Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance

Ying-Li Tian and Arun Hampapur
IBM T.J. Watson Research Center
PO BOX 704, Yorktown Heights, NY 10598
{yltian,arunh}@us.ibm.com

Abstract

Moving object detection is very important for video surveillance. In many environments, motion maybe either interesting (salient) motion (e.g., a person) or uninteresting motion (e.g., swaying branches.) In this paper, we propose a new real-time algorithm to detect salient motion in complex environments by combining temporal difference imaging and a temporal filtered motion field. We assume that the object with salient motion moves in a consistent direction for a period of time. No prior knowledge about object size and shape is necessary. Compared to background subtraction methods, our method does NOT need to learn the background model from hundreds of images and can handle quick image variations; e.g., a light being turned on or off. The average speed of our method is about 50fps on images at size 160x120 in 1GB Pentium III machines. The effectiveness of the proposed algorithm to robust detect salient motion is demonstrated for a variety of real environments with distracting motions such as lighting changes, swaying branches, rippling water, waterfall, and fountains.

1. Introduction

Detection of moving objects in video streams is known to be a significant and difficult research problem [12]. In many real environments, the motion is caused by both interesting (salient) and uninteresting motion. As in [13], salient motion is here defined as motion from a typical surveillance target (a person or a vehicle) as opposed to other distracting motions such as the scintillation of specularities on water and the swaying of vegetation in the wind. The distracting motions in the complex environments make the problem of motion detection more challenging.

Background subtraction is a conventional and effective approach to detect moving objects in the stationary background. To detect moving objects in a dynamic scene, adaptive background subtraction techniques have been developed [8, 9, 11]. Ren *et al.* [9] proposed a spatial distribution of Gaussians (SDG) model to deal with moving object detection having motion compensation which is only approximately extracted. Their results demonstrated the capability of the system to detect small moving objects with a highly textured background with pan-tilt camera motion. Stauffer *et al.* [11] modeled each pixel as a mixture of Gaussians and used an on-line approximation to update the model. Their system can deal with lighting changes, slow-moving objects, and introducing or removing objects from the scene. Monnet *et al.* [8] proposed a prediction-based online method for the modeling of dynamic scenes. Their approach has been tested on a coast line with ocean waves and a scene with swaying trees. However, they need hundreds of images without moving objects to learn the background model, and the moving object cannot be detected if they move in the same direction as the ocean waves. Recently, some hybrid change detectors are developed which combine temporal difference imaging and adaptive background estimation to detect regions of change [1, 5]. Huwer *et al.* [5] proposed a method of combining a temporal difference method with an adaptive background model subtraction scheme to deal with lighting changes. However, none of these methods can adapt to quick image variations such as a light turning on or off. The major drawbacks of adaptive background subtraction are summarized as following:

- It makes no allowances for stationary objects in the scene that start to move.
- It needs hundreds images to learn the background model.
- It cannot handle quick image variations and large distracting motion.

Motion based methods for detecting salient motion have also been developed [13, 14]. Wildes [13] proposed a measure of motion salience using spatiotemporal filtering and assumes the object is moving with a certain velocity due to the velocity-dependent nature of the spatiotemporal filters. This method didn't work for a slow moving object. Wixson [14] presented a method to detect salient motion by accumulating directionally-consistent flow. They calculated subpixel optical flow and integrated frame-to-frame optical flow over time for each pixel to compute a rough estimate of the total image distance it has moved. On each frame, they update a salient measure that is directly related to the distance over which a point has traveled with a consistent direction. However, their method was very time consuming and have the salience "trails" left by objects.

In this paper, we propose a real-time algorithm to detect salient motion in complex environments by combining temporal difference imaging and temporal filtered optical flow. Similar to [14], we assume that the object with salient motion moves in an approximate consistent direction in a time period. The motion is calculated by the Lucas-Kanada optical flow algorithm. The temporal difference imaging helps to detect slow moving objects, to give better object boundaries, and to speed up the algorithm because the temporal filter of optical flow is only applied to the regions of change which are detected by temporal difference imaging. In that region, for each pixel, the motion is salient motion if the pixel and its neighborhood move in the same direction in a period of time. Compared with [14], our method is faster and more robust. Our method runs in 15fps to 60fps (depending on the size of the region of changes which is detected by temporal differencing) in 1GB Pentium III 1 processor machines. The effectiveness of the proposed algorithm robustly to detect salient motion is demonstrated in Section 3 for a variety of real environments with distracting motion such as lighting changes, swaying branches, rippling water, waterfall, and fountains.

2. Salient Motion Detection Method

As shown in Figure 1, background subtraction does not work well and provides many false positives for the environment with motions not only caused by the objects of interest but also distracting motion such as specularities on water and vegetation in the wind. In this sequence, when a person walks in front of the oscillated branches in the strong wind, both the person and the moving leaves are detected as foreground by the background subtraction methods. Figure 1(b) shows an example of the method which is described in paper [4].

Generally, interesting moving objects tend to have consistent moving directions over time. In order to detect the objects with salient motion, we perform five steps: (1) the temporal difference of subsequent images are subtracted to get the region of change; (2) the frame-to-frame optical flow is calculated; (3) the temporal filter is applied to the region of changes which is detected by the first step to detect the pixels continually move in the same direction (either in X-component or in Y-component); (4) the pixels which continually move in the same direction are used as seed pixels for the X-component and Y-component of optical flow respectively and then grows the pixels to form larger region if its *NXN* neighborhood move in the same direction; (5) the objects with salient motion are finally detected by combining all the temporal difference imaging, temporal filtered motion, and region information.

(a) Original Image (b) Background Subtraction

Figure 1: In this sequence, a person walks in front of the oscillating branches in a strong wind. Background subtraction [4] does not work for sequences with *distracting motion,* both the person and the moving leaves are detected as foreground. (a) original image, (b) result of background subtraction [4]. The green regions present the detected foreground.

2.1 Temporal Difference

Temporal difference is the simplest method to extract moving objects and adapt to dynamic environments, but cannot detect the entire shape of a moving object with uniform intensity. In this paper, subsequent images $I(x, y, t)$ and $I(x, y, t+1)$ are subtracted and the difference image is thresholded to get the region of changes. The threshold T_d can be derived from image statistics. In order to detect cases of slow motion or temporally stopped objects, a weighted accumulation with a fixed weight for the new observation is used to compute the temporal difference image $I_{difference}(x, y, t)$ as shown in following equations:

$$I_{difference}(x, y, t+1) = \begin{cases} 1, if\ (I_{accum}(x, y, t+1) > T_d) \\ 0, otherwise \end{cases}.$$

And

$$I_{accum}(x, y, t+1) = (1 - W_{accum}) I_{accum}(x, y, t)$$
$$+ W_{accum} \left| I(x, y, t+1) - I(x, y, t) \right|,$$

where W_{accum} is the accumulation parameter which describes the temporal range for accumulation of difference images. $I_{accum}(x, y, t-1)$ is initialized to an empty image. In our system, we set T_d = 15 and W_{accum} = 0.5 for all the results.

A rectangular region of changes in the image now is available. The size of the region can change from (0, 0) to (*w, h*), where *w* and *h* is the width and height of the input image respectively. The size of the original image is 320x240 pixels. In order to speed up the whole system, we first downsample it to 160x120 pixels. Instead of using the whole image, this region of changes will be used later to apply the temporal filter for motion.

2.2 Motion Extraction

To extract a 2D motion field, we employ a modified Lucas-Kanade method to compute optical flow in our system because of its accuracy and efficiency. Barron *et al.* [2] compared the accuracy of different optical flow techniques on both real and synthetic image sequences, they found that the most reliable was the first-order, local differential method of Lucas and Kanade. Liu *et al.* [6] studied the accuracy and the efficiency trade-offs in different optical flow algorithms. They focused on the motion algorithms implementations in real world tasks. Their results showed that Lucas-Kanade method is pretty fast. Galvin *et al.* [3] evaluated eight optical flow algorithms. They found that a modified version of Lucas-Kanade algorithm has superior performance but sparse flow maps. The Lucas-Kanade method consistently produces accurate depth maps, has a low computational cost, and good noise tolerance. In our system, to identify an incorrect flow between two frames, a forwards-backwards checking is performed to examine whether the flow fields map to the same points.

The Lucas-Kanade algorithm assumed that intensity values of any given region do not change but merely shift from one position to another. Consider a moving object in the image, the displacement of the object over a *nxn* window from time *t* to *t+1* is *d*. The above assumption is represented by:

$$I_{t+1}(x+d)-I_t(x)=0$$

We wish to find the translation *d* of this window by minimizing a cost function *E* defined as:

$$E=\sum_{x\in R}[I_{t+1}(x+d)-I_t(x)]^2 ,$$

and the minimization for finding the translation *d* can be calculated in iterations:

$$d_{n+1}=d_n+\left\{\sum_{x\in R}\left(\frac{\partial I}{\partial x}\right)^T \Big|_{x+d_n}[I_t(x)-I_{t+1}(x)]\right\}\left[\sum_{x\in R}\left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial x}\right)^T \Big|_{x+d_n}\right]^{-1},$$

where d_0, the initial estimate, can be taken as zero if only small displacements are involved. See paper [9] for details of the Lucas & Kanade algorithm. In our system, the window size is 13x13.

2.3 Temporal Filter

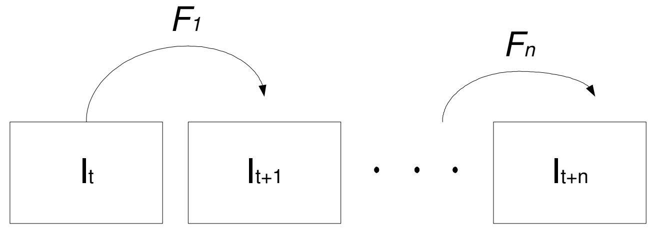

Figure 2: The temporal filter is applied to the calculated optical flow of frames in a period time [*t, t+n*].

As shown in Figure 2, the optical flow of frames $I_t, I_{t+1}, \cdots I_{t+n}$ in a period time [*t, t+n*] are represented as $F_1, F_2, \cdots F_n$. The *X*-component is $F_{1,x}, F_{2,x}, \cdots F_{n,x}$ and the *Y*-component is $F_{1,y}, F_{2,y}, \cdots F_{n,y}$ respectively. We assume that the object with salient motion moves in a consistent direction in a period time on either the *X*-component or the *Y*-component. It means that the optical flow of the region with salient motion in the time period [*t, t+n*] should be in same direction. In our system, we deal with the *X*-component and *Y*-component of optical flow separately.

For each pixel in the region of changes which is detected by temporal differencing, we first build a chain of optical flow in the time period [*t, t+n*]. The pixel (x, y) in frame I_t moves to $(x+dx_1, y+dy_1)$ in frame I_{t+1}, where dx_1 and dy_1 can be obtained from $F_{1,x}$ and $F_{1,y}$. If $dx_1 > 0$, we define the moving direction of this pixel in X-component is positive. Otherwise, it is moving in negative direction. In the frame I_{t+n} of this chain, the pixel is in position $(x+dx_1+dx_2+\cdots+dx_n, y+dy_1+dy_2+\cdots+dy_n)$. Then, we count the number P_{iX} (the *X*-component of optical flow is positive) and the number N_{iX} (the *X*-component of the optical flow is negative) in the time period [*t, t+n*] by using $F_{1,x}, F_{2,x}, \cdots F_{n,x}$, where $i \subset [1,n]$. The algorithm to detect whether a pixel belongs to an object with salient motion is described as following:

(1) $P_{iX} = 0,\ and\ N_{iX} = 0\ if\ i = 1$

$$(2)\ P_{iX} = \begin{cases} P_{(i-1)X} + 1, if\ (F_{i,X} > 0) \\ P_{(i-1)X}, otherwise \end{cases}\ and$$

$$N_{iX} = \begin{cases} N_{(i-1)X} + 1, if\ (F_{i,X} < 0) \\ N_{(i-1)X}, otherwise \end{cases}$$

(3) The pixel belongs the object with salient motion if $P_{nX} \geq T_1$ *or* $N_{nX} \geq T_1$.

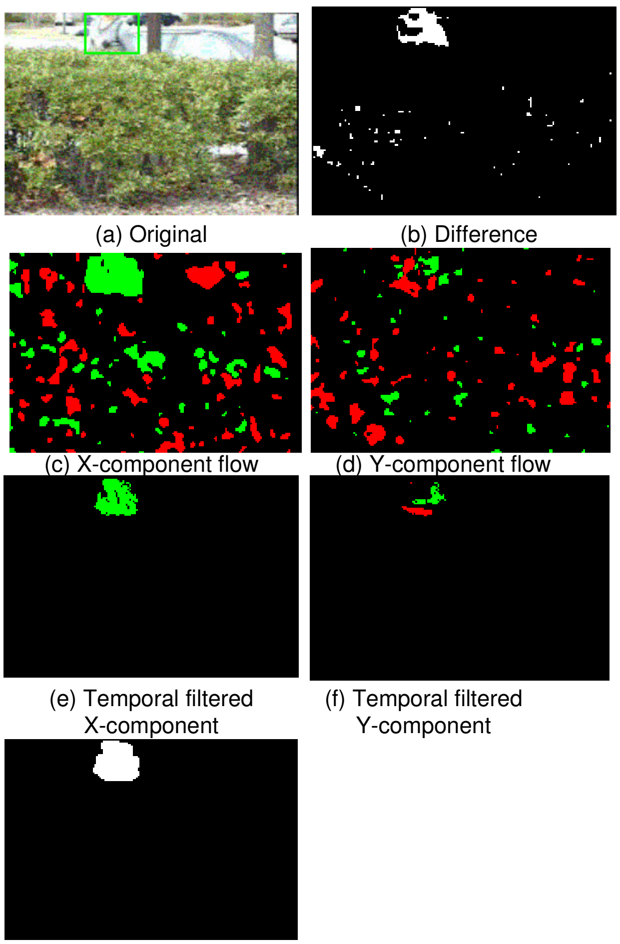

Figure 3: An example of the salient motion detection. (a) the original image, a green rectangular region shows the object with salient motion for display purpose only, (b) the accumulated difference image, (c) the X-component of optical flow, (d) the Y-component of optical flow, (e) the X-component of flow after temporal filter, (f) the Y-component of flow after temporal filter, (g) the final detected object with salient motion.

For the *Y*-component, we deal with it in the same way. In our system, we set $n = 10,\ T_1 = n - n/3$. The same parameter settings are used for both *X*-component and *Y*-component and all the experimental sequences in our system. Figure 3 shows an example of the procedure to detect a person walking behind wildly swaying leaves based on the temporal difference imaging, the *X*-component and *Y*-component of optical flow, and the temporal filtered flow components. Figure 3(a) is one frame of the sequence. A rectangle is added to show the walking person for display purposes. The accumulated temporal difference imaging is shown in Figure 3(b). Figure 3(c) and 3(d) show the X-component and Y-component of the optical flow respectively, where the red and green color indicate the directions of the flow. The X-component and Y-component of temporal filtered flow are shown in Figure 3(e) and 3(f). Because the person is walking from right to left, the X-component of temporal filtered flow dominates the salient motion detection in this sequence. The final detected region with salient motion is shown in Figure 3(g).

2.4 Region Growing

Comparing Figure 3(c) with 3(e), we notice that all the distracting motion has been filtered by the temporal motion filter. Also the region of salient motion after the temporal motion filter is smaller than that in the original flow. To avoid the problem of splitting one object into several objects, the pixels which are detected by the temporal motion filter are used as seed pixels for the X-component and Y-component of original optical flow and then the algorithm grows the pixels to form a larger region if its *NXN* neighborhood moving in the same direction. In our system, we use *3X3* neighborhood for region growing.

2.5 Multi-sources Fusion

The salient motion image ($I_{salient}(x, y, t)$) is obtained by combining the image of temporal difference ($I_{differnce}(x, y, t)$), temporal filtered image ($I_{X-temporal}(x, y, t)$ and $I_{Y-temporal}(x, y, t)$) and the region information of the motion. The output salient motion image is obtained by following equations:

$$I_{salient}(x, y, t) = I_{difference}(x, y, t) \cap (I_{X-temporal}(x, y, t) \cup I_{Y-temporal}(x, y, t))$$

3. Experimental Results

In this section, the effectiveness of the proposed algorithm to robust detect salient motion is demonstrated for a variety of real environments with distracting motions such as waterfall, fountain, rippling water, swaying branches, and large lighting changes. Notice that **same parameters** were used for **all sequences.**

Figure 4 illustrates the algorithm on a video sequence in which a person walks around while a fountain and a waterfall are observed in the background. The whole sequence includes 882 frames. In Figure 4, the left-top image shows one frame of the original image. A green rectangular region was added to show the object with salient motion for display purpose only. The right-top image shows the difference image. The bottom-left image shows the X-component of the flow and the final detected region with salient motion is shown in the bottom-right. The result shows that the person is well detected and the waterfall and the fountain are eliminated. The average speed for this sequence is about 40fps in 1GB Pentium III machines.

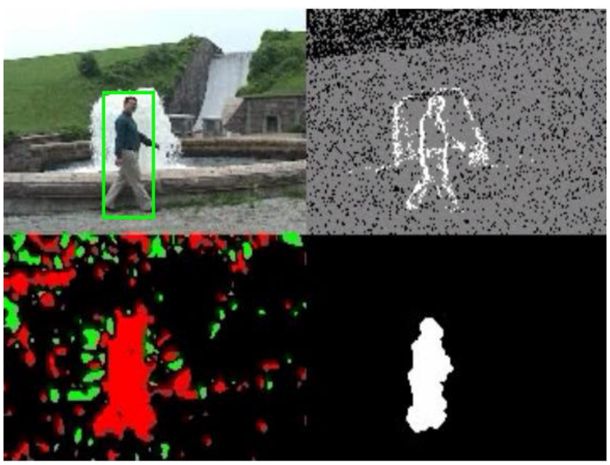

Figure 4: A person walks around while fountain and waterfall are in background. Left-top: original image, a green rectangular region shows the object with salient motion for display purpose only; right-top: difference image; bottom-left: X-component of the flow; bottom-right: final detected region with salient motion.

Figure 5, 6, and 7 provide more examples of the algorithm on five other sequences. In Figure 5, a person walks around while there are specularities on water in the background. The bottom-right image shows the well detected walking person. The average speed for this sequence runs 61fps in a 1GB Pentium III machine.

In Figure 6, a car (the one in the green rectangular) moves from left to right while wildly swaying leaves are in the foreground. The average speed for this sequence runs only 11Hz because the temporal difference region is almost same as the whole image.

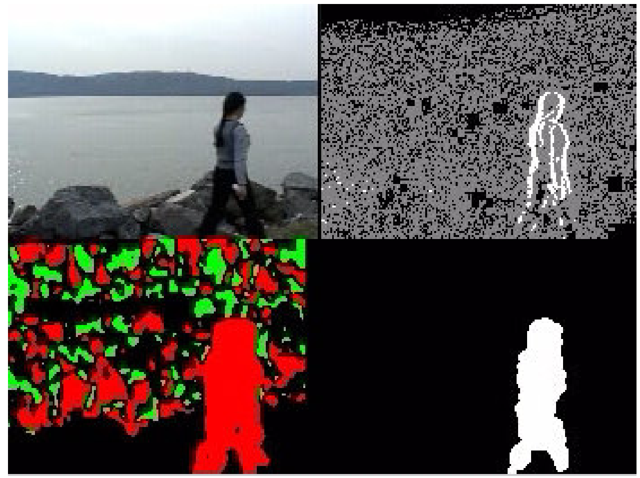

Figure 5: A person walks around while there are specularities on water in background. Left-top: original image; right-top: difference image; bottom-left: X-component of the flow; bottom-right: final detected region with salient motion.

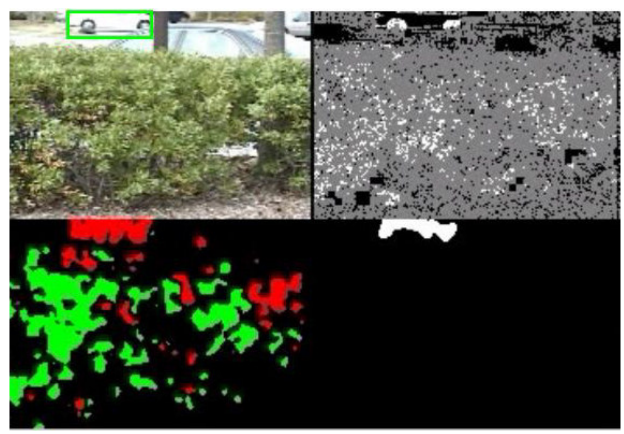

Figure 6: A car is moving while wildly swaying leaves are in foreground. Left-top: original image, a green rectangular region shows the object with salient motion for display purpose only; right-top: difference image; bottom-left: X-component of the flow; bottom-right: final detected region with salient motion.

Figure 7 demonstrates the capability of our algorithm to handle lighting changes. A person walks from lower-right to upper-left to turn off the lighting and then walk back. The original images and detected moving person are shown when the light is on and off respectively. The speed for this sequence runs about 50fps.

4. Discussions and Conclusions

We presented a new method for the detection of salient motion in complex background with distracting motion for real-time surveillance applications. We have assumed that the object with salient motion moves in a consistent direction in a period time. No prior knowledge about object size and shape is necessary. Our method combined temporal differencing with temporal filtered optical flow to robustly detect object with salient motion in real-time. The effectiveness of proposed algorithm to robust detect salient motion was demonstrated for a variety of real environments with distracting motions such as lighting changes, swaying branches, rippling water, waterfall, and fountains.

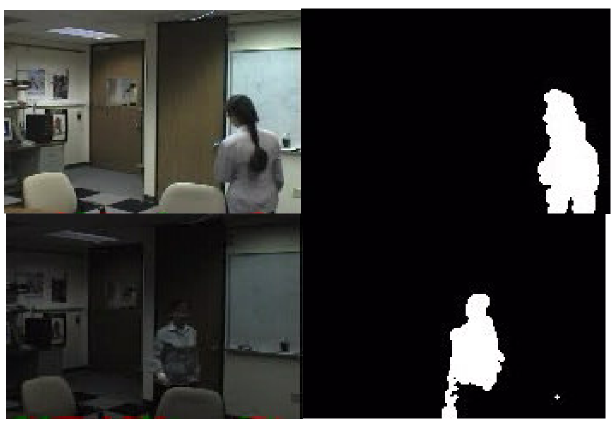

Figure 7: A person walks when turn on or off the light. Left-top: original image when light is on; right-top: final detected region with salient motion for the frame shown in the left; bottom-left: original image when light is off; bottom-right: final detected region with salient motion for the frame shown in the left.

The algorithm has some limitations. First, an object that moves in different direction such as moving in zigzag cannot be detected because we assume that the object with salient motion moves in a consistent direction in a period time. Second, if the object stops a while, we will lose it. But it can be detected when it stars to move again.

5. References

[1] N. Amamoto and K. Matsumoto, "Obstruction Detector by Environmental Adaptive Background Image Updating," In ERTICO, editor, 4th World Congress on Intelligent Transport Systems, No. 4, pp1-7, Berlin, Oct. 1997. Traffic Technology International.

[2] J. Barron, D. Fleet, and S. Beauchemin, "Performance of Optical Flow Techniques." International Journal of Computer Vision, Vol. 12, No. 1, 1994, 42-77.

[3] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, "Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms." In Proc. Of the 9th British Machine Vision Conference (BMVC'98), Vol. 1, p195-204, Sept. 1998.

[4] T. Horprasert, D. Harwood, and L. Davis. A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection. Proceedings of IEEE Frame-Rate Workshop, Kerkyra, Greece, 1999.

[5] S. Huwer and H. Niemann, "Adaptive Change Detection for Real-time Surveillance applications," In the Proc. of the 3rd IEEE Workshop on Visual Surveillance, 2000, pp.37-45.

[6] H. Liu, T. Hong, M. Herman and R. Chellappa, "Accuracy vs. Efficiency Trade-offs in Optical Flow Algorithms", In the Proc. of Europe Conf. Of Computer Vision, 1996.

[7] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application in Stereo Vision," The 7th International Joint Conference on Artificial Intelligence, 1981, 674-679.

[8] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background Modeling and Subtraction of Dynamic Scenes", In Proc. of International Conference on Computer Vision (ICCV), 2003, Pages 1305–1312.

[9] Y. Ren, C. Chua, and Y. Ho, "Motion Detection with Non-stationary Background," In Proc. Of 11th Int'l Conf. Image Analysis and Processing, 2001, 78-83.

[10] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background Subtraction Based on Cooccurrence of Image Variations", In Proc. of international conference on computer vision and pattern recognition (CVPR), 2003.

[11] C. Stauffer and W.E.L. Grimson, "Adaptive Background mixture Models for Real-time Tracking", CVPR99, June, 1999.

[12] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principles and practice of background Maintenance." In Proc. International Conference on Computer Vision, 1999, 255-261.

[13] R. P. Wildes, "A Measure of Motion Salience for Surveillance Applications," In Proc. Of IEEE International Conference on Image Processing, p183-187, 1998.

[14] L. Wixson, "Detecting Salient Motion by Accumulating Directionally Flow," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22. No. 8. pp774-779, August, 2000.