Foundations and Trends® in Robotics
Vol. 4, No. 4 (2013) 225–269

DOI: 10.1561/0600000047

now
the essence of knowledge

Robust Estimation and Applications in Robotics

Michael Bosse
Autonomous Systems Lab, ETH Zürich
mike.bosse@mavt.ethz.ch

Gabriel Agamennoni
Autonomous Systems Lab, ETH Zürich
gabriel.agamennoni@mavt.ethz.ch

Igor Gilitschenski
Autonomous Systems Lab, ETH Zürich
igilitschenski@ethz.ch

Contents

Abstract

Solving estimation problems is a fundamental component of numerous robotics applications. Prominent examples involve pose estimation, point cloud alignment, or object tracking. Algorithms for solving these estimation problems need to cope with new challenges due to an increased use of potentially poor low-cost sensors, and an ever growing deployment of robotic algorithms in consumer products which operate in potentially unknown environments. These algorithms need to be capable of being robust against strong nonlinearities, high uncertainty levels, and numerous outliers. However, particularly in robotics, the Gaussian assumption is prevalent in solutions to multivariate parameter estimation problems without providing the desired level of robustness.

The goal of this tutorial is helping to address the aforementioned challenges by providing an introduction to robust estimation with a particular focus on robotics. First, this is achieved by giving a concise overview of the theory on M-estimation. M-estimators share many of the convenient properties of least-squares estimators, and at the same time are much more robust to deviations from the Gaussian model assumption. Second, we present several example applications where M-Estimation is used to increase robustness against nonlinearities and outliers.

M. Bosse, G. Agamennoni and I. Gilitschenski. *Robust Estimation and Applications in Robotics*. Foundations and Trends® in Robotics, vol. 4, no. 4, pp. 225–269, 2013.

DOI: 10.1561/0600000047.

1

Introduction

Parameter estimation is the problem of inferring the value of a set of parameters through a set of noisy observations. Many tasks in robotics are formulated as an estimation problem. Most notable examples involve odometry, simultaneous localization and mapping (SLAM), or calibration. In case of odometry, the parameters often involve the sequence of robot poses and locations of landmarks that were seen (as in Leutenegger et al. (2015)). This is also true for SLAM, where additionally a map is built that can be used for later relocalization. For calibration, the estimated quantites usually involve the pose of a sensor and some of its internal parameters, e.g. the focal length of a camera lens. Since observations are subject to noise, the parameter estimate will always be afflicted with some level of uncertainty.

To model uncertainty, sensor and system noise are usually characterized by a probability distribution, one of the most common distributions being the Gaussian. Assuming Gaussian noise models leads to convenient simplifications due to its analytical properties and compact mathematical representation. Theoretically, the *central limit theorem* (CLT) is the main justification for the use of the Gaussian distribution.[1] The CLT can be applied in applica-

[1] The Gaussian distribution arises as the limit distribution of a sum of arbitrary independent, identically distributed random variables with finite variance.

tions where random variables are generated as the sum of many independent random variables. This assumption is known as the *hypothesis of elementary errors* and discussed in more detail in Fischer (2011). There are also several computational properties that make the Gaussian distribution an attractive choice. Namely, the fact that any linear combination of Gaussian random variables is Gaussian, and that the product of Gaussian likelihood functions is itself Gaussian. These properties allow additive Gaussian noise to be easily integrated into the parameter estimation framework of linear systems, where variables are assumed to be jointly Gaussian-distributed.[2]

Unfortunately, there is a tendency to invoke the Gaussian in situations where there is little evidence about whether or not it is applicable. Although the CLT provides a justification, to some extent and in some situations, the use of the Gaussian is rarely motivated by the nature of the actual stochastic process that generates the noise. There are situations that arise in practice which violate the CLT conditions. Many real-world systems contain strongly non-linear dynamics that destroy Gaussianity, since a non-linear transformation of a Gaussian random variable is not generally Gaussian-distributed. In certain applications the noise is multiplicative rather than additive, and the Gaussian assumption is inadequate due to the nature of the process.

The success of parameter estimation hinges on the assumptions placed on the noise distribution. Assuming a Gaussian distribution might still be a reasonable approximation even in the presence of non-linearity or non-additive noise, provided that the non-linearity is mild and the noise level is low. However, as these effects increase, there is neither a theoretical justification nor a practical advantage for using methods that rely on this assumption. If the Gaussian assumption is violated, then the parameter estimate may be misleading, which leads to the possibility of drawing incorrect conclusions about the parameter.

Outliers are a common type of a non-Gaussian phenomenon. An outlier may stem from hidden factors or characteristics that are intrinsic to the problem, but are tedious or otherwise impractical to model. Systems that rely on high-quality parameter estimates, such as robots, are especially sensitive to outliers. In certain cases, outliers can cause the system to fail catastrophically

[2]There are a number of other properties motivating the use of the Gaussian distribution. An introductory discussion of these properties can be found in Kim and Shevlyakov (2008).

to the point where a full recovery is no longer possible. For instance, a SLAM solution is vulnerable to false data associations, which may introduce strong biases or even lead to divergence in filter estimates.

Least-squares estimators are particularly prone to bias, outliers, or non-Gaussian noise. The squared-error loss is extremely sensitive, and its performance quickly degrades in the presence of these effects. The reason for this is that the estimator is an unbounded function of the residuals. From a probabilistic perspective, the Gaussian distribution is light-tailed, *i.e.* the tails of the Gaussian account for a very small fraction of the probability mass. This essentially rules out the possibility that an observation is wrong. Therefore, when a large discrepancy arises between the bulk of the observations and an outlier, the parameter estimate becomes an unrealistic compromise between the two.

The main goal of this tutorial is to make robust statistical tools accessible to the robotics community. Specifically, to provide the basis necessary for addressing the problems described above using M-estimators. Hence the contributions of this tutorial are twofold. On one hand, it provides an introduction to robust statistics that only requires preliminary knowledge of probability theory. In particular, the notion of random variables, probability distributions, probability density functions, and multi-variate linear regression are assumed to be known to the reader. On the other hand, this tutorial includes examples of robotics applications where robust statistical tools make a difference. It also includes corresponding Matlab scripts, and discusses how robust statistics improves parameter estimation in these examples.

The remainder of this tutorial is structured as follows. Chapter 2 gives an overview of the history and development of robust statistics and briefly discusses introductory material and existing applications in robotics. Chapter 3 starts with an overview of the challenges of non-linear least-squares estimation, and motivates the use of robust statistics for tackling some of these challenges. It also introduces basic concepts such as *loss functions*, and *iteratively re-weighted non-linear least-squares*. Chapter 4 describes qualitative and quantitative criteria for characterizing the robustness of M-estimators and provides definitions of concepts such as estimator bias, the influence function and the breakdown point are found here. Chapter 5 presents example applications that illustrate the advantage of using robust estimation in robotics.

Specifically, robust approaches to pose graph optimization, parameter estimation under non-Gaussian noise, and state-estimation in the presence of outliers and biases. Finally, chapter 6 concludes with a discussion of further reading and applications of robust statistics to robotics.

2

Related Work

Since the 1960s, researchers have been working on ways to make statistical analysis procedures resilient to deviations from their idealized assumptions. The seminal work by Huber (1964) laid the foundation of what has come to be known as "robust statistics", (see also Hampel (1992); Hampel et al. (1986)) an extension of classical statistics that takes into account the fact that parametric models are but an approximation of reality. Classical statistical analyses behaves quite poorly under violations of the underlying assumptions. The goal of robust statistics is to develop procedures that are still reliable under small deviations, *i.e.* when the true distribution lies in a neighborhood of the assumed model. Peter J. Huber was the first to develop the notion of approximate validity of a parametric model and extend the theory in this direction. In doing so, Huber introduced key concepts and derived useful practical tools that became standards in robust statistics and its applications.

One of the most significant implications of Huber's work was that it offered a way of dealing with outliers that is more general and well-behaved than classical approaches. The typical approach to safeguarding an estimator against outliers is to remove them entirely, by trimming observations that deviate somehow from the bulk of the data. For example, the Chi-square test, which is sometimes used as preprocessing before Kalman filtering, rejects ob-

servations that lie outside of the ellipsoid containing the majority, *e.g.* 99%, of the probability mass of the predictive distribution. The Random Sample Consensus (RANSAC) algorithm, discussed in Fischler and Bolles (1981), is another approach, that discards observations considered as outliers. It is based on randomly selecting a minimal subset of datapoints for parameter estimation and subsequently checking whether the resulting model produces more inliers than the best previous iteration. Unlike the methods discussed in this work, the outcome of RANSAC is not deterministic and might therefore be not reproducible. Additionally, rejecting observations, poses a number of problems. First of all, there is a loss of information due to the reduced sample size. Second, detecting outliers is inherently difficult, particularly in high-dimensional spaces where number of data is of the same order as the number of dimensions. Thus, it may be desirable to develop a notion of outliers that do not fully discard a potentially outlying datapoint but merely reduce its influence within the inference procedure. Robust statistical estimators provide ways of automatically dealing with outliers without necessarily discarding them. Also the decision happens gradually and can therefore be revised during iterations. These estimators, known as M-estimators, play a central role in modern robust statistics.

2.1 M-Estimators

An M-estimator is nothing more than a maximum-likelihood estimator, albeit for a non-standard parametric model. M-estimators arise as solutions of certain optimization problems (Huber, 1981; Hampel et al., 1986; Rousseeuw and Leroy, 1987; Staudte and Sheather, 1990). The objective function, known in this context as the *loss function*, is designed so that the resulting estimator has a number of desirable properties. Namely, a redescending *influence curve*, a small *gross error sensitivity* and a large *breakdown point*. Each of these terms has a concrete mathematical definition, which will be given in chapters 3 and 4. Intuitively, the influence curve describes the sensitivity of the overall estimate with respect to the data, the gross error sensitivity quantifies robustness as the maximum effect that an observation has on the estimator, and the breakdown point is the maximum proportion of outliers beyond which the estimator may develop an arbitrarily large bias.

2.2 M-Estimation in Robotics

There is a number of tutorial-style presentations related to estimation in robotics. In Zhang (1997), a number of different parameter estimation techniques (involving robust approaches) are introduced putting a strong emphasis on vision applications. Stewart (1999) also considers vision applications and restricts the focus to robust approaches. A discussion of robust estimation with focus on signal processing is presented in Zoubir et al. (2012). A more fundamental discussion of robust statistics and influence functions is given in Huber (1981); Hampel et al. (1986).

An approach based on Dynamic Covariance Scaling (DCS) is proposed in Agarwal et al. (2013a,b), It can be thought of as a generalization of classical gating by dynamically rejecting potential outliers. Carlone et al. (2014) proposed an outlier detection approach that is based on ℓ_1 relaxation and a linear programming problem formulation. Furthermore, mixture models have also been proposed in Olson and Agarwal (2012, 2013) to account for the fact that errors might be non-Gaussian. This approach also provides a stochastic model that considers potentially wrong loop-closures rather than a priori classifying them. An approach that uses switchable constraints was discussed in Sünderhauf and Protzel (2013, 2012a,b) where the topology of the pose graph is optimized as part of the entire optimization problem. Hee Lee et al. (2013) discusses an approach that assigns weights to each loop-closure constraint and then uses an expectation maximization at each iteration of the graph-optimizer in order to assign a low weight to outlier constraints. Another approach using expectation maximization for robust Kalman filtering is discussed in Ting et al. (2007). Removing the Gaussian assumption in sparse factor graphs is discussed in Rosen et al. (2013). Agamennoni et al. (2011) proposes a robust approach to inference of principal road paths. A combination of RANSAC and M-Estimation was presented by Torr and Murray (1997) and applied to fundamental matrix estimation. Further applications making use of robust methods are presented in Loxam and Drummond (2008); Kerl et al. (2013); Zach (2014).

3

Basic Concepts

Many estimation problems in robotics can be formulated as solving a non-linear least-squares problem of residuals, $\mathbf{r}$, in the form

$$\hat{\theta} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{n} \|\mathbf{r}_k(\boldsymbol{\theta})\|^2 \tag{3.1a}$$

$$\mathbf{r}_k(\boldsymbol{\theta}) = \mathbf{z}_k - \mathbf{h}_k(\boldsymbol{\theta}) \tag{3.1b}$$

where the $\mathbf{z}_k$ are indirect, noisy measurements of the unknown parameter $\boldsymbol{\theta}$. The parameter is usually high-dimensional, and is observed indirectly via a set of non-linear, low-dimensional projections $\mathbf{h}_k(\boldsymbol{\theta})$. The problem is often solved with a non-linear optimization method such as Gauss-Newton or Levenberg-Marquardt. In most cases it is safe to assume that, for optimization purposes, the parameter, $\boldsymbol{\theta}$, lies in Euclidean space.[1]

Estimation problems such as these arise frequently in robotics, *e.g.* pose graph optimization and bundle adjustment. For instance, in bundle adjustment, $\boldsymbol{\theta}$ is a set of landmark positions and camera configurations, the $\mathbf{z}_k$ are

[1] In certain cases, *e.g.* when estimating poses or transformations, $\boldsymbol{\theta}$ may contain elements such as rotation matrices and quaternions. These elements lie on manifolds. In these cases, assuming that the manifold has a differential structure, it is possible to parameterize $\boldsymbol{\theta}$ locally so that (3.1) is expressed as a function of local coordinates that lie in Euclidean space.

image keypoint observations, and $\mathbf{h}_k(\boldsymbol{\theta})$ are the projections of the landmarks into their corresponding images. In pose graph optimization, $\boldsymbol{\theta}$ is a block vector containing the trajectory of robot poses, and the $\mathbf{h}_k(\boldsymbol{\theta})$ express pairwise pose constraints with $\mathbf{z_k} = 0$. The solution to (3.1) is a least-squares estimate of the robot pose trajectory, or the landmark positions and camera configurations.

3.1 Why Non-Linear Least-Squares is Hard

Non-linear least-squares estimation is hard, especially for large problems. When the parameter is high-dimensional, solving (3.1) can be challenging, since the optimization method becomes computationally expensive and prone to numerical instability and local minima. Large-scale numerical solvers (Dellaert and Kaess, 2006; Agarwal et al., 2010; Kummerle et al., 2011) leverage the structure and sparsity of the optimization problem to maximize computational efficiency. Subspace methods (Eriksson and Wedin, 2004; Gratton et al., 2007; Maye et al., 2013) improve numerical stability by searching along directions in parameter space where the sum-of-squares error function is numerically well-conditioned. Local minima are much harder to deal with, and there is yet no general approach that guarantees finding the global optimum, from any starting point that at the same time scales well with dimension.

On the other hand, least-squares estimators are not robust. This is not a fault in the estimator itself, but rather the least-squares criterion. The sum-of-squares error metric is overly sensitive to data contamination and model misspecification —this stems from the fact that the estimator is an unbounded function of the residuals. Hence the accuracy degrades quickly when the data contain outliers, or the noise is non-Gaussian. In fact, the least-squares estimator is so sensitive that a single outlier is enough to introduce an arbitrarily large bias in the final solution.[2]

[2]It is important to note that, even under correct model assumptions, the non-linear least-squares estimator is generally biased. That is, the expected value of the estimator is different from the true value of the parameter. On the upside, though, the non-linear least-squares estimator is *weakly consistent*, meaning that, under certain mild conditions the sequence of estimates converges in probability to the true parameter in the limit $n \to \infty$. In other words, as the number n of data increases, the probability that the estimated parameter and the true parameter are arbitrarily close becomes asymptotically equal to one.

Least-squares' sensitivity can also be interpreted from a statistical perspective. From (3.1), the sum-of-squares error is equal, up to an additive constant, to the negative log-likelihood of a Gaussian distribution. Hence the least-squares estimator is the maximum-likelihood estimator under a Gaussian model,

$$Z_k \sim \mathcal{N}\left(\mathbf{h}_k(\boldsymbol{\theta}), \mathbf{I}\right) \tag{3.2}$$

where $\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ stands for a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and Z_k denotes a random variable.[3] In other words, it solves

$$\arg\max_{\boldsymbol{\theta}} \prod_{k=1}^{n} \mathcal{N}(\mathbf{z}_k; \mathbf{h}_k(\boldsymbol{\theta}), \mathbf{I}) \tag{3.3}$$

which, as $\mathcal{N}(\mathbf{z}_k; \mathbf{h}_k(\boldsymbol{\theta}), \mathbf{I})$ denotes the p.d.f. of a $\mathcal{N}(\mathbf{h}_k(\boldsymbol{\theta}), \mathbf{I})$ distribution evaluated at $\mathbf{z}_k$, is equivalent to

$$\arg\max_{\boldsymbol{\theta}} \ \exp\left(-\frac{1}{2}\sum_{k=1}^{n} \|\mathbf{z}_k - \mathbf{h}_k(\boldsymbol{\theta})\|^2\right) .$$

Choosing the covariance $\boldsymbol{\Sigma} = \mathbf{I}$ is merely for simplified presentation. It, however, is not a restrictive assumption as the rescaling of the covariance matrix may be encoded as part of $\mathbf{h}_k(\boldsymbol{\theta})$.

Since the Gaussian distribution places most of its probability mass in a small region around the mean, it cannot account for outliers. That is, it is unable to explain realizations of Z_k that are far away from the mean $\mathbf{h}_k(\boldsymbol{\theta})$, and thus does not conform to the Gaussian model.[4]

3.2 Loss Functions and Robust Estimation

M-estimators aim to reduce the sensitivity of least-squares by replacing the sum-of-squared error with another, more robust criterion. Namely, an M-estimator replaces (3.1) with

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{n} \rho\left(\|\mathbf{r}_k(\boldsymbol{\theta})\|^2\right) \tag{3.4}$$

[3] Note that a clear distinction is made between the random variables Z_k, and their realizations $\mathbf{z}_k$.

[4] For example, under a one-dimensional Gaussian model, an outlier 5 standard deviations away from the mean has less than *one in a million* chances of occurring. Although the probability is non-zero, it is unrealistically small.

where $\rho : t \mapsto \rho(t) \geq 0$ is a non-negative, non-decreasing function, usually with a unique minimum at $t = 0$. Note that t is the *squared* magnitude of the residual. This function is called the *loss function*, and (3.4) is a robust non-linear least-squares problem.[5]

Intuitively, the role of the loss function is to reduce the impact of outliers on the solution of (3.4). For example, the Huber loss function, defined with the scale parameter s as

$$\rho(t) = \begin{cases} t & t \leq s \\ 2\sqrt{st} - s & t > s \end{cases} \quad \text{with} \quad s > 0$$

is linear for t close to zero, and sub-linear for t far away from zero. The idea is that, if a squared error is extremely large, then it is most likely an outlier. Instead of discarding it, the Huber loss reduces the penalty to avoid biasing the estimator. This is in contrast to least-squares, where $\rho(t) = t$ and, therefore, larger and larger errors receive greater and greater penalties. Some common loss functions (and their derivatives, whose role will be explained subsequently) are listed in Table 3.1 and visualized in Figure 3.1.

The "M" in the term "M-estimator" stands for maximum-likelihood-type estimator Huber (1964). This name stems from the fact that an M-estimator can be loosely interpreted as a maximum-likelihood estimator, albeit for an unknown, non-Gaussian model. Agamennoni et al. (2015) showed that, in certain cases, this model follows an elliptical distribution Fang et al. (1987). In these cases an M-estimator is a maximum-likelihood estimator under an elliptical model,

$$Z_k \sim \mathcal{E}\left(\mathbf{h}_k(\boldsymbol{\theta}), \mathbf{I}, \rho\right) \tag{3.5}$$

where $\mathcal{E}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho)$ denotes an elliptical distribution with location $\boldsymbol{\mu}$, scale $\boldsymbol{\Sigma}$ and loss function ρ. That is, it solves

$$\arg\max_{\boldsymbol{\theta}} \prod_{k=1}^{n} f_k(\mathbf{z}_k) \propto \exp\left(-\frac{1}{2} \min_{\boldsymbol{\theta}} \sum_{k=1}^{n} \rho\left(\left\|\mathbf{z}_k - \mathbf{h}_k(\boldsymbol{\theta})\right\|^2\right)\right) \tag{3.6}$$

[5]There are different definitions of the loss function in the literature. Classic approaches, which deal with one-dimensional data, define the loss as a symmetric function of the *residual* r_k. For generality, in this tutorial the loss is defined as a function of the *squared error*, *i.e.* the squared of the residual, r_k^2, which in the case of multi-dimensional measurements simply becomes the squared norm $\|\mathbf{r}_k\|^2$.

Table 3.1: Commonly used M-estimators

	$\rho(t)$	$\rho'(t)$
Gaussian	t	1
Laplace	$2\sqrt{st}$	$\sqrt{\frac{s}{t}}$
Huber	$\begin{cases} t & t \leq s \\ 2\sqrt{st} - s & t > s \end{cases}$	$\min\left\{1, \sqrt{\frac{s}{t}}\right\}$
"Fair"	$2s\left(\sqrt{\frac{t}{s}} - \ln\left(1 + \sqrt{\frac{t}{s}}\right)\right)$	$\frac{1}{1+\sqrt{t/s}}$
Cauchy	$s \ln\left(1 + \frac{t}{s}\right)$	$\frac{1}{1+t/s}$
Geman-McClure	$s\left(1 - \frac{1}{1+t/s}\right)$	$\frac{1}{(1+t/s)^2}$
Welsch	$s\left(1 - \exp\left(-t/s\right)\right)$	$\exp\left(-t/s\right)$

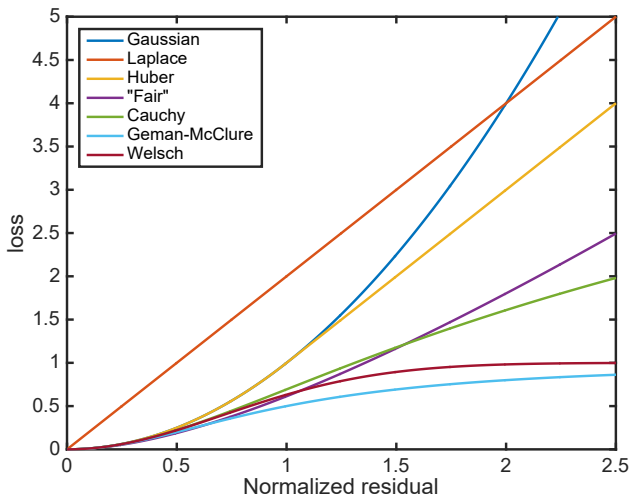

Figure 3.1: Loss curves for the M-estimators listed in table 3.1. To give a better intuition for the difference compared to least-squares, we maintained the squaring within the loss curve. That is, this plot shows $\rho(t^2)$.

compare these last two equations with equations (3.2) and (3.3) in section 3.1. See Agamennoni et al. (2015) for details on when the equivalence between M-estimators and elliptical distributions holds.

3.3 Iteratively Re-Weighted Non-Linear Least-Squares

An M-estimator maps a set of data $\mathbf{z}_k$ and a loss function ρ to a parameter estimate $\hat{\boldsymbol{\theta}}$, which solves the M-estimation problem (3.4). The M-estimation problem inherits many of the challenges of the least-squares problem. However, the solution, once found, is much more robust to deviations from idealized model assumptions. Qualitative and quantitative robustness will be discussed in more detail in chapter 4. The remainder of this chapter will briefly describe two widely used non-linear optimization methods for tackling (3.4). Namely, the Gauss-Newton and Levenberg-Marquardt methods. A thorough derivation of these methods is outside the scope of this tutorial. For an excellent introduction to numerical optimization, refer to Nocedal and Wright (1999).

Broadly speaking, there are essentially two families of iterative methods for finding a local minimum of a non-linear function such as (3.1) and (3.4). These are: line search methods and trust region methods. The main difference between them lies in their notion of "locality" and the way they enforce it, either by explicitly controlling the step size, or by regularizing the objective function. Gauss-Newton and Levenberg-Marquardt, respectively, belong to the families of line search and trust region methods.

Both types of methods solve (3.4) in an iterative fashion by repeatedly refining an estimate of the parameter $\boldsymbol{\theta}$. Starting from an initial guess, the estimate is updated by applying a sequence of update steps

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \, \Delta\boldsymbol{\theta} \tag{3.7}$$

where $\Delta\boldsymbol{\theta}$ is the update step at a given iteration, and α is the step size. The update step is computed by solving a linearized version of the original, non-linear problem around the current estimate. Specifically, by solving a weighted least-squares problem of the form

$$\min_{\Delta\boldsymbol{\theta}} \frac{1}{2} \left(\mathbf{J}\Delta\boldsymbol{\theta} - \mathbf{r}\right)^\top \mathbf{W} \left(\mathbf{J}\Delta\boldsymbol{\theta} - \mathbf{r}\right) + \frac{\lambda}{2} \left\|\Delta\boldsymbol{\theta}\right\|^2 \tag{3.8}$$

where $\mathbf{r}$ and $\mathbf{J}$ are the residual vector and Jacobian matrix, respectively, and $\mathbf{W}$ is a diagonal weight matrix. The scalar λ is a damping constant, and the step size α is chosen so that $\boldsymbol{\theta} + \alpha\,\Delta\boldsymbol{\theta}$ leads to a sufficiently large decrease in the overall loss. The vector and matrices appearing in (3.8) and (3.10) are given by

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{bmatrix} \qquad \mathbf{J} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_n \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w_1\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & w_n\mathbf{I} \end{bmatrix}$$

where $\mathbf{r}_k$ is the residual, and $\mathbf{J}_k$ and w_k are, in that order, the Jacobian and the weight for observation $\mathbf{z}_k$,

$$\mathbf{J}_k(\boldsymbol{\theta}) = \frac{\partial \mathbf{h}_k}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{3.9a}$$

$$w_k(\boldsymbol{\theta}) = \rho'\Big(\|\mathbf{r}_k(\boldsymbol{\theta})\|^2\Big) \tag{3.9b}$$

Note that evaluating $\mathbf{J}_k$ and w_k involve evaluating the first derivative of $\mathbf{h}_k$ and ρ, respectively. Hence it is implicitly assumed that $\mathbf{h}_k$ and ρ are continuously differentiable. Intuitively, the weighting function can be thought of as a correction ensuring the consideration of the robust loss in the underlying least-squares problem. More formally, it is a consequence of the chain-rule when applied to the nonlinear considered robust cost function.

The basis for solving (3.4) is to approximate the non-linear model locally by a weighted linear model. Each iteration is as follows. First, the residual vector and the Jacobian and weight matrices are evaluated based on the current estimate of $\boldsymbol{\theta}$. Then, the update step $\Delta\boldsymbol{\theta}$ is computed by solving the weighted least-squares problem (3.8). And finally, the estimate is updated according to (3.7). Note that (3.8) is a linear least-squares problem, and so it can be solved in closed form. The solution —which may or may not be unique—satisfies the damped normal equations,[6]

$$\Big(\mathbf{J}^\top\mathbf{W}\mathbf{J} + \lambda\mathbf{I}\Big)\Delta\boldsymbol{\theta} = \mathbf{J}^\top\mathbf{W}\mathbf{r} \tag{3.10}$$

Solutions to these equations are often computed via specialized methods that exploit the structure and sparsity of the problem, and scale well when $\boldsymbol{\theta}$ is high-dimensional (Saad, 2003; Davis, 2006).

[6]The normal equations are so-called because they state that the residual must be normal, i.e. orthogonal, to the columns of the weighted Jacobian matrix.

The Gauss-Newton and Levenberg-Marquardt methods are special cases of the method just described, with specific policies for choosing α and λ. In Gauss-Newton, the damping factor is set to $\lambda = 0$, while the step size α is computed by searching along the line defined by $\Delta\boldsymbol{\theta}$, a procedure known as line-searching (Armijo, 1966; Nocedal and Wright, 1999). For Levenberg-Marquardt, the step size is fixed at $\alpha = 1$ and the damping factor λ is chosen according to how much the overall loss decreases in the linear vs. the non-linear problems Conn et al. (2000).

4

Theoretical Background on M-Estimation

The quality of an estimator, under the ideal model assumptions, is usually measured in terms of its bias and efficiency. An estimator it unbiased if, on average, the parameter that it estimates matches the true parameter, and it is efficient if, out of all possible estimators, it has the lowest uncertainty. The ideal model assumptions, however, rarely hold in practice. Near-optimality under slight deviations from these assumptions is measured quantitatively by the gross error sensitivity and breakdown point. Desirable properties of an M-estimator are a high efficiency in a neighborhood of the ideal model, and a large gross error sensitivity and breakdown point. There is always a compromise between an estimator's quality and robustness, and M-estimation provides a mechanism for striking a trade-off between the two.

Before moving on to the definitions, some notation is in order. Let $\boldsymbol{\eta}$ denote an M-estimator of $\boldsymbol{\theta}$ with loss function ρ. $\boldsymbol{\eta}$ is a mapping from a set $\{\mathbf{z}_k\}$ of observations to the solution of (3.4),

$$\boldsymbol{\eta}\left(\mathbf{z}\right) = \arg\min_{\boldsymbol{\theta}} \ell\left(\mathbf{z};\boldsymbol{\theta}\right) \tag{4.1}$$

where ℓ is the robust sum-of-squares function,

$$\ell\left(\mathbf{z};\boldsymbol{\theta}\right) = \frac{1}{2}\sum_{k=1}^{n} \rho\left(\left\|\mathbf{z}_k - \mathbf{h}_k(\boldsymbol{\theta})\right\|^2\right) \tag{4.2}$$

and $\mathbf{z}$ is the block-vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_n \end{bmatrix}$$

Let Z be the random variable formed by grouping the Z_k into a vector, where the Z_k is the random variable such that $\mathbf{z}_k$ is the realization of Z_k.

Note that the M-estimator can be regarded as a random variable, as it is function of the observations, and the observations themselves are realizations of random variables. As such, $\boldsymbol{\eta}$ has a probability distribution. The distribution of $\boldsymbol{\eta}$, often referred to as the *sampling distribution*, serves as a basis for performing statistical tests and making decisions. In the remainder of this chapter, $\boldsymbol{\eta}(Z)$ will denote the M-estimator as a random variable, and the distribution over the M-estimator induced by Z will be referred to the sampling distribution of $\boldsymbol{\eta}$.

Classical terminology from estimation, also applies when using M-Estimators. Particularly, the CramÃl'r-Rao bound can be used to characterize a lower bound on the error variance of the estimator. Its definition requires revisiting the concept of biased estimators and the Fisher Information Matrix first.

Definition 4.1 (Bias)**.** The bias of $\boldsymbol{\eta}$ is defined as the difference between the mean of its sampling distribution, and $\boldsymbol{\theta}$. That is,

$$\mathcal{B}_{\boldsymbol{\theta}}(\boldsymbol{\eta}) = \mathrm{E}_Z[\boldsymbol{\eta}(Z)] - \boldsymbol{\theta} \tag{4.3}$$

where $\mathrm{E}_Z[\,\cdot\,]$ denotes the expectation with respect to Z. Hence $\boldsymbol{\eta}$ is unbiased if $\mathcal{B}_{\boldsymbol{\theta}}(\boldsymbol{\eta}) = \mathbf{0}$.

The bias is the extent to which the estimator differs from the parameter in a systematic manner, *i.e.* non-randomly. Intuitively, it quantifies how accurate the estimator would be, on average, if it were used repeatedly on different sets of observations.

Definition 4.2 (Fisher information matrix)**.** The Fisher information matrix of $\boldsymbol{\theta}$ is defined as the covariance matrix of the score.[1] That is,

$$\mathcal{I}_{\boldsymbol{\theta}} = \mathrm{E}_Z\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln f\left(Z\right) \frac{\partial}{\partial \boldsymbol{\theta}} \ln f\left(Z\right)^\top\right]$$

where f is the probability density function of the distribution over Z.

The Fisher information matrix is a function of the model. It can be loosely interpreted as a quantitative measure of how observable $\boldsymbol{\theta}$ is, on average. For any given set of observations, if the log-likelihood is peaked around $\boldsymbol{\theta}$, then the observations carry a lot of information about $\boldsymbol{\theta}$. On the other hand, if the log-likelihood is spread out, then the observations are ambiguous about $\boldsymbol{\theta}$. The Fisher information matrix $\mathcal{I}_{\boldsymbol{\theta}}$ quantifies this notion of curvature of the log-likelihood function, and averages over all possible sets of observations.

Theorem 4.1 (Cramér-Rao lower bound)**.** The covariance matrix of $\boldsymbol{\eta}$ around $\boldsymbol{\theta}$ is bounded from below,[2]

$$\mathrm{E}_Z\left[\left(\boldsymbol{\eta}\left(Z\right) - \boldsymbol{\theta}\right)\left(\boldsymbol{\eta}\left(Z\right) - \boldsymbol{\theta}\right)^\top\right] \\ \succeq \left(\mathbf{I} + \frac{\partial \mathcal{B}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right) \mathcal{I}_{\boldsymbol{\theta}}^{-1} \left(\mathbf{I} + \frac{\partial \mathcal{B}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\right)^\top + \mathcal{B}_{\boldsymbol{\theta}} \mathcal{B}_{\boldsymbol{\theta}}^\top$$

where $\mathcal{B}_{\boldsymbol{\theta}}$ and $\mathcal{I}_{\boldsymbol{\theta}}$ are as defined in 4.1 and 4.2, respectively. This bound is also known as the Fréchet-Cramér-Rao lower bound.

Proof. For a proof, refer to a standard statistics textbook, or to the original papers Rao (1945); Cramér (1946). □

The Cramér-Rao bound establishes a limit on the expected squared error between an estimator and the true parameter. It states that, for a given number of observations, the variance of the estimator cannot be arbitrarily small. For the special case where θ is scalar and η is unbiased, the bound becomes

$$\mathrm{Var}_Z[\eta\left(Z\right)] \geq \mathcal{I}_\theta^{-1}$$

[1] The score is the gradient of the log-likelihood function with respect to the parameter. It is possible to show, under mild assumptions, that $\mathrm{E}_Z\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln f\left(Z\right)\right] = \mathbf{0}$, *i.e.* that the score has zero mean. Hence the Fisher information matrix is the covariance matrix of the score.

[2] The notation $\mathbf{A} \succeq \mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is positive semi-definite.

since $\mathrm{E}_Z[\eta(Z)] = \theta$. In this case, the *efficiency* of η is defined as

$$\mathrm{Eff}(\eta) = \frac{\mathcal{I}_\theta^{-1}}{\mathrm{Var}_Z[\eta(Z)]} \tag{4.4}$$

By definition, an estimator cannot have an efficiency greater than 100%. The only estimator with an efficiency of 100% is the minimum-variance unbiased estimator.

4.1 The Influence Curve

The influence function is a design tool that can be used to visually assess the robustness of an M-estimator. Intuitively, it quantifies how the estimator would react if a small perturbation was added to a single observation. Formally, it is defined as the asymptotic bias caused by contaminating an observation by an infinitesimal amount, standardized by the probability mass of the contamination. The gross error sensitivity, a quantitative measure of robustness, is defined in terms of the influence function.

It is possible to derive an analytic expression for the influence function of $\boldsymbol{\eta}$ in (4.1) by applying the measure-theoretic definition in Hampel (1974), and following the steps in Chapter 4 of Neugebauer (1996) for a non-linear least-squares regression estimator. This leads to

$$\mathrm{IF}(\mathbf{z}; \boldsymbol{\eta}) = \mathrm{E}_Z\left[\frac{\partial^2 \ell}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top}(Z; \boldsymbol{\eta}(Z))\right]^{-1} \frac{\partial \ell}{\partial \boldsymbol{\theta}}(\mathbf{z}; \boldsymbol{\eta}(\mathbf{z})) \tag{4.5}$$

This expression is general and holds for any M-estimator of the form (4.1) where ρ' and the $\mathbf{h}_k$ are continuously differentiable.

It is interesting to consider special instances of the estimation problem, where the influence function takes a much simpler form. For instance, in the linear regression problem $\mathbf{h}_k(\boldsymbol{\theta}) = \mathbf{x}_k^\top \boldsymbol{\theta}$ for all $k = 1, \ldots, n$ and the z_k are scalar. In this case the influence function simplifies to

$$\mathrm{IF}(\mathbf{z}; \boldsymbol{\eta}) = \mathrm{E}_Z\left[\sum_{k=1}^{n} Z_k Z_k^\top\right]^{-1} \sum_{k=1}^{n} w_k r_k|_{\boldsymbol{\theta}=\boldsymbol{\eta}(\mathbf{z})}\, \mathbf{z}_k \tag{4.6}$$

where the notation $\cdot|_{a=b}$ denotes evaluated at $a = b$. Equation (4.6) states that, in a linear model, the influence exerted by a single observation y_k is proportional to the weighted residual $w_k r_k$. For the special case of a Gaussian

model, the $w_k = 1$ since $\rho(t) = t$, and thus the influence is a linear, unbounded function of the residuals. For a detailed derivation of the influence function for a linear measurement model with scalar observations, see Cheng (1992).

The definition of the influence function can be simplified, when considering the simplest possible problem, i.e. assuming a scalar $\theta \in \mathbb{R}$ and a measurement model $h_k(\theta) = \theta$ that directly measures θ. These simplifications allow the influence function to be expressed as a function of the residual, or the squared error. Replacing the derivatives of ℓ in (4.5) and separating out the contribution of a single residual r leads to

$$\mathrm{IF}(r;\eta) \propto \rho'\left(r^2\right) r \tag{4.7}$$

This is the form of the influence function defined in most tutorials and introductory textbooks on M-estimation.

The derivative ρ' of ρ is known as the weight function, and s is a tuning constant that controls the scale of the loss function.[3] Note that some M-estimators have influence functions that tend to zero as $r \to \infty$; in other words, as the residual becomes increasingly large, the observation is eventually ignored. These are known as *redescending* M-estimators. Examples of influence curves are shown in Figure 4.1.

4.2 Gross Error Sensitivity

The gross error sensitivity of an estimator is defined as the maximum norm of the influence curve. It is a global robustness criterion, as it expresses the maximum effect that an outlier —a gross error— can have on the estimator. Formally, the gross error sensitivity is defined as

$$\mathrm{GES}(\boldsymbol{\eta}) = \sup_{\mathbf{z}} \|\mathrm{IF}(\mathbf{z};\boldsymbol{\eta})\| \tag{4.8}$$

For the special case of a scalar parameter and a linear location model, this expression simplifies to

$$\mathrm{GES}(\eta) \propto \sup_{t>0} \rho'(t)\sqrt{t} \tag{4.9}$$

[3] Agamennoni et al. (2015) propose a method for automatically tuning these constants to a set of data, based on an equivalence between certain types of M-estimators and elliptical distributions.

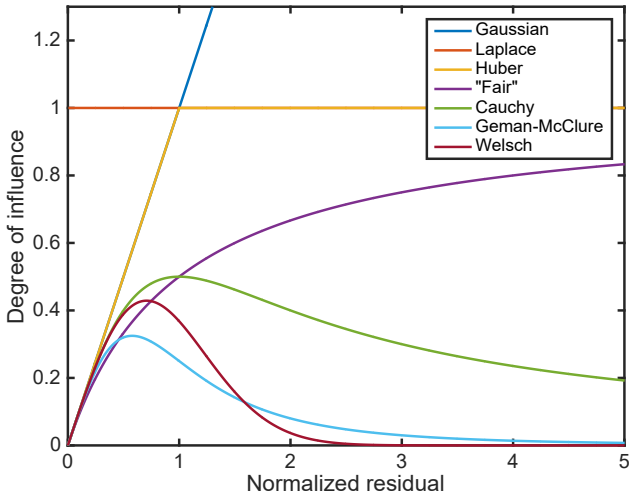

Figure 4.1: Influence curves for the M-estimators listed in table 3.1, under a linear location model. The influence function for a linear location model is given by (4.7). All of the influence functions shown here are bounded from above. The upper bound is the gross error sensitivity.

This is the expression adopted by most tutorial-level textbooks and introductory material. Note that the gross error sensitivity can be infinite; in fact, this is the case for the Gaussian distribution.

4.3 The Maximum Bias Curve

The maximum bias function is the maximum possible bias for a given proportion of outliers. In other words, suppose that a fraction ϵ of the observations are contaminated, *i.e.* are generated from a different distribution than the rest. The maximum bias is equal to the maximum value of $\|\mathcal{B}_{\boldsymbol{\theta}}(\boldsymbol{\eta})\|^2$ over all possible contaminating distributions. Formally,

$$\mathrm{MB}\left(\epsilon;\boldsymbol{\eta}\right)=\sup_{g}\left\{\|\mathcal{B}_{\boldsymbol{\theta}}(\boldsymbol{\eta})\|:Z\sim\left(1-\epsilon\right)\mathcal{F}+\epsilon\,\mathcal{G}\right\} \tag{4.10}$$

where $\mathcal{F}$ is the uncontaminated, probability distribution over Z with probability density function f, and $\mathcal{G}$ is the contaminating distribution with density g.[4]

[4]Hence the notation $\left(1-\epsilon\right)\mathcal{F}+\epsilon\,\mathcal{G}$ denotes a joint probability distribution over Z with probability density function $\left(1-\epsilon\right)f\left(\mathbf{z}\right)+\epsilon\,g\left(\mathbf{z}\right)$.

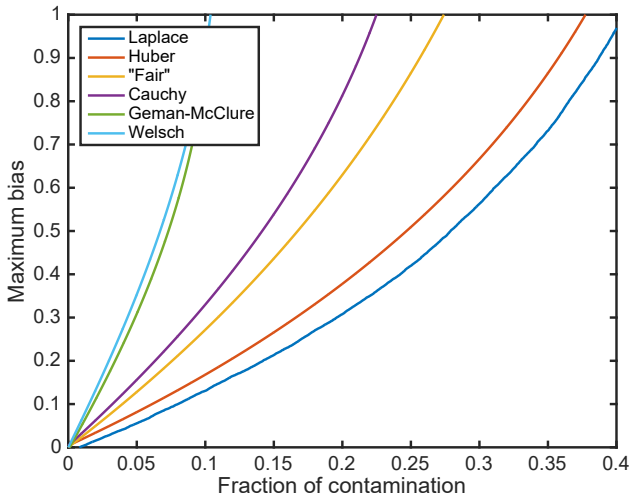

Figure 4.2: Maximum-bias curves for the M-estimators listed in table 3.1, under a linear location model. The maximum-bias function for a linear location model is given by (4.11a) and (4.11b). Beyond a certain fraction of contamination the maximum bias becomes infinite. This is the breakdown point.

For the special case where the parameter is scalar and the model is a linear location model, the maximum bias function becomes

$$\mathrm{MB}\left(\epsilon;\eta\right) = \left|\theta_{\epsilon}\right| \tag{4.11a}$$

where θ_{ϵ} is the solution to the implicit equation

$$\left(1-\epsilon\right)\mathrm{E}_{Z}\left[\rho'\left(\left(Z-\theta\right)^{2}\right)\left(Z-\theta\right)\right]+\epsilon=0 \tag{4.11b}$$

Refer to Martin and Zamar (1993) for a detailed derivation of the maximum bias function for a slightly more general model.

Figure 4.2 plots the maximum-bias curves for the M-estimators in table 3.1. The curve is plotted for values of ϵ in between 0% and 50%, since beyond 50% it becomes impossible to distinguish between the nominal and the contaminating distributions. For each loss function, the curve is plotted by approximating the expectation in (4.11b) by a sample average, and solving the implicit equation with an iterative root-finding method. Note that, beyond a certain ϵ, the maximum bias becomes infinite.

4.4 The Breakdown Point

The breakdown point is a quantitative indicator of robustness. It indicates the maximum proportion of outliers that the estimator can handle without resulting in an arbitrarily large bias. Formally, the breakdown point is is defined in terms of the maximum bias curve as

$$\mathrm{BP}\left(\boldsymbol{\eta}\right) = \sup_{0 \leq \epsilon < 1/2} \left\{\epsilon : \mathrm{MB}\left(\epsilon; \boldsymbol{\eta}\right) < \infty\right\} \tag{4.12}$$

A higher breakdown point corresponds to higher robustness. For the sample mean the breakdown point is 0%, while for the sample median it is 50%, which is the maximum possible value.

The breakdown point can be read directly from the maximum-bias curve. For instance, from figure 4.2, both the Huber and Laplace M-estimators have a breakdown point of 50%, since their maximum-bias curves become infinite at $\epsilon = 1/2$. On the other hand, the Cauchy and Welsch M-estimators have lower breakdown points —around 25% and 10%, respectively.

5

Robust Estimation in Practice

So far, we have established the theoretical framework for robust estimation. In this chapter, we want to take a closer look at applications that may benefit from these robust methods. The considered examples, cover different application areas within robotics and consider three different challenges. Outlier removal, consideration of non-Gaussian noise, and improving the convergence basin for non-linear optimization. As example applications of robust estimation, we consider simple polynomial least-square problems, the ICP point-cloud alignment algorithm, and pose graph optimization. All examples were implemented in Matlab and are made available within the supplementary material.

5.1 Outlier Removal

Situations in which sensors or some of the subsequent processing systems fail often result in measurement outliers. Thus, they are typically not accounted for by the noise models used for processing their signals. A simple way to address this, is throwing away measurements which suffers from the problems discussed above. Use of robust statistics offers a systematic approach to handle outliers within the estimator. Thus, characterizing outliers and in-

cluding an additional detection step becomes unnecessary. In a polynomial least-squares and a 2d ICP example, we will consider robust estimation in the presence of outliers.

5.1.1 Polynomial Least-Squares Example

Polynomial least-squares aims at finding coefficients of a polynomial. Based on input data, and noisy observations, minimization of the squared-error is used in order to obtain the estimate. At first glance, nonlinearity of the polynomial might appear challenging. This, however, is not the case as polynomial least-squares can be interpreted as a linear regression problem with a higher amount of input data. That is, we consider the measurement model

$$h_k(\boldsymbol{\theta}) = \sum_{i=0}^{n} \theta_i \, x_k^i \ ,$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$. Interpretation as a linear regression is straight forward by defining a new set of independent input variables $x_{i,k} := x_k^i$. This results in the linear regression observation model

$$h_k(\boldsymbol{\theta}) = \sum_{i=0}^{n} \theta_i \, x_{i,k} \ .$$

Experiment Setup

In this example we consider polynomial least-squares in the presence of outliers. Inliers are generated by evaluating the independent variables on the ground truth model and corrupting the result with Gaussian noise. A predefined share of dependent variables will be replaced by outlier values. These values follow a different model that is not depending on the input.

For our experiment we consider the case of $n = 2$ with the ground truth of the parameter $\boldsymbol{\theta}$ given by $\boldsymbol{\theta} = (1,\ 30,\ -30)$. The error distribution of inliers is $\mathcal{N}(0, 1)$. That is, for a given value of the independent variable x, an inlier is generated by drawing a random sample from a $\mathcal{N}(1 + 30x - 30x^2,\ 1)$ distribution. Outliers were generated using a uniform distribution on $[0, 20]$.

The independent input of the polynomial was generated as a set of 300 equally spaced values on $[0, 1]$. We performed the experiments using all different robust loss functions and different shares of outliers ranging between

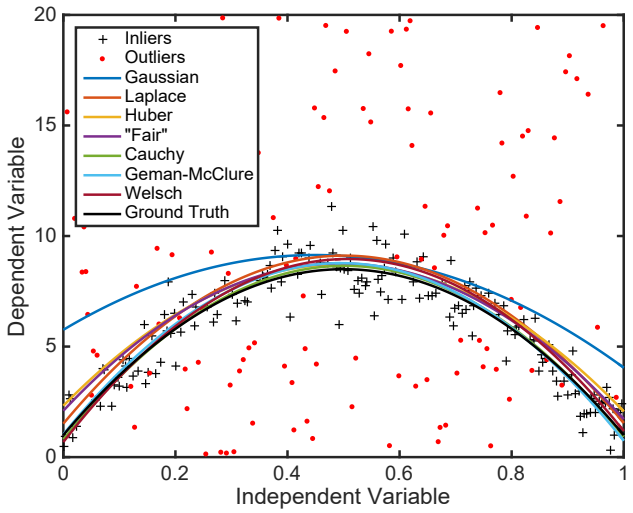

Figure 5.1: Groundtruth and resulting fit with 45% outliers.

0% and 90%. For each case, 100 runs were used. The (classical) least-squares solution was used as initial starting value for the iterative reweighted least-squares algorithm. An example run with corresponding fits for the case of 45% outliers is visualized in Figure 5.1.

Results

The results of this experiment are visualized in Figure 5.2. It shows the RMSE (averaged over the number of runs) of the inlier. Good performance for a low share of outliers can be explained by the relationship between polynomial least-squares and the linear case. However, as the number of outliers grows, the Gaussian assumption becomes more and more strongly violated. It is also not surprising that the robust estimators outperform classical least-squares as they put a weaker emphasis on high residuals and therefore the estimates are less harmed by the presence of outliers.

5.1.2 ICP in a Simulated 2D Box World

Given two sets of points (typically referred to as point clouds in robotic and computer vision literature) as input, the iterative closest point (ICP) algo-

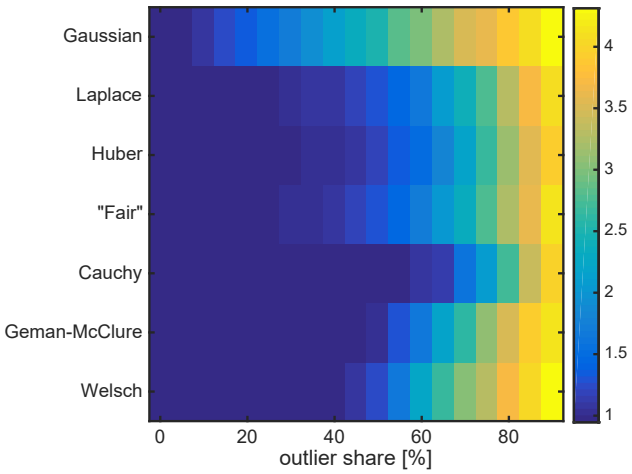

Figure 5.2: RMSE of inliers.

rithm, originally proposed in Besl and McKay (1992), is used to minimize the difference between these sets. This is useful (and therefore applied) in a broad range of robotic applications involving localization, mapping, and path planning. A typical example is aligning scans from two laser-scanners with each other. The algorithm is carried out using three steps. First, it associates each point of one point cloud with the closest point of the other point cloud. Second, a transformation (involving rotations and translations) is estimated that minimizes the mismatch between associated points. Finally, this transformation is applied. This entire process is repeated until convergence. The estimation of the transformation within ICP is usually based on a mean-squared error criteria, i.e., it can be understood as a nonlinear least-squares problem. Wrong associations within the first iterations of ICP appear as outliers in this nonlinear least-squares problem. Thus, use of robust regression approaches promises better results and faster convergence.

The general algorithm is visualized in Algorithm 1. Formulated in this generality, the algorithm involves two choices which resulted in an (at least partially ongoing) academic debate. First, the choice of the algorithm that associates the points with each other (named getClosestPonts in the pseudocode). Throughout this work, we will use the k-Nearest Neighbour search for computing this. Second, the procedure that computes a transformation for

a given association (name computeTransformation in the pseudocode) minimizing some error measure between the points.

Algorithm 1: ICP

Data: Point clouds P, Q
Result: Transformation $(\mathbf{R}, \mathbf{t})$
$\mathbf{R} \leftarrow$ identity matrix;
$\mathbf{t} \leftarrow$ zero vector;
while *not converged* **do**
  $association :=$getClosestPoints(P, Q);
  $(\tilde{\mathbf{R}}, \tilde{\mathbf{t}}) :=$ computeTransfomration$(P, Q, association)$;
  $P :=$ applyTransformation$(\tilde{\mathbf{R}}, \tilde{\mathbf{t}}, P)$;
  $\mathbf{R} \leftarrow \tilde{\mathbf{R}}\,\mathbf{R}$;
  $\tilde{\mathbf{t}} \leftarrow \tilde{\mathbf{R}}\,\mathbf{t} + \tilde{\mathbf{t}}$;
end

A very efficient solution exists, when the least-squares error measure is used, that is, when minimizing

$$r(\mathbf{R}, \mathbf{t})^2 = \sum_{i=1}^{N} ||\mathbf{y}_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}||^2$$

for two sets of vectors where $\mathbf{y}_i$ is assumed to be the transformed vector $\mathbf{x}_i$. It was proposed by Arun et al. (1987) and is based on three steps. First, the means $\overline{\mathbf{y}}$, $\overline{\mathbf{x}}$ of both sets are computed. Then we compute the singular value decomposition (see Golub and Van Loan (2013)) $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ of

$$\sum_{i=1}^{N} (\mathbf{y}_i - \overline{\mathbf{y}}) \cdot (\mathbf{x}_i - \overline{\mathbf{x}}_i)^\top$$

Finally, a minimizer of $r(\mathbf{R}, \mathbf{t})^2$ of obtained by choosing $\mathbf{R} = \mathbf{U}\mathbf{V}^*$ and $\mathbf{t} = \overline{\mathbf{y}} - \mathbf{R}\overline{\mathbf{x}}$. This, so called, point-to-point error metric was used in the original paper by Besl and McKay (1992).

A point-to-plane metric was proposed by Chen and Medioni (1992). It is given by

$$r(\mathbf{R}, \mathbf{t})^2 = \sum_{i=1}^{N} \left((\mathbf{y}_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}) \cdot \mathbf{n}_i\right)^2 .$$

In this metric $\mathbf{y}_i$ and $\mathbf{x}_i$ are the same as above and $\mathbf{n}_i$ is the normal at $\mathbf{y}_i$. As the actual plane (or line in case of 2d ICP) is not directly known for a

given set of points, $\mathbf{n}_i$ is computed empirically from neighboring points. In the literature, it was observed that the point-to-plane metric outperforms the point-to-point approach, e.g. see Rusinkiewicz and Levoy (2001). However, it does not give rise to a computational similarly nice solution. Therefore, linear least-squares is used to obtain the desired result as presented in Low (2004).

Example Setup and Results In our example, we will use the point-to-plane error metric and improve its convergence properties by replacing linear least-squares with a robust M-Estimator. In our setup, we consider a 2d case with two laser scanners positioned in a room with dimensions (width $\times$ length) 10m$\times$20m. The first scanner is located at $(2, 3)$ with an angle of $0°$. The second scanner is located nearby, that is at $(2.2,\ 3.1)$ with an angle of $5°$. Both scanners have an opening angle of $180°$ and measure ranges with a resolution of $1°$. The standard deviation of the range measurements is 0.03m. A typical (outlier-free) scan is visualized in Figure 5.3. Outliers were generated by corrupting a predefined share of correct range measurements (which are randomly selected) with additional noise. The distribution for this outlier noise is a zero-mean gaussian with 1m standard deviation, i.e. it is much stronger compared to the true measurement noise.

The experiments were carried out by considering different shares of outliers (between 0% and 90%). For each share of outliers we performed 2000 scans and subsequent runs of the ICP algorithm using different m-estimators. Our results are shown in Figure 5.4. Similar, to the polynomial least-squares example, it can be seen that the Gaussian approach is outperformed by all m-estimators.

5.2 Non-Gaussian Noise Modeling

In a probabilistic view on linear regression, it was seen as a maximum likelihood estimator for the case of Gaussian. That is, the noise distribution is symmetric around the mean. From the relationship between the presented robust estimators and elliptical distributions, it can be seen that these estimators also implicitly assume the underlying noise model to be symmetric. This ex-

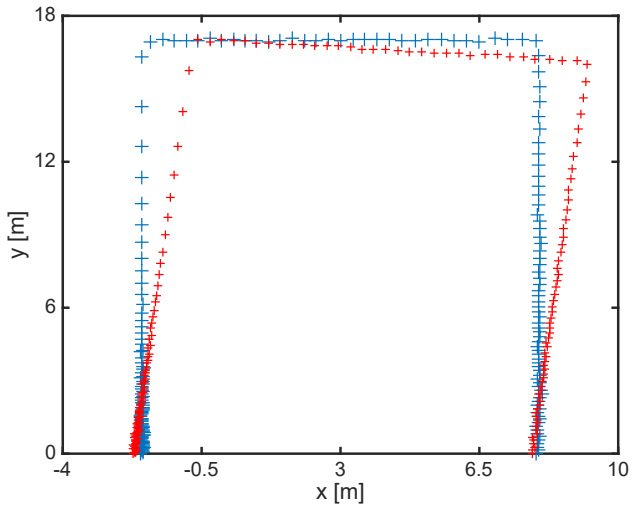

Figure 5.3: Outlier free scans from both scanners (blue is used for the first scanner and red for the second) represented in their respective coordinate system. The main goal of the ICP algorithm is to align these two scans, i.e., to find a planar transformation such that both scans match each other as close as possible.

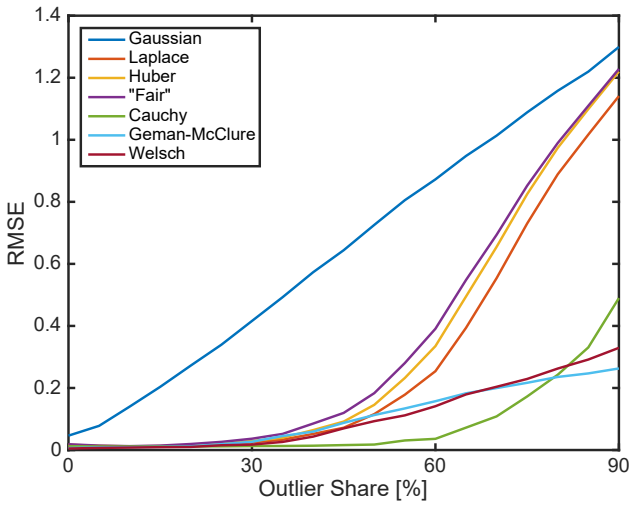

Figure 5.4: RMSE of obtained transformation for different shares of outliers. The least-squares error measure is outperformed by all other approaches. In this example use of the Cauchy loss yielded satisfactory results even for very high shares of outliers.

ample focuses on discussing how iteratively reweighted least-squares can be used in scenarios involving skewed noise.

Once again we consider a polynomial regression example. We assume, our observations to be corrupted by a biased noise model. As linear regression is minimizing the RMSE, it seems to be an unsuitable error measure in this case. Therefore, we perform reweighting that takes the skewness of the underlying distribution into account. The weighting function used in this example is given by

$$w_k(\boldsymbol{\theta}) = \begin{cases} q \cdot \left(\sqrt{r_k(\boldsymbol{\theta})^2 + 1}\right)^{-1} & r_k(\boldsymbol{\theta}) \geq 0\,, \\ (1-q) \cdot \left(\sqrt{r_k(\boldsymbol{\theta})^2 + 1}\right)^{-1} & r_k(\boldsymbol{\theta}) < 0\,. \end{cases}$$

The parameter $q \in (0, 1)$ denotes the probability mass of the error distribution which is smaller than 0. That is, random samples coming from the side with the stronger probability mass are downweighted and vice versa. Furthermore, the weighting term $\left(\sqrt{r_k(\boldsymbol{\theta})^2 + 1}\right)^{-1}$ can be thought of a modified variant of Huber's m-estimator. It resembles Huber in that the loss function is (approximately) quadratic near the origin and (approximately) linear far away from the origin. It differs in the transition between these two types of loss, which in this case occurs gradually as the residual increases.

For our example, we used a two-sided exponential distribution with different rate parameters as a model for (skewed) observation noise. It is given by the p.d.f.

$$f(x;\, q,\, \lambda_1,\, \lambda_2) = \begin{cases} q\,\lambda_1 \mathrm{e}^{-\lambda_1\, x} & x \geq 0\,, \\ (1-q)\,\lambda_2 \mathrm{e}^{-\lambda_2\, x} & x < 0\,. \end{cases}$$

Here $q \in [0, 1]$ and $\lambda_1,\, \lambda_2 > 0$. The parameter q decides on the amount of probability mass on each side, whereas the parameters λ_1 and λ_2 are the parameters for the exponential distribution on each side respectively. A Laplace distribution is obtained as a special case when $q = 0.5$ and $\lambda_1 = \lambda_2$.

In the example shown in Figure 5.5, we have generated a random polynomial of order 3 and computed 20 evaluations equally distributed on $[-5,\, 5]$. The evaluation result is corrupted with noise distributed according to the two-sided exponential distribution presented above. As noise distribution parameters we used $q = 0.9$ and $\lambda_1 = 1 - q$, $\lambda_2 = q$. The resulting error density is

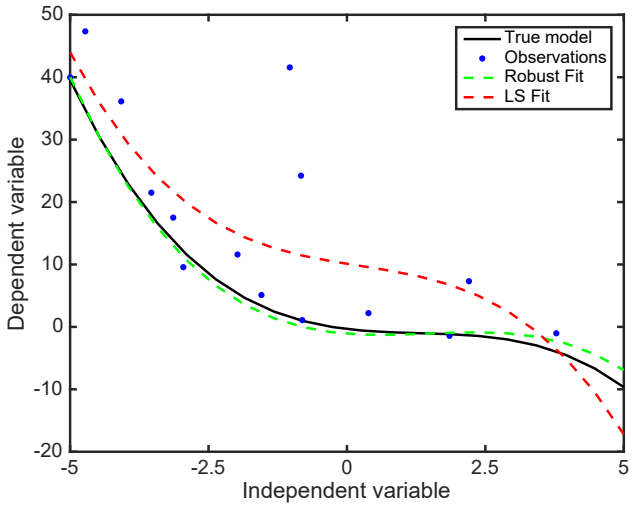

Figure 5.5: Linear regression fails in the presence of skewed noise as its underlying assumption considers the noise to be Gaussian.

visualized in Figure 5.6. Overall, from the results presented in Figure 5.5, it is seen how linear regression may fail in the presence of skewed noise.

5.3 Improved Convergence for Nonlinear Optimization

Viewing iteratively reweighted least sqaures as an optimization procedure gives rise to another interpretation of robust estimation techniques. In this view, they can be seen as a method for improving convergence

5.3.1 Annealed M-Estimation for Improved Convergence Basin for 2D ICP Problem

In the first ICP example, both scanner positions were close to each other and had similar viewing angles. This is, because ICP is a local alignment method assuming the initial value to be close to the actual solution. Use of m-estimators cannot fully overcome this limitation. However, it is possible to improve the convergence basin of ICP by using m-estimators. The general setting of this example is similar to the first ICP example. It mainly differs in the fact that it considers different distances between the laser scanners rather than different shares of outliers.

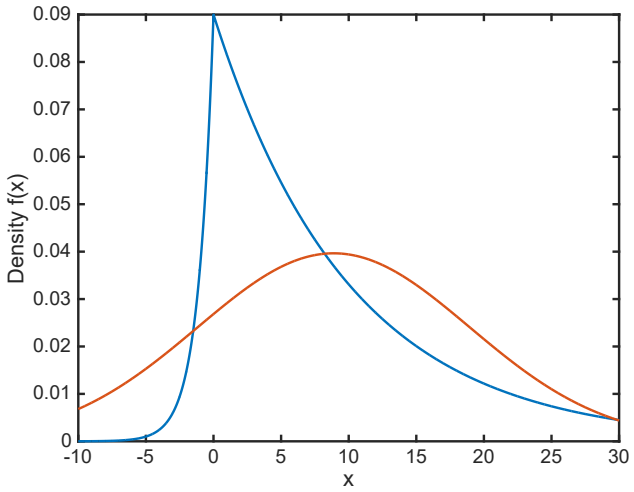

Figure 5.6: Density of the two-sided exponential distribution with different rate parameters which was used for generating the noisy observations (blue). And, for comparison, a Gaussian density with same mean and covariance (red).

Additionally to the m-estimators presented so far, this chapter will involve a slightly modified version of the Cauchy loss. This estimator involves an iteration dependent weighting function. That is, as the optimization procedure evolves, the impact of the estimator is reduced. The idea behind this approach is that as the estimation procedure evolves, it becomes less necessary to make use of the downweighting introduced by the loss function, because this downweighting mainly serves to punish wrong point associations. The resulting weighting is simply the original Cauchy weight with the parameter σ being replaced by $k \cdot \sigma$, where k denotes the current iteration.

In this example, we again consider a 10m$\times$20m room with two laser scanners. To make the involved optimization problem more challenging, several rectangular obstacles are placed in the room. The first scanner is fixed at $(4.1,\ 3.1)$ heading up with a 5° rotation to the left. The second scanner is also heading upwards with a small 5° rotation to the right. However, throughout the example, its position is varied within the lower, obstacl-free part of the room. That is, the second scanner is located at different positions on a 20×20 grid spread between 0.05 and 9.95 on the x-axis, and 0.05 and 7.95

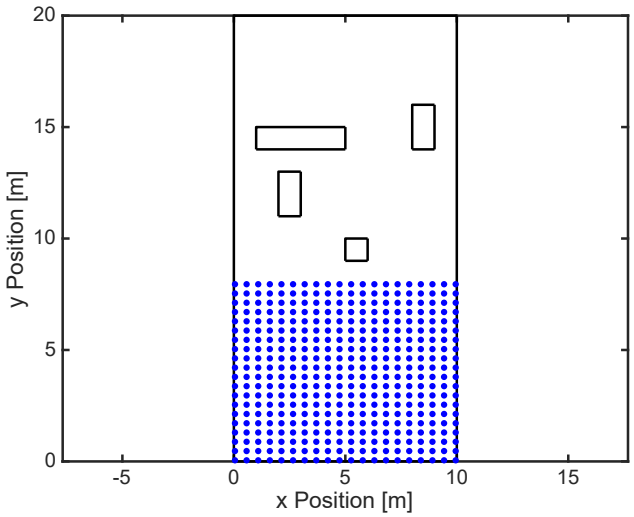

Figure 5.7: Room used in the ICP 2d convergence example. The blue dots indicate considered positions of the second scanner.

on the y-axis. The room geometry and all locations of the second scanner are visualized in Figure 5.7.

The results are visualized in Figures 5.8 and 5.9. They show the RMSEs that were obtained depending on the position of the second scanner. While all esimators achieve good results for certain positions, the convergence basin of the annealed cauchy is the largest for all estimators.

5.3.2 Robustness to Winding Error in Pose Graph Optimization

Pose graph optimization is used in order to build a map of the environment and obtain an estimate of the robots trajectory from relative pose measurements. These measurements can be obtained from wide range of sensors including inertial sensors and cameras. A first presentation of this approach was given in Lu and Milios (1997). Use of robust estimation techniques helps to avoid local minima as it systematically reduces the influence of outliers.

Here, we consider the famous Manhattan 3500 dataset as presented in Olson et al. (2006). It consists of relative pose measurements from an robot traveling in a 2d grid-like world. That is, in the absence of noise all turns

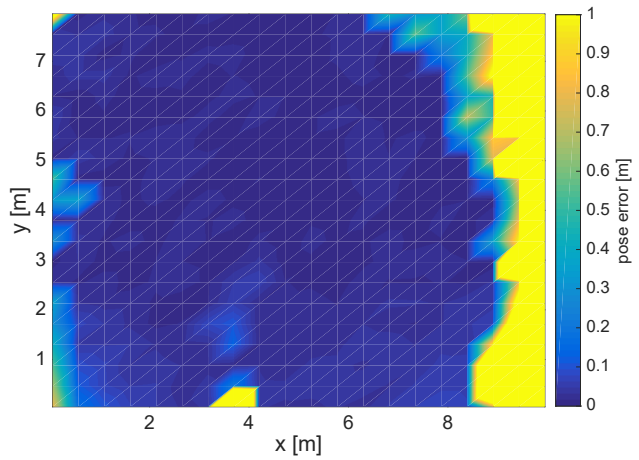

Figure 5.8: Results of the annealed Cauchy estimator. Annealing leads to additionally improving the convergence basin of the estimator.

would be $90°$. Additionally, the dataset contains an initial estimate of the ground truth, which consists of 3500 robot poses.

In this particular example, we consider the use of robust estimators in context with a winding error within the data. For this purpose, the Manhattan dataset is adapted by introducing a fixed additional angular bias into each odometry measurement. The magnitude of this bias is around $0.24°$ for each measurement. Due to the fact, that these measurements are relative, introducing this bias in each measurement results in a strong winding error for the entire posegraph. This served as a starting point for a nonlinear regression based posegraph optimization in which, once again, the previously presented loss functions were used. The results of this optimization are visualized in Figure 5.10. With the exception of the "Fair" loss function, all other losses outperform the Gaussian loss, i.e. simply using nonlinear least-squares. An example visualizing the process of convergence, here for the particular case of a Cauchy loss, is shown in Figure 5.11. There it is seen, how unwinding happens within the optimization process.

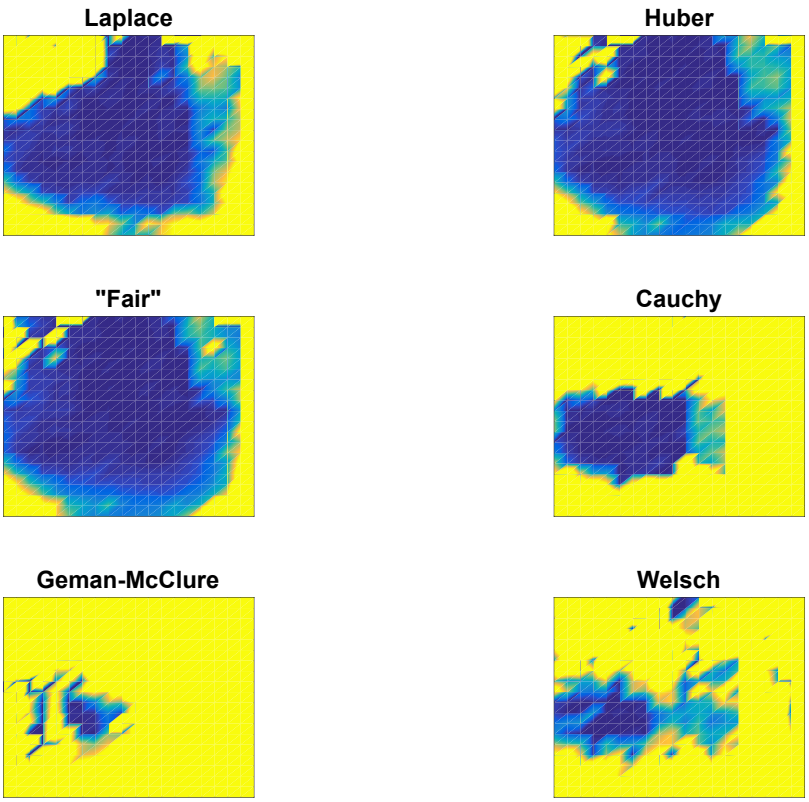

Figure 5.9: Convergence results for the m-estimators considered in this tutorial.

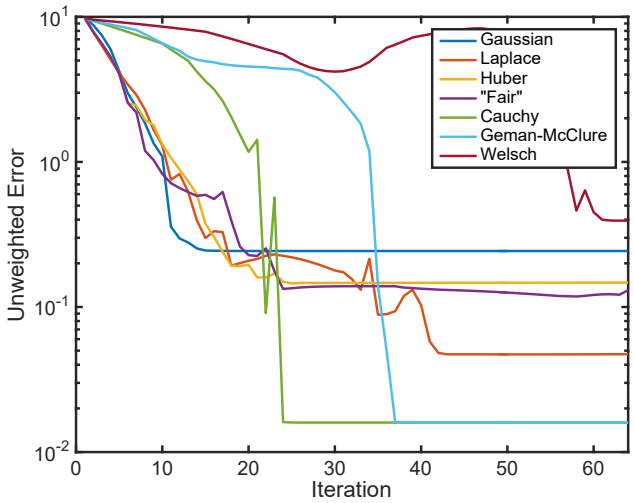

Figure 5.10: Unweighted error for pose graph optimization using different M-Estimators. Once again the Cauchy loss achieves the best performance in convergence speed and error minimization.

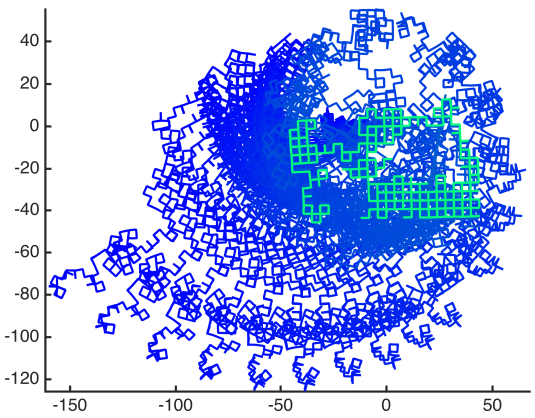

Figure 5.11: Visualization of the posegraph optimization for the case where the Cauchy loss was used.

6

Discussion and Further Reading

In this paper, our aim was to provide a first introduction to robust estimation for robotics. The underlying approach considered in this work is based on robust statistics. Thus, we focused on a concise presentation of the theoretical foundations of robust statistics. Within the discussion of examples, we focused on several typical challenges that arise in real-world robotic applications. First, handling of outliers within measurements, that can arise from sensor failure and interferences. Second, consideration of biased and, more generally, non-gaussian noise models. This becomes necessary as in practical applications sensor models might be not precise enough and potentially not account for hidden factors affecting the measurements. Finally, consideration of nonlinearities is necessary as most real-world system models exhibit a nonlinear structure, and are prone to errors even if linearity is assumed implicitly as is done when assuming estimated quantities to be jointly Gaussian.

There are several potential topics that are worth further reading for a better understanding of robust statistics in robotics. First, a thorough discussion on the theoretical background is given in the book by Hampel et al. (1986). Second, a detailed account of (nonlinear) regression analysis is given in Seber and Wild (2003), Seber and Lee (2003). Particularly, in the special case of nonlinear regression it is worth taking a look at nonlinear optimization

methods, see e.g. Nocedal and Wright (1999). Finally, deeper understanding of the inference problem at hand is helpful for better choosing the proper M-Estimator. Thus, it is also worth revisiting discussions of these inference problems, e.g. Grisetti et al. (2010) in case of graph-based SLAM.

References

G. Agamennoni, J. Nieto, and E. Nebot. Robust inference of principal road paths for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):298–308, March 2011.

G. Agamennoni, P. Furgale, and R. Siegwart. Self-tuning M-estimators. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2015.

P. Agarwal, G. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust map optimization using dynamic covariance scaling. In *International Conference on Robotics and Automation*, 2013a.

P. Agarwal, G. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Covariance scaling for robust map optimization. In *ICRA Workshop on Robust and Multimodal Inference in Factor Graphs*, 2013b.

S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski. Bundle adjustment in the large. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.

L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1), 1966.

K. Arun, T. Huang, and S. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

L. Carlone, A. Censi, and F. Dellaert. Selecting good measurements via ℓ_1 relaxation: A convex approach for robust estimation over graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2667–2674, 2014.

Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, April 1992.

C.-L. Cheng. Robust linear regression via bounded influence M-estimators. *Journal of Multivariate Analysis*, 40(1):158–171, 1992.

A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. SIAM, 2000.

H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

T. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, December 2006.

J. Eriksson and P.-Å. Wedin. Truncated gauss-newton algorithms for ill-conditioned non-linear least-squares problems. *Optimization Methods and Software*, 19(6): 721–737, December 2004.

K.-T. Fang, S. Kotz, and K. Ng. *Symmetric Multivariate and Related Distributions*. Chapman & Hall, 1987.

H. Fischer. *A History of the Central Limit Theorem*. Springer, 2011.

M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, jun 1981.

G. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, 2013.

S. Gratton, A. Lawless, and N. Nichols. Approximate gauss-newton methods for non-linear least-squares problems. *SIAM Journal of Optimization*, 18(1):106–132, 2007.

G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31 –43, 2010.

F. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, June 1974.

F. Hampel. Introduction to Huber (1964): Robust estimation of a location parameter. In S. Kotz and N.L. Johnson, editors, *Breakthroughs in Statistics*, Springer Series in Statistics, pages 479–491. Springer, 1992.

F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, March 1986.

G. Hee Lee, F. Fraundorfer, and M. Pollefeys. Robust pose graph loop closures with expectation-maximization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

P. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

P.J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3754, May 2013. .

K. Kim and G. Shevlyakov. Why Gaussianity? *IEEE Signal Processing Magazine*, 25(2):102 –113, March 2008.

R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, May 2011.

S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, March 2015.

K.-L. Low. Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Technical report, University of North Carolina at Chapel Hill, February 2004.

J. Loxam and T. Drummond. Student t mixture filter for robust, real-time visual tracking. In *Proceedings of the 10th European Conference on Computer Vision*, 2008.

F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997.

R.D. Martin and R.H. Zamar. Bias-robust estimation of scale. *The Annals of Statistics*, 21(2):991–1017, 1993.

J. Maye, P. Furgale, and R. Siegwart. Self-supervised calibration for robotic systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 473–480, June 2013.

S. Neugebauer. Robust analysis of M-estimators of non-linear models. Master's thesis, School of Electrical and Computer Engineering, 1996.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. In *Robotics: Science and Systems*, 2012.

E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. Technical report, University of Michigan & Universität Freiburg, 2013.

E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the International Conference on Robotics and Automation*, pages 2262–2269, 2006.

C. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3):81–91, 1945.

D. Rosen, M. Kaess, and J. Leonard. Robust incremental online inference over sparse factor graphs: Beyond the gaussian case. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1025–1032, May 2013.

P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE Comput. Soc, 2001.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

G. A. F. Seber and A. J. Lee. *Linear Regression Analysis*. John Wiley & Sons, 2003.

G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, 2003.

R. Staudte and S. Sheather. *Robust Estimation and Testing*. Wiley, 1990.

C. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3): 513–537, 1999.

N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph SLAM. In *International Conference on Robotics and Automation*, 2012a.

N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012b.

N. Sünderhauf and P. Protzel. Switchable constraints vs. max-mixture models vs. RRR — a comparison of three approaches to robust pose-graph slam. In *Proceedings of the International Conference on Robotics and Automation*, pages 5178–5183, May 2013.

J. Ting, E. Theodorou, and S. Schaal. A Kalman filter for robust outlier detection. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2007.

P. Torr and D. Murray. The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

C. Zach. Robust bundle adjustment revisited. In *Computer Vision – ECCV*, volume 8693 of *Lecture Notes in Computer Science*, pages 772–787. Springer, 2014.

Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1):59–76, 1997.

A. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 29(4):61–80, July 2012.