Monte-Carlo-Based Parametric Motion Estimation Using a Hybrid Model Approach

Michael Tok, *Student Member, IEEE,* Alexander Glantz, *Student Member, IEEE,* Andreas Krutz, *Member, IEEE,* and Thomas Sikora, *Senior Member, IEEE*

***Abstract*—Parametric motion estimation is an important task for various video processing applications, such as analysis, segmentation, and coding. The process for such an estimation has to satisfy three requirements. It has to be fast, accurate, and robust in the presence of arbitrarily moving foreground objects. We introduce a two-step simplification scheme, suitable for Monte-Carlo-based perspective motion model estimation. For complexity reduction, the Helmholtz tradeoff estimator as well as random sample consensus are enhanced with this scheme and applied on Kanade–Lucas–Tomasi features as well as on video stream macroblock motion vector fields. For the feature-based estimation, good trackable features are detected and tracked on raw video sequences. For the block-based approach, motion vector fields from encoded H.264/AVC video streams are used. Results indicate that the complexity of the whole estimation process can be reduced by a factor of up to 10 000 compared to state-of-the-art methods without losing estimation precision.**

***Index Terms*—Global motion model, Helmholtz tradeoff estimator, Monte-Carlo method, parametric motion estimation, robust regression.**

I. Introduction

MOTION is an essential property of video signals. It is caused by either arbitrarily moving objects or camera motions such as panning, zooming, rotating, and so on. The process of estimating parameters that describe background deformation through camera motion is often referred to as global motion estimation or, in a more general case, parametric motion estimation (PME). Various applications for PME include video coding and filtering, object segmentation in video sequences, or analysis issues, such as classification by motion and summarization of scene content.

The solution space for a parametric motion model (PMM) describing zoom, shearing, rotation, translation, and perspective deformation is of eighth dimension (as eight parameters have to be estimated at the same time), which makes it difficult to find a correct background deformation model for a pair of frames. To overcome this issue, iterative approximation methods that assume an initial model and refine it stepwise are applicable. Various PME methods working that way have already been proposed. Most of them work on pixel data, in the frequency domain, or on motion vector fields from encoded video data. A well-known way to estimate global motion on pixel data by iterative refinement is to start with a purely translational background motion model and to successively refine it by stepwise minimization of the registration error of a frame pair. Dufaux and Konrad [1], e.g., built a three-step image pyramid by low-pass filtering and downsampling a frame pair twice. For the pyramid level with the lowest resolution (twice downsampled), first a translational model is estimated as initial assumption. For each level, the model is then refined by a gradient descent approach. Each model for a given level is taken as initial assumption for gradient descent on the next level. That way, only a few steps per pyramid level are necessary. The whole estimation process results in a precise perspective motion model. For further refinement, Krutz *et al.* [2] proposed several improvements. They make use of an additional upsampling step for estimation on the image pyramid and replace the sampling filter by wavelet functions. Moreover, they use phase correlation in the frequency domain for the first translational motion model initialization. For further improvement, the PME is done on several image windows. Finally, only the estimation corresponding to the window with the smallest registration error is taken into account. One general disadvantage of pixel-based PME methods using gradient descent on image pyramids is the high computational complexity since for every step at every level an error gradient has to be calculated. To reduce the complexity of pixel-based PME using gradient descent, Yang *et al.* [3] introduced several simplification steps such as error gradient reusage and frame binarization.

Another way of finding PMMs is to derive them from a set of simpler translational models that are much easier to be determined. When global motion estimation on compressed video data is needed, for instance, macroblock-based estimation approaches, working on translational motion vectors of encoded video streams are suitable. Tarannum *et al.* [4], e.g., proposed a clustering method for identifying macroblocks belonging to background regions. Following this study, a robust M-estimator was applied [5]. Another way for block-based PME was proposed by Su *et al.* [6]. They applied gradient descent on motion vectors with iterative outlier rejection and used an adaptive motion model selection. A survey of such so-called compressed-domain features and their utilization in video analysis is given by Wang *et al.* in [7].

Manuscript received January 2, 2012; revised May 4, 2012; accepted June 21, 2012. Date of publication August 2, 2012; date of current version April 1, 2013. This paper was recommended by Associate Editor B. Zeng.

The authors are with the Communication Systems Group, Technische Universität Berlin, Berlin 10587, Germany (e-mail: tok@nue.tu-berlin.de).

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/TCSVT.2012.2211173

All PME methods mentioned so far have one point in common. They start with initial model assumptions and perform iterative refinement of the model to be estimated. Another well-known way of estimating PMMs is based on the Monte-Carlo method. This means that a numerical problem such as PME is solved in a stochastical way. One example is the random sample consensus (RANSAC) [8], which is subsequently explained in more detail. Another robust estimator based on the Helmholtz principle [9] is introduced by Felip *et al.* [10]. It also follows the Monte-Carlo approach by evaluating randomly formed subsets of measured feature points to estimate a final model. In [11], this estimator is implemented in a very simplified way to estimate PMMs on motion vectors of encoded video streams. The simplification consists of two steps. First, six straight lines are placed on the motion vector field of the stream and thus associated with a small subset of all motion vectors of a frame. Then the transformation of each line is estimated by the Helmholtz tradeoff estimator (HTE). Subsequently, a perspective model is derived for the six line transformations. The selection of only a few motion vectors from a whole frame leads to a lack of robustness in cases where larger foreground objects appear in the scene. These drawbacks are pointed out in [12].

Due to the power of higher-order motion models, the application scenarios for robust and precise PME are numerous. The MPEG-4 visual standard [13], e.g., uses parametric motion compensation for inter-frame redundancy reduction to obtain high compression rates. An additional encoding mode of MPEG-4 visual uses generated background sprites to model the background information of a scene. Kunter *et al.* [14] presented a technique to extend H.264/AVC [15] with an additional sprite mode to combine the advantages of sprite coding and H.264/AVC. This leads to higher compression efficiency over a large bit rate range, which even increases with the usage of multiple sprites. A way of using PMMs for filtering purposes is introduced with global motion temporal filtering. Glantz *et al.* [16] presented this technique as in-loop filtering for improving the coding performance of H.264/AVC. Wiegand *et al.* proposed to generate additional reference frames by affine image warping with PME to get higher H.264/AVC coding performance [17].

Another field for PME is automatic foreground object segmentation in video sequences for various applications. In [18], Farin *et al.* proposed to generate multiple background sprites by long-term PME for segmentation purposes. Krutz *et al.* introduced local background sprite models to obtain highly precise segmentation masks based on PME [19]. A comparison of integrated global motion-based object segmentation algorithms for automatic MPEG-4 sprite coding is given by Glantz *et al.* [20].

In [21], Irani *et al.* described how to use PME for video indexing. They generated background mosaics (or sprites) to deliver a quick overview of a given sequence. Another way of getting such a compact representation of video sequences by using PME was introduced by Sawhney and Ayer in [22]. Ye *et al.* [23] described how to create highly precise superresolution sprites for overview purposes by using PME.

All these techniques can benefit from higher PMM estimation quality. The main challenge is to derive algorithms that estimate quickly while remaining highly precise. In this paper, we present a two-step hybrid PME scheme for simplifying Monte-Carlo-based PME on encoded video data streams as well as on feature vector fields generated by feature tracking on raw pixel data. This scheme uses motion models with differing parameter amounts and so reduces the complexity of the whole estimation process dramatically. To demonstrate the possible complexity reduction with this scheme, it is applied on RANSAC and a second robust Monte-Carlo regression method based on the Helmholtz principle. For feature-based PME, Kanade–Lucas–Tomasi (KLT) feature tracking is applied on raw video data to generate a reliable feature motion vector field. On this field, the PME with the proposed scheme is then applied.

The remainder of this paper is organized as follows. Section II describes how robust regression on motion vector fields can be used for PME. Subsequently, two different Monte-Carlo-like methods for robust regression are described. The first one is the RANSAC algorithm. For further comparison, an estimation method based on the Helmholtz principle is also described shortly as a related regression method. To achieve computational complexity reduction for the estimation process, several simplification steps are explained in Section III. Details of the proposed simplification are described in Section III-B. Section IV presents and discusses results for macroblock-based PME on H.264/AVC video data as well as for feature-based PME on KLT feature vector fields. Therefore, the generation of these motion vector fields used for PME is described in short and their properties are discussed. Estimation quality results in terms of registration error values are shown for evaluation. Additionally, complexity analysis containing runtime comparisons of the discussed methods complete the evaluation and point out the complexity reduction potential of the hybrid model scheme. To summarize, the performance advantages in terms of runtime as well as estimation quality of a hybrid motion model approach in combination with Monte-Carlo principles are pointed out. Finally, conclusions are drawn in Section V.

II. Robust Regression for PME

The basic principle of every PME technique is to find a PMM representing the background transformation between two given frames. As background regions are assumed to be planar geometric objects, 3×3 homographies can represent such a transformation with high precision. These homographies can be calculated in different ways. One possibility is to reduce the background registration error directly. Another possibility is to obtain local motion models describing the translational motion of background area parts. By combining these local motion models in an appropriate manner, a higher-order motion description, containing zoom, shear, rotation, translation, and perspective deformation, can be derived.

For example, a description of the blockwise motion between two frames can be obtained by motion vectors of macroblocks from H.264/AVC-encoded video streams. Another possibility

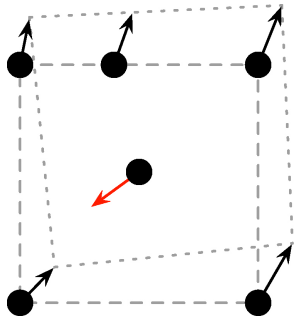


Fig. 1. Example of outlier rejection on motion vectors for PME. The dashed quadrangles represent the perspective deformation derived from the correct motion vectors. The vector in the middle does not belong to the model so it is declared as an outlier.

is to generate motion vector fields by feature detection and tracking. Both kinds of motion vector fields describe the displacement of background regions as well as the arbitrary motion of foreground objects. When the macroblock or feature motion vectors belonging to foreground objects are defined as outliers, the process of finding a higher-order background motion model can be seen as a robust regression problem. Fig. 1 illustrates this process. As the determination of such parametric models from local motion models can be done linearly, utilizing linear regression methods for outlier removal is a reasonable option. Although the motion model itself may not be linear, the underlying equation system often is.

A. Linear Regression in General

The task of linear regression in general is to find a model for a set of N observations. As described in [5], this model can be represented by a set of linear equations connecting the p parameters of the model $\boldsymbol{\theta}$ with the observations ($\mathbf{y} \leftrightarrow \mathbf{X}$)

$$y_i = x_{i,1} \cdot \theta_1 + \cdots + x_{i,p} \cdot \theta_p. \tag{1}$$

A model parameter set $\hat{\boldsymbol{\theta}}$ can be estimated by

$$\hat{y}_i = x_{i,1} \cdot \hat{\theta}_1 + \cdots + x_{i,p} \cdot \hat{\theta}_p. \tag{2}$$

The regression task is then modified to minimize the sum of estimation errors $r_i = y_i - \hat{y}_i$, each rated by an error function $\rho(r_i)$. The most common ρ is the square function, leading to least squares (LSs) solution

$$\min_{\hat{\theta}} \sum_{i=1}^{N} r_i^2. \tag{3}$$

This simple error-weighting function is often not useful for estimating a parametric background motion model out of local motion. A single outlier in a set of local motion models would lead to severe misestimation. Nevertheless, when applied to a set of noisy inliers in terms of very small local motion estimation errors, LS is able to deliver an unbiased result. A combination of robust outlier rejection and LS is a suitable way of implementing robust regression.

B. Regression Via RANSAC

The commonly used robust Monte-Carlo regression method is the so-called RANSAC, introduced by Fischler *et al.* [8].

Algorithm 1 RANSAC

```
t ← error threshold for all observations
X ← all observations
X_bestset ← ∅
m ← log(1 − P) / log(1 − (1 − ϵ)^p)
N ← 0
while N < m do
    X_consensus ← ∅
    select randomly p observations {x_1, ..., x_p} out of X
    derive model ŷ from {x_1, ..., x_p}
    for all observations x in X do
        if x fits ŷ with an error ≤ t then
            add x to X_consensus
        end if
    end for
    if #(X_consensus) > #(X_bestset) then
        X_bestset ← X_consensus
    end if
    N ← N + 1
end while
derive model y from X_bestset
return y
```

It is based on a simple statistical assumption. For a set of observations containing a percentual amount ϵ of outliers, the probability P of taking at least once a subset of p elements containing only inliers when trying m times is

$$P = 1 - (1 - (1 - \epsilon)^p)^m. \tag{4}$$

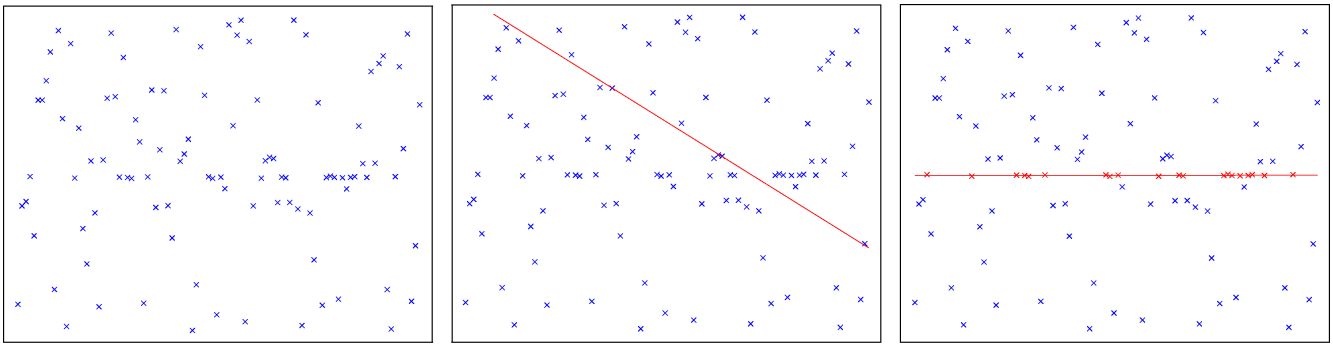
Conversely, a minimal amount of samples m to be evaluated for model estimation is derivable for a desired model correctness probability P and a model parameter amount p

$$m = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^p)}. \tag{5}$$

Then, for each subset, an assumed model can be calculated to classify the elements of the whole set either as inliers or as outliers concerning this model. To decide if an element fits to an assumed model, a fixed threshold, defining an elements' error distance to the assumed model has to be defined. The consensus set for the assumed model then only consists of fitting elements. After at least m random trials, the largest consensus set for an assumed model can be used to get a final model describing the inliers of the whole set best by ignoring outliers at the same time. An overview of the RANSAC algorithm is given in Algorithm 1.

C. Regression Via Helmholtz Principle

Another highly robust regression method is the HTE. Its technique is inspired by a fundamental law called the Gestalt theory, in which a group is perceptually meaningful if its number of occurrences would be very small in a random situation [9]. A Monte-Carlo approach for robust regression based on this principle is presented in [10].



(a) (b) (c)

Fig. 2. Illustration of the robustness of the HTE: 20 noisy points of a line are placed between 80 outliers. (a) Original set. (b) Estimation result using LS. (c) Estimation result using HTE.

In general, a subset of inliers regarding an expected model can be rated by its size and its model error standard deviation

$$\gamma = \frac{f\left(n_{\text{inliers}}\right)^{\alpha}}{g\left(\sigma_{\text{inliers}}\right)^{\beta}}. \tag{6}$$

The rating functions f and g and weighting parameters α and β are helpful for prioritizing the number of inliers of the population over their error standard deviation or vice versa. Theoretically, when evaluating every possible subset, the model derived from the subset with the highest rating γ should describe the most meaningful elements of a whole set best.

The PMM to be found is described by a 3×3 perspective transformation matrix $\mathbf{H}$ that transforms a given position $\mathbf{p}_i = (x_i, y_i)^T$ to a new position $\tilde{\mathbf{p}}_i = (\tilde{x}_i, \tilde{y}_i)^T$ by

$$\begin{bmatrix} \tilde{h} \cdot \tilde{x}_i \\ \tilde{h}_i \cdot \tilde{y}_i \\ \tilde{h}_i \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{7}$$

Following [10], a robust Helmholtz-based estimation procedure for PME that estimates such a perspective motion model on motion vector fields can be implemented as follows.

1) Take m samples of $p/2$ motion vectors $\mathbf{d}_i = (\Delta x_i, \Delta y_i)^T$ and their positions $\mathbf{p}_i$. Since one motion vector has two dimensions and thus two parameters, $p/2$ vectors already cover p parameters. Hence, for a perspective motion model estimation ($p = 8$), four vectors and their positions are necessary.
2) For each sample, do the following.
   a) Assume a model $\mathbf{H}_s$ by solving the determined equation system defined by the $p/2$ samples taken.
   b) Transform every motion vector position $\mathbf{p}_i$ with $\mathbf{H}_s$ to a new position $\tilde{\mathbf{p}}_{i,s}$ and calculate the error distance between the new motion vector position and the endpoint of that motion vector

$$r_{i,s} = \left\| \mathbf{p}_i + \mathbf{d}_i - \tilde{\mathbf{p}}_{i,s} \right\|. \tag{8}$$

   c) Build the sorted error space and find the error percentile value $v_{\lambda,s}$ defined by the inlier percentage $\lambda = (\epsilon - 1)$.
   d) Categorize each observation as an outlier or an inlier

$$w_i = \begin{cases} 1, & \text{if } |r_i/\hat{\sigma}| \le 2.5 \\ 0, & \text{else} \end{cases} \tag{9}$$

   where $\hat{\sigma}_s$ is an expected error standard deviation in an environment with a percentual inlier amount of λ

$$\hat{\sigma}_s = \frac{1}{\Phi^{-1}(0.75)} \cdot \left(1 + \frac{5}{n-p}\right) \cdot \sqrt{v_{\lambda,s}}. \tag{10}$$

   The threshold of 2.5 in (9) is quite reasonable because in a Gaussian situation as assumed here, there will be very few residuals larger than $2.5\hat{\sigma}_s$. The factor $1/\Phi^{-1}(0.75) = 1.4826$ in (10), with $\Phi^{-1}(x)$ being the inverse error function, is part of a consistent estimator of σ_s for normally distributed residuals. For very small subsets, the correction factor $1 + \left[5/(n-p)\right]$ is introduced. Further details about the estimation of $\hat{\sigma}$ can be found in [5].
   e) Count the amount of elements declared as inliers

$$I_s = \sum_{i=1}^{n} w_i \tag{11}$$

   and estimate a new model $\mathbf{H}'_s$ using only these inliers and a common parameter estimator such as LS.
   f) Find the standard deviation σ'_s for these inliers and the reestimated model $\mathbf{H}'_s$

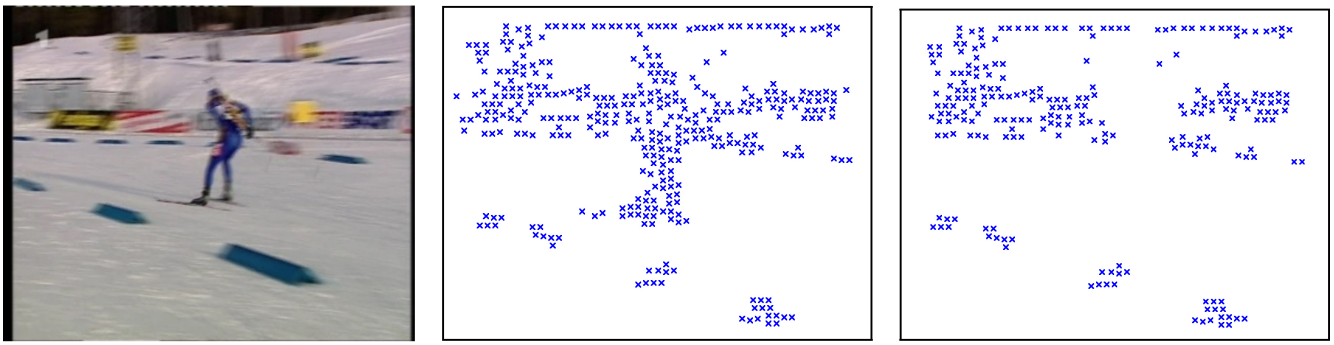
$$\sigma'_s = \sqrt{\sum_{k \in \text{inliers}} \frac{\left(r'_{k,s} - \mu'_s\right)^2}{I_s}} \tag{12}$$

   with an estimated subset error mean

$$\mu'_s = \frac{1}{I_s} \sum_{k \in \text{inliers}} r'_{k,s}. \tag{13}$$

   g) Calculate the simplified energy function value for the actual subfit following [10]

$$\gamma_s = \frac{I_s}{\sigma'_s}. \tag{14}$$



(a) (b) (c)

Fig. 3. Outlier rejection result for simplified HTE (with two-step scheme) for frame 30 of the *Biathlon* sequence. (a) Frame 30. (b) Selected features. (c) Filtered inliers.

TABLE I
SUBSET AMOUNTS FOR VARIOUS OUTLIER AND CONFIDENCE PERCENTAGES FOR AN EIGHT-PARAMETER MODEL

P	0.70	0.75	0.80	0.85	0.90
ϵ 0.70	18 350	21 129	24 530	28 915	35 094
0.75	78 903	90 852	105 476	124 329	150 902
0.80	470 302	541 521	628 686	741 062	899 447
0.85	4 697 714	5 409 104	6 279 776	7 402 266	8 984 328

3) Return the parameters of the model $\mathbf{H}'_s$ associated with the highest value of γ_s.

This estimation process is very robust to outliers. Fig. 2(a) shows a noisy test set of 100 positions. Only 20 of them belong to a line. The other 80 are randomly placed outliers that lead to an erroneous line model estimation with LS. The HTE [see Fig. 2(c)], however, has the ability to find the correct line model as the estimation result illustrates.

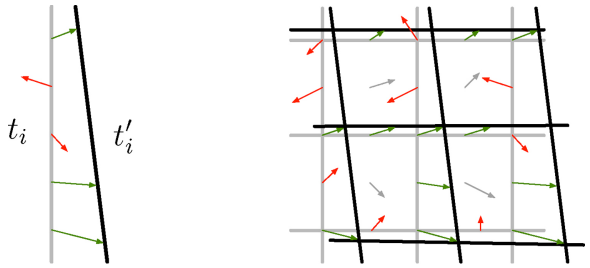
Fig. 3 shows the result of a foreground feature removal process by outlier rejection of the HTE. All features belonging to the arbitrarily moving *biathlete* are removed reliably.

III. SIMPLIFICATIONS FOR SAMPLE AMOUNT REDUCTION

A disadvantage of both RANSAC and HTE is exponential runtime dependency on the complexity (parameter amount) of the models to be estimated. For an estimation of a perspective motion model ($p = 8$), e.g., in an environment with $\epsilon = 80\%$ outliers and with $P = 95\%$ confidence, $m \approx 1\,170\,000$ subsets have to be evaluated. Table I shows examples of needed subsets for various ϵ and P. Thus, a direct usage of the original RANSAC or HTE is very complex. To reduce the amount of subset evaluations for RANSAC or HTE calculable by (5), several techniques exist. A selection of existing simplifications is explained in the following. Subsequently, the proposed simplification scheme is presented.

A. Conventional Simplification Methods

When assuming an outlier percentage of ϵ in n observations, an expected amount of $k = n(1-\epsilon)$ inliers exists. Hence, when a possible consensus set with size k or larger is found, no further sets are evaluated (preemptive). A more robust and adaptive way to simplify RANSAC is based on recalculating m each time a set is evaluated (adaptive) [24].



(a) (b)

Fig. 4. Illustration of the two estimation steps proposed by Barcelo *et al.* for deriving perspective motion models. (a) Single line transformation. (b) PME from six line correspondences (gray vectors are not used).

As the quality evaluation per subset of the HTE is very expensive in terms of computational steps, a reduction of needed iterations m is necessary for using the Helmholtz principle for PME with justifiable estimation time. Inspired by the work of Barcelo *et al.* [25], Felip *et al.* [11] presented a way to utilize the HTE for PME in the compressed domain without the need of evaluating millions of randomly selected subsets. Barcelo *et al.* proposed to estimate the transformation of six control lines between two frames. These lines are placed vertically and horizontally on a small subset of macroblock motion vectors. In general, the endpoints of these vectors describe a second set of lines, representing the perspective transform of the afore placed control lines as shown in Fig. 4(a).

A set of control lines and their derived correspondences is illustrated in Fig. 4(b).

By representing each control line in an implicit way ($t_i x + u_i y = 0$), a perspective model can be derived through four line correspondences $(t_i, u_i) \leftrightarrow (t'_i, u'_i)$ by

$$\begin{bmatrix} t'_i & 0 & -t_i t'_i & u'_i & 0 & -t_i u'_i & 1 & 0 \\ 0 & t'_i & -u_i t'_i & 0 & u'_i & -u_i u'_i & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} m_0 \\ \vdots \\ m_7 \end{bmatrix} = \begin{bmatrix} t_i \\ u_i \\ \vdots \end{bmatrix}. \quad (15)$$

As only four line correspondences are necessary to (15), additional robustness is given by estimating six of them and use a common regression system to obtain a final perspective transformation eventually as described by Felip *et al.* in [11].

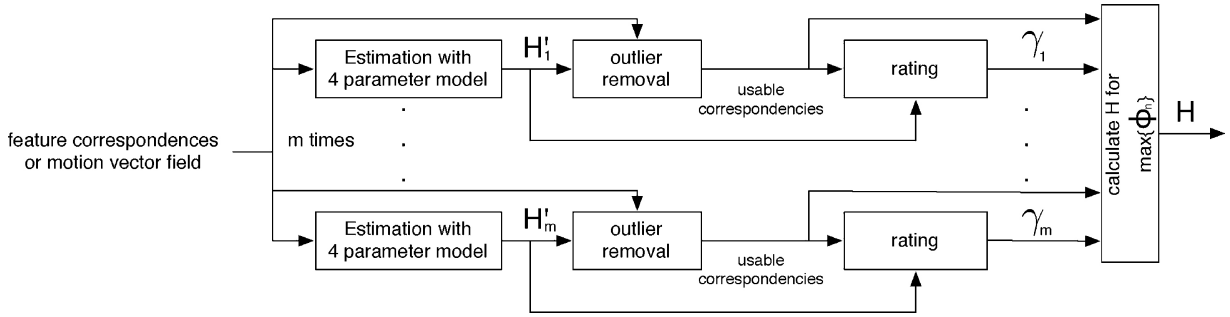


Fig. 5. Motion estimation on feature correspondences or motion vector fields based on the simplified HTE.

This approach has two drawbacks. First of all, by just taking vectors on the described control lines, only a small subset of motion vectors is involved in the PME process. Second, for solving (15), at least two horizontal and two vertical line correspondences are necessary. This subsequently means that always two out of three horizontal and two out of three vertical line correspondences have to be estimated correctly to obtain a correct model. This reduces the outlier acceptance rate ε in (5) to about 33% or less.

B. Hybrid Model Simplification

The estimation time could also be reduced by deriving simpler PMMs with fewer parameters (e.g., affine or similarity models). Since this would lead to higher registration errors resulting in lower compensation quality, a combination of outlier rejection with low parameter count models and following perspective model derivation is introduced. The most salient background changes between two frames can be modeled by a combination of simple camera-induced transformations.

1) *Uniform zoom*

$$\mathbf{H}_{\text{zoom}} = \begin{bmatrix} z & 0 & 0 \\ 0 & z & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

2) *Rotation*

$$\mathbf{H}_{\text{rot}} = \begin{bmatrix} \cos(r) & \sin(r) & 0 \\ -\sin(r) & \cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

3) *Vertical and horizontal translation*

$$\mathbf{H}_{\text{trans}} = \begin{bmatrix} 0 & 0 & t_x \\ 0 & 0 & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

Assuming additional camera distortions such as shearing or perspective deformation to be relatively small, models representing at least these four partial transformations are suitable to describe camera motion exactly enough for motion vector outlier detection and rejection. The combination of these models leads to the similarity transformation

$$\mathbf{H}_s = \begin{bmatrix} m_{0,s} & m_{1,s} & m_{2,s} \\ -m_{1,s} & m_{0,s} & m_{3,s} \\ 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

TABLE II
SUBSET EVALUATION SAVING RATIO WHEN USING A FOUR-PARAMETER MODEL INSTEAD OF AN EIGHT-PARAMETER MODEL FOR OUTLIER REJECTION WITH DIFFERENT EXPECTED OUTLIER PERCENTAGES ϵ

ϵ (%)	20	40	60	80
Saving ratio	2.87	8.19	39.56	625.50

It has only four parameters and is calculated by selecting only two randomly selected motion vectors and solving a simple 4×4 linear equation system

$$\begin{bmatrix} x_{1,s} & y_{1,s} & 1 & 0 \\ y_{1,s} & -x_{1,s} & 0 & 1 \\ x_{2,s} & y_{2,s} & 1 & 0 \\ y_{2,s} & -x_{2,s} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} m_{0,s} \\ m_{1,s} \\ m_{2,s} \\ m_{3,s} \end{bmatrix} = \begin{bmatrix} x'_{1,s} \\ y'_{1,s} \\ x'_{2,s} \\ y'_{2,s} \end{bmatrix} \tag{17}$$

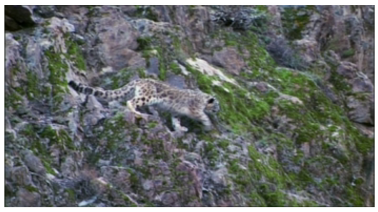
for inlier and outlier classification of each subset s. By replacing the perspective model by a similarity transformation, p in (5) is reduced from $p_{\text{perspective}} = 8$ to $p_{\text{similarity}} = 4$. Let n_1 be the amount of needed subset evaluations when taking a perspective model and n_2 the needed evaluations in the similarity model case. Then, by using n_1 and n_2 as m in (4) for the same probability P, a subset saving ratio can be defined as

$$\frac{n_1}{n_2} = \frac{\log\left(1-(1-\epsilon)^4\right)}{\log\left(1-(1-\epsilon)^8\right)}. \tag{18}$$

Table II presents ratios for selected outlier probabilities ϵ.

It has to be mentioned that the same model can directly be used to approximate an estimation error for every inlier and a variance for the particular subset in the HTE. A reestimation of a more accurate model is not necessary, which reduces the complexity of step 2e) of HTE drastically. This simplification is possible since most of the motion vectors either fit a model almost optimally or not at all. Unfortunately, zoom, rotation, and translation are much too dominant in common video sequences to use a model with even fewer parameters for further complexity reduction.

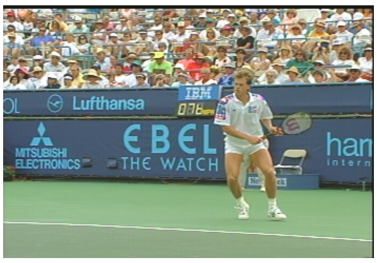
Fig. 5 illustrates, how the proposed simplification scheme is incorporated into the HTE exemplarily. The following section evaluates the time savings achievable by the hybrid model approach and compares it to the other mentioned simplification schemes.



Mountain

352 × 192, 25 Hz, 130 frames

This sequence shows one moving foreground object on highly textured background. The camera performs pan and zoom.



Stefan

352 × 240, 30 Hz, 300 frames

This sequence shows one moving foreground object on moderately textured background. The camera performs fast pan and zoom, typical for sport sequences.

Allstars CIF

352 × 288, 25 Hz, 250 frames

This sequence shows several small foreground objects with larger distance to the camera which move on lowly textured background. The camera performs slow pan.

Biathlon

352 × 288, 25 Hz, 200 frames

This sequence shows one foreground object on lowly textured background. The camera performs fast pan and slow zoom.

Monaco

352 × 288, 25 Hz, 150 frames

This sequence shows highly textured background and contains no foreground objects. The camera performs very slow pan.

Race

544 × 336, 25 Hz, 100 frames

This sequence shows three foreground objects on moderately textured background. The camera performs fast pan.

Flowervase

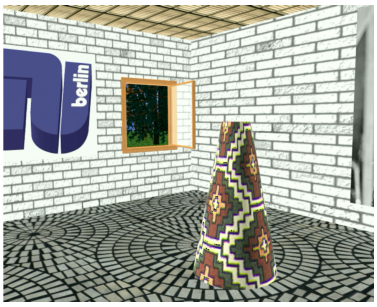
832 × 480, 30 Hz, 300 frames

This sequence shows highly textured background and contains no foreground objects. The camera slowly moves forward.

Allstars

704 × 576, 25 Hz, 250 frames

This sequence has the same properties as *Allstars CIF* except resolution.



Room 3D

720 × 576, 25 Hz, 60 frames

This sequence shows a highly textured 3D environment containing no foreground objects. The camera performs pan.

Palace

720 × 576, 25 Hz, 120 frames

This sequence shows moderately textured background and contains no foreground objects. The camera performs pan.

Penguins

1280 × 720, 25 Hz, 349 frames

This sequence shows highly textured background and contains no moving foreground objects. The camera performs slow pan.

Blue Sky

1920 × 1080, 25 Hz, 217 frames

This sequence shows a blue sky, few treetops and contains no foreground objects. The camera performs rotational motion.

Fig. 6. Overview of the test sequences used.

TABLE III
LIST OF COMPARED COMBINATIONS OF PME METHODS AND THE DISCUSSED SIMPLIFICATION SCHEMES

RANSAC original	RANSAC as described in Section II-B without any simplifications.
RANSAC 4p/8p	RANSAC as described in Section II-B with using a similarity model for steps 9 and 11 of Algorithm 1.
HTE 4p/8p	HTE as described in Section II-C with using a four similarity model for subset evaluation (as done for RANSAC 4p/8p).
RANSAC preemptive	RANSAC as described in Section II-B but with early break condition as described in Section III-A.
RANSAC adaptive	RANSAC as described in Section II-B with an additional refresh of m after step 18 of Algorithm 1: if the new $X_{\text{consensus}}$ is larger than the former X_{bestset}, the percentage of outliers ϵ and thus the amount of needed evaluations m is reduced. Hence, the amount of iterations is lowered adaptively.
RANSAC adaptive + 4p/8p	RANSAC as described in Section II-B with using a similarity model for steps 9 and 11 of Algorithm 1 and with adaptive reduction of m.

IV. EXPERIMENTAL EVALUATION

Twelve test sequences with varying properties like resolution and frame rate were selected for evaluation of the proposed schemes. Fig. 6 describes the sequences used.[1]

The PMMs were obtained by combining RANSAC and HTE, respectively, with the discussed simplification schemes and applying them either on H.264/AVC motion vector fields or on KLT feature vector fields. Table III describes in short which simplification schemes are used with RANSAC and HTE. As discussed in [11], a straightforward Monte-Carlo implementation of the HTE without simplifications is too complex in terms of memory usage and runtime to be used in real applications. The whole evaluation process is done with an expected outlier percentage of $\epsilon = 80\%$ and a probability of $P = 99.5\%$ for obtaining at least one subset without any outliers. The aforementioned error distance threshold for RANSAC is set to $t = 1.0$.

To measure the motion estimation quality, we warped each frame of a given test sequence onto its successive one with the use of the estimated parameters. The frame warping is done with bicubic spline interpolation of degree three. Background PSNR (BPSNR) values between the warped frames and their correspondences have been calculated using manually segmented ground truth masks of the background regions. BPSNR (derived from the background pixel-MSE) is not a direct measurement of how well a model fits scene motion. But it can be used as an indirect measurement for interpreting parametric background motion model quality. For complexity comparison, computation times for all tested combinations of estimation methods and simplification schemes and for motion vector field generation with the H.264/AVC reference coder [26] and the KLT feature tracking was used. Additionally, for each method, the per frame iteration amount has been counted. In the following, the generation of the motion vector fields for PME is described in short.

A. Motion Vector Fields

The presented PME method can be applied on different kinds of motion vector fields used for motion model estimation. Thus, it can be used to estimate the global motion of an encoded video stream directly in the compressed domain by utilizing the motion vectors, e.g., of given H.264/AVC macroblock structures. This obviously reduces complexity for specific applications since these motion vectors are readily available at the decoder. Additionally, it is possible to use motion vector fields with properties differing from H.264/AVC vector fields.

[1]For further results, see http://www.nue.tu-berlin.de/research/pmehma.

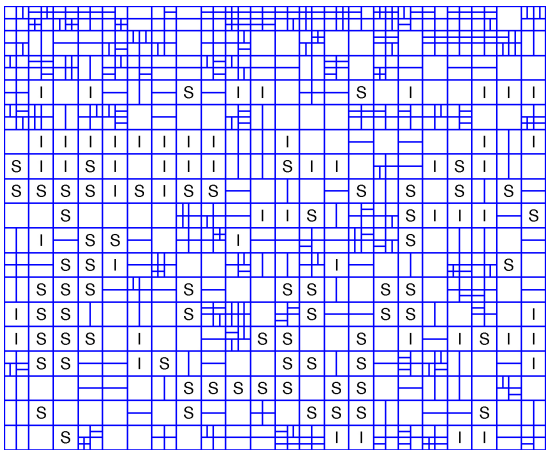


Fig. 7. Typical macroblock structure of a frame from the *Allstars CIF* sequence encoded with H.264/AVC. "I" blocks are of type intra, "S" blocks are of type skip, and empty blocks are of type inter and contain the motion vectors used for PME (motion vectors are omitted for clarity).

1) *H.264/AVC Macroblock Structures:* Hybrid video coding standards such as H.264/AVC rely on the reduction of spatiotemporal redundancy to achieve high compression ratios. Therefore, blockwise motion compensation is performed delivering motion vector fields for each inter-frame. Since a full search for block correspondences is very complex, enhanced predictive zonal search (EPZS) [27], a fast and precise local motion estimation method, is often used. Further details on existing video coding standards using motion compensation can be found in [15], [28], and [29]. Fig. 7 shows an example of a typical H.264/AVC macroblock structure. Skip blocks as a special case of inter prediction are not used for PME as described in this paper. In addition to Skip and inter blocks that contain the motion vector data used for PME, these structures also contain intra blocks for regions where a spatial redundancy exploit is more efficient than a temporal one.

To perform a better local motion estimation for inter prediction, an interpolation filter is used to obtain quarter-pel reference image resolution. This means that the resulting motion vectors used for PME also have quarter-pel precision. Another technique for improving the vector field quality is the rate-distortion optimized submacroblock partitioning utilized in inter blocks. This partitioning grants high-quality motion vectors also at borders of objects that move in different directions.

TABLE IV
MEAN BPSNR VALUES FOR THE DISCUSSED ALGORITHMS APPLIED ON H.264/AVC MOTION VECTOR FIELDS

Sequence	RANSAC Original (dB)	RANSAC 4p/8p (dB)	HTE 4p/8p (dB)	RANSAC Preemptive (dB)	RANSAC Adaptive (dB)	RANSAC Adaptive + 4p/8p (dB)
Mountain	38.39	38.40	37.46	38.40	38.44	38.43
Stefan	30.07	30.43	30.36	28.70	30.29	30.56
Allstars CIF	42.04	42.28	41.75	40.80	42.31	42.47
Biathlon	39.21	39.12	39.15	36.48	39.16	39.10
Monaco	37.48	37.70	39.24	38.25	38.39	38.83
Race	37.30	37.26	36.95	35.82	37.26	37.22
Flowervase	36.54	36.56	36.57	34.94	36.34	36.51
Allstars	39.89	39.95	39.60	38.09	39.78	39.87
Room 3-D	35.79	35.67	35.10	34.65	35.78	35.53
Palace	37.65	37.66	37.35	37.38	37.62	37.65
Penguins	32.03	32.36	32.21	31.06	32.15	32.34
Blue Sky	39.47	39.38	39.40	39.27	39.42	39.29
Mean	37.16	37.23	37.10	36.15	37.25	37.32

2) *KLT Feature Correspondence Fields:* A quality limiting factor of block motion data is the mentioned motion information resolution that is often limited to half- or quarter-pel. Hence, when choosing only well trackable features in a video frame and tracking them with an accuracy much higher than quarter-pel, better PME results are possible. Therefore, for feature-based PME with RANSAC and the HTE, a KLT feature tracker as described in [30] is used to generate an irregular motion vector field with an accuracy much higher than quarter-pel.

B. H.264/AVC Macroblock-Based Estimation

Each test sequence has been encoded with KTA 2.4 [26]. The encoder was set to use an IPPP ... group of picture structure and a quantization parameter (QP) of 4 for all inter-frames and the initial intra frame, which almost corresponds to lossless coding and thus minimizes entropy-coding-induced mismatch errors. This QP is chosen since lower QPs result in more intra and so fewer inter blocks in the coding structure while larger quantization steps lead to motion vectors with worse quality. Subsequently, the resulting motion vector fields of the encoded streams were used for PME with the modified RANSAC and HTE approaches. Table IV presents results of the selected combinations of PME methods and simplification schemes.

Almost all simplification schemes only have a slight influence on the motion estimation quality. For the *Mountain* sequence, the HTE misestimates several motion parameters resulting in a BPSNR reduction to only 37.46 dB in comparison to about 38.40 dB achieved by other estimation techniques. The RANSAC with preemptive abort criterion lacks robustness as can be seen by low BPSNR values for this method applied on the whole set of test sequences. The second fastest method—RANSAC with the proposed hybrid model usage and adaptive iteration amount calculation—delivers precise and robust results for all sequences.

C. Feature-Based Estimation

For each frame, 400 feature points were selected and tracked (which has been empirically found to be a sufficient amount for estimation). During the tracking process, some features were lost. The PME was performed on the remaining feature vectors. For feature rating and rejection, the standard settings of [31] were used. Table VI presents the results of PME with RANSAC and HTE in combination with the discussed simplification schemes.

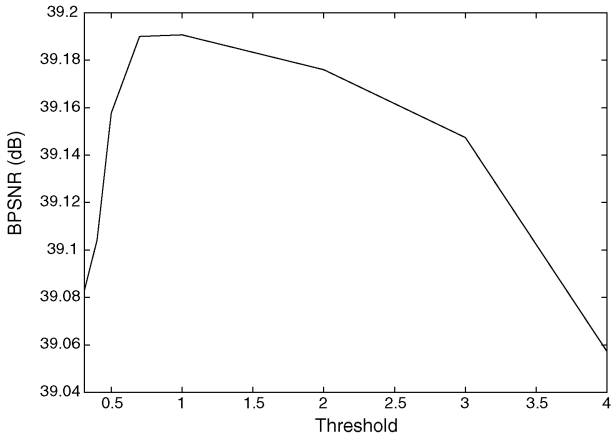


Fig. 8. BPSNR values for feature-based estimation with adaptive RANSAC on the *Biathlon* sequence with different error distance thresholds.

Same as in the macroblock-based case, RANSAC with preemptive abort criterion lacks robustness when applied on feature vector fields. All other estimation methods deliver high-quality estimation results again. The estimation accuracy of the simplified HTE, the adaptive RANSAC, and adaptive RANSAC with hybrid model simplification is almost identical.

It has to be emphasized that the quality of RANSAC-based PME depends on the fixed error distance threshold value for subset evaluation, as described in Section II-B. In contrast, HTE has no thresholds to be set. An inauspiciously selected threshold leads to erroneous or too small consensus sets. Fig. 8 demonstrates this fact by comparing BPSNR values for feature-based estimations on the *Biathlon* sequence with differing thresholds for adaptive RANSAC. So even if the HTE with hybrid model usage is neither the fastest nor the most robust method, it is highly adaptive due to the complex rating criterion, considering not only the amount but also the error variance of each subset to be evaluated.

TABLE V
AVERAGE PER FRAME RUNTIMES OF THE DISCUSSED ALGORITHMS APPLIED ON H.264/AVC MOTION VECTOR FIELDS

Sequence	RANSAC Original (ms)	RANSAC 4p/8p (ms)	HTE 4p/8p (ms)	RANSAC Preemptive (ms)	RANSAC Adaptive (ms)	RANSAC Adaptive + 4p/8p (ms)
Mountain	27253.75	177.55	730.82	3.92	4.01	3.58
Stefan	1727.53	230.22	1036.04	4.25	12.06	5.19
Allstars CIF	13364.50	299.47	1351.53	5.62	11.56	5.88
Biathlon	15308.40	209.39	965.53	3.13	17.57	5.09
Monaco	16485.85	287.80	1269.05	5.78	6.92	5.75
Race	3334.94	421.57	1974.88	7.72	18.04	9.42
Flowervase	10084.84	539.19	2644.79	8.91	17.56	11.33
Allstars	3805.34	701.30	3631.14	13.25	176.10	16.98
Room 3-D	49937.39	1191.03	6172.26	28.34	31.62	28.39
Palace	6036.38	424.50	2153.57	7.76	15.54	10.92
Penguins	4197.37	1441.99	7994.92	29.06	49.62	34.05
Blue Sky	17715.79	2740.67	15721.10	76.19	97.28	85.98
Mean	14104.34	722.06	3803.80	16.16	38.16	18.55

TABLE VI
MEAN BPSNR VALUES FOR THE DISCUSSED ALGORITHMS APPLIED ON FEATURE VECTOR FIELDS

Sequence	RANSAC Original (dB)	RANSAC 4p/8p (dB)	HTE 4p/8p (dB)	RANSAC Preemptive (dB)	RANSAC Adaptive (dB)	RANSAC Adaptive + 4p/8p (dB)
Mountain	38.23	38.30	38.51	38.27	38.52	38.41
Stefan	29.95	30.39	30.75	28.58	30.80	30.59
Allstars CIF	42.20	42.36	42.42	39.63	42.31	42.43
Biathlon	39.15	39.21	39.22	35.31	39.14	39.19
Monaco	40.05	40.19	40.94	40.53	40.88	40.68
Race	37.27	37.28	37.28	36.72	37.27	37.28
Flowervase	36.54	36.66	36.53	36.65	36.47	36.54
Allstars	39.58	40.29	40.25	38.94	40.03	40.23
Room 3-D	36.27	36.27	36.27	36.27	36.27	36.27
Palace	37.66	37.66	37.66	37.65	37.62	37.66
Penguins	32.63	32.63	32.63	32.63	32.62	32.63
Blue Sky	39.49	39.48	39.43	39.49	39.45	39.41
Mean	37.42	37.56	37.66	36.72	37.62	37.61

TABLE VII
AVERAGE PER FRAME RUNTIMES OF THE DISCUSSED ALGORITHMS APPLIED ON FEATURE VECTOR FIELDS

Sequence	RANSAC Original (ms)	RANSAC 4p/8p (ms)	HTE 4p/8p (ms)	RANSAC Preemptive (ms)	RANSAC Adaptive (ms)	RANSAC Adaptive + 4p/8p (ms)
Mountain	14 161	35.89	110.68	0.77	0.84	0.57
Stefan	10 419	46.88	170.47	0.85	7.55	0.89
Allstars CIF	12 927	66.35	240.79	1.24	49.66	1.21
Biathlon	18 158	60.91	221.49	1.06	941.80	1.49
Monaco	9989	68.43	246.25	1.52	1.48	1.22
Race	26 199	75.00	269.52	1.47	3.68	1.40
Flowervase	6889	85.09	318.14	1.95	1.81	1.57
Allstars	3964	79.62	300.09	1.45	15.35	1.47
Room 3-D	4863	85.58	314.87	1.96	1.62	1.56
Palace	2194	86.83	316.63	1.74	4.80	1.60
Penguins	4055	86.92	341.84	1.98	2.07	1.61
Blue Sky	18 574	86.37	331.94	2.02	1.75	1.58
Mean	11 033	71.99	226.23	1.50	86.03	1.35

D. Runtime and Complexity Consideration

As platform for the experimental evaluation (in terms of estimation quality and estimation complexity), a 2.2 GHz AMD Opteron 8354 system with 48 GB RAM has been used. The times for block matching have been taken from the EPZS implementation in the H.264/AVC reference software. For KLT feature tracking, the Birchfield implementation was used [31].

Table VIII provides an overview of the average time needed per frame for EPZS motion estimation and for KLT feature tracking on the test sequences. As the amount of features to be processed has a great impact on the overall runtime of the evaluated regression methods, the average amount of generated motion vectors per frame and sequence is presented as well. Obviously, it is more time efficient to only track up to 400 features than to generate a 16×16 up to 4×4 blockwise dense motion vector field. Additionally, as pointed out in Section IV-C, the higher accuracy and reliability of the KLT features (see Section IV-A2) leads to higher PME precision.

The accuracy of all presented approaches, except for the RANSAC implementation with preemptive termination, is very similar in general. In contrast to that, the discussed methods have very different complexity. Table V shows the

TABLE VIII
RUNTIMES AND AVERAGE AMOUNT OF GENERATED MOTION VECTORS FOR EPZS IN KTA AND KLT FEATURE TRACKING (400 SELECTED FEATURES PER FRAME), RESPECTIVELY

	EPZS		KLT Tracking	
Sequence	Time Per Frame (ms)	Features Per Frame	Time Per Frame (ms)	Features Per Frame
Mountain	690	911.9	154	139.2
Stefan	1128	1297.6	208	204.0
Allstars CIF	870	1500.0	253	299.3
Biathlon	1301	1174.1	266	288.3
Monaco	927	1499.1	251	307.1
Race	2331	2346.1	497	351.5
Flowervase	4530	2932.3	1085	395.2
Allstars	4263	3908.8	1136	374.2
Room 3-D	4533	6501.6	1143	390.7
Palace	26 143	2405.3	1129	392.3
Penguins	10 801	8090.5	2858	398.7
Blue Sky	26 385	14879.6	6526	396.5
Mean	6992	3953.9	1292	328.1

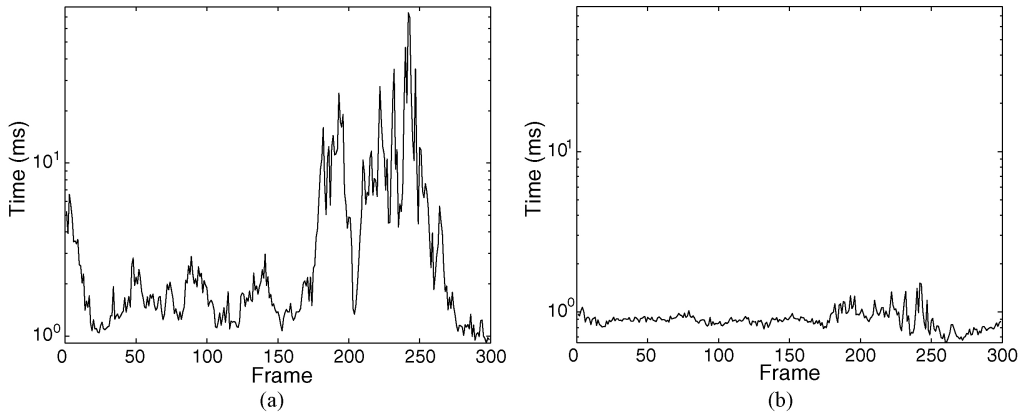


Fig. 9. Per frame runtime (logarithmic) comparison for adaptive RANSAC application on KLT feature vector fields for the *Stefan* sequence. (a) Without two-step simplification. (b) With two-step simplification.

average time per frame needed for PME on H.264/AVC motion vector fields with the proposed methods and schemes. Encoded videos with higher resolutions have more motion vectors per frame and, accordingly, PME on these motion vector fields needs more computation time. The slowest algorithms with an average estimation quality among the considered methods are the unoptimized RANSAC and the HTE with hybrid motion model usage. With PME runtimes of 10 s per frame and more, the usefulness of these two methods for PME on larger H.264/AVC vector fields is doubtful. RANSAC with both simplifications (hybrid model usage and adaptive iteration recalculation) is almost as fast as RANSAC with preemptive abort criterion but does not lack robustness. This can be seen by an estimation quality difference of 1.17 dB between these two methods.

Table VII shows the runtimes of the discussed methods applied on KLT generated feature vector fields. Again, with up to 26 s per frame, the unoptimized RANSAC is the slowest PME method among the discussed ones. Nevertheless, the estimation times are lower, in general, as expected. When assuming 25 f/s as real time, all methods with a PME runtime of 4 ms per frame and below are real-time capable. The fastest average estimation times are 1.5 ms for RANSAC with preemptive abort criterion and 1.35 ms for RANSAC with both simplifications. Additionally, RANSAC with hybrid model usage and adaptive iteration recalculation is highly robust as the BPSNR results show. Thus, this method has the lowest runtime while still delivering global motion models with very high quality.

For a closer look at the runtime savings reachable with the hybrid motion model scheme, Fig. 9 compares the PME runtime per frame of the adaptive RANSAC implementation without [see Fig. 9(a)] and with [see Fig. 9(b)] hybrid motion model simplification on the *Stefan* sequence. As can be seen, sporadically appearing high estimation times per frame (up to 75 ms) of adaptive RANSAC caused by complex motion can be lowered dramatically by the simplification scheme.

E. Final Quality-Complexity Evaluation

To get a better look at the reasons for the measured runtimes, Table IX summarizes the average per frame iterations for estimation on H.264/AVC motion vector fields exemplarily. With the aforementioned settings for ϵ and P, the original RANSAC has a fixed per frame amount of about $2.1 \cdot 10^6$ iterations, HTE with hybrid model usage (also called 4p/8p) and RANSAC with hybrid model usage do about $3.3 \cdot 10^3$ iterations per frame. Among the methods with varying iteration count, the highest amount of per frame iterations in average

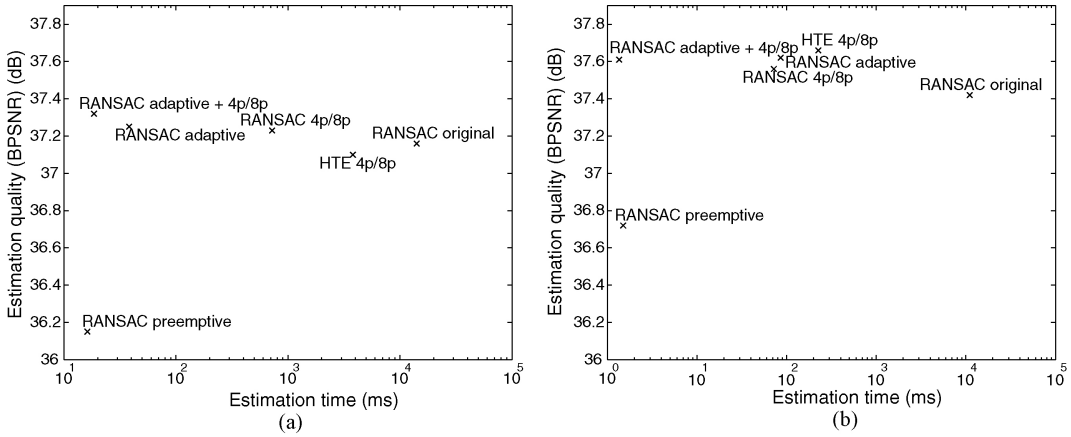


Fig. 10. Average estimation quality in terms of BPSNR (dB) over average estimation time (logarithmic) for the discussed PME methods (top left is better). (a) Estimation on H.264/AVC motion vector fields. (b) Estimation on KLT motion vector fields.

TABLE IX
AVERAGE PER FRAME ITERATIONS FOR APPLICATION OF THE EVALUATED PME METHODS ON H.264/AVC MOTION VECTOR FIELDS

Sequence	RANSAC Original	RANSAC 4p/8p	HTE 4p/8p	RANSAC Preemptive	RANSAC Adaptive	RANSAC Adaptive + 4p/8p
Mountain	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.0	11.4	5.0
Stefan	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.2	110.0	21.6
Allstars CIF	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.2	135.8	11.8
Biathlon	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.8	254.0	36.9
Monaco	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.0	18.5	6.6
Race	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.4	85.0	16.1
Flowervase	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.0	51.3	10.4
Allstars	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	15.5	3176.3	48.3
Room 3-D	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.0	16.6	13.1
Palace	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	7.9	52.9	29.2
Penguins	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.0	55.3	11.6
Blue Sky	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.1	12.0	8.5
Mean	$2.07 \cdot 10^6$	$3.3 \cdot 10^3$	$3.3 \cdot 10^3$	9.6	331.6	18.3

is performed by adaptive RANSAC when applied on vector fields of the *Allstars* sequence.

A summarizing overview of estimation times and estimation qualities on H.264/AVC and KLT motion vector fields with the discussed methods is given in Fig. 10.

For estimation on H.264/AVC motion vectors, preemptive RANSAC performs fastest but highly lacks in estimation quality, while adaptive RANSAC with the proposed hybrid model simplification has comparable per frame estimation. For KLT feature-based estimation adaptive RANSAC, the HTE approach from [32] and adaptive RANSAC with the simplification show comparable results in terms of quality while the simplified version of adaptive RANSAC estimates much faster. Thus, for both application scenarios (estimation on encoded video and estimation on KLT features) adaptive RANSAC with hybrid model simplification performs best in terms of estimation quality and time.

V. CONCLUSION

Highly robust PME methods for the compressed domain as well as for the pixel domain were compared in terms of estimation quality and runtime. A new simplification scheme based on hybrid model utilization for PME was presented and evaluated. Results for this scheme combined with other well-known simplification techniques for Monte-Carlo-based estimation methods were given and the advantages of this scheme were pointed out. The well-known RANSAC and the more recent HTE were revisited and successively improved with the discussed simplification schemes. BPSNR results in combination with runtimes of the compared methods provided an overview of the potential of the presented methods. Due to the high-quality PMMs generated by these PME techniques, most of them can be used for different application scenarios such as motion-based segmentation, higher-order motion compensation for inter prediction, efficient transcoding of video streams, or motion characterization for analysis purpose.

RANSAC, in combination with the proposed hybrid motion model utilization and adaptive iteration amount calculation, outperformed all other proposed methods in terms of runtime while still estimating parameters with robustness and accuracy. It estimated PMMs with a quality comparable to that of RANSAC without simplifications but estimated up to 10 000 times faster. Applied on feature vector fields, the twice optimized RANSAC is 60 times faster than the well-known RANSAC implementation of [24], which is optimized with only the adaptive iteration amount calculation. This combination of low runtime and high PME precision can lead to reliable high-quality real-time PME applications.

REFERENCES

[1] 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, Mar. 2000.

[2] A. Krutz, M. Frater, M. Kunter, and T. Sikora, "Windowed image registration for robust mosaicing of scenes with large background occlusions," in *Proc. 13th IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 353–356.

[3] K. Yang, M. Frater, E. Huntington, M. Pickering, and J. Arnold, "Low precision global motion estimation for video compression: A generalized framework," in *Proc. Dig. Image Comput.: Tech. Applicat.*, Dec. 2008, pp. 405–411.

[4] N. Tarannum, M. Pickering, and M. Frater, "A new M-estimator approach for global motion estimation," in *Proc. Dig. Image Comput.: Tech. Applicat.*, Dec. 2008, pp. 9–15.

[5] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[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, Feb. 2005.

[7] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, and H. Sun, "Survey of compressed-domain features used in audio-visual indexing and analysis," *J. Vis. Commun. Image Representat.*, vol. 14, no. 2, pp. 150–183, 2003.

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

[9] A. Desolneux, L. Moisan, and J.-M. More, "A grouping principle and four applications," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 4, pp. 508–513, Apr. 2003.

[10] R. L. Felip, X. Binefa, and J. Diaz-Caro, "A new parameter estimator based on the Helmholtz principle," in *Proc. 12th IEEE Int. Conf. Image Process.*, vol. 2. Nov. 2005, pp. 306–309.

[11] R. L. Felip, L. Barcelo, X. Binefa, and J. Kender, "Robust dominant motion estimation using MPEG information in sport sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 12–22, Jan. 2008.

[12] M. Tok, A. Glantz, M. G. Arvanitidou, A. Krutz, and T. Sikora, "Compressed domain global motion estimation using the Helmholtz tradeoff estimator," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 777–780.

[13] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[14] M. Kunter, P. Krey, A. Krutz, and T. Sikora, "Extending H.264/AVC with a background sprite prediction mode," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2128–2131.

[15] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[16] A. Glantz, A. Krutz, and T. Sikora, "Global motion temporal filtering for in-loop deblocking," in *Proc. 17th IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 957–960.

[17] T. Wiegand, E. Steinbach, and B. Girod, "Affine multipicture motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 197–209, Feb. 2005.

[18] D. Farin, P. de With, and W. Effelsberg, "Video-object segmentation using multi-sprite background subtraction," in *Proc. IEEE Int. Conf. Multimedia Expo.*, vol. 1. Jun. 2004, pp. 343–346.

[19] A. Krutz, A. Glantz, T. Borgmann, M. Frater, and T. Sikora, "Motion-based object segmentation using local background sprites," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1221–1224.

[20] A. Glantz, A. Krutz, T. Sikora, P. Nunes, and F. Pereira, "Automatic MPEG-4 sprite coding: Comparison of integrated object segmentation algorithms," *Multimedia Tools Applicat.*, vol. 49, pp. 483–512, Sep. 2010.

[21] M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proc. IEEE*, vol. 86, no. 5, pp. 905–921, May 1998.

[22] H. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, no. 8, pp. 814–830, Aug. 1996.

[23] G. Ye, M. Pickering, M. Frater, and J. Arnold, "A robust approach to super-resolution sprite generation," in *Proc. 12th IEEE Int. Conf. Image Process.*, vol. 1. Sep. 2005, pp. 897–900.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[25] L. Barcelo, R. Felip, and X. Binefa, "A new approach for real time motion imation using robust statistics and MPEG domain applied to mosaic images construction," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2005, pp. 1–4.

[26] (2010, Nov.) [Online]. Available: http://www.tnt.uni-hannover.de/∼vatis/kta

[27] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. SPIE Conf. Vis. Commun. Image Process.*, Jan. 2002, pp. 1069–1079.

[28] *MPEG-2*, document ISO/IEC IS 11172-2, 2000.

[29] *MPEG-4*, document ISO/IEC IS 14496-2, 2004.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vision Patt. Recognit.*, Jun. 1994, pp. 593–600.

[31] *Birchfield Implementation of the KLT Feature Tracker*. (2010, Nov.) [Online]. Available: http://www.ces.clemson.edu/$\sim$stb/klt

[32] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Feature-based global motion estimation using the Helmholtz principle," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2011, pp. 1561–1564.

Michael Tok (S'10) received the Dipl.-Ing. degree in computer engineering from the Technische Universität Berlin, Berlin, Germany, where he is currently pursuing the Dr.-Ing. degree with the Communication Systems Group.

He is currently involved in the International Telecommunication Union and International Organization for Standardization/IOC standardization activities leading to the next video coding standard high-efficiency video coding. His current research interests include multidimensional signal processing, hybrid video coding, object-based video coding, global motion estimation, and motion modeling.

Alexander Glantz (S'08) received the Dipl.-Ing. degree in computer engineering from the Technische Universität Berlin, Berlin, Germany, where he is currently pursuing the Dr.-Ing. degree with the Communication Systems Group.

He is involved in the European Networks of Excellence VISNET II and PetaMedia and in the International Telecommunication Union and ISO standardization activities. His current research interests include multimedia signal processing, hybrid video coding, rate-distortion theory, global motion estimation, and object segmentation in video sequences. He is a member of the German Society for Information Technology.

Andreas Krutz (M'10) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the Technische Universität Berlin, Berlin, Germany, in 2006 and 2010, respectively.

He is currently a Post-Doctoral Researcher with the Communication Systems Group, Technische Universität Berlin. He was a Student Research Assistant with the University College, University of New South Wales, Canberra, Australia, where he researched image processing and video coding. He was involved in four European network of excellences, namely, 3-DTV, K-space, VISNET II, and PetaMedia. Within the K-space project, he worked on video analysis techniques, especially on object segmentation. Within the remaining three projects, he worked on new techniques for hybrid video coding based on analysis or synthesis approaches. In 2008, he was a Guest Researcher with the University of New South Wales and the Co-Chair of a Special Session at MMSP in Cairns, Australia.

Dr. Krutz is currently an active contributor to the JCT-VC, a joint standardization activity of the International Telecommunication Union and ISO/IEC MPEG leading to the next video coding standard high-efficiency video coding. He is a member of VDE/ITG.

Thomas Sikora (SM'96) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from Bremen University, Bremen, Germany, in 1985 and 1989, respectively.

He is currently a Professor and the Director of the Communication Systems Group (Nachrichtenübertragung), Technische Universität Berlin, Berlin, Germany. In 1990, he joined Siemens Ltd. and Monash University, Melbourne, Australia, as a Project Leader, responsible for video compression research activities in the Australian Universal Broadband Video Codec consortium. Between 1994 and 2001, he was the Director of the Department of Interactive Media, Heinrich Hertz Institute, Berlin. In 2011, he was elected as a member of the Berlin-Brandenburg Academy of Science, Berlin. He is a co-founder of Vis-a-Pix GmbH (www.visapix.com) and imcube media GmbH (www.imcube.com), two Berlin-based startup companies involved in research and development of audio and video signal processing and compression technology. He is an Appointed Member of the Advisory and Supervisory Board of a number of German companies and international research organizations. He frequently serves as an Industry Consultant and Patent Reviewer on issues related to interactive digital audio and video. He has been involved in the International Telecommunication Union and ISO standardization activities as well as several European research projects for more than 15 years. As the Chairman of the ISO-MPEG video group, he was responsible for the development and standardization of the MPEG-4 and MPEG-7 video algorithms. He was also the Chairman of the European COST 211ter Video Compression Research Group. He was appointed as the Research Chair of the VISNET and 3-DTV European Networks of Excellence. He is a frequent Invited Speaker at international conferences. He has published more than 300 journal and conference papers related to audio and video processing. He co-authored three books: *Introduction to MPEG-7: Multimedia Content Description Interface* (New York: Wiley, 2002), *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval* (New York: Wiley, 2005), and *3D Videocommunication: Algorithms, Concepts, and Real-Time Systems in User-Centered Communications* (New York: Wiley, 2005).

Dr. Sikora was a recipient of the ITG Award in 1996. As a member of the ISO MPEG-2 Video Standards Group, he also received the Engineering Emmy Award of the U.S. National Academy of Television Arts and Sciences in 1996. He is a member of the German Society for Information Technology. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. From 1998 to 2002, he was an Associate Editor of the *IEEE Signal Processing Magazine*. He is an Advisory Editor of the *EURASIP Signal Processing: Image Communication Journal* and was an Associate Editor of the *EURASIP Signal Processing Journal*.