Detection and recognition of uneaten fish food pellets in aquaculture using image processing

Conference Paper *in* Proceedings of SPIE - The International Society for Optical Engineering · October 2014

DOI: 10.1117/12.2179138

Detection and Recognition of Uneaten Fish Food Pellets in Aquaculture using Image Processing

Huanyu Liu[1] , Lihong Xu[1,*], Dawei Li[1]
[1]College of Electronics and Information Engineering, Tongji University

ABSTRACT

The waste of fish food has always been a serious problem in aquaculture. On one hand, the leftover fish food spawns a big waste in the aquaculture industry because fish food accounts for a large proportion of the investment. On the other hand, the left over fish food may pollute the water and make fishes sick. In general, the reason for fish food waste is that there is no feedback about the consumption of delivered fish food after feeding. So it is extremely difficult for fish farmers to determine the amount of feedstuff that should be delivered each time and the feeding intervals. In this paper, we propose an effective method using image processing techniques to solve this problem. During feeding events, we use an underwater camera with supplementary LED lights to obtain images of uneaten fish food pellets on the tank bottom. An algorithm is then developed to figure out the number of left pellets using adaptive Otsu thresholding and a linear-time component labeling algorithm. This proposed algorithm proves to be effective in handling the non-uniform lighting and very accurate number of pellets are counted in experiments.

Keywords: Fish food pellets recognition, image processing, Otsu algorithm, connected area algorithm.

1. INTRODUCTION

Optimizing input-output ratio has always been the ultimate task in aquaculture. Feed conversion ratio (FCR) [1] is an index that is commonly used to represent the input-output ratio in aquaculture. It is defined as the ratio of the weight of the consumed feedstuff (kg) over the fish mass gain (kg). Typically, the waste of fish food pellets is the key factor affecting FCR. Therefore, how to eliminate the number of uneaten food pellets has been a frequently discussed topic in aquaculture.

The significance and challenge on detecting leftover fish food have made it an area of great interest for researcher. Foster [2] uses an underwater camera heading straight down in a sea cage to observe and count the fish food pellets falling down during a feeding event. Ang and Petrell [1] design a method to reduce the waste rate of fish food. They also present some experimental results about indices such as FCR, dispensation rate and mortality. Due to the lack of illumination underwater, the camera must be compensated by external light resources. However, the light source usually produces non-uniform lighting condition on the surface being illuminated. This phenomenon poses a great challenge to existing underwater image processing techniques. Moreover, when the water becomes turbid, those methods for fish food counting would fail because the discernibility is heavily degraded.

In section 2, we elaborate the experimental platform. Then we propose an adaptive method for segmenting fish food pellets from unevenly illuminated images and the proposed method is compared with many other algorithms for image segmentation. Section 4 gives some fish food pellets detection and counting results in different situations. Finally, section 5 concludes the paper and discusses future directions of our research.

2. EXPERIMENTAL PLATFORM AND SYSTEM OVERVIEW

Our experimental platform contains three parts. The first part is a large glass-made water tank covered by black shades. The size of the tank is 1.2m×0.6m×0.5m. When we are capturing underwater video, a paperboard is placed on top of the tank to block indoor light. The second part of our platform is an underwater imaging system containing an underwater camera, a white plate and a stand for fixing the camera. Plate is placed right beneath the stand, and the camera is placed vertically above the plate. Camera is fixed in the top of the stand. The distance between the camera and the plate can be adjusted between the range of 30cm and 60cm, therefore the height of the camera could be adjusted via the stand according to variable water turbidity. The camera consists of a SHARP 1/4 CCD camera (420 lines resolution)

*Lihong Xu is the corresponding author. Email: dahuan1016@126.com, xulhk@163.com.

Sixth International Conference on Graphic and Image Processing (ICGIP 2014),
edited by Yulin Wang, Xudong Jiang, David Zhang, Proc. of SPIE Vol. 9443, 94430G

with underwater housing (ABS) which tolerate water pressure as high as 5 atmospheres, and a compensating LED light source. The data line and the power line of camera are both 20 meters long. The third part of our platform is an upper computer with a video signal converter. The camera's output data line has to be connected to the signal converter to transfer the analog video signal to digital video signal. The upper computer then receives and stores the video signal as AVI files in a mass storage hard disk. All videos' resolution is fixed at 640×480.

3. ADAPTIVE OTSU AND SEGMENTATION

All of the underwater images have unevenly illuminated conditions. So the classical segmentation algorithms such as Otsu [3], two-dimensional Otsu [4], entropic thresholding [5,6], two-dimensional maximum entropy segmentation [7] and Yen thresholding [8] fall short on producing desirable thresholding results.

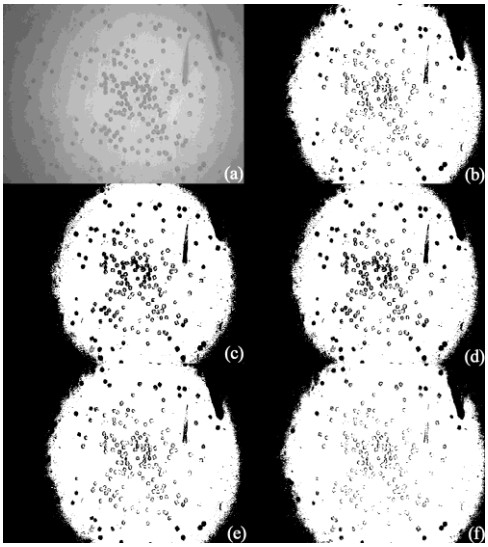

Figure 1. (a): the original image. (b)-(f): result imagesof Yen, Otsu, 2-D OTSU, entropic thresholding, 2-D maximum entropy segmentation respectively.

Fig.1 is the results of some typical thresholding algorithms. We can see that all of the result images have the same black margin and only the bright center is segmented successfully. This phenomenon is caused by the unevenly illumination that leads to low grey value in the margin. In short, classical methods using a global threshold are not able to deal with unevenly illuminated images.

In experiment, we got grey-level histogram of Fig.1(a). It's shape is single-peaked in general. The unimodal feature is the result of illumination attenuation from the center to margin because the distance from the margin to the camera is longer than the center. Edge pixels have a considerable drop in grey value compared to the central pixels, which leads to the unimodal shape in the histogram.

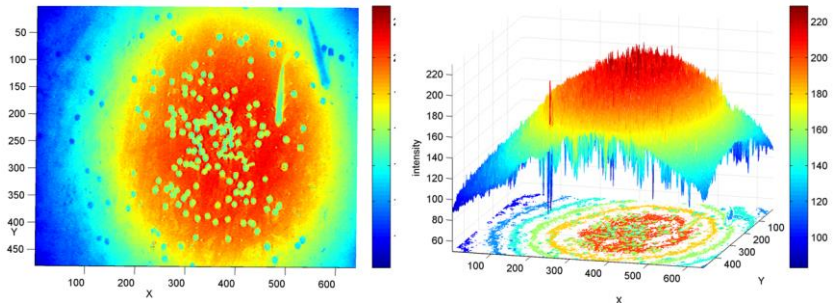

Figure 2. The 3D grey intensity values of Fig.1(a). The left figure is the vertical view, and the right figure is the view from the side. Please be noted that in the right figure, the horizontal plane stands for the image plane, and the height stands for the grey intensity.

To solve this problem, many robust segmentation algorithms are proposed. The most common methods consist of local threshold segmentation, adaptive threshold [9. Bradski Gary and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc., (2008).138], quick adaptive thresholding algorithm [10], adaptive thresholding using integral image [11] and so on. All these algorithms have their own merits and drawbacks. In this paper we propose an improved adaptive thresholding algorithm based on Otsu.

Otsu is sensitive to foreground object size and noise. It is best suitable for images with double-peaked histogram. In some cases, the threshold obtained through Otsu may not be the most proper threshold, but Otsu is the fastest algorithm among the classical algorithms. In our experiment, the grey-value histogram is roughly single-peaked. So Otsu is not the right technique to use. But in a small local window of the whole image, the illumination distribution can be regarded as uniform.

We get many histograms of different small windows in the image and find that the histograms of background windows are generally unimodal, while histograms of windows containing foreground objects are bimodal. Thus Otsu could be used within the small window to obtain the threshold of each local part. Then the global image could be segmented according to every local threshold. This is the main idea of local segmentation. It is robust against uneven lighting conditions. However, because of the discontinuity of adjacent windows' thresholds, the outcome obtained by local thresholding has a serious block phenomenon (see Fig.4(b)), which is intolerable.

Therefore, we introduce the concept of mask instead of local window. In principle, a mask is a moving window. It can be considered as an image matte. Before processing, the image border must be extended; otherwise the edge pixels can't be processed properly. All the added lines are replicas of four outmost borders. When processing, the mask begins with the first original pixel in the top left corner and moves pixel by pixel. Every time after a move, all pixels within the mask are processed with Otsu algorithm and a resultant threshold can be obtained. Thereby for each pixel, there is a corresponding threshold which is determined by the pixel's neighborhood information within the mask. In our experimental environment, the fish food is darker than the background; therefore the pixels that have higher intensity values are more likely to be the background in a local mask region. The segmentation result is given by a binary image, in which the white pixels represent fish food, while the black pixels stand for the background. We compare the center pixel's grey value to its threshold, if the value is greater than threshold, then the pixel is set to 0 (black) in the resultant binary image; otherwise, the pixels are set to white. The selection of the mask size is based on the size of foreground objects.

Processing with only mask is not enough. Because there are also many masks containing only background pixels. These masks' local grey-value histograms are unimodal and hence they are not suitable for applying Otsu. Besides, the closer the mask approaches the image edges, the more unimodality we will observe. Obviously, the tendency of sharper single peak is caused by the dim light in the margin. In the margin area of an underwater image, food pellets' values are close to that of the background. Otsu may not be able to distinguish the slight difference between food pellets and background. To deal with this problem, we introduce a Gaussian model in our algorithm as a penalty function to help deriving a better threshold.

In a continuous video, the background scene is relatively static. According to the work of Wren et al. [12], we consider that the background illumination distribution is similar to Gaussian distribution. But only knowing the pattern of background light distribution helps little in determining proper threshold. Light condition information must be transformed into a sort of light-weight-indicator. Thus penalty function which is related to light condition is an applicable way here. Grey value difference is a reflection of aberration. We have verified that the distribution of difference within mask obeys Gaussian distribution approximately. Thus the difference of every mask can be the independent variable of penalty function and determine the corresponding penalty value in masks. Similarly, a Gaussian function is introduced as the penalty function here, which is written as:

$$f(x) = a \cdot \exp\left[-\frac{x^2}{c^2}\right], \tag{1}$$

Where a and c are parameters to be determined, and x is the difference between the highest grey value and the lowest value in a mask. The penalty $f(x)$ is computed within each mask, and the corresponding threshold is recomputed by subtracting the penalty from the original threshold obtained from Otsu algorithm. Consequently, how to use image information to get the proper value of a and c is important. To compute parameters a and c, a least two data points of

function (1) should be obtained. As we have observed, the intensity difference in the bright center mask is much larger than that of the margin mask. So we search one data point in the bright center and another data point in the margin area.

For an image I(x, y), we traverse it to find the pixel that has the largest grey value. Then in its neighborhood mask, we record the mean value m_1 and the difference x_1. As to the marginal area, we select the mask with the lowest difference among the four masks at the four corner of the image, and compute its mean m_2 and difference x_2. On the basis of numerous experiments, we have found out that the penalties with respect to x_1 and x_2 can be approximated by:

$$f_1(x_1) = \left[3 - E\left(\frac{x_1 + 20}{40}\right)\right] \cdot 2^{E\left(\frac{|m_1 - 140|}{40}\right)}, \tag{2}$$

$$f_2(x_2) = E\left(\frac{x_2}{20}\right) \cdot 2^{4 - E\left(\frac{|m_1 - 140| + |m_2 - 100|}{40}\right)}. \tag{3}$$

In (2) and (3), the function $E(\cdot)$ represents rounding down. With these equations, two points of the penalty function can be calculated. Therefore, parameters a and c can be obtained. The ultimate threshold in every mask is decreased by subtracting the corresponding penalty value which is calculated from (1). The modified threshold is adaptive, because the received penalty is dependent on the lighting condition in a neighborhood.

Fig.3 is results of the proposed adaptive thresholding algorithm on some underwater images with different turbidity. The mask size is fixed at 49×49. In Fig.3(a), the two differences used to compute the penalty function (1) are 81 and 22, and the two means are 147 and 100 in the bright center and the marginal area respectively. The penalty function of Fig.3(a) is:

$$f(x) = 20 \cdot \exp\left(-\frac{x^2}{2100}\right) \tag{4}$$

Thus resultant binary image of Fig.3(a) is shown by Fig.3(e). We have also tested the proposed thresholding algorithm on several images (Fig.3(b), (c), (d)) with diverse water turbidity. The results are given in Fig.3(f), (g), and (h) respectively. It should be noted that the equation (2) and (3) are only empirical formulas. The penalty function derived from these equations may not be the best choice. But they surely can improve the segmentation.

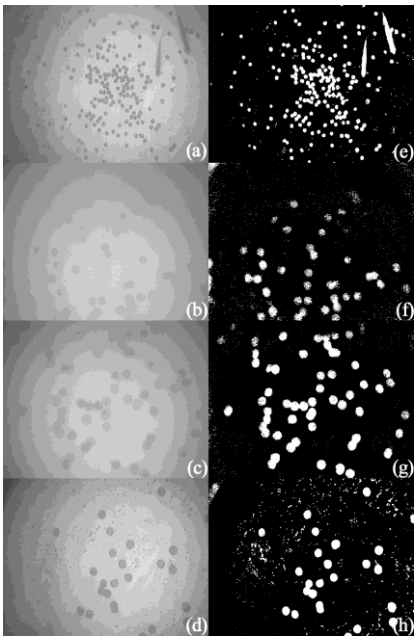

Figure 3. Binary images obtained from the proposed thresholding algorithm. (a)-(d) are original test images, and (e)-(h) are segmentation results.

Our adaptive Otsu method uses both the local neighborhood information and a penalty function to determine threshold for segmentation. Experiments show that this method can effectively segment fish food pellets from unevenly illuminated images. Fig.4 exhibits a comparison of some classical segmentation algorithms on an image with non-uniform illumination. Fig.4(a) shows a ground truth image that is manually labeled in a pixel-by-pixel way; this ground truth image represents the ideal segmentation result, and we expect results of the algorithms to approximate it as close as possible. Fig.4(b) is the result of a local Otsu algorithm. The window size is set to 49×49. This method suffers from a severe block phenomenon, in which large areas of false segmentation can be observed along many window edges. Fig.4(c) is the result of the adaptive thresholding using integral image [11], and it is better than Fig.4(b) but still has failed to detect some pellets. Image Fig.4(d) and (e) are results of the proposed mask thresholding method with two constant penalties, respectively. The mask size is 49×49 and the penalty is 12 in Fig.4(d), while the mask is 25×25 and 10 for Fig.4(e). Fig.4(f) is the result of the proposed thresholding method with adaptive penalty function, and it has the best segmentation result among all compared methods. Table 1 lists the corresponding segmentation indicators—TPRs and FPRs of the five images in Fig.4. TPR is defined as the quotient of the number of correctly segmented foreground pixels in resultant image generated by algorithms over the number of the whole foreground pixels in the ground truth image [13,14]. Likewise, FPR is defined as the quotient of mistakenly segmented foreground pixels in result image over the number of the background pixels in the ground truth image. It is obvious that result image obtained by our algorithm has a relatively high TPR (0.9344, not the highest) and the lowest FPR (0.0157). Though Fig.4(b) has a slightly higher TPR (0.9624), its FPR (0.3174) is more than twenty times of FPR in Fig.4(f). In general, the result of the proposed method (Fig.4(f)) has the best segmentation performance across all methods compared.

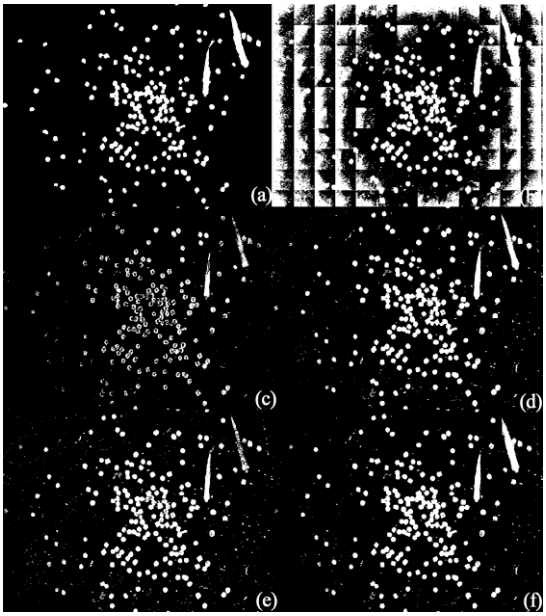

Figure 4. The comparative experimental results of several classical algorithms including ours.(a) is the ground truth image corresponding to Fig.1(a). (b) is the result of a local Otsu algorithm based on windows. (c) is the result of an adaptive thresholding algorithm based on integral image. (d) and (e) are the results of the proposed mask thresholding method with two constant penalties, and (f) is the result of the proposed thresholding method with adaptive penalty function. It is obvious that the proposed method has the best detection performance among all the methods compared.

Table. 1. The TPRs and FPRs of result images in Fig.4

	TPR	FPR
Fig.4(b)	0.9624	0.3174
Fig.4(c)	0.5886	0.0045
Fig.4(d)	0.9292	0.0178
Fig.4(e)	0.8790	0.0181
Fig.4(f)	0.9344	0.0157

All in all, the adaptive Otsu's thresholding algorithm is robust against the non-uniform lighting condition, and we do not have to pre-determine a lot of parameters for the algorithm. From Fig.3 we can see this algorithm is also very suitable for turbid water environments, which makes it very compelling in practical implementations.

4. FISH FOOD PELLETS COUNTING

After obtaining the binary image, we can figure out the number of uneaten fish food pellets. A fast connected component labeling method based on contour tracing is used to count the number of white foreground objects [15].

In the counting experiment, we use two types of fish food. The first type's diameter is 4 millimeters and the second's diameter is 8 millimeters. For these two types of fish food pellets' images, we set two different limit values of pixels. Only those connected areas whose pixel number is more than the limit value will be considered a fish food pellet. The smaller pellet's limit value is set to 25, while the bigger pellet's limit number is 80. Fig.5 shows the counting results for three test images. The counted numbers of fish food are 42, 23, 34, labeled in Fig.5 (d) to (f) respectively. From Fig.5 we can see that almost all uneaten fish food pellets can be recognized. In turbid water condition which is difficult even for people to recognize, the outcomes are also desirable. In all these three experiments, the automatic counting algorithm can recognize nearly all of the uneaten fish food pellets except some illegible ones in the dark corner. After getting the number of uneaten fish food pellets, the upper computer can regulate the controller of a fish food feeding machine to provide precise fish food quantity in the next feeding process.

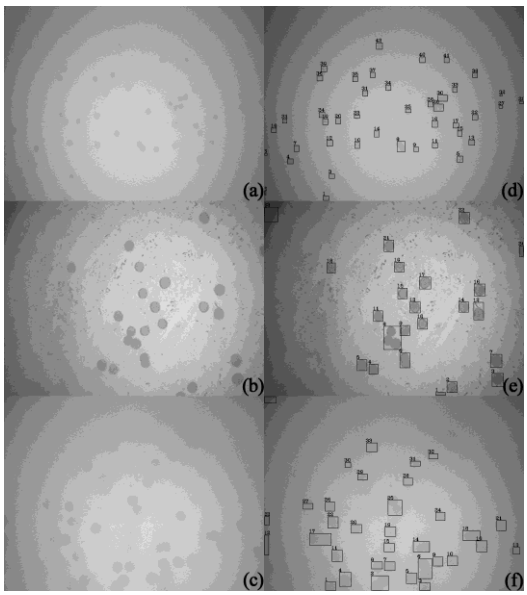

Figure 5. Countingfish food pellets from three underwater images. The left column is three test images, and the right column shows the respective counting results.

5. CONCLUSION

Aiming to detect and count the uneaten fish food pellets via underwater images, we propose an adaptive Otsu method that is effective under uneven illumination conditions. Based on the binary images obtained from the proposed adaptive thresholding algorithm, the uneaten fish food pellets number can be figured out using a linear-time labeling algorithm. Experiments prove that the average counting accuracy error is less than 8%, which is accurate to be applied for practical implementations.

However, there still exist many challenging tasks on underwater image processing. In future work, we plan to design an algorithm that can classify uneaten fish food pellets and other objects such as fishes and underwater plants.

ACKNOWLEDGEMENTS

This work was supported in part by the National High-Tech R&D Program of China under Grant 2013AA102305, the National Natural Science Foundation of China under Grant 61174090 and 61374094, China Postdoctoral Science Foundation under Grant 2013M540385, and in part by the U.S. National Science Foundation's BEACON Center for the Study of Evolution in Action, under cooperative agreement DBI-0939454.

REFERENCES

[1] Ang, K. P., and R. J. Petrell, "Control of feed dispensation in seacages using underwater video monitoring: effects on growth and food conversion, "Aquacultural Engineering 16.1, 45-62, (1997).

[2] Foster, M., Petrell, R., Ito, M. R., and Ward, R., "Detection and counting of uneaten food pellets in a sea cage using image analysis, " Aquacultural Engineering 14.3, 251-269, (1995).

[3] Otsu, Nobuyuki, "A threshold selection method from gray-level histograms, "Automatica 11.285-296, 23-27, (1975).

[4] Jianzhuang, Liu, Li Wenqing, and Tian Yupeng, "Automatic thresholding of gray-level pictures using two-dimension Otsu method, " Circuits and Systems, 1991. Conference Proc, China., 1991 International Conference on. IEEE, 325-327, (1991).

[5] Pun, Thierry, "Entropic thresholding, a new approach, " Computer Graphics and Image Processing 16.3, 210-239, (1981).

[6] Kapur, J. N., Prasanna K. Sahoo, and Andrew KC Wong, "A new method for gray-level picture thresholding using the entropy of the histogram, " Computer vision, graphics, and image processing 29.3, 273-285, (1985).

[7] Abutaleb, A., and A. Eloteifi, "Automatic Thresholding of Gray-Level Pictures Using 2-D Entropy, " 31st Annual Technical Symposium. International Society for Optics and Photonics, 29-35, (1988).

[8] Yen, Jui-Cheng, Fu-Juay Chang, and Shyang Chang, "A new criterion for automatic multilevel thresholding, " Image Processing, IEEE Trans. on 4.3, 370-378, (1995).

[9] Bradski Gary and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, Inc., (2008).

[10] Wellner, Pierre D, "Adaptive thresholding for the DigitalDesk, " Xerox, EPC1993-110, (1993).

[11] Bradley, Derek, and Gerhard Roth, "Adaptive thresholding using the integral image, " Journal of graphics, gpu, and game tools 12.2, 13-21, (2007).

[12] Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P., "Pfinder: Real-time tracking of the human body, " Pattern Analysis and Machine Intelligence, IEEE Trans. on19.7, 780-785, (1997).

[13] Li Dawei, Lihong Xu, and Erik Goodman, "Online background learning for illumination-robust foreground detection, " Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on. IEEE, 1093-1100, (2010).

[14] Li Dawei, Lihong Xu, and Erik D. Goodman, "Illumination-Robust Foreground Detection in a Video Surveillance System, " Circuits and Systems for Video Technology, IEEE Trans. on 23.10, 1637-1650, (2013).

[15] Chang Fu, Chun-Jen Chen, and Chi-Jen Lu, "A linear-time component-labeling algorithm using contour tracing technique, " computer vision and image understanding 93.2, 206-220, (2004).