

Road User Tracker Based On Robust Regression With GNC And Preconditioning

Andreas Leich^a, Marek Junghans^a, Karsten Kozempel^a, Hagen Saul^a

^aInstitute of Transportation Systems, Rutherfordstr. 2, 12489 Berlin, Germany;

ABSTRACT

In this paper an early vision tracking algorithm particularly adapted to the tracking of road users in video image sequences is presented. The algorithm is an enhanced version of the regression based motion estimator in Lucas-Kanade style. Robust regression algorithms work in the presence of outliers, while one distinct property of the proposed algorithm is that it can handle with datasets including 90% outliers. Robust regression involves finding the global minimum of a cost function, where the cost function measures if the motion model is conform with the measured data. The minimization task can be addressed with the graduated non convexity (GNC) heuristics. GNC is a scale space analysis of the cost function in parameter space. Although the approach is elegant and reasonable, several attempts to use GNC for solving robust regression tasks known from literature failed in the past. The main improvement of the proposed method compared with prior approaches is the use of a preconditioning technique to avoid GNC from getting stuck in a local minimum.

Keywords: Motion Estimation, Optical Flow, Robust Regression, Object Tracking, Gaussian Continuation, GNC

1. INTRODUCTION

A novel tracking algorithm is introduced and tested on real world video image data of road users that relies on the evaluation of image intensities of all pixels in some predefined image region (region of interest ROI) over consecutive images. A motion model is fit to the pixel data so that a cost function is at its minimum, when the model parameters fit the dominant motion in the image region. This approach is known in literature as the robust regression based approach to motion estimation.¹ In contrast to least squares regression based motion estimation (section 2), robust regression promises improved dealing with outliers in the data. In case of robust regression, the surface of the cost function is multimodal in general, especially, when the ROI contains more than one moving object. In order to address this issue and in contrast with widely adopted approaches that rely on random sampling (RANSAC), in this paper we investigate the graduated non convexity (GNC) approach, introduced by BLAKE and ZISSERMAN² in 1987. The idea of GNC is to do a scale space analysis in the parameter space where the cost function is defined (section 3).

In this paper we present an analysis of the reasons why prior approaches to use GNC for regression based motion estimation failed¹ (sections 3.1 and 3.2). Our studies show that failure stems from numeric issues. We name this finding the condition problem of GNC. We present a preconditioning method as a solution for this problem and extend known algorithms accordingly 3.2.

We analyze the effectiveness of the method both on synthetic data with randomly sampled outliers and on real world data within an experimental real time tracking system (section 4.1). The results on synthetic data show, that the proposed method stably works when the number of uncorrelated outliers exceeds 90% (section 4.2). Real world tests show, that using the novel algorithm, road users can be tracked over some hundreds of meters (section 4.3).

Further author information: (Send correspondence to A.L.)

A.L.: E-mail: andreas.leich@dlr.de, Telephone: +49 30 67055409

M.J.: E-mail: marek.junghans@dlr.de, Telephone: +49 30 67055214

Video Surveillance and Transportation Imaging Applications 2015, edited by Robert P. Loce, Eli Saber,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9407, 940702 · © 2015 SPIE-IS&T
CCC code: 0277-786X/15/\$18 · doi: 10.1117/12.2082520

2. REGRESSION BASED MOTION ESTIMATION

Motion can be estimated by comparing two images $\mathcal{I}_1(\mathbf{x})$ (template image) and $\mathcal{I}_2(\mathbf{x})$ which have been registered at different times and which contain the same moving object. The image \mathcal{I}_2 is being transformed by some argument transform $\mathbf{x}_t = f(\mathbf{x}, \vec{\phi})$ and compared with the template \mathcal{I}_1 afterwards. Motion parameters $\vec{\phi}$ are considered optimal if the difference between the images is minimal. A common choice to measure the differences between the images $\mathcal{I}_1(\mathbf{x})$ and $\mathcal{I}_2(\mathbf{x})$ is the euclidean L2-norm. Using the L2-norm, a continuous cost function can be introduced as follows:

$$\mathcal{E}(\vec{\phi}) = \sum_{p=1}^{N_p} (\mathcal{I}_1(\mathbf{x}_p) - \mathcal{I}_2(f(\mathbf{x}_p, \vec{\phi})))^2. \quad (1)$$

Here, the index p runs over all N_p pixels of a region of interest (ROI) in \mathcal{I}_2 . The pixel is defined by its coordinate vector \mathbf{x}_p . The argument transform function is called motion model. In general, a motion model is nonlinear. It is reasonable to approximate a motion model by a first order taylor series expansion, especially when the model is used to describe the motion in a small region of interest. A first order taylor series expansion gives a so called affine motion model. An affine motion model describes translation, rotation, shear and zoom of the object in the ROI. For tracking road users in image sequences recorded by a roadside camera, a further reduced four parametric motion model can be applied, which only describes zoom and translation of the tracked object.

$$\begin{aligned} \mathbf{x}_t &= \mathbf{W}(\mathbf{x})\vec{\phi} \\ &= \begin{pmatrix} 1 & x & 0 & 0 \\ 0 & 0 & 1 & y \end{pmatrix} [\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4]^T \end{aligned} \quad (2)$$

The motion model is expressed using a warp matrix \mathbf{W} here according to BAKER and MATTHEWS.³ The location of the global minimum of cost function \mathcal{E} can be considered as an optimal motion. The cost function is used to measure the difference between \mathcal{I}_1 and \mathcal{I}_2 and is defined in the N_ϕ -dimensional parameter space. In case of the L2-norm, the minimum of \mathcal{E} is found by solving an ordinary least squares problem (OLS). The OLS in general has the following form:

$$\mathcal{E}_{OLS}(\Delta\vec{\phi}) = \sum_{p=1}^{N_p} \left(e_p(\vec{\phi}_0) + \frac{\partial e_p}{\partial \vec{\phi}}(\vec{\phi}_0)\Delta\vec{\phi} \right)^2 = \min, \quad (3)$$

Premise for an OLS formulation of a problem is that the first order taylor series expansion does not leave a remainder term. Applied to the motion estimation problem we find:

$$e_p(\vec{\phi}) = \mathcal{I}_1(\mathbf{x}_p) - \mathcal{I}_2(\mathbf{W}(\mathbf{x}_p)\vec{\phi}). \quad (4)$$

Since the image \mathcal{I} is not necessarily a linear function, the motion estimation problem is a non-linear least squares problem. A non-linear least squares problem can be solved iteratively. In each iteration, a solution $\Delta\vec{\phi}$ of the OLS of a linearized approximation of the non-linear problem is calculated and the current estimate $\vec{\phi}_0$ corrected:

$$\vec{\phi}_0(k+1) = \vec{\phi}_0(k) + \Delta\vec{\phi}. \quad (5)$$

As mentioned earlier, $\Delta\vec{\phi}$ is being computed by solving the OLS of motion estimation. We introduce the residual \mathbf{r} and write (3) in matrix form:

$$\mathcal{E}_{OLS}(\vec{\phi}) = \|\mathbf{r}\|_2^2 = \min \quad (6)$$

$$\mathbf{r} = \mathbf{A}\vec{\phi} - \mathbf{b}. \quad (7)$$

For simplicity here we write $\vec{\phi}$ instead of $\Delta\vec{\phi}$. The solution of (6) is:

$$\Delta\vec{\phi} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \quad (8)$$

where $\mathbf{A}^T \mathbf{A}$ is referred to as Gauss-Newton-Hessian.

By repeatedly evaluating (8) and inserting into (5) one gets a sequence $\vec{\phi}_0(k)$ of solutions which can be expected to converge to the minimum of the nonlinear cost function \mathcal{E} . This algorithm is well known in the field of image processing as the LUCAS-KANADE algorithm⁴ or, combined with an interest point detector, as KANADE-LUCAS-TOMASI-Tracker (KLT-Tracker).⁵

ALGORITHM 2.1 (LUCAS-KANADE-ALGORITHM).

1. Define a starting value $\vec{\phi}_0(k=0)$.
2. Calculate an optimal correction $\Delta\vec{\phi}$ for $\vec{\phi}$ using (8) and insert it into $\vec{\phi}_0(k+1) = \vec{\phi}_0(k) + \Delta\vec{\phi}$.
3. Transform \mathcal{I}_2 using $\vec{\phi}_0(k+1)$. In order to do that, sub-pixel intensity values in the image must be interpolated.
4. Repeat steps 2. and 3. until some termination condition $|\Delta\vec{\phi}(i)| < \varepsilon$ true.
5. The sum of all vectors $\Delta\vec{\phi}(k)$ is the optimal motion vector.

Figure 1 contains a visualization of this algorithm. The image \mathcal{I}_2 is transformed using the parameters $\vec{\phi}_0$ and compared with \mathcal{I}_1 afterwards. The correction $\Delta\vec{\phi}$ for the motion vector is calculated by finding the minimum of the OLS cost function \mathcal{E}_{OLS} .

In the following chapters we introduce an improved version of this algorithm. Notes:

- The derived algorithm for motion estimation is a regression based method. Regression analysis in general is being applied when a big amount of measured data (here: image pixel intensities and image derivatives) is available for analysis and when there is some model that is being fit to the data (here: motion model). The regression approach is effective in the presence of noise, if the impact of noise is averaged at the minimum of the cost function (e.g. Gaussian noise). The algorithm can behave arbitrarily poor in the presence of any other noise than Gaussian noise.^{6,7} We suggest an improved version of the algorithm in the sense of robust statistics.⁷ Our novel version is tailored to the situation that the ROI contains multiple occluding and/or non-occluding moving objects. It is known from literature, that this situation is relevant in traffic surveillance and is difficult to handle with state of the art algorithms.⁸
- Matrix equation 7 is an overdetermined system of equations. Here, we introduce a method for finding a so called closest point solution for it.⁹ Every row equation in (7) defines a $N_\phi - 1$ -dimensional hyper-plane in the parameter space. The closest point solution is a solution that is as close as possible to as many as possible hyper-planes. Known approaches for finding closest point solutions are

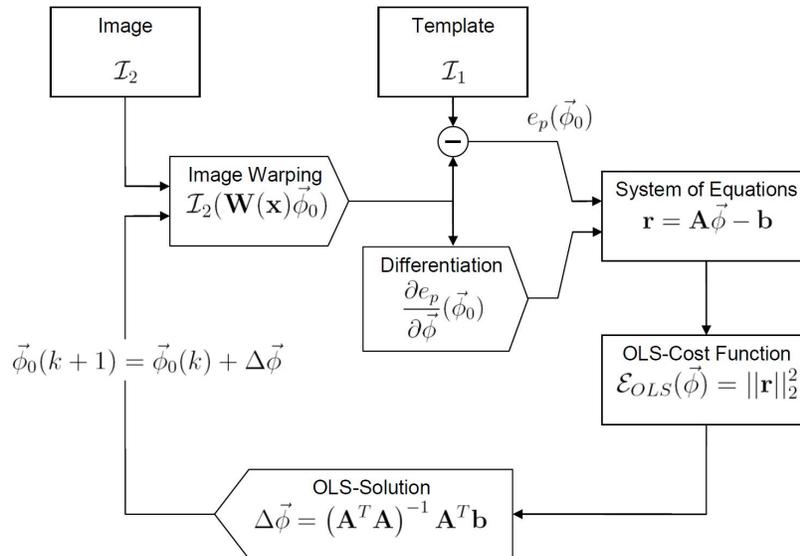


Figure 1. Visualization of the LUCAS-KANADE-Algorithm

- Random Sampling Consensus (RANSAC)¹⁰ and improved versions of RANSAC. Among the latter, locally optimized random sampling consensus has been investigated recently.¹¹
 - Inverse histogramming methods like the Hough Transform¹² and improved versions of the Hough Transform. Among the latter, the hash table implementation of the Hough Transform has been introduced by LOWE.¹³
 - Using the least median of squares as an error function instead of the L2-norm.¹⁴
 - Re-formulation of the problem as an optimal labeling problem and solving using graph cuts. One can look for an optimal assignment of labels (e.g. model1, model2,..., outlier) so that deviations from each model get minimal. This approach is especially useful, if there is additional information, e.g. measurements come from the neighborhood of some pixel (PEaRL¹⁵).
 - Graduated Non-Convexity.² The hyper-planes in parameter space are low pass filtered in order to produce a convex cost function as a starting point. The minimum of the convex cost function is determined using a gradient based approach and tracked while the smoothing parameter is being lowered gradually until some termination condition is met. This approach was first introduced by BLACK and ANANDAN.¹
- The novel algorithm is the result of studying the properties and the fail conditions of the robust regression approach by BLACK and ANANDAN.¹ The reasons for failure are (a) the way the linear system (7) is being formed (see section 3.1) and (b) the way it is being solved.
 - The novel algorithm can be used in any application, where multiple models have to be fit to measurement data and the problem can be expressed as an overdetermined linear system of equations (7). Here, it can compete and can be combined with other approaches, like locally optimized RANSAC.

3. SCALE SPACE ANALYSIS IN PARAMETER SPACE

In order to make our findings comprehensible, we use line fitting examples in a 2D parameter space. Our model is the slope intercept line parametrization. Model parameters are the slope m of the line and the interception n of the line and the ordinate axis. The estimation error is the distance Δy between measurement and line characterized by the model parameters.

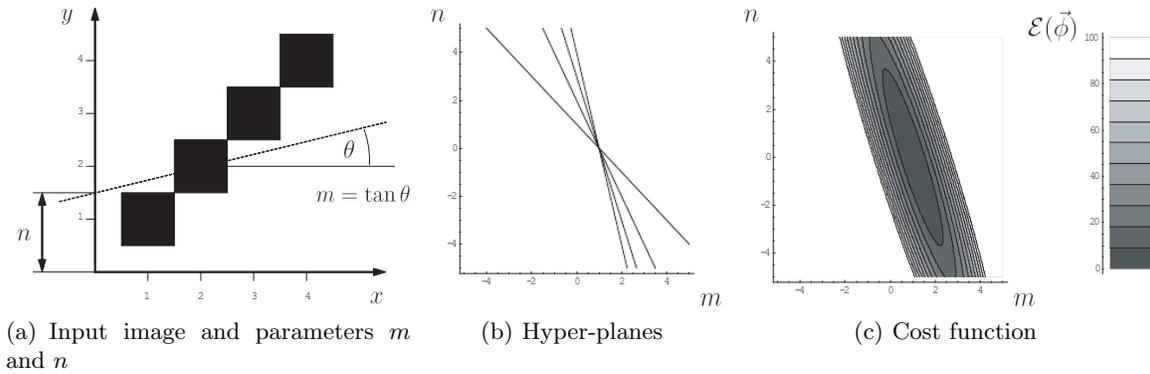


Figure 2. Visualization of the overdetermined linear system in example 3.1

EXAMPLE 3.1 (LINE FITTING (1)).

The task is to find lines in images. The input images have zero and non-zero pixels or binary images. We are looking for lines that go through as much pixels as possible. The first input image is image 2(a). The overdetermined linear system of equations is $\mathbf{A}\vec{\phi}=\mathbf{b}$, where \mathbf{A} is a $N_p \times 2$ -Matrix und N_p is the number of measurements.

$$\vec{\phi} = \begin{bmatrix} m \\ n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_p} \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_{N_p} & 1 \end{bmatrix}$$

The vector \mathbf{b} is an N_p -dimensional vector containing the y -coordinates of the measured data samples. The parameter vector $\vec{\phi}$ contains the parameter m and n . We solve the linear system of equations in OLS-fashion using equation (8). This means that we find the minimum of the quadratic cost function \mathcal{E}_{OLS} (equation (6)). This cost function is displayed in figure 2(c). After one iteration we find $n = 0$ and $m = 1$ as the solution of line fitting.

If we add two outliers to the input image of example 3.1, the OLS solution fails, because it finds some compromise between the two possible lines $[m = 1; n = 0]^T$, which is supported by four pixels and $[m = -1; n = 4]^T$, which has a support of three pixels (figure 3).

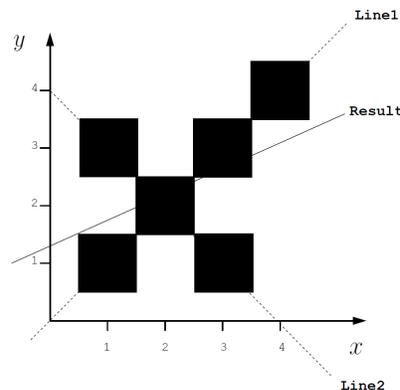
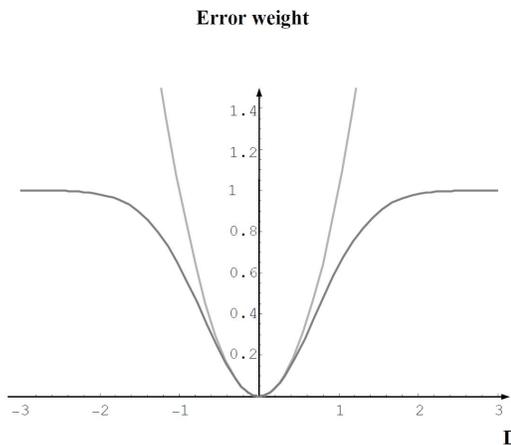


Figure 3. The OLS solution fails in example 3.1 where the measured data contains outliers.

If we apply the GNC* approach of BLAKE and ZISSERMAN² in this context, we replace the quadratic error function by a robust one (figure 4), e.g. the LeClerc¹⁷ function (9) and implement algorithm 3 conform with the approach of BLACK and ANANDAN.¹ This works with the input image shown in figure 3. Details on the calculation steps are given in example 3.2.



$$\rho(r, \sigma) = 1 - \exp\left(-\frac{r^2}{\sigma^2}\right). \quad (9)$$

Figure 4. The squared difference (gray) versus a robust error function (black). The robust error function penalizes big differences between the model estimate and measured data (outliers) less.

EXAMPLE 3.2 (LINE FITTING (2)).

Instead of using the L2-norm $\mathcal{E}_{OLS} = \|\mathbf{r}\|_2^2$ for the cost function, applying the LeClerc function yields:

$$\mathcal{E}_\rho(\vec{\phi}, \sigma) = \sum_{N_p} \exp\left(\frac{(y_p - x_p m - n)^2}{(x_p^2 + 1)\sigma^2}\right).$$

For more details on the denominator refer to section 3.1. This function is continuous and twice differentiable. The Newton method can be applied for minimizing this function:

$$\Delta\vec{\phi} = -\mathbf{H}^{-1}\nabla\mathcal{E}_\rho(\vec{\phi}, \sigma); \mathbf{H} = \nabla^2(\mathcal{E}_\rho(\vec{\phi}, \sigma))$$

After each iteration of the Newton method, the parameter σ is being lowered by the factor $c_\sigma = \frac{1}{2}\sqrt{2}$ until the termination criterion $\sigma(k) < \sigma_{\min}$ is true. In this example, $\sigma_{\min} = 0.1$ and $\sigma(k=0) = 10$ are reasonable choices. After 14 iterations, the algorithm converges at point $\vec{\phi} = [m = 1; n = 0]^T$.

Using the robust regression approach with GNC, the graphical representation of the motion estimation algorithm shown in figure 1 is extended as shown in figure 6.

*Recently, the term Gaussian Continuation had been introduced by MOBAHI and FISHER for this sort of algorithm.¹⁶

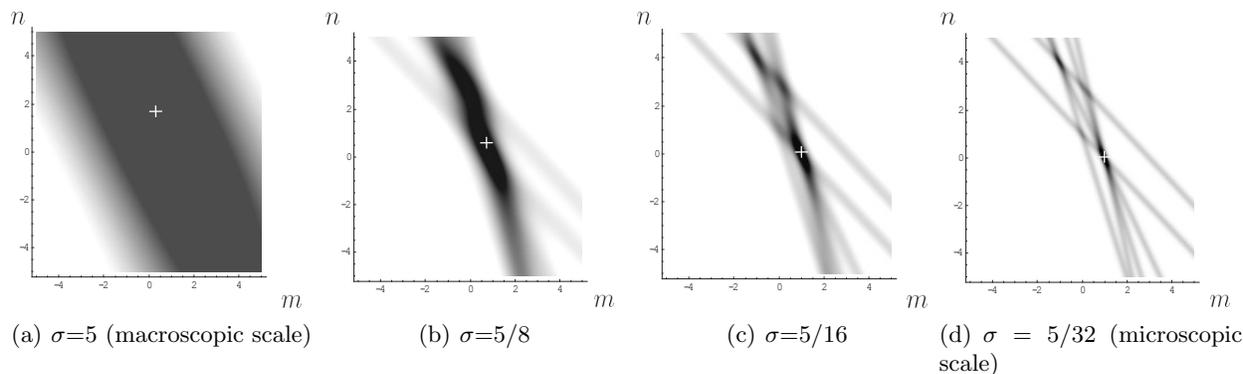


Figure 5. Course of the iteratively updated cost function minimum when scale space analysis in parameter space is applied. The minimum is marked with a white cross.

ALGORITHM 3.1 (GNC).

1. Pre-select an interval $[\vec{\phi}_{min}; \vec{\phi}_{max}]$ in parameter space, where the global minimum is expected.
2. Define σ_{min} for your problem. To do so, one needs to account for the expected amplitude of noise and the desired resolution.
3. Create a cost function $\mathcal{E}(\vec{\phi}, \sigma)$, that is convex in the whole interval of parameter space. Select a big scale parameter value $\sigma(k=0)$.
4. Select a start vector in parameter space $\vec{\phi}(k=0)$ within the pre-selected interval $\vec{\phi}_{min} < \vec{\phi}(k=0) < \vec{\phi}_{max}$.
5. Find a vector $\vec{\phi}_0(k)$, so that \mathcal{E} is minimal. An iterative minimization technique can be used, e.g. the Newton-Method.
6. Lower the scale parameter using some rule, e.g. $\sigma(k+1) = c_\sigma \sigma(k)$.
7. repeat steps 5. and 6. until $\sigma(k) < \sigma_{min}$ and a termination criterion of the iterative minimization technique are reached, e.g. $|\vec{\phi}_0(k+1) - \vec{\phi}_0(k)| < \varepsilon$.

Extensive tests show that algorithm sometimes works and sometimes fails. It can be extended for fitting circles and other patterns as has been shown by LEICH, JUNGHANS and JENTSCHL.¹⁸ But it fails even in simple 4x4 images shown in the examples in this paper. Failure is connected with two problems, which are discussed in sections 3.1 and 3.2.

3.1 CONTROL OF SCALE IN PARAMETER SPACE

In example 3.2 we used the term $(x_p^2 + 1)^{-1/2}$ in the denominator. In this section we explain why.

CONDITION 3.1 (INVARIANCE OF SCALE IN PARAMETER SPACE). *Algorithm 3 requires that the effective scale in parameter space σ does not depend on $\vec{\phi}$ itself in the whole interval $[\vec{\phi}_{min}; \vec{\phi}_{max}]$.*

If we use, as proposed by BLACK and ANANDAN,¹ the residual of the line equation as argument for the error function, then we introduce an implicit argument transformation in parameter space. For ease of understanding, consider the case of a 1D-translation motion model, where v is the translation parameter. Then the deviation of measurement and residual of the row equation is:

$$r = \mathcal{I}_2 - \mathcal{I}_1 + v\nabla\mathcal{I}_2, \quad (10)$$

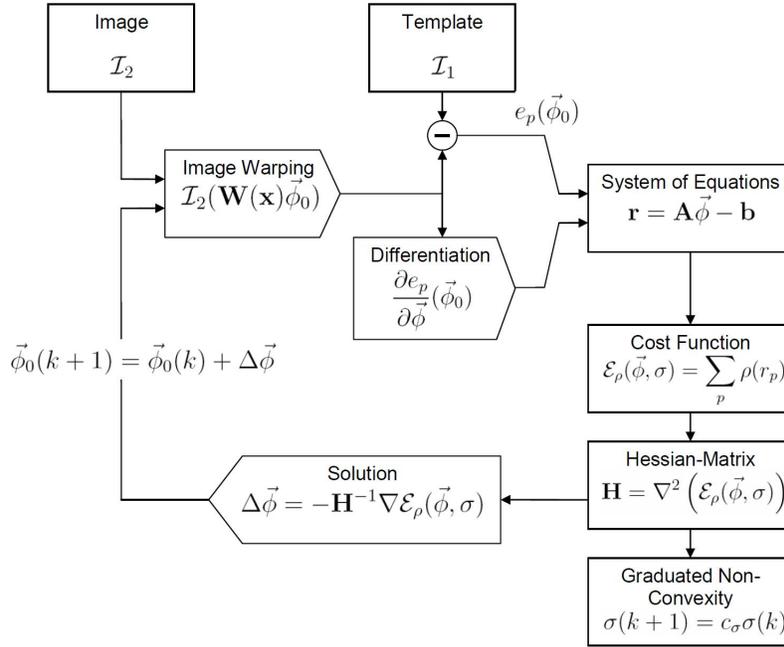


Figure 6. Motion estimation using robust regression and GNC (straightforward approach).

This is a linear function with the slope ∇I_2 (Note: in example 3.2, the slope is given by the parameter m). Since the argument of the error function contains this linear function, we have an argument transformation with a scale factor s in the direction of the v axis and a shift t :

$$\mathcal{E}(v) = \rho(I_2 - I_1 + \nabla I v) = \rho\left(\frac{v-t}{s}\right) \quad (11)$$

$$= \rho\left(\nabla I \left(\frac{I_2 - I_1}{\nabla I} + v\right)\right) \rightarrow s = \frac{1}{\nabla I}; t = \frac{I_1 - I_2}{\nabla I} \quad (12)$$

Small gradients ∇I_2 cause an expansion of the robust error function in the direction of the v axis and big gradients cause a compression respectively. This makes it impossible to control the convexity of the cost function. Experiments show that results improve, when a compensation for s is introduced.¹⁹ The intuitive general solution is using the perpendicular distance⁹ between the hyperplane of the row equation and the current estimate $\vec{\phi}(k)$. This is calculated using the hessian normal form:

$$\mathcal{E}(\vec{\phi}) = \sum_p \rho\left(\frac{a_{p1}\phi_1 + a_{p2}\phi_2 + \dots + a_{pN_\phi}\phi_{N_\phi} - b(p)}{\sqrt{a_{p1}^2 + a_{p2}^2 + \dots + a_{pN_\phi}^2}}\right),$$

where the elements of matrix \mathbf{A} are denoted as a . This formula was applied in example 3.2.

3.2 CONDITION PROBLEM

Any subset of row equations from (7) that contains N_ϕ linear independent row equations form a sub-problem that has exactly one solution.

CONDITION 3.2 (CONDITIONING OF SUB-PROBLEMS). *Algorithm 3 requires that all sub-problems of an overdetermined linear system are equally conditioned.*

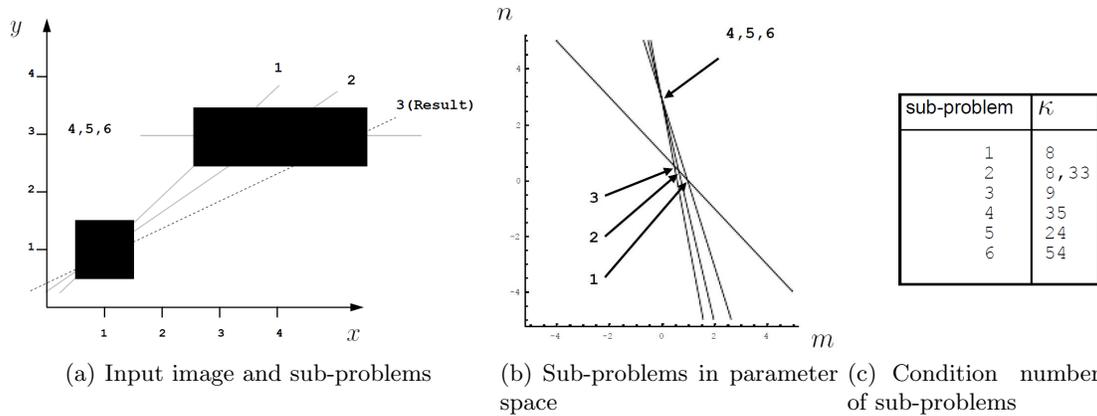


Figure 7. Overdetermined linear system that is problematic from the point of view of the condition problem.

Violation of condition 3.2 leads to a problem that we call condition problem of regression based GNC. The condition problem is of practical relevance, especially in motion estimation and road user tracking. It is illustrated in example 3.3.

EXAMPLE 3.3 (CONDITION NUMBERS OF SUB-PROBLEMS).

We try to fit lines in the input image 7(a). Instead of the global minimum $\vec{\phi} = [m = 0; n = 3]^T$ algorithm 3 finds the local minimum $\vec{\phi} = [m = 0.5; n = 0.5]^T$. In figure 7(a), the solutions of the sub-problems are displayed. Obviously, algorithm 3 converges to sub-problem 3 while the global solution is the same as for sub-problems 4,5 and 6. Figure 7(b) illustrates that the intersection angle of the lines that represent the hyper-planes of sub-problem 1, 2 and 3 is bigger than of the intersection angle of sub-problems 4,5 and 6. This intersection angle is a geometric representation of the condition number. The smaller the intersection angle, the worse the condition number. Obviously, GNC tends to converge to sub-problems that are better conditioned instead of converging to the global minimum.

We write \mathbf{A}_t for the matrix of the sub-problem of \mathbf{A} . The condition number κ is defined as:²⁰

$$\kappa = \|\mathbf{A}_t\|_{\infty} \|\mathbf{A}_t^{-1}\|_{\infty} \quad (13)$$

Figure 7(c) shows the condition numbers of the sub-problems 1 to 6.

Condition 3.2 can be hardly true for all instances of an overdetermined linear system. We propose a method in this paper that improves the situation within the scope of what is mathematically feasible. Our proposed method uses preconditioning to address the condition problem. Preconditioning means, that a linear system is transformed into a better conditioned one:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \rightarrow \hat{\mathbf{A}}\hat{\mathbf{x}} = \hat{\mathbf{b}} \quad ; \quad \kappa(\hat{\mathbf{A}}) < \kappa(\mathbf{A}).$$

In general, there is the option to use left and right preconditioning,²¹ but for an overdetermined linear system, only right preconditioning can be applied. In this context, a unity matrix $\mathbf{I} = \mathbf{M}_2^{-1}\mathbf{M}_2$ is inserted into the linear system's matrix equation:

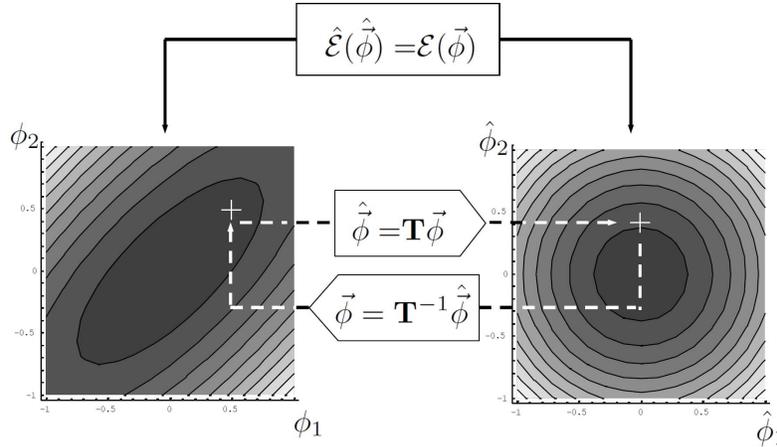


Figure 8. Preconditioning is the transformation of the cost function into an isotropic shape

$$\begin{aligned}
 (\mathbf{A}\mathbf{M}_2^{-1})(\mathbf{M}_2\vec{\phi}) &= \mathbf{b} \\
 \hat{\mathbf{A}}\hat{\mathbf{x}} &= \hat{\mathbf{b}}.
 \end{aligned}
 \tag{14}$$

A good preconditioner can be constructed by exploiting the issue, that a cost function of a well conditioned problem is isotropic. The isotropy properties of the cost function can be estimated by evaluating the Eigensystem of its hessian. The eigenvalues and eigenvectors of the hessian can be used to find optimal parameters for warping the parameter space in a way that improves the isotropy properties of the cost function.²² The argument transform is calculated by the matrix multiplication

$$\hat{\vec{\phi}} = \mathbf{T}\vec{\phi},
 \tag{15}$$

where the transformation matrix is found by a matrix decomposition of the hessian.

$$\mathbf{H} = \mathbf{T}^T\mathbf{T}.
 \tag{16}$$

There are numerous ways to find an appropriate factorization.²⁰ We focus on decomposition by diagonalization. The hessian is decomposed as follows:

$$\mathbf{H} = \mathbf{R}\mathbf{D}\mathbf{R}^T,$$

We compute the rotation matrices \mathbf{R} and the diagonal matrix \mathbf{D} using the Jacobi eigenvalue algorithm. The elements of the diagonal matrix \mathbf{D} are the eigenvalues of \mathbf{H} . The transformation matrix \mathbf{T} for the argument transform is therefore $\mathbf{T} = \sqrt{\mathbf{D}}\mathbf{R}^T$. With these findings, we extend algorithm 3 and get algorithm 3.2.

ALGORITHM 3.2 (PRECONDITIONING OF GNC).

1. Pre-Select an interval $[\vec{\phi}_{min}; \vec{\phi}_{max}]$ in parameter space, where the global minimum is expected.
2. Define σ_{min} for the problem.
3. Set a start vector $\vec{\phi}_0(k=0)$, initialize matrices $\mathbf{T}(k=0) = \mathbf{I}$ and $\hat{\mathbf{A}} = \mathbf{A}$.
4. Create a cost function $\mathcal{E}(\vec{\phi}, \sigma)$ that is convex at $\vec{\phi}_0(0)$ using a scale parameter $\sigma(k=0)$ that is coarse enough.
5. Create the preconditioned linear system $\hat{\mathbf{A}}\hat{\phi} = \hat{\mathbf{b}}$, calculate the hessian and its decomposition $\mathbf{H} = \mathbf{RDR}^T$.
6. Determine the smallest Eigenvalue λ_{min} of \mathbf{D} .
7. If $\lambda_{min} > 0$:
 Compute $\Delta\hat{\phi}$, e.g. using the Newton method.
 Update \mathbf{T} using the formula $\mathbf{T}(k) = c_\lambda \mathbf{T}_k \mathbf{T}(k-1)$.
 If Newton method converges: set $\sigma(k+1) = c_\sigma \sigma(k)$.
8. else if $\lambda_{min} \leq 0$, minimize the concave cost function.
9. Repeat steps 5 to 8 until $\sigma(k) < \sigma_{min}$ and $|\vec{\phi}(k+1) - \vec{\phi}(k)| < \varepsilon$
10. Back transform the result $\vec{\phi}_{opt} = \mathbf{T}^{-1}(k)\hat{\phi}(k)$

Notes:

- The factorization of the hessian in step 5 delivers intermediate results that can be used to speed up further processing: $\mathbf{H}^{-1} = \mathbf{RD}^{-1}$ and $\mathbf{T}^{-1} = c_T^{-1} \mathbf{RD}^{-1/2}$.
- It is a good practice to limit the step size of the Newton method in step 7. e.g. $(\Delta\vec{\phi})_{max} = \sigma/2$. We assume convergence of the Newton method (Step 7), if the step size is smaller than $\sigma(k)/2$.
- We use incremental preconditioning in step 7: $\hat{\mathbf{A}}(k) = \mathbf{AT}^{-1}(k-1)$.
- In step 7 the preconditioner has to be normalized because otherwise the effective scale σ in parameter space is changed by preconditioning. We use the normalization factor $c_\lambda = \frac{1}{\sqrt{\lambda_{min}(\mathbf{T}_k \mathbf{T}_k^T)}}$.

EXAMPLE 3.4 (PRECONDITIONED GNC).

GNC with preconditioning converges to the global minimum within eight iterations for the input image in figure 7(a). Figures 9(a) through 9(d) give a visual impression of the effectiveness of preconditioning. The plots are made for the interval $|\vec{\phi} - \vec{\phi}(k-1)| < \sigma(k)$, so that the center of the graphics is the current parameter estimate. We see, that preconditioning is an iterative process which converges to the optimally conditioned solution after numerous iterations.

Applied to the motion estimation problem, the motion estimation method illustrated in figure 6 can be extended. The novel method is shown in figure 10.

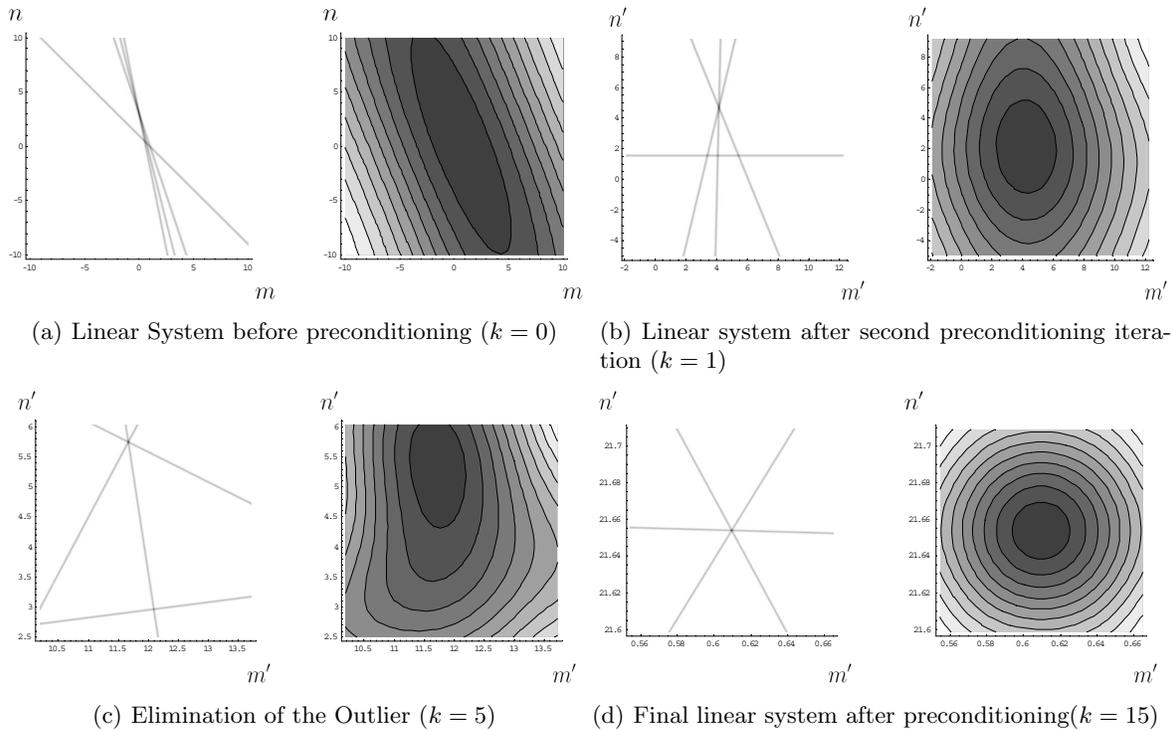


Figure 9. Illustration for 3.4. Left: Linear System. Right: Cost Function

4. EXPERIMENTS

We have implemented algorithm 3.2 in C++ using the numeric functions library of OpenCV. The corresponding software module was tested with synthetic and real world data.

4.1 IMPLEMENTATION ISSUES

The implementation includes a template class for the minimizer itself, a test class for the minimizer and a class for vehicle tracking. A vehicle trap based on a Histogram of Oriented Gradients (HoG²³) and a ground truth evaluation module were created as additional software modules. Notes:

- We implemented a real time tracking module where the processing framerate is being adapted to the computational load.
- We found that inliers and outliers are more effectively separated if the time gap between the template \mathcal{I}_1 and \mathcal{I}_2 is increased, until the parameter vector has some minimum length (e.g. 20 Pixels. Another good criterion is $7\% > \phi_2 > 5\%$; $7\% > \phi_4 > 5\%$ (equation (2)). While both methods work, best results were achieved with the latter criterion.
- We found that the algorithm works faster and equally effective if the preconditioner is not reset upon computation of a new image.
- BLACK and ANANDAN reported that in their implementation they had to compute the translation parameters first and zoom parameters afterwards.¹ Doing so we interpret as a workaround for the condition problem, because the two dimensional translational motion estimation problem is well conditioned (if there is no aperture problem). However, this approach is problematic if the affine motion parameters grow, because zoom can be particularly expressed by translation. In our implementation we optimize all parameters at the same time.

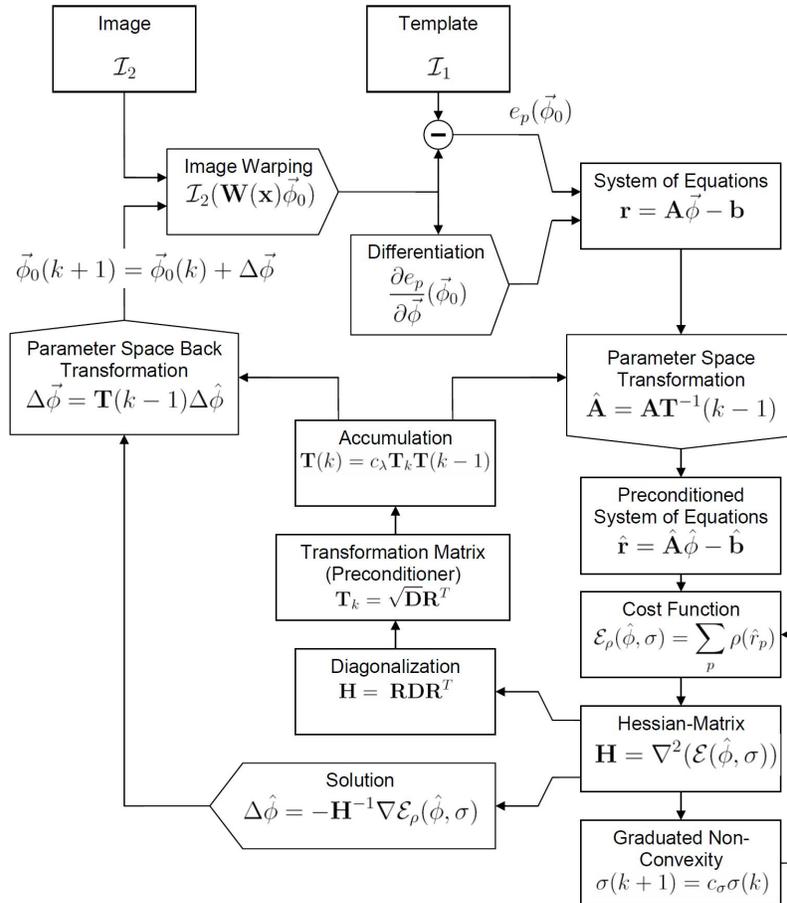


Figure 10. Preconditioning extension for motion estimation using robust regression and GNC

4.2 SYNTHETIC TEST DATA

The template class for the minimizer itself was tested with synthetic test data. The synthetic test data was generated with a random number generator. The test data includes inliers and outliers. Only two dimensional test data was used. Test parameters included the number of data point, the number of outliers and the noise σ_{noise} of the inliers. Only uncorrelated outliers were generated. Tests focused on finding the breakdown point of the algorithm, which is the number of outliers it can tolerate (see table in image 13).

We found that the more the value of the slope parameter m of the inlier data rises, the earlier the algorithm tends to break down. This is in line with expectation, because the steeper the slope m of the inlier data set, the more the condition numbers of inlier sub-problems rise, while the condition numbers of outliers sub-problems remain constant. We therefore added the constraint $m < 1$ to the test data generator.

4.3 REAL WORLD TEST DATA

We used the algorithm within the multi-sensor traffic surveillance and traffic situation detection software framework (MUSE) which is being developed at the German Aerospace Center. The MUSE framework is a platform for road user trajectory recording and situation evaluation. Within our tests, trajectories were recorded in a trap/track approach. Regions with elliptic shape were defined on the road. In learning mode, the support vector machine of the HoG detector was trained with a number of vehicles (we used training sequences of 15 min length with about one hundred cars). In operation mode, the HoG detector triggered the tracking algorithm, which used to track the detected road user over the whole road interval of sight. We assigned one tracker to one road user,



Figure 11. Screenshot from tracking software

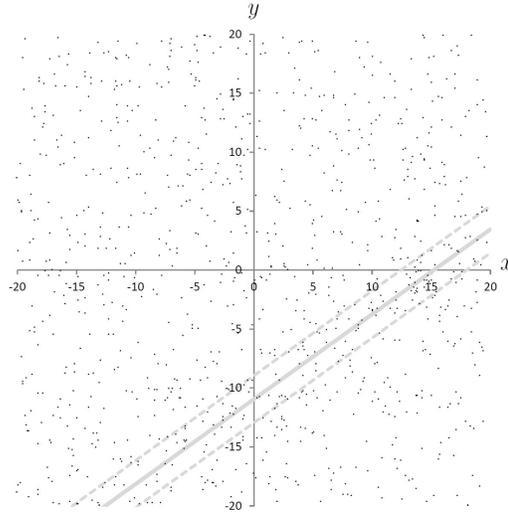


Figure 12. Synthetic test data with 90% outlier ratio. Ground truth is $m = 0.72$ and $n = -10.96$, noise is ($\sigma_{\text{noise}} = 3$).

Outliers Ratio [%]	Success Rate ($\sigma_{\text{noise}}=4$)
50%	100%
55%	100%
60%	100%
65%	100%
70%	99%
75%	99%
80%	97%
85%	94%
90%	83%
95%	59%

Figure 13. Success rate of the algorithm on synthetic data sets ($\sigma_{\text{noise}} = 4$) with rising outlier ratio

which is in contrast with state of the art feature tracking, where the additional computational step of feature grouping is required. Compared with tracking small features, tracking the whole vehicle has the advantage, that the regression algorithm evaluates a larger number of measurement data and therefore gives more precise results. Notes:

- We evaluated the algorithm in several traffic scenes. Since a four parametric zoom model is used, the tracker shows better results in scenes where vehicles were moving straight to/from the camera. Of the 1675 vehicles in the scene, 1609 vehicle were tracked correctly. If the camera is looking from above on a road crossing with turning movements, the tracker tries to find motion model parameters that express a rotation by zoom. This fails frequently.
- We found that $c_{\sigma} = \frac{1}{\sqrt{2}}$ gives good results over the range of tests. A lower value of c_{σ} leads to higher fail rates, higher values slow down the algorithm speed unnecessarily. This finding is in contrast to prior implementations of the GNC algorithm for robust regression motion estimation, where $c_{\sigma} = 0.95$ was used.¹ We attribute this issue to the effectiveness of preconditioning.
- Motion models of multiple tracked objects can be computed simultaneously on a state of the art PC. On an Intel Core i5 quad Core processor, a total number of about 20 000 Pixels can be processed in real time at 20fps. At higher computational load, the software starts dropping frames. If the frame rate drops below 5 fps, the tracking results become significantly worse.
- The most frequent failure phenomenon is expanding the ROI to an extent, where the background motion model is the dominant motion.
- While lowering σ , the minimizer encounters the situation, where the cost function becomes concave, because there are simply two valid models with nearly equal support in parameter space. We called this situation a branch point. Currently, no handling for branch points was implemented, except of that in case of a concave cost function, σ is being increased by $\frac{1}{\sqrt{c_{\sigma}}}$. Our studies show that every time the algorithm failed to track the vehicle, there was a branch point.

5. CONCLUSIONS AND OUTLOOK

We have derived, implemented and tested a novel tracking algorithm. The algorithm extends known approaches for robust regression based motion estimation in the presence outliers. Tests show that the algorithm can tolerate high levels of outliers and high levels of gaussian noise added to inliers. Future work will be dedicated to handling branch-points, extending the model for the full affine parameter set and porting the computational costly algorithm to massively parallel hardware. e.g. graphics card hardware.

6. ACKNOWLEDGMENTS

We would like to thank Prof. Hans-Joachim Jentschel for supporting us with valuable suggestions and discussions on the topic of robust signal processing. This work has been granted by the Ministry of Science and Culture of the German State of Lower Saxony in the scope of the project OptiSiLK (Optimizing Safety and Capacity of Intersections).

REFERENCES

- [1] Black, M. J. and Anandan, P., “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Computer Vision and Image Understanding* **63**, 75–104 (1996).
- [2] Blake, A. and Zisserman, A., [*Visual Reconstruction*], MIT Press (1987).
- [3] Baker, S. and Matthews, I., “Lucas-kanade 20 years on: A unifying framework,” (2002).
- [4] Lucas, B. and Kanade, T., “An iterative image registration technique with an application to stereo vision,” in [*International Joint Conference on Artificial Intelligence*], (1981).
- [5] Shi, J. and Tomasi, C., “Good features to track,” in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*], (June 1994).
- [6] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., [*Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*], Cambridge University Press, New York, NY, USA (1992).
- [7] Huber, P. J., [*Robust Statistics*], Wiley, New York (1981).
- [8] Jodoin, J.-P., Bilodeau, G.-A., and Saunier, N., “Urban tracker: Multiple object tracking in urban mixed traffic,” in [*IEEE Winter Applications of Computer Vision Conference (WACV)*], 885–892, IEEE (Mar. 2014).
- [9] Bab-Hadiashar, A. and Suter., D., “Robust optic flow computation,” *International Journal of Computer Vision* **29** (1998).
- [10] Fischler, M. A. and Bolles, R. C., “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM* **24**, 381–395 (June 1981).
- [11] Lebeda, K., Matas, J., and Chum, O., “Fixing the locally optimized RANSAC,” (2012).
- [12] Hough, P., “Method and means for recognizing complex patterns,” (1962). US Patent 3069654.
- [13] Lowe, D. G., “Distinctive image features from scale-invariant keypoints,” (2003).
- [14] Bab-Hadiashar, A. and Suter, D., “Robust optic flow estimation using least median of squares,” in [*Proc. ICIP, Lausanne, Switzerland, Sept. 1996*], 513–516 (1996).
- [15] Isack, H. and Boykov, Y., “Energy-based geometric multimodel fitting,” tech. rep. (2010).
- [16] Mobahi, H. and Fisher III, J. W., “A theoretical analysis of optimization by gaussian continuation,” (2015).
- [17] Black, M., *Robust Incremental Optical Flow*, PhD thesis, Yale University, Department of Computer Science (1992).
- [18] Leich, A., Junghans, M., and Jentschel, H.-J., “Hough transform with GNC,” in [*12th European Signal Processing Conference (EUSIPCO)*], (2004).
- [19] Richter, M., “Untersuchungen zur Fahrzeugzählung in digitalen Bildfolgen unter Nutzung robuster Fehlerfunktionen,” (2003).
- [20] Golub, G. and van Loan, C., [*Matrix Computations*], The John Hopkins University Press (1996).
- [21] Meister, A., [*Numerik linearer Gleichungssysteme - Eine Einführung in moderne Verfahren*], Vieweg Verlag (1999).

- [22] Leich, A., [*Ein Beitrag zur Realisierung der videobasierten weiträumigen Verkehrsbeobachtung*], Dresdner Forschung / Verkehrswissenschaften, TUDpress (2006).
- [23] Dalal, N. and Triggs, B., “Histograms of oriented gradients for human detection,” in [*Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*], **1**, 886–893 vol. 1 (June 2005).