Spring 5-15-2015

Automatic Classification of Fish in Underwater Video; Pattern Matching - Affine Invariance and Beyond

madhuri gundam
University of New Orleans, madhuguns@gmail.com

Madhuri Gundam
University of New Orleans

Automated Classification of Fish in Underwater Video; Pattern Matching - Affine Invariance and Beyond

A Dissertation

Submitted to the Graduate faculty
of the University of New Orleans
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Engineering and Applied Science
Electrical Engineering

by

Madhuri Gundam

B.Tech. ECE, Jawaharlal Nehru Technological University, 2007
M.S. EE, University of New Orleans, 2010

May 2015

Dedicated to my parents..

Mr. Laxminarayan Gundam and Mrs. Vijaya Gundam

Acknowledgements

I would like to express my deepest appreciation to my advisor, Dr. Dimitrios Charalampidis, for his continuous encouragement and support during my graduate studies. Without his vision and guidance this dissertation would not have been possible. I would like to thank Dr. George Ioup and Dr. Juliette Ioup for their guidance and encouragement during my doctoral career. I would like to thank my committee members Dr. Vesselin P. Jilkov and Dr. Rasheed M. Azzam for their valuable comments. I would like to express thanks to Charles H. Thompson from NOAA National Marine Fisheries Service (NMFS) for the support during the Fish project. I would like to thank NOAA NMFS in Mississippi, USA for providing the underwater video sequences used in this work. A special thanks to my family and friends for their unconditional love and support.

Contents

List of Figures

List of Tables

Abstract

Underwater video is used by marine biologists to observe, identify, and quantify living marine resources. Video sequences are typically analyzed manually, which is a time consuming and laborious process. Automating this process will significantly save time and cost. This work proposes a technique for automatic fish classification in underwater video. The steps involved are background subtracting, fish region tracking and classification using features. The background processing is used to separate moving objects from their surrounding environment. Tracking associates multiple views of the same fish in consecutive frames. This step is especially important since recognizing and classifying one or a few of the views as a species of interest may allow labeling the sequence as that particular species. Shape features are extracted using Fourier descriptors from each object and are presented to nearest neighbor classifier for classification. Finally, the nearest neighbor classifier results are combined using a probabilistic-like framework to classify an entire sequence.

The majority of the existing pattern matching techniques focus on affine invariance, mainly because rotation, scale, translation and shear are common image transformations. However, in some situations, other transformations may be modeled as a small deformation on top of an affine transformation. The proposed algorithm complements the existing Fourier transform-based pattern matching methods in such a situation. First, the spatial domain pattern is decomposed into non-overlapping concentric circular rings with centers at the middle of the pattern. The Fourier transforms of the rings are computed, and are then mapped to polar domain. The algorithm assumes that the individual rings are rotated with respect to each other. The variable angles of rotation provide information about the directional features of the pattern. This angle of rotation is determined starting from the Fourier transform of the outermost ring and moving inwards to the innermost ring. Two different approaches, one using dynamic programming algorithm and second using a greedy algorithm, are used to determine the directional features of the pattern.

Keywords: Pattern Recognition and Matching, Background Segmentation, Object Tracking, Nearest Neighbor Classifier, Affine Invariance, Dynamic Programming, Greedy Algorithm.

1 Introduction

1.1 General

Pattern recognition and matching find numerous applications in the fields of image processing and computer vision [1]. Pattern recognition assigns an object of unknown classification to a particular class based on some similarity measure. Learning methods in pattern recognition can be divided into two types, namely, supervised learning and unsupervised learning. Supervised learning algorithms can be categorized into: classification algorithms and regression algorithms. In classification algorithms, the classifier is trained using the features extracted from each class. This is called the training phase. After having been trained, the classifier accepts the test inputs and assigns labels to them according to the training data. Some of the examples of classification techniques include neural networks, support vector machines, and nearest neighbor algorithms. Unlike supervised learning, the unsupervised learning techniques assign the test input to one of the several clusters without assigning a label to the test input. Some of the unsupervised learning techniques include k-means clustering, self-organizing maps, Gaussian mixture models, and Hidden Markov models.

In contrast to pattern recognition, pattern matching attempts to perform exact matching. The objective of pattern matching is to find a target pattern in the source image. Pattern matching can be divided into: template matching and feature matching [2]. Template matching is applied directly to pixel intensities. Template-based methods are usually employed when the goal is to match an entire pattern with a source image. A brute-force way to match two patterns is by computing the cross correlation between the source image and all geometrically transformed versions of the target pattern. The target pattern may be rotated, scaled, shifted or sheared. Thus geometrically transformed versions of the target are to be considered.

Let an image $s(x,y)$ of size (m_1,m_2) be the source image and $t(x,y)$ of size (n_1,n_2) be the target pattern. The normalized cross correlation (NCC) between the source image, s, and the target pattern, t, is given by Eq. (1).

$$\gamma(u,v) = \frac{\sum_{x,y}(s(x,y) - \bar{s}_{u,v})(t(x-u,y-v) - \bar{t})}{\sqrt{\sum_{x,y}(s(x,y) - \bar{s}_{u,v})^2 \sum_{x,y}(t(x-u,y-v) - \bar{t})^2}} \tag{1}$$

where

$$\bar{s}_{u,v} = \frac{1}{n_1 n_2} \sum_{x=u}^{n_1+u-1} \sum_{y=v}^{n_2+v-1} s(x,y) \tag{2}$$

$$\bar{t} = \frac{1}{n_1 n_2} \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} t(x,y) \tag{3}$$

In Eq. (1), $\bar{t}$ is the mean of all pixel intensities in target pattern, and $\bar{s}_{u,v}$ is mean intensity of the source image in the correlation window (u,v) given by Eq. (2). It can be observed from Eq. (1) that the correlation is performed by sliding the target pattern around the source image. However, NCC is computationally expensive and time consuming.

On the other hand, feature-based matching uses local features, such as edges, corners, lines, curves, extracted from pattern to perform matching. Usually, feature-based matching techniques use a combination of different features to build feature vectors. The feature vectors of the source image are built by taking all geometrically transformed versions of the source image. To perform the matching, the feature vector of the target pattern are compared with the feature vectors of the source image. Feature-based matching is computationally efficient when compared to the template based matching as the only the feature vectors are used for matching.

1.2 Objective of this work

In this work, pattern recognition is used for fish classification in underwater video sequences using supervised classification. The images are gray scale images, obtained from the Gulf of Mexico.

The five species of fish used in this work are Red Grouper, Yellowtail Snapper, White Grunt, Queen Angelfish and Queen Triggerfish. These five species are chosen because of their importance to the Gulf of Mexico ecosystem. Moreover, the features of these species, such as their shapes, make them good candidates to test the performance of the proposed classification system. The underwater video frames are provided by the National Oceanic and Atmospheric Administration, National Marine Fisheries Service in Mississippi, USA.

The second part of this dissertation deals with pattern matching. Although, this work does not attempt to implement a complete pattern matching, an algorithm to complement existing affine estimation algorithms is proposed. Existing algorithms assume that the source and target pattern are associated by an affine transformation. Therefore, several affine transformed variations of the target image can be computed. If the affine parameters can be estimated, then the similarity between two patterns can be determined. However, sometimes, the patterns are not rigid but are deformed to some degree. In such cases, the proposed algorithm can be used to assist in determing the affine parameters, which are then used to perform pattern matching.

The rest of the dissertation is organized as follows. Chapter 2 presents some background information regarding the preprocessing steps to extract fish regions from their surrounding environment, and fish feature extraction using Gabor filters. Moreover, some information about affine transformation, and some of the existing Fourier transform based methods for affine parameter estimation is presented. Chapter 3 presents the proposed classification system to track and classify five species of fish. Finally, Chapter 4 presents the proposed Fourier transform-based pattern matching algorithm.

2 Background

This chapter is divided into two parts: first some background regarding automated classification of fish in underwater video, followed by affine transformation-based pattern matching. The first part presents detailed descriptions of steps to separate the fish regions from their surrounding environment. In addition, two species, namely Red Grouper and Yellowtail Snapper, used in this work have features that are specific to their species. To extract these features, Gabor filtering is used. The second part present the concepts of affine transform and a few existing Fourier transform-based algorithms to estimate the affine parameters.

2.1 Automated Classification of Fish in Underwater Video

Marine biologists use underwater videos sequences to identify and quantify fish, and to study fish behavior and migration patterns. Typically, these videos are analyzed by experts manually. However, manual analysis is a tedious process and consumes many hours to analyze one video. Automating this process, at least partially, will significantly save the time and labor required to analyze the videos. The majority of the previous research has been performed on classifying fish taken out of water [3]-[6] or in human controlled environments, such as in fish tanks with adequate lighting [7]. Moreover, some techniques can classify only one or two species of fish. This section presents brief descriptions of some existing fish classification techniques.

A deformable template matching algorithm used for aligning the test images with training images is presented in [8]. The aligned test images are classified using 2D texture features for two species, namely, *Striped Trumpeter* and *Western Butterfish*. A classification rate of above 80% is reported for about 300 test images. However, extracting texture features from fish in underwater videos might not be always possible due to poor lighting. A edge detection-based algorithm using Sobel and Gabor filters is proposed to classify *Portugese Shark*, *Rough Sagre Shark* and *Smooth Dog Shark* in [9]. In [10], *Scythe Butterfly* fish is classified using a 16 stage Haar classifier with

83 features. Another classification system to detect, track, and classify *Rathbunaster Californicus* from video sequences consisting of *Rathbunaster Californicus*, *Parasticopus Leukothete* and sometimes *Flatfish* and *Rockfish* is proposed in [11]. The algorithm uses feature vectors, based on local jets [12] and modeled as a mixture of Gaussians, are classified using Bayesian classifier with a classification rate of 90%.

A fish classification system to classify *Acanthuridae* and *Scaridae* in videos obtained using a moving camera is proposed in [13]. The algorithm extracts fish using edge detection and performs classification using Zernike moments. As the camera is moving, the background is not constant. It is assumed that either water, in blue color, or the coral, in non-blue color, will be the background against which the fish are present. Therefore, the histogram templates for water and coral are computed from a few manually selected frames. Background of a frame is extracted by dividing the frame into smaller blocks and by comparing its histogram with the two histogram templates and choosing the closest histogram. Classification of fish in videos obtained using a moving camera is an interesting problem. However, the proposed method is a slow process requiring an average of 60 min to analyze a 12 min video sequence.

Other algorithms have attempted to classify more than one or two species of fish. A fish classification technique using shape features, such as fin shapes and their locations, tail shape and body shape and length is presented in [14]. A database of 22 images for 9 species is used. However, the test images are also built from 22 images in database. Each fish is classified by calculating the distance of its feature vector from the feature vectors of the 22 images in the database. A similar classification algorithm proposed in [15] extracts features using Scale-Invariant Feature Transform (SIFT) and Principal Component Analysis (PCA). The algorithm performs clustering using Artificial Immune Network (AIN) and Adaptive Radius Immune Algorithm (ARIA). Finally, classification is carried out using the nearest neighbor classifier. Two separate cases, dead fish and live fish, are considered for classification. To classify dead fish, a database of 162 prototypes of 6 species is created by rotating them. On the other hand, for live fish, the images are acquired at a fish ladder. For this case, 4 species are considered with 12 prototypes for each species. The images in database are divided into sets, one set is used as training data while the other set is used as testing data. The authors report an overall classification rate of 92%. However, the algorithm does not perform any background subtraction. In addition, only single fish are considered for classification at a time.

In [16], a fish classification system using texture features and shape features is proposed. A 320 image database of 10 species of fish is created using affine transformations of the original fish. The technique extracted trajectories of fish motion to study their behavior and detect any unusual behavior. A classification rate of above 90% for 10 species is reported. However, the algorithm was tested for only 10 test images per species. A disadvantage of the works proposed in [5, 14, 15, 16] is that the number fish images are limited. Moreover, the test images are created by rotating or applying affine transformed versions of images in database. In other words, since the test images are artificially generated from training images, the classification performance of these systems is expected to be high.

The following sections 2.1.1 and 2.1.2 present a detailed description of background processing and species-specific feature extraction using Gabor filters.

2.1.1 Background Processing

Background subtraction is used to distinguish the objects of interest from their surrounding environment. Background subtraction is an important step in image processing and computer vision, such as in object detection [17], [18], gesture estimation [19], video survelliance [17], [18]. The main steps involved are computation of the background image from multiple frames, subtraction of the background image from the current frame, and thresholding of the background-subtracted frame [20].

First, to calculate the background image, L consecutive frames, F_l, $l = 1,...,L$, are considered. As the contrast in all the frames does not remain constant, the contrast of all frames is adjusted to that of the first frame, F_1. The background image can be computed by averaging the pixel intensities in L frames. The average of pixels is affected by the extreme pixel intensities, which are caused due to noise or when a fish enters the frame. However, the median is not affected by such fluctuation in pixel intensities. Contrast adjustment is achieved by multiplying all pixels in F_l with M_{F_l}/M_{F_1}. From here onwards, the term *frame* and the notation F_l refer to the contrast-adjusted frames.

To reduce memory resources, the background image is calculated by dividing the set of L frames in groups of N frames, such that $MN = L$. For each group of N frames, a partial background image, $B_m(x,y)$, $m = 1,...,M$, where (x,y) are the horizontal and vertical image coordinates, is calculated as shown in Eq. (4), i.e., a pixel at location (x,y) in $B_m(x,y)$ is computed as the median value of pixels at (x,y) in all M frames. Finally, the background image is calculated as the median image of the M partial background images as shown in Eq. (5).

$$B_m(x,y) = \underset{l}{med}(F_l(x,y)), \forall(x,y) \tag{4}$$

$$B(x,y) = \underset{m}{med}(B_m(x,y)), \forall(x,y) \tag{5}$$

The next step is to subtract each frame from background image, $B(x,y)$. A variance-like measure, $S^2(x,y)$, is calculated for each pixel as shown below. It can be observed that $S^2(x,y)$ is equivalent to the sample variance of pixels at location (x,y) in L frames if the if the median is replaced by the average.

$$S_m^2(x,y) = \underset{l}{med}((F_l(x,y) - B(x,y))^2), \forall(x,y) \tag{6}$$

$$S^2(x,y) = \underset{m}{med}(S_m(x,y)), \forall(x,y) \tag{7}$$

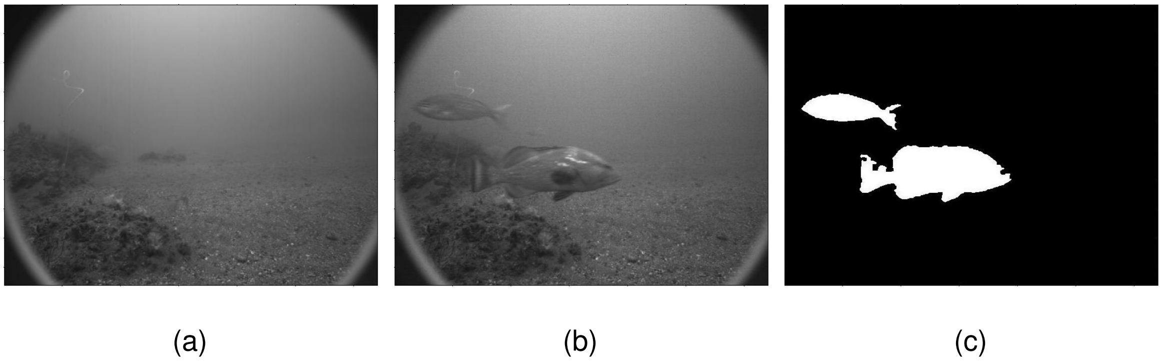


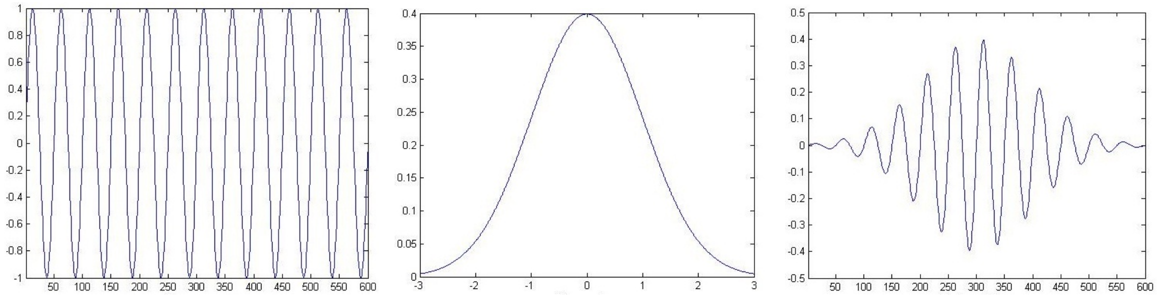
Figure 1: Background processing steps: (a) Background image, (b) An original frame, (c) Thresholded frame

Finally, the background-subtracted frames a thresholded using a user-defined threshold parameter, T. The threshold T determines the pixels that are different from the background pixels. More specifically, if $|F_l(x,y) - B(x,y)| > S(x,y)T$, $F_l(x,y)$ is considered to be associated with a fish. Thresholding results in binary images, where ”white” correspond to the fish and ”black” region corresponds to background. Lastly, to merge regions that are closely located, region growing is used. Regions smaller than a certain number of pixels are considered to be noise and are eliminated. The remaining white regions are the potential fish regions.

2.1.2 Feature Extraction using Gabor Filters

Gabor filters (GF) are linear filters widely used for edge detection, texture segmentation and feature extraction [22]. A GF is a complex sinusoid modulated by a Gaussian as shown in Fig. 2. Eq. (8) represents a spatial GF, where A is a constant, F is the spatial frequency, σ_x^2 and σ_y^2 are the horizontal and vertical standard deviations. A GF oriented vertically exhibits a strong response for horizontal details but smooths vertical details. Similarly, a GF oriented horizontally emphasizes vertical details and smooths horizontal details.

$$h(x,y) = Ae^{(-x^2/2\sigma_x^2 - y^2/2\sigma_y^2)}e^{2\pi jFx_r} \tag{8}$$



(a) (b) (c)

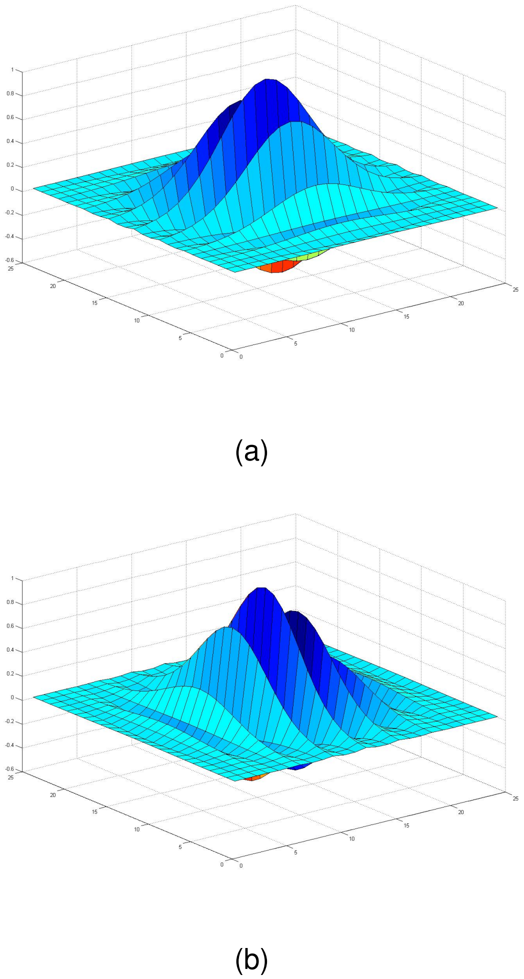
Figure 2: 1D Gabor filter: (a) Sine wave (b) Gaussian function (c) Gabor filter

In [21], Gabor filters are used to extract specific features from EM and OC. EM has a light colored vertical stripe along its caudal fin whereas OC has a yellow colored horizontal stripe extending along its body until the anterior end of its head. The GF is scaled proportional to the area of the fish being tested. In other words, the standard deviation of the filter is adjusted according to the size of the fish, i.e., $\sigma_x^2 = \sigma_y^2 = \alpha(Area)$, where α is a user defined constant, $Area$ is the size of the fish.

Stripe Detection for Epinephelus Morio (Red Grouper):

Epinephelus Morio (EM) has a bright stripe on its tail, a feature that is specific to its species. To detect this feature, the fish is separated from its surroundings by multiplying the original gray scale image with its thresholded image as shown in Fig. 4. Horizontal edges in Fig. 4(c) are detected by filtering the image with vertical GF to obtain $I_{hor}(x,y)$ as shown in Fig. 5(a). Similarly, the vertical edges are detected by filtering Fig. 4(c) with horizontal GF to obtain $I_{ver}(x,y)$ as shown in Fig. 5(b). The image ratio, $I_r(x,y) = I_{hor}(x,y)/I_{ver}(x,y)$, is used to emphasize the stripe as shown in Fig. 5(c). The intensity difference between the fish region and the background causes vertical stripe-like edges around fish. In order to eliminate the effect of vertical stripe-like edges in $I_{hor}(x,y)$ (Fig. 5(a)), a zone of pixels with width equal to $3\sigma_x$ around the fish outline are set equal to zero. The resulting image is shown in Fig. 5(d).

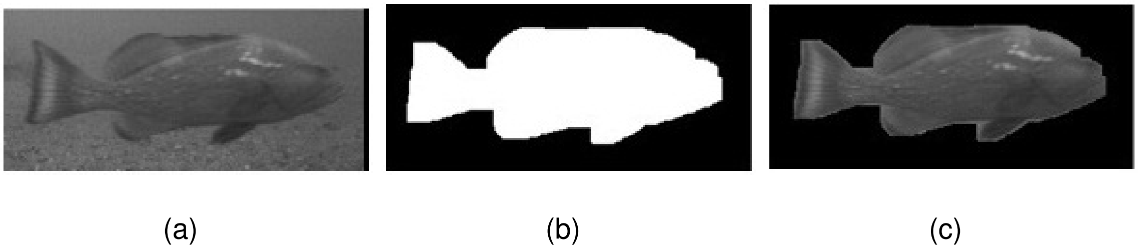
In order to emphasize the presence of vertical stripes, a moving average (MA) filter $f_{MA}(y)$, of size



(a)

(b)

Figure 3: 2D Gabor filters with $\sigma_x = \sigma_y = 4$ oriented (a) Vertically (b) Horizontally

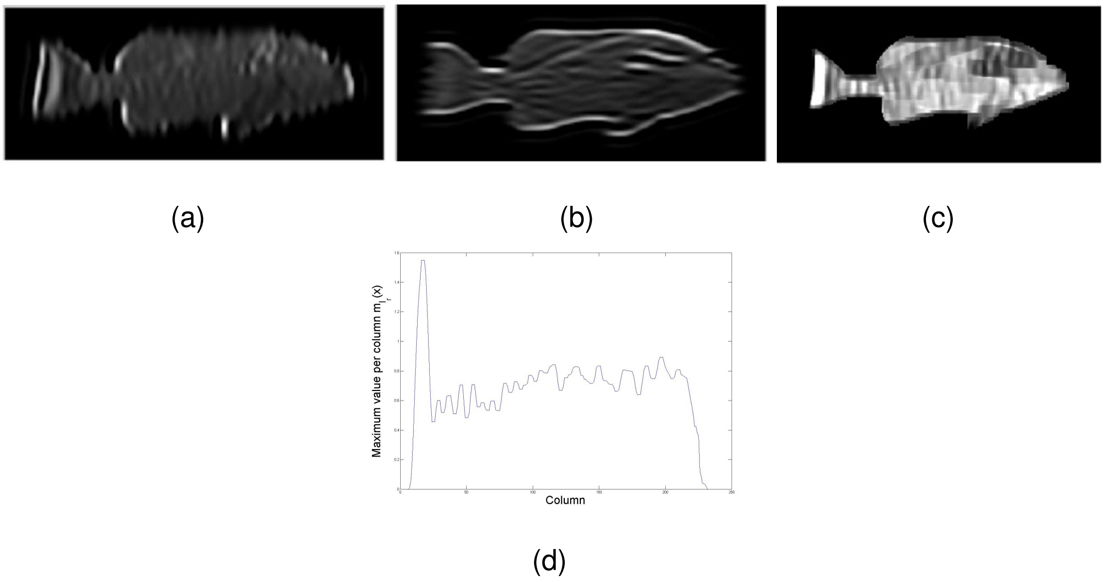


(a) (b) (c)

Figure 4: E. Morio pre-processing steps: (a) Original image (b) Thresholded image (c) Isolated fish region

$W \times 1$ is applied on $I_r^{EM}(x,y)$. The maximum value per column, $m_{I_r}(x)$, is computed for EM as follows:

$$m_{I_r}^{EM}(x) = \max_{y}(f_{MA}(y) * I_r(x,y)) \tag{9}$$


(a) (b) (c)

(d)

Figure 5: (a) Vertical Gabor filtering (b) Horizontal Gabor filtering (c) Stripe detection (d) Maximum value along columns

where ”$*$” represents the convolution operation. The maximum value of $m_{I_r}(x)$, namely $m^{EM}_{I_r max} = \underset{x}{max}(m^{EM}_{I_r}(x))$, quantifies the presence of a stripe in the fish region.

Stripe Detection for Ocyurus Chrysurus (Yellowtail Snapper):

A similar approach is followed to detect the horizontal stripe along the body of Ocyurus Chrysurus(OC). Fig. 6 presents the results for preprocessing steps while Fig. 7 presents the results of GF to detect stripe. However, the ratio image is computed as $I_r(x,y) = I_{ver}(x,y)/I_{hor}(x,y)$, and an MA filter of size $1 \times W'$ is applied along the rows of $I_r(x,y)$. The maximum value for each row is computed as:

$$m^{OC}_{I_r}(y) = \underset{x}{max}(f_{MA}(x) * I_r(x,y)) \tag{10}$$

Sometimes, the fish is oriented at an angle with respect to the x-axis. To detect the stripe in such cases, $I_r(x,y)$ is rotated by different angles, and the largest $m^{OC}_{I_r max}$ is considered.

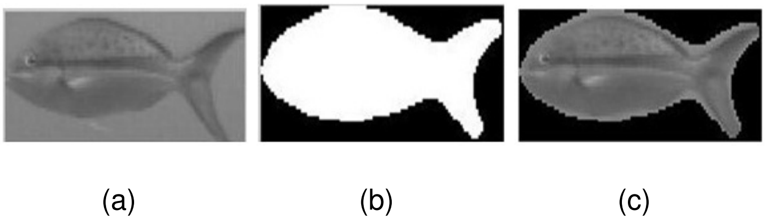


Figure 6: O.Chrysurus pre-processing steps: (a) Original image (b) Thresholded image (c) Isolated fish region

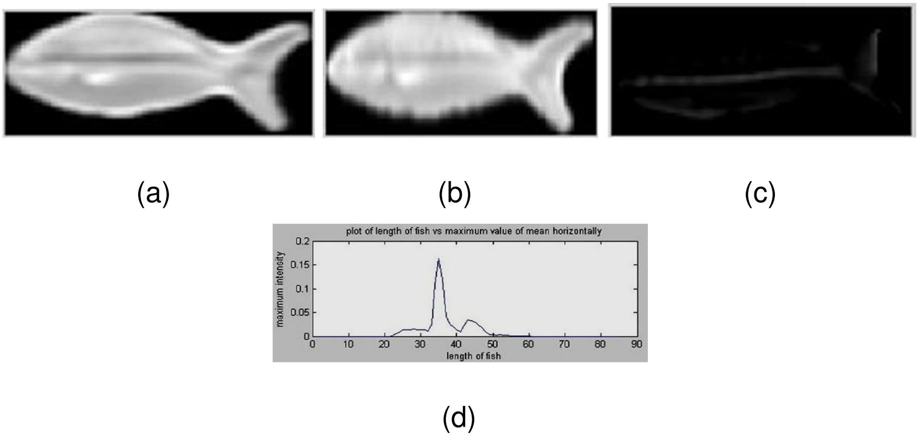


Figure 7: (a) Horizontal Gabor filtering (b) Vertical Gabor filtering (c) Stripe detection (d) Maximum intensity along rows

2.2 Affine Transformation

This section presents the concepts of affine transformation and a few previously proposed Fourier transform (FT) based algorithms to estimate affine transform between a source and a target pattern. By estimating the affine parameters, the similarity between two patterns can be determined.

Affine transformation is a geometric transformation that maps the pixels of input image to new locations in the target image while preserving the collinearity of the points and the distance rations. Affine transformation is a linear combination of translation, rotation, scaling and shearing operations. The standard affine transformation operator, A_t, is defined in Eq.(11), where L and $\mathbf{t}$ respectively represent the linear and translational parts of A_t.

$$A_t \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (11)$$

2.2.1 Affine Transform Estimation

Affine transform parameters can be determined using a brute-force search of 6-D space and by calculating error, e_{st} as shown in Eq. (12), for all possible linear and translational changes, between the target block, f_t, and the source block, f_s. However, brute force approach is computationally expensive and time consuming.

$$e_{st} = |f_t(\mathbf{u}) - T(f_s(\mathbf{u}))| \quad (12)$$

Among the other methods that attempt to estimate the affine parameters, FT-based methods seem to provide a good alternative. Fourier transform decomposes the geometric transform, A_t, defined in Eq. (11), into a linear part, L, which affects only the magnitude spectrum and a translational part, $\mathbf{t}$, which affects only the phase gradient of the Fourier transform. The affine theorem [23] states that given the linear relation, L, between the Fourier transforms of the source block, $F_s(\mathbf{u})$, and the target block, $F_t(\mathbf{u})$, there also exists a linear relationship $(L^{-1})^T$ in spatial domain. In other words, given

$$F_t(\mathbf{u}) = A_t F_s(\mathbf{u}) \quad (13)$$

then

$$F_t(\mathbf{u}) = \frac{1}{detA_t} \mathrm{e}^{i\mathbf{t}u^T} |F((A_t^{-1})^T f_s(\mathbf{u}))| \quad (14)$$

The translational part, $\mathbf{t}$, of A_t can be estimated using the shift theorem, which states that a translational change in spatial domain is transformed to phase change in frequency domain. The translational part, $\mathbf{t}$, can be determined by finding the location at which the correlation between $F_s(\mathbf{u})$

and $F_t(\mathbf{u})$ is maximum.

$$\mathbf{t} = \underset{x,y}{\operatorname{argmax}}\{F^{-1}[\frac{F_s(\mathbf{u})F_t^*(\mathbf{u})}{|F_s(\mathbf{u})||F_t^*(\mathbf{u})|}]\} \tag{15}$$

The remaining part of this subsection presents a few existing FT-based algorithms to estimate the linear part, L, of the affine transformation A_t whereas the translation part is determined using the Shift theorem described above.

2.2.2 Least Square-Based Estimation

Least square-based estimation of affine parameters, estimates the Fourier power spectrum with a Gaussian Mixture Model (GMM) using Levenberg-Marquardt (LM). The algorithm estimates the affine parameters using LM [24], [25]. A Gaussian mixture model is a probability density function represented as a weighted sum of N constituent Gaussian densities as shown in Eq. (16).

$$GMM(\mathbf{u}|\lambda) = \sum_{j=1}^{N} w_j G(\mathbf{u}|\mu_j, \sigma_j) \tag{16}$$

where $\mathbf{u}$ is a data vector, w_j are the weights, μ_j is the mean vector, σ_j is the covariance matrix of component Gaussian densities denoted by $G(\mathbf{u}|\mu_j, \sigma_j)$ and λ is collective representation of the Gaussian parameters σ_j, w_j and μ_j.

The magnitude spectrum, is modeled as a two-dimensional N component Gaussian mixture model with Gaussians centered about the origin, i.e, $\mu_j = 0$, of the spectrum. The Gaussian parameters are estimated using LM by minimizing the squared error between the GMM and magnitude spectrum of the source block, $F_s(\mathbf{u})$, as shown in Eq. (17). Each Gaussian component G_j represents a distinctive directional energy cluster. LM is a conjugate gradient descent method that requires the gradient with respect to the fitting parameters. Therefore, a set of partial derivates of the covariance matrix of the GMM are calculated using Eq. (18) and Eq. (19).

$$\sum_{\mathbf{u}}(G_j(\mathbf{u}) - |F_s(\mathbf{u})|)^2 \tag{17}$$

$$\frac{\partial G_j(\mathbf{u})}{\partial w_m} = \frac{G_j(\mathbf{u})}{w_m} \tag{18}$$

$$\frac{\partial G_j(\mathbf{u})}{\partial \sigma_m^{-1}} = \frac{G_j(\mathbf{u})\mathbf{u}\mathbf{u}^T}{2} \tag{19}$$

Similarly, the LM algorithm searches for the linear transformation, L, by minimizing the squared error between the GMM of the source block spectrum and the spectrum target block as shown in Eq. (20). Although least-square optimization approach is simple, its computational requirement is large and is likely to be trapped in local minima.

$$\sum_{\mathbf{u}}(L(G_j(\mathbf{u})) - |F_t(\mathbf{u})|)^2 \tag{20}$$

$$\frac{\partial G_j(L(\mathbf{u}))}{\partial L} = -G_j(L(\mathbf{u}))\sum_{m}^{N}\sigma^{-1}L\mathbf{u}\mathbf{u}^T \tag{21}$$

2.2.3 Phase Correlation-Based Estimation

A phase correlation-based technique to estimate rotation, scaling and translation parameters is proposed in [26]. Rotation and scale can be efficiently estimated using the Fourier-Mellin estimation by using phase-correlation relationship. The magnitude spectrum is translation invariant, i.e., any translation change is reflected in phase gradient of the Fourier transform. Moreover, a rotation of source block, $f_s(\mathbf{u})$, rotates its magnitude spectrum, $|F_s(\mathbf{u})|$, by same angle and scaling of $f_s(\mathbf{u})$ by σ scales $|F_s(\mathbf{u})|$ by factor of σ^{-1} as in Eq. (23). By converting the image to polar coordinates, (θ, r), rotation is converted to translation along angular, θ, axis. In addition, by converting the radial axis to logarithmic scale, scaling is reduced to translation along radial axis.

As mentioned earlier, using polar-logarithmic representation, both image rotation and scaling can be reduced to translation as shown in the Eq. (25), where θ_0 is the rotation angle, σ is the scaling factor, and x_0 and y_0 are translations along x and y directions. Although this method accurately computes scaling, rotation and translation, it fails to estimate shear deformation.

$$f_t(x,y) = f_s[\sigma(x\cos\theta_0 + y\sin\theta_0) - x_0, \sigma(-x\sin\theta_0 + y\cos\theta_0) - y_0] \tag{22}$$

$$F_t(m,n) = \mathrm{e}^{-j2\pi\phi}\sigma^{-2}|F_s[(m\cos\theta_0 + n\sin\theta_0)/\sigma, (-m\sin\theta_0 + n\cos\theta_0)/\sigma]| \tag{23}$$

$$|F_t(\theta,r)| = \sigma^{-2}|F_s(\theta-\theta_0, r/\sigma)| \tag{24}$$

$$|F_t(\theta,r)| = \sigma^{-2}|F_s(\theta-\theta_0, \log r - \log\sigma)| \tag{25}$$

$$F_t(v,w) = \sigma^{-2}\mathrm{e}^{-j2\pi(v\log\sigma + w\theta_0)}F_s(v,w) \tag{26}$$

2.2.4 Spectral Alignment-Based Estimation

A Spectral alignment-based method assuming the presence of double energy clusters in the Fourier spectrum is proposed in [25]. An illustration of double energy clusters in spectrum is shown in Fig. (8). By identifying the centroids of each energy clusters, the affine transformation between two blocks can be solved for using Eq. (29). The two representative centroids are identified by performing angular variance analysis to find the angles, θ_1 and θ_2, for which the sum of the variances, $\sigma^2(\theta_1,\theta_2)$, is minimum.

$$\sigma^2(\theta_1,\theta_2) = \frac{1}{E_\theta}\sum_{\mathbf{u}\varepsilon\alpha_{(\theta_1,\theta_2)}} |F(\mathbf{u})|.||\mathbf{u} - \mu(\theta)||^2 \tag{27}$$

$$\mu(\theta) = \frac{1}{E_\theta}\sum_{\mathbf{u}\varepsilon\alpha_{(\theta_1,\theta_2)}} |F(\mathbf{u})|\mathbf{u} \tag{28}$$

where $F(\mathbf{u})$ is the Fourier spectrum, $\alpha_{(\theta_1,\theta_2)}$ represents the coordinates of the half plane subdivision

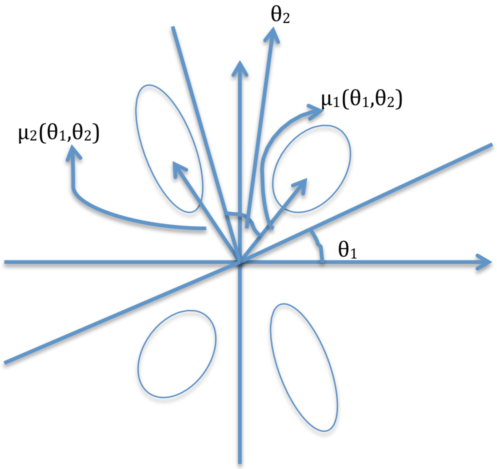


Figure 8: Double energy clusters

defined by the angles θ_1 and θ_2, E_θ is the sum of energy in the half plane, μ is the representative centroid of the corresponding subdivision. The linear part, L, can be directly computed by aligning the centroids of the segments of source block, (μ_{sx}, μ_{sy}), and target block, (μ_{tx}, μ_{ty}). The affine parameters can be easily computed by aligning the centroids. However, this method fails if the spectrum does not contain double energy clusters.

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} = \begin{bmatrix} \mu_{tx}^T & \mu_{ty}^T \end{bmatrix} \begin{bmatrix} \mu_{sx}^T & \mu_{sy}^T \end{bmatrix}^{-1} \tag{29}$$

2.2.5 Fourier Slice-Based Estimation

A Fourier slice-based affine parameter estimation method is proposed in [27]. Fourier slice at an angle θ is calculated using the Radon projection, $p(\theta)$, which is the 1-D integral of a function, $f(x,y)$, along angle θ as shown in Eq. (30) [28]. Projection Slice theorem, which states that the Fourier transform of Radon projection at angle θ is a slice of Fourier spectrum, $F(u,v)$ through the origin at angle θ, is used to calculate the Fourier slices as shown in Eq. (31).

$$p_\theta(r) = \int_\theta f(x,y)dl \tag{30}$$

$$F_\theta(k) = \sum_{n=0}^{N-1} p_\theta(r) \mathrm{e}^{-i2\pi kn/N} \tag{31}$$

For every Fourier slice, a 1-D Gaussian window, centered on the DC component is computed using Eq. (32), where σ represents the deviation of Fourier coefficients in a slice, and N is the number of Fourier coefficients. Every Gaussian window is weighted as per Eq. (34).

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \tag{32}$$

$$\sigma = \sqrt{\frac{2}{N} \sum_{i}^{N/2} (x_i)^2} \tag{33}$$

$$w_\theta = \frac{\sum_r |F_C(x,y)| \delta(x\cos\theta + y\sin\theta) G_\theta(r)}{\sum_\theta \sum_r |F_C(x,y)| \delta(x\cos\theta + y\sin\theta) G_\theta(r)} \tag{34}$$

The next step is to find two representative Gaussians. The two Gaussians with highest weights corresponding to the significant directional features are determined using Eq. (35). First, each polar bin is populated with Gaussian components. In each bin, the Gaussians with highest weight (> 0.1), are selected. The two Gaussians with highest weights are selected as the representative slices. The bins containing Gaussians with very small weights do not correspond to any significant features. Therefore, such bins are treated as empty bins.

$$B_i = \{G_j | \frac{(2i-1)\pi}{2k} + \theta(G_M) < \theta(G_j) \leq \frac{(2i+1)\pi}{2k} + \theta(G_M)\} \tag{35}$$

In Eq. (35), the parameter k is set to 3/8 times the size of image as proposed in [27], l is the number of gaussians, $(0 \leq j \leq l)$, $(0 \leq i \leq k-1)$, $\theta(G_j)$ is the direction of gaussian component, G_j and G_M is the gaussian component with highest weight, w_θ. Finally, the coordinates of the centroids of the two representative slices are computed, mapped to rectangular coordinate system and used to calculate the linear part, L, of affine transform using Eq. (29). As mentioned earlier, this algorithm is based on Radon projection, which computes the integral of a function along a straight line at an angle, θ. Therefore, the technique fails when the high intensities of Fourier transform do not fall on straight lines.

3 Automated Classification of Fish in Underwater Video

This work proposes a classification system to automatically track and classify fish in underwater video sequences. The system can identify and classify five species, namely Epinephelus Morio (EM), Ocyurus Chrysurus (OC), Balistes Vetula (BV), Holacanthus Ciliaris (HC) and Haemulon Plumierii (HP), found frequently in the Gulf of Mexico. Their common names are respectively Red Grouper, Yellowtail Snapper, Queen Triggerfish, Queen Angelfish and White Grunt. Chapter 2.1.1 presented a detailed description of background subtraction and thresholding, as well as feature extraction using Gabor filters. This chapter presents tracking, which associates multiple views of a fish in consecutive frame, feature extraction using Fourier descriptors to describe the shape, which are then presented to a classifier for classification. Lastly, a probabilistic-like framework to classify an entire sequence using the combination of the shape based classification results and Gabor filter results is presented.

3.1 Tracking

Tracking is used to detect the path of a fish from the point it enters until the point it exits the camera's field of view. Thus, providing information about a fish from multiple frames. The white regions obtained after thresholding the background-subtracted image are the potential fish regions. An example is shown in Fig. 1, where white regions represent the potential fish regions.

Tracking is useful especially when one single view of a fish is not suitable for classification. For example, when a fish is swimming towards the camera, it is difficult even for a human expert to identify the species. In such cases, it is assumed that fish will eventually turn providing a side view to the camera lens. Thus as the fish is tracked from frame to frame, it is expected to provide at least one good view for classification. Moreover, as the number of good views of a fish increases, the classification confidence of the proposed system increases.

In this work, Kalman filter is used for tracking fish by assuming a constant velocity model [29, 30]. The *state vector* has information about the fish center, (c_x, c_y), the coordinates of the top left and bottom right points, (bt_x, bt_y) and (bb_x, bb_y), of the rectangular bounding box enclosing the fish region, and the velocity of the center (v_x, v_y). Therefore, the state vector of the fish in the $l-th$ frame is $s_l = [c_x^{(l)}, c_y^{(l)}, v_x^{(l)}, v_y^{(l)}, bt_x^{(l)}, bt_y^{(l)}, bb_x^{(l)}, bb_y^{(l)}]^T$. The bounding box endpoints are chosen since the size of the bounding box is associated to the size of the fish area. A separate Kalman filter is assigned to each fish region, which are tracked simultaneously.

The Kalman filter operations can be divided into two stages: prediction and correction. During the prediction stage, the filter obtains the *a priori* estimate for the current state, $\hat{s}_l^-$, using the previous state estimate, s_{l-1}, as shown in Eq. (37). The vector of observations, z_l, shown in Eq. (38) includes the same variables as the state. However, the variables in z_l are determined directly from the image data. The first time a fish region enters the camera's field of view, its state variables are initialized to the real observations, z_l, except the velocities which are set equal to zero. The state transition matrix, A, relates the previous state, s_{l-1}, with the present state, s_l.

In order to associate the i-th fish region in frame F_{l-1}, namely $R_i^{(l-1)}$, to one of the regions in frame F_l, namely $R_j^{(l)}, j = 1, .., J$, the Euclidean distances, E_{ij}, between the estimates of the center coordinates of region $R_i^{(l-1)}$ that are obtained from state estimate $\hat{s}_l^-$ of $R_i^{(l-1)}$ and the center coordinates determined from observations z_l of all $R_j^{(l)}$ are computed as shown in Eq. (36). The $R_j^{(l)}$ associated to the smallest distance, E_{min}, is associated to $R_i^{(l-1)}$ only if E_{min} and the size difference of the two regions are each smaller than user-defined thresholds. Otherwise, it is assumed that either a region splitting or a merging has occurred, and a new Kalman filter is assigned to track the new region.

$$E_{ij} = \sqrt{(c_{ix}^{(l-1)} - c_{jx}^{l})^2 + (c_{iy}^{(l-1)} - c_{jy}^{l})^2} \tag{36}$$

During the correction stage, the filter corrects $\hat{s}_l^-$ using the respective z_l to obtain the a posteriori estimate, $\hat{s}_l$. The estimate $\hat{s}_l$ is used as $\hat{s}_{l-1}$ in the next frame. The measurement matrix, H, in Eq.

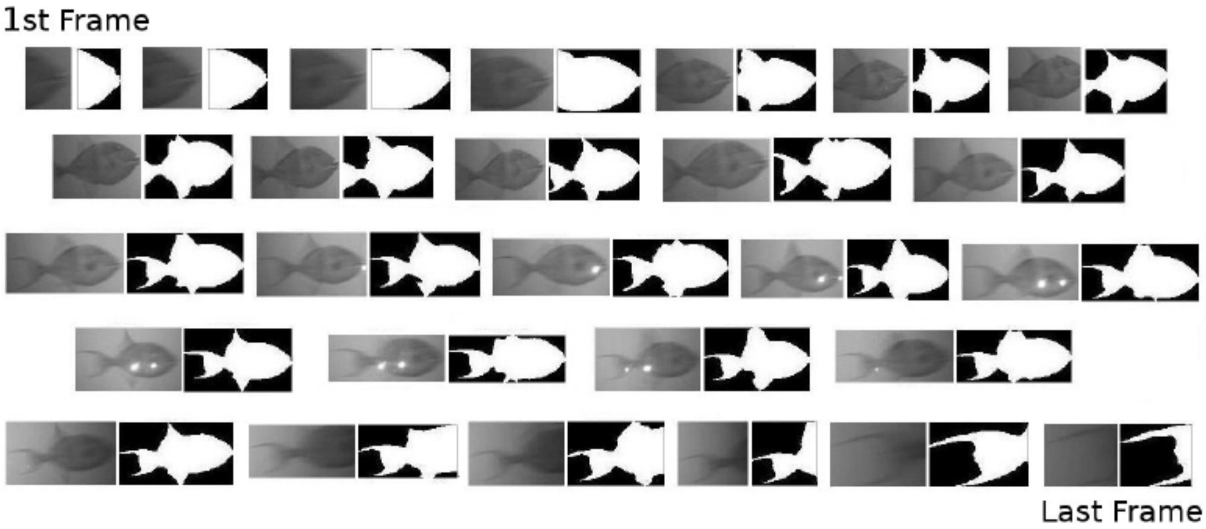


Figure 9: Tracking a fish in consecutive frames

(38) associates state predictions with observations, z_l, and K is the Kalman gain.

$$\hat{s}_l^- = A\hat{s}_{l-1} \tag{37}$$

$$\hat{s}_l = \hat{s}_l^- + K_l(z_l - H\hat{s}_l^-) \tag{38}$$

As the fish regions are tracked, they are classified on-the-fly using the nearest neighbor classifier presented in section 3.3. An example of tracking a fish from the frame it enters camera's point of view untill it exists is presented in Fig. 2. In this example, the top leftmost image shows the fish as detected for the first time. The black and white image to its right is the corresponding thresholded image obtained after background subtraction. Moving from left to right, and then from top to bottom, pairs of images show the same fish and its corresponding thresholded image as seen in consecutive frames. The identification of the bounding box enclosing the fish, as well as the association of the fish region from one frame to the next are performed automatically by the algorithm.

In some cases, background processing is unable to detect a fish using the original threshold, T, although the fish is still present in the frame. As a result, the Kalman filter fails to track the fish. In this situation, it is assumed that the fish is present in the same area. In an attempt to separate such fish from background, variable-thresholds are used for background subtraction. More specifically, a threshold, $T_2 > T$, is first applied. This threshold is decreased until the fish region is detected or until a specified minimum threshold, $T_n < T$ is reached. The search is performed in the region enclosed by the predicted bounding box specified by the coordinates, (bt_x, bt_y) and (bb_x, bb_y), obtained from Eq.(37). If a region is detected, it is associated with the sequence and is tracked in the next frame. However, this region is not classified. If the missing fish is not found for 5 consecutive frames, it is assumed that the fish has exited the camera's field of view. Therefore, the Kalman filter associated with the fish is terminated. If the same fish is found again in the frame, it is assigned a new Kalman filter and is tracked as a new fish.

3.2 Micro Feature Extraction

This section describes the shape features extracted from fish that are used to classify fish. The micro features consists of Fourier descriptors, representing the shape of a fish, and features specific to species EM and OC described in Section 2.1.2.

3.2.1 Fourier Descriptors

Fourier descriptors (FD) represent an object in the frequency domain [31, 32, 33, 34]. The FDs of a shape are computed using the coordinates of its boundary points. Consider an P-point outline with (x_p, y_p) as the horizontal and vertical coordinates of the p^{th} point, and $c_p = x_p + iy_p$, as the complex representation of the point coordinates. The FDs are calculated as shown in Eq. (39).

$$D(k) = \sum_{p=0}^{P-1} c_p e^{-j2\pi pk/P}, k = 0, ..., P-1 \tag{39}$$

The FDs of a fish region are calculated using all points on the boundary. The lower- and higher-frequency descriptors represent, respectively, the general shape and the finer details of the shape. As fish regions tend to have relatively smooth edges, the significantly high-frequency FDs can be ignored. Moreover, by ignoring the FD phases, the absolute FD values, $|D(k)|$, are made invariant to rotation of fish.

To make the FDs invariant to fish size, the FDs magnitudes are normalized using the average of $D(1)$ and $D(P-1)$. As fish tend to have relatively smooth boundary, only the FDs corresponding to lower frequencies, i.e., for $k = 1, ..., 40$ and $k = P-40, ..., P-1$, are used in this work. The FDs corresponding to EM, HP, OC, BV and HC are shown in Fig. (10).

3.3 Classification

This section presents the Nearest Neighbor classifier (NNC) used to classify the feature vectors consisting of absolute normalized FD values. Based on the NNC algorithm, feature vectors with known classification (exemplars) are used to represent each of C classes. The FDs of the EM, HP, OC, HC and BV shown in Fig. (10) are used as the exemplars. The distance measure between a feature vector with unknown classification and each of the exemplars is used to classify the feature vector, and therefore its associated object, to one of the classes [35]. The q-th exemplar of the c-th class is defined as $D_{c,q}$, where $c = 1, ..., C$, and $q = 1, ..., Q$. In this work, the total number of classes, $C = 5$, for EM, HP, OC and BV the number of exemplars, $Q = 3$ whereas $Q = 4$ for HC.

As mentioned earlier, to classify the feature vector, a weighted Euclidean distance is used to measure the distance between the unknown classification and exemplars. The square of the distance is defined as shown in Eq. (40). The weight, w_c, is the average square Euclidean distance between c-th class, D_c, and a few FD vectors extracted from subjectively ”good” views of fish that are known to belong to class c. Essentially, w_c is the sample variance associated with the multivariate feature

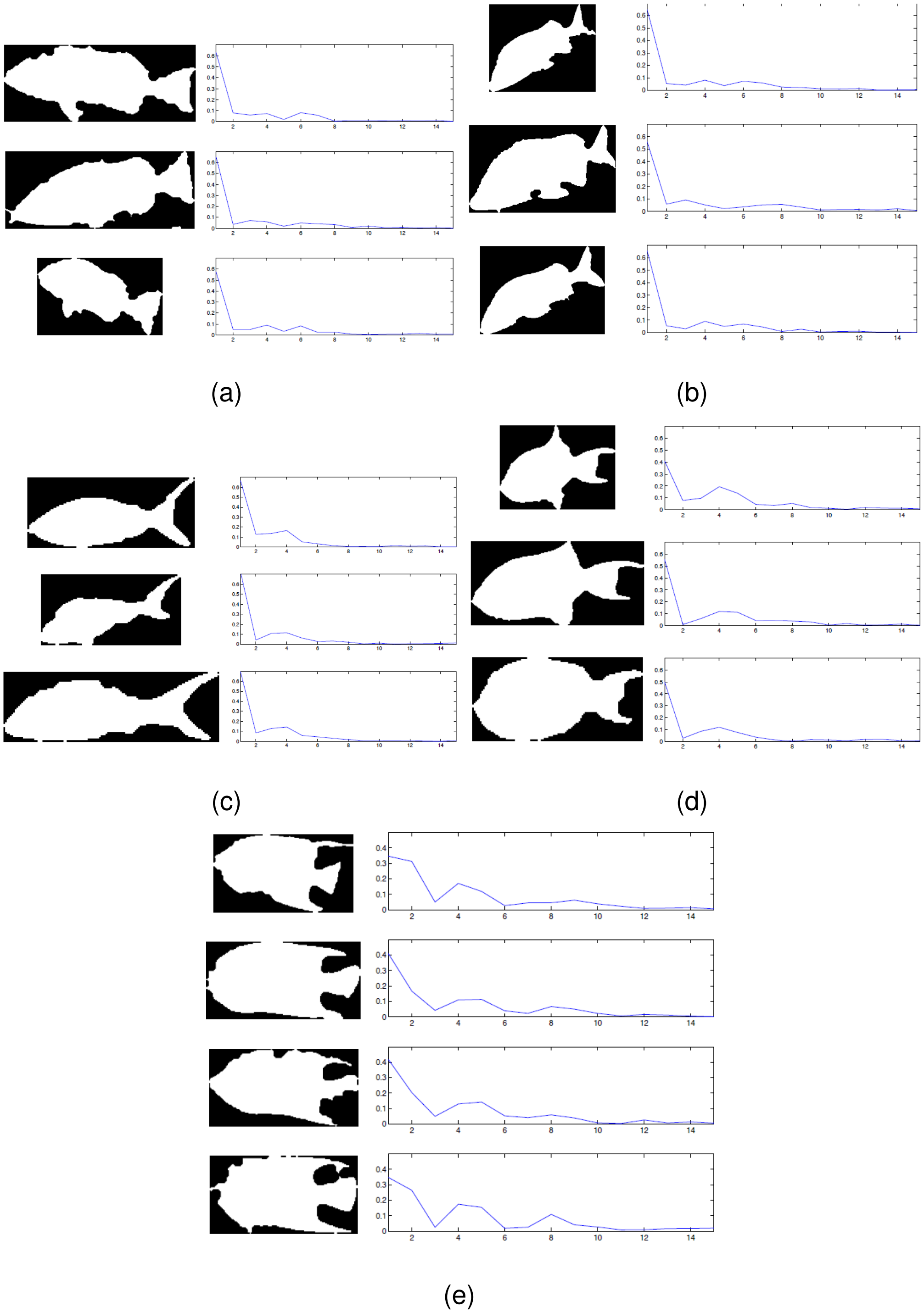


Figure 10: Templates and FDs (a) E. Morio (b) H. Plumieri (c) O. Chrysurus (d) B. Vetula (e) H. Ciliaris

distribution of class c, assuming that the covariance matrix of such distribution is diagonal with all diagonal elements equal to w_c. A fish region with corresponding vector D_t is temporarily assigned to the class c of minimum $dist^2(D_c, D_t)$.

$$dist^2(D_{c,q}, D_t) = \frac{1}{w_c}(D_{c,q} - D_t)^T(D_{c,q} - D_t) \tag{40}$$

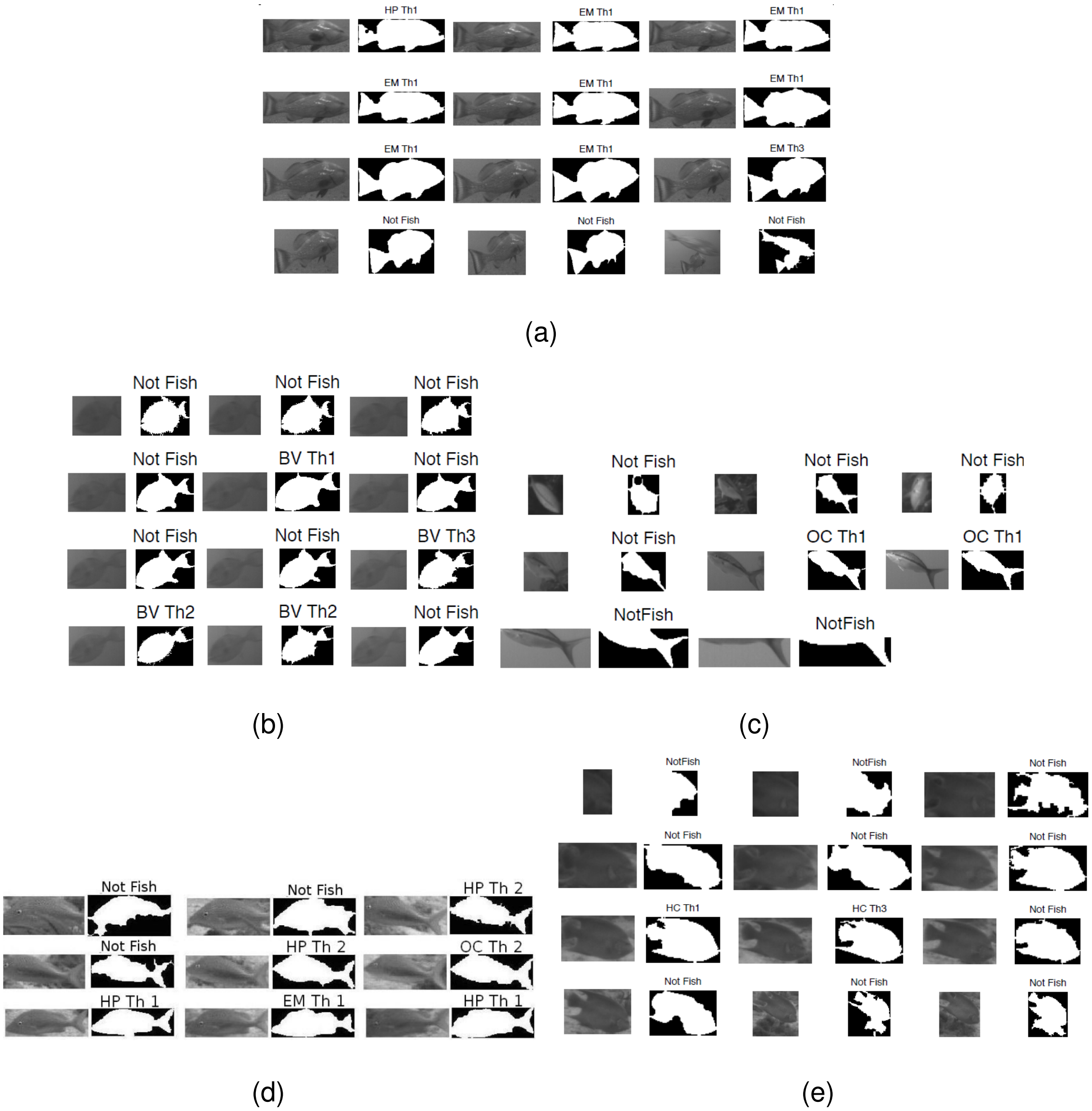


Figure 11: Microfeature classification results (a) E. Morio (b) B. Vetula (c) O.Chrysurus (d) H. Plumieri (e) H. Ciliaris

Three different thresholds, Th_1, Th_2, Th_3 are used for the distance given by Eq. (40). The three thresholds are chosen to be $Th_1 = 0.56$, $Th_2 = 0.59$ and $Th_3 = 0.62$. If the distance, $dist^2(D_{c,q}, D_t)$

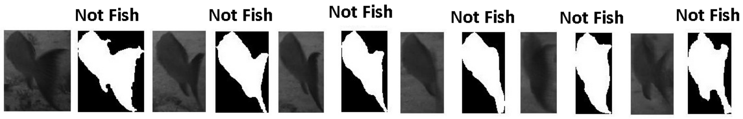


Figure 12: NNC results for pose unsuitable for classification

is greater than the highest threshold Th_3, the region is labelled as 'Not Fish'. Fig. 11 presents examples of classification for EM, BV, OC, HP and HC sequences. In every sequence, the top left most image is the fish as detected for the first time and the binary image to its right is its corresponding image obtained after background subtraction. Moving from left to right, the pairs of gray scale and binary images represent the same fish in consecutive frames. The labels in the figures indicate the threshold which was satisfied for classification. For example, in Fig. (11(c)), the label 'OC Th1' indicates that the fish has been classified as OC according to its shape based on threshold Th_1.When NNC does not classify a region as one of the five classes, the region is labelled as 'Not Fish', which implies that the region does not correspond to a fish or at least not to a fish of interest in this work. Moreover, the fish regions, which are present at the borders of the frame are also labelled as 'Not Fish' as they are usually incomplete and therefore are not suitable for classification.

As mentioned earlier, in addition to FDs, the species-specific features, $m^{EM}_{I_r max}$ and $m^{OC}_{I_r max}$, are also extracted for every region as described in section 2.1.2. Section 3.5 describes how NNC results and $m^{EM}_{I_r max}$ and $m^{OC}_{I_r max}$ are combined together to classify an entire sequence.

3.4 Adaptive Region Segmentation using NNC and Morphological Operations

An adaptive region segmentation approach is used for the regions that are labelled as 'Not Fish' by the NNC classifier, i.e., only the fish which are not classified by NNC are subjected to adaptive morphing operations and background thresholds. As mentioned earlier, the background subtracted images are subjected to dilation, erosion and region filling operations to remove noise and merge closely located regions of the same fish. However, sometimes the shapes extracted using the original threshold T are not suitable for NNC classification. Therefore, an adaptive approach using variable background thresholds and morphological operations is used to determine the shapes of the final region.

In addition to the original threshold used for background subtraction, *T*, two additional thresholds, *T'* and *T"* are applied to the frame portion from where the region was extracted using *T*. The two thresholds are lower than original threshold, i.e., $T' < T'' < T$. As *T'* and *T"* are lower than *T*, it is expected that a larger number of pixels will be assigned to the fish region. Hence, the size of the bounding box is increased so that when thresholds *T'* and *T"* are applied, the region remains inside the bounding box. If thresholding using *T'* and *T"* produces multiple regions within the bounding box, only the largest region is considered.

In this step, three different morphological operations are used. First, dilation and erosion are applied along the directions 0°, 45°, 90° and 145°. Second, dilation and erosion are applied along 0° and 90° directions. Third, no dilation and erosion are applied. However, region filling is applied for all three cases. The combination of three different thresholds for background subtraction and morphological operations, produces a total of nine different shapes. Fig. (13) presents an example of adaptive morphing and thresholding on HC. It can be observed that only Fig. 13(b) produces a HC good shape. Next, all shapes are presented to NNC classifier, and the shape which produces the smallest $dist^2(D_c, D_t)$ is finally selected. In Fig. 13, first, second and third rows respectively correspond to threshold T, T' and T". Similarly, first column corresponds to morphological operations along 0°, 45°, 90° and 145°, the second column corresponds to morphological operations along 0° and 90° and finally, no morphological operations are applied for the third column.

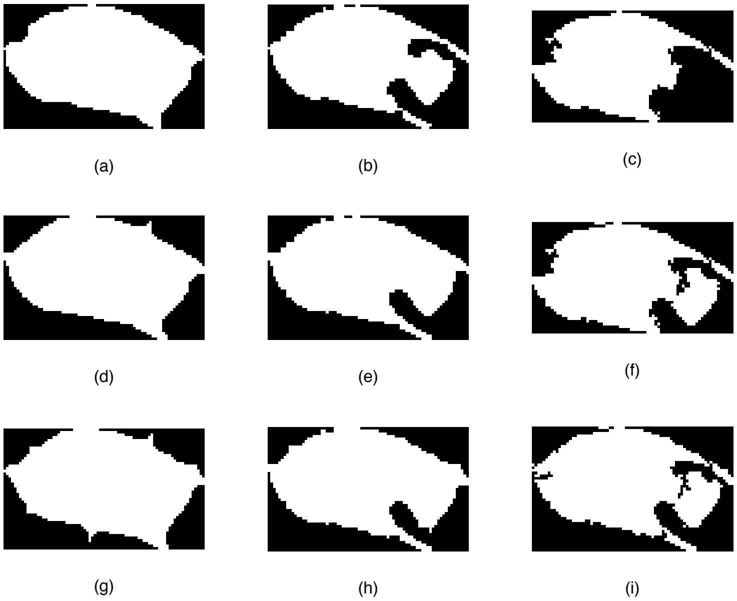


Figure 13: Multiple morphing operations: Each row corresponds to different threshold while each column corresponds to different morphological operation

3.5 Macrofeatures and Macrofeature Classification

This section presents a probabilistic-like approach to classify an fish entire sequence. As the Kalman filter tracks a fish from frame to frame, the microfeatures, FDs and species-specific features, are used to classify a region in a single frame whereas the macrofeatures are used to classify an entire fish sequence.

The macrofeature vector is built by assigning probabilities to each NNC result in a fish sequence. Specifically, the NNC is labels a region according to the relation between the distance, $dist^2(D_c, D_t)$, and the three thresholds Th_1, Th_2 and Th_3. If FD vector, $\hat{D}$, of a region, satisfies threshold Th_1, it is more likely that $\hat{D}$ corresponds to a species than if Th_2 is satisfied. Therefore, a probability quantifying the likelihood of $\hat{D}$ is assigned depending on the threshold that is satisfied; a higher probability is assigned to the smallest threshold. The thresholds Th_1, Th_2 and Th_3 are respectively assigned the values of 0.6, 0.3 and 0.1. However, if no threshold is satisfied, a zero probability is assigned. Similarly, species-specific features are assigned a probability of 0.2 if $m^{EM}_{I_r max} \geq 1.2$ and

$m^{OC}_{I_r max} \geq 0.1$. However, if a stripe is not detected, a zero probability is assigned. The thresholds and probabilities are user-defined and are chosen by observing the sequences in which the exemplars are included.

A final aggregate of all the probabilities is calculated using Eq. (41) for sequence of length n where*j = HP, EM, OC, HC and BV*, indicates the species, $P_j(A_i)$ represents the probability that i-th region, A_i is associated with j-th species and $P_j(\overline{A_i})$ represents the probability that i-th region, A_i is not associated with j-th species.

$$P_j(\bigcup_{i=1}^{n} A_i) = 1 - \prod_{i=1}^{n} P_j(\overline{A_i}) \tag{41}$$

As mentioned in section 3.3, the microfeature set consisting of FDs, $m^{EM}_{I_r max}$ and $m^{OC}_{I_r max}$, are extracted for all regions. However, the aggregate probabilities for species HP, HC and BV is determined from $P_j(A_i) = P^{FD}_j(A_i)$, where $P^{FD}_j(A_i)$ is probability assigned to the NNC classification of FDs. For species EM and OC, $P_j(A_i) = P^{FD}_j(A_i)P^{SS}_j(A_i)$, where $P^{SS}_j(A_i)$ indicates the probability assigned to species-specific features according to $m^{EM}_{I_r max}$ and $m^{OC}_{I_r max}$. It has to be mentioned that although all regions are filtered using GF, while calculating the aggregate probabilities, the algorithm only assigns $P^{SS}_j(A_i)$ to P_{EM} if $m^{EM}_{I_r max} > 1.2$ and to P_{OC} if $m^{OC}_{I_r max} > 0.1$.

In Eq. (41), $P_j(\bigcup\limits_{i=1}^{n} A_i)$ is calculated for all five species separately. The highest $P_j(\bigcup\limits_{i=1}^{n} A_i)$ defines the final classification of the entire sequence. For example, consider the sequence shown in Fig. 11(a). NNC classifies the fish once as 'HP Th 2', 'EM Th 2' and 'EM Th 3', and six times as 'EM Th 1' . The probability of this sequence being EM is $P_{EM}(\bigcup_{i=1}^{n} A_i) = 0.9998$ and H. Plumieri is $P_{HP}(\bigcup_{i=1}^{n} A_i) = 0.3$ where as $P_{BC}(\bigcup_{i=1}^{n} A_i) = 0$, $P_{AF}(\bigcup_{i=1}^{n} A_i) = 0$, $P_{OC}(\bigcup_{i=1}^{n} A_i) = 0$. Hence, although fish in individual frames are classified as different species, using macrofeatures, this sequence is classified as EM.

3.6 Results and Analysis

This section presents performance evaluation results for the proposed classification system. A total of 4163 frames consisting of fish were tested. An average of 3.34 fish regions were found in each frame. Out of 4163 frames, about 1400 frames consist of OC, EM, BV, HC and HP. All fish regions within a frame are automatically identified and segmented out of the overall frame. In all the experiments, the Kalman filter covariance matrices for process noise and measurement noise are assumed to be diagonal and their values are respectively set equal to 0.05 and 0.9.

A. Macrofeature Classification Results

Table 1 presents evaluations results including the sequences corresponding to fish that are at the edges of frame, fish with low resolution and fish with view or pose, which is unsuitable for classification. A few examples of such fish are shown in Fig. 17-20. It can be noted that it would be difficult even for an human expert to classify fish with unsuitable view or with only some part of its body in camera's view. According to Table 1, the classification rate of the proposed system is 72.25%.

In Table 1, the 19% misclassification of EM is either due to fish pose, i.e., the view of fish is not suitable for classification or because the fish is at the edges of the frame. Fig. 17 presents examples in which EM is facing the camera. All the 19% misclassified HC belong to a same fish that faces the camera as shown in Fig. 16(e). An example of BV swimming down is shown in Fig. 19(a). Here BV looks like a OC and is thus misclassified as OC using Th_2. Fig. 19(b) presents BV example with low resolution. Out of the 33% of HP that are misclassified, majority of the fish have very low resolution, thus resulting in poor segmentation of fish from its background and as a result are misclassified by NNC. However, if these problematic cases were eliminated, the performance of the proposed system improves significantly as shown in Table 2.

Table 3 shows results for 1^{st} classification and the combined results for 1^{st} or 2^{nd} classifications,

Table 1: Performance Evaluation

Species	Number of appearences	Classified correct (%)	Misclassified (%)	Not classified (%)
E. Morio	295	72%	19%	9%
O. Chrysurus	120	100%	0%	0%
H. Ciliaris	299	68%	19%	13%
B. Vetula	167	78%	14%	8%
H. Plumieri	286	63%	33%	4%

Table 2: Performance Evaluation Excluding the Problematic Cases

Species	Number of appearances	Classified correct (%)	Misclassified (%)	Not classified (%)
E. Morio	269	79%	21%	0%
O. Chrysurus	120	100%	0%	0%
H. Ciliaris	243	84%	0%	16%
B. Vetula	144	94%	0%	6%
H. Plumieri	286	70%	28%	2%

obtained using macrofeatures. When $P_j(\bigcup_{i=1}^{n} A_i)$ are calculated, a sequence is classified as a particular species *j* if the highest macrofeature, $P_j(\bigcup_{i=1}^{n} A_i)$, corresponds to species *j*. Sometimes, the second highest macrofeature, $P_{j'}(\bigcup_{i=1}^{n} A_i)$, such that $j' \neq j$, also aids in classification. It is possible that *j'* might actually correspond to the correct species.

Table 4 presents the total of number of sequences found in test frames for each species. It has to be mentioned here that object merging and splitting are not addressed in the tracking part of the proposed classification system. For example, when an object splits into two or more fish regions, the Kalman filter does not associate these fish back to their original tracks before they merged.

Table 3: Performance evaluation using first or second classifications

Species	Number of appearances	Classified correct		Not classified	
		1^{st} CR	1^{st} or 2^{nd} CR	1^{st} CR	1^{st} or 2^{nd} CR
E. Morio	269	79%	100%	0%	0%
O. Chrysurus	120	100%	100%	0%	0%
H. Ciliaris	243	84%	84%	16%	16%
B. Vetula	144	94%	94%	6%	6%
H. Plumieri	286	70%	88%	2%	2%

Hence, they are tracked as new sequences. However, the number of sequences in Table 4 are obtained by associating such merge or split sequences manually. In Table 4, second and third columns indicate respectively 1^{st} and 2^{nd} classification results.

All 5 EM sequences are identified correctly as EM in 1^{st} classification whereas 1 sequence is classified as OC and 1 sequence has been classified as BV in 2^{nd} classification. For a sequence with 1^{st} classification EM, $P_{EM}(\bigcup_{i=1}^{n} A_i) > P_j(\bigcup_{i=1}^{n} A_i)$ where j = HP, OC, HC, BV. For example, consider the sequence shown in Fig. 11(d). In this sequence, fish is classified 4 times as HP, $P_{HP}(\bigcup_{i=1}^{7} A_i)$ = 0.9216, once as EM, $P_{EM}(\bigcup_{i=1}^{7} A_i)$ = 0.6, and once as 'OC', $P_{OC}(\bigcup_{i=1}^{7} A_i)$ = 0.3, i.e., $P_{HP}(\bigcup_{i=1}^{7} A_i) > P_{EM}(\bigcup_{i=1}^{7} A_i) > P_{OC}(\bigcup_{i=1}^{7} A_i)$. Therefore, for this sequence 1^{st} classification is HP and 2^{nd} classification is EM.

Fig. 14 presents cases in which the pose of EM is not suitable for NNC classification, i.e., using Fourier descriptors of shape. However, in these cases, the vertical stripe on its tail aids in classification. For the sequence shown in Fig. 14(a), $m_{I,max}^{EM} \geq 1.2$, thus this fish is classified as EM with $P_{EM}(\bigcup_{i=1}^{n} A_i) = 0.9313$. In Fig. 14(b), although the NNC result for one frame is EM using Th_3, the vertical stripe on its tail is detected making $P_{EM}(\bigcup_{i=1}^{n} A_i) = 0.424$ for this sequence. Moreover, the fish in this sequence is not misclassified as any of the other species. Therefore, this sequence

Table 4: Macro Feature Result In Terms of Sequences

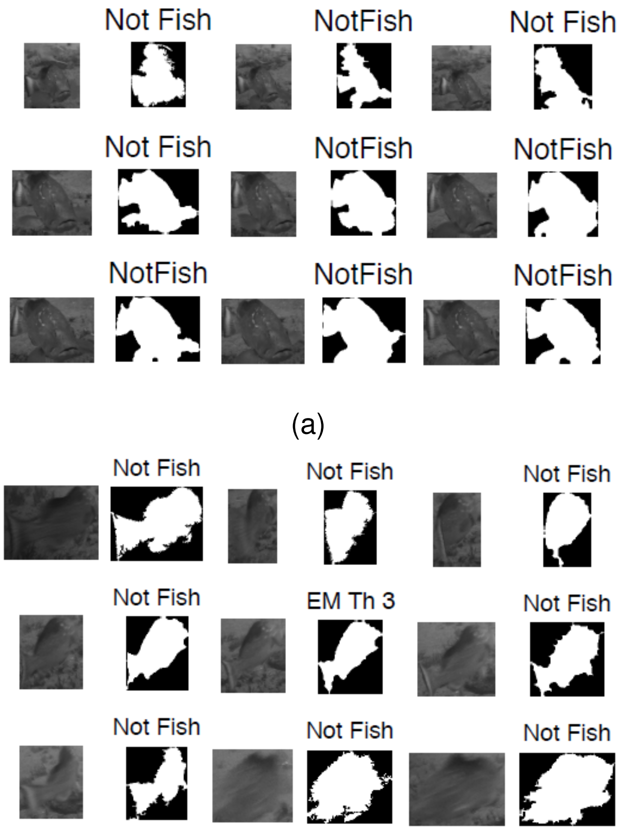
Species	No. of sequences	Sequences classified in first classification	Sequences classified in second classification
E. Morio	5	5 EM	1 OC, 1 BV
O. Chrysurus	14	14 OC	2 EM, 1 HP
H. Ciliaris	14	10 HC, 1 EM	1 BV, 1 OC, 1 as both BV and HP
B. Vetula	5	3 BV, 1 OC, 1 as both HP and BV	1 HC
H. Plumieri	25	15 HP, 4 EM, 5 OC, 1 as both HP and EM	2 HP, 6 EM, 1 as both HP and EM

can be classified as EM.

B. False Alarms

Table 5 presents the false alarms detected. Majority of the false alarms detected occur because the fish shape resembles the shapes of exemplars and due to presence of a stripes as shown in Fig. 15 and 16. A few examples of false alarms, based on the shapes, are presented in Fig. 15. It can be observed the fish in Fig. 15(b) and Fig. 15(c) is a *Grouper*. However, it is not EM. It is misclassified as EM as its shape is similar to that of EM. The *Grouper* in Fig. 15(d) is misclassified as HP because the shape of its tail resembles the shape of HP's tail. On the other hand, the *Calamus Calamus* in Fig. 15(f), 15(g) and 15(h) are respectively misclassified as OC, EM and HP. The *Shark* in Fig. 15(e) is misclassified as OC as its shape resembles to that of an OC. However, in low resolution images as shown in example shown in Fig. 15(j), it is difficult even for a human expert to identify the fish correctly.

Next, a few examples of false alarms caused due to detection of stripe are presented in Fig. 16. In



(a)

(b)

Figure 14: Classification using species-specific features

Table 5: False Alarms

Total number of all fish detected in 4163 frames	Number of false alarms	False alarms (%)
13898	580	4.2%

the examples shown in Fig. 16(a) and 16(b), the camera light creates a bright spot on fish, thus, detecting a vertical stripe $m^{EM}_{I_r max} \geq 1.2$. Although, the NNC classifier does not classify fish in Fig. 16(b) as one of the species of interest, it is labelled as EM due to detection of a vertical stripe. Fig. 16(e) presents an example of HC that is swimming towards the camera. In this case, the NNC does not classify it as a fish, but its fins appear as a vertical stripe making $m^{EM}_{I_r max} \geq 1.2$. Thus this fish is misclassified as an EM with $P_{EM}(A_i) = 0.2$. Similarly, horizontal stripes are detected in fish examples shown in Fig. 16(c) and 16(d). The fish in Fig. 16(c) is labelled as EM by NNC.

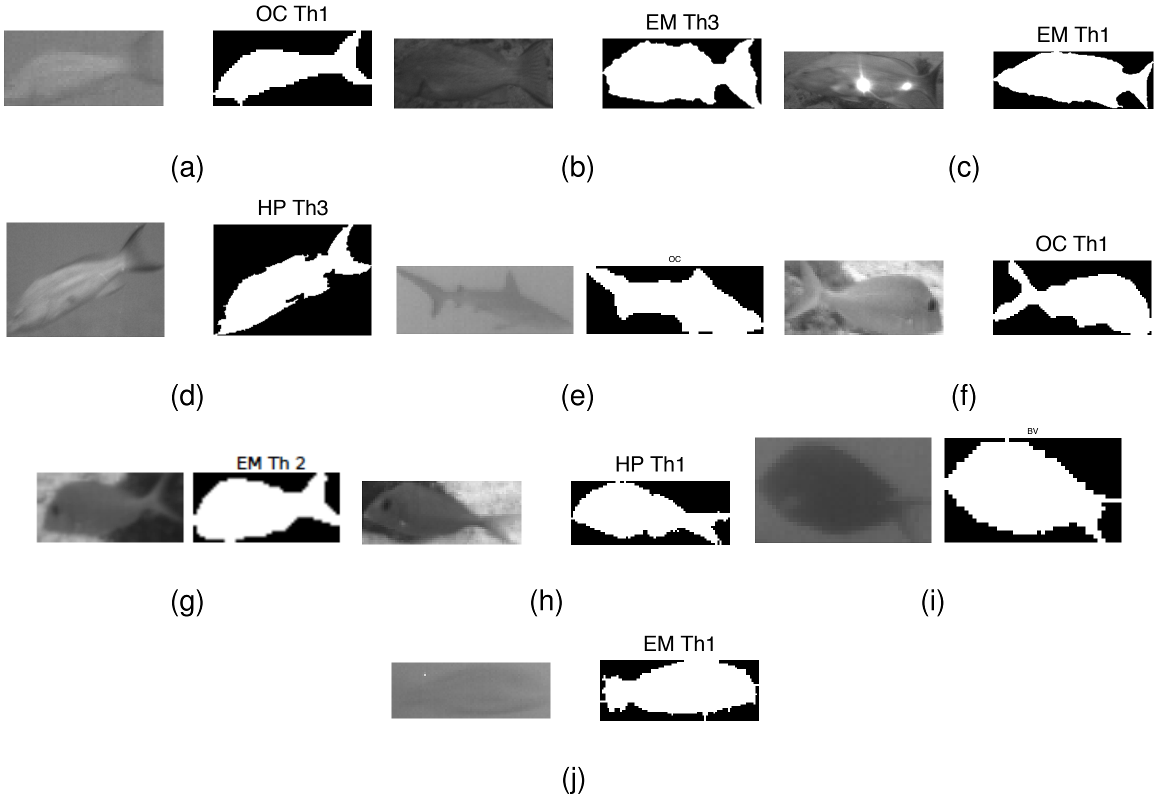


Figure 15: False alarms

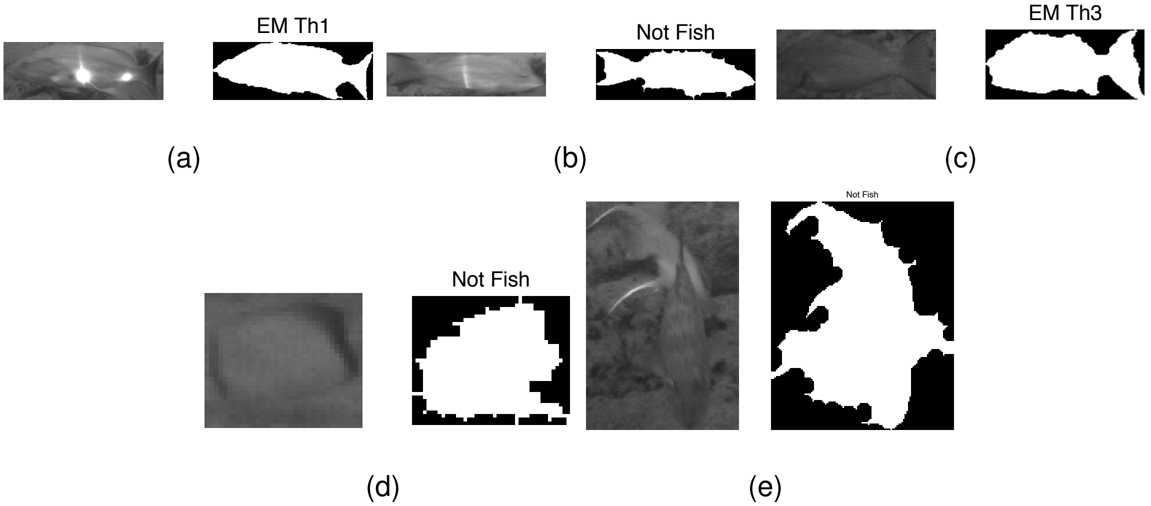


Figure 16: False alarms caused due to detection of stripe

However, a horizontal stripe is detected at the edge of its body. The fish shown in Fig. 16(d) has dark stripes along its anterior and posterior, which lead to detection of vertical stripes $m^{OC}_{I_r max} \geq 0.1$.

C. Other Cases

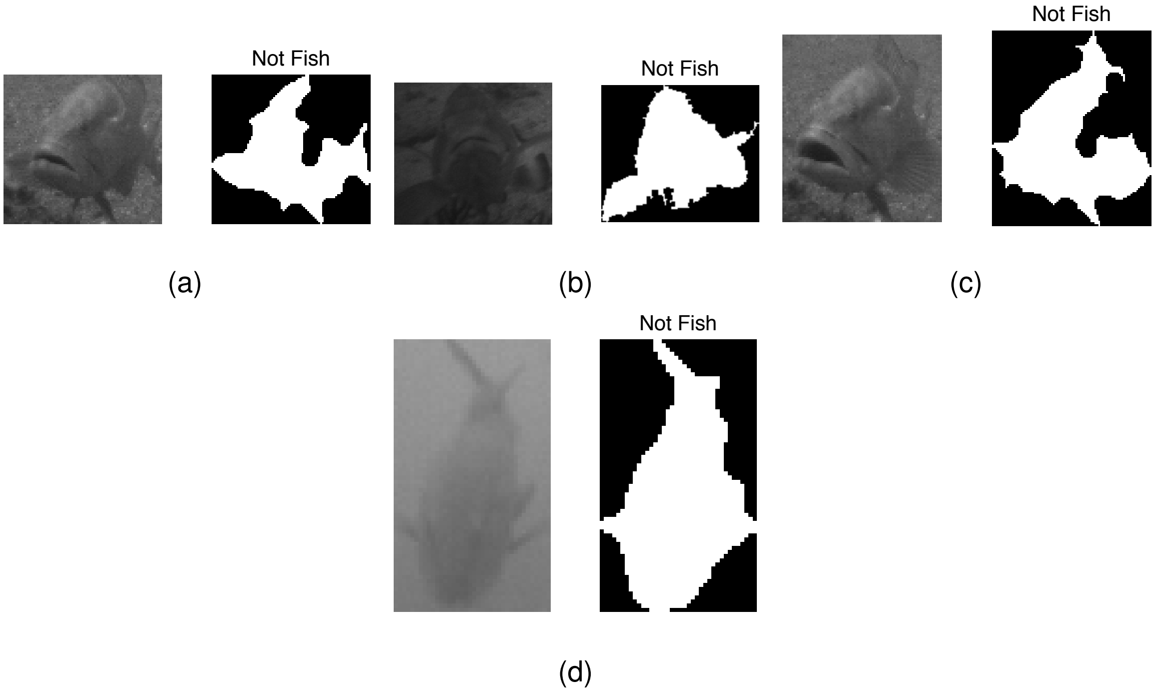


Figure 17: Classification results for EM with poses unsuitable for classification

Consider the examples shown in Fig. 17, in which the fish faces camera. In such cases, the fish cannot be classified until it turns to side, thus providing a good view for classification. The fish shown in Fig. 17(a)-17(c) eventually turns providing a good side view for classification. However, for most of the frames in this sequence, the fish faces the camera and its mouth is detected as a horizontal stripe. Therefore, when macrofeature vectors are computed, the $P_{OC}(\bigcup_{i=1}^{n} A_i) > P_{EM}(\bigcup_{i=1}^{n} A_i)$, thus this sequence is classified as OC.

Although HC has never been misclassified, a high percentage of HC are not classified. An example is shown in Fig. 18. However, 16% of frames have not been classified due to poor resolution which leads to poor segmentation of fish from background. The adaptive segmentation presented in section 3.4 improves segmentation of fish as shown in Fig. 13. Another example of HC being misclassified is shown in in Fig. 16(e). This fish is misclassified as EM because the white stripe on its fins is detected by Gabor filters.

An example of BV swimming down is shown in Fig. 19(a). Here BV looks like a OC and is

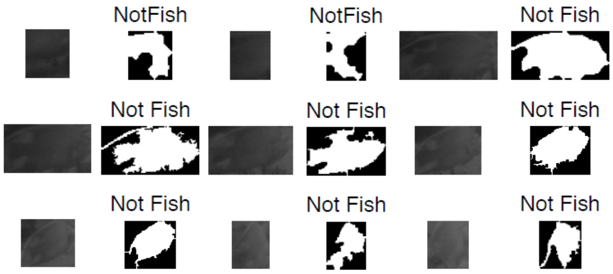


Figure 18: H. Ciliaris low resolution frames

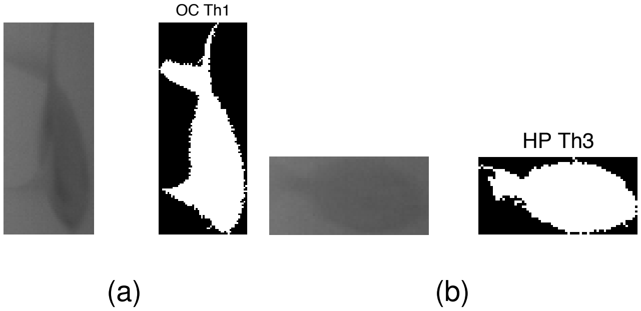


Figure 19: BV Misclassification: (a) B. Vetula swimming vertically down (b) Low resolution frame

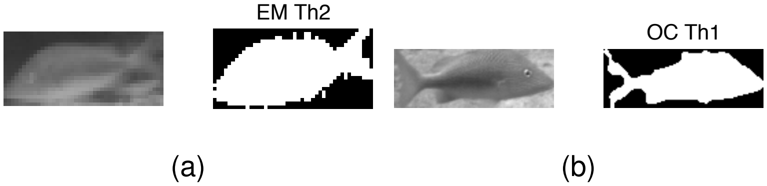


Figure 20: HP Misclassification: (a) HP misclassified as EM (b) HP misclassified as OC

thus misclassified as OC using Th_1. Fig. 19(b) presents BV example with low resolution. In this sequence, BV is classified once as BV using Th_3 and once misclassified as HP using THh_3.

High percentage of HP misclassification can be explained using Fig. 20. The overall shape of HP's body resembles that of EM with an exception of its tail. When the details of HP tail are lost due to improper segmentation or due to low resolution, HP is misclassified as EM. In addition, HP has a blurred stripe along its body. Sometimes, the intensity of this stripe is ≥ 0.1. In such cases, HP is misclassified as OC. It can be noted that HP is misclassified only as one of the two species, namely, EM and OC, because its shape resembles to the shapes of EM and OC.

3.7 Sensitivity Evaluation to the Threshold Variations

This section presents the sensitivity analysis of the proposed classification system to variations in the NNC classification thresholds Th_1, Th_2 and Th_3. It can be observed from the tables presented in this section that any variations in the thresholds Th_1, Th_2 and Th_3 do not result in significant change in the classification performance of the system.

3.7.1 Results for 2nd set of thresholds

The following tables present the classification results for thresholds $Th_1 = 0.54$, $Th_2 = 0.57$ and $Th_3 = 0.6$. The results shown in the tables include all the cases, i.e., including the sequences in which fish is the edges of frames and the sequences with view or pose that is unsuitable for classification. From the tables it can be observed that classification rate for HC has dropped significantly. In addition, the number of false alarms has dropped to 3.1%.

Table 6: Performance Evaluation

Species	Number of appearances	Classified correct (%)	Misclassified (%)	Not classified (%)
E. Morio	295	72%	19%	9%
O. Chrysurus	120	100%	0%	0%
H. Ciliaris	299	55%	27%	18%
B. Vetula	167	78%	14%	8%
H. Plumieri	286	63%	33%	4%

Table 7: Macro Feature Result In Terms of Sequences

Species	No. of sequences	Sequences classified in first classification	Sequences classified in second classification
E. Morio	5	5 EM	1 OC, 1 BV
O. Chrysurus	14	14 OC	2 EM
H. Ciliaris	14	7 HC, 1 EM, 1 BV	1 BV, 1 OC
B. Vetula	5	3 BV, 1 OC, 1 as both BV and HP	-
H. Plumieri	25	15 HP, 4 EM, 5 OC, 1 as both HP and EM	2 HP, 6 EM, 1 as both HP and EM

Table 8: False Alarms

Total number of all fish detected in 4163 frames	Number of false alarms	False alarms (%)
13898	431	3.1%

3.7.2 Results for 3rd set of thresholds

The following tables present the classification results for thresholds $Th_1 = 0.58$, $Th_2 = 0.61$ and $Th_3 = 0.64$. Similar to the results presented for the previous set of thresholds, the results shown in the tables include the sequences in which fish is the edges of frames and the sequences with view or pose that is unsuitable for classification. From the tables it can be observed that the number of false alarms has increased to 5.7%. In addition, classification rate for HC has increased. On the other hand, the classification rate for EM, OC, HP and BV did not change.

Table 9: Performance Evaluation

Species	Number of appearances	Classified correct (%)	Misclassified (%)	Not classified (%)
E. Morio	295	72%	19%	9%
O. Chrysurus	120	100%	0%	0%
H. Ciliaris	299	72%	25%	3%
B. Vetula	167	78%	14%	8%
H. Plumieri	286	62%	34%	4%

Table 10: Macro Feature Result In Terms of Sequences

Species	No. of sequences	Sequences classified in first classification	Sequences classified in second classification
E. Morio	5	5 EM	1 OC, 2 BV
O. Chrysurus	14	14 OC	2 EM, 1 HP, 1 BV
H. Ciliaris	14	10 HC, 1 as both HC and BV, 3 BV	1 EM, 1 OC
B. Vetula	5	3 BV, 1 OC, 1 as both BV and HP	1 HC
H. Plumieri	25	14 HP, 5 EM, 5 OC, 1 as both HP and EM	2 HP, 7 EM, 1 as both HP and EM

Table 11: False Alarms

Total number of all fish detected in 4163 frames	Number of false alarms	False alarms (%)
13898	792	5.7%

3.8 Conclusions

This work proposes an automatic classification system to identify five species of fish in underwater video, which are obtained from uncontrolled environment. The five species of fish used in this work are Red Grouper, Yellowtail Snapper, White Grunt, Queen Angelfish, and Queen Triggerfish. The steps involved to perform recognition of fish species are background processing, object tracking and classification. An existing temporal median filtering-based background subtraction method is used to separate the fish regions from their surroundings. To associate the fish in multiple frames, Kalman filter-based object tracking is applied. Every non-background object is tracked from frame to frame until it exits the camera's field of view. Nearest neighbor classifier is used to label the Fourier descriptors of the object with unknown classification. Finally, macrofeature based classification using a probabilistic-like framework to classify an entire sequence is proposed. One of the advantages of macrofeature classification is it is independent of the length of the sequence. The classification rate of the proposed system is 72.25% for thresholds $Th_1 = 0.56$, $Th_2 = 0.59$ and $Th_3 = 0.62$. The performance of the proposed system has been evaluated for two addition sets of thresholds.

4 Fourier Transform based Pattern Matching

Pattern matching is a tool used extensively in image processing [36], [37] and computer vision [38]. The majority of the existing pattern matching techniques assume that the target and source images are related by a purely affine transformation. However, sometimes the target pattern can be transformed by a small deformation in addition to affine transformation. In such cases, the FT-based techniques presented in section 2.2 fail to determine the affine parameters. The algorithm presented in this chapter aims to complement the existing FT-based affine transform estimation techniques when an additional deformation, which rotates a pattern by different angles with respect to the pattern's center, is present in the pattern. The algorithm uses two approaches to determine the angles of rotation. The first approach is using dynamic programming and the second approach is using a greedy algorithm. The following sections present the concepts of dynamic programming and greedy algorithm, proposed FT-based pattern matching algorithm. Finally, results and conclusions of the proposed algorithm are presented.

4.1 Dynamic Programming

Dynamic programming is a discrete recursive optimization technique which divides a problem into smaller subproblems [39]. Dynamic programming has been used to solve various in image processing [40]-[45], such as, deformable object matching [40], [41], curve detection [42]-[44], image segmentation [44]. Each subproblem is called a *stage*. Solving each subproblem results in a local optimal solution. Dynamic programming arrives at the global optimal solution by sequentially solving each sub-problem, moving from one stage to the next stage until all stages are covered. Each stage is associated with *states*. A transition from one state in a stage to a state in the next stage is associated with a weight, called the *reward*. Dynamic programming is performed either by moving from initial stage to the final stage, called forward induction, or by moving from final stage to the initial stage, called backward induction. The following equations represent recursive formulae for forward induction and backward induction to compute the maximum cost, where s_n is the current stage, $c_n(s_n)$ is the cost associated with the current state and $p_{n-1}(s_n - 1)$ and $p_{n+1}(s_n + 1)$ represent the maximum cost for the stages completed. However, in this work, *reward* is associated with choosing a *state*. Therefore, from here on, *cost* is replaced with term *reward*.

$$p_n(s_n) = \max\{p_{n-1}(s_{n-1}) + c_n(s_n)\} \tag{42}$$

$$p_n(s_n) = \max\{p_{n+1}(s_{n+1}) + c_n(s_n)\} \tag{43}$$

4.2 Greedy Algorithm

A greedy algorithm is an optimization technique that solves a problem by choosing locally optimal solutions. Similar to dynamic programming, a greedy algorithm divides a problem to sub-problems and finds the local optimum solutions for each sub-problem. The global solution is determined by solving each sub-problem moving from the initial stage to the final stage. However, as the algorithm chooses the solution that appears to be the optimum solution locally, it does not necessarily lead to a globally optimum solution [46]. Eq. (44) presents an example of formula to compute the maximum reward, where s_n is the current stage, $c_n(s_{jn})$ is the reward for choosing the j-th state associated with the current stage.

$$p_n(s_n) = p_{n-1}(s_{n-1}) + \max\{c_n(s_{jn})\} \tag{44}$$

Fig. 21 presents an example of a greedy algorithm to find the maximum sum. As mentioned, the greedy algorithm picks the solution that gives the immediate optimal solution. The path chosen by the greedy algorithm is shown with *dark gray* arrows. However, it can observed that the path leading to maximum sum is given by *black* arrows.

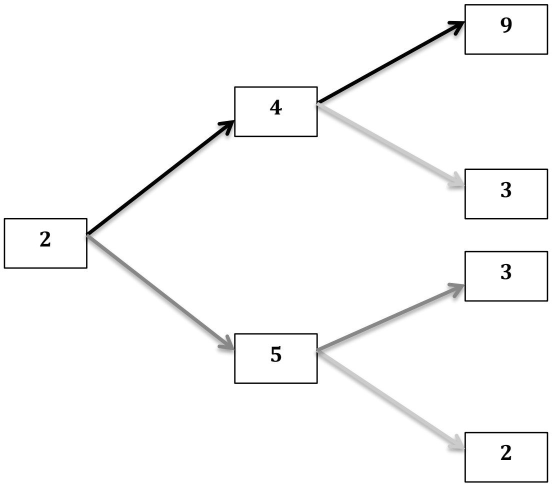


Figure 21: Greedy algorithm

4.3 Proposed Method

This section presents the proposed Fourier transform-based pattern matching algorithm. As mentioned earlier, the Fourier transform-based pattern matching techniques presented in section 2.2 assume that the source and target patterns are related by an affine transformation. However, sometimes a small deformation in addition to affine transform may be present in the target pattern. For example, if Fig. 22 represents an original pattern, previously discussed FT-based algorithms cannot clearly identify the directional features.

The proposed method aims to identify one or two major directional features, called as *path*, in a pattern. In patterns similar to Fig. 22, it can be assumed that the pattern is rotated by a different angle with respect to the distance from center of the pattern. Therefore, a *path* is a sequence of distance-dependent angles of rotation, $\{\phi_n, n = 0, 1, 2, ..., N\}$, where ϕ_n is the angle of rotation at a distance r_n from the center of the pattern. The algorithm starts by decomposing the spatial domain image, $f(\mathbf{x})$ of size $S \times S$, into N non-overlapping circular rings, $f_n(\mathbf{x})$, where $n = 0, 1, 2, ..., N$, as shown in Eq. (45). Radius of each circle is given by Eq. (46), where (m_x, m_y) are the coordinates of pattern center, $r_{max} = S/2$. The centers of the circular rings is at the middle of the original image.

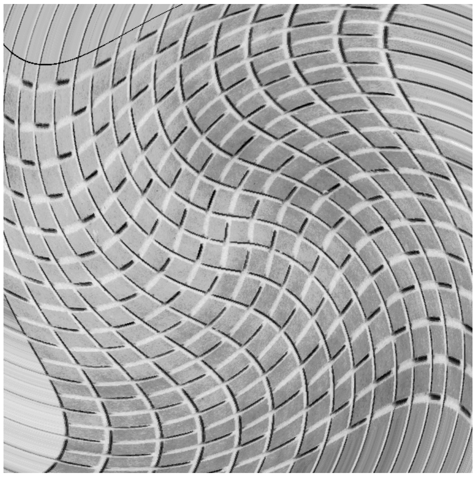


Figure 22: An original pattern

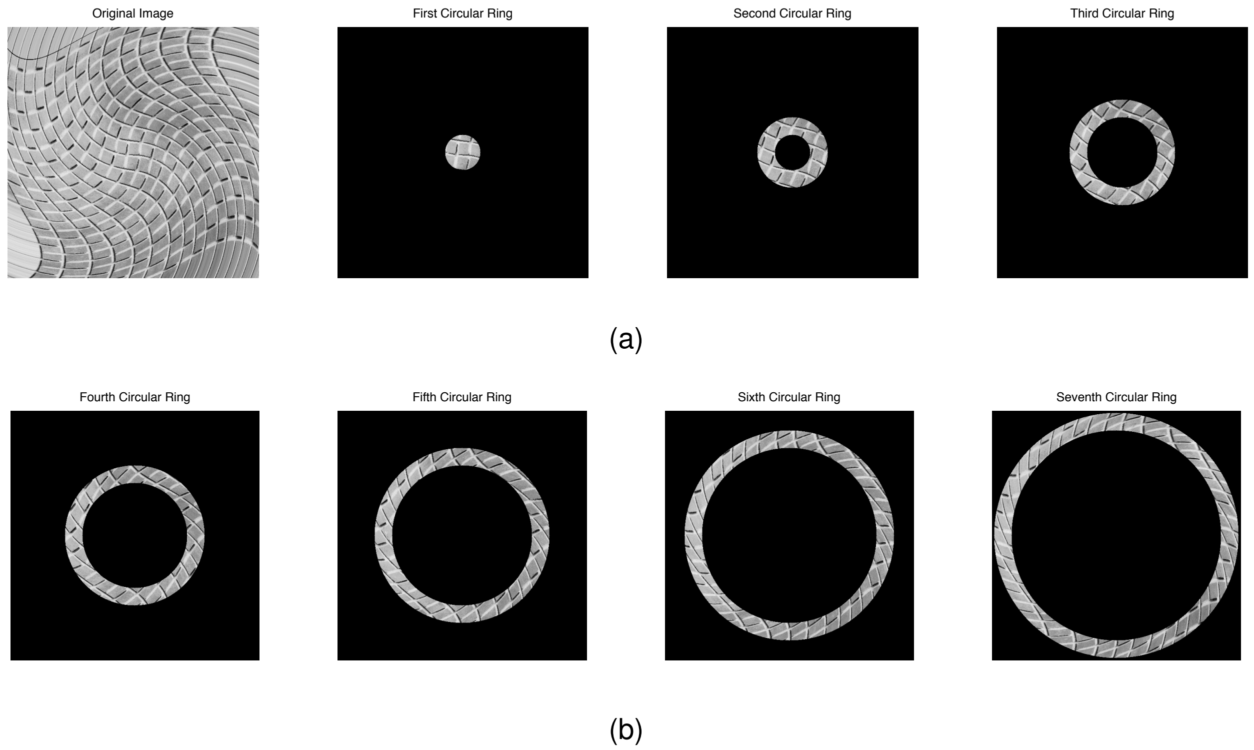
An example of an image decomposed into non overlapping circular rings is presented in Fig. 23.

$$f_n(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & (n-1)r_{max} \leq r_{x,y} < nr_{max} \\ 0 & \text{otherwise} \end{cases} \tag{45}$$

$$r_{x,y} = \sqrt{(x-m_x)^2 + (y-m_y)^2} \tag{46}$$

Next, the FTs of the N rings, $F_n(\mathbf{u}) = \mathscr{F}\{\mathbf{f_n}(\mathbf{x})\}$, are the computed and are mapped to polar domain using the nearest neighbor interpolation. In polar domain, only the angles $[0°, 179°]$ are considered as the directional features are symmetric, i.e., angles $[180°, 359°]$ provide the same information as the angles angles $[0°, 179°]$. After mapping the FTs of the circular rings into polar domain, the next step is to determine the directional features of the pattern. For this two approaches, namely, dynamic programming and greedy algorithm, are used. In this work, each ring is associated with a *stage* while each angle in polar domain is associated with a *state*. In both approaches, backward induction is applied, i.e., starting from the outermost ring, N-th ring, the algorithm moves inwards towards the innermost ring, i.e., the first ring.

The objective function for the dynamic programming-based approach is given by Eq. (48). The



(a)

(b)

Figure 23: Decomposition of original image into circular rings

reward, $c_n(s_n)$ associated with choosing state s_n is given by Eq. (47), i.e., the reward of choosing an angle s_n is the sum of the FT magnitudes at s_n, where $F_n(r,s_n)$ represents the FTs in polar domain; a higher reward indicates a stronger directional component along the angle s_n.

$$c_n(s_n) = \sum_{r=0}^{r_{max}} |F_n(r,s_n)| \tag{47}$$

The total reward, p_n, associated with choosing a state is given by Eq. (48), where p_{n+1} represents the total reward associated with $(n+1)$th stage and $c_n(s_n)$ is the reward associated with the current state s_n given by Eq. (47). As the proposed algorithm aims to find the two paths associated with the highest sum of magnitudes of FTs, the dynamic programming algorithm picks the angle that gives the maximum total reward.

$$p_n(s_n) = \max_{s_{n+1}}\{p_{n+1}(s_{n+1}) + c_n(s_n)\} \tag{48}$$

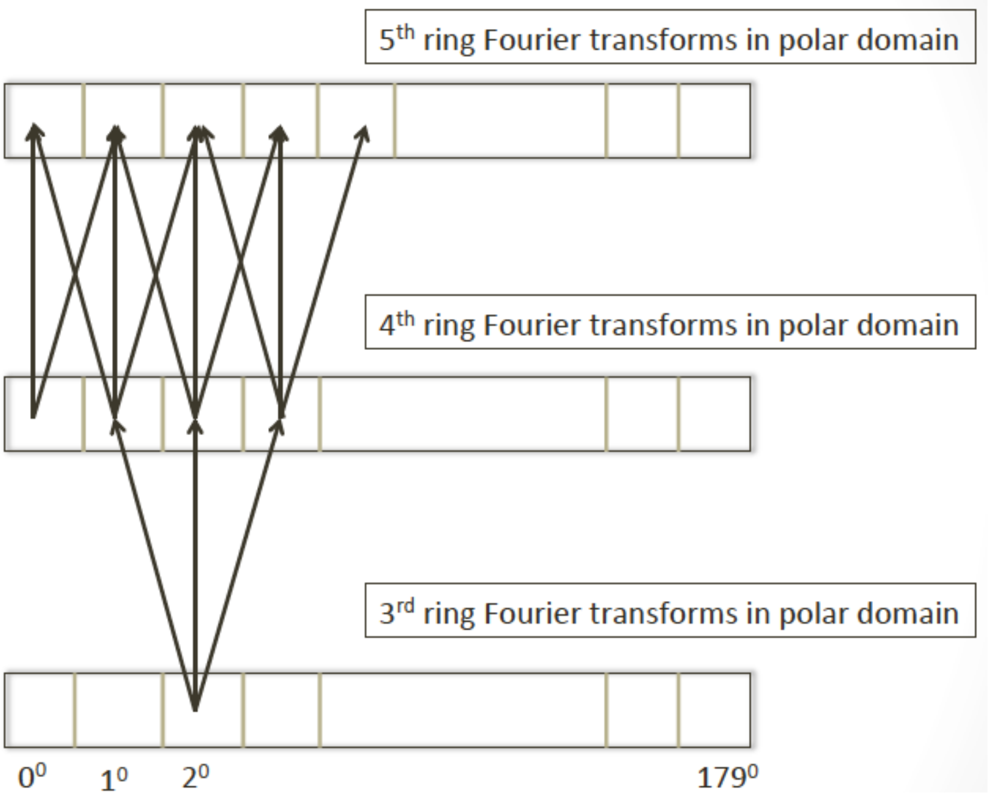

5th ring Fourier transforms in polar domain
4th ring Fourier transforms in polar domain
3rd ring Fourier transforms in polar domain
0⁰
1⁰
2⁰
179⁰

Figure 24: Path computation

In Eq. (48), for each state $s_{n+1} = [s_n - \psi, s_n + \psi]$, i.e., for each angle in the n-th ring, the algorithm search for angles $[s_n - \psi, s_n + \psi]$ in the $(n+1)$-th ring. The angle ψ is a user-defined parameter and defines the maximum search angle as the algorithm moves from $(n+1)$-th ring to n-th ring. Finally, backtracking is applied on s_n to find the two paths corresponding to the two maximum sums of magnitudes of FTs. A constraint, θ is used so as to avoid overlapping of the first and second paths, so that the second path always stays from the first path by angle θ.

The objective function for greedy algorithm-based approach is given in Eq. (49). It can be observed from Eq. (48), the dynamic programming finds the two paths by assuming that the summation of the magnitudes of FTs of individual rings is equal to the result obtained by adding the complex-valued FTs of individual rings. However, the dynamic programming approaches ignores the phase information. As the greedy algorithm computes the summation of complex-valued FTs, the phase information of the FTs is retained. In addition, the greedy algorithm adds the absolute value of

FTs only after reaching the final stage, $n = 1$.

$$s_{n+1} = \underset{s'_{n+1}}{\operatorname{argmax}}\{\sum_r |P_{n+1}(r, s'_{n+1}) + F_n(r, s_n)|\} \tag{49}$$

$$P_n(r, s_n) = P_{n+1}(r, s_{n+1}) + F_n(r, s_n) \tag{50}$$

subject to $s_{n+1} = [s_n - \psi, s_n + \psi]$, where ψ is a user-defined search angle.

To associate Eq. (48) with Eq. (44), it can be mentioned that the reward $c_n(s_{jn})$ is equivalent to $\sum_r |P_{n+1}(r, s'_{n+1}) + F_n(r, s_n)|$. This algorithms falls in the greedy algorithm category because maximizing $\sum_r |P_{n+1}(r, s'_{n+1}) + F_n(r, s_n)|$ in one step does not imply that the final sum, considering all stages, is globally optimum. For example, consider three complex numbers, $a = 1 + 2j$, $b = 3 - 2j$ and $c = -4 - 0j$. Then, $a + b = 4$ and $a + b + c = 0$. Addition of a and b results in a positive number. However, the sum of all three numbers results in 0. It can observed from the example that addition of complex numbers does not necessarily lead to maximum sum. Similarly, the proposed greedy algorithm-based approach adds the complex-valued FTs. Therefore, this approach may not always lead to global optimum solution.

4.4 Results

This section presents simulation results for the proposed FT-based pattern matching algorithm. The three user-defined parameters, expected maximum change in angle as the algorithm moves from $(n+1)$-th ring to n-th ring, $\psi = 40°$, the minimum angle to avoid overlapping of the first and second paths in backtracking algorithm , $\theta = 70°$ and the number of circular rings into which the original pattern is decomposed is N = 7. Fig. 25(a) and Fig. 25(c) are the original patterns whereas Fig. 25(b) and Fig. 25(d) are the images obtained after applying deformation on the original patterns. The white and black lines on the patterns represent the two paths detected. The direction of the line segment between two consecutive dots on the paths provides the direction of

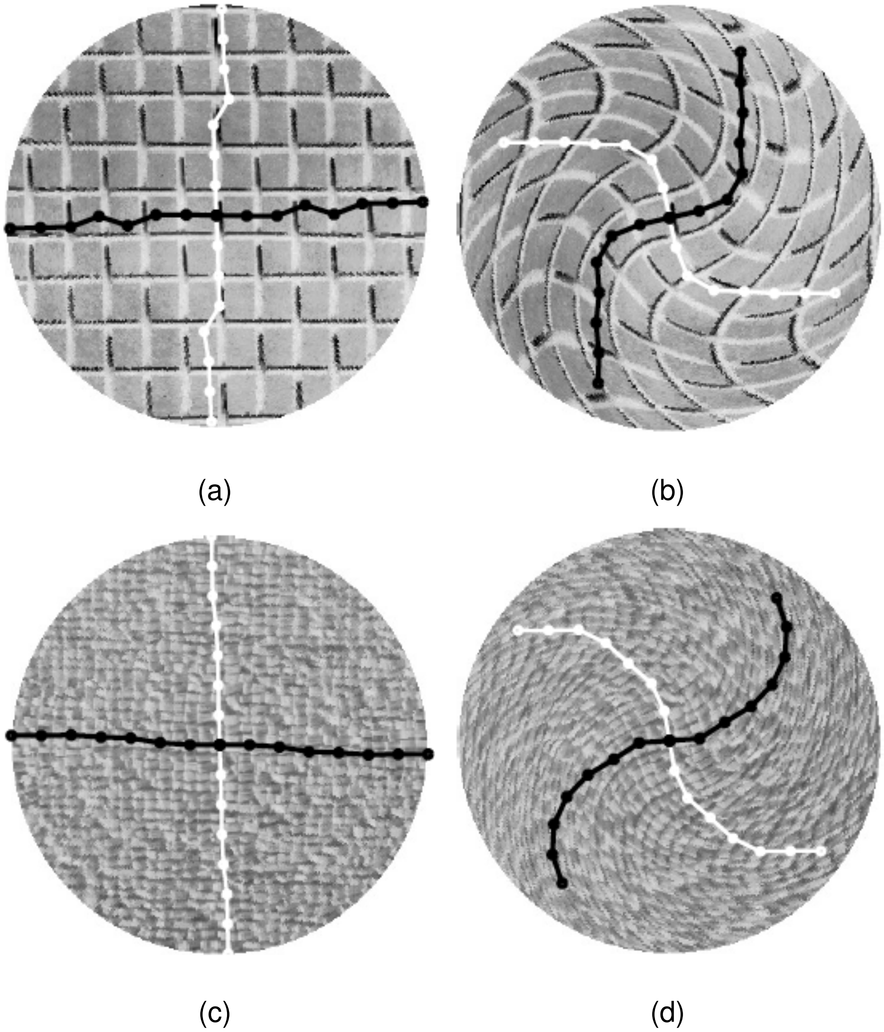


Figure 25: Results of the proposed algorithm using greedy algorithm

rotation, ϕ_n, of the rings.

An example of the proposed algorithm using dynamic programming is shown in Fig. 26(c). It can be observed that the dynamic programming fails to detect the directional features whereas in Fig. 26(b), the greedy algorithm correctly detects the directions in same pattern.

Another example is presented in Fig. 27, where single path is detected using greedy algorithm. From the FT of the example shown in Fig. 27(a), it can be observed that the major direction of this pattern cannot be determined without decomposing it into rings.

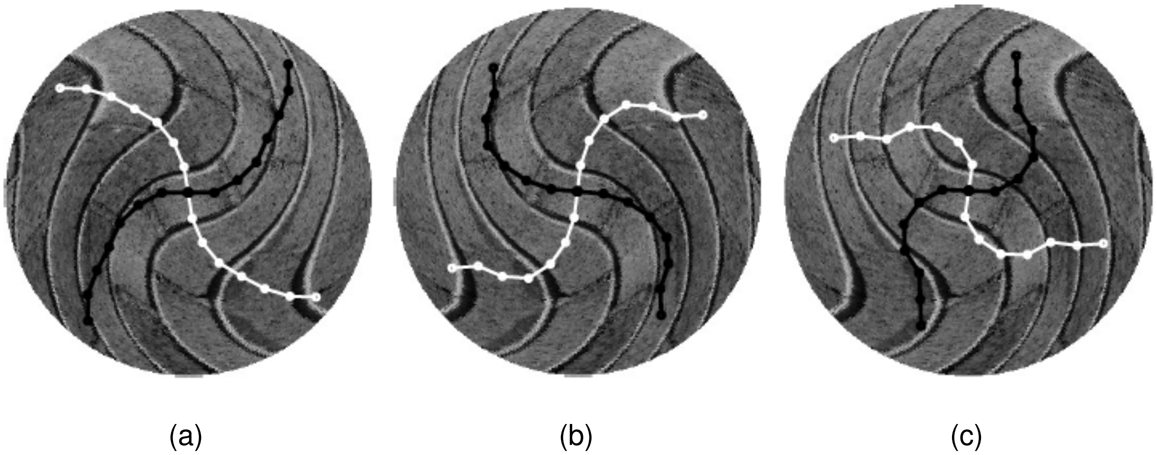

(a) (b) (c)

Figure 26: Results of the proposed algorithm using: (a), (b) Greedy algorithm (c) Dynamic programming

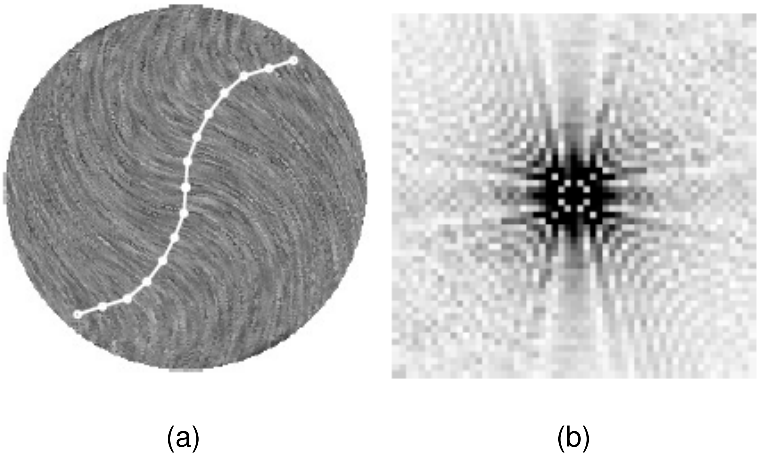

(a) (b)

Figure 27: Results of the proposed algorithm (a) Pattern and path detected using greedy algorithm (b) Zoomed in version of Fourier transform magnitude of the pattern

4.5 Conclusions and Future Work

A Fourier transform-based technique has been proposed to detect directional features in patterns which have deformed by a warp-like transformation. The algorithm complements the existing FT-based techniques to estimate the affine parameters. Two different approaches are studied in this work. The first approach uses dynamic programming, while the second uses a greedy algorithm. The dynamic programming-based approach computes the paths by assuming that the sum of magnitudes of FTs of the individual concentric circular rings is equal to the FT of the original images, thus ignoring the phase information in FTs. On the other hand, the greedy algorithm-based approach uses the complex-valued FTs to compute the paths. The purpose of this work is not to compare dynamic programming itself with the greedy algorithm. Such a comparison is not appropriate since the objective functions used in the two approaches are different. However, one can compare the performances of the two techniques overall. Further experimentation is required to confirm that the dynamic programming does not detect the directional features for the objective function used in this work.

Future work includes extending the proposed algorithm to other types of deformations and automatically determine the user defined parameters. These include the number of circular rings into which the original image is to be decomposed, N, the search angle as the algorithm moves from one ring to the next ring, ψ, and finally, the separation angle used in backtracking algorithm, θ.

References

[1] http://www.mathworks.com/discovery/pattern-recognition.html

[2] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT11/node5.html#SECTION00052000000000000000

[3] White, D. J., Svellingen, C., Strachan, N. J. C., *Automated measurement of species and length of fish by computer vision*, Elsevier, 2006.

[4] F. Storbeck and B. Daan, *Fish species recognition using computer vision and a neural network*, Elsevier, 2000.

[5] Nery, M. S., Machado, A. M., Campos, M. F. M., Padua, F. L. C., Carceroni, R., Queiroz-Neto, J. P, *Determining the appropriate feature set for fish classification tasks*, Proceeding of the XVIII Brazilian Symposium on Computer Graphics and Image Processing, 2005.

[6] M. C.Chuang, J. N.Hwang, C. Rose, *Aggregated Segmentation of Fish from Conveyor Belt Videos*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1807-1811, 2013.

[7] N. Castignolles, M. Catteon, M. Larinier, *Identification and counting of live fish by image analysis*, SPIE. Vol 2182, Image and Video Processing II, 1994.

[8] Rova, A., Mori, G., Dill, L. M., *One Fish, Two Fish, Butterfish, Trumpeter: Recognizing Fish in Underwater Video*, MVA2007 IAPR Conference on Machine Vision Applications, 2007.

[9] G. T.Shrivakshan, *An Analysis of SOBEL and GABOR Image Filters for Identifying Fish*, Proceedings of the International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), 115-119, 2013.

[10] Benson, B., Cho, J., Goshorn, D., Kastne, R.*Field Programmable Gate Array Based Fish Detection using Haar Classifier*, American Society of Underwater Science, 2009.

[11] Edgington, D., Cline, D., Davis, D., Kerkez, I., Mariette, J. *Detecting, Tracking and Classifying Animals in Underwater Video*, OCEANS 2006. 1-5, 2006.

[12] Ranzato, M.A. *Automatic Visual Recognition of Biological Particles*, California Institue of Technology: Pasadena, CA, 2004.

[13] Fabic, J. N., Turla, I. E., Capacillo, J. A., David, L. T., Naval Jr, P. C, *Fish Populating Estimation and Species Classification from Underwater Video Sequences using Blob Counting and Shape Analysis*, Proceeings of Underwater Technology Symposium (UT), 1-6, 2013.

[14] Lee, D. J., Redd, S., Schoenberger, R., Xu, X., Zhan, P., *An Automated Fish Species Classification and Migration Monitoring System*, 29th Annual Conference of the IEEE Industrial Electronics Society. Vol. 2, 1080-1085, 2003.

[15] Rodrigues, T. A. M., Padua, L. C. Flavio., Gomes, M. R., Soares E. G. *Automatic Fish Species Classification Based on Robust Feature Extraction Techniques and Artificial Immune Systems*, 2010.

[16] Spampinato, C., Giordano, D., Salvo, R. D., Chen-Burger, Y., Fisher, R. B., Nadarajan, G. *Automatic Fish Classification for Underwater Species Behaviour Understanding*, ARTEMIS'10 Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams, 2011.

[17] Tsai, D., Lai, S. *Independent Component Analysis-Based Background Subtraction for Indoor Surveillance*, IEEE Transactions On Image Processing, vol. 18, no. 1, 2009.

[18] Maddalena, L.; Petrosino, A. *A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications*, IEEE Transactions On Image Processing, vol. 17, no. 7, 2008.

[19] Rosales, R., Sclaroff, S. *Learning and synthesizing human body motion and posture*, Proceeding of Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 506-511, 2000.

[20] Gundam, M., Charalampidis, D., Ioup, G., Ioup, J., Thompson, C. *Automated Fish Classification in Underwater Video*, Proceedings of 66-th Annual Conference Gulf and Carribean Fisheries Institute, Corpus Cristi, vol 66, pp 276-282, 2013.

[21] Joginipelly, A.K., Charalampidis, D., Ioup, G., Ioup, J., Thompson, C. H. *Species-specific Fish Feature Extraction Using Gabor Filters*, Proceedings of 66-th Annual Conference Gulf and Carribean Fisheries Institute, Corpus Cristi, volume 66, pp 283-291, 2013.

[22] Fogel, I., Sagi, D. *Gabor Filters as Texture Discriminator*, Journal of Biological Cybernetics, vol. 61, Issue 2, pp 103-113, 1989.

[23] R. N. Bracewell, K. Y. Chang, A. K. Jha, Y. H. Wang, *Affine Theorem for two-dimensional FT*, Electronics Letters, 29(3), p. 304, 1993.

[24] A. Bhalerao and R. Wilson, *Affine invariant image segmentation*, Proc. British Machine Vision Confonference, Kingston University, UK, pp. 157-166, 2004.

[25] A. Bhalerao and R. Wilson, *Warplet: An image dependent wavelet representation*, Proc. IEEE International Conference on Image Processing, pp 490-493, 2005.

[26] Q. Chen, M. Defrise, and F. Deconinck, *Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition*, IEEE Transactions Pattern Analaysis and Machine Intelligence, vol. 16, no. 12, pp. 1156-1168, 1994.

[27] H. Park, G. R. Martin, A. Bhalerao, *Local Affine Image Matching and Synthesis Based on Structural Matching*, IEEE Transactions On Image Processing, vol. 19, no. 8, 2010.

[28] A. Kak, M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press,1998.

[29] Li, X., Wang, K., Wang, W., Li, Yang. *A Multiple Object Tracking Method Using Kalman Filter*, International Conference on Informations and Automation, 2010.

[30] Li, J., Shao, C., Xu, Wangu., Yue, Hao. *Real Time Tracking of Moving Pedestrians*, International Conference on Measuring Technology and Mechatronics Automation, 2009.

[31] Zahn, C., Roskies, R. *Fourier Descriptors for Plane Closed Curves*, IEEE Transactions on Computers, C-21: 269-281, 1972.

[32] Lin, C. C., Chellappa, R. *Classification of Partial 2-D Shapes Using Fourier Descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 9, No. 5, 1987.

[33] Folkers, A., Samet, H. *Content-based Image Retrieval Using Fourier Descriptors on a Logo Database*, Proceedings of the 16^{th} International Conference on Pattern Recognition, Vol III, 521-524, 2002.

[34] Sarfraz, M., Hassan, M., Iqbal, M. *Object Recognition Using Fourier Descriptors and Genetic Algorithm*, Proceedings of International Conference of Soft Computing and Pattern Recognition, 318-323, 2009.

[35] Cover, T. M., Hart, P. E. *Nearest Neighbor Pattern Classification, IEEE Transactions On Information Theory*, Vol. 13, No. 1, pp 21-27, 1967.

[36] M.Uenohara, and T. Kanade, *Use of Fourier and Karhunen-Loeve Decomposition for Fast Pattern Matching With a Large Set of Templates*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, no. 8, pp. 891-898,1997.

[37] D. Nair, R. Rajagopal, and L. Wenzel, *Pattern Matching based on a generalized FT*, Proceedings of the SPIE, vol. 4116, pp. 472-480, 2000.

[38] X. Pengdong, N. Barnes, T. Caetano, and P. Lieby, *An MRF and Gaussian Curvature Based Shape Representation for Shape Matching*, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-7, 2007.

[39] R. E. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[40] Y. Amit, and A. Kong, *Graphical Templates for Model Registration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pp. 225-236, 1996.

[41] P. F. Felzenszwalb, *Representation and Detection of Deformable Shapes*, EEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 208-220, 2005

[42] A. Amini, T. Weymouth, and R. Jain, *Using Dynamic Programming for Solving Variational Problems in Vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 9, pp. 855-867, 1990.

[43] U. Montanari, *On the Optimal Detection of Curves in Noisy Pictures*, Communications of the ACM, Vol. 14, Issue 5, pp. 335-354, 1971.

[44] D. Geiger, A. Gupta, L. A. Costa, J. Vlontzos, *Dynamic Programming for Detection, Tracking, and Matching Deformable Contors*, IEEE Transacttions on Pattern Analysis and Machine Intelligence, vol. 17, No. 3, pp. 294-302, 1995.

[45] P. F. Felzenszwalb, and D. P. Huttenlocher, *Pictorial Structures for Object Recognition*, International Journal of Computer Vision, vol. 61, no.1, pp. 55-59, 2005.

[46] `https://www.cs.berkeley.edu/~vazirani/algorithms/chap5.pdf`

5 Vita

Madhuri Gundam was born and bought up in Hyderabad, India. She received her Bachelor's in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in May 2007. In Fall 2007, she joined the Department of Electrical Engineering at University of New Orleans as a Graduate student and received her Master's in 2010. She has worked as a teaching assistant and research assistant for the Department of Electrical Engineering during her graduate career at the University of New Orleans. She is currently working as an Electrical Engineering intern at the American Bureau of Shipping.