# Fish Population Estimation and Species Classification from Underwater Video Sequences using Blob Counting and Shape Analysis

J.N. Fabic[†], I.E. Turla[†], J.A. Capacillo[†], L.T. David[‡], and P.C. Naval, Jr.[†]

[†]Department of Computer Science
[‡]Marine Science Institute
University of the Philippines, Diliman, Quezon City, Philippines

***Abstract*—Fish population estimation and classification of fish species have been an integral part of marine science research. These tasks are important for the assessment of fish abundance, distribution and diversity in marine environments.**

**We describe an efficient method for fish detection, counting, and species classification from underwater video sequences (UWVS) using blob counting and shape analysis. The video sequences were obtained with a moving camera resulting in rapid viewpoint changes thereby making it difficult to employ motion detection schemes in extracting fish images from background.**

**Video preprocessing involved blackening out the corals from the underwater videos. This is done in order to effectively estimate fish count in the environment, though excluding those that are against a coral background. We then applied histogram comparison to initially blacken out the occlusions using blue and non-blue templates obtained randomly from the UWVS. We then introduced an erasure procedure to further aid in removing the coral background For fish detection, Canny edge detection was applied to extract fish contours. After the latter have been delineated, blob counting is then employed to in order to compute the fish count. Due to rapid frame changes, the average fish count per unit time is computed from the counts in each frame. For shape analysis, blob size is initially estimated and when a threshold is exceeded, Zernike moment-based shape analysis is performed on the blob for comparison with moment signatures of selected fish species stored in a database. The label of the best matching moments identifies the species of the fish blob. The shape-based classification algorithm is designed to identify the two most common species of fish found in the Tubbathaha reef in Sulu Sea, Philippines.**

***Keywords—Fish Census from Video, Shape Analysis, Zernike Moments***

## I. Introduction

The estimation of fish count and determination of species found in coral reefs are important tasks in the field of marine science research. The traditional method makes use of experienced divers to manually do the counting and identify the fish species. However, this method suffers many obvious limitations, among them are the lack of trained divers whose services can only be contracted after an indeterminate waiting period, likelihood of human error resulting from diver fatigue, emotional conditions, biological limitations and environmental factors.

Numerous proposals have been made for more automated methods for assessment of fish abundance, distribution and diversity ranging from towed equipment containing sonar, resistivity counters and infrared beams [1] to remotely operated vehicles [2] fitted with underwater cameras that collect high-resolution video [3]. Among the numerous sensing modalities, live video imaging has the key advantages of human verifiability down to the individual organism and non-intrusiveness of the collection process [4]. In recent years, embedded video cameras have been deployed for unattended video capture by underwater observatories in midwater and benthic habitats. The subsequent manual analysis of the collected video data is tedious, very time consuming, requires human concentration, and therefore prone to errors. Estimates for time spent for annotation and classification specify around 15 minutes of manual analysis for every minute of video [5]. It is therefore evident that machine-assisted analysis is necessary for conversion of raw video data into meaningful information for use by marine scientists.

The assessment task reduces to the those of accurate detection and recognition of fish species from which aggregate count and other statistics could be made. These subtasks are fundamentally challenging due to fish movement, uneven and frequent changes in illumination, camera motion, etc. [6]. Fortunately, advances in image analysis have reached the point where these challenges can be overcome with sophisticated algorithms that achieve good performance through incorporation of domain-specific information. A number of video-based fish detection and identification systems with reasonable accuracies are now available and deployed by various marine science laboratories [1], [5], [6]. These systems perform color, shape and texture analyses on the collected video data in an automated fashion to yield bulk fish statistics.

In this paper we describe an approach for an efficient means of fish count estimation and species identification what will serve as a tool for marine scientists that will alleviate the burden of manual analysis.

## II. System Overview

Our system processes underwater video streams (UWVS) obtained from a moving video camera. Processing of video data obtained from this set up is difficult due to rapid viewpoint changes arising from camera movement and the presence of

978-1-4673-5948-1/13/$31.00 ©2013 IEEE

corals and other organisms in the benthic environment that complicates object segmentation. Several steps need to be completed in order to successfully count and identify fish with this challenging data.

The system subjects the collected video data to the following operations: Pre-processing, Contour Detection, Blob Counting and Species Identification (Fig. 1). Pre-processing aims to remove unwanted background objects by blackening them out. It is composed of the following three steps: Coral Blackening, Inward-Outer Block Erasure, and Edge Cleaning. Contour Detection utilizes the Canny Edge Algorithm to capture the outline of fish which is subsequently converted into blobs using Connected Components Analysis. The blobs are counted and the Zernike moment signature for each blob is computed and compared with known moment signatures for species identification.

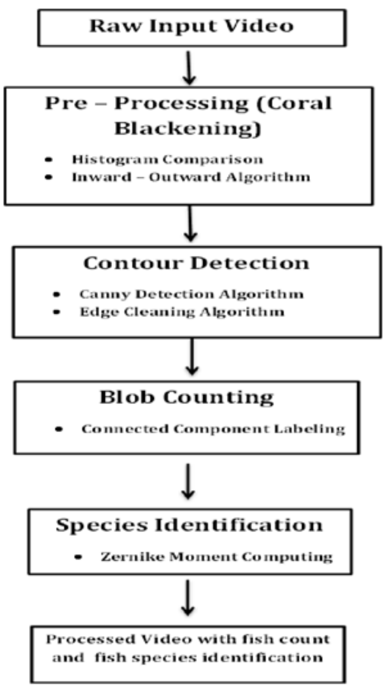


Fig. 1. System Flowchart

## III. PRE-PROCESSING

The underwater video sequences (UVWS) undergo pre-processing in order to eliminate unwanted complex background. This involves Coral Blackening to blacken out corals, and the use of Inward-Outer Block Erasure and Edge Cleaning Algorithms to refine the blackening process.

### *A. Coral Blackening Procedure*

In order to determine which parts are blackened and which are not, the entire image is divided into $20 \times 20$ pixel blocks and color histograms are computed for each block. Since the background against which the fish are clearly visible is predominantly blue, we only compute the histogram of the blue component of the image.

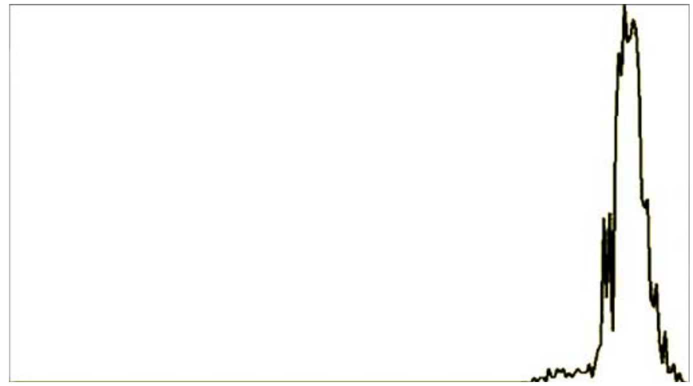

Fig. 2. **Histogram of a Blue Image Block**

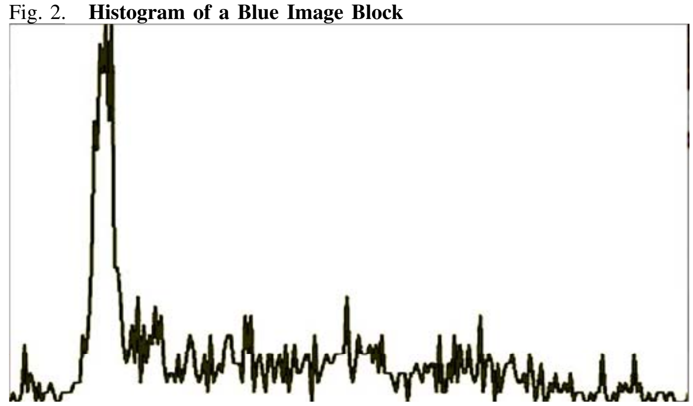

Fig. 3. **Histogram of a Non-Blue Image Block**

We manually selected from across 30 frames as templates from the Underwater Video Sequences, 250 samples of $20 \times 20$ pixel blocks, half of which are from the bluish water background and the remaining half are from benthic (mainly coral) background. The Blue and Non-Blue histogram templates were averaged and saved for later comparison. Each frame, having a dimension of $720 \times 480$ pixels are then divided into blocks, resulting in a total of $36 \times 24$ or 864 blocks. The histograms for each of these blocks are computed and compared with the Blue and Non-Blue histogram templates. Figs. 2 and 3 show the Blue and Non-Blue histogram templates. If the value of a block's histogram is closer to that of a Non-Blue template, that block will be blackened.

Although this step is effective in blackening out most of the benthic portions of the image, minimal errors are still present. These errors are mostly from small coral parts which remain unchanged due to the square shape of the block and fish which are also blackened due to their non-blue color. Thus, we developed two procedures to address the problem.

### *B. Inward-Outer Block Erasure Algorithm*

The Inward-Outer Block Erasure Algorithm works by block-level scanning the image frame left to right and downwards starting from the topmost left block. Fig. 4 depicts how the algorithm works. Starting from the block on the second row, targeted blocks are histogrammed compared to their outer neighbors. If their outer neighbors have mean histogram values that are black, they are also blackened, otherwise the block could possibly be surrounded by blocks corresponding to water which makes it more likely that it is also block for water but containing fish that caused it to be blackened. This block is then converted back to its original state.


Fig. 4. **Inward-Outer Block Erasure Example**

### *C. Edge Cleaning Algorithm*

The Edge Cleaning Algorithm performs a pixel-wise scan starting from the left most bottom pixel of a frame and then moves upward checking the RGB value of the current pixel. If the pixel's value is near black, it moves up and repeats the process; if the current pixel's value is near blue, it will check the pixel above. If the upper pixel is black, the current pixel shall be blackened because this pixel is probably a part of a coral which the Inward-Outer Erasure Algorithm has missed since the latter only processes blocks and not pixels. After doing this, the algorithm continues its way up. The scanning then proceeds one pixel to the right and repeats the process again from the bottom.

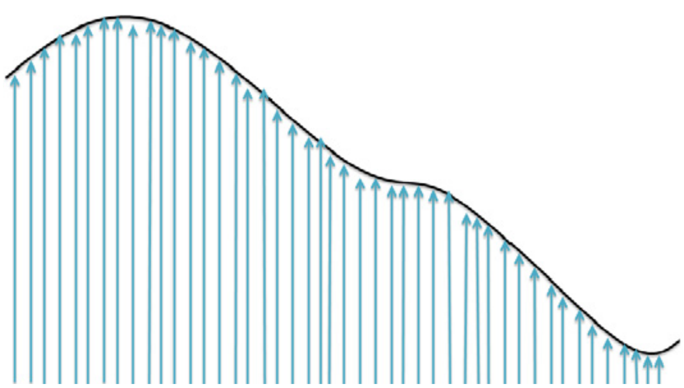

Fig. 5. **How the Edge cleaning works**

## IV. Contour Detection

After all the corals are blackened out, the next step is to find where the fishes are. The Canny Edge Detector [7] is used in each frame of the UWVS for detecting fish contours and filling up the spaces to allow blob counting. The Canny Edge Detector first convolves the raw image with a Gaussian filter to reduce noise and then computes the gradient and gradient direction for each possible edge. Pixels with large local edges are determined from the image gradients and double thresholding is applied to remove edge-like noise and retain possible edges in the image. The sensitivity of the Canny Edge Detection Algorithm may be controlled by specifying the threshold value. Extraction of the suspected fish edges to a certain extent, depends on this threshold value since higher or lower values produce inaccurate results. High values will fail to detect lightly outlined fish while low values tend to highlight the other objects in the benthic environment. We have determined this threshold through several trials. Fig. 7 is the resulting image after Canny edge detector is applied on the raw image (Fig.6).

Fig. 6. **Raw Underwater video**

Fig. 7. **Underwater video with Canny detection applied**

## V. Blob Detection

After the contours are detected the next step is to fill up the contours to form blobs. The Blob detector used is based on the Laplacian of the Gaussian (LoG). An image is first convolved by a Gaussian kernel. A multi-scale blob detector with automatic scale selection is then obtained using a Scale-normalized Laplacian operator. The results is then used to detect scale-space maxima or minima. Thus given a discrete 2D input image, a 3D discrete scale-space volume is computed. A bright (dark) Blob is then identified if the corresponding point is greater (smaller) than the values of its 26 neighbours [8].

The algorithm used in labeling blobs is Connected Component Labeling which is used to detect connected regions in digital binary images. Given a heuristic, subsets of connected components can be uniquely labeled enabling blobs to be extracted and/or detected from the resulting binary image. The results can now be used for counting, filtering or tracking [8].

## VI. SPECIES IDENTIFICATION

Image moments are a powerful shape descriptor that capture the global features of objects. These are widely used in the field of image pattern recognition due to their invariance properties.

Identification of fish species rely on the image moment features of the fish blob. Since small fish blobs may not contain enough information for identification of fish species, we observe a size threshold that need to be exceeded for image moment calculation to proceed.

### A. Zernike Moments

The set of orthogonal Zernike moments are known to be superior compared to other image moments due to their nice rotational, translational and scale invariant properties [9]. We chose Zernike moments for our system because of these important properties match the requirements for fish species identification.

These are the three steps in computing the Zernike moments of a given input image: computation of radial polynomials, computation of Zernike basis functions, and computation of Zernike moments by projecting the image on to the basis functions [10].

The complex-valued Zernike moment of order $n$ and repetition $m$ of an intensity image $f(x, y)$ is defined as

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho, \theta) V_{nm}^*(\rho, \theta) \rho d\rho d\theta$$

where $\rho = \sqrt{x^2 + y^2}$ is the length of the vector from the origin to the pixel $(x, y)$ and $\theta = \arctan(y/x)$ is the angle that the vector makes with the $x$-axis. The order $n$ and repetition $m$ are integers that satisfy $n \geq 0$, $n - |m| = (even)$ and $|m| \leq n$. The complex-valued 2-D Zernike basis functions (which are defined within a unit circle) are formed by function

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) exp(jm\theta), \ |\rho| \leq 1$$

where $j = \sqrt{-1}$ and the real-valued Zernike 1-D radial polynomial is given by

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} c(n, m, s) \rho^{n-2s}, \tag{1}$$

where

$$c(n, m, s) = (-1)^s \frac{(n-s)!}{s!((n+|m|)/2 - s)!((n-|m|)/2 - s)!} \tag{2}$$

The radial polynomials satisfy the orthogonal properties for the same repetition,

$$\int_0^{2\pi} \int_0^1 R_{nm}(\rho, \theta) R_{n',m}(\rho, \theta) \rho d\rho d\theta \tag{3}$$

$$= \begin{cases} \frac{1}{2(n+1)} & \text{if } n = n', \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The Zernike basis functions are orthogonal which implies that there is no redundancy of information among the Zernike moments with different orders and repetitions. Thus, each moment is unique and independent of each other [10].

### B. Identifying fish Species using Zernike moments

Templates from known species of fish are gathered. The Zernike moments up to $12^{th}$ order are calculated for each template. These moments are from different combinations of order $n$ and repetition $m$ where $n \geq 0$, $m \leq n$, $|n - m| = 2$. The averages of the moments with the same order and repetition of each species template are calculated. The average of one species template is subtracted from another to get the average difference distance. This result is then halved to get the Zernike moment threshold of the templates.

The templates gathered is the basis for species identification of each unknown fish blob. If the fish blob size exceeds a certain value, the blob's Zernike moments are computed. These moments are then subtracted from the pre-calculated species template moments. Through sorting, the top 5 smallest differences are then compared with the computed Zernike moment threshold. If none of them exceed the threshold, the fish type is then determined by the majority of type of fish correlated with the top 5 smallest differences. Otherwise, the fish can not be identified.

## VII. IMPLEMENTATION

Sample Blue and Non-blue histogram templates are gathered over several frames of the video sequences (UWVS). These are then used to generate the respective mean values for the two types of image templates. Then for every block in the image frame, color histograms are obtained and compared with the previous mean values calculated. The Non-Blue blocks are blackened while the Blue blocks are left unchanged.

When all of the benthic portions of the image have been blackened out, the Canny Detector is used to detect fish contours. It is used twice in the process, one for outlining the contours, and for filling up the space outlined from detection. The Canny detector will also detect contours from the remaining occlusions referring to the unblackened coral boundaries (Fig. 8). We managed to fix this by developing a cleaning algorithm for the coral boundaries (as detailed in Section IV).

In order for fish counting to be accomplished, blobs have to be formed from the fish outlines. Since blobs are computed per frame, difficulties arise when two fish outlines overlap thereby resulting in a single blob count. This brings about inaccurate counts, so that some procedural adjustments are undertaken (e.g. using average counts per time interval instead of computing values for every frame).

Fish templates from two known species of fish (*Acanthuridae* and *Scaridae*) captured in the video are obtained. The Zernike moments for these templates are computed and stored as previously discussed. The image blobs are colored according to its identified species. If it cannot be classified, the blob's color (white) is unchanged.

Fig. 8. **Detecting edges using Canny Detector from remaining corals (unblackened)**

## VIII. Experiments

We tested our system on three Underwater Video Sequences provided by the University of the Philippines Marine Science Institute. These videos were taken by divers in the Tubbataha Reef, Sulu Sea.

We selected the underwater videos with the least viewpoint changes. This video also contains more fish thereby making population estimation and identification more feasible. Any fish that is against the benthic background will be blackened by the algorithms thus excluding it from the fish count.

| Machine Count | Human Count | Overcount |
|---|---|---|
| 79 | 74 | 5 |
| 67 | 62 | 5 |
| 81 | 83 | -2 |
| 106 | 99 | 7 |
| 74 | 70 | 4 |
| 85 | 78 | 7 |
| 68 | 64 | 4 |
| 63 | 60 | 3 |
| 82 | 76 | 6 |
| 89 | 85 | 4 |
| 105 | 102 | 3 |
| 70 | 67 | 3 |
| 79 | 75 | 4 |
| 73 | 70 | 3 |
| 69 | 65 | 4 |
| 76 | 73 | 3 |
| 70 | 67 | 3 |
| 72 | 68 | 4 |
| 76 | 73 | 3 |
| 67 | 62 | 5 |

TABLE I. Machine and Human Fish Counts

| MSI Video | Length(min) | Process time (mins) |
|---|---|---|
| 000 | 13:10 | 67.28 mins |
| 001 | 12:25 | 62.67 mins |
| 002 | 11:11 | 58.18 mins |

TABLE II. Processing time

For comparison purposes, the video stream was frozen at random time intervals and each fish was manually counted. This was then compared with the program's count. The two counts are tabulated in Table I. We observe that in most cases there is an overcount of less than 10 %.

For a twelve minute video the processing time from the pre-processing, detection and blob counting was computed and

| Video at 00:05 | | Machine | |
|---|---|---|---|
| | Human Count | Machine Count | Correctly Identified |
| Identifiable Fish | 5 | 5 | 4 |
| Sturgeon Fish | 1 | 2 | 1 |
| Parrot Fish | 4 | 3 | 3 |
| Video at 00:27 | | Machine | |
| | Human Count | Machine Count | Correctly Identified |
| Identifiable Fish | 6 | 7 | 5 |
| Sturgeon Fish | 3 | 5 | 2 |
| Parrot Fish | 3 | 2 | 1 |
| Video at 00:27 | | Machine | |
| | Human Count | Machine Count | Correctly Identified |
| Identifiable Fish | 6 | 7 | 5 |
| Sturgeon Fish | 3 | 5 | 2 |
| Parrot Fish | 3 | 2 | 1 |

TABLE III. Machine vs Human (Species Identification)

the ratio was 1:5 (one minute of video takes about 5 minutes of processing). Further optimization is expected to reduce this ratio considerably. Table II shows the processing time of the rest of the video sequences.

The image blobs are individually computed per frame for species identification hence it takes the system longer to process the video.

Fig. 9 shows the fish that can be identified visually by a human (left, manually highlighted in red) compared with fish identified by the system (right). Table III shows the results of the comparisons from three different time instances of the input video.

It may be gleaned from Table II that machine count is greater than the human counterpart. This is attributed to the fact that some small parts of the corals that were not blackened were counted as blobs, despite the pre-processing and cleaning stages. Clearly, background elimination is difficult to achieve, leading to this overcount.

Species identification by the system shows results which are almost the same as human count. However, there are instances of wrong associations of fish type, which is deemed within allowable limits.

## IX. Conclusion

We have developed methods for fish counting and species determination using color histograms, Canny Edge detection, blob counting through connected component labeling and Zernike moment based - shape analysis. The overall performance appear to be a viable alternative to manual methods. Further work include extending species identification to more than two species and the use of color and texture for more accurate determination of fish species.

## References


[1] S. Cadieux, F. Lalonde, and F. Michaud, Intelligent system for automated fish sorting and counting, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[2] B.H. Robinson, The coevolution of undersea vehicles and deep-sea research, *Marine Technology Society Journal*, 33: 69-73, 2000.

[3] D.R. Edgington, D.E. Cline, D. Davis, I. Kerkez, and J. Mariette, Detecting, Tracking and Classifying Animals in Underwater Video *OCEANS 2006*, Moss Landing, CA, Sept. 2006.


Fig. 9. **Screenshots of the processed video for Species Identification**


[4] A. Rova, G. Mori, and L.M. Dill, . Recognizing Fish in Underwater Video, *IAPR Conference on Machine Vision Applications*, 2007.

[5] C. Spampinato, D. Giordano, R. Di Salvo, Y.H. Chen-Burger, R.B. Fisher, and G. Nadarajan, Automatic Fish Classification for Underwater Species Behavior Understanding, *Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, pp. 45-50, 2010.

[6] P.X. Huang, B.J. Boom, and R.B. Fisher, , Underwater Live Fish Recognition using a Balanced-Guaranteed Optimized Tree, *Proceedings of the Asian Conference on Computer Vision*, Daejeon, Korea, 2012.

[7] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (1986): 679-698.

[8] Gonzalez, R.C. and Woods, R.E., *Digital Image Processing (3rd ed.)*, Prentice Hall, 2007., pp. 719 - 725

[9] A. Khotanzad, Invariant image recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 (1990): 489-497.

[10] S.K. Hwang and W.Y. Kim, A novel approach to the fast computation of Zernike moments, *Pattern Recognition*, 39(11), 2065-2076