Background Subtraction in Varying Illuminations Using an Ensemble Based on an Enlarged Feature Set

Brendan Klare
Dept of Computer Science and Engineering
Michigan State University
klarebre@msu.edu

Sudeep Sarkar
Dept of Computer Science and Engineering
University of South Florida
sarkar@cse.usf.edu

Abstract

Image sequences with dynamic backgrounds often cause false classification of pixels. In particular, varying illuminations cause significant changes in the representation of a scene in different color spaces, which in turn results in the high levels of failure in such conditions. Because mapping to alternate color spaces has largely failed to solve this problem, a solution of using alternate image features is proposed in this paper. In particular, the use of gradient and texture features along with the original color intensities are used in an ensemble of Mixture of Gaussians background classifiers. A clear improvement is shown when using this method compared to the Mixture of Gaussians algorithm using only color intensities. In addition, this work shows that performing background subtraction using only gradient magnitude as an image feature performs at a much higher rate in varying illuminations then using color intensities. Results are generated on three separate datasets, each with unique, dynamic, illumination conditions.

1. Introduction

Automated tracking systems have progressed significantly over the past decade, particularly when used in indoor and controlled environments. Unfortunately autonomous tracking in outdoor environments is still crippled by many of the complex characteristics of outdoor environments. Dynamic backgrounds (such as moving trees, rippling water, etc.) and illumination changes are some of the most difficult types of background events to correct classify.

In background classification each frame in an image sequence is initially segmented into foreground and background regions, where the foreground regions represent moving objects that are of interest to a high level tracking system. Generally, these objects are limited to animals and moving vehicles. Background regions are the complement of the foreground regions. The process of pixelwise segmentation of an image into these two classes is commonly referred to as background subtraction, foreground/background segmentation, and background classification. It is common to build a statistical model that can be updated with an iterative algorithm to characterize the background. Outliers from the model are cast as foreground.

Accurate background subtraction is vital to tracking systems. Tracking systems seek to infer high level semantics from an observed scene, such as traffic monitoring, target acquisition, and biometric identity. The more false hypotheses the high level algorithms receive from a segmentation algorithm the worse the expected overall performance of the system. For this reason there has been a significant amount of research for improving foreground/background segmentation.

1.1. Related Works

In [24], Stauffer and Grimson first proposed using the Mixture of Gaussians algorithm for background subtraction, which has since become the most common approach to background subtraction. This is a progression from modeling each pixel as a single Gaussian distribution [27]. In the Mixture of Gaussians algorithm, each pixel is characterized by multiple, weighted, Gaussian distributions. Each distribution corresponds to an observed mode, and its weight represents each modes frequency of occurrence. As a new pixel $p_{i,j}$ is processed, each Gaussian distribution $D_{i,j}(l) \sim N(\mu, \sigma)$ is checked to see if it matches $p_{i,j}$, where ($1 \leq l \leq K$) and K is the number of distributions. A match is when a pixel is within 2.5 deviations from the distribution's mean. $M_{i,j}(D_{i,j}(l), p_{i,j})$ is 1 if distribution $D_{i,j}(l)$ matches pixel $p_{i,j}$, and 0 otherwise. The weights of the distributions are updated using Equation 1, where $0 < \alpha < 1$ is the learning rate, and $w_{i,j}(l)$ is the weight of the lth distribution at location (i, j). The pixel $p_{i,j}$ is considered a member of the background if $\beta_{i,j}$ from Equation 2 is greater than the predefined threshold τ, where typically $\tau \approx .2$. So the more times a distribution is matched,

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

the heavier its weight becomes. The heavier a distribution's weight, the more likely it represents the background.

$$w_{i,j}(l) = M_{i,j}(D_{i,j}(l), p_{i,j}) \cdot \alpha + w_{i,j}(l) \cdot (1 - \alpha) \quad (1)$$

$$\beta_{i,j} = \sum_{l=1}^{K} w_{i,j}(l) \cdot M_{i,j}(D_{i,j}(l), p_{i,j}) \quad (2)$$

Many variations of the Mixture of Gaussians algorithm have been proposed. In [12], KaewTraKulPong and Bowden use Mixture of Gaussians with varying learning rates that allows the algorithm to adapt to change faster. Shadow detection is explicitly addressed by working in a chromatic color space, similar to the one used by Horprasert *et al.* in [9]. In [7], Gordon *et al.* incorporate depth information into the Mixture of Gaussians algorithm. While additional sensors are required, combining depth information creates an algorithm more robust to illumination changes. In [10], Jabri *et al.* were one of the first to use image gradient information as a feature for tracking. Javed *et al.* used gradient magnitude and orientation, as well as the RGB intensities, to create a five dimensional Mixture of Gaussian algorithm in [11]. While computing the edge features incurs additional computation, the use of these features results in a more illumination invariant feature space.

In [8], a non-parametric background subtraction algorithm was presented using texture features in the form of a local binary pattern. In [28], Zhong and Sclaroff used texture information in conjunction with a Kalman filter background subtraction algorithm. Other Kalman filter based approaches can be found in [16, 22].

Background classification using motion information has also been used for background classification and tracking. Examples of these works can be found in [26, 17].

Another family of background subtraction algorithms uses global image information in order to determine which pixels belong to the background and foreground processes. Eigenbackgrounds [18] and pixel layering [6, 21] are some examples of these methods.

The use of ensembles in background subtraction has not been heavily investigated. Avidan uses an ensemble of classifiers via the Ada Boost algorithm in [4]. This approach requires labeled training data that contains the locations of a specific object to be tracked. While this algorithm is highly effective, it is only capable of tracking one specific object, and requires labeling data offline.

1.2. Proposed Method

In this paper, a new algorithm for background subtraction is proposed. This algorithm is a meta-learning algorithm that incorporates multiple instantiations of the Mixture of Gaussian algorithm. By operating on 13 different image features, this algorithm demonstrates a significantly heightened performance on image sequences with varying illuminations. A surprising consequence of this research was the exceptional performance of using only edge features for background subtraction.

The motivation for this work arose from the failures of mapping RGB color spaces to alternate color spaces that are ideally illumination invariant. Figure 1 shows two frames taken from the OTCBVS dataset [1]. The images on the left are the scene under sunny condition, and the images on the right are the same scene after a cloud covers the sun. The RGB, hue, saturation, and gradient magnitude values for both frames are shown. Of particular note is the fact that the hue and saturation are both highly variant to this change, while the gradient magnitude is to a much lower extent. Similar robustness was observed in the texture features used in this work. Continued failures using alternate color spaces in illumination varying datasets inspired the approach presented in this paper.

This paper represents the first known work to fuse multiple unsupervised background classifiers. Additionally, it is the first known work to use texture information derived from Haar features in background subtraction. The remainder of the paper is outlined as follows: Section 2 provides a description of the feature set used for classification. Section 3 describes the ensemble algorithm used for background classification. Results of our algorithm compared against an open source implementation of the Mixture of Gaussians classifier on three publicly available datasets are shown in Section 4. Concluding remarks are presented in Section 5.

2. Feature Space

The first step in the proposed background classification algorithm is to separate an image into distinct features. Thirteen different features are used. The first three features are simply the intensities values for the red, green, and blue image channels. These features may take a range of 0 to 255.

The next two features are the image gradient and magnitude. A simple Canny edge detector [5] is used, with $\sigma = 2$. If the gradient in the x and y directions are

$$G_x(i,j) = \tilde{p}_{i,j} - \tilde{p}_{i-\sigma,j}$$

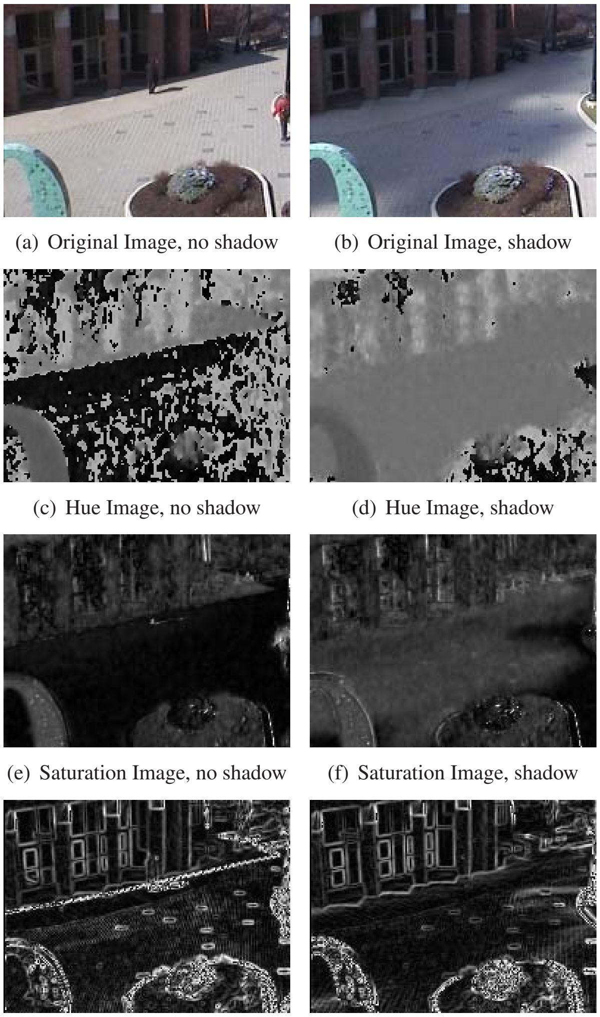
and

$$G_y(i,j) = \tilde{p}_{i,j} - \tilde{p}_{i,j-\sigma}$$

where $\tilde{p}$ is a Gaussian smoothed pixel, then the magnitude and orientation feature values for pixel $p_{i,j}$ can be calculated using Equations 3 and 4, respectively.

$$G_M(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)} \quad (3)$$

$$G_\theta(i,j) = \arctan\left(G_y(i,j)/G_x(i,j)\right) \quad (4)$$



(a) Original Image, no shadow (b) Original Image, shadow

(c) Hue Image, no shadow (d) Hue Image, shadow

(e) Saturation Image, no shadow (f) Saturation Image, shadow

(g) Gradient Magnitude, no shadow (h) Gradient Magnitude, shadow

Figure 1. Effects of varying illumination on different features

The final eight features used are Haar features, which act as low cost texture estimates over a region of pixels. The emergence of Haar features in pattern analysis and computer vision began with Papageorgiou *et al*. in [19, 20], and have received particular attention after their use in object detection by Viola and Jones in [25].

Haar features offer simple texture estimations over an image subregion at a low computational cost. The eight Haar features being used in this paper can be viewed in Figure 2. Haar features are particularly desirable because they can be quickly computed by using an integral image, which was shown by Lienhart and Maydt in [15] to be computable in one pass of the image.

By consequence of computing the texture in the 12x12 region surrounding each pixel, spatial information is incorporated into Haar feature. So while the algorithm being presented remains a local background subtraction method, the Haar features incorporate information regarding neighboring pixels.

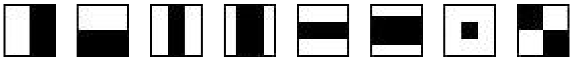


Figure 2. Set of Haar features being used

3. Algorithm

The algorithm used is an ensemble algorithm that consists of 13 Mixture of Gaussian classifiers C_k $(1 \leq k \leq 13)$. Each classifier operates exclusively on one of the 13 features from the feature set described in Section 2. After each classifier C_k has processed a frame, the hypothesis $H_{i,j}(k)$ of each classifier is then fused, resulting in a single hypothesis $H_{i,j}$ for each pixel $p_{i,j}$, where $H_{i,j} \in \{B, F\}$.

The fusion method used is the average rule, which has been shown to be the most effective method of fusing equally weighted hypotheses [13, 14]. Equations 5 and 6 show how the single, per-pixel hypothesis $H_{i,j}$ is generated using the average rule.

$$\hat{H}_{i,j} = \frac{1}{n} \sum_{k=1}^{13} H_{i,j}(k) \tag{5}$$

$$H_{i,j} = \begin{cases} B & \text{if } \hat{H}_{i,j} \geq \tau \\ F & \text{otherwise} \end{cases} \tag{6}$$

The ability to use average rule fusion is because the Mixture of Gaussians algorithm is able to generate a degree of membership, as opposed to a classifier which only generates a binary decision. While the final result of the Mixture of Gaussians classifier is a binary decision, this results after a membership degree β is thresholded. Therefor, $H_{i,j}(k) = \beta_{i,j}$. As was seen in Equation 2, the value $\beta_{i,j}$ is the sum of the weights of all distributions that match pixel $p_{i,j}$. Averaging each of the classifier's $\beta_{i,j}$ values and then thresholding based on τ allows for more information to be used in the fusion of the classifiers.

It is tempting to think that the ensemble algorithm presented can be reduced significantly in complexity by using a 13 dimensional Gaussian mixture model instead. While using a single classifier that models the data in 13 dimensions would drastically reduce the computational complexity of the algorithm, it would also result in a very different algorithm. To better understand this fact, consider Equation 1. A distribution's weight is only updated if it is matched. Yet, in order for a match to occur under a 13 dimensional model, the incoming feature vector must be within 2.5σ of the mean for each dimension. Therefor, if one of the 13 dimensions does not match then the entire feature vector is considered non-matching. Using an ensemble, each dimension has its

Table 1. Dataset details

Algorithm	GT Frames	FG Objects	Resolution
OTCBVS	16	75	320 x 240
PETS 2001	22	53	768 x 576
PETS 2006	20	66	720 x 576

own classifier with one dimensional distributions. A match is based only on that dimensions current value. This allows each classifier to update independently of the other feature values.

Because each classifier operates on a single feature and has a limited access to information, it makes each classifier more along the lines of a theoretically "weak" classifier. According to predominate ensemble classifier theory, presented originally by Schapire in [23], these weaker hypotheses can be fused into a single strong hypothesis.

4. Results

Three publicly available datasets were used to evaluate the performance of the ensemble algorithm. These sets are the PETS 2001 dataset [2], the PETS 2006 dataset [3], and the OTCBVS dataset [1]. The PETS 2001 and OTCBVS datasets are both outdoor datasets. The OTCBVS dataset is plagued with varying illumination caused from passing clouds, and is the most difficult dataset. The PETS 2001 dataset has a gradual illumination change at the end of the testing sequence. The PETS 2006 dataset is an indoor dataset, though the floor causes specular reflections of the foreground objects. The number of ground truth frames, total foreground objects, and image resolution for each dataset can be found in Table 1. The ground truth frames used were spaced throughout the entire dataset, i.e. they are not from consecutive frames and they span the entire set.

Three separate algorithms were evaluated. The first is the ensemble algorithm presented in this paper, which we will refer to as Ensemble MofG. The next is the Mixture of Gaussians algorithm implemented in Intel OpenCV, which is based on the work in [12]. This algorithm will be called OpenCV MofG, and serves as a baseline performance indicator. The final algorithm used is a single dimensional Mixture of Gaussians algorithm which operates only on the gradient magnitude feature. This algorithm will be referred to as Gradient MofG. Parameters for each algorithm were optimized by evaluating their performance on separate training sequences for each dataset.

ROC analysis was used to evaluate the performance of each algorithm. Bounding boxes of foreground objects were used as the ground truth. For each foreground object, a true positive classification was considered to be when at least 25% of the pixels within that objects bounding box were classified as foreground. False positives were measured as the percentage of pixels outside the bounding box that were incorrectly classified as foreground. In order to generate the various points on the ROC plots, the threshold τ from Equation 6 was varied. A lower τ results in less true positives and less false positives, while a higher τ results in more true positives and more false positives.

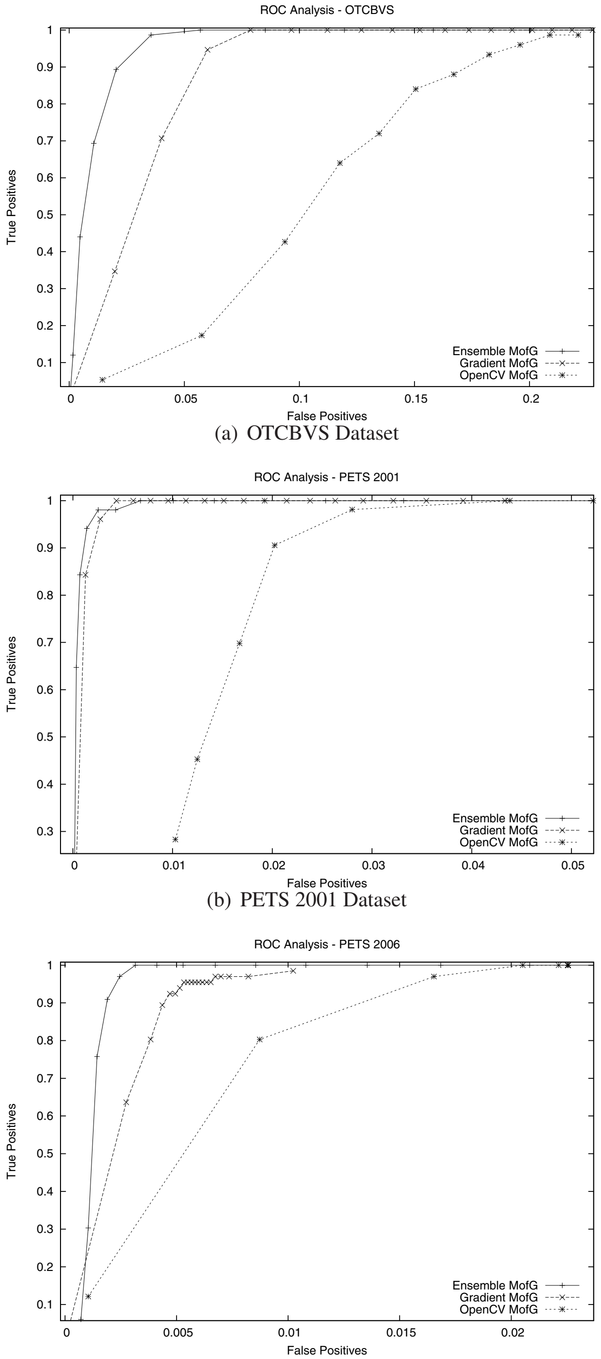
Table 2. Each algorithm's false positive rate at 90% true positive performance

Algorithm	OTCBVS	PETS 2001	PETS 2006
Ensemble	.0204	.0014	.0019
Gradient	.0601	.0028	.0044
OpenCV	.1825	.0202	.0165

The results over the entire datasets are seen in Figure 3. In all three datasets the ensemble algorithm offered the optimal performance. The Gradient MofG algorithm outperformed the OpenCV MofG, which was an initially unexpected result when conducting this research. Table 2 shows each of these algorithms false positive rates when operating at a 90% true positive rate over the entire ground truth. When compared to the proposed ensemble algorithm, the baseline OpenCV Mixture of Gaussians algorithm had 50 times as many false positive pixels in the OTCBVS dataset, 14 times as many in the PETS 2001 dataset, and 8 times as many in the PETS 2006 dataset.

Figure 4 shows each algorithm's performance at each ground truth frame. These results help substantiate the claim that the heightened performance when using Ensemble MofG and Gradient MofG over OpenCV MofG are due to greater illumination invariance. Consider the frame by frame results of the PETS 2001 dataset in Figure 4(b). The performance of the OpenCV MofG is equal to the two proposed algorithms until the final few frames of ground truth. These are frames from the end of the data set when a global illumination change occurs. Both the Ensemble MofG and Gradient MofG are largely invariant to this illumination change, while the OpenCV MofG has a significantly deteriorated performance. In the PETS 2006 dataset there is no illumination change. Only the reflection of the foreground objects on the floor causes false positive classification. This explains why the OpenCV MofG algorithm cleanly scales the Ensemble MofG and Gradient MofG. Finally, while the consistent passage of clouds in the OTCBVS dataset caused such a significant illumination change that the Ensemble MofG and Gradient MofG were not entirely invariant to them, their performance in these conditions was a drastic improvement over the OpenCV MofG algorithm.

It is important to note the improved performance using Gradient MofG. While the Ensemble MofG performed better on average, the computational demands are greatly reduced in the Gradient MofG algorithm. This fact suggests the use of the Gradient MofG classifier for systems with limited resources. The performance of the other individual features, as well as subsets of the entire feature set, were

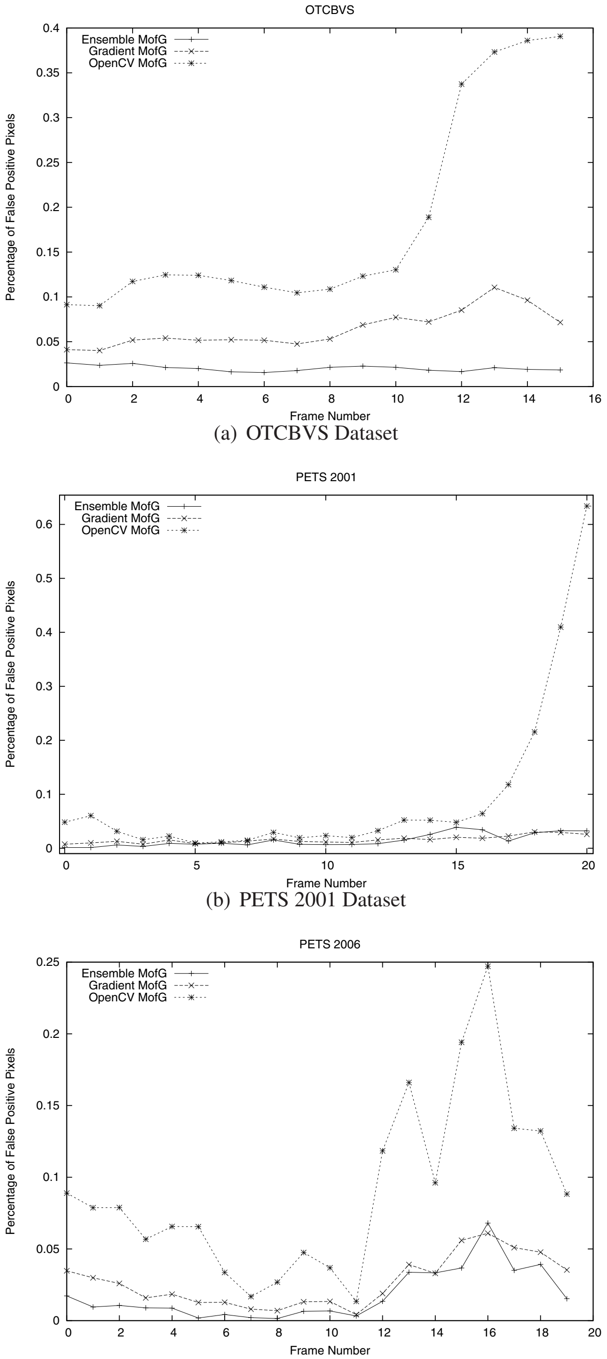


(a) OTCBVS Dataset

(b) PETS 2001 Dataset

(c) PETS 2006 Dataset

Figure 3. Global results for each background subtraction algorithm



(a) OTCBVS Dataset

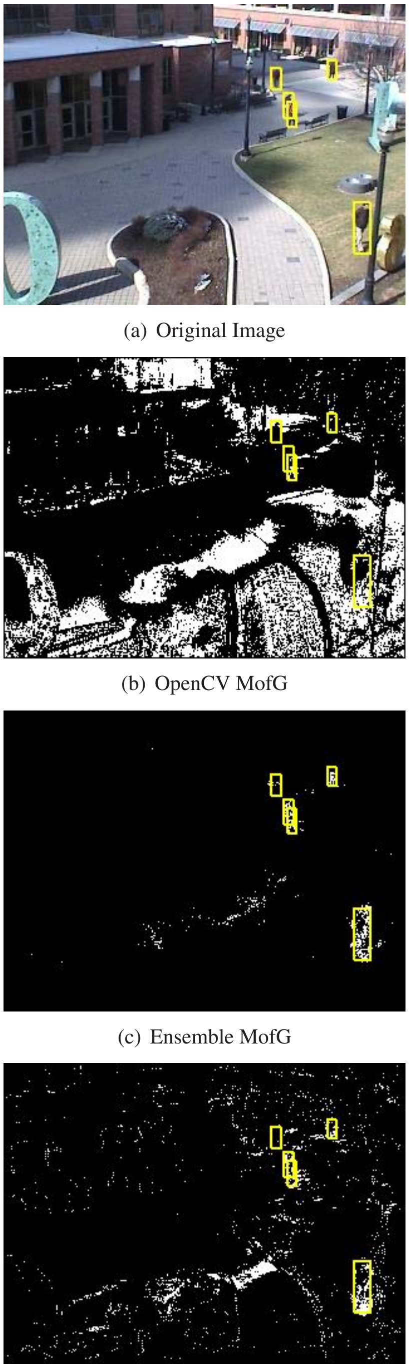
(b) PETS 2001 Dataset

(c) PETS 2006 Dataset

Figure 4. Percentage of false positive pixels in each frame when classifying at a 90% true positive rate

tested and no significant performances occurred.

Figures 5 and 6 show the classification results for each pixel of a frame undergoing an illumination change in the OTCBVS and PETS 2001 datasets. It is easy to see the clear failure of classification when only using the color features in these varying illuminations. The best classification in each case is when using the Enesemble MofG algorithm, though this algorithm still has some false classification, and in the OTCBVS frame there is one missed person.



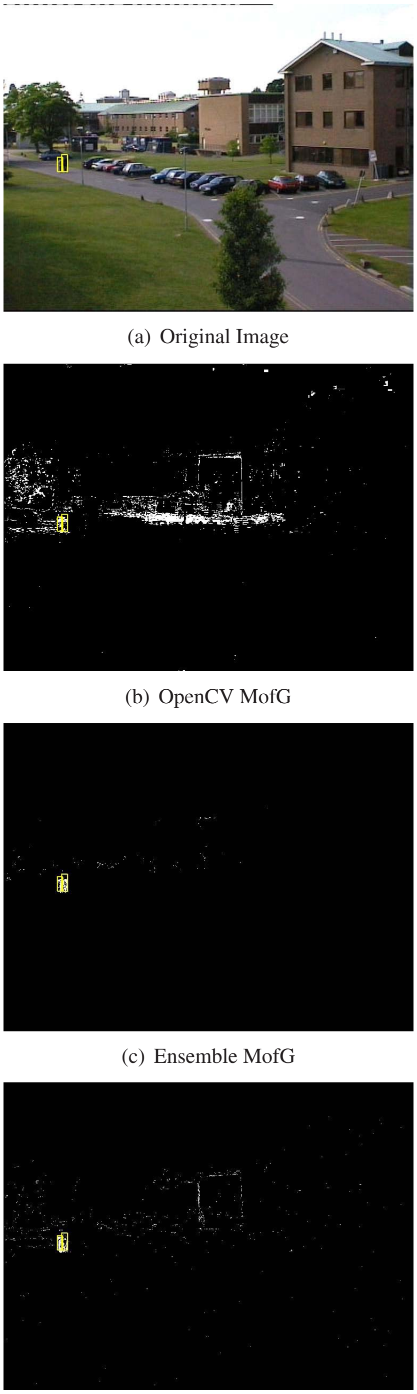
(a) Original Image

(b) OpenCV MofG

(c) Ensemble MofG

(d) Gradient MofG

Figure 5. Classification of a frame from the OCTCBVS dataset during an illumination change



(a) Original Image

(b) OpenCV MofG

(c) Ensemble MofG

(d) Gradient MofG

Figure 6. Classification of a frame from the PETS 2001 dataset during an illumination change

5. Conclusion and Analysis

Background subtraction in image sequences with dynamic illumination conditions was greatly improved on by both the Ensemble Mixture of Gaussians and the Gradient Magnitude Mixture of Gaussians algorithms presented. Results were compared against an open source implementation of the Mixture of Gaussians algorithm over three separate datasets. The best results observed were from the Ensemble

Mixture of Gaussians algorithm.

The ensemble algorithm was able to combine 13 separate (and generally "weak") hypotheses into a single strong hypothesis. Each classifier used a separate feature from a feature set that included the three RGB features, two gradient based features, and eight Haar features. The strong performance of the Mixture of Gaussians classifier that only used the gradient magnitude image feature was an unexpected result of the research conducted. No other single classifier had significant, individual performances.

The reason that classification was improved upon in varying illumination conditions is that spatial information was used while still maintaining a focus on a per pixel classification approach. When spatial information is not used a classifier must determine if a future pixel belongs to a foreground or background process based solely on some color intensity feature. The lack of a truly illumination invariant color space when being restricted to a discrete range of color intensities results in the change of color intensity during an illumination change. This change in intensity cannot easily be distinguished from an intensity change that results from a new foreground object. When using spatial information as a feature for a pixel (such as edge features or texture features), a pixel is defined based on its relationship to neighboring pixels. If each pixel in a neighborhood undergoes the same change, then the difference between neighboring pixels remains static in ideal circumstances. During an illumination change, neighboring pixels often experience the same change in illumination which causes a feature such as edge intensity or a Haar feature to remain at the same value. This illumination invariance is precisely what is desired for background subtraction in varying illuminations.

At the same time it is important to incorporate spatial information for background subtraction, it is also important to still use the original color intensities as well. We observed this empirically as our ensemble classifier performed much worse without the RGB features. This fact is also intuitive as there needs to be some preciseness to a pixels observations. That is, incorporating only spatial information loses focus on the pixel in question. This is believed to be a reason why using only the gradient magnitude performed well in a single feature classifier. When using a small standard deviation, the response of the edge detector keeps a tight neighborhood while still incorporating spatial information.

The ensemble algorithm implemented is not a real time classifier, yet. However, a high level of parallelism is inherent to the algorithm. The code paths for each classifier in the ensemble are disjoint, implying each could run on a separate processor or core. Also, shared memory accesses by each classifier are read-only for accessing the incoming frame. The only shared memory writes that are needed are to update the classifiers per pixel hypotheses. So while a real time implementation was not generated for this paper, it is believed that using a parallel system would result in real time performance.

References

[1] OTCBVS Benchmark Dataset Collection. http://newton.ex.ac.uk/tex/pack/bibtex/btxdoc/node6.html.

[2] PETS 2001 Dataset. ftp://ftp.pets.rdg.ac.uk/pub/PETS2001.

[3] PETS 2006 Dataset. http://www.cvg.rdg.ac.uk/PETS2006/data.html.

[4] S. Avidan. Ensemble tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):261–271, 2007.

[5] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8:679–714, 1986.

[6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. *CVPR*, 1:53–60, 2006.

[7] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 395–398, 2001.

[8] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):657, 2006.

[9] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *ICCV*, 1999.

[10] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *International Conference on Pattern Recognition*, volume 4, pages 627–630, 2000.

[11] O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. In *Motion and Video Computing, Proceedings of Workshop on*, pages 22–27, 2002.

[12] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems*, 2001.

[13] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–269, 1998.

[14] L. Kuncheva. A theoretical study on six classifier strategies. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(2):281–286, 2002.

[15] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing*, 2002.

[16] S. Messelodi, C. Modena, N. Segata, and M. Zanin. A kalman filter based background updating algorithm robust to sharp illumination changes. In *ICIAP 2005, 13th International Conference on Image Analysis and Processing*, pages 163–170, 2005.

[17] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *Computer Vision and Pattern Recognition*, 2:302–309, 2004.

[18] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:831–843, 2000.

[19] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. Computer Vision and Pattern Recognition*, pages 193–199, 1997.

[20] C. Papageorgiou, M. Oren, , and T. Poggio. A general framework for object detection. In *In International Conference on Computer Vision*, 1998.

[21] K. Patwardhan, G. Sapiro, and V. Morellas. Robust foreground segmentation using pixel layers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 4:746–751, 2008.

[22] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive bbackground estimation and foreground detection using kalman filtering. In *In Proc. ICAM*, pages 193–199, 1995.

[23] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[24] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 1999.

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[26] L. Wixson. Detecting salient motion by accumulating directionally-consistent flow. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):774–780, 2000.

[27] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[28] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. *Computer Vision, IEEE International Conference on*, 1:44, 2003.