Motion Vector Outlier Rejection Cascade for Global Motion Estimation

Yue-Meng Chen and Ivan V. Bajić, *Member, IEEE*

***Abstract*—Global motion estimation (GME) from motion vector (MV) field in compressed domain greatly reduces the complexity of conventional pixel-based GME. However, outlier MVs, caused by noise or foreground objects, may reduce the accuracy of MV-based GME. In this paper, we propose a cascade-of-rejectors approach for removing MV outliers to achieve efficient and accurate GME. Experimental results show that the proposed MV outlier rejection cascade significantly lowers the complexity MV-based GME, with an accuracy close to or better than state-of-the-art methods.**

***Index Terms*—Global motion estimation, outlier removal.**

I. Introduction

GLOBAL MOTION ESTIMATION (GME) is used to estimate camera motion in a video sequence, which can be useful in content-based video analysis, such as video object segmentation, background modeling, video indexing, etc. GME can be done in either pixel domain [1], [2] or compressed domain [3]–[5]. The compressed domain approaches utilize coarsely sampled (i.e., block-based) motion vector (MV) field from compressed video, and therefore greatly reduce the computational complexity of GME compared to pixel-based approaches. However, MVs in the compressed bitstream are often imperfect and inconsistent with real camera motion. To remove the outliers poorly fit into the global motion model, an iterative approach is usually used [3], but the computational cost of outlier removal can be fairly high.

In [6], Dante and Brookes proposed a MV outlier removal method for epipolar geometry, where the detection of an outlier is accomplished by examining the magnitude difference between a MV and its 8-neighbors. Our proposed method extends this work by also examining the phase difference among neighboring MVs, and by replacing the hard-decision thresholding from [6] with a soft-decision removal of a prescribed fraction of worst-fitting MVs from the MV field. The proposed method can help significantly reduce the number of iterations in a state-of-the-art GME method from [3], while simultaneously improving its accuracy.

The paper is organized as follows. In Section II, we investigate the smoothness of the MV field generated by pure camera motion, and deduce several statistical parameters which are subsequently used to set the filter thresholds in the proposed MV outlier rejection cascade. The cascade itself is described in Section III, the experimental results are presented in Section IV, and the conclusions are drawn in Section V. The work was performed in a reproducible research manner, and the MATLAB code needed to reproduce all reported results is available at http://www.sfu.ca/~ibajic/software.html.

Manuscript received October 11, 2009; revised November 09, 2009. First published November 17, 2009; current version published December 23, 2009. This work was supported in part by the NSERC Grant STPGP 350740. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lisimachos Paul Kondi.

The authors are with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6 Canada (e-mail: yuemengc@sfu.ca; ibajic@ensc.sfu.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2009.2036879

II. Characterization of the Motion Vector Field Generated by Camera Motion

In [3], four 2-D motion models (translational, geometric, affine and perspective) are summarized, with the eight-parameter perspective model being the most general one. The perspective model is described by a vector of its parameters, $\mathbf{m} = [m_0, \ldots, m_7]$. Given (x, y) and (x', y') as the coordinates in the current and the reference frame, respectively, the perspective transformation is defined as:

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}, \quad y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}. \quad (1)$$

Defining $\mathbf{x} = [x, y, 1]^T$ and $\mathbf{x}' = [N_x, N_y, D]^T$, where N_x and N_y are the numerators in (1) and D is the denominator, (1) can be represented by a homographic mapping:

$$\begin{aligned} \mathbf{x}' &= \mathbf{H}(\mathbf{m}) \cdot \mathbf{x} \\ &= \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \\ x' &= N_x/D, \quad y' = N_y/D. \end{aligned} \quad (2)$$

The homographic mapping can also be decomposed into a product of matrices containing the focal lengths and rotation angles [7]. In this model, the X- and Y-components of the MV field at (x, y) in the current frame are given by:

$$\mathbf{MV}^X(x, y; \mathbf{m}) = x' - x, \quad \mathbf{MV}^Y(x, y; \mathbf{m}) = y' - y. \quad (3)$$

MVs that come from a given motion model (a given vector of motion parameters $\mathbf{m}$) usually exhibit fairly strong spatial correlation. We will use this property in our proposed outlier rejection cascade to remove the MVs that do not seem to fit the model. In order to estimate how similar or how different the neighboring MVs from a given model can be, we performed the following experiment. We used a range of camera parameters in the homographic mapping that are commonly found in practice [8], [9]: focal length: 200–1000; focal length change ratio between consecutive frames: 0.9–1; angular velocity: [−1.6, 1.6] degrees per frame for x-, y- and z-axis. We created 14 000 combinations

TABLE I
THE 90TH PERCENTILE OF MAGNITUDE AND PHASE DIFFERENCE

Block Size	D_{mag}	D_{ph} (Degrees)
32×32	1.0	45
16×16	0.4	19
8×8	0.2	9
4×4	0.1	4

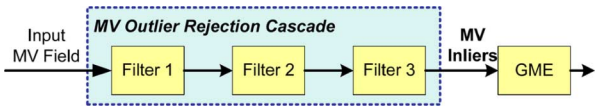

Fig. 1. Proposed MV outlier removal cascade.

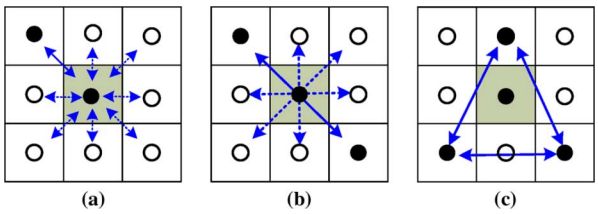

Fig. 2. Construction of MVs in S_i^j, (a): S_i^1, (b): S_i^2, and (c): S_i^3.

of camera parameters from these ranges, and for each combination, we synthesized an MV field in floating-point MV precision according to (2) and (3). We then measured the average relative difference in magnitude (D_{mag}) and phase (D_{ph}) between each MV and its 8-neighborhood. The histograms of these quantities were then created, and the 90th percentile was computed. Assuming a CIF (352 × 288) resolution video, Table I lists the 90th percentile values of D_{mag} and D_{ph} for block sizes of 4 × 4, 8 × 8, 16 × 16, and 32 × 32 pixels. For example, for 16 × 16 blocks, the 90th percentile for D_{mag} was 0.4, and the 90th percentile for D_{ph} was 19 degrees, meaning that 90% of the MVs in a MV field described by a perspective model, with the camera parameters from the ranges listed above, have the average relative magnitude difference from their neighbors of 0.4 or less, and the average phase difference of 19 degrees or less. These 90th percentile values are used in the next section to set the thresholds in the proposed outlier rejection cascade.

III. MV OUTLIER REJECTION CASCADE

The cascade-of-rejectors approach has been very successful in fast object detection [10], where it is used to quickly verify the presence or absence of certain object features. We propose using a similar approach to remove outliers from an input MV field, in order to speed up GME. The proposed cascade consists of three filters, as shown in Fig. 1. Input MV field is subject to testing in the first filter, then the MVs declared as inliers are further tested in the second filter, and so on.

To test each input MV, the filters in the cascade employ the following strategy. Let $\mathbf{MV}_i$ be the input MV to be tested in filter j, where $j \in \{1,2,3\}$. Associated with $\mathbf{MV}_i$ is the set S_i^j of MVs computed from the 8-neighborhood of $\mathbf{MV}_i$ as shown in Fig. 2, where the location of $\mathbf{MV}_i$ is shown in gray. For filter 1, S_i^1 consists of individual MVs from the neighborhood of $\mathbf{MV}_i$, as shown in Fig. 2(a). For filter 2, S_i^2 consists of the averages of diagonally opposite MVs from the neighborhood of $\mathbf{MV}_i$, as shown in Fig. 2(b). Finally, for filter 3, S_i^3 consists of the averages of triangularly opposite MVs from the neighborhood of $\mathbf{MV}_i$, as shown in Fig. 2(c). There are at most eight MVs in S_i^1, and at most four MVs in each of S_i^2 and S_i^3.

TABLE II
MAGNITUDE AND PHASE THRESHOLDS

Filter j	$T^j{}_{mag}$		$T^j{}_{ph}$ (Degrees)	
	16×16	8×8	16×16	8×8
1	0.4	0.2	19	9
2	0.2	0.1	9.5	4.5
3	0.1	0.05	4.75	2.25

Once S_i^j is constructed, we test the following conditions for each $\mathbf{MV}_k \in S_i^j$, and count how many times the following conditions are satisfied:

$$\|\mathbf{MV}_i - \mathbf{MV}_k\| / \|\mathbf{MV}_i\| < T_{mag}^j, \tag{4}$$

$$|\varphi(\mathbf{MV}_i) - \varphi(\mathbf{MV}_k)| < T_{ph}^j \tag{5}$$

where T_{mag}^j and T_{ph}^j are the thresholds for maximum relative magnitude difference, and maximum phase difference, respectively. To avoid the computation of phase $\varphi(\cdot)$, (5) can be rewritten as $\langle \mathbf{MV}_i, \mathbf{MV}_k \rangle > \|\mathbf{MV}_i\| \cdot \|\mathbf{MV}_k\| \cdot \cos(T_{ph}^j)$. Let N_i^j be the number of times the above conditions are satisfied. Note that $N_i^1 \leq 16$, and $N_i^j \leq 8$ for $j \in \{2,3\}$. The weighted count is given by $WN_i^j = W_i^{j-1} \cdot N_i^j$, where $W_i^j = \exp(-(WN_{\max}^j - WN_i^j))$, $WN_{\max}^j = \max_i WN_i^j$ and $W_i^0 = 1$ for all i. The weight W_i^j is a measure of how similar is $\mathbf{MV}_i$ to vectors in S_i^j.

The magnitude and phase thresholds in (4) and (5) are determined based on the 90th percentile values for the relative magnitude and phase difference found in Section II. The thresholds for MV field with 16 × 16 and 8 × 8 blocks are set as shown in Table II. For filter 1, the thresholds are equal to the 90th percentile values from Table I. These thresholds are halved for filter 2, and further halved for filter 3.

Each of the three filters in the cascade is set to keep the same fraction of inliers in order to satisfy the target fraction of inliers. If $p \in [0,1]$ is the fraction of inliers we want from the cascade, then each filter is set to keep $p^{1/3}$ of the input MVs, and remove the rest as outliers. For example, if we want to keep 70% of MVs as inliers, then $p = 0.7$, $p^{1/3} \approx 0.888$, so each filter will keep approximately 88.8% of its input MVs as inliers, and remove 11.2% as outliers. The filtering operation is summarized as follows.

1) Symmetrically extend the MV field across frame boundaries, flag all MVs as inliers, and set $j = 1$.
2) For each inlier $\mathbf{MV}_i$, find the weighted count WN_i^j. Note that previously declared outlier MVs are included in the neighborhoods (S_i^j) of inlier MVs.
3) Sort MVs in descending order of their weighted counts.
4) Declare the target number of MVs at the bottom of the sorted list as outliers.
5) If $j = 3$, then stop. Otherwise, set $j = j + 1$, and move on to the next filter, repeating steps 2–5.

IV. EXPERIMENTAL RESULTS

An extensive evaluation of MV-based GME approaches was conducted in [11], where the iterative Gradient Descent (GD)

TABLE III
TEST GLOBAL MOTION PARAMETERS

Model	Motion parameters
GM 1	**m**=[0.9, 0, 10.4238, 0, 0.95, 5.7927, 0, 0]
GM 2	**m**=[0.9964, −0.0249, 1.0981, 0.0856, 0.9457, −7.2, 0, 0]
GM 3	**m**=[0.9964, −0.0249, 6.0981, 0.0249, 0.9964, 2.5109, −2.7e−5, 1.9e−5]
GM 4	**m**=[1, 0, 4.4154, 0, 1, 0, −1.13e−4, 0]

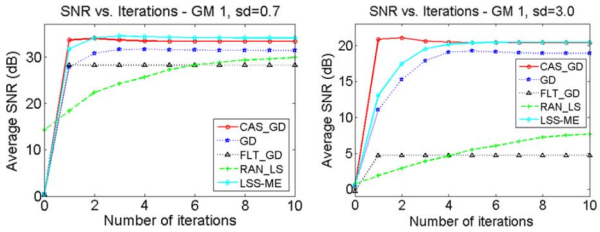

Fig. 3. SNR versus iterations with synthetic MV field (**GM 1**) corrupted by Gaussian noise with standard deviation {0.7, 3.0}.

approach [3], the least square solution using an M-Estimator (LSS-ME) [4], and RANdom SAmple Consensus (RANSAC) [13] were identified as powerful approaches used to compute GM parameters from MVs. We chose GD as the platform to examine the effects of using the proposed cascade to pre-process the MV field (CAS_GD), as illustrated in Fig. 1. We compare MV outlier rejection capabilities of our cascade against the filter from [6] (FLT_GD). We also implemented LSS-ME [4] and RANSAC with least-square regression (RAN_LS) [12] for further performance comparison. Parameter C in LSS-ME is set to 2 in this work.

A. Evaluation on Synthetic MV Fields

We use the same four sets of motion parameters as in [3], shown in Table III, to synthesize the test MV fields. These models also fall within the range of parameters used in Sections II and III to set the filter thresholds. After the MV field is synthesized, assuming CIF resolution and 16 × 16 blocks, as in [3], the MVs are corrupted by independent zero-mean Gaussian noise in both X- and Y-components, and outlier MVs (groups of connected MVs pointing in a random direction) simulating foreground moving objects. As in [3], the performance criterion is the signal-to-noise ratio (SNR) between the MV field generated by parameters $\mathbf{m}$ (Table III), and the MV field generated by the estimated parameters $\hat{\mathbf{m}}$.

Fig. 3 shows the results of five GME approaches in terms of SNR versus the number of GME iterations. In this experiment, MV field, generated using parameters from GM 1, is only corrupted by noise. Simulation results with $\sigma \in \{0.7, 3.0\}$ are shown in Fig. 3 (the results with $\sigma \in \{1.5, 2.2\}$, as well as results with other models from Table III, follow similar trends). Each result is averaged over 50 runs. The filters in our cascade were set to give 70% of inliers overall, in order to facilitate a fair comparison with the results in [3]. We observe that the CAS_GD converges faster than GD, LSS-ME and RANSAC, and achieves a very close SNR to LSS-ME. Both CAS_GD and LSS-ME have a higher SNR than other methods, especially as the noise variance increases. In Table IV, we list the converged SNR values of these four methods for all four GM models from Table III. FLT_GD yields the worst performance among the tested methods, because the filter from [6] usually removes too many input MVs.

TABLE IV
SNR IN THE MV FIELD (dB), CORRUPTED BY ONLY GAUSSIAN NOISE

GM Model	GME Algorithms	Standard Deviation of Gaussian Noise 0.7	1.5	2.2	3.0
GM 1	**CAS_GD**	**33.98**	**27.60**	**23.87**	**21.50**
	GD	31.71	24.63	21.85	20.11
	FLT_GD	28.15	15.83	11.19	8.73
	RAN_LS	29.61	17.60	13.58	9.08
	LSS-ME	34.23	27.79	23.47	20.56
GM 2	**CAS_GD**	**37.28**	**31.28**	**27.92**	**25.11**
	GD	35.02	29.01	26.33	23.01
	FLT_GD	33.92	23.04	17.37	13.62
	RAN_LS	31.57	20.87	17.28	14.20
	LSS-ME	38.11	31.12	27.39	24.65
GM 3	**CAS_GD**	**33.65**	**27.16**	**23.05**	**21.02**
	GD	33.01	26.53	23.73	21.18
	FLT_GD	30.49	18.43	12.23	9.77
	RAN_LS	29.38	18.18	14.51	12.39
	LSS-ME	34.64	29.05	25.51	21.55
GM 4	**CAS_GD**	**37.67**	**30.64**	**26.39**	**23.19**
	GD	35.56	29.11	26.22	23.14
	FLT_GD	34.51	23.27	17.29	13.53
	RAN_LS	33.72	21.27	17.78	14.26
	LSS-ME	38.11	31.48	28.11	24.23

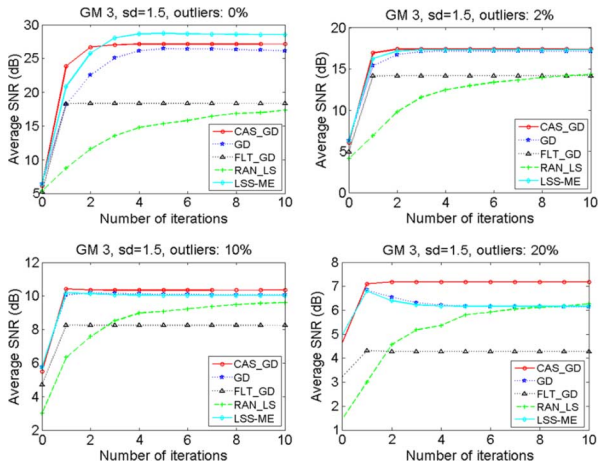

Fig. 4. SNR versus iterations with synthetic MV field (**GM 3**) corrupted by Gaussian noise ($\sigma = 1.5$) and outliers {0%, 2%, 10%, 20%}.

Next, we corrupted the MV fields by both noise ($\sigma = 1.5$) and outliers made up of connected regions of 3 × 3, 6 × 6, 9 × 9 MVs in the center of the frame (2%, 10% and 20% of the total MV field size). These MV outliers are generated by adding the vector (5, 5) to the MVs generated by the global motion model. The results for GM 3 are shown in Fig. 4, where we can observe that the performance of RANSAC improves compared to other methods as the percentage of MV outliers increases. Average (converged) SNRs for all four GMs are listed in Table V, while the number of iterations needed to achieve SNRs from Tables IV and V is listed in Table VI.

B. Evaluation on MV Fields From Real Test Sequences

Another way to test MV-based GME is to estimate the MVs in a test sequence that contains mostly camera motion, use GME on these MVs to estimate the model, and perform global motion compensation by warping the target frame onto the reference image plane according to the model using bilinear interpolation [11]. If the sequence indeed contains only camera motion, and if GME is accurate, we should expect the frames

TABLE V
SNR IN THE MV FIELD (dB), CORRUPTED BY BOTH NOISE AND OUTLIERS

GM Model	GME Algorithms	Outlier Percentage ($\sigma = 1.5$) 0 %	2 %	10 %	20 %
GM 1	**CAS_GD**	**27.56**	**16.16**	**8.74**	**5.35**
	GD	24.75	15.81	8.57	4.40
	FLT_GD	15.46	12.97	7.97	3.58
	RAN_LS	17.13	13.44	7.94	4.68
	LSS-ME	27.67	15.93	8.52	4.42
GM 2	**CAS_GD**	**31.24**	**19.59**	**11.64**	**6.90**
	GD	29.31	19.38	11.59	6.46
	FLT_GD	23.05	17.82	11.25	6.65
	RAN_LS	20.34	16.69	11.01	6.32
	LSS-ME	31.30	19.47	11.49	6.47
GM 3	**CAS_GD**	**26.81**	**17.35**	**10.38**	**7.19**
	GD	27.04	17.21	10.12	6.15
	FLT_GD	17.54	13.61	8.20	3.11
	RAN_LS	18.14	14.96	9.66	5.90
	LSS-ME	28.32	17.31	10.04	6.17
GM 4	**CAS_GD**	**30.06**	**20.13**	**12.66**	**8.93**
	GD	29.11	19.93	12.43	7.93
	FLT_GD	23.20	18.99	11.81	6.01
	RAN_LS	21.10	17.37	11.97	7.77
	LSS-ME	31.24	20.06	12.39	7.79

TABLE VI
AVERAGE NUMBER OF ITERATIONS TO ACHIEVE PERFORMANCE ABOVE

GME Algorithms	Noise only (No outliers) $\sigma \in \{0.7, 1.5, 2.2, 3.0\}$ 0.7	1.5	2.2	3.0	Outliers and noise ($\sigma = 1.5$) 2%	10%	20%
CAS_GD	**2**	**2**	**2**	**2**	**2**	**2**	**2**
GD	6	6	6	6	6	6	6
FLT_GD	2	2	2	2	2	2	2
RAN_LS	14	132	>500	>500	144	199	327
LSS-ME	4	4	4	4	4	4	4

TABLE VII
(Top) GLOBAL MOTION COMPENSATION PERFORMANCE (PSNR in dB). (Bottom) GLOBAL MOTION COMPENSATION SPEED (TIME IN ms)

Sequences	CAS_GD	FLT_GD	GD	RAN_LS	LSS-ME
Flower Garden	22.19	21.44	22.30	21.87	22.48
Stefan	24.60	22.16	24.51	24.74	24.60
City	29.48	29.25	28.70	29.62	29.88
Tempete	27.83	24.98	26.51	27.86	27.66
Waterfall	34.86	24.25	34.71	35.48	34.69
Mobile	23.47	22.69	23.91	24.72	24.97
Coastguard	26.78	26.90	26.54	26.97	26.82
Average	**27.03**	**24.52**	**26.73**	**27.32**	**27.30**

Sequences	CAS_GD	FLT_GD	GD	RAN_LS	LSS-ME
Flower Garden	25.3	16.9	43.1	137.6	298.6
Stefan	25.0	16.2	42.2	179.2	442.0
City	25.1	15.9	44.0	143.2	479.3
Tempete	24.9	15.8	42.5	98.9	488.2
Waterfall	23.9	15.0	42.7	97.1	478.5
Mobile	24.8	15.3	42.2	117.4	477.5
Coastguard	24.8	16.2	44.2	117.7	464.6
Average	**24.8**	**15.9**	**43.0**	**127.3**	**446.9**

compensated by global motion to be very close to the original frames. The similarity can be measured using the conventional PSNR. We performed this experiment on seven test sequences listed in Table VII. Exhaustive search on 8×8 blocks is used to estimate the MVs prior to GME. We compare the same five GME methods as in the previous section. This time, CAS_GD and FLT_GD are followed by a single iteration of GD, plain GD uses six iterations while LSS-ME set to use three iterations. The thresholds for 8×8 blocks listed in Table II are used in our cascade. The total processing time per frame was measured in MATLAB on a standard desktop PC with Intel Pentium CPU at 3.0 GHz, with 2 GB of RAM. This processing time includes all filtering and GME operations.

Table VII lists the average PSNR in dB and the average processing time on the seven test sequences. FLT_GD yields the fastest performance (36% faster than our CAS_GD), but also has the lowest PSNR performance (about 2.5 dB worse than our CAS_GD). Its PSNR performance, compared to other methods, seems to be especially poor on sequences with small camera motion, like *Waterfall*, where its thresholding strategy removes too many MVs. Our CAS_GD is the next fastest method, and achieves better PSNR (by about 0.3 dB) than plain GD, simultaneously with a 73% speedup compared to plain GD. Finally, RAN_LS and LSS-ME give the best PSNR (both about 0.3 dB higher than our CAS_GD), but at a significantly higher computational cost. Overall, our CAS_GD gives the best tradeoff between accuracy and complexity. These results should be taken with a grain of salt, though, because the motion present in these sequences is not entirely due to camera motion. Nonetheless, the results provide some insight into the GME performance that can be expected on real sequences.

V. CONCLUSION

We have proposed a cascade-of-rejectors approach for removing outliers from the MV field prior to Global Motion Estimation (GME). The proposed approach was tested on both real and synthetic MV fields, and the results indicate that it can significantly reduce the complexity of conventional GME while achieving similar or better accuracy.

REFERENCES

[1] A. Krutz, M. Frater, M. Kunter, and T. Sikora, "Windowed image registration for robust mosaicing of scenes with large background occlusions," in *Proc. IEEE ICIP'06*, Oct. 2006, pp. 353–356.

[2] H. Alzoubi and W. D. Pan, "Efficient global motion estimation using fixed and random subsampling patterns," in *Proc. IEEE ICIP'07*, Sep. 2007, pp. 477–480.

[3] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 232–242, Feb. 2005.

[4] A. Smolic, M. Hoeynck, and J.-R. Ohm, "Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 application," in *Proc. IEEE ICIP'00*, Sep. 2000, pp. 271–274.

[5] R. Wang and T. S. Huang, "Fast camera motion analysis in MPEG domain," in *Proc. IEEE ICIP'99*, Oct. 1999, pp. 353–356.

[6] A. Dante and M. Brookes, "Precise real-time outlier removal from motion vector fields for 3D reconstruction," in *Proc. IEEE ICIP'03*, Sep. 2003, pp. 393–396.

[7] M. Kunter, A. Krutz, M. Mandal, and T. Sikora, "Optimal multiple sprite generation based on physical camera parameter estimation," in *Proc. VCIP'07*, Jan. 2007, vol. 6508, no. 2, pp. 0B.1–0B.10.

[8] S. Tubaro and S. Rocca, "Motion field estimators and their application to image interpolation," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Norwell, MA: Kluwer, 1993, pp. 153–187.

[9] D. Farin, "Automatic Video Segmentation Employing Object/Camera Modeling Techniques," Ph.D. Thesis, Technische Univ. Eindhoven, Eindhoven, Netherlands, 2005.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE CVPR'01*, Dec. 2001, pp. 511–518.

[11] M. Haller, A. Krutz, and T. Sikora, "Evaluation of pixel- and motion vector-based global motion estimation for camera motion characterization," in *Proc. WIAMIS'09*, May 2009, pp. 49–52.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2004.

[13] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.