Offline Generation of High Quality Background Subtraction Data

Etienne Grossmann, Amit Kale, Christopher Jaynes and Sen-ching Samson Cheung
UK Center for Visualization and Virtual Environments
1 Quality Street; Lexington KY 40507-1464, USA
{etienne,amit}@cs.uky.edu, jaynes@vis.uky.edu, cheung@engr.uky.edu

Abstract

Ground truth is important not only for performance evaluation but also for a principled development of computer vision algorithms. Unfortunately obtaining ground truth data is difficult and often very labor intensive. This is particularly true of video analysis due to the immense cost of producing pixel-wise ground truth in potentially thousands of frames. In this paper, we propose a method to produce foreground/background segmentation for video sequences captured by a stationary camera, that requires very little human labor as compared to complete manual segmentation, while still producing high quality results. Given a sequence, we use a few hand labeled images and Adaboost to train a classifier that segments the rest of the sequence. We demonstrate the effectiveness of our approach on two sequences and discuss the new horizons opened by these encouraging results.

1 Introduction

Ground truth is very important in computer vision because of the central role it can play in the empirical analysis and development of algorithms. Important uses of labeled image databases, for example, are the training of image classifiers and detectors [21] or of face recognition methods [14]. Ground truth also provides "perfect" data to test existing methods that were built on the premises that this data is available. For example, having exact silhouettes of walkers helps in the evaluation of gait recognition algorithms [16]. Perhaps the most common usage of ground truth is in quantitative performance evaluation, where it sets the gold standard, whether for ego-motion estimation [1], pattern recognition, background subtraction [6], or others tasks.

Unfortunately ground truth is usually hard to obtain. For instance, precise camera motion can be obtained from robot arm encoders [1] or precisely known calibration targets; however this can be cumbersome to obtain. Labeling image databases or videos requires a person to view and annotate hundreds or thousands of images [7]. Foreground/background ground truth, the object of this article, is obtained by hand-segmenting each image, each of which may require two to thirty minutes [6].

We propose an off-line method to generate near-perfect foreground segmentation of video sequences requiring significantly less manpower than extensive hand-labeling. The method is semi-automatic and only requires the user to manually segment foreground regions in a few (1-5) images. These few manually-segmented images are used to train an

automatic classifier that will then segment the whole sequence. The automatic classifier, based on Adaboost [20], combines the output of many "weak classifiers" to classify the training data with arbitrary precision. In our case, the weak classifiers are ordinary image and video filters. Our method requires a single parameter to be set viz. the desired classification error of Adaboost on the training dataset or the total number of training steps. In short, we use supervised learning and image processing tools to complete the labeling work started by the user.

Recently there has been an upsurge in the use of supervised learning methods in computer vision. The main reason is that supervised learning methods e.g. Adaboost, decision trees, and neural networks combine simple decision rules (weak classifiers, stumps, neurons) to obtain classifiers that outperform ad-hoc methods [23]. Moreover the latter require more field-specific knowledge from the designer. Examples of recent uses of supervised learning approaches in computer vision include novel view generation [9, 8], face or pedestrian detection [23, 24].

The reader should note that, while these approaches solve existing computer vision problems, our method serves to alleviate the human effort required for accurate foreground segmentation. In this respect our work is related to that of Agarwala et al.[2] where a human user interactively uses curve tracking tools to produce rotoscope sequences, saving a huge amount of labor with respect to prior methods.

Our contribution, then, will be apparent in the ease of ground truth production for domains that typically require significant effort and will, hopefully, lead to more complete and commonplace use of ground-truth both in the development and analysis of vision algorithms.

1.1 Applications

A straightforward method for obtaining high quality foreground/background classification has several direct benefits related to algorithm analysis. In addition, the methodology described here suggests potential directions for new background modeling algorithms.

Performance Evaluation in other fields Recently there has been a significant effort dedicated at comparing gait recognition performance [16] on a dataset collected at the University of South Florida. Most gait recognition algorithms rely on background subtraction to compute the binarized silhouette of the person from which different gait features such as width of the outer contour of the silhouette [12] or moment features [13] are extracted. These features can be affected differently based on the quality of the foreground segmentation. In order to assess the absolute goodness of a gait recognition algorithm independently of the specific background subtraction method used, it is important to extract the binarized silhouettes as accurately as possible.

Performance Evaluation in Background Subtraction Very few quantitative evaluations are published in background subtraction. Exceptions include Migdal and Grimson [15], Erdem et al. [6] and Black et al [3]. This is the case despite an increased awareness about the importance of performance evaluation in background subtraction, as exemplified by the dedicated PETS workshop series.

The output of our method can be used directly to bracket the error of a background segmentation method: calling the true segmentation Y_{True}, the output of our method Y_{Boost}

and that of the tested method Y, one has:

$$\mathrm{E}(Y, Y_{\mathrm{Boost}}) - \mathrm{E}(Y_{\mathrm{True}}, Y_{\mathrm{Boost}}) \;\leq\; \mathrm{E}(Y_{\mathrm{True}}, Y) \;\leq\; \mathrm{E}(Y, Y_{\mathrm{Boost}}) + \mathrm{E}(Y_{\mathrm{True}}, Y_{\mathrm{Boost}}),$$

where the error measure E can be the proportion of misclassified pixels or any other distance measure that verifies the triangular inequality. In these inequalities, $\mathrm{E}(Y, Y_{\mathrm{Boost}})$ is computed directly from Y_{Boost} and Y, while $\mathrm{E}(Y_{\mathrm{True}}, Y_{\mathrm{Boost}})$ has been estimated from the small amount of available hand-segmented data that was used by our method.The bracketing above may be preferred to an estimate of $\mathrm{E}(Y_{\mathrm{True}}, Y)$ obtained from the small amount of available hand-segmented data, because it measures the performance over the whole sequence rather than over only a few frames.

This bracketing is only useful, of course, as long as $\mathrm{E}(Y_{\mathrm{True}}, Y_{\mathrm{Boost}})$ is much smaller than $\mathrm{E}(Y_{\mathrm{True}}, Y)$, which is seen empirically to be the case in section 3. This approach to performance evaluation is only valid as long as BGS methods do not reach errors as low as those of our semi-automatic method, which is currently the case. Our endeavor is that this situation lasts as *little* as possible and that our method contributes to the development of high-performance BGS methods.

Development of new background subtraction methods It has not been possible to treat background subtraction as a supervised learning problem until now because training data was not available.[1] The method we present in this paper allows to produce high quality foreground segmentation that can be used by supervised learning methods that are able to cope with a few fractions of percent of error [19].

Finally, our work gives some insight into the background segmentation problem. The composition of the boosted classifiers suggests an answer to the question: "what are the useful features in background subtraction in a particular sequence?"

2 Methodology

Having introduced the principle of labeling, learning and validation rounds, we detail how Adaboost is used to learn a classifier.

2.1 Supervised learning with Adaboost

We use Adaboost [10, 20] for many reasons, one of them being that it has been studied extensively [18, 20, 11, 17] and has been observed to generalize well. Moreover, it requires a single parameter to be set, the number of training rounds T.

The training process in Adaboost results in a classifier

$$H(X) = \mathrm{Sign}\left(\sum_{t=1}^{T} \alpha_t h_t(X)\right) \in \{-1, 1\}, \tag{1}$$

where $X \in \mathscr{X}$ is the observed data used in classification, $h_t : \mathscr{X} \longrightarrow [-1, 1]$ is a "weak classifier" belonging to a class of functions $\mathscr{H}$ and $\alpha_t \in \mathbb{R}$ is a weighing coefficient. G returns $+1$ to indicate foreground and -1 to indicate background.

[1] We are only aware of a single well-known video sequence for which hundreds of frames have been hand-labeled, the "hall-monitor" sequence [6]

Adaboost requires that the weak classifiers perform "better than guessing." Mathematically, this means that there exists a positive ε (which does not need to be known), such that, given a sample of data $(X_1,y_1),...,(X_N,y_N)$, where $y_n \in \{-1,1\}$ represents the class (background and foreground, in our case) of input X_n, and given positive weights $D(1),\ldots,D(N)$ that sum to one, there exists a classifier $h \in \mathscr{H}$ such that its error

$$\sum_{n=1}^{N} D(n)\,[\![h(X_n)y_n < 0]\!]$$

is less than $1/2-\varepsilon$, where $[\![h(X_n)y_n < 0]\!]$ is 1 (resp. 0) if $h(X_n)y_n$ is negative or not, i.e if h wrongly (resp. correctly) predicts the class of X_n. If this assumption holds, then the classifier (1) built by Adaboost will have an error on the training data that decreases exponentially with T.

The input X may e.g. be the RGB values and location of a pixel, and may also include information on its spatial and temporal neighborhood. At most, X could include, beyond the pixel location (x,y) and RGB value, the whole image and perhaps the whole sequence too. What the set $\mathscr{X}$ is exactly is not important here because Adaboost only "sees" the values $h(X)$, for $h \in \mathscr{H}$, and not X itself.

The weak classifiers h_t and weights α_t in Equation (1) are determined by the Adaboost rule at the t^{th} training step. The training data consists of examples $(X_1,y_1),...,(X_N,y_N)$. At each training step, Adaboost chooses the classifiers $h_t \in \mathscr{H}$ and weights $\alpha_t \in \mathbb{R}$ that minimize a criterion related to the error. In the present work, we use the criterion used in [20, Sec. 4].

2.2 Image filters as weak classifiers

We now detail how image filters, i.e. image processing operations, can be used as weak classifiers suitable for Adaboost. Assume we have image filters $f_1,\ldots,f_M$ (listed below) that produce, for a given pixel location and value $X \in \mathscr{X}$, a value $f_m(X) \in \mathbb{R}^2$. Then, for every $m \in \{1,\ldots,M\}$ and every threshold $\tau \in \mathbb{R}$, we define the weak classifier

$$h^{m,\tau}(X) = \text{Sigmoid}(f_m(X) - \tau). \tag{2}$$

The set of weak classifiers is: $\mathscr{H} = \{h^{m,\tau} \mid 1 \le m \le M,\, \tau \in \mathbb{R}\}$. We may now detail the filters that are used in our experiments.

2.2.1 Spatial Correlation filter

Classification with these filters is based on spatial correlation of different sized neighborhoods of the input images with a mean background-only image obtained from the beginning of the sequence. The correlation for each pixel in the output image is computed as:

$$f_m^{\text{Corr}}(X) = \text{Corr}\,(B(N_m,X), T(N_m,X)), \tag{3}$$

where $B(N_m,X)$ (resp. $T(N_m,X)$) is a neighborhood of width N_m around X in the input (resp. mean background) image. The correlation will be low in foreground regions and high in background regions. We used nine different neighborhood sizes $N_m \in \{3, 5, 7, 9, 11, 15, 21, 27, 33\}$, leading to varying levels of smoothing in the correlation image outputs.

[2] or $\{1,\ldots,255\}$.

2.2.2 Spatio-temporal Filters

In these filters, image pixels are classified based on spatio-temporal consistency in successive images. The output image is generated as follows: the current frame t is spatially smoothed by a binomial filter of width σ, resulting in values $\hat{X}_t^{\sigma}$. These values are time-smoothed using a first order AR filter:$\hat{X}_t^{\sigma,\lambda} = \lambda \hat{X}_t^{\sigma} + (1-\lambda)\hat{X}_{t-1}^{\sigma}$. Finally the image filter is

$$f_m^{ST}(X_t) = |\hat{X}_t^{\sigma,\lambda} - \hat{X}_{t-1}^{\sigma,\lambda}| \tag{4}$$

We use 16 filters, corresponding to $\sigma \in \{0,2,4,16\}$ and $\lambda \in \{0,1/2,4/5,16/17\}$

2.2.3 Pixel-wise probabilistic filters

Classification with these filters is based on the probability of the current RGB (or HSV or Laplacian of Gaussian (LOG)) value X at a given pixel to belong to the background, assuming a kernel probability model:

$$f_m^{\text{Kernel}}(X) = \sum_{i=1}^{P} \Pi_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{m,j}} e^{-\left(X_j - Z_{i,j}\right)^2 / 2\sigma_{m,j}^2}, \tag{5}$$

where $Z_1, \ldots, Z_P$ are $d = 3$-dimensional vectors of RGB (or HSV or LOG) background values observed in the first P frames of the sequence, which are supposed to be background-only. The parameter $\sigma_{m,j}$ for each pixel is allowed to take 10 different values around the value suggested in in Elgammal et al [5]. We thus have $10 \times 3 = 30$ different pixel-wise probabilistic filters at our disposition.

2.2.4 Morphological operators

Morphological operators are applied at each training step $t \in \{2, \ldots, T\}$ to the current output of the unthresholded classifier,

$$H_t(X_n) = \sum_{s=1}^{t-1} \alpha_s h_s(X_n), \; n \in \{1, \ldots, N\}.$$

We use grey-level operators of erosion, dilation, opening and closing, with radii of 1, 2, 3 and 4 pixels, which result in grey-level images. There are thus 16 morphological operators at our disposition.

Note that using morphological operators in this fashion changes theoretically the boosting framework, since the set of weak classifiers varies during boosting. Our preliminary study tends to show that this change has no adverse consequence in theory and that it may be useful in practice.

Altogether, there are (9+16+30+16=) 71 image filters available and each one yields a family of weak classifiers indexed by the threshold τ in Equation (2).

3 Experiments

Having detailed our methodology, we now show its results on two sequences. Our implementation runs under Octave [4] and Matlab.

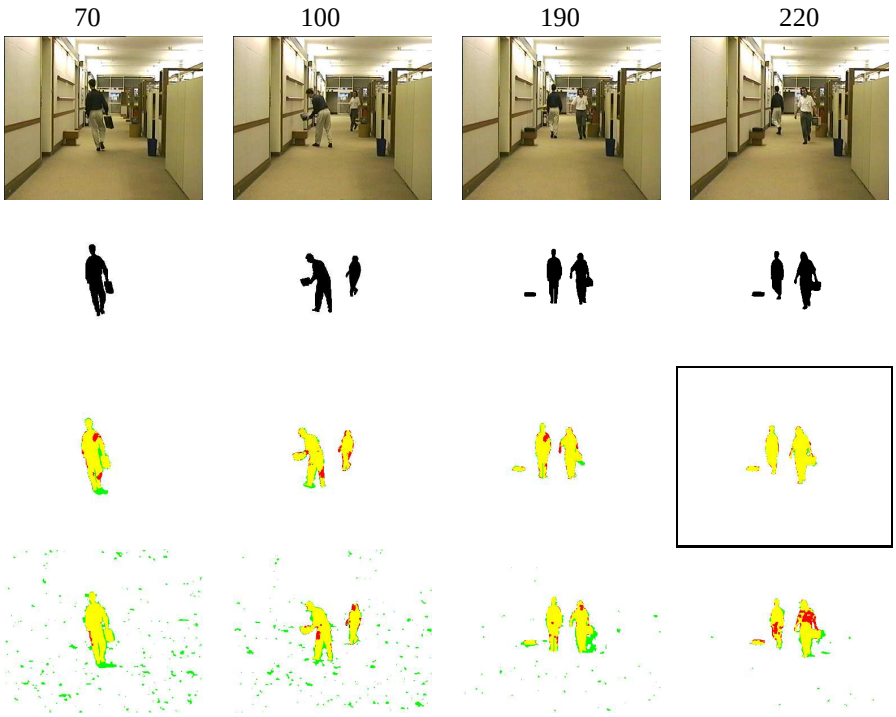


Figure 1: **Row 1:** "Hall-monitor" images 70, 100, 190 and 220. **Row 2:** ground-truth. **Row 3:** Output of boosted classifier. Image 220 (circled) was used for training, the others for validation. White and yellow (light gray) are correctly classified background and foreground while red (dark gray) and green (gray) are misclassified foreground and background.**Row 4:** Output of the method of Elgammal et al [5].

Hall-monitor The first dataset used was the well-known 300-image "hall-monitor" sequence, shown in the top row of Figure 1. Prof. Erdem [6] provided hand-labeled ground-truth for the left sides of images 32-240 while the right side of images 70, 100, 130, 160, 190 and 220 were hand-labeled by the authors (second row of Figure 1). These images constitute the ground-truth for training and validation.

The third row of this figure shows the results of our method after 100 boosting steps, trained using only image 220. In white and yellow (light gray) are correctly classified background and foreground pixels, respectively, while red (dark gray) and green (gray) denote false positives and negatives. During training, the error on the training data decreased from 0.75% to 0.34%. Table 1 shows the contribution of each type of weak classifier in the boosted classifier. Note that the correlation filters are not used at all. The validation errors (computed on the five other images) are 11.4% (false negatives), 0.43% (false positives) and 0.81% (overall). The false negative rate is the ratio of the number of pixels wrongly classified as negative to the number of true positives, the false positive rate is the ratio of number of pixels wrongly classified as positive to the number of true negatives and the overall error is the fraction of misclassified pixels.

Family	Kernel				Spatio Temporal			Corr.	Morph.
Colorspace	RGB	HSV	HS	LOG	RGB	HS	V	RGB	-
Number	24	15	14	12	6	5	4	0	20
Weights (%)	23.3	14.9	5.8	9.2	2.0	2.9	1.3	0	40.7
Total Num.	65				15			0	20
T. Wgt. (%)	53.2				6.2			0	40.7

Table 1: Contributions of the individual image filter families and colorspaces, in number of weak classifiers and total absolute weight. These values come from the classifier trained on image 220 of the Hall-Monitor sequence.

The results[3] were compared with the unsupervised Kernel-based approach of Elgammal et al. [5] (shown in row 4). The false positive rate using their method was found to be 0.37%, the false negative rate was 15.7% while the overall error was 1.76%. As can be seen, our approach gives foreground segmentation which is much more closer to the ground truth by leveraging the power of a larger arsenal of unsupervised learners.

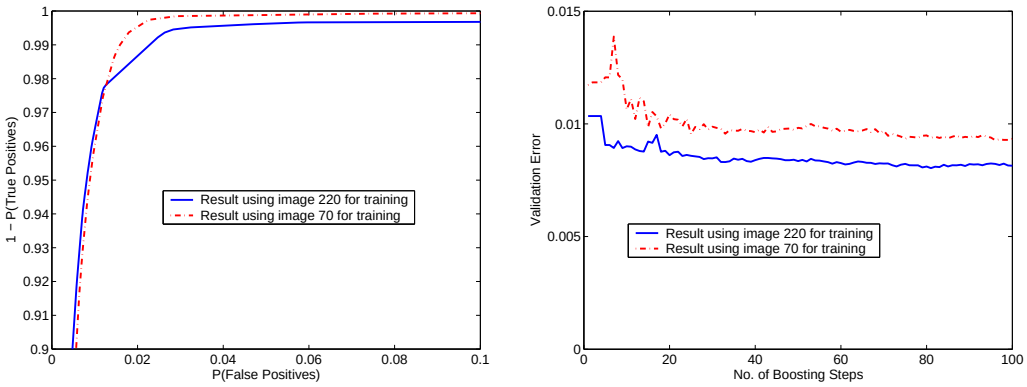
We found that the choice of the training image influences the performance: if image 70 was used instead of 220 the error values obtained were higher. Specifically we got a false negative rate of 15.7% (vs. 11.4% when training with image 220), a false positive rate of 0.37% (vs. 0.43%) and an overall error rate of 0.93% (vs. 0.81%). This is to be expected however, since image 70 contains much less foreground than image 220.

The ROC curves corresponding to both situations are shown in Figure 3(a). Note that, due to the prevalence of negatives in the data, the error is measured in the left part of the ROC curve. This explains that the error obtained when training with image 220 is lower than that obtained using image 70 while the ROC of the latter is often above that of the former. The validation error tends to decrease with the number of boosting steps as shown in Figure 3(b), which shows that Adaboost does not overfit for this dataset.

In other experiments, using different subsets of five training and one validation image, we found that the validation errors do not change significantly. These results are better than with a single image, but not much better, showing that a single labeled images may suffice to obtain very good results.

MIT indoor data This image sequence was taken at the MIT Artificial Intelligence Laboratory and provided by Joshua Migdal [15]. It consists of a person entering the field of view of the camera on the right, walking to the left out of the camera's view. The top row of Figure 3 shows a few images from the sequence. Hand labeled ground truth was provided for frames 115, 120, 125, 130 and 135, four of which have been shown in the second row of Figure 3. These images constitute the ground-truth for training and validation. The third row of this figure shows the results of our method after 50 boosting steps, trained using images 115, 120, 125 and 130. We found that the validation error did not decrease by increasing the number of boosting steps. In white and yellow (light gray) are correctly classified background and foreground pixels, respectively, while red (dark gray) and green (gray) denote false positives and negatives. The validation error

[3] We experimented extensively with the σ parameter of Eq. (5) and with the "α" parameters We may ask the authors of [5] whether better results can be obtained.



(a) (b)

Figure 2: (a) ROC curves corresponding to training images 70 (red dash-dotted curve) and 220 (blue solid line). (b) Validation error as a function of number of boosting steps corresponding to training images 70 (red dash-dotted curve) and 220 (blue solid line).

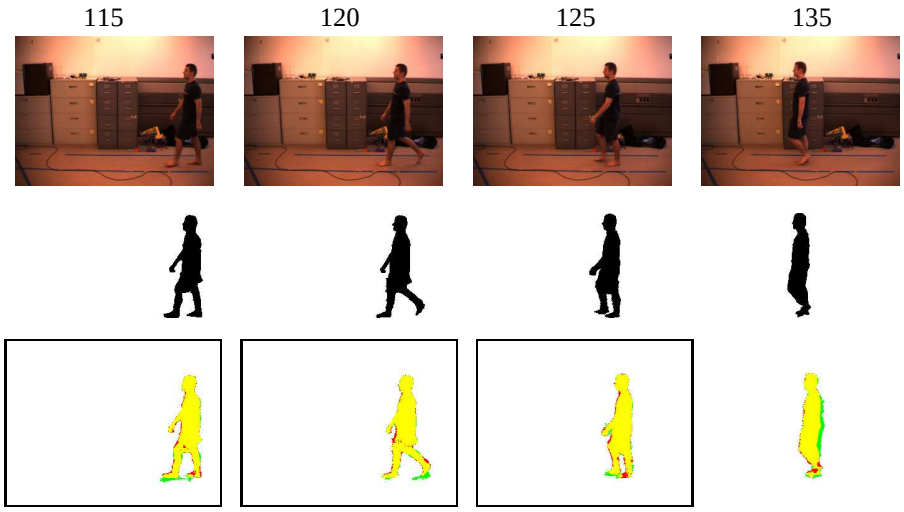


Figure 3: **Row 1:** MIT Indoor images 115,120,125 and 135. **Row 2:** ground-truth. **Row 3:** Output of boosted classifier. Validation uses image 135, training, image 115, 120, 125 and 130 (circled).

computed on the test image was 0.9% (false positives), 5% (false negatives) and 1.1% (overall).

4 Conclusions

In this paper, we presented an approach based on supervised learning to generate foreground segmentation for video sequences with much less labor than hand-labeling and much higher accuracy than fully automatic methods. It would be interesting to compare

the error of our method with the variability of manually segmented "ground truth", as suggested by Unnikrishnan and Hebert [22].

Our objective is, in the future, to improve the quality and further reduce the user's burden. We believe this is possible by using different image filters as base classifiers, for example, more advanced morphological operators and spatio-temporal operators. Since Adaboost accepts any type of image filter, these extensions can easily be integrated in the proposed framework. If this objective is achieved, the difficulty of obtaining ground-truth for whole sequences would be immensely reduced, with the following benefits:

- Quantitative evaluation of unsupervised background subtraction methods, as done in [6], would be possible with very little extra labor.
- Methods relying on segmented background could be quantitatively evaluated independently of any given background subtraction method; again, this would benefit the development of such methods.
- The availability of large amounts of segmentation data enables new approaches to background subtraction. Research in that area, constrained by the lack of ground-truth data, resulted in methods that essentially solve a single-class learning problem. We hope the data resulting from our method will induce researchers to address background subtraction with tools of supervised learning.

References

[1] H. Adams, S. Singh, and D. Strelow. An empirical comparison of methods for image-base motion estimation. In *IROS*, Lausanne, Switzerland, 2002.

[2] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, 23(3):584–591, 2004.

[3] P. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. *IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 125–132, 2003.

[4] John W. Eaton. *GNU Octave Manual*. Network Theory Limited, 2002.

[5] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *proc. of the IEEE*, 90(7):1151–1163, 2002.

[6] C. E. Erdem, A. M. Tekalp, and B. Sankur. Metrics for performance evaluation of video object segmentation and tracking without ground-truth. In *IEEE Intl. Conf. on Image Processing (ICIP)*, 2001.

[7] R. Fisher, J. Santos-Victor, and J. Crowley. CAVIAR: Context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, 2003. EC IST project IST 2001 37540.

[8] A. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *Proceedings of the International Conference on Computer Vision*, October 2003.

[9] W. T. Freeman and E. C. Pasztor. Learning low-level vision. In *International Conference on Computer Vision*, volume 2, pages 1182–1189, 1999.

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[11] J. Friedman, T. Hastie, and R.J. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 2:337–374, 2000.

[12] A. Kale, N. Cuntoor, B.Yegnanarayana, A.N. Rajagopalan, and R. Chellappa. Gait analysis for human identification. *Proceedings of the 3rd International conference on AVBPA*, 2003.

[13] L. Lee and W.E.L. Grimson. Gait analysis for recognition and classification. *Proceedings of the IEEE Conference on Face and Gesture Recognition*, pages 155–161, 2002.

[14] A.M. Martinez and R. Benavente. The AR face database. 24, Computer Vision Center of the Universitat Autònoma de Barcelona, 1998.

[15] J. Migdal and W.E.L Grimson. Background subtraction using markov thresholds. *Proceedings of IEEE Workshop on Motion and Video Computing*, January 2005.

[16] P. J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. W. Bowyer. The gait identification challenge problem: Data sets and baseline algorithm. *Proc of the International Conference on Pattern Recognition*, 2002.

[17] C. Rudin, R. Schapire, and I. Daubechie. Analysis of boosting algorithms using the smooth margin function: A study of three algorithms. *submitted*, 2004.

[18] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML*, 1997.

[19] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *proc. ICML*, 2002.

[20] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[21] A. Torralba, K. P. Murphy, and W. T. Freeman. The MIT-CSAIL database of objects and scenes. web.mit.edu/torralba/www/database.html.

[22] Ranjith Unnikrishnan and Martial Hebert. Measures of similarity. In *Seventh IEEE Workshop on Applications of Computer Vision*, pages 394–400, January 2005.

[23] P. Viola and M. Jones. Robust real-time object detection. In *Proc. ICCV workshop on statistical and computational theories of vision*, 2001.

[24] P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, pages 734–741, 2003.