Automated fish counting using image processing

Y. H. Toh
National University of Singapore

T.M. Ng, B.K. Liew
Republic Polytechnic of Singapore

Abstract—**This paper presents a simple method of counting feeder fish automatically using image processing techniques. A video of a school of fish is taken and every frame is processed singly and independently. The first step is to obtain blobs marking the positions of the fish. Several ways of accomplishing this task are discussed. Noise and background objects are filtered from the image of the blobs. Area information of the blobs is used to count the number of fish in one frame, and the average number of fish over all frames is then recorded. Experimental results show that the correct number of fish can be obtained for a school of 5, 10, 15, and 50 fish. Errors within frames increase with the number of fish, mainly resulting from the fact that area thresholding can be quite sensitive. Finally, a discussion about the method's effectiveness and possible improvements are provided.**

Keywords: blobs, background estimation, morphologically closing

1. INTRODUCTION

In fish farms and shops, feeder fish are usually packed in plastic bags, to be sold to customers. Packing involves counting the fish to get the right quantity required by customers. However, the counting process is time consuming and can be subjected to human error.

In this paper, we explore how this feeder fish counting can be automated using image processing [1]. Studies such as those in [2] and [3] include some possible ways of fish counting. However, they are designed for bodies of water with different varieties of fish so they include some form of fish species detection algorithm so as to be able to count the number for each species. In addition, fish tracking algorithms are also used in the counting of fish. Thus, their suggested methods are computationally intensive and will increase costs and slow down the counting speed. For feeder fish counting, the problem is simplified because it only involves one species of fish which look similar in shape and size to each other so fish species detection is not needed. Fish tracking will be important in cases in which fish may swim in and out of the video frame so the total number of fish may vary. Since the number of fish in the plastic bag is fixed, fish tracking is not essential although it will improve the accuracy. We instead aim to find a simple-to-execute method that can do the counting operation quickly with good accuracy.

The video of a school of fish is taken and processed to obtain the number of fish present. The key idea behind our proposed method is the use of blobs to mark the position of the fish, and then analyzing the area of each blob to decide how many fish are captured in each blob. This in turns yields the number of fish in a video frame. Individual frames are processed separately, and the number of fish computed from each frame are then added up and divided by the number of frames in the video to obtain an average figure.

2. PROPOSED METHOD

The method that we propose here is specifically meant for counting feeder fish. Feeder fish are mostly uniform in size, and will thus be more likely to produce an accurate count of the number of fish present in a blob.

2.1 Methods of obtaining the initial blob

To obtain a blob, a frame is extracted from the video [4] and then converted to grayscale (see Fig. 1a). The colour information is not needed because the fish have a higher intensity value than the surroundings. Thus, only the intensity information needs to be retained. Next, the image contrast is enhanced [5], with more weights being applied to the lighter background. As a result, the background becomes much lighter while there is only a small change in intensity to the foreground objects (see Fig. 1b). Proceeding from here, we have explored a few methods.

In the first method, the image can be changed to a black and white image by using an intensity threshold. All pixels with luminance greater than the threshold are changed to white and the rest are changed to black. When the threshold is set too low, too little of the shape of each fish is captured (see Fig. 2). On the other hand, when the threshold is too high, due to non-uniform lighting, the top-most corner of the container which is quite dark, is changed to black as well, thus merging with the shape of the fish (see Fig. 3). It is not easy to have a very high contrast image and a uniform background for this method to work well.

Another method is to use edge detection to identify the fish [6]. The canny edge detector is used because it provides the most number of true edges [7]. To eliminate the edges of the container, the image is filled with redundant black data starting from the sides of the image outwards. This set up an additional rectangular edge denoting the original outer border of the image. Morphological closing is done next to join the edges that were not terminated properly and to fill the areas (see Fig. 4). Blobs are thus created from the edges. This step is also necessary to connect the round edges of the container to the original outer border of the image so they can all be eliminated by eliminating blobs which has a major axis much longer than the length of five fish. The final result after eliminating noise and background objects is shown in Fig. 5. Several problems are noted with this method. Firstly, edge detection performed on objects without well-defined borders results in incomplete edges. This can be remedied through morphological closing but this process in turn affects the blob area. Fish that appears singly in the original image may join with neighbouring fish (see Fig. 5). This results in inaccuracies when we use the area to count the number of fish in a blob.

The last method is our preferred method. It is a variation of the first method. For example, the contrast of the image in Fig. 1a is still adjusted to produce image in Fig. 1b but instead of directly applying the threshold to the adjusted image, the process is performed on the difference image which is obtained by subtracting the background from the adjusted image. The technique used to obtain the background is called background estimation [8]. It involves morphologically closing the frame with a suitable structuring element. In our case, a disk with a radius of four pixels is used. As shown in Fig. 6, this allows the container to be retained while the fish are no longer visible. After background subtraction, the resulting image is as shown in Fig 7. This difference image is better than the adjusted image as the container is largely not visible and the fish are more prominent. The complement of the background subtraction result is then subjected to a luminance threshold to yield Fig. 8. The thresholding result is much better than that obtained by directly applying a threshold to the image.

2.2 Noise and Background Object Elimination

All particles with areas smaller than an area threshold are considered as noise and are removed. This area threshold should be adjusted based on the expected size of the fish [9]. An area threshold of 25 pixels is used in our experiment. Similarly, large background objects such as rocks and decorative ornaments can be removed by eliminating blobs which have areas too large to consist of any possible intersection of fish. The probability that an additional fish intersects with a blob decreases as the number of fish already present in the blob increases. A threshold of an estimated area occupied by five intersected fish is thus used, above which the blob is counted as a background object. Fig. 9 shows the result produced by applying this thresholding operation to the image in Fig. 8. Note that the lower right part of the container is not fully eliminated due to the imperfect estimation of the background. This blob is removed by setting all pixels with x-direction coordinates higher than a dimension threshold beyond which fish cannot be found to black (see Fig. 10). Such an error is not encountered in the experiments for 5, 10 and 15 fish.

2.3 Counting the Fish

After filtering noise and background objects, the remaining blobs correspond to only fish. Now to count the fish the standard area for one fish has to be estimated. A simple way to estimate this is to take the median area of all the blobs. Since most of the fish are likely to appear singly, the median area corresponds well to the area occupied by one single fish. Blobs that have an area lower than 140% of the median area are classified as having only one fish. This percentage is chosen to accommodate fish that are slightly larger than the median area, yet small enough such that it has only a small probability of including intersecting fish. The remaining blobs are considered to have intersecting fish. For these blobs, we divide the blob area by the median area and round the answer to get the number of fish in each of the blobs. A summary of the algorithm is shown in Fig. 11.

Fig. 12 shows the image after blobs with single fish have been eliminated. The white blobs indicate areas detected to contain two fish while the gray blob indicate the area detected to contain three fish. Thus when compared with the original image, our proposed method yields the accurate result.

2.4 Setup

A simple setup for practical implementation is shown in Fig. 13. Fish required to be counted flow from a tank to a smaller container. The container acts as the background while the fish are the foreground objects to be extracted. Therefore the container must be bright in colour to provide an acceptable contrast to the fish. A video camera records images of the fish for a predefined amount of time. These images are processed to produce the count of the fish.

3. EXPERIMENTAL DATA

Experimental data for 5, 10, 15 and 50-fish are shown in Table 1 and Table 2. For the 5 and 10-fish videos, all frames register the correct number. Accuracy is reduced when 15-fish are used. Eight frames register an answer that has an error of one fish. However, this error is smoothened out by averaging when a much larger number of frames register the correct number.

In the 50-fish case, three different settings are used to investigate how a small change in the standard area affects the accuracy of the proposed method. First, a median area of 58 pixels is used. As noted in Section 2.2, in this test case, the lower right portion of the container has to be removed. Table 2 shows that the number of frames that give the incorrect number increased to 30, around four times that of the 15-fish case. However, this still means that about 80% of the frames register the correct number. None of the frames gives an error of more than two fish. Only one frame out of the 30 frames that register the wrong number has an error of two fish. After averaging, the final result is 50.0338 fish. Therefore, when the number of frames registering a count greater than actual count is comparable to the number registering a smaller count than the actual count, the errors cancel each other out partially giving rise to an answer that deviates only slightly from the correct answer.

When the standard area used is 57 pixels, the number of frames that are in error is 33. However, none of them has an error of more than one fish. The final result still rounds to 50. When the standard area is increased to 60 pixels, the number of frames in error increases to 55, but the final result still rounds to close to 50.

4. ANALYSIS OF RESULTS

This method is very useful for counting small number of fish. This is because they are more widely-spaced apart, so they are more likely to appear singly, reducing the number of times we have to process intersecting fish. When fish intersect, the amount of overlap differs, so blob sizes can have a huge range even for the same number of intersecting fish. For example, if one fish overlaps almost completely over another, the blob is likely to be only slightly larger than that of a single fish. The range of areas permitted to be occupied by a singly found fish is larger than that for intersecting fish, so if more of them are singly found, the likelihood that more fish will be categorized in the correct group is higher. Thus, as the number of fish increases, the number of intersecting fish increases and accuracy drops.

Using the median area as the standard area for comparison is found to be good. In all the cases, the correct answer is obtained. The tests show that even if the median area deviates a little from the ideal threshold, the final result obtained is still correct. This holds true even for large numbers of fish, say 50 fish.

The averaging technique allows us to obtain accurate results for any number of fish. When more fish are introduced, although the likelihood of having an error of one fish is higher, the absolute percentage error is reduced since the final error after averaging is not increased significantly. Thus, the proposed method is also suitable for cases involving large numbers of fish if a highly accurate but not perfect result is needed.

There are also shortcomings with the proposed method that need to be explored further. Firstly, there must be a good contrast between the background and the fish. Otherwise, thresholding using luminance values will be difficult. The background estimation method is also not perfect, resulting in the error blob in the lower right corner of the 50-fish image. A large part of each fish but not all of it is captured, affecting the area of the blob, so using a global luminance threshold may not be ideal. Lastly, while the final result remains largely unaffected by the standard area used, the number of frames in error can vary a lot even if it changes by a few pixels. Therefore, selecting a suitable standard area is very important.

5. CONCLUSION

In conclusion, we note that using area information to measure the number of feeder fish is computationally efficient without sacrificing accuracy. The only criterion is to have fish of roughly the same area.

Certain measures can be taken to improve the accuracy of measurements. One possible modification can be to take the reference median area by averaging the median area of test cases involving fewer fish. The accuracy can also be improved by double checking with other parameters of the blobs.

To capture more of the shape of the fish and possibly eliminate some noise, a local luminance threshold using the average luminance values of nearby pixels can be employed in place of a global threshold.

To obtain a more accurate reproduction of the background, a snapshot of the background can be taken before the fish are added. The background is then extracted by averaging the first few frames of the video pertaining to the snapshots before any fish are added.

Finally, it is hoped that the proposed method can serve as a starting point for research into counting objects in specialised situations in which complex counting problems can be converted into simpler tasks.

ACKNOWLEDGEMENT

Special thanks to Chen Xin Ju, Neo Wei Xiang and Yeo Chao Sheng (Nicholas) from Republic Polytechnic for the experimental setup and the capturing of videos of fish.

REFERENCES

[1] Feature Detection (Computer Vision), July 30, 2008, http://en.wikipedia.org/wiki/Feature_detection_(computer_vision)

[2]Morais, E.F., Campos, M.F.M., Pádua, F.L.C. & Carceroni, R.L., "Particle Filter-based Predictive Tracking for Robust Fish Counting," Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing, 2005. SIBGRAPI 2005, pp.367-374, October 2005

[3]Spampinato, C., Chen-Burger, J., Nadarajan, G. & Fisher, R., "Detecting , tracking and counting fish in low quality unconstrained underwater videos," Proceedings of 3rd International Conference on Computer Vision Theory and Applications, 2008. VISAPP 08, pp.514-520, January 2008

[4]Standford University. (2005). Assignment #2 : Image Processing and Object Tracking, July 30, 2008, http://www.stanford.edu/ class/cs377s/assignments/assignment2.pdf

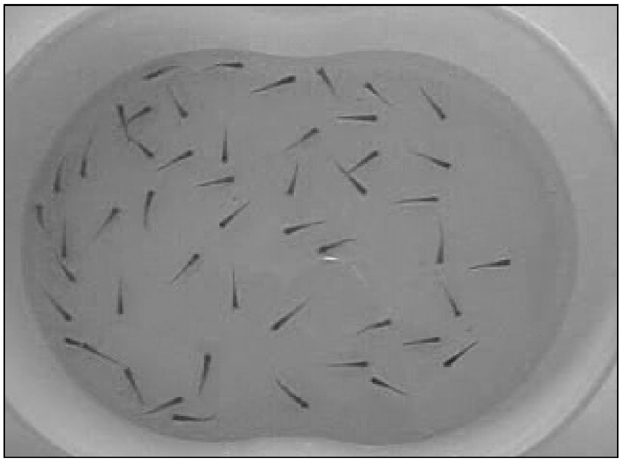
[5]The Mathworks, Contrast Enhancement Techniques, July 30, 2008, http://www.mathworks.com/products/image/demos.html?file=/products/ demos/shipping/images/ipexcontrast.html

[6]The Mathworks, Image Processing Toolbox, July 30, 2008, http://www.mathworks.com/access/helpdesk/help/toolbox/images

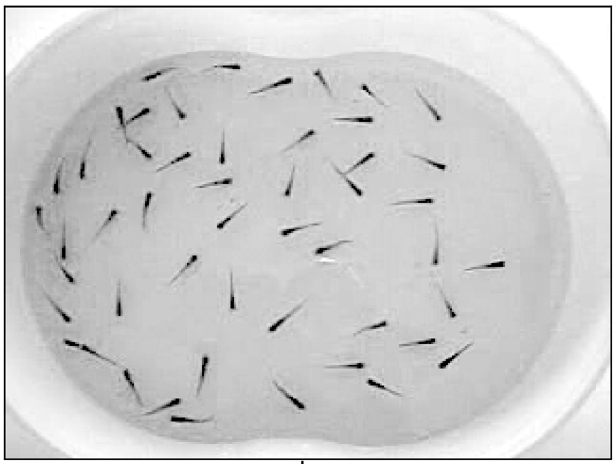
[7]Adlan, H.H.A., Ramli, A.R., Mansor, S., Taib, T., "Performance analysis of edge detection methods for macroscopic images," KKU Science Journal, vol.31, no.3, pp. 141-147, July-September 2003

[8] The Mathworks, Acquiring And Analyzing Image Sequences In MATLAB, July 30, 2008, http://www.mathworks.com/company/newsletters/news_notes/win03/tracking.html

[9]The Mathworks, Identifying Round Objects, July 30, 2008, http://www.mathworks.com/products/image/demos.html?file=/products/ demos/shipping/images/ipexroundness.html



a



b

Fig. 1. A 50-fish frame a) after conversion to grayscale b) after contrast adjustment

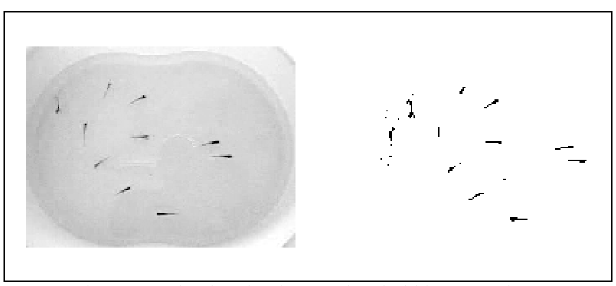


Fig. 2. Result when luminance threshold is set too low.

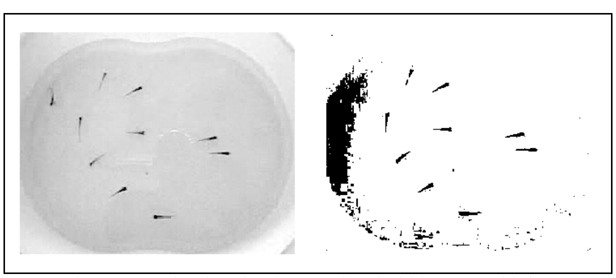
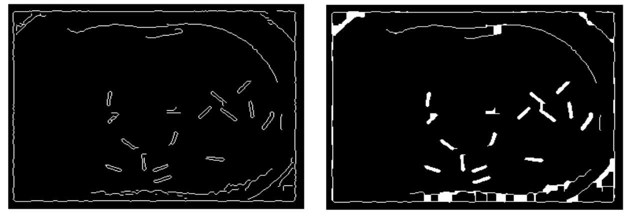


Fig. 3. Result when luminance threshold is set too high.



a b

Fig 4. A 15-fish frame a) after edge detection b) after morphological closing.

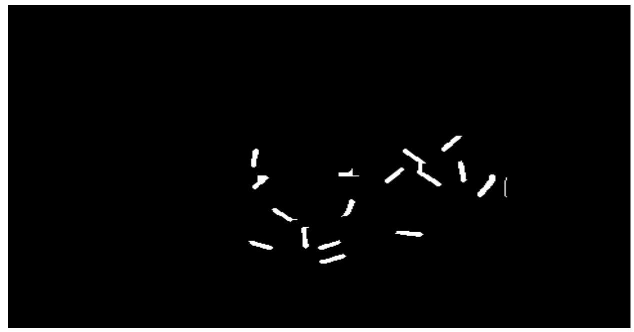


Fig. 5. A 15-fish frame after removing background objects.

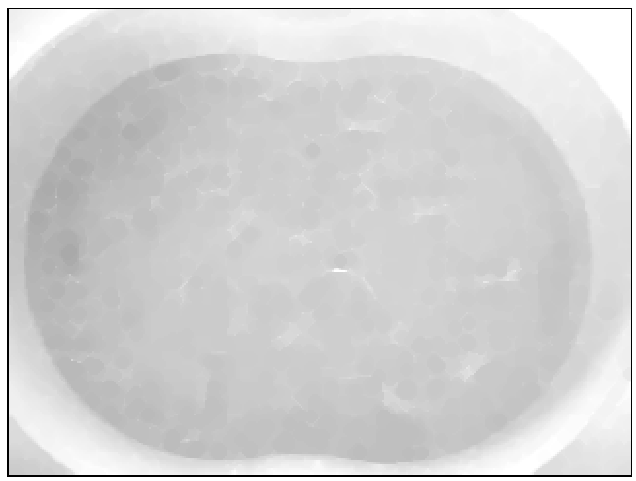


Fig 6. Estimated background of image in Fig. 1b.

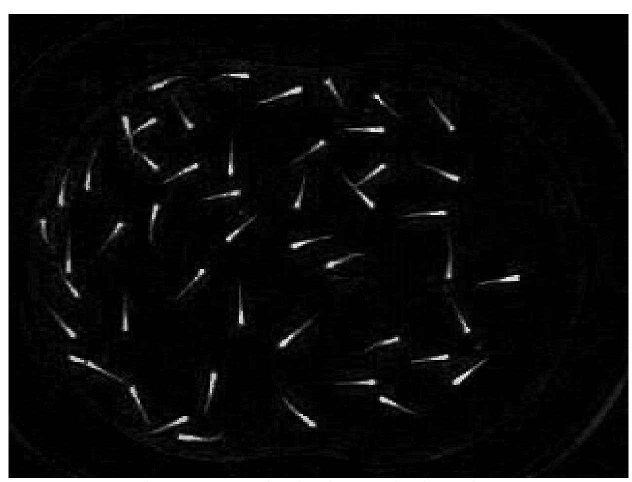


Fig. 7. Image after background subtraction.

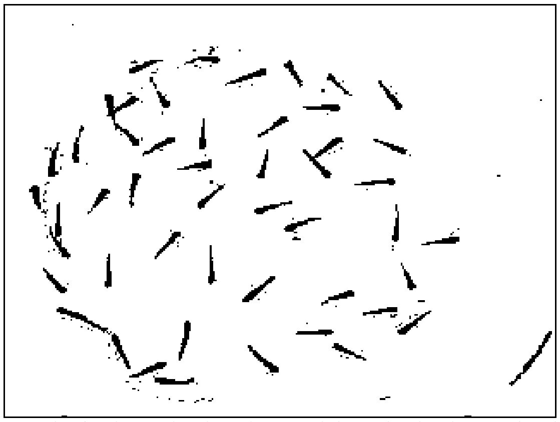


Fig 8. Result after image in Fig. 7 is passed through a luminance threshold.

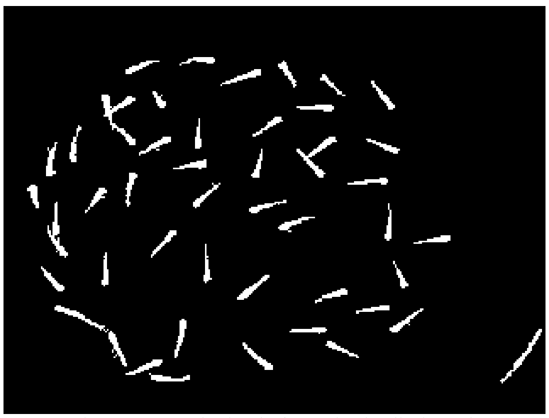


Fig. 9. Image obtained after noise is eliminated.

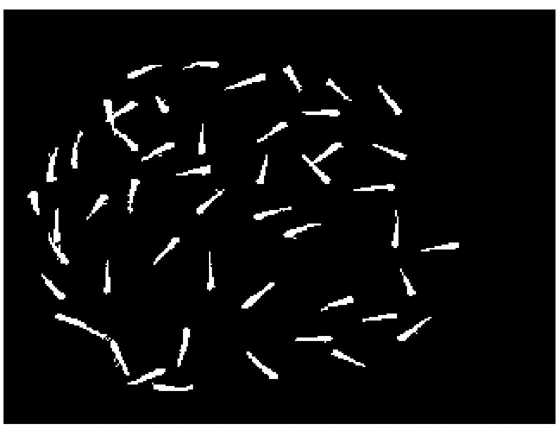


Fig. 10. Image obtained after the container is totally eliminated from image in Fig. 9.

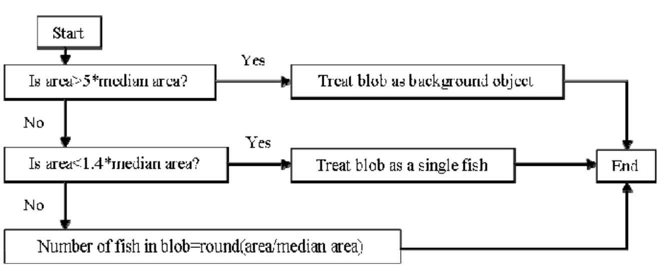


Fig. 11. Algorithm to calculate number of fish in blob

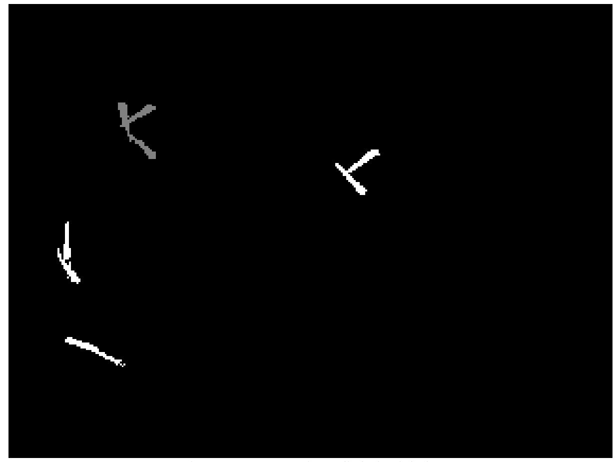


Fig. 12. Counting fish in blobs

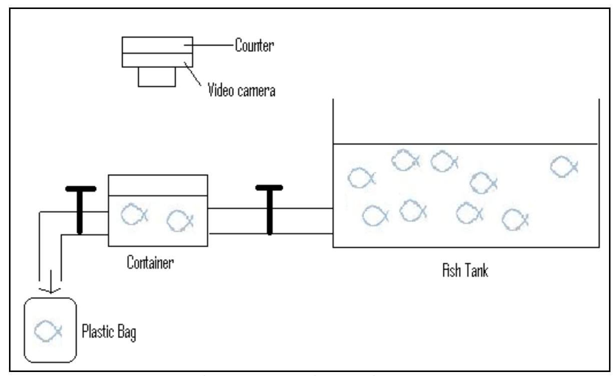


Fig. 13. Experimental Setup

Table 1. Results after analysing all frames

	5 fish	10 fish	15 fish	50 fish
Median Area	56	58.5	56	58
Mean Number of Detected Fish	5	10	15.028	50.0338
Percentage Error(Error/1)	0%	0%	2.80%	3.38%
Percentage Error(Error/Actual Number)	0%	0%	0.19%	0.07%

Table 2. Breakdown of errors in frames

	Number of Frames with absolute error -1	0	1	2	No. of frames in error	Total no. of frames	% of frames in error	Mean number
15-fish (Median Area)	2	135	6	0	8	143	5.59	15.0280
50-fish (Median Area)	13	118	16	1	30	148	20.27	50.0338
50-fish (57 pixels)	24	115	9	0	33	148	22.30	49.8986
50 fish (60 pixels)	53	92	3	0	56	148	37.84	49.6621