ROBUST GLOBAL MOTION ESTIMATION USING MOTION VECTORS OF VARIABLE SIZE BLOCKS AND AUTOMATIC MOTION MODEL SELECTION

Martin Haller, Andreas Krutz, Thomas Sikora

Technische Universität Berlin
Communication Systems Group
EN 1, Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

A new approach for highly robust and precise global motion estimation (GME) using motion vectors (MVs) is presented. We show that this approach obtains precise higher-order short-term motion parameters for global motion using motion vectors solely. The approach is general and works for different mathematical methods including least-squares and Newton-Raphson method. We show that the approach is suitable for fixed block sizes from plain full-search block-matching as well as for arbitrary block sizes from video streams compressed with H.264/AVC reference encoder. The proposed approach is compared against four other known MV-based GME (MV-GME) methods. Our results show that the approach is significantly more robust and obtains higher precision for global motion parameters in terms of background motion compensation, especially if moving objects occur. In addition, the results are as good as results from precise pixel-based GME methods or even better while the presented MV-GME methods have very low computational costs.

Index Terms— Motion analysis, Motion estimation, Motion compensation, Video coding, Video analysis

1. INTRODUCTION

Higher-order motion parameters estimated by GME algorithms are necessary for many image processing applications such as video mosaicing, moving object segmentation, video coding, camera motion characterization, video analysis, etc.

Several GME algorithms were proposed recently, where pixel-based GME algorithms [1, 2, 3] use the luma signals of an image pair and MV-GME methods like [4, 5, 6, 7] use MVs obtained by block-matching. Motion vectors are included in video streams of motion-compensated video codecs. The vectors are essentially reused by MV-GME with the motivation to lower the computational complexity for GME and avoid a repetition of motion estimation with block-matching or pixel-based GME due to their significantly higher computational costs.

However, previously proposed MV-GME methods do not obtain the quality of pixel-based GME methods as we will show in the experimental section. The approach, which is presented in this paper, solves this problem. The computational complexity remains significantly low while pixel-based GME quality is obtained. This becomes possible when appropriate initialization is combined with a robust estimator.

We evaluate two pixel-based and six MV-GME algorithms by means of background Y-PNSR for global motion compensation on five sequences with and three sequences without moving foreground objects.

The paper is organized as follows. Section 2 introduces the proposed approach for GME using MVs. Section 3 presents the experimental results. The paper concludes with Section 4.

2. ROBUST MV-GME APPROACH

The block diagram of our robust MV-GME approach is shown in Fig. 1. All blocks and symbols are introduced and described in this section.

Motion vectors obtained by block-matching as used within hybrid video codecs describe the displacement of a block in a current frame with respect to the best match in the reference frame. This work assumes that the reference frame is the previous frame in temporal order (P frame). The partition of blocks in the current frame can be fixed as in MPEG-2 or variable in size as in MPEG-4 part 2 and 10 (H.264/AVC). The i-th motion vector $\mathbf{v}_i$ of a motion vector field (MVF) has $v_{x,i}$ and $v_{y,i}$ as horizontal and vertical displacements for a block at position (x_i, y_i) which are the horizontal and vertical coordinate values within the frame. A block weight n_i is assigned to the i-th motion vector due to the different size of block partitions in the current frame. This enables MV-GME to support fully modern tree-structured motion vector fields. We define the block weight n_i as

$$n_i = \begin{cases} 16 & \text{, if block size is } 16 \times 16, \\ 8 & \text{, if block size is } 16 \times 8 \text{ or } 8 \times 16, \\ 4 & \text{, if block size is } 8 \times 8, \\ 2 & \text{, if block size is } 8 \times 4 \text{ or } 4 \times 8, \\ 1 & \text{, if block size is } 4 \times 4. \end{cases}$$

The block weight n_i is currently adapted for block sizes in the range of 4×4 to 16×16. This weight can be readapted for other limits of variable block sizes if desired.

Skip macro-blocks contained in compressed video streams are ignored by default and are considered only if too few motion vectors are available. It may occur that these few motion vectors belong to a moving object, so that it is necessary to consider the skip blocks as zero-motion to stabilize the GME. Skip blocks are used if $\sum_{\forall i} n_i$ is

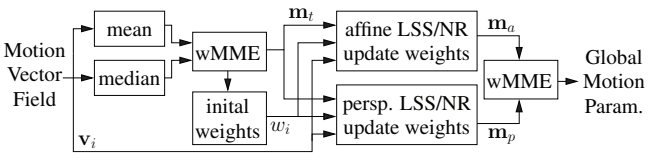
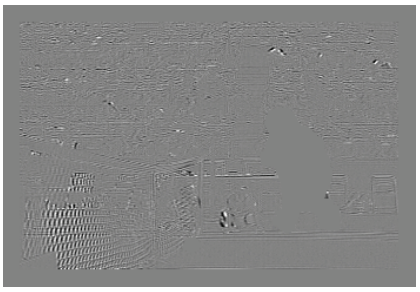


Fig. 1. Block diagram for the robust MV-GME approach

(a) Motion vectors $\mathbf{v}_i$ (color symbolizes the direction and luma represents logarithmically the magnitude, intra-MB is cyan, skip-MB is yellow)

(b) Final weights w_i for perspective MVGME-NR (white equals 1, black equals 0)



(c) Global motion compensation error combined with background mask and a border mask of 15 pixels

Fig. 2. Example for $\mathbf{v}_i$, w_i, and motion compensation error for Stefan sequence (MVF for frames 295-296) for H.264 JM 16.2, EPZS, QP30

smaller than $T_{\text{skip}}N_{\text{max}}$ where T_{skip} is chosen experimentally as 20 % and N_{max} is the maximum number of blocks for a notional fixed 4×4 partition and is computed by 16 times the number of macro-blocks for a fixed 16×16 partition.

A displacement originated by a parametric motion model at point (x_i, y_i) can be seen as a motion vector estimate $\tilde{\mathbf{v}}_i$. Such estimates can be computed for the translational motion model $\mathbf{m}_t = (m_2, m_5)$ so that

$$\begin{aligned}\tilde{v}_{x,i}(x_i, y_i, \mathbf{m}_t) &= m_2\\ \tilde{v}_{y,i}(x_i, y_i, \mathbf{m}_t) &= m_5.\end{aligned}$$

For affine motion model $\mathbf{m}_a = (m_0, m_1, m_2, m_3, m_4, m_5)$ the vector components are

$$\begin{aligned}\tilde{v}_{x,i}(x_i, y_i, \mathbf{m}_a) &= m_0x_i + m_1y_i + m_2 - x_i\\ \tilde{v}_{y,i}(x_i, y_i, \mathbf{m}_a) &= m_3x_i + m_4y_i + m_5 - y_i,\end{aligned}$$

and for perspective model
$\mathbf{m}_p = (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7)$ we obtain

$$\begin{aligned}\tilde{v}_{x,i}(x_i, y_i, \mathbf{m}_p) &= \frac{m_0x_i + m_1y_i + m_2}{m_6x_i + m_7y_i + 1} - x_i\\ \tilde{v}_{y,i}(x_i, y_i, \mathbf{m}_p) &= \frac{m_3x_i + m_4y_i + m_5}{m_6x_i + m_7y_i + 1} - y_i.\end{aligned}$$

With these estimated motion vectors $\tilde{\mathbf{v}}_i$ using a global motion model, new coordinate values can be computed by

$$\begin{aligned}x'_i &= x_i + \tilde{v}_{x,i}(x_i, y_i, \mathbf{m})\\ y'_i &= y_i + \tilde{v}_{y,i}(x_i, y_i, \mathbf{m}).\end{aligned}$$

The M-estimator principle is used during iterative estimation to remove successively outlier. For this, a weight w_i is introduced for each MV. Our approach uses a binary weighting function, so that either MVs are used for estimation or ignored completely [8].

The error between the motion vector $\mathbf{v}_i$ and estimated motion vector $\tilde{\mathbf{v}}_i$ using global motion parameters is defined as

$$\begin{aligned}e_{x,i}(\mathbf{m}) &= w_i(v_{x,i} - \tilde{v}_{x,i}(x_i, y_i, \mathbf{m}))\\ e_{y,i}(\mathbf{m}) &= w_i(v_{y,i} - \tilde{v}_{y,i}(x_i, y_i, \mathbf{m})).\end{aligned}$$

We then define the weighted mean match error (wMME)

$$wMME(\mathbf{m}) = \frac{1}{\sum\limits_{\forall i \in W} n_i} \sum_{\forall i \in W} n_i \left(|e_{x,i}(\mathbf{m})| + |e_{y,i}(\mathbf{m})|\right)$$

for a given set of motion parameters $\mathbf{m}$ where $W = \{i : w_i > 0\}$ is the index set of non-zero weighted elements of all i. The wMME considers the block weights n_i and is used as objective criterium to decide between different global motion parameter sets.

Initialization: The estimation of global motion parameters needs an appropriate initialization. To this end, weighted mean is determined as

$$\begin{aligned}\text{mean}(v_{x,i}) &= \frac{1}{\sum\limits_{\forall i} n_i} \sum_{\forall i} n_i v_{x,i}\\ \text{mean}(v_{y,i}) &= \frac{1}{\sum\limits_{\forall i} n_i} \sum_{\forall i} n_i v_{y,i}\end{aligned}$$

as well as the weighted median values for $v_{x,i}$ and $v_{y,i}$ are computed on $\sum_{\forall i} n_i$ elements wherein motion vector $\mathbf{v}_i$ occurs n_i times. These values are used to form translational motion parameters

$$\begin{aligned}\mathbf{m}_{\text{mean}} &= (\text{mean}(v_{x,i}), \text{mean}(v_{y,i})),\\ \mathbf{m}_{\text{median}} &= (\text{median}(v_{x,i}), \text{median}(v_{y,i})).\end{aligned}$$

The further used set is selected according to the lowest weighted mean match error. This set is used to compute the initial weights w_i.

Updating w_i: The updating is only accomplished if $\sum_{\forall i \in W} n_i$ exceeds $T_{\text{up}}N_{\text{max}}$. The threshold used for outlier rejection is $\mu + \sigma$, where

$$\begin{aligned}\mu &= \frac{1}{N_{\text{max}}} \sum_{\forall i \in W} n_i \left(|e_{x,i}(\mathbf{m})| + |e_{y,i}(\mathbf{m})|\right)\\ \sigma^2 &= \frac{1}{N_{\text{max}} - 1}\Big(\big(N_{\text{max}} - \sum_{\forall i \in W} n_i\big)\mu^2 +\\ &\qquad \sum_{\forall i \in W} n_i\big(\,|e_{x,i}(\mathbf{m})| + |e_{y,i}(\mathbf{m})| - \mu\big)^2\Big).\end{aligned}$$

The update of weights w_i is performed for a non-zero mean error μ. Before the updating step is finalized, the number of outliers is determined. Each motion vector with absolute error $|e_{x,i}(\mathbf{m})|+|e_{y,i}(\mathbf{m})|$ greater than the threshold $\mu + \sigma$ is considered as an outlier. Before the weights are updated, it is ensured that the sum of n_i for non-zero weighted motion vectors has to be at least $T_{\text{up}}N_{\text{max}}$. Otherwise the weights remain unchanged. T_{up} is chosen experimentally as 20 %. The binary weighting function is defined by

$$w_i = \begin{cases}0 & \text{if } |e_{x,i}(\mathbf{m})| + |e_{y,i}(\mathbf{m})| > \mu + \sigma,\\ 1 & \text{otherwise.}\end{cases}$$

The threshold $\mu + \sigma$ decreases rapidly after initial update of weights w_i and in subsequent update steps. In each update step outliers are removed and within each iteration the motion model improves as well. For example, the thresholds in each iteration for MVF in Fig. 2 are $9.81, 2.20, 0.68, 0.35$, and 0.23.

The iterative estimation begins with the initial motion parameter set $\mathbf{m}$ and the initial updated weights w_i. The iteration steps are as follows:

1. Step: Use current motion parameter set $\mathbf{m}$ and current weights w_i to determine the incremental change $\Delta\mathbf{m}$.
2. Step: Determine the new set of motion parameters $\mathbf{m}^+ = \mathbf{m} + \Delta\mathbf{m}$.
3. Step: Iterative estimation is terminated for a maximum number of iterations N_{it} or for certain conditions of $\Delta\mathbf{m}$.
4. Step: Update weights w_i using new motion parameters $\mathbf{m}^+$, increment the number of iterations, and proceed with step 1.

The estimation is done independently for affine and perspective motion models. The resulting parameters are compared with wMME and the set with the smallest error is selected as the final estimated global motion parameters. This implies an automatic decision whether affine or perspective motion model is used.

The first step of each iteration differs for different mathematical methods and is described in more detail in the following sections as well as corresponding termination conditions from step three.

2.1. MV-GME using Least-Squares Solution

Matrix $\mathbf{H}_a$ and vector $\mathbf{r}_a$ consider all $\mathbf{v}_i$ with $i \in W$ so that

$$\mathbf{H}_a = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$\mathbf{r}_a = \begin{pmatrix} \dots & x'_i & y'_i & \dots \end{pmatrix}^T.$$

The new affine motion parameters $\mathbf{m}_a^+$ are computed using least-squares solution (LSS) similar to [5] and the corresponding change in parameters is given as $\Delta\mathbf{m}_a$

$$\begin{aligned} \mathbf{m}_a^+ &= ((\mathbf{H}_a^T\mathbf{H}_a)^{-1}\mathbf{H}_a^T\mathbf{r}_a)^T \\ \Delta\mathbf{m}_a &= \mathbf{m}_a^+ - \mathbf{m}_a. \end{aligned}$$

Matrix $\mathbf{H}_p$ and vector $\mathbf{r}_p$ are defined for the perspective motion model for all $\mathbf{v}_i$ with $i \in W$. The new perspective parameters $\mathbf{m}_p^+$ and the difference $\Delta\mathbf{m}_p$ are computed analog as described above.

The iteration of MV-GME-LSS stops if the maximal number of iterations is reached or no change in parameters occurs due to unchanged weights w_i in the last iteration step so that $||\Delta\mathbf{m}_p||_1 = 0$. The maximal number of iterations N_{it} is set to 5 for comparison with MV-GME using Newton-Raphson method.

2.2. MV-GME using Newton-Raphson Method

Hessian matrix $\mathbf{A}$ and gradient vector $\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5)^T$ for affine motion model are computed by differentiating the squared error $e_{x,i}^2 + e_{y,i}^2$ according to the Newton-Raphson (NR) method as used in [6]. $\mathbf{A}$ and $\mathbf{b}$ consider only motion vectors $\mathbf{v}_i$ with $i \in W$. The updated affine motion parameters $\mathbf{m}_a^+$ are computed as

$$\begin{aligned} \Delta\mathbf{m}_a &= (\mathbf{A}^{-1}\mathbf{b})^T \\ \mathbf{m}_a^+ &= \mathbf{m}_a + \Delta\mathbf{m}_a. \end{aligned}$$

For perspective motion model, Hessian matrix $\mathbf{P}$ and gradient vector $\mathbf{c} = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7)^T$ are computed for all motion vectors $\mathbf{v}_i$ with $i \in W$. The new perspective motion parameters $\mathbf{m}_p^+$ are determined by

$$\begin{aligned} \Delta\mathbf{m}_p &= (\mathbf{P}^{-1}\mathbf{c})^T \\ \mathbf{m}_p^+ &= \mathbf{m}_p + \Delta\mathbf{m}_p. \end{aligned}$$

MV-GME-NR stops its iterations if the maximal number of iterations N_{it} is reached or the difference for translational parameters m_2 and m_5 is below threshold $T_{\text{NR},1}$ and for all other parameters below $T_{\text{NR},2}$. We follow [6] by choosing thresholds as $N_{\text{it}} = 5$, $T_{\text{NR},1} = 10^{-3}$, and $T_{\text{NR},2} = 10^{-5}$.

3. EXPERIMENTAL EVALUATION

We selected the global motion compensation (GMC) background error measured as Y-PSNR in dB for comparing our proposed approach and other methods in the experiments. The background error is determined by applying a mask that excludes foreground pixels if moving objects are present in an image sequence. The GMC background error for a given set of estimated global motion parameters is measured with original image sequences to exclude compression related artifacts from the comparison besides the influence on the resulting MVs. We used eight test videos, which are Stefan (352×240), Foreman, Horse, Biathlon, and Allstars (each 352×288) with large and small moving objects and Monaco, Room3D, and Castle (each 352×288) without any moving objects.

The comparison considers plain pixel difference as baseline, pixel-based GME using optical-flow [1] (OF-GME), and pixel-based GME based on gradient descent [3] using feature tracking for initialization (GD-GME-FT). For fixed block size (4×4 with quarter-pel precision) and full-search block-matching (FSBM), other MV-based GME methods such as robust least-squares solution using a Tukey's biweight M-estimator (Smolić) [5] and adaptive motion model selection using Newton-Raphson method (Su) [6] are evaluated together with two MVF filters (CAS-Su, FLT-Su) [7, 9] as preprocessing for Su's method as examined in [7]. Our approach uses least-squares solution (MV-GME-LSS) or Newton-Raphson method (MV-GME-NR). Results for these methods were obtained using motion vectors estimated for fixed block sizes as well as variable block sizes and different quantization step sizes (QP). For the latter case, the H.264/AVC reference encoder JM 16.2 was used with prediction structure IPPP... and enhanced predictive zonal search (EPZS) for motion estimation.

The results are shown as average background Y-PSNR values in Tab. 1. Visual examples[1] for MV-GME-NR are given in Fig. 2. Here, the motion vector field, the final weighting for perspective MV-GME based on Newton-Raphson method, and the motion compensation background error are shown.

Table 2 compares computational costs[2] for our proposed MV-based GME methods using motion vector fields (QP30) from the Stefan sequence. MV-GME-NR has less computational costs while obtaining very similar background Y-PSNR values as MV-GME-LSS. However, if motion vectors are not available from compressed video then additional computational costs for block-matching motion (BM) estimation has to be considered for a combination of BM with MV-GME methods.

[1] http://www.nue.tu-berlin.de/research/mvgme

[2] AMD Opteron™ processor 8354, 2.2 GHz, single core used

Table 1. Average background Y-PSNR values in dB for 8 video sequences, for H.264/AVC videos also bitrates and Y-PSNR in dB are given

	Stefan	Foreman	Horse	Biathlon	Allstars	Monaco	Room3D	Castle
Methods	with moving foreground objects					only background		
Pixel Difference	17.73	27.79	18.11	24.20	30.72	25.97	18.13	23.31
Pixel-based Global Motion Estimation								
OF-GME [1]	29.60	37.09	23.09	31.83	42.55	40.91	33.42	36.49
GD-GME-FT [3]	30.44	36.81	27.14	37.30	41.20	41.02	39.29	36.51
Full-Search Block-Matching, 4×4 fixed block size, quarter-pel, bicubic spline interpolation 9th order								
MV-GME-LSS	29.93	38.04	29.04	38.72	42.27	39.64	36.58	36.87
MV-GME-NR	29.92	38.03	29.04	38.72	42.27	39.64	36.58	36.87
Smolić [5]	24.41	32.44	22.60	32.92	40.41	38.18	38.15	34.86
Su [6]	23.17	32.35	21.36	33.60	40.72	40.42	32.06	35.20
CAS-Su [7]	23.42	32.40	21.77	33.60	40.68	40.42	32.05	34.84
FLT-Su [9]	24.41	32.48	22.27	30.97	41.01	40.40	32.12	34.92
H.264 JM 16.2, EPZS block-matching, quarter-pel, QP24								
Bitrate in kbit/s	2793	808	2820	822	533	844	1595	579
Y-PSNR in dB	39.45	39.60	38.32	40.84	40.54	39.54	38.61	39.84
MV-GME-LSS	30.41	37.71	32.92	38.82	42.51	39.59	35.84	36.97
MV-GME-NR	30.40	37.71	32.92	38.82	42.51	39.59	35.83	36.97
H.264 JM 16.2, EPZS block-matching, quarter-pel, QP30								
Bitrate in kbit/s	1307	273	1129	339	193	258	590	182
Y-PSNR in dB	34.13	35.23	33.34	36.69	36.47	34.19	33.00	35.10
MV-GME-LSS	30.46	36.86	32.56	38.18	41.61	40.19	37.37	37.06
MV-GME-NR	30.46	36.86	32.56	38.19	41.61	40.18	37.37	37.06
H.264 JM 16.2, EPZS block-matching, quarter-pel, QP36								
Birate in kbit/s	487	101	425	144	70	71	171	63
Y-PSNR in dB	28.78	31.36	29.25	32.94	32.44	29.34	27.91	30.90
MV-GME-LSS	30.48	34.71	27.38	35.03	36.55	36.89	37.75	35.61
MV-GME-NR	30.48	34.72	27.37	35.03	36.55	36.89	37.75	35.61
H.264 JM 16.2, EPZS block-matching, quarter-pel, QP42								
Bitrate in kbit/s	163	50	156	68	26	28	61	30
Y-PSNR in dB	23.94	27.89	25.94	29.37	28.88	25.53	23.69	27.44
MV-GME-LSS	29.39	29.97	19.89	28.88	30.72	26.30	36.72	25.65
MV-GME-NR	29.39	29.98	19.89	28.88	30.72	26.30	36.72	25.65

Table 2. Computational costs per motion vector field for Stefan sequence (30 Hz, methods implemented in C/C++)

	MV-GME	
Method	LSS	NR
GME Time	18.06 ms	6.04 ms
GME per s	55.37	165.56

4. CONCLUSIONS

The proposed approach for robust MV-GME obtains results that are as good as results from pixel-based GME or even better. The approach is highly robust against moving objects, fast camera motion, and video compression related artifacts in a wide range of bitrates. The approach works for different optimization techniques, selects automatically affine or perspective motion model and is applicable to tree-structured motion vectors for variable block sizes as used in modern video codecs such as H.264/AVC.

5. REFERENCES

[1] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Visual Commun. Image Represent.*, vol. 6, no. 4, pp. 348–365, 1995.

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

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

[4] R. Wang and T. S. Huang, "Fast camera motion analysis in MPEG domain," in *Proc. ICIP*, 1999, pp. 691–694.

[5] A. Smolić, M. Höynck, and J.-R. Ohm, "Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 applications," in *Proc. ICIP*, 2000, pp. 271–274.

[6] 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, 2005.

[7] Y.-M. Chen and V. Bajić, "Motion vector outlier rejection cascade for global motion estimation," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 197–200, 2010.

[8] A. Smolić and J.-R. Ohm, "Robust global motion estimation using a simplified M-estimator approach," in *Proc. ICIP*, 2000, pp. 868–871.

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