International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.1 (2014), pp.123-132
http://dx.doi.org/10.14257/ijmue.2014.9.1.12

Exploiting Adaptive Background Image and Dynamic Search Window for Fast Object Tracking

Daewon Jung[1] and Changsoo Lee[2]

[1] R&D Strategy Department, The Attached institute of ETRI
P.O.Box 1, Yuseong Daejon, 305-600 South Korea
[2] Dept of Computer Science, Soongsil University
Sangdo dong, Dongjak-gu, Seoul, 156-743, South Korea

dwjung@ensec.re.kr, powerofmicro@naver.com

Abstract

Recently, due to the interest for personal safety, intelligent image recognition technology using the CCTV has received a lot of attention in many areas. Real-time video or image tracking technology is the technology to continuously track the movement of the target object and maintain the specified information of the object. However, the problem is that unnecessary noise or environmental changes of the input video from CCTV make it difficult to accurately detect and track the target object. In this paper, we propose a fast object tracking algorithm, which is divided into two parts: i) object detection process and ii) object tracking process. In the background image replacement technique, we compute the difference between the initial input image and input video image. Then, it replaces N×M blocks based on the changed pixels, finally leading to replacing the background images. If it changes over the pre-defined threshold value, we set the block size and generate the variable search window for object tracking. Since the variable search window only generates the minimum search area for object tracking, our proposed method is quickly able to compute the variable region for object tracking. Through various experiments, our proposed scheme outperforms the other object detection and tracking methods in terms of object tracking counts and tracking error counts.

Keywords: *Multimedia, Image Processing, Object Tracking, Object Detection, Camera Security*

1. Introduction

In recent years, due to the rapid increase of crime and interest for personal safety, intelligent image recognition technology using the CCTV has become more important. Furthermore, the demand of image and video analysis solutions for building intelligent and self-monitoring-enabled system has been constantly increasing. With respect to these demands, most of security solution providers have a plan to provide intelligent monitoring system in the near future. However, as the applicable areas are getting expanded, research on image resolution, transmission rate, object region recognition and tracking technologies is highly required. Real-time video tracking technology is one of the core technologies for building the intelligent video recognition technology. This technology continuously tracks the moving object and main the key information about the object.

In general, video storage technology uses the motion compensation method to remove the temporal redundancy of time and to compress the video. Under the motion compensation

ISSN: 1975-0080 IJMUE

method, it estimates the object movement by using the motion vector, which can make it possible to transmit the background image and object independently from the video. Note that we need the algorithm that significantly reduces the computational complexity, because the motion vector needs to be processed in real-time.

The proposed method for detecting and tracking the object in this paper is one of the way using kernel tracking. Specifically, our proposed tracking algorithm composes of two stages: one is the object detection stage using the background image replacement, and the other one is the stage for tracking the object.

This paper is organized as follows: In Section 2, we provide a summary of related work dealing with object detection and tracking methods. In Section 3, we propose a fast object tracking algorithm using background image changes and dynamic search window. The empirical evaluation of the proposed tracking algorithm follows in section 4. Finally we conclude the paper in Section 5.

2. Related Work

In this section, we briefly provide the related work for object detection and object tracking methods. In order to investigate the existing object detection and tracking technology, we will introduce a kernel tracking algorithm, a difference image method, and a block matching method, which are mostly used for object detection and tracking methods.

In terms of the object detection technology to distinguish between the background image and the object image from a real-time video, the difference image method and the block matching methods have been widely used in this field. When it comes to the object tracking technologies, object tracking methods can be categorized into three methods: point tracking, kernel tracking, and silhouette tracking [1-4] as indicated in Figure 3. Point tracking is the method that estimates the position of the object based on position, velocity, and acceleration. This kernel tracking method is similar to the Kalman Filters in terms of the input data. Kernel tracking generates color histogram similar to Camshift algorithm, and then estimates the position of pixels. By using this method, it can easily track the movement of the target object [8]. However, there are some computational drawbacks using the kernel tracking method in the case of a large number of pixels or a high computational complexity.

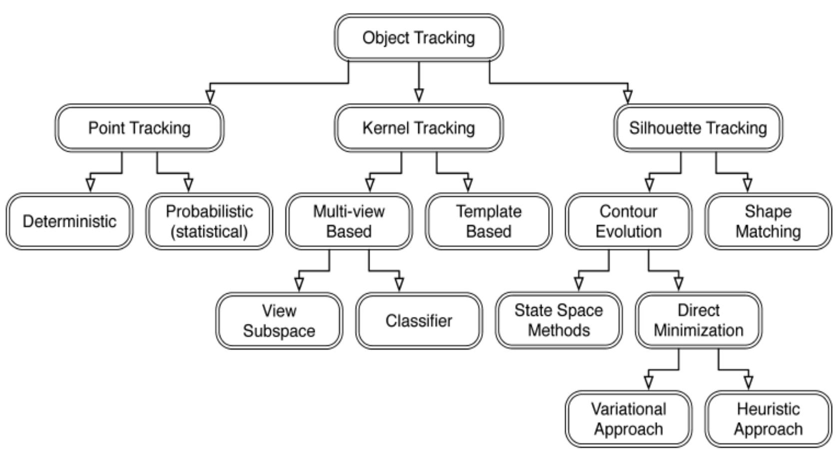


Figure 1. Categorization for object tracking technologies

2.1. Difference image method

In the difference image method for detecting the object movement, two methods are genera. The first one is the method using the differences between the current frame and the previous frame. The second method is using the differences between the background image and input frame. Under the method using the differences between two frames, if the difference between the current and previous frames is less than the threshold, it does not detect the object because there are not so many pixels changed. To the contrary, if the difference exceeds the threshold, it is regarded as an object inflow so that the object is detected [4-5].

The method using the differences between the background image and input frame first sets the standard background image. Then it computes the differences between this standard image and the current frame. This method is shown in Figure 2.

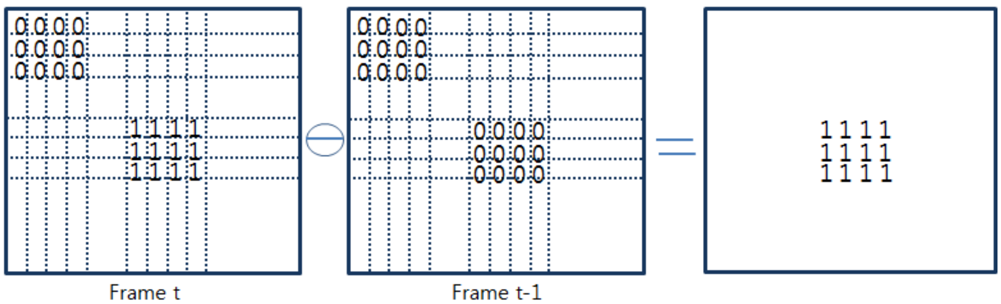


Figure 2. Difference image method

2.2. Block matching method

The block matching method is a technique for estimating the current frame from the previous frame by the unit of block under the assumption that every pixel has the same motion vector. First, the frame is divided into constant size of the block. Then, it can estimate the current block to be decoded based on the most similar block in the previous frame. In Figure 2, the size of the current block and the candidate block is $N \times M$, the navigation area has a value between –W and +W in the horizontal and vertical, respectively [6].

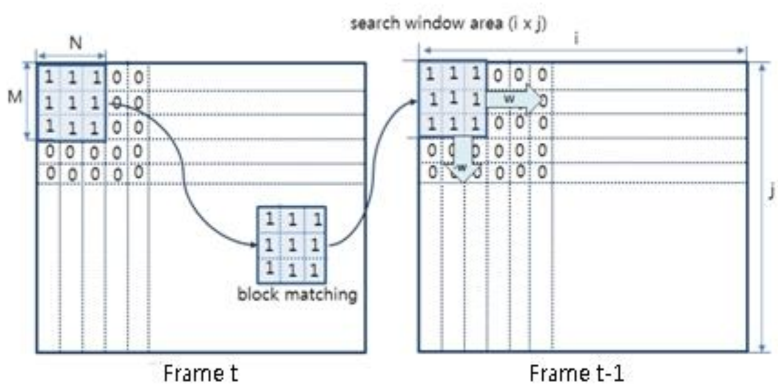


Figure 3. Block matching method

3. Proposed Adaptive Background Image and Dynamic Search Window

In this section, we will describe our proposed fast object tracking algorithm by using adaptive background image and dynamic search window. When we compare the inter-frame pixel for object detection, some of pixel in the background acts like a noise due to the time of illumination changes and environment changes. Note that these factors make it difficult to detect the object from the background.

In the background image method, we first compare the background image with the input image by the block. If the computed value is in the range between 0 to the threshold, we change the block image in the input frame to the background image. To the contrary, if the computed values exceed the threshold, we determine this situation as the influx of the object so that we generate the multiple of two dynamic search windows and the object block by computing the maximum value in the horizontal and vertical axis of the block changed. In the dynamic search window, the rectangle that shapes the object keeps track of the objects within the window area in the opposite direction, the vertical and horizontal direction, respectively.

We emphasize here that the proposed method maintains a constant computational complexity regardless of the size of the image resolution, finally leading to a fast object tracking. Figure 4 depicts the overall system configuration of the proposed fast object tracking method.

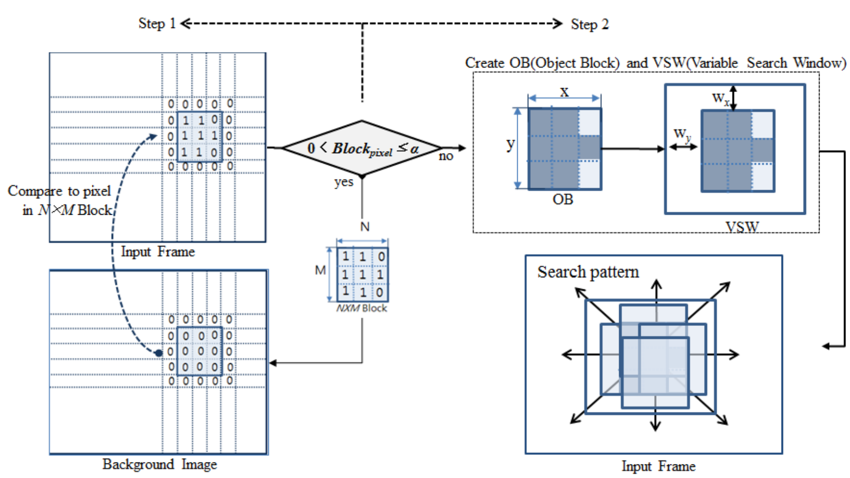


Figure 4. Object tracking algorithm using adaptive background changes and variable search window

3.1. Adaptive background changes

In general, the background image detects the object by comparing the initial input image with the next frame. However, some parts of images changed due to the illumination or environment changes are the factors for making it hard to detect the object. As we mentioned previously, our proposed adaptive background change method changes the background image when the computed value is in the range of between 0 and the threshold.

Since one pixel in image has all RGB colors, it updates the pixel by the unit of block when the difference between the value of pixel for RGB colors and that of the next frame is less than the threshold value.

In (1), it indicates the average value for the differences between the current frame and the next frame in N×M block.

$$P_{RGB\_chanel} = AVG\{\sum_{i=0}^{N}\sum_{j=0}^{M} Background_{RGB\_channel}(i,j) - NextFrame_{RGB\_channel}(i,j)\}, \quad (1)$$

where $P_{RGB\_channel}$ is the average difference value of RGB channels between two frames and $Background_{RGB\_channel}$ is the pixels within a block of background image. The last one, $NextFrame_{RGB\_channel}$ indicates RGB channel for the next frame.

$$CP_{Pixel\_check} = \begin{cases} 1, & 0 < P_{RGB\_channel} \le \alpha \\ 0, & Otherwise \end{cases} \quad (2)$$

In (2), based on the computed value of $P_{RGB\_channel}$ is less than or equal to the threshold (defined by α), it changes the N×M block or it detects the object.
Figure 5 shows the method for background image replacement.

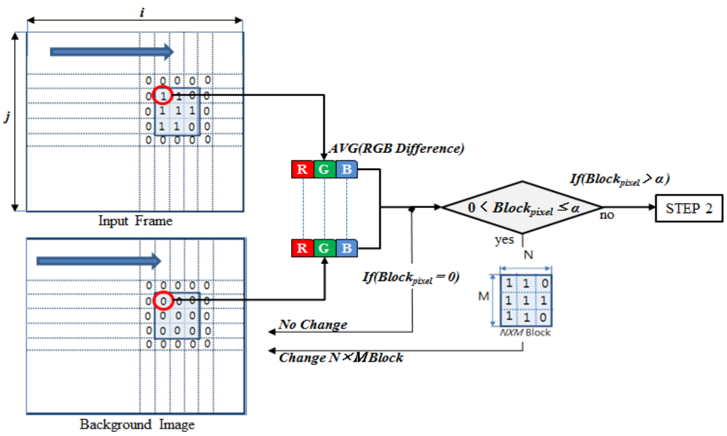


Figure 5. Method for changing background image

3.2. Exploiting variable search window for object detection

At Step1, when $P_{RGB\_channel}$ exceeds the threshold α, it is regarded as object detection case such that it creates an object area of the rectangle based on the minimum and maximum values of X-axis and Y-axis for the object.

$$Object_{Location} = \begin{pmatrix} x_{\max\_\min}(x_1, x_2, ..., x_n) \\ y_{\max\_\min}(y_1, y_2, ..., y_n) \end{pmatrix} \quad (3)$$

From (3), $Object_{Location}$ includes the minimum and maximum values of X-axis and Y-axis, respectively (*i.e.*, x_{min}, x_{max}, y_{min}, y_{max}).

Based on (3), we also compute the rectangle area of the object in (4).

$$Object_{rect} = R((x_{\min}, y_{\min}), (x_{\max}, y_{\min}), (x_{\min}, y_{\max}), (x_{\max}, y_{\max})) \quad (4)$$

In the case that we are tracking an object in real-time, if the object moves in a constant speed or movement patterns, we can easily estimate the movement by the estimation patterns. However, if the speed and movement patterns are varied, it is not so easy to track the object. In addition, it takes long time to compute the whole area of the image. This is a kind of drawbacks for existing methods. In order to overcome this problem, we propose to use the variable search window, which adaptive changes the search area according to the movement direction so that we can improve the speed of object detection by reducing the computational complexity.

We create the variable search window by using (4) and searching area W as depicted in (5).

$$VSW_{rect} = S((x_{\min} - W, y_{\min} - W), (x_{\max} + W, y_{\min} - W), (x_{\min} - W, y_{\max} + W), (x_{\max} + W, y_{\max} + W) \quad (5)$$

Figure 6 indicates the process for generating variable search window and tracking method.

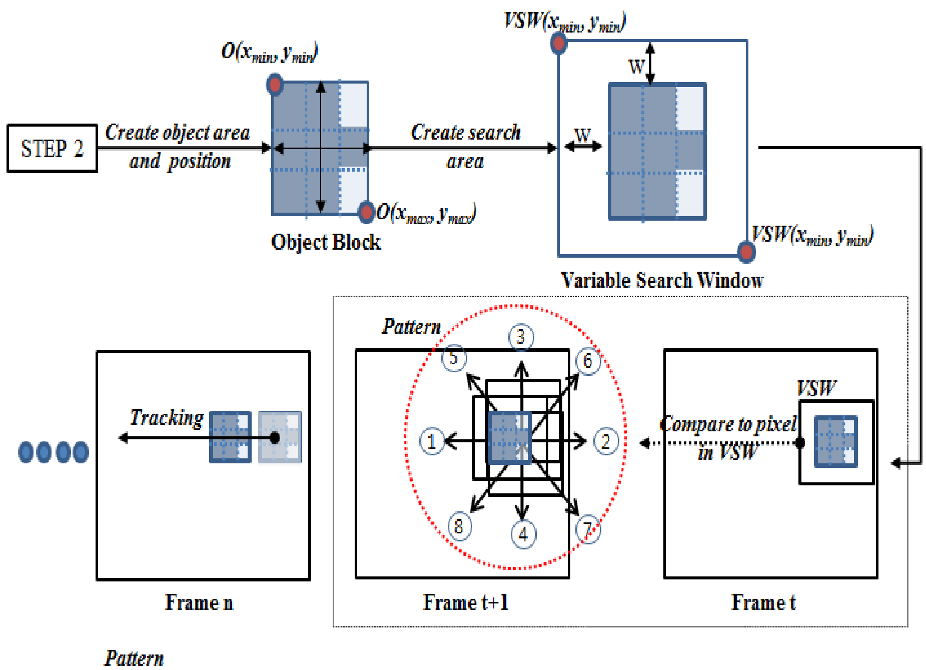
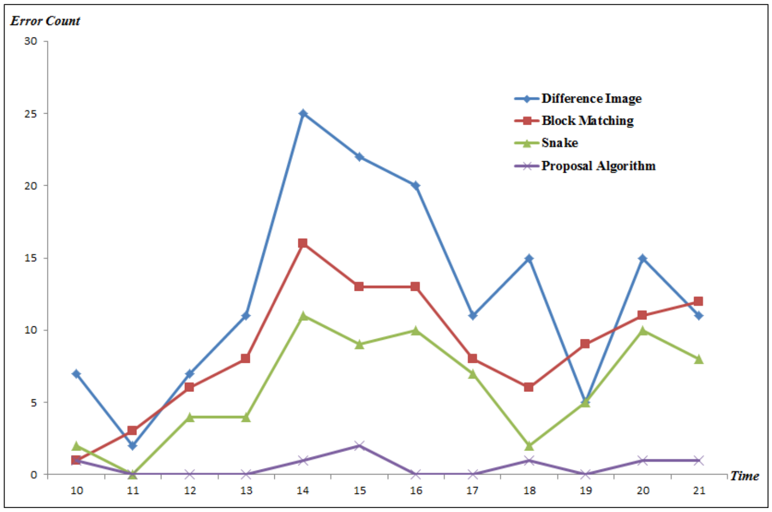


Figure 6. Variable search window generation and tracking process

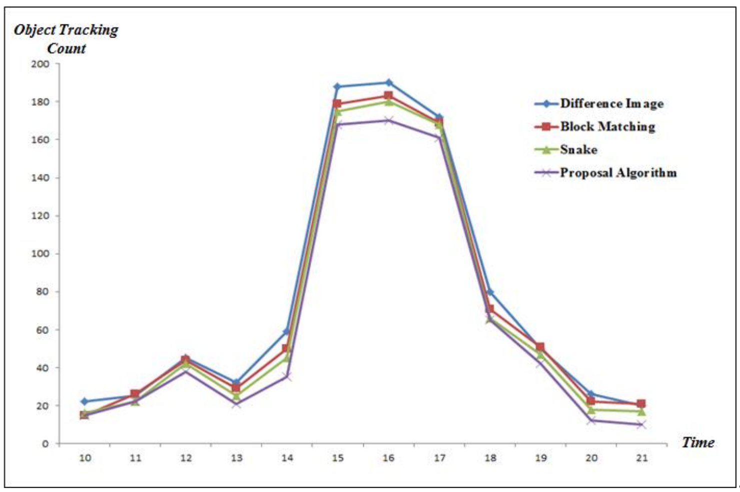
4. Performance Evaluation

To evaluate the performance of our proposed object tracking algorithm, we use both Intel Core2Duo E6300 model and Intel i7 3770K model for lower speed and higher speed computing environment, respectively. For the operating system, Windows XP and Windows 7 are used, respectively.

First, we evaluate our proposed algorithm under the experiment environment where it does not have a lot of people movement. To evaluate our proposed tracking algorithm, we compare it with following three methods: i) difference image method, ii) block matching method, and iii) snake method. We intentionally make a situation that a person or a moving object puts into the experimental environment at a certain time.



(a) **Comparison of object tracking count**



(b) **Comparison of tracking error count**

Figure 7. Object tracking comparison for object tracking count and tracking error count

Figure 7 shows the results for object tracking count and tracking error count, respectively. Note that we evaluate our proposed algorithm in comparison with three other existing methods.

As shown in Figure 7, under the difference image method, we can see a large number of tracking count in the time between 10am to 2pm and 7pm to 9pm. This is due to the fact that this difference image method sensitively reacts with respect to the tiny changes in the environment. In the case that block matching method is used, this block matching method is robust than that of the difference image in terms of the environment changes. However, we found that there exist some tracking errors caused by the object standing in the middle position at 3pm to 5pm or afterimages. Even though the snake algorithm also has the robustness to the environment changes, it also experiences the similar errors of the block matching method at the time from 3pm to 5pm. Different from the previous three methods, our proposed method has the most robustness in respect with the environment changes, especially, our proposed method significantly outperforms three other methods in the afternoon.

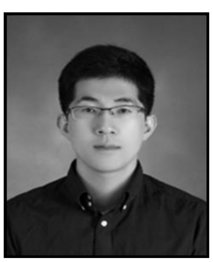
5. Conclusions

The object detection and tracking using the background image is not easy to distinguish between object and noise due to the lighting or environmental changes. However, we can improve the accuracy for detecting the object by updating the background image with respect to the environment changes. Also, detection and tracking for object coming from various angles are also possible by exploiting the method of implementation. Our proposed scheme by using the concept of background changes and adaptive search window outperforms the other object detection and track methods. Our proposed method minimizes the computational time for the navigation area so that our method is suitable for the real-time system. In the future, we will conduct research on the detection and tracking of multiple objects,

References

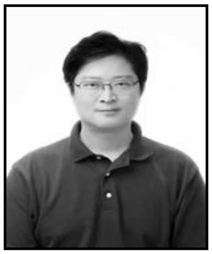
[1] D. Chen and J. Yang, "Robust Object Tracking Via Online Dynamic Spatial Bias Appearance Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, **(2007)**, pp. 2157-2169.
[2] S. Kim, S. Lee, S. Kim and J. Lee, "Object Tracking of Mobile Robot using Moving Color and Shape Information for the aged walking", IJAST, vol. 3, **(2009)**, pp. 59-68.
[3] J. C. Sousaa, H. A. Biswasa, R. Brito and A. Silveirab, "A Multi Objective Approach to solve Capacitated Vehicle Routing Problems with Time Windows Using Mixed Integer Linear Programming", IJAST, vol. 28, **(2011)**, pp. 1-8.
[4] V. K. Singh and R. C. Tripathi, "Fast and Efficient Region Duplication Detection in Digital Images Using Sub-Blocking Method", IJAST, vol. 35, **(2011)**, pp. 93-102.
[5] N. Annabi-Elkadri and A. Hamouda, "The Multiresolution Spectral Analysis for Automatic Detection of Transition Zones", IJAST, vol. 36, **(2011)**, pp. 95-110.
[6] W. -C. Hu, "Adaptive Template Block-Based Block Matching for Object Tracking", IEEE Eighth International Conference on Intelligent Systems Design and Applications (ISDA), vol. 1, **(2008)**, pp. 61-64.
[7] H. Fang, S. -H. Kim and J. W. Jang, "Snake Algorithm for Automatically Tracking Multiple Objects", 18th IEEE International Conference on Image processing (ICIP), **(2011)**, pp. 469-472.
[8] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", Transactions on Pattern Analysis and Machine Intelligence, vol. 25, **(2003)**, pp. 564-577.

Authors



Daewon Jung

He received his B.S. degree in Electronics and Communications Engineering from Kwangwoon University, Seoul, Korea in 2006, and his M.S. and Ph.D. degrees in Information and Communications from Gwangju Institute of Science and Technology, Gwangju, Korea, in 2008 and 2012, respectively. He has been working for the Attached Institute of Electronics and Telecommunications Research Institute since 2012. His research interests include performance evaluation and protocol design for wireless communication systems and distributed protocols in wireless networks.



Changsoo Lee

He received his B.S. degree in Computer Science from Hanseo University, Korea, in 1999, and M.S. and Ph.D. degrees in Computer Science from Soongsil University, Seoul, Korea, in 2002, and 2005, respectively. He is currently a researcher at network security lab in Soongsil University, Seoul, Korea. His research interests include multimedia applications and multimedia security, RFID/USN Solution.