Texture-Based Foreground Detection

Csaba Kertész
Vincit Oy
Hermiankatu 3 A, 33720 Tampere, Finland
E-mail: csaba.kertesz@ieee.org

Abstract

The foreground detection can be utilized in tracking, segmentation or object recognition. The Local Binary Pattern (LBP) texture descriptor has been introduced for various purposes (e.g. texture classification, image indexing) and a foreground detection use is presented in this paper. Experiments on image sequences prove that the proposed algorithm, compared with other existing state of the art methods, achieves notable performance and computation speed.

Keywords: *Foreground, background, Local Binary Pattern, Markov Random Field*

1. Introduction

The foreground detection is one of the well-known problems of the image processing used in intruder detection, surveillance or object tracking. To have a closer look on the motivation of this paper, the literatures of the LBP and the Mixture of Gaussians (MoG) are discussed in the next paragraphs pointing out some similarities between these two paradigms and the further development of the LBP for foreground detection based on the observations.

The LBP texture descriptor was developed for texture classification [9], however, new aspects has been discovered: recognizing facial expressions [15], following human activities [8] or classifying genders [3]. Heikkilä et al. developed a foreground detection method, which was based on the LBP [4] and it used a block-matching measurement in order to calculate the similarity to the background model. The next version of algorithm (TBMOD), extended by Heikkilä and Pietikäinen [5], worked on pixel basis and each pixel had independent statistics. The advantage of this pixel basis approach is providing pixel-level accuracy and detecting more subtle details of the foreground. On the other hand, the computational complexity was increased and the higher level classification of the pixels was not considered.

Many popular methods from the last decade base on the Gaussian distribution assumption of the individual pixels. After the first version of this algorithm, more Gaussian distributions have been used [13] to handle multi-modal backgrounds with rippling water or flickering monitors. More authors have proposed improvements to the MoG like transforming the color information to detect shadows using the mixture model [7] or adapting the components of the mixture for each pixel [18].

A new variant of the Mixture of Gaussian algorithm proposed by White and Shah [17] uses a particle swarm optimization to tune the parameter set of the MoG to the ground-truth images. A fitness function automates the hand-tuning of the parameters and reduces the sum of the false detections of the MoG efficiently.

The MoG and the LBP methodologies with joint machine learning methods have been advanced the original algorithms in the recent years. Schindler and Wang optimized the

output of the MoG with a Markov Random Field in their smooth foreground-background segmentation (SFBS) [11] and the results were impressive. Dalley et al. proposed a generalization to the MoG model [2] where the Gaussian models of a pixel are affected by the pixels in the neighborhood and the MRF classifies the pixels as foreground or background finally. Kellokumpu et al. used a Hidden Markov Model (HMM) with LBP feature histograms to detect and classify human activities [8].

The TBMOD adapted some pieces of the MoG model to update the LBP histograms of the pixels. This similarity brings the straightforward direction to the machine learning algorithms, which have been used for the MoG models successfully, in order to optimize the foreground estimation based on the LBP histograms.

This paper proposes the MRF as a higher level classification of the LBP histograms into foreground and background. The large changes in the scene (e.g. turning off the lights) raise difficulties for the quick adaptation of the models, therefore, an initial stage has been developed here to reset the histogram models in such situations. The next sections describe the proposals and the experimental results.

2. Proposed Method

The model update of the TBMOD was taken over with modifications that boost the early phase of the model building and optimize the computation time. Figure 1 shows how the images are acquired from the camera and the models cleared when large changes happen in the scene. The next stage of the algorithm incorporates the image into the model and starts over with a new image, if only model updates are necessary; otherwise a graph is built upon the initial guesses of the LBP histograms and cut to find an optimal approximation of the foreground regions.

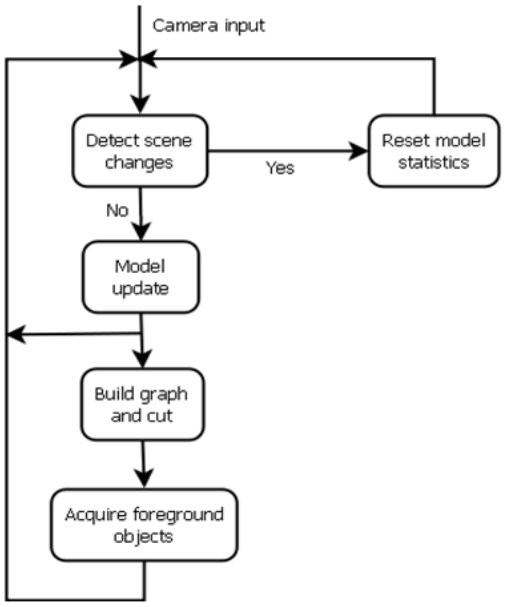

Figure 1. The Steps of the Algorithm

2.1. Detect global scene changes

The sudden changes in the illumination, either global (e.g. sun going behind the clouds) or local changes (e.g. partial reflection from a bright object nearby), are challenging for modeling the background because of the quick temporal change of the pixel values. Some authors proposed modifications to the existing algorithms [6, 14], however, a separate function is implemented here.

To recognize the illumination changes in the scene, the camera image is converted to Luv (or Lab) color space and the L channel is extracted and normalized in the range [0, 255]. The difference between the last and the previous frames in the image sequence is defined as:

$$L_{diff}(t-1,t)=\frac{\sum_{u=0}^{w-1}\sum_{v=0}^{h-1} g(u,v)}{w*h}*100, \tag{1}$$

where w and h correspond to the resolution of the extracted L channel and the $g(u, v)$ is defined as follows:

$$g(u,v)=\begin{cases}1 & : \ |(l_{t-1}(u,v)-l_t(u,v))|>b \\ 0 & : \ otherwise\end{cases}, \tag{2}$$

where the $l_t(u, v)$ denotes the luminance value in the (u, v) position in a certain t time and the b denotes a threshold value in bin units.

A scene change is global if L_{diff} exceeds a high percentage of the scene area ($L_{diff} >= S_{limit}$). The Figure 2 shows five examples where the intersection of the areas below the curves on the figure corresponds a good value space for (b; S_{limit}). Higher values of the L_{diff} are preferred and smaller bins can be caused by the image noise, therefore good sets of values are for example (15; 75), (10; 80) or (20; 70). Any of these options can be used for the Wallflower and the TBMOD videos successfully.

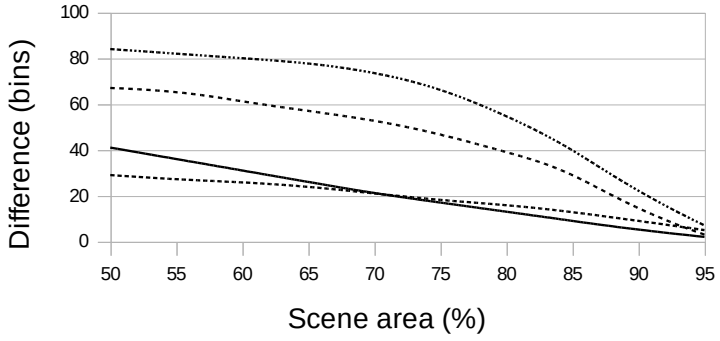

Figure 1. Five examples of illumination changes/sudden reflections from a huge object in various indoor scenes and recorded by multiple cameras. The figure shows the calculated b in the function of L_{diff} values between two sequential frames before and after an illumination change.

The illumination changes are transient during multiple frames in the records, hence it is a good practice to ignore some frames (e.g. 5 frames in this study) after a change is detected. The L_{diff} measure is quite invariant for the image scaling. The L channel scaled to 1/8 of the original size modifies the results of the L_{diff} measure less than 1 % and the computation time becomes insignificant at such low resolutions.

2.2. New LBP Operator

The Local Binary Pattern (LBP) is a grayscale invariant texture primitive statistic, which was introduced in the middle of the nineties. The first version of the operator worked with eight-neighbors of a pixel, using the value of the center pixel as a threshold. The thresholded neighborhood was multiplied with powers of 2 and eventually summing up [10].

Wide variety of the LBP exists in the literature and the version in this paper uses a larger neighborhood than the original. The following formula defines the new operator:

$$\begin{aligned} LBP_{new}(x,y) &= n_1(x-2,y-2)*2^0 + n_2(x-2,y)*2^1 + \\ & n_1(x+1,y-2)*2^2 + n_3(x-2,y)*2^3 + n_3(x+1,y)*2^4 + \\ & n_1(x-2,y+1)*2^5 + n_2(x,y+1)*2^6 + n_1(x+1,y+1)*2^7, \\ n_1(i,j) &= s\left(\frac{\sum_{u=i}^{i+1}\sum_{v=j}^{j+1} g(u,v)}{4} - C\right), \\ n_2(i,j) &= s\left(\frac{g(i,j)+g(i,j+1)}{2} - C\right), \\ n_3(i,j) &= s\left(\frac{g(i,j)+g(i+1,j)}{2} - C\right), \\ s(r) &= \begin{cases} 1 & : r \geq 0 \\ 0 & : r < 0 \end{cases}, \end{aligned} \tag{3}$$

where g denotes the grayscale value in a certain position and C denotes the average value of the 4-connected neighborhood of the center pixel (x, y).

The new operator is calculated inside a 5x5 neighborhood, therefore, the LBP_{new} is not defined two-pixels wide on the image borders. The TBMOD specified a correction value to the LBP calculation in order to handle the flat color areas where the color values almost do not change. This correction has negligible influence with the LBP_{new}, hence omitted here.

2.3. Model Update on Pixel Basis

First, the camera image is converted to CIE Luv color space and the L, u and v channels are normalized to the range [0, 255]. The LBP_{new} works on a single color channel, therefore, the image is converted to grayscale and eventually the LBP_{new} operator is applied. In the case of the large scene changes (Section 2.2), the LBP_{new} histograms (described below) are cleared with the data generated from the current image and everything else is set to default values in the model. (No further processing is required in this case.)

The model of each pixel and its corresponding update process are identical. Let us denote the actual pixel, whose model is being processed, as $p(x, y)$ and the current frame number:

$$f = \begin{cases} 1 & : \ first\ frame \vee clearing\ the\ model \\ f+1 & : \ otherwise \end{cases}. \tag{4}$$

An LBP_{new} histogram (h) is computed from the disk of a circle with the radius R and the center p. The model of the pixel contains K pieces of histograms $\{m_0, ..., m_{K-1}\}$, each model histogram has a weight $\{\omega_0, ..., \omega_{k-1}\}$, with a value range [0, 1] and their sum up to one. h is compared to the current K model histograms using a proximity measure, the histogram intersection, which represents the common part of two histograms:

$$I(a,b) = \frac{\sum_{i=0}^{N-1} min(a_i, b_i)}{N}, \tag{5}$$

where a and b are two histograms and N is the number of the histogram bins. It neglects features occurred only in one of the histograms and the complexity is very low, linear for the number of the histogram bins: O*(N)*.

There is a threshold value for the proximity measure, T_p. If the proximity is below the threshold T_p for all model histograms, the model histogram with the lowest weight is replaced by h. A low initial weight (e.g. 0.01) is given to the new histogram and the weights of the model are normalized to sum up to one. The new model histogram is marked as foreground histogram and no further processing is required in this case.

More processing is required if matches were found. The model histogram with the highest proximity value is selected and the best matching model histogram is updated as follows:

$$\begin{aligned} m_k &= (\alpha_h + D) * h + (1 - \alpha_h - D) * m_k, \\ D &= \begin{cases} (100 - f)/100 & : f < 100 \\ 0 & : otherwise \end{cases}, \end{aligned} \tag{6}$$

where $\alpha_h \in [0,1]$ is a user-settable learning rate, D provides fast adaption in the begin of the model building (e.g. after fast illumination changes). Furthermore, the weights of the model histograms are updated:

$$\omega_k = (\alpha_w + D) * M + (1 - \alpha_w - D) * \omega_k, \tag{7}$$

where α_w is an other user-settable learning rate and M is 1 for the best matching histogram and 0 for the others. The adaptation speed of the model is controlled by the learning rate parameters α_h and α_w.

All of the model histograms are not necessarily produced by the background silhouette. The persistence of the histograms in the model can be used to decide whether the histogram models the background or not. The persistence is directly related to the weight of the histogram: the bigger the weight, the higher the probability of being a background histogram. As a last stage of the updating procedure, the model histograms are sorted in decreasing order according to their weights and the first L histograms are marked as background histograms ($B_t = B_{t,0}...B_{t,L-1}$):

$$\omega_0 + ... + \omega_{L-1} > T_b, \quad T_b \in [0,1], \tag{8}$$

where T_b is a user-settable threshold.

The probability of being a background pixel is defined by the maximal proximity between h and the background histograms:

$$P(p \in \beta) = max(I(h, B_{t,i})), \quad (i = 0 \ldots L-1), \tag{9}$$

where β is the set of the background pixels. This probability is calculated when the proximity of the current h histogram is compared against the model histograms of the pixel. $P(p \in \beta)$ is used in the MRF model in the next section.

A "chess pattern" can optimize the computation time of the update process. Instead of all pixels, every even pixel in the even rows and every odd pixel in the odd columns are calculated. The rest pixels are labeled as foreground if at least two pixels in their 4-connected neighborhood are foreground pixels, otherwise it is background pixel. The computation time is reduced by half approximately and the accuracy is still kept near pixel-level.

2.4. Markov Random Field

Updating the LBP_{new} histograms is similar to the MoG model, which was optimized with an MRF in the SFBS [11]. That MRF contained background and foreground labels and its global minimum was equivalent with the result of the min-cut of a graph [1].

A modified version of the graph is built here. A node is associated to each pixel (p) on the image, one node to the "foreground" and one node to the "background". The nodes of the pixels are connected with an arc to each node in their 4-connected neighborhood, with an arc to the foreground node and with an arc to the background node. The weights of the arcs in the graph as follows:

$$\begin{aligned} &C = 1, \\ &W(p \in \beta) = 1, \\ &W(p \in F) = \gamma * (1 - P(p \in \beta)), \end{aligned} \tag{10}$$

where the C is the cost between the members of the 4-connected neighborhoods, $W(p \in F)$ is the cost of the arcs connected to the foreground node and the $W(p \in \beta)$ is the cost of the arcs connected to the background node. The next section advices some good values for the constant γ.

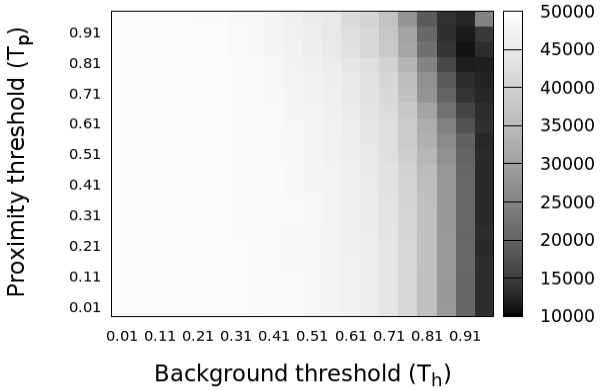

Figure 3. The sum of the false positive and the false negative pixels, while the background and the proximity thresholds are varied. The darker color means less false pixels.

A graph can be built and cut for each frame after updating the LBP_{new} histograms, but it is optional. A pixel belongs to the foreground if its graph node is connected to the foreground node after the cut, otherwise it is a background pixel. Because of the LBP_{new} works on the neighborhood of the pixels, an erode filter is applied to the result of the min-cut algorithm.

3. Parameter Selection

The algorithm contains several variable parameters and their value spaces are large. First, the TBMOD [5] test videos have been used to research the value spaces and the results have been verified on the Wallflower test videos [16] separately. It has been found that the changes in the parameter values cause the same (increasing/decreasing) trends of the false and true pixels on the two different test video sets, suggesting the expectable good results of an optimal parameter set on 3rd party videos as well.

A good set of reference values is $R = 2$, $K = 3$, $T_b = 0.95$, $T_p = 0.75$, $\alpha_h = 0.01$, $\alpha_w = 0.01$ and $\gamma = 8$. These values provide good results, but to have a better insight, how sensitive the algorithm for the parameter values, measurements were made with five TBMOD videos with one varied parameter at a time.

The Figure 3 shows the sum of the false pixels, while the T_p and T_h are varied in the range [0, 1]. The deep dark area [0.8, 0.95] in the diagram corresponds to the range of the good values for both parameters.

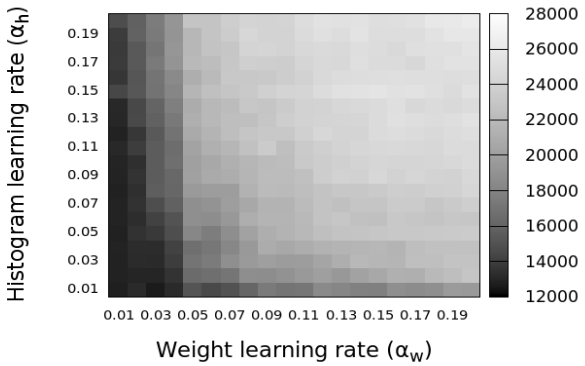

Figure 4. The sum of the false positive and the false negative pixels, while the histogram and the weight learning rates are varied. The darker color means less false pixels.

Parameters α_w and α_h are responsible for the balanced persistence and the update of the histograms. High rates do not serve the temporal learning well, therefore, the Figure 4 shows these rates in the range [0, 0.21]. The range [0, 0.03] contains good values with minimal false pixels for both parameters.

Small sampling regions are not suitable for the dynamic backgrounds (e.g. waving trees) and large sizes can not recognize the small objects. A good choice is $R = 2$ according to the Figure 5. The number of histograms assigned to a pixel have impact on the speed, hence $K = 2, 3, 4, 5$ are good candidates and the lower values can be preferred for less computation time and smaller memory footprint. γ is a constant for the graph of the MRF and good values are in the range [7, 10].

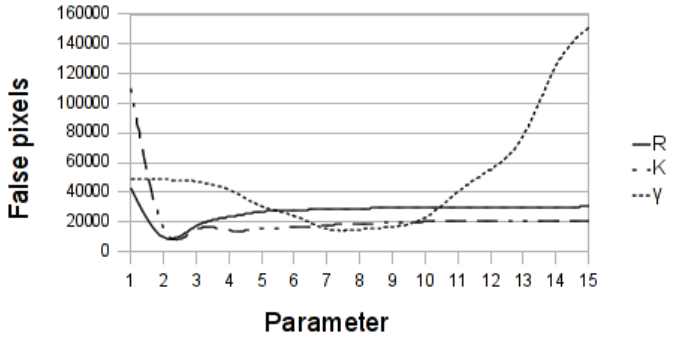

Figure 2. The sum of the false positive and the false negative pixels, while R, K and γ are varied.

4. Experiments

In the contrast to the parameter selection, based on the TBMOD test videos in the previous section, some experimental results with the Wallflower test videos [16] are discussed here, using the same reference parameters. To emphasize an advantage of the new algorithm (LBP-MRF), one set of fixed parameters for all videos are used, while the TBMOD authors specified separate set of optimal parameters on video basis [5] in their tests.

The definition of the foreground is not straightforward because the changes in the scene can be originated from either the locomotion or the self-movements of the background components (e.g. waving trees or rippling water). Several challenges are well-demonstrated in the Wallflower's videos (in the order of the Figure 6): *Time of Day* (TOD) and *Light Switch* (LS) – Slow/fast lightning changes, *Waving Trees* (WT) and *Camouflage* (C) – Dynamic background and flat color areas, *Boostrapping* (B) – Small moving objects with shadows, *Foreground Aperture* (FA) – Flat color areas. The video, named Moved Object, is left out on purpose as the algorithms have only a few false detections with it.

Since the new algorithm is similar to the MoG, the comparison is done against MoG variants: the SFBS [11], a MoG with a swarm optimization (MoG-PSO2) [17], a MoG based on IHLS color space [12]; the results of the new algorithm are in the last row of the Figure 6. Comparing the ground truths (second row) and the output of the algorithms, the foreground objects were detected by the new algorithm (LBP-MRF) as closed regions in all cases. It identified the feet of the sitting person in TOD, but the standing person on the right side in B was missed, similar to the other algorithms. In some cases, the SFBS approximates the shapes of the foreground objects better than LBP-MRF, in other cases, the latter. All in all, the new algorithm performs well to detect the foreground as continuous regions contrary to others (e.g. MoG-PSO2). These are the visual impressions, but take a look on the performance in numbers.

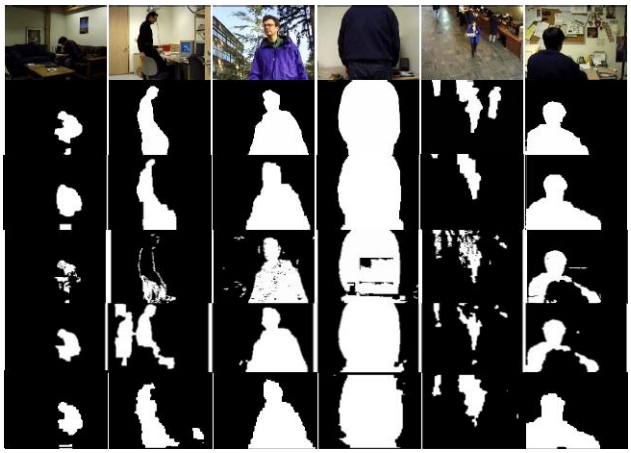

Figure 6. Experiments with the Wallflower image sequences. The first row contains the original test images and the second the ground truths. From the third row, the figure shows the results of the following algorithms: 3th SFBS, 4th MoG-PSO2, 5th MoG-IHLS and 6th LBP-MRF.

The Figure 7 shows the sum of *false positives* (background pixels marked as foreground) and *false negative* (foreground pixels marked as background) of the examined algorithms. The LBP-MRF has a promising result, near half of the false alarms of the MoG-PSO2 and MoG-IHLS; the result of the SFBS is approximated, which is the best.

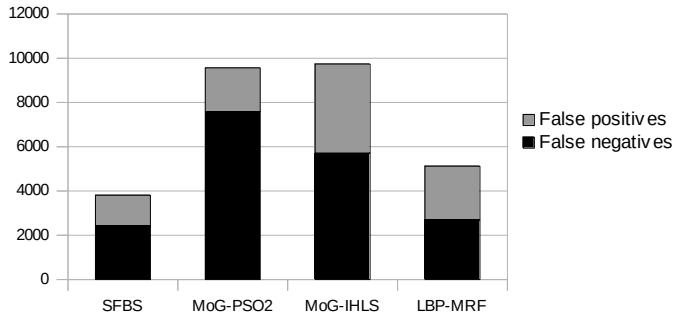

Figure 7. The test results for the Wallflower image sequences where the unit of the vertical axis is the sum of the false positive and false negative pixels.

When the TBMOD videos with the original size (320x240) were processed by the LBP-MRF, using the same reference parameters, it was discovered that the algorithm is quite invariant for the resolution and provides the good performance at higher resolutions, despite the smaller sampling area relative to the resolution.

The computation time is an important part of the capabilities of the algorithm. A test configuration was a laptop with a single-core Pentium 4 (1.8 Ghz) and the LBP-MRF is more than three times faster compared to the original TBMOD at the resolution 160x120, nonetheless, extra steps are introduced in the new version; it requires 16 ms to process the model update of a frame and 6 ms for the MRF if the foreground regions are estimated based on the model. The computation times are decreased by half on a Core2

Duo CPU (utilizing one core at 1.6 Ghz) and thus the new algorithm is suitable for real-time processing. Comparing the LBP-MRF to SFBS algorithm, the latter needs an average 14 ms for the MRF over the MoG model updates (~50-60 ms/frame).

5. Conclusion

An enhanced, open-source[1] version of the TBMOD has been proposed in this paper and the new algorithm (with LBP_{new} operator and MRF) estimated the foreground efficiently with low computational cost. The detection of the changes helped to rebuild the model in the cases of the temporal fluctuations of the illumination and thus the new algorithm performs notable and its accuracy can score marks against the TBMOD or MoG variants.

The TBMOD authors proposed variant value sets of the parameters for each test video in their study, however, the same set of values performed well for all of the TBMOD and the Wallflower videos here and the fine-tuning is possible in a range of recommended values for specific image sequences if necessary.

The future work can include the development of new changes to reduce the influence of the image noise, because cheap webcams are used in computer vision projects and the image quality is limited.

Following the camera movements can be an other direction for improvements. The fixed camera was an assumption in this paper and a tilting or panning camera can not be handled with the current version of the algorithm. Nevertheless, the model is pixel based and if the vector of camera movement is known, it is possible to move the statistics of the pixels according to the vector.

Acknowledgment

Thanks to Marko Heikkilä, providing some TBMOD source codes for analysis and videos for testing purposes.

References

[1] Y. Boykov, V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision", Proceedings 3rd EMMCVPR, 2001.

[2] G. Dalley, J. Migdal, and W.E.L. Grimson, "Background Subtraction for Temporally Irregular Dynamic Textures", Workshop on Applications of Computer Vision, 2008.

[3] A. Hadid, M. Pietikäinen, "Manifold learning for gender classification from face sequences", Proceedings 3rd IAPR/IEEE International Conference on Biometrics (ICB), Alghero, Italy, 2009.

[4] M. Heikkilä, M. Pietikäinen, and J. Heikkilä, "A texture-based method for detecting moving objects", Proceedings the 15th British Machine Vision Conference (BMVC), London, UK, 1:187-196, 2004.

[5] M. Heikkilä, M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4):657-662, 2006.

[6] V. Jain, B. Kimia, and J. Mundy, "Background Modeling Based on Subpixel Edges", IEEE International Conference on Image Processing (ICIP), pp. 321-324, 2007.

[7] P. KaewTraKulPong, R. Bowden, "An Improved Adaptive Background Mixture Model for Real-Time Tracking with Shadow Detection", Proceedings European Workshop on Advanced Video Based Surveillance Systems, 2001.

[8] V. Kellokumpu, G. Zhao, and M. Pietikäinen, "Human activity recognition using a dynamic texture based method", Proceedings The British Machine Vision Conference (BMVC), Leeds, UK, 2008.

1 The project web address: http://aiboplus.sf.net

[9] T. Mäenpää, M. Pietikäinen, and T. Ojala, "Texture classification by multi-predicate local binary pattern operators", Proceedings 15th International Conference on Pattern Recognition, Barcelona, Spain, 3:951-954, 2000.

[10] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distribution", Pattern Recognition, 29(1):51-59, 1996.

[11] K. Schindler, H. Wang, "Smooth Foreground Background Segmentation for Video Processing", Asian Conference on Computer Vision (ACCV), Hyderabad, India, Volume 3852, pages 581-590, 2006.

[12] N. Setiawan, S. Hong, J. Kim, and C. Lee, "Gaussian mixture model in improved IHLS color space for human silhouette extraction", 16th Int. Conf. on Artificial Reality and Telexistence (ICAT), Hangzhou, China, 732- 741, 2006.

[13] C. Stauffer, W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking", Proceedings IEEE Computer Society Conference Computer Vision and Pattern Recognition, Vol. 2, pp. 246–252. 1999.

[14] T. Su, J. Hu, "Background removal in vision servo system using gaussian mixture model framework". ICNSC, 1:70-75, 2004.

[15] M. Taini, G. Zhao, and M. Pietikäinen, "Weight-based facial expression recognition from near-infrared video sequences", Proceedings 16th Scandinavian Conference on Image Analysis (SCIA), Oslo, Norway, 2009.

[16] K. Toyama, K. Krumm, J. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance", The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.

[17] B. White, M. Shah. "Automatically tuning background subtraction parameters using Particle Swarm Optimization", IEEE Conference on Multimedia and Expo, 2007.

[18] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction". Proceedings Int'l Conf. Pattern Recognition, vol. 2, pp. 28–31, 2004.

Author

Csaba Kertész received the BSc degree in Computer Science focused on Artificial Intelligence from Budapest Tech and MSc in Computer Science from University of Szeged, Hungary. He is a Lead Engineer at the Vincit Oy in Tampere, Finland. His research interests include image processing in robotics, behavior-based systems and specialization in the Sony AIBO robot dog.