# A Coder for Affine Transformation in H.263

M. Tabandeh, F. Marvasti, S.M. Bouzari

***Abstract:*** **Coding for video transferring in communication networks is essential because the digital representation of image sequences requires a very large number of bits. Similarity in image sequences causes transferring whole of image data to be unnecessary. The statistical analysis of video signals indicates that there is a strong correlation between successive picture frames as well as within the picture elements themselves. Many methods and coders to find this information have been proposed up to now. In this research we propose a new coder using Affine Transformation along with BMA technique for motion compensation in H.263 standard. In order to decrease the amount of computations in online communications, we have utilized new conditional motion estimation. Simulation results clearly prove that our method reduces bit-rate with respect to H.263 standard.**

***Keywords*:** **H.263, Affine Transformation, BMA, Long-term memory.**

## I. INTRODUCTION

In image sequence coding, the correlation between consecutive frames can be reduced by the motion compensation technique [1]. Motion compensation plays an important role in reducing bit-rate for transmission or storage of video signals. ITU-T H.263 [2] is a coding standard developed for low bit-rate video communication. In this standard, motion estimation is based on matching a block of pixels in the current frame against a similar size block of pixels in the previous frame. This technique is also called block matching algorithm (BMA). In a typical BMA, a frame of the picture is divided into either blocks of $M \times N$ pixels or square blocks of $N^2$ pixels. Then, for a maximum motion displacement of W pixels per frame, the current block of pixels is matched against a corresponding block at the same coordinate but in the previous frame, within the square window of (N+2W)× (N+2W) size.

This technique relies on the assumptions that the motion of objects is purely translational and that the illumination also is uniform, an assumption that can not always be true. In practice, any motion has a complex nature that can be decomposed into translation, rotation, shear, expansion and other deformation components as well as nonuniform illumination. To compensate for these nonuniform changes between consecutive frames, a deformed block of pixels in the current frame can be matched against a nondeformed block in the previous frame. The deformation should be such that all the components of the complex motion and illumination changes are included.

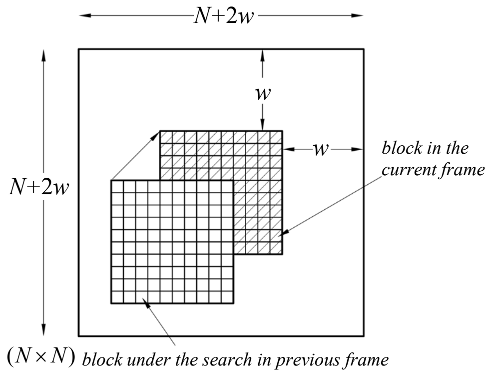



Fig. 1. Block Matching Algorithm (BMC)

The improved rate distortion performance of long-term memory motion-compensating prediction is shown by means of simulation results when integrating long-term memory prediction into an H.263 codec [3]. The computation time of task related to ME can be significantly reduced by fast search techniques, where mathematical inequalities are utilized that give lower bounds on norms of vector differences [4]. One example for these mathematical inequalities is the triangle inequality. Incorporating the triangle inequality into the SAD or SSD yields the following inequality for the distortion:

$$D(s,c) = \sum_{[x,y]\in B} |s[x,y] - c[x,y]|^p \geq \hat{D}(s,c) = \left| \left( \sum_{[x,y]\in B} |s[x,y]|^p \right)^{1/p} - \left( \sum_{[x,y]\in B} |c[x,y]|^p \right)^{1/p} \right|^p \quad (1)$$

Where $s[x,y]$ and $c[x,y]$ are the original and motion-compensated prediction signals respectively. We get by varying the parameter $p = 1$ for SAD and $p = 2$ for SSD. We have used these methods in our research.

We can view MCP as a source coding problem with a fidelity criterion closely related to VQ. Some methods have proposed that the criterion for the block motion estimation is the minimization of a Lagrangian cost function

$$J = D_{DFD} + \lambda_{Motion} R_{Motion} \qquad (2)$$

In which the distortion $D_{DFD}$ , representing the prediction error, is weighted against the rate $R_{Motion}$ associated with the motion parameters using a Lagrange multiplier $\lambda_{Motion}$. The Lagrange multiplier imposes the rate constraint, and its value directly controls the rate-distortion trade-off [5, 6, 7, 8 and 9]. However, these methods have a problem which due to computation complexity can not be used in online communication. To reduce computations we have proposed a new video codec.

## II. USED CODEC

Motion compensation efficiency can be further improved by Affine Transformation. In our research, in order to reduce computation complexity of finding Affine parameters, we propose a coder that chooses next step with respect to condition of current step. Schematic algorithm of proposed coder is illustrated in figure (2). BMA in comparison with Affine Transformation algorithm takes lower time; however, its bit error rate has a greater value. Therefore, we have used Affine Transformation with BMA in motion estimation and compensation sections and the following algorithm is proposed:

First step: Get a block of pixels from picture.

Second step: Use conditional motion estimation if block is not stationary go to Third step else go to First step.

Third step: Use Block matching algorithm to find motion vectors of block. If error is less than threshold go to First step else next step.

Fourth step: Find Affine Transformation parameters. Go to step one.

In the conventional block-matching motion estimation, for a given matching criterion, the number of search locations determines the required computation. A large number of search locations helps in finding a better match, but would require more computation and therefore more computation time. Thus, there is a tradeoff between the complexity and the coding efficiency in the motion estimation technique. In conditional motion estimation method, we can distinguish between moving and stationary blocks before using motion estimation to find motion vector's block. Puri [10] as well as Rath [11] used a motion detector with two thresholds. If the gray value difference between a target pixel and the pixel at the same location in the previous frame is greater than a threshold Tg, then the pixel is moving, else it is stationary. If the number of moving pixels in a target block is less than another threshold Tp, then the block is consider stationary, else it is supposed to be moving. If a block is stationary its MV equals zero, else it is necessary to search for the value of MV. In spite of the fact that most of the blocks in pictures with a non moving camera are either stationary or have simple movements the BMA search seems convenient to be used. However, for more complex motions, the use of Affine Transformation will be unavoidable. To achieve this aim, one threshold is defined as $T_{BMA}$ . If MAE obtained using BMA in first step is greater than this threshold, then Affine Transformation is used. The motion compensation in the core H.263 is based on one motion vector per macro block of 16 × 16 pixels, with half pixel precision. Finding half pixel precision with Affine Transformation requires a very large number of computations. For this reason, we used zero order interpolation instead of one or two order interpolation. Encoder structure has illustrated in figure (2).

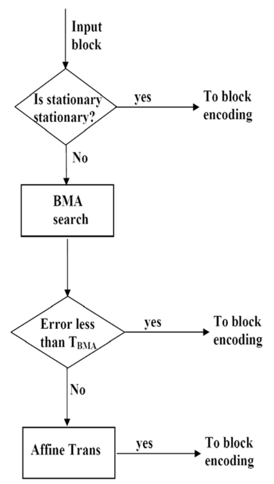



Fig. 2. Proposed Encoder

## III. AFFINE MOTION COMPENSATED PREDICTION

Motion estimation and compensation are parts of coder used to estimate motion vectors in images and find error of the estimation. They also have a capability of compensating for estimated images. In the proposed coder as mentioned above, combination of BMA and Affine Transformation has been used. Affine Transformation in motion estimation is utilized in academic centers in recent years, but usage of that is not common because of large amount of process requirement. This transformation is a mapping method remaining parallel manner of lines after applying; however, angles between lines are not kept. For such transformation, we need three points from block in present image to map to other three points in previous image. Based on these three points we can map other points or pixels of current block to point or pixels in previous frame. This transformation as shown in figure (3).

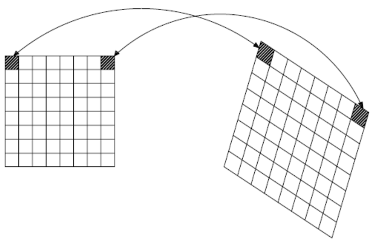


Fig. 3. An Affine Transformation

The commonly used motion compensation technique utilizes translational motion model, which assumes that all pixels within the considered block move with the same velocity. In reality, a rigid object motion in 3D space gives rise to more complex motions in the image plane, which cannot be compensated using only the translational model. Clearly, the assumption of constant velocity is only valid for small blocks overlapping a single moving object. Affine Transformation capable of estimating complex motion such as rotation, shear, expansion. How much the amount of block size grows, Affine Transformation is more effective than BMA in motion estimation of block. Affine Transformation can be expressed as a matrix operation:

$$\mathbf{p} = \mathbf{A}\,\mathbf{q} \qquad (3)$$

where **A** is the transformation matrix, **p** and **q** are pixels in predicted and source image respectively.

Affine motion model, which uses six parameters, these parameters are mentioned in relation (4). Affine Transformation can be expressed as:

$$\begin{aligned} U(x,y) &= a_1 x + a_2 y + a_3 \\ V(x,y) &= a_4 x + a_5 y + a_6 \end{aligned} \qquad (4)$$

where $u$ and $v$ are components of a pixel displacement vector with coordinates (x,y) and the coefficients $a_i$ are parameters that defined the transformation.

## IV. Results

In this section we examine the implementation of the proposed method and compare its performance with H.263 standard. Results presented in this section are obtained using six video samples. We utilize conditional coder with a window with fifteen pixels in length for each point search. In TSS algorithm, for a maximum motion displacement of $w$ pixels/frame, the total number of computations becomes $(5 + 4\log_2 w)$. For this reason, in order to find Affine parameters with maximum motion displacement of $w$ pixels/frame in Affine Transformation, the total number of computations becomes $(5+4\log_2 w)^3$, since three search points will be required. In this setting, the total number of computations for a picture is:

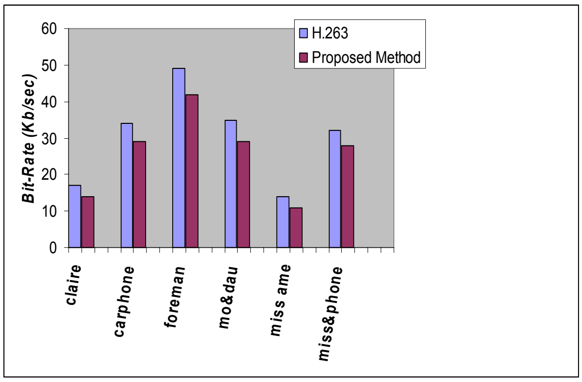
$$\begin{aligned} &K_{affine} \times (5+4\log_2 w)^3 + \\ &N_{frame} \times K_{BMA} \times (5+4\log_2 w) + K_{sta} \end{aligned} \qquad (5)$$

Where $K_{affine}$ , $K_{BMA}$ and $K_{sta}$ are number of macro blocks in picture to which Affine Transformation, BMA and stationary procedures were applied respectively. $N_{frame}$ is number of previous frame in memory. The proposed algorithm was tested using a set of test sequences ( “Claire” , ”Carphone” , ”Foreman” , ”Mother & daughter” , ”Miss America” , ”Miss & phone”). Table 1 shows the average of $K_{affine}$, $K_{BMA}$ and $K_{sta}$ in test sequences. In this experiment, each picture contains 99 macro blocks and $T_g$, $T_p$ and $T_{BMA}$ equal 5, 10, 1000 respectively. In this experiment we have utilized Long Term Memory method with 5 pictures in memory. In relation (5), the value of the coefficient $K_{affine}$ plays an important role since $(5+4\log_2 w)^3$ is a large number and should $K_{affine}$ increases slightly, the total number of computations will increase rapidly. The results show that Affine Transformation is applied on 41.03 macro blocks in average and therefore, the total number of computations decreased by 50 percent as compared with Affine Transformation without utilizing our proposed method.

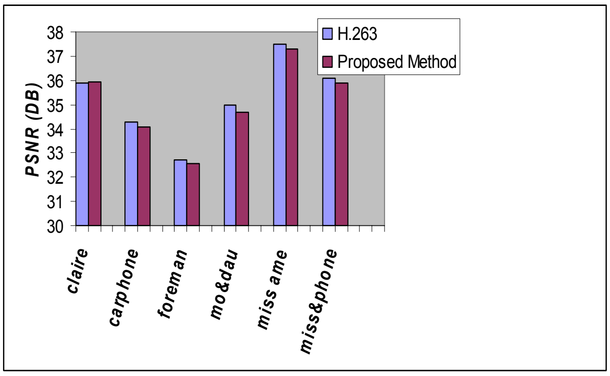
Table 1
AVERAGE OF PROPOSED PARAMETER

| Video samples | Coefficient | | |
|---|---|---|---|
| | $K_{Bili}$ | $K_{BMA}$ | $K_{sta}$ |
| Foreman | 68.6 | 90.1 | 99 |
| Claire | 19.05 | 35.95 | 99 |
| Mother & Daughter | 45.95 | 61 | 99 |
| Carphone | 51.2 | 74.5 | 99 |
| Miss America | 17.8 | 36.55 | 99 |
| Miss & phone | 43.6 | 67.9 | 99 |

As shown in figure 4(a), the bit-rate to transform video sequences in the proposed method is smaller than the bit-rate in the H.263 standard. We obtained an overall improvement of about 17.8% reduction in the bit-rate. That is why Affine Transformation has the ability of estimating the rotation, shear and translation. As shown in figure 4(b), PSNR becomes slightly less than H.263 standard using our proposed method. Generally, the PSNR (in average) in our method is 0.48% lower than for the case of H.263.




(a)




(b)

Fig. 4. Compare proposed algorithm with H.263 standard
(a) Bit Rate for video samples; (b) PSNR for video samples

## V. CONCLUSION

In our research, conditional motion estimation is applied in H.263 standard coder. We used three threshold parameters Tg, $T_p$, $T_{BMA}$ to switch between different conditions and steps. In our proposed method, we reduced bit-rate clearly and attempted to decrease computations for on-line communications. This idea, decreasing computations on Affine Transformation, hardly ever has been attended. Experimental results show that in our method bit-rate decrease by approximately 18% as compared with H.263 standard. In this research with five steps, each step has lower computation complexity compared to next step; however, next step has lower error bit rate. With change in threshold parameters we can choose complexity in computation and error bit rate.

## REFERENCES

[1] H.G. Musmann, P. Pirsch, and H.J. Grallert, "Advances in picture coding." *Proc. IEEE, vol.73*, pp.523-548, Apr.1985.

[2] Draft ITU-T Recommendation, H.263: "Video coding for low bit rate communication." July 1995

[3] T. Wiegand, X Zhang, and B. Girod, "Motion-Compensating Long-Term Memory Prediction", in *Proc.ICIP*, Santa Barbara, USA, Oct.1997.

[4] T. Wiegand, B. Lincoln, and B. Girod, "Fast search for long-term memory motion-compensated prediction", *in Proc. IEEE Int. Conf. Image Processing, Chicago*, IL, 1998, vol.3, pp.619-622.

[5] A. Puri, H.M. Hang and D.L. Schilling, "An efficient block-matching algorithm for motion compensated coding." *Proceedings of IEEE ICASSP*'87, 1987, pp.25.4.1–25.4.4.

[6] T. Wiegand, E. Steinbach, and B. Girod, "Long-Term Memory Prediction Using Affine Motion Compensation", in *Proc ICIP'99*.

[7] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-Constrained Vector Quantization", *IEEE Trans. ACOUSTIC.*, VOL. 37, NO. 1, JANUARY 1989.

[8] T. Wiegand, X. Zhang, and B. Girod, “Long-Term Memory Motion-Compensated Prediction”, *IEEE Trans.* Circuit and System, VOL.9, NO.1, FEBRUARY 1999.

[9] T. Wiegand, E. Steinbach, and B. Girod, “Affine Multipicture Motion-Compensated Prediction“, *IEEE Trans. Circuit and System*., VOL.15, NO.2, FEBRUARY 2005.

[10] A. Puri, H.M. Hang and D.L. Schilling, “Interframe coding with variable block-size motion compensation.” *In Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, 1987, pp.65-69.

[11] G.B. Rath, A.Makur, “Subblock matching-based conditional motion estimation with automatic threshold election for video compression.” *IEEE Transactions on Circuits and Systems for video Technology*. Volume 13, Issue 9, sept.2003, pp.914-924