Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Neeraj Iyer

Entitled
Machine Vision Assisted In Situ Ichthyoplankton Imaging System

For the degree of Master of Science

Is approved by the final examining committee:

Dr. Gavriil Tsechpenakis
Chair

Dr. Rajeev Raje

Dr. Mihran Tuceryan

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Dr. Gavriil Tsechpenakis

Approved by: Dr. Shiaofen Fang 04/09/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Machine Vision Assisted In Situ Ichthyoplankton Imaging System

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Neeraj Iyer

Printed Name and Signature of Candidate

04/09/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

MACHINE VISION ASSISTED IN SITU ICHTHYOPLANKTON IMAGING SYSTEM

A Thesis

Submitted to the Faculty

of

Purdue University

by

Neeraj Iyer

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2012

Purdue University

Indianapolis, Indiana

This work is dedicated to my family and friends.

ACKNOWLEDGMENTS

I am heartily thankful to my supervisor, Dr. Gavriil Tsechpenakis, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I also want to thank Dr. Rajeev Raje and Dr. Mihran Tuceryan for agreeing to be a part of my Thesis Committee.

Thank you to all my friends and well-wishers for their good wishes and support. And most importantly, I would like to thank my family for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Iyer, Neeraj. M.S., Purdue University, May 2012. Machine Vision Assisted In Situ Ichthyoplankton Imaging System. Major Professor: Gavriil Tsechpenakis.

Recently there has been a lot of effort in developing systems for sampling and automatically classifying plankton from the oceans. Existing methods assume the specimens have already been precisely segmented, or aim at analyzing images containing single specimen (extraction of their features and/or recognition of specimens as single targets in-focus in small images). The resolution in the existing systems is limiting. Our goal is to develop automated, very high resolution image sensing of critically important, yet under-sampled, components of the planktonic community by addressing both the physical sensing system (e.g. camera, lighting, depth of field), as well as crucial image extraction and recognition routines. The objective of this thesis is to develop a framework that aims at (i) the detection and segmentation of all organisms of interest automatically, directly from the raw data, while filtering out the noise and out-of-focus instances, (ii) extract the best features from images and (iii) identify and classify the plankton species. Our approach focusses on utilizing the full computational power of a multicore system by implementing a parallel programming approach that can process large volumes of high resolution plankton images obtained from our newly designed imaging system (In Situ Ichthyoplankton Imaging System (ISIIS)). We compare some of the widely used segmentation methods with emphasis on accuracy and speed to find the one that works best on our data. We design a robust, scalable, fully automated system for high-throughput processing of the ISIIS imagery.

1 INTRODUCTION

The name plankton is derived from the Greek adjective planktos, meaning "errant", "wanderer" or "drifter" [1]. Plankton typically flow with ocean currents. They are a crucial source of food to larger aquatic organisms such as various fishes and whales. Planktonic photosynthesis accounts for roughly half of the primary productivity on earth and plays an important role in the ocean's carbon cycle. Plankton abundance and distribution are strongly dependent on factors such as ambient nutrient concentration, the physical state of the water column, and the abundance of other plankton. The study of plankton is termed Planktology and an individual plankton is referred as a plankter.

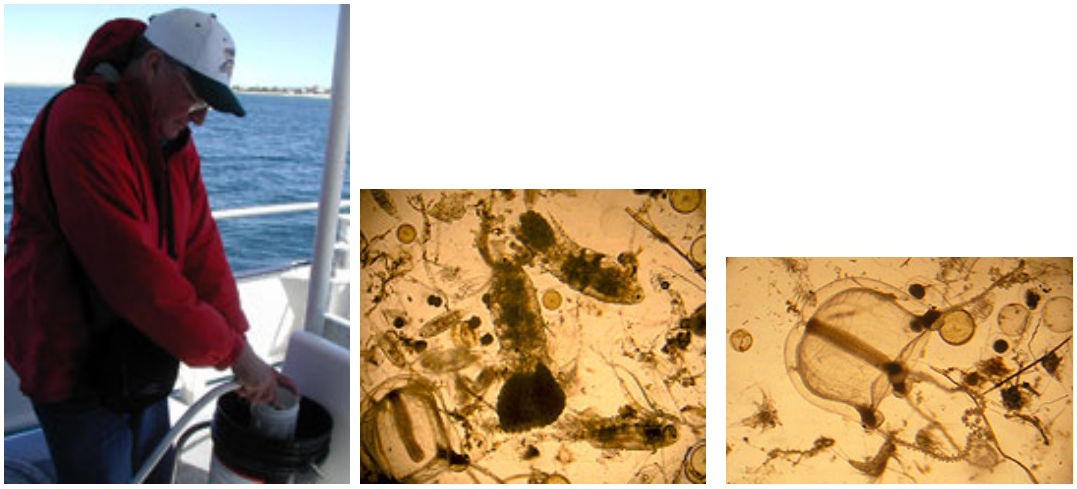
By studying the patterns in plankton distribution we can learn about the effects of climate change on the marine ecosystem. Since plankton are not harvested or exploited like fish or intertidal organisms, adjustments in distribution and abundance can be attributed to changing environmental factors [2]. As plankton are indicators of healthy aquatic environments, long-term studies have been carried out on plankton since the 1930s with numerous research projects continuing today [2].

Current plankton net-based approaches as shown in Fig. 1.1 to studying meso- and macro-zooplankton distributions result in many preserved samples. When high frequency net sampling is conducted, the resulting effort to sort, identify, and quantify organisms in the net samples can be extreme (e.g. our Straits of Florida (SOF) study of billfish larvae yielded 156 net samples every month for two years, which required ca. 12 person-years of microscope time to analyze). In comparison, digitally collected data have the potential advantage of being sampled and analyzed much more rapidly [3]. Where higher frequency (and higher resolution) sampling can be

Figure 1.1. Collection of plankton using nets

accomplished while at the same time allowing for much faster data analysis, there is tremendous capacity for improved scientific inquiry and monitoring.

With plankton net in hand as shown in Fig. 1.2(a), you can collect both microscopic and macroscopic organisms that form the base of the marine food chain. The species of plankton collected would be dependent on the depth of the water column being sampled and the mesh size of the net. After the towing process is complete, rinse the sides of the net with salt water. This pushes any plankton that are caught in the mesh. Thus a concentrated sample of plankton would be obtained at the bottom of the net. These samples can be observed using various magnifying devices. Current technologies available for the study of many zooplankters remain limited in comparison to the spatial-temporal resolution and data acquisition rate available for physical oceanographic measurements. Though net technology has become quite sophisticated (e.g. MOCNESS), enabling vertically discrete net samples coupled with detailed environmental data, net samples still require the task of being processed manually, a time-consuming and costly effort. The use of nets significantly reduces resolution as the nets integrate the organisms over the sampling distance and depth. Biological oceanographers have been advancing methodologies for more rapid, higher resolution sampling of phyto- and zooplankton via various acoustic and video technologies (e.g. OPC [4], VPR [5], ZOOVIS [6], SIPPER – see Wiebe and Benfield



(a) Obtaining Planktons from the Nets (b) Planktons under the microsope (c) Planktons under the microsope

Figure 1.2. ISIIS

2003 for major review of zooplankton sampling advancements). These technologies resulted in high-resolution data suitable for identifying copepods and benthic invertebrate larvae with spectacular results (Davis et al. 1992). However, these techniques are typically not applicable to the substantially rarer meso- and macro-zooplankton owing to small image volumes.

The critical issue for our interests (i.e. Ichthyoplankton and other dilute plankton) is that the VPR (Video Plankton Recorder), and its cousins, sample a relatively small volume of water that is inadequate to quantify plankton in a wider size range. For example, while copepods and some invertebrate larvae may exceed densities of 1-10 l-1, ichthyoplankton and larger zooplankton typically occur at densities of ca. 0.01-0.1, i.e. 1-2 orders of magnitude less. To more broadly sample rarer zooplankters, other techniques (SIPPER and OPC) have involved imaging and/or counting plankters by size as they pass through a narrow tube, but this approach does not enable in situ observations and can distort fragile plankton into non-identifiable shapes.

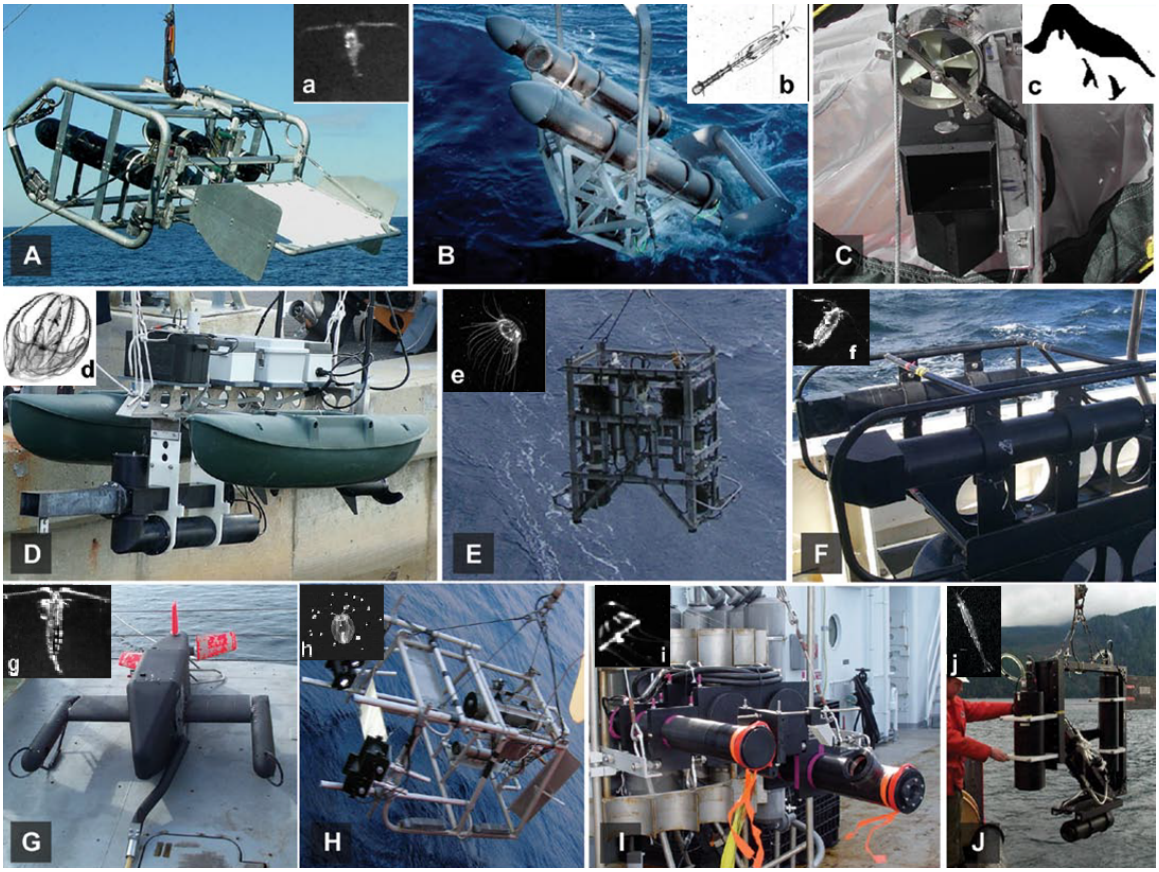


Figure 1.3. [3]. The number of in situ imaging systems is increasing rapidly. These are examples of some zooplankton and micronekton imaging systems (A-J) along with their corresponding (a-j) representative regions of interest (ROI's). Note that in most cases, the ROI's have been cropped from a larger image and have been resized to fit in the figure. None of the ROI s are to the same scale. A. Ocean DiVA: Digital Video Acquisition System. Image: C. Pilskaln, SMAST B. ISIIS : In Situ Ichthyoplankton Imaging System. Image: R. Cowen, RSMAS C. LOPC: Laser Optical Plankton Counter mounted in a ring net. Image: A. Herman, DFO Canada D. SIPPER : Shadowed Image Particle Profiler and Evaluation Recorder mounted below an autonomous pontoon vehicle. Image: A. Remsen, USF E. UVP: Underwater Video Profiler. Image: G. Gorsky, Laboratoire Oceanography Villefranche surmer F. VPR: Video Plankton Recorder mounted on BIOMAPPER II vehicle. Image: M. Benfield, LSU G. VPR II : Video Plankton Recorder II mounted in the Flying Fish high-speed towbody. Image C. Davis, WHOI H. LAPIS : Large-Area Plankton Imaging System. Image: E. Horgan, WHOI I. ZOOVIS -SC: Self-Contained Zooplankton Visualization System. Image: M. Sutor, LSU J. ZOOVIS : Zooplankton Visualization System. Image: M. Benfield, LSU.

Table 1.1
Comparison of plankton imaging systems

System	sampled volume	Pixel Resolution	Size of smallest object theoretically resolved (based on 10 pixel object)	Deployment speed
ZOOVIS	1 L/s	56 m	0.5 mm	0.5 m/s
VPR	2 L/s	10 m	0.1 mm	6 m/s
SIPPER	10 L/s	50 m	0.5 mm	1 m/s
UVP	6 L/s	175 m	0.2 mm	1.5 m/s
LAPIS	360 L/s	500 m	5 mm	1 m/s
ISIIS-1	70 L/s	70 m	0.7 mm	2.5 m/s
ISIIS-2	140 L/s	70 m	0.7 mm	2.5 m/s

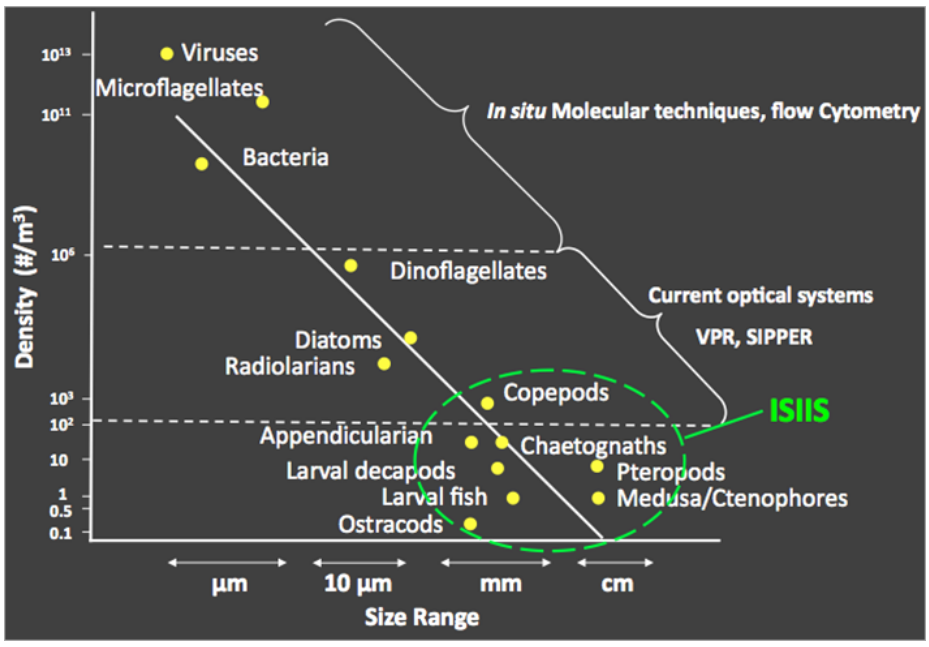
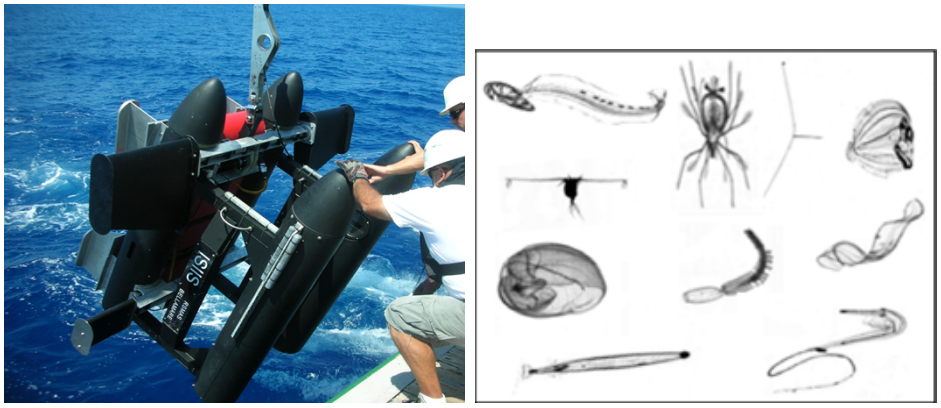


Figure 1.4. Plankton Size and Density

These approaches to plankton classification either assume that specimens or regions of interest (ROIs) are already segmented (this is done manually) and thereby focus on the recognition methodology or they focus on the visual features to be used for recognition, assuming that the data is free of noise (intensity ambiguities). Thus the first step involves a tedious work of manually segmenting large amounts of data, which produces isolated plankton images which are easy to recognize. The second approach assumes the existence of a single extracted specimen in the examined image, where the specimen outline or region features can be clearly distinguished visually and computationally. They focus on the visual features to be used for recognition, assuming that the data is free of noise (intensity ambiguities) [3, 7].

To address these problems, a high-resolution towed plankton imaging system, the In-Situ Ichthyoplankton Imaging System (ISIIS) as shown in Fig. 1.5(a), was built, capable of imaging water volumes sufficient to accurately quantify even rare plankton (e.g. larval fish) in situ [8]. This imaging system produces very high resolution imagery at very high data rates, necessitating automated image analysis. Since the goal is the identification and quantification of a large number of specimens, whose shapes can be relatively similar to each other, an automated system for detection and recognition of specimens of interest is developed using computer vision and machine learning tools.

We use the In Situ Ichthyoplankton Imaging System (ISIIS) to get high resolution images of the planktons (see Fig 1.5(b). The vehicle frame is divided into four compartmentalized enclosures with imaging and optical equipment seamlessly integrated into ISIIS ventral housings, with environmental sensors (e.g. CTD, O2, PAR, fluorometry, ADCP) and electronics in the dorsal housings. The dive fins are positioned ahead of the vehicle aligned with the tow point and away from the imaging pods. The vehicle is designed to undulate between the surface and a maximum depth of 200 m.



(a) Launching of ISIIS-2 (b) Scanned Planktons

Figure 1.5. ISIIS

This Thesis describes an approach that automatically extracts and classifies specimens of multiple classes of plankton from the digital images. In this work our goal is to segment individual planktons from raw images and extract the best features from a very large volume of data which will be used for classification and recognition of planktons. We tackle both the problems of automatic segmentation of planktons and recognition of multiple classes (around 20) in a scalable and efficient way. The data collected by the ISIIS during few hours of collection would need atleast 20 man years for manual processing which is not practical. If we cannot process this vast amount of data faster the data would be use useless. Thus our challenge is not only to achieve accuracy but also achieve high speed which can make the whole process of segmentation, recognition and classification of plankton a fully automated high throughput system.

We focus on achieving a balance between accuracy and speed for processing this large amount of data. Some plankton are deformable and may have different shapes depending on the point of view. They can vary in size and some of them can be identifiable only if some small parts of the planktons are identified. The quality of

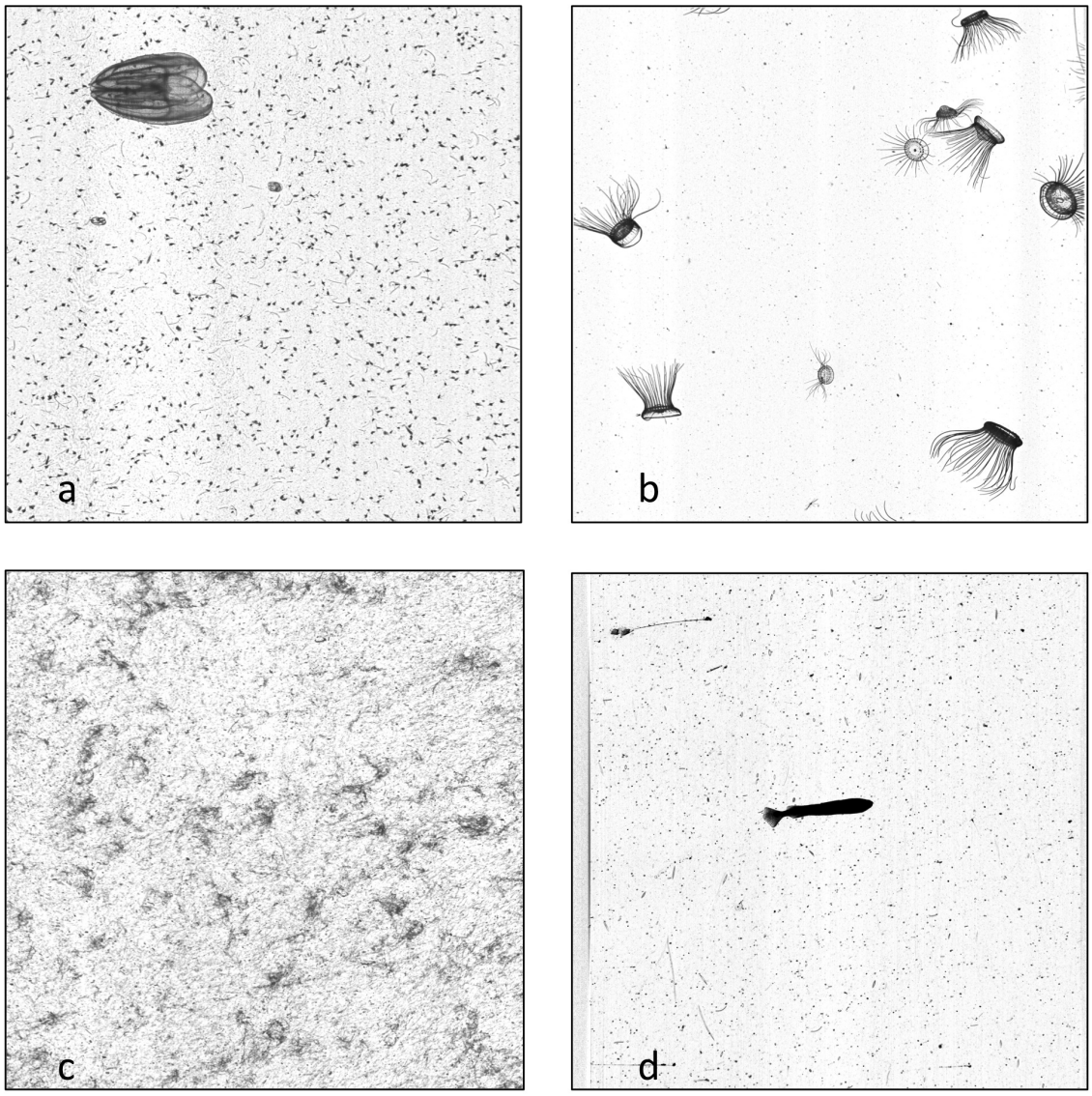


Figure 1.6. Example images from each of 4 unique water columns: a) 5 m depth at Stellwagen Bank, b) 15 m in southern California Bight, c) 10 m in Monterey Bay, d) 30 m 40 km south of Rhode Island. Note differences in background particles, ranging from very dense copepods (ca. 100 l-1) in a), relatively clear water in b) very dense particulates in c) to dense diatoms and some marine snow in d).

the images also varies a lot depending on the quality of the water bodies. Different planktons have different features which have to be used to classify them. We cannot give the same weight to a set of features across all the classes. Thus all these issues can make identifying planktons very difficult.

In our first part we focus on segmentation of ISIIS images to extract separate plankton images and extract a set of features that are used for recognition. We did a comparative study by implementing various segmentation algorithms. In order to achieve high throughput we implement parallel algorithms and try to incorporate high level as well as low level parallelism. For the segmentation we performed speed and accuracy analysis to determine which algorithm works better in our particular case. We implemented segmentation using K-means, Fuzzy C-means, Isodata Clustering, Spectral clustering and K-harmonic means based clustering.

The issues we had to deal with during segmentation were preprocessing for removal of noise, accurate segmentation and distinguishing noise and dust particles from the plankton images. These images would then be used during the recognition phase to quantify the various plankton.

Recognition of various plankton has its own set of challenges such as the plankton may be deformable, the images being affected by the position with respect to the line scanner and the quality of the water sampled. Therefore we cannot have a fixed model for each plankton class. Also we cannot give the same weight to some features over all classes. To overcome these issues we build a classification tree based on intra class similarity and inter class variance. The tree is build bottom up with each leaf node representing a concrete class, the internal nodes representing group of classes having similar features and the edges of the tree having the set of features that distinguish the classes. Our method is different from the existing methods in the way that our method is scalable to higher number of classes and gives different importance to

features for different classes of plankton. Once the tree is built using the training samples we simply traverse the tree based on the features instead of a "one vs one" comparison to classify the plankton. Thus our method scales well even as the number of classes increase.

2 PREVIOUS WORKS

This thesis draws motivation from the work done by R.K. Cowen, C. Guigand, C. Cousin, G. Tsechpenakis [9–13] which describes the ISIIS device and the methods that have been implemented for recognition of plankton from the ISIIS images. This thesis takes a different approach from what has already been explored in [9]. In [9], they use the Scale Invariant Feature Transform (SIFT) for matching between the detected regions and the organism images in our database. This method does not scale well with the huge dataset of unknown images and with the increase in the number of plankton classes. In [13] an active learning approach was taken to visual multiple object class recognition, using Conditional Random Field (CRF) formulation. This approach worked better but involved a human oracle that was responsible for selecting the samples for active learning and therefore was not suitable for huge volumes of data. Speed and Scalability was again an issue here. There have been similar attempts in automatic classification and quantification of plankton. Some of the approaches are explained below.

Current larval fish sampling studies are typically carried out with towed net systems, which offer limited versatility and data analysis [10]. Nets collect organisms over the sampling distance/depth profile(s) and hence do not provide a fine scale resolution of organism population. Though net technology has become quite sophisticated e.g., the multiple opening/closing net and environmental sensing system), enabling vertical resolution coupled with detailed physical data, Net tows require massive sums of time to perform data analysis (approximately one man-year of post-processing work for every two days at sea). Also since this system does not work in situ, it can damage the more delicate planktons. Due to these major disadvantages, there has been more focus on Visual Recorders for physical sampling of plankton. The visual recorders can

be dividied into particle detection and image-forming systems. The particle detectors e.g. optical plankton counter [4] use the interruption of light source by plankton to detect and count the targets as they pass through a sampling tunnel. These might damage volatile plankton while passing through the tunnel.

(a) Plankton tow net, 153-m pore size, 0.5-m diameter (D:L=1:3)

(b) Multiple opening/closing net system

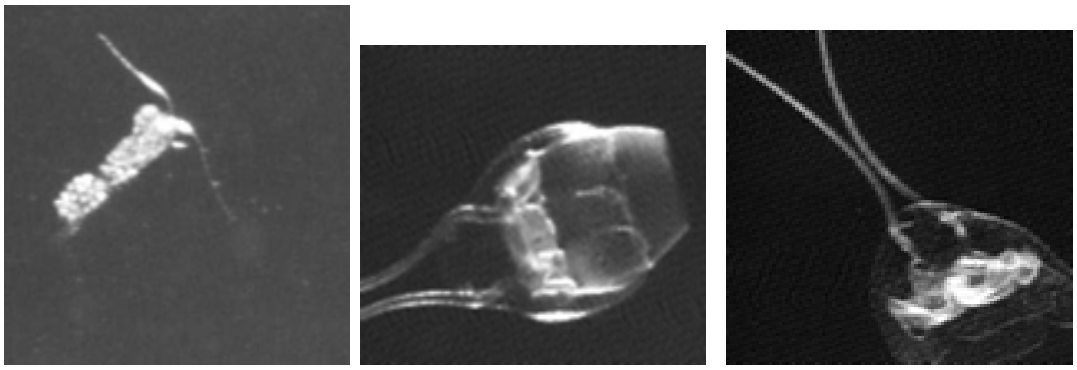
Figure 2.1. Net system for plankton study

Image-forming optics put to use various cameras to capture the organisms while towing the instrument. These began with a photographic camera in a net and currently include towed system as the camera net system [14], the ichtyoplankton recorder [15], Video plankton recorder (VPR) [5], in situ video recorder [16], in situ ichthyoplankton recording system [9] and the shadowed Image Particle platform and Evaluation Recorder (SIPPER) [17]. Also there are profiling systems, such as underwater video profiler (UVP) [18] and holographic instruments [19, 20]. See Wiebe and Benfield [6] for a review.

The majority of optical systems use video and typically scan small volumes of water to achieve acceptable image resolution characteristics. The VPR scans plankton of size between 0.1 mm to 1 cm [5]. It is capable of scanning 60 images per second.

It is an in situ imaging system. It Cannot identify plankton to species level and undersamples rare taxa (e.g., $< 50/m^3$). The VPR and the scanned plankton are shown below.

Figure 2.2. Video Plankton Recorder



(a) Copepod (b) Jelly Fish (c) Jelly Fish

Figure 2.3. VPR Scanned Plankton

[21] was one of the early Automatic Plankton Image Recognition systems that used the images obtained from the VPR. they combined traditional invariant moment features and Fourier boundary descriptors with gray-scale morphological granulometries to form a feature vector capturing both shape and texture information of plankton images. They used a Learning Vector Quanitization (LVQ) neural network classi-

fier. [7] uses texture based feature, co-occurance matrices (COM) as the feature, and a Support Vector Machine (SVM) as the classifier. This method does not scale well with large data set and a large number of plankton classes.

The Zooplankton Imaging System (ZOOVIS) [22] has a camera is aimed downward into a sheet that is 12 cm wide and 3 cm deep. By setting the depth of field to match or slightly exceed the depth of the light sheet, only targets that are in focus are illuminated. It has a depth range of 0-250 m, sampling rates of up to 4Hz. They are typically limited to macrozooplankton (0.1 cm to 10 cm) and provide resolution of 50 microns. It scans relatively small volumes of water.

Figure 2.4. ZOOVIS Zooplankton Imaging System

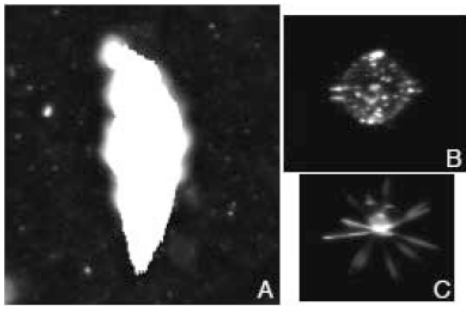


Figure 2.5. Image of a ciliate (Laboea, panel A), dinoflagellate (Protoperidinium, panel B), and radiolarian taken with ZOOVIS-SC in Monterey Bay in July 2006

SIPPER [17] was developed by Center for Ocean Technology of the College of Marine Sciences at USF in St.Petersburg. It uses a line scan laser camera to take a cross section of all particles that flow through a 4 by 4 tube. This results in a continuous digital image that is 4 inches wide. Its purpose is to enable scientists to get an accurate count of types of marine plankton in a region of water. It uses High-speed digital line-scan cameras and scans 36000 lines per second. It scans plankton of size $< 100\mu$m at 96mm depth of field and 96 mm width. The towing speed is 3 knots and scans 14 litres/sec. It does not scan in-situ and scans images and therefore might affect the plankton while flowing through the tube.

In [23] an active learning approach for multiclass SVM classification was proposed. From the training samples they select 15 strongest features from 29 features and the recognition is based on these 15 features. These 15 features have the same weightage across all the plankton classes. Multiclass SVM is basically multiple bi-class SVM. Thus this method cannot scale well with the number of plankton classes.

(a) Sipper device (b) Tube through which planktons pass (c) sample SIPPER Image

Figure 2.6. SIPPER Images

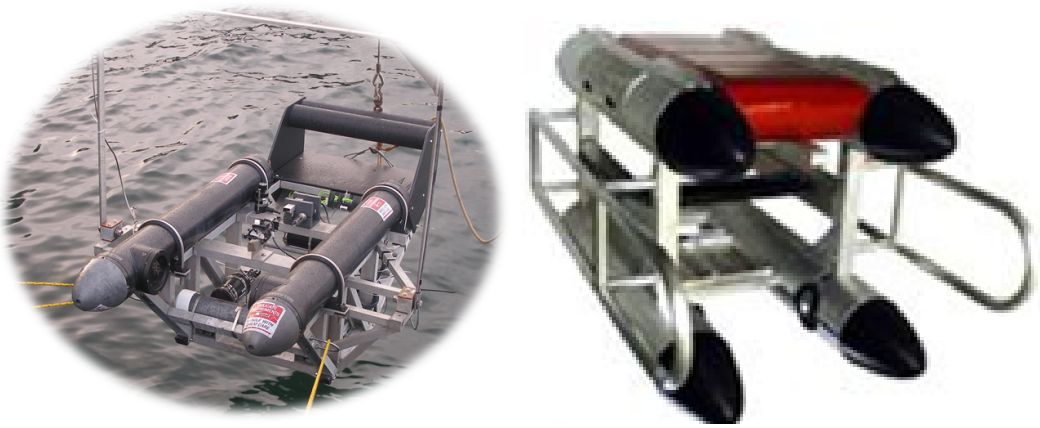
The Optical Plankton Counter (OPC) was originally designed at the Bedford Institute of Oceanography as a remotely-towed sensor providing continuous real-time information on zooplankton [24, 25]. The OPC complemented information obtained from net tows and povided information overlap and higher resolution measurements.

(a) Optical Plankton Counter (b) Tube through which planktons pass (c) OPC Image

Figure 2.7. Optical Plankton Counter Images

The ISIIS [9] uses line scan cameras from the machine vision industry to scan the water and provide a continuous imaging with 60 micron pixel at 4-5 knots speed. The line scan camera is coupled with a back illumination technique (shadowgraph) that provides exceptional resolution and depth of field while providing a telecentric image (magnification is not affected by distance from object to lens). The data is either ported to shipboard via Fiberoptic towing cable or recorded internally. ISIIS instruments are designed to image large volume of water in order to study relatively rare organisms. The vehicles and their imaging system are configurable and give the versatility needed for studying a range of organism from small, abundant plankton, to larger and rarer specimens (i.e fish larvae). It has a depth rating of 200m, the vehicle is capable of pre-programmed undulation and can be towed off the side of the ship

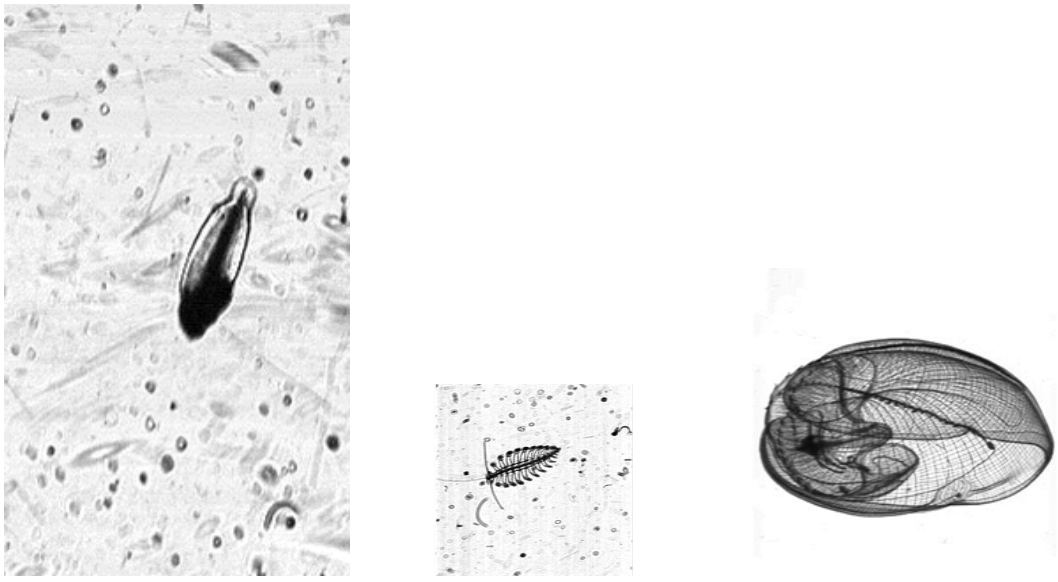
to avoid the disturbances created by the tether. Particular attention has been given to provide for undisturbed flow path in front of the imaging viewports. This vehicle is equipped with environmental sensors (CTD, PAR, Fluorometer) and a navigation ADCP. Plankton images are transferred via fiber-optic to the ship. Below is the ISIIS instrument and the images obtained from the device.



(a) ISIIS-device version 1 (b) ISIID-Device version 2

Figure 2.8. ISIIS device used to record plankton images

The ISIIS system utilizes a high-resolution, line-scanning camera with a Light Emitting Diode (LED) light source, modified by plano-convex optics, creating a collimated light field to back-light a parcel of water (Fig. 2.10). The imaged parcel of water passes between the forward portions of two streamlined pods (pressure housings), and thereby remains unaffected by turbulence. This results in very high-resolution plankton images in their natural orientation and position. Quantification of organism concentration and fine scale distribution is possible when a sufficient volume of water is imaged this way. The imaging data and associated oceanographic data is ported to the surface via 0.322 in copper/fiber optic oceanographic wire and recorded onto a computer controlled raid array.



(a) Larval Fish (b) Polychaete larva (c) Ctenophore

Figure 2.9. ISIIS scanned plankton

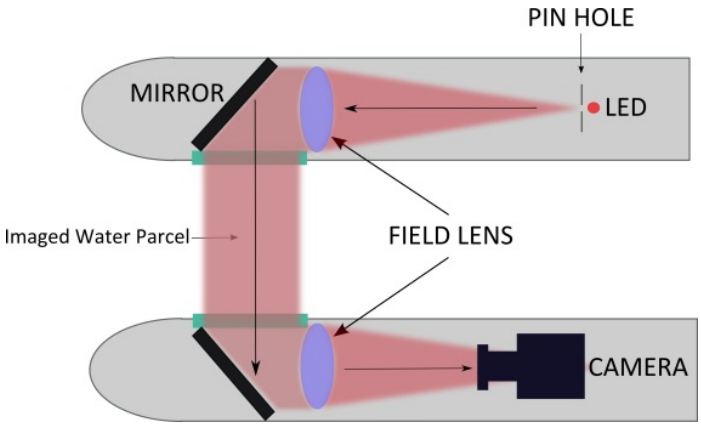


Figure 2.10. Light scheme using shadowgraph technique

Light passes through plano-convex lenses thereby establishing a pseudo-collimated light beam. This is then refocused by a second field lens and it then impinges on an imaging lens. The advantages this approach offers over other lighting techniques include: high depth of field (Arnold and Nuttall-Smith 1974, Settles 2001), very sharp

outlines of organisms and internal structures (facilitate automated recognition) and telecentric image (magnification level not affected by distance from object to the lens). Since the light rays are directed toward the imaging sensor, extremely low intensity of light is required compared to any other lighting technique. Due to this we do not need to use bright light that may deter organisms away from the imaging area. We used a line-scan camera for imaging. This type of camera creates a continuous image at high speed scanning rates, which allows for high-resolution images (vertical resolution of 2048 lines and a 36 KHz scanning rate). This combination provides for a continuous visual field that is approximately 13.5 cm tall with a 40 cm depth of field [10]. Vertical lines on the plane are put together to form "continuous" images: the horizontal direction corresponds to recording time.

The ISIIS provides high resolution images and scans a large volume of water compared to the other instruments. The challenge here is that the data is so overwhelming that we need to come up with parallel processing that is highly efficient in both speed and accuracy inorder to quantify the plankton from the images obtained. The table below shows the comparison between ISIIS with the other systems. We therefore use the ISIIS images to develop an automated segmentation and recognition system that is able to segment the planktons from the ISIIS raw images, extract the features and is able to recognize and quantify them in a highly efficient manner. It performs the scan in situ i.e. the planktons are not disturbed and therefore provide better images.

3 METHODOLOGY

This thesis is broadly divided into two parts: segmentation and recognition. Segmentation process involves an extensive comparative study to determine which segmentation method works better than segmentation results in terms of speed and accuracy. For classification we compare K-means [26], Fuzzy C-means [27], Isodata Clustering [28,29], Spectral Clustering [30] and K-harmonic means [31]. For plankton recognition and classification, we propose a novel classification tree approach which is highly scalable. This chapter contains detailed descriptions of all work done as part of this thesis.

3.1 Preprocessing and Noise Removal

Due to the use of line scanners any dust or particles on the sensor will appear to be a line over the course of the entire scan. Other errors include blurring, spurious region pixels. In order to remove the vertical lines introduced due to the dust or particles on the sensor we take the fourier transform of the image, shift the fourier transform to the origin and mask the mid range and then reverse shift and take the inverse fourier transform. This eliminates the vertical lines as shown in Fig. 3.1.

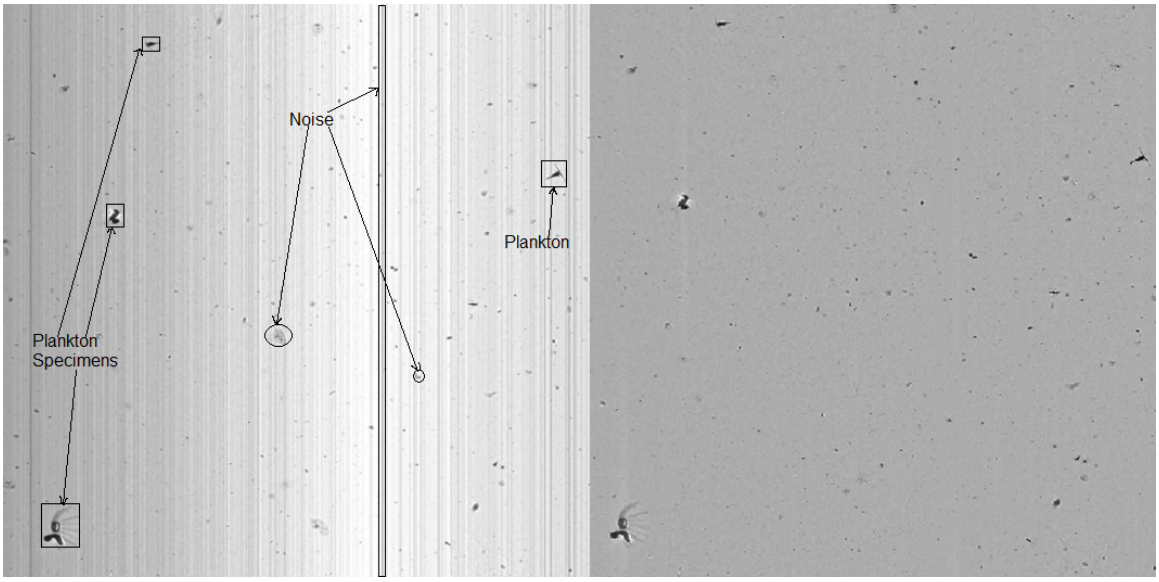


Figure 3.1. Example of original and cleaned image

3.2 Plankton Segmentation

Every single image obtained from the ISIIS contains multiple planktons of different types. In order to be able to be able to quantify those we need to identify them individually. Segmentation is very crucial for feature extraction and recognition. Accurate segmentation is challenging because quality of images vary depending on water quality, level of noise introduced, relative position of plankton with respect to the system. The plankton are deformable and there are multiple classes of plankton, we therefore cannot find predict the shape of the plankton to use during segmentation. Segmentation methods have not been explored as most of the recognition systems focus on recognition by assuming perfect segmented images. The existing plankton classification systems assume that the images are manually segmented. This is not possible due to the large volume of data and therefore we need an automated segmentation process to achieve this. We concentrate on segmentation of the ISIIS images to segment planktons by reducing the noise. We need to find the minimum bounding

polygon which can hold the entire plankton in a way such that we are able to extract the best features to distinguish the planktons. Some of these are the shape and appearance features such as transparency ratios, convex hull ratios, eigenvalues, morphological granulometric features, geometric moments, intensity distribution. Due to the challenges involved in identifying the planktons, care has to be taken to avoid any loss of data during the segmentation phase. The segmentation process should be able to neglect the noise and still be able to identify and extract the planktons. As we scan a large volume of water, processing of these images at real time is also one of the major challenges. This challenge is tackled by exploiting the power of multicore systems by introducing different levels of parallelism. The end goal is to have a single system on board the ship that can process these images as they are captured by the ISIIS-2 equipment. We discuss the parallelism introduced in later sections. Thus we work with the constraint of having a single system with multiple cores instead of processing these images off shore on a distributed system of multiple processors.

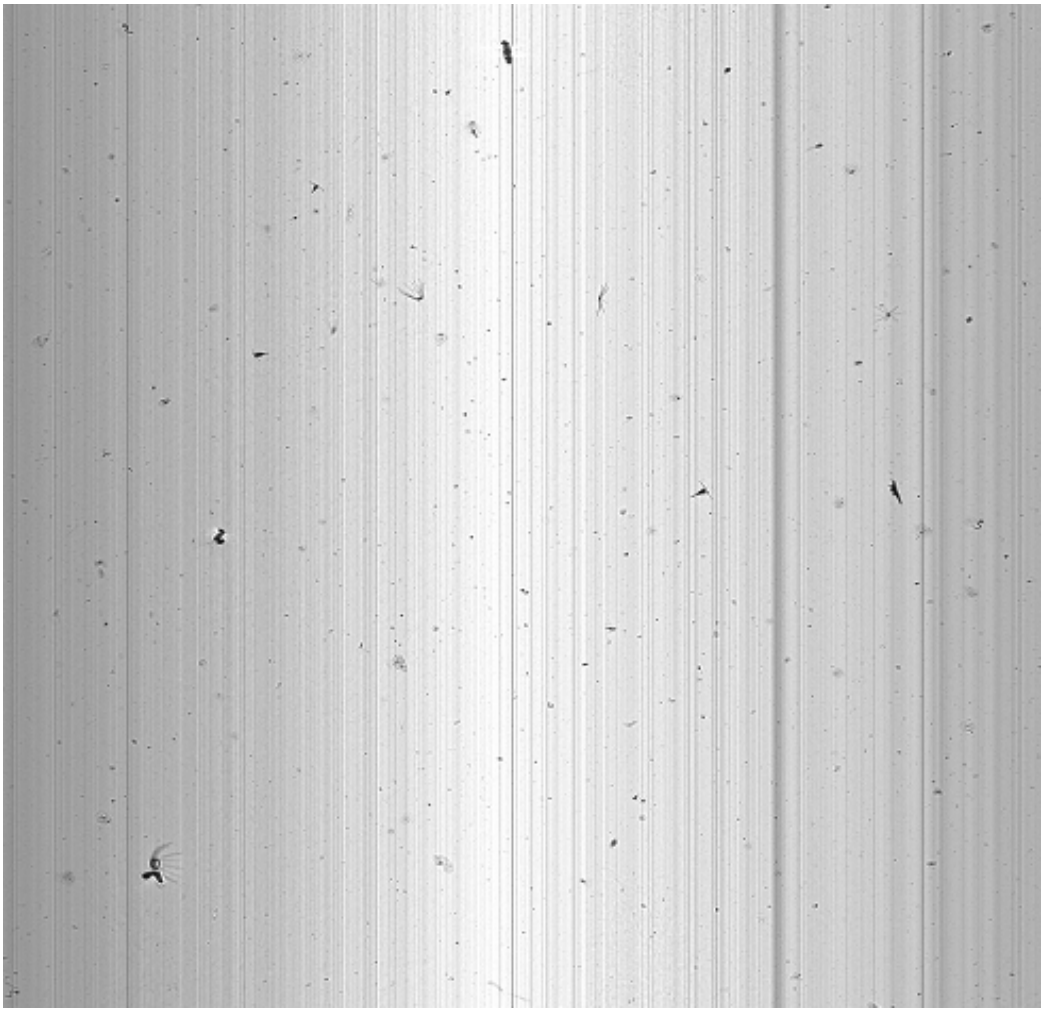


Figure 3.2. ISIIS Sample Image

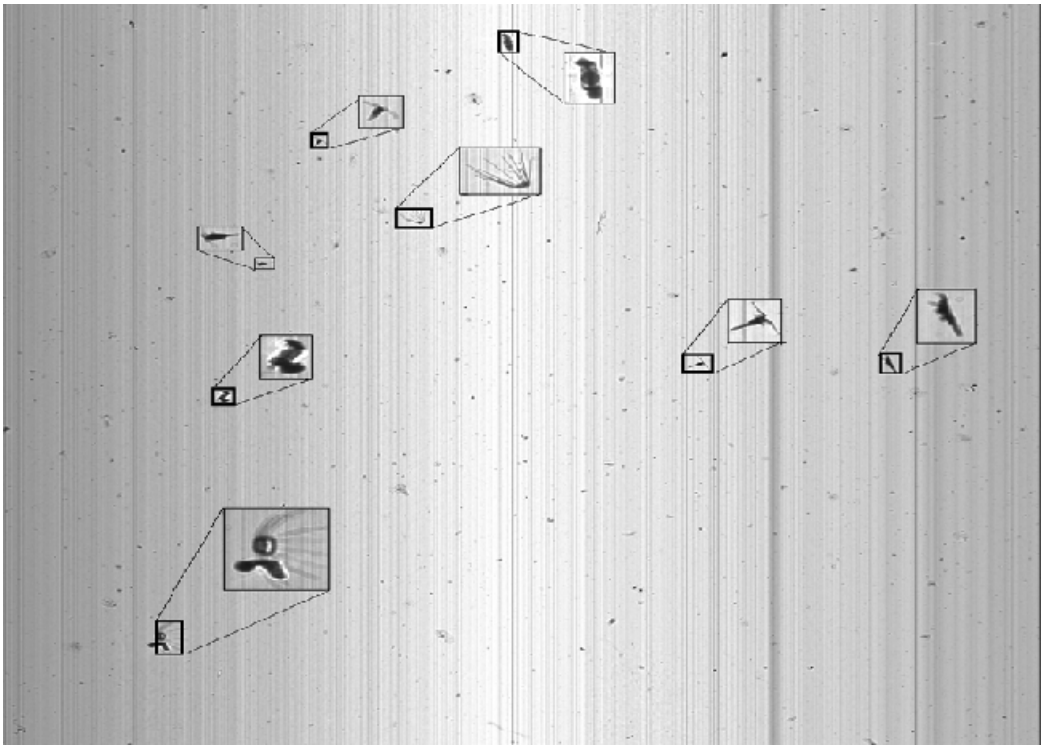


Figure 3.3. Identifying plankton

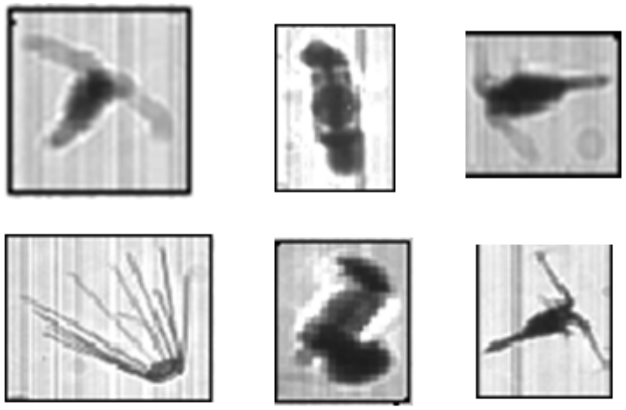


Figure 3.4. Planktons Segmented from ISIIS Image

Image segmentation [32] is considered to be the most common problem in computer vision. It refers to the process of partitioning a digital image into multiple segments.

The goal is to extract meaningful information from the images. Image segmentation is typically used to locate objects in images. Image segmentation finds use in a wide variety of applications such as medical imaging, face recognition, machine vision, biomedical and biological applications etc. Many algorithms have been suggested in this regard. Finding a best segmentation method is a non-trivial process. Also with the large quantity of data, the speed of processing is also an important factor. We attempt to find an approach that would run efficiently on a single system which could be used on the ship to process the images at real time instead of transferring these huge volumes of data to be processed elsewhere. Thus the challenge is to fully utilize a single system in terms of efficiency without compromising on the quality of results. We introduce different levels of parallelism in our approach to make the solution as efficient as possible and to utilize the full functionality of a current multicore system [33].

Some of the methods of image segmentation are model based and appearance based. The simplest form of segmentation is thresholding [32]. Thresholding classifies the pixels of a given image into two groups (e.g. foreground and background). One group would be the pixels with their gray values above a certain threshold while the other group being those wth gray values equal to below the threshold. This approach is very naive and as the appearance of plankton changes over time, spatially and over different water quality / depth it is difficult to set a threshold. Model based segmentation assumes the regions to be segmented have a repetitive form of geometry. As we are scanning a large volume of data for multiple planktons, we cannot restrict our segmentation to look for a particular set of shapes or geometry. Therefore we cannot use model based techniques. The other approach could be supervised or unsupervised clustering. As we have no training samples to start with we cannot use supervised clustering. This leads us to unsupervised clustering. Unsupervised clustering refers to the problem of trying to find hidden structure in unlabeled data.

The simplest and widely used unsupervised learning is clustering. Since the intensity of the pixels is the key feature to separate the foreground and the background we have used some well known center based clustering and a graph based clustering. We implemented the segmentation of planktons using K-means clustering [26], iterative K-means, Fuzzy C-means [27, 34, 35], Isodata clustering [28, 29], K-harmonic Means algorithm [31] and Spectral Algorithm (Shi and Malik) [30]. Many algorithms offer better clustering than K-means [26]. We implemented both sequential and parallel versions of these algorithms. The constraints for parallelism is to have the system work on a single multicore system instead of a cluster of systems. This constraint will allow us to process the image at real time on the ship itself while capturing the images from the ISIIS-2 equipment. We found that K-harmonic means provided the best balance in terms of speed and accuracy for the ISIIS images and therefore used this in our system of automatic segmentation and recognition.

3.2.1 K-means

K-means is one of the simplest unsupervised learning algorithms that solves the well known clustering problem proposed by MacQueen in 1967 [26]. It classifies a given data set into "K" number of clusters, "K" being fixed a priori. The centroids are initially selected randomly and each point is associated to the nearest centroid. Then the centroids for each cluster are recalculated based on the number of points associated to that cluster. This process continues till the point the centroids no longer move.

Consider N data points and K disjoint subsets S_j containing N_j data points so as to minimize the sum-of-squares criterion which is the euclidean distance in the given feature space [26].

$$J = \sum_{j=1}^{K} \sum_{n \epsilon S_j} |x_n - \mu_j|^2 \tag{3.1}$$

where x_n is a vector representing the n^{th} data point and μ_j is the geometric centroid of the data points in S_j. We use pixel intensity to calculate the centroids using the Euclidean distance. The K-means finds clusters and stops when the clusters are fairly constant over multiple runs. K-means is simple and easy to implement, it is very fast and provides decent clustering. It however depends on the initial clusters that are selected, it finds local instead of global maxima.

Complexity: Let t_{dist} be the time to calculate the distance between two objects.
Each iteration time complexity $O(KmNt_{dist})$ where
K = number of clusters (centroids)
m = number of dimensions
N = number of data points
For average I iterations to converge giving $O(IKmNt_{dist})$
The timing and accuracy results are provided in the experiments section.

3.2.2 Iterative K-means

K-means results are dependent on the initial clusters that are selected. Thus the quality of K-means is highly dependent on the initial clusters and might vary with different set of initial clusters. In order to eliminate the dependence on the initial clusters selected, we could run the K-means multiple times with different initializing centroids and then take the best result from the different runs. This will help us eliminate the dependence on the initial clusters. It can be noted that due to multiple runs, this method is will take a longer time to yield the clusters.

Suppose the K-means is run 'p' times to find the best clustering. The complexity would be $O(IKmNpt_{dist})$ where all the other terms are similar to the K-means. Though iterative K-means could give better result for some images, the same number of runs might not be needed to process the other images. Thus deciding how many

times to run the K-means depends on the images and therefore cannot be adjusted or decided separately for each image. Therefore this approach is not very practical under our current constraints.

3.2.3 Fuzzy C-means

K-means divides the sample space into clusters in a way such that each data point can belong to only 1 cluster. This is considered to be hard membership. There is a notion of assigning each data point to multiple clusters depending on the probability of each data point belonging to each cluster. In Fuzzy clustering, each point can belong to multiple clusters to a certain degree instead of belonging to just one cluster. The points on the edge of a cluster, maybe in the cluster to a lesser degree than points in the center of cluster. An overview of the various fuzzy clustering algorithms is available in [36].

Consider N data points and K disjoint subsets so as to minimize the sum-of-squares criterion which is the as shown below.

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N} {w^p}_{i,j} \left| x_n - \mu_j \right|^2 \tag{3.2}$$

where p is a parameter that determines the influence of the weights $p \epsilon [1..\infty]$, x_n is a vector representing the n^{th} data point and μ_j is the centroid of the data points in cluster j.

$$\mu_j = \frac{\sum_x w_k(x) x}{\sum_x w_k(x)} \tag{3.3}$$

Any point x has a set of coefficients giving the degree of being in the K^{th} cluster $w_k(x)$. With fuzzy C-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster.

The degree of belonging, $w_k(x)$, is inversely proportional to the distance from x to the cluster in the previous pass. It is also dependent on a parameter m which controls how much weight is given to the closest center. The fuzzy C-means algorithm is very similar to the K-means algorithm [27].

3.2.4 Isodata clustering

K-means, Fuzzy C-means have to set the number of clusters fixed a priori. But there are situations when we would want the number of clusters to vary depending on the situation. The Isodata clustering helps us to do the same. The Iterative Self-Organizing Data Analysis Technique (ISODATA) method is a modification of the K-means clustering developed by Ball et al. [28]. The ISODATA algorithm is similar to the K-means algorithm with the distinct difference that the ISODATA algorithm allows for different number of clusters while the K-means assumes that the number of clusters is known a priori.

The procedure of the ISODATA is as follows:

1. Parameters required for the algorithm such as convergence condition for rearrangement, deciding small clusters, conditions for splitting and merging clusters are determined and the initial cluster centroids are selected.

2. According to the convergence condition, clusters are rearranged using the K-means method.

3. If all the clusters are in the given threshold and there is no variation, the processing terminates.

4. Clusters are merged if either the number of members (pixel) in a cluster is less than a certain threshold or if the centers of two clusters are closer than a certain threshold. Clusters are split into two different clusters if the cluster standard

deviation exceeds a predefined value and the number of members (pixels) is twice the threshold for the minimum number of members.

Though the ISODATA method can adjust the number of clusters by division and fusion, global optimal cannot be guaranteed and as it has more parameters than the K-means method, adjustment of the parameter is still more difficult.

3.2.5 Spectral Algorithm

As opposed to K-means clustering, which results in convex sets, spectral clustering does not make any assumptions on the form of the cluster. It can therefore solve problems, such as intertwined spirals [30]. Spectral clustering foots on graph theory and appeals to intuition. We take the help of adjancency matrix for partitioning the data. An adjacency matrix is a means of representing which vertices (or nodes) of a graph are adjacent to which other vertices. We explain below how such a matrix can help in efficiently clustering the data points.

Given our data points $x_1, ..., x_n$, we construct a graph on the n objects where 2 objects are connected by an edge if they are sufficiently similar. The similarity condition can be any condition such as the distance between the 2 points. For e.g. we can add an edge between every set of objects x_i , x_j whose distance is less than any ϵ. Other ways to create this graph would be to use K-nearest neighbor graphs.

For every graph of this form, we can construct an n x n matrix M which is the adjancency matrix, where $M_{ij} = 1$ if there is an edge between x_i and x_j and $M_{ij} = 0$ otherwise. We look at the eigen values and eigen vectors of this matrix to use for clustering. Let us see how the eigen values and eigen vectors can help us with clustering.

Consider a very basic example, where we have 2 clusters and when we construct a graph, we put edges between every pair of objects in the same cluster, and put no edges across clusters. In this case, the adjacency matrix M of the graph is block diagonal. (Assuming 4 objects in this example)

$$M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

The eigenvectors of this matrix are $\begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^T$ and $\begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix}^T$. If we consider the first one of these, the coordinates for which the eigenvector is 1 correspond exactly to the items in the first cluster and the second eigen vector identifies the objects in the 2^{nd} cluster. In this simple example we can see how the eigen values can help us in clustering the data. We can extend this to k clusters by using k eigen vectors (corresponding to the largest eigenvalues). In the example we assumed there are no edges between clusters, usually that is not the case. For these problems we take the knowledge of graph cut to partition the data.

Given a similarity graph with adjacency matrix W, the simplest and most direct way to construct a partition of the graph is by solving the mincut problem. For a given number K of subsets, the mincut approach simply consists in choosing the partition $A_1 ... A_k$ which minimizes

$$cut(A_1, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, A_i\prime) \tag{3.4}$$

Where $A_i\prime$ is the complement of A. Here factor 1/2 is for consistency reasons, as we are dealing with undirected graphs. For k > 2 we use the following equation given by [30]

$$Ncut(A_1, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, A_i\prime)}{vol(A_i} = \sum_{i=1}^{k} \frac{cut(A_i, A_i\prime)}{vol(A_i)} \tag{3.5}$$

The running time of the normalized cut algorithm is O(mn) where n is the number of pixels and m is the number of steps, the eigensolver takes to converge. Spectral algorithms offer better segmentation results than K-means or other center based algorithms but is comparatively slower. Thus processing of large volumes of data takes fairly long time and it fails in the constraints of processing the images at real time.

3.2.6 K-harmonic Means

The performance of K-means depends on the initialization of the centers. This is a major problem and can vary the quality of clustering results. K-harmonic means algorithm (KHM) is a center-based clustering algorithm which uses the harmonic averages of the distances from each data point to the centers as components to its performance function [31]. It has been proven that K-harmonic means is insensitive to the initialization of the centers. KHM is an interative algorithm that refines the K clusters.

Let C = $\{c_j | j = 1, ..., K\}$ be K centers and

S = $\{x_i | i = 1, ..., N\}$ be N given data points, the KHMs performance function is

$$Perf_{KHM}\left([x_i]_{i=1..n}, [c_j]_{j=1..k}\right) = \sum_{i=1}^{N} \frac{K}{\sum_{j=1}^{k} \frac{1}{||x_i - c_j||^2}} \tag{3.6}$$

The quantity inside the outer summation is the harmonic average of K squared distances. The K-harmonic means has a "built-in" dynamic weighting function, which boosts the data that are not close to any center by giving them a higher weight in the next iteration. The complexity of K-harmonic means is the same as K-means. It results in very good clustering results and is insensitive to the initial points chosen. Thus on most occasions it provides better results over K-means and also has a competitive speed when compared to the other segmentation methods. Thus for our segmentation problem we use this inorder to achieve highly accurate segmentation results at a fairly high speed. The results and comparison of the performance is provided in the experiments section.

3.3 Recognition

Recognition of plankton is the major portion of the thesis work. There have been many methods that have been tried and tested for plankton recognition. This is not a trivial job for the following reasons:

1. The planktons of the same class may vary in size and also shape to some degree.
2. The images are dependent on the position of the plankton with respect to the instrument.
3. They might be partially occluded due to the noise, dirt or other plankton.
4. Same set of features cannot be used for recognition across all the plankton.
5. Some plankton are rare with less training samples.
6. Many classes of plankton and large data set of unknown images.
7. Image quality varies depending on quality of water sampled.
8. Inter class similarity.
9. Intra-class variance.

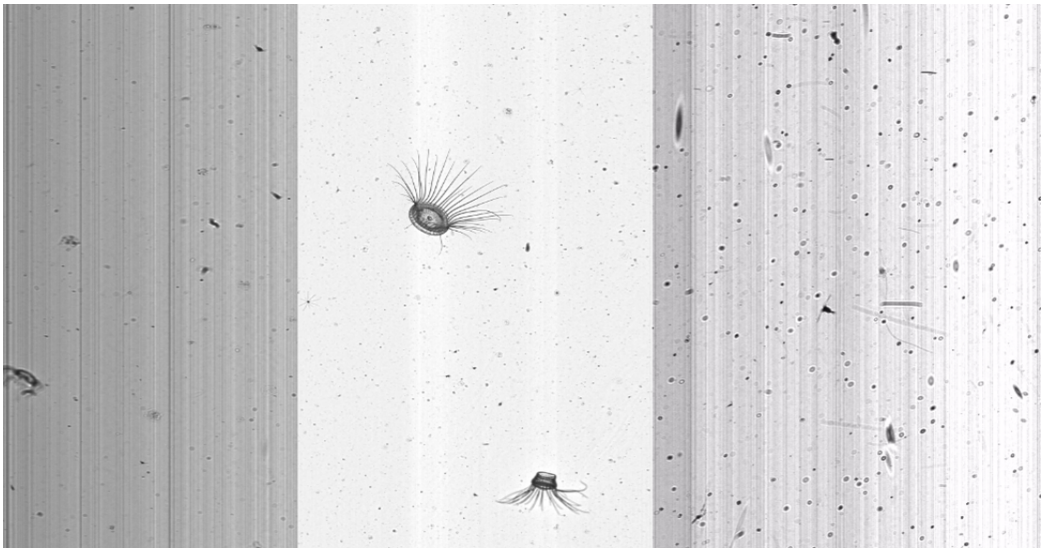


Figure 3.5. ISIIS images from varying water quality

There have been a variety of efforts directed at the computer-based analysis and recognition of plankton images, from a theoretical, general and system-specific perspective (see overview by Benfield et al. 2007 [3]). Recent community efforts have attempted to evaluate the most promising image analysis approaches, with an eye to future developments in computing power and imaging capabilities. A variety of image analysis techniques have resulted in success at categorizing extracted images using both support vector machines (SVMs) that learn vector quantization (LVQ) and artificial neural networks [7, 23, 37]

Existing methods assume the specimens have already been precisely segmented, or aim at analyzing images containing single specimens (extraction of their features and/or recognition of specimens as single targets in-focus in small images. To the best of our knowledge, even software that process images that contain more than one specimen (e.g. ZooImage) cannot be used for large-scale processing of raw data for three main reasons.

1. Such approaches largely depend on the clarity and resolution of the images, which limits their scalability and robustness. Our goal is a recognition system that is robust to noisy information directly from the acquired data, without the need of any human interaction.

2. The recognition is often based on large amounts of manually cropped and labeled specimens that are used for training the classification modules. This means that tedious manual work is required to build extensive training libraries. One of our goals is to minimize the manual effort for such libraries, using 10-30 sample specimens per category.

3. The classifiers used for recognition strongly depend on the "goodness" of the estimated specimens' features (appearance), which implies that only "entirely in-focus" specimens can be reliably recognized (again implying "perfect" data).

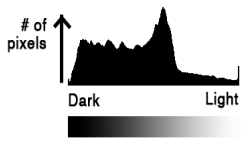
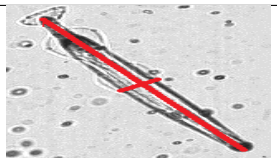
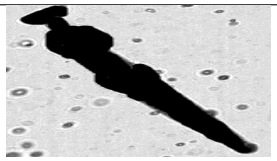
As there are more than 15 classes of plankton, we cannot have a "one vs one" comparision to determine which class the plankton belongs to. Also we cannot build a model for each plankton as the planktons can be deformable and the images are also dependent on their position relative to the ISIIS system. We therefore propose a decision tree based approach that has loose constraints as to how to define a class and how to distinguish between plankton of different classes. We take the help of inter class variance and intra class similarity to create a classification tree from the training samples. The tree is generated in a bottom up fashion and depend on a probabilistic model that decides how similar 2 classes are. A set of granulometric features are extracted from each of the plankton image. The following sections explain the feature extraction, construction of the decision tree and the recognition of plankton species using this decision tree.

3.3.1 Feature Extraction

We extract general granulometric features. As we cannot model the plankton class granulometric features are ideal enough and any additional features might result in overfitting problem where the model becomes so tight that slighly varying shapes of the planktons might be misinterpreted as some other class or an unknonwn class. Features used are in the following table.

Table 3.1

Features extracted

Feature	Description
Histogram	
Min-Max Axis Ratio	
Solidity	$\frac{Area}{ConvexArea}$
Eccentricity	The relative difference in magnitude of the eigenvalues are thus an indication of the eccentricity of the image, or how elongated it is.
Mass	
Hu Moment [38]	moments which are invariant under translation, changes in scale, and also rotation [38].
Transparency Ratio	$\frac{\#\text{pixels in original image}}{\#\text{pixels within the contour}}$
Convex Ratio	$\frac{\#\text{pixels in original image}}{\#\text{pixels in the convex hull}}$
Eigen Value Ratio	$\frac{min(f_1,f_2)}{max(f_1,f_2)}$ where f_1, f_2 are egienvalues of cov(X,Y)

We extract features for the training samples and use these feature vectors in the construction of the Classification Tree. The same set of features are extracted from unknown images for recognition process. The set of featues play an important part in how accurately you are able to classify the plankton.

3.3.2 Recognition

There are many different classes of plankton. We group the classes based on similarity and differences of features. we construct a tree based on these similarities and differences of features. The tree is constructed in a bottom up fashion with each leaf node representing a concrete class. The recognition process can be divided into 2 parts.

1. Building the tree (training) - A bottom up approach
2. Recognition (Testing) - A top down approach

In training, we start from the leaves that are our desired classes. Each class has Feature Vectors. In order to see which 2 classes are highly similar we calculate the feature difference between every 2 classes and select the classes which have the least feature difference. We merge those two classes to make a parent class. This new class is now used again with the remaining classes to find the feature similarity and the process is repeated until we reach the root node. Once we obtain the root node we can start with the recognition process. The leaf nodes represent the concrete classes. The intermediate nodes represent the node generated from similar classes and the edges of the tree represent the features that distinguish the left and right child of the node.

In the recognition process (test) we start from the root node. We first extract the features of the unknown image and calculate the feature difference of this unknown image with the left and right child of the root. We then traverse to the node that

has the minimum feature difference. We repeat this process until we reach a leaf node. This leaf node is then recorded as the class of the unknown image. The figure represents the feature vector from training samples for the various classes.

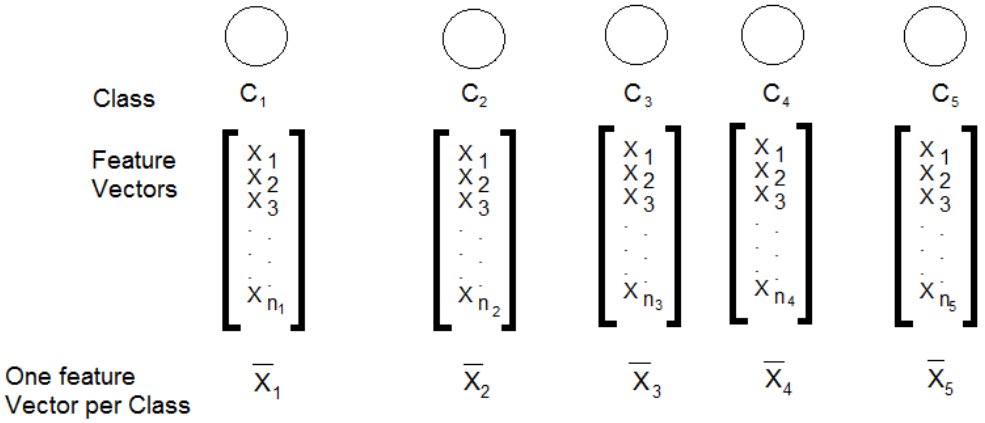


Figure 3.6. Leaf nodes of the tree

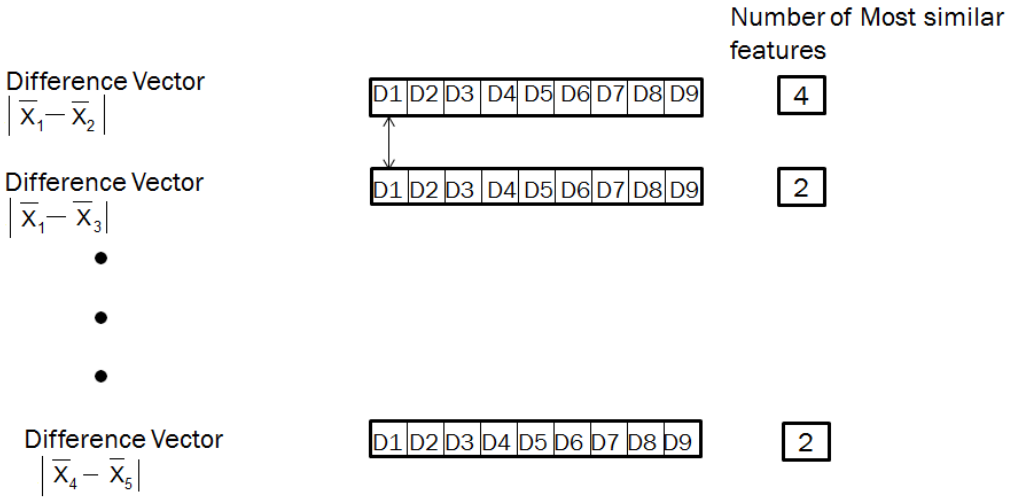


Figure 3.7. Feature Difference

Assuming $\bar{X}_1$ and $\bar{X}_2$ are highly similar, we combine those two classes to form a parent node.

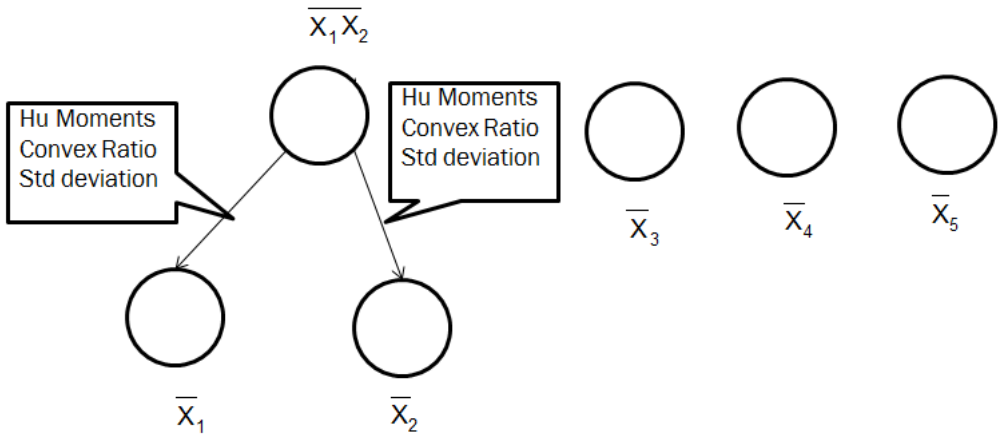


Figure 3.8. Merging 2 leaf nodes to generate an intermediate node

We follow the same process until we obtain the root node. A sample tree is shown below

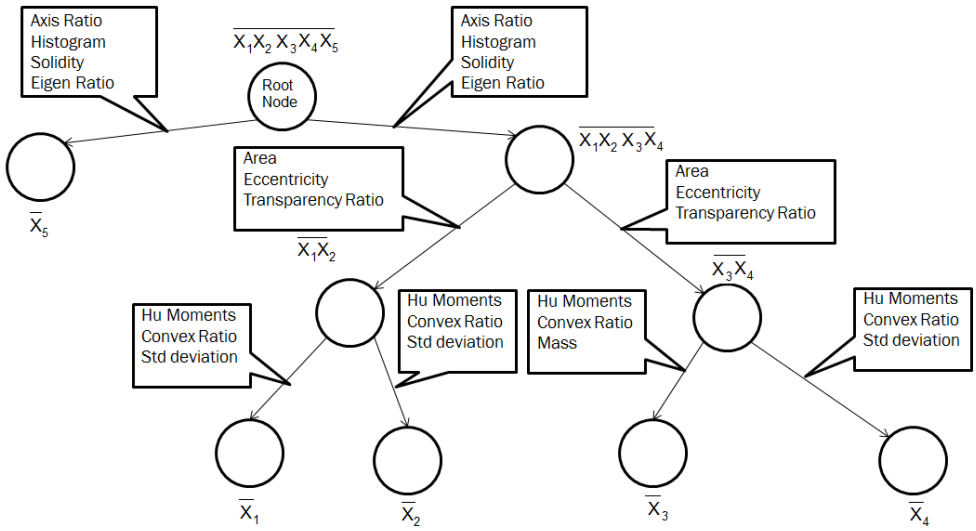


Figure 3.9. Final Tree Generated

Various methods have been proposed and implemented for recognition. These methods do not scale well as the number of classes increase. With many classes we can no longer go for one vs one type of comparision to classify the plankton. The other alternative to this is to use a decision tree. The tree gives us some advantages over the other implementations. The complexity for recognition of each plankton after the tree is constructed is O(lg n). Plankton as we know are deformable and therefore it is hard to model and then fit the plankton. We therefore build an intuitive way of distinguishing and thus classifying the plankton based on feature similarities and differences.

We calculate the feature difference among every 2 class combinations. This feature difference vector is the measurement of how much each feature varies from the other. Each class can contain different number of feature vectors. We combine these feature vectors to get a single feature vector for each class using histogram normalization as it performs better than considering the mean value of the feature vectors. We use histogram difference for the features and this would give us $\frac{n*(n-1)}{2}$ difference vectors. From these feature difference vectors we find which vector has the maximum number of minimum values with respect to each feature.

We can thus obtain the 2 classes that are highly similar with respect to all the other classes. The features that were the minimum are the features that are highly similar and all other features can distinguish between these 2 classes. We merge the features of these 2 classes to form a parent node. The distinguishing features are put on the edges of this parent node which will be used during the recognition testing process. In the list of tree nodes we replace the 2 leaf nodes with this parent node and repeat the same process of feature difference, comparison and merging till there is just 1 node.

The edges in the tree are features that can distinguish between a left child and a right child. Thus this method is an adaptive tree construction which will give different im-

portance to different features (based on level) and requires no additional information about the classes to be provided manually. This approach is highly scalable and works quite well for large number of classes (15-20).

Once the tree is constructed, recognition process is fairly trivial. Firstly the feature vector for the unknown image is calculated. Then we start from the root node and get the probability of that feature value to the left and right child. This probabilistic distribution will give us an idea about the inclination of the class to the left or right node. Only the features on the edges are taken into consideration as the other features are fairly similar. For each feature either the left or right node gets a point based on how close it was to the value of the unknown plankton. We then traverse to the child which gets the most score in this process. We continue the same process until we reach a leaf node. The leaf node is then recorded as the class of the unknown plankton. It can thus be seen that the recognition process takes O(lg n) time on average where n is the number of classes. We only compare the features present on the edges of the tree. Thus this method is very fast. The results and performance of the classification and recognition are explained in the Results section.

3.4 Parallelism

The ISIIS scans a huge volume and transfers around 80Mb/sec. The data is so large that processing them in a sequential fashion would take a lot of time. Also our attempt was to build a system that could be used on the deck of the ship that could process these images at real time. Image processing has a huge scope of parallelism and we wanted to exploit this to our advantage.

Multicore processors have become the new mainstream architecture, and hence require great attention from software developers. Multi-core chips will become ubiquitous in the next few years. Soon embedded systems will have multicore chips as with a small

increase in power, there will be a large increment in computational power. The ability to utilize the full potential of multiple execution cores in a single chip for software has proven to be difficult. Some of the challenges with programming parallel programs are finding and implementing parallelism, dealing with race conditions and deadlocks and eliminating performance bottlenecks. Determining how many cores should be allocated for an application, use of heterogeneous cores for specific applications is also a big challenge. Different types of parallelism can be achieved. These can be fine grained parallelism (Instructions are executed in parallel, these require frequent communication between threads), coarse-grained parallelism (block of codes run in parallel, communication among the threads is not so frequent), Task Decomposition, Data decomposition and data flow decomposition.

Deciding which parallelism would work best for a particular requirement is a tough task and requires a lot of experience. With an improper parallelism option, the speedup could be negative and thus the resulting solution could be much slower than the actual single processor solution. Ideally, multiple cores should offer linear speedup. But more often it is observed that sub-linear is achieved. This is due to the sequential part of the program. If an application has half part parallelized and the other half sequential, the speedup achieved would be only 75% instead of 100%. Other reasons for the sub linearity of the code include hardware bottlenecks. Adding more threads increases the communication and synchronization cost between the threads and could also to contention of resources. This could decrease the throughput of the system.

As we can see, the process of segmenting and recognizing the plankton images is an embarrasingly parallel situation where multiple images can be processed in parallel. We try to introduce as much parallelism as possible to increase the speed of processing. As our final goal is to build a system that can utilize the multicore features if available, we identify parallel sections and test if introducing this parallelism would be beneficial. As these images are high resolution we cannot process a large number

of images in parallel due to memory restrictions. Thus our problem is to determine at what granularity the parallism has to be introduced. We have a bit of coarse grain parallelism where we try to process multiple images at the same time. Some mid-level parallelism where processing of single image is split in parallel and also fine grain parallelism where a set of independent instructions are executed in parallel. The center based clustering methods are a perfect example where low level parallelism can be highly exploited as the pixel level operations can be done in parallel. We have not worked on specialized parallel algorithms for the segmentation but have worked with parallelism as an add-on feature by recognizing the code sections that could be parallelized without compromising on the quality and performance of the overall system. The overall goal was to develop a system that does not necessarily need a multicore system but would be able to utilize them if available. Also the system would be on board the ship which should be able to provide real time processing of the images. this restricts us from using multiple systems and processing the images in a distributed environment.

Apart from the parallelism in the segmentation, the recognition is also parallelized with the system recognizing multiple unknown plankton in parallel. The code for feature extraction has also been improved with the addition of various parallel constructs in the code by extracting different features simultaneously and also optimizing each feature extraction.

3.4.1 Sequential K-means pseudocode

K-means algorithm can be explained easily as below [39].

1. Choose some manner in which to initialize the mi to be the mean of each group (or cluster), and do it.

2. For each example in your set, assign it to the closest group (represented by mi).

3. For each mi, recalculate it based on the examples that are currently assigned to it.

4. Repeat steps 2-3 until mi converge.

3.4.2 Parallel K-means pseudocode

This section adds coarse grain parallelism where multiple images are processed in parallel. The code itself is sequential but each core is assigned a separate entity to work with coarse grain parallelism. Calculating the distance of each pixel from the centroid can be simultaneously done in parallel for all the data points. This can be easily parallelized by parallelizing the for loop.

```
1  #pragma omp parallel sections
2  {
3      #pragma omp section
4      Kmeans(im_{1},k);
5      #pragma omp section
6      Kmeans( im_{2},k);
7      #pragma omp section
8      Kmeans(im_{3},k);
9  }
```

Figure 3.10. Coarse Grain Parallelism

X: a set of N data vectors
C_i : $initializedkclustercentroids$
C : $theclustercentroidsofk-clustering$
$P = p_i \mid i = 1,...,N$ is the cluster label of X
$(C,P) \leftarrow KMEANS(X,C_i)$
repeat
$i \leftarrow 0$
$C_{previous} \leftarrow C_i$;

for all i $\varepsilon[1,N]$ **do**
  $p_i \leftarrow min(x_i,c_j)$;
  $1 \leq j \leq k$
  for all j $\varepsilon[1,k]$ **do**
  $c_j \leftarrow$Average of x_i, whose $p_i = j$;
  end for
end for
until $C = C_{previous}$

(A) Sequential K-means

X: a set of N data vectors
C_i : $initializedkclustercentroids$
C:the cluster centroids of k-clustering
$P = p_i \mid i = 1,...,N$ is the cluster label of X
$(C,P) \leftarrow KMEANS(X,C_i)$
repeat
$i \leftarrow 0$
$C_{previous} \leftarrow C_i$;
#pragma parallel for
for all $i \varepsilon[1,N]$ **do**
  $p_i \leftarrow min(x_i,c_j)$;
  $1 \leq j \leq k$
  $\#pragmaparallelfor$
  for all j $\varepsilon[1,k]$ **do**
  $c_j \leftarrow$Average of x_i, whose $p_i = j$;
  end for
end for
until $C = C_{previous}$

(B) Parallel K-means

Figure 3.11. K-means Pseudocode

3.4.3 Sequential Isodata Clustering

This algorithm is based on the K-means algorithm, and employs processes of eliminating, splitting, and clustering [35].

3.4.4 Sequential Isodata Clustering

Along with processing multiple images at the same time similar to the code shown for K-means we can introduce parallelism in the isodata clustering to speed up the process. As isodata clustering is similar to K-means to find the centroids, we can use the same logic as in the parallel K-means to compute the euclidean distance of each

point with the centroid in parallel. If there are more than 2 clusters that need to be merged, the same can also be done in parallel.

3.4.5 Parallel Isodata Clustering

Along with processing multiple images at the same time similar to the code shown for K-means we can introduce parallelism in the isodata clustering to speed up the process. As isodata clustering is similar to K-means to find the centroids, we can use the same logic as in the parallel K-means to compute the euclidean distance of each point with the centroid in parallel. If there are more than 2 clusters that need to be merged, the same can be also be done in parallel. Thus the pseudocode for this is as shown below.

X: a set of N data vectors
C_i : $initializedkclustercentroids$
C:the cluster centroids of k-clustering
P = p_i $|i = 1,...,N$ is the cluster label of X
$(C,P) \leftarrow KMEANS(X,C_i)$
repeat
$i \leftarrow 0$
$C_{previous} \leftarrow C_i$;
for all $i \; \varepsilon[1,N]$ **do**
$p_i \leftarrow min(x_i, c_j)$;
$1 \leq j \leq k$
for all j $\varepsilon[1,k]$ **do**
count ←number of elements in centroid j
if count nmin$preset_value$ **then**
reassign those vector elements to other clusters to yield k clusters
end if
compute a new cluster center as the average of all the feature vectors in each cluster
end for
end for

• Eliminate clusters that contain less than nmin feature vectors and reassign those vectors to other clusters to yield K clusters.
• Compute a new cluster center as the average of all feature vectors in each cluster.
• For each k^{th} cluster compute the mean-squared error $\sigma_n^2 k$ of each n^{th} component x_n over that cluster and find the maximum $\sigma_{n*}^2 k$ component mean-squared error over within cluster k for over n = 1, , N, where the index n* is for the maximum component.
• If there are not enough clusters $K_{init} < K/2$ and this is not the last iteration, then if $\sigma_{max(k)} > \sigma_{split}$ for any cluster k, split that cluster into two.
• If this is an even iteration and $K_{init} > 2K$,then compute all distances between cluster centers. Merge the clusters that are close than a given value.
until C = $C_{previous}$

(A) Sequential Isodata Clustering

X: a set of N data vectors
C_i : $initializedkclustercentroids$
C:the cluster centroids of k-clustering
P = p_i $|i = 1,...,N$ is the cluster label of X
$(C,P) \leftarrow KMEANS(X,C_i)$
repeat
$i \leftarrow 0$
$C_{previous} \leftarrow C_i$;
#pragma parallel for
for all $i \; \varepsilon[1,N]$ **do**
$p_i \leftarrow min(x_i, c_j)$;
$1 \leq j \leq k$
for all j $\varepsilon[1,k]$ **do**
count ←number of elements in centroid j
if count nmin$preset_value$ **then**
reassign those vector elements to other clusters to yield k clusters
end if
$#pragmaparallelfor$
compute a new cluster center as the average of all the feature vectors in each cluster
end for
end for
• Eliminate clusters that contain less than nmin feature vectors and reassign those vectors to other clusters to yield K clusters. This check can be done in parallel for all the clusters
• Compute a new cluster center as the average of all feature vectors in each cluster.
#pragma parallel for
• For each k^{th} cluster compute the mean-squared error $\sigma_n^2 k$ of each n^{th} component x_n over that cluster and find the maximum $\sigma_{n*}^2 k$ component mean-squared error over within cluster k for over n = 1, , N, where the index n* is for the maximum component.
• If there are not enough clusters $K_{init} < K/2$ and this is not the last iteration, then if $\sigma_{max} k$ σ_{split} for any cluster k, split that cluster into two.
• If this is an even iteration and $K_{init} > 2K$, then compute all distances between cluster centers. Merge the clusters that are close than a given value.
until C = $C_{previous}$

(B) Parallel Isodata Clustering

Figure 3.12. Isodata Pseudocode

3.4.6 Sequential K-harmonic Means Clustering

1. Choose any k points from N.
2. Calculate distances from all points in N to all centers in K.
3. Nmin = minimum distance for to any center for each point in N.
4. Recompute harmonic Averages and update K.

3.4.7 Parallel K-harmonic Means Clustering

As K-harmonic means is similar to K-means and works on each data point individually we can easily parallelize the code along with processing multiple images at the same time. The pseudocode is shown below.

1. Choose any k points from N.
2. Calculate distances from all points in N to all centers in K.
3. Nmin = minimum distance for to any center for each point in N.
4. Recompute harmonic averages and update K.

```
for all j = 0 to n do
   for all i = 0 to k do
      U[j] = U[j] + (Nmin[j]/N[j,I]) ;
   end for
end for
U[j] = U[j]  1;
for all i = 0 to k do
   for all j = 0 to n do
      Q[j,I] = [(Nmin[i]^(p-2)                    *
      (Nmin[i]/N[j,i])^(p+2)]/[(1+U[j]^p)^2]
   end for
end for
for all i = 0 to k do
   for all j = 0 to n do
      QQ[i] += Q[j,I]
   end for
end for
for all i = 0 to k do
   for all j = 0 to n do
      R[j,I] = Q[j,i] / QQ[i];
   end for
end for
for all i = 0 to k do
   for all j = 0 to n do
      R[j,I] = Q[j,i] / QQ[i];
   end for
end for
for all i = 0 to k do
   for all j = 0 to n do
      K[i] = K[i] + R[j,i]*N[j] ;
   end for
end for
```

(A) Sequential K-harmonic means

```
#pragma parallel for
for all j = 0 to n do

for all i = 0 to k do
   U[j] = U[j] + (Nmin[j]/N[j,I]) ;

end for
end for
U[j] = U[j]  1;
#pragma parallel for

for all i = 0 to k do
   for all j = 0 to n do
   Q[j,I] = [(Nmin[i]^(p-2)                    *
   (Nmin[i]/N[j,i])^(p+2)]/[(1+U[j]^p)^2]
   end for
end for
#pragma parallel for

for all i = 0 to k do
   for all j = 0 to n do
      QQ[i] += Q[j,I]
   end for
end for
#pragma parallel for

for all i = 0 to k do
   for all j = 0 to n do
      R[j,I] = Q[j,i] / QQ[i];
   end for
end for
#pragma parallel for

for all i = 0 to k do
   for all j = 0 to n do
      K[i] = K[i] + R[j,i]*N[j] ;
  end for
end for
```

(B) Parallel K-harmonic Means

Figure 3.13. K-harmonic Pseudocode

4 RESULTS

We run our system for the evaluation of the Gulf of Mexico oil spill. By quantifying and classifying the various plankton, the marine biologists can conclude the extent of the effects of the oil spill.

We conducted our experiments on various systems with multicore processors and found that the K-harmonic means provides better balance of accuracy and speed in comparison with the other segmentation approaches. We also noticed that the parallel implementation provides a better speedup compared to sequential approach. We developed the code using OpenCV and OpenMP. The images were of 2048 x 2048 resolution gray scaled images. A single image could contain multiple planktons of the same type or of different types. Quality of each image could be different for each image which makes prediction of noise even more difficult.

4.1 Accuracy

Speed alone is not important. We need to get highly accurate segmentation results to improve the recognition phase.

1. Accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{\text{\# correct predictions of positive and negative instances}}{\text{\# of predictions}} \tag{4.1}$$

2. Recall or true positive rate (TP) is the proportion of positive cases that were correctly identified.

$$TP = \frac{\text{\# correct predictions of positive instances}}{(\text{\# incorrect predictions of negative instances} + \text{\# correct predictions of positive instances.})} \tag{4.2}$$

3. False positive rate (FP) is the proportion of negative cases that were incorrectly classified as positive.

$$FP = \frac{\text{\# incorrect predictions of positive instances}}{(\text{\# correct predictions of negative instances} + \text{\# incorrect predictions of positive instances.})} \tag{4.3}$$

4. True negative rate (TN) is defined as the proportion of negative cases that were classified correctly, as calculated using the equation.

$$TN = \frac{\text{\# correct predictions of negative instances}}{(\text{\# correct predictions of negative instances} + \text{\# incorrect predictions of positive instances.})} \tag{4.4}$$

5. False negative rate (FN) is the proportion of positive cases that were incorrectly classified as negative, calculated using the equation:

$$FN = \frac{\text{\# incorrect predictions of negative instances}}{(\text{\# incorrect predictions of negative instances} + \text{\# correct predictions of positive instances.})} \tag{4.5}$$

6. Precision (P) is the proportion of predicted positive cases that were correct.

$$P = \frac{\text{\# correct predictions positive instances}}{(\text{\# incorrect predictions of positive instances} + \text{\# correct predictions of positive instances.})} \tag{4.6}$$

We used 2000 images to determine the percentage of properly segmented planktons. The results for various algorithms are as follows.

4.2 Segmentation Results

In this section we discuss the timing and accuracy results for the segmentation alone.

Table 4.1
Confusion Matrixfor K-means clustering

		Predicted	
		Negative	Positive
Actual	Negative	300	200
	Positive	180	1340

Table 4.2
Confusion Matrix for iterative K-means clustering

		Predicted	
		Negative	Positive
Actual	Negative	320	110
	Positive	170	1400

Table 4.3
Confusion Matrix for fuzzy C-means clustering

		Predicted	
		Negative	Positive
Actual	Negative	290	190
	Positive	190	1370

Table 4.4
Confusion Matrix for ISODATA clustering

		Predicted	
		Negative	Positive
Actual	Negative	400	90
	Positive	70	1440

Table 4.5
Confusion Matrix for Spectral clustering

		Predicted	
		Negative	Positive
Actual	Negative	410	35
	Positive	25	1530

Table 4.6
Confusion Matrix for K-harmonic means clustering

		Predicted	
		Negative	Positive
Actual	Negative	400	55
	Positive	45	1500

Table 4.7

Segmentation Results

Algorithm	Accuracy	TP	FP	TN	FN	P
K-means	0.82	0.88	0.40	0.60	0.11	0.87
Iterative K-means	0.86	0.89	0.25	0.74	0.10	0.92
Fuzzy C-means	0.88	0.93	0.39	0.60	0.13	0.87
ISODATA	92	0.95	0.18	0.81	0.04	0.94
Spectral	97	0.98	0.07	0.92	0.016	0.97
K-harmonic Means	95	0.97	0.12	0.87	0.02	0.96

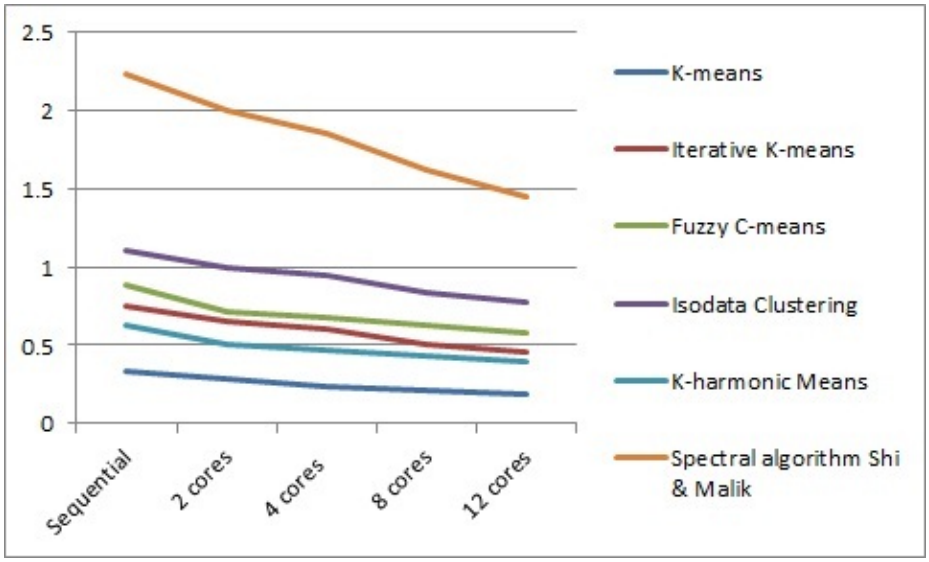


Figure 4.1. Speedup for different cores where the y axis represents the time in seconds

Table 4.8
Percentage scaleup between a sequential process and a parallel implementation run on a 12 core system. The speedup is the ratio of the best sequential running time of that particular algorith to the parallel implementation of that same algorithm.

Approach	Sequential	Parallel Implementation	Speedup
K-means	0.33	0.183	1.80
Iterative K-means	0.75	0.46	1.63
Fuzzy C-means	0.89	0.58	1.53
ISODATA	1.103	0.78	1.41
Spectral	2.233	1.39	1.67
K-harmonic Means	0.624	0.39	1.60

4.3 Classification Results

For classification we first generate the classification tree based on training data. After the tree is generated we supply a set of unknown images to recognize.

Table 4.9
Confusion Matrix for Narcomedusae class

		Predicted	
		Negative	Positive
Actual	Negative	1665	94
	Positive	70	720

Table 4.10
Confusion Matrix for Copepod class

		Predicted	
		Negative	Positive
Actual	Negative	1940	55
	Positive	119	440

Table 4.11
Confusion matrix for 4 classes

Actual-predicted	Narcome-dusae	Jelly	Chaeto-gnath	Copepod
Narcomedusae	780	30	20	16
Jelly	40	290	0	35
Chaetognath	0	0	190	20
Copepod	25	30	20	480

Table 4.12
Confusion matrix for 5 classes

Actual-predicted	Narcome-dusae	Jelly	Chaeto-gnath	Copepod	Appendi-cularian
Narcomedusae	720	30	10	15	15
Jelly	55	340	0	40	0
Chaetognath	0	0	230	0	30
Copepod	24	15	20	440	60
Appendicularian	20	10	50	0	430

The following is the tree generated while with 5 classes. Classes having common parent are highly similar with the distinguishing features on the tree edges. Thus inorder to distinguish between 2 classes of the same parent only the features on the edges are enough and every feature need not be compared.

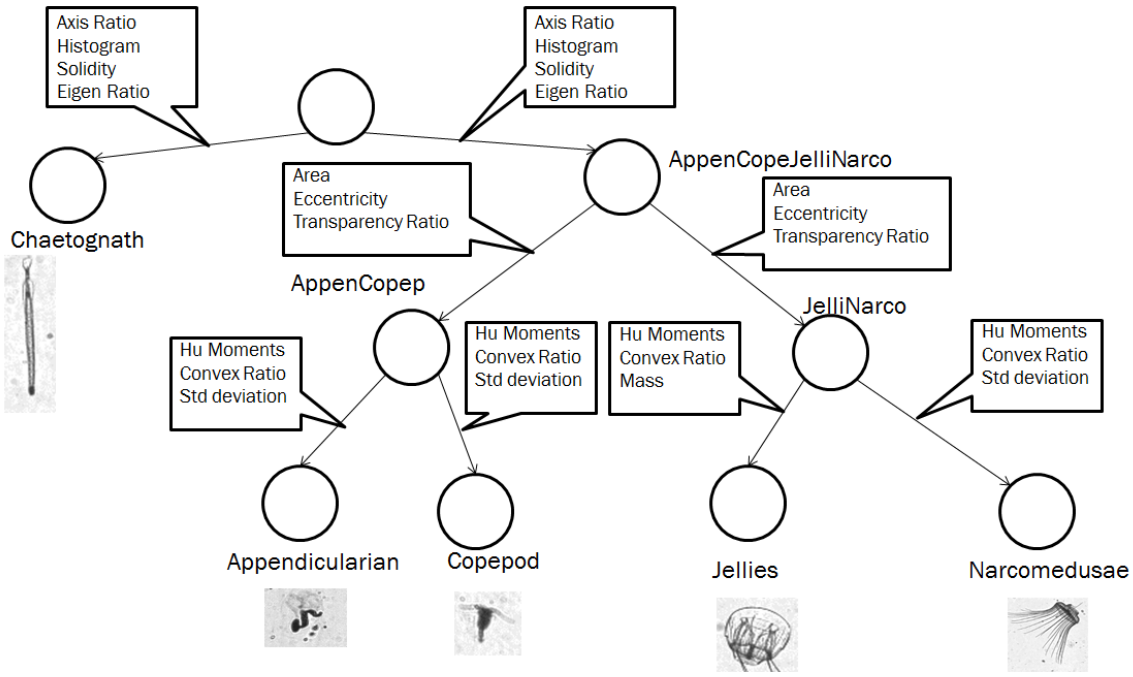


Figure 4.2. Tree for 5 classes

Thus we obtain accuracy upto 95 %. The implementation is not specific on the classes of planktons. The tree generation process is generic and will adapt to the different classes based on how similar or different the features are with respect to the other classes. It takes O(lg n) time to recognize each unknown image. where n is the number of unknown classes. Thus this approach is much better than the existing approaches in that it gives the result in logarithmic time. Also since not all the features are actually compared it fares faster.

Table 4.13

Confusion matrix for 14 classes

Actual-predicted	Appendi-cularians	Chaeto-gnath	Cope-pods	Cteno-phores	Di-atoms	Fish1	Fish2	Narco-medusae	Jelly	Lines	Marine-Snow	Polych-aetes	Shrimp	Siphono-phores
Appendi-cularian	530	40	0	0	10	13	14	5	12	20	9	0	0	10
Chaetognath	35	572	0	0	14	0	15	0	0	15	18	4	8	8
Copepods	30	10	335	5	3	8	6	20	15	2	7	4	18	0
Ctenophores	0	0	5	375	0	5	8	30	45	0	3	7	4	4
Diatoms	20	25	14	2	347	3	6	0	0	19	9	0	4	0
Fish1	15	0	6	0	2	230	26	0	0	0	15	6	12	4
Fish2	5	0	13	0	0	40	240	0	0	0	13	8	14	7
Narcomedusae	15	10	15	15	0	3	0	760	30	0	5	9	11	7
Jelly	0	0	5	45	0	5	6	33	442	0	14	8	10	15
Lines	20	25	5	0	35	0	5	0	0	285	9	7	8	11
MarineSnow	25	23	10	5	8	4	0	5	0	35	253	35	9	42
Polychaetes	2	6	0	3	6	11	5	5	0	0	18	210	20	35
Shrimp	8	0	11	4	0	6	8	11	8	0	11	6	376	7
Siphonophores	3	0	6	0	0	3	10	7	5	4	13	23	11	225

5 SUMMARY

As part of my thesis work we have developed the automated system for processing the large volumes of high resolution ISIIS images. With this automated system we are now in a position to make use of the large data that is collected that we have and can quantify and classify the different plankton accurately and efficiently. We are able to segment and recognize the plankton obtained by sampling water at different depths and different quality of water. Without this system we could not make sense of the large volumes of data obtained from the ISIIS image. Existing systems are able to recognize only 6-8 classes and do not scale well with increasing number of plankton classes. Our approach is highly scalable and yields results at a much higher speed. We also did a comparative study of the various segmentation methods and their performance with respect to our problem and found that K-harmonic means works better than other clustering methods for automatic segmenation of the ISIIS images.

The future work would be to tune the system to improve the accuracy, to add much more levels of parallelism to achieve higher throughput and to see if this system works well in the recognition of other organisms or in other domains. The approach is simple, yet effective and could be targeted to be an industry standard atleast in the plankton recognition system. Along with various performance tuning, feature set could be improved by exploring other features that could improve the accuracy. Also a way to design a system that recognizes new classes instead of misclassification could be a considered a part of future work.

LIST OF REFERENCES

LIST OF REFERENCES

[1] H. V. Thurman. *Introductory oceanography*. Prentice-Hall, Inc., 1997.

[2] C. L. Dybas. On a collision course: ocean plankton and climate change. *BioScience*, 56(8):642, 2006.

[3] M. C. Benfield, P. Grosjean, P. F. Culverhouse, X. Irigoien, M. E. Sieracki, A. Lopez-Urrutia, H. G. Dam, Q. Hu, C. S. Davis, A. Hansen, C. H. Pilskaln, E. M. Riseman, H. Schultz, P. E. Utgoff, and G. Gorsky. Rapid: research on automated plankton identification. *Oceanography Society*, 2007.

[4] A. W. Herman. Simultaneous measurement of zooplankton and light attendance with a new optical plankton counter. *Continental Shelf Res.*, 8:205–221, 1998.

[5] C. S. Davis, S. M. Gallager, M. S. Berman, L. R. Haury, and J. R. Strickler. The video plankton recorder (vpr): design and initial results. *Ergebnisse der Limnologie ERLIA6*, 36:67–81, july 1992.

[6] P. H. Wiebe and M. C. Benfield. From the hensen net toward four-dimensional biological oceanography. *Progress In Oceanography*, 56(1):7–136, 2003.

[7] Q. Hu and C. Davis. Automatic plankton image recognition with co-occurrence matrices and support vector machine. *Marine Ecology Progress Series*, 295(12):21–31, 2005.

[8] R.K. Cowen and C. Guigand. In-situ ichthyoplankton imaging system (isiis): system design and preliminary results. *Limnology and Oceanography Methods*, (6):126–132, 2008.

[9] G. Tsechpenakis, C. Guigand, and R. K. Cowen. Image analysis techniques to accompany a new in situ ichthyoplankton imaging system. In *OCEANS 2007*, pages 1–6, june 2007.

[10] C. Cousin and C. Guigand. Imaging plankton at 5 knots: biological oceanography and near real time processing. *Marine Technology Reporter*, pages 38–40, october 2008.

[11] G. Tsechpenakis, C. Guigand, and R. K. Cowen. Machine vision assisted in situ ichthyoplankton imaging system. *Sea Technology*, 49(12):15–20, december 2008.

[12] R. K. Cowen, Guigand C., C. Cousin, G. Tsechpenakis, S. Chatzis, and A. Greer. An operational in situ ichthyoplankton imaging system (isiis). *OceanObs*, september 2009.

[13] O. Martinez and G. Tsechpenakis. Integration of active learning in a collaborative crf. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08*, pages 1–8, june 2008.

[14] Peter B. Ortner, Leonard C. Hill, and Harold E. Edgerton. In-situ silhouette photography of gulf stream zooplankton. *Deep Sea Research Part A. Oceanographic Research Papers*, 28(12):1569–1576, 1981.

[15] J. Lenz, D. Schnack, D. Petersen, J. Kreikemeier, B. Hermann, S. Mees, and K. Wieland. The ichthyoplankton recorder: A video recording system for in situ studies of small-scale plankton distribution patterns. *ICES Journal of Marine Science: Journal du Conseil*, 52(3-4):409–417, 1995.

[16] P. Tiselius. An in situ video camera for plankton studies: design and preliminary observations. *Marine Ecology-Progress Series*, 164, s. 293-299, 1998.

[17] S. Samson, T. Hopkins, A. Remsen, L. Langebrake, T. Sutton, and J. Patten. A system for high-resolution zooplankton imaging. *Oceanic Engineering, IEEE Journal of*, 26(4):671–676, october 2001.

[18] G. Gorsky, M. Picheral, and L. Stemmann. Use of the underwater video profiler for the study of aggregate dynamics in the north mediterranean. *Estuarine, Coastal and Shelf Science*, 50(1):121–128, 2000. Visualization in Marine Science.

[19] V. Chalvidan, J. P. Chambard, G. Craig, A. Diard, G. L. Foresti, B. Forre, S. Gentili, P. R. Hobson, R. S. Lampitt, P. Maine, J. T. Malmo, H. Nareid, A. Pescetto, G. Pieroni, S. Serpico, K. Tipping, and A. Trucco. High-resolution in situ holographic recording and analysis of marine organisms and particles (holomar). In *OCEANS '98 Conference Proceedings*, volume 3, pages 1599–1603 vol.3, october 1998.

[20] E. Malkiel, O. Alquaddoomi, and J. Katz. Measurements of plankton distribution in the ocean using submersible holography. *Measurement Science and Technology*, 10:1142–1152, december 1999.

[21] X. Tang, W. K. Stewart, H. Huang, S. M. Gallager, C. S. Davis, L. Vincent, and M. Marra. Automatic plankton image recognition. *Artificial Intelligence Review*, 12:177–199, 1998. 10.1023/A:1006517211724.

[22] M. C. Benfield, C. J. Schwehm, R. G. Fredericks, G. Squyres, S. F. Keenan, and M. V. Trevorrow. *Chapter 2: measurement of zooplankton distributions with a high-resolution digital camera system*. CRC Press, 2003.

[23] T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins. Recognizing plankton images from the shadow image particle profiling evaluation recorder. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(4):1753–1762, august 2004.

[24] A. W. Herman, B. Beanlands, M. Chin-Yee, A. Furlong, J. Snow, S. Young, and E. F. Phillips. The moving vessel profiler (mvp): in-situ sampling of plankton and physical parameters at 12 knts and integration of a new laser/optical plankton counter. *Oceanology*, 102:123–135, 1998.

[25] A. W. Herman, B. Beanlands, and E.F. Phillips. The next generation of optical plankton counter: the laser-opc. *Journal of Plankton Research*, 2004.

[26] J. B. MacQueen. Some methods for classifcation and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Mathematical Statist. Probability*, pages 281–297, 1967.

[27] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[28] G. H. Ball and D. J. Hall. Some fundamental concepts and synthesis procedures for pattern recognition preprocessors. *International Conference on Microwaves, Circuit Theory, and Information Theory*, september 1964.

[29] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[30] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, aug 2000.

[31] B. Zhang, M. Hsu, and U. Dayal. K-harmonic means - a data clustering algorithm. *HP Software Technology Laboratory*, october 1999.

[32] L. Shapiro and G. Stockman. *Computer vision*, volume 2004. Prentice Hall, October 2001.

[33] S. Akhter and S. Akhter. *Multi-core programming: increasing performance through software multithreading*. Intel Press, 2006.

[34] E. Ruspini. Numerical methods for fuzzy clustering. *Information Sciences*, 2(3):319–350, 1970.

[35] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[36] R. Nock and F. Nielsen. On weighting clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1223–1235, august 2006.

[37] X. Tang, F. Lin, S. Samson, and A. Remsen. Binary plankton image classification. *Oceanic Engineering, IEEE Journal of*, 31(3):728–735, july 2006.

[38] M. K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, february 1962.

[39] E. Alpaydin. *Introduction to machine learning*. The MIT Press, 2nd edition, 2010.