Handling of False Stationary Detections in Background Subtraction in Video Preprocessing

Jiri Pytela[1], *Matej Smid*[1]

[1]Dept. of Cybernetics, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic

pyteljir@fel.cvut.cz, smidm@fel.cvut.cz

Abstract. *This paper describes a method for identifying and avoiding false stationary detections in background subtraction caused by movement of objects belonging to the background. This method is independent on used background subtraction algorithm. The proposed algorithm examines edges on contours of detected foreground objects to decide if it is a true detection or a false alarm. We propose and test 4 specific criteria functions that can be used to evaluate detected objects, we use the function with best results to test the effect of this method on performance of a tracking system. Both indoor and outdoor video sequences are used for evaluation.*

Keywords

video preprocessing, background subtraction, false stationary detection, ghost detection

1. Introduction

In many surveillance applications background subtraction is an important first step. It is used to separate the possible interesting moving objects - *the foreground*, from the uninteresting part of the input image - *the background*.

The simplest method to achieve this goal is to compare every frame of the input video sequence with a static background image. However, having a single constant image to represent the background proves insufficient in most real-world applications. Sometimes the background image is not available at all. The background can contain unimportant moving objects like water and trees, the lighting conditions may vary or the scene can be permanently changed when objects are added or removed. To handle these issues, it is necessary to maintain a background model which can be updated with new information.

There are many different techniques that were proposed to model the background [1]. One of the popular methods models the value of each pixel by a mixture of Gaussians [6].

We focus on background subtraction as a preprocessing step for object tracking. It is common that an object stays relatively still for longer periods of time - long enough for the background model to learn it as a background. A resumed movement results in a false detection on the previous location of the motionless object. The detection does not correspond to any real object. We call this type of false detection a ghost. The same situation occurs when there are moving objects present at the time of the background initialization. These issues are caused by the background modelling and occur in many background subtraction algorithms. In this paper we propose a method to identify and avoid these false detections.

The reminder of this paper is structured as follows. Overview of relevant previous work is given in Section 2. Section 3 describes the method itself. Details of each step, namely background modelling, ghost detection and background model update are presented in Sections 3.1, 3.2 and 3.3 respectively. Description of the evaluation process and data specifications are in Section 4. Results of all experiments and comparison of proposed criteria functions are presented in Section 5. We summarize our work and propose possible future enhancements in Section 6

2. Related Work

Dealing with the problem of ghost detections is common for many background subtraction algorithms. General overview of the topic is given in [7].

A popular solution, which can be convenient for many applications, assumes that every detection which does not move during multiple frames is a ghost [4][5][11][8], we explain why it is not suitable for our purpose.

A method presented in [10] utilize temporal image analysis and uses currently invisible background pixels in the background model update. In another work [9] the ghost detection elimination is achieved by examining a pixel neighbourhood during the evaluation process and is built-in in the background subtraction process.

In [3], the authors combine gradient of pixel intensity and pixel colour. Gradient difference is used in [2] to validate object detections. We take similar but simpler approach.

3. Proposed Approach

First, it is important to define what is considered a ghost. In some of the works dealing with ghost detections, any detection that does not move in a defined period of time is labelled as a ghost. But it is common in tracking applications that some objects do not move significantly for a long time, it is undesirable to label all stationary detections as ghosts.

This approach is based on the same idea as Javed et al. [2]. In an image, every real object is separated from its surroundings by an edge. If an object does not have an edge on its border, it is not a real object but a ghost detection. In this sense every stationary detection that do not correspond to any real object in the image is considered a ghost.

The entire process can be split into three parts: background subtraction, ghost detection and background model update. The background subtraction can be any method that works with pixel by pixel model of the background. In our case the pixels are modelled by a mixture of Gaussians [6]. The output is a foreground binary mask.

The ghost detection step splits the foreground mask into single connected components, each should correspond to one object. Objects contours are evaluated for a presence of image edges and labelled as valid detections or ghosts.

The background model update step takes all the pixels in all the connected components classified as a ghost detections and updates the corresponding pixels in the background model to the current value of those pixels. This way, ghost detections are quickly absorbed into the background.

The details of each step follow.

3.1. Background Subtraction

The proposed method can be used with any background subtraction algorithm that maintains an independent model for each pixel. Let us denote I an image being processed, x being the image pixel and $I(x)$ the pixel value. Here, we use Gaussian mixture model. The detailed description can be found in [6], we present a short overview below.

Let us denote the background model BG, then $BG(x)$ is the model of pixel x, three Gaussians $\mathcal{G}_i^x$, $i = 1, 2, 3$, are used to model its value. μ_i^x is the mean of $\mathcal{G}_i^x$, $\Sigma_i^x = \sigma_i^{x2}\mathbf{1}$ its covariance and w_i^x its weight in the model.

To derive the background pixel value the Gaussians are first ordered by the fitness value w_i^x/σ_i^x and then the first N^x are used to compute the expected pixel value.

$$N^x = \arg\min_n (\sum_{i=1}^{n} w_i^x > T), \tag{1}$$

where the threshold T is the minimum prior probability of the background in the scene.

We evaluate the pixels to create a foreground mask F, where $F(x)$ is its value at pixel x. $F(x) = 1$ classifies the pixel x as a foreground, $F(x) = 0$ as a background.

A pixel x is considered as a background if there is a Gaussian in the model which matches the pixel value. We say that a Gaussian matches pixel value if its intensity is closer than 2.5 standard deviations from the Gaussian mean:

$$F(x) = \begin{cases} 1 & \text{if } \exists i < N^x : |I(x) - \mu_i^x| < 2.5 \cdot \sigma_i^x \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The matched Gaussian parameters are updated as follows:

$$w_i^x(t+1) = w_i^x(t) \cdot (1-\alpha) + \alpha, \tag{3}$$
$$\mu_i^x(t+1) = \mu_i^x(t) \cdot (1-\alpha) + \alpha I(x), \tag{4}$$
$$\Sigma_i^x(t+1) = \Sigma_i^x(t) \cdot (1-\alpha) + \alpha(I(x) - \mu_i^x(t))^2, \tag{5}$$

where $\alpha \in \langle 0, 1 \rangle$ is a learning coefficient which determines how fast the background model reacts to a change. If there is no matching Gaussian in the model, the Gaussian with the lowest fitness is replaced. The $I(x)$ is used as the new mean with a high value for variance and a low weight.

At $t = 0$, when the first frame is processed, the situation is same as if the learning coefficient was selected to $\alpha = 1$. The second frame at $t = 1$ provides 50% of the available information to the background model, however according to the rules above has a smaller effect, depending on the value of the learning coefficient α.

To deal with this problem we propose to use a modified learning coefficient α', with value $\alpha' = 1$ at $t = 0$. α' is gradually lowered until the desired value α is reached:

$$\alpha' = \max(\alpha, \frac{1}{1+t}) \tag{6}$$

This approach better reflects information value of the new frames. The difference between α and α' is illustrated in Figure 1.

For the purposes of background subtraction we decided to use CIE LUV colour space, because colour difference in the model better corresponds to the human perceptual difference [14].

This background model can handle both sudden and gradual changes in illumination, moving objects in the background and also permanent changes in the scene.

3.2. Ghost Detection

This step takes a foreground mask F produced in the background subtraction step and evaluates validity of the detections. First the connected components are extracted from the foreground mask, each is then evaluated independently.

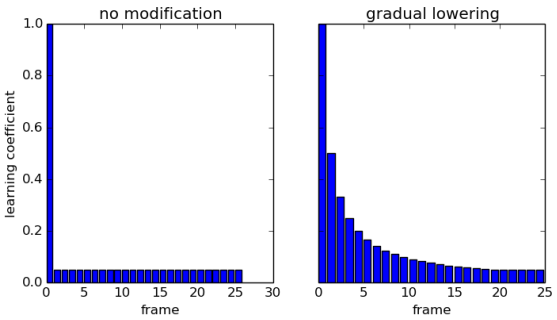

Fig. 1. At the beginning of an image sequence the background model has to be initialized. When a background image without moving objects is not available, the model has to be learned from the sequence itself. On the left subfigure the learning coefficient $\alpha = 1$ for the first frame. The background model is completely initialized with the fist frame. On the right subfigure the learning coefficient is smoothly decreased from 1 to the predefined value.

$$F = C_1 \vee C_2 \vee \cdots \vee C_n \tag{7}$$

where $C_{i \in 1..n}$ is a single connected component. Only C_i formed by larger number of pixels than a size threshold T_S continues to be evaluated.

The border of C_i is denoted C_i'. Because the region C_i' will be used for edge detection, it needs to be wider than one pixel, so it is expanded as illustrated in Figure 2. Canny edge detector with Otsu threshold is used to compute the edges. By using the Otsu algorithm we avoid selecting the threshold value manually.

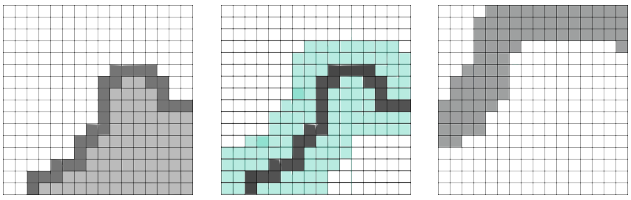

Fig. 2. On the left figure is an object C_i: it's border (dark grey) and interior (light grey). In the centre is the expanded object border C_i' (both light blue and dark grey), true border C_i^{tb} (dark grey) and its expansion C_i^{eb} (light blue). On the right figure the outer object border C_i^{ob} is displayed. It is the region adjacent to the expanded object border.

Edge mask obtained from currently processed image I in the region C_i' is denoted $E^I_{C_i'}(x)$. Image I contains the moving objects as well as the background so it follows that edges E^I from any region contain edges of the moving objects and also background edges. We are only interested in the edges belonging to the moving objects, which we can get by subtracting the background edges. The background image B is reconstructed from the background model, described in previous section, by:

$$\forall x \in B : B(x) = \mu_{k_m}, \quad k_m = \arg\max_k (w_k/\sigma_k). \tag{8}$$

Note that k_m is an index of a Gaussian with the highest fitness value w_k/σ_k.

Edge mask computed using the background image B in the region C_i' is denoted $E^B_{C_i'}$, similar to the case of I. By combining the image edges $E^I_{C_i'}$ with the background edges $E^B_{C_i'}$ we get the foreground edges:

$$E^F_{C_i'} = E^I_{C_i'} \wedge \neg E^B_{C_i'}. \tag{9}$$

Edges belong to the foreground objects if they are present in the current image I but not in the background image B.

By performing this evaluation only on the detected objects, it is possible to increase efficiency by computing the edges in small regions corresponding to the object's boundary instead of the whole image

We propose several criteria functions that can be used to determine if object C_i is a valid detection or not. Object C_i is valid detection if the condition is satisfied. Output of this step is a binary ghost mask G.

- **Foreground edge probability:**

$$\frac{|E^F_{C_i'}|}{|C_i'|} > T_G, \tag{10}$$

where $|X|$ is the number of (non-zero) pixels in X. The threshold T_G was empirically chosen to be 0.18.

- **Foreground edge ratio:**

$$\frac{|E^F_{C_i'}|}{|E^I_{C_i'}|} \cdot \max\left(\frac{|E^F_{C_i'}|}{|E^B_{C_i'}|}, 1\right) > T_G \tag{11}$$

is a proportion of the foreground edges in all detected edges modified so that edges appearing in regions with no small number of background edges have higher value.

- **Narrow border.** As described above, for the purpose of edge detection the borders of objects were expanded. The region C_i' can be viewed as a conjunction of the one pixel wide true border region C_i^{tb} with the part obtained by the expansion C_i^{eb}, see Figure 2. Then using only edges on the true border C_i^{tb}:

$$\frac{\sum_{\forall x \in C_i^{tb}} E^F_{C_i'}}{\sum_{\forall x \in C_i^{tb}} E^B_{C_i'}} > T_G \tag{12}$$

is a ratio of foreground edges to background edges on the true border of an object. We chose $T_G = 1$.

- **Edge probability difference.** We can select outer object border C_i^{ob}, as shown in Figure 2, so that C_i^{ob} and C_i' are disjunctive. By comparing probability of edges

on the object border and outside border, we get a criteria function:

$$\frac{\frac{|E^I_{C'_i}|}{|C'_i|}}{\frac{|E^I_{C^{ob}_i}|}{|C^{ob}_i|}} > T_G. \tag{13}$$

This criteria function has the advantage, that it does not require pixel by pixel background model.

3.3. Background Model Update

The task of maintaining up-to-date background model is handled by the background subtraction algorithm described in step 1. We integrated the ghost detection results into the update process.

Taking the ghost mask G, all pixels x labelled as ghost detections are updated in the background model to the current value $I(x)$. The update is done by increasing the weight of the Gaussian $\mathcal{G}_i$ which matches the current value $I(x)$. When next frame is processed, $\mathcal{G}_i$ will be considered by the background model (Equation 1) and x correctly evaluated as background:

$$\forall x : \forall \mathcal{G}_i \in BG(x) : w_i^x = \begin{cases} \frac{w_i^x+\beta}{1+\beta} & \text{if } \mathcal{G}_i \text{ matches } I(x) \\ \frac{w_i^x}{1+\beta} & \text{otherwise,} \end{cases} \tag{14}$$

where β is a constant parameter. Following must also hold:

$$\forall x : \forall \mathcal{G}_i \in BG(x) : \sum w_i^x = 1. \tag{15}$$

4. Evaluation

Because we lack foreground groundtruth, the results are evaluated visually by comparing different criteria functions and results without using the proposed ghost detection.

We are interested in background subtraction for tracking applications and so we also evaluate the effect of ghost detection on overall performance of our tracking system.

Three different types of data are used:

- **Floorball** - A floorball match recorded from multiple viewpoints. The sequence is acquired indoors.
- **PETS2009** - Outdoor dataset recorded from multiple viewpoints. Pedestrians are walking in an university campus.[1] [13]
- **Parking** - Video sequence of a parking car captured on a security camera.[15]

[1] Available online at http://www.cvg.reading.ac.uk/PETS2009/.

5. Experimental Results

Results of the proposed method on video sequence of parking car is presented in Figure 5. It shows that using the ghost detection eliminates false detection caused by movement of a car previously belonging to the background. Outputs of the foreground edge probability (Equation 10) and the narrow border (Equation 12) criteria functions are satisfactory. Foreground edge ratio (Equation 11) has almost no effect in this situation and edge probability difference (Equation 13) produces worse results than no ghost detection. We use only the first three criteria for the data sequences bellow.

Figure 5 shows comparison of the 3 criteria on the floorball dataset. Foreground edge probability and narrow border criteria functions have again good results. The ghost artefacts are soon eliminated, and valid foreground objects remain in the foreground. The method works correctly on objects broken into several parts.

To simulate movement of objects that are considered background, we did not use the gradually changing learning coefficient at the initialization of the experiment on the PETS dataset. Figure 5 shows the results, which are similar to the floorball sequence.

To evaluate the effect of ghost detection on the overall performance of our tracking system, we selected foreground edge probability criteria function (Equation 10). This function provided good results in sense of correctly identifying most ghost detections while it often allows small and only partially detected objects to remain as foreground.

We evaluated the tracking system on 1000 frames of the floorball sequence and 794 frames of the PETS sequence. In Section 3.3 we use the CIE LUC colour space for the background subtraction algorithm. We present a comparison to the results obtained with the RGB colour space.

Using gradual lowering of the learning coefficient discussed in the same section, we simulate scene where objects considered background start to move.

The tracking results are in Figure 3.3, they are evaluated using CLEAR MOT metrics. The multiple object tracking accuracy (MOTA) value accounts for object detection errors, that is false positives, false negatives and identity mismatches. The multiple object tracking precision (MOTP) is the average distance of the true positive detections to the ground truth positions. Details can be found in [12]. We can see that the method has a small negative impact on the performance of the tracking systems in scenes, where it is rare for background objects to move. However, it offers a significant improvement in the cases where this occurs frequently.

	floorball sequences with gradual lowering of the learning coefficient				
ghost detection, colour space	MOTA	MOTP	false negatives	false positives	identity mismatches
no ghost detection, RGB	0.7594	0.2224	61	4	12
ghost detection, RGB	0.7594	0.2256	61	4	12
ghost detection, LUV	0.7563	0.2329	64	4	10
	floorball sequences without gradual lowering of the learning coefficient				
no ghost detection, RGB	0.6063	0.2243	91	22	13
ghost detection, RGB	0.6750	0.2327	87	5	12
ghost detection, LUV	0.6969	0.2326	85	4	8
	PETS sequences with gradual lowering of the learning coefficient				
no ghost detection, RGB	0.9474	8.4979	113	87	8
ghost detection, RGB	0.9315	8.559	177	91	3
ghost detection, LUV	0.9302	8.480	178	91	7
	PETS sequences without gradual lowering of the learning coefficient				
no ghost detection, RGB	0.7884	8.440	207	621	9
ghost detection, RGB	0.9358	8.766	193	56	5
ghost detection, LUV	0.9358	8.518	195	54	5

Fig.3. This table summarizes influence of the ghost detection on the overall performance of a tracking system. The tracking is performed on 1000 frames of the floorball sequence and 794 frames of the PETS sequence. The foreground edge probability criterion function (Equation 10) is used for the ghost detection. An influence of the colour space selection in the background subtraction algorithm is evaluated. The CIE LUV and the RGB colour spaces were used. The gradual lowering of the learning coefficient is used to simulate a movement of objects included in the background model.

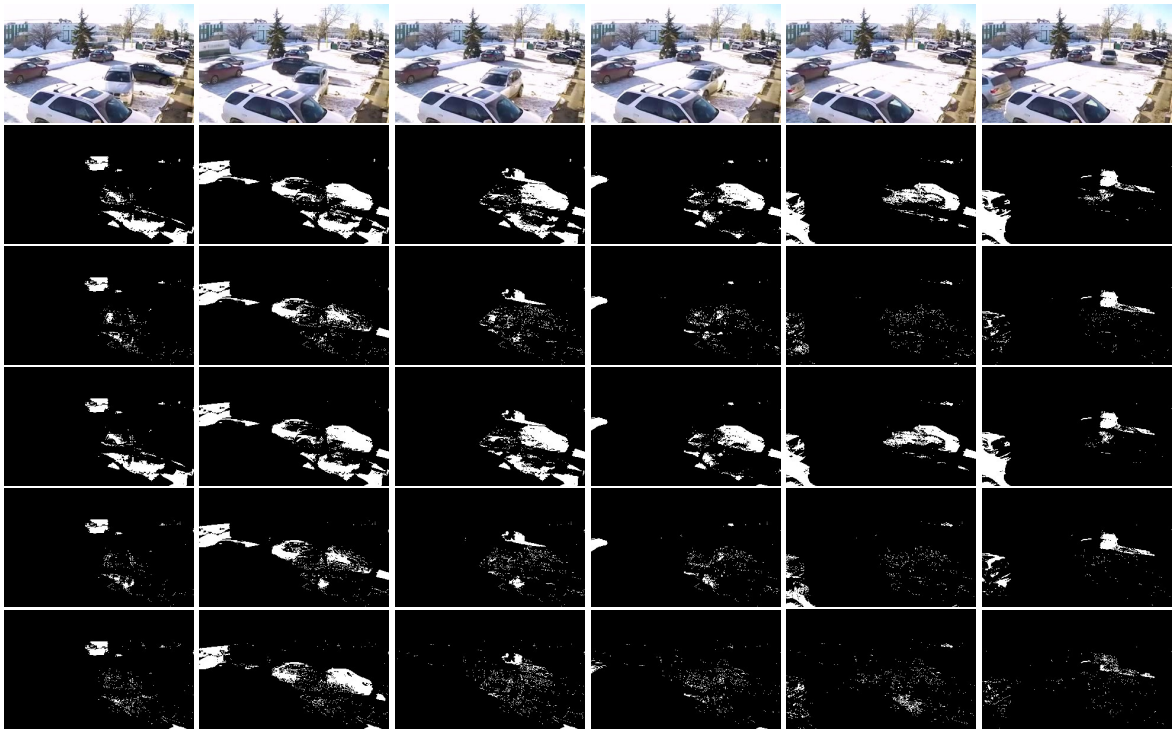

Fig.4. The background subtraction results of different criteria functions for ghost detection on the parking video sequence. The first row contains images from the sequence, the first image comes from the middle of the sequence, following images are selected after 15, 40, 80, 120, 200 frames respectively. In the rows bellow are the corresponding foreground masks. Second row: background subtraction without ghost detection. The foreground masks obtained using different criteria functions follow. Third row: foreground edge probability. Fourth row: foreground edge ratio. Fifth row: narrow border. Sixth row: edge probability difference

6. Conclusions and Future Work

In this paper, we have presented a method to deal with ghost detections based on the presence of edges on the border of detected objects. To identify such detections, we pro-

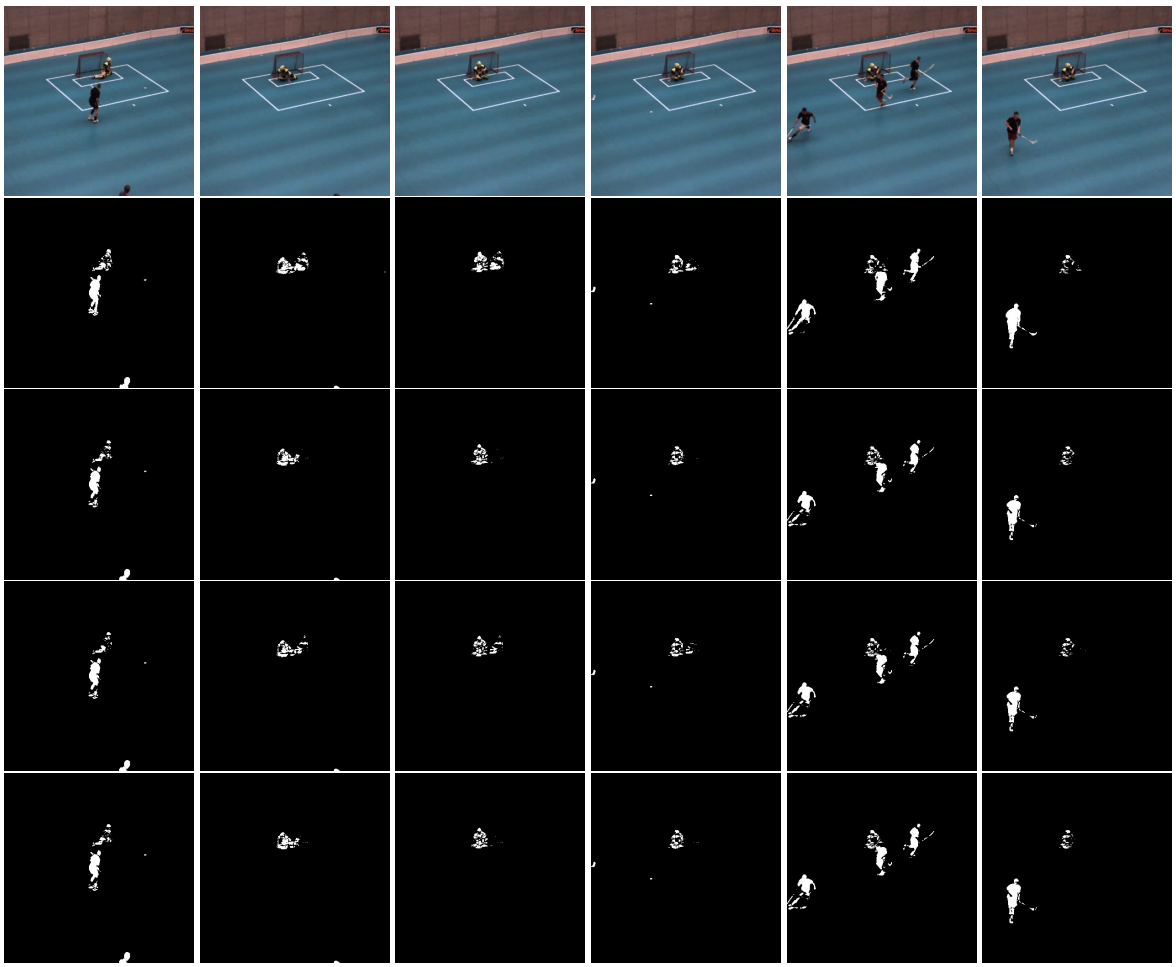

Fig.5. The background subtraction results of different criteria functions for the ghost detection on the floorball video sequence. The first row contains images from the sequence. The first image comes from the middle of the sequence, following images are selected after 40, 60, 110, 150, 190 frames respectively. In the rows bellow are the corresponding foreground masks. Second row: background subtraction without ghost detection. The foreground masks obtained using different criteria functions follow. Third row: foreground edge probability. Fourth row: foreground edge ratio. Fifth row: narrow border.

posed and evaluated 4 different criteria functions. Both outdoor and indoor video sequences were used to evaluate the results.

We showed that the method performs well with two out of the four proposed criteria functions. The background subtraction results are improved in case when there are moving objects present in the scene at the time of initialization, or when objects considered as background start to move. Ghost detections are eliminated and the background model is less affected by the initial values, which results in improved detections in those regions in the subsequent frames.

There are weak spots of the proposed method also. Valid objects may be detected as ghosts when only part of the object is detected as a foreground or the object is broken into parts. This can happen when objects, or their parts, have similar colours as their background or when there are background objects in front of them, e.g. tree branches or road signs.

A missing ghost detection may occur when it overlaps with a moving object. This may result in a delay in the ghosts elimination in case of very slowly moving objects. Problems with recognizing ghost detections may also arise when a ghost detection occurs in a region with complicated edges present in the background.

Finally we have shown a positive impact on the performance of a tracking system. We have incorporated the ghost detection into the background subtraction module of the tracking system and compared the performance with the original version.

The main advantages of the method are:

- independence on a background subtraction algorithm,
- computational efficiency - only foreground regions of the input image are evaluated.

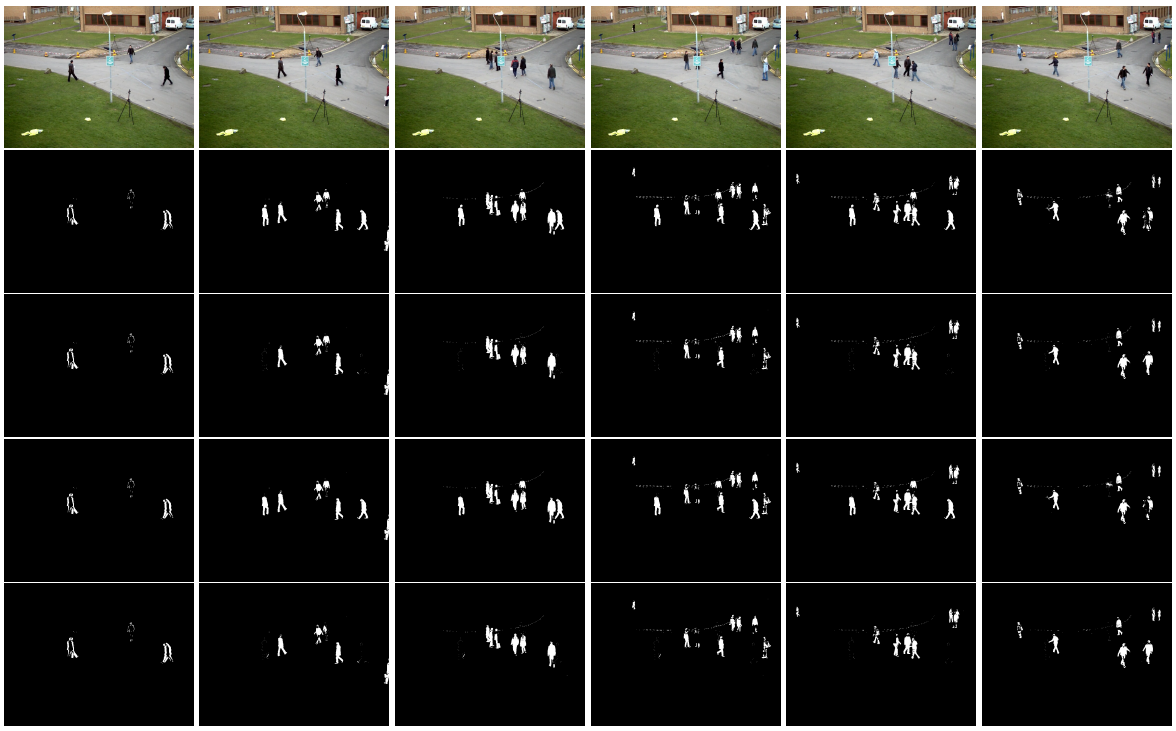

Fig.6. The background subtraction results of different criteria functions for the ghost detection on the PETS video sequence. For this experiment, the gradual change of learning coefficient was not used in order to simulate situations, when objects in the background start to move. The first row contains 0th, 10th, 60th, 120th, 160th and 190th frames. In the rows bellow are the corresponding foreground masks. Second row: background subtraction without ghost detection. The foreground masks obtained using different criteria functions follow. Third row: foreground edge probability. Fourth row: foreground edge ratio. Fifth row: narrow border. Sixth row: edge probability difference.

Acknowledgements

The authors would like to thank prof. Ing. Jiri Matas, Ph.D. for kind supervision.

References

[1] BENEZETH, Y., et al. "Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms." *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on.* IEEE, 2008.

[2] JAVED, O., KHURRAM, S., SHAH, M. "A Hierarchical Approach to Robust Background Subtraction Using Colour and Gradient Information." *Motion and Video Computing, 2002. Proceedings. Workshop on.* IEEE, 2002.

[3] JABRI, S., DURIC, Z., WECHSLER, H., ROSENFELD, A. "Detection and Location of People in Video Images Using Adaptive Fusion of Colour and Edge Information." Pattern Recognition, 2000. Proceedings. 15th International Conference on. Vol. 4. IEEE, 2000.

[4] CALDERARA, S., et al. "Reliable Background Suppression for Complex Scenes." *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*. ACM, 2006.

[5] CUCCHIARA, R., et al. "Detecting Moving Objects, Ghosts, and Shadows in Video Streams." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.10 (2003): 1337-1342.

[6] KAEWTRAKULPONG, P., BOWDEN, R. "An Improved Adaptive Background Mixture Model for Real-Time Tracking with Shadow Detection." *Video-based surveillance systems.* Springer US, 2002. 135-144.

[7] CHEUNG, S., Kamath, C. "Robust Techniques for Background Subtraction in Urban Traffic Video." *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004.

[8] WANG, H., SUTER, D. "Background Subtraction Based on a Robust Consensus Method." *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on.* Vol. 1. IEEE, 2006.

[9] BARNICH, O., VAN DROOGENBROECK, M. "ViBe: A Universal Background Subtraction Algorithm for Video Sequences." *Image Processing, IEEE Transactions on* 20.6 (2011): 1709-1724.

[10] SPAGNOLO, P., Leo, M., Distante, A. "Moving Object Segmentation by Background Subtraction and Temporal Analysis." *Image and Vision Computing* 24.5 (2006): 411-423.

[11] SHOUSHTARIAN, B., BEZ, H. E. "A Practical Adaptive Approach for Dynamic Background Subtraction Using an Invariant Colour Model and Object Tracking." *Pattern Recognition Letters* 26.1 (2005): 5-26.

[12] KENI, B., Rainer, S "Evaluating multiple object tracking performance: the CLEAR MOT metrics." *EURASIP Journal on Image and Video Processing 2008* (2008).

[13] Ellis, A.,Shahrokni, A. Ferryman, J.M., PETS2009 and Winter-PETS 2009 results: A combined evaluation, *in 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009, pp. 18.

[14] SCHANDA, J. *Colourimetry: Understanding the CIE system.* John Wiley and Sons, (2007): 61-64

[15] KIDZIDZI, K. *Youtube video, Dec 4, 2014*, 2014, Retrieved from https://www.youtube.com/watch?v=HYcsW48HHXM

About Authors…

Jiri PYTELA - born in Prague. Currently student of a Master study program Robotics and Cybernetics at FEE, CTU in Prague. During his Bachelor study period he spent 6 months

as intern in INRIA Grenoble working on a research project in the field of affective computing.

Matej SMID - currently PhD student at the Center for Machine Perception, Department of Cybernetics, FEE, CTU interested in sports tracking and articulated objects tracking. After finishing Master studies at CTU, Matej spent more than 5 years working in the field of computer vision at Software Compentence Center Hagenberg GmbH. He also taught computer vision at University of Applied Sciences Upper Austria.