Fish Population Estimation from Underwater Video Sequences Using Blob Counting and Shape Analysis

Suman Sharma, Aman Shakya

Institute of Engineering, Central Campus, Pulchowk, IOE, TU, Nepal

Corresponding Email: 069msice619@ioe.edu.np, aman.shakya@ioe.edu.np

Abstract: Fish population estimation and classification of fish species have been an integral part of marine science. These tasks are important for the assessment of fish abundance, distribution and diversity in marine environments. Underwater video measurement systems are used widely for counting and measuring fish in aquaculture, fisheries and conservation management. This paper is presented the techniques used for the detection, identification, measurement, and counting of fish in underwater video image sequences, including consideration of the changing body shape, color and texture of fish. It presented simple method of counting fish as blob counting, which automatically using image processing techniques. The detection algorithm, canny edge detection algorithm is used. Coral-blackening process is used to distinct fish and the background. A video of fish was taken every frame is processed singly and independently. Finally each Zernike moment is calculated for each blob with the template of type one fish, the blobs is counted the number of type one fish for the blobs whose amplitude moment is near about zero and the average number of type one fish over the frame is recorded.

Keywords: video, underwater, fish counts, image matching, motion analysis, blob counting, and Zernike moment.

1. Introduction

The study of underwater species is a fascinating topic to marine biologists and environmental experts. Video survey is a popular approach to study marine life. To determine the size and distribution, or to study the behavior of species, the researchers need to locate them in images or videos, but this can be very time consuming when processing a large volume of data. Automated image and video segmentation helps to speedup this tedious work. The purpose of this paper is to identify type of fish in the video or images from given sample, by analyzing and exploring from various visuals information [1].

The monitoring of fish for stock assessment in aquaculture, commercial fisheries and in the assessment of the effectiveness of biodiversity management strategies such as Marine Protected Areas and closed area management is essential for the economic and environmental management of fish populations. Video based techniques for fishery independent and non-destructive sampling are widely accepted. The advantages of using stereo-video for counting the numbers of fish, measuring their lengths and defining the sample area have been well demonstrated. In order to study the effects that climate change and pollution has on the environment, long-term monitoring of the environment is necessary. One of the most important natural environments on earth is coral reefs, however monitoring the fish population and biodiversity is still a challenging task. Data collection in this kind of environment is labor intensive, requiring divers to count the fish species in a certain area. In recent years, stereo-video recording has become much cheaper which makes underwater cameras a good alternative for data collection. Furthermore, automatic video processing and pattern recognition is able to process this kind of data [2, 3, 4].

However, the time lag and cost of processing video imagery decreases the cost effectiveness and uptake of this technology. Current research aims to minimize or completely eliminate the involvement of the human operator in the process of recognition and length measurement of fish recorded by underwater stereo-video surveys. The ultimate goal is to fully automate the recognition and measurement, in order to deal with the many thousands of hours of stereo-video footage that is routinely captured each time. Advances in automated techniques will substantially decrease the cost of processing and make the technology more accessible to a broad spectrum of end users [4,5].

The next significant advances in the technology of stereo-video monitoring of wild and aquaculture fish must be the automated candidate identification and body shape reconstruction of the fish in order to directly extract volumes, and potentially the identification of individual animals to validate sampling and monitor growth. Automated measurement will also enable monitoring of the condition of fish, at least to the extent of estimating the frequencies of superficial injuries and potentially identify infestation levels or secondary infections caused by parasites such as sea lice or skin and gill flukes. For obtaining accurate measurements, it is quite important to precisely locate the reference points (e.g.

tip of the snout and the valley point of the tail). Once the Haar detector gives the location of snout and tail, pin-pointing reference points can be done by matching templates of snout and tail having the reference point explicitly marked.

The last step is to establish correspondences between the stereo image pairs such that the snout and tail of each fish is correctly associated with the corresponding image points in the stereo pair. Corresponding points in stereo image pairs lie on epipolar lines as shown in figure 1. Therefore, the search space for correspondences is one dimensional. Automatically establishing correspondences in stereo image pairs is a well-studied problem in photogrammetry and computer vision, both in a dense and a sparse fashion. Dense methods for correspondence establishment usually rely on block matching to compute disparity map. Sparse methods, on the other hand, first identify key points in the images and then capture texture around those locations in a feature descriptor such that correspondences can be established between key-points extracted from the stereo image pairs. Use of a Haar classifier followed by template matching already provides the locations to be matched in the stereo pair. Hence, the search for the corresponding snout / tail reference point in one image can be restricted to the epipolar line corresponding to the snout / tail reference point in the second image [6,7,8].

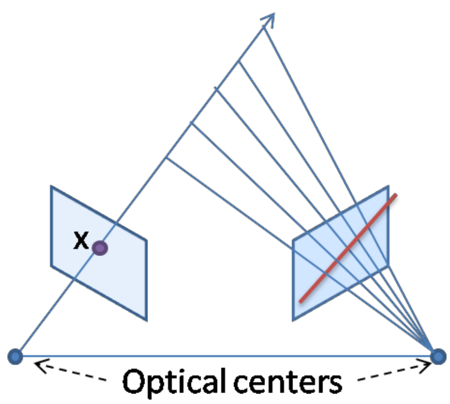

Figure 1: An illustration showing stereo-image geometry and the epipolar line in the right image of the stereo pair.

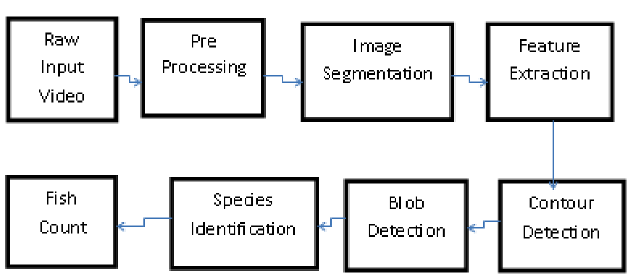

Figure 2: Block Diagram of Proposed System

1.1 Pre-Processing

The underwater video sequences undergo pre-processing in order to eliminate unwanted complex background. This involves Coral Blackening to blacken out corals, and the use of Canny Edge detection, edge cleaning algorithms to refine the blackening process.

1.2 Coral Blackening Procedure

In order to determine which parts are blackened and which are not, the entire frame of size 320 by 240 pixel histograms are computed for a frame. Since the background against which the fish are clearly visible is predominantly water color, only compute the histogram of the background component of the image. The manually selected frames as templates from the Underwater Video Sequences, different samples of 20 × 20 pixel blocks, half of which are from the water back- ground and the remaining half are from benthic (mainly coral) background. The water color and non-water histogram templates were averaged and saved for later comparison. Each frame, having a dimension of 320 × 240 pixels are then divided into different blocks. The histograms for each of these blocks are computed and compared with the water and non-water histogram templates. If the value of a block's histogram is closer to that of a non-water template, that block will be blackened. Although this step is effective in blackening out most of the 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-water color.

1.3 Histogram Analysis

Intensity transformation functions based on information extracted from image intensity histograms play a central role in image processing, in areas such as enhancement, compression, segmentation, and description. The focus of this section is on obtaining, plotting, and using histograms for image enhancement.

1.3.1 Generating and Plotting Image Histograms

The histogram of a digital image with L total possible intensity levels in the range [0, *G*] is defined as the discrete function

$h(r_k) = n_k$ .... (1)

Where r_k is the kth intensity level in the interval [0, *G*] and n is the number of pixels in the image whose intensity level is r_k. The value of *G* is 255 for given images. Sometimes it is necessary to work with normalized histograms, obtained simply by dividing all elements of n_k by the total number of pixels in the image, which we denote by n.

$$p(r_k) = \frac{n_k}{n} \quad (2)$$

Where, for integer images, k= 0, 1, 2…L-1. From basic probability, we recognize $p(r_k)$ as an estimate of the probability of occurrence of intensity level r [12].

1.4 Canny Edge Detection

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several algorithms exists, and this worksheet focuses on a particular one developed by John F. Canny (JFC) in 1986. Even though it is quite old, it has become one of the standard edge detection methods and it is still used in research. The aim of JFC was to develop an algorithm that is optimal with regards to the following criteria.

Detection: The probability of detecting real edge points should be maximized while the probability of falsely detecting non-edge points should be minimized. This corresponds to maximizing the signal to noise ratio.

Localization: The detected edges should be as close as possible to the real edges.

Number of responses: One real edge should not result in more than one detected edge (one can argue that this is implicitly included in the first requirement).

Algorithm

Step 1: Smoothing: Smoothing is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the image is first smoothed by applying a Gaussian filter.

Step 2: Finding gradients: The Canny algorithm basically finds edges where the grayscale intensity of the image changes the most. These areas are found by determining gradients of the image. Gradients at each pixel in the smoothed image are determined by applying what is known as the Sobel-operator.

Step 3: Non-maximum suppression i.e. only local maxima should be marked as edges.

Step 4: Double thresholding: Potential edges are determined by thresholding.

Step 5: Edge tracking by hysteresis, final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge [6].

1.5 Zernike Moments

Zernike moments are the mappings of an image onto a set of complex Zernike polynomials. Since Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlap of information between the moments. Zernike moments are significantly dependent on the scaling and translation of the object in an ROI. Nevertheless, their magnitudes are independent of the rotation angle of the object. Hence, we can utilize them to describe shape characteristics of the objects. For instance, we took the advantage of Zernike moments to extract the shape information of benign and malignant breast masses.

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. Here choosing Zernike moments for this system because of these important properties match the requirements for fish species identification.

$$Z_{nm} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(\rho,\theta) V_{nm}^{*} \rho \, d\rho \, d\theta \quad (3)$$

Where $\rho = (x^2 + y^2)^{1/2}$ 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 axis. The order n and repetition m are integers that satisfy

$$n \geq 0,\ n - |m| = (even) \text{ and } |m| \leq n. \quad (4)$$

The complex-valued 2-D Zernike basis functions (which are defined within a unit circle) are formed by function.

$$V_{nm}(\rho,\theta) = R_{nm} \exp(jm\theta), \quad |\rho| \leq 1 \quad (5)$$

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

$$R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} C(n,m,s) \rho^{n-2s} \quad (6)$$

Where, $C(n,m,s) = (-1)^s \dfrac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}$ … (7)

The radial polynomials satisfy the orthogonal properties for the same repetition

$$\int_0^{2\pi} \int_0^1 R_{nm}(\rho,\theta) R_{n',m}(\rho,\theta) \rho \, d\rho \, d\theta = \begin{cases} \frac{1}{2(n+1)} & if\ n = n' \\ 0 & otherwise \end{cases} \quad (8)$$

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 [6].

1.6 Blob Counting

The Blob detector used is based on the Laplacian of the Gaussian . 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 are 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 neighbors. 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 image. The results can now be used for counting, filtering or tracking [6].

2. Algorithm of Proposed System

Step 1: Raw input video

Step 2: Convert video to frame.

Step 3: Apply the identified thresholds and create a binary image, remove any regions less than threshold, crop the original image to the rectangle containing the remaining candidate regions.

Step 4: Apply coral blackening for a selected image sample.

Step 5: Apply canny edge detection for the detection of fish.

Step 6: Image areas classified as non-background fish candidates are then subject to morphological filters, erosion and closing, dilation and opening, and a median filter;

Step 7: The count of fish in the frame is then determined by a blob counting. Apply dilation to remove noise, expand and merge adjacent regions, then apply erosion to restore the external boundaries of the regions;

Step 8: Count the total blob on a selected frame.

Step 9: Calculate Zernike moment form a template fish of type one with the selected frame.

Step 10: If the Zernike moment is found as less than one then the count the blob is type one fish and go back to step 9.

Step 11: Display the total fish count on a selected frame.

2.1 Curvature Calculation

After retrieving the smoothed fish contour, the next step is to compute the curvature of each point along the boundary. The computing curvatures of the boundary pixels are to find the corners of the fish contour. By doing that, important locations, such as tail tips, the joints points of tail and body are supposed to be located.

The equation for computing the curvature of point u = {x, y} is:

$$K(u) = \frac{x'y - x''y'}{(x'^2 + y'^2)^{3/2}} \ldots\ldots\ldots\ldots\ldots\ldots \quad (9)$$

where x' and y' represent the first derivative of x and y, x'' and y'' denote the second derivative of x and y.

2.2 Species Identification

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

Figure 3: Template of Fish type

Identification of fish species as shown in figure 3, the image moment features of the fish blob is to be calculated. Since small fish blobs may not contain enough information for the identification of selected fish type. So observe a selected type of image is the moment calculation is near about zero and phase is reversed.

3. Implementation

Sample water and non-water color histogram templates are obtained from the selected frame of the video sequences. It is then used to generate the respective mean values for the two types of image templates. When all of the benthic portions of the image have been blackened out, the Canny Detector is used to detect fish contours. 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. Fish templates from known species of fish captured in the video are obtained. The Zernike moments for the template is computed and stored as previously discussed. The image blobs are colored according to its identified species

4. Experiments

Here the video was taken from the Sesnarayan pond located at Farping. The selected underwater video was 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. Average of data is counted from a three conjugative frames. Average data is counted for a three different sequence of video. Each fish was manually counted. This was then compared with the program's count. The counts are shown in table 1, which shows machine versus human count for each frame. The processing time from the pre-processing, detection and blob counting was computed. The image blobs are individually computed per frame for species identification hence it takes the system longer to process the video. The average data of three different sequences are shown in table 2. Figure 5, figure 6 and figure 7 are shown the original frames for first sequence. It may be gleaned from the bar shown in figure 4 that machine count is less than the human count for every sequence. This is attributed to the fact that some fish that were blackened during coral blackening. Species identification by the system shows results which are almost the same as human count.

Table 1: Date from three different frames for 1st sequence

Frame	Machine Count	Human count	Matched Template
Frame 1	27	35	10
Frame 2	32	38	9
Frame 3	28	35	14

Table 2: Average data for three different sequences

Sequence	Machine count	Human count	Matched Template
1st	29	36	11
2nd	33	37	14
3rd	26	35	14

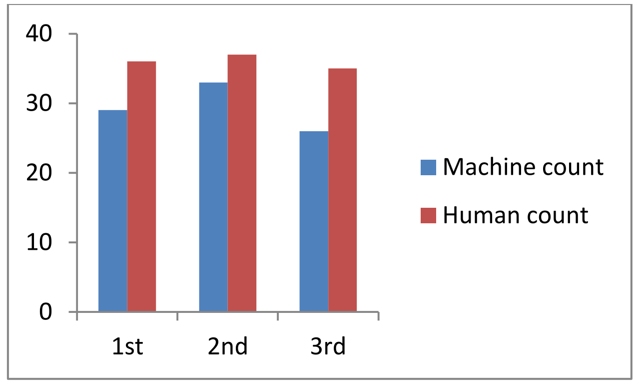

Figure 4: Machine versus Human count bar diagram

5. Conclusion

The goal of this system is to explore methods to segment fish components, investigate the possible methods of counting fish components and develop classifiers that can discriminate different component categories. Automated detection of underwater species from a large volume of underwater video is useful in marine studies, e.g., marine biology, fisheries, and coral reef research. The task can be difficult because of the complexity and variability within the benthic environments. Some key factors include the water depth and visibility, lighting condition, time of day, and certainly the image resolution. The intensity characteristics of fish and the water column are so different; that the water column is typically brighter than fish, so it is used to detect fish within the water column.

Figure 5: Original image of frame 1

Figure 6: Original image of frame 2

Figure 9: Coral Blackening of frame 1

Figure 7: Original image of frame 2

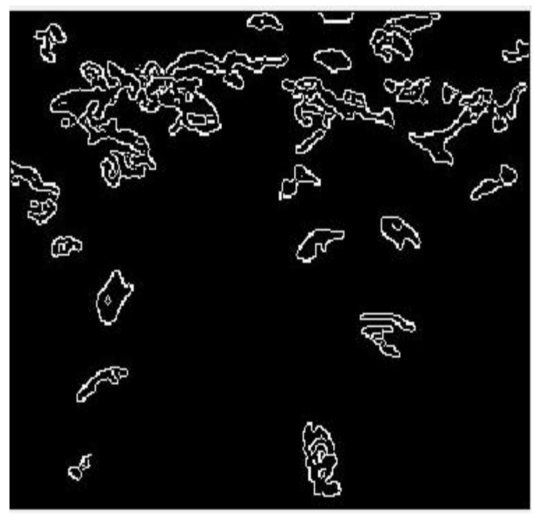

Figure 10: Canny Edge Detection of frame 1

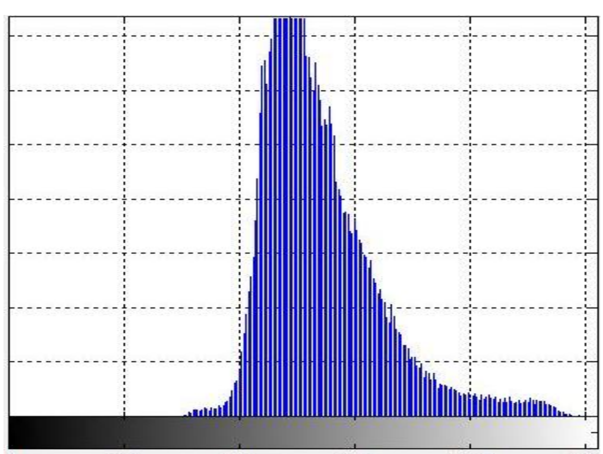

Figure 8: Histogram Analysis of frame 1

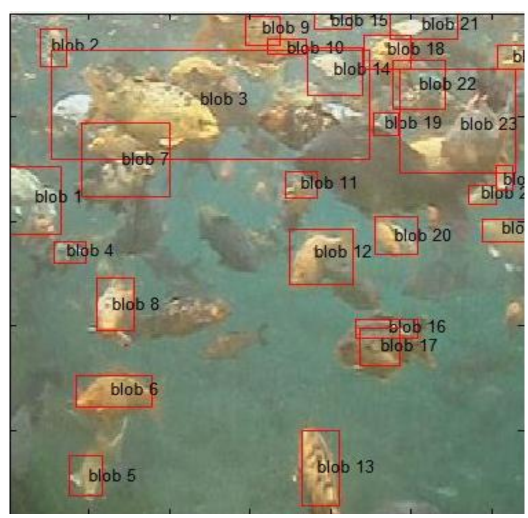

Figure 11: Blob Counting of frame 1

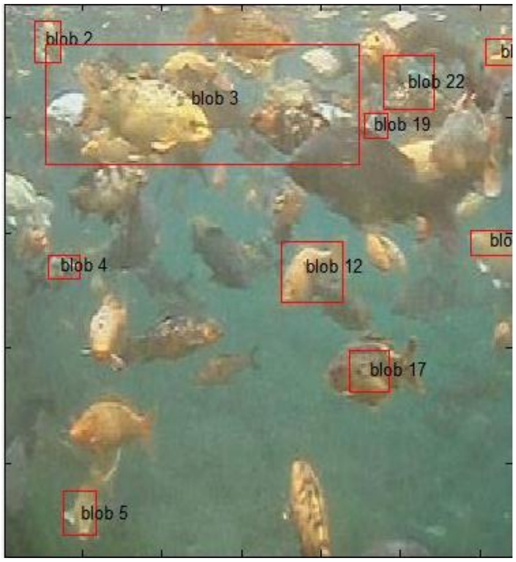

Figure 12: Applied Zernike moment to frame 1

References

[1] Spampinato, Concetto, et al. "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*. ACM, 2010.

[2] Shortis, Mark R., et al. "A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences." *SPIE Optical Metrology 2013*. International Society for Optics and Photonics, 2013.

[3] Boom, Bastiaan J., et al. "Long-term underwater camera surveillance for monitoring and analysis of fish populations." *VAIB12* (2012).

[4] 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.

[5] Toh, Y. H., T. M. Ng, and B. K. Liew. "Automated fish counting using image processing." *Computational Intelligence and Software Engineering,2009. CiSE 2009. International Conference on*. IEEE, 2009.

[6] Fabic, J. N., et al. "Fish population estimation and species classification from underwater video sequences."*Underwater Technology Symposium (UT), 2013 IEEE International*. IEEE, 2013.

[7] Cootes, Timothy F., et al. "Active shape models-their training and application."*Computer vision and image understanding* 61.1 (1995): 38-59.

[8] Sclaroff, Stan, and Lifeng Liu. "Deformable shape detection and description via model-based region grouping." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.5 (2001): 475-489.

[9] Clement, Pierre M., et al. *Fishery applications of optical technologies*. Eds. James Churnside, Michael Jech, and Eirik Tenningen. International Council for the Exploration of the Sea, 2012.

[10] Schuster, Stefan, and Silke Amtsfeld. "Template-matching describes visual pattern-recognition tasks in the weakly electric fish Gnathonemus petersii."*Journal of experimental biology* 205.4 (2002): 549-557.

[11] Mahmood, Arif, and Sohaib Khan. "Correlation-coefficient-based fast template matching through partial elimination." *Image Processing, IEEE Transactions on*21.4 (2012): 2099-2108.

[12] Yu, Ning, et al. "Underwater range-gated laser imaging system design with video enhancement processing." *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on*. IEEE, 2013.

[13] Boom, Bastiaan J., et al. "A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage." *Ecological Informatics* (2013).

[14] Mokhtarian, Farzin, and Riku Suomela. "Robust image corner detection through curvature scale space." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.12 (1998): 1376-1381.

[15] Lee, D. J., et al. "An automated fish species classification and migration monitoring system." *Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE*. Vol. 2. IEEE, 2003.