Automatic Feeding Control for Dense Aquaculture Fish Tanks

Yousef Atoum, Steven Srivastava, and Xiaoming Liu, *Member, IEEE*

***Abstract*—This paper introduces an efficient visual signal processing system to continuously control the feeding process of fish in aquaculture tanks. The aim is to improve the production profit in fish farms by controlling the amount of feed at an optimal rate. The automatic feeding control includes two components: 1) a continuous decision on whether the fish are actively consuming feed, and 2) automatic detection of the number of excess feed populated on the water surface of the tank using a two-stage approach. The amount of feed is initially detected using the correlation filer applied to an optimum local region within the video frame, and then followed by a SVM-based refinement classifier to suppress the falsely detected feed. Having both measures allows us to accurately control the feeding process in an automated manner. Experimental results show that our system can accurately and efficiently estimate both measures.**

***Index Terms*—Bag-of-Words (BoW), correlation filter (CF), feeding control, fish, HOG, particle filter.**

I. Introduction

BASED on the statistics from Fisheries and Aquaculture Department [1], aquaculture is growing at a very high rate internationally, and its contribution to the world's total fish production reached 42.2% in 2012, up from 25.7% in 2000. The fish feeding process is one of the most important aspects in managing aquaculture tanks, where the cost of fish feeding is around 40% of the total production costs [6].

Monitoring several aquaculture tanks with highly populated fish is a challenging task. Many researchers adopt a *telemetry*-based approach to study fish behavior [5], [7]. In addition, some scientists prefer a computer vision (CV)-based approach for fish monitoring [8], [10], [17]–[19]. Unfortunately, all these studies are conducted at a *small* scale, i.e., a small number of fish in small tanks. Compared to fish behavior, excess *feed* detection is rarely addressed except [12], where feeding control is achieved by estimating fish appetite. However, the tank in [12] is also small and fish can be easily segmented.

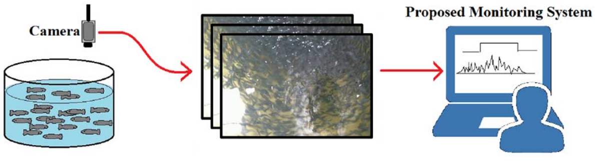

Fig. 1. Given the video input, our system performs real-time monitoring and feeding decision for a highly dense fish tank.

By collaborating with an active aquaculture fish farm, we have developed a CV-based automated feeding control system. A video camera is placed above the water surface of a highly *dense* fish tank with around 10,000 fish, as shown in Fig. 1. The camera captures only part of the water surface due to the large tank size. Videos are directly transferred to a host computer that performs real-time analysis on the state of fish behavior. Moreover, the system is also programmed to take immediate actions in stopping the feeding process when needed.

In this paper we present an efficient CV system to continuously monitor fish eating activity, detect excess feed, and automatically control the feeding process. A two-class classifier is learned to distinguish whether fish are actively consuming feed or not. To detect the amount of feed floating on the water surface, we propose a novel two-stage approach. First, a supervised learned correlation filter is applied to the test frame in order to detect every individual feed. Second, a Support Vector Machine (SVM) classifier is deployed as a refinement step of the correlation filter output, which attempts to suppress falsely detected feed while preserving true feed. Furthermore, we propose to detect feed in an *optimum* local region only, rather than the entire frame whose accuracy and efficiency are both less than ideal. Using the particle filter technique, the local region is estimated by maximizing the correlation between the number of locally detected feed and that of true feed in the entire frames. Finally, based on continuous measurements from fish activity and feed detection, various actions take place to control the feeding process.

This paper makes the following contributions: 1) a fully automated aquaculture monitoring system that controls feeding for a highly dense fish tank, 2) an accurate measure of the fish activity, and a continuous detection of excess feed from an optimum local region, and 3) the video dataset and the labels that are publicly available for future research.

II. Proposed Method

Monitoring the fish eating activity along with making sure the fish are provided with the correct amount of feed are the main goals of our system. Our system architecture is shown in Fig. 2, and we present each part in the following sections.

Manuscript received July 31, 2014; revised December 01, 2014; accepted December 09, 2014. Date of publication December 24, 2014; date of current version January 13, 2015. This work was supported in part by the Herrick Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kai-Kuang Ma.

Y. Atoum is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: atoumyou@msu.edu).

S. Srivastava is with Aquaculture Research Corporation, Tecumseh, MI 49286 USA (e-mail: s.k.srivastava@aquaresearchcorp.com).

X. Liu is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: liuxm@cse.msu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2014.2385794

1070-9908 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

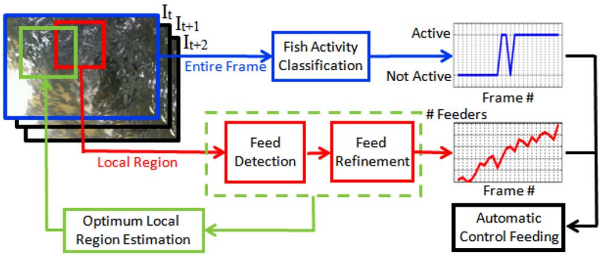

Fig. 2. The architecture of the proposed feeding control system.

A. Fish Activity Classification

To classify the fish activity, a set of features is extracted to best characterize the fish while consuming feed. Due to the large size of video frames, as well as the non-uniform spatial distribution of fish activity within the tank, a video frame $\mathbf{I}$ is uniformly divided by a 3×3 grid. Features $\mathbf{v}^j$ are extracted from each of 9 different regions and concatenated to a single feature vector $\mathbf{v} = (\mathbf{v}^{1\top}, \cdots, \mathbf{v}^{9\top})^\top$ to represent one video frame.

When the feeding starts, ripples and waves are generated throughout the tank due to fish breaking the water surface. Hence, one simple feature is the difference between the consecutive frames $\Delta \mathbf{I}^j = \mathbf{I}_t^j - \mathbf{I}_{t-1}^j$, which indicates the abrupt changes in waves and ripples on the water surface. Furthermore, the existence of waves is normally accompanied with light reflections. Therefore, by setting a proper threshold (τ_l) to $\mathbf{I}$, we obtain a binary image $\mathbf{I}_l$ that captures the brighter light reflections. While a fixed τ_l is used for all our data, a learning-based adaptive threshold τ_l (e.g., [15]) is desired in the future for applying our system to different fish farms. The feature of one region is computed by $\mathbf{v}^j = (\mu(\Delta \mathbf{I}^j), \mu(\mathbf{I}_l^j), \sigma(\mathbf{I}_l^j))^\top$, where μ and σ denote the mean and standard deviation respectively. A 27-dim feature is extracted for one video frame.

By using training videos with labels on the fish activity, we learn a two-class classifier via standard learning schemes. Given a test video, such a classifier makes a decision on a per-frame basis. However, in our application the most desired capability is to accurately measure the duration when fish are actively eating by estimating the *onset* and *offset*, i.e., the beginning and end of this active eating duration. Therefore, we perform two post-processing steps following the frame-based classifier decision. First, to reduce the noise in classification results, a sliding voting window (SVW) process is applied, where the majority vote within a 10-frame window is treated as the classification result of the *last* frame within the window. Second, using a larger window of 100 frames, we estimate the onset and offset of the active eating duration by monitoring *when* the fraction of "active" frames has increased or decreased to the half. Note that, in both post-processing steps, our system operates in an *online* processing mode, i.e., only the past, instead of the future, frames are used in decision making.

B. Feed Detection

Accurately estimating the amount of excess feed floating on the water is a critical component for any *intelligent* aquaculture system. However, detecting individual feed is very challenging due to the tiny feed size, partially submerged into the water, and light reflection. Further, feed detection should be conducted in real time for *immediate* feeding control. These challenges motivate us to develop a carefully designed feed detector with three components: 1) correlation filter is used to detect all possible feed, 2) a classifier built on rich features suppresses non-feed from the first component, and 3) a local region is searched to maximize the computational efficiency and accuracy. We now discuss each component in detail.

Correlation Filter for Feed Detection: The efficiency challenge is attributed by the contrast between the large frame size (1080 × 1960 pixels) and tiny feed size (~30 pixels), i.e., a huge number of local candidates to be classified as feed vs. non-feed. To address this challenge, we like to *efficiently* rule out the majority of non-feed candidates while preserving most true feed. Correlation filter (CF) is chosen for this purpose due to its proven success in object detection [11], [4].

Specifically, we adopt the unconstrained scalar feature approach [3], which is learned by minimizing the average Mean Square Error between the cross correlation output and the desired correlation output for all training images, i.e.,

$$\min_{\mathbf{h}} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i \oplus \mathbf{h} - \mathbf{g}_i\|_2^2 + \lambda \|\mathbf{h}\|_2^2, \tag{1}$$

where $\mathbf{h}$, $\mathbf{x}_i$, $\mathbf{g}_i$, λ, and $\oplus$ are the CF, visual features of ith image, desired output, regularization weight, and convolution, respectively. Converting into the frequency domain results in,

$$\hat{\mathbf{h}} = \left[\lambda \mathbb{I} + \frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{X}}_i^\dagger \hat{\mathbf{X}}_i \right]^{-1} \left[\frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{X}}_i^\dagger \hat{\mathbf{g}}_i \right], \tag{2}$$

where $\hat{}$ is the FFT operation, $\hat{\mathbf{X}}$ is the diagonal matrix with $\hat{\mathbf{x}}$ on its diagonal, $\dagger$ is conjugate transpose, and $\mathbb{I}$ is the identity matrix. A set of N local patches (of size LXL) with true feed in the center are used as the training images. For *efficiency* the raw intensity is used as $\mathbf{x}$. Given a test image $\mathbf{x}_t$, the convolution output $\mathbf{x}_t \oplus \mathbf{h}$ containing peaks larger than a threshold τ are detected as the candidate feed. We choose τ where the maximum true detection and minimal false alarm are achieved.

Classifier for Feed Refinement: While the CF can efficiently nominate candidate feed, it is likely to have false alarm due to its simple intensity feature. Therefore, the second component of feed detection focuses on the *accuracy* challenge, by performing an accurate two-class (feed vs. non-feed) classification on the set of candidate feed resulted from the CF. To learn the classifier, the same patches in learning the CF are also used as positive training samples, while the false alarm candidate patches are used as negative samples.

Given the much fewer number of candidate feed to classify than the CF, we can afford to extract a *rich* feature representation for classification. First, feed is visually distinguishable by its *color*. We employ the Bag-of-Words (BoW) [16] to learn the color feature. Using K-means clustering on the Cartesian representation of the RGB color space, $d_c (= 20)$ code words, $\{s_d\}_{d=1}^{d_c}$, indicate the representative colors in all training samples. For a training sample $\mathbf{P}$, we convert each pixel to the nearest color words, and generate a d_c-dim BoW histogram $\mathbf{f}_c(i) = \sum_{(u,v) \in \mathbf{P}} \delta(i = \arg\min_d \|\mathbf{P}(u,v) - s_d\|_2)$, where δ is the indicator function. This histogram is further normalized by $\mathbf{f}_c = \frac{\mathbf{f}_c - \min(\mathbf{f}_c)}{\max(\mathbf{f}_c) - \min(\mathbf{f}_c)}$. Second, since feed has unique *edges* with certain *orientation*, for each training sample we also compute the 36-dim Histogram of Oriented Gradients (HOG) feature $\mathbf{f}_h$ from 2×2 cells [9]. Finally, a 56-dim feature $\mathbf{f} = (\mathbf{f}_c^\top, \mathbf{f}_h^\top)^\top$ is extracted from each sample, the collection of which is fed into kernel SVM to learn the classifier.

TABLE I
RULES FOR CONTROLLING THE FEEDING PROCESS

# of Feed	Fish Active	Feeding Machine	Action
High	Yes	On	Off
High	No	On	Off
Low	Yes	Off	On
Low	No	On	Off

Locating the Optimum Local Region: While the two-step feed detection can be applied to the entire video frame, we propose to use a *specific local region* instead. First, due to inevitable light reflection on certain areas, detection in the entire frame may lead to worse performance than in a local region. Second, the latter is also more efficient. Finally, a *specific* local region should be chosen rather than a *random* one. Due to non-uniform spatial distributions of light reflection, feed in certain local region are more *detectable*. Further, since the subset of feed detected locally may not well represent the overall amount of excess feed, we propose to search the optimum local region by maximizing the correlation between the number of locally detected feed and that of true feed in the frame. This strategy is also motivated by the fact that the feed control relies more on the *dynamics*, than the absolute number, of feed.

One may use brute-force search to find the optimum local region. Alternatively, we adopt a more efficient particle filter-based approach [2], [14]. The basic idea is to iteratively update a set of particles $\{\mathbf{c}^k, w^k\}_{k=1}^K$, where $\mathbf{c}^k$ is the location and size of a local region and w^k is its weight, so that all particles converging toward the optimum local region. We manually label the ground truth feed for randomly chosen n entire frames, each with the number of feed being G^i. All K particles are initially distributed uniformly within the image frame, and the weight is computed by Pearson's correlation coefficient,

$$w^k = \frac{\sum_{i=1}^n (T_i^k - \mu(T^k))(G_i - \mu(G))}{\left(\sum_{i=1}^n \left(T_i^k - \mu(T^k)\right)^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^n \left(G_i - \mu(G)\right)^2\right)^{\frac{1}{2}}}, \quad (3)$$

where T_i^k is the estimated number of feed in $\mathbf{c}^k$, and $\mu(T^k)$ is the mean of T_i^k over n images. At each iteration, a CDF-based resampling is conducted where the particles with larger w^k will have higher chances to be selected in the next iteration than the ones with smaller w^k. Once a particle is selected, we add a random perturbation to $\mathbf{c}^k$. With sufficient iterations, all K particles converge to regions with larger weights, where the one with the largest w^k is chosen as the optimum local region.

C. Automatic Control of the Feeding Process

The purpose of classifying the fish behavior at every frame, as well as detecting the amount of excess feed, is to automatically control the feeding process without the need of human intervention. Ultimately, the goal is to prevent both *overfeed* and *underfeed* to the fish tank. Based on the per-frame results obtained from both the fish activity classification and the feed detection, a continuous decision is made on whether to stop or continue the feeding. We use a rule-based method as listed in Table I. It represents some critical conditions under which the "action" of stopping or continuing feeding will be taken immediately. For example, if the number of feed is high and continuously increasing over a long period of time while the machine is still feeding, the machine needs to be stopped until the number of feed drops below a certain level.

TABLE II
THE ACCURACY OF FISH ACTIVITY CLASSIFICATION

Method	Error rate (%)	Error rate w/ SVW(%)	Onset error (Sec.)	Offset error (Sec.)
MLE	1.79	1.64	6.16	6.41
Adaboost	1.63	**1.49**	**5.56**	**7.24**
SVM	1.62	1.62	6.08	7.33

III. EXPERIMENTAL RESULTS

Our dataset consists of 21 videos of a top-view aquaculture fish tank. These videos were captured at 10 FPS, with 1080 × 1960 pixels and an average length of 5, 684 frames. The first 20 videos are captured under normal circumstances. Each video is manually labeled with the onset and offset frames of the fish eating activity, which is used to evaluate the fish activity classification. The last video exhibits a huge amount of excess feed, since the feeding machine is intentionally switched on for a longer period of time. To evaluate the feed detection, we manually label feed in $n = 12$ frames randomly taken from the video at different stages of the feeding process. We conduct the labeling twice and only the feed labeled in both trials are claimed as true feed. The number of true feed ranges from 22 to 856 per frame, with the total of 4, 485 feed.

A. Fish Activity Classification

The experiment of the fish activity classification follows a Leave-One-Video-Out-Cross-Validation scheme. We use three classifier learning schemes: Maximum Likelihood Estimations (MLE), Adaboost and SVM. As shown in Table II, the fact that all three methods achieve good performance demonstrates the effectiveness of our feature representation. The best performing classifier is Adaboost with a per-frame-based error rate of 1.49%. The "onset error" measures the difference between the estimated onset and the labeled ground-truth onset. The offset error is slightly larger than the onset, partially because there is more inconsistency in labeling the offset among videos.

B. Feed Detection

We set the parameters as $N = 2,000$, $L = 25$, $\tau_l = 229$, $\tau = 0.53$, and g_i is a 2D Gaussian centered at the targets locations with a variance of 2 and peak amplitude of 1. The default parameters in LibSVM are used for SVM learning.

Fig. 3 compares the results of the CF in the local region alone vs. having a SVM refinement classifier following the CF. The Normalized False Alarm (NFA) is the number of falsely detected feed divided by the number of true feed. Remarkably, the refinement classifier reduces the amount of false alarm by nearly 50%, while maintaining the similar true detection rate. For example, one good point on ROC has the detection rate of 90.8% at a NFA of 0.3. Further, the results of operating on the entire frame is much worse than on the local region. Finally, we also employ the SVM classifier for feed detection *without* first applying the CF. It can detect 85.3% of feed, but the NFA is considerably high at 4.5, not to mention the much lower efficiency. The superior over this baseline demonstrates the excellent accuracy and efficiency of our two-step approach.

An illustration of feed detection procedure is shown in Fig. 6. Columns 1-2 are successful at detecting all feed with no false alarms. Columns 3-4 have missing detection, but no false alarms. Columns 5-8 illustrate variations of false alarms.

C. Local Region Estimation

The number of particles for localizing the optimum local region is 100. Since the results of the particle filter depend on the

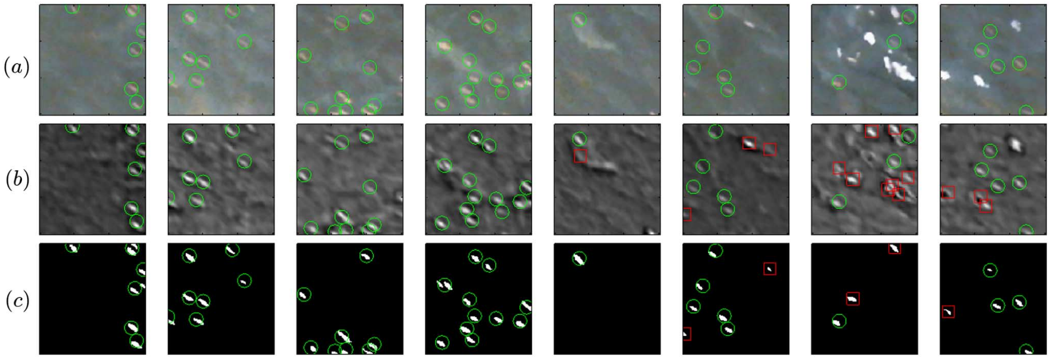

Fig. 6. Feed detection procedures with each column being one local region of 150 × 150 pixels: (a) the original image with green circles indicating labeled ground-truth feed, (b) the CF output (green and red), where the red squares are false alarms, and (c) the results of the SVM classifier in a binary image where the white regions are the final detected feed. Note the reduced false alarms from (b) to (c) (best viewed in color).

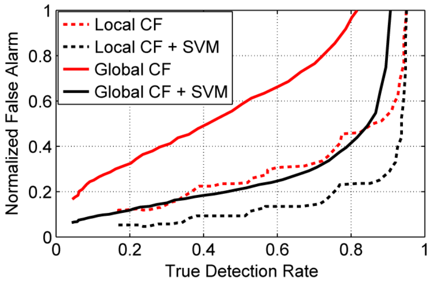

Fig. 3. Comparison of feed detection.

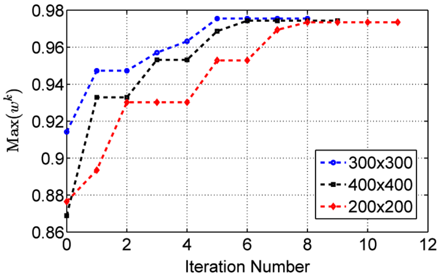

Fig. 4. Local region optimization.

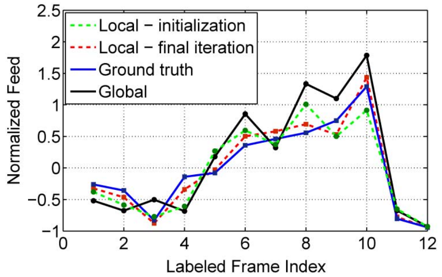

Fig. 5. Comparison of normalized feed.

initialization, we repeat this experiment three times with different initial sizes of local regions. The maximum w^k for three runs are shown in Fig. 4. Note that the final iterations of all runs achieve a similar weight of 0.97, due to the huge overlap in the final optimum local region. The optimum local region is found to be centered at $(224 \pm 3, 256 \pm 4)$, with a size of 258 × 258. The fact that all three runs converge to the same local region gives a strong indication of achieving the global optimal solution for this optimization.

To illustrate the effectiveness of the particle filter, we plot four signals: ground-truth feed in the entire frame G_i, T_i when feed detection is applied to the frame, T_i^k with the maximum w^k at the initialization and at the final iteration. To compensate different data ranges, we plot the normalized feed as $\frac{T_i^k-\mu(T^k)}{T_i^k}$ in Fig. 5. Compared to the initialization and the global feed detection, the feed estimation at the optimum local region has the highest correlation with the ground-truth feed. Therefore, the feeding control based on the local region is almost the same as based on the true feed of the entire frame.

D. Computational Efficiency

The computational efficiency is an important metric for any computer vision system. We evaluate the efficiency using a Matlab implementation on a conventional Windows 8 desktop computer with an Intel i5 CPU at 3.0 GHz with 8 GB RAM. First, for classifying fish activity, most of the computation is on the feature extraction, which is 0.106 sec. per frame. The total time for classifying the activity of one frame via MLE, Adaboost and SVM are 0.156, 0.108 and 0.123 sec., respectively. Second, the efficiency of feed detection depends on several factors, such as the size of the local region, the number of candidate feed for the SVM classifier. The total time for the CF step in the optimum local region is 0.006 seconds. The refinement classifier requires 0.004 sec. to extract features and classify a single candidate feed resulted from the CF. The average total time to detect feed in the local region is 0.085 seconds. In summary, our entire system operates at 5+ FPS. With the future C++ implementation, we believe that our system can operate in real time on a conventional PC.

IV. Conclusions

A fully automatic system is developed to understand fish eating behavior in a highly dense aquaculture tank. The ability to classify whether the fish are actively consuming feed along with the continuous detection of excess feed provides valuable information for feeding control in the tank. In the future, we will enhance the system by providing a continuous measure of how active the fish are in a scale ranging from zero to one, and leverage shot boundary detection [13] for more precise onset/offset detection. We also plan to extend the system to infer fish growth from the behavior of fish movement, with the goal of developing *computational* and *quantitative* approaches toward a comprehensive understanding of fish growth.

REFERENCES

[1] FAO global aquaculture production volume and value statistics database updated to 2012, FAO Fisheries and Aquaculture Dept., Tech. Rep., 2014.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[3] V. N. Boddeti, T. Kanade, and B. Kumar, "Correlation filters for object alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2291–2298, IEEE.

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2544–2550, IEEE.

[5] C. J. Bridger and R. K. Booth, "The effects of biotelemetry transmitter presence and attachment procedures on fish physiology and behavior," *Rev. Fisheries Sci.*, vol. 11, no. 1, pp. 13–34, 2003.

[6] C. Chang, W. Fang, R.-C. Jao, C. Shyu, and I. Liao, "Development of an intelligent feeding controller for indoor intensive culturing of eel," *Aquacult. Eng.*, vol. 32, no. 2, pp. 343–353, 2005.

[7] S. G. Conti, P. Roux, C. Fauvel, B. D. Maurer, and D. A. Demer, "Acoustical monitoring of fish density, behavior, and growth rate in a tank," *Aquacult. Eng.*, vol. 251, no. 2, pp. 314–323, 2006.

[8] C. Costa, A. Loy, S. Cataudella, D. Davis, and M. Scardi, "Extracting fish size using dual underwater cameras," *Aquacult. Eng.*, vol. 35, no. 3, pp. 218–227, 2006.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 1, pp. 886–893, IEEE.

[10] S. Duarte, L. Reig, and J. Oca, "Measurement of sole activity by digital image analysis," *Aquacult. Eng.*, vol. 41, no. 1, pp. 22–27, 2009.

[11] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. IEEE Conf. Int. Conf. Computer Vision (ICCV)*, 2013, pp. 3072–3079, IEEE.

[12] J.-V. Lee, J.-L. Loo, Y.-D. Chuah, P.-Y. Tang, Y.-C. Tan, and W.-J. Goh, "The use of vision in a sustainable aquaculture feeding system," *Res. J. Appl. Sci., Eng. Technol.*, vol. 6, no. 19, pp. 3658–3669, 2013.

[13] X. Liu and T. Chen, "Shot boundary detection using temporal statistics modeling," in *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, vol. 4, p. IV–3389, IEEE.

[14] X. Liu and T. Chen, "Face mosaicing for pose robust video-based recognition," in *Proc. IEEE Conf. Asian Conf. Computer Vision (ACCV)*, 2007, pp. 662–671, Springer.

[15] J. Roth, X. Liu, and D. Metaxas, "On continuous user authentication via typing behavior," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4611–4624, 2014.

[16] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, 2009.

[17] L. H. Stien, S. Bratland, I. Austevoll, F. Oppedal, and T. S. Kristiansen, "A video analysis procedure for assessing vertical fish distribution in aquaculture tanks," *Aquacult. Eng.*, vol. 37, no. 2, pp. 115–124, 2007.

[18] J. Xu, Y. Liu, S. Cui, and X. Miao, "Behavioral responses of tilapia (oreochromis niloticus) to acute fluctuations in dissolved oxygen levels as monitored by computer vision," *Aquacult. Eng.*, vol. 35, no. 3, pp. 207–217, 2006.

[19] B. Zion, V. Alchanatis, V. Ostrovsky, A. Barki, and I. Karplus, "Real-time underwater sorting of edible fish species," *Comput. Electron. Agricult.*, vol. 56, no. 1, pp. 34–45, 2007.