

Research Article

Sensor Selection and Integration to Improve Video Segmentation in Complex Environments

Adam R. Reckley,¹ Wei-Wen Hsu,¹ Chung-Hao Chen,¹ Gangfeng Ma,¹ and E-Wen Huang²

¹ Old Dominion University, Norfolk, VA 23529, USA

² National Central University, Jhongli 32001, Taiwan

Correspondence should be addressed to Chung-Hao Chen; cxchen@odu.edu

Received 1 November 2013; Accepted 28 December 2013; Published 9 February 2014

Academic Editor: Yi-Hung Liu

Copyright © 2014 Adam R. Reckley et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Background subtraction is often considered to be a required stage of any video surveillance system being used to detect objects in a single frame and/or track objects across multiple frames in a video sequence. Most current state-of-the-art techniques for object detection and tracking utilize some form of background subtraction that involves developing a model of the background at a pixel, region, or frame level and designating any elements that deviate from the background model as foreground. However, most existing approaches are capable of segmenting a number of distinct components but unable to distinguish between the desired object of interest and complex, dynamic background such as moving water and high reflections. In this paper, we propose a technique to integrate spatiotemporal signatures of an object of interest from different sensing modalities into a video segmentation method in order to improve object detection and tracking in dynamic, complex scenes. Our proposed algorithm utilizes the dynamic interaction information between the object of interest and background to differentiate between mistakenly segmented components and the desired component. Experimental results on two complex data sets demonstrate that our proposed technique significantly improves the accuracy and utility of state-of-the-art video segmentation technique.

1. Introduction

Background subtraction is often considered to be a key part of any video surveillance system being used to detect objects in a single frame and/or track objects across multiple frames in a video sequence. Most of the current state-of-the-art techniques for object detection and tracking utilize some form of background subtraction that involves developing a model of the background at a pixel, region, or frame level and designating any elements that deviate from the background model as foreground. Robust object detection and tracking algorithms must be able to maintain satisfactory performance in dynamic backgrounds where the background of the image is in motion such as rippling water and illumination fluctuations. Although there are many proposed methods for object detection, a number of them achieving moderate success in dynamic backgrounds, most current state-of-the-art techniques treat objects of interest and background as separate entities ignoring any interaction. If the background of a scene

is defined as any object other than the object of interest, there are many situations when background motion affects the motion of objects of interest. For instance, the motion of an object in the ocean is going to be strongly influenced by wave action. Another example, if the object of interest is a specific vehicle, any vehicles in front of the object of interest that slow or stop will cause the object of interest to begin to slow or slow to a stop. In this paper, we hypothesize that the dynamic interaction between an object of interest and the background can provide useful information and by understanding and modeling the dynamic interaction between an object of interest and background, we can improve the performance of state-of-the-art object detection techniques. In order to test our hypothesis, we introduce heterogeneous sensing modalities to first model the dynamic interaction between an object of interest and background. Then, we select the best quality of information in terms of relevant parameters and dynamically assessing these parameters in a multisensor setting to be integrated image segmentation process.

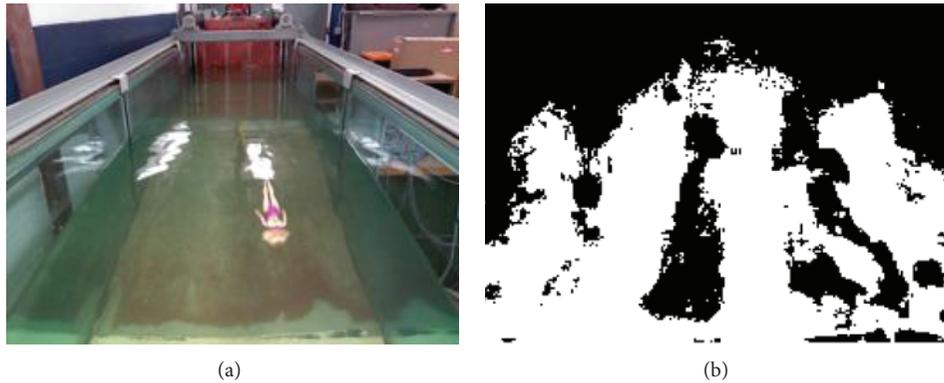


FIGURE 1: Example of SCS-LBP algorithm performed on our data set. Image on the left is the original image and the image on the right is the result from our implementation of the SCS-LBP algorithm.

The experimental results demonstrate convincing evidence that the dynamic interaction between an object of interest and background provide valuable information that can be utilized to improve segmentation results. The structure of this paper is organized as follows: Section 3 shows the related work. The sensor selection and integration is introduced in Section 4. Section 5 illustrates the segmentation approach. Section 6 shows experimental results. Section 7 concludes this paper.

2. Motivation

At the outset of this process, we sought to detect and track an object of interest in a complex, dynamic environment. As described previously, there have been many proposed methods for video segmentation that attempt to segment objects from video sequences of dynamic environments. We chose one state-of-the-art video segmentation technique to perform segmentation on our data set of image sequences. The method that we chose to use was the spatial extended center-symmetric local binary pattern (SCS-LBP) proposed by Xue et al. [1]. When the method was utilized for segmentation on our complex data set the performance was inadequate for executing object detection and tracking. In our dataset, a water tank is used to generate dynamic backgrounds, breaking waves, high reflectance, and inconsistent motion of the object. An example of the results from the implementation by Xue et al. [1] is shown in Figure 1. As a result, Xue's method is unable to handle this case.

3. Related Work

Due to its importance in nearly all video segmentation contexts, background subtraction is a widely studied topic in computer vision and has had an abundance of literature published to its effect including surveys and evaluations of the most current and prevalent techniques [2–7]. In its most general form, background subtraction involves first creating an unambiguous model of the background in the image. Once the background model is built, the subsequent incoming frames are compared to the background model with any pixels that differ from the background model by

more than a certain threshold determined to be foreground objects. Using the determination of pixels as foreground or background, a binary foreground mask is created, completing the basic background subtraction. Robust background subtraction algorithms for real-world applications (i.e., video surveillance) must be capable of performing in dynamic environments. Rarely, in real world applications the background can be expected to remain static over an entire video sequence. The large majority of techniques presented for background subtraction focus on differing schemes for modeling the background and processes to improve current background modeling methods. In the following subsections, some popular state-of-the-art techniques are discussed.

3.1. Gaussian Mixture Model Background Subtraction. Many of the proposed techniques for background subtraction exploit the Gaussian probability density function to model the background. The parametric method proposed by Wren et al. [8] models each pixel with a single Gaussian distribution. Parametric methods assume an underlying distribution and use a set of training images that do not contain any objects of interest to estimate parameters of the underlying distribution for background modeling. Wren et al. [8] utilize the Gaussian distribution and estimate the mean background color and the covariance for each pixel in a frame. The single modal background model only provides satisfactory results when the camera and background are both static. With the exception of the simplest cases in computer vision, the assumption that both the camera and background are static is unrealistic and reduces the utility of the method in the majority of real-life situations. In order to account for slow variations in the background of an image, Stauffer and Grimson [9] proposed modeling each pixel with a mixture of Gaussians to build a background model of a sequence of images. For the purpose of this paper, slow variations in the background or slowly moving background objects can be defined as background objects whose movement in the video sequence is slower than the movement of any objects of interest in the video sequence. Using the persistence and variation of each of the Gaussians in the mixture, the Gaussians that constitute the background are determined.

Any pixel values that do not fit within any of the Gaussians that are considered part of the background are resolved to be part of a foreground object. Pixel values that are determined to be a part of the foreground are combined using connected components. The system proposed by Stauffer and Grimson [9] uses a set of K Gaussians in the mixture for modeling the background that are continuously updated based on their accuracy in modeling the background.

Due to the fact that a set number (K) of Gaussians are used in the mixture for modeling the background, during certain images, the K Gaussians may not be sufficient to automatically fully adapt the background model to the scene [10]. Zivkovic and van der Heijden [10, 11] proposed an improvement to the Gaussian mixture model to provide for the required number of components necessary to be calculated for each pixel, to allow for full adaptation to the observed scene. By using a recursive function with the weight values for each Gaussian in the mixture, Zivkovic and van der Heijden [10, 11] calculate the necessary number of components for each pixel increasing the efficiency of the system by reducing unnecessary components in the mixture of Gaussians and allowing for full adaptation of the model to the observed scene. Another proposed improvement to the Gaussian mixture model came from Lee [12] who proposed a scheme to improve the convergence rate of the Gaussian mixture model without compromising the model stability. The improved convergence rate of the Gaussian mixture model reduces the time and number of images necessary for training the algorithm. However, even with the improvements proposed by Zivkovic and van der Heijden [10, 11] and Lee [12], assuming that the pixel intensity distribution follows a Gaussian distribution may be inaccurate in dynamic scenes causing the method to fail, as stated by Xue et al. in [1].

3.2. Nonparametric Background Subtraction. In addition to the Gaussian mixture model for background subtraction, many authors proposed nonparametric methods for modeling the background to perform background subtraction. Nonparametric methods make no assumption of the underlying distribution in each pixel, instead relying on previous samples from the data to perform the background modeling. Elgammal et al. [13, 14] propose a kernel density estimation algorithm that models the background by estimating the probability of observing pixel intensity values based on a sample of previous intensity values for each pixel. Essentially, the algorithm is “estimating the probability density function by averaging the effect of a set of kernel function centered at each data point” [13]. Once the background model is estimated, background subtraction is performed. After background subtraction is completed to locate foreground objects, Elgammal et al. [13, 14] build a representation of the foreground areas to aid in the tracking of the objects and resolving any object occlusion. The kernel density estimation technique proposed by Elgammal et al. [13, 14] performs satisfactorily when the background has slow moving variation, but performance declines with the introduction of significant background movement [15]. Sheikh and Shah [16] proposed a technique for improving the performance of the

kernel density estimation. Rather than treating each image pixel as an independent random variable, Sheikh and Shah [16] contend that useful correlation can be found in pixel intensities over spatially proximal pixels that can be exploited in order to maintain accuracy over increasingly dynamic background. Additionally, Sheikh and Shah [16] propose maintaining joint background and foreground models of each pixel that can be used competitively in a maximum a posteriori probability estimation of a Markov random field (MAP-MRF) decision framework to increase the accuracy of the foreground segmentation. Even with maintaining models for both background and foreground objects to improve segmentation performance, the method proposed by Sheikh and Shah [16] requires that foreground objects have faster movement than any of the background objects [17].

Kim et al. [18, 19] propose a different nonparametric technique for background subtraction that models the background using a quantization/clustering technique. The method proposed by Kim et al. [18, 19] takes data samples at each pixel and clusters them into a set of codewords, specifically focusing on the color and brightness information. The background subtraction is performed by calculating the color distortion of the incoming pixel from the nearest cluster. If an incoming pixel has color distortion to a codeword that is less than a set detection threshold and its brightness is within the range of that codeword, the incoming pixel is classified as background. Otherwise, the incoming pixel is classified as foreground. Kim et al. [18, 19] include adaptive background model updating during illumination changes to increase performance during slowly moving background changes. However, the method is susceptible to problems when permanent structural changes occur in the background of the image due to the fact that the codeword update method does not allow for the creation of new codewords [20].

Another popular, state-of-the-art nonparametric technique, proposed by Barnich and van Droogenbroeck [20, 21], is called the visual background extractor (VIBE). Similar to other nonparametric methods, VIBE builds an estimation of the background model using data samples. However, unlike other methods, VIBE uses a random policy to select which values to include in the estimation to build the background. Barnich and van Droogenbroeck's [20, 21] random policy method in VIBE creates a smooth exponentially decaying lifespan for the values that constitute the pixel models. Additionally, the VIBE method only requires one frame to estimate the background model and can begin detecting foreground objects in the second frame. Unlike the Gaussian mixture model and kernel density estimation techniques, which update the global probability density function or estimation with each pixel value, the VIBE technique only allows the incoming pixel values to have local influence over neighboring pixels. One of the major shortcomings with the VIBE algorithm is that it has a high false detection rate [22]. The high false detection rate is due to the fact that if there are any moving objects in the initial frame from which the model is built, the moving objects will be incorporated into the background model creating a ghost region in subsequent frames [23]. Li et al. [22] proposed an improvement to the VIBE algorithm to address the problem of high false detection

rates. Li et al. [22] improve the VIBE algorithm using an adjacent frame difference algorithm which takes into account the time domain correlation between frames prior to the video. Using the time domain correlation, the improved VIBE algorithm aims to quickly remove the ghost regions in the model. The VIBE algorithm updates the model over time and would eventually eliminate the ghost, but the aim of Li et al. [22] is to remove the ghost much more quickly. Another improvement to the VIBE algorithm was proposed by van Droogenbroeck and Paquot in [24]. van Droogenbroeck and Paquot [24] improved the VIBE technique by removing foreground blobs with areas smaller or equal to 10 pixels and filling holes in the foreground objects with an area smaller or equal to 20 pixels. Additionally, van Droogenbroeck and Paquot [24] utilized the distance measure proposed by Kim et al. [19] to calculate a color distortion to improve performance of the system, upgrading from the simpler Euclidean distance measure that was used previously in VIBE. Using the new color distortion measure in conjunction with an adaptive threshold, van Droogenbroeck and Paquot [24] were able to significantly improve the performance of VIBE. However, the VIBE technique fails to reach the same level of performance on dynamic backgrounds as the Gaussian models described earlier [4]. Recall that Gaussian mixture models could perform well on dynamic background with only slowly moving background objects.

3.3. Local Binary Pattern Background Subtraction. With the exception of the VIBE algorithm, all the techniques described in the previous sections update the background model over the global probability density function or estimation. However, even the VIBE algorithm, as well as all other previously described methods, treats each incoming pixel value as independent for the purpose of creating a background model. Based on these shortcomings, the background of the image must be assumed to be static or nearly static with only slowly moving background objects for these methods to perform adequately, which prevents them from working well when attempting to detect moving objects in dynamic scenes. Comparable to the idea in the VIBE algorithm that each pixel should only affect its neighbors, Heikkilä and Pietikäinen [25] proposed a texture-based method that models each pixel as a group of adaptive local binary pattern (LBP) histograms. The local binary patterns are calculated by thresholding a P number of neighbors of a center pixel with the result being the binary pattern. This is performed for each pixel in a structure element. All of the binary patterns are placed together to form a LBP histogram for the center pixel. Background subtraction is performed by comparing the histogram for the incoming pixel against the background histograms using a histogram intersection proximity measure. If the proximity is calculated to be higher than a user-defined threshold for at least one of the background histograms, the pixel is classified as background. Otherwise, the pixel is classified as foreground.

Although the local binary pattern method proposed by Heikkilä and Pietikäinen [25] is robust to monotonic gray-scale changes and very fast to compute [25], it still does not

work adequately in dynamic scenes with intense background variations [26]. In order to address this shortcoming, Zhang et al. [26] offered an improvement to the local binary pattern technique that extended the local binary patterns from the spatial domain to the spatiotemporal domain and included an online dynamic texture extraction operator. The results from Heikkilä and Pietikäinen [25] showed improved results over the mixture of Gaussian method and kernel density estimation method; however a direct comparison to the original local binary pattern was never presented in [25]. Additionally, the inclusion of temporal information in the local binary pattern caused the method proposed by Heikkilä and Pietikäinen [25] to increase the computational load. However, even with the improvement of Heikkilä and Pietikäinen [25] to make LBP robust against local illumination changes, it struggles with uniform foreground objects in a uniform background [27].

Another improvement to the local binary pattern was introduced by Heikkilä et al. in [28]. Heikkilä et al. [28] present a new texture feature that simplifies the original local binary pattern by modifying the scheme used for comparing the neighboring pixels with the center pixel. Rather than comparing each neighboring pixel with the center pixel, Heikkilä et al. [28] compare center-symmetric pairs of pixels reducing the number of comparisons by half. Heikkilä et al. [28] incorporate their new texture feature, called center-symmetric local binary pattern (CS-LBP), into the SIFT descriptor to improve performance. The SIFT descriptor originally proposed by Lowe [29] is based on the idea that the appearance of an interest region can be characterized by the distribution of its local features. The SIFT descriptor is a 3D histogram that uses gradient as the local feature. The proposed CS-LBP feature reduced the number of required histograms increasing the computational simplicity, while providing a tolerance to illumination changes and is more robust to noise than the original local binary pattern. Although Heikkilä et al. [28] improved the LBP pattern; their improvements only utilized spatial information without taking into account temporal information [1]. Xue et al. [1] extended the CS-LBP operator from the spatial domain to the spatiotemporal domain which was designated spatial extended center-symmetric local binary pattern (SCS-LBP). SCS-LBP is capable of extracting spatial and temporal information simultaneously increasing accuracy of detection in dynamic scenes while sustaining low computational complexity by utilizing the center-symmetric scheme. A limitation of the local binary pattern and its subsequent improvements is that they are not efficient when handling large flat regions in an image (i.e., sky) due to the fact that the gray values of the neighboring pixels are very close to the value of the center pixel [25]. Chua et al. [27] proposed yet another improvement to rectify this limitation using local color features that can be represented as a local color pattern (LCP). Chua et al. [27] presented a technique that incorporates both the local binary pattern and the local color pattern to handle both texture rich areas for which the local binary pattern is effective and uniform regions where local color pattern is more effective. Using a fuzzy rule-based system, a weight is assigned and updated to the color and

texture features based on a pixel's local properties, specifically the current pixel's texture similarity score, uniformity of the binary pattern, color similarity score, and saturation value. The assigned weights are then utilized to select which features (color or texture) should be used for modeling the background at each location.

Based on the accuracy data and experimental images provided by Chua et al. [27] over nine video sequences, the method proposed by Chua et al. [27] appears to be one of the most accurate methods that were discovered during our research. Therefore, for the purpose of our experimentation, we chose to use the method proposed by Chua et al. [27] as the base method that we sought to improve.

4. Sensor Selection and Integration

Measuring the sensor belief has been an interesting research topic due to the uncertainty and imprecision involved in the sensor-based information gathering. To determine the sensor belief, the works in [18] have found out the rate of change in successive measurements from the sensor and argued that the greater the rate of change, the lower the belief. The rate of change is obtained based on the past data, the writers have defined some fuzzy rule sets to determine the self-belief of the sensors. Elgammal et al. [13] compared the performance of one sensor with another and derived a model for calculating the belief of the sensors. The performance of a sensor is determined based on the current detection outcome that supports an activity. The evidence from multiple sensors that support an activity from an abstract level is used to derive the belief value. Brutzer et al. [4] propose a dynamic belief calculation approach in the framework of a multimedia surveillance system. Using this mechanism, the belief of a set of nontrusted sensory streams evolves based on their association with other trusted streams. However, it is apparent that the determination of trusted streams would require certain precomputation, which might cause some overhead in obtaining the overall belief of the sensors and the information of interest.

4.1. Sensing Model and Measure of Uncertainty. Estimation problem is clarified using standard estimation theory. The time-dependent measurement, $\mathbf{z}_i(t)$, of sensor i with characteristics, $\lambda_i(t)$, is related to the parameters, $\mathbf{x}(t)$, that we wish to estimate through the following observation model [19]:

$$\mathbf{z}_i(t) = \mathbf{h}(\mathbf{x}(t), \lambda_i(t)), \quad (1)$$

where \mathbf{h} is a (possibly nonlinear) function depending on $\mathbf{x}(t)$ and parameterized by $\lambda_i(t)$, which represents the (possibly time dependent) knowledge about sensor i . Typical characteristics, $\lambda_i(t)$, about sensor i include sensing modality, which refers to what kind of sensor i is, sensor position \mathbf{x}_i , and other parameters, such as the noise model of sensor i and node power reserve. In (1), we consider a general form of the observation model that accounts for possibly non-linear relations between the sensor type, sensor position, noise model, and the parameters we wish to estimate. A special case of (1) would be $\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{f}_i(\mathbf{x}(t)) + \mathbf{w}_i(t)$, where \mathbf{f}_i

is an observation function and \mathbf{w}_i is additive, zero mean noise with known covariance. In case \mathbf{f}_i is a linear function on the parameters, (1) reduces to the linear equation:

$$\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{H}_i(t) \mathbf{x}(t) + \mathbf{w}_i(t). \quad (2)$$

In order to illustrate our technique, we will later consider the problem of stationary target localization with stationary sensor characteristics. Here, we assume that all sensors are acoustic sensors measuring only the amplitude of the sound signal so that the parameter vector $\mathbf{x} = [x, y]^T$ is the unknown target position, and

$$\lambda_i = [\mathbf{x}_i, \sigma_i^2]^T, \quad (3)$$

where \mathbf{x}_i is the known sensor position and σ_i^2 is the known additive noise variance. Note there is no longer a time dependence for \mathbf{x} and λ_i . Assuming that acoustic signals propagate isotropically, the parameters are related to the measurements by

$$\mathbf{z}_i = \frac{a}{\|\mathbf{x}_i - \mathbf{x}\|^{\alpha/2}} + \mathbf{w}_i, \quad (4)$$

where a is a given random variable representing the amplitude of the target, α is a known attenuation coefficient, and $\|\mathbf{x}_i - \mathbf{x}\|$ is the Euclidean norm. \mathbf{w}_i is a zero mean Gaussian random variable with variance σ_i^2 .

In the remainder of this paper, we define the *belief* as a representation of the current *a posteriori* distribution of \mathbf{x} given measurements $\mathbf{z}_1, \dots, \mathbf{z}_N : p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N)$.

Typically, the expectation value of this distribution $\bar{\mathbf{x}} = \int \mathbf{x} p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N) d\mathbf{x}$ is considered the estimate (i.e., the minimum mean square estimate), and we approximate the residual uncertainty by the covariance: $\Sigma = \int (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_N) d\mathbf{x}$.

In order to calculate the belief based on measurements from several sensors, we must pay a cost for communicating that information. Thus, maintaining what information each sensor node has about other sensor nodes is an important decision. This is why the sensor characteristics $\lambda_i(t)$ are clearly represented because it is important to know what information is available for various information processing tasks. Since combining measurements into the belief are now assigned costs, the problem is to intelligently choose a subset of sensor measurements which provide "good" information for constructing a belief state as well as minimizing the cost of having to communicate sensor measurements to a single node. In order to choose sensors to provide "good" updates to the belief state, it is essential to understand a measure of the information.

4.2. Sensor Selection. Given the current belief state, we wish to incrementally update the belief by incorporating measurements of other nearby sensors. Among all available sensors in the network, however, not all provide useful information that improves the estimate. Furthermore, some information might be useful but redundant. The task is to select an optimal subset and to decide on an optimal order of how

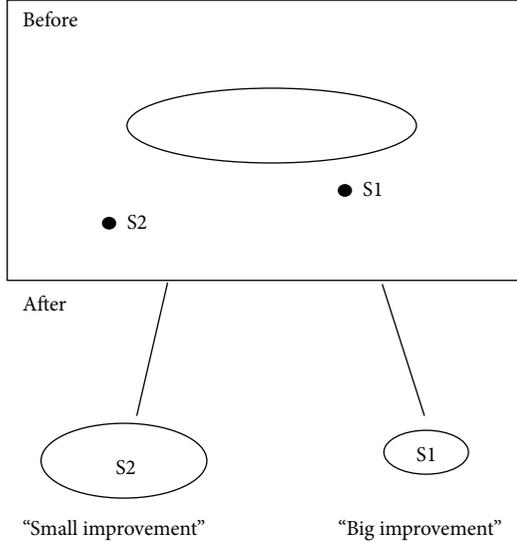


FIGURE 2: Sensor selection based on information gain of individual sensor contributions.

to incorporate these measurements into our belief update. Due to the distributed nature of the sensor network, this selection has to be done without explicit knowledge of the measurement residing at each individual sensor to avoid communicating less useful information. Hence, the decision has to be made solely based upon the sensor characteristics such as the sensor position or sensing modality and the predicted contribution of these sensors. Figure 2 shows the basic idea of optimal sensor selection. The image is based upon the assumption that estimation uncertainty can be effectively approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space. In Figure 1, the solid ellipsoid indicates the belief state at time t and the dashed ellipsoids are the incrementally updated belief after incorporating an additional measurement from a sensor, S1 or S2, at the next time step.

Although in both cases, S1 and S2, the area of high uncertainty is reduced by the same amount, the residual uncertainty in the case of S2 maintains the longest principal axis of the distribution. Based on the underlying measurement task, we will choose case S1 over S2.

4.3. Measures on Expected Posterior Distribution. It is essential to define a measure of information utility to quantify the information gain provided by a sensor measurement. We want to show that information content is inversely related to the “size” of the high probability uncertainty region. We first introduce an information-theoretic definition of the utility measure. There are many kinds of measuring methods (*Covariance-Based, Fischer Information Matrix, Entropy of Estimation Uncertainty, Volume of High Probability Region, and Sensor Geometry Based Measures*) [19]. In this paper we only describe “*Expected Posterior Distribution measures*” [12] that prove to be practically useful. Our objective is to predict the information utility of a piece of nonlocal sensor data before obtaining the data. In practice, the prediction must

be based on the currently available information: the current belief state and the characteristics of the sensor of interest which includes information such as the sensor position and sensing modality that can be established beforehand. We assume there are N sensors labeled from 1 to N and the corresponding measurements of the sensors are $\mathbf{z}_1, \dots, \mathbf{z}_N$. Let $U \subset \{1, \dots, N\}$ be the set of sensors whose measurements have been incorporated into the belief. That is, the current belief is $p(\mathbf{x} | \{\mathbf{z}_i\}_{i \in U})$. The sensor selection task is to choose a sensor whose data has not been incorporated into the belief yet and which provides the most information. To be specific, let us define an information utility function φ_{Utility} that assigns a value to each probability distribution. In this case, we ignore the cost term in the objective function. The best sensor, defined by the earlier objective function, is given by

$$\hat{j} = \arg \max_{j \in V} \varphi_{\text{Utility}} \left(p(\mathbf{x} | \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}) \right), \quad (5)$$

where V is the set of sensors whose measurements are potentially useful. The idea of using expected posterior distribution is to predict what the new belief state (posterior distribution) would look like if a simulated measurement of a sensor from the current belief state is incorporated. The utility of each sensor can then be quantified by the entropy or other measures on the new distribution from the simulated measurement. We use the tracking problem to derive an algorithm for evaluating the expected utility of a sensor. When a real new measurement is available, the new belief or posterior is evaluated using the familiar sequential Bayesian filtering [12]:

$$p(\mathbf{x}^{(t+1)} | \mathbf{z}^{\overline{(t+1)}}) = C \cdot p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \cdot \int p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}) \cdot p(\mathbf{x}^{(t)} | \mathbf{z}^{\overline{(t)}}) d\mathbf{x}^{(t)}, \quad (6)$$

where $p(\mathbf{x}^{(t)} | \mathbf{z}^{\overline{(t)}})$ is the current belief given a history of the measurement up to time t : $\mathbf{z}^{\overline{(t)}} = \{\mathbf{z}^{(0)}, \dots, \mathbf{z}^{(t)}\}$, $p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)})$ specifies the predefined dynamics model, $p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)})$ is the likelihood function from the measurement of sensor j , and C is a normalization constant. How do we compute the expected value of $p(\mathbf{x}^{(t+1)} | \mathbf{z}^{\overline{(t+1)}})$ without having the data $\mathbf{z}_j^{(t+1)}$ in the first place? The idea is to guess the shape of likelihood function from the current belief and the sensor position.

Without loss of generality, the current belief is represented by a discrete set of samples on a grid of the state space. This nonparametric representation of the belief state allows to represent highly non-Gaussian distribution and nonlinear dynamics. Figure 3 shows an example of the grid-based state representation. The gray squares represent the likely position of the target as specified by the current belief. The brighter the square, the more likely the target is there. For a sensor i , given the observation model $\mathbf{z}_i^{(t+1)} = h(\mathbf{x}^{(t+1)}, \mathbf{w}_i^{(t)})$, where $\mathbf{w}_i^{(t)}$ is the sensor noise, we can estimate the measurement

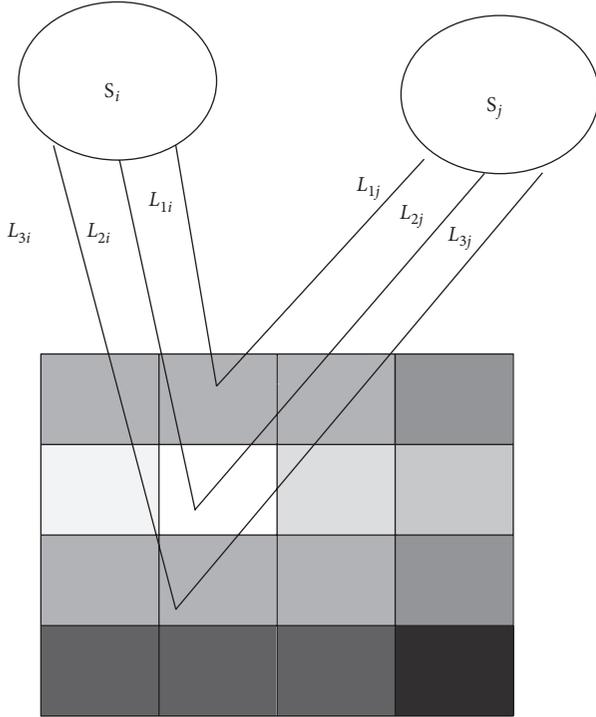


FIGURE 3: The expected likelihood function for each sensor (i or j) is a weighted sum of the marginal likelihood function conditioned at each grid in the predicted belief distribution.

$\mathbf{z}_j^{(t+1)}$ from the predicted belief and compute the expected likelihood function:

$$\begin{aligned} \hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \\ = \sum_{v_k \in S(\mathbf{x}^{(t+1)})} L_{ki}(\mathbf{x}^{(t+1)}, v_k) \times \left[p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)}) \Big|_{\mathbf{x}^{(t+1)}=v_k} \right], \end{aligned} \quad (7)$$

where the marginal likelihood is defined as $L_{ki}(\mathbf{x}^{(t+1)}, v_k) \triangleq \hat{p}(\mathbf{z}_j^{(t+1)}(\mathbf{x}^{(t+1)} = v_k) | \mathbf{x}^{(t+1)})$ and the prediction as

$$\begin{aligned} p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)}) \\ \triangleq \sum_{u_k \in S(\mathbf{x}^{(t)})} \left[p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}) \Big|_{\mathbf{x}^{(t)}=u_k} \right] \\ \times \left[p(\mathbf{x}^{(t)} | \mathbf{z}^{(t)}) \Big|_{\mathbf{x}^{(t)}=u_k} \right]. \end{aligned} \quad (8)$$

Using the estimated likelihood function $\hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)})$ from sensor i , the expected posterior belief can be obtained as follows: $\hat{p}(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)}) = C \cdot \hat{p}(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \cdot p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t)})$. We can then apply measures such as the entropy to the expected belief $\hat{p}(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)})$, as an approximation to the true belief $p(\mathbf{x}^{(t+1)} | \mathbf{z}^{(t+1)})$. This approach can apply to non-Gaussian belief since the discrete approximation of the

belief state assumes a general form. To compute the expected belief, however, we have conditioned the expected likelihood function on the predicted belief state.

5. Object Segmentation Approach

In order to investigate the hypothesis, a method needed to be developed that incorporated the information of the dynamic interaction between the desired object of interest and the background into a state-of-the-art segmentation technique to resolve whether that information could improve the performance results of the state-of-the-art segmentation technique. However, to incorporate the dynamic interaction information for object detection and tracking, the dynamic interaction needed to be quantified in a manner that allowed it to be integrated into a state-of-the-art technique. Since the fuzzy rule-based method was able to provide a number of distinct components including the desired object of interest, in order to improve upon the performance, we sought to detect the component that corresponded to the desired object of interest from the results of the fuzzy rule-based method and remove all other components. Due to the fact that only the desired object of interest will move in a consistent, nonaccidental manner in accordance with the dynamic interaction information extracted, incorporating the dynamic interaction information will allow for detecting and tracking the desired object of interest.

In order to accomplish the detection and tracking of the desired object of interest, we developed an algorithm using dynamic interaction information that was implemented after the fuzzy rule-based method completed segmenting each frame. The dynamic information that we sought to utilize in our algorithm was the expected movement of the object of interest. Drawing inspiration from the human visual system once again, we exploited the research completed by Palmer et al. [30] that discussed spatiotemporal relatability, specifically the hypotheses of persistence and position updating. The hypothesis of persistence described by Palmer et al. [30] states that object fragments remain perceptually available for a short time in the human visual system after occlusion so that they can be integrated into later-appearing fragments. The proposed algorithm uses this insight in the way that the algorithm is implemented. Between consecutive frames in the image sequence, any detected components are maintained by the proposed algorithm. The components maintained are available for the next frame only, comparably to the way that Palmer et al. [30] describe perceptually available objects for a short period. Additionally, Palmer et al. [30] hypothesize that the human visual system maintains a representation of the velocity of an object as it moves behind occluding surfaces. Using this information, Palmer et al. [30] describe how based on the velocity of the object, even though an object may not be visible, the human visual system updates the position so that it if the object parts become visible again, it can be integrated into other visible object parts. Likewise, the proposed algorithm uses the computed dynamic movement of the desired object of interest to update the expected position of the components detected. Based upon whether

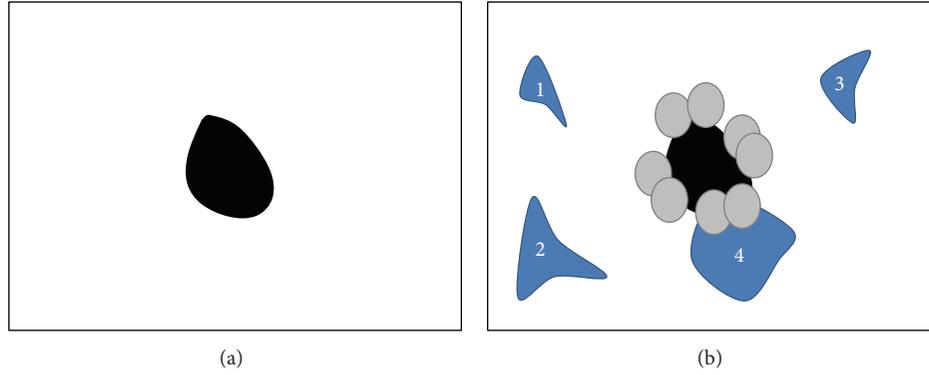


FIGURE 4: Visual representation of the algorithm process. (a) Representation of a single stored component. (b) Representation of a subsequent frame.

a detected component fulfills the updated expected position, object detection and tracking are performed.

For the first image in the sequence when the desired object of interest is detected, all of the distinct components detected by the fuzzy rule-based segmentation method are stored obeying the idea of persistence described by Palmer et al. [30]. For each component detected, every pixel included in the component, the centroid, and eight points on the extreme edge of the component were recorded. After the fuzzy rule-based method completes the segmentation on subsequent frames and detects distinct components, each of the components detected are compared against the expected motion of all of the stored components assuming that the stored components follow the motion of the desired object of interest following the concept of position updating from Palmer et al. [30]. The expected motion of the stored components is computed using the dynamic interaction information determined previously. The eight points along the edge of each of the stored components recorded earlier are at the following extreme locations: top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom, and left-top. At each of the eight locations around the extreme edges of each component, a circle with a radius corresponding to the maximum movement expected based on the dynamic interaction information calculation is positioned. Each of the positioned circles represents the possible movement of the component in each direction if the object was following the expected dynamics. In the subsequent frame, any of the components detected by the fuzzy rule-based method that are not located within the expected movement of a stored component are discarded because they do not follow the dynamic motion of the desired object of interest. In order to determine if any of the newly detected components are within the expected motion of a stored component, each of the extrema points is compared against each pixel of each newly detected component. For a single extrema point of a single stored component, the distance to every pixel in each of the newly detected components is calculated in accordance with the equation below:

$$\text{Measured Distance} = \sqrt{(x_n - x_e)^2 + (y_n - y_e)^2}, \quad (9)$$

where (x_n, y_n) are the coordinates of a pixel in one of the newly detected components and (x_e, y_e) are the coordinates of the extrema point. The measured distance is compared against the expected movement per image calculated previously. If the measured distance is less than the expected movement per image, then the newly detected component to which the pixel belongs is considered to have a presence within the expected movement of the stored component. Any detected components that have a presence within the expected movement of a stored component are accumulated for the next frame since it could possibly represent the desired object of interest. Pseudocode depicting the process for determining if a new detected component has a presence within the expected movement of a stored component is shown in Algorithm 1.

A visual representation of this process is displayed in Figure 4. The image on the left of Figure 4 shows a representation of a stored component. The image on the right of Algorithm 1 shows a visual representation of the comparison process of the algorithm. The blue components represent newly detected components in the current frame whereas the black component represents the stored component from the previous frame. Since only new detected component number four exists inside the expected movement of the object of interest (represented by the gray, transparent circles), it is the only newly detected component that is stored for the next frame. The rest of the newly detected components (1–3) would be discarded. After a series of subsequent frames, using the same process, ultimately only the desired object of interest will remain. However, in rare occurrences, the fuzzy rule-based method will mistakenly detect a component that has no directional movement but appears continuously in a majority of frames and would be retained throughout the sequence of images using the method described.

Therefore, in addition to the method described, another evaluation is performed that will establish whether or not the detected components that are retained for multiple frames maintain an overall directional movement. In addition to calculating the extreme points on the edge of stored components, the centroid of the component is also calculated and maintained. After each image where a component is maintained, the previous centroids that correspond to that

```

Determine the stored components of the previous frame
Determine the detected components of the current frames
FOR (each extreme point on the stored component)
  FOR (each pixel included in the new component detected)
    Calculate the distance ( $M_D$ ) between the current extreme pixel and the
    current pixel from the new detected component
  IF ( $M_D$  is less than the expected movement)
    (i) New detected component has presence in the expected
    movement, store the new detected component for next frame
    (ii) Break to next detected component
  END-IF
End-FOR
END-FOR

```

ALGORITHM 1: Pseudocode for determining if a new detected component falls within the expected movement of a stored component.

component are maintained, creating a register of the history of the component's centroid. Using this register of centroid data, every twenty frames, the distance the component has traveled is calculated and compared to a user-defined threshold. If the distance that the component has traveled is less than the user-defined threshold, the component is discarded in subsequent frames removing the mistakenly detected components from the fuzzy rule-based technique that have no directional movement. Figure 5 shows a visual representation of this process. For clarity, only five centroids are represented for two objects, whereas in the proposed algorithm the process is performed after twenty frames. In Figure 6, the distance traveled by the object represented by the black centroids is obviously greater than the distance traveled by the object represented by the blue centroids. Therefore, depending on the user-defined threshold, the object represented by the black centroids would be retained and stored, while the object represented by the blue centroids would be discarded.

Additionally, in order to facilitate understanding of the proposed algorithm, pseudocode of the proposed algorithm is provided in Algorithm 2. The pseudocode in Figure 5 describes the proposed algorithm from the completion of the fuzzy rule-based method proposed by Herrero and Bescos [5] until the completion of the proposed algorithm. In Algorithm 2, user-defined threshold is based on the maximum speed of the desired target and the number, 20, is based on the processed frames per second in our case.

Recall that the fuzzy rule-based method utilized two features for segmentation, the LBP (texture) and the LCP (color). Starting with the LBP, there are a number of parameters that will have an impact on the number of iterations of the algorithm. Let F represent the number of frames where each frame size is M rows by N columns. For the purpose of extracting the LBP, an R by R structure element is used with each LBP referencing P neighboring points around the center position of the structure element. In addition to finding the LBP for the R by R structure element, the algorithm also calculates the histogram statistic of three different color spaces for each pixel in the structure element as well. The fuzzy rule-based technique will take $F * M * N * (R * R * P + R * R * 3 + K)$ iterations to complete. If we regard the parameters

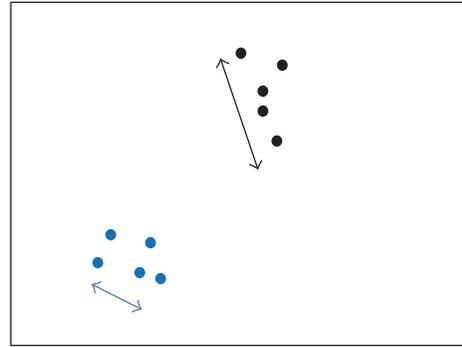


FIGURE 5: Visual representation of the algorithm process for removing objects mistakenly detected in a majority of frames with minimal directional movement.

P and K as the fixed constants, using the big O asymptotic notation, the complexity is in the $O(F * M * N * R^2)$ level.

However, in addition to the fuzzy rule-based, the proposed algorithm is performed after the completion of the fuzzy rule-based segmentation for every frame. The proposed algorithm compares the connections of the segmented components between two successive frames. Each pixel of each component in the new frame is compared against the eight extrema points from the stored components. Once we locate any connection between the current component with some previous component, we do not have to compare the current component against the rest of the stored components. Consequently, if there were C components in the previous frame, each pixel in the current component will check C times at most, and the total number of pixels from the components in the current frame that will have to be checked will be at most the entire image or M by N . Therefore, the iteration for checking the connection will take at most $M * N * C * 8$. Due to the fact that the number of components detected for each frame (C) is very limited, we can regard the number of components detected for each frame as a constant. Using these constraints, the complexity of the proposed algorithm is $O(F * M * N)$ for processing F frames. Therefore, the total complexity for the fuzzy rule-based method and the proposed

```

Determine the maximum motion dynamic expected per image in the form of pixel radius
FOR (each frame in the image sequence)
  (i) Perform fuzzy rule-based technique
  (ii) Perform connected component analysis on result of fuzzy rule-based technique
IF (current frame is the first frame in sequence to contain object of interest)
  Store all components detected by connected component analysis
  Each component stored has the following auxiliary information:
  (i) All pixels included in the component
  (ii) Centroid of the component
  (iii) Eight extrema points on the edge of the component
Else
  FOR (each new component detected)
    FOR (each of the sorted components from previous frame)
      FOR (each of the extrema points of stored components)
        FOR (each pixel included in the current new component)
          Calculate the distance from current extrema point to current
          pixel in current new component
          IF (distance < max motion dynamic expected)
            (i) Store current new component and auxiliary
            information for subsequent frames
            (ii) Break to next new component
          END-IF
        END-FOR
      END-FOR
    END-FOR
  IF (frame processed in a multiple of 20)
    Calculate the total distance travelled by the current new component
    IF (distance travelled is greater than user-defined threshold)
      Store current new component and auxiliary information for
      subsequent frames
    ELSE
      Discard the current new component
    END-IF
  END-IF
END-FOR
END-IF
END-FOR

```

ALGORITHM 2: Pseudocode of the proposed algorithm.

algorithm will be $O(F * M * N * R^2) + O(F * M * N)$, which simplifies into the $O(F * M * N * R^2)$ level.

6. Experimental Results

After the development of the improvement to the fuzzy rule-based method proposed by Chua et al. [27], the improved method was tested on three sequences of images from our data set. In order to numerically compare our technique with the results of the fuzzy rule-based technique, we utilized the same method that was used by Chua et al. [27] when evaluating their fuzzy rule-based technique in the original paper called F -measure [27]. The F -measure method used by Chua et al. [27] is defined as follows:

$$F = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}, \quad (10)$$

where

Recall

$$= \frac{(\# \text{ of pixels correctly classified as foreground})}{(\# \text{ of pixels classified as foreground in the ground truth})^{-1}}, \quad (11)$$

Precision

$$= \frac{\# \text{ of pixels correctly classified as foreground}}{\# \text{ of pixels classified as foreground}}.$$

For the first sequence of images, trial 1, the wave tank was set up to produce waves with a height of 0.05 meters and a period of 1.5 seconds. Based upon the manual segmentation of the desired object of interest from the sequence of images, it was calculated that the maximum expected movement of the desired object of interest per image in the sequence was

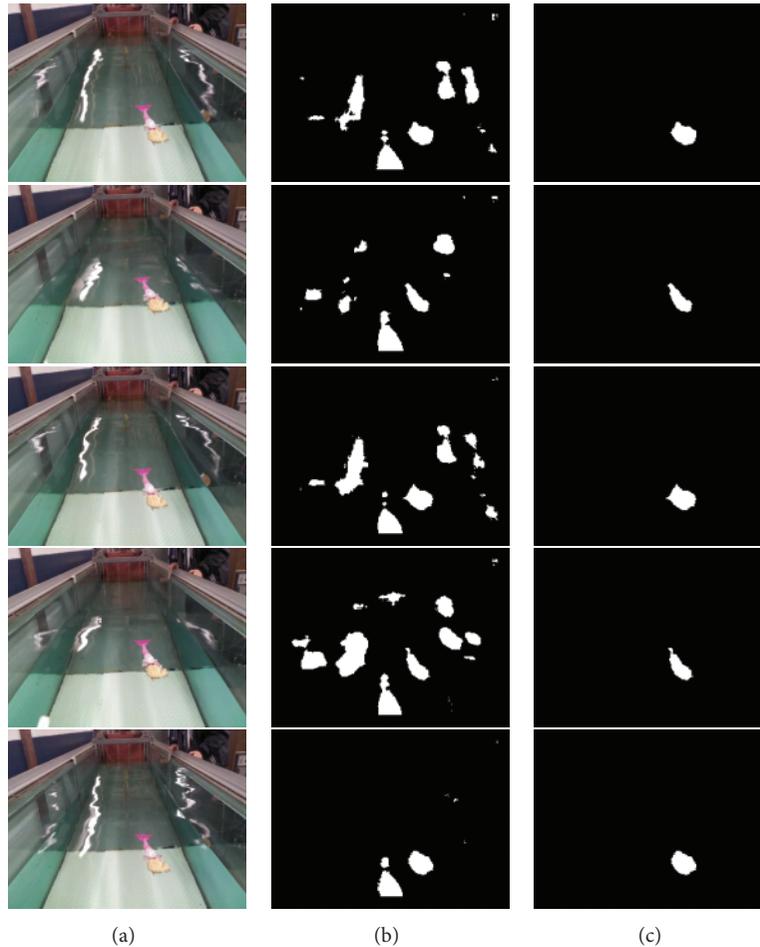


FIGURE 6: Example of five images from trial 2. (a) Original image. (b) Resulting images from fuzzy rule-based technique. (c) Resulting images from our improvement technique.

a pixel radius of 5.1. However, when the fuzzy rule-based method was performed on the sequence of images of trial 1, the method was unable to segment the desired object of interest. Based upon the fact that, of the three trials that were performed, trial 1 had the smallest movement per image, we believe that the fuzzy rule-based method failed on this image sequence due to the limited movement of the desired object of interest that caused the desired object of interest to be incorporated into the background of the image. Since the fuzzy rule-based method failed to detect the desired object of interest among the detected components, we were unable to test our improvement to the method.

For trial 2, the wave tank was set to produce waves with a height of 0.08 meters and a period of 1.5 seconds. Based upon the manual segmentation of the desired object of interest from the sequence of images from trial 2, it was calculated that the maximum expected movement of the desired object of interest per image in the sequence was a pixel radius of 6.5. Using the calculated dynamic interaction information and the results from the fuzzy rule-based method, we applied our improvement to detect and track the desired object of interest from the sequence of images. As shown in Table 1, our improvement to the fuzzy rule-based technique yielded

significantly improved results from the original fuzzy rule-based technique. Table 1 shows the recall, precision, and F -measure values for five example images from each of the two image sequences where our improved method was performed and the corresponding fuzzy rule-based technique results. Five consecutive example images and corresponding fuzzy rule-based technique images are shown in Figure 6 from the image sequence of trial 2. For trial 3, the wave tank was set to produce waves with a height of 0.1 meters and a period of 1.5 seconds. The calculated maximum expected movement per image based on the manual segmentation of the desired object of interest was a pixel radius of 6.4. Again, we applied our improvement to the fuzzy rule-based technique for the trial 3 sequence of images. Five consecutive example images and corresponding fuzzy rule-based technique images are shown in Figure 7 from the image sequence of trial 3.

7. Conclusion

In this paper, we hypothesized that the dynamic interaction between an object of interest and the background can provide

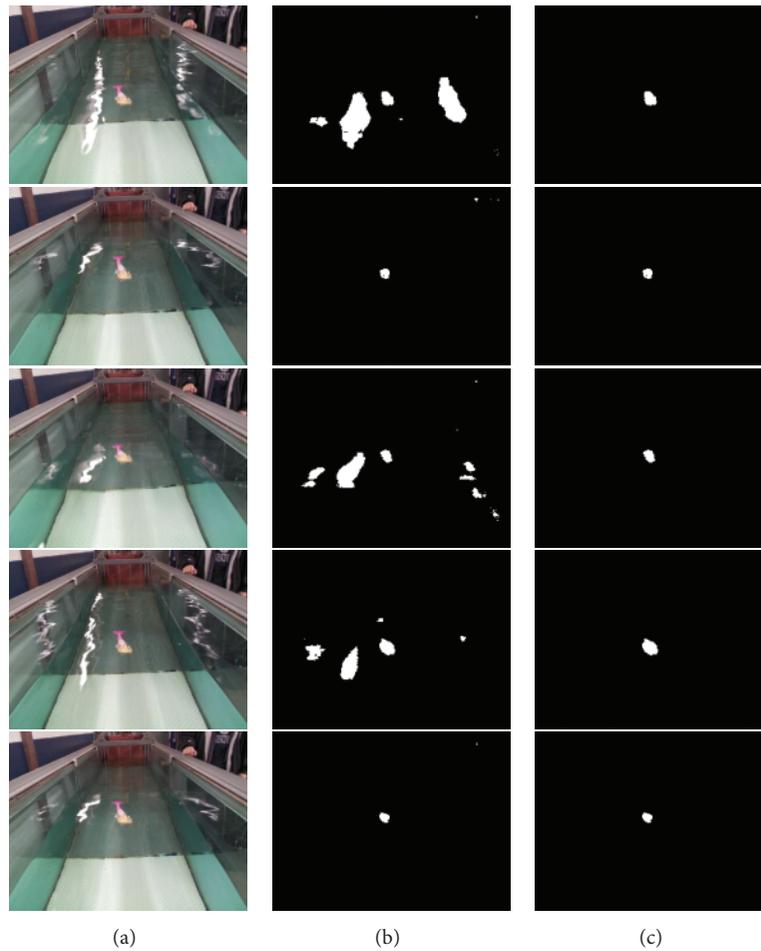


FIGURE 7: Example of five images from trial 3. (a) Original image. (b) Resulting images from fuzzy rule-based technique. (c) Resulting images from our improvement technique.

useful information and by understanding and modeling the dynamic interaction between an object of interest and background, we could improve the performance of state-of-the-art object detection techniques. After implementing two current state-of-the-art techniques and evaluating the performance when conducted on our dynamic water tank environment, we observed that the fuzzy rule-based technique proposed by Chua et al. was able to segment the frames of image sequences into a number of distinct components, one of which was the desired object of interest. Understanding and modeling the dynamic interaction between the object of interest and the background by manually segmenting and recording the movement information for a specific dynamic environment (i.e., wave height and period) allowed us to develop an algorithm to incorporate the dynamic interaction information into the segmentation process. Using our algorithm and the fuzzy rule-based technique, we performed segmentation on three image sequences from our wave tank data set. During the first trial, the fuzzy rule-based technique was unable to detect the object of interest among the detected components. Since the fuzzy

rule-based technique failed, we were unable to implement our algorithm for the first trial. However, on the second and third trial, the fuzzy rule-based technique performed adequately enough for us to implement our algorithm and compare the results of our algorithm against the results of the fuzzy rule-based technique. Based on the recall, precision, and F -measure data calculated for the trials, our proposed algorithm significantly improves upon the results observed from the fuzzy rule-based algorithm proposed by Chua et al. The experimental results achieved in this paper demonstrate convincing evidence that the dynamic interaction between an object of interest and background provides valuable information that can be utilized to improve segmentation results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

TABLE 1: Recall, precision, and F -measure data.

Image sequence	Metric	Fuzzy rule-based method	Our method
Trial 2			
Image 1	Recall	0.28261	0.56522
	Precision	0.023851	0.68421
	F -Measure	0.043989	0.61905
Image 2	Recall	0.31382	0.62763
	Precision	0.026282	0.91781
	F -Measure	0.048502	0.74548
Image 3	Recall	0.29924	0.59848
	Precision	0.022281	0.60305
	F -Measure	0.041473	0.60076
Image 4	Recall	0.31504	0.62053
	Precision	0.022303	0.93525
	F -Measure	0.041657	0.74605
Image 5	Recall	0.3093	0.6186
	Precision	0.025775	0.6838
	F -Measure	0.047585	0.64957
Trial 3			
Image 1	Recall	0.76389	0.76389
	Precision	0.017909	0.67073
	F -Measure	0.034998	0.71429
Image 2	Recall	0.54	0.54
	Precision	0.016232	0.8617
	F -Measure	0.031518	0.66393
Image 3	Recall	0.62581	0.62581
	Precision	0.017382	0.776
	F -Measure	0.033824	0.69286
Image 4	Recall	0.81006	0.81006
	Precision	0.026571	0.68075
	F -Measure	0.051455	0.7398
Image 5	Recall	0.50562	0.50562
	Precision	0.018072	0.98901
	F -Measure	0.034897	0.66914

Acknowledgment

The project was in part supported by National Science Council (NSC) Programs NSC101-2221-E-008-039-MY3 and NSC-102-3113-P-007-014.

References

- [1] G. Xue, J. Sun, and L. Song, "Dynamic background subtraction based on spatial extended center-symmetric local binary pattern," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '10)*, pp. 1050–1054, July 2010.
- [2] Y. Benezeth, P.-M. Jodoin, B. Emile, and H. Laurent, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, Article ID 033003, 2010.
- [3] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, December 2008.
- [4] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1937–1944, June 2011.
- [5] S. Herrero and J. Bescos, "Background subtraction techniques: systematic evaluation and comparative analysis," in *Proceedings of the 11th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS '09)*, pp. 33–42, Berlin, Germany, 2009.
- [6] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari, "Evaluation of background subtraction methods," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA '08)*, pp. 357–364, December 2008.
- [7] D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proceedings of the IEEE 5th International Conference on Advanced Video and Signal Based Surveillance (AVSS '08)*, pp. 192–199, September 2008.
- [8] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "P finder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.
- [9] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, pp. 246–252, June 1999.
- [10] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 28–31, August 2004.
- [11] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.
- [12] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 827–832, 2005.
- [13] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1162, 2002.
- [14] A. M. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the 6th European Conference on Computer Vision—Part II (ECCV '00)*, pp. 751–767, London, UK, 2000.
- [15] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 1305–1312, October 2003.
- [16] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778–1792, 2005.
- [17] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 171–177, 2010.

- [18] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, vol. 5, pp. 3061–3064, October 2004.
- [19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [20] O. Barnich and M. van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.
- [21] O. Barnich and M. van Droogenbroeck, "ViBE: a powerful random technique to estimate the background in video sequences," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 945–948, April 2009.
- [22] Y. Li, W. Chen, and R. Jiang, "The integration adjacent frame difference of improved ViBe for foreground object detection," in *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '11)*, pp. 1–4, September 2011.
- [23] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.
- [24] M. van Droogenbroeck and O. Paquot, "Background subtraction: experiments and improvements for ViBe," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '12)*, pp. 32–37, 2012.
- [25] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.
- [26] S. Zhang, H. Yao, and S. Liu, "Dynamic background modeling and subtraction using spatio-temporal local binary patterns," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 1556–1559, October 2008.
- [27] T. W. Chua, K. Leman, and Y. Wang, "Fuzzy rule-based system for dynamic texture and color based background subtraction," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '12)*, pp. 1–7, 2012.
- [28] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognition*, vol. 42, no. 3, pp. 425–436, 2009.
- [29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [30] E. M. Palmer, P. J. Kellman, and T. F. Shipley, "A theory of dynamic occluded and illusory object perception," *Journal of Experimental Psychology*, vol. 135, no. 4, pp. 513–541, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

