

# Robust Modified $L^2$ Local Optical Flow Estimation and Feature Tracking

Tobias Senst, Volker Eiselein, Rubén Heras Evangelio and Thomas Sikora  
Communication Systems Group, Technische Universität Berlin  
Einsteinufer 17, 10587 Berlin, Germany  
senst,eiselein,heras,sikora@nue.tu-berlin.de

## Abstract

*This paper describes a robust method for the local optical flow estimation and the KLT feature tracking performed on the GPU. Therefore we present an estimator based on the  $L^2$  norm with robust characteristics. In order to increase the robustness at discontinuities we propose a strategy to adapt the used region size. The GPU implementation of our approach achieves real-time (>25fps) performance for High Definition (HD) video sequences while tracking several thousands of points. The benefit of the suggested enhancement is illustrated on the Middlebury optical flow benchmark.*

## 1. Introduction

The computation of 2D image velocity, or optical flow, is a common topic in computer vision. A primary goal in the field is to estimate the scene or object motion as precisely as possible. Most recent 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*:

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

with  $I(x, y, t)$  the image intensity of a grayscale image,  $\mathbf{d} = (u, v)^T$  denoting the displacement of a point and  $\delta t$  a small time difference at a position  $\mathbf{x} = (x, y)$ . The most successful methods to compute  $\mathbf{d}$  use a linearisation of Eq. 1 performed by a first order Taylor-approximation and are therefore gradient-based. This leads to an under-determined linear system. To solve this system, two kinds of spatial coherence have been introduced. By introducing an additional global constraint Horn and Schunck [13] apply a soft spatial coherence forcing the partial derivatives of neighbouring motion vectors to be minimal. A strong spatial coherence was introduced by Lucas and Kanade [11]

that is categorised as local constraint expecting the motion in a small region to be constant. Both techniques minimize the cost function by using the mean square error.

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 [1] the violations result in gross measurement errors which are referred to as outliers. Since Horn/Schunck and Lucas/Kanade penalize the minimization in a quadratic way the model does not handle outliers robustly. Black and Anandan [2] proposed a robust estimation framework exploiting the Lorentzian robust norm.

So far 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.* [5] or the Huber- $L^1$  norm used by Werlberger *et al.* [20]. An additional benefit can be achieved by combining this with more sophisticated Total-Variation techniques as in e.g. Papenberg *et al.* [12] and Zach *et al.* [22]. Generally global optical flow methods achieve a superior accuracy compared to local optical flow methods.

Yet, applications as robot navigation, augmented reality, visual attention and camera self-calibration require very fast detection of interest points (feature points) and the subsequent search for potential correspondences in real-time. Therefore, methods with excellent runtime performance exploiting local optical flow techniques as the popular KLT tracking algorithm [19] in many cases are still applied. For that reason research of local methods is often motivated by improving the run-time performance. E.g. Senst *et al.* [14] propose integral images to decrease the computational complexity per interest point. Sinha *et al.* [18], Zach *et al.* [21] and Fassold *et al.* [6] improve the run-time performance by parallelising the algorithm and porting it onto a GPU. While in many global optical flow methods robust estimation techniques are established, most local methods are currently based on least-square optimisation. Gain adaptive modifications were proposed by Zach *et al.* [21] and Kharbat *et*

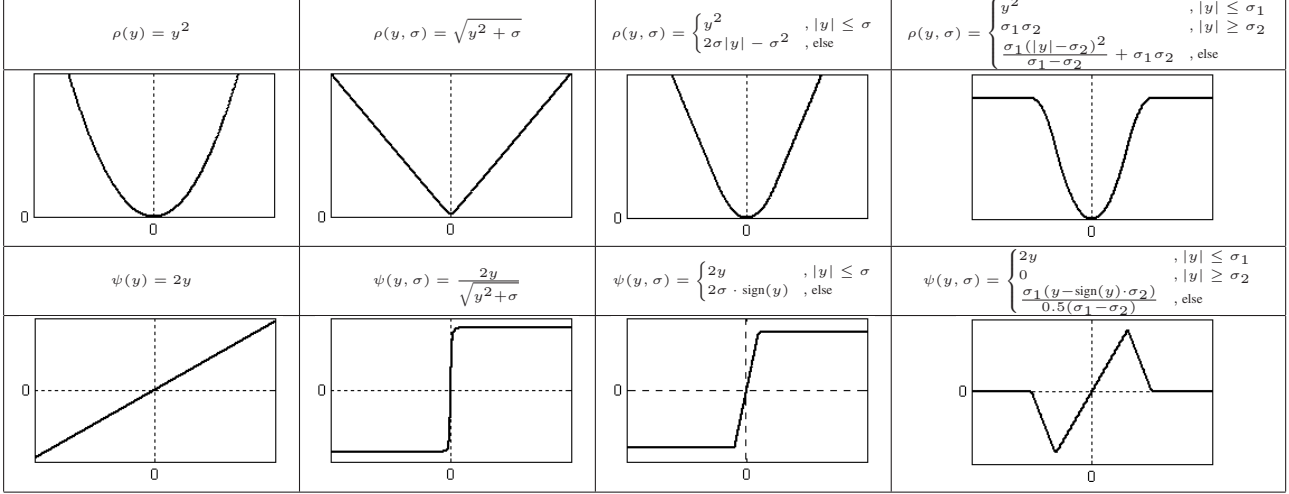


Figure 1. Common estimators, from left to right,  $L^2$ , modified  $L^1$ , Huber's minmax estimator and the proposed modified  $L^2$ .

al. [9]. Kim *et al.* [10] proposes a varying illumination robust approach using a least-median of squares method which is robust but increases the runtime drastically.

## 2. Robust Local Estimation

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

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

To find a displacement  $\mathbf{d}$ , the *residual error*  $E_{local}$  is minimised for a small image region  $\Omega$ , with the spatial derivatives  $\nabla I = (I_x, I_y)^T$  and the temporal derivative  $I_t$ ,  $w(\mathbf{x})$  a weighting function and a norm  $\rho$ , with its scale parameters  $\sigma$ .

The influence of different weighting functions was evaluated by Singh *et al.* [17], studying 2D-Gaussian and Laplace of Gaussian (LOG). In this paper we want to focus on the study of a robust estimator  $\rho$ . If the errors of the measurement were normally distributed, the optimal estimator would be the quadratic  $\rho(\mathbf{y}) = \mathbf{y}^2$  leading to a least-squares solution. Combined with a rectangular weighting function ( $w(\mathbf{x}) \in \Omega = 1$ , else  $0$ ), Eq. 2 formulates the residual error established by Lucas and Kanade [11] and lays the foundation for the popular KLT tracking algorithm [19]. To solve the least-square minimization, the derivative of the residual error is set to zero so that the displacement is given by:

$$\mathbf{d} = \mathbf{G}^{-1} \cdot \mathbf{b} \\ \mathbf{G} = \sum_{\Omega} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \mathbf{b} = \sum_{\Omega} \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \quad (3)$$

To get an accurate solution, Bouguet [4] introduces a pyramidal iterative implementation solving Eq. 3 in a

Newton-Raphson fashion, which is comparable to simultaneous overrelaxation techniques used by Black and Anandan [2]. In general the Newton iteration converges fast but only if the solved function is convex around zero. The computational complexity of the Lucas/ Kanade algorithm for  $n$  feature points,  $k$  iterations and a region  $\Omega$  of the size  $\Omega_N \times \Omega_N$  is  $\mathcal{O}(k \cdot n \cdot \Omega_N^2)$ .

The drawback of the least-squares solution is that outliers are assigned a high weight by the quadratic estimator, as shown in figure 1. The influence function  $\psi(\mathbf{y}, \sigma)$  as the derivative of the estimator  $\rho(\mathbf{y}, \sigma)$  characterizes the bias that a particular measurement has on the solution [7]. Indeed, the quadratic model describes the optimal estimator for gaussian noise. More robust norms are the modified  $L^1$  [12, 22] norm and Huber's minmax norm [3, 20], 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 [3]. In contrast, Huber's minmax norm is designed to be quadratic for small values, which makes it applicable for Newton. However the influence function of both norms does not descend all the way to zero - consequently very strong outliers still affect the solution.

**Proposed modification of the  $L^2$  norm:** A possible norm with an influence function tending to zero for large values is Tukey's biweight. But because we want to keep the computational effort low and thereby make the algorithm as fast as possible, we decided to use a norm which is composed by quadratic functions. Therefore we propose a modification of the  $L^2$  norm to estimate the robust local optical flow which is quadratic for small values and has an influence function being zero for strong outliers. The proposed estimator is modified by two scale parameters  $\sigma_1$  and  $\sigma_2$ . If set  $\sigma_2 = \sigma_1$  the modified  $L^2$  is equivalent to the

truncated quadratic, while setting  $\sigma_2 \rightarrow \infty$  the norm becomes equivalent to Huber’s minmax. The modification of the Eq. 2 by the modified  $L^2$  estimator is formulated as:

$$E_{local} = \sum_{\Omega_1} (\nabla I \cdot d + I_t)^2 + \sum_{\Omega_3} \sigma_1 \sigma_2 + \sum_{\Omega_2} \left( \frac{\sigma_1}{\sigma_1 - \sigma_2} (|\nabla I \cdot d + I_t| - \sigma_2)^2 + \sigma_1 \sigma_2 \right) \quad (4)$$

with  $\Omega_1$  denoting the set of data measurements fulfilling  $|y| \leq \sigma_1$ ,  $\Omega_2$  denoting the set fulfilling  $\sigma_1 < |y| < \sigma_2$  and  $\Omega_3$  the set which fulfils  $|y| \geq \sigma_2$ . We can derive the residual function and compute the displacement vector by:

$$G = \sum_{\Omega_1} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} + \sum_{\Omega_2} \begin{bmatrix} \frac{\sigma_1}{\sigma_1 - \sigma_2} I_x^2 & \frac{\sigma_1}{\sigma_1 - \sigma_2} I_x I_y \\ \frac{\sigma_1}{\sigma_1 - \sigma_2} I_x I_y & \frac{\sigma_1}{\sigma_1 - \sigma_2} I_y^2 \end{bmatrix} \quad (5)$$

and

$$b = \sum_{\Omega_1} \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} + \sum_{\Omega_2} \begin{bmatrix} \frac{\sigma_1 I_x (I_t - \text{sign}(I_t) \sigma_2)}{\sigma_1 - \sigma_2} \\ \frac{\sigma_1 I_y (I_t - \text{sign}(I_t) \sigma_2)}{\sigma_1 - \sigma_2} \end{bmatrix} \quad (6)$$

To get an accurate solution the iterative scheme and pyramidal coarse-to-fine strategy proposed Bouguet [4] is used. In addition we applied an overrelaxation parameter  $\omega$  that overcorrects the result  $d_k$  at iteration  $k$ .

Figure 2 shows plots of the residual error surfaces  $E_{local}$  for different estimators. The surfaces are rendered for a region of size  $15 \times 15$  at the position  $(340, 200)$  of the *Grove2* sequence taken from the Middlebury dataset. This region is affected by two motions. Figure 2 (bottom right) shows the region lying at a leaf moving to the bottom left while some parts of the region belong to the background moving to the left  $\hat{d}_{BG} = (-2.62, 0.13)$ . The respective minimal displacement vectors  $\hat{d}$  are marked with a dot and are computed by varying  $d$  in Eq. 4.

Figure 2 illustrates the resulting minima of the  $L^2$  surface is inaccurate but smooth. The error is caused by the partial motions and the corrupted intensity constancy assumption of leaf- and background pixels who appear and disappear. More precise results are computed with the Huber’s minmax (middle left) and the modified  $L^1$  norm (bottom left). The most accurate result is performed with the truncated  $L^2$  (middle right) and the modified  $L^2$  (top right) caused by their influence function descending to zero for large values. The truncated  $L^2$  gives the best guess for the displacement but the error surface is more irregular which has a destabilising affect to the Newton iteration. Finally the error surface of the modified  $L^2$  illustrates the improvement of the proposed norm which combines accuracy and smoothness.

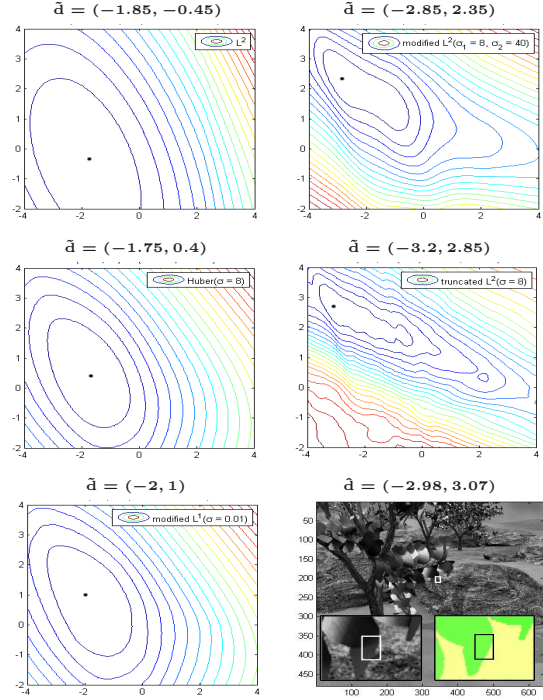


Figure 2. Error surface of different estimators. Groundtruth is denoted by  $\hat{d}$  and the respective minima  $\hat{d}$  is marked as a dot.

### 3. Generalized Aperture Problem

The local optical flow formulation of Eq. 2 is affected by a common problem which is the choice of the region size  $\Omega$ . While a large region is needed to constrain the solution and provide insensitivity to noise (see aperture problem), a large region increases the risk of violating the assumption that a region represents only one motion. That dilemma is referred to as generalized aperture problem [8].

Using a robust norm does not solve this problem generally. Yet, the experiments done by Black and Anandan [2] show promising results. However they used two moving synthetic random noise patterns. Their experiment shows that for a robust norm the resulting motion depends on the motion of the pattern dominating. Data measurements caused by the second moving pattern were recognised as outliers and suppressed successfully. Our own experiments, done with the Middlebury dataset, do not show the same results of suppressing the non-dominant motion in the region. This is due to the effect that the influence of a data measurement on the solution depends on its cornerness. The relation between cornerness and solving the local flow is analysed by Shi and Tomasi [15], who use it to find good feature points. As a new result of our experiments and Black’s and Anandan’s observation, the generalized aperture problem can only reduced by robust estimation if the differently moving patterns have nearly the same cornerness.

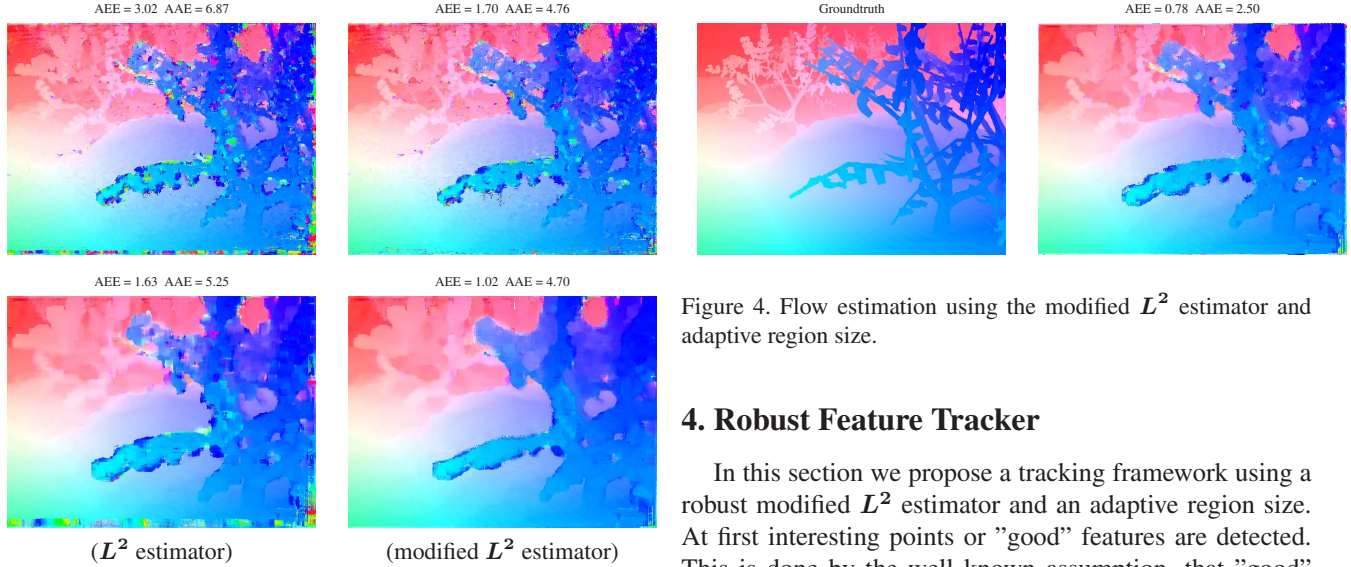


Figure 3. Flow estimation using the  $L^2$  (left) and the modified  $L^2$  estimator (right) and varying region sizes: top -  $7 \times 7$  and bottom -  $17 \times 17$ .

In figure 3 the dilemma is shown for the  $L^2$  and the more robust modified  $L^2$  estimator. The average endpoint error (AEE) and the average angular error (AAE) are given for a numerical feedback. The right column shows the improvement realized when the robust estimator is used. But the dilemma still exists. As illustrated by using small region the results are noisy while by using large region the results are blurred.

**Proposed strategy to adapt the region size:** Therefore we propose a strategy to adapt the region size successively. At first, the displacement vector is computed applying a large region ( $\Omega_{large}$ ) for  $k_l$  iterations with  $k_l \ll k$  or until a minor convergence is reached. Then the iteration is applied to a small region ( $\Omega_{small}$ ) to achieve a better performance at the motion boundaries. To avoid the aperture problem caused by small regions, the minimal eigenvalue of the matrix  $\mathbf{G}$  is computed to decide if the tracking was successful [15, 19]. Another decision if the feature is trackable with  $\Omega_{small}$  is taken by comparing the normalised residual error  $E_{local}$  of the small and large region. In this case, the iteration is continued until reaching the final convergence or maximal number of iterations  $k$ . If  $\Omega_{small}$  is not trackable the region size will be increased successively until a trackable size or  $\Omega_{large}$  is reached.

Figure 4 (right) shows the improvement by the adaptive strategy. Related to the large region at figure 3 (bottom right) the motion structure is less blurred and related to the small region, figure 3 (top left), the outliers at the motion boundaries are reduced.

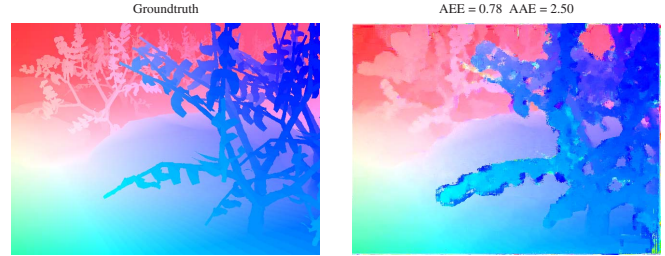


Figure 4. Flow estimation using the modified  $L^2$  estimator and adaptive region size.

#### 4. Robust Feature Tracker

In this section we propose a tracking framework using a robust modified  $L^2$  estimator and an adaptive region size. At first interesting points or "good" features are detected. This is done by the well known assumption, that "good" features should have a rich structure tensor [15], where the measure is defined as the minimal eigenvalue of the gradient matrix  $\mathbf{G}$ . In addition a two-pass texture analysis and non-maximum suppression procedure, as introduced by Sinha *et al.* [18], is applied to avoid concentration of features at few highly textured regions. After displacement computation the features are kept if convergence is reached and the residual error  $E_{local}$  falls below a threshold.

We implemented our proposed tracking algorithm in C++ using the OpenCV 2.0 interfaces and in addition we have a parallelised and GPU-accelerated implementation using the OpenCL framework.

#### 5. Evaluation

The evaluation is based on two criteria: the common performance measures for dense flow fields as the average angular error (AAE) and the average endpoint error (AEE)[16] and the tracking efficiency  $\eta$ , which was introduced by Singh *et al.* [17]. This measure is defined as the quotient of the number of successfully tracked features and the total number of features initialized. We propose to combine both criteria into a new plot, the tracking performance, see figure 5. The tracking performance plot is a measurement describing the ability to track features( $\eta$ ) including their tracking errors (AAE, AEE).

Varying tracking efficiency is produced by varying the threshold of the residual error between each initial and tracked image patch. All experiments are done by choosing a small region of  $7 \times 7$ , a large region of  $17 \times 17$  ( $\Omega_{large}$  is set to not adaptive methods, comparing figure 3 large regions produced less AEE and AAE), overrelaxation parameter 0.8 and norm parameter  $\sigma_1 = 8$  and  $\sigma_2 = 40$ .

A convergence experiment is shown in figure 5 (top). As shown, iteration numbers of 11 to 21 are a good compro-



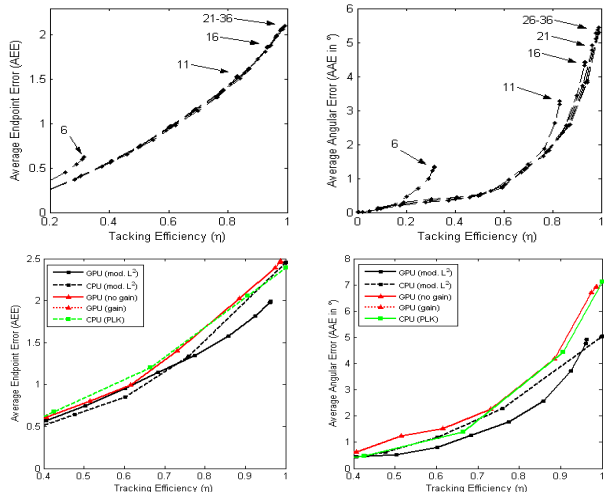


Figure 5. Comparison of tracking performance plot (*Grove3*) for modified  $L^2$  with varying numbers of iterations (top) and for different methods: proposed CPU and GPU modified  $L^2$ , GPU(no gain) [18], GPU(gain) [21] and OpenCV implementation CPU(PLK) [4] (bottom).

mise in terms of accuracy and in order not to unnecessarily increase the execution time. For further experiments the maximal number of iterations is therefor set to 21.

A comparison with state-of-the-art methods is illustrated in figure 5 (bottom). The GPU and CPU implementation were compared to the GPU implementation of the gain adaptive KLT Tracker (GPU(gain)) by Zach *et al.* [21] and the GPU KLT Tracker (GPU (no gain)) proposed by Sinha *et al.* [18] and the pyramidal Lucas-Kanade method (CPU (PLK)) proposed by Bouguet [4] available in OpenCV. The results show the improvements realized when the displacement vector is computed by using a robust estimator instead of  $L^2$  estimator and using adaptive region sizes. Because *Grove3* has no changing illumination, the curves of Sinha and Zach are nearly equal.

Figure 7 shows the dense flow computed by the proposed method. By comparing the results with global optical flow methods [22, 20] our method could not increase the accuracy. Generally in terms of accuracy recent global flow methods are superior to the local ones. But in respect of their run-time performance just local flow methods are able to perform these tracking tasks in real time. With the increasing video cameras resolution (from VGA resolution ( $640 \times 480$  pixel) to HD ( $1920 \times 1080$  pixel)), the speed of the methods becomes even more important.

Figure 6 shows the run-time of the GPU implementation by Zach *et al.* (run-time by Sinha *et al.* is similar) and our method applied on a Nvidia GTX 275 graphic device. In contrast the GPU implementation of the Dual-TV- $L^1$  has a run-time of 484ms on VGA and 2375ms on HD content and the method of Werlberger has a run-time of 1672ms

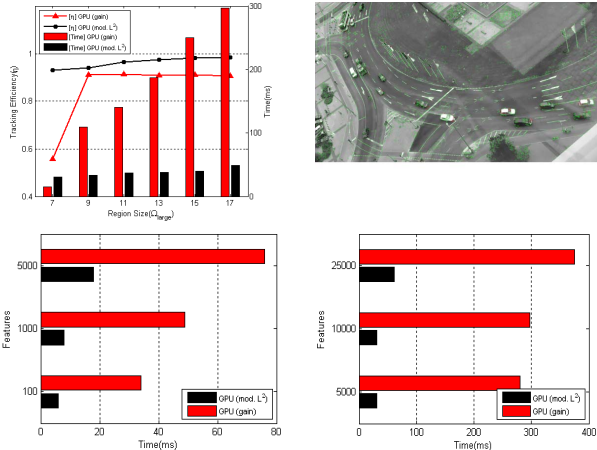


Figure 6. Run-time and tracking efficiency for 10.000 feature points and varying region sizes (top left) on HD *Car* sequence (top right) and for varying number of feature points for VGA (*Grove3*) (bottom left) and HD (*Car*) (bottom right).

on VGA and 5687ms on HD content. The figure shows that our method and the method of Zach *et al.* has a much higher run-time performance which could be categorised with 76-370ms as real-time tracking on HD content. Also an additional benefit in terms of run-time rely to the adaptive region strategy of our method could be seen. While Zach *et al.* iterate all the time with the large region our strategy shrinks the region so that most of the time iteration are done with lower region sizes. As a consequence the run-time for our method is decreasing.

Indeed, in many applications a high accuracy is less important than a very fast run-time. Figure 6 shows the run-time and tracking efficiency for a huge number of features and varying large region size on a HD sequence. By choosing a very small region size Zack *et al.* achieve a run-time of 15ms (66.6fps) but their tracking efficiency decreases. By 31ms (32.2fps) our method fulfills the real-time requirement also in HD sequences while maintaining a good tracking efficiency.

## 6. Conclusion

In this paper we illustrated the benefit of a robust framework for the widely used KLT Tracker. We proposed a new estimator based on the  $L^2$  norm with robust characteristics. To cope with the generalized aperture problem a strategy to adapt the region size was given. We demonstrate the improvements in terms of accuracy of these two new aspects by comparing this method with state-of-the-art implementations [4, 18, 21]. We also demonstrate the excellent run-time performance of our implementation. It is possible to track up to 10.000 feature points on HD material in real-time (>25fps). In the future we will include gain-adaption as proposed by Zach *et al.* [21].

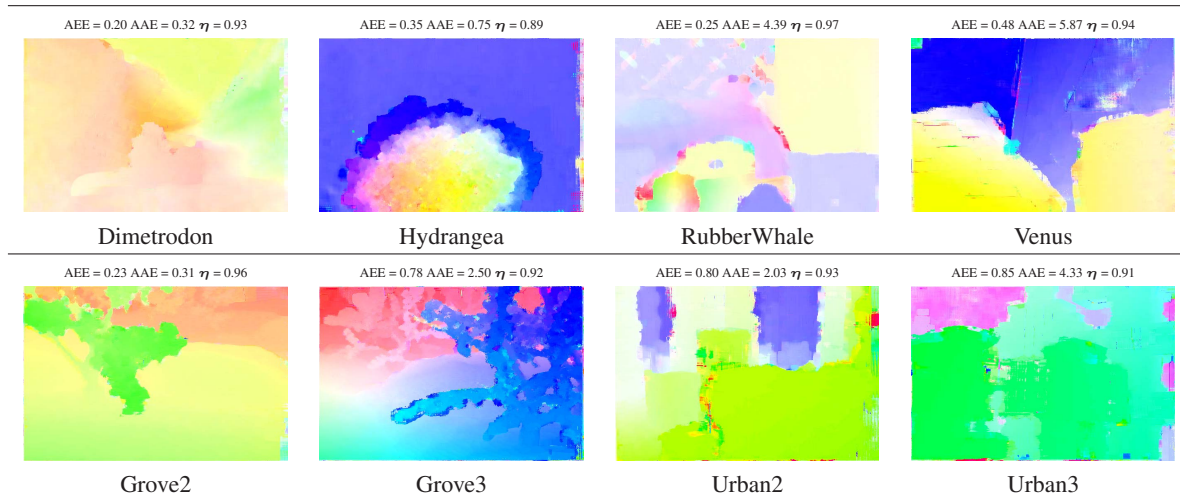


Figure 7. Evaluation results for dense optical flow on the Middlebury dataset.

## References

- [1] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *IEEE International Conference on Computer Vision (ICCV 93)*, pages 231–236, 1993. 1
- [2] M. J. Black and P. Anandan. The robust estimation of multiple motion: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 1, 2, 3
- [3] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–432, 1998. 2
- [4] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. Technical report, Intel Corporation Microprocessor Research Lab, 2000. 2, 3, 5
- [5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV 04)*, pages 25–36, 2004. 1
- [6] H. Fassold, J. Rosner, P. P. Schallauer, and W. Bailer. Realtime klt feature point tracking for high definition video. In *Computer Graphics, Computer Vision and Mathematics (GraVisMa 09)*, 2009. 1
- [7] 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)*. Wiley-Interscience, New York, 2005. 2
- [8] A. Jepson and M. Black. Mixture models for optical flow computation. In *Computer Vision and Pattern Recognition (CVPR 93)*, pages 760–761, 1993. 3
- [9] M. Kharbat, N. Aouf, A. Tsourdos, and B. White. Robust brightness description for computing optical flow. In *British Machine Vision Conference (BMVC 08)*, 2008. 2
- [10] Y. Kim, A. Martinez, and A. Kak. A local approach for robust optical flow estimation under varying illumination. In *British Machine Vision Conference (BMVC 04)*, 2004. 2
- [11] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981. 1, 2
- [12] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67:141–158, 2006. 1, 2
- [13] B. K. P.Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. 1
- [14] T. Senst, V. Eiselein, and T. Sikora. II-LK a real-time implementation for sparse optical flow. In *International Conference on Image Analysis and Recognition (ICIAR 10)*, pages 240–249, 2010. 1
- [15] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition (CVPR 94)*, pages 593–600, 21-23 1994. 3, 4
- [16] J. L. S. R. M. J. B. Simon Baker, Daniel Scharstein and R. Szeliski. A database and evaluation methodology for optical flow. Technical report MSR-TR-2009-179, Microsoft Research, 2009. 4
- [17] M. Singh, M. K. Mandal, and A. Basu. Gaussian and laplacian of gaussian weighting functions for robust feature based tracking. *Pattern Recognition Letters*, 26(13):1995–2005, 2005. 2, 4
- [18] M. P. Sudipta N. Sinha, Jan-Michael Frahm and Y. Genc. Gpu-based video feature tracking and matching. Technical report 06-012, UNC Chapel Hill, 2006. 1, 4, 5
- [19] C. Tomasi and T.Kanade. Detection and tracking of point features. Technical report CMU-CS-91-132, CMU, 1991. 1, 2, 4
- [20] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber- $L^1$  optical flow. In *British Machine Vision Conference (BMVC 09)*, 2009. 1, 2, 5
- [21] C. Zach, D. Gallup, and J. Frahm. Fast gain-adaptive klt tracking on the gpu. In *Visual Computer Vision on GPUs workshop (CVGPU 08)*, pages 1–7, 2008. 1, 5
- [22] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV- $L^1$  optical flow. In *DAGM-Symposium*, pages 214–223, 2007. 1, 2, 5