Real-time Stable Texture Regions Extraction for Motion-Based Object Segmentation

Dubravko Ćulibrk
http://www.dubravkoculibrk.org
Borislav Antić
http://www.ursusgroup.com
Vladimir Crnojević
http://www.ursusgroup.com

Faculty of Technical Sciences
University od Novi Sad
Trg Dositeja Obradovica 6
21000 Novi Sad
Serbia
http://www.ftn.ns.ac.rs

Abstract

Object segmentation is a fundamental task in various computer vision applications. Although used extensively for object recognition, texture has lately been ignored as a feature used for background modelling and object segmentation. The complexity of working with texture descriptors for segmentation in videos is two-fold: the descriptive features cannot be calculated in real time and features extracted based on arbitrarily chosen regions or blocks in the frame are not stable enough to allow for building models sufficiently accurate, yet simple enough to be used for real-time segmentation.

The paper proposes an approach that can be used to detect regions of texture, stable enough to be modelled using probabilistic models commonly used for foreground segmentation. Based on the evaluated stable texture regions, a discriminative texture descriptor is proposed that can be evaluated in real time. Features based on this descriptor are able to enhance the segmentation performance of segmentation algorithms on some very "hard" sequences.

1 Introduction

Object segmentation plays a key role in scene understanding and various computer vision applications. Moving object segmentation in particular is extensively used in automated surveillance [13][21].

Over the past 20 years significant research effort has been spent on developing ways of segmenting the moving foreground objects from the background. When video is captured from a stationary camera, the background is expected to be stationary to a degree and an adaptive model can be built to serve as basis for segmentation.

For reasons of efficiency most adaptive models are learnt on a per pixel basis. The video sequence is treated as a set of pixel processes, as Stauffer and Grimson dubbed them [21]. The disregard for the spatial information contained in the frames leads to noisy segmentation output that is later morphologically processed to produce the final segmentation. In addition, treating the sequence as a set of single pixel changes leads to an inability to represent the texture of the objects in the background and perform adequate segmentation based on this important property of the objects.

BMVC 2009 doi:10.5244/C.23.17

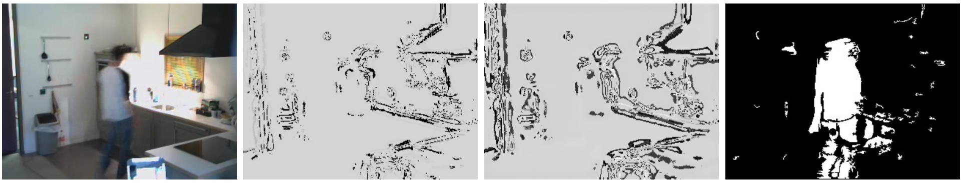


Figure 1: Stable texture regions based segmentation: frame from a sequence (left), region stability map (center-left), extracted texture descriptors (center-right), segmented foreground (right). Lighter regions in the stability map are more stable than the dark ones.

Joint domain-space modelling [19], as well as modelling of the sequence at two different scales, combined with advanced features adopted from the pattern recognition research community, has been suggested as a way to enhance the segmentation and escape the limitations of single-pixel-based models[3]. These approaches still fail to consider the texture properties of the objects in the background and suffer from the noise introduced by joining objects being modelled as one.

Texture of the background can be used to build models in a more informed way. The complexity of working with texture descriptors for segmentation in videos is two-fold: the descriptive features cannot be calculated in real time and features extracted based on arbitrarily chosen regions or blocks in the frame are not stable enough to allow for building models sufficiently accurate, yet simple enough to be used for real-time segmentation. Large stable regions in the texture of the background allow for larger scale modelling and more precise extraction of texture related features. Conversely, unstable regions in terms of texture correspond to transitions between different objects (or different object patches) in the background, and the features used should be extracted for each of the objects separately. An approach to background modelling and foreground segmentation derived from these principles is proposed in the paper. In each frame of the sequence stable texture regions are determined, texture descriptor (feature) values extracted using the stable regions information and used to model the background and segment the foreground using standard probabilistic approaches. The process is illustrated in Fig. 1, where the brighter patches in the stability map correspond to more stable regions, in terms of texture.

While the principle of determining stable regions proposed here is applicable to any feature set and a number of probabilistic background modelling methods could be used to model feature related statistics, the proposed approach is evaluated using texture descriptors designed for fast calculation and a Mixture of Gaussians model [11] adapted from the OpenCV implementation [1]. The segmentation algorithm has been evaluated on several diverse sets of sequences. Quantitative evaluation has been performed indicating improved performance of the proposed approach over the original Mixture of Gaussians relying on single-pixel derived information.

The rest of the paper is organized as follows: Section 2 reviews the relevant published work. Section 3 describes the proposed background modelling approach. Section 4 presents the experiments conducted and segmentation results achieved. Concluding remarks and some directions for future work can be found in Section 5.

2 Background and Related Work

The earliest segmentation approaches relied on background models in the form of a background image obtained by low-pass filtering from the frames of the sequence. Rosin provides a good overview [17]. The application of such approaches has lately been limited to relatively easy segmentation problems[4].

The state of the art in segmentation are probabilistic methods that approximate the statistics of the pixels corresponding to background and use them to distinguish between the background and the foreground[6][21][19][13]. For a common frame of reference, the early filter-based approaches can be thought of as probabilistic approaches that model the background at a pixel using a single unimodal probability density function, described by its peak value, and a single global threshold to determine how likely is the currently observed value of the pixel to be due to a background object. The first probabilistic methods introduced the threshold dependant on local variance of the pixel values [9] and single-mode parametric modelling and then moved toward multimodal parametric representation of pixel statistics with Mixtures of Gaussians. The Gausian Mixture model of Staufer and Grimson [21], remains one of the most widely used and compared against. Recently more sophisticated non-parametric estimation methods have been proposed in an effort to achieve better and more accurate background models [19][13][6][8]. However, Mixtures of Gaussians remain the methodology most widely used in applications, probably due to the fact that the new approaches are seen as limited improvements in terms of segmentation performance.

Most background models are built on single-pixel basis. The sequence is treated as a set of single-pixel values changing over time, what Stauffer and Grimson [21] referred to as pixel processes. Region based models have been emerging recently, since the use of spatial information adds a new level of computational complexity to the segmentation methods [8][19][20][5]. It is interesting to note that simple incorporation of the spatial information in the model as Sheikh and Shah propose[19], leads to models that are highly sensitive to small camera vibrations, which commonly occur in surveillance videos. This inability seems to stem from the fact that they disregard the fact that the features of two joining objects cannot reasonably be expected to have similar distribution of feature values, although the objects are close to one another.

Some researches use block-based approaches instead of pixel-based approaches for background modelling and subtraction. In a block-based approach, an image is divided into overlapped or non-overlapped blocks, and specific block features are used for background modelling. Since a block can monitor more global changes in the scene than a single pixel, block-based approaches are less sensitive to local movements in the background. Primary limitation of block-based approaches is the fact that only coarse-resolution foreground can be extracted, making them unsuitable for applications that require detailed shape information. In addition, such an approach results in the complex problem of selecting the correct block size (scale) that should be used for background modelling.

While pixel-based approaches rely mostly on pixel intensity and color approaches for segmentation, block based approaches allow for extraction of more sophisticated features. Mason *et al*. [14] calculated edge and color histograms in each block as features to describe the block, and histogram similarity is computed to detect the foreground region. Monnet and Duric [15] used incremental PCA and an online auto-regressive model to predict a dynamic scene. Heikkilä [10] used local binary pattern (LBP) [16] histograms to capture background statistics of each block. Chen *et al*. propose contrast histograms as block features used for segmentation [3]. Relying on blocks reduces the computational load required to evaluate features for the whole frame, but creates the problem of transition from the block based foreground map to more accurate segmentation and effectively discards pixel-level features for the sake of block based features. Some authors even propose maintaining separate models at different scales [3].

The issue of texture-based object segmentation in videos, as it is considered here, should not be confused with the dynamic texture approach. The latter is aimed at modelling the textures created by dynamic background objects such as rippling water and smoke [7][25][2], rather than modelling the static texture of background objects and distinguishing it from the texture of moving foreground objects in the more classical framework of probabilistic modelling.

The idea of determining the stable texture regions of the background in order to determine the scale at which the background model should be used has not been explored before, to the best of our knowledge. Integral images [24] are used to calculate the stable regions for each frame of the sequence in real time. This enables the methodology proposed to extract accurate features, using only pixels pertinent to background objects. In addition, novel, efficient to calculate, texture descriptors proposed here.

3 Real-time Stable Texture Regions Detection

The basic idea behind the proposed approach is that an arbitrary region around each location in the frame can be used to extract features pertinent to that location only if the texture within the region is perceived as stable. The texture within the inner regions of objects in the scene should be stable and can be modelled at coarse as well as fine scales. The features of the background should be extracted based on progressively smaller regions as the location approaches the boundaries of the objects.

Such a background model benefits from the largest possible amount of information extracted from the pixels in the neighborhood of the location modelled, but does not include information from unrelated pixels near to the location (pixels related to other objects).

The size of the stable texture regions can be determined online, using integral images. Once the size of the stable-texture region at each location is determined, arbitrary descriptive features can be extracted from the frames of the sequence. In addition, one can choose any of a number of probabilistic methods[8][21][13][6] to learn the statistics of the features.

Here, the approach has been used to enhance the Mixture of Gaussians based segmentation [11]. Texture descriptive features, that can be evaluated efficiently are proposed to guide the segmentation.

3.1 Detecting Stable Texture

The crucial stage of the proposed background modelling and object segmentation method is the detection of the regions in the frames with stable texture. Proposed texture descriptors are based on the following observation made by Varma and Garg[22].

Given an image I, let $\mu(\bar{B}(x,r))$ be the sum of all pixel intensities that lie within a closed disk $\bar{B}$ of radius r centered at an image point x, i.e. $\mu(\bar{B}(x,r)) = \sum_{\|y-x\|\leq r} I(y)$. They show empirically that

$$\log\mu(\bar{B}(x,r)) = D(x)\log r + L(x) \qquad (1)$$

where $D(x)$ is the local fractal dimension, and $L(x)$ is the fractal length. Experimental evaluation conducted by the same authors showed that $L(x)$ is a good discriminative feature for texture classification. Varma and Garg read it directly off a graph, having plotted the values of $\log \mu(\bar{B}(x,r))$ for progressively larger r.

Equation 1 is intuitive even if it is not linked with the fractal properties. Observe that:

$$\begin{aligned} \mu(\bar{B}(x,r)) &= r^2 \pi \bar{I}(x,r) \\ \Rightarrow \log \mu(\bar{B}(x,r)) &= 2\log r + \log \pi \bar{I}(x,r) \end{aligned} \tag{2}$$

where $\bar{I}(x,r)$ represents the average intensity of the pixels in the disk. Bearing the Eq. 2 in mind, it is to be expected that the logarithm of the pixel intensity sum would change somewhat linearly as the disk size grows, if the disks encompass pixels with the same texture. On the other hand, if the texture changes between successive disk sizes, this trend would be violated.

In the initial stage of the approach proposed here, this observed property of the texture is used to detect the size of the region around each pixel location that exhibits no significant changes in the texture. Rather than using circular disks, square regions (blocks) are used to enable efficient computation. The appropriate from of Eq. 2 for this case is:

$$\begin{aligned} \mu(\bar{B}(x,r)) &= 4r^2 \bar{I}(x,r) \\ \Rightarrow \log \mu(\bar{B}(x,r)) &= 2\log r + \log 4\bar{I}(x,r) \end{aligned} \tag{3}$$

Within the stable texture regions, there should not be any abrupt changes in the value of $L(x)$. Since any sudden changes in the graph gradient $D(x)$ would also cause the value of the intercept $L(x)$ to change, one can detect the sudden changes in the texture by monitoring how the average intensity changes when the block size is increased. The proposed approach for detecting stable texture regions and subsequently the maximum scale at which a pixel can be effectively modelled proceeds as follows:

1. The logarithms of sums of intensity values for successive blocks of "radius" $r \in \{1,2,\ldots,R\}$ centered at each pixel location x are computed.

2. The derivative of these values across the scales is computed.

3. If the variance of the derivative values for a location is less than the mean variance for the whole frame the region is marked as stable.

4. If not, the maximum derivative value in the sequence is selected as the radius at which the change in the texture is encountered and the radius of the stable region around location x set to the radius of the last stable block.

The process is illustrated in Fig. 1. The figure shows the detection results (texture stability map) for sample sequence. The term "radius" is used loosely, and describes the distance between the center of the square and the closest point on its sides.

To enable the computation of the stability map for each frame of the sequence in real time, integral images, as proposed by Viola and Jones [24], can be used. The integral image of an image at location x,y contains the sum of the pixels above and to the left of x,y, inclusive

$$II(x,y) = \sum_{x' \leq x, y' \leq y} I(x,y) \tag{4}$$

where $II(x,y)$ is the integral image and $I(x,y)$ is the original image. Using the following pair of recurrences:

$$S(x,y) = S(x,y-1) + I(x,y) \tag{5}$$

$$II(x,y) = II(x-1,y) + S(x,y) \tag{6}$$

(where $S(x,y)$ is the cumulative row sum, $S(x,-1) = 0$ and $II(-1,y) = 0$) the integral image can be computed in one pass over the original image. Using the integral image any rectangular sum can be computed in just four array references.

3.2 Texture-related Features for Background Modelling

Any number of descriptive features can be extracted once the stable-texture regions are known. The discussion within the paper, however, is limited to a single type of texture-descriptive features, for the sake of brevity.

Texture is an important characteristic of objects that cannot be modelled within the pixel-based framework. Not surprisingly, it is often at the focus of block-based and larger scale models [3]. Extensive research has been done in the domain of texture description for purposes of classification. Leung and Malik [12], Varma and Zisserman [23] and Serre *et al.* [18] use large banks of filters to extract texture related features (48 and 38 filters respectively). The banks are to mostly composed of oriented filters, with different orientation and scale. Features based on large filter banks are unfortunately not applicable to video object segmentation due to computation complexity involved. Recent work of Serre *et al.* state that the typical running time of their algorithm is some 10s per image.

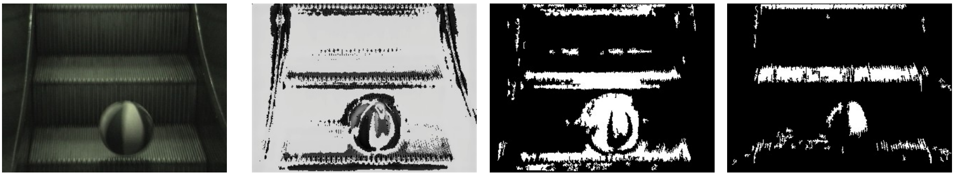
Features based on the observations of Varma and Garg [22], however, can be computed efficiently using integral images. They showed that the y-axis intercept of the line regressed from the points on their plots can be used to classify texture effectively. Through the experiments performed within the research discussed here it became apparent that the feature proposed by Varma and Garg, although sufficient to achieve texture classification, was not discriminative enough to enable accurate segmentation in all cases considered. To create a more discriminative feature, the plots' axes were exchanged and $\log(r)$ plotted against the intensity sums. The y-axis intercept for such plots proved more discriminative, since it is more sensitive to small changes in the line gradient.

Rather than reading the intercept off the plot, linear least mean squares regression is proposed here to calculate the value of the intercept. The feature values are evaluated according to equation 7

$$f(x) = \frac{(\sum_r u)(\sum_r uv) - (\sum_r v)(\sum_r u^2)}{(\sum_r u)^2 - R_s(\sum_r u^2)}, u = \log\mu(\bar{B}(x,r)), v = \log r \tag{7}$$

where R_s is the maximum stable region size and $f(x)$ is the extracted feature value for pixel x.

Any type of background modelling approach can be used to learn the statistics of the extracted features and perform the segmentation. Mixture of Gaussians model, as it has been extended by Kaewtrakulpong and Bowden [11], has been selected as widely used and efficient approach. The evaluated texture features are provided as input to the model, instead of intensity values, in order to achieve texture-based segmentation.



(a) Original frame. (b) Extracted features. (c) Proposed approach. (d) MoG result.

Figure 2: Segmentation of an escalator sequence.

4 Experiments and Results

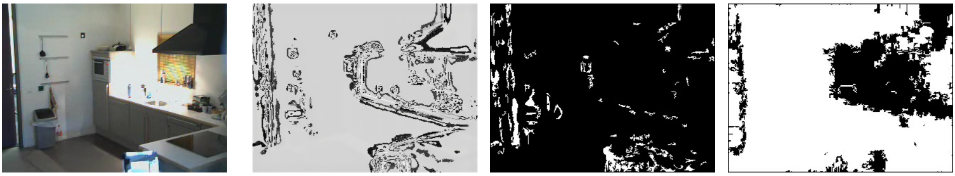
To evaluate the segmentation performance of the proposed methodology, the Mixture of Gaussians segmentation [11] as implemented within the OpenCV framework[1] has been extended to include texture-based features proposed in Section 3.2 and compared to the original segmentation approach based on pixel intensity values.

Several sets of sequences have been used to evaluate the segmentation performance. The first set consists of sequences that contain strongly textured objects (escalators) in the background and have been proven to be difficult to segment within the MoG framework [3]. The second and third set of sequences were provided as part of the challenges for the International Conference on Distributed and Smart Cameras (`http://www.icdsc.org`). The second set is concerned with a typical smart home scenario, while the third provides sequences taken at a basketball game. The two data sets represent the opposite ends of the spectrum when it comes to the number and the size of the objects in the sequences. Both are concerned with segmentation for the purpose of human tracking and activity detection. The final set is comprised of sequences taken from traffic surveillance cameras, mounted on bridges and tunnels, and enables the evaluation of the applicability of the approach to outdoor and rigid body segmentation problems. The resolution of the sequences varies across data sets from 320×240 pixels for the first two, 384×288 for the fourth and 800×600 for the third. Intensity values have been used as the basis for segmentation, as they are commonly related to texture.

No fine tuning of the MoG algorithm was performed. The default OpenCV values have been used for both the reference and proposed approach. The texture feature value was scaled to the range of the intensity and incorporated into the segmentation algorithm. Same initial variance was used for both texture and intensity features. The algorithm performs post-processing by removing connected components smaller than 15 pixels and filling holes within the object, for both the reference and proposed approach results. Figures 2, 3, 4 and 5 show sample segmentation results for each data set.

The only parameter provided to the stable region detection procedure is the maximum size of the regions ($S = 2R + 1$). Smaller values yield faster computation and less over-segmentation, and should be used when the objects are smaller. Larger values mean better generalization ability and should be used for larger objects. In the experiments presented R was set to 3 for first and second data sets, and to 9 for the other two.

On an Intel Core2Duo processor, running on a single 2.4GHz core, the segmentation of a single frame of the 320×240 sequences, and maximum block size 7×7 ($R = 3$) pixels, was performed in environ 50ms. The reference MoG approach performed the same task 2 times faster. Increasing the maximum size of the blocks increases the processing time, but segmentation of 10 frames per second can still be achieved for blocks of size 19×19 ($R = 9$).



(a) Original frame. (b) Extracted features. (c) Proposed approach. (d) MoG result.

Figure 3: Segmentation of a smart home sequence.

Method	Escalators	Smart house	Basketball	Traffic
MoG	0.1024	0.3252	0.3358	0.3205
Proposed	0.2211	0.5565	0.7569	0.4782

Table 1: Similarity measure comparison.

MoG segmentation benefited from the introduction of texture-related features on all data sets. For the first data set the proposed features allowed the segmentation to detect more foreground pixels (see Fig. 2). In addition to being more discriminative, the new features made the segmentation far less sensitive to global illumination changes, which occurred due to person entering the scene and occluding the light sources in smart home sequences and due to camera auto gain adjustment in traffic surveillance sequences. Fig. 3 shows this effect occurring when a person is about to enter the frame, for a sample smart home sequence. Finally, introduction of texture-related features enabled the segmentation algorithm to suppress shadows more efficiently. This is illustrated by figures 4 and 5.

4.1 Quantitative Evaluation

For each of the three test sequences a measure of the segmentation accuracy was calculated following the methodology used in [13]. If D is a detected(segmented) region and G the corresponding ground truth, then the similarity measure between these two regions is defined as:

$$S = \frac{D \cap G}{D \cup G} \tag{8}$$

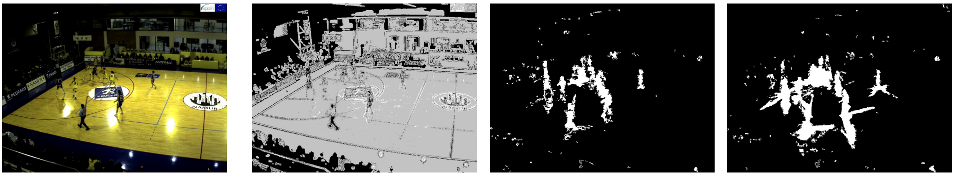
The similarity of the regions (S) reaches the maximum value of 1 if they are the same. Otherwise, it varies between 1 and 0 according to how similar the regions are. S is the measure of the overall misclassification. Similarity has been measured based on randomly selected frames, the ground truth being hand segmented.

Table 1 shows the results obtained for the original MoG method and the proposed method employing texture-based features for sample sequences within each data set.

Quantitative evaluation results show that the proposed methodology improves the segmentation.

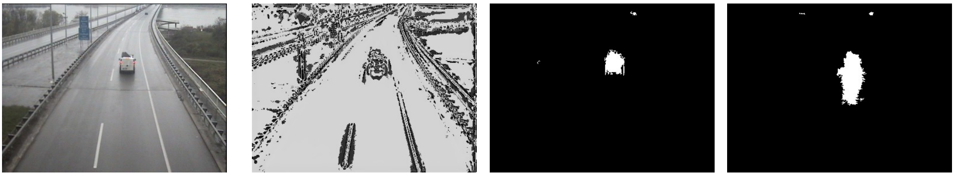
5 Conclusions and Future Work

Although texture plays a key role in the ways humans recognize object, texture-related information has effectively been ignored within mainstream background modelling and moving



(a) Original frame. (b) Extracted features. (c) Proposed approach. (d) MoG result.

Figure 4: Segmentation of a basketball sequence.



(a) Original frame. (b) Extracted features. (c) Proposed approach. (d) MoG result.

Figure 5: Segmentation of a traffic surveillance sequence.

object segmentation research. The paper addressed the problem of determining on-line local stable texture regions and extracting texture-related features that are suitable for background modelling. These features are stable, discriminative enough and can be computed efficiently, allowing for real time foreground segmentation, using any of the standard probabilistic approaches. Experiments conducted on diverse sets of sequences show that the widely used MoG segmentation benefits from the introduction of such features.

A number of improvements can be made, when the proposed methodology is concerned. The stable texture detection described in Section 3.2 can be enhanced if median, rather than the mean filtering was performed to detect the stable regions, since median filtering is more robust to outliers. This, however, requires more computation. As more computational power becomes available, other texture descriptors, as well as features commonly used for object recognition can be explored as basis for segmentation. More importantly, the methodology should be extended to include color information. This venue is currently being pursued by the authors. Finally, since the proposed features can be used with different modelling and segmentation approaches, the benefits to each should be explored and documented.

References

[1] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008.

[2] A.B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pages 1–6, 2007.

[3] Yu-Ting Chen, Chu-Song Chen, Chun-Rong Huang, and Yi-Ping Hung. Efficient hierarchical method for background subtraction. *Pattern Recogn.*, 40(10):2706–2715, 2007. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/j.patcog.2006.11.023.

[4] S.Y. Chien, Y.W. Huang, B.Y. Hsieh, S.Y. Ma, and L.G. Chen. Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. In *IEEE Trans. on Multimedia, vol. 6, pp. 732-748*, 2004.

[5] M. Cristani and V. Murino. A spatial sampling mechanism for effective background subtraction. In *Proc. VISAPP 2007, vol. 2, pp. 403-410*, 2007.

[6] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht. Neural network approach to background modeling for video object segmentation. In *IEEE Trans. on Neural Networks*, volume 18, pages 1614–1627, 2007.

[7] G. Doretto, A. Chiuso, Y. N.Wu, and S. Soatto. Dynamic textures. 51(2):91Ű109, 2003.

[8] A. ElGammal, R. Duraiswami, D. Harwood, and L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. In *Proc. of the IEEE, vol. 90, No. 7, pp. 1151-1163*, 2002.

[9] I. Haritaoglu, D. Harwood, and L. Davis. W4 real-time surveillance of people and their activities. In *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, pp. 809-830*, 2000.

[10] Marko Heikkila. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):657, 2006. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2006.68.

[11] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. In *In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing*. Kluwer Academic Publishers, September 2001.

[12] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44, 2001. ISSN 0920-5691. doi: http://dx.doi.org/10.1023/A:1011126920638.

[13] L. Li, W. Huang, I.Y.H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. In *IEEE Trans. Image Processing, vol. 13, pp. 1459-1472*, 2004.

[14] M. Mason and Z. Duric. Using histograms to detect and track objects in color video. In *AIPR '01: Proceedings of the 30th on Applied Imagery Pattern Recognition Workshop*, page 154, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1245-3.

[15] Antoine Monnet, Anurag Mittal, Nikos Paragios, and Visvanathan Ramesh. Background modeling and subtraction of dynamic scenes. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1305, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.

[16] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2002.1017623.

[17] L.R. Rosin. Thresholding for change detection. In *Proc. of the Sixth International Conference on Computer Vision (ICCV'98)*, 1998.

[18] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2007.56.

[19] Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(11):1778–1792, 2005. ISSN 0162-8828. doi: http://dx.doi.org.ezproxy.fau.edu/10.1109/TPAMI.2005.213.

[20] D. Socek, D. Culibrk, O. Marques, H. Kalva, and B. Furht. A hybrid color-based foreground object detection method for automated marine surveillance. In *Proc. of the Advanced Concepts for Intelligent Vision Systems Conference (ACIVS 2005)*, 2005.

[21] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, pp. 747-757*, 2000.

[22] Manik Varma and Rahul Garg. Locally invariant fractal features for statistical texture classification. In *ICCV '07: Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[23] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62(1-2):61–81, 2005. ISSN 0920-5691. doi: http://dx.doi.org/10.1007/s11263-005-4635-4.

[24] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001. doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.2751.

[25] Jing Zhong and Stan Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 44, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.