

At All Costs: A Comparison of Robust Cost Functions for Camera Correspondence Outliers

Kirk MacTavish and Timothy D. Barfoot

Abstract—Camera-based localization techniques must be robust to correspondence errors, i.e., when visual features (landmarks) are matched incorrectly. The two primary techniques to address this issue are *RANSAC* and *robust M-estimation* – each more appropriate for different applications. This paper investigates the use of different robust cost functions for M-estimation to deal with correspondence outliers, and assesses their performance under varying degrees of data corruption. Experimental results show that using an aggressive re-descending cost function (e.g., *Dynamic Covariance Scaling (DCS)* or *Geman-McClure (G-M)*) best improves accuracy by excluding outliers almost entirely. Additionally, adjusting an error-scaling parameter for the robust cost function over the course of the optimization improves convergence with poor initial conditions.

I. INTRODUCTION

Cameras have become ubiquitous for robotic navigation and localization, as they are both inexpensive and effective. These techniques often rely on identifying unique image patches (visual features) in an image, and matching them to corresponding patches in another [1], [2]. Determining which features match (feature correspondence) is difficult – changing lighting or perspective and perceptual aliasing can lead to false matches. These errors result in extreme measurement outliers and a highly non-Gaussian error distribution.

Rejecting correspondence errors is often handled by Random Sample Consensus (RANSAC) [3]. RANSAC consists of randomly selecting the minimum number of measurements required to satisfy the model, counting the total number of measurements in agreement, and repeating many times. The largest consensus is used, and the remaining measurements are discarded as outliers. RANSAC is very popular since it is usually quite effective, very fast, and easy to implement.

However, RANSAC does have its drawbacks. It can be quite sensitive to the inlier threshold – too high and outliers are used, too low and valid measurements are discarded. The inlier classification is typically performed only once, prior to optimization, though an improved state estimate might better inform the outlier rejection. RANSAC also requires a simple model that can be uniquely solved with only a few measurements, so that the odds are favourable that an uncorrupted random sample can be found. Conversely, it needs enough measurements to uniquely solve the model at all times – RANSAC will perform poorly on datasets with sparse measurements and frequent drop-outs.

The authors are with the Autonomous Space Robotics Lab (ASRL) at the University of Toronto Institute for Aerospace Studies (UTIAS) kirk.mactavish@mail.utoronto.ca, tim.barfoot@utoronto.ca

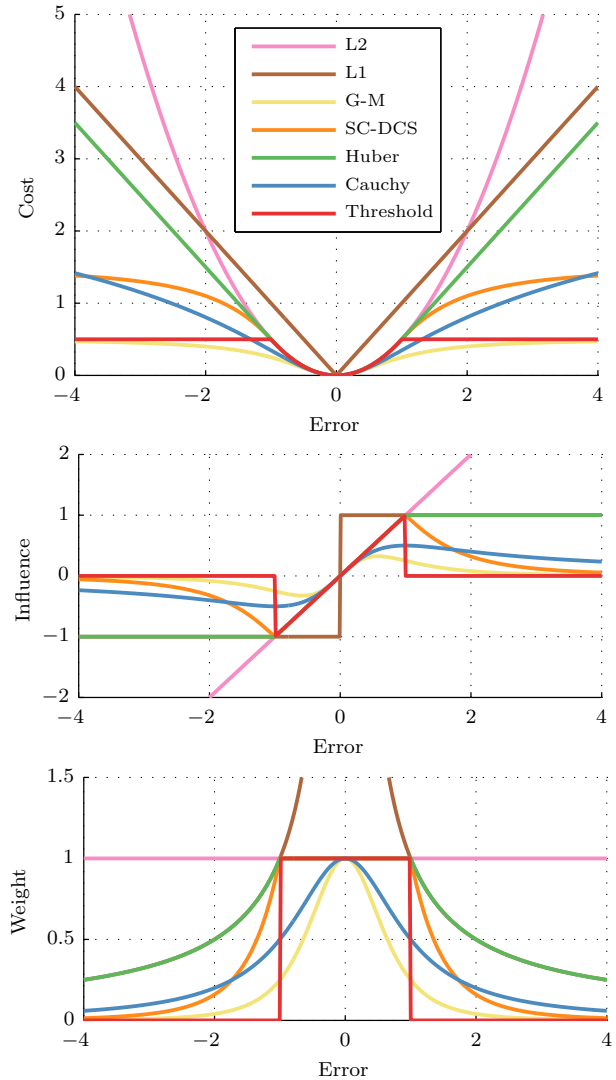


Fig. 1: Plots of the robust cost, influence, and weight functions for the robust M-estimators defined in Table I. In all cases the tuning parameter is set to its default value, 1. The *Huber*, *Dynamic Covariance Scaling (DCS)*, and *RANSAC-like Threshold* functions are all the same as *L2* when the error is small. *DCS* is the same as *Geman-McClure (G-M)* ($\times 4$) for larger errors, *Threshold* is completely off, and *Huber* matches *L1*. The *L1* weights are infinity at zero error, making it unsuitable for robust estimation, though common for regularization. Both *L1* and *Threshold* have discontinuous influence functions, which may cause chatter during optimization.

Robust M-estimation is an alternative method to address the outlier problem [4]. Instead of a least-squares, maximum-likelihood cost function, a robust cost function is used to decrease the influence of outliers. A robust cost function is sub-quadratic, with several examples shown in Figure 1.

Rodriguez *et al.* [5] published an example of a robust estimator being used for camera-based localization using the Beaton Tukey Biweight robust function [6], though they did not give much insight into the robust function selection. Hu *et al.* [7] propose using robust M-estimation to bootstrap Simultaneous Localization and Mapping (SLAM) graph optimization. Their work uses the *Geman-McClure* (*G-M*) and *Cauchy* cost functions to reject bad loop-closure proposals [7]. Lee *et al.* [8] also propose using the *Cauchy* robust function in a SLAM optimization framework, primarily comparing *Cauchy* to the *Huber* cost function. While Zhang [9] makes remarks on a comprehensive list of robust cost functions, there is not a thorough comparison of the common robust cost functions as applied to visual navigation.

There are many named examples of robust cost functions [9], with dramatically differing behaviours. This paper will show how several of these robust functions perform on real data, and under which conditions they are useful. We employ a visual localization dataset that is of particular interest, as it has very sparse landmark correspondences and frequent visual drop-outs. The dataset is too difficult for RANSAC, and robust estimation is a very good candidate for outlier rejection.

II. ROBUST COST FUNCTIONS

This section introduces robust cost functions in the context of robust M-estimation. We begin with a maximum-likelihood Gaussian (least-squares) estimator,

$$J(\mathbf{x}) \triangleq e(\mathbf{x})^2, \quad e(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|_{\Lambda}, \quad (1)$$

where J is the cost we want to minimize, \mathbf{x} is the state we are estimating, $\mathbf{f}(\mathbf{x})$ is the measurement model, \mathbf{y} is the measurement vector, Λ is our measurement covariance, and $e(\mathbf{x})$ is our error as the Mahalanobis distance. We often have a stacked measurement vector, \mathbf{y} , with K multi-dimensional measurements that are conditionally independent given the state, \mathbf{x} ; for example, these groups might correspond to observations of landmarks in a stereo image pairs. The cost function can be written as a sum over these K independent measurements:

$$J(\mathbf{x}) \triangleq \sum_{k=1}^K J_k(\mathbf{x}), \quad J_k(\mathbf{x}) \triangleq e_k(\mathbf{x})^2, \quad e_k(\mathbf{x}) \triangleq \|\mathbf{y}_k - \mathbf{f}_k(\mathbf{x})\|_{\Lambda_k}. \quad (2)$$

Robust M-estimation redefines the cost function, replacing it with a sub-quadratic function, $\rho(e)$:

$$J_k(\mathbf{x}) \triangleq \rho(e_k(\mathbf{x})). \quad (3)$$

We can iteratively set the derivative to zero to identify the optimal state, \mathbf{x}^* , corresponding to the minimum of this new cost function; we also introduce the influence function, $\psi(e)$:

$$\frac{\partial J_k(\mathbf{x})}{\partial \mathbf{x}} = \psi(e_k(\mathbf{x})) \frac{\partial e_k(\mathbf{x})}{\partial \mathbf{x}}, \quad \psi(e_k) \triangleq \frac{\partial \rho(e_k)}{\partial e_k}, \quad (4)$$

$$\Rightarrow 0 \triangleq \psi(e_k(\mathbf{x}^*)) \left. \frac{\partial e_k(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}^*}, \quad (5)$$

where \mathbf{x}^* is the optimal state at the previous iteration. The solution to (5) depends on the form of ψ , and can be quite difficult. However, we can define a weight function, w , that allows us to match the Iteratively Reweighted Least-Squares (IRLS) problem [10],

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_k w(e_k(\mathbf{x}^*)) e_k(\mathbf{x})^2, \quad (6)$$

$$\Rightarrow 0 \triangleq w(e_k(\mathbf{x}^*)) e_k(\mathbf{x}^*) \left. \frac{\partial e_k(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}^*}, \quad w(e_k) \triangleq \frac{\psi(e_k)}{e_k}, \quad (7)$$

where we iteratively solve the least-squares problem using robust weights computed from the previous solution.

To summarize, a robust cost provides

$$J_k(\mathbf{x}) \triangleq \rho(e_k(\mathbf{x})), \quad \psi(e_k) \triangleq \frac{\partial \rho(e_k)}{\partial e_k}, \quad w(e_k) = \frac{\rho(e_k)}{e_k}. \quad (8)$$

Figure 1 shows several types of these robust cost, influence, and weight functions, which are defined in Table I.

A. Switchable Constraints and Dynamic Covariance Scaling

Two recent techniques that have had good success for robotic mapping are *Switchable Constraints* (SC) [11] and *Dynamic Covariance Scaling* (DCS) [12]. They are not presented in the literature as robust cost functions, but they can be expressed as such. Both methods minimize the following objective function:

$$J(\mathbf{x}, \mathbf{s}) \triangleq \sum_k J_k(\mathbf{x}, \mathbf{s}), \quad (9)$$

$$J_k(\mathbf{x}, \mathbf{s}) \triangleq s_k^2 e_k^2 + \phi(1-s_k)^2, \quad (10)$$

where $e_k \triangleq \|\mathbf{y}_k - \mathbf{f}_k(\mathbf{x})\|_{\Lambda_k}$, and $s_k \in [0, 1]$ is a switch variable from \mathbf{s} , that enables or disables measurement k . DCS solves for the optimal value for s_k in closed form:

$$\frac{\partial J_k(x_k, s_k)}{\partial s_k} = 2s_k e_k^2 - 2\phi(1-s_k) \triangleq 0 \quad (11)$$

$$\Rightarrow s_k = \frac{\phi}{\phi + e_k^2}. \quad (12)$$

Substituting into the original cost function,

$$J_k(x_k, s_k) = \left(\frac{\phi e_k}{\phi + e_k^2} \right)^2 + \phi \left(1 - \frac{\phi}{\phi + e_k^2} \right)^2 = \frac{\phi e_k^2}{\phi + e_k^2}, \quad (13)$$

which is exactly the *G-M* robust cost function ($\times 2$). Agarwal *et al.* [12] elect to scale the cost by another factor of two (which has no effect on the solution), and constrain the weight value, s_k^2 , to be in the range $[0, 1]$. This constraint (similar to *Huber*) makes *SC/DCS L2* for small errors, and *G-M* for large. The final robust function definition is provided in Table I.

III. DATASET

To evaluate the robust functions in Table I, we introduce a visual-inertial localization experiment. This experiment uses the *Starry Night Dataset* prepared by Barfoot [15]. The sensor system consists of an Inertial Measurement Unit (IMU) mounted to a stereo camera. The camera observes a scene

TABLE I: The table below shows the cost, influence, and weight functions for the robust M-estimators compared in this paper. See Figure 1 for visualization and additional observations.

Type	$\rho(e)$	$\phi(e)$	$w(e)$
$L2$	$e^2/2$	e	1
Huber [13]	$\begin{cases} e^2/2 & e \leq k \\ k(e - e/2) & e \geq k \end{cases}$	$\begin{cases} e & e \leq k \\ k \operatorname{sgn}(e) & e \geq k \end{cases}$	$\begin{cases} 1 & e \leq k \\ k/ e & e \geq k \end{cases}$
$L1$	$ e $	$\operatorname{sgn}(e)$	$1/ e $
Cauchy	$\frac{e^2}{2} \log(1 + \frac{e^2}{c^2})$	$\frac{e}{1 + (e/c)^2}$	$\frac{1}{1 + (e/c)^2}$
SC/DCS [11], [12]	$\begin{cases} e^2/2 & e^2 \leq \phi \\ \frac{2\phi e^2}{\phi + e^2} - \phi/2 & e^2 \geq \phi \end{cases}$	$\begin{cases} e & e^2 \leq \phi \\ \frac{4\phi^2 e}{(\phi + e^2)^2} & e^2 \geq \phi \end{cases}$	$\begin{cases} 1 & e^2 \leq \phi \\ \frac{4\phi^2}{(\phi + e^2)^2} & e^2 \geq \phi \end{cases}$
$G\text{-}M$ [14]	$\frac{e^2/2}{\sigma + e^2}$	$\frac{\sigma^2 e}{(\sigma + e^2)^2}$	$\frac{\sigma^2}{(\sigma + e^2)^2}$
Thresh.	$\begin{cases} e^2/2 & e \leq t \\ t^2/2 & e > t \end{cases}$	$\begin{cases} e & e \leq t \\ 0 & e > t \end{cases}$	$\begin{cases} 1 & e \leq t \\ 0 & e > t \end{cases}$

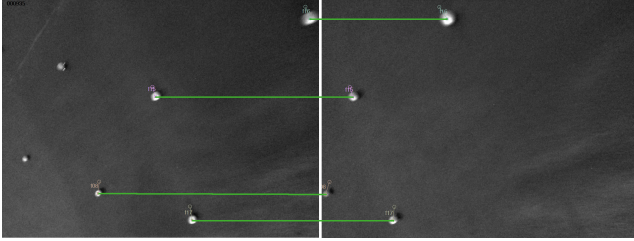


Fig. 2: A sample stereo image pair from the dataset, showing Vicor markers as landmarks, and stereo correspondence in green. Landmark correspondence for each stereo pair is given a priori, and corrupted in simulation (source [15]).

of infrared-reflective markers for the Vicor motion-capture system. These markers are used as landmarks in the localization problem, and their positions are known a priori. Figure 2 shows an example of a stereo image pair from the dataset. The dataset provides landmark correspondence, i.e., which Vicor marker corresponds to each dot seen in each image. The number of landmarks seen in each image are shown in Figure 3a. Since three landmarks are required to uniquely solve for the camera pose, not all images provide a full solution; there is a fairly long drop-out in the middle of the dataset that is particularly challenging. The IMU measurements have been sanitized to remove biases. Barfoot [15] provides additional information.

A. The Localization Problem

We would like to solve for the maximum-likelihood camera trajectory given intrinsic velocity measurements in translation and rotation, and extrinsic stereo-camera landmark measurements. We use a batch IRLS Gauss-Newton estimator to minimize the cost function,

$$J(\mathbf{x}) \triangleq \sum_k^K \|\mathbf{e}_{\text{int},k}(\mathbf{x})\|_{\mathbf{Q}}^2 + \sum_k^K \sum_l^{L_k} \rho(\|\mathbf{e}_{\text{ext},kl}(\mathbf{x})\|_{\mathbf{R}}), \quad (14)$$

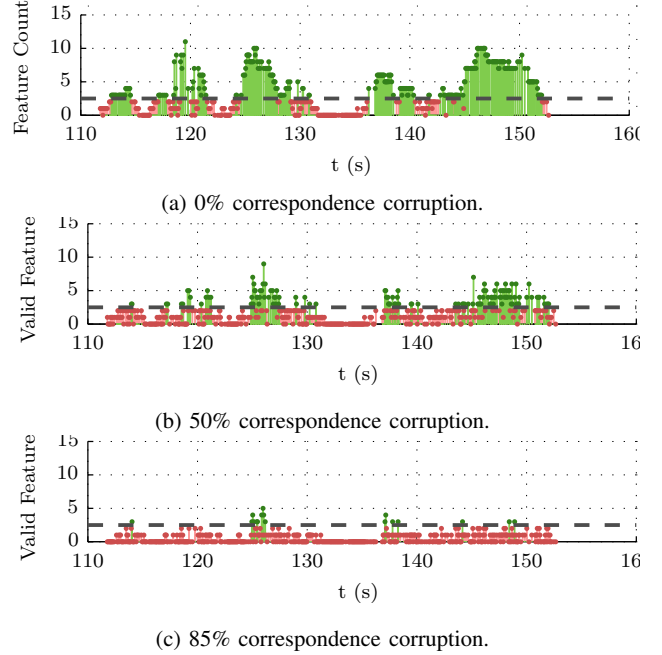


Fig. 3: The number of landmarks with correct correspondence seen at each image in the dataset. Correct landmark visibility decreases as the correspondences become increasingly corrupt. Images with at least three valid landmarks (the minimum number required to uniquely solve for the state) are coloured green, those with fewer are coloured red.

where e is the error between the model-predicted and actual intrinsic (int) and extrinsic (ext) measurements, \mathbf{Q} and \mathbf{R} are the intrinsic and extrinsic measurement covariances, ρ is a robust cost function, K is the number of timesteps, and L_k is the number of landmarks visible at the k -th timestep.

It can be noticed in Table I that many of the functions have a tunable parameter; all of these parameters were set to their default value, 1. To allow adjustment, we instead deflate the error by d , this provides a more uniform and predictable tuning across robust types. Our modified robust function is now:

$$\rho'(e) = \rho(e/d). \quad (15)$$

Many of the robust weights have a quadratic trust region when $|e| < 1$. This deflation term lets us scale the error so that the trust region is active for a specified number of standard deviations. For example, by starting with $s \triangleq 10$ and decaying to 3 over the course of the optimization, we expect our inliers errors to settle into a smaller trust region as our state estimate improves. This lets us maintain the influence of our inliers, while excluding as many outliers as possible. Since the localization problem is nonlinear, and we are already iterating Gauss-Newton to solve the least-squares problem; iteratively reweighting does not add much overhead.

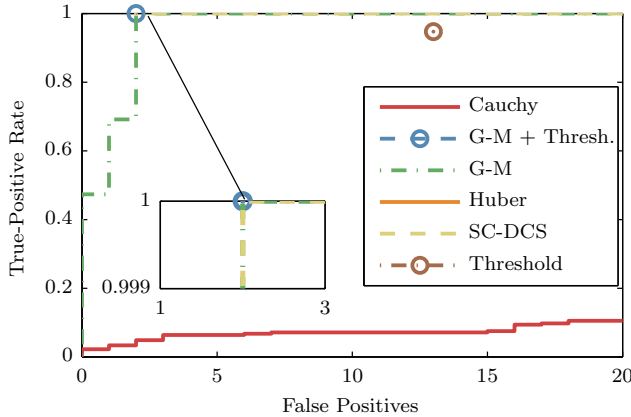


Fig. 4: The Receiver Operating Characteristic (ROC) curve for outlier rejection at 85% corruption. To find the number of false positives (trusted corrupt measurements) at a given true-positive rate (fraction of trusted valid measurements), we can threshold the final robust weights leaving those above as trusted. Sweeping this threshold generates the ROC curve in this figure. Ideally, we would have a single point in the top left at (0, 1). The inset shows that *SC-DCS* and the two *G-M* variants are close to this optimal performance, with only two false positives – this is less than 0.2% of the 1162 corrupt measurements. *Threshold* produces only a single point, as the weights are already thresholded. Both *Huber* and *Cauchy* did not converge; in fact, *Huber* accepts 40 false positives before any true, so it is not visible with these false-positive limits.

B. The Corruption Scheme

To simulate feature correspondence errors in a typical vision-based localization system, we manually introduce correspondence errors from image landmarks to Vicon landmarks. We randomly select landmark observations to corrupt; for each of these landmarks, we randomly generate a new landmark identification (id), from the total set of 20 unique landmarks. If the landmark id is unchanged, the correspondence remains uncorrupted. Otherwise, the new landmark id is assigned. If the new landmark id was already observed in the image, that old observation is discarded. Therefore, there is a chance that the corruption could overwrite an additional measurement.

While these outliers make the localization problem more challenging by themselves, they also greatly reduce the number of valid measurements. Figure 3 shows the number of valid measurements in each image for two different corruption levels. At 85% corruption, there are very few images with three or more landmarks – the minimum number required by RANSAC to uniquely solve for the camera pose.

IV. EXPERIMENTAL RESULTS

To test performance under varying degrees of data corruption, three trials were run: the first with no correspondence corruption, the second with 50%, and third with 85%. All of the robust functions were able to cope with 50% outliers with the exception of *Huber* and *L1* which are not redescending

TABLE II: A comparison of the RMS Error (RMSE) for the robust functions at different levels of correspondence corruption. *L1* and *L2* are included for reference. *DCS* reliably achieves low RMSE, though *G-M + Threshold* shows faster convergence. *Huber* has poor outlier rejection as it is not redescending. Best and worst values are marked in blue and red, and ‘—’ indicates failure to converge to a reasonable solution.

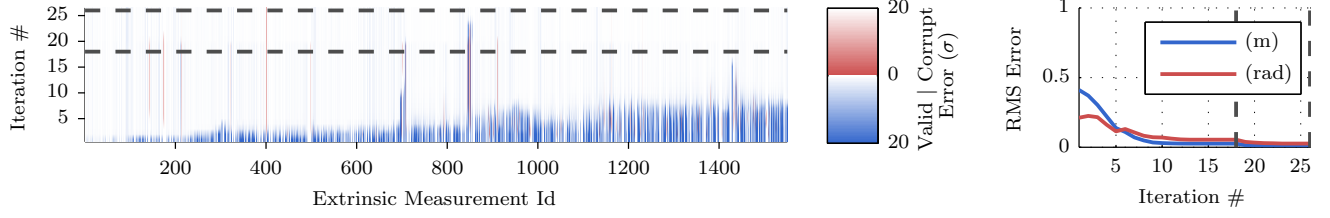
Type	0% Corrupt	50% Corrupt	85% Corrupt
	RMSE (m, rad), iterations		
<i>L2</i>	0.0121, 0.0241	—	—
<i>L1</i>	0.0116, 0.0242	—	—
<i>Huber</i>	0.0121, 0.0241	—	—
<i>Cauchy</i>	0.0123, 0.0245	0.0307, 0.0484, 33	—
<i>G-M</i>	0.0126, 0.0250	0.0160, 0.0291, 22	0.0294, 0.0490, 75
<i>G-M + Th.</i>	0.0120 , 0.0246	0.0141, 0.0265 , 20	0.0286, 0.0486, 78
<i>DCS</i>	0.0121, 0.0242	0.0149, 0.0274, 26	0.0280, 0.0478, 99
<i>Thresh.</i>	0.0120 , 0.0246	0.0138 , 0.0267, 19	0.1181, 0.1739, 90

functions. Only *G-M* and its variant, *DCS*, were able to deal with 85% outliers (which was near their limit). The final outlier rejection results with 85% outliers are shown in Figure 4. Table II shows a breakdown of the final errors for each robust function with varying correspondence corruption.

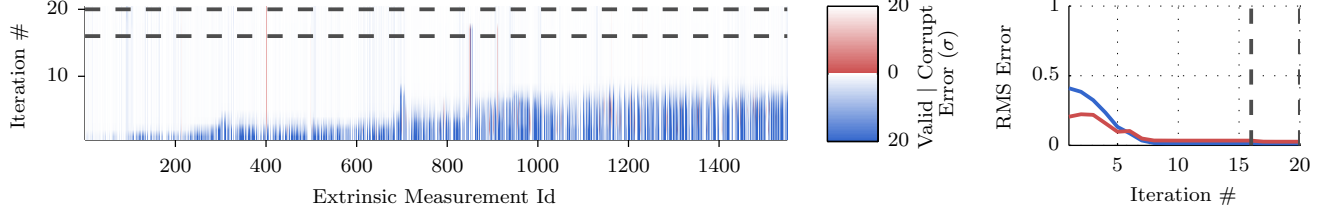
DCS shows a good balance of accuracy and convergence, though *G-M + Threshold* often converged more quickly with similar accuracy. The *Threshold* robust function achieved impressive accuracy once converged, but displayed very poor convergence characteristics on its own.

In order to test performance under varying initial state uncertainty, the initial conditions are the integrated IMU measurements, which drift over time. The beginning of the trajectory has small state uncertainty, and the end, large. To deal with these highly varying initial errors, we apply a changing error deflation over the course of the optimization, initial discussed in (15). We call the changing error deflation values the *deflation schedule*. Additionally, in the *G-M + Threshold* sequence, the last few iterations use the *Threshold* robust function to improve accuracy. We use a naive implementation that iterates Gauss-Newton IRLS until convergence, for each deflation value in the schedule. The solution to the previous is used as the initial conditions for the next. We use a single deflation term for all of the measurements, despite the fact that the initial conditions are much worse at the end of the trajectory than the beginning. Ideally this scaling would be set automatically using the confidence we have on our state estimate; Tong and Barfoot [16] discuss this strategy in depth.

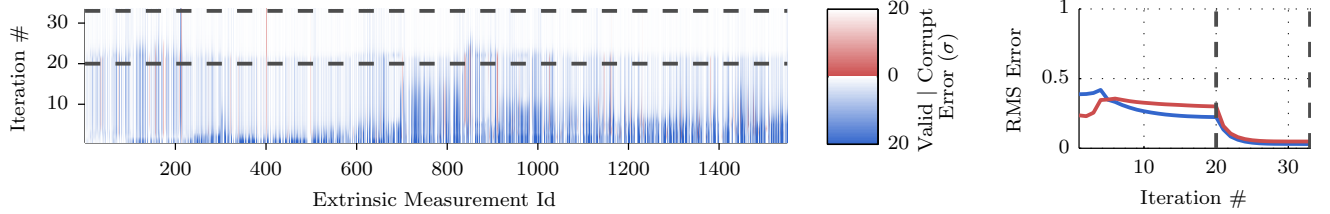
Figures 5 and 6 show the evolution of the Mahalanobis distances for each of the measurements over the course of the optimization. The deflation schedules are listed in the captions for each of the robust cost functions. These figures provide insight into the unique convergence of each robust type. Figure 7 shows the final estimate using *DCS*, under the varying data corruption.



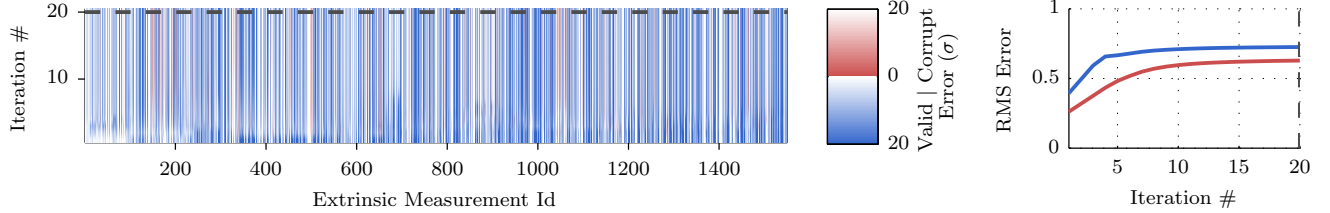
(a) Results for the *DCS* robust function with error deflation of (10, 3) with 50% corrupt. We see that after 10 iterations, there is only one stubborn region in the middle of the dataset that is not white, but has some trusted corrupt measurements and rejected valid measurements. When the error deflation is reduced, this region is corrected, and we see the desired white error state across nearly all of the measurements.



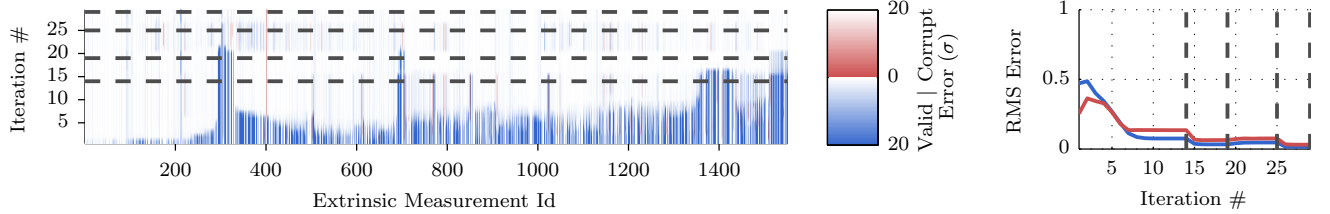
(b) Results for the sequence of *Geman-McClure* and *Threshold* robust functions with error deflation of (10) and (3) with 50% corrupt. The convergence is quite similar to (a); it is slightly slower in the first few iterations, but does a better job with the centre measurements.



(c) Results for the *Cauchy* robust function with error deflation of (10, 3) with 50% corrupt. Cauchy shows slow convergence, as its error rejection is not as strong as the above methods.

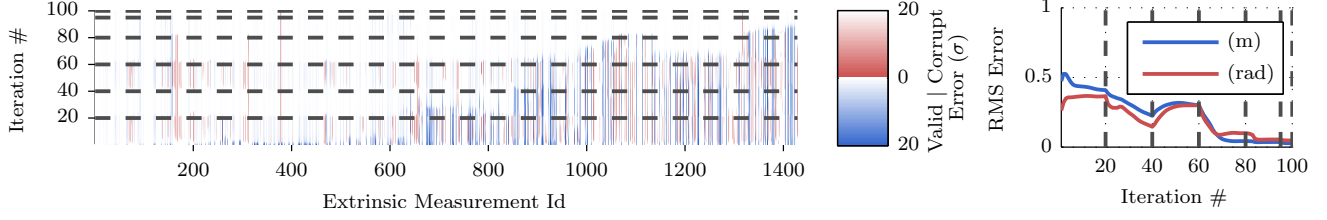


(d) Results for the *Huber* robust function with error deflation of 10 with 50% corrupt. Huber cannot fully reject outliers, as it is not a redescending robust function. No error deflation schedule leads to convergence.

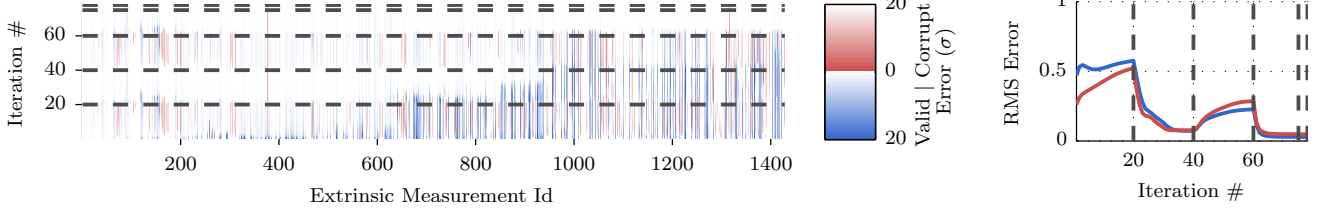


(e) Results for the *Threshold* robust function with error deflation of (20, 3, 30, 3) with 50% corrupt. This function requires a more complex error deflation schedule, since its region of influence ends very abruptly. This abruptness makes it difficult to deal with poor initial conditions, since moderate errors very quickly have no influence. Performance is very good at the beginning of the trajectory with good initial conditions.

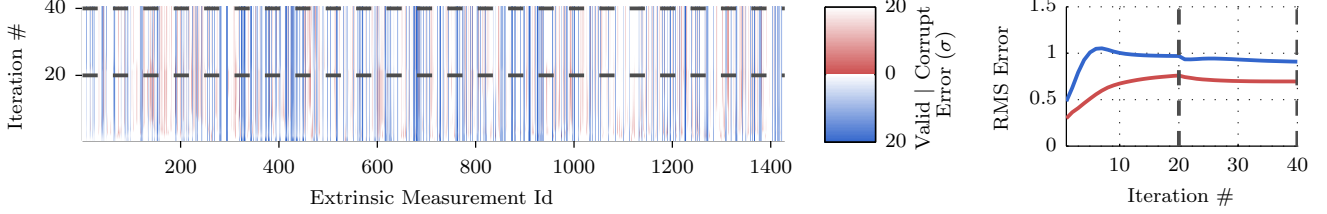
Fig. 5: Results for 50% correspondence corruption. The figures to the left show per-measurement, per-iteration Mahalanobis error. The bottom of the plot is at initial conditions, and the top is the final state. The left is the beginning of the trajectory, and the right is the end. Blue indicates large error on valid measurements, red indicates low error (and thus high weights) on corrupt measurements, and white indicates the desired error – we would like to see a quick fade to white. Grey dashed lines indicate a transition in the error deflation. The figure to the right shows the state estimate vs. ground-truth RMSE for each iteration. The *G-M*-based costs (a-c) show the best convergence and rejection. *Huber* fails to converge as it is not redescending.



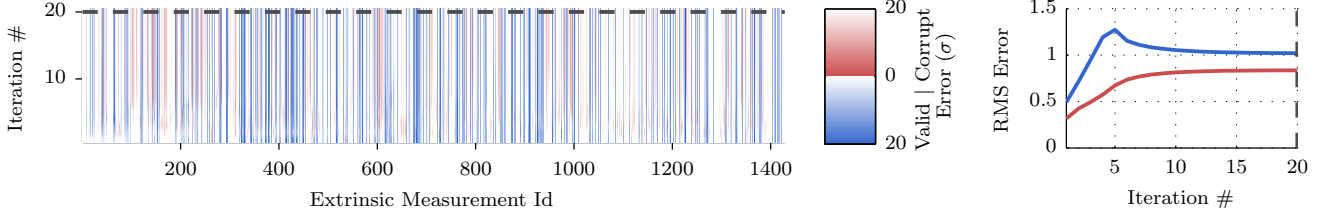
(a) Results for the *DCS* robust function with error deflation of (10, 3, 10, 3, 1, 3) with 85% corrupt. We see that at high iteration numbers, the error is white across all measurements, meaning we have kept the inliers and rejected the outliers.



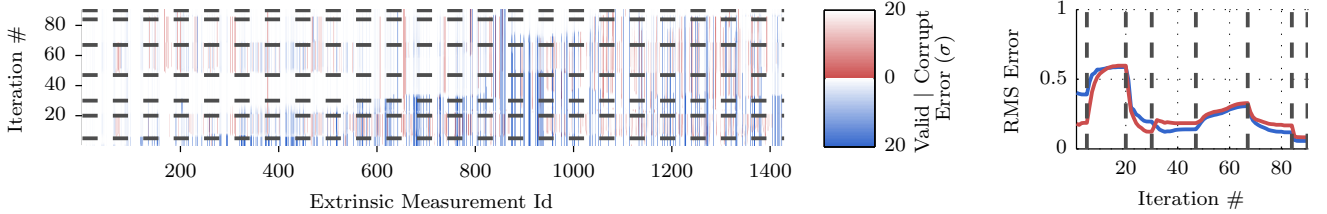
(b) Results for the sequence of *Geman-McClure (G-M)* and *Threshold* functions with error deflation of (15, 3, 15, 3) and (3) with 85% corrupt. We see at iteration 75, *G-M* has kept the inliers and rejected the outliers, *Threshold* is used for the last few iterations to improve accuracy. There are fewer iterations and error deflation values required when compared to (a), despite the fact that *DCS* is *G-M*-based.



(c) Results for the *Cauchy* robust function with error deflation of (10, 3) with 85% corrupt. Though successful with 50% corruption, *Cauchy* does not reject outliers strongly enough to converge under this increased data corruption.

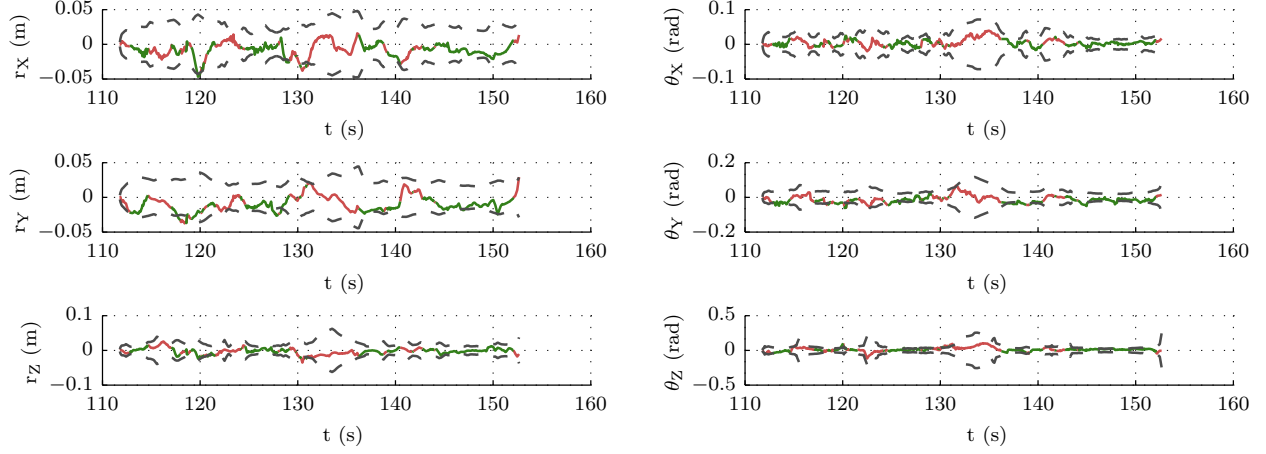


(d) Results for the *Huber* robust function with error deflation of 10 with 85% corrupt. *Huber* cannot fully reject outliers, as it is not a redescending robust function. No error deflation schedule leads to convergence.

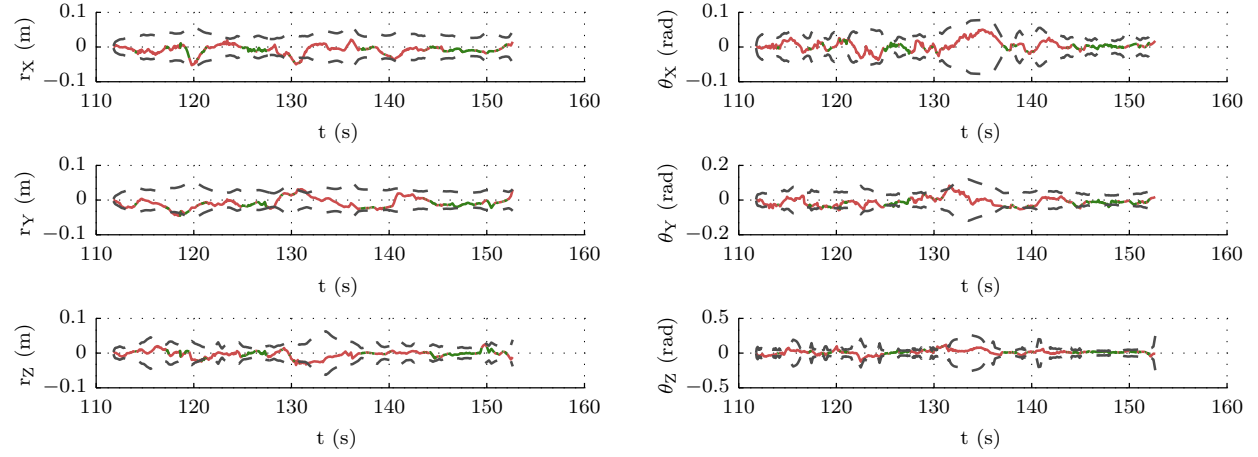


(e) Results for the *Threshold* robust function with error deflation of (5, 30, 3, 20, 30, 10, 5) with 85% corrupt. The *Threshold* function alone does not achieve outlier rejection as good as the *G-M*-based functions. Since its region of influence ends very abruptly, it struggles to recover the second half of the dataset, which begins with poor initial conditions.

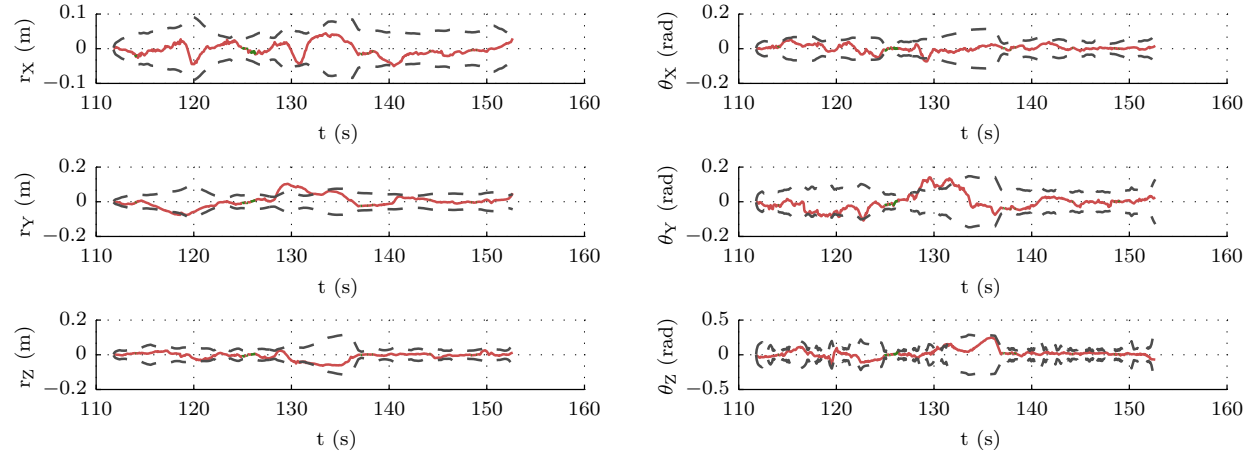
Fig. 6: Results for 85% correspondence corruption. The figures to the left show per-measurement, per-iteration Mahalanobis error. The bottom of the plot is at initial conditions, and the top is the final state. The left is the beginning of the trajectory, and the right is the end. Blue indicates large error on valid measurements, red indicates low error (and thus high weights) on corrupt measurements, and white indicates the desired error – we would like to see a quick fade to white. Grey dashed lines indicate a transition in the error deflation. The figure to the right shows the state estimate vs. ground-truth RMSE for each iteration. The *G-M*-based robust functions (a-c) show the best convergence and rejection. Both *Huber* and *Cauchy* fail to converge.



(a) Error for the L2 cost function with no correspondence corruption. This serves as a baseline for the corruption experiments.



(b) Error for the DCS robust function with 50% Corruption. The error is not much larger than (a).



(c) Error for the DCS robust function with 85% Corruption. The error is notably larger than the other experiments, but is still quite reasonable considering the level of data corruption – nearly the entire trajectory contains fewer than three valid landmarks per image.

Fig. 7: Error plots for the trajectory, using *L2* for the uncorrupted dataset, and *DCS* when corrupted. Three- σ error bounds are given by the black dashed line, showing fairly consistent performance due to good outlier rejection. The red regions of the trajectory contained fewer than three uncorrupted landmarks per image, the green regions contained at least three.

V. CONCLUSIONS

We have shown that using a robust function in an IRLS scheme is an effective way to deal with correspondence outliers commonly present in visual navigation. We have also presented successful localization in the presence of up to 85% correspondence outliers, frequently with fewer than 3 landmarks per image. The comparison of several popular robust functions has shown that the *DCS* robust cost function has a good balance of convergence and accuracy; in general, all of the *G-M*-based robust costs performed well. *G-M* had good convergence but less accuracy, making it useful as an *initial* function. The *Threshold* robust function was accurate under good initial conditions, making it a good function with which to finish.

We can also conclude that both convergence and accuracy are sensitive to the error deflation term (or other equivalent tuning parameter). This term should be carefully selected, especially in the presence of poor initial conditions. Future work will investigate the application of the adaptive threshold of Tong and Barfoot [16] to robust M-estimation.

ACKNOWLEDGEMENTS

We would like to extend our deepest thanks to the Canada Foundation for Innovation, the Canadian Space Agency, and MDA Space Missions for providing us with the financial and in-kind support necessary to conduct this research. This work was also supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN).

REFERENCES

- [1] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment - A Modern Synthesis," *Vis. Algorithms Theory Pract.*, vol. 1883, pp. 298–372, 2000.
- [2] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment," *Int. J. Rob. Res.*, vol. 29, no. 8, pp. 958–980, May 2010.
- [3] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [4] X. Zhang, J. Zhang, A. B. Rad, X. Mai, and Y. Jin, "A novel mapping strategy based on neocortex model: Pre-liminary results by hierarchical temporal memory," in *2012 IEEE Int. Conf. Robot. Biomimetics*, IEEE, Dec. 2012, pp. 476–481.
- [5] F. Rodriguez, V. Frémont, P. Bonnifait, *et al.*, "An experiment of a 3d real-time robust visual odometry for intelligent vehicles," in *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, IEEE, 2009, pp. 1–6.
- [6] A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics*, vol. 16, no. 2, pp. 147–185, 1974.
- [7] G. Hu, K. Khosoussi, and S. Huang, "Towards a reliable SLAM back-end," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 37–43, 2013.
- [8] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust Pose-Graph Loop-Closures with Expectation-Maximization," *Intell. Robot. Syst. (IROS), 2013 IEEE/RSJ Int. Conf.*, pp. 556–563, 2013.
- [9] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image Vis. Comput.*, vol. 15, pp. 59–76, 1997.
- [10] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.
- [11] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 1879–1884.
- [12] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," *2013 IEEE Int. Conf. Robot. Autom.*, pp. 62–69, May 2013.
- [13] P. J. Huber *et al.*, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.
- [14] S. Geman, D. E. McClure, and D. Geman, "A nonlinear filter for film restoration and other problems in image processing," *CVGIP: Graphical models and image processing*, vol. 54, no. 4, pp. 281–289, 1992.
- [15] T. D. Barfoot, "State estimation for aerospace vehicles - aer1513 course assignments," University of Toronto Institute for Aerospace Studies, Tech. Rep., 2011.
- [16] C. H. Tong and T. D. Barfoot, "Evaluation of heterogeneous measurement outlier rejection schemes for robotic planetary surface mapping," *Acta Astronaut.*, vol. 88, pp. 146–162, Jul. 2013.