

Robust Local Optical Flow for Feature Tracking

Tobias Senst, *Student Member, IEEE*, Volker Eiselein, *Student Member, IEEE*,
and Thomas Sikora, *Senior Member, IEEE*

Abstract—This paper is motivated by the problem of local motion estimation via robust regression with linear models. In order to increase the robustness of the motion estimates, we propose a novel robust local optical flow approach based on a modified Hampel estimator. We show the deficiencies of the least squares estimator used by the standard Kanade–Lucas–Tomasi (KLT) tracker when the assumptions made by Lucas–Kanade are violated. We propose a strategy to adapt the window sizes to cope with the generalized aperture problem. Finally, we evaluate our method on the Middlebury and MIT dataset and show that the algorithm provides excellent feature tracking performance with only slightly increased computational complexity compared to KLT. To facilitate further development, the presented algorithm can be downloaded from <http://www.nue.tu-berlin.de/menue/forschung/projekte/rlof>.

Index Terms—Feature tracking, Hampel, Kanade–Lucas–Tomasi (KLT), long-term trajectories, optical flow, robust estimation.

I. INTRODUCTION

THE COMPUTATION of 2-D image velocities, or optical flow, is a common topic in computer vision. Our challenge is to estimate the scene or object motion as precisely and computationally efficiently as possible. Common optical flow techniques exploit two constraints: data conservation and spatial coherence. Data conservation is derived from the observation that the observed objects generally persist in time. Thus, the intensity of a small region in two consecutive images remains constant, although its position is changing. This leads to the mathematical formulation of the intensity constancy assumption as follows:

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t) \quad (1)$$

with $I(x, y, t)$ as the image intensity of a grayscaled image, $\mathbf{d} = (u, v)^T$ denoting the displacement of a point, and δt as a small time difference at a position $\mathbf{x} = (x, y)$. Different approaches to solve this equation have been described widely in the literature [1]. The most successful methods to compute \mathbf{d} use a linearization of (1) performed by a first-order Taylor approximation and are therefore gradient based. This leads to

Manuscript received July 20, 2011; revised November 29, 2011; accepted June 1, 2012. Date of publication June 1, 2012; date of current version August 30, 2012. This work was supported by the European Community FP7, under Grant 261743 (NoE VideoSense). This paper was recommended by Associate Editor J. F. Arnold.

The authors are with the Communication Systems Group, Technische Universität Berlin, Berlin 10623, Germany (e-mail: senst@nue.tu-berlin.de; eiselein@nue.tu-berlin.de; sikora@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.2012.2202070

an underdetermined linear system. To solve this system, two kinds of spatial coherence conditions have been introduced and established: the techniques of *global* and *local* gradient-based approach. By introducing an additional *global constraint*, Horn and Schunck [2] applied a soft spatial coherence forcing the *partial derivatives of neighboring motion vectors to be minimal*. A strong spatial coherence was introduced by Lucas and Kanade [3] that is categorized as *local constraint expecting the motion in a small region to be constant*.

These assumptions are simplifications and hence may be violated in practice. For example, motion boundaries violate the common assumption that the optical flow varies smoothly. As described by Black and Anandan [4], the violations result in gross measurement errors that are referred to as outliers. Since Horn–Schunck and Lucas–Kanade penalized the minimization in a quadratic way, the model does not handle outliers robustly. Black and Anandan [5] proposed a robust estimation framework exploiting the Lorentzian robust norm.

Most of the state-of-the-art global optical flow methods are using robust estimation frameworks. Common norms are the modified L^1 , which is successfully used in different solutions, e.g., by Brox *et al.* [6] or the Huber- L^1 norm used by Werlberger *et al.* [7]. An additional benefit can be achieved by combining this with more sophisticated total-variation techniques, as in Papenberg *et al.* [8] and Zach *et al.* [9]. Generally, global optical flow methods achieve a superior accuracy compared to local optical flow methods. Sand and Teller [10] proposed the particle video framework to compute a dense set of long-term trajectories from dense optical flow, which is rather slow. In [11], a large displacement optical flow as introduced in [12] was used to create dense point trajectories with a high performance.

Yet applications such as robot navigation, augmented reality, visual attention, and camera self-calibration require very fast detection of interest points and the subsequent search for potential correspondences in real time. *Methods with excellent runtime performance exploiting local optical flow techniques, such as the popular Kanade–Lucas–Tomasi (KLT) tracking algorithm [13], are still applied in many cases*. Comparative studies indicate that the Lucas–Kanade algorithms provide accurate results [1] while being significantly more efficient [14] than other optical flow methods.

Research of local methods is often motivated by improving the runtime performance, e.g., Senst *et al.* [15] proposed integral images to decrease the computational complexity per interest point. Sinha *et al.* [16], Zach *et al.* [17], and Fassold *et al.* [18] improved the runtime performance by parallelizing

the algorithm and porting it onto a graphics processing unit (GPU). Although in many global optical flow methods, robust estimation techniques are established, most local methods are currently based on least square optimization. Gain adaptive modifications were proposed by Zach *et al.* [17] and Kharbat *et al.* [19]. Kim *et al.* [20] proposed an approach robust to varying illumination using a least median of squares method that is robust but increases the runtime drastically. Robust motion estimation for tracking has also been investigated in [21] as an application of [22]. Baker *et al.* [23] have investigated Lucas–Kanade in detail regarding the image alignment problem and a higher ordered parameterized warping model.

The aim of this paper is to introduce an efficient and robust local gradient-based feature tracking system that was earlier motivated and proposed in [24]. In this paper, we evaluate shortcomings of the established KLT method. Based on empirical findings, we establish a novel robust local optical flow (RLOF) algorithm based on a modified Hampel estimator. We show that this algorithm provides excellent feature tracking performance with only slightly increased computational complexity compared to KLT.

The remainder of this paper is organized as follows. Section II describes the Lucas–Kanade approach analogous to the regression of a linear model. It will be shown that this least squares estimate behaves badly when assumptions are violated. In Sections III and IV, we introduce our feature tracker based on a robust estimation framework, and in Section V, we evaluate our results regarding the MIT database [25].

II. LUCAS–KANADE IN A STATISTICAL CONTEXT

In this section, we focus on the Lucas–Kanade algorithm viewed in a statistical context.

The general gradient-based local optical flow constraint is formulated as follows:

$$E_{\text{local}} = \sum_{\Omega} w(\mathbf{x}) \cdot \rho(\nabla I(\mathbf{x})^T \cdot \mathbf{d} + I_t(\mathbf{x}), \sigma). \quad (2)$$

To find a displacement \mathbf{d} , the residual error E_{local} is minimized for a small image region $\mathbf{x} \in \Omega$, with the spatial derivatives $\nabla I(\mathbf{x}) = (I_x(\mathbf{x}, t), I_y(\mathbf{x}, t))^T$ and the temporal derivative $I_t(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x}, t+1)$, $w(\mathbf{x})$ a weighting function, and a norm ρ , with its scale parameters σ and $\mathbf{x} = (x, y)$ image pixel positions.

Consider a linear model and its residual as follows:

$$\epsilon_i = \hat{y}_i - \hat{x}_{i0}\theta_0 - \dots - \hat{x}_{ij}\theta_j \quad (3)$$

for the i th of n observations $(\hat{x}_{i0}, \dots, \hat{x}_{ij}, \hat{y}_i)$. If ϵ_i are independent and normally distributed, Gauss proved that the optimal fit $\hat{\theta}$ of parameters $\theta = (\theta_0, \dots, \theta_j)$ can be estimated with the least sum of squares

$$\min_{\theta} \sum_{i=0}^{n-1} \epsilon_i^2. \quad (4)$$

The least squares estimator relies on a very efficient computational complexity. The optimal parameter $\hat{\theta}$ (4) can be directly computed by an explicit formulation. However, this estimator is very sensitive to outliers [26], [27].

A. Lucas–Kanade Locally Constant Flow

The Lucas–Kanade method is given by the gradient-based formulation as follows:

$$\min_{\mathbf{d}} \sum_{\Omega} (\nabla I(\mathbf{x})^T \cdot \mathbf{d} + I_t(\mathbf{x}))^2. \quad (5)$$

To find a displacement \mathbf{d} , the sum of least squares is minimized for a small image region Ω . In the statistical context, a regression for a linear model $\hat{y} - \hat{\mathbf{x}}^T \cdot \theta$ with the parameters $\theta = \mathbf{d}$ is done using the observations $(\hat{x}_0, \hat{x}_1, \hat{y}) = (I_x, I_y, I_t)$ of the region Ω .

1) *Linearization Problem*: The original Lucas–Kanade method [3] is restricted by the first-order Taylor approximation. The assumption of a linear relation between the gradients ∇I and the motion is only accurate for small motions. To cope with small linearization errors, Bouguet [28] proposed an iterative solution in a Newton–Raphson fashion. \mathbf{d} is iteratively solved for increments to the displacement $\Delta \mathbf{d}$ as follows:

$$\Delta \mathbf{d}^i = \mathbf{G}^{-1} \cdot \left[\sum_{\Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}) \right] \quad (6)$$

where \mathbf{G} denotes the Hessian. The resulting displacement is updated as follows:

$$\mathbf{d}^i \leftarrow \mathbf{d}^{i-1} + \Delta \mathbf{d}^i \quad (7)$$

with the second frame being updated at each iteration i so that $I_t^{i-1}(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x} + \mathbf{d}^{i-1}, t+1)$. The iterative solution is initialized with $d = (0, 0)^T$. In the literature, (6) has also been described by [23] as the inverse compositional algorithm for translational warps. To cope with motion larger than a single pixel, a coarse-to-fine strategy is employed in which pyramids of spatially filtered and subsampled images are created.

2) *Aperture Problem*: Equation (6) has a limitation, which is commonly referred to as the aperture problem. There exists only a solution of \mathbf{d} if \mathbf{G} is not singular. This implies the existence of gradients in x and y direction in the observed region Ω . In consequence, the Lucas–Kanade algorithm could not be applied on homogeneous image content. To overcome this problem, a large region Ω is needed to increase the probability that the region contains edges.

3) *Generalized Aperture Problem*: While a large region is needed to constrain the solution and provide insensitivity to noise, it also increases the risk of violating the local constancy assumption, whereby a region should be described by only one motion. Contrarily, a small region decreases the probability that a region contains discriminative edges. That dilemma is referred to as generalized aperture problem [29].

B. Observation Distributions

As discussed in Section II-A, the assumptions underlying the Lucas–Kanade algorithm can be violated by homogeneous regions, motion boundaries, the appearance and disappearance of pixels, e.g., by occlusion, and changes of illumination. This section studies the characteristics and distribution of the observed data I_x , I_y , and I_t to motivate the robust framework proposed in Section IV. It should not be seen as a complete investigation but as an illustration of the potential problems of

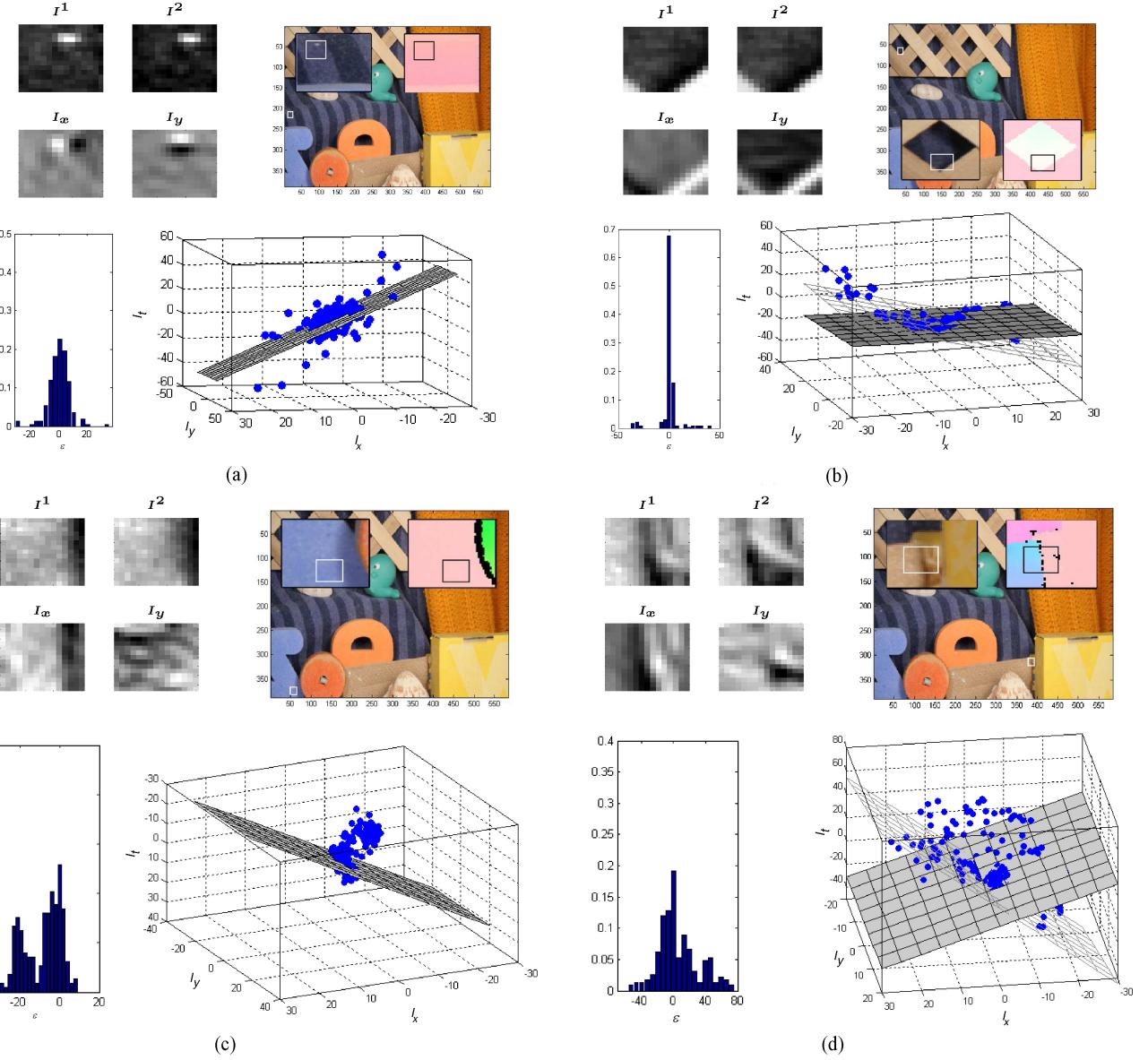


Fig. 1. (a) Distribution of the residual error ϵ (bottom right) and the data of the spatial and temporal derivatives (bottom left) fulfilling the Lucas–Kanade assumptions. The data scatter plot includes the ground truth motion as plane. (b) Different motions where the motion of the region center is illustrated as gray plane and the motion of the bottom-right region border as transparent plane. (c) Setting with changing illuminations affected by shadows. (d) Appearance of occluded pixels by different motions. The illustration at the top shows the derivatives of the region (left) and an overview of the region position containing a magnification of the relevant area and a magnification of the color-coded ground truth.

the standard Lucas–Kanade method. Similar experiments for settings of different motion were already conducted by Black and Anandan [5]. By the use of two different synthetic moving patterns they show that a robust estimation could suppress the nondominant motion efficiently in theory. To apply the experiments in a more realistic scene, the *RubberWhale* sequence of the Middlebury dataset [30] is used. The Middlebury dataset includes synthetic and realistic pairs of consecutively captured images and provides the optical flow as ground truth for each pair. To show the distribution of the observed data, two types of plots are used. At first, the distribution of the residual (3) is displayed (see Fig. 1, bottom left). To get a more detailed view, the distribution of the observed data is displayed with a scatter plot for each region (see Fig. 1, bottom right). The used region size (17×17) corresponds to a common size for the

Lucas–Kanade algorithm. Additionally, the spatial gradients I_x , I_y and different temporal pixel values I^1 , I^2 contained by the region are shown at the top left while the coarse position of the region can be seen at the top right. The picture contains a magnification of the relevant area and a magnification of the color-coded ground truth, see [30].

The first test is shown by Fig. 1(a), where the data is captured at position (17, 213) and the region fulfills the Lucas–Kanade assumption. It includes a single motion; constant illuminations and spatial gradients in x and y direction prevent the gradient matrix \mathbf{G} from singularity. The distribution of the residual and data is near to be normally distributed. The second test shown in Fig. 1(b) is captured at the position (33, 64) in which the region contains two different motions. The plane of the second motion is displayed with transparent faces.

Obviously, the distribution of the residual and observed data does not follow a Gaussian distribution.

A test of changing illumination is shown in Fig. 1(c), captured at the position (50, 374). The different illumination is affected by the moving shadow of the red wheel. Obviously, the distribution of the residual and data is biased and not normally distributed. The last test at Fig. 1(d) plots a sample of a region that includes an appearing texture. It is captured at position (386, 312). In contrast to the experiment shown in Fig. 1(b), the data includes outliers, which are not affected and do not belong to one of the two motions. This results in an asymmetric and non-Gaussian distribution.

Most of the problems in realistic scenes occur at motion boundaries. The violation of the local constancy assumption mostly coincides with the violation of the intensity constancy assumption by occlusion. In relation to the KLT-Tracker [13], this becomes an important aspect. To avoid the aperture problem, this often-used tracker performs a feature selection algorithm (e.g., good features to track [31]) to detect corner points with high edges in both directions and thus a high minimal eigenvalue of \mathbf{G} . In practice, these edges lie in all likelihood at motion boundaries. This gives the motivation to use a more robust estimator than the least squares estimator.

III. ROBUST ESTIMATOR WITH PIECEWISE LINEAR INFLUENCE FUNCTIONS

Huber [32] formed the first basis for a theory of robust estimation and introduced a class of estimator, called *M-estimator* as follows:

$$\min_{\theta} \sum_{i=0}^{n-1} \rho(\epsilon_i, \sigma) \quad (8)$$

where ρ is an error norm, with its scale parameters σ , which may or may not be present. The M-estimator is a generalization of the well-known maximum likelihood estimator without assuming that ρ is of the form $-\log f$ for any probability density f . The robustness of the estimator depends on its error norm. An approach based on the influence function was introduced by Hampel [33]. The influence function $\psi(y, \sigma)$ as the derivative of the estimator $\rho(y, \sigma)$ characterizes the bias that a particular measurement has on the solution. The estimation of θ can be given by the influence function

$$\sum_{i=0}^{n-1} \psi(\epsilon_i, \sigma) \cdot \frac{\partial \epsilon_i}{\partial \theta} = 0 \quad (9)$$

Fig. 2 shows the drawback of the least squares solution: outliers are assigned a too high weight by the quadratic error norm $\rho = y^2$. More robust norms are the L^1 norm [8], [9], [32] and Huber's minmax norm [7], [34] because their influence functions are limited. Both are equivalent for large values, but for normally distributed data, the L^1 norm produces estimates with higher variance than the optimal quadratic L^2 norm [34], [35]. In contrast, Huber's minmax norm is designed to be quadratic for small values, which makes it applicable for Newton.

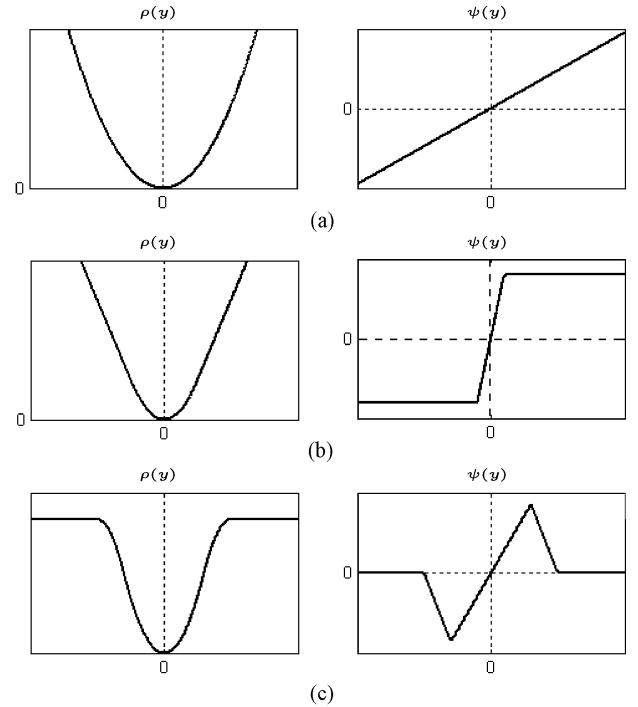


Fig. 2. Common error norms ρ with piecewise linear influence functions ψ . (a) Quadratic. (b) Huber. (c) Shrunked Hampel.

A. Redescending Influence Function

To increase the robustness, the influence of extremely discordant observations should be reduced to zero. So ψ should be designed to vanish for large values. This was a reason to develop redescending M-estimators that Huber described in [35]. An advantage of redescending M-estimators is that they have very low breakdown points [33]. Black and Anandan [5] proposed an implementation for the optical flow using the Lorentzian norm

$$\psi(\epsilon_i, \sigma) = \frac{2\epsilon_i}{2\sigma^2 + \epsilon_i^2} \quad (10)$$

to improve the behavior of motion computation at motion boundaries. Odobez and Bouzemy [22] proposed a redescending M-estimator implementation based on Tukey's biweight norm [36] as follows:

$$\psi(\epsilon_i, \sigma) = \begin{cases} \epsilon_i (\sigma^2 - \epsilon_i^2)^2, & |\epsilon_i| < \sigma \\ 0, & \text{else.} \end{cases} \quad (11)$$

The M-estimator problems using the robust norms (11) and (10) cannot be solved analytically, which is a drawback with regard to computational complexity. The solution can instead be found using the iteratively reweighted least squares method by transforming the M-estimation problem into an equivalent weighted least squares problem [22] as follows:

$$\sum_{i=0}^{n-1} \rho(\epsilon_i, \sigma) = \frac{1}{2} \sum_{i=0}^{n-1} w_i \cdot \epsilon_i^2 \quad (12)$$

where the weight at each observation x_i is given by

$$w_i = \frac{\psi(\epsilon_i)}{\epsilon_i}. \quad (13)$$

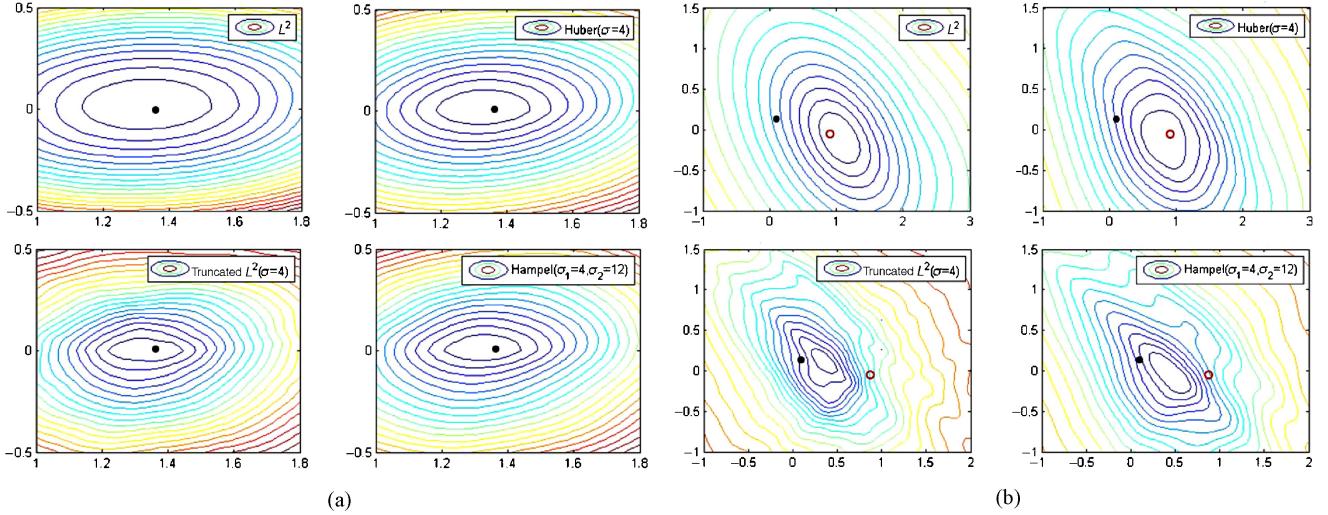


Fig. 3. Error surfaces of the residual error sum obtained by varying (u, v) with the error functions shown in Fig. 2 in the scenario of Fig. 1. (a) Not violated. (b) Motion boundaries. The ground truth motion of (a) is $\mathbf{d}(17, 213) = (1.36, -0.01)$ and the center motion of (b) is $\mathbf{d}(33, 64) = (0.09, 0.15)$ including a second motion at a neighboring pixel $\mathbf{d}(41, 70) = (0.91, -0.08)$. Ideally, the minima in the plots should be at the ground truth motion and smooth surfaces should simplify the gradient descent. Black dot: ground truth of the center motion. Red circle: corresponding second motion. Blue lines: small values. Red lines: big values.

Another alternative to the M-estimator is the least median of squares estimator (LMedS), advantages of which lie in its theoretical high robustness. But as Odobez and Boutheny stated, the computational cost of LMedS is very high and increases rapidly with the amount of data.

A primary aim of this paper is to keep the computational effort as low as possible. We therefore want to focus on estimator classes with an influence function composed of linear functions. Fig. 2 shows common error norms based on composed quadratic functions. We base our approach on the Hampel estimator but reduce the number of its parameters by shrinking the high and low flat segment to

$$\rho(\epsilon_i, \sigma) = \begin{cases} \epsilon_i^2, & |\epsilon_i| \leq \sigma_1 \\ \sigma_1 \sigma_2, & |\epsilon_i| \geq \sigma_2 \\ \frac{\sigma_1(|\epsilon_i| - \sigma_2)^2}{\sigma_1 - \sigma_2} + \sigma_1 \sigma_2, & \text{else} \end{cases} \quad (14)$$

with the influence function

$$\psi(\epsilon_i, \sigma) = \begin{cases} 2\epsilon_i, & |\epsilon_i| \leq \sigma_1 \\ 0, & |\epsilon_i| \geq \sigma_2 \\ \frac{\sigma_1(\epsilon_i - \text{sign}(\epsilon_i) \cdot \sigma_2)}{0.5(\sigma_1 - \sigma_2)}, & \text{else.} \end{cases} \quad (15)$$

Several algorithms were investigated by Dutter [37] including Newton's method, which we want to use to solve (9). Newton's method has the remarkable property of reaching the theoretically exact solution in one single step, if the start value is close to the solution and a (composed) quadratic error norm is used [35]. A comparative study of these redescending M-estimators is given by Shevlyakov *et al.* [38]. Table I shows the evaluation of dense optical flow from different M-estimator for images in the Middlebury dataset. The reference KLT method based on the quadratic norm is compared to the M-estimator implementations regarding [22], available at <http://www.irisa.fr/vista/Motion2D/about.html>, for the Talwar, Tukey, Cauchy, and Welsh norm, and to the RLOF. RLOF

shown in three variants: 1) as described in Section IV; 2) with the shrunked Hampel norm and having a fixed region size (RLOF*); and 3) with quadratic norm and adapted region size (RLOF#). With a relatively short runtime, the accuracy of the RLOF* is comparable to the other robust norms. Further improvement is reached by adapting the region size of the RLOF*. The runtime was computed at the *Grove3* 640×480 sequence by CPU specification regarding Section V. All methods use default parameters and a region size of 17×17 .

B. Experiments With Composed Quadratic Error Norms

In this section, we want to study the effect of the different norms (see Fig. 2) to the Lucas–Kanade method with regard to exemplary data distributions shown in Fig. 1. Figs. 3 and 4 show error surfaces of the observations rendered for a region of size 17×17 . The error surfaces display the sum of the residual error $\epsilon_i(u, v)$ weighted with the L^2 , truncated L^2 , and Huber or shrunked Hampel error function while varying (u, v) . Ground truth motion is taken from the dataset and used as solution information for the center pixel of the region.

To validate the normally distributed data, the experiment with Fig. 1(a) is shown in Fig. 3(a). The minima of all error surfaces are near the expected ground truth value.

Fig. 4(a) and (b) shows cases of appearing pixels, in Fig. 4(a), the appearance of the shadow at the right region side, and in Fig. 4(b), uncovered pixels, both violating the intensity constancy assumption.

Minima of the Huber norm are closer to the ground truth than the minimum of the least squares norm. However, redescending influence (i.e., bounded influence) gives an additional improvement of the behavior concerning gross outliers. As stated by Hampel [33], nonmonotone influence functions should be used with caution. In general, they should also not descend too steeply. Fig. 4(b) shows that the truncated L^2 norm produces a more nonconvex error surface with local minima that could corrupt the minimization of (8).

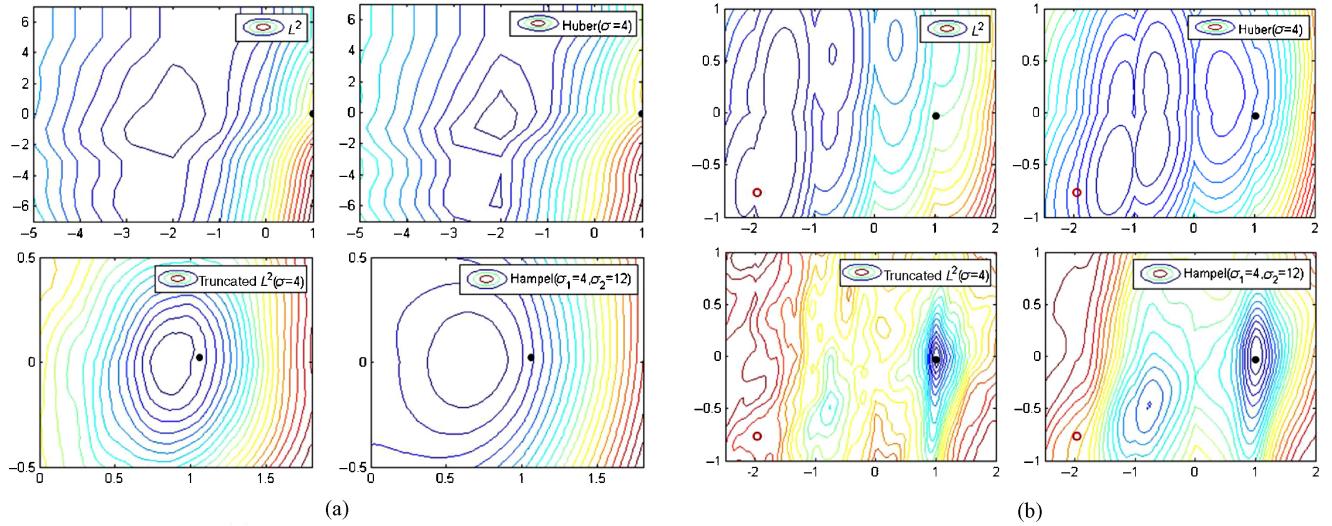


Fig. 4. Error surfaces of the residual error sum obtained by varying (u, v) with the error functions shown in Fig. 2 in the scenario of Fig. 1. (a) Changing illuminations. (b) Appearing pixels. The ground truth motion of (a) is $\mathbf{d}(50, 374) = (1.06, 0.02)$ and the center motion of (b) is $\mathbf{d}(386, 312) = (0.96, -0.02)$ including a second motion $\mathbf{d}(378, 312) = (-1.94, -0.76)$. Ideally, the minima in the plots should be at the ground truth motion and smooth surfaces should simplify the gradient descent. The ground truth of the center motion are shown as a black dot and the corresponding second motion as a red circle. The blue lines denote small values while the red lines denote big values.

TABLE I
AEE AND THE R0.5 ERROR FOR MIDDLEBURY DATASET

| | Dimetrodon | | Grove2 | | Grove3 | | | Hydrangea | | RubberWhale | | Urban2 | | Urban3 | | Venus | |
|--------|-------------|-------------|-------------|------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | AEE | R0.5 | AEE | R0.5 | AEE | R0.5 | Runtime (s) | AEE | R0.5 |
| KLT | 1.24 | 0.14 | 2.50 | 0.15 | 1.75 | 0.36 | 76 | 2.13 | 0.24 | 1.76 | 0.15 | 1.83 | 0.21 | 2.30 | 0.25 | 0.96 | 0.16 |
| RLOF* | 0.93 | 0.14 | 0.30 | 0.10 | 1.04 | 0.29 | 75 | 1.07 | 0.22 | 0.32 | 0.13 | 1.27 | 0.20 | 1.31 | 0.24 | 0.86 | 0.15 |
| RLOF# | 2.69 | 0.16 | 12.78 | 0.26 | 6.33 | 0.39 | 72 | 0.79 | 0.30 | 0.39 | 0.15 | 10.36 | 0.30 | 9.12 | 0.37 | 0.93 | 0.23 |
| RLOF | 0.20 | 0.13 | 0.23 | 0.09 | 0.78 | 0.26 | 73 | 0.35 | 0.20 | 0.25 | 0.11 | 0.80 | 0.17 | 0.85 | 0.23 | 0.48 | 0.16 |
| Talwar | 0.24 | 0.14 | 0.27 | 0.09 | 1.27 | 0.28 | 839 | 0.50 | 0.23 | 0.32 | 0.13 | 5.90 | 0.41 | 4.81 | 0.40 | 1.33 | 0.27 |
| Tukey | 0.24 | 0.14 | 0.27 | 0.09 | 1.27 | 0.28 | 789 | 0.50 | 0.24 | 0.32 | 0.13 | 5.89 | 0.41 | 4.84 | 0.40 | 1.31 | 0.27 |
| Cauchy | 0.24 | 0.14 | 0.28 | 0.10 | 1.22 | 0.29 | 811 | 0.48 | 0.23 | 0.32 | 0.14 | 5.75 | 0.39 | 4.80 | 0.40 | 1.29 | 0.26 |
| Welsh | 0.24 | 0.14 | 0.27 | 0.09 | 1.24 | 0.28 | 819 | 0.49 | 0.23 | 0.32 | 0.13 | 5.82 | 0.40 | 4.85 | 0.40 | 1.28 | 0.27 |

RLOF* (RLOF without adaptive region size, see Section IV) shows similar accuracy compared to the Talwar, Tukey, Cauchy, and Welsh robust norms [22] but a relatively short runtime. RLOF# denotes the RLOF without using the modified estimator. R0.5 denotes the ratio of pixels that have an endpoint error above 0.5 and illustrates the outlier resulting from each method.

While the result of Fig. 4(b) corresponds to the multiple motion experiments of Black and Anandan, Fig. 3(b) shows different characteristics. The minimum is nearer to the second motion of the region border than to the motion of the region center, while a second minimum is not visible. This behavior can be observed at motion boundaries where at least one object is homogeneous. The data at homogeneous areas has no or only a small impact on the result of the estimates of the linear Lucas–Kanade model because the derivative $\partial\epsilon_i/\partial\theta$ at homogeneous regions is zero and shrinks its influence. In consequence, the improvement of distinguishing different motions in a region by robust estimation depends on the ratio of the textures introduced by each object.

IV. ROBUST LOCAL OPTICAL FLOW

The outcome of our experiments leads us to propose a variation of the Lucas–Kanade method using the shrinked Hampel norm (14): the RLOF method. The inverse compo-

sitional RLOF residual error is formulated as

$$E_{\text{RLOF}} = \sum_{\Omega_1 \subset \Omega} \epsilon^2 + \sum_{\Omega_3 \subset \Omega} \sigma_1 \sigma_2 + \sum_{\Omega_2 \subset \Omega} \left(\frac{\sigma_1}{\sigma_1 - \sigma_2} (|\epsilon| - \sigma_2)^2 + \sigma_1 \sigma_2 \right) \quad (16)$$

with

$$\epsilon = \nabla I(\mathbf{x})^T \cdot \mathbf{d} + I_t(\mathbf{x}) \quad (17)$$

and Ω_1 as the subset of data in Ω fulfilling $|\epsilon_i| \leq \sigma_1$, Ω_2 denoting the subset fulfilling $\sigma_1 < |\epsilon_i| < \sigma_2$, and Ω_3 for which holds $|\epsilon_i| \geq \sigma_2$. As stated in Section III-A, (7) can be solved as

$$\Delta \mathbf{d}^i = \mathbf{G}_{\text{RLOF}}^{-1} \cdot \left[\sum_{\Omega_1 \subset \Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}) + \sum_{\Omega_2 \subset \Omega} \frac{\sigma_1}{\sigma_1 - \sigma_2} \cdot \nabla I(\mathbf{x}) \cdot (I_t^{i-1}(\mathbf{x}) - \text{sign}(I_t^{i-1}(\mathbf{x})) \cdot \sigma_2) \right] \quad (18)$$

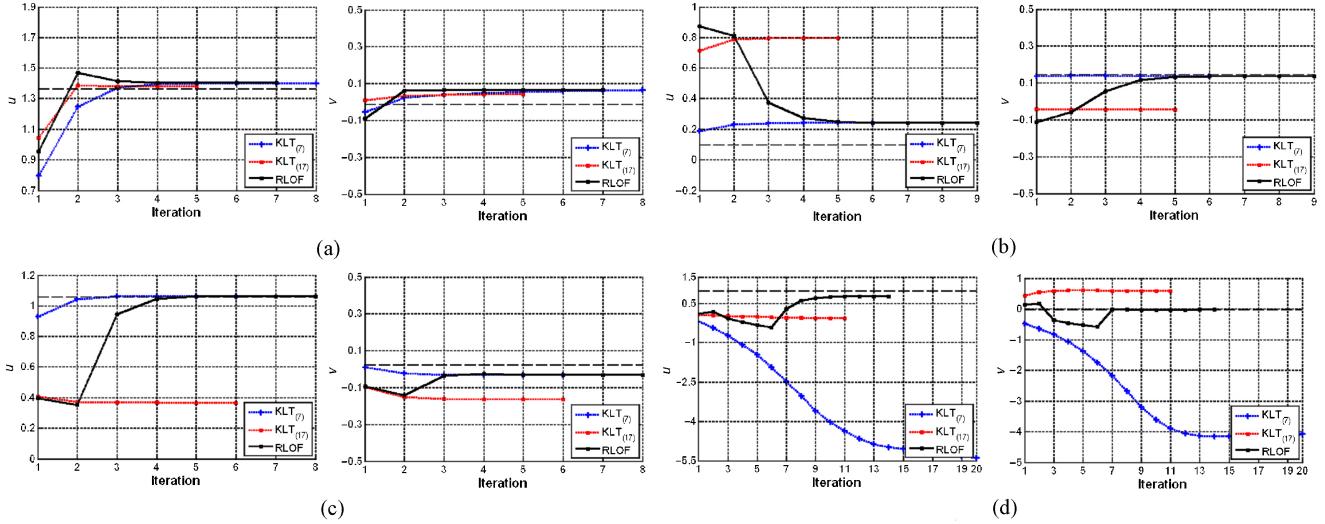


Fig. 5. Example of the RLOF and KLT algorithm with different region sizes converging at the four case studies (Section II-B). The resulting motion (u, v) is plotted against the iteration number. Black dashed line: ground truth. Termination criteria are set with 20 maximal iterations and a minimal change of $\Delta\mathbf{d}$ by 0.001. (a) Not violated. (b) Motion boundaries. (c) Changing illuminations. (d) Appearing pixels.

where \mathbf{G}_{RLOF} is the modified Hessian matrix

$$\begin{aligned} \mathbf{G}_{\text{RLOF}} = & \sum_{\Omega_1 \subset \Omega} \nabla I(\mathbf{x}) \cdot \nabla I(\mathbf{x})^T \\ & + \sum_{\Omega_2 \subset \Omega} \frac{\sigma_1}{\sigma_1 - \sigma_2} \nabla I(\mathbf{x}) \cdot \nabla I(\mathbf{x})^T. \end{aligned} \quad (19)$$

As stated in [35], it is important that the influence function ψ of the norm does not descend too steeply as long as the value of ψ is still high. We incorporate this by using the L^2 norm as a monotone ψ for the first iteration. This can easily be done by setting $\sigma_1 = \infty$ and $\sigma_2 = \infty$. For the following iterations, we append cycles with the nonmonotone ψ , since the corresponding shrunk Hampel norm is nonconvex and the determination of the minimum may be trapped in local minima far away from the true minimum.

A. Generalized Aperture Problem

As shown in Fig. 4(a) and (b) and by the experiments done in [29], redescending norms are able to cope with perturbations affecting the local constancy and brightness constancy assumption. An essential requirement therefore is that the observed underlying moving object owns the dominant textures in the observed domain. For example, this is not fulfilled in Fig. 3(b). The motion boundaries example contains two moving objects: the wooden fence in the foreground containing the dominant textures and the background. While the center of the observed region lies in the background, we are interested in computing its motion. The observations of the background x_{ij} (which correspond to its spatial and temporal derivatives) are less correlated than the observations of the foreground fence. Thus, the estimation of the parameter set θ is determined by the fence motion, while the background motion is treated as outlier. This example shows that the observation done in Fig. 4(a) and (b) and [29] cannot always be valid. To cope with the violation of the local constancy assumption, the observed region has to be as small as possible with regard to the generalized aperture problem to increase the probability to contain no motion

boundary. An additional reason to set the observed region Ω as small as possible is caused by the computational effort.

Therefore, we propose a strategy to adapt the region size depending on the residual error E_{RLOF} and the contained texture. At first, \mathbf{d} is computed for a few number of iterations i_l applying a large region Ω_{large} . This results in an overall coarse solution, containing most likely all important texture. To achieve a better performance at motion boundaries, the iteration is applied to a drastically shrunk region Ω_{small} after one cycle. To avoid the aperture problem, the minimal eigenvalue of the matrix \mathbf{G}_{RLOF} is computed to decide if the feature is trackable [13], [31]. Another criterion to decide if the feature could be tracked with the small region is taken by comparing the normalized residual error E_{RLOF} of the current and the large region. In the positive case, the iteration is continued until reaching final convergence or a maximal number of iterations, otherwise the region size is increased step by step by the default value two until a trackable size or Ω_{large} is reached. In this field, there are still a lot of research opportunities, but topics such as variable window shapes or an exact investigation on the influence of the size step are far beyond the focus of this paper.

B. Empirical Validation

We experiment with the four case studies of Section II-B comparing the convergence of RLOF with the original KLT applied to the small and the large region size used in RLOF. The iterative solution for each of the four displacements is shown in Fig. 5. The termination criteria are set to a maximal number of 20 iterations and a minimal change of $\Delta\mathbf{d}$ by 0.001. The black dashed lines display ground truth motion at the respective positions. The results confirm our expectation that a small region favors a higher resolved motion field, in particular at motion boundaries [see Fig. 5(b) and (c)]. A large region converges on average with fewer iterations. It achieves a low resolved motion (in our case, it is not able to separate the motions). We observe in Fig. 5(d) that the small region

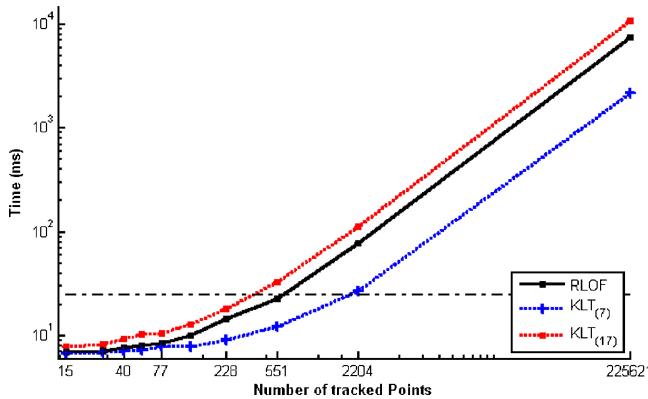


Fig. 6. Comparison of execution times: KLT and RLOF method with varying number of tracked points. Black dashed line: 25 f/s threshold.

KLT fails to converge with the appearing pixels, while a large region converges to an inaccurate solution. By adapting the region size and applying a robust estimator, the capability of the RLOF to separate different motions and to neglect outliers introduced by appearing pixels is increased. We observe that RLOF can increase the accuracy and stability of current KLT methods that is paid for by a higher number of iterations than the KLT with the large region.

C. Computational Complexity

With n as the number of computed motion vectors and N being the number of pixels of the region Ω , the computational cost of the Lucas–Kanade method for one iterative step is given by: 1) computing the warped spatial derivative ∇I ($O(nN)$); 2) computing the gradient matrix \mathbf{G} ($O(nN)$); and 3) computing the incremental solution ($O(nN)$). (6). Baker and Matthews proved in [23] that the inverse compositional algorithm also used in [28] is the same as the Lucas–Kanade algorithm but far more efficient because the spatial derivatives and the gradient matrix only have to be computed initially. The computational cost of this variation, used by the KLT method, is given by $O(nNi)$. Thus, for a fast tracking method, on the one hand, the convergence must be fast while the region size should be kept small. As N is quadratic to the region size Ω in most cases, a small region size is preferred. However, Fig. 5 shows how a small region tends to converge more slowly to the correct solution as the region is more likely to contain less gradient information (see Fig. 1 for the respective regions).

Within the robust norm, the gradient matrix (19) has to be computed for each iterative step by revalidating each pixel N . Thus, the computational complexity $O(inN)$ is increased by recalculating \mathbf{G}_{RLOF} ($O(nNi)$). Through the varying region size, the computational complexity of the RLOF is bounded by $O(nN_{\text{small}}i)$ and $O(nN_{\text{large}}i)$. Fig. 6 shows the runtime of the KLT and RLOF method related to the number of tracked points. The implementation is tested on an AMD Phenom II X4 960 running at 2.99 GHz and without multithreading.

As the computation of each motion vector is independent, the runtime can be decreased by parallelization (using, e.g., OpenMP). Generally, due to the changing window size, the RLOF converges more slowly than the KLT using one large region (see Fig. 5). This is due to the smaller amount of gradi-

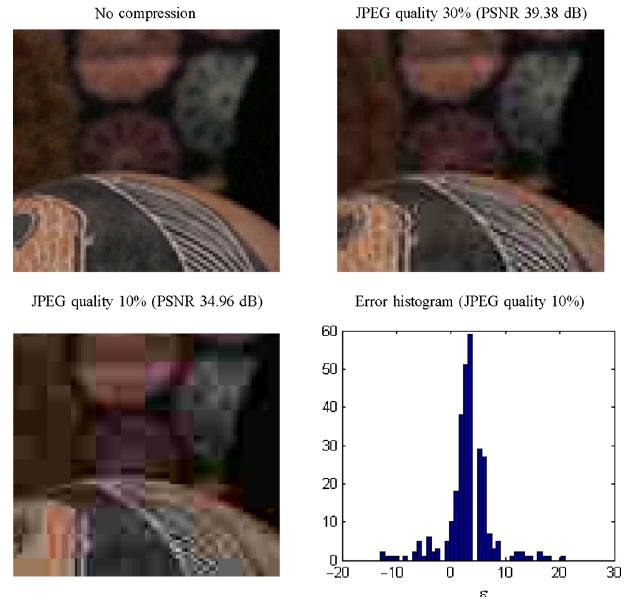


Fig. 7. Robustness test is performed with different JPEG compressions. This kind of error is not Gaussian distributed.

ent information contained in small windows. However, thanks to the adaptive region size, the RLOF is less time consuming. So the adaptive region size is not only advantageous for accuracy at motion boundaries but also results in a decreased computational cost compared to KLT with large region size. A general advantage of local optical flow methods is shown in Fig. 6 with the linear scalability due to the number of tracked points n . In our experiment, we vary the number of motion vectors by applying a grid with different sizes to find the points to track. The RLOF processes 551 features in 44.38 f/s using the CPU on an image of the size 584×388 . Thus, it needs only 70% of the runtime of KLT using a large region. By using not more than a few sets of points to track, the runtime depends mainly on the initial calculation of the image gradients and is thus in this experiment bounded by 6.7 ms.

V. EXPERIMENTS ON VIDEO SEQUENCES

In the following section, we show the results of our experiments for long-term trajectories. We compare the RLOF to the KLT and the state-of-the-art dense trajectory methods LDOF [11] and particle video [10]. To deal with the large motions, the RLOF and KLT are implemented in a pyramidal manner [24], [28]. For our experiments, we apply four levels. The small and large region sizes are set to 7×7 and 17×17 and the maximal number of iterations is 20. RLOF works with a robust norm specified by $\sigma_1 = 5$ and $\sigma_2 = 50$. The KLT experiments are performed with all valid region sizes between 7×7 and 17×17 .

A. Evaluation Methodology

We evaluate the performance of the tracking methods with the MIT dataset [25]. In contrast to the Middlebury dataset, this dataset provides the ground truth optical flow for whole sequences of up to 75 frames. We compare the accuracy of the

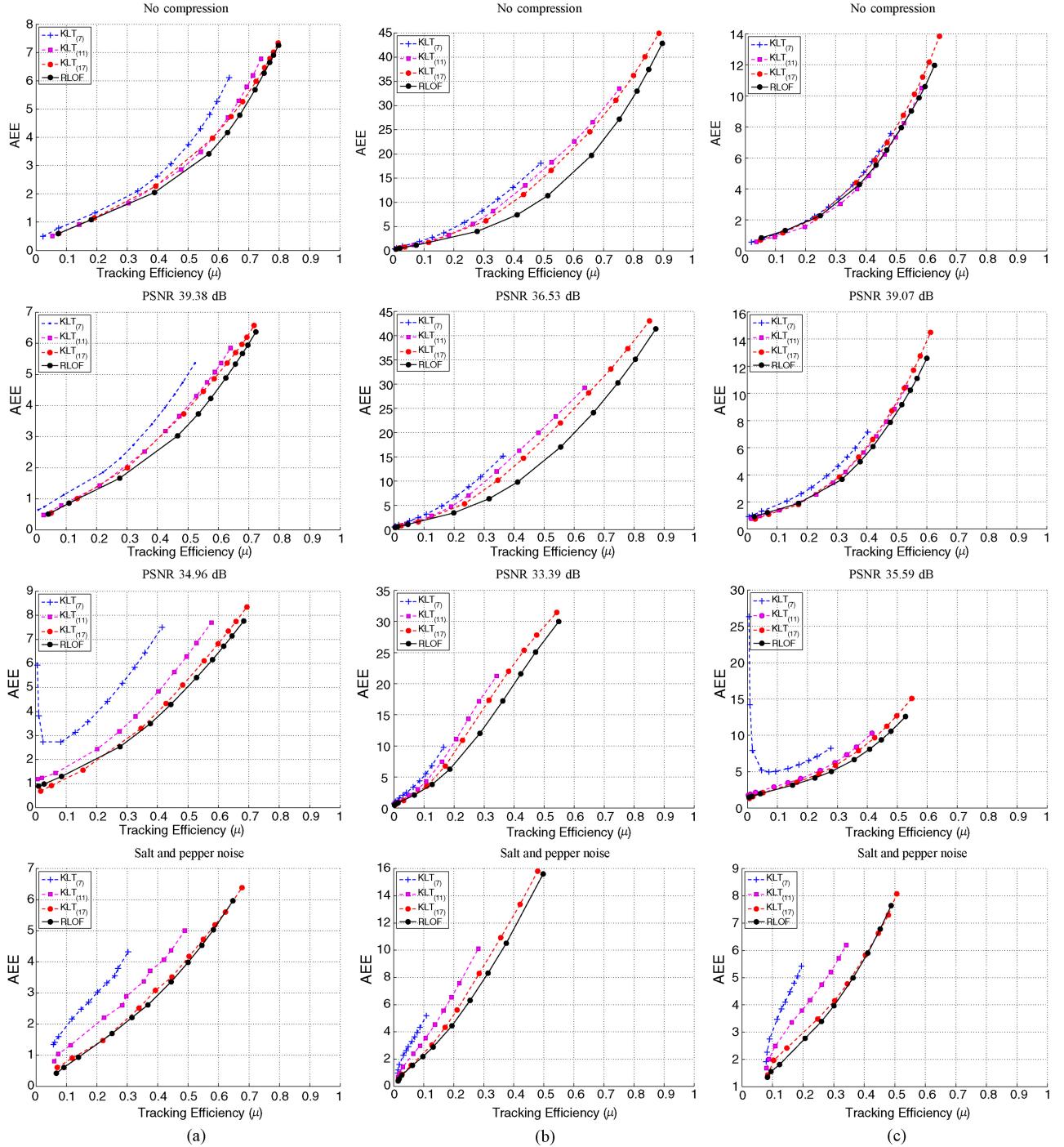


Fig. 8. Evaluation of the tracking performance for the (a) *Toy*, (b) *Fish*, and (c) *Table* sequence with different compression rates of the MIT [25] dataset and salt and pepper noise with a noise density of 0.02. The region size of the KLT varied from 7×7 to 17×17 , which is the range of the window sizes for RLOF. 7×7 , 11×11 , and 17×17 are shown.

trackers for the entire length of the sequences by comparing trajectories. A trajectory is created for each pixel of the first image and tracked through all the frames. Ground truth trajectories are obtained from the ground truth optical flow. To compute subpixel accurate tracks, a bilinear interpolation is applied. A trajectory is stopped as soon as a point gets occluded or is incorrectly tracked. We detect these for the KLT and RLOF method by checking the consistency of the forward and backward flow. The consistency check is done by thresholding the forward displacement from image $I(t) \rightarrow I(t+1)$

and the respective warped reverse one $I(t+1) \rightarrow I(t)$ as follows:

$$\|\mathbf{d}_{I(t) \rightarrow I(t+1)} + \mathbf{d}_{I(t+1) \rightarrow I(t)}\| < \epsilon_d. \quad (20)$$

The measurement is based on two criteria: 1) the average endpoint error (AEE) between the set of trajectories T and the set of ground truth trajectories T_{GT}

$$AEE = \frac{1}{|T|} \sum_{|T|} \|\dot{x}_T - \dot{x}_{T_{GT}}\|^2 \quad (21)$$

with \dot{x} as the endpoint of each trajectory, and 2) the tracking efficiency (η)

$$\eta = \frac{1}{|T_{\text{GT}}| \cdot L} \sum_{|T|} l_T. \quad (22)$$

In contrast to [24], this measure is defined as the quotient of the average of l_T , the length of all successfully tracked features, and L , the total number of frames of each sequence. The combination of these two criteria is plotted as tracking performance by varying ϵ_d (see Fig. 8). Note that the AEE is accumulated for all trajectories T and not only for the trajectories, which are tracked until the last images successfully.

B. Results

In real-life scenarios, e.g., video surveillance, videos are provided in a compressed way, e.g., H.264, MPEG4, or motion JPEG. Thus, the video is affected by compression artifacts. In the statistical point of view, these artifacts are a source of noise, which is not Gaussian distributed (see Fig. 7). This is why we are in addition interested to compare the tracking methods related to different compressions. The results for the *Toy*, *Fish*, and *Table* sequence are shown in Fig. 8.

In the *Toy* sequence (18 frames), the camera is moving transversally in front of a set of teddy bears. Large areas behind the bear (left) are uncovered while computation of optical flow is hampered by the very homogeneous areas of the black and white panda. The *Table* sequence (13 frames) shows a circular motion around a table with different objects on top occluding each other. Due to depth of field, the background is very blurred and contains little texture information that makes it a hard area for estimating optical flow. The *Fish* sequence (75 frames) is the longest of the sequences. The backgrounds are blurry while the video also suffers from strong noise and low brightness. The motion estimation is further aggravated by the transparent motion that covers the whole sequence caused by dust particles moving in the water.

As described above, the tracking efficiency η is a value between 0 and 1 that describes the percentage of the trajectories the algorithm is able to maintain in relation to their length. Tracking all trajectories over all frames would correspond to $\eta = 1$ while tracking only half of the pixels for the full length (or all pixels for half the number of frames) corresponds to $\eta = 0.5$.

Fig. 8 shows the tracking efficiency compared to the AEE. Regardless of the compression level and the salt and pepper noise with a density of 0.02, by these examples it can be seen that RLOF usually achieves better results than the KLT variants. While the proposed CPU implementation of the RLOF achieves a frame rate of 44 f/s by tracking 551 features, we provided in [24] a parallelized version of the RLOF, which is able to compute up to 10.000 feature points on high-definition content in real time (>25 f/s). The implementation was done using OpenCL and runs on a NVidia GTX 275 GPU.

The region size used in RLOF is varied between 7×7 and 17×17 that are the minimal and maximal region sizes of the KLT algorithm used in these experiments. We conduct thus from the higher performance that varying the region size is a favorable extension of the standard KLT. The performance of all algorithms decreases with higher compression

TABLE II
EVALUATION OF TRACKING PERFORMANCE FOR THE LDOF AND
PARTICLE VIDEO METHODS OVER MIT SEQUENCES

| | LDOF | | | | | |
|------------------|-------|--------|--------|--------|-------|--------|
| | Toy | | Fish | | Table | |
| | AEE | η | AEE | η | AEE | η |
| JPEG quality 30% | 3.10 | 0.87 | 19.18 | 0.37 | 3.85 | 0.65 |
| JPEG quality 10% | 5.78 | 0.65 | 21.85 | 0.38 | 5.08 | 0.49 |
| Salt and pepper | 3.32 | 0.63 | 27.40 | 0.26 | 3.51 | 0.45 |
| Particle Video | | | | | | |
| | Toy | | Fish | | Table | |
| | AEE | η | AEE | η | AEE | η |
| | 13.07 | 0.84 | 68.45 | 0.69 | 24.09 | 0.75 |
| JPEG quality 30% | 10.21 | 0.83 | 49.35 | 0.63 | 31.31 | 0.71 |
| JPEG quality 10% | 34.70 | 0.76 | 102.86 | 0.58 | 52.01 | 0.69 |
| Salt and pepper | | | | | | |

rates, which shows that all methods suffer from the loss of information during JPEG compression. Especially, smaller window sizes have disadvantages with increasing compression rate as they suffer more from compression noise. Still, by varying different region sizes, RLOF is able to achieve better results than standard KLT trackers. The evaluation of the dense LDOF (available at <http://lmb.informatik.uni-freiburg.de/resources/binaries>) and the particle video tracker (available at <http://rvsn.csail.mit.edu/pv>) are shown in Table II using the default parameter set given by the authors. While the LDOF outperforms the RLOF and KLT in terms of accuracy except in the *Fish* sequence caused by its transparent motion, the RLOF still achieves a good accuracy compared to the particle video method. We observed that the optimization step of the particle video method, which often improves this method, fails at areas containing little textured information, e.g., the table in the *Table* sequence. The runtime of the dense optical flow methods was measured for the *Toy* sequence of size 972×723 . The runtime is almost constant and independent of the number of tracked points, which is why we prefer local optical flow methods in time-critical applications. The CPU implementation of the LDOF requires about 139 s and the particle video requires about 135 s per frame. While we evaluate a dense trajectory set, i.e., 702 756 points to track at the first image, the RLOF also requires 173 s.

VI. CONCLUSION

In this paper, we illustrated the benefit of a robust framework for feature tracking via local optical flow. Motivated by an extensive analysis of data distributions, we proposed the RLOF approach based on a modified Hampel estimator with robust characteristics. To cope with the generalized aperture problem, a strategy to adapt the region size was developed. The effectiveness of our approach was shown under various scenarios as motion boundaries, changing illuminations, and appearing pixels, all violating standard Lucas–Kanade assumptions. In our experiments, it could be shown that a robust estimator gives better results. Evaluations on the MIT database showed an excellent long-term feature tracking performance of RLOF with only slightly increased computational complexity compared to KLT.

REFERENCES

- [1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *Int. J. Comput. Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [2] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

- [3] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 674–679.
- [4] M. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *Proc. IEEE ICCV*, May 1993, pp. 231–236.
- [5] M. J. Black and P. Anandan, "The robust estimation of multiple motion: Parametric and piecewise-smooth flow fields," *Comput. Vision Image Understand.*, vol. 63, no. 1, pp. 75–104, 1996.
- [6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. ECCV*, 2004, pp. 25–36.
- [7] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic Huber-L1 optical flow," in *Proc. BMVC*, 2009, pp. 231–236.
- [8] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *Int. J. Comput. Vision*, vol. 67, no. 2, pp. 141–158, 2006.
- [9] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Proc. DAGM Symp.*, 2007, pp. 214–223.
- [10] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *Int. J. Comput. Vision*, vol. 80, no. 1, pp. 72–91, 2008.
- [11] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpuaccelerated large displacement optical flow," in *Proc. ECCV*, 2010, pp. 438–451.
- [12] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, Jun. 2009, pp. 41–48.
- [13] C. Tomasi and T. Kanade, "Detection and tracking of point features," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, 1991.
- [14] H. Liu, T.-H. Hong, M. Herman, T. Camus, and R. Chellappa, "Accuracy vs efficiency trade-offs in optical flow algorithms," *Comput. Vision Image Understand.*, vol. 72, no. 3, pp. 271–286, 1998.
- [15] T. Senst, V. Eiselein, and T. Sikora, "II-LK: A real-time implementation for sparse optical flow," in *Proc. ICIAR*, 2010, pp. 240–249.
- [16] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "GPU-based video feature tracking and matching," Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Chapel Hill, Tech. Rep. 06-012, 2006.
- [17] C. Zach, D. Gallup, and J. Frahm, "Fast gain-adaptive KLT tracking on the GPU," in *Proc. Vis. CVGPU Workshop*, 2008, pp. 1–7.
- [18] H. Fassold, J. Rosner, P. P. Schallauer, and W. Bailer, "Realtime KLT feature point tracking for high definition video," in *Proc. Comput. Graphics, Comput. Vision Math. (GravisMa)*, 2009.
- [19] M. Kharbat, N. Aouf, A. Tsourdos, and B. White, "Robust brightness description for computing optical flow," in *Proc. BMVC*, 2008, pp. 46.1–46.10.
- [20] Y. Kim, A. Martinez, and A. Kak, "A local approach for robust optical flow estimation under varying illumination," in *Proc. BMVC*, 2004, pp. 91.1–91.10.
- [21] M. Gelgon, P. Bouthemy, and T. Dubois, "A region tracking method with failure detection for an interactive video indexing environment," in *Proc. 3rd Int. Conf. Vis. Inform. Inform. Syst.*, 1999, pp. 261–268.
- [22] J. Odobezi and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Vis. Communun. Image Represent.*, vol. 6, no. 4, pp. 348–365, Dec. 1995.
- [23] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, Feb. 2004.
- [24] T. Senst, V. Eiselein, R. Heras Evangelio, and T. Sikora, "Robust modified L2 local optical flow estimation and feature tracking," in *Proc. IEEE WMVC*, Jan. 2011, pp. 685–690.
- [25] C. Liu, W. Freeman, E. Adelson, and Y. Weiss, "Human-assisted motion annotation," in *Proc. CVPR*, 2008, pp. 1–8.
- [26] P. J. Huber, "Robust regression: Asymptotics, conjectures and Monte Carlo," *Ann. Statist.*, vol. 1, no. 5, pp. 799–821, 1973.
- [27] P. J. Rousseeuw, "Robuste regression mit ausreisern in den erklärenden variablen," *Österreichische Zeitschrift für Statistik und Informatik (ZSI)*, vol. 15, no. 1, pp. 36–42, 1986.
- [28] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," Microprocessor Res. Lab., Intel Corp., Hillsboro, OR, 2000 [Online]. Available: http://robots.stanford.edu/cs223b04/algo_tracking.pdf
- [29] A. Jepson and M. Black, "Mixture models for optical flow computation," in *Proc. CVPR*, 1993, pp. 760–761.
- [30] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," Microsoft Res., Redmond, WA, Tech. Rep. MSR-TR-2009-179, Dec. 2009.
- [31] J. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR*, 1994, pp. 593–600.
- [32] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.
- [33] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions* (Wiley Series in Probability and Statistics). New York: Wiley-Interscience, 2005.
- [34] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, Mar. 1998.
- [35] P. J. Huber, "Robust methods of estimation of regression coefficients," *Statistics*, vol. 8, no. 1, pp. 41–53, 1977.
- [36] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Statist.: Theory Methods*, vol. 6, no. 9, pp. 813–828, 1977.
- [37] R. Dutter, "Numerical solution of robust regression problems: computational aspects, a comparison," *J. Statist. Computat. Simulat.*, vol. 5, no. 3, pp. 207–238, 1977.
- [38] G. Shevlyakov, J. W. Lee, K. M. Lee, V. Shin, and K. Kim, "Robust detection of a weak signal with redescending m-estimators: A comparative study," *Int. J. Adap. Contr. Signal Process.*, vol. 24, no. 1, pp. 33–40, 2010.



Tobias Senst (S'11) received the Dipl.-Ing. degree in computer engineering from Otto von Guericke Universität Magdeburg, Magdeburg, Germany. He is currently pursuing the Ph.D. degree with the Communication Systems Group, Technische Universität Berlin, Berlin, Germany.

His current research interests include image and video processing, optical flow, feature tracking, and video surveillance.



Volker Eiselein (S'11) received the Dipl.-Ing. degree in computer engineering from Technische Universität Berlin, Berlin, Germany, where he is currently pursuing the Ph.D. degree with the Communication Systems Group.

His current research interests include image and video processing, video surveillance, object tracking, and optical flow.



Thomas Sikora (SM'97) 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 Director of the Communication Systems Group, 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. From 1994 to 2001, he was the Director of the Department of Interactive Media, Heinrich Hertz Institute, Berlin GmbH, Germany.

Prof. Sikora was a recipient of the German Society for Information Technology (ITG) Award in 1996. He was the Co-Founder of 2SK Media Technologies and Vis-a-Pix GmbH, two Berlin-based start-up companies involved in research and development of audio and video signal processing and compression technology. He has been involved in international ITU and ISO standardization activities as well as in several European research activities for a number of 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 VISNET and 3DTV European Networks of Excellence. He is an Advisory and Supervisory Board Member of a number of German companies and international research organizations. He is an Industry Consultant on issues related to interactive digital audio and video. He is a member of the ITG. He has published more than 150 papers related to audio and video processing. 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 EURASIP Signal Processing: Image Communication. He was an Associate Editor of EURASIP Signal Processing.