Robust GME in Encoded MPEG Video

Vahid Kiani
Ferdowsi University of Mashhad
Azadi Square
Mashhad, Iran
vahid.kiani@rocketmail.com

Hamid Reza Pourreza
Ferdowsi University of Mashhad
Azadi Square
Mashhad, Iran
hpourreza@um.ac.ir

ABSTRACT
In this paper, we propose a novel block-based approach for global motion estimation (GME) and panorama image construction in encoded MPEG-1 video streams. Direct extraction of motion vectors (MV) from MPEG stream greatly improves efficiency of our method against pixel-based methods. However, some MVs in MPEG stream do not indicate real object or camera motion in the scene. Therefore, we introduce a reliability measure to discriminate real MVs from noisy or outlier MVs. In addition, an iterative reweighting process is applied to increase accuracy of GME. Finally, panorama image of several sequences is constructed by using estimated camera motion. Experiments show that the proposed method has high accuracy and can produce high quality panorama images for compressed videos faster than real-time.

Categories and Subject Descriptors
I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement – *Feature representation.* I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *Motion.* I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding – *Video analysis.*

General Terms
Algorithms, Reliability.

Keywords
global motion estimation, motion vector reliability, MPEG, IRLS, panorama image construction.

1. INTRODUCTION
Rapid growth of archived video content has lead to huge video databases worldwide. Effective and efficient video processing algorithms in compressed domain are needed to process these archives. Global motion is one of the most promising semantic features in video analysis, indexing and retrieval applications. Two main approaches for global motion estimation (GME) are pixel domain and compressed domain [10]. The compressed domain approaches diminish computational complexity of GME by exploiting block-based motion vector (MV) field from compressed video. However, MVs in the compressed video stream are often noisy and inconsistent with real motion, and need to examine by a filtering process [1].

Several works are done to estimate global motion from block motion vectors in compressed or pixel domain. Chen in [1] proposed a cascade of MV filters to reject outliers among block based MVs extracted from raw images in pixel domain. No iterative enhancement stage is used in this method to increase accuracy of GME. An iterative least square error minimization method is proposed in [10] to extract global motion form coarsely sampled MV field. However, in this method block motion vectors are not extracted from compressed video and outlier MVs are not filtered before iterative GME process; this causes necessity for more iterations in the iterative GME process and expands computational burden of the proposed method. Haller in [9] proposed an iterative GME method for motion vector field from variable size blocks. Different motion models are examined in their work and profits of using appropriate motion model for each problem domain are discussed. These methods are not directly applicable to compressed video processing because they extract MVs by block based MV estimation in pixel domain. Panorama image construction is a qualitative way of camera motion evaluation [3,7]. Moreover, panorama image has used as a static summary for video [11,19].

In this research work, we propose a method for global motion estimation and panorama image construction in encoded MPEG-1 videos by directly exploiting MV field of compressed domain. Section 2 describes process of motion vectors extraction from MPEG bit-stream. In section 3, a reliability measure is introduced to discriminate unreliable MVs form real motion vectors in the scene. A simple yet applicable motion model is proposed in section 4 and initial camera parameters are estimated based on reliable motion vectors. In section 5, an iterative reweighting method is applied to enhance accuracy of GME. Section 6 is dedicated to panorama image construction and experiments. The block diagram for the proposed method is shown in Figure 1.

2. MOTION VECTOR EXTRACTION
In the first step, motion vectors are extracted from forward coded macro-blocks (MBs) in each P-Picture of the MPEG stream as follows:

$$x'(m,n) = x(m,n) + MV_x(m,n) \quad (1)$$

$$y'(m,n) = y(m,n) + MV_y(m,n) \quad (2)$$

Where x(m,n) and y(m,n) are center point location of block (m,n) in current picture, x'(m,n) and y'(m,n) are center point location of most similar area in previous picture.

MoMM2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

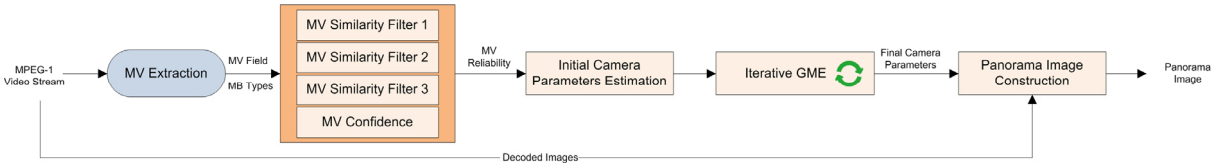


Figure 1. Block diagram for the robust panorama construction approach.

The $MV_x(m,n)$ and $MV_y(m,n)$ are horizontal and vertical components of MB (m,n) motion vector in current picture. A zero motion vector is assigned to intra-coded MBs. We modified a MPEG-1 video decoder in Java to extract these motion vectors by partial decoding of the MPEG bit-stream.

3. MOTION VECTOR RELIABILITY MEASUREMENT

Accidental similarity of MBs in current and reference picture around moving object boundaries leads to outlier MVs in the MPEG stream. In addition, fast object and camera motion in the scene can produce noisy MVs in the picture. By defining a reliability measure, we try to discriminate noisy and outlier MVs from real MVs.

3.1 Outlier Removal

While real MVs are similar to their neighboring MVs, outliers are different from their neighbors. Therefore, local similarity measures are used to discriminate outliers from real MVs [1,16,18]. In this study, we modify soft-threshold similarity measures proposed in [1] and apply them in MPEG domain. These filters are shown in Figure 2.

3.1.1 First Similarity Filter

Magnitude and phase similarity of the center MV with its 8 neighbor MVs are measured by the first similarity filter. First magnitude similarity measure is defined as:

$$PM_1(m,n) = \frac{1}{8}\sum_{i=1}^{8} SM_i^1(m,n) \tag{3}$$

Where $SM_i^1(m,n)$ denotes magnitude similarity of i'th neighbor MV to central MV(m,n) in a 3*3 neighborhood; and is computed as:

$$SM_i^1(m,n) = \begin{cases} 1 & \dfrac{\|\mathrm{MV}_i - \mathrm{MV}(m,n)\|}{\|\mathrm{MV}_i\|} < T_1^{mag} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The MV_i is the i'th neighbor MV, and T_1^{mag} is magnitude similarity threshold of the first filter. In our experiments, as suggested in [1] we used $T_1^{mag} = 0.4$ for CIF resolution videos, and $T_1^{mag} = 0.25$ for higher resolution videos.

First phase similarity measure is defined as:

$$PA_1(m,n) = \frac{1}{8}\sum_{i=1}^{8} SA_i^1(m,n) \tag{5}$$

Where $SA_i^1(m,n)$ denotes phase similarity of i'th neighbor MV to central MV(m,n) in a 3*3 neighborhood; and is computed as:

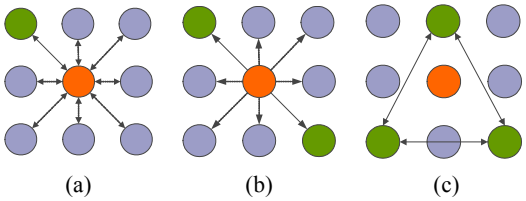


Figure 2. Three MV similarity filters for outlier rejection [1].

$$SA_i^1(m,n) = \begin{cases} 1 & |\varphi(\mathrm{MV}_i) - \varphi(\mathrm{MV}(m,n))| < T_1^{ph} \\ 0 & otherwise \end{cases} \tag{6}$$

The MV_i is the i'th neighbor MV, and T_1^{ph} is phase similarity threshold of the first filter. In our experiments, as suggested in [1] we used $T_1^{ph} = 19°/180°$ for CIF resolution videos and $T_1^{ph} = 10°/180°$ for higher resolution videos.

3.1.2 Second Similarity Filter

The second filter measures magnitude and phase similarity of the center MV with average of its diagonally opposite neighbor MVs. The measures are similar to (3) and (5), but for average of diagonally opposite neighboring MVs [1]. In our experiments, we used $T_2^{mag} = 0.2$ for CIF resolution videos, and $T_2^{mag} = 0.125$ for higher resolution videos. In addition, we used $T_2^{ph} = 9.5°/180°$ for CIF resolution videos, and $T_2^{ph} = 5°/180°$ for higher resolution videos.

3.1.3 Third Similarity Filter

The third filter measures magnitude and phase similarity of the center MV with average of its triangularly opposite neighbor MVs. This filter pays more attention to cornerwise neighbors by using them two times in computations. Third magnitude and phase similarity measures are defined like (3) and (5), but for average of triangularly opposite neighbor MVs [1]. In our experiments, we used $T_3^{mag} = 0.1$ and $T_3^{ph} = 4.75°/180°$ for CIF resolution videos. In addition, we used $T_3^{mag} = 0.0625$ and $T_3^{ph} = 2.5°/180°$ for higher resolution videos.

3.2 Noise Removal

In each P-Picture, intra-coded MBs indicate regions in the picture with no similar area in reference picture. Therefore, we expect to see noisy MVs around intra-coded MBs. By this assumption, confidence measure for each MV is defined as follows:

$$Cnf(m,n) = \begin{cases} 1 - \dfrac{1}{8}\sum_{i,j \in neighbors(m,n)} IntraMB(i,j) & \text{non intracoded MB} \\ 0 & \text{intracoded MB} \end{cases} \tag{7}$$

Which computes ratio of non intra-coded MBs around the MV(m,n) in a 3*3 neighborhood. The IntraMB(i,j) is one for intra-coded MBs and zero for non intra-coded MBs.

3.3 Reliability Measure

In our framework, noisy and outlier MVs are two kinds of unreliable MVs. To distinguish them from real MVs we combine local MV similarity measure with MV confidence measure as:

$$\forall m,n \qquad MV_{Reliablity}(m,n) = PM_1(m,n) * PA_1(m,n) * PM_2(m,n) * PA_2(m,n) * PM_3(m,n) * PA_3(m,n) * Cnf(m,n) \tag{8}$$

Where $MV_{Reliability}(m,n)$ is reliability of MV(m,n) and is a continuous value between zero and one.

We use an adaptive thresholding strategy to distinguish reliable MVs from unreliable MVs. The adaptive threshold is defined as:

$$Th_{MVRel} = \max(1.2 * M_{MVRel}, 0.3) \tag{9}$$

Where M_{MVRel} is mean MV reliability in current picture and is defined as:

$$M_{MVRel} = \frac{1}{M * N} \sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) \tag{10}$$

The M and N are number of rows and columns of the MB grid in current picture. All MVs with reliability higher than Th_{MVRel} are considered as reliable MV.

4. CAMERA MOTION MODEL

As discussed in [17], to model 3D motion in a scene a 2D parametric motion model can be utilized. The affine motion model with six parameters has extremely used in the literature [2,3,8,10,13,20]. However, in most real world applications such as sport video analysis, camera movements can be categorized into panning, tilting, and zooming. So, we simplify affine motion model to a three-parameter motion model as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & s \\ s & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p \\ t \end{bmatrix} \tag{11}$$

Where s, p, t are scaling, panning and tilting factors respectively. By this model, each pixel (x,y) in the current picture is mapped to corresponding pixel (x',y') in the previous reference picture.

Putting start and end point of each motion vector of the picture in (11) results in a linear equation system. By solving this equation system, initial camera parameters are estimated. Using these initial parameter values, the iterative GME process in the next step could start from a promising point.

5. ITERATIVE GLOBAL MOTION ESTIMATION

Since most motion vectors in each picture are caused by camera motion; an iterative reweighting method could be used to remove remaining errors in GME [5,15]. Therefore, the Iteratively Reweighted Least Squares (IRLS) [2,4,6,15] method is used in this section to refine MV reliability values and reestimate parameters of camera motion. In this process, the estimated reliability value of each MV is considered as MV weight during parameters estimation. In IRLS, the following error function is minimized in order to find optimal camera parameter values:

$$J = \sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) * \left[(s \cdot x(m,n) + p - x'(m,n))^2 + (s \cdot y(m,n) + t - y'(m,n))^2 \right] \tag{12}$$

To minimize error J, partial derivatives of J are derived with respect to s, p, and t. After a simple displacement, following formulas will appear for iterative camera parameters estimation:

$$p = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) * (x'(m,n) - s \cdot x(m,n))}{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n)} \tag{13}$$

$$t = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) * (y'(m,n) - s \cdot y(m,n))}{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n)} \tag{14}$$

$$s = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) * [x(m,n) \cdot (x'(m,n) - p) + y(m,n) \cdot (y'(m,n) - t)]}{\sum_{m=1}^{M} \sum_{n=1}^{N} MV_{Reliability}(m,n) * [x(m,n)^2 + y(m,n)^2]} \tag{15}$$

After parameters estimation, MV reliability values are recalculated based on last parameter values by following equation:

$$MV_{Reliability}(m,n) = \exp\left(-k * \left[(s \cdot x(m,n) + p - x'(m,n))^2 + (s \cdot y(m,n) + t - y'(m,n))^2 \right]\right) \tag{16}$$

The k is an arbitrary positive constant which determines contribution of MV error in MV reliability during the GME process. Iterative estimation of camera parameters and MV reliabilities will continue for a determined number of iterations. In our experiments, we used k = 4 and max iterations of 30.

6. Experimental Results

In this section, we evaluate performance of our proposed method called robust IRLS (R-IRLS) on several benchmark and soccer videos by panorama image construction and global motion compensation. We implemented the iterative GME method proposed by Liu in [14] on reliable MVs of section 3 and use it as an opponent in our experiments.

6.1 Test Videos

Several benchmark videos are used for evaluation. These test videos are in CIF (352 × 288) resolution and available at http://media.xiph.org. They are encoded with FFMPEG encoder as MPEG-1 videos. Important encoding parameters are summarized in Table 1. In addition, three soccer sequences in (720 × 480) resolution are used to show the robustness of the proposed method for soccer domain. Important encoding parameters for soccer videos are summarized in Table 2.

6.2 Panorama Image Construction

Construction of panorama image can be utilized to evaluate estimated global motion qualitatively [7]. To construct a panorama image, consecutive images must coincide on each other according to estimated camera motion. Therefore, a misestimation error propagates along the sequence in panorama image construction process and results in a panorama image with apparent errors [7]. When major motion in the scene is camera motion against object motion, we can suppose that there is no moving object in the scene. In such a situation, accurate and robust estimation of camera motion will produce a bright and clear panorama image.

In prior section, camera motion parameters are estimated for each image in the sequence. To coincide successive images, their transformation matrices must refer the same initial frame.

Therefore, accumulated transformation matrix of each image in the sequence is computed as [7]:

$$\begin{aligned} H^{P}_{1\to1} &= I_{3*3} \\ H^{P}_{2\to1} &= H_{1\to1}H_{2\to1} = H^{P}_{1\to1}H_{2\to1} \\ H^{P}_{3\to1} &= H_{1\to1}H_{2\to1}H_{3\to2} = H^{P}_{2\to1}H_{3\to2} \\ H^{P}_{4\to1} &= H_{1\to1}H_{2\to1}H_{3\to2}H_{4\to3} = H^{P}_{3\to1}H_{4\to3} \\ &\vdots \\ H^{P}_{N\to1} &= H_{1\to1}H_{2\to1}H_{3\to2}H_{4\to3}.....H_{N\to N-1} = H^{P}_{N-1\to1}H_{N\to N-1} \end{aligned} \quad (17)$$

Where N is number of images in the sequence, and $H^{P}_{i\to j}$ denotes accumulated transformation matrix which transforms i'th picture to the j'th picture in the sequence. The symbol I denotes a 3*3 identity matrix, and $H_{i\to i-1}$ is transformation matrix of i'th picture constructed as:

$$H_{i\to i-1} = \begin{bmatrix} s(i) & 0 & p(i) \\ 0 & s(i) & t(i) \\ 1 & 1 & 1 \end{bmatrix} \quad (18)$$

Where s(i), p(i) and t(i) are three computed camera parameters s, p, and t of i'th picture in the sequence.

After computation of accumulated transformation matrix, pixels in each picture of the sequence are transformed to the first picture by following equations:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{k} \end{bmatrix} = H^{P}_{i\to1} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (19)$$

$$x' = \frac{\tilde{x}}{\tilde{k}} \qquad y' = \frac{\tilde{y}}{\tilde{k}} \quad (20)$$

Table 1. FFMPEG encoder settings for benchmark videos.

Parameter	Value
GOP Size	16 (fix)
B-Picture count	0
Frame rate	25
Bit-rate	5120 kbps
Frame Size	352 × 288
Maximum Motion Vector Range	50 pixel

Table 2. FFMPEG encoder settings for soccer videos.

Parameter	Value
GOP Size	16 (fix)
B-Picture count	0
Frame rate	25
Bit-rate	10240 kbps
Frame Size	720 × 480
Maximum Motion Vector Range	100 pixel

Which transforms each pixel (x,y) in i'th image of the sequence to pixel(x',y') in the first picture.

After transformation of each image in the sequence to the first picture, all transformed pictures must combine to constitute panorama image. Transformed images could synthesize by simply overlaying each picture on prior pictures or using average or median filtering in the time axes. Average filter in the time axes is faster than median filter. Therefore, we used average filter for pictures combination in CIF resolution sequences. In this situation, errors in camera parameters estimation cause blurring or distortions in the resulting panorama image. For soccer sequences, images are combined by overlaying method.

Figure 3 shows panorama images of bus CIF sequence generated by using our method R-IRLS, Liu [14] method, and linear equations method of section 4. According to Figure 3, iterative GME refinement in R-IRLS and Liu [14] leaded to finer panorama images. Also, panorama images of three soccer sequences generated by R-IRLS are shown in figure 4. The proposed method constructed clear and fine panorama images for both general and soccer sequences.

6.3 Global Motion Compensation

In video sequences with dominant camera motion, motion compensation error of each picture is correlated with camera motion estimation accuracy [3]. We use PSNR measure in dB to compare our proposed method and other approaches [1,10,12,14]. Motion compensation error is computed by matching current picture with previous reference picture using camera motion compensation. Then, square difference between two matched pictures is considered as global compensation error. The PSNR measure in dB is defined as:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (21)$$

Where MAX_I denotes maximum possible value in the picture; that is 255. The MSE denotes Mean Square Error and is computes as:

$$MSE = \frac{1}{3*H*W}\sum_{c=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W}\left[I^{c}(i,j) - K^{c}(i,j)\right]^2 \quad (22)$$

Where I and K are current and reference picture in RGB color format respectively, c denotes color component index, H and W are height and width of picture respectively.

Table 3 represents performance of our method against CAS_GD [1] and Liu [14]. Computation of PSNR is done on original raw pictures of each sequence to avoid effects of quality loss caused by MPEG compression. As shown in Table 3, the proposed method has high accuracy comparing to other state-of-the-art methods.

6.4 GME Speed

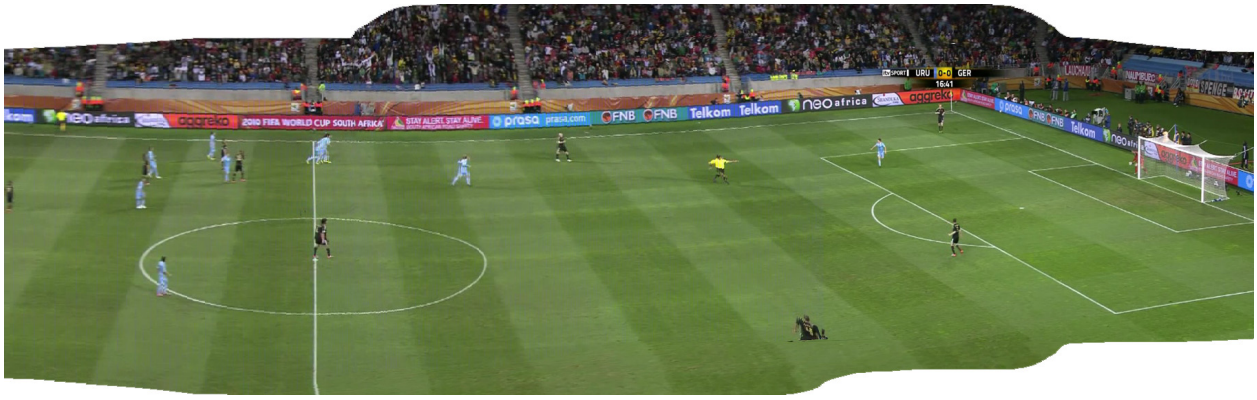
The GME, reliability measurement, and synthesization modules are implemented in MATLAB. The total processing time for motion extraction from MPEG bit-stream, reliability computation, and GME per frame is 12 milliseconds for CIF resolution videos and 16.7 milliseconds for soccer videos. These experiments are done on a standard laptop PC with Intel Pentium CPU at 2.53 GHZ, with 3 GB of RAM. Comparing to [1], our method is more efficient due to matrix operations in MATLAB and direct extraction of MVs in Java.

(a)

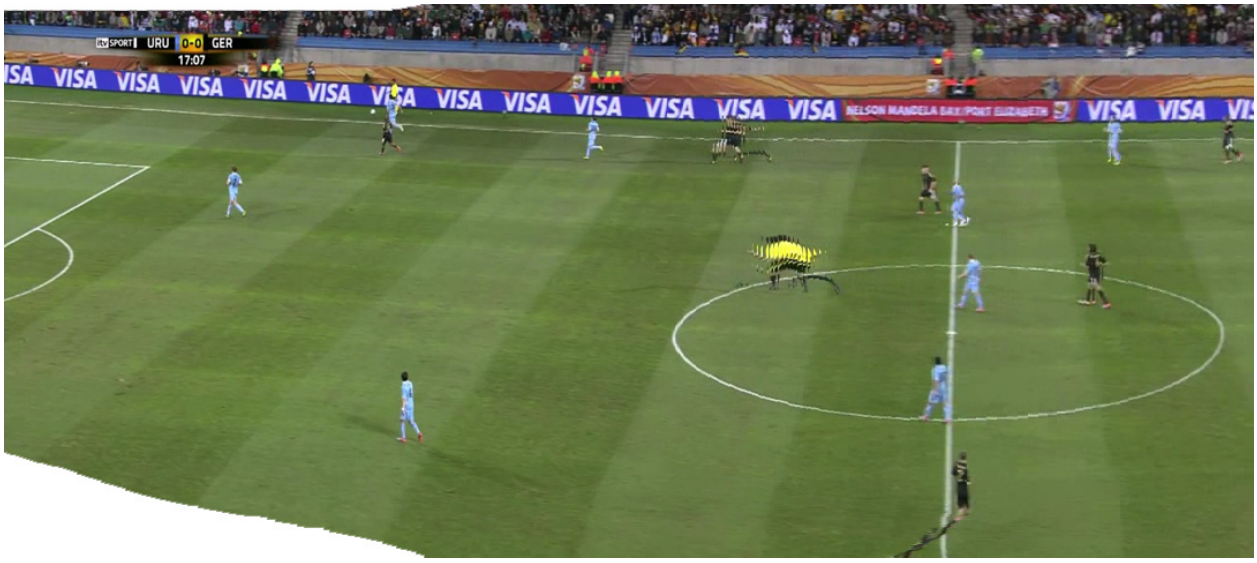
(b)

(c)

Figure 3. Panorama images for bus sequence by averaging filter in the time axes; (a) linear equations solving method; (b) R-IRLS; (c) Liu method.



(a)



(b)

(c)

Figure 4. Panorama images for three soccer videos by overlaying method; (a) panorama image of Soccer1 sequence; (b) panorama image of Soccer2 sequence; (c) panorama image of Soccer3 sequence.

Table 3. Performance Comparison by Average PSNR in dB.

Sequences	R-IRLS	CAS_GD [1]	Liu [14]
Bus	23.26	---	23.24
City	28.57	29.48	28.65
CoastGuard	28.28	26.78	28.27
FlowerGarden	23.38	22.19	23.83
Mobile	25.36	23.47	25.38
Stefan	26.02	24.60	26.00
Tempete	27.74	27.83	27.51
Waterfall	36.10	34.86	35.43
Soccer1	29.70	---	29.71
Soccer2	28.84	---	28.84
Soccer3	29.27	---	29.24

7. CONCLUSION

In this paper, we presented a robust and efficient method for panorama image construction in compressed videos. Estimating motion vector reliabilities and improving estimated global motion by an iterative process led to robustness of our method against MV noise in MPEG bit-stream. On the other hand, direct extraction of motion vectors from compressed sequence by partial decoding of MPEG video caused high efficiency in the overall process. Experiments show that the proposed method is accurate and efficient comparing to state-of-the-art research works.

8. REFERENCES

[1] Chen, Y.-M. and Bajic, I.V. 2010. Motion Vector Outlier Rejection Cascade for Global Motion Estimation. *IEEE Signal Processing Letters 17*, 2 (Feb. 2010), 197-200. DOI= http://dx.doi.org/10.1109/LSP.2009.2036879.

[2] Coldefy, F. and Bouthemy, P. 2004. Unsupervised Soccer Video Abstraction Based on Pitch, Dominant Color and Camera Motion Analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia* (New York, USA, October 10-16, 2004). MULTIMEDIA'04. ACM, New York, NY, 268-271. DOI= http://doi.acm.org/10.1145/1027527.1027588.

[3] Corrigan, D., Kokaram, A., Coudray, R., and Besserer, B. 2006. Robust Global Motion Estimation From Mpeg Streams with a Gradient Based Refinement. In *Proceedings of the IEEE International Conference on Acoustics Speed and Signal Processing* (Toulouse, France, May 14-19, 2006). ICASSP 2006. IEEE, Piscataway, NJ, II-285-II-288. DOI= http://dx.doi.org/10.1109/ICASSP.2006.1660335.

[4] Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, C.S. 2010. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics 63*, 1 (Jan. 2010), 1-38. DOI= http://dx.doi.org/10.1002/cpa.20303.

[5] Duan, L.-yu, Wang, J., Zheng, Y., et al. 2005. Shot-Level Camera Motion Estimation Based on a Parametric Model. In *Proceedings of the TREC Video Retrieval Evaluation Online Proceedings'2005*.

[6] Endra, O. and Gunawan, D. 2010. Comparison of l1-Minimization and Iteratively Reweighted least Squares-l p-Minimization for Image Reconstruction from Compressive Sensing. In *Proceedings of the Second International Conference on Advances in Computing, Control, and Telecommunication Technologies* (Jakarta, Indonesia, December 02-03, 2010). ACT 2010. IEEE, Piscataway, NJ, 85-88. DOI= http://dx.doi.org/10.1109/ACT.2010.31.

[7] Felip, R.L., Barcelo, L., Binefa, X., and Kender, J.R. 2008. Robust Dominant Motion Estimation Using MPEG Information in Sport Sequences. *IEEE Transactions on Circuits and Systems for Video Technology 18*, 1 (Jan. 2008), 12-22. DOI= http://dx.doi.org/10.1109/TCSVT.2007.903804.

[8] Gheissari, N., Bab-Hadiashar, A., and Suter, D. 2006. Parametric Model-Based Motion Segmentation using Surface Selection Criterion. *Computer Vision and Image Understanding 102*, 2 (May 2006), 214-226. DOI= http://dx.doi.org/10.1016/j.cviu.2006.02.002.

[9] Haller, M., Krutz, A., and Sikora, T. 2010. Robust Global Motion Estimation using Motion Vectors of Variable Size Blocks and Automatic Motion Model Selection. In *Proceedings of the International Conference on Image Processing* (Hong Kong, China, September 26-29, 2010). ICIP 2010. IEEE, Piscataway, NJ, 737-740. DOI= http://dx.doi.org/10.1109/ICIP.2010.5652035.

[10] Hsu, V. 2005. Global Motion Estimation from Coarsely Sampled Motion Vector Field and the Applications. *IEEE Transactions on Circuits and Systems for Video Technology 15*, 2 (Feb. 2005), 232-242. DOI= http://dx.doi.org/10.1109/TCSVT.2004.841656.

[11] Hua, T.M.X.-sheng. 2008. Structure and event mining in sports video with efficient mosaic. *Multimedia Tools and Applications 40*, 1 (Oct. 2008), 89-110. DOI= http://dx.doi.org/10.1007/s11042-007-0186-8.

[12] Huang, A.-M. and Nguyen, T.Q. 2008. A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation. *IEEE transactions on image processing 17*, 5 (May 2008), 694-708. DOI= http://dx.doi.org/10.1109/TIP.2008.919360.

[13] Jang, S., Pomplun, M., Kim, G., and Choi, H. 2005. Adaptive Robust Estimation of Affine Parameters from Block Motion Vectors. *Image and Vision Computing 23*, 14 (Dec. 2005), 1250-1263. DOI= http://dx.doi.org/10.1016/j.imavis.2005.09.003.

[14] Liu, S., Xu, M., Yi, H., Chia, L.-T., and Rajan, D. 2006. Multimodal Semantic Analysis and Annotation for Basketball Video. *EURASIP Journal on Advances in Signal Processing 2006*, (2006), 1-14. DOI= http://dx.doi.org/10.1155/ASP/2006/32135.

[15] Montoliu, R. and Pla, F. 2009. Generalized least squares-based parametric motion estimation. *Computer Vision and Image Understanding 113*, 7 (Jul. 2009), 790-801. DOI= http://dx.doi.org/10.1016/j.cviu.2009.01.006.

[16] Ojo, O. a and Haan, G. de. 1997. Robust Motion-Compensated Video Upconversion. *IEEE Transactions on*

Consumer Electronics 43, 4 (Nov. 2009), 1045-1056. DOI= http://dx.doi.org/10.1109/30.642370.

[17] Peyrard, N. and Bouthemy, P. 2005. Motion-Based Selection of Relevant Video Segments for Video Summarization. *Multimedia Tools and Applications 26*, 3 (Aug. 2005), 259-276. DOI= http://dx.doi.org/10.1007/s11042-005-0891-0.

[18] Saur, D.D., Kulkami, S.R., and Ramadge, P.J. 2000. Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation. *IEEE Transactions on Circuits and Systems for Video Technology 10*, 1 (Feb. 2000), 133-146. DOI= http://dx.doi.org/10.1109/76.825867.

[19] Truong, B.T. and Venkatesh, S. 2007. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications 3*, 1 (Feb. 2007), Article 3. DOI= http://dx.doi.org/10.1145/1198302.1198305.

[20] Wang, R. and Huang, T. 1999. Fast Camera Motion Analysis in MPEG Domain. In *Proceedings of the International Conference on Image Processing* (Kobe, Japan, October 24-28, 1999). ICASSP 2006. IEEE, Piscataway, NJ, 691-694. DOI= http://dx.doi.org/10.1109/ICIP.1999.817204.