Vision-based counting of scallops from AUV benthic images: targeting false positives

Prasanna Kannappan
Department of Mechanical Engineering
University of Delaware, USA

Herbert G. Tanner
Department of Mechanical Engineering
University of Delaware, USA

Art Trembanis
School of Marine Science and Policy
University of Delaware, USA

Justin H. Walker
Department of Geological Sciences
University of Delaware, USA

Abstract

INTRODUCTION

Marine Surveys using Underwater Robotic Platforms

Using underwater robotic platforms to perform optical imaging surveys offers high data densities for habitat assessment surveys. Though the large volume of image data (in the order of thousands to millions of images) helps to provide a detailed picture of the species habitat, they also present a challenge to data managers and marine scientists: they generate a need for extensive manpower and time to process the data. While technological developments in form of improvements in robotic platforms and image acquisition systems have enhanced our capabilities for observing and monitoring a species habitat, we still lack the required arsenal of data processing tools. This need motivates the development of automated tools to analyze benthic imagery data containing scallops.

In previous video-based scallop surveys [Rosenkranz et al., 2008] it is reported that the data processing time to review and analyze one hour of collected data is in the order of 4 to 10 hours. The literature also suggests that automated computer techniques would greatly benefit imaging surveys, but there is no available automated tools to this date. There has been anecdotal evidence of development of automated scallop assesment tools by HabCam group [Gallager et al., 2005], though there is no such tool available to the research community. Manual data processing time estimates from our own Autonomous Underwater Vehicle (AUV) data indicates that the counting can be performed at a rate of 2080 images/hour for scallops. If this were to be extended to all benthic-macro organisms, the rate would drop to 600 images/hr. Other scallop survey reports [Oremland et al., 2008] documents a manual processing rate of 1-10 hours per person per tow transect—there is no mention of the number of images per transect. According to this same reference, subsampling 1% of the images can reduce the processing time to approximately 1–2 hours per tow.

Vision-based Detection and Counting of Marine Creatures

Counting marine life using automated techniques have been tried on fish [Spampinato et al., 2008; Edgington et al., 2006; Williams et al., 2006] and aquaculture [Zion, 2012]. This work involves mostly the use of stationary cameras that detect the presence of species through background subtraction. Once the species is detected, techniques like contour matching are used to identify and count them. But when counting sedentary animals like scallops using a stationary camera does not make sense, and one would opt for moving robotic platforms with cameras. Background estimation and subtraction from images collected by a moving platform becomes problematic due to the rapidly changing background.

One approach to detect the presence of animals without prior knowledge of the background is finding points that are significantly different from the rest of the pixels in a image, i.e., singling out "anomalous" pixels. The regions around these pixels are most likely to contain objects, though not necessarily the object of interest. Further processing of these image regions is required to remove false positives. Still, such algorithms that can be used to detect points of interest can also be biased towards detecting specific types of objects to decrease false positives. Statistically, these points of interest can be thought of as sudden changes to the underlying distribution generating background pixels. Using mathematical approaches designed to detect a change in data distribution [Basseville and Nikiforov, 1993; Poor and Hadjiliadis, 2009] requires prior knowledge of the background distribution. In this application context, modeling this background distribution from noisy image data is inherently challenging, if not impossible.

Counting scallops on artificial scallop beds [Enomoto et al., 2009, 2010] would normally employ trained feature descriptors capable of detecting intricate fluted patterns on scallop shells. The problem in using such techniques on datasets collected in natural environments is that fluted pattens are typically indistinguishable in the presense of non-uniform lighting, high levels of speckle noise, and poor resolution due to images acquired from moving platform several meters away from the target. Yet, techniques that attempt to count scallops in natural environments do exist. Fearn et al. [2007] use machine learning methods coupled with Bottom-Up Visual Attention (BUVA) (for the latter, see [Itti et al., 1998] and refer to the next section for more detail). Interestingly, Fearn et al. [2007] do not use any ground truth to validate the results. At the same time, Dawkins [2011] and Einar Óli Guòmundsson [2012], present results obtained over a small set dataset of less than 100 images. It is thus not clear if these methods generalize over typical AUV missions containing thousands of images.

This work is the extension of the scallop counting work in [Kannappan et al., 2014; Kannappan and Tanner, 2013].

TECHNICAL BACKGROUND

Visual Attention

Visual attention is a neuro-physiologically inspired theory [Koch and Ullman, 1985] that tries to explain the cognitive mechanism behind human visual processing. According to this theory, the human visual system assigns priorities to regions of an acquired image before processing. The priorities are encoded in the form of special weights, also known as *salience* values of a pixel. A region with high salience value is processed first. These salience values are indicators of the degree of interest associated with a pixel, and can be used to identify sub-regions of an image containing visually distinct objects.

According to the model of visual attendion discussed in [Koch and Ullman, 1985], human visual system first generates a series of feature streams from an input image. These feature maps are then combines into a set of so called *center-surround* feature maps to highlight regions that are locally different in each of the feature streams. All the center-surround feature maps are then accumulated into a saliency map. The peak in the saliency maps are called fixation and correspond to points that are significantly different from the rest of the

image. These fixations are processed in the decreasing order of their salience values.

According to a known computational model for the visual attention [Itti et al., 1998], an image is first processed along three feature streams which are color, intensity, and orientation. The color stream is further divided into two sub-streams (red-green and blue-yellow) and the orientation stream into four sub-streams, for instance along directions $\theta \in \{0°, 45°, 90°, 135°\}$. The image information in each sub-stream is scaled by nine different factors, indexed in $0, \ldots, 8$. Then for each factor k, the image data is essentially scaled to size $\frac{1}{2^k}$, resulting in some loss of information as the scale increases. The resulting image data for each factor constitutes the *spatial scale* for the particular sub-stream. The sub-stream feature maps are then resized again to a uniform size and compared in order to isolate differences between them. These differences are introduced by the loss of information and the subsequent interpolation that takes place during resizing, which results in the resized feature maps not matching exactly across different scales, even when brought down to the same size. The *center-surround* operator $\ominus$ takes pixel-wise differences between resized sub-streams and exposes those mismatches. For the intensity stream, for example, the center-surround operator is being used in the form

$$I(c, s) = |I(c) \ominus I(s)| \quad , \tag{1}$$

where c and s are indices for two different spatial scales, with c ranging in $\{2, 3, 4\}$, and $s = c + \delta$, for $\delta \in \{3, 4\}$. Center-surround feature maps are thus computed for each sub-stream in color and orientation streams too.

A set of a total of 42 center-surround feature maps are generated from the seven sub-streams (two for color, one for intensity and four for orientation), The center-surround feature maps for each original stream (color, intensity, and orientation) are then combined back into three *conspicuity maps*: one for color $\bar{C}$, one for intensity $\bar{I}$, and one for orientation $\bar{O}$. For instance, the intensity conspicuity map is given as

$$\bar{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} w_{cs}\, \mathcal{N}(I(c, s)) \tag{2}$$

where $\oplus$ is a cross-scale operator that works in a fashion similar to $\ominus$; the difference here is that the data in resized maps from different scales is now pixel-wise added. The map normalization operator $\mathcal{N}(\cdot)$ in (2) scales a whole map, by multiplying the values in its argument with the scaling factor $(M - \bar{m})^2$, where M is the global maximum over the map and $\bar{m}$ is the mean over all local maxima present in the map. Conspicuity maps are then simply combined—no cross-scale operation is used at this stage—to get the *saliency map*

$$S = w_{\bar{I}}\, \mathcal{N}(\bar{I}) + w_{\bar{C}}\, \mathcal{N}(\bar{C}) + w_{\bar{O}}\, \mathcal{N}(\bar{O}) \quad , \tag{3}$$

where $w_{\bar{k}}$ is in general a user-selected stream-specific weight. In BUVA all streams are weighted equally, so $w_{\bar{I}} = w_{\bar{C}} = w_{\bar{O}} = 1$. On this saliency map, a winner-takes-all neural network is typically used [Itti et al., 1998; Walther and Koch, 2006] to compute the maxima—other methods are of course possible. When a local maximum is found, the focus of attention is supposedly shifted to this point, and the point becomes a visual fixation.

The weights in (2) and (3) can be selected judiciously to bias fixations toward specific targets of interest. The resulting variant of this method is known as Top-Down Visual Attention (TDVA) [Navalpakkam and Itti, 2006]. One method to select these weights is [Navalpakkam and Itti, 2006]:

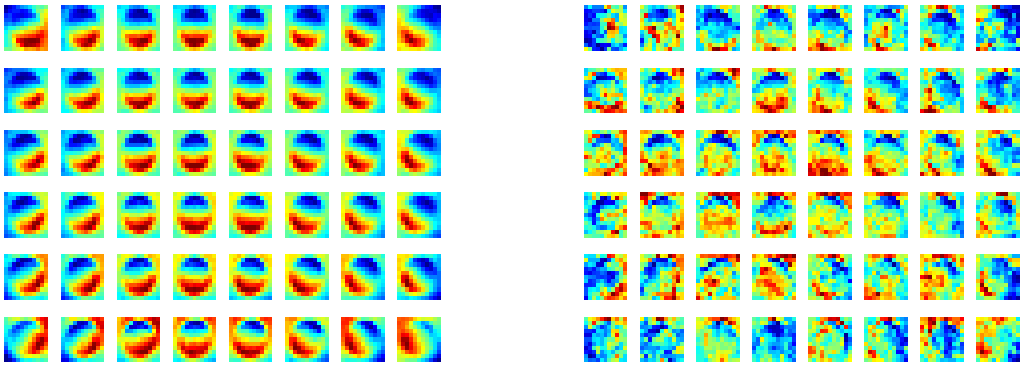
$$w_j = \frac{w'_j}{\frac{1}{N_m} \sum_{j=1}^{N_m} w'_j} \quad , \tag{4a}$$

where N_m is the number of feature (or conspicuity) maps, and

$$w'_j = \frac{\sum_{i=1}^{N} N_{iT}^{-1} \sum_{k=1}^{N_{iT}} P_{ijT_k}}{\sum_{i=1}^{N} N_{iD}^{-1} \sum_{k=1}^{N_{iD}} P_{ijD_k}} \quad , \tag{4b}$$

where N is the number of images in the learning set, N_{rT} and N_{rD} are the number of targets (scallops) and distractors (similar objects) in the r-th learning image, P_{uvT_z} is the mean salience value of the region around the v-th map containing the z-th target (T) in the u-th image. P_{uvD_z} is similarly defined for distractors (D).

Figure 1: Seabed image with scallops shown in circles



(a) (b)

Figure 2: (a) Mean map of scallops in each quadrant (b) Standard deviation map of scallops in each quadrant. Red corresponds to higher numeric values and blue correspond to lower numeric values.

MAIN FOCUS OF THE CHAPTER

[here is where you make a summary of the ASLO paper approach, note the successes and the limitations. You then use this assessment to justify the need for the fourth layer]

Issues, Controversies, Problems

SOLUTIONS AND RECOMMENDATIONS

Layer IV: False Positives Filter

To decrease the false positives that are produced in the classification layer, two methods were evaluated as possible candidates: a high-dimensional Weighted Correlation Template Matching (WCTM) technique and a Histogram of Gradients (HOG) method. The main objective here is to find a method that will retain a high percentage of true positive scallop and at the same time eliminate false positives from the classification layer.

High-dimensional weighted correlation template matching

In this method, the templates used are generated from scallop images that are not preprocessed, i.e., images that are not median-filtered in the first three layers. The intuition behind this is that although median filtering reduces speckle noise and may improve the performance of segmentation, it also weakens the edges and gradients in an image. Avoiding median filtering helps to generate templates that are a little more accurate than the ones already used in the classification layer.

Based on the observation that the scallop templates are dependent on their position in the image (see figure 2), a new scallop template is generated for each object coming from classification layer. Consider an object filtered down from the classification layer. It is represented by a 3-tuple (a_o, b_o, R_o), where a_o and b_o represent the x–y coordinates of object center and R_o corresponds to a radius. The representative scallop template is now generated from all scallops in the learning set (containing 3 706 scallops) with their centers lying within a 40×40 window in the neighborhood of the object center (a_o, b_o). Each of these scallops is cropped using a square window of size $2.5R \times 2.5R$ (R is the scallop radius). Since each of these scallops is of different dimensions, it is resized to a window of size $2.5R_o \times 2.5R_o$. All the learning scallop instances are finally combined through a pixel-wise mean to obtain the mean representative template. Similarly, a standard deviation map that captures the standard deviation of each pixel in the mean template is also obtained. The templates produced here are of larger size compared to the templates in Layer III (a Layer III template is of size 11×11). These larger templates are more accurate since they hold more scallop information.

The templates and the object pixels are first processed through normalization and mean subtraction before comparison. If template and object are represented by vectors $X^t = (X_1^t, X_2^t, \ldots, X_k^t)$ and $X^u = (X_1^u, \ldots, X_v^u)$, respectively, where $v = (2.5R_o)^2$ is the total number of pixels in both the template and the object, then template normalization for the p^{th} component of the vector is given by

$$X_p^{\bar{t'}} = \min_k X_k^u + \left(\frac{\max_k X_k^u - \min_k X_k^u}{\max_k X_k^{t'} - \min_k X_k^{t'}} \right) \left[X_p^{t'} - \min_k X_k^{t'} \right] .$$

Normalization is followed by mean subtraction of the template. The resulting mean subtracted template $X^{\bar{t}}$ is computed as

$$X_p^{\bar{t}} = X_p^{t'} - \frac{1}{v} \sum_{k=1}^{v} X_k^{t'}$$

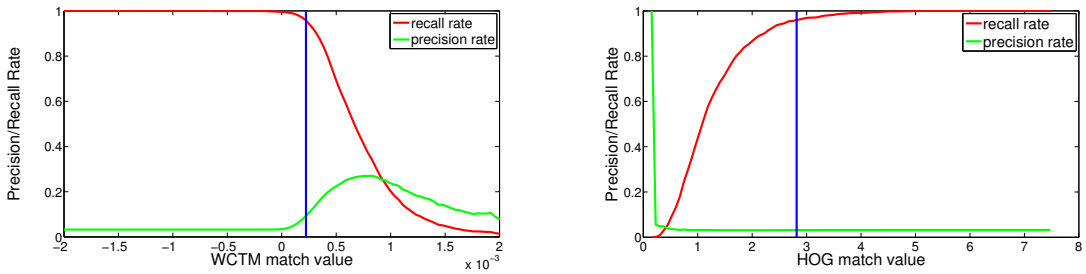
The mean subtracted object vector $X^{\bar{u}}$ is also similarly obtained. The standard deviation map is also normalized.

$$\bar{\sigma}_p = \frac{\sigma_p}{\sum_{k=1}^{v} \sigma_k} .$$

The mean subtracted template and the corresponding object vector are compared using a correlation metric weighted by the variance of each feature. The higher the value of this metric, the better the match between the object and the template. The WCTM similarity measure is given by

$$D_{\mathsf{wctm}} = \prod_{k=1}^{v} \frac{X_k^{\bar{t}} X_k^{\bar{u}}}{\bar{\sigma}_k} .$$

The threshold for the weighted correlation metric D_{wctm} is set as 0.0002222, i.e., any object with a similarity score lower than this threshold is rejected. This threshold value is justified from the precision-recall curves (see Figure 3(a)) of the weighted correlation metric values for the objects filtering down from the classification layer. The threshold shown by the blue line corresponds to 96% recall rate, i.e., 96% of the true positive scallops from the classification layer pass through WCTM. At the same time, the WCTM decreases the false positives by over 63%.



(a) (b)

Figure 3: Precision recall curve for layer IV candidate methods (a) WCTM and (b) HOG. The blue line marks thresholds $D_{\mathsf{wctm}} = 0.0002222$ and $D_{\mathsf{hog}} = 2.816$. It is important to note that WCTM is a similarity measure and HOG is a dissimialrity measaure. This implies that only instances below the indicated threshold D_{wctm} in WCTM and likewise instances above the threshold D_{hog} in HOG are rejected as false positives.

Histogram of Gradients

The Histogram Of Gradients feature descriptor encodes an object by capturing a series of local gradients in neighborhood of the object pixels. These gradients are then transformed into a histogram after discretization and normalization. There are several variants of HOG feature descriptors. The R-HOG used for human detection in [Dalal and Triggs, 2005] was tested here as a possible Layer IV candidate.

According to R-HOG, the image is first tiled into a series of 8×8 pixel groups referred to here as cells (the image dimensions need to be multiples of 8). The cells are further divided into a series of overlapping blocks each containing groups of 2×2 cells. For each cell a set of 64 gradient vectors (one per pixel) is computed. Each gradient vector contains a direction and magnitude component. In the gradient directions, the sign is ignored reducing the range to 0-180 from 0–360. The gradient vectors are then binned into a 9–bin histogram ranging from 0–180 degrees with a bin width of 20 degrees. The contribution of each gradient vector is computed as half its gradient magnitude. The other half of the gradient magnitude is split between the two neighboring bins (in case of boundary bins, the neighbors are determined by wrapping around the histogram). The histograms from the 4 cells in each block is then concatenated to get vector v of 36 values (9 per cell). These vectors from each block are then normalized using an L_2-norm. The normalized vector $\bar{v}$ from each block is concatenated into a single feature vector F to get the HOG descriptor for the input image. The expression for L_2 norm computation is shown below,

$$\bar{v} = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}}$$

where ϵ is a small constant (here $\epsilon = 0.01$).

Since this method imposes a constraint on the image dimensions being multiples of 8, the learning samples (each cropped using a square window of size of $3 \times$ radius) is resized to 24×24. The object samples used here contain both positive scallop instances and negative instances obtained from other objects detected by the segmentation layer. A HOG feature vector F of length 144 (4 blocks $\times$ 4 cells $\times$ 9 values) is computed for each object instance obtained from the classification layer.

Any machine learning method can be applied using the positive and negative object instances as learning samples. As per the original implementation in [Dalal and Triggs, 2005], an Support Vector Machine (SVM) was tested. The SVM learning algorithm failed to converge even after a large number of iterations. This could be attributed to the fact that the scallop profiles vary significantly based on their position in the image. To overcome this problem, a lookup table similar to the one used to learn the scallop profiles is the classification layer is generated here. Only difference here is that instead of saving a reference scallop template vector, a reference HOG vector for only positive scallop instances from the learning set is recorded. The reference HOG

Table 1: Comparison of tested false positive filter layer methods

	HOG		WCTM	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2
True positives from Classification Layer	183	1759	183	1759
False positives from Classification Layer	7970	52456	7970	52456
True positives after Layer IV Method	179	1689	176	1685
False positives after Layer IV Method	7752	51329	2924	16407
Decrease in true positives after Layer IV	4 (2.2%)	70 (4%)	7 (3.8%)	74(4.2%)
Decrease in false positives after Layer IV	218 (2.7%)	1127 (2.1%)	5046 (63.3%)	36049 (68.7%)

descriptor for a pixel coordinate in the image is taken to be the mean of all the HOG descriptors of scallop instances inside a 40×40 window around the point.

For each instance classified as a scallop from the classification layer, its HOG descriptor is compared with its corresponding learned reference HOG descriptor from the lookup table. Since the HOG feature vectors are essentially histograms, the Earth Mover's Distance (EMD) metric [Rubner et al., 2000] used to measure the dissimilarity between histograms can be used. Technically, the EMD quantifies the difference between two histograms. Intuitively, if histogram bins are thought of as containers of dirt with the volume of dirt in each bin is analogous to the bin frequency, then the amount of work done moving piles of dirt to match a different histogram arrangement is equivalent to the EMD between the two histograms. The EMD measure D_{emd} between two histograms A and B is given by

$$D_{\mathsf{emd}}(A, B) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

where m and n are the number of bins in histograms A and B respectively, d_{ij} is a measure of the distance between bin i in A and bin j in B and f_{ij} is the optimal *flow* or the amount of material moved between the bins i and j that solves the problem. For more details about the optimization and linear programming involved in this algorithm refer to [Rubner et al., 2000].

A precision-recall curve (shown in Figure 3(b)) with the classification threshold set as 2.816 (which corresponds to 96% recall rate, same rate used to set the wctm threshold). Any object with EMD distance value less than this threshold is considered as a scallop. Though this threshold can capture 96% of the scallops, very few false positives get eliminated (less than 3%).

Results and Discussion

The three-layered detection approach with the addition of a fourth false positives filter layer was tested on two separate datasets containing 1 299 and 8 049 images respectively. Among the two candidate methods tested for the fourth layer, WCTM was chosen over HOG due to it superior performance. The difference in performance between HOG and WCTM for both the datasets can be seen in Table 1. Rows 1 and 2 show the true positives and false positives that are filtered down from the initial 3 layers (Layers I-III) respectively. With these values as the baseline, the thresholds for both HOG and WCTM were chosen to retain a high recall rate of close to 96%. This ensures that very few true positives are lost and their performance is primarily assessed through the reduction in false positives (row 6 of Table 1).

Since the thresholds are set such that the recall rate is high in both methods, the decrease in true positives is less than 5% in both HOG and WCTM. However there is a significant reduction in false positives (63.3% for dataset 1 and 68.7% for dataset 2) from WCTM. On the other hand, the decrease in false positive is relatively small (less than 3%) in HOG. It is not clear why the HOG filter fails to remove false positives. One

Table 2: Results of multi-layer scallop classification

	Dataset 1	Dataset 2
Number of images	1,299	8,049
Ground Truth Scallops	363	3,698
Valid Ground Truth Scallops	250	2,781
True Positives After Visual Attention Layer	231 (92.4%)	2,397 (86.2%)
True Positives After Segmentation Layer	185 (74%)	1,807 (64%)
True Positives After Classification Layer	183 (73%)	1,759 (63.2%)
True Positives After False Positive Filter Layer	176 (70.4%)	1,685 (60.6%)
False Positives After Classification Layer	7,970	52,456
False Positives After False Positives Filter Layer (WCTM)	2,924	16,407
Decrease in false positives (due to WCTM)	63.3%	68.7%

reason could be that the HOG filter derived from its native implementation for human detection in [Dalal and Triggs, 2005] might need further customization and even weighting through standard deviation weights like in WCTM. Further study and detailed analysis is required to improve its performance. These results led to the inclusion of WCTM as the false positive filter layer in the scallop counting process pipeline.

The overall performance of the four-layer pipeline is shown in Table 2. The results are compared to manually labeled ground truth. Only a subset of the available scallops, scallops at least 80 pixels horizontally and 60 pixels vertically away from the image boundaries were used as ground truth. This was done to leave out scallops near the boundaries that were affected by severe vignetting effects. Such scallops were often too dark (see Figure 1) and very difficult to correct using standard vignetting correction algorithms. Furthermore, the scallop templates for scallops near the boundaries are such that their prime feature, the dark crescents, blend into the dark borders of the image (see Figure 2(a)). Inclusion of the boundaries would cause almost any objects near the boundary to be classified as scallops, resulting in a large number of false positives. It is also interesting to note that scallops only partially visible near the image boundaries were excluded in the manual counts performed in [Walker, 2013].

Table 2 shows the results of the 3-layer pipeline along with the improvements in terms of the reduction in false positives as a result of introducing the fourth processing layer. The true positive percentages shown are computed with reference to the valid ground truth scallops (row 3 of table 2), i.e., scallops away from image boundaries. In dataset 1, which contains 1 299 images, the four-layer filtering results in a 70.4% overall recall rate, while in dataset 2 that contains 8 049 images the overall recall rate is 60.6%. Though the addition of the fourth false positive layer results in a small drop of 2.6% in recall rate, it eliminates over 63% of the false positives in both datasets. There is no clear reason to explain the better performance of this pipeline on dataset 2 both in terms of recall rate and decrease in false positives compared to dataset 1.

FUTURE RESEARCH DIRECTIONS

CONCLUSION

References

Basseville, M. and Nikiforov, I. V. (1993). *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, NJ.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition, 2005., volume 1, pages 886–893. IEEE.

Dawkins, M. (2011). Scallop detection in multiple maritime environments. Master's thesis, Rensselaer Polytechnic Institute.

Edgington, D. R., Cline, D. E., Davis, D., Kerkez, I., and Mariette, J. (2006). Detecting, tracking and classifying animals in underwater video. In *Oceans'06 MTS/IEEE-Boston Conference and Exhibition*, pages 1–5. IEEE.

Einar Óli Guòmundsson (2012). Detecting scallops in images from an auv.

Enomoto, K., Masashi, T., and Kuwahara, Y. (2010). Extraction method of scallop area in gravel seabed images for fishery investigation. *IEICE Transactions on Information and Systems*, 93(7):1754–1760.

Enomoto, K., Toda, M., and Kuwahara, Y. (2009). Scallop detection from sand-seabed images for fishery investigation. In *2nd International Congress on Image and Signal Processing*, pages 1–5. IEEE.

Fearn, R., Williams, R., Cameron-Jones, M., Harrington, J., and Semmens, J. (2007). Automated intelligent abundance analysis of scallop survey video footage. *AI 2007: Advances in Artificial Intelligence*, pages 549–558.

Gallager, S., Singh, H., Tiwari, S., Howland, J., Rago, P., Overholtz, W., Taylor, R., and Vine, N. (2005). High resolution underwater imaging and image processing for identifying essential fish habitat. In Somerton, D. and Glentdill, C., editors, *Report of the National Marine Fisheries Service Workshop on Underwater Video analysis*, NOAA Technical Memorandum NMFS-F/SPO-68, pages 44–54.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

Kannappan, P. and Tanner, H. G. (2013). Automated detection of scallops in their natural environment. In *21st Mediterranean Conference on Control and Automation*, pages 1350–1355. IEEE.

Kannappan, P., Walker, J. H., Trembanis, A., and Tanner, H. G. (2014). Identifying sea scallops from benthic camera images. *Limnology and Oceanography: Methods*, 12(10):680–693.

Koch, C. and Ullman, S. (1985). Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227.

Navalpakkam, V. and Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimizing detection speed. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2049–2056. IEEE.

Oremland, L., Hart, D., Jacobson, L., Gallager, S., York, A., Taylor, R., and Vine, N. (2008). Sea scallop surveys in the 21st century: Could advanced optical technologies ultimately replace the dredge-based survey? Presentation made to the NOAA Office of Science and Technology.

Poor, H. V. and Hadjiliadis, O. (2009). *Quickest detection*. Cambridge University Press.

Rosenkranz, G. E., Gallager, S. M., Shepard, R. W., and Blakeslee, M. (2008). Development of a high-speed, megapixel benthic imaging system for coastal fisheries research in alaska. *Fisheries Research*, 92(2):340–344.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

Spampinato, C., Chen-Burger, Y.-H., Nadarajan, G., and Fisher, R. B. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *3rd International Conference on Computer Vision Theory and Applications*, pages 514–519. Citeseer.

Walker, J. (2013). Abundance and size of the sea scallop population in the mid-atlantic bight. Master's thesis, University of Delaware.

Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407.

Williams, R., Lambert, T., Kelsall, A., and Pauly, T. (2006). Detecting marine animals in underwater video: Let's start with salmon. In *Americas Conference on Information Systems*, volume 1, pages 1482–1490.

Zion, B. (2012). The use of computer vision technologies in aquaculturea review. *Computers and Electronics in Agriculture*, 88:125–132.