Feature-Based Object Tracking Using Spatial Matching of Differential Directional-Edge Images

Sihwan Kim
Department of Electronic Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
Email: sihwan@else.k.u-tokyo.ac.jp

Tadashi Shibata
Department of Frontier Informatics
The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa-shi, Chiba, 277-8561, Japan
Email: shibata@ee.t.u-tokyo.ac.jp

***Abstract*—A robust object tracking system has been developed based on the directional edge information extracted from a video sequence. By introducing the concept of the differential directional-edge image (DDEI), a map of edge flags produced from the difference of two consecutive edge images, the information from background has been effectively removed. As a result, the algorithm is robust against the influence of cluttered background, as well as illumination change and speed variation. In addition, in order to enhance the robustness against the shape transformation and partial occlusion occurring during the object movement, adaptive algorithms have been developed and integrated in the system. The basic operation of the algorithm is the spatial matching of differential directional-edge histograms, which is compatible to direct VLSI hardware implementation, making it possible to build a real-time responding system. The performance of the proposed object tracking system has been demonstrated by computer simulation using video sequences including a variety of disturbances.**

I. Introduction

Object tracking is one of the most important subjects in video processing. It has a wide range of applications including traffic surveillance, vehicle navigation, gesture understanding, security camera systems, and so forth. In biological systems, visual object tracking is quite essential and very robust and sophisticated systems are established in animals. This is because moving objects can be either their prey, an attacking enemy or a potential mate, it is very important for an animal's survival to accurately find and track moving objects and subsequently understand their behavior. Therefore, developing electronic system somehow mimicking the biological principle would be quite important to build robust object tracking systems.

Provided an object in motion is localized in a scene, tracking can be carried out by searching for the object's image in consecutive frames of a video sequence. However, such an approach poses several difficulties. First of all, the object image needs be separated from the background. Otherwise, simple pattern matching can not work due to the disturbances arising from the cluttered background. Illumination change during the tracking can also confuse correct identification of the target image. In addition, since the algorithms for motion tracking are usually computationally very expensive, it is desirable to develop algorithms compatible to direct VLSI hardware implementation in order to achieve real-time performances [1],[2].

Several algorithms have been proposed so far for object tracking. The mean-shift method [3]-[5] utilizes the color histogram to separate the object from the background. Since the color histogram is a viewpoint-invariant feature, it is stable against changes in scale, pose and shape of the target object. However, the technique is vulnerable to variation in illumination, whereas the human vision system is capable of tracking objects in gray scale without difficulty. In object tracking by active contours ("snakes") [6]-[8], objects are separated from the background using the energy minimizing technique, demonstrating a good performance for non-rigid objects. However, it needs the prior information, i.e., the object contour has to be defined or trained before tracking, which is not always available in practice. Also, when the object contour is complicated, the initial target contour definition is often laborious and difficult to generate automatically.

It is well known that the visual perception in animals relies heavily on edge information in various orientations to recognize not only the static image of an object but also its motion. Being inspired by such a biological principle, a robust image recognition system has been developed based on the image representation algorithm using directional-edges extracted from an input image [9],[10].

Then, the purpose of the present work is to develop a robust object tracking system utilizing the directional edge information to represent the shape of the object under tracking. In order to erase the information from the background, the concept of a differential directional-edge image has been introduced. A differential directional-edge image is produced from edge maps of two consecutive images in a video sequence by taking exclusive OR of edge flags at every pixel site. As a result, edge flags in the static background are effectively erased and edges are only retained primarily at locations of moving objects. Since such an image representation is based on edge information, it is very robust against illumination change as already verified in [11]. We have also developed adaptive algorithms for differential edge image generation as well as motion vector detection. As a result, the present object tracking system have shown robust performances against shape transformation of objects during motion, partial occlusion, speed variation as well as the presence of confusing background sceneries. The key operation of the system is the shift

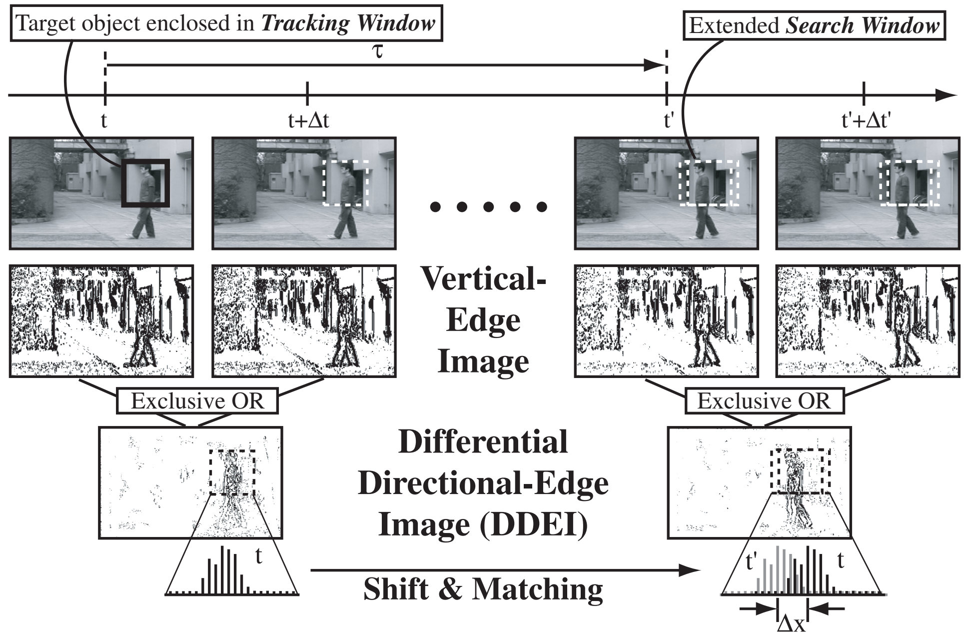


Fig. 1. Object tracking algorithm in x direction using four frames of images taken from a video sequence. The same algorithm is employed for tracking in y direction.

and matching of differential edge histograms to determine the motion. Specialized VLSI hardware systems for this operation have already been developed in [1],[2], the present algorithm is compatible to direct VLSI hardware implementation to achieve a real-time performance.

The paper is organized as follows. After presenting the total system organization and the object tracking algorithm in Section 2, experimental results are demonstrated in Section 3. Finally conclusions are given in Section 4.

II. System Organization

The object tracking algorithm employed in the present system is illustrated in Fig. 1. Firstly, four grayscale images are taken from a video sequence. We assume that the object being tracked at time t is enclosed in the tracking window. Then the search window is determined by extending the tracking window by ± 8 pixels at t', $t'+\Delta t'$. At the very beginning of the tracking session, the initial location of the tracking window is determined by some other means and is specified as the initial condition. Then the system starts to track the object and continues to track it autonomously.

Directional-edge flags are obtained by binarizing grayscale images obtained with appropriate filtering kernels (see an example shown in Fig. 2) in the tracking window as well as in the search window. In the present system, x-direction tracking is performed using vertical edges since they are sensitive to horizontal motion, while horizontal edges are used for y-direction tracking. From directional-edge images at t and $t+\Delta t$, the differential directional-edge image (DDEI) at t is generated by taking XOR (exclusive OR) of edge flags at every pixel site. At this time, Δt is determined adaptively as it is explained in section 2.1 and 2.2. The differential directional-edge image (DDEI) at t' is generated by the same procedure. If we assume there is no movement in the camera position, DDEI edge flags in the static background are effectively removed and edges are only retained primarily at moving object locations. To calculate the x component of a motion vector (Δx), data of DDEI are projected onto horizontal axes. Then the shift and match of the projected data of DDEI at t and t' yields the motion vector of the object as the minimum in the matching residue. By this processing, the motion vector at t' is obtained. t' starts from $t+\Delta t$, and increases as the frame advances. If Δx reaches ± 4 pixels (half the search range of ± 8 pixels), then the location of the tracking window is shifted accordingly. When the tracking window is shifted, a new DDEI of the target is generated from the window and utilized as the template for tracking. Since the tracking template is regenerated each time, the algorithm can track an object even if it changes its shape or partial occlusion occurs.

In the present system, there are two threshold values that are important to make the system adaptive to environmental change. First is the threshold used in edge detection from the original image, denoted Th_edge. The second threshold is used for generating differential directional edge images, Th_DDEI. In the next section, the two thresholds are explained.

A. Directional-Edge Detection and Differential Edge Image Generation

Detection of directional-edges and generation of a differential edge image are illustrated in Fig. 2. Edge filtering is carried out at every pixel site in the tracking window using a 5×5-

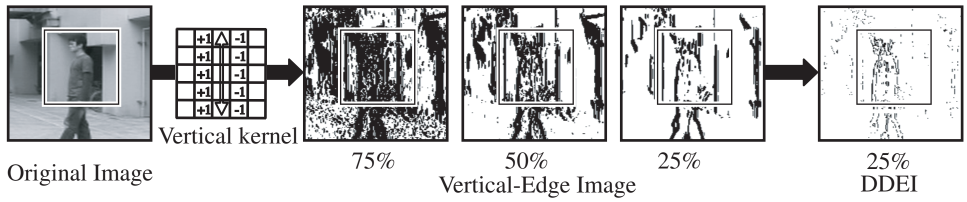


Fig. 2. Thresholding for the vertical edge detection in the tracking window yielding the number of edge flags of 75%, 50%, and 25% of the total number of pixels in the window. DDEI at 25% is also shown on the right.

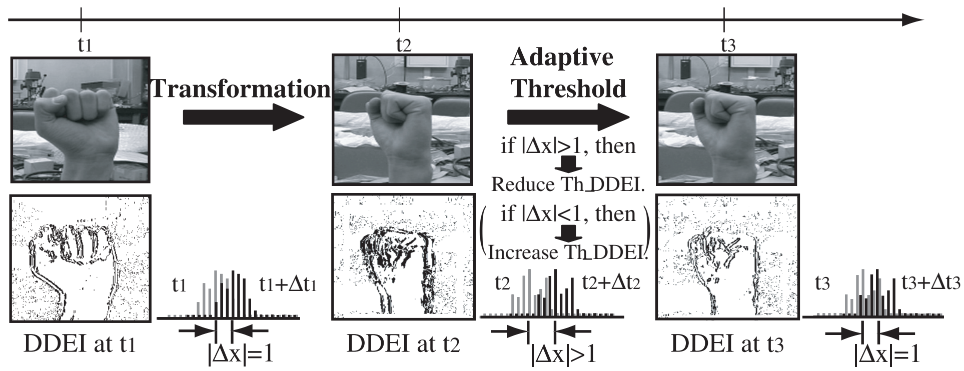


Fig. 3. Adaptive threshold of differential image by Δx depending on transformation, partial occlusion.

pixel kernel. The absolute value after filtering represents the directional-edge intensity at each pixel. Th_edge is determined by observing the total number of edge flags detected in the tracking window. The vertical edge images produced with three different thresholds are shown in Fig. 2, where the number of edge flags are 75%, 50%, 25% of the total number of pixel sites in the window. In the present system, 25% was employed as the condition for Th_edge. With this threshold, the essential features from the original image are mostly retained in the vertical edge image.

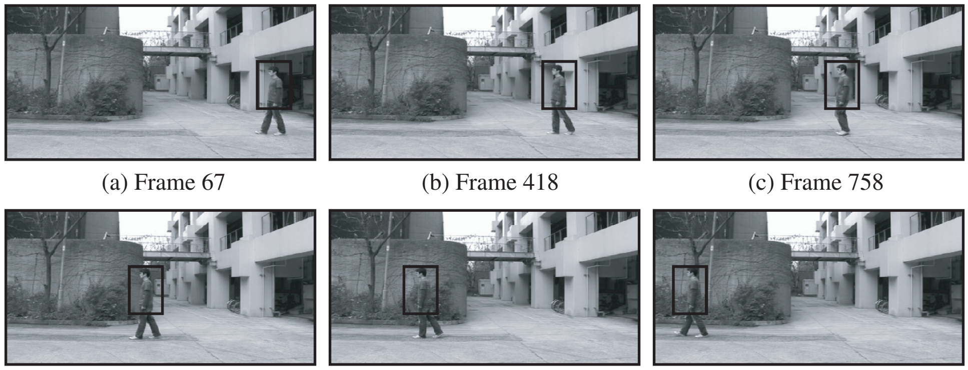
From the two directional-edge images at t and t+Δt, DDEI is generated by taking XOR as already explained. In generating DDEI, the number of edge flags after XOR is counted in every frame. And when the edge count reaches a predetermined value, i.e., Th_DDEI, the edge flag map at the moment is accepted as the DDEI at t. In the case of a target window size of 200×200 pixels (as shown in Fig. 2), 5% of the total number of pixels in the window ($2,000$ pixels) was adopted as the threshold. Initially, Th_DDEI is set in this way, and then Th_DDEI is controlled by feedback automatically. We explain it further in detail in the next section.

B. Adaptation to Shape Transformation and Partial Occlusion

The procedure to determine Th_DDEI adaptively in the presence of transformation is illustrated in Fig. 3. At first, let's consider the case where an object is moving at a constant speed without shape transformation. Then we can track the object with constant values of Th_DDEI and Δt, and the resultant Δx would be also a constant. Now we consider the case where a gradual shape transformation occurs during the constant speed movement of an object. If we assume the object's edge flags are reduced by transformation, the edge flag count after taking XOR would also reduce. In such a case, the edge flag count after XOR does not reach Th_DDEI when the system waits the time interval of Δt. This is because Th_DDEI was determined in the previous time slot where the edge flag count is larger than in the present time slot. As a result, the value of Δt increases, and Δx also increases. In order to keep the value of Δx at a constant value, we have introduced an adaptive algorithm as shown in the Fig. 3. The constant value of $|\Delta x|$ was set to 1 pixel to achieve an accurate object tracking. Namely, if $|\Delta x|$ is larger than 1 pixel in the present time slot, then the threshold Th_DDEI is reduced in the next time slot. This reduces Δt and makes $|\Delta x|$ approach the constant value of 1 pixel. If $|\Delta x|$ is smaller than 1 pixel, then Th_DDEI is increased. Since the partial occlusion of an object can be regarded as a kind of shape transformation, the problem can also be resolved by this adaptive algorithm.

III. Experimental Results

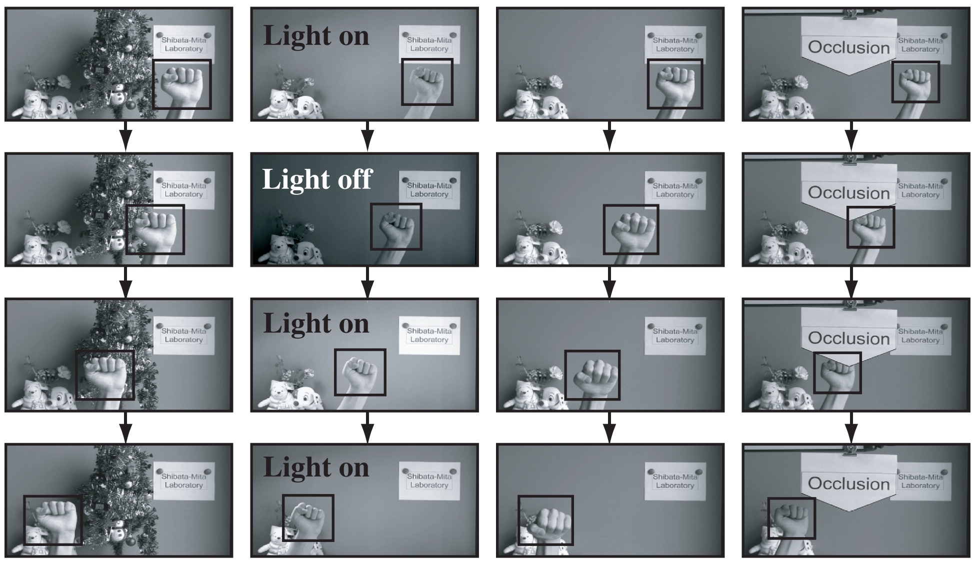
In order to evaluate the performance of the proposed system, computer simulation was carried out for a variety of video sequences. Samples of a person or a hand moving in a video sequence taken with a video camera were used for tracking experiment. Because the frame rate of an ordinary video camera is 30 frame/sec, which is too slow for the present algorithm to work, a target object in a scene was made to move very slowly. However, in CMOS image sensors with focal plane processing functions, it is practical to assume such processing can be carried out at a frame rate of 500-1000 frame/sec. The tracking window enclosing the target object was set by hand in the initial setting. The initial value of



(a) Frame 67 (b) Frame 418 (c) Frame 758

(d) Frame 1112 (e) Frame 1400 (f) Frame 1668

Fig. 4. Tracking the upper part of the body of a walking person in a scene with complicated background.



(a) Busy background (b) Illumination change (c) Transformation (d) Partial occlusion

Fig. 5. Tracking of a hand image under various disturbing conditions : (a) busy background, (b) illumination change, (c) shape transformation, (d) partial occlusion.

Th_DDEI is determined as 5% of the total pixel counts in the tracking window.

At first, we tested the sample of a person walking on a street. The tracking results of the upper part of the body are shown in Fig. 4. In this experiment, the size of the tracking window was set at 78×108 $(8,424)$ pixels. Although the background is very complicated with pillars of a building (Fig. 2(b)) and leaves moving in wind (Fig. 2(e)), the object is being tracked correctly.

The robustness of the system against a variety of disturbing conditions is demonstrated in Fig. 5. In Fig. 5(a), it is seen that the hand passing through the busy background of a decorated tree from left to right is correctly tracked. In this sequence, the tracking window size was 182×154 $(28,028)$ pixels. In Fig. 5(b), the light was turned on and off during the sequence. The tracking window size was 158×144$(22,752)$ pixels in this sequence. In Fig. 5(c), the shape of the hand was changed during movement. The tracking window size was 176×154 $(27,104)$ pixels. In Fig. 5(d), the hand was hidden partially behind the paper. The size of the tracking window was 148×130 $(19,240)$ pixels. Under all these disturbing conditions, the tracking was carried out successfully.

Fig. 6 shows the tracking results of an object moving with speed variation. The size of tracking window was 172×154 $(26,488)$ pixels. The calculated speed of the object is shown in Fig. 6(a). Despite the speed variation range as large as 46 times between the slowest and the fastest, the location of the object is correctly detected.

Finally, the performance of the system was evaluated for an object moving in two-dimensions with all disturbing conditions included. The size of the tracking window in this case was 168×144 $(24,192)$ pixels. The results are shown in Fig. 7, also demonstrating a successful tracking.

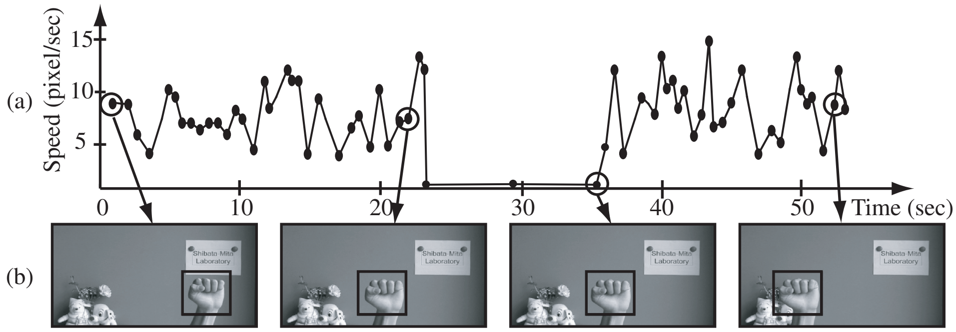


Fig. 6. Object tracking under speed variation. (a) Object speed as a function of time. (b) Results showing successful tracking.

Fig. 7. Tracking of an object in two dimensional motion including all disturbing options : busy background, object transformation, illumination variation, partial occlusion and speed variation. The object is accurately located in all trial scenes.

IV. Conclusion

A robust object tracking system utilizing the differential directional-edge image (DDEI) has been developed. DDEI is robust against the influence of background, illumination change, speed variation. In addition, in order to enhance the robustness against the shape transformation and partial occlusion, the adaptive algorithms have been employed. As a result, the efficiency of proposed object tracking system under a variety of disturbing conditions is proved by computer simulation.

Acknowledgment

The authors would like to thank Hitoshi Hayakawa of The University of Tokyo for his valuable discussions in developing the object tracking algorithm.

References

[1] T. Nakai, T. Shibata, T. Yamashita and T. Ohmi, “Neuron-MOS Parallel Search Hardware for Real-Time Signal Processing,” Analog Integrated Circuits and Signal Processing, Vol. 21, No. 2, pp. 173-191, 1999.

[2] H. Kimura and T. Shibata, “A Simple-Architecture Motion-Detection Analog VLSI Based on Quasi-Two-Dimensional Hardware Algorithm,” Analog Integrated Circuits and Signal Processing, Vol. 39, No. 3, pp. 225-235, 2004.

[3] D. Comaniciu and V. Ramesh, “Real-Time Tracking of Non-Rigid Objects using Mean Shift,” Proc. Computer Vision and Pattern Recognition, Vol. 2, pp. 142-149, 2000.

[4] D. Comaniciu, V. Ramesh and P. Meer, “Kernel-based object tracking,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, pp. 564-577, 2003.

[5] N. Peng and J. Yang, “Mean-Shift Blob Tracking with Kernel-Color Distribution Estimate and Adaptive Model Update Criterion,” Lecture Notes in Computer Science, Vol. 3247, pp. 83-93, 2004.

[6] M. Kass, A. Witkin and D. Terzopoulos, “Snakes: Active contour models,” International Journal of Computer Vision, Vol.1, No. 4, pp. 321-331, 1988.

[7] A. Blake, R. Curwen and A. Zisserman, “A framework for spatiotemporal control in the tracking of visual contours,” International Journal of Computer Vision, Vol.11, No.2, pp. 127-145, 1993.

[8] M. Yokoyama and T. Poggio, “A contour-based moving object detection and tracking,” IEEE Proc. Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 271-276, 2005.

[9] M. Yagi, T. Shibata, “An Image Representation Algorithm Compatible with Neural-Associative-Processor-Based Hardware Recognition Systems,” IEEE Trans. Neural Networks, Vol. 14, No. 5, pp. 1144-1161, 2003.

[10] Y. Suzuki and T. Shibata, “Multiple-Clue Face Detection Algorithm Using Edge-Based Feature Vectors,” IEEE Proc. International Conference on Acoustics, Speech, and Signal Processing, Vol.5, pp. 737-740, 2004.

[11] Y. Suzuki and T. Shibata, “Illumination-Invariant Face Identification Using Edge-Based Feature Vectors in Pseudo-2D Hidden Markov Models,” Proc. European Signal Processing Conference, 2006.