International Journal of Innovative
Computing, Information and Control
Volume 9, Number 1, January 2013

ICIC International ©2013 ISSN 1349-4198
pp. 179–190

TENSOR VOTING BASED OUTLIER REMOVAL FOR GLOBAL MOTION ESTIMATION

Toan Dinh Nguyen* and Gueesang Lee*

Department of Electronics and Computer Engineering
Chonnam National University
Bukgu, Gwangju, South Korea
*Corresponding authors: toan_mulmi@hotmail.com; gslee@jnu.ac.kr

Received October 2011; revised February 2012

Abstract. *Motion vector based global motion estimation methods have much lower complexity than pixel based ones. Therefore, they are widely used in the compressed domain to estimate the camera motion in video sequences. However, the accuracy of these motion vector based methods largely depends on the quality of the input motion vector field. In real applications, many motion vector outliers are present due to noise or foreground objects. In this paper, a novel tensor voting based motion vector outlier removal method is proposed to improve the quality of the input motion vector field. First, motion vectors are encoded by second order tensors. A 2-D voting process is then used to smooth the motion vector field. Finally, the smoothed motion vector field is compared with the input one to detect outliers. The experimental results on synthetic and real data show the effectiveness of the proposed method.*

Keywords: Tensor voting, Motion vector, Outlier removal, Global motion estimation, Camera motion

1. **Introduction.** Global motion estimation (GME), a process to estimate the motion of the background caused by the camera motion in a video sequence, has been widely used in many applications, such as image mosaicing, video coding, video stabilization, content-based video analysis, and motion segmentation. In order to define the global motion field, motion models with various parameters are used. Four popular motion models are translational (2 parameters), geometric (4 parameters), affine (6 parameters) and perspective (8 parameters). GME can be performed in either the pixel domain [1] or the compressed domain [2-4] by solving a set of linear equations with the help of feature points correspondence or a motion vector field. An evaluation of pixel based and motion vector based GME methods can be found in [5].

In general, pixel based GME methods exhibit good performance, but with high computational cost. In [1], a pixel based Gauss-Newton gradient descent algorithm is proposed. An image pyramid is also used to reduce the computational complexity. A simplified version of this method is described in [5]. The simplified pixel based method can provide good results when compared with the original pixel based method.

As motion vectors are available in the compressed domain, they can be utilized in motion vector based GME methods to reduce the computational complexity. The study done in this paper belongs to this category. The global motion parameters that are obtained from the motion vectors in the compressed domain are useful in several applications, such as image mosaicing (also called panorama) [6], video transcoding [2]. In panorama application shown in Figure 1(a), frame images taken from different view points according to the movement of the camera are merged together in order to build a bigger and seamless image using the global motion parameters. In Figure 1(b), the global motion is used for the analysis of a video sequence by estimating the path that connects the points of interest

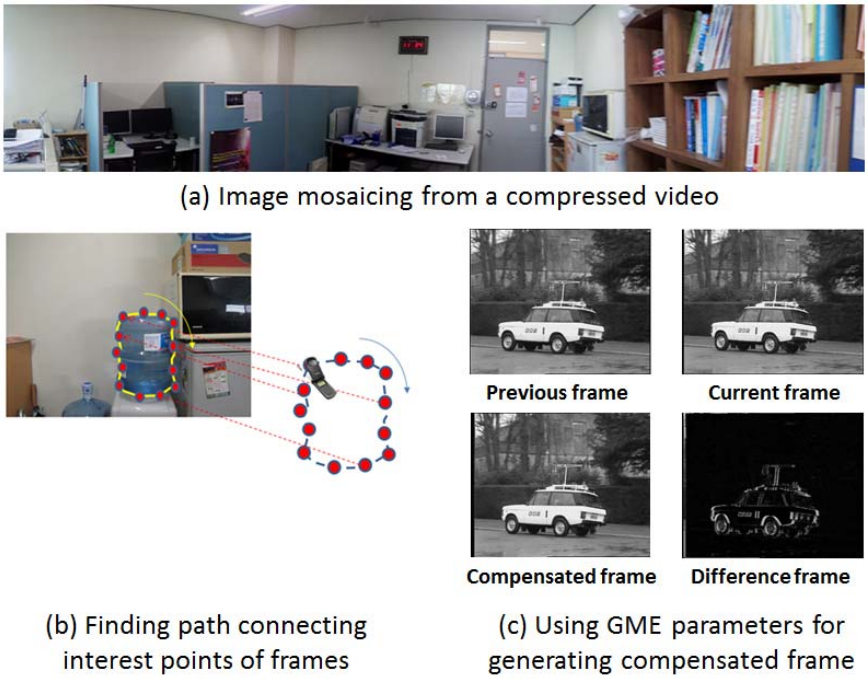


FIGURE 1. Some practical applications using global motion parameters in compressed domain

of frames from the video sequence captured by using a camera phone. From the resulting path, objects can be extracted and recognized more precisely. In Figure 1(c), the global motion parameters obtained from the previous and current frames in a video sequence can also be utilized to generate a compensated frame for video stabilization and moving object segmentation. As the goal of video stabilization is to create a new video sequence where the motion between the frames has been removed, the compensated frame is generated by removing the global motion from the current frame. This is used in the resulting video sequence instead of the original current frame. On the other hand, to detect the moving objects in a video sequence, the compensated frame is generated from the previous frame and the global motion. The difference frame is obtained by the subtraction of the current frame to the compensated one, and then it is used to detect the moving objects. As GME is also an important tool for video descriptors in MPEG-7, static-sprite generation in MPEG-4 Sprite coding, and global motion compensation (GMC) in MPEG-4 advanced simple profile (ASP), the global motion parameters obtained in the compressed domain can be used in a video transcoder that converts a video sequence from a source format (e.g., MPEG-2) to MPEG-4 or MPEG-7 format.

To estimate the global motion from a coarsely sampled motion vector field, the Newton-Raphson gradient descent method is used in [2] in order to minimize the fitting error between the input motion vectors and those generated from the estimated motion model. In [7], the least square solution for GME is obtained by using an M-Estimator. This method is robust and it obtains good results. Random sampled consensus (RANSAC) [8] can also be used for a motion vector based GME by defining iteratively a set of motion vectors which is used to obtain the global motion parameters. In [9], the RANSAC with the least square regression is used for GME.

Since coding efficiency is the first goad of compressed video sequences, the motion vectors in the video stream do not necessarily represent true motion. Therefore, they do not indicate global motion in some parts of the videos. Furthermore, many motion vector outliers are present due to noise and foreground objects. These outliers, the motion

vectors that fit poorly into the global motion, reduce the accuracy of the motion vector based GME methods. Several methods have been reported for the removal of the outliers. In [2], an iterative approach with fairly high complexity is used. In the preprocessing step, zero or near-zero motion vectors are screened first. Then, in each iteration of the gradient descent procedure, a match error histogram of all the input motion vectors is calculated. A hard-decision threshold is set to reject points with large errors in the histogram. In [3], the outliers are detected by examining the magnitude difference between a vector and its eight neighbors. The outlier filter is based on the assumption that the difference between the neighboring motion vectors is small. This method uses only the magnitude information and it tends to remove many motion vectors. Therefore, it generates poor results. In [4], a cascade method, which can take advantage of both magnitude and phase information, is presented. In this method, soft-decision thresholding is used instead of hard-decision thresholding as in [3]. This method is fast and accurate, as shown by the experimental results cited in [4]. However, in this method, only eight neighbors of a vector are considered to determine whether a particular vector is an outlier or not. These neighbors may be unreliable in the motion vector fields with high noise variance. Furthermore, predefined magnitude and phase thresholds are required.

In this paper, an efficient method for motion vector outlier removal based on 2-D tensor voting [10,11] is proposed to make use of the fact that motion vectors generated by the same motion model have strong spatial correlation. By encoding motion vectors by tensors and also by performing a tensor voting process among them, it is possible to take an efficient advantage of the correlation information among the motion vectors to smooth the motion vector field. The smoothed motion vector field is used as a reference for the detection of outliers. Previous works use only eight neighboring motion vectors and developing various filters and equations to check whether the input motion vector is an outlier or not. On the other hand, the proposed method can exploit the correlation between each motion vector with more surrounding ones and it can also detect outliers with only one voting step. Furthermore, the proposed method does not require any threshold values. The only input parameter of this method is the range of the voting process. It will be shown that the result is not sensitive to the value of this voting range. Since the tensor voting method is robust to noise, the proposed method can generate good results for motion vector fields with high noise variance.

In the remainder of this paper, Section 2 provides an introduction of global motion models. The proposed tensor voting based motion vector outlier removal method and experimental results are presented in Sections 3 and 4, respectively. Finally, conclusions and future works are given in Section 5.

2. **Global Motion Estimation Using a Motion Vector Field.** The perspective model with eight parameters is the most general one, among the four popular global motion models (translational, geometric, affine, and perspective). Detailed description of these models is given in [2]. Let us denote $m = [m_0, \ldots, m_7]$ as the vector that contains the eight parameters of the perspective model that are to be estimated, (x, y) and (x', y') as the coordinates of the current frame (anchor frame) and reference frame (target frame), respectively. The transformation can then be defined as:

$$\begin{aligned} x' &= f_x(x, y|m) = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}, \\ y' &= f_y(x, y|m) = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1} \end{aligned} \tag{1}$$

The estimation goal of pixel based GME methods is to minimize the sum of a weighted squared intensity errors, E, is the overall corresponding pairs of pixels or feature points inside the anchor image I and the reference image I', as stated in the following equation:

$$E = \sum_i w_i [I'(x'_i, y'_i) - I(x_i, y_i)]^2 = \sum_i w_i e_i^2 \tag{2}$$

where w_i is the weight of pixel i. Since a large number of pixels are considered, the minimization of this error function is very computationally intensive. In a compressed domain video sequence, motion vectors are available. Instead of pixels, the motion vectors are used in the motion vector based GME methods to reduce the computational complexity. The motion vector based GME methods estimate the global motion parameters by minimization of the sum of matching errors between the input motion vectors and the estimated ones. The input frame is divided into blocks. Let us denote i as the block index and (MVx_i, MVy_i) as the motion vector at the upper left spatial coordinates (x_i, y_i). According to the motion model m, the motion vector $(\Delta x_i, \Delta y_i)$ is equal to the displacement vector defined as follows:

$$(\Delta x_i, \Delta y_i) = (x'_i - x_i, y'_i - y_i) = (f_x(x_i, y_i|m) - x_i, f_y(x_i, y_i|m) - y_i) \tag{3}$$

The squared error to be minimized can then be calculated as:

$$E = \sum_i w_i ((MVx_i - x'_i + x_i)^2 + (MVy_i - y'_i + y_i)^2) \tag{4}$$

Since the motion vectors that are generated by the same motion model have strong spatial correlation, for each motion vector, its surrounding motion vectors can be used to determine whether it is an outlier or not. If the outliers are removed correctly then, the motion vector based GME methods converge faster and they generate more accurate results.

3. Tensor Voting Based Motion Vector Outlier Removal.

3.1. Overview of the method. As motion vectors generated by the same motion model have strong spatial correlation, it is possible to determine whether a particular motion vector fits a motion model or not by the examination of its surrounding motion vectors. The proposed method can be used effectively in compressed video sequences containing global motion of the camera. The tensor voting framework [10,11], which is very robust to noise, is used to smooth the input motion vector field by the application of a voting process among the tensors that represent the motion vectors. The motion vector outliers can be easily detected by the comparison of the input motion vectors with the corresponding ones in the smoothed motion vector field. For this purpose, each motion vector in the input vector field is encoded by a 2-D stick tensor that is represented by a 2×2 symmetric matrix. A stick voting process is then applied to propagate direction information from each motion vector to its neighbors. The main directions of the resulting tensors are extracted to detect the outliers.

3.2. Tensor encoding. Tensors, an extension of scalars and vectors, provide a natural and concise mathematical framework for formulating problems in various areas of mathematics and physics [12,13]. They are also widely used in computer vision and image processing fields [14]. Voting methods based on tensors [10,11] are used to represent and analyze the saliency of each type of perceptual structure (curve, surface, junction, or region) to which a token may belong. In 2-D, a second order, symmetric, non-negative definite tensor, T, is represented by a 2×2 matrix and it is visualized by an ellipse. The axes of the ellipse are the eigenvectors of the tensor. Their lengths are proportional to the

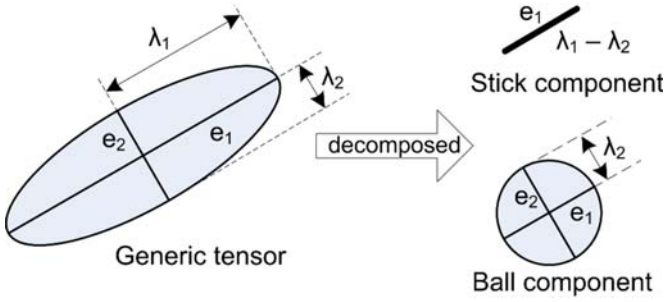
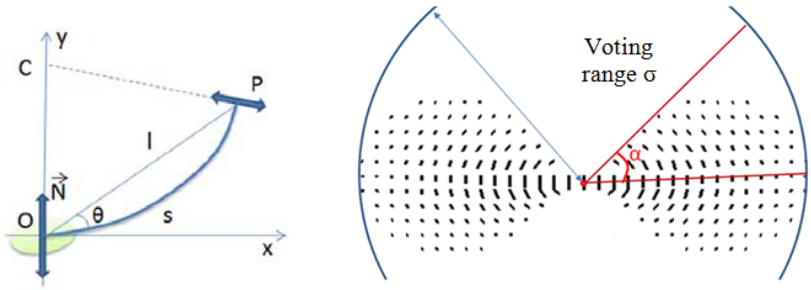


FIGURE 2. Tensor decomposition



(a) (b)

FIGURE 3. (a) 2-D stick voting process between 2 tensors and (b) 2-D stick voting field

values of the eigenvalues. This tensor, T, represented in Equation (5), can be decomposed into stick and ball components using Equation (6), as shown in Figure 2.

$$T = [\hat{e}_1\, \hat{e}_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \hat{e}_1^T \\ \hat{e}_2^T \end{bmatrix} \tag{5}$$

$$T = (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + \lambda_2(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) \tag{6}$$

where λ_i are the eigenvalues in a decreasing order and $\hat{e}_i$ are the corresponding eigenvectors. The major axis $\hat{e}_1$ is the preferred normal orientation of a potential curve segment. The magnitude of the stick component $(\lambda_1 - \lambda_2)$ indicates how certain it is that the corresponding point belongs to a curve. Each motion vector (MVx_i, MVy_i) of the input motion vector field is initialized by a stick tensor as follows:

$$T_i = \begin{bmatrix} (MVx_i)^2 & MVx_i * MVy_i \\ MVx_i * MVy_i & (MVy_i)^2 \end{bmatrix} \tag{7}$$

3.3. **Stick voting process.** A stick voting field is used to propagate each stick tensor into space in a certain neighborhood. Every token in the neighborhood defined by the stick voting field is included in this voting process. In order to calculate the total influence on a token, the tensor fields of all of its neighbors are simply summed. The voting kernel defines the most likely normal by selecting the most probable continuation curve between the two points, O (the voter) and P (the recipient), as shown in Figure 3(a). The vote received at P is a stick tensor with its main direction normal to the smooth curve at P. The magnitude of the vote is a function of confidence that the voter and the receiver indeed belong to the same perceptual structure. This magnitude can be calculated by the following equation:

$$DK(s, \kappa, \sigma) = e^{-\left(\frac{s^2 + c\kappa^2}{\sigma^2}\right)} \tag{8}$$

where the arc length $s = (l\theta)/\sin(\theta)$, the curvature $\kappa = 2\sin(\theta)/l$, σ is the scale of the voting field that controls the size of the voting neighborhood and the strength of the votes, and the parameter c (a function of the scale) is given by: $c = \frac{-16\log(0.1)\times(\sigma-1)}{\pi^2}$. Figure 3(b) shows a 2-D voting kernel for a voter who has a 90-degree normal vector and a cut-off angle α of 45 degrees. The tokens receive votes from their neighboring tokens by the voting process and encode them into new tensors.

3.4. **Motion vector outlier removal.** To remove outliers, the phase difference between the motion vectors in the input motion vector field and the ones in the smoothed motion vector field is used. First, the directions of the eigenvectors that correspond to the largest eigenvalues of the resulting tensors are used as the directions of the motion vectors in the smoothed tensor field. For block i, a similarity value S_i is calculated based on the cosine of the similarity between input motion vector and the one in the smoothed motion vector field, as follows:

$$S_i = |\cos(MV_i, MV_i')| = |\cos(MV_i, \hat{e}_{1,i})| \tag{9}$$

where MV_i, MV_i', and $\hat{e}_{1,i}$ are the input motion vector, smoothed motion vector, and eigenvector that correspond to the largest eigenvalue of the resulting tensor at block i. As the smoothed tensor voting field is taken as the reference motion vector field, the larger the value of S_i, the better this input motion vector fits the motion model. Therefore, the motion vectors that possess small similarity values are removed. The similarity of zero motion vectors is set to 0. In this research, 30% of the motion vectors were found to be outliers, and this is similar to the work in [4].

Figure 4 illustrates the steps in the proposed method. The synthetic motion vector field generated by using a perspective model $m = [1, 0, 4.4154, 0, 1, 0, -1, -1.13e-4, 0]$ is depicted in Figure 4(a). The image size is 288×352, and the block size is set to 16. The indices of the blocks are shown on the horizontal and vertical axes. Figure 4(b) shows the corrupted motion vector field which is generated by the addition of the independent zero-mean Gaussian noise with standard deviation SD = 1.5 to the motion vector field in Figure 4(a). The smoothed tensor voting field after the application of a 2-D stick tensor voting process is presented in Figure 4(c). Note that in the smoothed motion vector field, the direction of each vector is the direction of the eigenvector which corresponds to the largest eigenvalue and the magnitude is the magnitude of the corresponding input motion vector. Through the voting process, the input motion vectors are refined according to their neighbors. Therefore, the directions of the smoothed motion vectors are similar to those of the original ones. Figure 4(d) shows 30% motion vector outliers (vectors with rectangles) that correspond to the 30% smallest similarity values.

4. **Experimental Results.** At first, synthetic motion vector fields are used for evaluation. The same four sets of global motion parameters as in [2,4], which are shown in Table 1, are used. The synthetic motion vector fields of these four motion models are illustrated in Figure 5. The images have CIF resolution and are divided into 16×16

TABLE 1. Global motion parameters of four test models [2,4]

Model	Model parameters
M1	$m = [0.95, 0, 10.4238, 0, 0.95, 5.7927, 0, 0]$
M2	$m = [0.9964, -0.0249, 1.0981, 0.0856, 0.9457, -7.2, 0, 0]$
M3	$m = [0.9964, -0.0249, 6.0981, 0.0249, 0.9964, 2.5109, -2.7e-5, 1.9e-5]$
M4	$m = [1, 0, 4.4154, 0, 1, 0, -1, -1.13e-4, 0]$

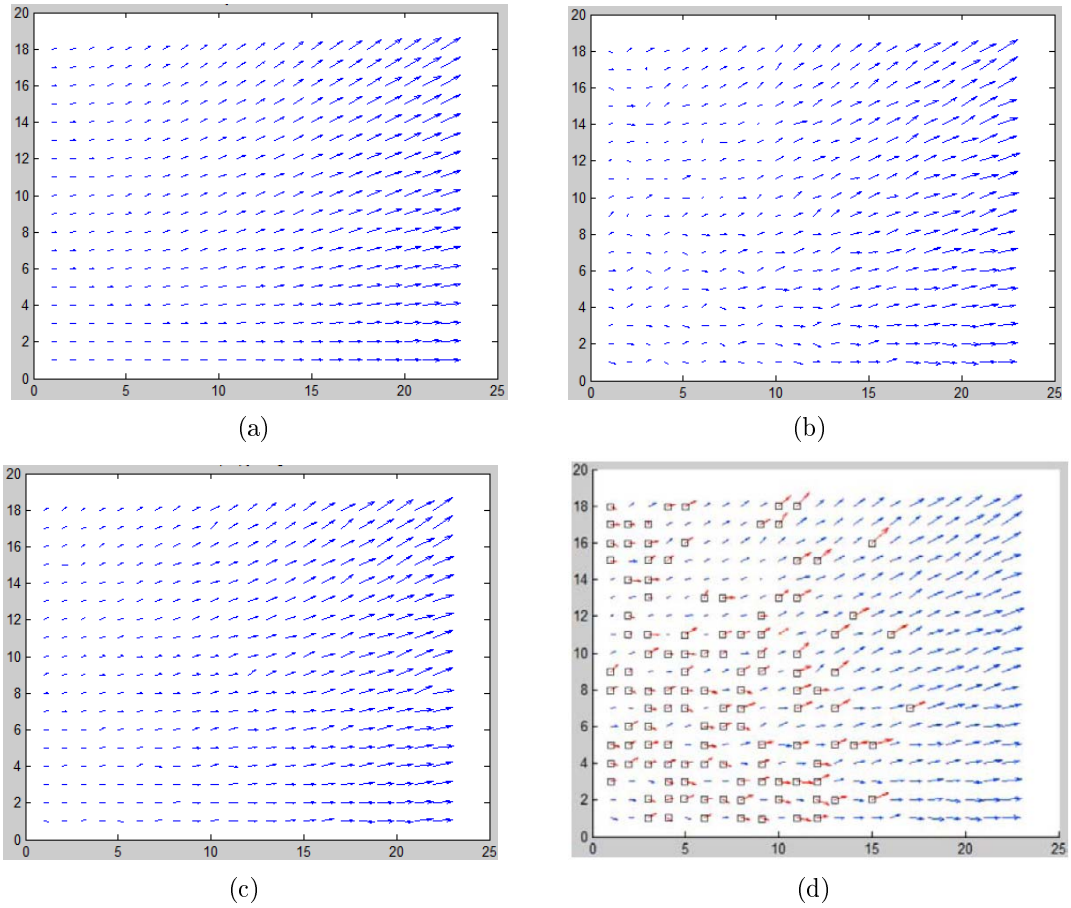
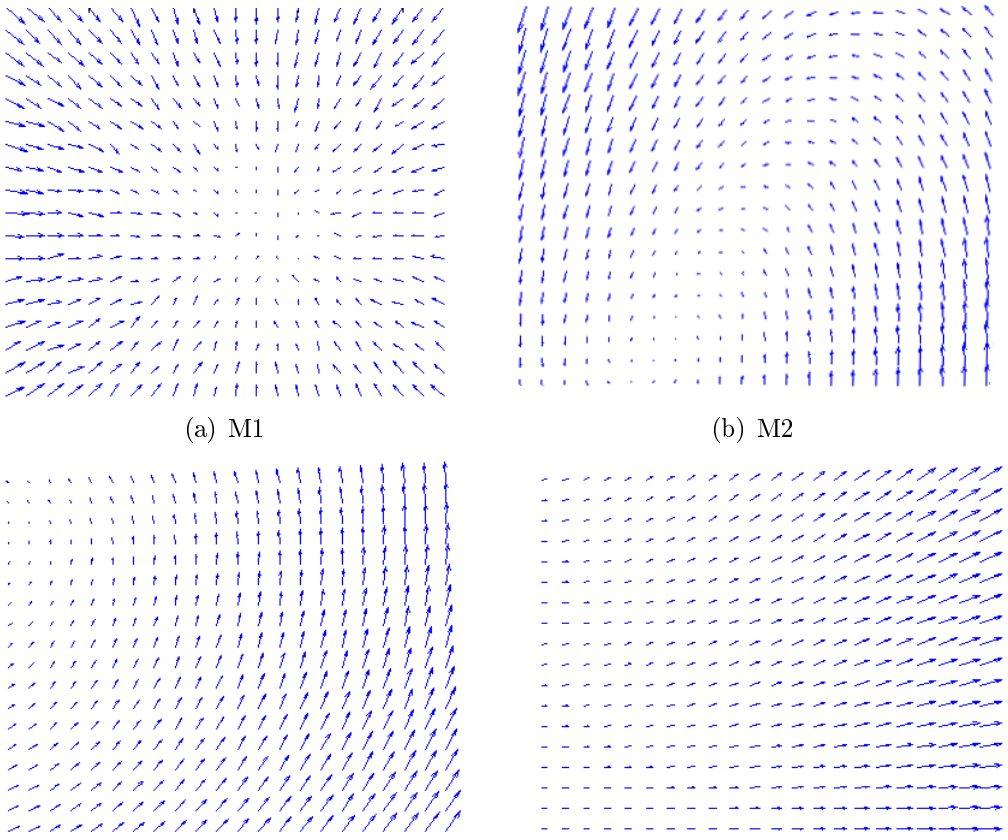


FIGURE 4. Outlier removal using tensor voting: (a) synthetic motion vector field, (b) motion vectors are corrupted by noise, (c) smoothed motion vector field using tensor voting, (d) outlier motion vectors (vectors with rectangles)

blocks. To generate input motion vector fields for GME methods, the motion vectors in the synthetic motion vector fields are corrupted by independent zero-mean Gaussian noise in both horizontal and vertical directions. Four levels of the standard deviation of the Gaussian noise are used: SD = {0.7, 1.5, 2.2, 3.0}. From the corrupted input motion vector fields, new global motion parameters are estimated by using GME methods. The signal-to-noise criterion (SNR) is then calculated between the synthetic motion vector field generated by the original global motion parameters and the one generated by the estimated global motion parameters. Five powerful approaches are selected for comparison: iterative gradient descent (GD) [2], the filter method (FLT_GD) as described in [3], the cascade filter method (CAS_GD) as described in [4], RANSAC with least-square regression (RAN_LS) [9], and the least square solution using an M-estimator (LSS_ME) [7]. Figure 6 shows a performance comparison between the proposed method (TV_GD) and the five comparison ones. In the proposed method, the number of iterations in GD is set to 2, and the voting range is set to 50. Thirty percent of the motion vectors are considered as outliers in the TV_GD and CAS_GD methods. Each method is run 50 times, and the average value is calculated. The experimental results show that the proposed method generates a better SNR than other state-of-the-art methods. By correct removal of the outliers, the proposed method improves the results of the GD method. As expected, the proposed method performs well in high noise motion vector fields by using the tensor



(a) M1 (b) M2

(c) M3 (d) M4

FIGURE 5. Synthetic motion vector fields of test models

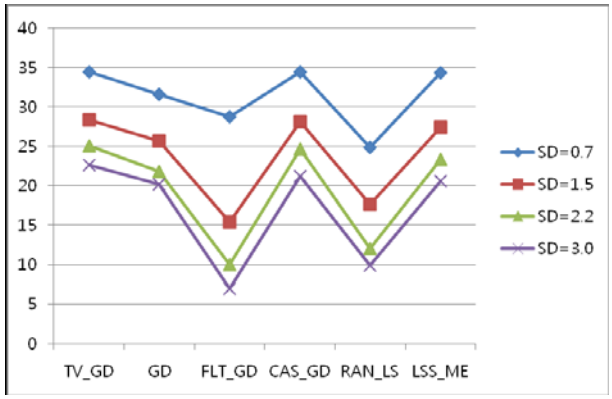
TABLE 2. Average number of iterations

Algorithm	SD = 0.7	SD = 1.5	SD = 2.2	SD = 3.0
TV_GD	**2**	**2**	**2**	**2**
GD	6	6	6	6
FLT_GD	2	2	2	2
CAS_GD	2	2	2	2
RAN_LS	14	132	> 500	> 500
LSS_ME	4	4	4	4

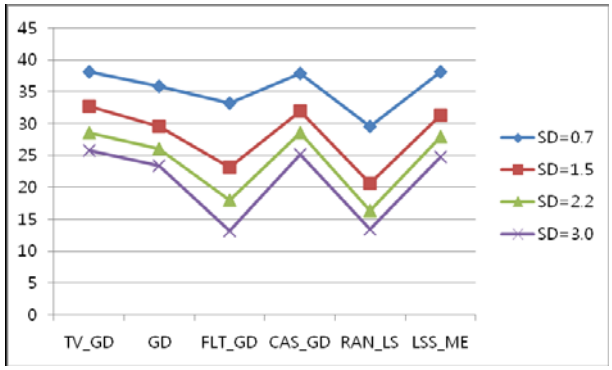
voting framework. As mentioned in [4], FLT_GD achieves very low SNR performance as it tends to remove too many input motion vectors.

By the removal of outliers before feeding the motion vector into the GD method, the number of iterations in the GME is also reduced. Table 2 shows the average number of iterations for the convergence for each of the GME method. The proposed method converges much faster than the GD, RAN_LS, and LSS_ME methods. Only two GME iterations are needed to achieve satisfactory results.

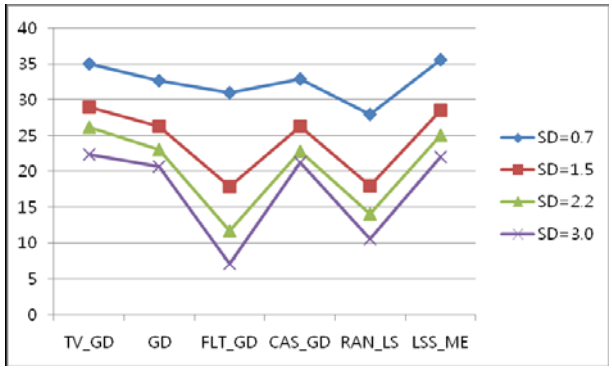
The only input parameter of the proposed method is the voting range. Fortunately, the tensor voting results are not very sensitive to the values of the voting range [10,11]. Figure 7 shows the SNR performance of the proposed method with different values of the voting range. Moreover, the proposed method can generate stable results with different values of the voting range.



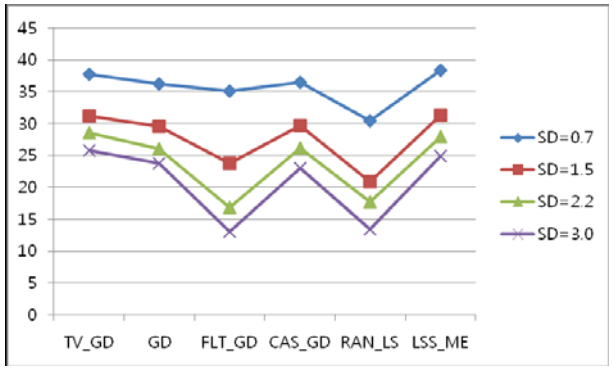
(a) M1



(b) M2



(c) M3



(d) M4

FIGURE 6. SNR comparison of motion vector fields which are corrupted by Gaussian noise

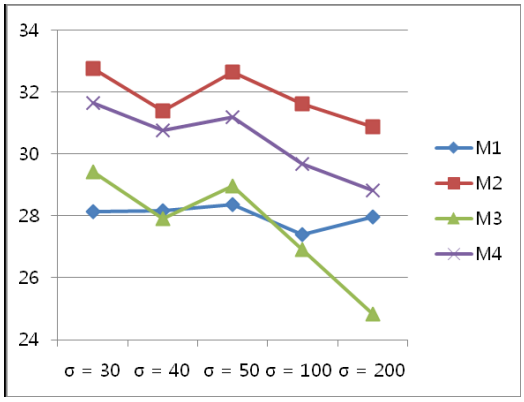


FIGURE 7. SNR performance with different values of the voting range (the standard deviation of Gaussian noise = 1.5)

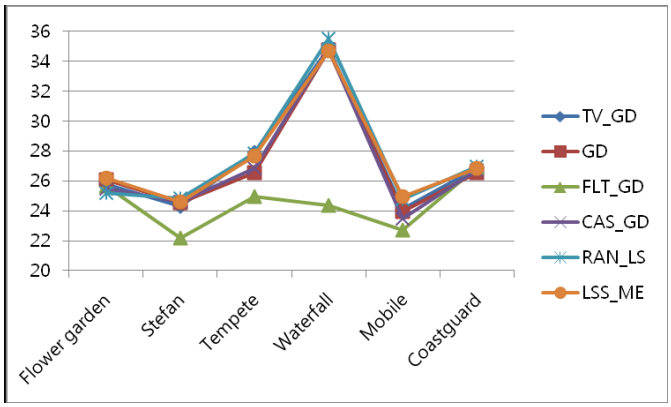


FIGURE 8. PSNR comparison of GME methods

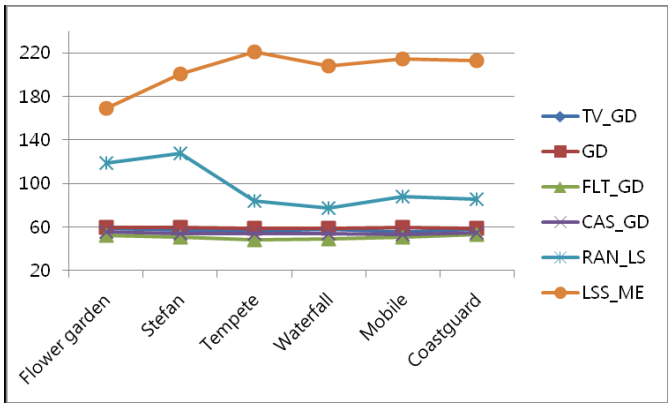


FIGURE 9. Processing time (ms) comparison of GME methods

The real test sequences that contain mostly camera motion are used to evaluate the performance of the motion vector based GME methods. The motion vectors estimated by the full-search motion estimation procedure are used to estimate the motion model. Global motion compensation is then applied by warping the current frame to the reference image plane. This is done according to the estimated motion model with bilinear interpolation, as described in [4]. The conventional PSNR is measured between the original reference frame and the compensated one. Figure 8 shows the PSNR performance of all GME methods on six test sequences. The average processing time per frame of the GME

methods is shown in Figure 9. All algorithms are implemented in MATLAB and they are run on an Intel Core 2 Quad CPU with a 2.66-GHz processor speed, 3GB RAM, and Windows 7. In this experiment, the block size is set to 8×8 and the voting range is set to 30. The number of iterations in the GME is set to one for the TV_GD, CAS_GD, and FLT_GD methods, six for the GD method, and three for the LSS_ME method. In the proposed method, most of the computational time is spent in the tensor voting process. In order to reduce process time, the closed form tensor voting [15] which is implemented in the C programming language was used.

The FLT_GD method is the fastest, but it generates the worst result. The RAN_LS and LSS_ME methods give good results, but they impose high computational costs. The proposed method is as fast as the CAS_GD method and it yields a good PSNR. Thus, the proposed method provides a good tradeoff between accuracy and computational requirement.

5. **Conclusions.** An efficient motion vector outlier removal method based on tensor voting is proposed. 2-D tensor voting is used to make use of the fact that motion vectors that are generated by the same global motion model have a strong spatial correlation. By using the phase difference between the input motion vectors and the smoothed ones, outliers are effectively detected. The experimental results show that the proposed method generates good results for both the synthetic and real motion vectors, especially as the noise variance increases.

Acknowledgement. This research was supported by Basic Science Research Program through the National Research of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0006109) and by Basic Science Research Program through the National Research of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0029429).

REFERENCES

[1] A. Krutz, M. Frater, M. Kunter and T. Sikora, Windowed image registration for robust mosaicing of scenes with large background occlusions, *Proc. of ICIP*, pp.353-356, 2006.

[2] Y. Su, M.-T. Sun and V. Hsu, Global motion estimation from coarsely sampled motion vector field and the applications, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.15, pp.232-242, 2005.

[3] A. Dante and M. Brookes, Precise real-time outlier removal from motion vector fields for 3D reconstruction, *Proc. of ICIP*, vol.1, pp.I-393-6, 2003.

[4] Y.-M. Chen and I. V. Bajic, Motion vector outlier rejection cascade for global motion estimation, *IEEE Signal Processing Letters*, vol.17, pp.197-200, 2010.

[5] M. Haller, A. Krutz and T. Sikora, Evaluation of pixel- and motion vector-based global motion estimation for camera motion characterization, *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, pp.49-52, 2009.

[6] X. Zhang, The improvement of a feature-based image mosaics algorithm, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2759-2764, 2008.

[7] A. Smolic, M. Hoeynck and J. R. Ohm, Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications, *Proc. of ICIP*, vol.2, pp.271-274, 2000.

[8] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. of the ACM*, vol.24, pp.381-395, 1981.

[9] M. Hartley, A. Krutz and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Edition, Cambridge University Press, 2004.

[10] C.-K. Tang and G. Medioni, Inference of integrated surface, curve and junction descriptions from sparse 3D data, *IEEE Trans. on PAMI*, vol.20, pp.1206-1223, 1998.

[11] G. Medioni, M. S. Lee and C.-K. Tang, *A Computational Framework for Segmentation and Grouping*, Elsevier, Amsterdam, 2000.

[12] I. Borisenko and I. E. Tarapov, *Vector and Tensor Analysis with Applications*, Dover, New York, 1968.
[13] E. C. Young, *Vector and Tensor Analysis*, New York, 1993.
[14] S. Aja-Fernndez, R. de L. Garca, D. Tao and X. Li, *Tensors in Image Processing and Computer Vision*, Springer Publishing Co., 2009.
[15] T.-P. Wu, J. Jia and C.-K. Tang, A closed-form solution to tensor voting for robust parameter estimation via expectation-maximization, *Technical Report*, HKUST, 2009 .