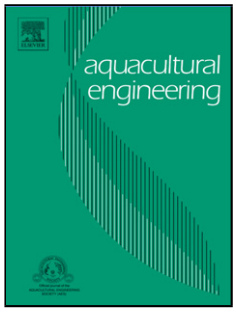

Title: Detection of Uneaten Fish Food Pellets in Underwater Images for Aquaculture

Authors: Dawei Li, Lihong Xu, Huanyu Liu

PII: S0144-8609(16)30224-2
DOI: http://dx.doi.org/doi:10.1016/j.aquaeng.2017.05.001
Reference: AQUE 1901

To appear in: *Aquacultural Engineering*

Received date: 15-12-2016
Revised date: 14-4-2017
Accepted date: 1-5-2017

Please cite this article as: Li, Dawei, Xu, Lihong, Liu, Huanyu, Detection of Uneaten Fish Food Pellets in Underwater Images for Aquaculture.Aquacultural Engineering http://dx.doi.org/10.1016/j.aquaeng.2017.05.001

Regular Papers:

Detection of Uneaten Fish Food Pellets in Underwater Images for Aquaculture

Dawei Li [1], Lihong Xu [2,*] and Huanyu Liu [3]

[1] College of Information Science and Technology, Donghua University, Shanghai, China 201620; daweili@dhu.edu.cn
[2] College of Electronics and Information Engineering, Tongji University, Shanghai, China 201804; xulihong@tongji.edu.cn
[3] Huawei Technologies Co. Ltd., Shanghai, China 200120; liuhuanyu2@huawei.com
* Correspondence: xulihong@tongji.edu.cn; Tel.: +86-1-363-636-2684

Abstract: The waste of fish food has always been a serious problem in aquaculture. On one hand, the leftover fish food spawns a big economic loss in the industry because feedstuff accounts for a large proportion of the investment. On the other hand, the left over fish food may pollute the water and worsen the living environment of aquatic products. In this paper, we develop an adaptive thresholding method for detecting uneaten fish food in underwater images. To deal with non-uniform illumination in underwater environments, we focus on analyzing the local histogram of intensities in a mask for each pixel. The Expectation-maximization-guided Gaussian mixture is used to fit the histogram to figure out its type, and then an adaptive threshold is computed accordingly. At last the central pixel of the mask is compared with the threshold to generate the binary detection result. Experimental results show that the proposed method obtains desirable detection of leftover fish food in many underwater environments with different water turbidity levels and with different extent of unevenness of illumination. In the four test underwater environments, the lowest True Positive Rate of the proposed method is higher than 80%, and the highest rate reaches 95.9%. The False Positive Rates of the proposed method are no higher than 2.7%.

Keywords: fish food detection; aquaculture; image segmentation; Gaussian Mixture Model; EM algorithm; non-uniform illumination

1. Introduction

With the growing demand of aquatic products, aquaculture today has been punching above its weight in human food production. Due to the lack of feedback information from actual consumption, waste of feedstuff has been a common and disturbing problem for workers in traditional aquaculture. On one hand, the leftover fish food spawns a big economic loss in the aquaculture industry because feedstuff accounts for a large proportion of the investment. Feed makes up as much as 82% and 70% of total costs for tilapia farming in ponds and cages respectively, more than 50% for milk fish, see-bass cage culture as well as for shrimp pond culture [1]. The energy use of feed production also takes up the majority of the total energy use in Atlantic salmon production, currently making up about 50% of total cost [2]. Thus, improving feed efficiency in industrial aquaculture systems is already a priority [3]. On the other hand, the leftover fish food may lead to eutrophication of water and worsen the aquatic environment. The most common sign of an environmental problem caused by nutrient discharges in aquaculture is the accumulation of organic

sediments and changes in the benthic fauna [4]. Therefore, how to decrease the number of uneaten food pellets is a frequently discussed topic in aquaculture.

Nowadays, the rapid development of image processing technology and the declining price of the underwater video systems have made the detection of fish food waste more readily accessible than ever before. The video system for aquaculture routinely applies underwater cameras to monitor the underwater environment and analyze the quantity of uneaten food pellets via a computing unit. Then the result is treated as a feedback, and sent to an automatic feeding device that uses the feedback as control input. By processing and analyzing the feedback data, the controller automatically determines whether to increase or decrease the quantity of fish food pellets at next feed.

Underwater fish food detection belongs to a class of underwater image processing tasks that have long been an area of interest. Underwater visual object detection and segmentation face challenges that are shared in some other image-based applications such as visual object recognition, image enhancement, optical flow estimation, depth estimation, and so on. The biggest challenge in underwater image processing originates from light, because water is a medium that strongly scatters and absorbs light, causing degradation of image contrast and deteriorating the quality of captured images. Model-based methods are effective in solving underwater image degradation problems; hence they are commonly used for enhancement of views. The work by Schechner and Karpel [5] captured two color images of the undersea scene through a polarizing filter to enhance visibility. Although the enhancement is significant, the method is best compatible with horizontal photography rather than acquisitions in the downwards direction. Drews Jr. *et al.* [6] applied the Dark Channel Prior (DCP) to underwater image recovery and depth estimation. The method can achieve accurate depth map of an underwater scene from color images under bright and uniform illumination. Sheinin and Schechner proposed a Next Best Underwater View (NBUV) method [7] to adjust the relative positions of a light source and an underwater camera for generating a better view that suppresses backscatter and shadows. NBUV can help to produce underwater images that have uniform illumination condition, which may be beneficial to underwater object segmentation tasks. Model-based methods usually process color underwater images rather than grayscale images because ample visual information can generate accurate models.

In underwater environments, infrared light sources and infrared cameras are commonly adopted for fewer disturbances on aquatic creatures' activities, so that the vision system only captures grayscale images. Different from model-based methods, image-based methods have fewer constraints on the underwater environment—e.g., they can work on grayscale images; hence they are sometimes more reliable in real applications for object segmentation than model-based methods. Researchers have been focusing on developing image-based techniques to separate interested foreground from underwater environments. Hsiao *et al.* [8] adopted Gaussian Mixture Model (GMM) to identify fish from ocean background. Shen *et al.* [9] developed a biological hierarchical model inspired by frog visual mechanism to detect underwater moving objects. Walther *et al.* [10] combined the background subtraction and the Itti's model to detect and track underwater objects. Chen [11] proposed a machine-learning based method to detect objects from underwater images. Fei *et al.* [12] used The Expectation-maximization (EM) algorithm to detect objects from underwater Synthetic Aperture Sonar (SAS) images. Evans [13] modeled the image as a bivariate Gaussian mixture model (GMM) and an EM approach was then used to estimate the mixture parameters. Finally, the mixtures were used to detect southern Bluefin tunas from underwater images. Singh *et al.* [14] used method of contrast limited adaptive histogram equalization to enhance subsea images, and then applied the Fuzzy C-Means clustering for segmentation. Schoening *et al.* [15] applied the ES4C algorithm to fully automated segmentation for benthic images of poly-metallic nodules, and achieved better results comparing to the Otsu's method.

Pelleted fish food detection has been drawing attentions from both computer vision and aquaculture engineering communities for quite a time. The existing applications mainly focus on cage farming, and most of them can be classified as image-based methods. Foster [16] used an underwater camera facing straight down in a sea cage to observe and count the fish food pellets that

fall down during a feeding event. Ang and Petrell [17] tried to reduce the waste rate of fish food in camera-monitored cages. They also presented some experimental results based on indices such as Feed Conversion Ratio (FCR), dispensation rate and mortality. Alver *et al.* [18] combined a 3-d diffusion model and a new feed input module to simulate the food pellets distribution in salmon cages. Skoien *et al.* [19] presented a study of the fish food pellet settling rate and diffusion process through underwater experiments, and they hoped it may improve estimates of fish behavior, growth and feed loss. Skoien *et al.* [20] also proposed a design based on an integration of Kalman filter and the cost minimization for quantifying feed density in sea cages. Parsonage [21] designed an image processing framework to detect and recognize fish feed pellets with a camera viewing upward for Atlantic salmon cage farming.

Many challenging issues still exist in segmenting biomass or feedstuff from underwater images. Objects on the seabed such as leftover fish food pellets usually remain still in field of view for quite a long time before dissolution, making the popular background–foreground detection methods [22-24] infeasible because still objects will be combined into background during the learning phase. For aquaculture monitoring and underwater exploration, artificial lights are a common approach to improve underwater visibility. However, the scene irradiance quickly decreases with the growing distance to the light source due to backscattering of water medium. The phenomenon directly leads to the non-uniform illumination problems in many underwater images, which causes many image segmentation algorithms to fail. To address these issues, we propose an adaptive histogram-based thresholding method for underwater fish food detection on a single image. This image-based method first generates an intensity histogram of a neighborhood area for each pixel; then a Gaussian mixture model (GMM) [25] is used to fit each histogram. The expectation-maximization (EM) algorithm [26] is applied to learn the parameters of GMM, and if the resultant mixture contains well-separated Gaussian distributions, a modified Otsu's algorithm is used to compute the threshold value that separates the histogram into two parts. At last, each pixel is classified into either background or foreground by comparing with the threshold. Although the EM-guided GMM has been widely applied in many types of image processing tasks, it is for the first time being used to classify leftover feedstuff from underwater environment by automatically recognizing the histogram modality of each single pixel. The proposed method is tested on a series of underwater images that have different conditions and results have shown that our method is superior to several other methods.

2. Material and methods

2.1. Platform and Data Aquisition

The underwater imaging platform was set up at a fresh water aquaculture site located at Qingpu District of Shanghai, China. The aquaculture site has two pools with average water depth of 1.8 meter. The structure of the underwater system is shown in Fig. 1(a) and 1(b). A customized underwater camera (640×480 resolution) with integrated LED lights is fixed to a rod that attached to a thin holder with a height of 1.9 meter. Right below the camera there sits a round plate (effective diameter of 35 cm) which is fixed to the holder. The camera and the plate are shown in Fig. 1(a), and the distance from the camera to the plate is adjustable by changing the connection point of the rod and the top beam of the holder. The distance between the camera and the plate can vary between 20 cm and 50cm with a step of 3 cm. In experiments, the actual distance is set to 20cm, 29cm, and 41cm. The image of the complete system is shown in Fig. 1(b). The underwater system was placed near the daily feeding spot of the pool as the way shown in Fig. 1(c). Fig. 1(d) shows a little boat that can distribute fish food pellets, and a worker can use this boat to relocate the underwater imaging system. A video surveillance server was connected to the camera, and underwater videos were stored on the server. The interface of the server is given by Fig. 1(e). After a feeding event, uneaten fish food pellets will remain on the plate.

2.2. Problem Statement

Histogram-based methods have always been a fundamental tool for image segmentation and classification. They usually use global intensity information to classify pixels when the pixel intensities of foreground and background congregate into separable classes. Generally speaking, most of the classical histogram-based methods have the merits of easy implementation and fast computing speed. Some well-known examples include the Otsu's algorithm [27], entropic thresholding [28], [29], and their two-dimensional variant [30]. These methods work globally because they generate only one threshold for each image.

Underwater images usually have unevenly distributed illumination. An image captured by our platform is given in Fig. 2, in which the left grayscale image is captured by a camera pointing straight down to the water bottom where some fish food pellets scatter. The water bottom is illuminated by an array of LED lights attached to the underwater camera, and because the center part of the image is closer to lights, it is brighter than the boundary. We display the intensity of each pixel of Fig. 2(a) as height in a 3D diagram shown by Fig. 2(b). It is clear that the image suffers from non-uniform intensity distribution. Directly applying global methods to an unevenly illuminated underwater image may cause poor results. Fig. 3(a) shows the result of the original Otsu's algorithm on Fig. 2(a). Large areas of boundary background are falsely classified as foreground (white pixels). A viable alternative is to apply the method independently on local regions, and then use different thresholds to segment different local regions. However, this may produce undesirable block artifacts as shown in Fig. 3(b) when severe non-uniform illumination occurs.

To deal with non-uniform illumination in underwater environments, we propose an adaptive thresholding method. As the illumination changes dramatically on the global scale, the local intensity distribution is more reliable. But to avoid block artifacts, we consider intensities within a mask—a local window moves pixel by pixel on the image—to form the local histogram. Each pixel is only associated with the mask centered at it, and its intensity is compared with the threshold obtained from the area to decide whether it should belong to foreground or background. The effectiveness of the Otsu's algorithm is based on the assumption that data is composed of two classes that are far enough to be distinguished—i.e., the data histogram should exhibit evident bimodal pattern. If a mask contains both fish food pellets (foreground) and background, its histogram may has bimodal shape and thereby the Otsu's algorithm is feasible. The biggest difficulty here becomes how to automatically judge the pattern (or shape) of the histogram. So we utilize an EM-guided GMM with two Gaussian distributions to fit each histogram. After convergence, the pattern of a histogram can be derived from the relation between the two learned Gaussians.

2.3. Use GMM to fit mask histogram via EM iterations

Let set $\mathcal{X} = \{x_1,...,x_n\}$ denotes the intensity dataset with n pixels in the mask. Assuming $\mathcal{X}$ can be explained by a mixture of density, then $\mathcal{Z} = \{z_1,...,z_n\}$ is $\mathcal{X}$'s label set whose elements vary over $\{1,2,...,K\}$. Therefore $z_i = j$ means x_i is supposed to be generated from the jth distribution in the mixture. Here we are only interested in dividing the mask into two classes—the foreground and the background; so $K = 2$, and each intensity should belong to either of the classes. Assuming data points of each class are amenable to a Gaussian distribution:

$$p(x_i \mid z_i = j, \boldsymbol{\theta}) = N(x_i \mid \mu_j, \sigma_j), \tag{1}$$

where $N(\cdot)$ represents a Gaussian probability density function; μ_j and σ_j are the mean and the standard deviation of the jth Gaussian, respectively. $\boldsymbol{\theta} = [\boldsymbol{\omega}^T, \boldsymbol{\mu}^T, \boldsymbol{\sigma}^T]^T = [\omega_1,...,\omega_K, \mu_1,...,\mu_K, \sigma_1,...,\sigma_K]^T$ is the vector of parameters of the mixture. The GMM representation is thereby given as,

$$p(x_i \mid \boldsymbol{\theta}) = \sum_j p(x_i \mid z_i = j, \boldsymbol{\theta}) \cdot p(z_i = j \mid \boldsymbol{\theta}). \tag{2}$$

The weight $\omega_j = p(z_i = j \mid \boldsymbol{\theta})$ is the *a priori* probability for the *j*th density. Then the above equation can be rewritten as:

$$p\ x_i \mid \boldsymbol{\theta}\ = \frac{\omega_1}{\sqrt{2\pi\sigma_1}} \exp\left[-\frac{x_i - \mu_1\ ^2}{2\sigma_1^2}\right] + \frac{\omega_2}{\sqrt{2\pi\sigma_2}} \exp\left[-\frac{x_i - \mu_2\ ^2}{2\sigma_2^2}\right]. \tag{3}$$

The goal of the EM algorithm is to recursively find the maximum likelihood estimation (MLE) of $\boldsymbol{\theta}$ from a log-likelihood function:

$$L\ \boldsymbol{\theta}\ = \sum_{i=1}^{n} \log p\ \mathbf{x}_i \mid \boldsymbol{\theta}\ . \tag{4}$$

Assuming we are now at time *t*, $\boldsymbol{\theta}^{t-1}$ is the estimation of parameter $\boldsymbol{\theta}$ at time $t-1$. $\boldsymbol{\theta}^t$ is the parameter we are estimating. The EM algorithm iterates between the "E" step and the "M" step. In the "E" step, we first compute the *a posteriori* probability $p(z_i = j \mid x_i, \boldsymbol{\theta}^{t-1})$ by Bayes' Theorem:

$$p\ z_i = j \mid x_i, \boldsymbol{\theta}^{t-1}\ = \frac{p(x_i \mid z_i = j, \boldsymbol{\theta}^{t-1})\, p(z_i = j \mid \boldsymbol{\theta}^{t-1})}{\sum_j p(x_i \mid z_i = j, \boldsymbol{\theta}^{t-1})\, p(z_i = j \mid \boldsymbol{\theta}^{t-1})}. \tag{5}$$

In the "M" step at time *t*, we compute $\boldsymbol{\theta}^t$ by maximizing a term [12], [13]:

$$\boldsymbol{\theta}^t = \arg\max_{\boldsymbol{\theta}}\ \sum_i \sum_j p(z_i = j \mid x_i, \boldsymbol{\theta}^{t-1}) \log p(x_i, z_i = j \mid \boldsymbol{\theta})\ . \tag{6}$$

Finally equation (6) is used to derive the EM-guided updating equations for GMM parameters as follows:

$$\omega_j^t = \frac{1}{n} \sum_{i=1}^{n} p\ x_i, z_i = j \mid \boldsymbol{\theta}^{t-1}\ , \tag{7}$$

$$\mu_j^t = \frac{\sum_{i=1}^{n} x_i \cdot p\ x_i, z_i = j \mid \boldsymbol{\theta}^{t-1}}{\sum_{i=1}^{n} p\ x_i, z_i = j \mid \boldsymbol{\theta}^{t-1}}, \tag{8}$$

$$\sigma_j^t = \sqrt{\frac{\sum_{i=1}^{n} x_i - \mu_j^{t-1}\ ^2 \cdot p\ x_i, z_i = j \mid \boldsymbol{\theta}^{t-1}}{\sum_{i=1}^{n} p\ x_i, z_i = j \mid \boldsymbol{\theta}^{t-1}}}. \tag{9}$$

For each mask, we iterate equations (7)-(9) until the parameters converge. If the value defined in equation (10) is less than 1×10^{-3}, we believe a convergence is reached.

$$err = \frac{\left\|\boldsymbol{\mu}^t - \boldsymbol{\mu}^{t-1}\right\|_2}{\left\|\boldsymbol{\mu}^{t-1}\right\|_2} + \frac{\left\|\boldsymbol{\sigma}^t - \boldsymbol{\sigma}^{t-1}\right\|_2}{\left\|\boldsymbol{\sigma}^{t-1}\right\|_2}. \tag{10}$$

The EM algorithm is sensitive to the selection of initial parameters. Proper initial values can speed up convergence. Here we select two gray values that have the highest and the second highest bar in the histogram as μ_1^0 and μ_2^0, and the initial standard deviation is set randomly between 2.0 and 9.0. Sometimes the EM algorithm may get stuck in a local maxima and fail in fitting the histogram, and this phenomenon is usually accompanied by extremely uneven weights of Gaussians in the mixture. To avoid this situation, we check the values of ω_j after convergence. If $|\omega_1 - \omega_2|$ is larger than 0.98, the EM algorithm will run a second time with randomized initial values. Fig. 4(a) shows the GMM fitting result of a mask containing both fish food pellets and the background. The histogram in Fig. 4(a) is a typical bimodal one, and it is correctly fitted by two Gaussians in the mixture with one shown in magenta and the other in yellow. It is also possible that the mask only contains background pixels—i.e., the histogram is unimodal. In this case, the two fitted Gaussians become very close (shown in Fig. 4(b)) because the data used to drive the EM process originate from

the same density, and we should combine the two distributions. Therefore, for each mask, we use the following equation to decide its modality pattern,

$$\left|\mu_1-\mu_2\right| \underset{\text{unimodality}}{\overset{\text{bimodality}}{\gtrless}} d\cdot\ \sigma_1+\sigma_2\ . \tag{11}$$

Equation (11) can be rewritten as in:

$$\frac{\left|\mu_1-\mu_2\right|}{\sigma_1+\sigma_2} \underset{\text{unimodality}}{\overset{\text{bimodality}}{\gtrless}} d. \tag{12}$$

The LHS of the above equation is the distance between the two classes normalized by combined standard deviation. This normalized distance can reveal the extent of "overlap" between two Gaussian densities. The threshold parameter d is set to between 1.2 and 1.5 to maintain a good result in our underwater environment.

If the histogram is regarded as bimodal, the Otsu's threshold [27] is used as the bimodal threshold: $Th = Th_{bimodal} = Th_{Otsu}$. But if the histogram is unimodal, the threshold is computed by equation (13), in which s is a control parameter set between 2.0 and 3.0.

$$Th = Th_{unimodal} = \begin{cases} \mu_1 - s\cdot\sigma_1 & if\ \ \mu_1 < \mu_2 \\ \mu_2 - s\cdot\sigma_2 & else \end{cases}. \tag{13}$$

Finally we compare the center pixel's intensity against the threshold Th . If the former is smaller than the latter, then the pixel is classified as a foreground pixel (white). Conversely, the pixel is regarded as a background pixel (black).

2.4. Adaptively Adjust the Bimodal threshold

For a mask with bimodality, it is expected that the threshold should fall in the valley and perfectly divide the two peaks of the histogram. But the Otsu's threshold often deviates from the valley area in practice. This is mainly because for a histogram with two unbalanced classes, data from the bigger class tends to attract the Otsu's threshold to its mean. In bimodal masks, the background pixels greatly outnumber the foreground pixels because the mask size is much larger than size of fish food pellets (please refer to Fig. 4(a) for example); so we always come across a bigger background class in a mask histogram. The Otsu's threshold is attracted towards the background class center—i.e., the threshold becomes larger than the optimal value, and it needs to be adjusted by an offset δ .

The between-class variance corresponding to the Otsu's threshold is represented as:

$$\sigma_b^2 = \underbrace{\tilde{\omega}_1\cdot\tilde{\omega}_2}_{\substack{\text{degree of}\\ \text{balance}}}\cdot\underbrace{\left(\tilde{\mu}_1-\tilde{\mu}_2\right)^2}_{\text{separability}}, \tag{14}$$

in which the parameters $\tilde{\omega}_1$ and $\tilde{\mu}_1$ are the probability and mean of class 1; while $\tilde{\omega}_2$ and $\tilde{\mu}_2$ are the probability and mean of class 2 [27]. Equation (14) reflects the degree of balance and the distance between the two classes. For those datasets that have balanced and well-separated classes, σ_b^2 is large. While for datasets with unbalanced classes, σ_b^2 is small. This is mainly because $\tilde{\omega}_1\cdot\tilde{\omega}_2$ reaches maximum when $\tilde{\omega}_1=\tilde{\omega}_2=0.5$. A large between-class variance needs only a small or zero offset because the Otsu's threshold is already desirable. Conversely, when σ_b^2 is small, we need a larger δ to shift the threshold for better results. A natural and easy assumption is:

$$\delta \propto \exp\ -\sigma_b^2/2c^2\ . \tag{15}$$

In equation (15), the "$\propto$" sign means direct proportion. The parameter c can be interpreted as weighted average deviation from one class center to the other of all local histograms in an

underwater image. At last, the RHS of equation (15) will be multiplied by a scaling factor a to control the magnitude of the offset, and the modified threshold for a bimodal mask is given as:

$$Th_{bimodal} = Th_{Otsu} - a \cdot \exp\left(-\sigma_b^2 / 2c^2\right) . \quad (16)$$

The offset δ is adaptive because it will automatically select a proper value according to the degree of balance as well as the separability of the two classes. The undesirable phenomenon that the Otsu's threshold is attracted to the background class can then be alleviated. The superiority of using equation (16) over the original Otsu's threshold can be illustrated by Fig. 5. The complete flow chart of the proposed method is given in Fig. 6.

3. Results

Our method uses EM algorithm in a local mask to determine which type the mask is and obtains an adaptive threshold accordingly for leftover fish food detection. Experiments show that this method can effectively segment uneaten fish food from underwater images with different water turbidity levels and different extent of unevenness with respect to illumination. Fig. 7 shows comparative results of our method and several other methods including the local method cvAdaptiveThreshold incorporated in OpenCV [31], a multi-threshold method based on Otsu's algorithm [27], and the 2-D entropic thresholding [30], on several test images, and our method exhibits the best detection performance. Two quantitative measures—False Positive Rate (FPR) and True Positive Rate (TPR), are used to assess the performances of methods compared. FPR is the ratio of the number of falsely classified foreground pixels in the result to the number of background pixels in ground truth. TPR is the ratio of the number of correctly classified foreground pixels in the result image to the number of total foreground pixels in the ground truth. A good method should obtain a TPR as high as possible meanwhile keeps the FPR at a low level. As our method obtains far better results than some global methods, we only quantitatively compare the method that has close performance to ours. The FPRs and TPRs of the 3rd and the 4th columns in Fig. 7 are listed in Table 1, and our method obtains higher TPR than the cvAdaptiveThreshold method at a lower FPR level. On Fig. 7(d), our method has an evident edge over cvAdaptiveThreshold by 52% decrease of FPR and 9% increase of TPR.

For all experiments in this paper, the mask size is fixed as 49×49 for local segmentation algorithms. The proposed method is realized in unoptimized C++ code, and it costs less than 240 seconds for a 640×480 resolution image on a laptop with Intel i5-4590 CPU and 4 GB memory. The speed of the proposed method is relatively slow compared to the cvAdaptiveThreshold method that costs less than 4 seconds for a single image. The slow speed is mainly due to the computation of Gaussians. We can restrict the maximum iteration number of EM to shorten the time for GMM parameter learning, because sometimes a convergence needs a long iteration process and in fact even one iteration is enough to bring the parameters to a neighborhood of their optima. Restricting the mask size is another way of speedup. By reducing the mask size to 29×29, the time cost of the proposed method reduces to less than 100 seconds (see Table 1). In a smaller mask, the difference in processing time between our method and cvAdaptiveThreshold narrows, and the segmentation performances of both methods deteriorate. However, our method still has an edge over cvAdaptiveThreshold by FPR for small mask regions.

4. Discussion

4.1. The EM algorithm for GMMs

There are two kinds of EM algorithms for estimating GMM parameters: one is the on-line style EM which updates parameters with streaming data; and the other is the batch style EM which updates parameters by using all the data simultaneously.

For moving object detection, a popular solution is comparing each new video frame against a dynamic background model to extract foreground. As a video system keeps capturing new frames, the background should be updated all the time. Thus, on-line EM algorithm is usually utilized to update parameters of the GMM background. Stauffer and Grimson [25] proposed an adaptive scheme (similar to EM framework) to update the parameters of the GMM background model, and the foreground objects can be extracted by comparing the background model with the newest frame. KaewTraKulPong and Bowden [32] used the on-line EM (Maximum-Likelihood-Estimation style) algorithm for GMM-background learning, the GMM parameters were updated in an incremental way that was different from the adaptive Gaussian Mixture Model [25] proposed by Stauffer and Grimson. A shadow detection process was also designed by them to avoid detecting shadows as foreground objects. Hsiao *et al.* [8] used the Stauffer and Grimson's method to update GMM parameters of the background, and then contrasted with the newest underwater image to extract moving fishes in undersea videos.

Image segmentation on a single image is another situation. A batch style EM algorithm is more suitable because there is no new information coming in when carrying out the segmentation. Fei *et al.* [12] used the batch EM for generalized GMM, and the equations for updating parameters were the same with equations (7), (8), and (9). Their GMM results were directly used to segment underwater synthetic aperture sonar (SAS) images for guiding autonomous underwater vehicles (AUVs), which is different from our methodology. In our paper, we apply the batch EM to a GMM containing two Gaussians to analyze the intensity modality within each mask. Then different thresholds are used for the unimodal case and the bimodal case, respectively. The threshold for unimodal case is computed by the parameters of the learned GMM. The threshold for bimodal case is computed by the Otsu's method with an adaptive offset. The center pixel of the window area is finally compared with the threshold to decide whether the pixel should be classified as foreground or background. Therefore, the framework of the proposed methodology is different from other work mentioned previously, and the contribution mainly lies in recognizing the intensity modality of a local region, rather than the modification on Otsu's algorithm.

4.2. Parameters

Four parameters are used in the proposed segmentation algorithm (see Table 2), and their values influence the results.

The parameter d in equation (12) determines the modality of each mask. In all of our experiments, d is fixed to 1.3. We could also allow the value to vary between 1.2 and 1.5 because the final segmentation result is not sensitive to it according to experiments in several different environments.

The parameter s in equation (13) controls the unimodal threshold $Th_{unimodal}$, and it should be between 2.0 and 3.0 because it is closely related to the Gaussian confidence region. When the mask is unimodal, the histogram can be regarded to follow a single Gaussian distribution, and the pixels in the mask are mainly background pixels. The value of $Th_{unimodal}$ is between $\mu-3\sigma$ and $\mu-2\sigma$. The intensity of the central pixel is then compared with this threshold to decide whether it is foreground or background. We set parameter s to be 2.0, 2.5, and 3.0 respectively to test the corresponding FPR and TPR measures on the sample image Fig. 7(a). The results are shown by Fig. 8. A larger value leads to both lower FPR and TPR, and a lower value increases the FPR and TPR at the same time. Therefore, we fix this parameter to 2.5 in experiments for reaching a good balance between FPR and TPR.

The two parameters a and c in equation (16) are globally chosen for each underwater scene. The parameter a directly controls the magnitude of the offset. The parameter c can be interpreted as the average deviation from one class center to the other of all masks in an image, and this parameter adjusts the offset by controlling the influence of the between-class variance of the current mask. According to our practice, the parameter a should better be chosen from the interval [4, 15],

and c should be chosen from interval [5, 25] for good results. Experiments were also carried out to figure out how to select parameters that are best for different water turbidity levels and other situations. Fig. 7 displays four different underwater environments as in (a), (b), (c), and (d). The environments (a) and (c) contains small fish food pellets, while the environments (b) and (d) contains big fish food pellets. In (a) and (d) the water is clean, but in (b) and (c) the water is turbid. The parameters a and c were manually tuned to obtain desirable segmentation results as shown in Fig. 7(i), (j), (k), and (l). For the environment in Fig. 7(a) we set $a=10$ and $c=16$; for Fig. 7(b) we set $a=8$ and $c=7$; for Fig. 7(c) we set $a=6$ and $c=22$, and for Fig. 7(d) we set $a=12$ and $c=7$. It can be seen that a higher value of parameter a is suitable for the environments with clean water. For environments with turbid water, using a lower value of a can obtain better segmentation results. This is because turbid water deteriorates the sharpness of the captured image, hence causing the background class and foreground class to move close to each other, so the offset should be reduced accordingly to maintain the separability between the two peaks. It also can be observed that a higher value of parameter c is more suitable for the environments having smaller fish food pellets. This is because the smaller the fish food pellets are, the higher chance we will have a small foreground area in the mask. A small foreground area means a larger background area in the mask, and it will increase the degree of unbalance of the two classes as the background area is usually much larger than the foreground area. In this case, a larger c brings a larger offset to counteract the undesirable deviation of Otsu's threshold. By analyzing the experimental results for parameter tuning, we find that the parameter a should be chosen from interval [4, 10) for turbid water, and from interval [10, 15] for clear water. The parameter c should be chosen from interval [5, 15) for segmenting big fish food pellets, and chosen from interval [15, 25] for small fish food pellets.

The difference between using the proposed offset δ and using a constant offset is also analyzed. Fig. 9(a) shows the segmentation result using 10.0 as the offset; Fig. 9(b) shows the segmentation result using 12.0 as the offset, and Fig. 9(c) shows the segmentation result using the proposed adaptive offset. Fig. 9(a) and Fig. 9(c) have similar TPRs, but Fig. 9(c) has lower FPR. This can be easily observed by comparing the blue dashed square areas in the two figures, because the area in (a) contains much more false positives (scattered white pixels) than (c). Fig. 9(b) and Fig. 9(c) have similar FPRs, but the latter has higher TPR. This can be observed by comparing the red square areas in the two figures; because the fish food pellets of the area in Fig. 9(c) have more complete contours than those in Fig. 9(b). In general, the proposed offset is superior to the constant offset in two aspects: (i) the proposed offset reaches a balance between low FPR and high TPR, and (ii) the proposed offset automatically adjusts itself to the local environment by including the between-class variance, and avoids the intractable tuning process to some extent.

5. Conclusions

In this paper, we proposed an adaptive thresholding method for fish food detection in underwater images. To deal with non-uniform illumination in underwater images, we focus on the histogram of intensities in a local mask. The EM-guided GMM is used to fit the histogram to figure out whether the histogram is bimodal or unimodal, and then an adaptive threshold is computed accordingly. Finally, the central pixel of the mask is compared with the threshold to generate the binary detection result. Besides aquaculture, our method has potential to be applied on other segmentation problems that suffer from non-uniform background or illumination condition. In the underwater environments where the experiments were carried out, the foreground is always darker than the background. It is possible that other underwater environments have the opposite situation. If the foreground objects are darker, the proposed methodology can still apply with two modifications: (i) change the minus signs in equation (12) into plus signs, and (ii) change the minus sign in equation (15) into plus sign.

In the future, we are planning to use parallel computing technology such as GPU-based computing to speed up the segmentation. For example, NVIDIA's CUDA framework has potential

to make the pixel-wise computation hundreds of times faster than the CPU realization. Detection is only the first step of recognition, so another direction of our future endeavor is to design an intelligent underwater imaging system that is able to count uneaten fish food and recognize different types of foreground such as food pellets, underwater plants and fishes. We are also interested in combining a model-based method such as NBVU [7] with the proposed image-based segmentation to obtain better results in the future because many model-based methods are specialized in enhancing image quality and restoring underwater images.

Acknowledgments:

This work was supported in part by the National High-Tech R&D Program of China under Grant No.2013AA102305, National Natural Science Foundation of China under Grants 61603089, 61603090, and 61573258. Parts of this work were also supported by the Shanghai Sailing Program under Grant No.16YF1400100, and by the Fundamental Research Funds for the Central Universities of China under agreement No.233201600068.

Conflicts of Interest:

The authors declare no conflict of interest.

References

1. Hishamunda, N.; Ridler, N.B.; Bueno, P.; Yap, W.G. Commercial aquaculture in Southeast Asia: Some policy lessons. *Food Policy*, 2009, 34, 102-107. http://doi.org/10.1016/j.foodpol.2008.06.006
2. Troell, M.; Tydemer, P.; Rönnbäck, P.; Kautsky, N. Aquaculture and Energy use. In: Cleveland, C. (Ed.), Encyclopedia of Energy. Elsevier Inc., 2004, pp. 97–108. http://doi.org/10.1016/B0-12-176480-X/00205-9
3. Naylor, R.L.; Goldburg, R.J.; Primavera, J.H.; Kautsky, N.; Beveridge, M.C. Effect of aquaculture on world fish supplies. *Nature,* 2000, 405, 1017-1024. http://doi.org/10.1038/35016500
4. Asche, F.; Guttormsen, A.G.; Tveteras, R. Environmental problems, productivity and innovations in Norwegian salmon aquaculture. *Aquaculture Economics & Management,* 2008, 3, 19-29. DOI: 10.1046/j.1365-7313.1999.00034.x
5. Schechner, Y.Y.; Karpel, N. Recovery of Underwater Visibility and Structure by Polarization Analysis. *IEEE Journal of Oceanic Engineering,* 2005, *30, 570-587*. DOI: 10.1109/JOE.2005.850871
6. Drews. Jr. P.L.J.; Nascimento, E.R.; Botelho, S.S.C.; Campos, M.F.M. Underwater Depth Estimation and Image Restoration Based on Single Images. *IEEE Computer Graphics and Applications,* 2016, *36, 24-35*. DOI: 10.1109/MCG.2016.26
7. Sheinin, M.; Schechner, Y.Y. The Next Best Underwater View. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, pp. 3764 -3773. DOI:10.1109/CVPR.2016.409
8. Hsiao, Y.; Chen, C.; Lin, S.; Lin, F. Real-world underwater fish recognition and identification, using sparse representation. *Ecological Informatics,* 2014, *23*, 13-21. http://doi.org/10.1016/j.ecoinf.2013.10.002
9. Shen, J.; Fan, T.; Tang, M.; Zhang, Q.; Sun, Z.; Huang, F. A Biological Hierarchical Model Based Underwater Moving Object Detection. *Computational and Mathematical Methods in Medicine,* 2014, Article ID 609801. DOI: 10.1155/2014/609801
10. Walther, D.; Edgington, D.R.; Koch, C. Detection and tracking of objects in underwater video. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 544 -549. DOI: 10.1109/CVPR.2004.1315079
11. Chen, Z.; Wang, H.; Xu, L.; Shen, J. Visual-adaptation-mechanism based underwater object extraction. *Optics & Laser Technology,* 2014, *56*, 119-130. http://doi.org/10.1016/j.optlastec.2013.07.003
12. Fei, T.; Kraus, D.; Aleksi, I. An Expectation-Maximization Approach Applied to Underwater Target Detection. in: Proceedings of the 2012 International Conference on Detection and Classification of Underwater Targets, 2012, pp. 46-59.
13. Evans, F.H. Detecting fish in underwater video using the EM algorithm. in: Proceedings of ICIP (International Conference on Image Processing), 2003, vol. 2, pp. 1029-1032. DOI: 10.1109/ICIP.2003.1247423

14. Singh, S.; Soni, M.; Mishra, R.S. Segmentation of Underwater Objects using CLAHE Enhancement and Thresholding with 3-class Fuzzy C-means Clustering. *International Journal of Emerging Technology and Advanced Engineering*, 2014, *4*, *798-805*.
15. Schoening, T.; Kuhn, T.; Jones, D.O.B.; Simon-Lledo, E.; Nattkemper, T.W. Fully automated image segmentation for benthic resource assessment of poly-metallic nodules. *Methods in Oceanography*, 2016, *15-16*, *78-89*. http://doi.org/10.1016/j.mio.2016.04.002
16. Foster, M.; Petrell, R.; Ito, M.R.; Ward, R. Detection and counting of uneaten food pellets in a sea cage using image analysis. *Aquacultural Engineering*, 1995, *14*, 251-269. https://doi.org/10.1016/0144-8609(94)00006-M
17. Ang, K.P.; Petrell, R.J. Control of feed dispensation in sea cages using underwater video monitoring: effects on growth and food conversion. *Aquacultural Engineering*, 1997, *16*, 45-62. https://doi.org/10.1016/S0144-8609(96)01012-6
18. Alver, M.O.; Skoien, K.R.; Fore, M.; Aas, T.S.; Oehme, M.; Alfredsen, J.A. Modeling of surface and 3D pellet distribution in Atlantic salmon (*Salmo salar* L.) cages. *Aquacultural Engineering*, 2016, *72-73*, 20-29. http://doi.org/10.1016/j.aquaeng.2016.03.003
19. Skoien, K.R.; Aas, T.S.; Alver, M.O.; Romarheim, O.H.; Alfredsen, J.A. Intrinsic settling rate and spatial diffusion properties of extruded fish feed pellets. *Aquacultural Engineering*, 2016, *74*, 30-37. http://doi.org/10.1016/j.aquaeng.2016.05.001
20. Skoien, K.R.; Alver, M.O.; Alfredsen, J.A. A computer vision approach for detection and quantification of feed particles in marine fish farms. IEEE International Conference on Image Processing, 2014, pp. 1648-1652. DOI: 10.1109/ICIP.2014.7025330
21. Parsonage, K.D. Detection of fish-food pellets in highly-cluttered underwater images with variable illumination. Master Thesis, University of British Columbia, Canada, 2001. DOI: 10.14288/1.0058652
22. Porikli, F. Achieving real-time object detection and tracking under extreme conditions. *Journal of Real-Time Image Processing*, 2006, *1*, 33-40. DOI: 10.1007/s11554-006-0011-z
23. Li, D.; Xu, L.; Goodman, E.D. Illumination-Robust Foreground Detection in a Video Surveillance System. *IEEE Trans. on Circuits and Systems for Video Technology*, 2013, *23*, 1637-1650. DOI: 10.1109/TCSVT.2013.2243649
24. Li, D.; Xu, L.; Goodman, E. Online background learning for illumination-robust foreground detection. IEEE the 11th International Conference on. Control Automation Robotics & Vision (ICARCV), 2010, pp. 1093-1100. DOI: 10.1109/ICARCV.2010.5707245
25. Stauffer, C.; Grimson, W.E. Adaptive background mixture models for real time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999, pp. 246-252. DOI: 10.1109/CVPR.1999.784637
26. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 1977, *39*, 1-38.
27. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979, *9*, 62-66. DOI: 10.1109/TSMC.1979.4310076
28. Pun, T. Entropic thresholding: a new approach. *Computer Graphics and Image Processing*, 1981, 16, 210-239. https://doi.org/10.1016/0146-664X(81)90038-1
29. Kapur, J.N.; Sahoo, P.K.; Wong, A.K.C. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 1985, *29*, 273-285. https://doi.org/10.1016/0734-189X(85)90125-2
30. Abutaleb, A.; Eloteifi, A. Automatic Thresholding of Gray-Level Pictures Using 2-D Entropy. 31st Annual Technical Symposium, International Society for Optics and Photonics, 1988, pp. 29-35. DOI: 10.1117/12.942103
31. Gary, B.; Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV library*, O' Reilly Media, Inc., 2008; pp. 138-141.
32. KaewTraKulPong, P.; Bowden, R. An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In: Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, Sept. 2002, pp. 135-144. DOI: 10.1007/978-1-4615-0913-4_11

Dawei Li received the Bachelor of Engineering degree in Automation in 2006 from Tongji University, Shanghai, China. He received his PhD degree from Tongji University in January, 2013. During 2013-2015, he worked in the Department of Computer Science and Technology Department as a postdoc. In 2015, he became a lecturer in College of Information Science and Technology, Donghua University, Shanghai, China.

During 2009-2010, he was a visiting researcher with the Genetic Algorithms Research and Applications Group (GARAGe) at Michigan State University. His research there was about computer vision and video surveillance technology. His current research interests include image processing, computer vision, pattern recognition and computational intelligence.

Lihong Xu received the Ph.D degree in engineering from the Department of Automatic Control, Southeast University, Nanjing, China, in 1991.In 1994, he was appointed as a Professor at Southeast University. He transferred to Tongji University in August, 1997, and has been a Professor with Tongji University since then. His research fields include control theory, computational intelligence, and optimization theory.

In 1998 he was funded by the Daylight Project of Shanghai. In 1999 he was funded by the University Backbone Young Tutors Project. He is now doing joint research work as a Visiting Professor and Advisor of the green house research team of BEACON, USA. Prof. Xu is a member of ACM, and the President of IEEE CIS's Shanghai Chapter. He was the Co-Chair of the 2009 GEC Summit in Shanghai. He is also a Standing Director of the Chinese Society of Agricultural Engineering.

Huanyu Liu received the Bachelor of Engineering degree in Automation in 2012 from Tongji University, Shanghai, China. He received Master Degree in Engineering from Tongji University in April, 2015. Later, he became an engineer in Huawei Technologies Co. Ltd., Shanghai, China.

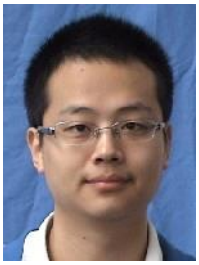

Figure 1. The underwater imaging platform and the aquaculture site. (a) and (b) are images of the underwater system including a 1.9-meter high holder, an underwater camera with LED lights, and a plate for holding the uneaten fish food pellets. (c) shows the situation of placing the underwater platform to the daily feeding spot of the pool. (d) shows the little boat that can distribute fish food and relocate the underwater imaging system. (e) shows the interface of the server that connects the underwater camera by a cable.

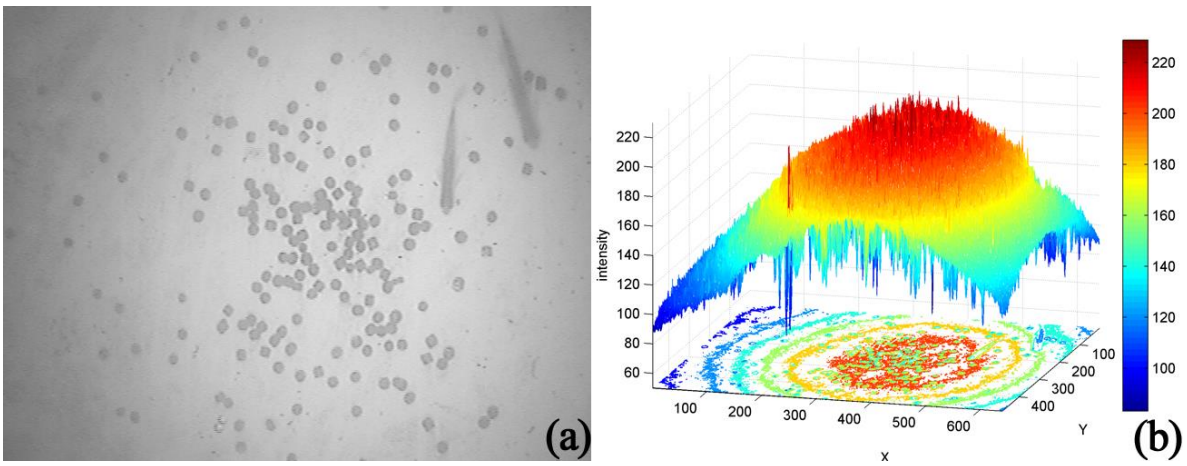

Figure 2. An underwater grayscale image with non-uniform illumination condition and its intensity distribution. (a) is the original image captured by a camera pointing straight down to the water bottom with conpensating LED lights. In this image uneaten fish food pellets and several fishes scatter around the water bed; (b) is the 3D plot of the intensities of all pixels of Fig. 2(a). The horizontal plane of Fig. 2(b) is the image plane, and the vertical axis stands for intensity value. The color encodes the intensity of each pixel, and the correspondence of the intensities and colors are given by the color-bar on the right side of Fig. 2(b).

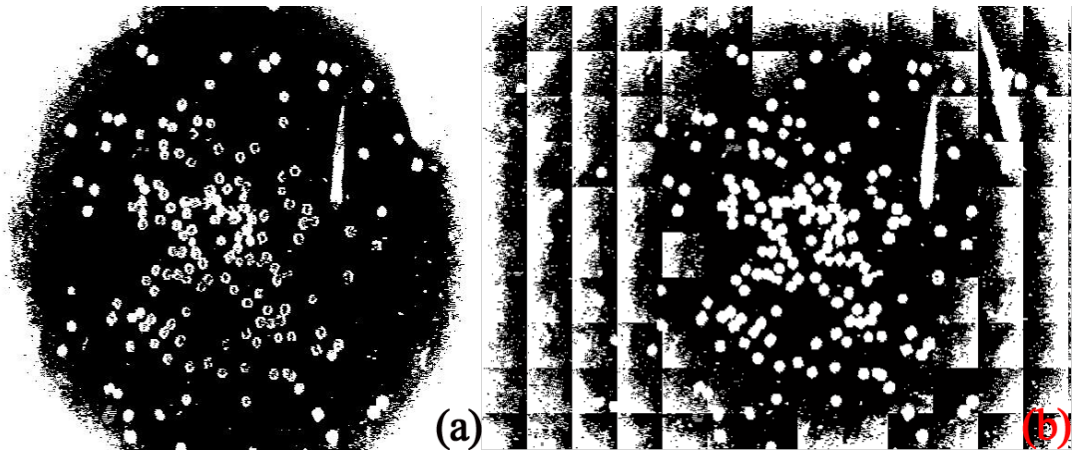

Figure 3. Results of the original Otsu's algorithm and the local version on Fig. 2(a). (a) is the result of the original Otsu's algorithm; although most food pellets are detected as foreground (white pixels), background area around image boundary is falsy classifed as foreground. (b) is the result of applying the Otsu's algorithm on local regions. The whole image is divided into 13*10 blocks, and then each block is segmented by a different threshold. The result suffers from numerous block artifacts.

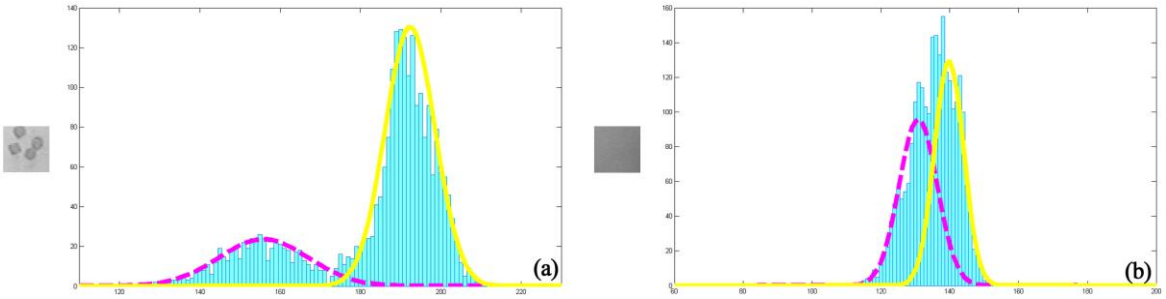

Figure 4. The GMM fitting results on histograms of some local masks. The horizontal axis stands for the pixel intensity, and the vertical axis stands for the number of pixels in each intensity bin. The mask in (a) contains both fish food pellets (foreground) and background, whereas the mask in (b) only contains background pixels. We directly display the two Gaussians of the converged mixture on each histogram with one in magenta dashed curve and the other in yellow solid curve. All the Gaussian pdfs in Fig. 4 were scaled up by 2700 to be tall enough for observation.

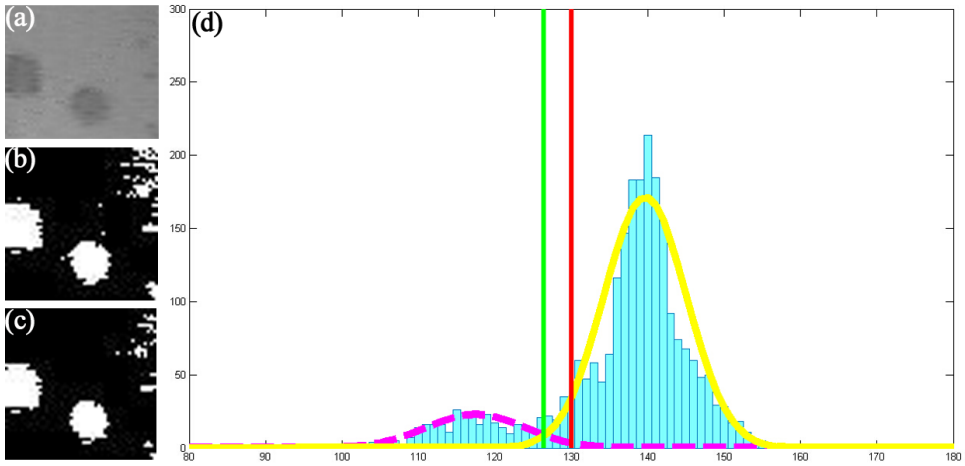

Figure 5. The histogram of the mask (a) is plotted in (d). The two peaks in the histogram (d) are not well-balanced. The Otsu's threshold is labeled by a red vertical line and the green vertical line is the threshold line computed by equation (16). Binary image (b) is the segmentation result using the Otsu's threshold only, and (c) is the result using the threshold with the offset. It is obvious that (c) has lower noise level than (b). The horizontal axis of (d) stands for the pixel intensity, and the vertical axis of (d) stands for the number of pixels in each intensity bin.

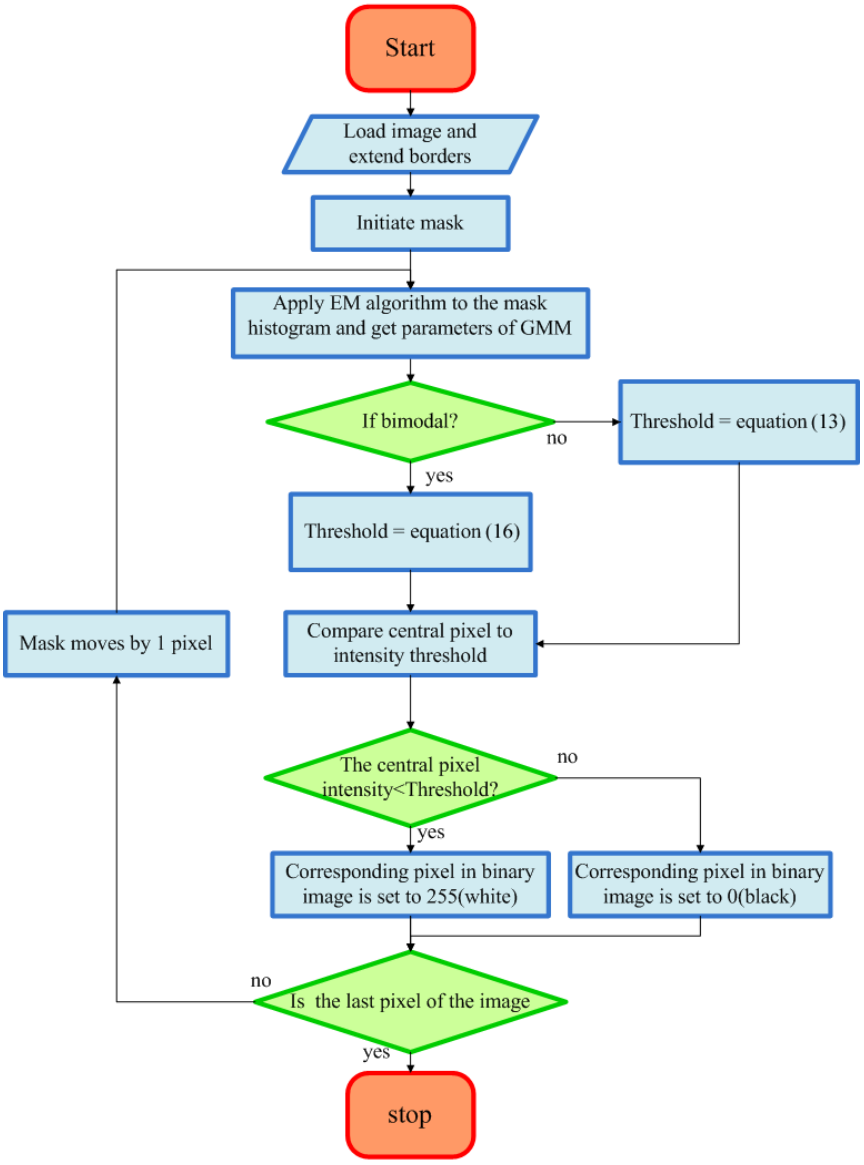

Figure 6. The complete flow chart of our method.

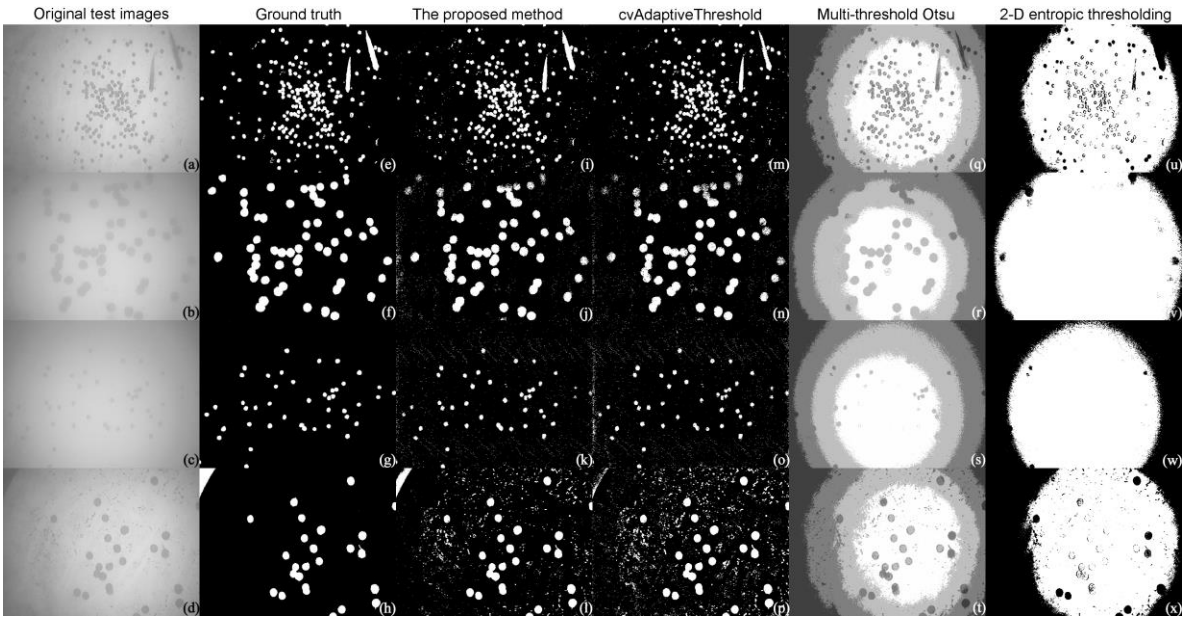

Figure 7. Comparative results of leftover fish food detection on 4 underwater images. The first column are four original grayscale images. The second column are manully labelled ground truth corresponding to the original images. The 3rd column gives the results of the proposed method, and the 4th column shows results of the cvAdaptiveThreshold method in OpenCV [31]. The 5th column shows results of the multi-threshold segmentation [27], and the last column shows the results of 2-D entropic thresholding [30]. Our method yields the closest results to the ground truth images across all methods compared. The performance of the cvAdaptiveThreshold method is the closest to ours, but our method presents better foreground contours near image boundaries and has lower detection noise (compare Fig. 7(l) and (p)). For results of our method (the 3rd column), the mask size is fixed as 49×49. Parameter d is fixed at 1.3, and parameter s is fixed at 2.0. We set $a=10$ and $c=16$ for (i); $a=8$ and $c=7$ for (j); $a=6$ and $c=22$ for (k); $a=12$ and $c=7$ for (l). For the cvAdaptiveThreshold method, we set the "block_size" parameter to 49 and the "param1" parameter to 10 for generating the best results shown in the 4th column.

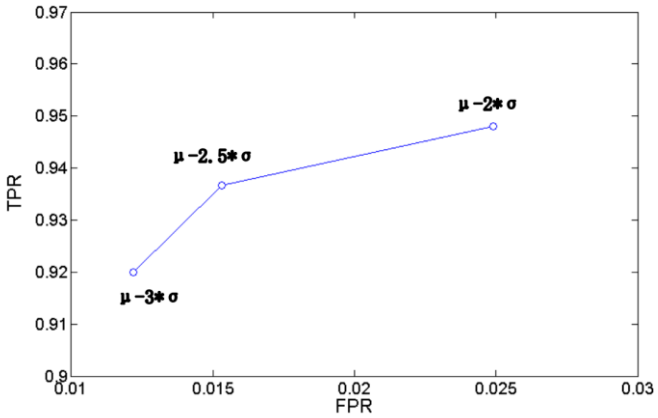

Figure 8. The parameter *s* is set to be 2.0, 2.5, and 3.0 respectively to test the corresponding FPR and TPR measures on sample image Fig. 7(a). It seems that *s*=2.5 reach a good balance between false positives and true positives.

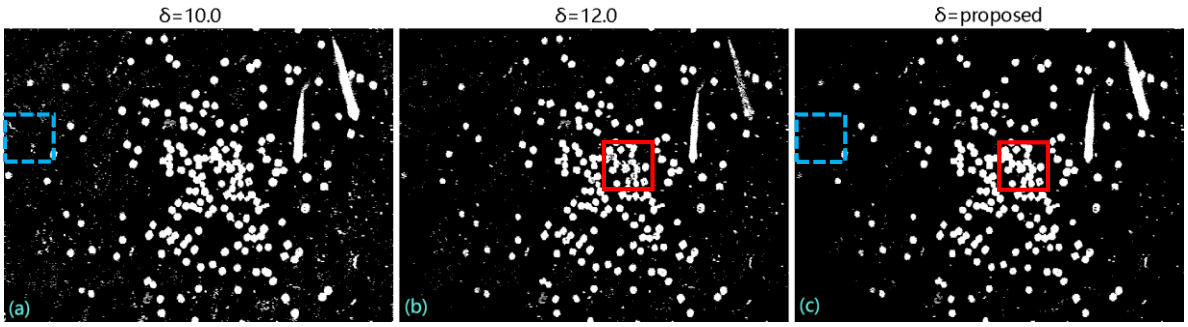

Figure 9. Comparison of segmentation results on an underwater image by using constant offset and using the proposed adaptive offset. (a) is the result using 10.0 as the offset; (b) is the result using 12.0 as the offset; (c) is the result using the proposed adaptive offset. (c) is the best result among the three because it maintains low false alarm rate and high detection rate at the same time. (a) and (c) have similar TPR; (b) and (c) have similar FPR. By comparing the dashed square areas in (a) and (c), and by comparing the square areas in (b) and (c), the proposed offset is superior to the constant offset.

Table 1. Quantitative comparison between our method and the cvAdaptiveThreshold method in OpenCV.

		Fig. 7(a)	**Fig. 7(b)**	**Fig. 7(c)**	**Fig. 7(d)**
FPR (49×49)	Our method	0.0153	0.0119	0.0116	0.0269
(lower is better)	cvAdaptiveThreshold	0.0158	0.0122	0.0176	0.0563
TPR (49×49)	Our method	0.9366	0.91	0.8144	0.9591
(higher is better)	cvAdaptiveThreshold	0.9281	0.8612	0.7974	0.8796
Time (49×49)	Our method	161	219	235	200
(Second)	cvAdaptiveThreshold	3	3	4	3
Time (29×29)	Our method	70	88	91	79
(Second)	cvAdaptiveThreshold	2	2	2	2

Table 2. Parameters used in the proposed method.

	Usage	Suggested value
d	Determine the modality of each mask	[1.2, 1.5]
s	Control the threshold for the unimodal case	[2.0, 3.0]
a	Directly control the magnitude of the offset for the bimodal threshold	[4, 10): turbid water [10, 15]: clear water
c	Control the offset value for computing the bimodal threshold	[5, 15): big object [15, 25]: small object