Seyit TUNÇ, Hakkı Alparslan ILGIN

Ankara University

Motion Estimation and Compensation using Video Object Tracking

Abstract. *Motion estimation is one of the most important processes of video compression standards that affect the quality and compression ratio of encoded video stream. In this paper we present a motion estimation algorithm, which is based on ASIFT method. We segment foreground objects into regions by exploiting features which are obtained on the object. Motion model of a region is determined according to the number of features in the region. Obtained motion model is applied on the region of objects and if holes are occurred, they are closed by bilinear interpolation. Motion compensation results of our method are compared with the most common motion estimation methods used in video coding standards.*

Streszczenie. *W artykule przedstawiono algorytm estymacji poruszenia w obrazie wideo, bazujący na metodzie ASIFT. Obiekt pierwszoplanowy zostaje podzielony na regiony, z których każdy otrzymuje model ruchu, uzależniony od nagromadzenia ogólnych cech obiektu. Występujące w modelu luki uzupełniane są za pomocą interpolacji biliniowej. Otrzymane wyniki działania metody porównano z najpopularniejszymi metodami estymacji, wykorzystywanymi w kodowaniu wideo. (**Śledzenie obiektu w obrazie wideo w estymacji i kompensacji położenia**).*

Keywords: Object based motion estimation and compensation, video objects, parameter estimation, affine and perspective transformation.
Słowa kluczowe: estymacja I kompensacja obiektowa położenia, obiekty wide, parameter estymacji, transformacja afiniczna i perspektyw.

Introduction

Video compression techniques enable decrease in bandwidth occupied by the encoded video and hardware cost. High speed transfer by compressing video signal at low bit rates also provides a gain from time which is crucial for real time video applications. Additionally a video signal that uses a small storage area with a high compression rate is not desired to lose much information for watching pleasure. Therefore reducing the storage requirement, providing a high quality watch and transmitting at high speed by compressing at low bit rates are very important for video coding.

Motion estimation is the most important process used in video compression for reducing temporal redundancy of video sequences. It is used in entire coding standards, from the first video coding standards, such as H.261, to state of the art H.264. While in MPEG-1 [1], MPEG-2 [2], MPEG-4 [5], H.261 [3], H.263 [4] and H.264, [6] frame based motion estimation is used, in MPEG-4, motion estimation for arbitrarily shaped objects are also supported [13].

There are two major factors which determine the success of a motion estimation algorithm. First of them is a low computation time because it is important for real-time coding; and the second one is a high estimation ability since it is important for increasing compression ratio. The method which we proposed focuses on high precise estimation.

At the second section, general information for motion estimation used in video coding standards is given. Then proposed motion estimation and compensation method is explained. At the fourth section the proposed method and frequently used motion estimation methods are compared. Finally, results are given at the last section.

Motion estimation used in video coding

Most of the progress made in video compression is obtained by improvements in the estimation of motion which are used in video compression algorithms [7]. In video compression standards like H.261, only full pixel motion estimation, and in the MPEG-1 and MPEG-2, which is developed later, a half pixel motion model for making better estimations is used [3]. To obtain a more sensitive motion resolution, in MPEG-4, quarter pixel motion estimation is utilized and with a transformation, for warping of wide regions of frame, a global motion estimation is also applied [14]. In H.264, motion prediction from multi reference frames is used. Also unlike the previous video compression standards which supports only one motion, a multi motion support inside the macro block is added [6].

Block Based Motion Estimation

Block based motion estimation is a technique to compute motion of rectangular pixel regions between consecutive frames in MPEG-1, MPEG-2, MPEG-4 and in H.264 [15]. In MPEG-1 and MPEG-2, fixed size square blocks at 16x16 pixels, called macro-block, is used. In addition to the blocks, motion vectors can be computed for arbitrarily shaped regions in MPEG-4. Thus, representation of the movement of an object by only one vector is provided. In H.264, the block size is not limited at 16x16 pixels dimensions, and can be at 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 pixels dimensions. In this way, the blocks can align to the regions, which consist of motion discontinuities, like edges of a moving object.

Object Based Motion Estimation

Object based video coding used in MPEG-4 provides high coding efficiency as well as opportunity to perform content based applications because it relies on shape information of moving objects [14]. In object based motion estimation, frames are segmented into moving foreground objects rather than splitting into the blocks. Motion of each object is estimated independently from the others [8, 16].

Proposed motion estimation and compensation method

In the proposed method, corresponding point pairs are obtained on foreground objects in consecutive frames for motion estimation. Extracted corresponding point pairs are clustered and these clustered pairs generate the seeds of the regions which will be segmented on the objects. With the help of the clustered corresponding point pairs, regions arise on the object. According to the number of corresponding point pairs, motion transformation of a region which will be applied is determined. When motion transformation for each region is determined, motion estimation for foreground object is obtained.

Estimated motion transformation for each region, is applied to the foreground object at the reference frame and the holes emerged are filled by interpolation. Thereby, compensation of the motion estimation is provided.

Segmentation of the Regions on the Objects

General block diagram for segmentation of the regions on the object is shown in Fig. 1. By using ASIFT on the moving objects at two consecutive frames, corresponding point pairs are extracted [9] as shown in Fig. 2 (a). Extracted corresponding points on the foreground object are shown in Fig. 2 (b).

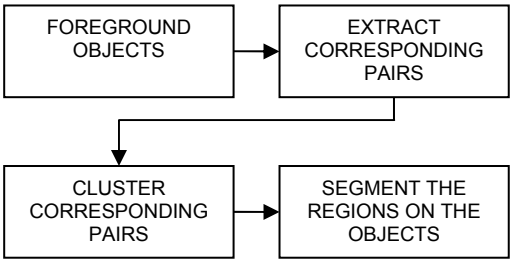

Fig. 1. Block diagram for segmentation.

After obtaining corresponding point pairs, they are clustered by using K-Means Clustering [10] algorithm, as shown in Fig. 2 (c). Then by using a method such as Region Growing [11], pixels around each point pair are joined to the cluster of those point pairs iteratively. Although this method is similar to region growing algorithm, similarity between intensity values of adjacent pixels is not taken into consideration in this method. By using this method, regions where motion will be estimated on the foreground object are segmented. Segmented regions are shown in Fig. 2 (d).

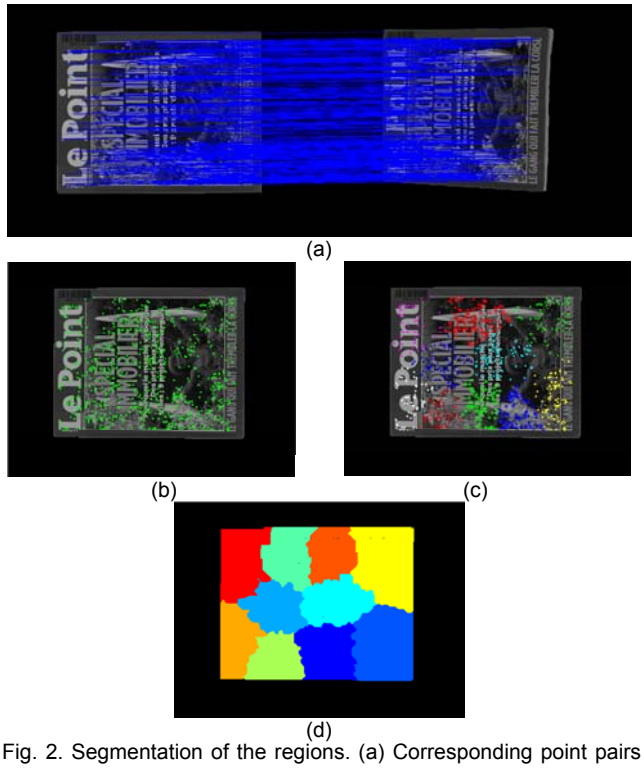

Fig. 2. Segmentation of the regions. (a) Corresponding point pairs on the foreground object at two consecutive frames (b) Corresponding points on the foreground object (c) Pairs clustered by k-means clustering. (d) Regions segmented on the object.

Motion Models of the Regions

After foreground object is segmented to the regions, depending on the number of corresponding points in the region, the motion model of the region is determined. For example, to find unknowns m (translation at x direction) and n (translation at y direction) at the translation motion model in (1), only one corresponding point in the region is enough because every single corresponding point pair ((x, y), (x', y')) gives us two equations:

$$x' = x = m;\ y' = y + n \tag{1}$$

If another pair exists in the region, anisotropic scaling and translation motion will be as follows

$$\begin{aligned} x' &= ax + m \\ y' &= by + n \end{aligned} \tag{2}$$

or rotation, isotropic scaling and translation motion will be modeled as in (3).

$$\begin{aligned} x' &= a\cos\theta x - a\sin\theta y + m \\ y' &= a\sin\theta x - a\cos\theta y + n \end{aligned} \tag{3}$$

If there is also a third corresponding point in the region, affine motion model in (4) can be calculated [17].

$$\begin{aligned} x' &= ax + by + m \\ y' &= cx + dy + n \end{aligned} \tag{4}$$

If there exist four or more corresponding points in the region, in addition to translation, rotation and scaling, a perspective motion model including perspective is given as follows [18]:

$$\begin{aligned} x' &= (ax + by + m) / (px + qy + 1) \\ y' &= (cx + dy + n) / (px + qy + 1) \end{aligned} \tag{5}$$

Motion Estimation

Motion models given by (1)-(4) are rearranged as below to estimate motion:

$$P' = PT \tag{6}$$

Here, P is matrix of points transformation will be applied on, T is transformation and P' is obtained matrix after transformation is applied. Since it is not possible to write translation motion in matrix form like (6), a third component (one) is added to the point pairs ((x, y), (x', y')) as follows [25].

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} \tag{7}$$

To solve affine motion model given by (4), (7) is rewritten as (6):

$$\begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \left[\begin{array}{cc|c} a & b & 0 \\ c & d & 0 \\ \hline m & n & 1 \end{array}\right] \tag{8}$$

When a is 1, b is 0, c is 0 and d is 1, only translation motion model is obtained. Translation motion is also determined by leaving m and n in (1). Affine motion estimation is done by multiplying (6) by the inverse of the matrix P as follows:

$$P^{-1}P' = P^{-1}PT = T \tag{9}$$

SVD [12] is used to compute the inverse of the matrix P.To solve motion model in (5), we rewrite for n amount of corresponding point pairs; in matrix form

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \end{bmatrix} \begin{bmatrix} a \\ c \\ m \\ b \\ d \\ n \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

is obtained. This equation is solved by homogeneous linear least squares to find transformation coefficients since it is in $\boldsymbol{Ax} = 0$ form in general. Transformation coefficients can be also calculated by using SVD in this approach. By definition, the matrix $\boldsymbol{A}$,is written in the following form

$$\mathbf{A} = \mathbf{U}\sum\mathbf{V}^T = \sum_{i=1}^{9}\sigma_i\mathbf{u}_i\mathbf{v}_i^T \quad (11)$$

The last column of matrix $\boldsymbol{V}$ ($\boldsymbol{v_9}$) is the right singular vector which corresponds to the least singular value. This is the vector which makes the left hand side of the equation (10) converge mostly to zero. Thus, $\boldsymbol{v_9}$ becomes the best solution vector for (10):

$$\mathbf{v}_9 = \begin{bmatrix} a \\ c \\ m \\ b \\ d \\ n \\ p \\ q \\ r \end{bmatrix} \quad (12)$$

If $\boldsymbol{v_9}$ is reshaped,

$$\mathbf{T} = \left[\begin{array}{ccc} a & b & p \\ c & d & q \\ \hline m & n & r \end{array}\right], \quad (13)$$

which is the perspective transformation matrix, is obtained.

Motion Compensation

The steps of motion compensation are shown in Fig. 3. Motion transformation calculated for each region on the foreground object is applied to that region. After applying the transformation, especially in cases where the object gets closer to the camera, holes can occur on the object. In such cases, the holes are filled by bilinear interpolation since its complexity is lower and it gives adequate results.

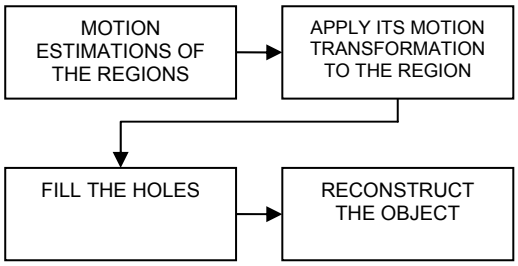

Fig. 3. Block diagram for motion compensation.

Experimental Results

The achievement of the proposed method is compared with block based motion estimation methods, which are Exhaustive (Full) Search (ES), Three Step Search (TSS), New Three Step Search (NTSS) [19], Simple and Efficient Three Step Search (SESTSS) [20], Four Step Search (FSS) [21], Diamond Search (DS) [22] and Adaptive Rood Pattern Search (ARPS) [23]. Small car video sequence taken by fixed camera with 800x582 resolution is used to test achievement of the proposed method for translational motion of small objects. In this test, according to the full frame, a small object (car) whose size is about 150x60 pixels moves in the orthogonal direction of the camera's direction. The proposed motion estimation method and block based motion estimation methods are applied and objective video quality is acquired by PSNR (Peak Signal to Noise Ratio). 16x16 pixels block size and +/-7x7 pixels search region are used for block based motion estimation methods. 8x8 pixels block size is also used for ES which gives the best result among block based motion estimation methods. 14th and 15th frames of the small car video sequence are shown in Fig. 4 (a) and (b), respectively. PSNR of the frame, compensated by ES method using 8x8 pixels block size, shown in Fig. 4 (c) is 32.3038 dB. PSNR of the frame, compensated by the proposed method, is obtained as 36.0549 dB.

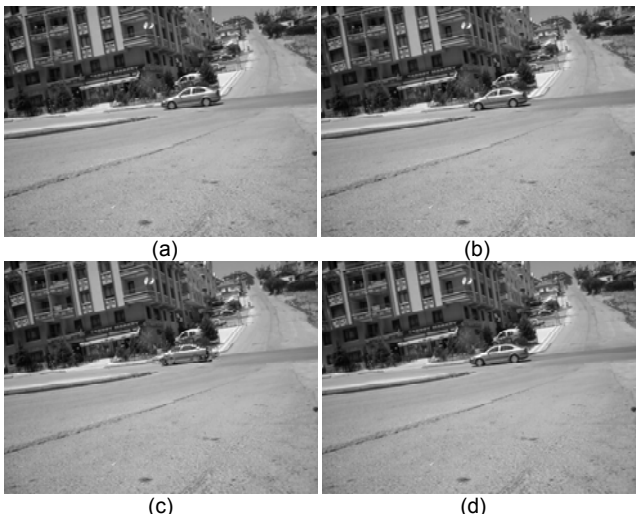

Fig. 4. Frames of the translation test video for small object. (a) Original 14th frame (b) Original15th frame (c) Compensated frame by ES method with 8x8 pixels block size (PSNR=32.3038 dB) (d) Compensated frame by proposed method (PSNR=36.0549 dB).

PSNR results acquired by different methods for consecutive 30 frames of small car video sequence are shown in Fig. 5.

Since the car moves slower at the first five frames than the other frames in the test video, all motion estimation methods give almost equal results. In the other frames, where the car moves fast, it is seen that the proposed method (PM) gives better performance than the other methods. The main reason of this achievement is that motion estimation is independent from speed of object motion in the proposed method. Average PSNR results calculated for each method in the small car video sequence are shown in Fig. 6. It is seen from Fig. 6 that the proposed method (PM) gives better result approximately 3.5 dB than ES with 16x16 pixels block size and approximately 1.5 dB than ES with 8x8 pixels block size.

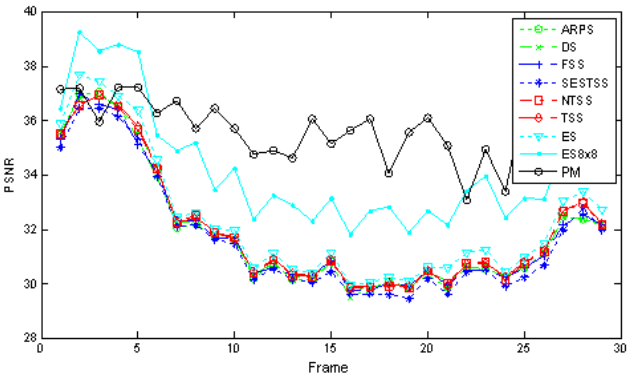

Fig. 5. PSNR (dB) comparison for small object translation test video

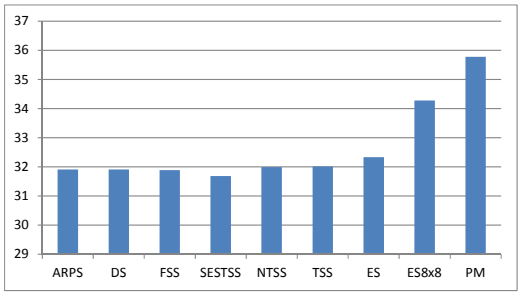

Fig. 6. Average PSNR (dB) values for each methods in the translation test for the small object.

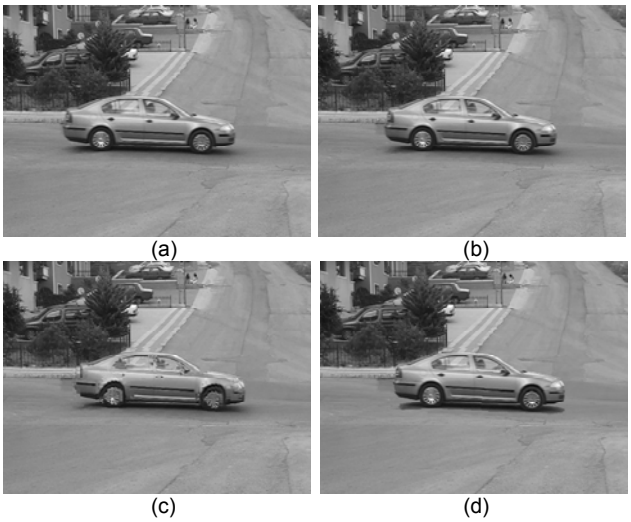

Fig. 7. Frames of the translation test video for the big object. (a) Original 10th frame (b) Original 11th frame (c) Compensated frame by ES method with 8x8 pixels block size (PSNR=29.6773 dB). (d) Compensated frame by proposed method (PSNR=31.0734 dB).

Big car video sequence taken by fixed camera is also used to test success of the proposed method for translational motion of big objects. It differs from the other translational motion test sequence with the size of the object seen in the frame. In this translational motion test, a big sized car approximately 360x120 pixels according to the full frame 640x480 moves in the orthogonal direction of the camera's direction.

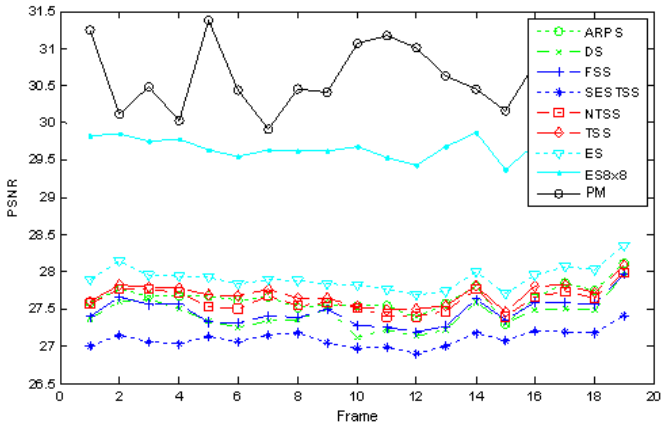

Fig. 8. PSNR (dB) values obtained in the translation test video for the big object.

The proposed motion estimation method and block based motion estimation methods are applied to the big car video sequences. 16x16 pixels block size and +/-7x7 pixels search region are used for block based motion estimation methods. 8x8 pixels block size is also used for ES which gives the best result through block based motion estimation methods. 10th and 11th frames of the big car video sequences are shown in Fig. 7 (a) and (b) respectively. PSNR of the frame, compensated by exhaustive search method fulfilled using 8x8 pixels block size, shown in Fig. 7 (c) is 29.6773 dB. PSNR of the frame, compensated by the proposed method, is obtained as 31.0734 dB. PSNR results comparisons for consecutive 20 frames of big car video sequence are shown in Fig. 8.

Since the speed of the car is constant, PSNR values of the block based motion estimation are not hugely different between the frames as shown in Fig. 8. It is seen that the proposed method (PM) gives better performance than the other methods. The main reason of this achievement is that motion estimation is independent from speed of object motion in the proposed method. Average PSNR results obtained for each method in the big car video sequence are shown in Fig. 9.

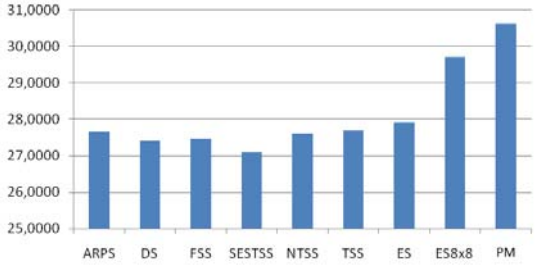

Fig. 9. Average PSNR (dB) values for each methods in the translation test for big object.

It is seen from Fig. 9 that the proposed method (PM) gives better result approximately 3 dB than ES with 16x16 pixels block size and approximately 1 dB than ES with 8x8 pixels block size. Both of the translational motion tests with big and small sized objects proved that performance of the proposed method is independent from object to frame ratio in size manner.

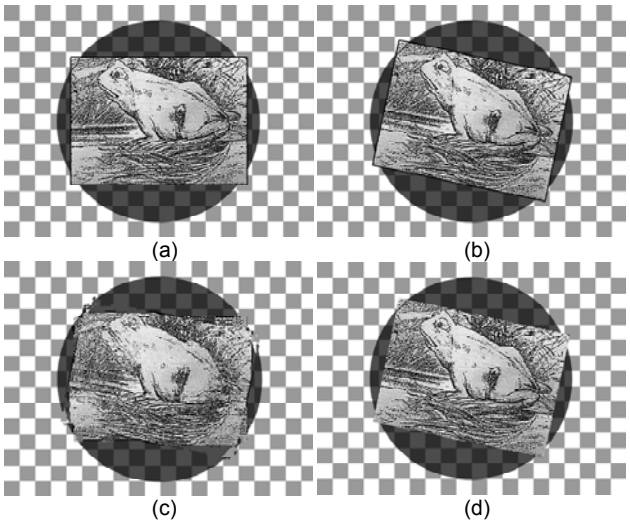

Fig. 10. Images of the rotation test video (a) The first frame (b) The second frame (c) Compensated frame by ES method with 8x8 pixels block size (PSNR=16.9524 dB) (d) Compensated frame by the proposed method (PSNR=22.3752 dB).

A test video is also prepared with an illusion image [24] in 784x592 resolution to test the proposed motion estimation method's performance for rotational movements. When the image is rotated 90° angle clock-wise, a horse changes to a frog. The proposed motion estimation method and block based motion estimation methods are applied to 30 consecutive frames. 16x16 pixels block size and +/-7x7 pixels search region are used for block based motion

estimation methods. 8x8 pixels block size is also used for ES which gives the most successful result among block based motion estimation methods. The first and the second frames of the rotation test video are shown in Fig. 10 (a) and (b), respectively.

PSNR results obtained by different methods applied in consecutive 30 frames of the rotation test video are shown in Fig. 11.

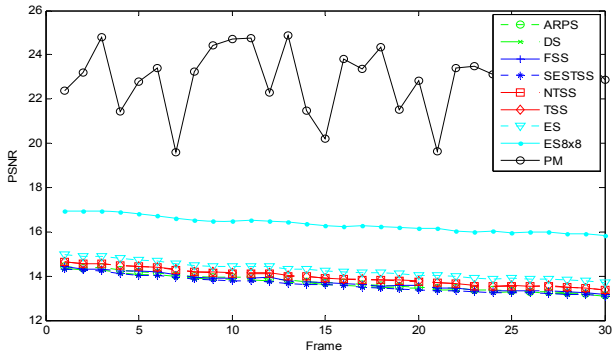

Fig. 11. PSNR (dB) values obtained in the rotation test video.

When the PSNR results obtained in the rotation test video are examined, it is seen that the proposed method is more successful than the other methods. The details are also clearly seen in the second compensated frame using the proposed method. The main reason of this performance is that object motion is also modelled by affine transformation in the proposed method. On the other hand, only translation motion is considered by default in the other motion estimation methods. Average PSNR results acquired for each method in the rotation test are shown in Fig. 12.

It is seen from Fig. 12 that the proposed method (PM) gives better result approximately 8.5 dB than ES with 16x16 pixels block size and approximately 6.5 dB than ES with 8x8 pixels block size.

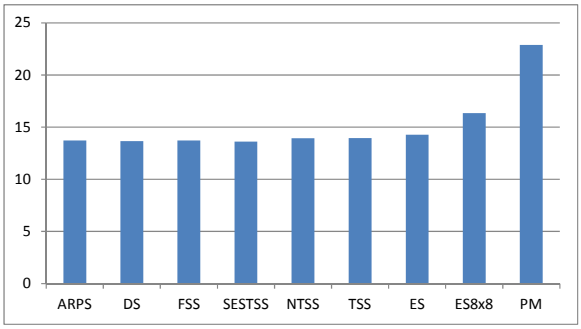

Fig. 12. Average PSNR (dB) values for each method in the rotation test.

Conclusion

An object based motion estimation method is proposed for videos taken by fixed cameras in order to increase the coding efficiency. Motion estimation is performed using silhouette of objects in the proposed method. Also motion of objects is modelled by scaling, rotation and perspective together with translation. Estimation is independent from speed and motion model of moving objects. As seen from the results that the proposed method gives better results than the block based motion estimation methods.

Based on the experiments, our method achieves higher PSNR values for the video sequences which have different motion properties such as rotation and translation since it considers not only translation information but also rotation, anisotropic scaling, isotropic scaling and perspective.

REFERENCES

[1] ISO/IEC 11172, Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, (1993)

[2] ISO/IEC 13818, Information technology: generic coding of moving pictures and associated audio information, (1995)

[3] ITU-T Recommendation H.261, Video CODEC for audiovisual services at px64 kbit/s, (1993)

[4] ITU-T Recommendation H.263, Video coding for low bit rate communication, Version 2, (1998)

[5] ISO/IEC 14496-2, Coding of audio-visual objects – Part 2: Visual, (2004)

[6] ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding, (2003)

[7] M. Ghanbari, Image Compression to Advanced Video Coding, UK, Hertz, (2003)

[8] M. Phadtare, Motion Estimation Techniques in Video Processing, *Electronic Engineering Times India*, (2007)

[9] J. M. Morel and G. Yu, ASIFT: A New Framework for Fully Affine Invariant Image Comparison, *SIAM Journal on Imaging Sciences*, (2009), vol. 2, issue 2

[10] S. P. Lloyd, Least squares quantization in PCM. *IEEE Transactions on Information Theory*, (1982), 129–137

[11] W. K. Pratt, Digital Image Processing 4th Edition, *John Wiley & Sons Inc.*, Los Altos, California, (2007)

[12] M. R. Hestenes, Inversion of Matrices by Biorthogonalization and Related Results, *Journal of the Society for Industrial and Applied Mathematics*, (1958), 51–90

[13] T. Sikora, The MPEG-4 Video Standard Verification Model, *IEEE Transaction on Circuits and Systems for Video Technology*, (1997), Vol. 7, No. 1

[14] M. C. Lee et al., A Layered Video Object Coding Sysem Using Sprite and Affine Motion Model, *IEEE Transaction on Circuits and Systems for Video Technology*, (1997), Vol. 7, No. 1

[15] I. E. G. Richardson, H.264 and MPEG-4 Video Compression, *Wiley*, (2003)

[16] Y. Yu, D. Doermann, Model of Object-Based Coding for Surveillance Video, *ICASSP*, (2005)

[17] G. Forsythe, M. Malcolm, and C. Moler, Computer methods for mathematical computations, *Prentice-Hall*, (1977)

[18] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, *Cambridge University Press*, (2003)

[19] Li, R., Zeng, B. and Liou, M. L., A New Three-Step Search Algorithm for Block Motion Estimation, *IEEE Trans. Circuits And Systems For Video Technology*, (1994), vol 4., no. 4, pp. 438-442

[20] Lu, J. and Liou, M. L., A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation, *IEEE Trans. Circuits And Systems For Video Technology*, (1997), vol 7, no. 2, pp. 429-433

[21] Po, L. M. and Ma, W. C., A Novel Four-Step Search Algorithm for Fast Block Motion Estimation, *IEEE Trans. Circuits And Systems For Video Technology*, (1996), vol 6, no. 3, pp. 313-317

[22] Zhu, S. and Ma, K. K., A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation, *IEEE Trans. Image Processing*, (2000), vol 9, no. 2, pp. 287-290

[23] Nie, Y. and Ma, K. K., Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation, *IEEE Trans. Image Processing*, (2002), vol 11, no. 12, pp. 1442-1448

[24] http://www.naute.com/illusions/frog.php

[25] Rogers, D. F. and Adams, J. A., Mathematical Elements for Computer Graphics, *McGraw-Hill*, (1976).

***Authors**: Seyit TUNÇ, Assist. Prof. Dr. Hakkı Alparslan ILGIN, Ankara University, Electronics Engineering Department, Dögol Caddesi, 06100 Tandogan, Ankara, Turkey, E-mail: stunc@eng.ankara.edu.tr, ilgin@eng.ankara.edu.tr.*