Automatic Fish Counting in Underwater Video

Cuenta Automática de Peces en Video Submarino

Comptage Automatique de Poisson dans la Vidéo Sous-Marine

HAZEM KHANFAR[1], DIMITRIOS CHARALAMPIDIS[1], E. J. YOERGER[2], GEORGE IOUP[2],
JULIETTE IOUP[2], and CHARLES THOMPSON[3]
[1]University of New Orleans, Department of Electrical Engineering, New Orleans, Louisiana 70148 USA. Hazem.Khanfar@aauj.edu. dcharala@uno.edu. [2]University of New Orleans, Department of Physics New Orleans, Louisiana 70148 USA. ejyoerge@my.uno.edu. geioup@uno.edu. jioup@uno.edu.
[3]NOAA Southeast Fisheries Center, Stennis Space Center, Mississippi 39529 USA. Charles.H.Thompson@noaa.gov.

ABSTRACT

Underwater video is currently being used by many scientists within NMFS to study fish populations and to do a census of species. The Southeast Fisheries Science Center uses such systems to assess populations of reef fish in the Gulf of Mexico. The manual nature of this analysis is time consuming and labor intensive. The goal is to develop software tools and algorithms to automate fish counting in underwater video. The main step is to recognize the presence of fish in the images and track the locations of individual fish from frame to frame. Recognizing the presence of fish is achieved via a histogram thresholding technique. Tracking is performed using a simple linear motion model. This allows automated counting of the number of fish in a time segment and exporting information about each fish. The relative size and shape of each individual fish changes as its location and viewed aspect change. However, most of the fish imaged will be moving against a more or less stationary background, and that motion is used to aid in detection. Enumerating fish targets imaged during a given time period requires tracking the location of each individual through the subset of images where it appears so that it is not counted multiple times. After each region containing fish is isolated, region growing is used. This makes possible the accurate counting of the number of fish and rejection of isolated regions which are not fish. Results demonstrate the performance of the proposed fish counting technique.

KEY WORDS: Computer counting, underwater video, census

INTRODUCTION

Underwater imaging systems are currently being used by many scientists within the National Marine Fisheries Service (NMFS) to observe, identify, and quantify living marine resources (Somerton and Gledhill 2005). The Southeast Fisheries Science Center uses such systems to access populations of reef fish in the Gulf of Mexico. Processing of image sequences is typically performed by human analysts who review the sequence and transcribe their observations. The manual nature of this analysis is time consuming and labor intensive; a twenty (20) minute segment of video data may require an entire workday for a human to analyze. In general, systems and algorithms for the collection and analysis of living marine resource data have been found to be of importance (Cowen et al. 2008, Hare et al. 2008, Semani et al. 2002). The goal of this work is to develop software tools and algorithms to automate analysis of sequences of underwater images of fish.

The first steps to automated processing are to develop algorithms to recognize the presence of fish in the images and track (Cohen et al. 1998) the locations of individual fish from frame to frame (Tsechpenakis et al. 2008). This will allow automated counting of the number of fish in a time segment and the exporting of information about each fish. The relative size and shape of each individual fish change as its location and view aspect change. However, most fish move against a more or less stationary background, and that motion can be used to aid in detection. Enumerating fish targets imaged during a given time period requires tracking the location of each individual through the subset of images where it appears so that it is not counted multiple times.

After each region containing fish is isolated, two methods of completing the fish images, edge detection and region growing are used. These make possible accurate counting of the number of fish and rejection of isolated regions which are not fish. Tracking is used to establish the number of fish correctly. The algorithm and examples are presented. With the success of the research model discussed in the following section, a graphical tool built on this model was developed to assist operators in the fish counting efforts. Details of this ongoing effort are discussed.

ALGORITHM OUTLINE

Counting

The algorithm flowcharts are presented in Figures 11 and 12. The first step in the analysis is to use a number of frames to generate an initial average frame representing the background (Figure 1). The same frames are re-read, and only the "no fish" frames (determined by the thresholding process described next) are averaged to produce the reconstructed background. The reconstructed background is dynamically updated with every new "no fish" frame.

Figure 1. Initial background from average of first 200 frames.

Each frame is used either for adjusting the background ("no fish" frame) or for fish counting and tracking (frame with fish). Figure 2(a) is an example of a frame with fish. Figure 2(b) shows the frame divided by the background. Regions with background values are approximately equal to 1, while other regions are significantly different to 1.

A histogram for the pixel amplitudes in the divided frame, shown in Figure 2(c), is used to define the thresholds in order to isolate regions containing fish. The peak in the histogram around 1 is mostly associated to the background. Other histogram peaks to the left and right of this peak are most likely associated to fish and are used to determine a low and high threshold. Intensity values between the low and high thresholds are replaced with zeros (black pixels) and values below the smaller threshold and above the higher threshold are set equal to one (white pixels), as shown in Figure 3(a).

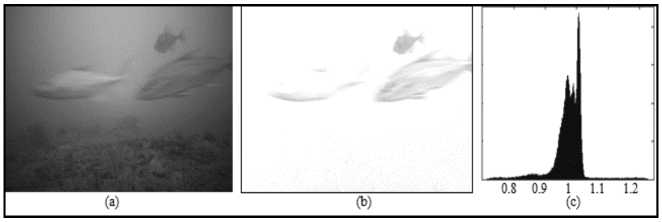


Figure 2. (a) Processed frame with three fish, (b) Divided frame (frame/divided by background), (c) the 10 point histogram.

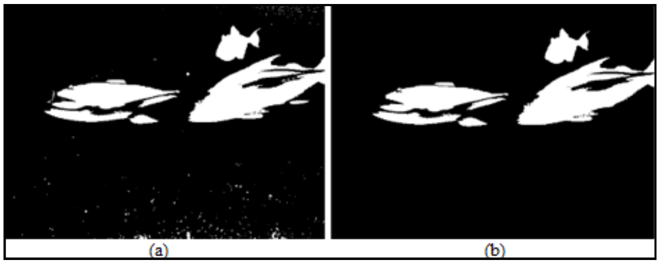


Figure 3. (a) Binary image after thresholding, (b) Binary images after removing small regions.

After removing regions smaller than 400 pixels, the frame is used either for updating the background (number of regions equal to zero) or for counting (number of regions greater than zero). In Figure 3(b), the total number of regions is six. Therefore, the counting and tracking processes are employed for this frame.

Two distinct approaches, edge detection and region growing, are used for fish counting. Canny edge detection (Gonzalez et al. 2008) is used to identify edges in the background-divided frame of Figure 2(b), as shown in Figure 4(a).

In order to connect nearby curves, which most likely correspond to the same fish region, the end-points for each curve are located and connected to the closest end-points associated to a different curve. Two end-points are connected if their Euclidean distance is less than √500 pixels. The result of the process is shown in figure 4(b). Small regions consisting of less than 120 pixels are removed. The number of connected regions shown in figure 4(c) corresponds to the number of fish.

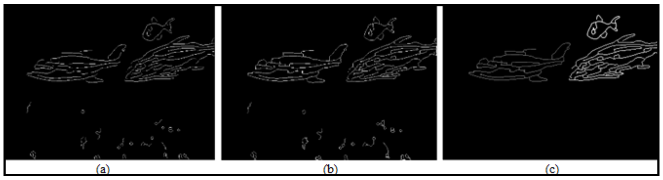


Figure 4. (a) Edge detection using Canny algorithm, (b) End points and curves connected, (c) Small regions removed.

Region growing is the second method for counting and tracking fish. Starting with figure 3(b), the size and location of a minimum box which contains all white regions is determined. The original frame is cropped based on this information, as shown in Figure 6(a). Values outside the box are set to zero. By cropping the image, we concentrate on image blocks with fish in order to eliminate part of the background that may affect the histogram and threshold calculations. The low and high thresholds are then determined for the cropped frame from the histogram in Figure 5(b).

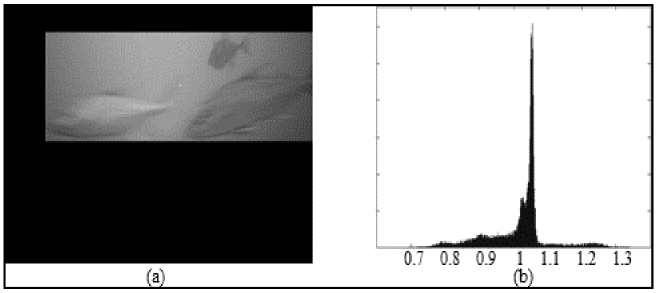


Figure 5. (a) Cropped frame, (b) histogram based on the cropped frame.

The binary image based on the new thresholds and the images identifying the fish regions are shown in figure 6. In Figure 6(b), small regions with fewer than 400 pixels are removed, and different regions are represented by different grayscale values. Figure 6(b) indicates that the area corresponding to the same fish may be split into many regions. Those regions, for a single fish, are located close to each other. Using a closing operation, closely located regions merge, as shown in Figure 6(c).

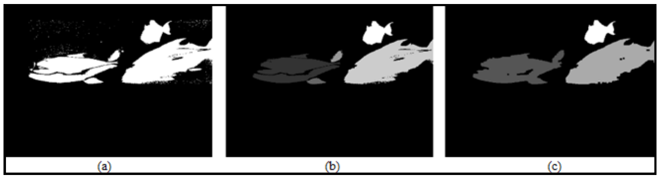


Figure 6. (a) Binary image after applying thresholding, and image regions produced (b) after removing small regions, (c) after employing an opening operation.

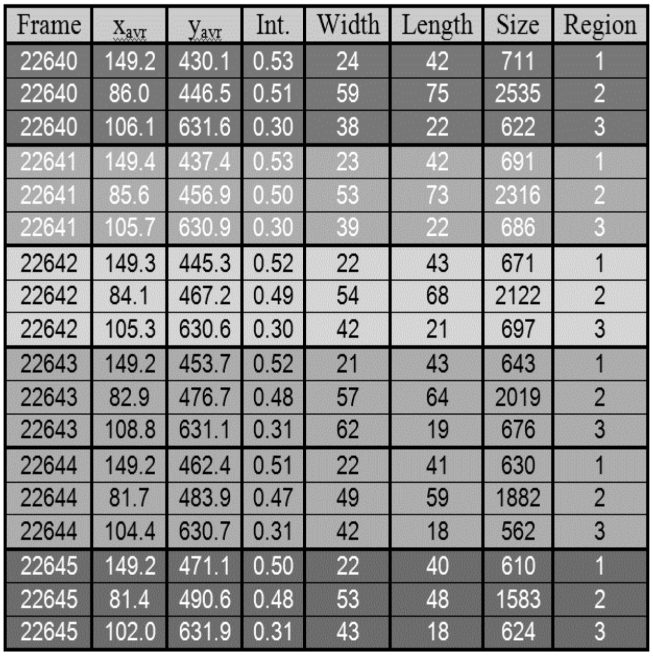
Tracking

Tracking fish from frame to frame helps in the fish counting process, especially when two or more fish regions merge, when fish exit or enter the field of view, or when a fish region appears to be split in two separate regions. In other words, using the information present in several frames provides additional robustness to the algorithm. For instance, assuming that two fish are correctly identified as two distinct regions in a particular frame captured at time *t*, a possible merging of the two regions in the subsequent frame at time *t*+1 may be detected using several criteria. First, the area of the merged region at frame *t*+1 is most likely similar to the total area of the two regions in frame *t*. Moreover, the location of the region in frame *t*+1 is probably close to the location of the two regions in frame *t*. In addition, the motion pattern of the two fish in frame *t* may be used to determine that the two fish regions are likely to be merged in the subsequent frame.

Once the tracks of fish are obtained, they can be analyzed for the purpose of counting the number of tracks associated with different fish, and thus the number of fish. The tracking algorithm extracts specific information from each region in a frame. This information is:

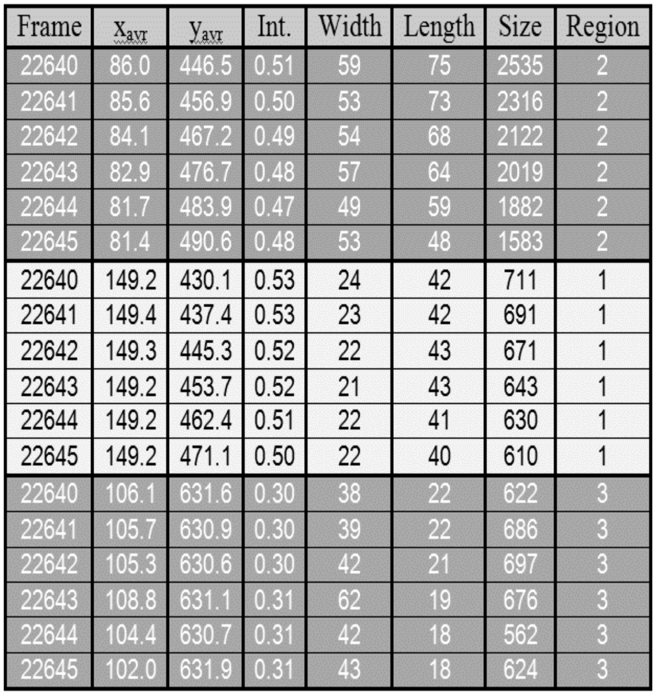
i) The location of the fish (*x* and *y* coordinates of the region's center),
ii) The intensity of the fish region (average intensity of all pixels belonging to the region),
iii) The length of the fish (difference between the two extreme *x* coordinate values which are part of the region),
iv) The width of the fish (difference between the two extreme *y* values which are part of the region), and
v) The size or area of the fish (number of pixels which are part of each region).

Table 1a. Information for regions in six successive frames without tracking.



Frame	x_{avr}	y_{avr}	Int.	Width	Length	Size	Region
22640	149.2	430.1	0.53	24	42	711	1
22640	86.0	446.5	0.51	59	75	2535	2
22640	106.1	631.6	0.30	38	22	622	3
22641	149.4	437.4	0.53	23	42	691	1
22641	85.6	456.9	0.50	53	73	2316	2
22641	105.7	630.9	0.30	39	22	686	3
22642	149.3	445.3	0.52	22	43	671	1
22642	84.1	467.2	0.49	54	68	2122	2
22642	105.3	630.6	0.30	42	21	697	3
22643	149.2	453.7	0.52	21	43	643	1
22643	82.9	476.7	0.48	57	64	2019	2
22643	108.8	631.1	0.31	62	19	676	3
22644	149.2	462.4	0.51	22	41	630	1
22644	81.7	483.9	0.47	49	59	1882	2
22644	104.4	630.7	0.31	42	18	562	3
22645	149.2	471.1	0.50	22	40	610	1
22645	81.4	490.6	0.48	53	48	1583	2
22645	102.0	631.9	0.31	43	18	624	3

Table 1b. Information for regions in six successive frames with tracking.



Frame	x_{avr}	y_{avr}	Int.	Width	Length	Size	Region
22640	86.0	446.5	0.51	59	75	2535	2
22641	85.6	456.9	0.50	53	73	2316	2
22642	84.1	467.2	0.49	54	68	2122	2
22643	82.9	476.7	0.48	57	64	2019	2
22644	81.7	483.9	0.47	49	59	1882	2
22645	81.4	490.6	0.48	53	48	1583	2
22640	149.2	430.1	0.53	24	42	711	1
22641	149.4	437.4	0.53	23	42	691	1
22642	149.3	445.3	0.52	22	43	671	1
22643	149.2	453.7	0.52	21	43	643	1
22644	149.2	462.4	0.51	22	41	630	1
22645	149.2	471.1	0.50	22	40	610	1
22640	106.1	631.6	0.30	38	22	622	3
22641	105.7	630.9	0.30	39	22	686	3
22642	105.3	630.6	0.30	42	21	697	3
22643	108.8	631.1	0.31	62	19	676	3
22644	104.4	630.7	0.31	42	18	562	3
22645	102.0	631.9	0.31	43	18	624	3

Each region may correspond to a single fish, part of a fish (when the fish is split into sub-regions), or more than one fish (when fish regions merge). For example, Table 1 (a) shows the information collected for regions in six successive frames. There are three fish regions in each frame. Each color in the table indicates the information extracted from one frame. The region number shown at the rightmost column does not necessarily remain the same for the same region in two consecutive frames, since region association from frame to frame is not employed for this example. It can be observed from the data presented in Table 1(a) that there is a large region that propagates from one frame to the next, and two other smaller regions. The *x* and *y* location can be used to distinguish one small region from another.

In order to automate the tracking process, we use the Euclidean distance between feature sets corresponding to two different regions. We compute the Euclidean distance for each region in the current frame with respect to all regions in the previous frame. The minimum Euclidean distance is selected to associate the region in a frame to a region in the following frame. The Euclidean distance used in the tracking process is defined as follows:

$$d = (x_{avr\_C} - x_{avr\_P})^2 + (y_{avr\_C} - y_{avr\_P})^2 + (int_C - int_P)^2 + (width_C - width_P)^2 + (length_C - length_P)^2 \qquad (1)$$

In Eq. (1), the subscripts *C* and *P* represent the current and previous frame, respectively. Table 1(b) shows the result of tracking the three regions for the six frames. At this point, region association has taken place, and the same region tracked in successive frames is represented by the same color in the table. It can be observed that the same fish was marked as "region 1" in all frames. This is also the case for the other two fish. Nevertheless, as mentioned earlier, this is merely a coincidence.

Another tracking example is shown in Figure 7. In this example, there is a merging of two fish, as shown in Figure 8(b). The processed frame, depicting four regions instead of five, is given in Figure 7(b).

The fact that the number of regions is reduced to four is also shown in Table 2(a). Since the number of regions is decreased from frame 22555 to frame 22556, it is expected that either region merging has occurred or the fish has exited the field of view. A simple inspection of Table 2(a) reveals that regions 1 and 3 in frame 22555 have merged into region 1 in frame 22556. The information presented in table 2(a) is prior to tracking, therefore the region number does not necessarily correspond to the same fish in consecutive frames.

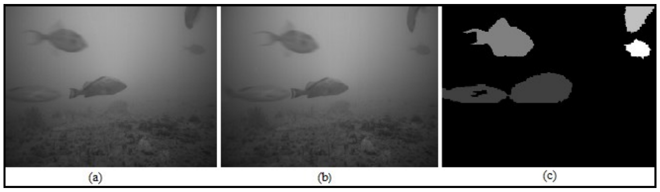


Figure 7. (a) Frame with 5 fish regions, (b) frame with 2 fish merged (4 regions), (c) frame with 5 regions (2 merged).

Figure 12 provides the flow chart for the two tracking methods used for the purpose of fish counting. The bottom half of the flow chart shows the steps when the number of regions changes from one frame to the next. Using the tracking algorithm presented in Figure 12, the regions shown in Table 2(b) are produced. Each color in the table corresponds to the same region. Apparently, the last region (represented by "green") disappears in frame 22556. Moreover, the information about the fourth region is incorrect in frame 22556 because of fish merging. However, by using the set of formulas presented as Eq. (2), we can split the merged region into two regions. Results are shown in Table 5. The information presented in Table 3 is used for counting the number of regions (fish). The region splitting procedure is described following Eq. (2).

$$x1 = x1_{-1} + D_{x1}, \qquad y1 = y1_{-1} + D_{y1}$$
$$D_{x1} = x1_{-1} - x1_{-2}, \qquad D_{y1} = y1_{-1} - y1_{-2}$$
$$x_{\mathrm{error}} = x_{\mathrm{merged}} - (x1+x2)/2, \quad y_{\mathrm{error}} = y_{\mathrm{merged}} - (y1+y2)/2 \qquad (2)$$
$$x1_{\mathrm{new}} = x1 + x_{\mathrm{error}}, \qquad y1_{\mathrm{new}} = y1 + y_{\mathrm{error}}$$

Table 2a. Information for regions in three successive frames without tracking.

Frame	x_{avr}	y_{avr}	Int.	Width	Length	Size	Region
22554	103.3	128.4	0.39	120	221	13640	1
22554	271.1	68.1	0.27	56	150	6788	2
22554	253.2	299.9	0.41	96	168	12280	3
22554	135.6	556.7	0.40	48	78	2424	4
22554	34.7	584.5	0.34	79	110	7025	5
22555	269.3	79.1	0.28	54	170	7261	1
22555	107.9	157.1	0.42	107	216	12050	2
22555	253.7	296.6	0.41	90	175	11968	3
22555	136.1	566.4	0.38	53	74	2360	4
22555	36.7	577.3	0.34	82	91	6125	5
22556	259.2	217.6	0.36	88	381	20046	1
22556	110.5	184.8	0.45	118	210	12527	2
22556	35.1	576.0	0.36	88	108	5950	3
22556	137.0	577.2	0.37	53	82	2538	4

Table 2b. Information for regions in three successive frames with tracking.

Frame	x_{avr}	y_{avr}	Int.	Width	Length	Size	Region
22554	103.3	128.4	0.39	120	221	13640	1
22555	107.9	157.1	0.42	107	216	12050	2
22556	110.5	184.8	0.45	118	210	12527	2
22554	135.6	556.7	0.40	48	78	2424	4
22555	136.1	566.4	0.38	53	74	2360	4
22556	137.0	577.2	0.37	53	82	2538	4
22554	34.7	584.5	0.34	79	110	7025	5
22555	36.7	577.3	0.34	82	91	6125	5
22556	35.1	576.0	0.36	88	108	5950	3
22554	253.2	299.9	0.41	96	168	12280	3
22555	253.7	296.6	0.41	90	175	11968	3
22556	**259.2**	**217.6**	**0.36**	**88**	**381**	**20046**	**1**
22554	271.1	68.1	0.27	56	150	6788	2
22555	269.3	79.1	0.28	54	170	7261	1

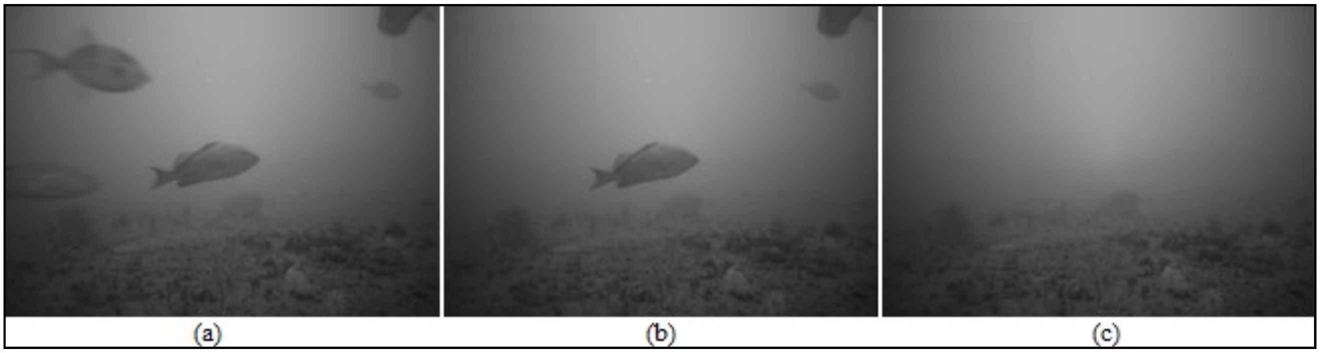


Figure 8. (a) A frame with five fish, (b) two fish are eliminated from original frame, (c) all fish are eliminated.

Table 3. Information for regions in three successive frames after splitting the merged region .

Frame	x_{avr}	y_{avr}	Int.	Width	Length	Size	Region
22554	103.3	128.4	0.39	120	221	13640	1
22555	107.9	157.1	0.42	107	216	12050	2
22556	110.5	184.8	0.45	118	210	12527	2
22554	135.6	556.7	0.40	48	78	2424	4
22555	136.1	566.4	0.38	53	74	2360	4
22556	137.0	577.2	0.37	53	82	2538	4
22554	34.7	584.5	0.34	79	110	7025	5
22555	36.7	577.3	0.34	82	91	6125	5
22556	35.1	576.0	0.36	88	108	5950	3
22554	253.2	299.9	0.41	96	168	12280	3
22555	253.7	296.6	0.41	90	175	11968	3
22556	**252.5**	**319.2**	**0.41**	**90**	**175**	**11968**	**1**
22554	271.1	68.1	0.27	56	150	6788	2
22555	269.3	79.1	0.28	54	170	7261	1
22556	**265.8**	**115.9**	**0.28**	**54**	**170**	**7261**	**5**

In Eq. (2), $x1_{-1}$ and $x1_{-2}$ are the x-coordinates of the center corresponding to region 1 at the previous frame and at the frame before the previous one, respectively. Then, D_{x1} describes the past movement of the region in terms of the x-coordinate. Therefore, assuming that region 1 travels with constant velocity, $x1$ is the expected x-coordinate of the region in the current frame. If a similar process is performed for a second region, i.e., region 2, then $x2$ is the expected x-coordinate of the second region in the current frame.

If the centers of both region 1 and region 2 are found to be closest to the merged region's center with coordinates x_{merged} and y_{merged}, then it can be indeed assumed that it was the overlapping of these two regions that produced the merged region. In this case, the expected x-coordinate of the combined region is equal to the average of the two regions' x-coordinates, namely $(x1+x2)/2$. The error, x_{error}, in Eq. (2) is defined as the difference of the actual x-coordinate of the merged region, i.e., x_{merged} , and the predicted x-coordinate, i.e., $(x1+x2)/2$. In order to find a better estimate than $x1$ regarding the x-coordinate of region 1 for the current frame, it is assumed that the prediction error, x_{error}, found for the merged region, is the same prediction error associated to region 1. Therefore, $x1_{\text{new}}$ is used as the corrected estimate for the x-coordinate of region 1's center for the current frame. The same process is applied to the regions' y-coordinate.

Reconstructed Background

In the described algorithm, the background was updated with "no fish" frames only. However, it is possible that in a particular frame sequence, there is not a significant number of "no fish" frames. This can occur, for instance, when a large number of fish is present nearby the camera location. In such cases, calculation of the background image may be inaccurate, and therefore, the performance of the algorithms may be considerably affected. In order to alleviate the problem, we use all frames to update the background, but we only use image portions with no fish. This is achieved by averaging all available frames pixel-wise, and by replacing, during the averaging process, the frame locations containing fish with the background values calculated at the same locations. As a result, the amount of data available for background calculation is significantly increased.

The background calculation process described above is illustrated via an example. Figure 8 shows a frame containing five fish. In Figure 8(b), two fish are replaced by the corresponding background values. Eventually, all fish are replaced by the corresponding background values, as shown in Figure 8(c). This modified frame is used to update the background.

Edge Detection by Separating the Regions

The edge detection algorithm can be improved by separating each region and applying the algorithm to each separate fish region. Finally, the regions are combined as shown in Figure 9.

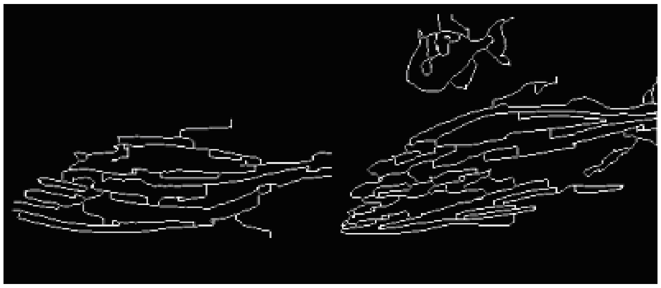


Figure 9. Edge detection technique result after combining all edge regions.

Filling the Edge Detected Regions

The general shape of the fish using edge detection is closer to the actual fish outline than the outline from the region growing method. This can be observed by comparing Figure 7 (region growing) and Figure 9 (edge detection). Therefore, the edge detection method is first used to produce accurate fish outlines, and is then followed by a region-filling approach to produce regions similar to the region growing method. The goal is to obtain more accurate shapes, while being able to extract features (location, area, etc.), as in the case of region growing, which can be used for tracking.

Figure 10 shows the improved edge detection method on a frame containing five fish. Using region growing, each closed region inside the fish edge curves is filled in using 4-neighbor connectivity. This model can detect, identify, and outline several fish within a frame. Next, we discuss a tool developed to implement the research model for use in the field (Figure 11).

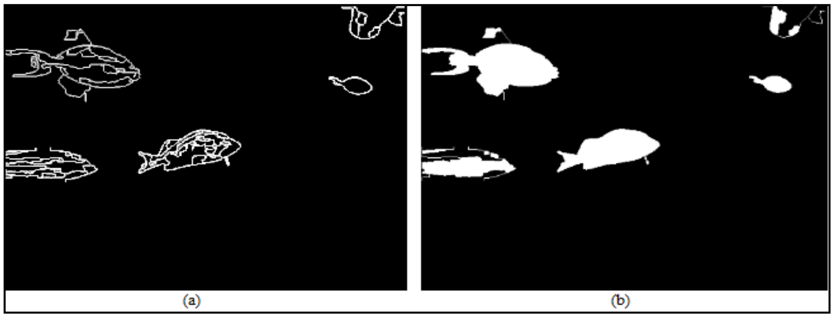


Figure 10. (a) Edge detection result, (b) filled regions.

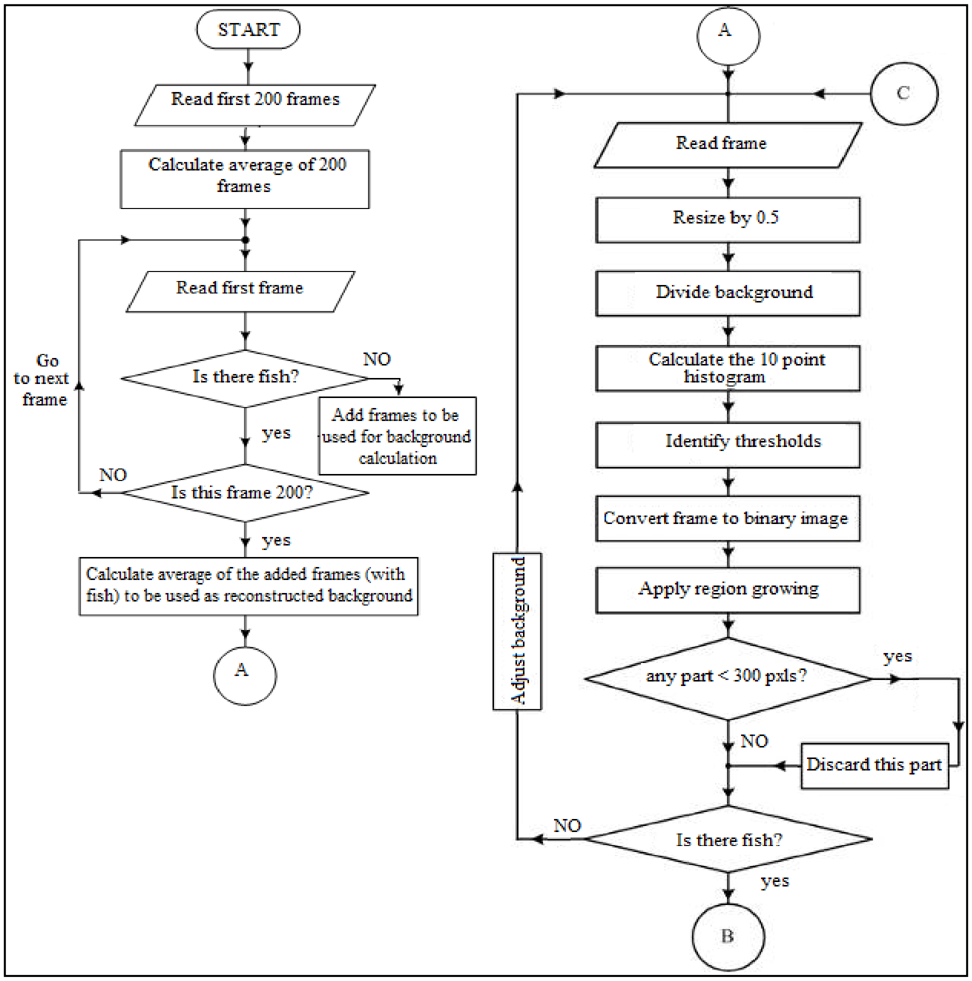


Figure 11. Flow chart of steps used to generate the background (left) and process the frames (right).

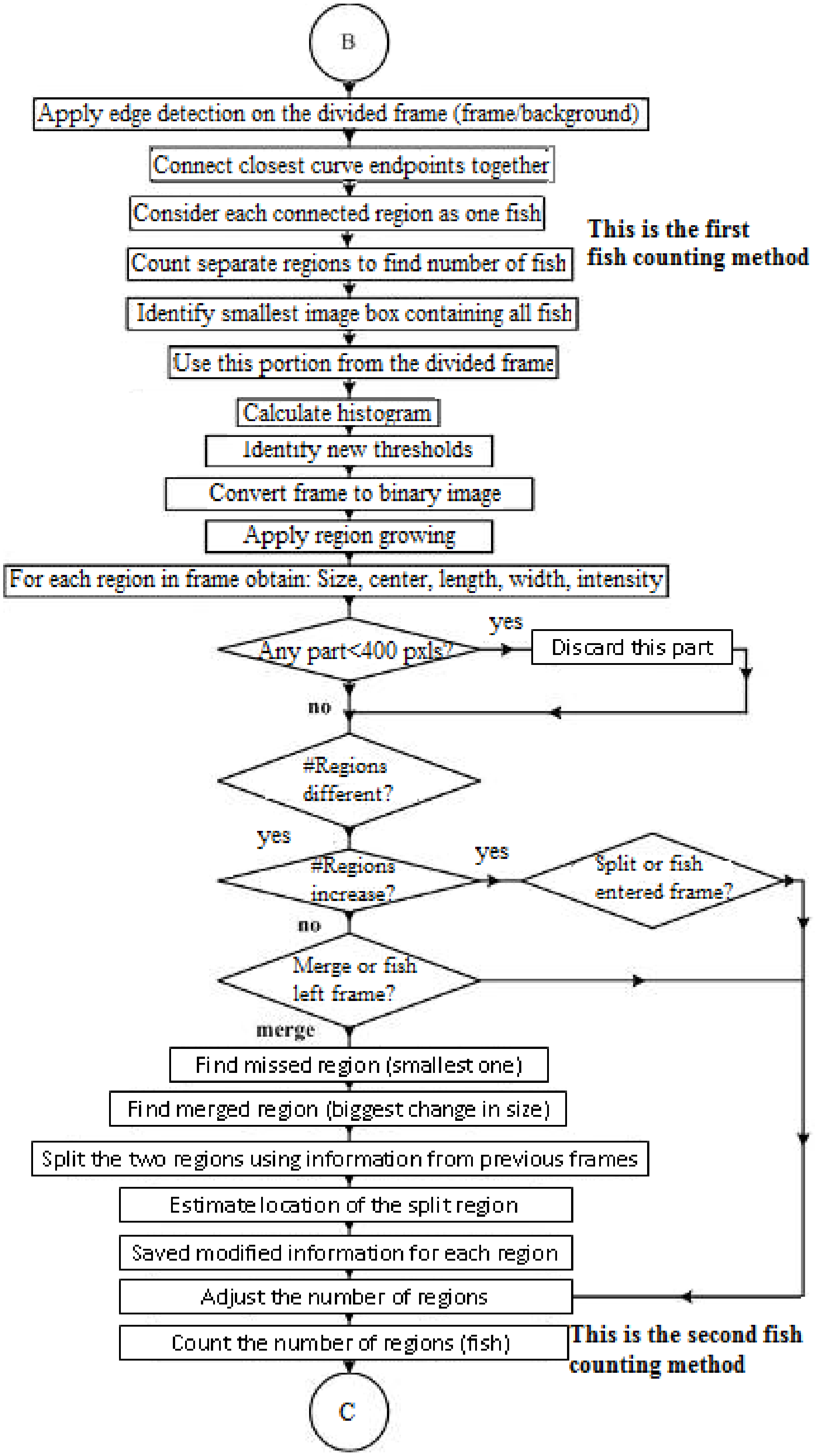

B
Apply edge detection on the divided frame (frame/background)
Connect closest curve endpoints together
Consider each connected region as one fish
Count separate regions to find number of fish
Identify smallest image box containing all fish
Use this portion from the divided frame
Calculate histogram
Identify new thresholds
Convert frame to binary image
Apply region growing
For each region in frame obtain: Size, center, length, width, intensity
Any part<400 pxls?
yes
Discard this part
no
#Regions different?
yes
#Regions increase?
yes
Split or fish entered frame?
no
Merge or fish left frame?
merge
Find missed region (smallest one)
Find merged region (biggest change in size)
Split the two regions using information from previous frames
Estimate location of the split region
Saved modified information for each region
Adjust the number of regions
Count the number of regions (fish)
C
This is the first fish counting method
This is the second fish counting method

Figure 12. Flow chart showing the two methods used for counting.

RESEARCH-BASED GRAPHICAL TOOL

With the promising results from the research model discussed above, the next logical step was to develop a program that could assist operators in the field to ease the burden of manual counting. To this end, a Graphical User Interface (GUI) program was built to automate the fish counting process. This GUI was developed in Matlab (2010a) using its GUIDE (Graphical User Interface Development Environment) program. A compiled executable of the GUI was developed in both 32- and 64-bit formats. The GUI was demonstrated and distributed to personnel and operators at the Southeast Fisheries Science Center in Pascagoula, MS. Presently, the program is undergoing testing with this group.

The featured aspects of the Fish-Counting GUI retain all of the science from the research model while allowing the operator flexibility in setting processing parameters (which were not available in programming the research version). For example, the GUI version can process a large database of video frames by processing equal-sized segments of the database, while the research model could only examine one of these segments. To further understand the various features of the GUI, snapshots of various screens from the processing are shown below in Figure 13 and explanation follows.

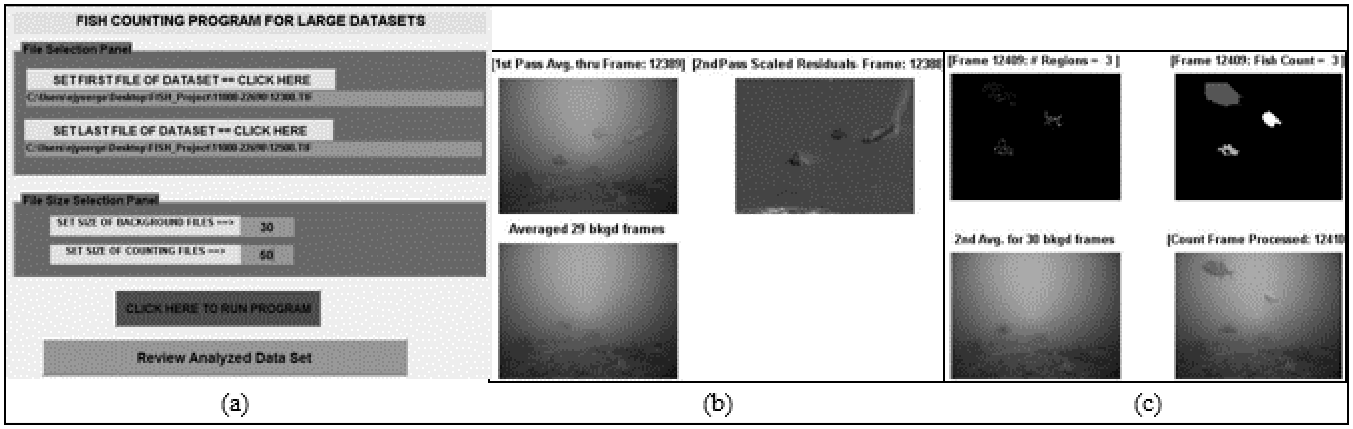
In order to understand the requisite parameters of the GUI, it is helpful to recall the type of data files used by the research model. In order to perform background subtraction, one needs to identify the frames (files, images) used to develop an averaged background file. Once identified and calculated, the frames in which one wishes to count fish must be specified. There is no reason why the background files and fish count files cannot overlap or even be identical. This is something which the operator may wish to vary depending on image quality, illumination, background motion, etc.

Turning to Figure 13, one can now explain how the above process is implemented by the GUI. Remember the purpose of the GUI is to process (i.e. count fish) in a large number of images which we will term a data set. In Figure 13(a), on the "File Selection Panel" of the GUI (top magenta panel), the user can enter the range of frames to be processed by specifying the beginning and ending frame numbers found in the data set. On the "File Size Selection Panel" (middle magenta panel), the user is allowed to set the number of files to be averaged to form a background file (background count) as well as the number of frames for which fish will be counted (fish count). At this point, the user can run the program (red "Run Program" button).

Once the program begins, the processing is straightforward. The main idea is that the program counts the fish in every frame by progressing through the data from the start file through the end file. It advances through the data set by processing contiguous segments of frames whose size is that specified by the fish count. Each segment uses its own, local background files formed by averaging the number of files specified by the background count. The particular files used for the background are always centered in the middle of the fish count segment. Of course, depending on the values for the number of background and fish-count files as well as the length of the data set to be processed, there will be issues with processing the immediate start and end of the data set. However, the program handles all occurrences of these issues.

Examples of the background and counting processes are shown in Figure 13(b) and 13(c). The parameters displayed in Figure 13(a) were used for these examples. Namely, a data set beginning with frame 12300 and ending in 12500 was processed with 30 background files and 50 fish-count files for each segment. Figure 13(b) shows the background processing while Figure 13(c) shows the fish-count processing. It should be noted that some frame-counts differ by 1. This is due to the difficulty in capturing real-time output as all subplots of a figure are generated and displayed at varying times.

Figure 13(b) is displaying the results for the 2nd fish-count segment. The center of this segment is at frame 12374 (12400 – 26). Adding to this frame, half of the background count, one finds the final background file processed for this segment is 12389 as indicated in the


FISH COUNTING PROGRAM FOR LARGE DATASETS
File Selection Panel
SET FIRST FILE OF DATASET == CLICK HERE
SET LAST FILE OF DATASET == CLICK HERE
File Size Selection Panel
CLICK HERE TO RUN PROGRAM
Review Analyzed Data Set
(a)
[1st Pass Avg. thru Frame: 12389]
[2ndPass Scaled Residuals Frame: 12388
Averaged 29 bkgd frames
(b)
[Frame 12409: # Regions = 3]
[Frame 12409: Fish Count = 3]
2nd Avg. for 30 bkgd frames
[Count Frame Processed: 12410
(c)

C:\Users\ejgeorge\Desktop\FISH_Project\1000-22690\12300.TIF

C:\Users\ejgeorge\Desktop\FISH_Project\1000-22690\12500.TIF

SET SIZE OF BACKGROUND FILES ==> 30

SET SIZE OF COUNTING FILES ==> 50

Figure 13. (a) GUI menu for fish counting, (b) background forming process, (c) fish counting process.

subplot titled “1st Pass Avg.” In applying the averaged background file on the 2nd pass, the residual for frame 12388 is shown in the next subplot. The fact that 12388 is one frame from the end of the background processing is evident since the “Averaged Background Plot” has used only 29 files in forming the average to this point.

Figure 13(c) shows the results in applying an averaged, background file to the fish counting process. The top 2 subplots show the result for region selection and growing (left to right subplots, respectively). The bottom 2 subplots show the average, background plot used for the counting and the next frame to be counted (left to right subplots, respectively).

After the entire data set is processed, the user may review the results by selecting “Review Analyzed Data Set” shown in Figure 13(a) (bottom green button). This will display a pop-up window as in Figure 14(a). This figure shows the required information already filled in. Figure 14(b) shows the distribution of counted fish plotted as “Number of Fish” vs “Frame Number.” Clicking on any of the data points (circles) in Figure 14b will automatically display the image corresponding to the selected frame number. In this case, frame 12407 was chosen and is displayed in Figure 14(c). Furthermore, by clicking within the image using the right or left mouse button, the display will move forward or backward, respectively, through adjoining images in the data set. This allows the operator to examine the validity of the fish-count for the frame he has chosen as well as review images adjoining selected frame.

SUMMARY AND CONCLUSIONS

The goal of this work was the development of algorithms that are capable of determining the number of individual fish in a given frame obtained using underwater non-stereo and stereo camera systems. Background subtraction using a histogram-based approach, region growing, and edge detection were used to identify fish regions. In order to assist the fish counting process, a tracking approach was developed. One goal of tracking was to be able to determine when two or more fish regions were merged. Fish region overlapping is an important factor in underestimating the number of fish in a frame. In addition to fish merging studied in this project, fish region splitting will be further investigated in future work.

It was found that edge detection was more successful in describing fish outlines when compared to region growing. On the other hand, region growing was more successful in identifying solid regions associated with fish. A combination of the two techniques was studied in this work and will be investigated further in the future.

Finally, as a further test of the research model, a GUI was developed implementing the fish-counting algorithms. The purpose of the GUI is to provide easy-to-use software for automatic fish counting. Such automation is hoped to relieve the burden of manual fish counting placed upon the operator and allow more time for scientific endeavors.

LITERATURE CITED

Cohen, I. and G. Medioni, “Detecting and Tracking Moving Objects in Video from an Airborne Observer,” DARPA Image Understanding Workshop, IUW98, Monterey, CA, November 1998.

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

Gledhill, et al. 2005. SEAMAP Reef fish survey of offshore banks. Pages 38-39 in: D.A. Sommerton and C.T. Gledhill (eds.) *Report of the National Marine Fisheries Service Workshop on Underwater Video Analysis*. NOAA Technical Memorandum NMFS-F/SPO-68.

Gonzalez, R.C. and R.E. Woods. 2008 *Digital Image Processing, 3rd Edition*. Prentice Hall, Upper Saddle River, New Jersey USA.

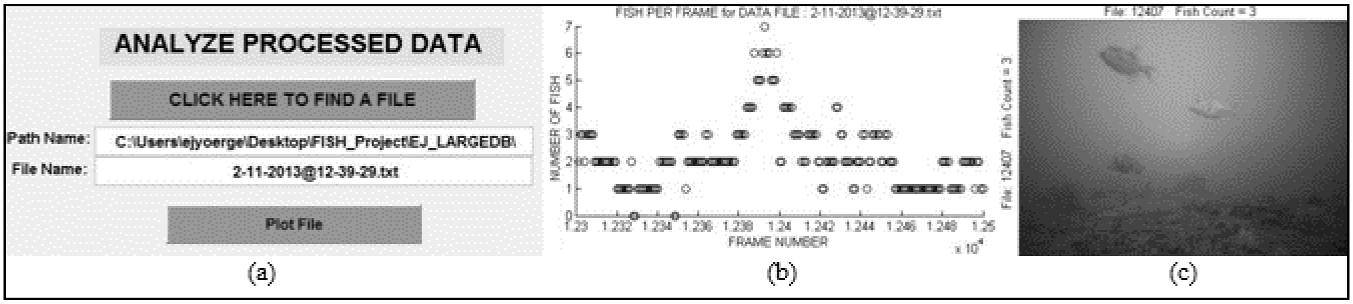
Grace, M., K.R. Rademacher, and M. Russell. 1994. Pictorial guide to the groupers (Teleostei: Serranidae) of the Western North Atlantic. NOAA Technical Report, NMFS 118.

Hare, J.A. and R.K. Cowen. 2008. Development and evaluation of new technology for the remote identification and enumeration of larval fish. *Progress Report to The Office of Science and Technology*.

Semani, D., C. Saint-Jean, C. Frélicot, T. Bouwmans, and P. Courtellemont. 2002. Alive fishes species characterization from video sequences. Pages 689-698 in: T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and D. De Ridder (eds.) *Lecture Notes in Computer Science Volume 2396: Advances in Pattern Recognition SSPR/SPR’02*.

Somerton, D.A, and C.T. Gledhill (eds.) 2005. Report of the National Marine Fisheries Service Workshop on Underwater Video Analysis, NOAA Technical Memorandum NMFS-F/SPO-68.

Tsechpenakis, G., L. Bianchi, D.N. Metaxas, and M. Driscoll. A novel computational approach for simultaneous tracking and feature extraction of *C. elegans* populations in fluid environments,” *IEEE Transactions on Biomedical Engineering* **55**(5):1539-1549.


ANALYZE PROCESSED DATA
CLICK HERE TO FIND A FILE
Plot File
FISH PER FRAME for DATA FILE : 2-11-2013@12-39-29.txt
NUMBER OF FISH
0
1
2
3
4
5
6
7
FRAME NUMBER
1.23
1.232
1.234
1.236
1.238
1.24
1.242
1.244
1.246
1.248
1.25
x 10⁴
File: 12407 Fish Count = 3
File: 12407 Fish Count = 3

Path Name: C:\Users\ejyoerge\Desktop\FISH_Project\EJ_LARGEDB\

File Name: 2-11-2013@12-39-29.txt

(a) (b) (c)

Figure 14. (a) Selection of data set to review, (b) graph of fish distribution, (c) Frame 12407 picked from graph in (b).