A Texture-based Method for Detecting Moving Objects

M. Heikkilä, M. Pietikäinen and J. Heikkilä
Machine Vision Group
Infotech Oulu and Department of Electrical and Information Engineering
P.O. Box 4500 FIN-90014 University of Oulu, Finland
{markot,mkp,jth}@ee.oulu.fi
http://www.ee.oulu.fi/mvg

Abstract

The detection of moving objects from video frames plays an important and often very critical role in different kinds of machine vision applications including human detection and tracking, traffic monitoring, human-machine interfaces and military applications, since it usually is one of the first phases in a system architecture. A common way to detect moving objects is background subtraction. In background subtraction, moving objects are detected by comparing each video frame against an existing model of the scene background. In this paper, we propose a novel block-based algorithm for background subtraction. The algorithm is based on the Local Binary Pattern (LBP) texture measure. Each image block is modelled as a group of weighted adaptive LBP histograms. The algorithm operates in real-time under the assumption of a stationary camera with fixed focal length. It can adapt to inherent changes in scene background and can also handle multimodal backgrounds.

1 Introduction

Background subtraction is often one of the first tasks in machine vision applications, making it a critical part of the system. The output of background subtraction is an input to a higher-level process that can be, for example, the tracking of an identified object. The performance of background subtraction depends mainly on the background modelling technique used to model the scene background. Especially natural scenes put many challenging demands on background modelling since they are usually dynamic in nature including illumination changes, swaying vegetation, rippling water, flickering monitors etc. A robust background modelling algorithm should also handle situations where new stationary objects are introduced to or old ones removed from the scene. Furthermore the shadows of the moving and scene objects can cause problems. Even in a static scene frame-to-frame changes can occur due to noise and camera jitter. Moreover, the background modelling algorithm should operate in real-time. In this paper, we propose a novel texture-based approach to background subtraction. The goal of the new approach was to address all of the above-mentioned difficulties except the handling of shadows

which turned out to be an extremely difficult problem to solve with background modelling.

2 Related Work

Different kinds of methods for detecting moving objects have been proposed in the literature. The most commonly used cue is pixel intensity. One very popular technique is to model each pixel in a video frame with a Gaussian distribution. This is the underlying model for many background subtraction algorithms. A simple technique is to calculate an average image of the scene, to subtract each new video frame from it and to threshold the result. The adaptive version of this algorithm updates the model parameters (mean and covariance) recursively by using a simple adaptive filter. This approach is used in [16]. The previous model does not work well in the case of dynamic natural environments since they include repetitive motions like swaying vegetation, rippling water, flickering monitors, camera jitter etc. This means that the scene background is not completely static. By using more than one Gaussian distribution per pixel it is possible to handle such backgrounds [2, 3, 13]. In [13], the pixel intensity was modelled by a mixture of weighted Gaussian distributions. The weights were related to the persistence of the distributions in the model. In [2], the mixture of Gaussians approach was used in a traffic monitoring application. The model for pixel intensity consisted of three Gaussian distributions corresponding to the road, the vehicle and the shadow distributions. Adaptation of the Gaussian mixture models can be achieved using an incremental version of the EM algorithm.

The Gaussian assumption for pixel intensity distribution does not always hold. To deal with the limitations of parametric methods, a non-parametric approach to background modelling was proposed in [1]. The proposed method utilises a general non-parametric kernel density estimation technique for building a statistical representation of the scene background. The probability density function for pixel intensity is estimated directly from the data without any assumptions about the underlying distributions.

In [5, 11], each image pixel was modelled with a Kalman filter. The proposed method can adapt to changes in illumination, but has problems with complex dynamic backgrounds. This approach was used in the automatic traffic monitoring application presented in [6]. In [17], the dynamic texture (background) is modelled by an ARMA model. A robust Kalman filter algorithm is used to iteratively estimate the intrinsic appearance of the dynamic texture as well as the regions of the foreground objects.

Hidden Markov Models (HMMs) have also been used to model pixel intensity. In this approach, the pixel intensity variations are represented as discrete states corresponding to scene modes. In [12], the approach was used in a traffic monitoring application where the pixel states represented the road, vehicles and shadows. In [14], HMM states corresponded to global intensity modes each modelled with a single Gaussian distribution. The model was capable of handling sudden illumination changes.

Modelling the scene background using edge features has also been under research. In [7], the background model was constructed from the first video frame of the sequence by dividing it into equally sized blocks and calculating an edge histogram for each block. The histograms were constructed using the pixel-specific edge directions as bin indices and incrementing the bins with the corresponding edge magnitudes. In [4], a fusion of edge and intensity information was used for background subtraction.

Motion based approaches have also been proposed for background subtraction. The algorithm presented in [15] detects salient motion by integrating frame-to-frame optical flow over time. Salient motion is assumed to be motion that tends to move in a consistent direction over time. The saliency measure used is directly related to the distance over which a point has traveled with a consistent direction.

Region-based algorithms usually divide an image into blocks and calculate block-specific features. Change detection is achieved via block matching. In [8], the block correlation is measured using the NVD (Normalised Vector Distance) measure. In [7], an edge histogram calculated over the block area is used as a feature vector describing the block. The region-based approaches allow only coarse detection of the moving objects unless a multi-scale approach is used.

3 New Approach

3.1 LBP Operator

The proposed texture-based method for background subtraction is based on the Local Binary Pattern (LBP) texture measure. The LBP is a powerful means of texture description. The operator labels the pixels of an image block by thresholding the neighbourhood of each pixel with the center value and considering the result as a binary number (LBP code):

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \tag{1}$$

where g_c corresponds to the grey value of the center pixel (x_c, y_c) and g_p to the grey values of the P neighbourhood pixels. The function $s(x)$ is defined as follows:

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} . \tag{2}$$

The original LBP operator worked with the 3×3-neighbourhood of a pixel [9]. See Figure 1 for an illustration of the operator. The general version of the LBP operator uses a circularly symmetric neighbour set as illustrated in Figure 2 [10]. In this case, g_p corresponds to the grey values of P equally spaced pixels on a circle of radius R.

The histogram of the LBP codes calculated over an image block can be used as a texture descriptor for the block. As can be seen from (1), the LBP is invariant to monotonic changes in grey scale.

3.2 The Method

The goal of the proposed method is to identify regions of a video frame that contain moving objects. The method divides each new video frame into equally sized blocks by using a partially overlapping grid structure as illustrated in Figure 3. By using partially overlapping blocks it is possible to extract the shape of the moving object more accurately than in the case of non-overlapping blocks.

Let us consider the feature vectors of a particular image block over time as a block process. Since we use an LBP histogram as the feature vector, the block process is defined as a time series of LBP histograms. We denote the block histogram at time instant t by $\mathbf{x}_t$.

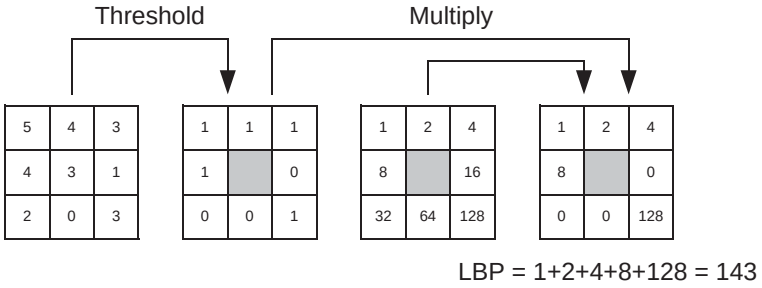

Figure 1: Example for calculating the original LBP code.

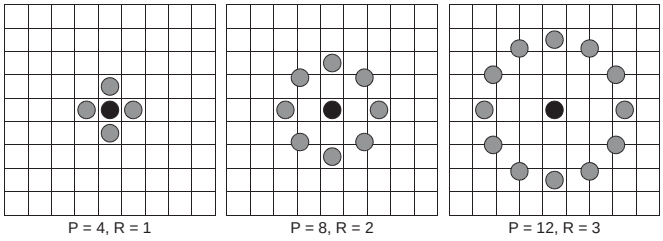

Figure 2: Circularly symmetric neighbour sets for different values of P and R. The values of neighbours that do not fall exactly on pixels are estimated by bilinear interpolation.

As mentioned in the previous section, LBP is invariant to monotonic changes in grey scale. This makes it robust against illumination changes. Since the LBP histogram does not contain information about the positions where the individual LBP codes were calculated, it also supports the modelling of multi-modal backgrounds. LBP is computationally very fast, which is an important property from the practical implementation point of view. Furthermore, we use more than one histogram to model each block.

The history of each block, $\{\mathbf{x}_1,...,\mathbf{x}_t\}$, is modelled by a group of K weighted LBP histograms. We denote the weight of the k^{th} histogram at time instant t by $\omega_{k,t}$. In the following, we explain the background model update procedure for one block, but the procedure is identical for each block. We utilise some ideas proposed by Stauffer and Grimson in [13].

The first thing to do is to compare the new block histogram $\mathbf{x}_t$ against the existing K model histograms using a distance measure. We used the histogram intersection as the distance measure in our experiments. The histogram intersection for the normalised histograms $\mathbf{x}_1$ and $\mathbf{x}_2$ is defined as follows:

$$H(\mathbf{x}_1,\mathbf{x}_2) = \sum_i min(x_{1,i},x_{2,i}), \tag{3}$$

where i is the bin index of the histogram. The user defines the threshold value T_D for the histogram intersection as a method parameter. Notice that it is also possible to use other measures such as Chi-square or Log-likelihood.

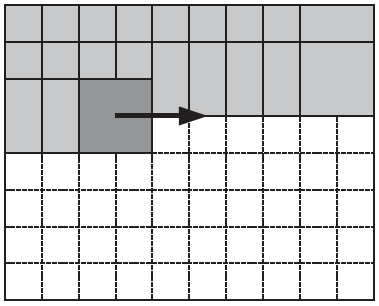

Figure 3: Partially overlapping grid structure used by the algorithm.

If none of the model histograms is close enough to the new histogram, the model histogram with the lowest weight is replaced with the new histogram and is given a low initial weight. After this, the weights are normalised so that they sum up to one.

If a model histogram close enough to the new histogram was found, the bins of this histogram are updated as follows:

$$\mathbf{x}_{k,t}[i] = \alpha_b \mathbf{x}_t[i] + (1-\alpha_b)\mathbf{x}_{k,t-1}[i],\ \alpha_b \in [0,\,1], \tag{4}$$

where α_b is the user-defined learning rate. In addition, the weights $\omega_{k,t}$ are updated as follows:

$$\omega_{k,t} = (1-\alpha_w)\omega_{k,t-1} + \alpha_w M_{k,t},\ \alpha_w \in [0,\,1], \tag{5}$$

where α_w is the user-defined learning rate and $M_{k,t}$ is 1 for the matched histogram and 0 for the others.

Next, we need to decide which of the histograms of the model are most likely produced by the background processes. We use the persistence of the histogram as an evidence for this. Because the persistence of the k^{th} histogram is directly related to its weight $\omega_{k,t}$, the histograms are sorted in decreasing order according to their weights. As a result the most probable background histograms are on the top of the list.

As a last phase of the updating procedure, the first B histograms are selected to be the background model as follows:

$$\omega_{1,t} + \ldots + \omega_{B,t} > T_B,\ T_B \in [0,\,1], \tag{6}$$

where T_B is the user-defined threshold for the selection. If we are modelling a unimodal background, a small value for T_B should be sufficient. In the case of a multi-modal background, a larger value for T_B is recommended. A small value for T_B selects only the most probable histogram to be the background model, whereas a large value allows the histogram $\mathbf{x}_t$ to get more bin configurations since the background model consists of more than one histogram.

Foreground detection is achieved via comparison of the new block histogram $\mathbf{x}_t$ against the existing B background histograms selected at the previous time instant. If a match is not found, the block is considered to belong to the foreground. Otherwise, the block is marked as background.

4 Experiments

The performance of the proposed method was evaluated using several video sequences. Both indoor and outdoor scenes were included. Figure 5 shows the results for the indoor test sequence where a person is walking in a room. As can be seen, the algorithm can separate the moving object almost perfectly. Because the algorithm is block-based, there occur some false positives on the contour areas of the object. Figure 6 shows the results for the indoor test sequence where a person walks towards the camera. Many adaptive pixel-based methods output a huge amount of false negatives on the inner areas of the moving object, because the pixel values stay almost the same over time. The proposed method gives good results, because it exploits information gathered over a larger area than a single pixel. The indoor test sequence illustrated in Figure 7 tests how well the algorithm handles a situation with a sudden illumination change. As can be seen the adaptation is pretty fast. The adaptation speed can be controlled with the learning rate parameters α_b and α_w. The bigger the learning rates are the faster the adaptation. Figure 8 shows the results of our algorithm for the outdoor test sequence, which contains relatively small moving objects. The original sequence has been taken from the PETS database (`ftp://pets.rdg.ac.uk/`). The proposed algorithm successfully handles this situation, but the shape information is not preserved well because of the block-based approach. The test sequence in Figure 9 contains heavily swaying trees in the wind and rippling water, which make it a very difficult scene from the background modelling point of view. There can also be some camera jitter present because of the wind. Since the method was designed to handle also multi-modal backgrounds, it manages the situation relatively well. The values of the method parameters used for the test sequences can be found from Table 1.

From the test results we can see that our algorithm preserves the shape of a moving object pretty well in the case of large objects. Because the approach is block-based, the detection of very small objects can be done only coarsely.

We measured the sensitivity of the algorithm to small changes of its parameter values. The most sensitive parameter is the distance threshold T_D which is used during the comparison of the histograms. The selected LBP operator mainly imposes the value of T_D. In summary, the algorithm proved not to be very sensitive to small changes in its parameter values, which makes the selection process quite easy.

The performance of the proposed algorithm was compared to four algorithms representing state-of-the-art pixel and block-based approaches [7, 8, 13, 5, 11]. We manually labelled five frames from all of the five test sequences and calculated the accumulated sums for the true negatives and the true positives reported by the algorithms for the test sequences. The results are illustrated in Figure 4. The overall performance of our approach seems to be better than the performance of the comparison methods. The new method detected moving objects very accurately (true positives). The proportion of the true negatives was also pretty good. As can be seen from Figures 5-9, most of the false positives occurred on the contour areas of the moving objects. This is satisfactory because our approach is block-based.

We also measured the real-time performance of our algorithm. For the parameter values used in the tests, a frame rate of 28 fps was achieved. We used a standard PC with a 1.8 GHz AMD Athlon XP CPU processor and 512 MB of memory in our experiments. The image resolution was 320×240 pixels.

Test Sequence	Block Size	$LBP_{P,R}$	K	α_b	α_w	T_D	T_B
1 (Figure 5)	12×12	$LBP_{6,1}$	4	0.010	0.010	0.35	0.5
2 (Figure 6)	12×12	$LBP_{6,1}$	4	0.010	0.010	0.35	0.5
3 (Figure 7)	12×12	$LBP_{6,1}$	4	0.050	0.020	0.35	0.8
4 (Figure 8)	12×12	$LBP_{6,2}$	3	0.005	0.005	0.50	0.9
5 (Figure 9)	12×12	$LBP_{6,2}$	5	0.015	0.015	0.50	0.9

Table 1: The parameter values used in the tests.

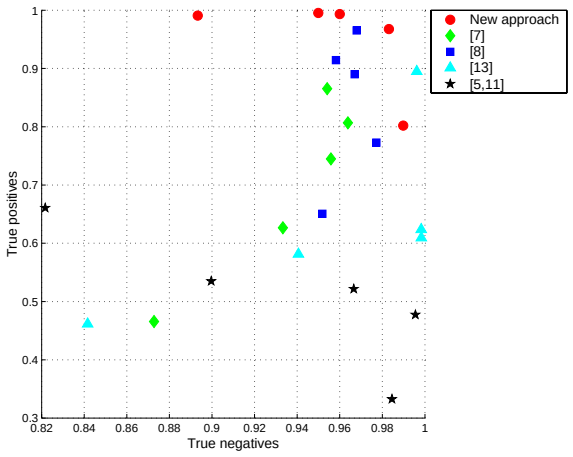

Figure 4: The proportions of the correctly classified pixels reported by the algorithms for the test sequences.

5 Discussion

In this paper, a novel block-based approach for detecting moving objects from video frames was proposed. Since the algorithm uses partially overlapping blocks, the shape information of the moving objects is preserved relatively well. The recent history of each block is modelled by a group of K weighted adaptive LBP histograms. The LBP is invariant to monotonic changes in grey scale, which makes it robust against illumination changes. The LBP also supports the modelling of multi-modal backgrounds. To make the model adaptive we update it at each discrete time instant.

The proposed algorithm was tested against several test sequences including both indoor and outdoor scenes. It proved to be tolerant to illumination changes and multimodality of the background. Since the algorithm is adaptive, it also manages situations where new stationary objects are introduced to or old ones removed from the scene background. Furthermore, the algorithm operates in real-time, which makes it well suited to systems that require real-time performance.

Acknowledgement

This work was supported by the Academy of Finland.

References

[1] 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*, 90:1151–1163, 2002.

[2] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *13th Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[3] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[4] 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 *15th International Conference on Pattern Recognition*, volume 4, pages 627–630, 2000.

[5] K. P. Karmann and A. Brandt. Moving object recognition using an adaptive background memory. In *Cappellini V. (ed.) Time-Varying Image Processing and Moving Object Recognition*, volume 2, 1990.

[6] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *12th IAPR International Conference on Pattern Recognition*, volume 1, pages 126–131, 1994.

[7] M. Mason and Z. Duric. Using histograms to detect and track objects in color video. In *30th Applied Imagery Pattern Recognition Workshop*, pages 154–159, 2001.

[8] T. Matsuyama, T. Ohya, and H. Habe. Background subtraction for non-stationary scenes. In *4th Asian Conference on Computer Vision*, pages 622–667, 2000.

[9] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[10] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[11] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using Kalman-filtering. In *International Conference on Recent Advances in Mechatronics*, pages 193–199, 1995.

[12] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. In *6th European Conference on Computer Vision*, volume 2, pages 336–350, 2000.

[13] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[14] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann. Topology free hidden markov models: Application to background modeling. In *8th IEEE International Conference on Computer Vision*, volume 1, pages 294–301, 2001.

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

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

[17] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. In *9th IEEE International Conference on Computer Vision*, volume 1, pages 44–50, 2003.

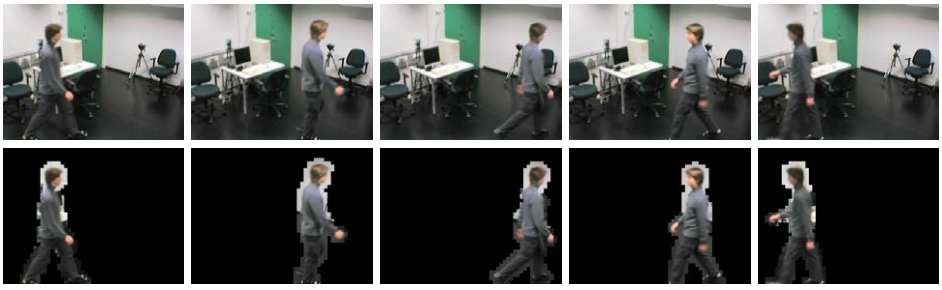

Figure 5: Frames 226, 240, 246, 328 and 344 from an indoor test sequence where a person is walking in a room.

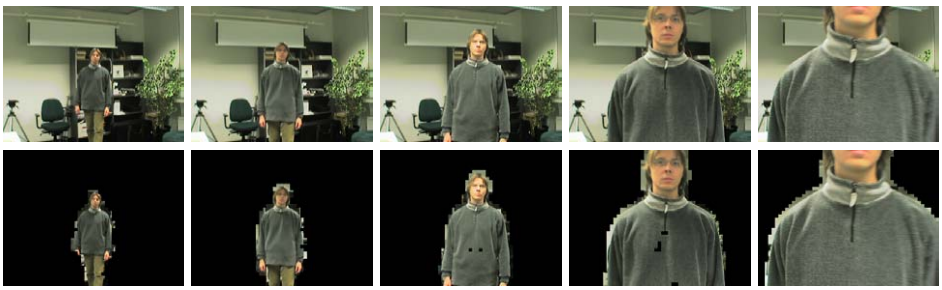

Figure 6: Frames 300, 340, 380, 420 and 440 from an indoor test sequence where a person is walking towards the camera.

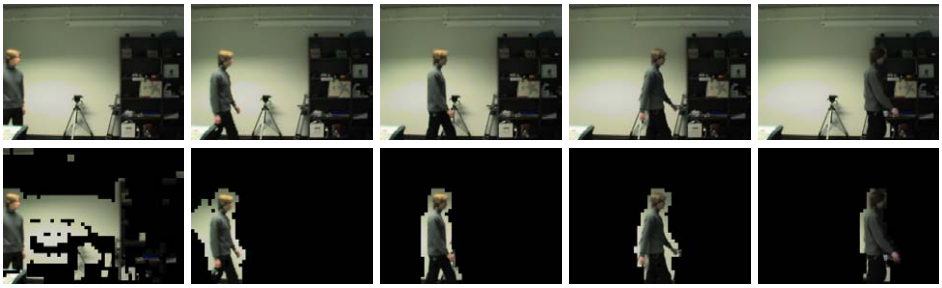

Figure 7: Frames 330, 340, 350, 360 and 370 from an indoor test sequence where the lights are just turned on.

Figure 8: Frames 150, 590, 920, 1000 and 1340 from an outdoor test sequence taken from the PETS database.

Figure 9: Frames 280, 300, 320, 350 and 380 from an outdoor test sequence where two persons are walking in front of heavily swaying threes and rippling water. There can also be some camera jitter present because of the wind.