2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Background Compensation Using Hough Transformation

Xuan Dai Pham, Jung Uk Cho, Jae Wook Jeon, *Member, IEEE*

***Abstract*— This paper proposes a method for detecting the camera motion between two successive images to compensate background motion. The camera motion is restricted with regard to panning, tilting, and zooming. For small panning and tilting angles and small values of zooming difference, we assume that the apparent background motion occurring between two consecutive images can be approximated. We perform this using a scaling transformation followed by a translation. The vertical and horizontal histograms of two successive images are created and then matched using Hough transformations. The transformation parameters are determined when the vertical and horizontal histograms are matched. A multi-resolution Hough transformation is employed to reduce processing time.**

I. Introduction

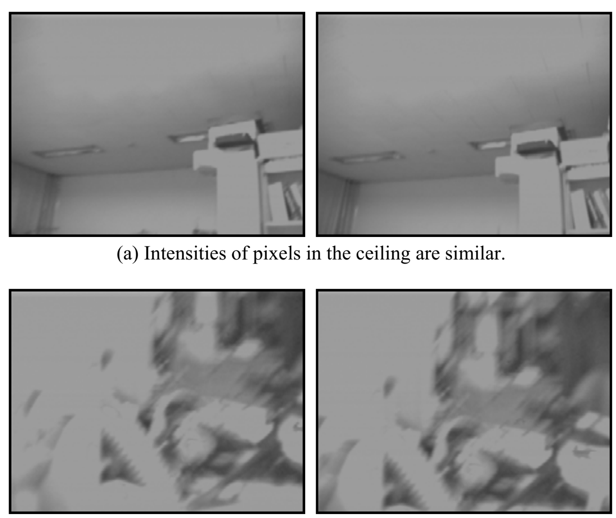
MOTION-based tracking with an active camera involves the difficulties of eliminating the camera motion (background motion), detecting the features of object motion, and the tracking mechanism. Several alternative approaches eliminate background motion.

In [1], the background motion is represented by an affine transformation. The affine parameters are estimated using the LMedS (least median of squares) method. Several feature points and their corresponding points in other frames are selected, and pixels selected from the moving object's regions are used as outliers for the parameter estimation. In approaches that estimate the transformation parameters using the positions of corresponding points, the points must be carefully chosen. These estimated parameters may be inaccurate if the points are not from the background as well as the moving object's regions. It is difficult to determine the correspondences of feature points due to motion blur. In [2], the relationship between pixels representing the same 3D point in images captured at different camera orientations is used to eliminate background motion. The values of the pan and tilt angles of the camera must be known for each captured image in order to use this approach. The disadvantage of this approach is that it is difficult to synchronize the time required to obtain the values of the pan and tilt angles and that required to capture the images. In [3], lines, polygons or other background line structures are matched to compensate the background motion. [4] describes a parametric motion model estimation algorithm. In that paper, they developed M-estimator like techniques in a multi-resolution framework. Some comparisons in this paper show that the main advantage of the proposed method with respect to [4] resides in the motion estimation in presence of importance motion blur.

This paper proposes a method to detect camera motion between two successive images. Camera motion is restricted with regard to panning, tilting, and zooming. Assume the panning and tilting angles as well as the zooming differences are small; the background motion between two successive images can be approximated by a scaling transformation followed by a translation.

Fig. 1 shows two pairs of two successive images captured when the camera was rotating and changing its zoom values. In Fig. 1-(a), it is difficult to select feature points and their corresponding points for parameter estimation in some parts of the background (the ceiling) because the gray intensities of pixels in this part are similar. In Fig. 1-(b), the two images are damaged by blur due to camera motion. It is difficult to select feature points and their corresponding points in the two photos.

The proposed method does not use the pairs of feature points and their corresponding points to determine the parameters of the transformation. This obviates the difficulties in selecting feature points and their corresponding points in successive images.



(a) Intensities of pixels in the ceiling are similar.

(b) Images are damaged by blur.

Fig. 1. Example difficulties for selecting feature points and corresponding points in successive images.

Manuscript received September 14, 2007. This research was performed as part of the Intelligent Robotics Development Program, one of the 21st Century Frontier Research and Development Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

Xuan Dai Pham is with School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea (e-mail: pdxuan@ece.skku.ac.kr).

Jung Uk Cho is with School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea (e-mail: ichead@ece.skku.ac.kr).

Jae Wook Jeon is with School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea (e-mail: jwjeon@yurim.skku.ac.kr).

Vertical and horizontal histograms are used instead. First, the previous and current grayscale images are separated into several binary images. Vertical and horizontal histograms are created from the binary images. Separating grayscale images into several binary images increases the tolerance to changes in illumination. The vertical and horizontal histograms of the previous image are matched with those of the current image via a Hough transformation. Paul V. C. Hough's transformation [5], [6] is a technique employed to extract the features of a particular shape within an image. The principal concept of the Hough transformation is to solve the original problem of locating collinear points via the mathematically equivalent problem of locating concurrent lines. The Hough transformation has been proven to be robust, and can even be successfully utilized for the detection of overlapping or semi-occluded objects in noisy images. Multi-resolution Hough transformation [7], [8], [9] reduces processing time. Implemented on a computer, the average time for processing one frame was 2.05s with image sizes of 640 × 480 pixels, for the proposed method.

This paper is organized as follows. Section II discusses background compensation using Hough transformation. Section III describes our experiments and their results. Section IV concludes with a brief summary.

II. Background compensation using Hough transformation

The motion features of a moving object can be detected using the image difference technique. This technique is widely used for detecting moving objects [2], [10], [11]. The difference image (interframe difference, IFD) is calculated by performing a pixel-by-pixel subtraction between two consecutive images. When the camera is mounted on a pan-tilt device and it rotates and moves, the background of the camera image changes. In order to detect the motion features using the image difference, the background motion must be compensated.

We assume that the camera motion is restricted with regard to panning, tilting, and zooming. Further, we assume that the pan-tilt angles as well as the zooming difference between two successive image frames are relatively small. It is thus assumed that the apparent background motion is dominant in the image sequences, and that the background motion between two consecutive image frames can be represented by a geometric image transformation such as a scaling transformation followed by a translation. Less occlusion and more accurate approximations are obtained when the motion is smaller.

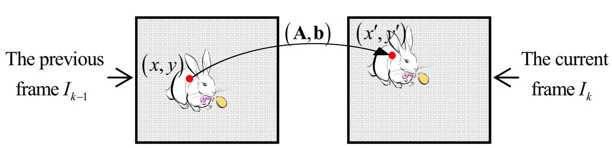


Fig. 2. Background motion represented by a scaling transformation followed by a translation

Let I_k and I_{k-1} be the images at time index k (current image) and $k-1$ (previous image), respectively. Given the above assumptions, a non-proportional scaling transformation followed by a translation is a geometric, two-dimensional transform used to transform I_{k-1} to I_k. The geometric transformation maps points in I_{k-1} to points in I_k. The transformation parameters can be expressed by a 2 × 2 matrix **A**, and a 2 × 1 matrix **b**. Fig. 2 shows the transformation that maps the point (x, y) in I_{k-1} to the point (x', y') in I_k.

The relation between (x, y) in I_{k-1} and (x', y') in I_k can be expressed as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{b} \tag{1}$$

in which **A** is a 2×2 matrix and **b** is a 2×1 matrix:

$$\mathbf{A} = \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad a_{11} > 0, a_{22} > 0$$

and the gray level of (x, y) in I_{k-1} is equal to that of (x', y') in I_k:

$$I_k(x', y') = I_{k-1}(x, y) \tag{2}$$

The IFD with background compensation between two successive images I_{k-1} and I_k is defined by

$$IFD_k(x', y') = \left| I_k(x', y') - I_{k-1}(x, y) \right| \tag{3}$$

where (x', y') is defined by equation (1). $IFD_k(x', y')$ is determined in points (x', y') belonging to I_k, and their corresponding points (x, y) belonging to I_{k-1}. This implies that IFD is determined in the region common to two successive images.

The transformation maps all points that have the same coordinates x in I_{k-1} to those having the same coordinates $x' = a_{11}x + b_1$ in I_k, i.e., it maps vertical lines to vertical lines. Similarly, the transformation maps all points having the same coordinates y in I_{k-1} to those having the same coordinates $y' = a_{22}y + b_2$ in I_k, i.e., it maps horizontal lines to horizontal lines.

First, the previous grayscale image I_{k-1} is separated into N binary images, $I_{k-1}^l(x, y)$, $0 \le l \le N-1$. In the original grayscale image I_{k-1}, all pixels whose intensity ranges from $\frac{255l}{N}$ to $\frac{255(l+1)}{N}$ are pixels 1 in the binary image I_{k-1}^l, otherwise they are pixels 0. Similarly, the current image I_k is also separated into N binary images I_k^l, $0 \le l \le N-1$. The illumination changes are not taken into account given equation (2). However, the fact that all pixels whose intensity range from $\frac{255l}{N}$ to $\frac{255(l+1)}{N}$ are pixels 1 in the binary image helps the algorithm tolerate illumination changes.

Let $h^V(I)$ be the vertical histogram of the binary image I

$$h^V(I) = \{h_j^V(I) : j = 0,...,W-1\} \quad (4)$$

where W is the number of columns in I and $h_j^V(I)$ is the number of pixels 1 in column j. Similarly, let $h^H(I)$ be the horizontal histogram of the binary image I

$$h^H(I) = \{h_i^H(I) : i = 0,...,H-1\} \quad (5)$$

where H is the number of rows in I and $h_i^H(I)$ is the number of pixels 1 in row i.

Since all points that have the same coordinates x have the same coordinates $x' = a_{11}x + b_1$ after mapping, column j is mapped into column j', where $j' = a_{11}j + b_1$. If column j' in the image $I_k^l(x,y)$ is a column obtained from the mapping of a column j in the image I_{k-1}^l, the number of pixels 1 in column j', $h_{j'}^V(I_k^l)$, is approximately equal to the number of pixels 1 in column j, $h_j^V(I_{k-1}^l)$,

$$h_j^V(I_{k-1}^l) \approx h_{j'}^V(I_k^l) \quad (6)$$

They are not equal because two successive images may be taken from different angles due to camera rotation, and changes in zoom. Condition (6) in the case of $h_j^V(I_{k-1}^l) \neq 0$ can be expressed by

$$\left| \frac{h_j^V(I_{k-1}^l) - h_{j'}^V(I_k^l)}{h_j^V(I_{k-1}^l)} \right| < T \quad (7)$$

where T denotes the tolerance, a parameter of the method. If $h_j^V(I_{k-1}^l) = 0$, t column j does not correspond to any column j'.

In order to determine values of parameters a_{11} and b_1, vertical histograms of the binary previous and current images, $h^V(I_{k-1}^l)$ and $h^V(I_k^l)$, $0 \le l \le N-1$, are used. For each value of l, $0 \le l \le N-1$, all pairs of (j, j') that satisfy (7) are searched. Assuming that there exists a transformation between the two successive images represented in equation (1), there exists a relationship between j and j' in all these pairs (j, j'). The relationship is represented by

$$j' = a_{11}j + b_1 \quad (8)$$

and the parameters a_{11} and b_1 can be determined by using all pairs (j, j').

Equation (8) demonstrates that the relationship between j and j' in the pairs is linear. Thus, the parameters a_{11} and b_1 can be determined via Hough transformations. Moreover, according to the value of T, noisy pairs (j, j') are found by using equations (7). The principle advantage of the Hough transformation is its strong immunity to noise [12].

In order to determine the parameters a_{11} and b_1 using Hough transformation, all points (j, j') are transformed into Hough (or $\rho\theta$) space as sinusoidal curves [5]

$$\rho = j\cos\theta + j'\sin\theta \quad (9)$$

and they specify the peak in the accumulator array of $\rho\theta$ space. Let the values of ρ and θ at the peak be ρ^V and θ^V respectively, and the two parameters a_{11} and b_1 be given by

$$a_{11} = -\frac{\cos\theta^V}{\sin\theta^V}$$
$$b_1 = \frac{\rho^V}{\sin\theta^V} \quad (10)$$

The range of θ in which the peak value θ^V is found depends on the search range of the parameter a_{11}. The range of θ should be reduced to reduce the Hough transformation process time. Parameter a_{11} is greater than zero, and is the scaling factor of the transformation. a_{11} represents the scaling in a horizontal direction. Therefore, it can be found in the range $[a_{\min}, a_{\max}]$ that contains 1. The range of θ can be determined according to the range $[a_{\min}, a_{\max}]$, as follows:

$$\operatorname{atan}\left(-\frac{1}{a_{\min}}\right) \le \theta \le \operatorname{atan}\left(-\frac{1}{a_{\max}}\right) \quad (11)$$

The advantages of the Hough transformation are its robustness to image noise and the pattern discontinuities the Hough transformation detects. The disadvantages of the standard Hough transform include its requirements for significant computing power and large storage. A multi-resolution Hough transform method has been proposed in [7], [8], and [9]. Using multi-resolution images and accumulator arrays, the multi-resolution Hough transformation reduces computing time. Logarithmic range reduction is used for faster convergence. The multi-resolution Hough transform is used to reduce the computation and storage requirements. Furthermore, it reduces the time for searching all pairs of (j, j') that satisfy the conditions given by equation (7). The multi-resolution Hough transform is first applied to the smallest image, subsequent iterations use images of increasing sizes.

The smallest range is used in the first iteration, and all possible pairs (j, j') are tested to satisfy the condition given in equation (7). The number of possible pairs (j, j') depends on the number of elements in the vertical set. Therefore, it depends on the number of columns in the smallest image. In subsequence iterations, parameters a_{11} and b_1, which are determined in the previous iteration, are used to estimate the values of j' using equation (8), and given values of j.

Therefore, pairs (j, j') are searched to satisfy the conditions given by equations (7) in the neighbourhood of the estimated values of j'. This reduces the time required to search pairs (j, j'). If the number of iterations L in the multi-resolution Hough transform is large, the computing time required for the Hough transform is reduced. However, if L is large, the proposed method is unstable, because the smallest image used in the first iteration lacks the distinct features required to detect the transformation parameters. This implies that the smallest images resulting from L down-sampling reductions of the previous and current images are too similar to use for determining the parameters of the transformation. In the proposed method, the number of iterations L was set to 2.

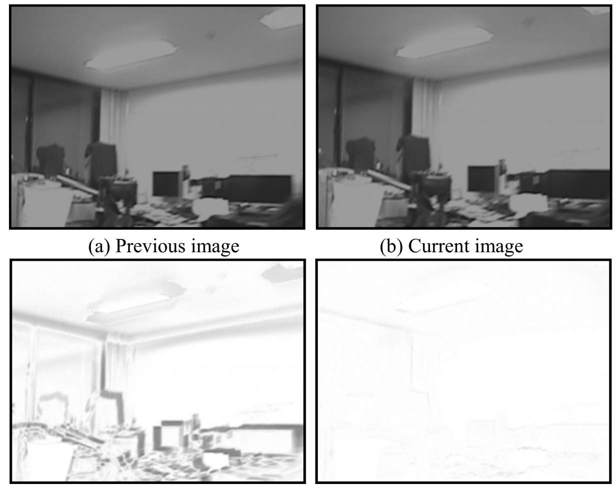
The parameters a_{22} and b_2 are determined in the same manner.

III. Experimental Results

We used a Sony EVI-D100 CCD video camera and a frame grabber board for our experiments. The EVI-D100 camera was connected to the frame grabber and to a PC through its RS-232C port. All settings to the camera were set through this serial port. Grayscale images, captured by the frame grabber board, had a resolution of 640×480 pixels. Its maximum pan speed was 300 degrees/sec, and the maximum tilt speed was 125 degrees/sec. The zoom position ranged from 0h (optical wide end) to 4000h (optical tele end). The PC used had a AMD 64 X2 Dual Core Processor 4400+ 2.21 GHz with 3 GB of RAM and the Windows XP operating system.

In our experiments, the number of binary images N was 32. The tolerance value T in equations (7) was set to 0.1. The range $[a_{\min}, a_{\max}]$ was set to $[0.9, 1.1]$. Therefore, the range of θ in equation (11) was $[-48.04^\circ, -42.30^\circ]$. The parameters for the multi-resolution Hough transform were as follows: the number of iterations L was set to 2; the reduction factor $\sigma = 2$, the parameter $\mu = -1$, and α was set to 3084. It implies that each parameter of the Hough transformation is to be solved to $\frac{1}{\alpha} = 0.0003$ of its full range.

Fig. 3-(a) and 3-(b) show two successive images. These two images were captured while the camera was rotating and changing its zoom. Fig. 3-(c) and Fig. 3-(d) shows the IFD without and with background compensation between these two images. The IFD with background compensation is calculated using equation (3). Images in Fig.3 –(c), (d) are inverted for showing.



(a) Previous image (b) Current image

(c) IFD without background compensation (d) IFD with background compensation

Fig. 3. Two successive images and their IFDs

In IFD, the border that corresponds to new appearing parts and the old disappearing parts is filled with black pixels. IFD is only determined in the common region of successive images. The parameters of the transformation determined from these two images are as follows

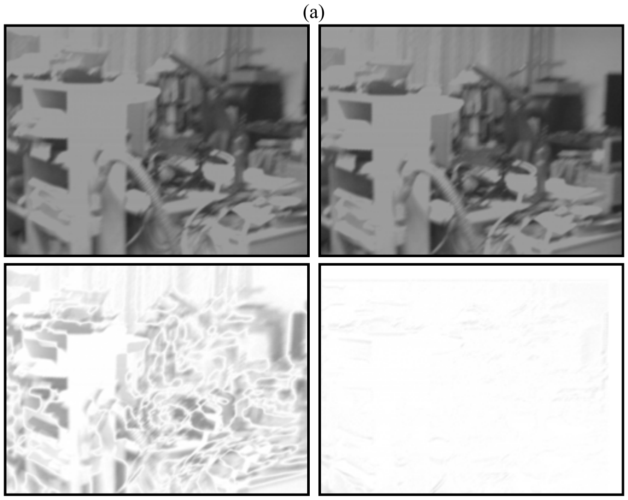
$$a_{11} = 1.0530 \qquad a_{22} = 1.0593$$
$$b_1 = 17.9097 \qquad b_2 = -18.9619$$

The intensity difference is calculated by $\frac{1}{N}\sum_{x,y}\left|I_k(x', y') - I_{k-1}(x, y)\right|$, where N is the number of calculated pixels in the IFD. In Fig. 3-(d), the intensity difference is 1.36.

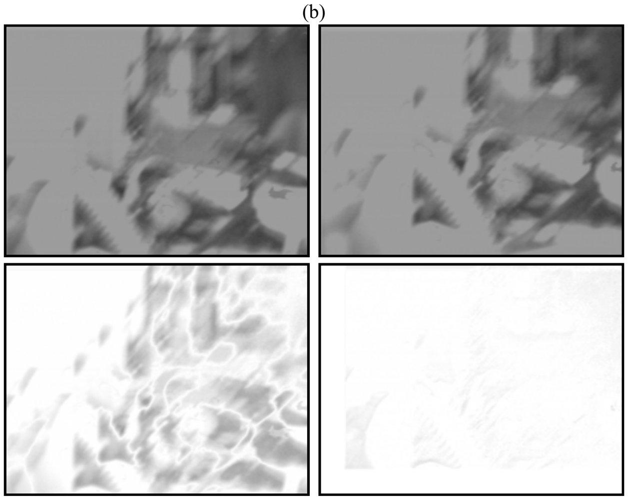
Fig. 4 shows successive images. Two successive images in Fig. 4-(c) are significantly damaged by blur. These successive images are captured while the camera is rotating and changing its zoom values. The pan/tilt speed of the camera is 34 deg/sec and the zoom speed is set to 4. The camera rotates continuously. When it reaches its limit, it reverses course, rotating in the opposite direction. Fig. 4 also shows their corresponding IFDs. Table I shows the parameters of the transformation and intensity differences determined by the method. The intensity difference is large if images are damaged by blur (Table I). The intensity difference in Fig. 4(c) is largest, the images are significantly damaged by blur.

Fig. 5 shows an example of a person who moves in the room. Fig. 5-(a) and 5-(b) show two successive images. Fig. 5-(c) and 5-(d) show the IFD without and with background compensation and the shape of the moving person.

(a)



(b)



(c)

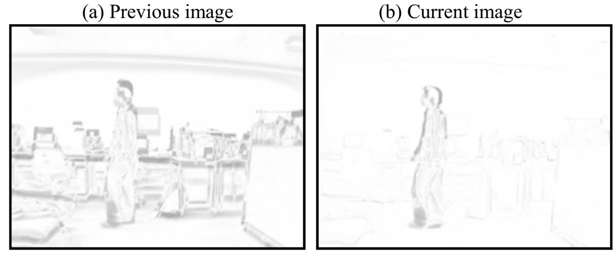
Fig. 4. Successive images and corresponding IFDs

In the next experiment, we compared the result of the proposed method with those of other methods. Table II shows comparisons of the proposed method with other methods. Each image sequence used for the experiment contains 1500 images. The sequence of images was captured while the camera was rotating and changing its zoom. All pairs that satisfy (7) and (9) contain outliers. RANSAC [14] are used to estimate parameters of the lines that best fit all pairs that satisfy (7) and (9). We compared the method RANSAC against the proposed method, using the same set of pairs of points. In the RANSAC algorithm, the number of iterations was 1000 and the threshold used to identify a point that fits well was set to 2. The performance of the RANSAC method depends on the number of iterations. With the same set of pairs of points, the proposed method obtained superior results in all cases and the average time required to process one frame is smaller. In the method [1], the number of feature points was set to 50, block size and search area were 21×21 and 101×101, respectively. The number of iterations in this method was set to 100. The intensity variance was set to 32. In comparison with the method [1], Table II shows that the proposed method yields smaller average intensity differences in all cases. However, in the method [1], the average time for processing one frame was 1305.59ms, and it is 2058.15ms in the proposed method. The implementation of the algorithm, Motion2D, described in [4] is freely available, and can be downloaded at http://www.irisa.fr/vista/Motion2D/. We used the free version Motion2D which is implemented in Linux. The two models in Motion2D which are close to the approximation in this paper are MDL_AFF_ROT_NULL (ARN) and MDL_AFF_HYP2_NULL (AH2N).

TABLE I
DETECTED PARAMETERS AND COMPENSATION ERRORS OF CASES IN FIG. 4

Cases	Detected Parameters		Intensity difference
Fig. 4-(a)	$a_{11} = 1.0023$ $b_1 = 9.6542$	$a_{22} = 0.9995$ $b_2 = -14.9642$	1.25
Fig. 4-(b)	$a_{11} = 0.9861$ $b_1 = -22.8087$	$a_{22} = 0.9733$ $b_2 = 22.5630$	1.67
Fig. 4-(c)	$a_{11} = 1.0216$ $b_1 = 54.9477$	$a_{22} = 1.0238$ $b_2 = -55.9490$	2.27

(a) Previous image (b) Current image



(c) IFD without background compensation (d) IFD with background compensation

Fig. 5. Two successive images with a moving person and their IFDs

TABLE II
COMPARISONS WITH THE OTHER METHODS

Pan/Tilt Speed (deg/sec)	Zoom Speed	RANSAC		The Method [1]		Motion2D [4] ARN		Motion2D [4] AH2N		The Proposed Method	
		Intensity Difference	Time (ms)	Intensity Differenc e	Time (ms)	Intensity Difference	Time (ms)	Intensity Difference	Time (ms)	Intensity Difference	Time (ms)
14	1	2.39	8096.26	2.86	1295.56	0.98	213.54	0.96	217.74	1.20	2223.21
14	2	2.54	7778.16	2.86	1295.42	1.10	249.44	1.08	259.06	1.37	2447.28
14	3	2.42	8333.51	2.89	1295.42	1.14	277.65	1.13	279.51	1.38	2196.27
14	4	2.54	7541.09	3.20	1295.36	1.17	249.89	1.15	257.66	1.39	2201.50
34	1	3.38	7642.20	4.03	1297.99	1.72	294.60	1.65	281.07	1.82	2133.93
34	2	3.65	7913.55	4.83	1348.22	1.91	319.48	1.83	310.93	1.95	2036.70
34	3	3.64	7791.81	4.75	1329.74	1.99	336.38	1.93	318.23	2.00	2066.81
34	4	4.21	7456.93	5.65	1321.70	2.42	363.67	2.34	352.41	2.28	2069.98
52	1	4.44	7780.31	6.26	1311.90	2.88	440.56	2.82	447.10	2.87	1908.20
52	2	4.09	8463.22	5.67	1311.28	2.47	409.17	2.40	393.07	2.49	1945.37
52	3	4.13	7938.56	5.68	1317.36	2.60	412.40	2.50	420.33	2.65	2046.97
52	4	4.37	7542.92	6.04	1301.10	2.82	449.06	2.75	428.19	2.88	1883.31
65	1	4.65	7700.49	6.93	1289.14	3.49	545.18	3.46	481.34	3.33	1853.76
65	2	4.53	7672.04	6.10	1289.61	2.99	465.16	2.91	442.72	2.84	1974.94
65	3	4.18	7882.39	6.31	1293.87	2.81	424.36	2.72	425.81	2.71	2027.01
65	4	4.53	7659.14	6.71	1295.81	3.28	485.12	3.24	477.63	3.19	1915.15
Average		3.73	7824.54	5.05	1305.59	2.24	370.98	2.18	362.05	2.27	2058.15

Table 2 shows that the average intensity difference and the average time for processing one frame of the Motion2D are smaller than those of the proposed method. However, at the pan and tilt speed of 65deg/sec, the proposed method gave the smaller intensity differences than Motion2D in two models ARN and AH2N.

IV. CONCLUSION

We have proposed a method by which the background motion represented by a scaling transformation followed by a translation between two successive frames can be determined using the Hough transformation. The proposed method provides better results than the method in [1], which uses corresponding points to determine the affine transformation parameters. The drawback of the method [1] is that it is difficult to correctly find the corresponding feature points when some areas contain similar images or when images are blurred due to camera motion. The proposed method does not find corresponding points to determine parameters for the transformation. However, the processing time of the proposed method is large, and it cannot be applied in real-time applications unless it is implemented in hardware. The main advantage of the proposed method with respect to [4] resides in the motion estimation in presence of importance motion blur. Background motion cannot be completely eliminated because of occlusion and motion blur. Background motion is approximated by a scaling transformation followed by translation. Therefore, IFD contains some noise. In particular, motion blur may significantly affect performance at high pan and tilt speeds.

REFERENCES

[1] S. Araki, T. Matsuoka, N. Yokoya, and H. Takemura, "Real-time tracking of multiple moving object contours in a moving camera image sequence", *IEICE Transactions on Information and System*, vol. E83-D, no. 7, pp. 1583-1591, 2000.

[2] Don Murray and Anup Basu, "Motion tracking with an active camera", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 449-459, 1994.

[3] Q. Cai, A. Mitiche and J. K. Aggarwal, "Tracking human motion in an indoor environment", *Proc. ICIP*, vol.1, pp.215-218, 1995.

[4] J. Odobez and P. Bouthemy, "Robust multi-resolution estimation of parametric motion models in complex image sequences", *Proc. 7th Eusipco, Edinburgh, Scotland,* 1994.

[5] Paul V. C. Hough, "Method and means for recognizing complex patterns", *U.S. Patent 3069654*, 1961.

[6] R.O. Duda, P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures", *Communications of the ACM*, 15, pp. 11-15, 1972.

[7] M. Atiquzzaman, "Multi-resolution Hough transform - An efficient method of detecting patterns in images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1090-1095, November 1992.

[8] M. Atiquzzaman, "Pipelined implementation of the multi-resolution Hough transform in a pyramid multiprocessor", *Pattern Recognition Letters*, vol. 15, no. 9, pp. 841-851, 1994.

[9] M. Atiquzzaman, "Coarse-to-Fine Search Technique to Detect Circles in Images", *The International Journal of Advanced Manufacturing Technology* , vol. 15, no. 2, 1999.

[10] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning", *Proc. ECCV*, pp. 189-196, 1994.

[11] T.J. Ellis, P. Rosin, and P. Golton, "Model-based vision for automatic alarm interpretation", *IEEE Aerospace and Electronic Systems Magazine*, vol. 6, no. 3, pp. 14-20, 1991.

[12] W. Niblack, D. Petkovic, "On improving the accuracy of the Hough. Transform", *Machine Vision and Applications*, vol. 3, pp. 87–106, 1990.

[13] W. H. Press, S.A. Teukolsky, W. T. Betterling, and B. P. Flannery, "Numerical Recipes in C", 2nd ed., *Cambridge University Press*, 1992.

[14] D.A. Forsyth and J. Ponce, "Computer Vision A Modern Approach", *Prentice Hall*, 2003.