Optical Flow Measurement Based on Boolean Edge Detection and Hough Transform

Min-hyuk Chang, Il-jung Kim, and Jong-an Park

Abstract: The problem of tracking moving objects in a video stream is discussed in this paper. We discussed the popular technique of optical flow for moving object detection. Optical flow finds the velocity vectors at each pixel in the entire video scene. However, optical flow based methods require complex computations and are sensitive to noise. In this paper, we proposed a new method based on the Hough transform and on voting accumulation for improving the accuracy and reducing the computation time. Further, we applied the Boolean based edge detector for edge detection. Edge detection and segmentation are used to extract the moving objects in the image sequences and reduce the computation time of the CHT. The Boolean based edge detector provides accurate and very thin edges. The difference of the two edge maps with thin edges gives better localization of moving objects. The simulation results show that the proposed method improves the accuracy of finding the optical flow vectors and more accurately extracts moving objects' information. The process of edge detection and segmentation accurately find the location and areas of the real moving objects, and hence extracting moving information is very easy and accurate. The Combinatorial Hough Transform and voting accumulation based optical flow measures optical flow vectors accurately. The direction of moving objects is also accurately measured.

Keywords: Optical flow, Boolean based edge detection, Hough transform, voting accumulation.

1. INTRODUCTION

Motion estimation, which may refer to image-plane motion (2-D motion) or object motion (3-D motion) estimation, is one of the fundamental problems in digital video processing and has been the subject of substantial research effort. In short, motion estimation is very important for dynamic scene analysis, such as 3D-object reconstruction [1], object tracking [2], robot navigation [3,4], and so forth. One way to find 3D motion information involves getting its perspective projection on the image plane. This is usually called the "velocity field" and represents the apparent velocity of the image pixels from one frame to another frame. One of the most notable approaches to finding the velocity field is based on the estimation of a measure of the change of image brightness in the frame sequence commonly referred to as optical flow [5]. Optical flow represents an approximation of the velocity field, which is a purely geometric concept. In many applications, optical flow is a sufficient approximation of the velocity field and can be reasonably employed in its place. Different approaches for optical flow estimation exhibit different behavior with respect to discontinuities and for different types of motion. The two most popular approaches of optical flow are the regularization-based and multiconstraint-based approaches.

Regularization-based [5,6,7] approaches consider velocity field estimation as an ill-posed problem. Solutions are obtained by minimizing a function where a smoothness constraint is appropriately weighted to regularize the solution. Usually, these methods lead to iterative solutions and the velocity is evaluated at every point of the image. Drawbacks of these approaches are that difficulties occur at the regions of object occlusions. Further the depth of propagation of the field depends on the number of iterations used and on the weighting factor of the regularizing.

Multiconstraint-based [8,9,10] approaches of optical flow are based on the possibility of defining a set of constraint equations for the point under consideration. This set of equations is usually solved by numerical methods. Traditional numerical methods, like the least-squares technique, are averaging methods

Manuscript received November 29, 2001; accepted July 18, 2002. This project was supported by Factory Automation Center for Parts of Vehicles (FACPOV) at Chosun University, Kwangju, Korea. FACPOV is designated as a regional research center of Korea Science and Engineering Foundation (KOSEF) and is operated by Chosun University.

Min-hyuk Chang, Il-jung Kim, Jong-an Park are with the Division of Electronics and Information and Communication Engineering, Chosun University, Korea. (e-mail: {millcre, japark} @chosun.ac.kr).

and are thus susceptible to errors in cases of occlusion and of noise. Another drawback of these optical flow based methods is the imprecise detected motion boundary because motion is usually nonhomogeneous near the motion boundary. Furthermore, the object contour remains undetermined when the motion of the object is similar to that of its neighborhood. Therefore, several methods have been proposed that use not only optical flow, but also other information such as color and edges [11-14].

In this paper, a new multiconstraint based optical flow along with Boolean based edge detection is used for finding motion information in the image sequences. We first find the locations and areas of the moving objects in the input image sequence so that we can reduce the computational time for the Hough transform and voting accumulation to find the optimal constraint lines for optical flow estimation. The Boolean based edge detector [15] is applied to the two input image sequences, and then we can find the location of moving objects by taking the binary difference of the two edge maps. After dilating the resulting difference image, we obtain the areas of possible moving objects. In this process, we also get erroneous results of moving objects due to noise. We remove these erroneous moving objects by holding the the possible moving objects at a certain threshold. The moving objects, which have considerable area (counting the number of pixels in the segmented moving objects), are called the real moving objects. After finding the real moving objects, we separately find the optical flow in those areas to estimate the direction in which the objects are moving.

For finding the optical flow of the detected moving objects, a method based on the multiconstraint-based approach is presented, which evaluates optical flow constraint (OFC) equations in the neighborhood of each pixel. The solution is obtained from the Combinatorial Hough Transform (CHT) [17] and vote accumulation, which avoids drawbacks associated with the least-squares methods. The calculation of many points along the constraint lines is also avoided, from the transformed slope-intercept parameter domain. For reducing the operating time, we use the logical operation method for computing brightness gradients. We use edge detection and segmentation to extract the real moving areas in the images and thus reducing the computational time of the CHT.

The simulation results show that the proposed method is very effective for extracting optical flow vectors and hence recognizing moving objects in the images. Finding the moving objects using edge detection and segmentation significantly reduces the computation time. . The Boolean based edge detector gives very thin edges and helps achieve accurate segmentation of moving objects. Optical flow constraints are solved using the multiconstraint approach and the Combinatorial Hough Transform, which avoid the problems associated with the least-square based optical flow methods.

This paper is organized as follows. Section 2 describes the detection of moving objects using the Boolean based edge detector. Section 3 explains the extraction of velocity vectors based on the CHT and vote accumulation. Simulation results are shown in Section 4.

2. EXTRACTION OF MOVING OBJECTS

Two consecutive input images are taken from an image sequence. Moving objects are extracted from the two input images. Edges of the input images are found with the help of the new Boolean based edge detector. The Boolean based edge detector that is used in the proposed algorithm is fast and accurate and provides very thin edges. Other popular edge detectors, such as Sobel, Laplacian, and so on, provide comparatively thick edges. The differential edge image, which will be explained later, provides much better results on thin edges. The Boolean based edge detector used in the algorithm is briefly explained in the next subsection.

2.1. Boolean based edge detector

The Boolean based edge detector is based on local operations, global operations, and Boolean algebra. We take a 3×3 window of the original gray-level image, and the local threshold is found based on the local mean value. This local threshold is used for setting a threshold for the image points, which converts the gray-level image into binary image. The resulting 3×3 binary image window is cross-correlated with sixteen edge-like binary patterns. We use Boolean functions in the cross-correlation of the image window. The resulting intermediate edge map contains true edges as well as false edges. The false edges are generated by noise in the image and are removed by another parallel path. In this path, the same 3×3 window of the image is globally held at a certain threshold based on the variance of the window. The global threshold is pre-selected, considering the presence of noise in the image. A logical AND combines the resulting intermediate edge map with the intermediate edge map from local threshold. The block diagram is

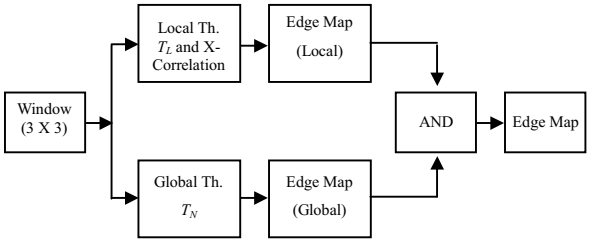


Fig. 1. A block diagram of the Boolean based edge detector.

shown in Fig. 1, and each step is now explained in detail.

2.1.1 Local threshold

The idea of global and multiple thresholds have been applied in the various edge detectors. Not all images can be neatly processed for edges using simple thresholds. The intensity histogram of an image has a different picture for different types of images. With global thresholds, we expect to see a distinct peak in the histogram for determining the global threshold for the entire image. But in most cases, such peaks do not exist, and a simple threshold is unlikely to produce a good result. In this case, an adaptive threshold may be a better answer.

An adaptive threshold changes the threshold dynamically over the image. This more sophisticated threshold can accommodate changing lighting conditions in the image, e.g., those occurring as a result of a strong illumination gradient or shadows. With an adaptive threshold, a threshold is calculated for each pixel in the image. Different methods are available for setting an adaptive threshold. One of them is creating a local threshold. To find the local threshold, we statistically examine the intensity values of the local neighborhood of each pixel. The statistic, which is most appropriate, depends largely on the input image. Simple and fast functions include the mean of the local intensity distribution.

The local threshold (T) value for each center pixel is selected as either

$$T = Mean,\ T = Median,\ T = (Max + Min) / 2,$$
$$\text{or } T = (Max - Min) / 2.$$

We use the mean value approach because it is already calculated for the variance function. On the margin, however, the mean of the local area is unsuitable as a threshold because the range of intensity values within a local neighborhood is very small and their mean is close to the value of the center pixel. The situation can be improved if the threshold employed is not the mean but is (*mean* - C) where C is a constant. From experiments, the trade-off value of C is equal to 5. We thus use (*mean* - C) as the local threshold. For a 3×3 image window $W(x,y)$, we calculate the mean as

$$Mean\ \mu = \frac{1}{N \times N} \sum_{x=0,y=0}^{x=N,y=N} W(x,\ y). \qquad (1)$$

The local threshold $T_L(x,y)$ for each center pixel of window $W(x,y)$ is selected as $T_L(x,y) = (\mu - C)$ where C is a constant. A threshold for window $W(x,y)$ is set as $W_L(x,y) = 1$ if $W(x,y) > T_L(x,y)$. Otherwise, $W_L(x,y) = 0$.

2.1.2 Boolean function (Local operation)

For edge finding, window $W_L(x,y)$ is cross-correlated with sixteen edge-like patterns. Any pattern that matches window $W_L(x,y)$ is called an edge at the center of window $W(x,y)$, and $B_L(x,y)$, called the local edge map, is made. The sixteen patterns are like Prewitt compass masks. These patterns cover nearly all possible edge patterns in every direction. The cross-correlation of window $W_L(x,y)$ with edge patterns is accomplished by Boolean functions. The Boolean equation for the first mask is expressed in Eq. (2). A logical OR is used in all similar equations to get either one or zero at the center of window $W(x,y)$.

$$B0 = !B(0,0) \times B(0,1) \times B(0,2) \times !B(1,0) \times B(1,1) \times B(1,2) \times !B(2,0) \times B(2,1) \times B(2,2). \qquad (2)$$

2.1.3 False edge removal (global threshold)

So far in this work, false edges are detected due to the presence of noise. We now remove the false edges with the global threshold approach. We take a new threshold, T_N, whose value is related to the noise level in the image. The selection of global threshold T_N is determined experimentally. However, its value is moderate. In gray-scale images, a global threshold of 20 to 35 is good enough in most cases. Its value plays no major role in the algorithm. If noise is higher, then the value should be higher, say, 35.

The variance function has its maximum value at an edge. So, the variance is calculated again locally for each window W (3×3) and a threshold is set as

if $\sigma^2_{x,y} > T_N$, $B_G(x,y) = 1$, otherwise $B_G(x,y) = 0$, and

$$\sigma^2_{x,y} = \frac{1}{N \times N} \sum_{x=0}^{x=N-1} \sum_{y=0}^{y=N-1} \left[g(x,y) - \mu_{x,y} \right]^2 \qquad (3)$$

where $g(x,y)$ is the intensity value of window $W(x,y)$, μ is the mean of the neighbors (3×3) at (x,y) position, and $N \times N$ is the window size. $B_G(x,y)$ is called the global edge map.

The local threshold (*mean* - C) gives better edge localization, while global threshold (variance) limits the spread of pixel intensity, above which edge presence is maximum.

A logical AND is used to combine the local edge map $B_L(x,y)$ and the global edge map $B_G(x,y)$, creating the final edge map.

2.2. Differential edge image

The locations and areas of moving objects in the input image sequence are found to reduce the computation time of the CHT, and vote accumulation is used to find the optimal constraint lines of the optical flow estimation. After using the Boolean based edge detector to get the edge maps (see Section 1), we find the binary differential edge image from the two resulting input edge maps to remove the background (the still part) in the images.

$$D(x,y) = ABS(E_2(x,y) - E_1(x,y)) \quad (4)$$

where $E_2(x,y)$, $E_1(x,y)$ are the two binary edge maps of the input image sequence and $D(x,y)$ is the resulting binary difference image. The resulting binary difference image $D(x,y)$ gives us the possible location of moving objects. To find the areas of moving objects, we binary dilate the difference image $D(x,y)$ as

$$DL = dilate\ (D) \quad (5)$$

where DL is the dilated image of the difference binary image D. The dilated image DL detects the areas of moving objects in the image sequence. The dilation of the difference image should be enough to ensure that the areas of the moving objects detected in the dilated image are greater than the actual areas of the moving objects.

Actually, in the dilated image DL, all possible moving objects (both real and erroneous moving objects) are detected. The erroneous moving objects are detected due to the presence of noise in the images. We applied a method for determining thresholds to extract the real moving objects from the dilated image DL. We first label the moving objects in the dilated image DL and then calculate the binary areas of each of the moving objects. We set a threshold for the real moving objects that have considerable area in the dilated image:

If $A[DL(j)] > T_{area}$
    Real Moving Object (keep it),
else
    Erroneous Moving Object (discard it)

where $A[DL(j)]$ calculates the binary area (count the number of 1s of the labeled object) of the jth labeled object in DL and where T_{area} is the threshold, the value of which depends on the size of the input images and the distance of camera from the scene. From experiments, the value of T_{area} indicates that 15% of the image gives accurate results. Fifteen percent seems high to discard the erroneous moving objects, but $A[\bullet]$ is a dilated image, so a threshold of 15% is actually just a 5–8% threshold. We discard the erroneous moving objects by replacing 1s with 0s in that area. Finally, we get the image, which contains only real moving objects in the image sequence. We then calculate the optical flow vectors in those actual moving areas.

We presented the CHT and vote accumulation based optical flow estimation method in an earlier paper [16]. In that paper, we used a difference image on the gray level images. We used median operators for removing the noise in the images. After noise removal, the process of erosion and dilation revealed the moving objects in the images. Median, erosion, and dilation operators are highly time-consuming processes, which highly increases the computation time. Here in this paper, we avoid median and erosion operators, and hence reduce the computation time. However, edge detection increases the computation time but considerably less than the median and erosion operators. We used the Boolean based edge detector, which offers very thin and accurate edges. The differential edge image obtains better results on thin edges.

3. EXTRACTION OF VELOCITY VECTORS BASED ON CHT AND VOTE ACCUMULATION

3.1. OFC equation

Assuming that the image brightness, $E(x(t), y(t), t)$, is stationary with respect to time (i.e., $dE/dt = 0$), the flow of its features pattern can be modeled by a sort of continuity equation:

$$E_x u + E_y v + E_t = 0 \quad (6)$$

where E_x, E_y, E_t represents partial derivatives of the image brightness in the x, y and t directions, respectively, and where u and v correspond to dx/dt and dy/dt , respectively, and represent the components of the velocity vector of the features pattern along the x and y axes, respectively, on the image plane (the optical flow components). Eq. (6) is usually called the OFC, and estimation methods based on the OFC equation are commonly referred to as gradient-based methods. The derivatives of brightness are estimated from the discrete set of image brightness measurements available.

3.2. Combinatorial Hough transform and vote accumulations

The OFC equation cannot provide a unique solution by itself. In fact, the OFC can be regarded as the equation of a line in the (u,v) plane.

$$v = m\,u + c \quad (7)$$

where $m = -Ex/Ey$ is the slope and $c = -Et/Ey$ is the intercept. Any point along this line is a possible solution for the optical flow estimation problem.

In this paper, we used the logical comparison method for improving the gradient operation speed. The operation algorithm for logical comparison is outlined in Table 1.

3.2.1 Transform to the (m,c) plane

A multiconstraint solution based on the OFC followed by the vote accumulation method identifies the most likely solution as the point (u,v), where most of the constraint lines lies in the vicinity of each pixel intersect. Using this approach, the characteristics of each constraint line are transformed from the (u,v) plane to the slope-intercept plane (m,c), where each constraint line is represented by

a point according to Eq. (7). The requirement of a common intersection point in the (u,v) plane of a set of constraint lines is equivalent to the requirement of colinearity of their corresponding points in the parameter plane (see Fig. 2). The estimation of optical flow at each pixel is thus reduced to finding the best line that matches the pattern of points corresponding to the constraint lines (around each pixel) in the parameter plane. The best line is the line on which the largest number of points reside, not the line that gives the minimum accumulative distance to the points. In practice, we consider 5×5 blocks, with 2 pixels of overlap.

Table 1. The algorithm for logical comparison.

```
//Initialization
    Start Position : x = 1, y =1;
    Operation Flag : flag[255][255] = 0
//if not equal first image and second image in same position
for (i=0; i<255;i++) {
   for(j=0; j<255;j++) {
      //f(t+1) : second image
      //f(t) : first image
      if(f(t+1) != f(t)) {
         if(flag(i, j) == 0) {
            Operate E_x, E_y, E_z at (i, j);
            flag(i, j) = 1;
         }
         if(flag(i+1, j) == 0) {
            Operate E_x, E_y, E_z at (i+i, j);
            flag(i+1, j) = 1;
         }
         if(flag(i, j+1) == 0) {
            Operate E_x, E_y, E_z at (i, j+1);
            flag(i, j+1) = 1;
         }
         if(flag(i+1, j+1) == 0) {
            Operate E_x, E_y, E_z at (i+1, j+1);
            flag(i+1, j+1) = 1;
         }
      }else {
         E_x=E_y=E_t = 0;
      }
   }
}
```

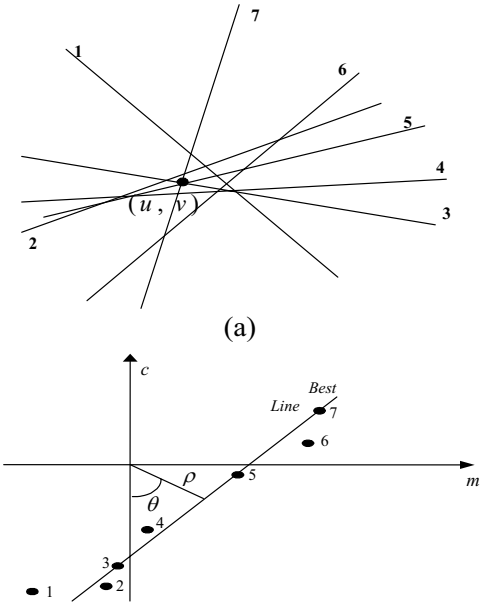


Fig. 2. Constraint line parameterization; (a) *uv*-plane, (b) corresponding best line in the *mc*-parameter plane.

3.2.2 Transform to the (ρ,θ) plane and vote accumulation

Votes are cast using the combinatorial hough transform. Each couple of points, (m_1,c_1) and (m_2,c_2), in the slope-intercept plane, corresponding to a couple of constraint lines, adds a vote to a monodimensional accumulation histogram of the best line □ as

$$\theta = \tan^{-1}(-\frac{m_2 - m_1}{c_2 - c_1}), \tag{8}$$

$$\rho_i = m_i \cos(\theta_{\max}) + c_i \sin(\theta_{\max}). \tag{9}$$

Therefore, according to the multiconstraint approach on an $N\times N$ area, $(N^4-N^2)/2$ couples of constraint equations exist, and thus their solutions also exist. This could lead to an asymptotical complexity equal to I^2N^2 ($I^2 = I\times I$, or the size of the image) and is simplified by considering only the combinations of the constraint lines associated with the pixels in the $N\times N$ area and the constraint line of the center of the multi-point area. Thus, (N^2-1) pairs of equations, and hence, N^2 votes, are obtained. The histogram in □ is inspected to find the most probable value, $\theta_{\max}$ (the value corresponding to the peak of the histogram). By using that value, a second stage of N^2 votes is used to define another histogram for the other line parameter. We computed $\rho_{\max}$ as

$$\rho_{mean} = \frac{1}{N^2}\sum_{i=1}^{N^2} \rho_i . \tag{10}$$

Then search ρ within the defined range in the ρ set.

$$\rho' = \{i \mid \rho_{mean}.\alpha < \rho_i < \rho_{mean}.\beta; i = 1,..., N^2\} \tag{11}$$

where α is 0.4 (minimum factor) and β is 1.5 (maximum factor) experimentally. ρ_{max} is the mean of ρ' within the value defined in second process.

$$\rho_{\max} = mean\ (\rho') . \tag{12}$$

3.2.3 Computing the velocity vector

The best approximation of the best line is defined by $(\rho_{max},\theta_{max})$. Therefore, the optical flow at each pixel can be directly derived from these line parameters.

$$u = \cot(\theta_{\max}), v = \frac{\rho_{\max}}{\sin(\theta_{\max})} . \tag{13}$$

3.3. Extracting optical flow for the moving objects

For extractingoptical flow for the moving objects, we used the mask operation between the dilated image having only real moving objects and the original image sequence. Then, the information of the moving object computed in Section 3.2 is displayed in those areas in the image sequence

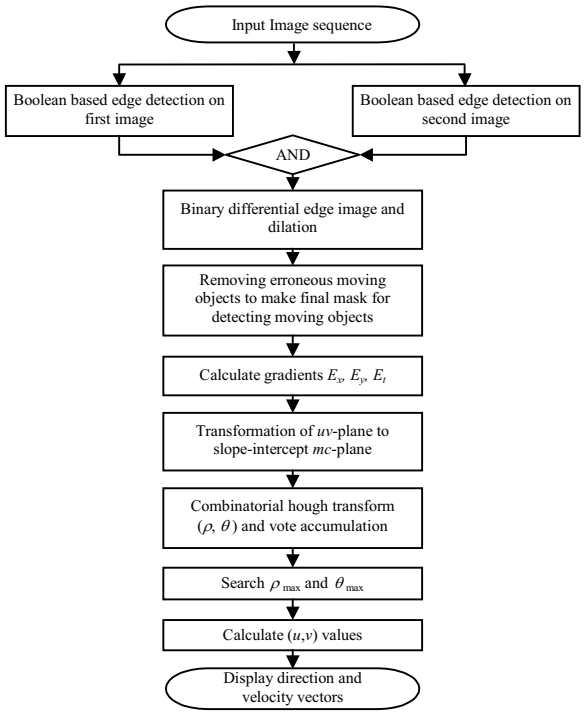


Fig. 3. Diagram of the proposed algorithm.

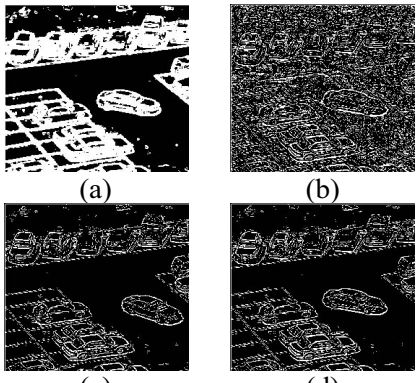
where motion is found (see Section 2). The diagram of the proposed algorithm is shown in Fig. 3.

4. SIMULATION RESULTS

For simulation, a two-frame (256×256), gray-level sequence is used, as shown in Fig. 4. A Boolean based edge detector is applied on the input image sequence. Fig. 5 shows the step-by-step details of the Boolean based edge detector on the input image. The results are obtained with conditions of global noise T_n = 20 and the constant C = 5. Fig. 5(a) shows the image after applying only the local threshold (*mean* – *C*) on the input image. Fig. 5(b) shows the image after applying the global threshold (variance). Finally, in Fig. 5(c), the two operations are combined to get the final edge image of the original image. Fig. 5(a) indicates that the maximum possible edges in the image are detected. This edge map contains both true as well as false edges. Fig. 5(a) is obtained by using only local threshold (*mean* – *C*) and after cross-correlation with the sixteen edge-like patterns. Fig. 5(b) shows the image obtained by only global threshold and variance of the image. The variance function limits the spread of pixel intensity, above which edge presence is at maximum. Fig. 5(c) shows the results of using a logical AND operation to combine the two edge maps obtained from local and global operations. It means we used both Fig. 5(a) and Fig. 5(b), via an AND operation, to get the final edge detection in Fig. 5(c). Fig. 5(d) illustrates the edge detection on the second input image.

(a) (b)

Fig. 4. Input image sequence: (a) first image, (b) second image.



(a) (b)

(c) (d)

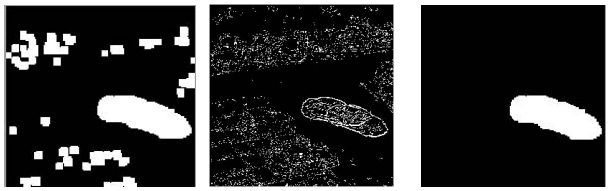
Fig. 5. Boolean based edge detection: (a) local operation, (b) global operation, (c) final edge map on first input image, (d) final edge detection on the second input image.

After getting the edge maps of the two input image sequence with the Boolean based edge detector, we find the differential edge image of the two edge maps as shown in Fig. 6(a). The dilated image of the resulting difference image is shown in Fig. 6(b). Setting a threhold removes the erroneous moving objects, and the areas of the real moving objects are shown in Fig. 6(c). The proposed algorithm of optical flow using the CHT and vote accumulation is applied on the detected moving objects instead of the whole image. The result of the final optical flow is shown in Fig. 7(a), and the direction vector of the moving object is shown in Fig. 7(b).

The results of the proposed method, method III, are compared with method I (Campani and Verri, 1990) and method II (CHT and voting accumulation, 2000). The operation time, computational complexity, and moving object angle estimation of the three methods are compared. The simulation results are shown in Table 2.

Method I uses the multi-point based algorithm, but it also has greater computational complexity than the proposed methodand has the local minimum problem at some part of the image.

Method II uses the CHT and vote accumulation based optical flow estimation method [16]. The calculation of many points along the constraint lines is avoided, taking into consideration the transformed slope-intercept parameter domain.



(a) (b) (c)

Fig. 6. Detection of the real moving objects; (a) the differential edge image of the two input edge maps, (b) the dilated image of the difference image, (c) the real moving objects after removing the erroneous moving objects.

(a) (b)

Fig. 7. Direction and velocity vectors; (a) optical flow vectors using the proposed method, (b) the direction vector of the moving object.

Table 2. Feature comparison of the three methods.

Algorithm	Complexity	Operation time	Moving distance	Angle
Campani & Verri	N^2I^2	843.66	25.1868	27.125
CHT & Voting	N^2I^2	286.96	24.2578	35.131
Proposed method	$N^2I_N^2$	151.52**	24.6752	36.105

$I \times I$: the size of the image,
$N \times N$: the size of the neighborhood,
$I_N \times I_N$: the size of the moving region,
**: pre-filtering of the N×N region.

Method III proposes using angle estimation to track the moving objects using base-line and optical flow and applying Boolean edge detection to the resulting difference image to find possible moving objects.

Method III offers less simulation time than the other two methods. Edge detection, dilation, and mask operations are time-consuming processes. But the optical flow is found only in the detected moving regions, so the overall computation time is much less than with the classical methods in cases of a small number of moving objects. However, with a large number of moving objects, the computation time of the proposed method will be higher. The proposed algorithm works very well in cases of a small number of large moving objects. The direction detection of the moving object is improved if the object is well-segmented. If object is not well-segmented, the direction detection of the moving object contains small errors, because the occluded part of the moving object misleads the algorithm.

5. CONCLUSION

The problem of tracking moving objects in a video stream is discussed in this paper. We discussed the popular technique of optical flow for moving object detection. Optical flow finds the velocity vectors at each pixel in the entire video scene. However, optical flow based methods require complex computations and are sensitive to noise. In this paper, we proposed a new method based on the Hough transform and on voting accumulation for improving the accuracy and reducing the computation time. Further, we applied the Boolean based edge detector for edge detection. Edge detection and segmentation are used to extract the moving objects in the image sequences and reduce the computation time of the CHT. The Boolean based edge detector provides accurate and very thin edges. The difference of the two edge maps with thin edges gives better localization of moving objects. The simulation results show that the proposed method improves the accuracy of finding the optical flow vectors and more accurately extracts moving objects' information. The process of edge detection and segmentation accurately find the location and areas of the real moving objects, and hence extracting moving information is very easy and accurate. The Combinatorial Hough Transform and voting accumulation based optical flow measures optical flow vectors accurately. The direction of moving objects is also accurately measured.

REFERENCES

[1] K. Prazdny, "On the information in optical flow," *Computer Vision Graphics Image Processing*, vol. 23, pp. 239-259, 1983.
[2] P. J. Burt et al., "Object tracking with a moving camera," *Proc. IEEE Workshop on Visual Motion*, Irvine, California, U.S.A., pp. 2-12, 1989.
[3] R. C. Nelson and J. Aloimonos, "Obstacle avoidance using field divergence," *IEEE Trans. on Pattern Analysis Mach. Intell.*, vol. 11, no. 10, pp. 1102-1106, 1989.
[4] M. Subbarao, "Bounds on time-to-collision and rotation component from first-order derivatives of image flow," *Computer Vision Graphics Image Processing,* vol. 50, pp. 329-341, 1990.
[5] B. K. P. Horn and B. G. Schunck, "Determin-

ing optical flow," *Artif. Intell*., vol. 17, pp. 185-203, 1981.
[6] H. H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Computer Vision Graphics Image Processing,* vol. 21, pp. 85-117, 1983.
[7] H. H. Nagel and W. Enkelmann, "Towards the estimation of displacement vector fields by 'oriented smoothness' constraints," *Proc. 7th IEEE Int. Conf. on Pattern Recognition*, pp. 6-8, 1984.
[8] C. Cafforio and F. Rocca, "Tracking moving objects in television images," *Signal Processing,* vol. 1, pp. 133-140, 1979.
[9] M. Campani and A. Verri, "Computing optical flow from an overconstrained system of linear algebraic equations," *Proc. 3rd IEEE Int. Conf. on Computer Vision*, pp. 22-26, 1990.
[10] P. Nesi, A. Delbimbo, and J. L. Sanz, "Multi-constraints-based optical flow estimation and segmentation," *Int. Workshop on Computer Architecture for Machine Perception*, Paris, pp. 419-426, 1991.
[11] W. B. Thompson, "Combining motion and contrast for segmentation," *IEEE Trans*. on *Pattern Analysis Mach. Intell*, vol. 2, no. 6, pp. 543-549, 1980.
[12] S. Gemen et. al., "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE Trans. on Pattern Analysis Mach. Intell*, vol. 6, no. 6, pp. 721-741, 1985.
[13] M. J. Black, "Combining intensity and motion for incremental segmentation and tracking over long image sequence, " *Proc. ECCV*, pp. 485-493, 1992.
[14] M. Etoh et. al., "Segmentation and 2D motion estimate by region fragments, " *Proc. 4th Int. Conf. on Computer Vision*, pp. 192-199, 1993.
[15] M. B. Ahmad and T.-S. Choi, "Local threshold and Boolean function based edge detection," *IEEE Trans. on Consumer Electronics*, vol. 45, no. 3, pp. 674-679, August 1999.
[16] S.-K. Kang and J.-A. Park, "Estimation of moving information for tracking of moving objects," *The Korean Society of Mechanical Engineers*, vol. 15, no. 3, March 2001.
[17] D. B.-T. and M. Sandler, "A combinatorial Hough transform," *Pattern Recognition Letter*, 11, pp. 167-174, 1990.

Min-hyuk Chang received the B.S. and M.S. degrees in Electronic Engineering from Chosun University in 1995 and 1997, respectively. He is currently proceeding toward the Ph.D. Degree in Electronic Engineering at Chosun University, Gwangju, Korea. His research interests are digital signal processing, vision system motion estimation, motion tracking, and pattern recognition.

Il-jung Kim received the B.S. and M.S. degrees in Electronic Engineering, from Chosun University in 1995 and 1997, respectively. He is currently proceeding toward the Ph.D. Degree in Electronic Engineering at Chosun University, Gwangju, Korea. His research interests are digital signal processing, pattern recognition, and home networking.

Jong-an Park received the Ph.D. degree in Electrical Engineering from Chosun University in 1986. Since March 1978, he has been a Professor at the School of Electronics, Information, and Communications Engineering, Chosun University, Gwangju, Korea. He was a visiting professor at the University of Massachusetts, U.S.A., from 1983 to 1984 and at the University of Surrey, United Kingdom, from 1990 to 1991. His research interests are digital signal processing and pattern recognition.