IMPROVED INITIAL VALUE PREDICTION FOR GLOBAL MOTION ESTIMATION

Adel Ahmadi[1], *Hojjat Salehinejad*[1], *Siamak Talebi*[1,2], *and Faroukh Koroupi*[3]

[1]Department of Electrical Engineering, Shahid Bahonar University of Kerman
22 Bahman Blvd., 76169-133, Kerman, Iran
email: adel.ahmadi@graduate.uk.ac.ir, h.salehi@mail.uk.ac.ir

[2]Advanced Communication Research Institute, Sharif University of Technology
Azadi Ave., 11365-11155, Tehran, Iran
email: siamak.talebi@mail.uk.ac.ir

[3]Islamic Azad University - Baft Branch
Baft, Iran
email: faroukh.koroupi@yahoo.com

ABSTRACT

Global motion estimation (GME) algorithms have an imperative role in object-based applications. Gradient-based GME is a well known method among these algorithms. Such algorithms require an initial value for their initialization step. Well estimation of this value plays a significant role in the accuracy of GME. This work introduces a simple but efficient technique for initial value prediction of GME. This technique employs a long-term predictor as well as global motions of previous frames. Simulations results demonstrate faster convergence and less computational complexity of the proposed method versus common presented techniques in the literature with almost same efficiency.

1. INTRODUCTION

Motion estimation and motion compensation are two of the most important and widely used techniques in video coding systems. The motions are divided into local motion (LM) and global motion (GM) categories [11]. LMs are caused by moving, rotation, and deformation of objects while GMs are caused by panning, rotation, and zooming of camera. The local motion estimation (LME) and global motion estimation (GME) are techniques for estimation of LM and GM respectively. The LME and local motion compensation (LMC) techniques are used in discrete cosine transform (DCT) based video compression standards such as H.264 [1] and MPEG-2 [2] for reducing existent redundancy between macroblocks of two frames. In MPEG-4 verification model, the LMC technique as well as global motion compensation (GMC) technique is employed for reducing existent redundancy between motions. In this standard, by considering the sum of absolute difference (SAD) criterion, each macroblock can employ one of the LMC or GMC techniques [3].

Most common GME methods can be categorized in matching-based, feature-based, gradient-based and their hybrids [5], [15]. In the matching-based methods, the GM is estimated by using generalized block matching algorithm [4]. The features of two frames and techniques such as RANSAC [6] are used in GM estimation using the feature-based methods [7]. In the gradient-based methods, GM is estimated by utilizing gradient search techniques. These techniques can estimate GM with a reasonable accuracy rate but high computational complexity [8-14].

The gradient-based methods are divided into hierarchical and non-hierarchical approaches. In these methods, techniques such as Levenberg-Marquardt algorithm (LMA) or Gauss-Newton algorithm (GNA) are employed for minimizing sum of squared differences (SSD). These iterative algorithms require an initial value in global minimum “basin”. The less difference between initial value and global minimum, results in faster convergence of algorithms [16].

In hierarchical methods, the GME is mostly performed on three levels which are coarse, intermediate, and bottom levels [8-12]. In these methods, each frame is filtered and downsampled respectively. This process results in a pyramid with three levels. First, GME is applied between two frames in coarse level. The resulted GM is projected on intermediate level and is considered as the initial value for this level. After GM estimation in this level, the estimated GM is projected again on bottom level and is considered as the initial value for this level. Such hierarchical techniques reduce complexity and computational load of the system [8]. These methods need an initial value for coarse level. In [8] and [9] a modified three-step search is done between two frames [18], and the result is considered as initial value of this level. The three-step search algorithm provides a reasonable initial value. This algorithm is consisted of 25 SAD calculation steps and only can predict GM translation parts. Therefore, it is not applicable in rotation or zoom modes of camera. On the other side, [5] have suggested feature matching methods instead of the three-step search algorithm. Chan *et al.* in [12] have introduced a motion vector prediction for finding an initial value. They have suggested six predictors which are zero motion vector (MV), past MV, acceleration MV, historical average, historical maximum MV, and historical minimum MV. After projecting GMs of previous frames on coarse level, six vectors are constructed by translation parts of six predictors. These six vectors are constructed by assuming absence of rotation and zoom. Then, SADs of re-

 ISSN 2076-1465

sulted vectors are calculated and later, translation parameters of vector with the least produced SAD are selected. By considering resulted translation parts and also other elements of six predictors, six new vectors are constructed. By calculating SAD of these six GM vectors, vector with the least SAD is considered as the initial value. Further readings are available at [12]. Qi *et al.* in [10] and [11] suggest using the three-step search algorithm just for the first six frames and only employing four predictors for the remaining frames. By following up this idea, SAD calculations decrease up to eight times as well as having a less complex and more efficient method than [8] and [12]. In [13] and [14], some non-hierarchical methods are presented. The authors in [13] have employed three-step search algorithm between filtered and downsampled frames for estimating a coarse initial value. In [14] three-step search is used between sub-sampled frames for finding initial value.

In this paper, we suggest reducing initial value calculation by using a long-term predictor and employing a non-hierarchical GME. This method will reduce average number of iterations for algorithm convergence in a sequence and therefore, reduces computational complexity.

In section 2 of the paper, motion model, GNA and hierarchical GME are reviewed. The proposed combinatory algorithm is discussed in details in section 3 and simulation results on several sequences are presented and compared in section 4. Finally, the paper is concluded in section 5.

2. GLOBAL MOTION ESTIMATION

2.1 Motion Model

By considering the MPEG-4 verification model [3], the perspective model is employed for GM throughout this paper. This is due to the simplicity of models such as affine and translational where they are special cases of this model. The perspective motion model is defined as

$$x_i' = \frac{m_1 x_i + m_2 y_i + m_3}{m_7 x_i + m_8 y_i + 1} \tag{1}$$

$$y_i' = \frac{m_4 x_i + m_5 y_i + m_6}{m_7 x_i + m_8 y_i + 1} \tag{2}$$

$$\mathbf{m} = \begin{bmatrix} m_1 & m_2 & \cdots & m_8 \end{bmatrix}^T \tag{3}$$

where $\mathbf{m}$ is the GM vector, m_3 and m_6 are translation parameters and m_1, m_2, m_4, m_5 are rotation and zoom parameters. Also, m_7 and m_8 are perspective geometric distortion parameters.

2.2 Gauss-Newton Algorithm Based Global Motion Estimation

This algorithm is a common technique for minimizing sum of squared function. For the GME in this work, the error of pixel i is defined by

$$e_i = F_k(x_i, y_i) - F_{k-1}(x_i', y_i') \tag{4}$$

where x_i and y_i are coordinates of pixel i in frame k and x_i' and y_i' are correspondent pixel location in frame $k-1$.

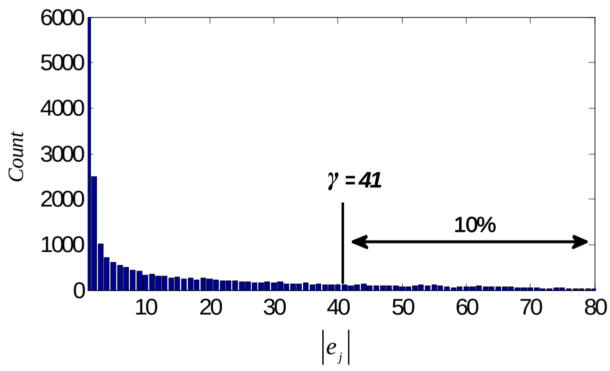

Figure 1 – Histogram of $|e_i|$ with the threshold γ that omits top 10% of the histogram.

The GM between two frames F_k and F_{k-1} is a vector which minimizes

$$E = \sum_{i=1}^{N} \rho(e_i) \tag{5}$$

where N is number of pixels and $\rho(e_i)$ is truncated quadratic error function defined as

$$\rho(e_i) = \begin{cases} e_i^2 & |e_i| \le \gamma \\ 0 & |e_i| > \gamma \end{cases} . \tag{6}$$

The γ is a threshold limit which reduces undesirable outliers effects of incorrect GM estimation. This threshold limit is set in the first iteration in a manner that omits 10% of pixels located above $|e_i|$ histogram throughout the algorithm as in Figure 1.

The iterative Gauss-Newton algorithm [8][16] is employed for minimizing (4) in this paper. In this algorithm, vector $\mathbf{m}$ is updated in each iteration t by

$$\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \Delta\mathbf{m}^{(t)} . \tag{7}$$

The updating term $\Delta\mathbf{m}^{(t)}$ is found by solving the following equation

$$\left[\mathbf{J}^T(\mathbf{m}^{(t)})\mathbf{J}(\mathbf{m}^{(t)})\right]\Delta\mathbf{m}^{(t)} = -\mathbf{J}^T(\mathbf{m}^{(t)})\mathbf{e}(\mathbf{m}^{(t)}) \tag{8}$$

where

$$\mathbf{e}\left(\mathbf{m}^{(t)}\right) = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix}^T \tag{9}$$

and $\mathbf{J}(\mathbf{m}^{(t)})$ is Jacobian matrix of error vector $\mathbf{e}(\mathbf{m}^{(t)})$ and is defined as

$$\mathbf{J}(\mathbf{m}^{(t)}) = \begin{bmatrix} \frac{\partial e_1}{\partial m_1} & \frac{\partial e_1}{\partial m_2} & \cdots & \frac{\partial e_1}{\partial m_8} \\ \frac{\partial e_2}{\partial m_1} & \frac{\partial e_2}{\partial m_2} & \cdots & \frac{\partial e_2}{\partial m_8} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N}{\partial m_1} & \frac{\partial e_N}{\partial m_2} & \cdots & \frac{\partial e_N}{\partial m_8} \end{bmatrix} . \tag{10}$$

In GNA, vector $\mathbf{m}$ is updated in each iteration using (8), (9) and (10), until a termination criterion is satisfied [8].

Totally, the GNA is employed in the GME algorithm as follows:

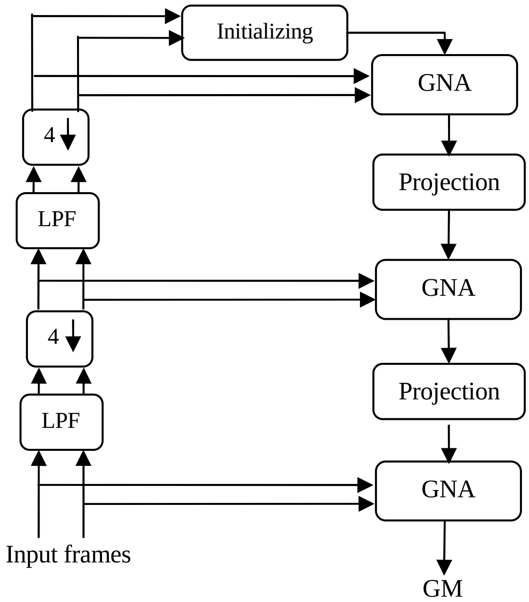

Figure 2 - Block diagram of the hierarchical GME.

1) The vector $\mathbf{m}$ is valued to a precomputed initial value.
2) The $\mathbf{J}(\mathbf{m})$ matrix and $\mathbf{e}(\mathbf{m})$ vector are calculated using $\mathbf{m}$. If $t=1$, the threshold limit γ is calculated by using the $|e_i|$ histogram. Otherwise, pixels with $|e_i| > \gamma$ in $\mathbf{J}(\mathbf{m})$ and $\mathbf{e}(\mathbf{m})$ calculation are omitted.
3) Equation (8) is solved to achieve $\Delta\mathbf{m}$.
4) The vector $\mathbf{m}$ is updated by (7). If $Number\ of\ iterations > N_{\max}$ or $\underset{\text{for all elements}}{\Delta\mathbf{m}} < \boldsymbol{\varepsilon}$ the algorithm ends, otherwise, it will be returned to the step 2.

2.3 Hierarchical Global Motion Estimation

The hierarchical GME structure is illustrated in Figure 2. In this structure, a Gaussian pyramid with three levels for two frames F_k and F_{k-1} is constructed. The calculations start from the highest level of the algorithm, which is called the coarse level, to the lowest level of the algorithm. In the first phase, a three-step search is done between the two frames in coarse level to achieve an estimation for translation elements m_3 and m_6. Then, the vector $\hat{\mathbf{m}}$ with $m_1 = m_5 = 1$, $m_2 = m_4 = m_7 = m_8 = 0$ and estimated m_3 and m_6 are considered as initial values for the GNA. Afterwards, the vector $\mathbf{m}$ is projected on the middle stage to be considered as the initial value for this stage. This projection is done by setting

$$m_3 = 2\times m_3,\ \ m_6 = 2\times m_6 \tag{11}$$

and

$$m_7 = \frac{m_7}{2},\ \ m_8 = \frac{m_8}{2}. \tag{12}$$

The calculations are continued in this stage and the resulted vector is projected on the lowest level to produce $\mathbf{m}$.

The computational load of GNA has a direct relation with number of pixels [13]. Therefore, if computational load of the lowest level with N number of pixels be 1, this load will be 0.25 and 0.125 in the middle and highest levels of algorithm, respectively.

3. PROPOSED METHOD

In video sequences each frame has a GM which demonstrates motions of camera. By considering the short time between successive frames ($1\,Sec\,/\,Frame\ Rate$), it is possible to assume that the camera motion is continuous within successive frames. Therefore, the GM is predictable using previous GMs of frames. In this case, it is possible to predict GM of a frame by using GMs of previous frames.

In [11] and [12], the resulted vectors from GME between previous frames are used for GM prediction of current frame. In fact, the achieved GM from the GME between two frames $k-1$ and $k-2$, demonstrates camera motion in frame $k-1$ with respect to the frame $k-2$. This vector is called short-term GM. The perspective motion model has a concatenation property [17]. In a way that if vector $\mathbf{m}_{k,k-1}$ be the GM vector between two frames k and $k-1$ and vector $\mathbf{m}_{k-1,k-2}$ be the GM vector between two frames $k-1$ and $k-2$, the GM vector $\mathbf{m}_{k,k-2}$ between two frames k and $k-2$ is achieved by concatenating $\mathbf{m}_{k,k-1}$ to $\mathbf{m}_{k-1,k-2}$. This property assists in calculating GM of each frame in respect to the first frame. Each $\mathbf{m}_{k,1}$ vector is named a long-term vector. These vectors provide better viewpoint of camera motions in respect to real environment along a sequence.

In our proposed method, we suggest employing long-term vectors in order to predict GM vector of frames. After the long-term calculation process, vector $\mathbf{m}_{k,k-1}$ is achieved by concatenating $\mathbf{m}_{k,1}$ to inverse of vector $\mathbf{m}_{k-1,1}$. In this work, we use acceleration predictor as

$$\hat{\mathbf{m}}_{k,1} = 2\mathbf{m}_{k-1,1} - \mathbf{m}_{k-2,1} \tag{13}$$

and the concatenating part as

$$Concatenating\left(\mathbf{m}^{-1}_{k-1,1}, \hat{\mathbf{m}}_{k,1}\right) \longrightarrow \hat{\mathbf{m}}_{k,k-1}. \tag{14}$$

Different steps of the proposed algorithm are as follow:

1) The first frame $k=1$ is encoded in intra-mode. For this frame $\mathbf{m}_{1,1}$ is defined as

$$\mathbf{m}_{1,1} = [1,0,0,0,1,0,0,0]^T. \tag{15}$$

2) The algorithm jumps to the next frame $k_{new} = k_{old} + 1$.
3) GM between two frames k and $k-1$ are calculated using hierarchical GME and three-step search algorithm.

4) Vector $\mathbf{m}_{k,1}$ is calculated by concatenation vector $\mathbf{m}_{k-1,1}$ to vector $\mathbf{m}_{k,k-1}$.

5) The algorithm jumps to the next frame $k_{new} = k_{old} + 1$.

6) Initial value of GM vector $\hat{\mathbf{m}}_{k,k-1}$ for frame k is predicted by (13) and (14).

7) By operation GNA with the predicted initial value in the stage 6, GM vector between two frames k and $k-1$ is estimated. If SSD of the estimated GM vector is more than the threshold $E_{\max}$, the algorithm is returned to the stage 3 to perform GME for the current frame again. Otherwise, the algorithm is returned to the stage 2.

The proposed algorithm not only decreases computational load and complexities, but also requires fewer number of iterations for convergence to a reasonable result. This is due to precise initial value prediction of our proposed method.

4. SIMULATION RESULTS

In this section, efficiency of the proposed method is compared with the presented methods in [8] and [11] for seven common sequences. These sequences are Carphone, Coastguard, Foreman, Mobile, Stefan, Table and Tempete. Frame dimension of the Carphone and Stefan sequences are 176×144 and 352×240 respectively and frame dimension of the other sequences is 352×288. For all the sequences except the Table, simulations are performed on the first 200 frames. Since in the Table sequence, scene changes from frame 132, simulations are performed on its first 131 frames. A computer with 4GB Ram, 2.66 GHz Core2Quad CPU, and MS Windows Vista operating system is employed for the simulations in the MATLAB environment.

In the simulations, the luminance of sequences is coded in the interframe mode (IPPP…) with fixed quantization size $Q=10$. The termination criteria $N_{\max}$ is set to 32 and $\boldsymbol{\varepsilon} = [0.001, 0.001, 0.1, 0.001, 0.001, 0.1, 0.001, 0.001]^T$, [8]. The simulations are performed against the presented hierarchical methods by Dufaux *et al.* and Qi *et al.* in [8] and [11] respectively. The hierarchical method in [8] is named H-TSS and has three levels where initial value is achieved by using a three-step search algorithm. The hierarchical method in [11] is named H-TPR and also has three levels. In this method, for the first six frames, initial value is reached by using a three-step search algorithm and for the rest of the frames, short-term GMs of previous frames are used as well as the four mentioned predictors in section 1.

In our proposed method, GME is performed on the frames without employing the hierarchical method and the initial value is achieved by using the described predictor in (13) and (14). In this method, for the frames that their achieved SSD from GME is more than the threshold limit $E_{\max}$, GM is performed again by the H-TSS method.

In the next subsection, number of necessary iterations for convergence of the mentioned methods is discussed and correspondent computational times are compared. Then, accuracy and peak signal-to-noise ratio (PSNR) and coding efficiency of predicted frames using the proposed method versus methods in [8] and [11] are evaluated in subsection 4.2.

4.1. Convergence time and number of iterations analysis

In the hierarchical methods, the GME is mostly performed on three levels which are coarse, intermediate, and bottom. Computational load of each iteration is correspondent to number of pixels [13]. Since number of pixels in the coarse and intermediate levels are 0.125 and 0.250 times that of the bottom level respectively, computational load of these levels are also 0.125 and 0.250 times that of the bottom level respectively. Based on this argument, in the simulations of the hierarchical methods H-TSS and H-TPR, each iteration in the coarse and intermediated levels are considered as 0.125 and 0.250 times that of an iteration in the bottom level respectively. For example, if in the hierarchical method we have 8, 4, and 3 iterations in the coarse, intermediate and bottom levels, number of iterations for the frame is considered as $\frac{8}{16}+\frac{4}{4}+3=4.5$. The average number of necessary iterations for each sequences frame convergence is presented in Table 1. Based on the results, the proposed method can reduce average number of necessary iterations for convergence to a percentage between 18% and 40%. In addition, the proposed method does not require constructing pyramid and calculating SAD for initial value estimation. As a result, this method is faster than proposed methods in [8] and [11]. Table 2 demonstrates GME computational time of the proposed method and the H-TSS and H-TPR methods for different sequences.

4.2. Peak signal-to-noise ratio & coding efficiency analysis

In order to evaluate accuracy and performance of the proposed method versus method in [8] and [11], the PSNRs of decoded frames luminance and also average size of encoded frames must be analyzed and compared. To do so, GME and LME are performed between two successive frames. Afterwards, for each macroblock from each frame, two macroblock predictions, one by using GM and the other by using LM, are made. The prediction with the less SAD is selected for the current frame reconstruction. PSNRs of the decoded frames are presented in Table 3. In most of the sequences, PSNRs are almost the same. The coding efficiency of these methods is also almost the same for different sequences, as shown in Table 4.

5. CONCLUSION

This paper introduces an efficient method for global motion (GM) initial value prediction. This technique has less computational complexity than common methods and also converges in fewer number of iterations. In addition, due to the reasonable estimation of initial value by the predictor, hierarchical pyramid is omitted in this technique, so that the global motion estimation (GME) is performed once between frames.

As the simulations results demonstrate, the proposed method in this work is 1.24 to 1.85 times faster than the proposed methods in [8] and [11], with almost similar accuracy and coding efficiency.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] Information technology—Generic coding of moving pictures and associated audio, ISO/IEC JTC1 IS 13818-2 (MPEG-2), 1994.

[3] “MPEG-4 video verification model version 18.0,” in: ISO/IEC JTC1/SC29/WG11 N3908, Pisa, Italy, 2001.

[4] W. C. Chan, O. C. Au, and M. F. Fu, “A novel predictive global motion estimation for video coding,”, *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol.3, pp.5-8, May, 2002.

[5] A. Smolic, J. R. Ohm and T. Sikora, “Object-based global motion estimation using a combined feature matching and optical flow approach,” in *Proc. VLBV’98*, Urbana, IL, pp. 165–168, Oct. 1998.

[6] M. A. Fischler, and R.C. Bolles, “RANSAC Random sample concensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Comm. ACM*, vol. 24, pp. 381-395, 1981.

[7] D. Farin, and P. H. N. de With, “Evaluation of a feature-based global motion estimation system,” in *SPIE Visual Communications and Image Processing*, pp. 1331–1342, Jul. 2005.

[8] F. Dufaux and J. Konrad, “Efficient, robust and fast global motion estimation for video coding,” *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 497–501, Mar. 2000.

[9] Hoi-Kok Cheung, Wan-Chi Siu, “Fast global motion estimation for sprite generation”, *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, pp. 1-4, May 2002.

[10] B. Qi, A. Amer, “Robust and Fast Global Motion Estimation Oriented to Video Object Segmentation”, *IEEE International Conference on Image Processing*, pp. 153-156, Sept. 2005.

[11] B. Qi, M. Ghazal, A. Amer, “Robust Global Motion Estimation Oriented to Video Object Segmentation,” IEEE Trans. Image Process., vol.17, no.6, pp.958-967, Jun. 2008.

[12] W. C. Chan, O. C. Au, and M. F. Fu, “Improved global motion estimation using prediction and early termination,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 285–288, Sep. 2002.

[13] H. Alzoubi and W. D. Pan, “Very fast global motion estimation using partial data,” *Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, HI, USA, pp. 1189–1192, April 2007.

[14] H. Alzoubi and W. D. Pan, “Efficient Global Motion Estimation Using Fixed and Random Subsampling Patterns,” in *Proc. of IEEE International Conference on Image Processing, San Antonio*, TX, pp. 477-480, September 2007.

[15] G. Ye, M. Pickering, and M. Frater, J.Arnold, “Arobust approach to super-resolution sprite generation,” in *Pro. of IEEE Int. Conf. Image Processing*, vol. 1, pp. 897–900, Sep. 2005.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C*, Cambridge Univ. Press, pp. 59–71, 1992.

[17] P. Heckbert, *Fundamentals of Texture Mapping and Image Warping*, Master’s Thesis, U.C. Berkeley, pp.17-21, 1989.

[18] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding of video conferencing,” in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, pp. G5.3.1–G.5.3.5, Dec. 1981.

TABLE 1 - Average number of necessary iterations per frame for first 131 and 200 frames of Table and other sequences respectively.

Sequence	H-TSS	H-TPR	Proposed
Carphone	3.24	3.25	2.65
Coastguard	4.53	4.55	3.65
Foreman	3.60	3.59	2.68
Mobile	2.92	2.94	1.76
Stefan	3.70	3.68	3.06
Table	4.26	4.31	2.56
Tempete	2.65	2.64	1.76
Avg.	3.56	3.57	2.59

TABLE 2 - Necessary time (Sec.) for GME of first 131 and 200 frames of Table and other sequences respectively.

Sequence	H-TSS	H-TPR	Proposed
Carphone	17.1	17.4	11.9
Coastguard	100.0	102.9	81.0
Foreman	81.3	83.6	63.8
Mobile	53.7	54.7	29.0
Stefan	58.0	58.3	44.7
Table	60.6	61.0	35.7
Tempete	56.1	57.4	33.6
Avg. Speedup	1.0	1.0	1.4

TABLE 3 - PSNR-Y average of each constructed sequence using the proposed method and the H-TSS and H-TPR methods.

Sequence	Without GMC	H-TSS	H-TPR	Proposed
Carphone	31.58	32.94	32.93	32.94
Coastguard	29.66	30.91	30.91	30.92
Foreman	31.05	32.32	32.32	32.32
Mobile	28.17	28.99	28.99	29.00
Stefan	29.34	30.45	30.44	30.46
Table	30.68	31.97	31.95	31.96
Tempete	29.13	30.32	30.32	30.32

TABLE 4 - Bitrate/Frame of compressed video frames (KByte).

Sequence	Without GMC	H-TSS	H-TPR	Proposed
Carphone	2.3284	2.3096	2.3077	2.3078
Coastguard	9.7573	9.2802	9.2785	9.2703
Foreman	9.5116	9.3556	9.3563	9.3615
Mobile	15.1698	13.1161	13.1186	13.1734
Stefan	11.5554	9.4996	9.5032	9.4872
Table	10.4538	9.8570	9.8584	9.8543
Tempete	10.8096	9.6818	9.6836	9.6876