheme is used to help identify and confirm the target fish.

Base color images are transformed into greyscale frames (introduced in chapter 2); the expression $f^k(x, y)$ denotes the value of a filtered image at pixel location (x, y) for the k^{th} frame of a video sequence. The image domain is D_f.

Special Thresholding

In our project, the sonar is used as the complementary sensor to provide the target distance for video and assist video identify the target fish. Since the underwater valid range of video camera is less than two meters, and yet sonar minimum measurement range is 5m, the sonar data of more than two meters can be neglected (threshold is T_1). Also, the data less than 10 centimeters cannot be calculated due to the sonar head surface reflection (threshold is T_2). Considering that a solid object should have stronger reflection, the value of data less than a threshold (T_3) will also be neglected.

$$f^k(x, y) = \{0, |x > T_1, x < T_2, f < T_3\} \tag{4.3}$$

Modified Median Filter

The median filter is used to remove isolated spurious measurements (spikes or dropouts) without disturbing the rest of the data. This approach gives us control over the amount of smoothing, noise reduction, and data enhancement that is achieved.

The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) The kernel is usually square but can be any shape [31]. The 2-D median filter can be denoted below. In our project, we use a $3 * 3$ window median filter.

$$f^k(x, y) = Med_{-W}\{f_{xy}\}; \text{where } W \text{ is an } n * n \text{ window} \tag{4.4}$$

The above median filter removes one pixel wide spikes and dropouts but does not remove spurious data that occupies a larger region. Using a larger median filter would remove larger regions but would also eliminate important details in the image. To get around this problem, a modified median filter is applied, removing only valleys, but leaving peaks intact. The idea is that peaks are likely to correspond to a point of interest whereas valleys are mostly drop outs or arise due to speckles. Also, peaks due to noise generally tend to be much lower and will be removed by filling up the valleys between them.

Morphological operators

The operations of dilation and erosion are fundamental to morphological image processing [30]. Effectively the dilation operator enlarges bright regions of a grayscale image, while the erosion operator shrinks them. But they are absolutely not a reversible operation. The morphological operators for erosion (Θ) and dilation ($\oplus$) are defined as follows:

$$
\begin{aligned}
(f \oplus q)(x,y) &= max\,\{f(x+m,y+n)| \\
&\quad (x+m,y+n) \in D_f \text{ and}(m,n) \in D_q\} && (4.5) \\
(f\Theta q)(x,y) &= min\,\{f(x+m,y+n)| \\
&\quad (x+m,y+n) \in D_f\text{and}(m,n) \in D_q\} && (4.6)
\end{aligned}
$$

These morphological filters require a structuring element, q, with domain D_q. For this work, structuring elements were chosen with a disk of radius 3 pixels.

Erosion and dilation operations are used to compute the morphological gradient and to create the snowless luminance image. The snowless image results from the application of a successive morphological operation called *opening*. This operation removes small speckles, floater, algae, etc. from an image, and is defined as:

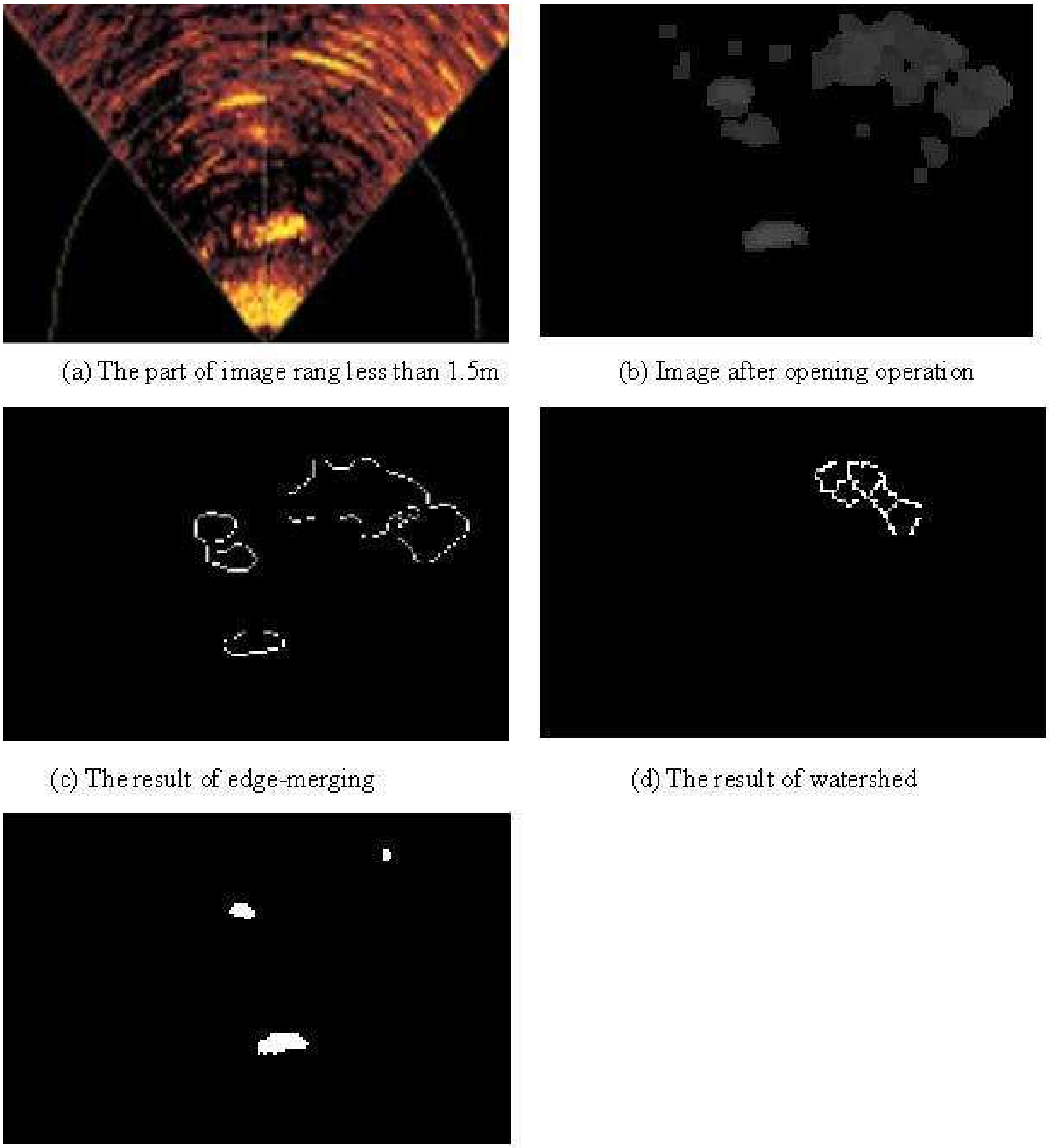
$$(f \circ q)(x,y) = (f\Theta q) \oplus q \qquad (4.7)$$

Region Grouping

The objective of segmentation is to partition an image into regions. Segmentation algorithms generally use topological clustering to group filtered pixels into sets. Typically, topological clustering methods identify either edges or regions. Edge-based methods distinguish between nonboundary pixels and pixels along the target boundary. Region-based methods classify pixels as members of either the closed set of target pixels or as members of the complementary background pixel set.

In our sonar images, we have tested several clustering techniques, such as convex edge-merging methods, and watershed methods.[65].It was found that a fast, reliable and efficient method was to simply cluster neighboring pixels above or below a selected threshold [68].In our work, the threshold is the eighty percent of the maximum intensity value.

Fig.4.4 shows the results of the edge-merging method, watershed method and thresholding segmentation method respectively. The edge-merging method can obtain the approximate edges of different regions, but the edges cannot be used



(a) The part of image rang less than 1.5m

(b) Image after opening operati on

(c) The result of edge-merging

(d) The result of watershed

(e) The result of threshold segmentation

Figure 4.4: The results of different segmentation methods.

as the basis to identify the target fish. The watershed method has no ability to find the object and extract the useful region from image. The thresholding region segmentation is common and easy to implement. However, it provides a valid segmentation noting that there are other undesirable regions that still remain but which can be removed using an object recognition scheme in the next step.

Object identification

In general, the above steps lead to the rejection of most background and small objects. Only several slightly larger objects with strong reflection remain in the image. However, even if only one object region remains, we still require an object identification algorithm to differentiate and confirm the target fish from background and other underwater animals.

To distinguish among multiple target regions and to reacquire targets following out-of-frame events, position related statistics need be computed for each segmented region. Other judgment criteria come from user experience to aid in discrimination over short time intervals. The algorithm is described as follow:

- Decide a reasonable area according to the fish position in former sonar image frame and in the current video image frame. We only consider those objects that fall into this region and neglect outsider areas.

- Sometimes, fish can give multiple echoes along a bearing line, which cause some false objects in sonar image frames. These false objects can be identified by being equally spaced in range. Therefore, the data will be discarded behind the nearest target along equivalent space of scanline.

- The size of a fish closer to ROV should be larger, so too small size segmentation regions will be abandoned.

After identifying the target fish, its range and bearing are extracted from the scanline through the central of this area.

4.4 Results and Analysis

Sonar data of a fish was taken in Paradise Lake, Ontario, Canada by using a VideoRay ProIII MicroROV. The sequences of images were obtained using the Tritech Micron DST forward looking Image Sonar which was fitted vertically on the frontal top of the VideoRay. It was scanning forwards using its smallest range of 5 m in a 90 degree scan sector at an operating frequency of 700 kHz.

The sonar data processing algorithm was applied to each frame of each sequence. An example of a typical sonar image being processed is shown in Fig.4.6. The fish

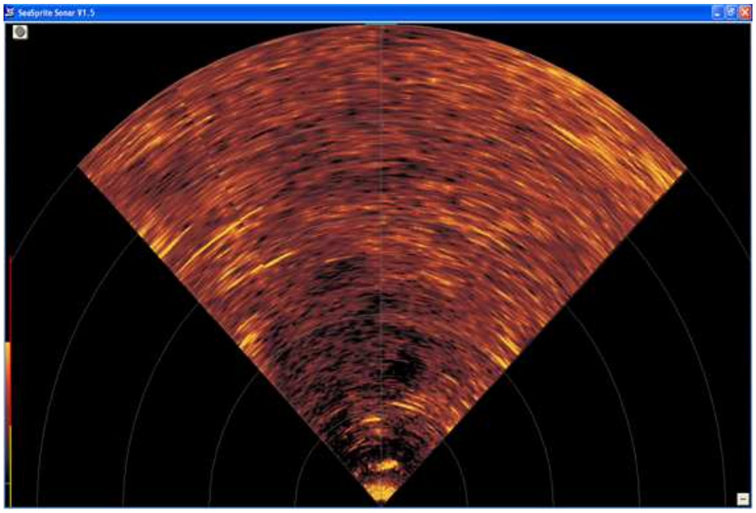


Figure 4.5: Original sonar image

recognition results from a sequence of sonar image frames are shown in Fig.4.7. The results indicate that the sonar image processing method provides sufficient relative range and orientation estimates for fish tracking under some situations.

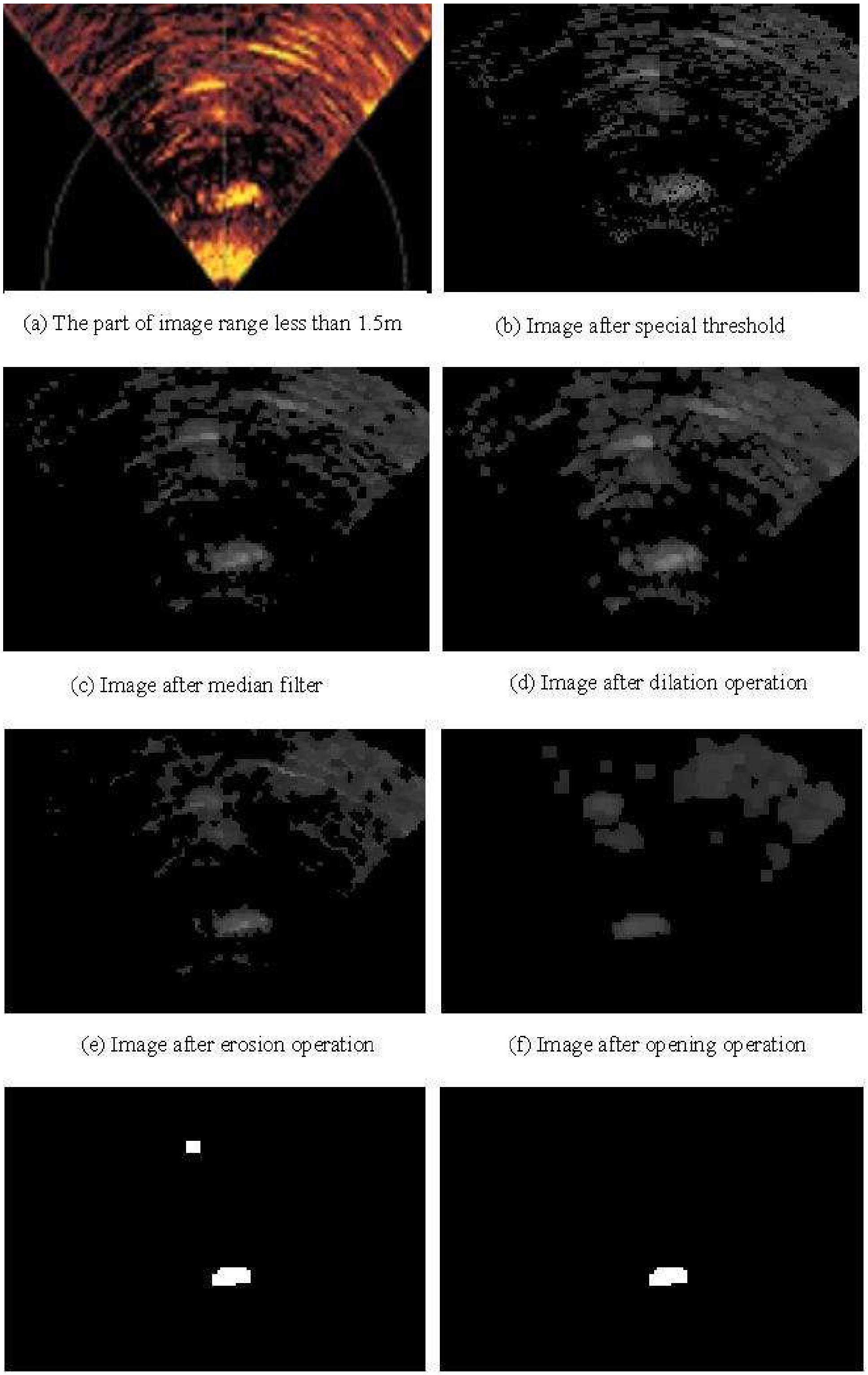
However, since the fish body is deformable and it is frequently swimming in a complex underwater environment, it should be noted that the following situations will lead to false object recognition and failure of the image processing algorithm:

- The intensity of reflection of fish is weak, i.e. less than 80 percent of the maximum value among all pixels.

- The size of the target fish in the reflected image is smaller than, or close to, other objects. For example, see the top image of Fig.4.8.

- If The horizontal positions of objects that have obvious segmented region are much closer, even the real range of these objects are in different, (seeing Fig.4.8(b), there are two obvious objects along the middle of vertical line in the image frame that mens very small horizontal distance, but they are in very different vertical position in image frame that means lager vertical distance, it is difficult to decide the target by using object identification algorithm.

- When the fish wanders in the underwater bottom, due to the strong reflection of coarse and uneven floor, recognizing it is extremely difficult, are shown in bottom image of Fig.4.8.



(a) The part of image range less than 1.5m

(b) Image after special threshold

(c) Image after median filter

(d) Image after dilation operation

(e) Image after erosion operation

(f) Image after opening operation

(g) Image after region-based cluster

(h) Result of object identification

Figure 4.6: Sonar image processing results

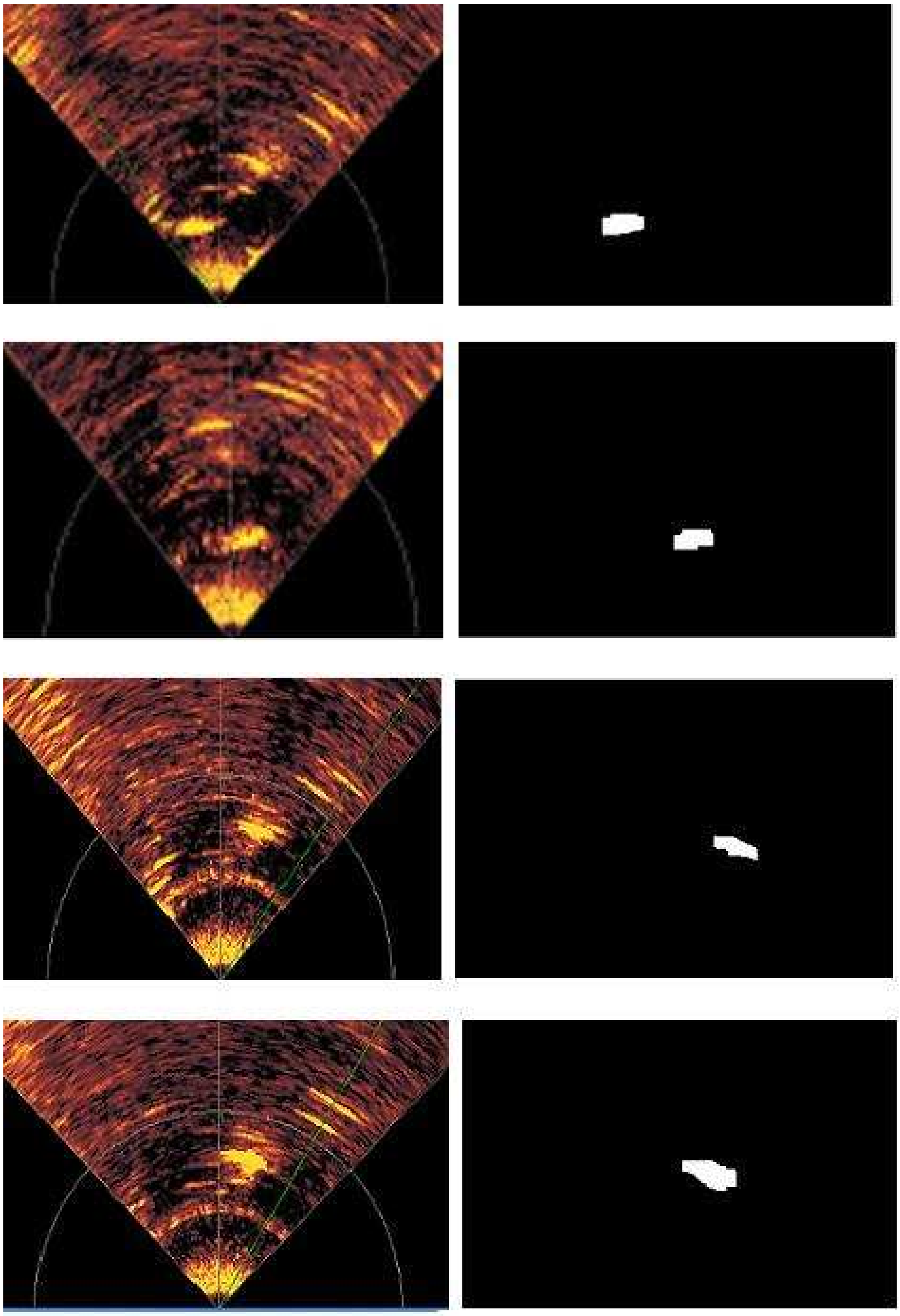


Figure 4.7: The results in a sequence of sonar image frames.

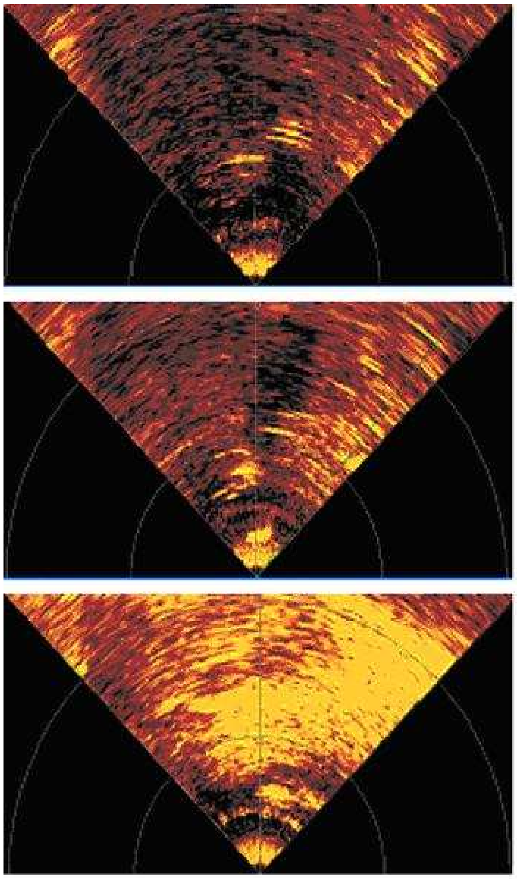


Figure 4.8: The examples of some situations that cannot recognize target fish.

4.5 Summary

In this chapter, the subsea sonar and forward-looking image sonar principle are briefly introduced. To accompany the MicroROV, the Tritech Micron DST (Digital Sonar Technology) Sonar was selected. Also, computer vision processing methods used to recognize the target fish were presented. The simple region clustering method successfully segments object regions from background after implementing a sequence of basic image processing methods: special threshold, media filter and Morphological operators. Identification schemes were used to decide the target fish location. Although the sonar image algorithm is shown to be successful under some conditions, it is not guaranteed to work in all conditions as described in section 4.4.

Chapter 5

State Estimation

5.1 Introduction

After recognition of a target fish from images, the next step in the system is to determine the system state that can guide the AUV to autonomously follow the fish. To do so, it is necessary to find some method of relating the range, bearing and depth of the fish to the AUV. This information is needed so that the AUV can effectively keep the underwater track within its field of view.

The image capturing devices, i.e. the video camera and sonar, are fixed on the underwater vehicle. Therefore, the geometric centre of an image can be defined as the relative position of the AUV. Once the location of target in the image plane is determined by calculating the centroid of object based on the coordinate axis, the relative position to the AUV coordinate axis can be obtained by a coordinate translation.

When all processing has been completed, these results of state estimation then are input to a control algorithm that will be developed in future work for positioning the AUV over the track. The general aim is to keep the centre of mass location within a small region about the centre of the image, in order to keep the fish in the optimum field of view.

5.2 Object Location in Image Plane

The location of the target in a 2-D image provides a position and orientation measurement in the camera reference frame. The most simple method used for determining various attributes of objects' position in an image, is by determining the location of the centre of mass or centroid of the object. [6].

We define the characteristic function of an object in an image to be

$$I[u,v] = \begin{cases} 1 & \text{for points on the object,} \\ 0 & \text{for background points.} \end{cases} \tag{5.1}$$

The area is given by the object:

$$A = \sum \sum I[u,v] \tag{5.2}$$

The centre of mass, denoted by $(\bar{u}, \bar{v})$, then the corresponding discrete equations is given:

$$\bar{u} = \frac{\sum \sum u^2 I[u,v]}{\sum \sum I[u,v]} \tag{5.3}$$

$$\bar{v} = \frac{\sum \sum v^2 I[u,v]}{\sum \sum I[u,v]} \tag{5.4}$$

In image processing calculation, the pixels are located according to the coordinates that the origin start at the top left of picture, therefore the centre point (x, y) is required transforming to the image reference frame (the origin is in the centre of picture). The formula is described:

$$\begin{cases} x = u - \frac{m-1}{2} \\ y = -\left(v - \frac{n-1}{2}\right). \end{cases} \tag{5.5}$$

where, m, n is the image dimensions $m * n$.

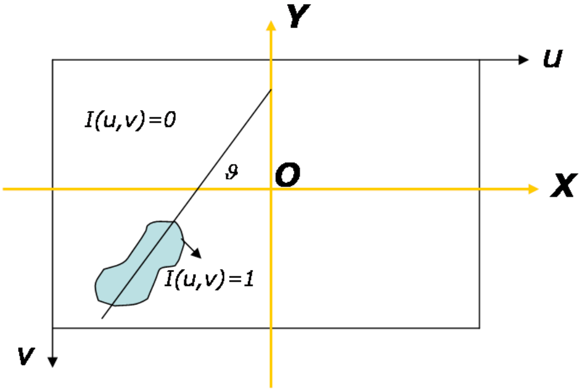
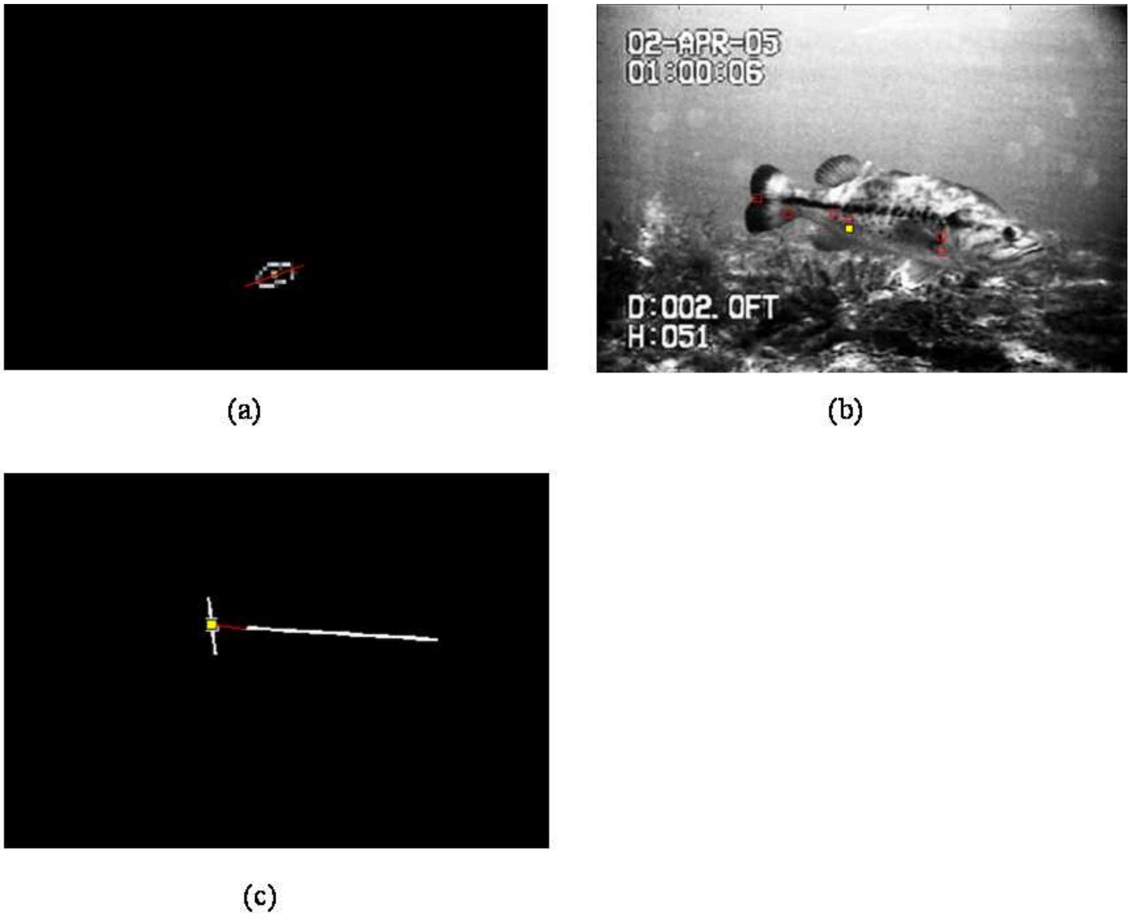


Figure 5.1: The object coordinates in the image frame

In the sonar image, the fish's position is the location of the centroid of object area. In the video image processed with SIFT based object recognition, the fish's position is defined as the center of the keypoints of interest matched between frames.

In the video image processed by the feature extracted algorithm, the intersection of the extracted body and tail line areas are considered as the fish's current position. Since the point of intersection is near the tail centre, an offset along the body line is added, which moves the point to the approximate centre of fish body. All of the results are shown in the Fig. 5.2.



(a)

(b)

(c)

Figure 5.2: Fish position and direction in image plane. The square indicates the central point and the line represents the fish pose in image plane. The squares are the matched keypoints. (a) The result in sonar image plane. (b) The position result in SIFT based video image plane. Due to the isolated points, there is no direction that can be represented as fish pose. In (c) the results of describing the point of intersection of two lines as fish location and the slope of lines as the direction of fish in video image plane processed by feature extraction algorithm.

5.3 System State Estimation

5.3.1 The Position and Orientation Relative to Sonar

The sonar visual images are mapped by the strength of the return signals collected for each scan against a color scale and plotted. Given the position of the fish within the image, the relative range and bearing of the fish to Sonar can be calculated by the data of corresponding scanline that have the echo reflection off Object. Generally, a scanline includes numbers of bins to store the strength value of each echo through scan range (seeing Fig. 5.3) If we know which bin the object locates in, its range ρ is given:

$$\rho_s = S_r \times n_b / N_b \tag{5.6}$$

here, S_r is Range Scale(unit:m), n_b is the current bin number, N_b is the total bin number.

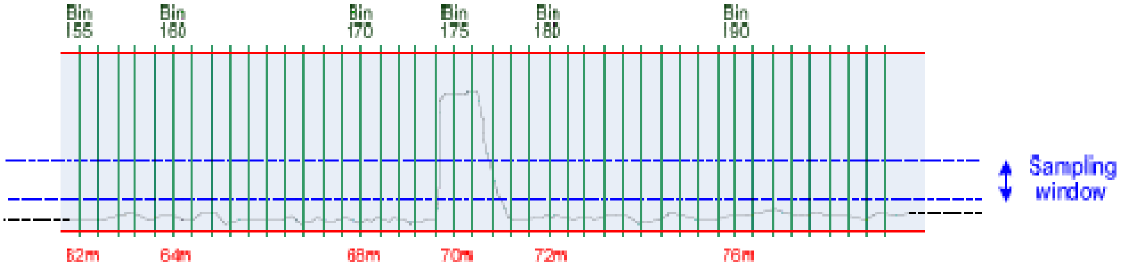


Figure 5.3: An example of Bin 175 including a object. [2]

Fig. 5.4 shows an example scanline with the sonar facing at a 90 degree angle. Given such scanlines with identifiable returns from objects, we can determine the bearing φ_s to such objects directly. The range and bearing can be changed into yaw, Y_s, and distance, X_s, in Cartesian coordinates.

$$\begin{aligned} X_s &= \rho_s \times \sin 270 - \varphi_s \\ Y_s &= \rho_s \times \cos 270 - \varphi_s \end{aligned} \tag{5.7}$$

5.3.2 The Position Relative to Video Camera

The relative position between camera and target is calculated by perspective geometry projection relation. Fig. 5.5 shows an example of camera geometry projection: the position of a point (P) on the space target is defined as $P(x, y, z)$, and it is taken

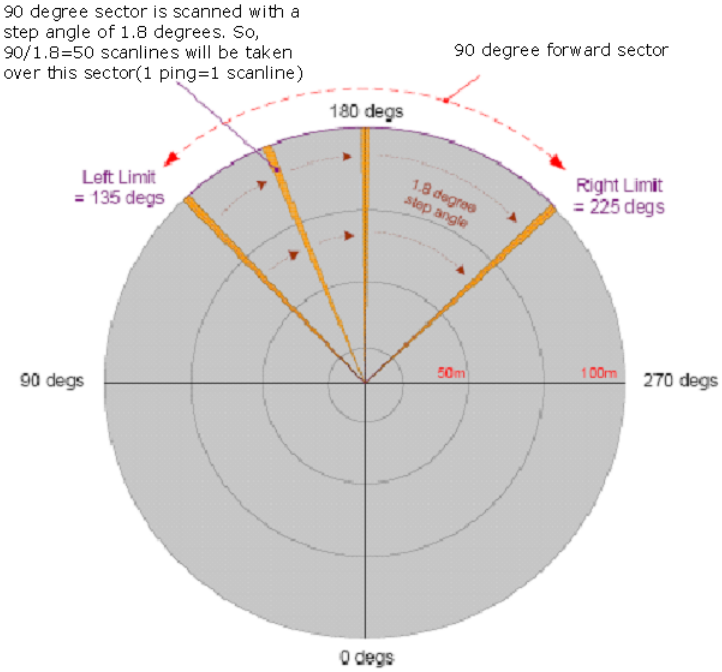

90 degree sector is scanned with a step angle of 1.8 degrees. So, 90/1.8=50 scanlines will be taken over this sector(1 ping=1 scanline)
90 degree forward sector
180 degs
Left Limit = 135 degs
Right Limit = 225 degs
1.8 degree step angle
90 degs
50m
100m
270 degs
0 degs

Figure 5.4: An example to scan a 90 degree.[2]

in the camera's image, defined as $P(x_i, y_i)$ based on camera coordinates. Through the geometry projection relation, if know the location in image,$P(x_i, y_i)$, and the distance, D between target plane and camera (provided from sonar), we can calculate the relative position of space target to camera. Our video camera has been calibrated, camera's real focus f (unit, mm) and the scaling factor k ($pixel/mm$) are known, and the image plane focus is given:

$$d = kf \tag{5.8}$$

The target's position in the image plane (x_i, y_i) , the relative depth, Z_v, and the yaw, Y_v, between the camera and target is given at:

$$\begin{aligned} Y_v &= \frac{y_i}{kf} \cdot D \\ Z_v &= \frac{x_i}{kf} \cdot D \end{aligned} \tag{5.9}$$

In addition, considering the image results after being processed by the first algorithm, *feature extraction algorithm*, the fish tail and body-line have been extracted. Among useful techniques, *feature based scaling*, can be used for approximately calculating the relative distance between the fish and camera for monocular vision system.

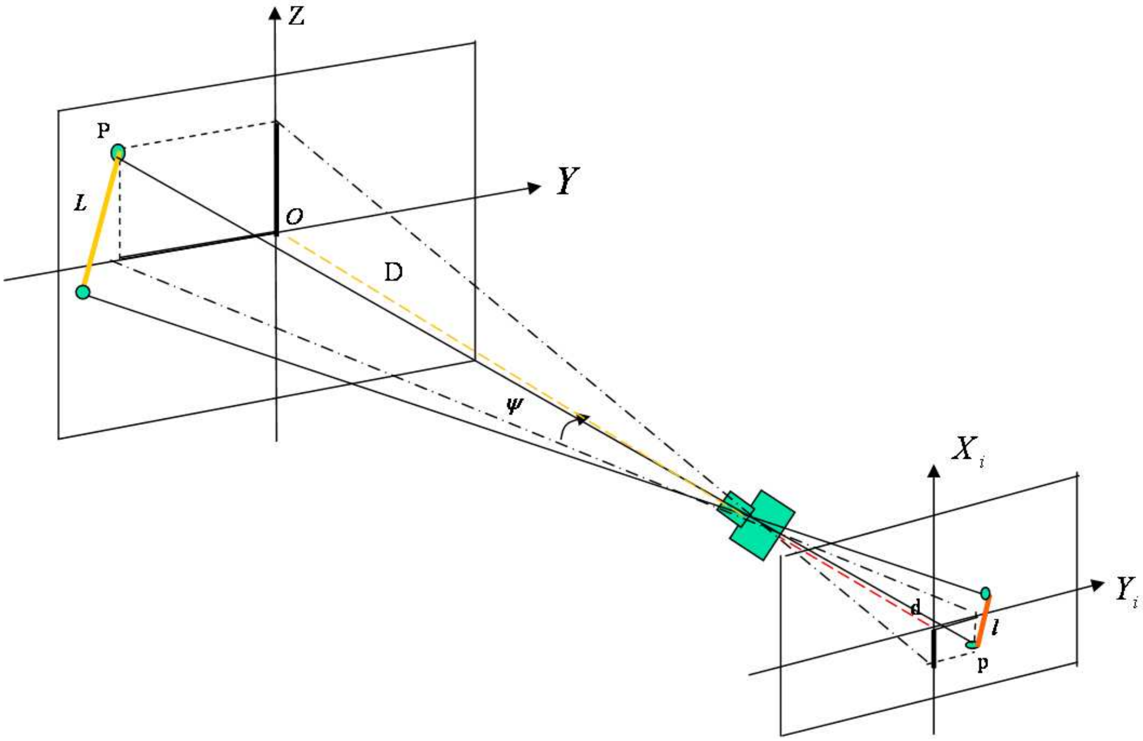


Figure 5.5: Camera geometry projection

Feature based scaling produces a relative range measurement with arbitrary length scale. An external reference measurement is required to fix the units of the arbitrary length scale. Typically, by the statistical data, the most size of target fish, Large Mouth Bass, is used to relate a relative initial length to absolute metric units.

Because feature extraction generally suffers from light intensity changes, the fish tail swaying and the body deforming, to help remedy such disturbances, we use tail length and body-line area to obtain two relative distances of image and object respectively and then combine them by a simple Kalman filter that weights the fusion based on variance. Compared with sonar data, the accuracy of this approximate distance estimation is low but it is still helpful when sonar data is lacking.

Relative feature scales are based on pixel measurements in the image plane. In the fish tracking system, pixel width is used to calculate the tail length. From the feature extraction, we have known the line representing the tail, but sometimes, the tail cannot be wholly extracted due to the fish being close to bottom or seaweed. We select the tail length,l_t, from the topmost point and the intersection of the extracted body and tail line.

Besides, since the fish body rotates and deforms or the light intensity changes, the body line is not always kept in full length, so we choose a section at the centre of the body-line segmented region and use its area as the valid value to estimate relative distance. The area value, s, is simply calculated by summing pixel's numbers in this section.

$$\begin{aligned} D_t &= \frac{kf}{l} \times L \\ D_b &= \frac{kf}{s} \times S \end{aligned} \tag{5.10}$$

here, f (mm) is the real focus is , k is the scaling factor that transforms f into the image plane.

If D_t is the depth calculated by the length of tail with variance σ_t^2, and D_b is the depth calculated by the width of body line of fish with variance σ_b^2, then D_k is the optimal depth calculated by the Kalman filter equations.

$$D_k = D_t + K(D_b - D_t) \tag{5.11}$$

$$K = \frac{\sigma_t^2}{\sigma_t^2 + \sigma_b^2}; \tag{5.12}$$

The relative depth, Z_v, and the yaw, Y_v, between camera and target is calculated using the above Eq. 5.10, using D_k instead of D.

Errors in the feature-based range estimate result from difficulties in precisely determining target lengths within images, and from the uncertain size of the target. Assuming the covariances of the image plane measurements are σ_u^2, σ_v^2 , σ_l^2 , σ_s^2 respectively, and the covariance of the size of fish is σ_L^2, then we can compute their relative variance $\Sigma\psi$, ΣD, ΣZ, and $\Sigma\rho$ according to the error propagation law:

$$\Sigma = \nabla f C_f \nabla f^T \tag{5.13}$$

Here,∇f is a Jacobian matrix and $C_f$is a function of covariance σ^2.

5.3.3 The System State Estimation

Our current VideoRay has three degrees of freedom that can be controlled, therefore, only three coordinates are necessary to describe fish position in the body frame: distance to fish, Xb , yaw, Yb, and depth, Zb. The VideoRay system reference frames are shown in Fig. 5.6. The video camera with assistant sonar just satisfies the requirement of controller in VideoRay. The translation is very simple from a video or sonar frame into the system frame.

$$P_b = T + P_i \tag{5.14}$$

Here, $P_b = [X_b, Y_b, Z_b]^T$ is system frame, $P_i = [X_i, Y_i, Z_i]^T$ $(i = s/v)$ represent sonar or video frames. T is the translation Matrix.

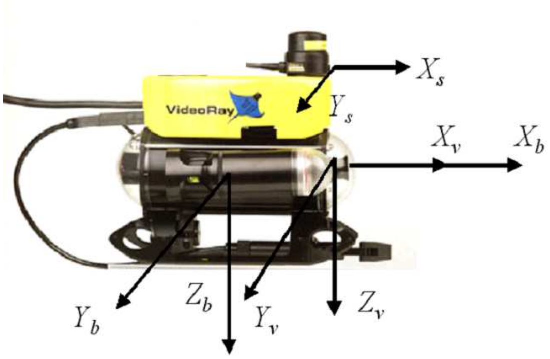


Figure 5.6: VideoRay reference frames: The "b" subscript indicates the body-fixed frame, while the "v" subscript indicates the camera frame, and "s" subscript indicates the sonar frame.

Since the time to generate each complete image frame from the sonar takes much longer than each of video frame captured by video camera, the calculation of the yaw,Y_b , and the depth, Z_b will come from video image processing and the value of distance form sonar will keep unchanged during the sonar form new image. When sonar data have been updated, the yaw and depth will be calculating according to this new distance until next updated data arriving. In addition, in lacking the sonar data, if we are lucky to obtain the approximate depth from video image by the feature extraction algorithm, we also can use the results of monocular camera itself, but this algorithm works well based on many assumptions, and does not reliably and efficiently extract the fish every time, while the SIFT approach based algorithm cannot estimate the distance between target and the AUV, it must depend on an assistant equipment such as sonar to provide the relative distance.

5.4 Off-line Experimental Result and Analysis

Tracking the Large Mouth Bass in the real lake, Paradise Lake, Ontario, Canada, by a MicroROV, VideoRay ProIII, is extremely difficult, due to the limitation of time, environment and equipment. We only obtained two experimental data that is valid and useful for off-line video and sonar image processing. In the first experiment, Video data images of a Large Mouth Bass were acquired in 2005, using the WDCC-6300 CCD color camera installed on human driven VideoRay ROV. Images were of dimensions $480*640$, and were grabbed at a frame rate at 20Hz. There was not any sonar mounted on VideoRay at that time. In the second one done in 2007, the aim was verify the based SIFT object algorithm and sonar data processing. The image were grabbed at a frame rate of 30Hz and dimensions were $240 * 320$. The Tritech Micron DST (Digital Sonar Technology) image Sonar was installed vertically on the frontal top of the VideoRay. As an auxiliary range measurement, since the shooting angle of the camera is less than 90 degree and the distance to target in which camera can capture a clear picture is less than 2 meters under 1.5 meters depth of the lake, the Tritech Micron DST sonar scan range is set its smallest range, 5 meters, the scan area is 90 degree area, and the time of finishing the whole scan area is 4 seconds.

Fig. 5.7 shows the results of the relative position between ROV and the target fish calculated from the sonar and the video image processing after the SIFT object algorithm. Because the captured images are blurry and small, the feature extraction algorithm is not applicable for texture segmentation in most sequence of frames, but only applies to some individual image frames. So the results did not include these data.

The relative yawing,Y_b, and distance, X_b, between the target fish and the ROV is calculated for 15 successive images provided by the sonar in 60 seconds and its Results are displayed in Fig. 5.7 (a3) and (a4). In the 3rd and 4th sonar images, due to the fish being close to bottom, rough seabed textures can blot out the target completely and the fish cannot be found out. In the 8th and 9th sonar image, the fish cannot be showed either because the fish turns to swim towards ROV, which can be seen from camera images. As for these lacks of data, the output value of the sonar will keep the same as the value in former image in the 8th, 16th, 36th and 40th seconds. The depth, Z_b, and yawing, Y_b, of from relative to the ROV is calculated from 200 successive frames taken by the video camera across a time span of 60 seconds, the depth, Z, and yawing, Y_v, relative fish with ROV is calculated. Results are displayed in Fig. 5.7 (a1) and (a2). Star marks are the values at every second. Beginning from the 52nd second , the fish cannot be taken in the image because it swims up exceeding the shoot area. When combining with the depth curve, at the 52nd second, the fish is about 0.8 meters over ROV. But at this time, the sonar still has the ability to find the fish. AUV will be able to adjust its pose to track the fish again according to the sonar data and the former information from video frames.

From video streams in the first experiment, we have the ability to obtain the relative position by using the feature extraction algorithm. Since there are not any sonar data, for the monocular system, we only use the feature based scaling method to estimate the approximate distance from the camera to the target. Statistically, the average length of Large Mouth Bass is 30cm, and the corresponding length of its tail is 9cm with $\sigma^2 = 0.5cm^2$ and the corresponding width of fish body central pattern is 0.7cm with $\sigma^2 = 0.05cm^2$. Fig. 5.8 shows the relative distance estimation results and the corresponding propagation errors from using 1) feature scaling the fish tail, 2) feature scaling of the body line, and 3) fusion of the two previous results via the Kalman filter. The fusion of distance measurements from both features of

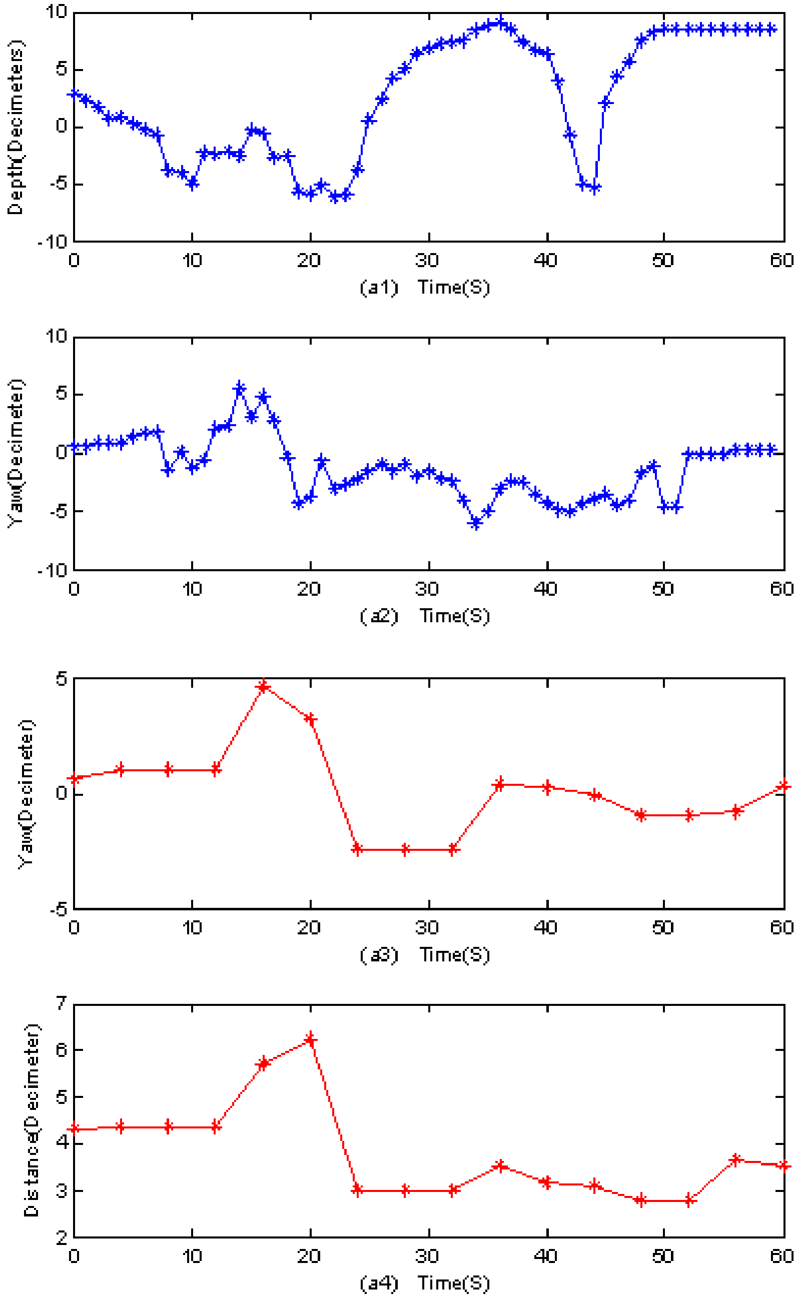


Figure 5.7: The results of the relative position estimation in 60 seconds tracking. (video data are shown in figure (a1)and (a2); sonar data are shown in figure (a3) and (a4))

fish tail and body line assist in decreasing errors, while the monocular vision system does have inherent difficulties in predicting camera's depth of field.

The relative position between the target fish and the ROV is calculated for ten successive images taken across a time span of 2 seconds. Results are displayed in Figure 5.8. At present, there is no truth data for comparison. Error results are based on the theoretical calculations of propagation error. Fig. 5.9 (b) shows the calculated depth of fish along the Z_b axis. Because the size of this species of adult fish is undetermined and can only be obtained from statistical data, higher error in the depth estimation occurs. This will affect the accuracy in estimating relative vertical and range positions. When the relative depth calculated has maximal value 0.1m, the propagation error is 0.02m. While the range position calculated from the image has a maximal value of 0.95m, the propagation error is 0.2m, seen from Fig. 5.9 (a). The result of Yaw is shown in Fig. 5.9 (c), the maximum propagation error is about 0.018 at the real value of yaw close 0.2m. Although these errors are little bigger, as monocular vision system and only as assistance when lacking of sonar range data, these errors should be acceptable in the proposed fish tracking system. Moreover, it is not required to keep the high precise position between the target fish and the ROV all the time, what is needed the ability to track the fish moving, without disturbing it. Fig. 5.9 (d) exhibit the fish bearing results. The accuracy of bearing is higher, because its propagation error is affected by the distance of camera leans to image plan that is the system error and can be gain from camera system calibration, and by the error of the measurement distance of the target point to the origin in the image coordinates. In according to the error propagation law, from the Eq.5.10 and Fig. 5.9 (d2), the propagation error is very little.

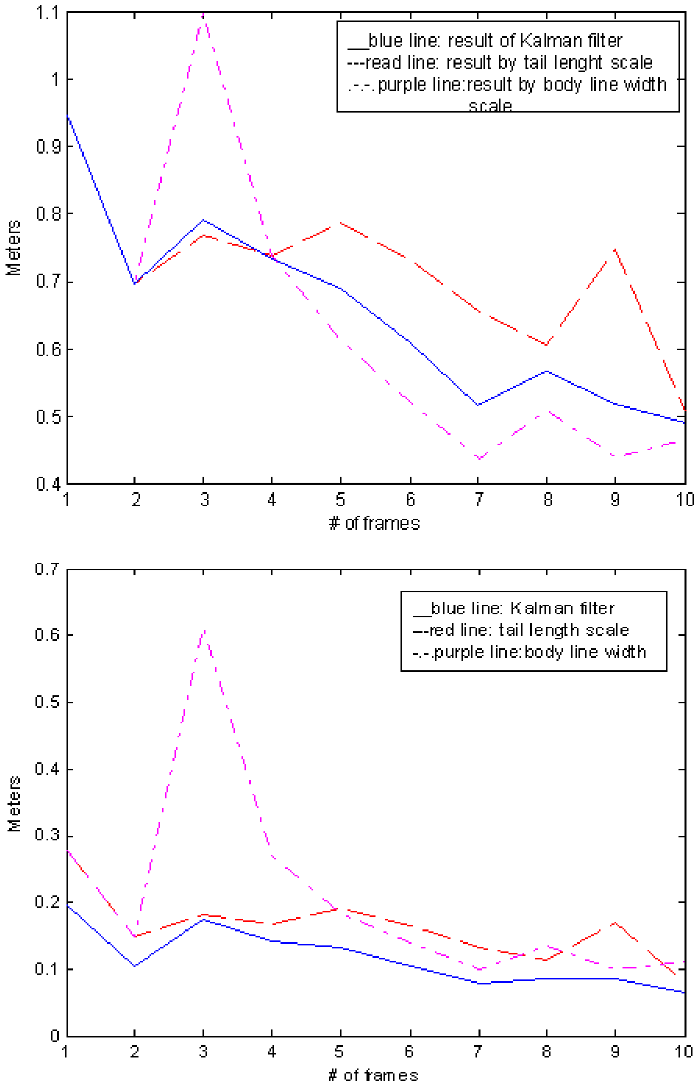


Figure 5.8: The result of depth estimation and propagation error by using feature scaling method and Kalman filter. Note: due to limitation of image processing, there is no clear body line in fist two frames; the depth value is equal to the depth value calculated from the tail length.

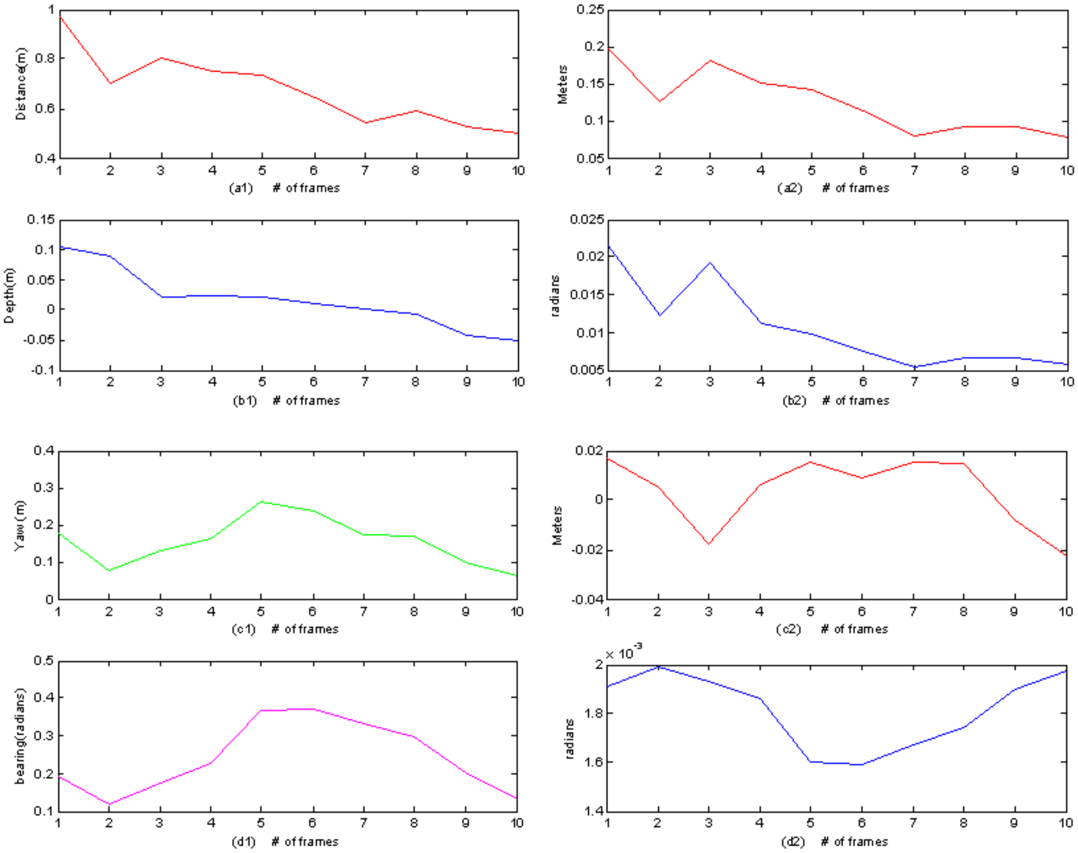


Figure 5.9: The vision sensor measurement result of the relative position at 2 second in ten successive frames. (a1): Distance, X_b; (b1): Depth, Z_b; (c1): Yaw, Y_b; (d1) bearing of fish, ψ. The right diagrams: the propagation error of three axes respectively.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this research, we presented the development and analysis of vision-processing methods for both video and sonar images, as well as state estimation methods for tracking a particular species of fish, the Large Mouth Bass, in natural underwater environments where the engineer cannot control or modify the visual appearance of the target or the surrounding scene.

With respect to image processing for identifying fish from the cluttered underwater background in camera video frames, we introduced two valid and efficient segmentation and recognition algorithms : one is feature extraction and the other is SIFT based object recognition. These algorithms extracted fish features from video images and calculated the relative position and orientation between the target fish and AUV.

A forward-looking image sonar was used to provide measurements of relative distance, while at the same time assisting in target fish identification. The sonar images were processed used computer vision approaches to allowed recognition of the fish and determination of the relative position and orientation.

A state estimator was used to combine the relative position and orientation information obtained from the two sources: camera video and sonar. Using a coordinate transformation, the relative range, bearing and depth between fish and AUV were obtained. In addition, the estimator can analyze some situations to decide which of the two devices will produce measurements with higher associated confidence.

Using data from a natural underwater experiment, off-line experiments were conducted. The results obtained from experiment 1 include a series of frames that span 2 seconds in duration. The results demonstrate that the feature extraction algorithm successfully extracts target fish features under some assumptions. These assumptions include 1) only one fish is in each frame, 2) the fish tail and side are visible, and 3) the fish is not among seaweed. The feature based scaling method is able to estimate the position of the fish relative to the camera.

In a second experiment, when the fish does not swim fast, results demonstrated that the vision-processing method of both video and sonar consistently and successfully identified tracking targets and the state estimator correctly and reliably outputted the system state with no parameter-tuning by a user. This second experiment used a sequence of 180 video frames and 15 sonar frames that spanned a duration of 60 seconds. Although the slow speed of the sonar scan and the presence of noise in image frames affect the accuracy of state estimation, the errors are acceptable for fish tracking applications.

6.2 Future Work

This project presented research to enable vision-based state estimation for tracking target fish using video camera and sonar. Several work still needs to be done for further investigation and realization. This includes:

- The 3-dimension position and heading estimation obtained through the vision algorithm is uncertain due to noise and other imperfections. Also, the velocity that is important information for the controller could not be measured directly by video or sonar sensor. In future work, we consider using the extended Kalman filter that allows motion estimation, attaches uncertainty estimates to the components of the vector state, enables inclusion of a model of the vehicle dynamics, and provides a framework for fusing data from different sensors.

- The analysis of feature extraction algorithm described the limitations used in the fish-tracking system. Further research is required to improve image segmentation and recognition technology to make the fish-tracking system robust to situations in which these assumptions do not hold.

- Since the speed of sonar scan is slower, the estimation may not be as adaptable to some real-time applications. A high-speed/high-accuracy image sonar may be required. Also, if there is more space, another video camera could be mounted to realize the stereo vision. Through stereo vision techniques, the distance between object and video camera can be obtained.

- Develop an efficient and robust controller integrated with visual information to implement the closed-loop real-time control. This visual servo system has ability to guide the AUV to autonomously follow the fish and monitor its behavior in natural underwater environment.

Bibliography

[1] http://www.codersource.net/csharp conversion color gray image.aspx.

[2] file 'decoding sonar scanline data' of tritech international limited. xi, 77, 78

[3] http://en.wikipedia.org/wiki/kd-tree.

[4] http://en.wikipedia.org/wiki/side-scan sonar. 58

[5] http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html. 27

[6] http://homepages.inf.ed.ac.uk/rbf/cvonline/local-copies/owens/lect2/node3.html. 74

[7] http://www.answers.com/topic/sonar?cat=technology. 58

[8] http://www.dosits.org/gallery/tech/osf/esm1.htm. 59

[9] http://www.rov.org/educational/pages/sonars.html. 5

[10] http://www.tritech.co.uk/products/products-micron-sonar.htm. xi, 57, 58, 60, 61, 62, 63

[11] O. Amidi, T. Kanade, and R. Miller. Vision-based autonomous helicopter research at carnegie mellon robotics institute (1991-1998). in m. vincze and g.d. hager, editors,. In *Robust Vision for Vision-Based Control of Motion*, pages 221–232. SPIE Optical Engineering/IEEE Press, 2000. 4

[12] V. Areekul, U. Watchareeruetai, K. Suppasriwasuseth, and S. Tantaratana. Separable gabor filter realization for fast fingerprint enhancement image processing. In *ICIP 2005. IEEE International Conference*, volume 3, pages III– 253–6, 2005. 21

[13] A. Balasuriya and T. Ura. Vision-based underwater cable detection and following using auvs. In *Oceans '02 MTS/IEEE*, volume 3, pages 1582–1587, 2002. 4

[14] D.H. Ballard. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 45:891–923, 1998. xi, 47, 48

[15] B.A.Moran, J.J.Leonard, and C. Chryssostomidis. Curved shape reconstruction using multiple hypothesis tracking. *IEEE journal of Oceanic Engineering*, October 1997. 61

[16] M. S. Bartlett. A comparison of gabor filter methods for automatic detection of facial landmarks. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 242, 2002. 19

[17] P.H. Batavia, D.A. Pomerleau, and C.E. Thorpe. Overtaking vehicle detection using implicit optical flow. In *Proceedings of the IEEE Transportation Systems Conference*, pages 729–734, 1997. 4

[18] J. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. of Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, June 1997. 45

[19] J. Canny. A computational approach to edge detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 8, pages 679–714, November 1986. 27

[20] M.J Chantler and J.P.Stoner. Automatics interpretation of sonar image sequences using temporal feature measures. *IEEE Journal of Oceanic Engineering*, 22(1):47–56, January,1997. 7

[21] M.J. Chantler and J.P. Stoner. Automatic interpretation of sonar image sequences using temporal feature measure. *IEEE Journal of Oceanic Engineering*, January 1997. 60

[22] C.Jennings, D.Murray, and J.J. Little. Cooperative robot localization with vision-based mapping. In *Proc. of Robotics and Automation, 1999 IEEE International Conference on*, volume 4, pages 2659 – 2665, 1999. 4

[23] C.Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997. 35

[24] C.Silpa-Anan, T. Brinsmead, S. Abdallah, and A.Zelinsky. Preliminary experiments in visual servo control for autonomous underwater vehicle. In *Proceedings of Intelligent Robots and Systems 2001, IEEE/RSJ International Conference on*, volume 4, pages 1824 – 1829, 2001.

[25] D.B.Westmore and W.J. Wilson. Direct dynamic control of a robot using an end-point mounted camera and kalman filter position estimation. In *IEEE International Conference on Robotics and Automation*, pages 2376 – 2384, 1991. 4

[26] A. Delopoulos, A.Tirakis, and S. Kollias. Invariant image classification using triple-correlation-based neural networks. *IEEE Trans. Neural Networks*, 5(3), 1994. 34

[27] E. Dougherty (ed.). Mathematical morphology in image processing. marcel dekker. 1992. 13

[28] S. Edelman, N.Intrator, and T. Poggio. Complex cells and object recognition. 42

[29] Y. Fan and A. Balasuriya. Autonomous target tracking by auvs using dynamic vision. In *Proc. of the 2000 International Symposium on Underwater Technology*, pages 187–192, 2000. 3, 5

[30] D. A. Forsyth and J.Ponce. Computer vision: A modern approach. In *Prentice Hall*, 2003. 12, 13, 14, 66

[31] R.C. Gonzalez and R.E.Woods. Digital image processing. In *Prentice Hall, upper Saddle River, NJ.*, 2nd ed. 2002. 15, 16, 17, 18, 65

[32] J.C. T. Hallam. Intelligent automatic interpretation of active marine sonar. *PhD thesis, University of Edinburgh, Edinburgh Scotland*, 1984. 7

[33] C. Harris. Geometry from visual motion. *In Active Vision. A. Blake and A. Yuille (Eds.), MIT Press*, pages 263–284, 1992. 35

[34] H.Tu, C.H. Chen, C.Y.Chen, and L.C.Fu. A system for 3-d target trajectory detection via stereo visual tracking. *Technical Report of Department of Electrical Engineering, National Taiwan University, Taipei, R.O.C.*, 1997. 4

[35] Reson Inc. Seabat 6012's operator manual. 7

[36] J.A.Catipovic. Performance limitations in underwater acoustic telemetry. *IEEE, Journal of Oceanic Engineering*, 5, No.3, July,1990. 3

[37] J.C.T.Hallam. Intelligent automatic interpretation of active marine sonar. *PhD thesis, University of Edinburgh, Edinburgh, Scotland*, 1984. 60

[38] D. Kocak, N. da Vitoria Lobo, and E. Widder. Computer vision techniques for quantifying, tracking, and identifying bioluminescent plankton. In *IEEE Journal of Oceanic Engineering*, volume 24(1), pages 81–95, 1999. 5

[39] K.Ohba and K.Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 19(9), 1997. 35

[40] K.Tanaka. Mechanisms of visual object recognition: monkey and human studies. *Current Opinion in Neurobiology*, 7:523–529, 1997. 35

[41] D.M. Lane, M.J. Chantler, and D.Dai. Robust tracking of multiple objects in sector-scan sonar image sequence using optical flow motion estimation. *IEEE journal of Oceanic Engineering*, January 1998. 61

[42] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 60(2), 2004. 36, 37

[43] L.Jin, X.Xu, S.Negahdaripour, C.Tsukamoto, and J.Yuh. A real-time vision-based station keeping system for underwater robotics applications. In *Proc. of OCEANS '96.MTWIEEE. Prospects for the 21st Century*, volume 3, pages 1076–1081, 1996. 3

[44] R. L.Marks, H. H.Wang, M. J.Lee, and S.M.Rock. Automatic visual station keeping of an underwater robot. In *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceeding*, volume 01.2, pages IY137–11/142, 1994. 3

[45] D. G. Lowe. Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, 21(2), 1994. 35, 36, 38, 39, 43, 50

[46] M.van Kreveld M. de Berg. *Computational Geometry (Algorithms and Applications).* Springer-Verlag, 2nd rev edition, 2000. xi, 45, 47

[47] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of CVPR '03*, June 2003. 35

[48] M. Minami, J. Agbanhan, and T. Asakura. Manipulator visual servoing and tracking of fish using a genetic algorithm. In *Industrial Robot*, volume 26(4), pages 278–289, 1999. 5

[49] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. In *PAMI*, volume 23, pages 349–361, 2001. 35

[50] M.Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 19(9), 1991. 35

[51] N.C.Hu, K.K.Yu, and Y.L.Hsu. Two-dimensional shape recognition using oriented-polar rrepresentation. *Opt. Eng.*, 36(10), 1997. 34

[52] N.J.Pioch, B.Roberts, and D.Zeltzer. A virtual environment for learning to pilot remotely operatedvehicles. In *Proc. of International Conference on Virtual Systems and MultiMedia*, volume 10-12, pages 218–226, Sep.1997. 2

[53] N.P.Papanikolopoulos and P.K.Khosla. Adaptive robot visual tracking", ieee transactions on automatic control. In *IEEE Transactions on Automatic Control*, volume 38(3), pages 429 – 445, 1993. 4

[54] N.R.Pal and S.K. Pa. A review on image segmentation techniques. *Pattern Recognition*, 1993. 64

[55] A. Ortiz, M. Simo, and G. Oliver. Image sequence analysis for real-time underwater cable tracking. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on*, pages 230–236, 2000. 4

[56] W. Patrick and A.F. Laine. Wavelet descriptors for moultrie solution recognition of hand printed characters. *Pattern Recognition*, 28(8), 1995. 34

[57] J.G. PAUL. Simulation and analysis of a digital focused beamformer for sonar. *PhD thesis, Department of Electrical and Electronic Engineering, Heriot-Watt University, Edinburgh*, June 1992. 59

[58] Y. Petillot, I.Tena Ruiz, and D.M.Lane. Underwater vehicle path planinf using a multi-beam forward looking sonar. In *OCEANS'98 IEEE proceeding*, pages 1194–1199, September 1998. 7

[59] H.H. Pien, D.E. Gustafson, and W.F. Bonnice. An auv vision system for target detection and precise positioning. In *Proc. of the 1994 Symposium on Autonomous Underwater Vehicle Technology*, pages 36–43, 1994. 2

[60] J. Rife. Automated robotic tracking of gelatinous animals in the deep ocean. *PhD thesis, Stanford University, Stanford,California*, December,2003. 1, 3, 5, 13

[61] R.Lu and Y. Shen. Noisy image segmentation by modified snake model. *Journal of Physics: Conference Series*, 48, Issue 1:369–372, 2006. 13

[62] R.N.Carpenter. Concurrent mapping and localization with fls. *1998 Workshop on Autonomous Underwater Vehicles, Cambridge, Massachusetts, USA*, 1998. 60

[63] R.Nock and F. Nielsen. Statistical region merging. In *IEEE Trans Pattern Anal Mach Intell*, volume 26(11), pages 1452–1458, 2004. 13

[64] N.S. Netanyahu R. Silverman A.Y Wu S. Arya, D.M.Mount. Distinctive image features from scale-invariant keypoint. *Pattern Recognition*, 13(2), 1981. xi, 48, 49

[65] P.K. Sahoo, S.Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. In *Computer Vision Graphics Image Processing*, volume 41, pages 233–260, 1988. 64, 66

[66] J. Santos-Victor and J. Sentieiro. The role of vision for underwater vehicles. In *Autonomous Underwater Vehicle Technology, 1994. AUV '94., Proceedings of the 1994 Symposium on*, pages 28–25, 1994. 4

[67] E. Trucco and A. Verri. Introductory techniques for 3-d computer vision. In *Prentice Hall*, pages 108–112, 1998. 13

[68] J.S. Weszka. A survey of threshold selection techniques. In *Computer Vision Graphics Image Processing*, volume 7, pages 259–265, 1978. 66

[69] X. Xu and S. Negahdaripour. Vision-based motion sensing for underwater navigation and mosaicing of ocean floor images. In *OCEANS '97. MTSOEEE Conference Proceedings*, volume 01.2, pages 1412–1417, 1997. 3

[70] Y.Petillot, I.T. Ruiz, D.M.Lane, Y.Wang, E.Trucco, and N.Pican. Underwater vehicle path planning using a multi-beam forward looking sonar. In *OCEANS'98 IEEE Proceeding*, pages 1194–1199, 1998. 60

[71] K. K. Yu and N. C. Hu. Two-dimensional gray-level object recognition using shape-specific points. *Journal of the Chinese Institute of Engineers*, 24(2), 2001. 34

[72] X. Yuan, Z. Hu, J. Chen, R. Chen, and P. Liu. Online learning and object recognition for auv optical vision. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 6, pages 857–862, 1999. 5

[73] Z.Zhang, R.Deriche, O. Faugeras, and Q.T.Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995. 35