Fast Two-Stage Global Motion Estimation: A Blocks and Pixels Sampling Approach

Adel Ahmadi[1], Farhad Pouladi[2], Hojjat Salehinejad[3], and Siamak Talebi[1,3]

[1] Electrical Engineering Department, Shahid Bahonar University of Kerman, Iran
[2] Scientific-Applied Faculty of Post & Telecommunications Ministry of Information and Communication Technology, Tehran, Iran
[3] Advanced Communication Research Institute, Sharif University of Technology, Tehran, Iran
adel.ahmadi@gmail.com, pouladi@ictfaculty.ir, {h.salehi,siamak.talebi}@mail.uk.ac.ir

Abstract. Global motion estimation (GME) is an important technique in image and video processing. Whereas the direct global motion estimation techniques boast reasonable precision they tend to suffer from high computational complexity. As with indirect methods, though presenting lower computational complexity they mostly exhibit lower accuracy than their direct counterparts. In this paper, the authors introduce a robust algorithm for GME with near identical accuracy and almost 50-times faster than MPEG-4 verification model (VM). This approach entails two stages in which, first, motion vector of sampled block is employed to obtain initial GME then Levenberg-Marquardt algorithm is applied to the sub-sampled pixels to optimize the initial GME values. As will be shown, the proposed solution exhibits remarkable accuracy and speed features with simulation results distinctively bearing them out.

1 Introduction

Motion estimation and compensation is one of the most essential techniques in video compression and processing. Motions in video are categorized into local motion (LM) and global motion (GM) [1]. The LMs are resulted from movement, rotation and reform of objects, while the GMs are due to movement, rotation, and camera zoom [2]. Global motion estimation (GME) has many applications such as video coding, image stabilization, video object segmentation, virtual reality and etc. In MPEG-4 standard, some techniques such as sprite coding and global motion compensation (GMC) are required for GME [3].

The common GME methods are divided into direct and indirect categories. In the direct category, which is pixel-based, prediction error is minimized by using optimization methods such as Levenberg-Marquardt algorithm (LMA) [1],[2],[4]-[7]. The indirect methods consist of two stages. In the first stage, motion vectors of blocks are calculated and by using these vectors, GM of frame is estimated in the second stage [8]-[14].

In MPEG-4 verification model (VM), GME is a direct type scheme where LMA is applied to the whole frame. Since LMA has high computational complexity, some methods have been devised by considering a limited number of pixels in the calculations. One such

G.A. Tsihrintzis et al. (Eds.): Intelligent Interactive Multimedia, SIST 11, pp. 143–151.
springerlink.com

technique is called FFRGMET that is used in MPEG-4 optimizing model. This technique just applies LMA to a number of pixels called feature pixels [15]. In [6], pixels are selected using gradient method. In this work, each frame is divided into 100 blocks and then 10% of pixels with the highest gradient are selected from each block. This procedure requires gradient calculations and pixels arrangement based on the gradients. Therefore, this method has a considerable computational complexity. The idea of random pixels selection is introduced in [16]. In spite of the method presented in [6], this technique has much lower computational complexity. However, random pixel selection causes numerical instabilities. In [4] and [5], pixels are selected based on a static pattern. In these papers, authors divide the frame into non-overlapped blocks and then select a few pixels with static pattern from each block. This method has low computational complexity and also does not cause numerical instabilities. In comparison to MPEG-4 VM, this scheme is faster with little accuracy degradation. An indirect GME for the affine model is proposed in [14]. In this study, firstly the amount of translation is estimated by using integral projection algorithm (IPA) and then based on that information a limited block-matching is performed for each sampled block.

In this paper, we have improved the proposed method in [14] and intend to use the perspective model. This is expected to achieve an improvement of peak signal to noise ratio (PSNR) at low complexity.

The reminder of this paper is organized as follows. The motion models are described in section 2 and in section 3, the proposed method including its different steps are discussed in details. The simulation studies are provided in section 4 and finally the paper is concluded in section 5.

2 Motion Models

The most comprehensive GM model in MPEG-4 is the perspective model. This model encompasses simpler models. This model is defined by:

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

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

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

where $\mathbf{m}$ is GM vector from current frame pixels (x_i, y_i) to reference frame pixels (x'_i, y'_i). This vector consists of translation parameters (m_3 and m_6), rotation and zoom parameters (m_1, m_2, m_4, and m_5), and perspective parameters (m_7 and m_8). Simpler models such as affine (with 6 parameters, $m_7= m_8=0$), Translation-Zoom-Rotation (with 4 parameters, $m_1=m_5$, $m_2=-m_4$, $m_7=m_8=0$), Translation-Zoom (with 3 parameters, $m_1=m_5$, $m_2=m_4=m_7=m_8=0$) and Translation (with 2 parameters, $m_1=m_5=1$, $m_2=m_4=m_7=m_8=0$) are special cases of perspective model.

3 Global Motion Estimation

The proposed algorithm consists of two stages. The first process calls for a rough estimation of GM. When this is obtained second stage takes place in which the initial estimation has to be optimized with greater precision. Structure of the proposed algorithm is as follows.

Stage I

- Estimating translation between two frames using IPA.
- Sampling blocks from the current frame as in Fig.1. Calculating motion vectors of sampled blocks using block matching (with shifted search centre and small searching range). Excluding 30% of blocks with maximum sum of absolute differences (SAD).
- Estimating eight parameters of GM vector using above motion vectors.

Stage II

- Sampling current frame pixels using 1:12×12 model as in Fig.2-d. Applying LMA to sampled pixels to optimize initially estimated GM of the first stage. The LMA iterations are continued until either of the following conditions is satisfied: reaching 10 iterations or updated term be lower than 0.001 for translational components *and* lower than 0.00001 for other components.

3.1 Initial Translation Estimation

In the first stage of GME, translation components must be estimated. In [1]-[5], a three-step search is used for this purpose. IPA is employed instead of a three-step search in our algorithm, because it is more accurate and robust [14].

To estimate translation between two frames, horizontal and vertical projection vectors are calculated as:

$$IP_k^{horiz}(y) = \frac{1}{M}\sum_{x=1}^{M} F_k(x, y) \tag{4}$$

$$IP_k^{vert}(x) = \frac{1}{N}\sum_{y=1}^{N} F_k(x, y) \tag{5}$$

where F_k denotes luminance of frame k and (M , N) are dimensions of frames. IP_k^{vert} and IP_k^{horiz} are integral projection values of F_k in vertical and horizontal directions respectively. By using the correlation between horizontal and vertical integral projection vectors of F_k and F_{k-1}, a translation value is calculated in vertical and in horizontal directions as below:

$$d_x = \min_{t=\{-s,s\}} \left\{ \sum_{x=1}^{M} (IP_k^{vert}(x) - IP_{k-1}^{vert}(x-t))^2 \right\} \tag{6}$$

$$d_y = \min_{t=\{-s,s\}} \left\{ \sum_{y=1}^{N} (IP_k^{horiz}(y) - IP_{k-1}^{horiz}(y-t))^2 \right\} \quad (7)$$

where (d_x,d_y) is translation of the current frame with respect to previous frame and s is maximum search range. The maximum search range is determined based on the size and contents of the video. To give some examples, s=8 for QCIF format and s=16 for CIF and SIF formats seems reasonable.

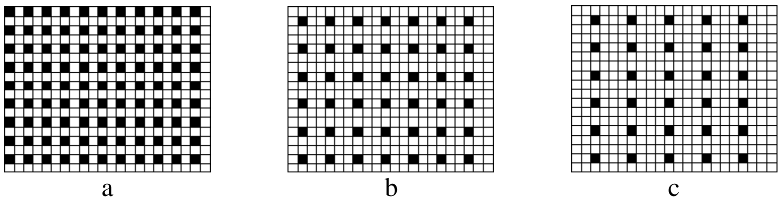

Fig. 1. Blocks sampling pattern [14]; a) 1:4; b) 1:9; c) 30:369

3.2 Block Sampling and Limited Block Matching

After translation estimation, one of the patterns in Fig.1 is employed for blocks sampling. Size of each block for different formats is considered as: 8×8 for QCIF, 16×16 for CIF and SIF and 32×32 for 4CIF. Then for each sampled block, a modified full search block matching algorithm (BMA) is obtained. In this search, the search centre is shifted (d_x,d_y) units and searching range is as small as (-3, +3). This results in less SAD computations and sufficient accuracy for motion vectors of background blocks. Since blocks with high SAD are mostly part of the foreground, 30% of them are excluded. The motion vectors of remaining blocks will be used in the next subsection.

3.3 Initial Estimation of Perspective Model GM Parameters

By considering (x_i , y_i) as central pixel coordinate of the current frame sampled block and (x_i' , y_i') as central pixel coordinate of the best matched block, we can have:

$$x_i' = v_{x,i} + x_i \quad (8)$$

$$y_i' = v_{y,i} + y_i \quad (9)$$

where $v_{x,i}$ and $v_{y,i}$ are motion vectors obtained from the previous step.

To find GM between two frames, we must minimize the Euclidean error:

$$E = \sum_{i=1}^{N_b} \left[\left(\frac{m_1 x_i + m_2 y_i + m_3}{m_7 x_i + m_8 y_i + 1} - x_i' \right)^2 + \left(\frac{m_4 x_i + m_5 y_i + m_6}{m_7 x_i + m_8 y_i + 1} - y_i' \right)^2 \right] \quad (10)$$

whrere N_b is number of blocks. Since the perspective model is nonlinear, (10) could be solved by using LMA which results in significant computational complexity. On the other hand, by using algebraic error definition [17], (10) can be modified as:

$$E = \sum_{i=1}^{N_b}\left[\left(\left(\frac{m_1x_i + m_2y_i + m_3}{m_7x_i + m_8y_i + 1} - x_i'\right)^2 + \left(\frac{m_4x_i + m_5y_i + m_6}{m_7x_i + m_8y_i + 1} - y_i'\right)^2\right)\times D_i^2\right] \tag{11}$$

where D_i is the denominator of motion model:

$$D_i = \left(m_7x_i + m_8y_i + 1\right) \tag{12}$$

Therefore, we can simplify (11) as:

$$E = \sum_{i=1}^{N_b}\left[\left(m_1x_i + m_2y_i + m_3 - x_i'D_i\right)^2 + \left(m_4x_i + m_5y_i + m_6 - y_i'D_i\right)^2\right] \tag{13}$$

At this stage, we can minimize (11) by solving $\partial E / \partial m_j = 0$ and arriving at:

$$\left(\sum_{i=1}^{N_b}\mathbf{A}_i\right)\mathbf{m} = \sum_{i=1}^{N_b}\mathbf{b}_i \tag{14}$$

where $\mathbf{m}$ is GM vector. The $\mathbf{A}_i$ matrix and $\mathbf{b}_i$ vector are defined as:

$$\mathbf{A}_i = \begin{bmatrix} x_i^2 & x_iy_i & x_i & 0 & 0 & 0 & -x_i^2x_i' & -x_iy_ix_i' \\ x_iy_i & y_i^2 & y_i & 0 & 0 & 0 & -x_iy_ix_i' & -y_i^2x_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_ix' & -y_ix_i' \\ 0 & 0 & 0 & x_i^2 & x_iy_i & x_i & -x_i^2y_i' & -x_iy_iy_i' \\ 0 & 0 & 0 & x_iy_i & y_i^2 & y_i & -x_iy_iy_i' & -y_i^2y_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_iy_i' & -y_iy_i' \\ x_i^2x_i' & x_iy_ix_i' & x_ix_i' & x_i^2y_i' & x_iy_iy_i' & x_iy_i' & -x_i^2s_i' & -x_iy_is_i' \\ x_iy_ix_i' & y_i^2x_i' & y_ix_i' & x_iy_iy_i' & y_i^2y_i' & y_iy_i' & -x_iy_is_i' & -y_i^2s_i' \end{bmatrix} \tag{15}$$

$$\mathbf{b}_i = \left[x_ix_i' \quad y_ix_i' \quad x_i' \quad x_iy_i' \quad y_iy_i' \quad y_i' \quad x_is_i' \quad y_is_i'\right]^T \tag{16}$$

$$s_i' = x_i'^2 + y_i'^2. \tag{17}$$

3.4 Subsampling Pixels and Levenberg-Marquardt Algorithm

In this stage, the estimated GM from the previous stage is optimized with greater accuracy by employing LMA. In this paper, we suggest subsampling from all pixels of current frame with a static pattern as in [4], instead of just selecting feature pixels

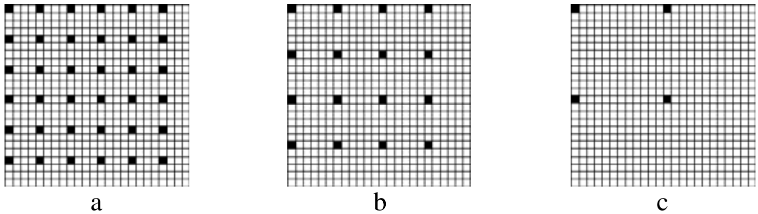

Fig. 2. Pixels subsampling pattern; a) 1:4×4; b) 1:6×6; c) 1:12×12.

among the remaining blocks as in [14]. This selection technique poses less computational complexity than [14] and it is more precise.

In this paper, the 1:12×12 sampling method is used which means that we select one pixel from each 12×12 block. After pixels subsampling, initial GM is optimized by applying LMA to these pixels. To reduce outlier effects, 10% of pixels with the most error are discarded after first iteration [4].

4 Simulation

In this section, the proposed method is examined and compared against MPEG-4 VM, [14] and [4] with a sampling factor 1:9×9.The following sequences with CIF format are considered for simulations: Akiyo (300 frames), Bus (150 frames), Carphone (300 frames), Coastguard (300 frames), Foreman (400 frames), Flower (150 frames), Mobile (300 frames), Stefan (300 frames), Tempete (260 frames), and Waterfall (260 frames). The simulations are run on a desktop computer featuring 2.66GHz Core2Quad CPU, 4GB RAM and MS Windows Vista operating system in MATLAB environment.

The GME time of different sequences are presented in Table 1. Judging from the Table, it is seen that the proposed method's GME time is less than that in [4] for most of the sequences. Furthermore, this is almost the same as the GME time in [14] with affine model.

Table 2 compares speed of the proposed method versus other methods in relation to the MPEG-4 VM method with perspective model. As these results illustrate, the proposed technique is 53 times faster than VM with perspective model. This is while the method in [14] is about 43 times faster than VM with affine model and about 60 times faster than VM with perspective model. The Proposed method as well as [4] both work with perspective model whereas [14] only works with affine model.

The PSNR of sequences is calculated by:

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{18}$$

Table 1. GME Time Comparison of 5 Different Methods (Sec.)

Sequence	VM(Pers.)	VM(Aff.)	[4]	[14]	Proposed
Akiyo	433.11	254.25	7.15	7.18	8.45
Bus	232.66	145.86	5.24	3.38	4.05
Carphone	152.68	99.86	4.47	3.77	4.10
Coastguard	436.75	299.95	6.96	6.67	7.75
Foreman	960.57	640.99	12.61	8.89	10.77
Flower	518.55	279.10	7.03	5.48	6.74
Mobile	354.57	222.69	11.12	6.70	8.10
Stephan	297.07	204.22	8.79	6.57	7.80
Tempete	225.25	153.99	7.00	5.64	7.01
Waterfall	345.65	190.19	6.84	6.15	7.27

Table 2. Speed Comparison of the [4] and MPEG-4 VM Perspective GM

Sequence	VM(Pers.)	VM(Aff.)	[4]	[14]	Proposed
Akiyo	1.00	1.70	60.57	60.32	51.26
Bus	1.00	1.60	44.40	68.83	57.45
Carphone	1.00	1.53	34.16	40.50	37.24
Coastguard	1.00	1.46	62.75	65.48	56.35
Foreman	1.00	1.50	76.18	108.05	89.19
Flower	1.00	1.86	73.76	94.63	76.94
Mobile	1.00	1.59	31.89	52.92	43.77
Stephan	1.00	1.45	33.80	45.22	38.09
Tempete	1.00	1.46	32.18	39.94	32.13
Waterfall	1.00	1.82	50.53	56.20	47.54
Avg.	1.00	1.60	50.02	63.21	53.00

Table 3. PSNR Comparison for Different Sequences (dB)

Sequence	VM(Pers.)	VM(Aff.)	[4]	[14]	Proposed
Akiyo	41.010	41.011	41.101	36.301	41.012
Bus	21.687	21.679	21.623	21.805	21.831
Carphone	30.811	30.739	30.403	28.855	29.729
Coastguard	26.376	26.384	26.358	26.242	26.599
Foreman	25.279	25.256	25.289	23.237	25.085
Flower	28.312	28.160	27.884	27.227	27.716
Mobile	25.538	25.495	25.583	25.206	25.581
Stephan	24.494	24.157	22.753	23.591	23.916
Tempete	27.786	27.778	27.726	27.434	27.715
Waterfall	35.675	35.634	35.573	34.918	35.725

Table 4. PSNR Degradation in Respect of MPEG-4 VM Perspective GM

Sequence	VM(Pers.)	VM(Aff.)	[4]	[14]	Proposed
Akiyo	0.000	0.001	0.090	-4.709	0.002
Bus	0.000	-0.008	-0.064	0.118	0.144
Carphone	0.000	-0.072	-0.408	-1.956	-1.082
Coastguard	0.000	0.008	-0.018	-0.135	0.222
Foreman	0.000	-0.152	-0.428	-1.086	-0.597
Flower	0.000	0.022	0.011	-2.042	-0.193
Mobile	0.000	-0.043	0.045	-0.332	0.043
Stephan	0.000	-0.336	-1.740	-0.903	-0.577
Tempete	0.000	-0.009	-0.060	-0.353	-0.071
Waterfall	0.000	-0.041	-0.103	-0.758	0.050
Avg.	0.000	-0.068	-0.268	-1.215	-0.206

where

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(F_k(x,y) - F_{k-1}(x',y'))^2 \tag{19}$$

In Table 3, PSNR of GME for each sequence is presented. Table 4 also displays PSNR degradation in respect of VM with perspective motion model. As the results demonstrate, the proposed method has on average reduced the PSNR by -0.2 dB while [14] method degrades the PSNR by -1.2 dB.

5 Conclusion

In this paper a fast two-stage algorithm for global motion estimation (GME) with perspective model is introduced. In the first stage, eight parameters of global motion (GM) are estimated by using sampled motion vectors of blocks. In the second stage, by subsampling of pixels and using Levenberg-Marquardt algorithm (LMA), the estimated GM of the first stage is estimated more accurately.

As the simulation results demonstrate, one key advantage of the proposed solution in this paper is that it is almost 53 times faster than the MPEG-4 VM method. Another outstanding feature of the innovative technique is its enhanced estimation accuracy which is more than FFRGMET's and [4]'s and almost the same as MPEG-4 VM's. Still, when compared against [14], the algorithm exhibits better precision under the same speed. This is while our method works with the perspective model and [14] estimates the simpler affine model.

References

1. Qi, B., Ghazal, M., Amer, A.: Robust Global Motion Estimation Oriented to Video Object Segmentation. IEEE Trans. Image Process. 17(6), 958–967 (2008)
2. Dufaux, F., Konrad, J.: Efficient, robust and fast global motion estimation for video coding. IEEE Trans. Image Process. 9(3), 497–501 (2000)
3. MPEG-4 video verification model version 18.0. In: ISO/IEC JTC1/SC29/WG11 N3908, Pisa, Italy (2001)
4. Alzoubi, H., Pan, W.D.: Fast and Accurate Global Motion Estimation Algorithm Using Pixel Subsampling. Information Sciences 178(17), 3415–3425 (2008)
5. Alzoubi, H., Pan, W.D.: Reducing the complexity of MPEG-4 global motion estimation using pixel subsampling. IET Electronic letters 44(1), 20–21 (2008)
6. Keller, Y., Averbuch, A.: Fast gradient methods based on global motion estimation for video compression. IEEE Transactions on Circuits and Systems for Video Technology 13(4), 300–309 (2003)
7. Chan, W.C., Au, O.C., Fu, M.F.: Improved global motion estimation using prediction and early termination. Proc. IEEE Int. Conf. Image Processing 2, 285–288 (2002)
8. Moscheni, F., Dufaux, F., Kunt, M.: A new two-stage global/local motion estimation based on a background/foreground segmentation. In: Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1995), Detroit, MI, vol. 4, pp. 2261–2264 (1995)
9. Rath, G.B., Makur, A.: Iterative least squares and compression based estimation for a four-parameter linear motion model and global motion compensation. IEEE Transactions on Circuits & Systems for Video Technology 9(7), 1075–1099 (1999)
10. Xu, G., Ding, M., Cheng, Y., Tian, Y.: Global motion estimation based on kalman predictor. In: Proc. of 2009 IEEE Int. Workshop on Imaging Systems and Techniques, Shenzhen, China, pp. 395–398 (2009)
11. Chung, Y., He, Z.: Reliability Analysis for Global Motion Estimation. IEEE Signal Processing Letters 11(11), 980–997 (2009)
12. Tarannum, N., Pickering, M.R., Frater, M.R.: An automatic and robust approach for global motion estimation. In: Proc. of IEEE Int. Workshop on Multimedia Signal Processing (MMSP 2008), Australia, pp. 83–88 (2008)
13. Shang, F., Yang, G., Yang, H., Tian, D.: Efficient Global Motion Estimation Using Macroblock Pair Vectors. In: Proc. of Int. Conf. on Information Technology and Computer Science (ITCS 2009), Ukraine, vol. 1(1), pp. 225–228 (2009)
14. Lei, L., Zhiliang, W., Jiwei, L., Zhaohui, C.: Fast Global Motion Estimation. In: Proc. of 2nd IEEE Int. Conf. on Broadband Network & Multimedia Technology (IC-BNMT 2009), Beijing, pp. 220–225 (2009)
15. ISO/IEC JTC1/SC29/WG11 MPEG Video Group. Optimization model. ISO/IECJTC1/SC29/WG11N3675 LaBaule, France (2000)
16. Dellaert, F., Collins, R.: Fast image-based tracking by selective pixel integration. In: ICCV Workshop on Frame-Rate Vision, Corfu., Greece (1999)
17. Farin, D.: Automatic Video Segmentation Employing Object/Camera Modeling Techniques. Ph.D. Thesis, Technical University of Eindhoven, pp. 110-114 (2005)