A NOVEL METHOD FOR TRACKING MOVING OBJECTS USING BLOCK-BASED SIMILARITY

M. Mahdavi*

Department of Computer Engineering, University of Guilan
P.O. Box 3756, Guilan, Iran
mehregan_m@hotmail.com

S.N. Shahrouzi and R. Hasanzadeh

Department of Electrical Engineering, University of Guilan
P.O. Box 3756, Guilan, Iran
s_n_shahrouzi@yahoo.com - hasanzadepak@gmail.com

*Corresponding Author

(Received: November 28, 2007 – Accepted in Revised Form: September 25, 2008)

Abstract Extracting and tracking active objects are two major issues in surveillance and monitoring applications such as nuclear reactors, mine security, and traffic controllers. In this paper, a block-based similarity algorithm is proposed in order to detect and track objects in the successive frames. We define similarity and cost functions based on the features of the blocks, leading to less computational complexity of the algorithm. Therefore, this method is suitable for real-time tracking. According to the experimental results, this method has a good performance and works well for occluded objects, cluttered environments and noisy sequences.

Keywords Object Tracking, Background Subtraction, Machine Vision, Cost Function, Similarity Function

چکیده تشخیص و ردگیری اشیاء متحرک دو مسئله عمده در کاربردهای مراقبتی و نظارتی نیروگاه‌های اتمی، امنیت معادن، و کنترل ترافیک می‌باشند. در مقاله حاضر یک روش تشابه مبتنی بر بلوک برای تشخیص و ردگیری اشیاء در تصاویر متوالی ارائه شده است. توابع تشابه و هزینه بر اساس خصوصیات بلاک‌ها محاسبه می‌گردند. در نتیجه این روش برای ردگیری بلادرنگ مناسب می‌باشد. بر اساس نتایج تجربی، این روش دارای کارایی خوبی بوده و در قبال اشیاء مسدود شده، محیط‌های شلوغ و دنباله‌های نویزداربه خوبی عمل می‌کند.

1. INTRODUCTION

There are a number of algorithms for automatic extraction and tracking objects. However, issues such as objects overlapping, changes in the objects' pose, and cluttered backgrounds are the most challenging research topics. An algorithm called mean-shift (MS) [1] is popular among vision tracking community and has been used in many tracking applications [2-7]. However, mean-shift tracker and blob model [8] are usually sensitive to the effect of occlusion and noise. Computational complexity is also an important factor for tracking in real-time applications.

In order to decrease the sensitivity to occlusion and noise, we propose an algorithm using block-based similarity measure. By using block-based similarity, the tracker uses features of the blocks instead of pixels. For segmenting foreground from background, we use a block-based background subtraction method. The objects, after being extracted from each frame, will be represented by the features of their corresponding blocks. Dimensions of the blocks can depend on the size of the smallest object being tracked. It can also depend on the amount of the desired clarity of the objects based on the condition of the scene.

By using the block-based similarity measure in the image sequences, noise reduction is actually applied. Moreover, the computational complexity is considerably reduced compared to the pixel-based approaches, which is an important issue in real-time applications.

The experiments show that the proposed algorithm has a good performance and works well in presence of occlusion, pose changing, and noise.

This paper is organized as follows: In Section 2, we discuss the architecture of the system. Section 3 explains how objects are extracted from the background. The tracking algorithm is described in Section 4. Section 5 presents the experimental results. Concluding remarks are given in Section 6.

2. SYSTEM ARCHITECTURE

Similar to common tracking, our tracking system has two main units: the foreground-background segmentation unit and the object tracking unit. The input frames are fed into the foreground-background segmentation unit for being segmented into foreground (objects) and background, as shown in Figure 1. Background subtraction method is used to implement this process.

The object tracking unit links up the objects in the current and previous frames. This unit is implemented by using block-based similarity measure of the objects. In order to achieve the desired results, this unit uses cost and similarity functions.

Since the system uses a block-based similarity measure, the input sequence is initially transformed into block view, which is shown in Figure 2. The features of these blocks are used by the segmentation and tracker units.

3. FOREGROUND-BACKGROUND SEGMENTATION

The foreground-background segmentation is performed in six steps. As shown in Figure 3, the input frame will be transformed from RGB to HSI format. The intensity of each frame contains very important information, which we use in our tracking algorithm. Therefore, this transformation is necessary to extract the intensity of the frames from color information. Next step, the grey scaled images are divided into blocks with n × n pixels. The extracted local and spatial features such as mean, median, maximum, minimum and pixel coordinates of the blocks will be used for similarity measures.

The first four features show the differences between the blocks. The last feature is used as a separator when similarity functions are being determined.

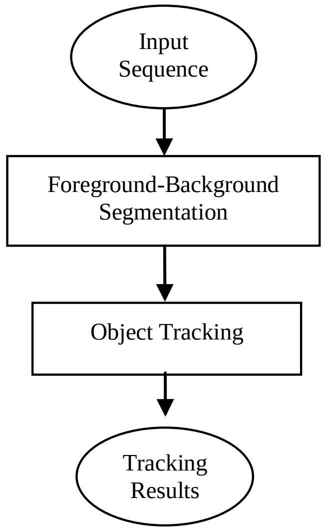


Figure 1. Overall tracking system structure.

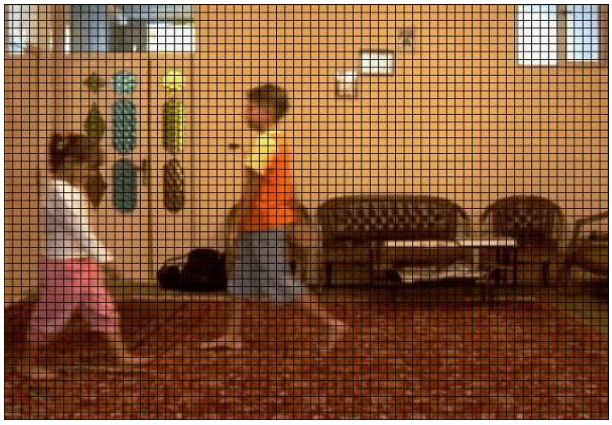


Figure 2. Block view of frames.

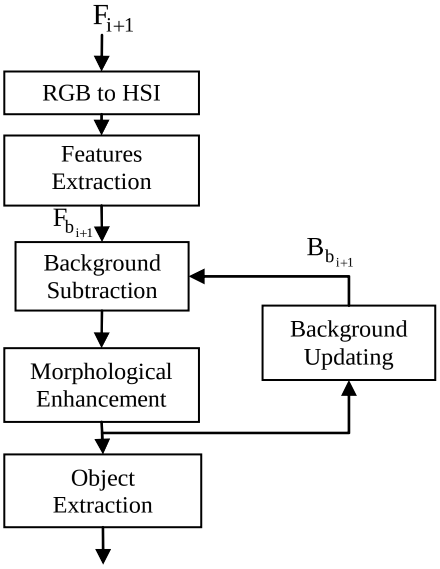


Figure 3. Foreground-background segmentation steps.

The block-based representation of images has the following advantages:

- Using the local features of each block such as the mean value results in a better noise elimination approach.
- It reduces the computational complexity, since the algorithm works with the features of the blocks instead of pixels.
- Since the local features are used, the algorithm is less sensitive to small changes in the pixels.
- By using blocks, intensity of each block is represented by the mean value of pixels. As a result, cluttered background has less effect on this algorithm.
- The median operator helps tracker to overcome impulse noise.
- Maximum and minimum values of blocks are useful features in noisy images.

It seems that using this approach for representing frames, the tracker with a proper cost function and a good aggregation of similarity features between old and new objects, works more efficiently than the one that uses pixel-based calculations. We have used background subtraction [9-14] to separate objects from background.

In background subtraction step, $B_{b_{i+1}}$ (the mean value of the background blocks) is subtracted form $F_{b_{i+1}}$ (the mean values of the frame blocks). i is the index of the current frame. Using an appropriate threshold, the objects will be extracted from the image. The threshold was selected intuitively by the average of subtracted values of 300 successive frames without foreground. It generally depends on the average changes of the illumination of images and the pattern of the background.

Because of some defects in the blocks, objects such as unfilled areas and winglets, some morphological tools are used to improve the representation of the moving objects.

Our model for background subtraction is focused on the mean values of blocks. It is being updated in each frame and because of block-based representation of pixels it is less sensitive to noise and also slight changes in the background. Therefore, it is expected to perform well for tracking moving objects outdoor.

The next background updating step, is performed by averaging the mean values of the current frame and the mean values of the current blocks as follows

$$B_{b_{i+1}}(s) = \begin{cases} \frac{1}{2}\left(B_{b_i}(s) + F_{b_i}(s)\right) & \forall s \in \text{Background} \\ B_{b_i}(s) & \forall s \in \text{Foreground} \end{cases} \tag{1}$$

Where, s is a block set of the image and $B_{b_{i+1}}$ is used as the input for the next background subtraction. This equation calculates the average mean values of the previous frame and the mean values of the previous background blocks of the current frame.

In object extraction step, related components are labeled as different moving objects and the results are fed into the tracker.

4. TRACKING UNIT

The correlation between target objects of two consecutive frames, perform an important role in tracking of the moving objects.

Figure 4 shows the process of a tracker, which includes three steps. After this process, the separated objects from their respective background in the new frame are classified into one of the following categories:

- Related to one of the objects in the previous frame.
- As an object that is generated by combining some of the objects in the previous frame.
- The other found objects are compared with the lost objects. If the result of the comparison is positive, the objects will be related to them. Otherwise, they will be classified as new objects.

Each object will be formed by the average mean values of the corresponding blocks which are computed after being fed into the tracker. This is recorded for the objects of the current frame until the process of the next frame begins. After that, the minimum value of similarity between each object in the previous and new frame is calculated, in order to determine the pairs of blocks with the lowest cost function.

The cost function is computed for each pair of blocks, where the difference between the numbers of blocks for each selected object does not exceed a threshold. A threshold is also applied to the average mean values of the blocks. These thresholds are experimentally determined. We have examined different values and noticed that choosing values equal to, or around the difference

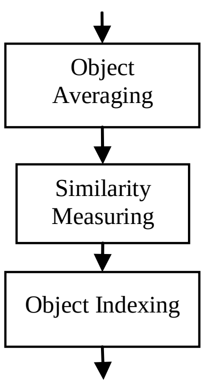


Figure 4. Tracking unit structure.

between the number of blocks and the difference of average mean values of two objects generate good results. However, they depend on maximum velocity of objects (when the distance of objects from the camera changes) and also the threshold used for background subtraction.

The cost function between each block of the object in the previous and also in the current frame is determined as follows:

$$\begin{aligned}
&\mathrm{CostF}\left(\mathrm{old}_p(j), \mathrm{new}_q(k)\right) = \\
&\left|\mathrm{mean}_p(j) - \mathrm{mean}_q(k)\right| + \\
&\left|\mathrm{median}_p(j) - \mathrm{median}_q(k)\right| + \\
&\left|\mathrm{max}_p(j) - \mathrm{max}_q(k)\right| + \\
&\left|\mathrm{min}_p(j) - \mathrm{min}_q(k)\right| + \\
&\sqrt{\frac{\left(x_p(j) - x_q(k)\right)^2}{N-1} + \frac{\left(y_p(j) - y_q(k)\right)^2}{M-1}}
\end{aligned} \tag{2}$$

Where, $\mathrm{mean}_p(j)$, $\mathrm{median}_p(j)$, $\mathrm{max}_p(j)$, $\mathrm{min}_p(j)$, $x_p(j)$ and $y_p(j)$ are respective mean, median, maximum, minimum and coordinates of j th block of p th object in the previous frame ($\mathrm{old}_p(j)$). Moreover, $\mathrm{mean}_q(k)$, $\mathrm{median}_q(k)$, $\mathrm{max}_q(k)$, $\mathrm{min}_q(k)$, $x_q(k)$ and $y_q(k)$ are respective mean, median, maximum, minimum and coordinates of k th block of q th object in the current frame ($\mathrm{new}_q(k)$). N and M are the width and length of the image. They are used to transform x and y values in the range of 0 and 1.

The values calculated for the cost function are then used in the similarity function that is explained below.

The similarity function between each object (q th) in the new frame and each object (p th) in the previous one is calculated using Equation 3.

$$\begin{aligned}
&\mathrm{Similarity\,F}(p, q) = \\
&\textstyle\sum_{k \in q} \min_{j \in p}\left(\mathrm{Cost\,F}\left(\mathrm{old}_p(j), \mathrm{new}_q(k)\right)\right) \\
&+ \left|N(p) - N(q)\right|
\end{aligned} \tag{3}$$

Where, N(p) and N(q) are the number of blocks for p th object in the previous frame and q th object in the new frame, respectively. In fact, similarity between two objects is the aggregation of minimum cost function values of their blocks added by the difference between the number of blocks that form each objects.

We use $\sum_{k \in q} \min_{j \in p}$ instead of $\sum_{j \in p} \min_{k \in q}$, in order to calculate the amount of similarity of each new object in the current frame and the old objects in the previous frame. These two terms could be exchanged and the results will be the same. However, the algorithm should be modified to reflect these changes for not processing the sequence in reverse.

The second term is put as a separator for unrelated objects. As a result the calculations are done on more similar objects.

In order to relate the objects in the successive frames, a criterion for similarity of objects should be used. We propose the above-mentioned equations based on the features of the blocks. However, they could be defined differently. Our experiments show that they are compatible with our algorithm and generate good result.

Similarity values are registered in a Similarity Matrix (SM). Rows in the SM represent the number of objects in the current frame and columns represent the number of objects in the previous frame. The elements of the matrix are the similarity values of selected rows and columns (q and p). For objects that are not similar, the elements are labeled as unknown.

Finally, the relationships between objects in the last two consecutive frames are extracted from the similarity matrix in object indexing step. This is done by the following algorithm:

```
While (SM is known)
{
        Select min(SM).
        Determine q and p of selected item.
        Mark SM(q,p) as unknown.
        q th Object is related to p th object.
}
```

As a result of this algorithm, q th object in the current frame and p th object in the previous frame are known as the same objects. This algorithm works similar to the Hungarian method [15]. After finishing the above iteration, some of the objects in the last two consecutive frames may still be unknown. Unknown objects in the previous frame and the current frame are called lost objects and appeared objects, respectively. A record of each lost object with information about the number of its blocks and their mean values will be produced. Since, it is possible to mix up a lost object with another object; we use proper thresholds (set using the size of objects) in the number of blocks and the distance between the centroids of two objects to specify whether these two objects are mixed up or not. However, appeared objects in the current frame are compared with lost objects and if they match, the lost object will be removed from the list. Otherwise, it will be considered as a new object in the scene. For applying, matching process to the lost and appeared objects, they should pass through a threshold determined based on the number of their blocks. Then, they are compared with a subtraction function that is defined in Equation 4.

$$\mathrm{SubtractionF}(e,f) = \frac{\sum_{c \in e} \min_{d \in f} (\mathrm{CF}(c,d))}{N(e)} \qquad (4)$$

Where, e and c are the number of lost objects and their number of blocks, respectively. f and d are the number of appeared objects and their number of blocks, respectively. N(e) is the number of blocks included in e th lost object. CF(c,d) is another cost function:

$$\mathrm{CF}(c,d) = \left| \mathrm{mean}_e(c) - \mathrm{mean}_f(d) \right| \qquad (5)$$

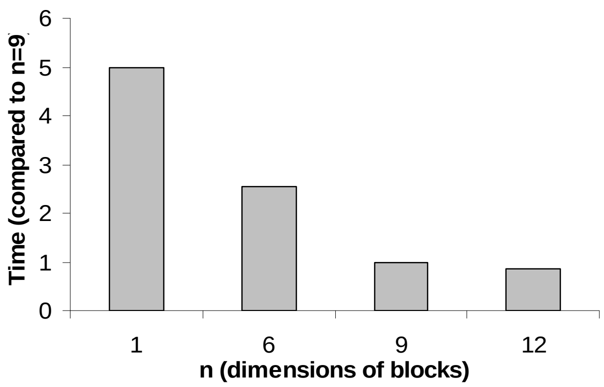
Where, $\mathrm{mean}_e(c)$ and $\mathrm{mean}_f(d)$ are the mean values of c th block of e th lost object and d th block of f th appeared object.

For each two objects, they are labeled as similar objects if the subtraction value is less than the threshold used in the background subtraction unit. Otherwise, the appeared object is called a new object in the scene. Tracking moving objects is performed by iterating this algorithm on the successive video frames.

5. EXPERIMENTAL RESULTS

This block-based similarity method is tested by several image sequences. The image sequences we used, to test our algorithm were caught by a digital video camera capable of recording 30 fps. The frames are 480 × 640 pixels. All of the image sequences were indoor videos with cluttered background including colorful carpets, windows and furniture. For this environment we use 9 × 9 pixel block sets (n = 9). We selected n = 9, by trial and error to get the best results. We applied several values for n. There was a trade-off between the time and accuracy. We have found that n = 9 is the best choice for the selected scene. Larger values for n could be selected. However, contour of objects would become rectangular and some pixels of the background would be added to the objects being tracked. The result is shown in Figure 5. The value of n actually depends on the pattern of the background and the objects, the size of the objects and the resolution of each frame. With some prediction about the size of the smallest object that should be detected, we can have good assumption for an unknown scene.

The first video, as shown in Figure 6, contains two people who walk towards each other and then pass by one another. The result of background subtraction is illustrated in Figure 7. It shows that this unit works properly to separate the objects from the background. Figures 8 and 9 show the outputs of tracking for the girl and the boy, respectively. The tracker could track targets after occlusion, but when the people passed by one another, the tracker realized the mixed object as the representation of both objects.



(a)

n	Accuracy
1	Noisy Objects
6	Good
9	Good
12	Rectangular Contour

(b)

Figure 5. Results for different values of n (a) Performance and (b) Accuracy.

Figure 6. The selected frames for the first input sequence.

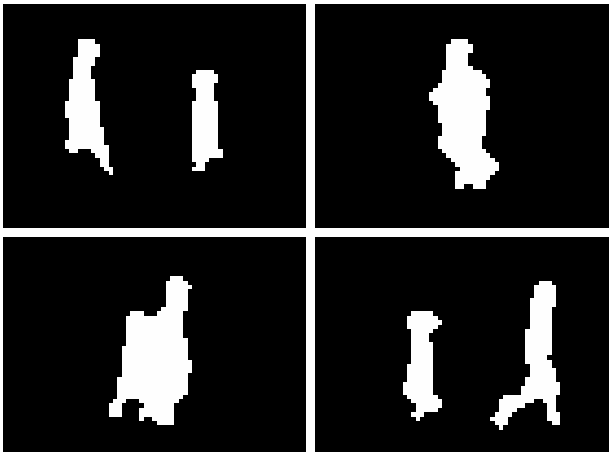


Figure 7. Background subtraction results for the selected frames.

Figure 8. Tracking results for the girl.

Figure 9. Tracking results for the boy.

In the second video, a person walks from one side of the scene to the other side, then sits, stands up, and then walks to the other side, turns around and walks until he goes out of the scene. Some frames are shown in the left column of Figure 10. The results of the tracker are shown in the right column. As shown in Figure 10, the tracker did not lose the boy, even while he was sitting. Moreover, no problem occurred by changing his distance and view from the camera.

The results indicate that this algorithm works properly for tracking purposes and is not sensitive to occlusion and changes in the objects' pose.

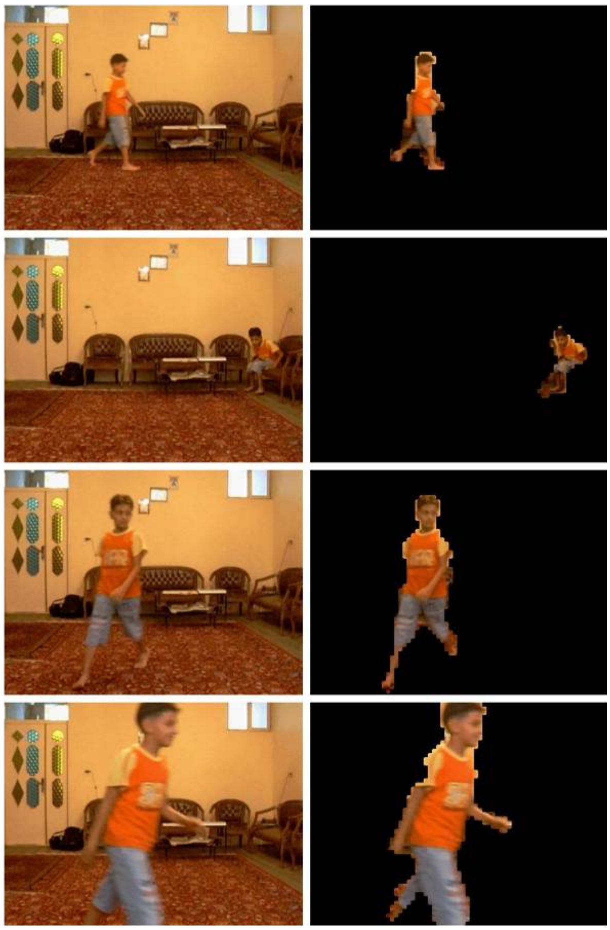


Figure 10. The selected frames for the second input sequence (right column).

6. CONCLUSIONS

In this paper, we have proposed an object tracking algorithm based on similarity of the blocks of objects. Block-based representation of frames provides less computational complexity. Therefore, it is suitable for real-time applications. It also addresses the issue of small changes in the background. This algorithm is able to track objects even if occlusion situation and cluttered background occurs. It also works well when the view of objects changes. The results are achieved by processing the intensity of images.

The number of objects that can be tracked simultaneously is not limited in the algorithm. However, it affects the amount of computations required.

Our algorithm is sensitive to illumination changes. This problem occurs when the side of the objects facing light source changes during the tracking period. It is because of big changes in the intensity of blocks of objects and the tracker fails to follow the objects.

There is a distance threshold for specifying an object as a mixed object. Therefore, when the objects move faster we should make some revisions in the threshold. In addition, to make the tracker more robust, we can dynamically set the size of the blocks. For example, in perspective scenes such as highways, roads, and intersections, smaller size of blocks can be considered for farther points in the image. These issues are left for the future.

7. REFERENCES

1. Li, Z., Tang, Q. L. and Sang, N., "Improved Mean Shift Algorithm for Occlusion Pedestrian Tracking", ***Electronic Letters***, Vol. 44, No. 10, (2008), 622-623.
2. Porikli, F. and Tuzel, O., "Multi-Kernel Object Tracking", ***In IEEE International Conference on Multimedia and Expo***, (2005), 1234-1237.
3. Shen, C., Brooks, M. J. and Van Den Hegel, A., "Fast Global Density Mode Seeking: Application To Localization and Tracking", ***IEEE Trans Image Processing***, Vol. 16, No. 5, (2007), 1457-1469.
4. Yang, C., Duraiswami, R. and Davis, L. "Efficient Mean-Shift Tracking Via a New Similarity Measure", ***In IEEE Conf. on Computer Vision and Pattern Recognition***, Vol. 1, (2005), 176-183.
5. Shan, C., Wei, Y., Tan, T. and Ojardias, F., "Real Time Hand Tracking by Combining Particle Filtering and Mean Shift", ***In IEEE International Conference on Automatic Face and Gesture Recognition***, (2004), 669-674.
6. Papanikolopoulos, N. P., Nelson, B. J., Khosla, P. K., "Six Degree-of-Freedom Hand/Eye Visual Tracking with Uncertain Parameters", ***IEEE Trans Robotics and Automation***, Vol. 11, No. 5, (1995), 725-732.
7. Deguchi, K., Kawanaka, O. and Okatani, T., "Object Tracking by the Mean-Shift of Regional Color Distribution Combined with the Particle-Filter Algorithms", ***International Conference on Pattern Recognition***, Vol. 3, (2004), 506-509.
8. Wren, C., Azarbayejani, A., Darrell, T. and Pentland, A., "Pfinder: Real-Time Tracking of the Human Body", ***IEEE Trans. Pattern Anal. Machine Intl***., Vol. 19, (1997), 780-785.
9. Maddalena, L. and Petrosino, A., "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications", ***IEEE Trans Image Processing***, Vol. 17, No. 7, (2008), 1168-1177.
10. Matsuyama, T., Wada, T., Habe, H. and Tanahashi, K., "Background Subtraction under Varying Illumination", ***Systems and Computers in Japan***, Vol. 37, (2006), 77-88.
11. Friedman, N. and Russell, S., "Image Segmentation in Video Sequences: A Probabilistic Approach", ***In Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence***, (1997), 1-3.
12. Elgammal, A., Harwood, D. and Davis, L., "Non-Parametric Model for Background Subtraction", ***In Proceedings of the 6th European Conference on Computer Vision***, (2000), 751-767.
13. Stauffer, C. and Grimson, W., "Adaptive Background Mixture Models for Real-Time Tracking", ***Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition***, (1999), 246-252.
14. Toyama, K., Krumm, J., Brumitt, B. and Meyers, B., "Wallflower: Principles and Practice of Background Maintenance", ***In Proceedings of IEEE International Conference on Computer Vision***, (1999), 255-261.
15. Kuhn, H. W., "The Hungarian Method for the Assignment Problem", ***Naval Research Logistics Quarterly***, Vol. 2, (1955), 83-97.