

Robust Navigation In GNSS Degraded Environment Using Graph Optimization

Ryan M. Watson and Jason N. Gross , *West Virginia University*

Biography

Mr. Ryan M. Watson is a Graduate Research Assistant within the Navigation Laboratory in the Department of Mechanical and Aerospace (MAE) Engineering at WVU and is pursuing his PhD in aerospace engineering. He completed his Master in aerospace engineering at WVU in 2017, with a thesis on the integration of Inertial Navigation in JPL's RTGx Precise Point Positioning software. Mr. Watson earned his B.S. degree in mechanical engineering from WVU in 2014.

Dr. Jason N. Gross is an assistant professor at West Virginia University (WVU) where he directs WVU's Navigation Laboratory. He received his Ph.D. in aerospace engineering from WVU in 2011 and worked at the Jet Propulsion Laboratory (JPL) until January 2014. His research is focused on Guidance, Navigation & Control (GNC) technologies and he currently serves as an associate member of the AIAA's GNC technical committee. He is a member of the ION, the AIAA, and the IEEE.

Abstract

Robust navigation in urban environments has received a considerable amount of both academic and commercial interest over recent years. This is primarily due to large commercial organizations such as Google and Uber stepping into the autonomous navigation market. Most of this research has shied away from Global Navigation Satellite System (GNSS) based navigation. The aversion to utilizing GNSS data is due to the degraded nature of the data in urban environment (e.g., multipath, poor satellite visibility). The degradation of the GNSS data in urban environments makes it such that traditional (GNSS) positioning methods (e.g., extended Kalman filter, particle filters) perform poorly. However, recent advances in robust graph theoretic based sensor fusion methods, primarily applied to Simultaneous Localization and Mapping (SLAM) based robotic applications, can also be applied to GNSS data processing. This paper will utilize one such method known as Incremental Smoothing and Mapping (ISAM2) in conjunction several robust optimization techniques to evaluate their applicability to robust GNSS data processing. The goals of this study are two-fold. First, for GNSS applications, we will experimentally evaluate the effectiveness of robust optimization techniques within a graph theoretic estimation framework. Second, by releasing the software developed and data sets used for this study, we will introduce a new open-source front-end to the Georgia Tech Smoothing and Mapping (GTSAM) library for the purpose of integrating GNSS pseudorange observations.

Introduction

Traditionally, GNSS data is not utilized to its full potential for autonomously navigating vehicles in urban environments. This is largely ascribable to the possibility of GNSS observables be degraded (e.g., multipath, poor satellite geometry) when confronted with an urban environment. However, the inclusion of GNSS data in such systems would provide substantial information to their navigation algorithms. So, the ability to safely incorporate GNSS observables into existing inference algorithms, even when confronted with environments that have the potential to degrade GNSS observables, is of obvious importance.

To overcome the aforementioned issues, we will leverage the advances made within the robotics community surrounding graph-based simultaneous localization and mapping (SLAM) to efficiently and robustly process GNSS data. Within this community, the advances surrounding optimization have been made on two fronts: optimization speed and robustness. For processing speed, the incremental smoothing and mapping (ISAM2) (Kaess et al., 2012) algorithm provides a real-time optimization routine (Whelan et al., 2015) and, as such, will be utilized as the optimization framework for this study. For robust graph optimization, the literature can be divided into two broad subsets: traditional M-Estimators (Huber, 1981), and more recent graph based robust methods (Sunderhauf and Protzel, 2012), (Agarwal et al., 2013), (Olson and Agarwal, 2012).

This paper will evaluate the effectiveness of the mentioned robust optimization techniques when applied specifically to GNSS pseudorange data processing.

The remainder of this paper is organized as follows. First, the technical approach utilized for this study is described, which begins with a discussion on factor graph optimization and then evolves into a discussion on how to make that optimization more robust to erroneous data. Next, the experiential setup and collected data sets are discussed. Then, the results obtained using the previously described models and data sets are provided. Finally, the paper ends with concluding remarks and proposed steps for continued research.

1 Technical Approach

This section provides concise overview of the sensor fusion approach utilized in this study. For the reasons mentioned above, it was decided to use a graph-theoretic approach for sensor fusion as opposed to the more common Kalman (Kalman, 1960) or particle filters (Thrun, 2002). To describe our approach, first, the factor graph (Kschischang and Frey, 2001) is discussed. Then, the evolution from the factor graph to Incremental Smoothing and Mapping (ISAM2) (Kaess et al., 2012) is covered. Next, a discussion is provided on the incorporation of GNSS pseudorange observables into the factor graph framework. Finally, a discussion is provided on methods to make these graph theoretic approaches more tolerant to measurement faults, specifically, in the context of GNSS.

1.1 Factor Graphs

The factor graph was purposed in (Kschischang and Frey, 2001) as a mathematical tool to factorize functions of many variables into smaller subsets. This formulation can be utilized to infer the posterior distribution of the GNSS based state estimation problem. The factorization is represents as a bipartite graph, $\mathcal{G} = (\mathcal{F}, \mathcal{X}, \mathcal{E})$, where there are two types of vertices: the states to be estimated, \mathcal{X} , and the probabilistic constraints applied to those states, \mathcal{F} . An edge \mathcal{E} only exists between a state vertex and a factor vertex if that factor is a constraint applied on that time-step. An example factor graph is depicted in Figure 1, where X_n represents the states to be estimate at time-step n (i.e., user position, user velocity, zenith point troposphere delay, GNSS receiver clock bias), e represents the constraint applied to the state by a measurement (i.e., a GNSS pseudorange observable), and b_n represents a probabilistic constraint applied between time-steps (e.g., incorporation of inertial navigation into the factor graph).

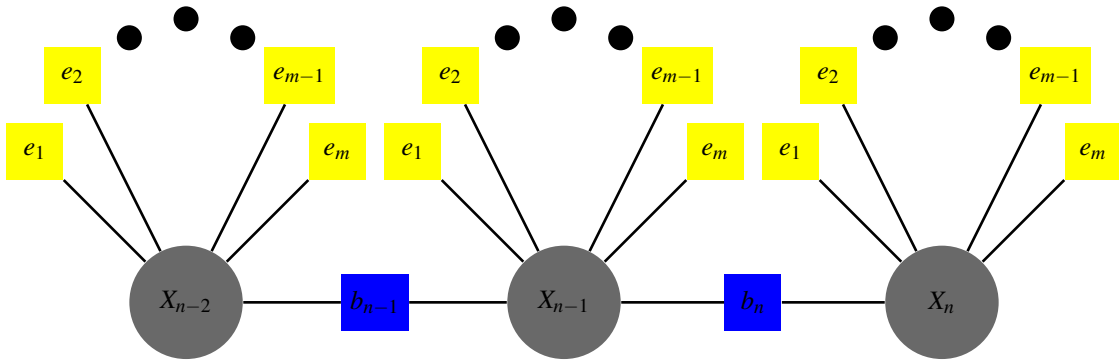


Figure 1: Example Factor Graph

In the graph, e_i represents an error function or probabilistic constraints applied to the state at the specified time-step. When Gaussian noise is assumed, the error function is equivalent innovation residual of the traditional Kalman filter, as shown in Eq 1. Utilizing this information it is easy to see that the optimal state estimate can be calculated through a traditional non-linear least squares formulation (i.e., Levenberg Marquardt (Lourakis, 2005)) that minimizes the error over the graph.

$$e_i = H_i(X_i) - Z_i \quad (1)$$

$$\hat{X} = \operatorname{argmin} \sum_i \|e_i\|_{\Sigma}^2 \quad (2)$$

1.1.1 Incremental Smoothing and Mapping

The factor graph works great when post-processing data; however, it is a batch estimator and cannot effectively operate in real-time when sufficiently large amounts of data have been incorporated into the graph (Indelmana et al., 2013). To overcome this limitation, the Incremental Smoothing and Mapping formulation was developed (Kaess et al., 2012). This formulation moves from an un-directed graphical structure known as the Bayes tree (Kaess et al., 2010). The additional information incorporated into the graph allows the optimizer to re-linearize only the vertices that are affected by the incorporation of the new measurement to the graph (e.g., most of the graph is left unchanged when a new pseudorange observable is incorporated).

1.1.2 Pseudorange Factor

With the general factor graph framework discussed, the discussion can now move to optimization of GNSS navigation applications. To construct the pseudorange factor, the dual-undifferenced GNSS observables are utilized. Because undifferenced data is being used, methods for mitigating other GNSS error-sources (e.g., troposphere and ionosphere delays, and GNSS receiver clock bias) are incorporated in the measurement models.

To mitigate the ionospheric delay, the dispersive nature of the ionosphere is leveraged, and a linear combination of the GPS L_1 and L_2 frequencies (1575.42 MHz and 1227.60 MHz, respectively) is formed to produce ionospheric-free (IF) pseudorange measurements in order to eliminate the delay to first order (Misra and Enge, 2006). The IF pseudorange combination can be seen in Eq. 3: where, f_1 and f_2 are the L_1 and L_2 frequencies, and ρ_{L1} and ρ_{L2} are the pseudorange measurements on the L_1 and L_2 frequencies. The superscript j in Eq. 3 is used to designate the measurement between the platform and satellite j .

$$\rho_{IF}^j = \rho_{L1}^j \left[\frac{f_1^2}{f_1^2 - f_2^2} \right] - \rho_{L2}^j \left[\frac{f_2^2}{f_1^2 - f_2^2} \right] = 2.546\rho_{L1}^j - 1.546\rho_{L2}^j \quad (3)$$

Using the IF combination, the pseudorange measurements are modeled as shown in Eq. 4: where δt_u is the receiver's clock bias, T_z is the tropospheric delay in the zenith direction, and $m(el^j)$ is a user to satellite elevation angle dependent mapping function. In this study, the Black and Eisner model (Black and Eisner, 1984) is used to model the dry component of the troposphere and an estimated parameters is used to compensate for the residual error of the dry delay model and wet delay.

$$\hat{\rho}_{IF}^j = \|x_{sat}^j - x_{user}\|^2 + c\delta t_u + T_z m(el^j) + \epsilon_p^j \quad (4)$$

In addition to the modeled parameters shown above, several additional error terms are generally incorporated in a dynamic platforms observation model. Some of the unmentioned error terms (i.e., relativistic, satellite clock and gravitation delay) are incorporated into our model through the utilization of GPSTk (Tolman et al., 2004), which is an open-source GNSS software package. All additional un-modeled error components are included in Eq. 4 as the ϵ term.

Using the modeled pseudorange observable, we can construct the pseudorange factor as shown in Eq. 5. In Eq. 5, the superscript j designate the measurement between the platform and j^{th} satellite, ρ_{IF}^j is the observed pseudorange value, $\hat{\rho}_{IF}^j$ is the estimated pseudorange value as calculated in Eq. 4, and Σ is the uncertainty in the pseudorange observable.

$$\|e_{pr}^j\|_{\Sigma_j}^2 = \|\rho_{IF}^j - \hat{\rho}_{IF}^j\|_{\Sigma_j}^2 = e_{pr}^j \Sigma_j^{-1} e_{pr}^j \quad (5)$$

1.2 Fault Tolerance

Due to the degraded nature of the GNSS data in urban environments (i.e., subject to multipath, poor satellite visibility), methods to make the state estimation process more robust must be incorporated into the optimization routine. For this study, we will broadly classify these robust optimization techniques as traditional M-Estimators and more recent graph based techniques. Both methods are described in detail below.

1.2.1 Traditional M-Estimators

The field of M-Estimators dates back to the seminal paper published by Huber (Huber, 1964), which was later extended into a comprehensive survey on the subject (Huber, 1981). The field of research related to M-estimation can be reduced to minimizing the influence of erroneous data by replacing the L_2 cost function with a modified cost function, $\rho(e_i)$, which penalizes observables that do not conform to the user defined observation model.

All cost functions can be classified as being redescending or non-redescending. A redescending cost function conforms to the property that $\lim_{e_i \rightarrow \infty} \psi(e_i) = 0$, where $\psi(e_i) = \frac{d\rho(e_i)}{de_i}$, which implies that the weight approaches 0 as the magnitude of the residual approaches ∞ . For this study, two M-estimators were selected, one M-estimator that is redescending — the Cauchy cost function (Olive, 2008) — and one M-estimator that is non-redescending — the Huber cost function (Huber, 1964).

The m-estimator utilized in this study are shown in greater detail in Table 1. In Table 1, $\rho(e_i)$ represents the modified cost function, $\psi(e_i)$ is the first derivative of the cost function with respect to the residual, $w(e_i)$ specifies the weight applied to corresponding entries in the information matrix, and k is the user defined M-estimator kernel width.

Table 1: Selected M-Estimators

M-Estimator	$\rho(e_i)$	$\psi(e_i)$	$w(e_i)$
Huber	$\begin{cases} \text{if } x \leq k & \frac{e_i^2}{2} \\ \text{if } x > k & k(e_i - \frac{k}{2}) \end{cases}$	$\begin{cases} e_i \\ k \text{sign}(e_i) \end{cases}$	$\begin{cases} 1 \\ \frac{k}{ e_i } \end{cases}$
Cauchy	$\frac{k^2}{2} \log(1 + \frac{e_i^2}{k^2})$	$\frac{e_i}{1 + e_i^2/k^2}$	$\frac{1}{1 + e_i^2/k^2}$

1.2.2 Graph Based Approaches

Next, three more recent advances in robust optimization: switchable constraints, dynamic covariance scaling, and max-mixtures are discussed in the context of robust GNSS optimization.

1.2.3 Switch Constraints

Switchable Constraints were first introduced by (Sunderhauf and Protzel, 2012) as a method to reject false positive loop-closure constraints in the simultaneous localization and mapping (SLAM) problem. Additionally, this method has been evaluated for its ability to mitigate the affect of multipath in urban environment (Sünderhauf et al., 2012, 2013). The robustness of this optimization technique is granted through the incorporation of a new switchable constraint for every factor of interest in the graph. The switchable constraint can be thought of as an observation weight that is be optimized concurrently with the state estimates. The modified cost function that must be optimized to find the minimum error over the entire graph is shown in Eq. 6, where $s \in \{0, 1\}$ is the switchable constraint, γ_i is the initial estimate for the switch variable, and $\psi(\cdot)$ is a real-valued function such that $\psi(s_{ij}) = w_{ij} : \mathcal{R} \rightarrow \{1, 0\}$. A graphical depiction of the modified graph is shown in Figure 2, where the residual associated with measurement m at epoch n exceeds the residual threshold.

$$\hat{X}, \hat{S} = \underset{i}{\operatorname{argmin}} \sum_i \|e_{pr}^i\|_{\Sigma}^2 + \sum_i \|\psi(s_i) * e_{pr}^i\|_{\Sigma}^2 + \sum_i \|\gamma_i - s_i\|_{\Xi}^2 \quad (6)$$

1.2.4 Dynamic Covariance Scaling

A pitfall of the switchable constraint robust optimization algorithm is that additional latent variable are added to the optimization process every time a observable's residual exceed the defined threshold. This means that the optimizer is not only working over the original search space composed of the state vector, but is now also optimizing over all added switchable constraints. The inclusion of additional variables into the optimization process not only increases computation cost, but could also potentially decrease convergence speed. To overcome these issues, a closed formed solution to switchable constraints was introduced in (Agarwal et al., 2013) as dynamic covariance scaling. The closed formed solution can be see in Eq. 7, where Φ is the inverse of the prior uncertainty on the switchable constraint, and χ is the initial error in the state.

$$s = \min(1, \frac{2\Phi}{\Phi + \chi^2}) \quad (7)$$

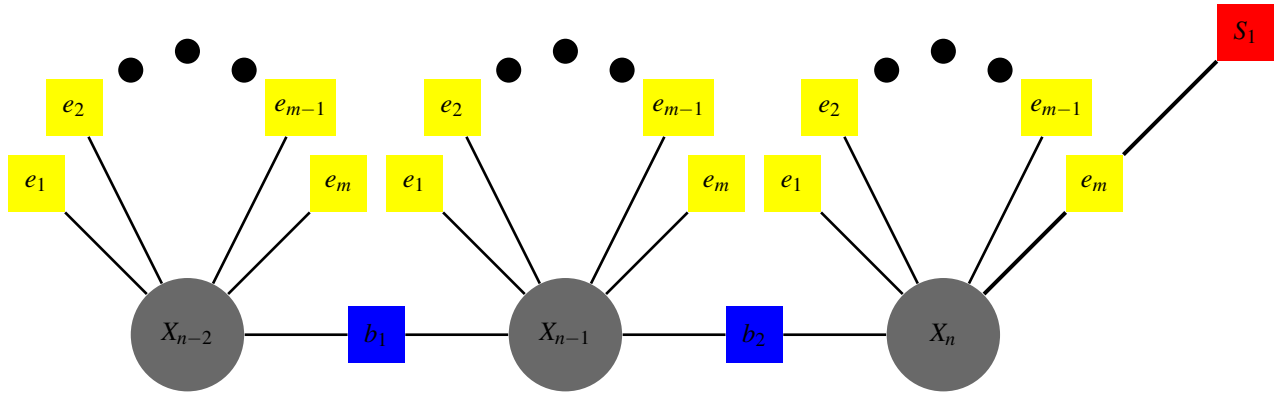


Figure 2: Example Factor Graph with Switchable Constraint

1.2.5 Max-Mixtures

The two previous robust models are still confined to uni-modal Gaussian error distributions, which in many situations will not fully capture the true distribution of the data in trying environments. To overcome this limitation, the next robust method extends that traditional uni-modal Gaussian factor graph optimization to a Gaussian mixture model. Generally, to move from a traditional uni-modal Gaussian optimization to a more complication distribution, the sum of multiple Gaussian components is utilized, as shown in Eq. 8.

$$p(z_i|x) = \sum_i w_i \mathcal{N}(\mu_i, \Lambda_i) \quad (8)$$

However, this solution greatly complicates the calculation of the maximum likelihood estimate because the logarithm cannot be “pushed” inside of the summation operator. To overcome this additional complexity, (Olson and Agarwal, 2012) recommend the utilization of the the max operator as a Gaussian component selector to approximation the true Gaussian summation. For GNSS data processing, this means that each observable can be modeled using two independent distributions: one distribution that defines data free of outliers, while a second distribution represents faulty data (i.e., the null hypothesis). The null hypothesis can be modeled as a Gaussian distribution centered at the mean of the error-free observable distribution, but with a substantially larger variance. This null hypothesis is tested against the hypothesis within the optimizer to iteratively find the weighting applied to the Jacobian and information matrix.

2 Algorithm Implementation And Performance

2.1 Software Implementation

For implementation, this work uses the Georgia Tech Smoothing and Mapping Library (GTSAM) (Dellaert, 2012) as the estimation engine. GTSAM is an open-source C++ library that abstracts the process of formulating an estimation problem as a factor graph. Important to our proposed application of robust GNSS, it allows developers to easily add custom measurement factor types, which allows for the flexible integration of different sensors. GTSAM also includes a variety of nonlinear optimizers for efficiently solving graphs over specified batches of measurements.

2.2 Experimental Data Sets

To evaluate the algorithms positioning performance, two urban data-sets were collected – the East-North ground trace for one of the collected data-sets is depicted in Figure 5 – that contain multiple scenarios which are known to degrade GNSS data. A depiction of the encountered scenarios are provided in Figure 3. To see how these adverse scenarios can affect GNSS positioning performance, the real-time, tightly-coupled GNSS/INS NovAtel positioning solution is compared to the post-processed RTK solution in Figure 4. As can be seen in Figure 4, the adverse environments when driving in an urban setting can induce large fluctuations in positioning error.

These data-sets provide approximately 4 hours of GNSS/INS data. Each data-set contains dual-frequency GNSS pseudo-range and carrier-phase observables at 10 Hz. Additionally, 50 Hz IMU data measured by the navigation grade IMU contained in the NovAtel SPAN system is included.

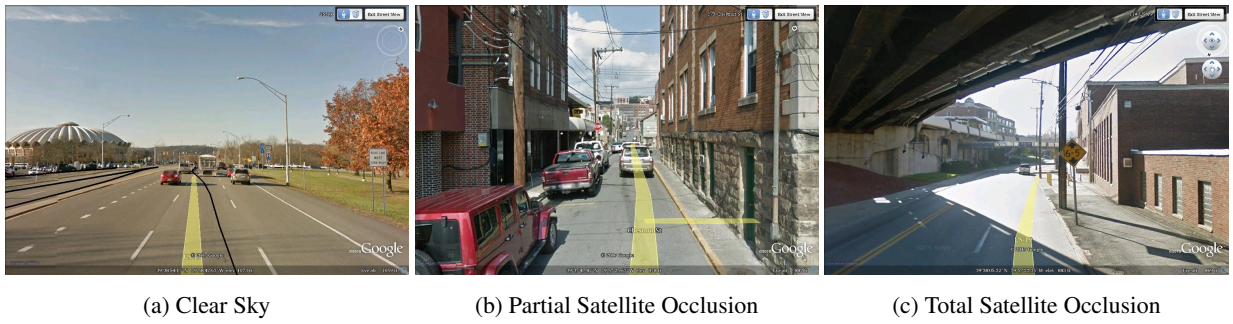


Figure 3: Multiple scenarios encountered during data collect that are known to degrade GNSS observables

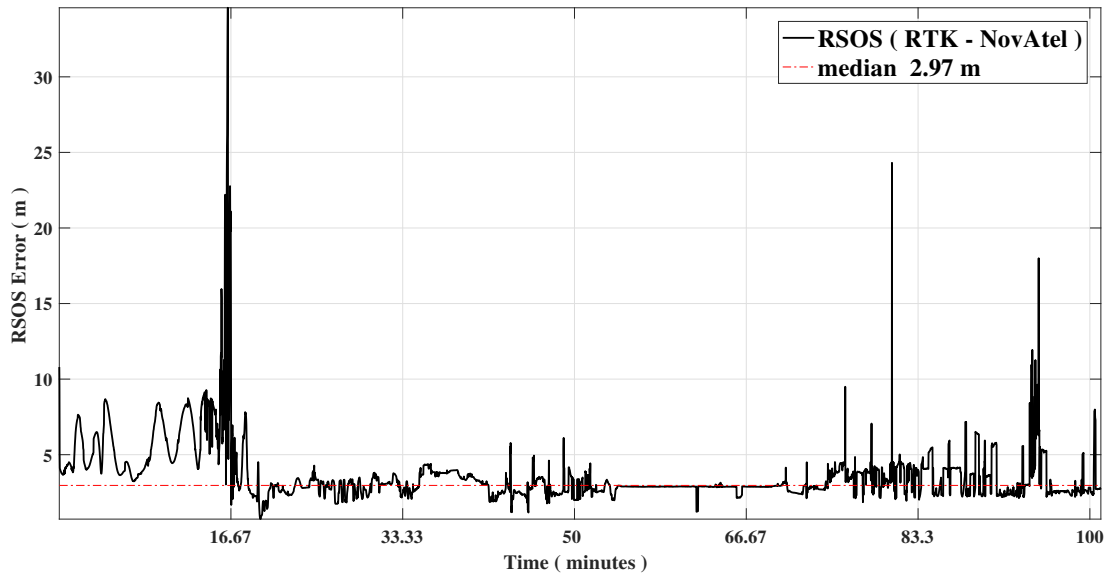


Figure 4: RSOS Positioning Difference Between RTK and NovAtel’s real-time positioning solution

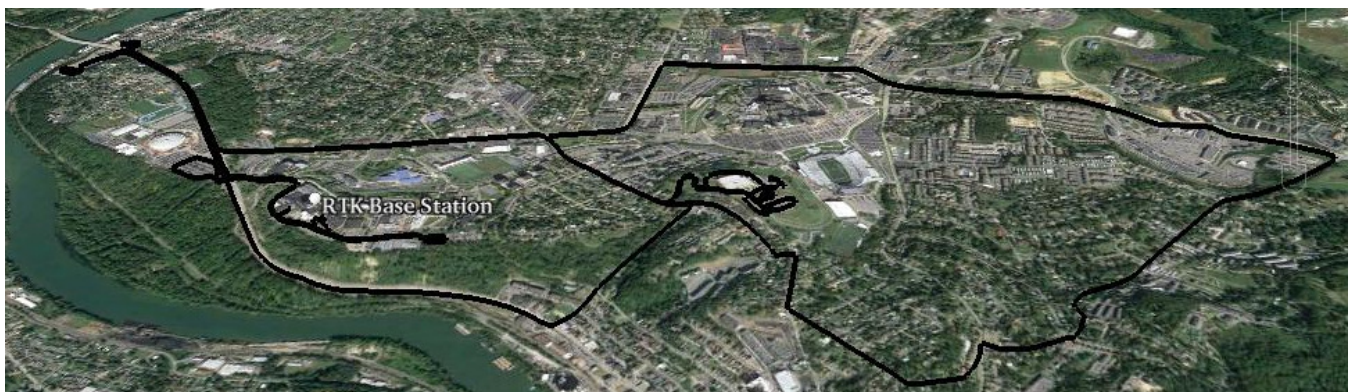


Figure 5: East-North ground trace for example data-set

2.3 Reference Solution

To generate an accurate comparison solution, static data was collected concurrently with the roving platform. The static data sets provide 1 Hz dual-frequency GNSS pseudorange and carrier-phase observables collected by a NovAtel SPAN receiver.

This data is used to generate a Kalman filter/smoothing Carrier-Phase Differential GPS (CP-DGPS) positioning solution with respect to the local reference station. The CP-DGPS reference solution was calculated using RTKLIB (Takasu, 2009).

2.4 Evaluation

Using the data-set shown in Figure 5, the factor graph formulation with the above mentioned noise models can be tested against the RTK positioning solution. As a visual comparison, the RTK positioning solution is compared to the solution obtained when utilizing a factor graph with L_2 optimization in Figure 6. As seen in the figure, the two solutions agree fairly well.

The results for the six cost functions detailed in the previous sections is provided in Table 2. This Table provides the mean, median, and max deviation of Residual Sum of Squares (RSOS) difference between the RTK positioning solution and the factor graph solution. From this result it should be noted that no one optimization routine has a clear advantage. This may be attributed to the fact that — although the data was collected in an environment that will degrade the observables (i.e., building induced multipath, and poor satellite geometry) — the environment may not have been harsh enough to fully leverage the benefits that can be gained by the robust noise models.

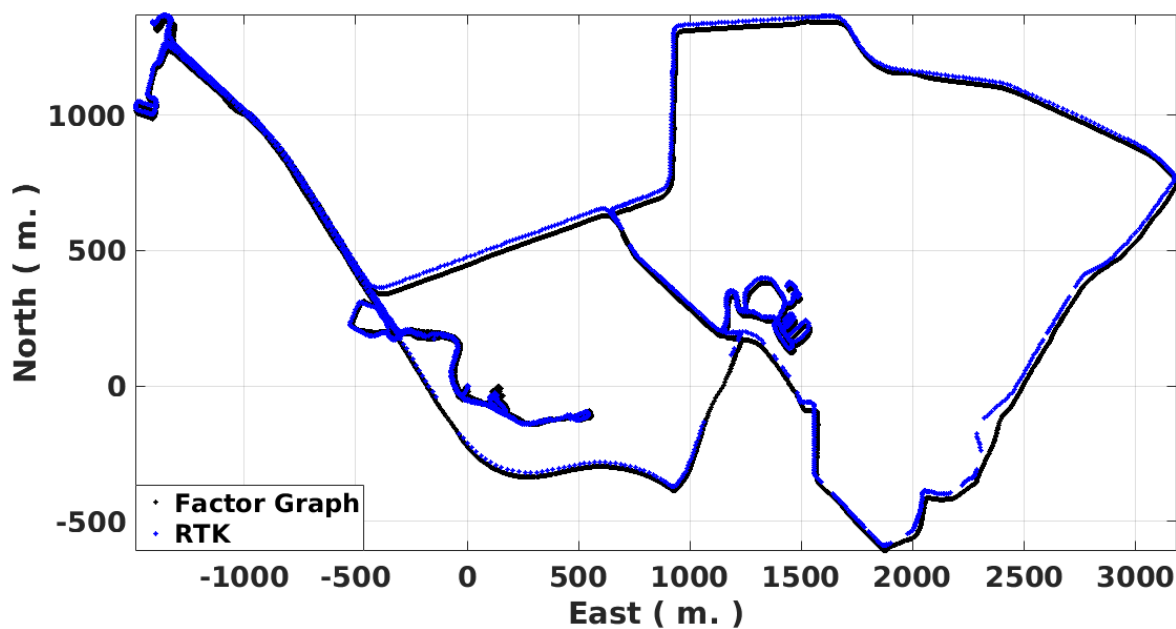


Figure 6: East vs. North Positioning Comparison between RTK and Factor Graph

To more clearly understand when the robust noise models become beneficial, simulated faults were incorporated into the study. The artificial faults were simulated as Gaussian random variable with a mean of zero and a standard deviation of 50 meters. The simulation faults were added onto the measured pseudorange observable at random, where it is possible that up to 49% of the observables can contain artificial faults.

With the artificial faults added to the pseudorange observables, we will first look at how the traditional M-Estimators, which are detailed in Table 1, perform. The relative error increase as the percentage of faults increases (i.e., each estimator is compared to the result obtained by that estimator when no artificial faults were present on the observables) is depicted in Figure 7. From Figure 7, a clear advantage can be seen with the M-Estimators with respect to both the mean and median RSOS positioning error when compared to traditional L_2 optimization. When a comparison is conducted between the M-Estimators, a relatively small positioning error decrease is present when the Cauchy cost function is utilized in place of the Huber. This benefit was also noted in (Lee et al., 2013), where they were analyzing robust loop closure methods for solving the SLAM problem.

The same analysis was conducted for the graph based robust inference methods. The relative positioning error increase as the percentage of artificial faults varies can be seen in Figure 8 for the graph based methods. From this plot it can be noted that the dynamic covariance scaling and max-mixtures methods closely follow the trend depicted in Figure 7 by the M-Estimators.

Table 2: Optimization result comparison of several cost functions

	median (m)	mean (m)	max (m)
L_2	29.69	28.71	147.36
Huber	29.69	28.71	147.36
Cauchy	29.67	28.67	147.15
DCS	29.70	28.71	147.36
Switch Factor	29.46	28.88	148.31
Max-Mix	29.73	28.42	147.1

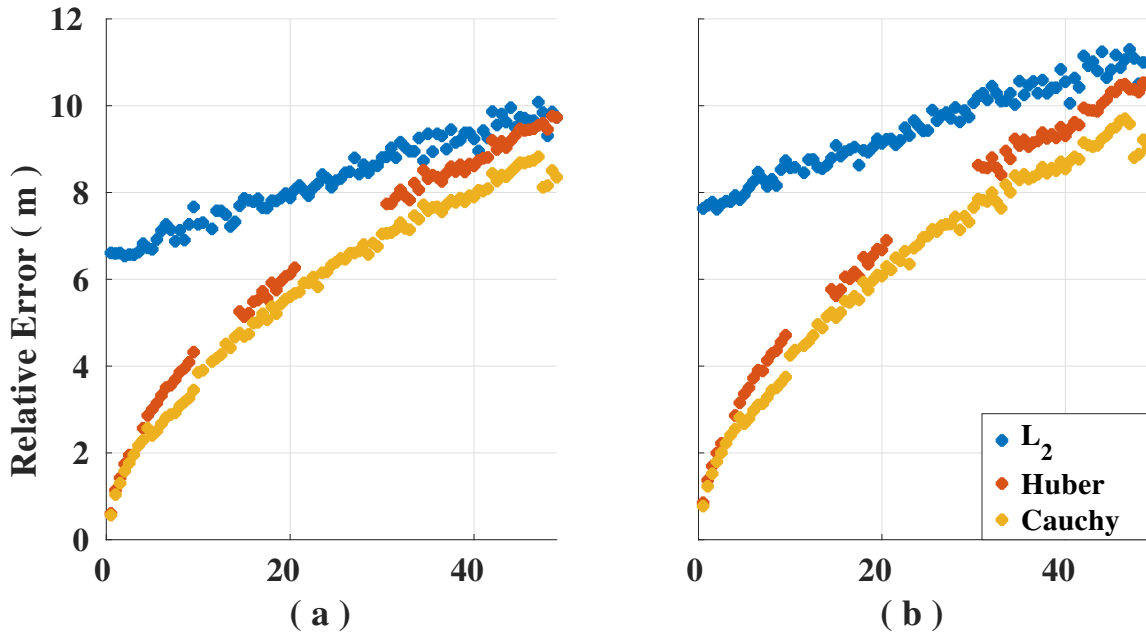


Figure 7: Positioning error increase for M-Estimators as artificial faults are added to the observables where (a) depicts the median relative RSOS positioning error and (b) depicts the mean RSOS positioning error.

This is to be expected because both robust methods are generalizations of M-Estimators, where DCS is a generalization of the Geman-McClure cost function (Agarwal et al., 2013) and max-mixtures is a generalization of the corrupted Gaussian cost function (Olson and Agarwal, 2012).

Another interesting results obtained from Figure 8 is how relatively little the positioning performance is affected when switchable constraints are utilized. This is most clearly depicted in panel (a) of Figure 8, where the relative error is almost constant as the percentage of faults increases from 0–49%.

To more thoroughly analyze the switchable constraint, we can first look at the distribution the solved-for weighting parameters as the percentage of faults are increased. The distribution of the optimized weighting parameter is depicted in

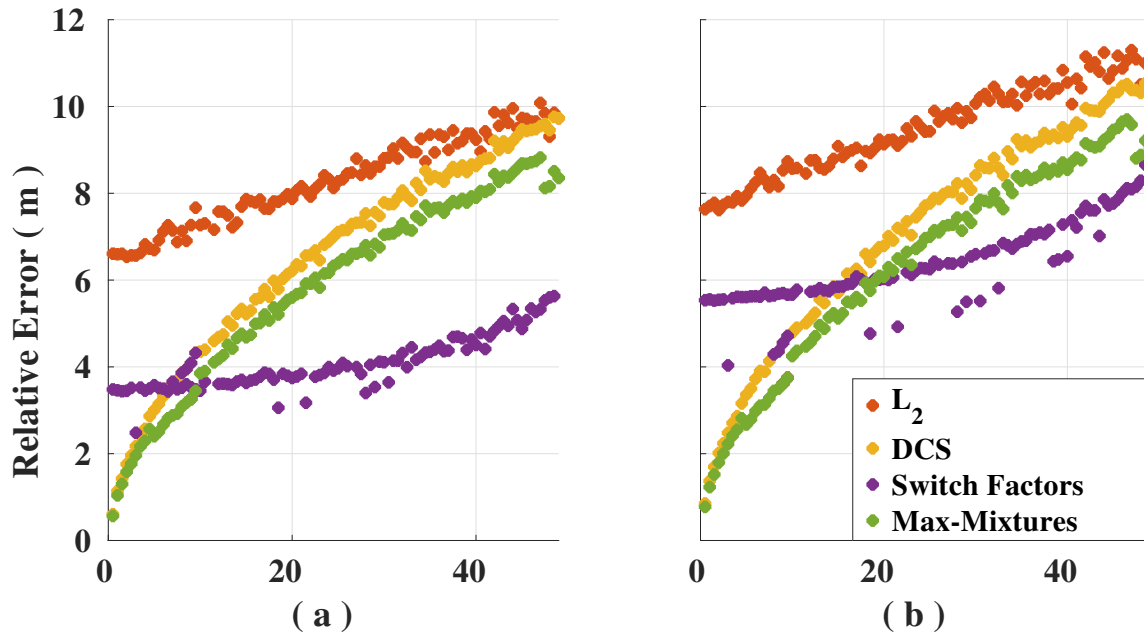


Figure 8: Positioning error increase for graph based robust methods as artificial faults are added to the observables where (a) depicts the median relative RSOS positioning error and (b) depicts the mean RSOS positioning error.

Figure 9, where panel (a) shown the distribution when no artificial faults are added to the observables and panel (b) shows the distribution when 25 %of the observations contain faults. Next, if we set an arbitrary threshold, we can use the switchable constraint weighting parameter as a classifier to denote when the optimizer believes an observable is faulty. Utilizing this technique, the classification accuracy for several weighting thresholds is shown in Figure 10.

Finally, we can evaluate the execution speed for each optimization scheme. To evaluate the execution speed, a 6 minute segment of the data-set depicted in Figure 5 was utilized. Using this data segment, four of the optimization routines were compared and the results are displayed in Figure 11. From this figure it can be seen that L_2 optimization and the traditional M-Estimator execute at approximately the same speed. Additionally, it can be seen that the execution speed of the Max-Mixtures algorithm is being noticeably affected as the graph grows.

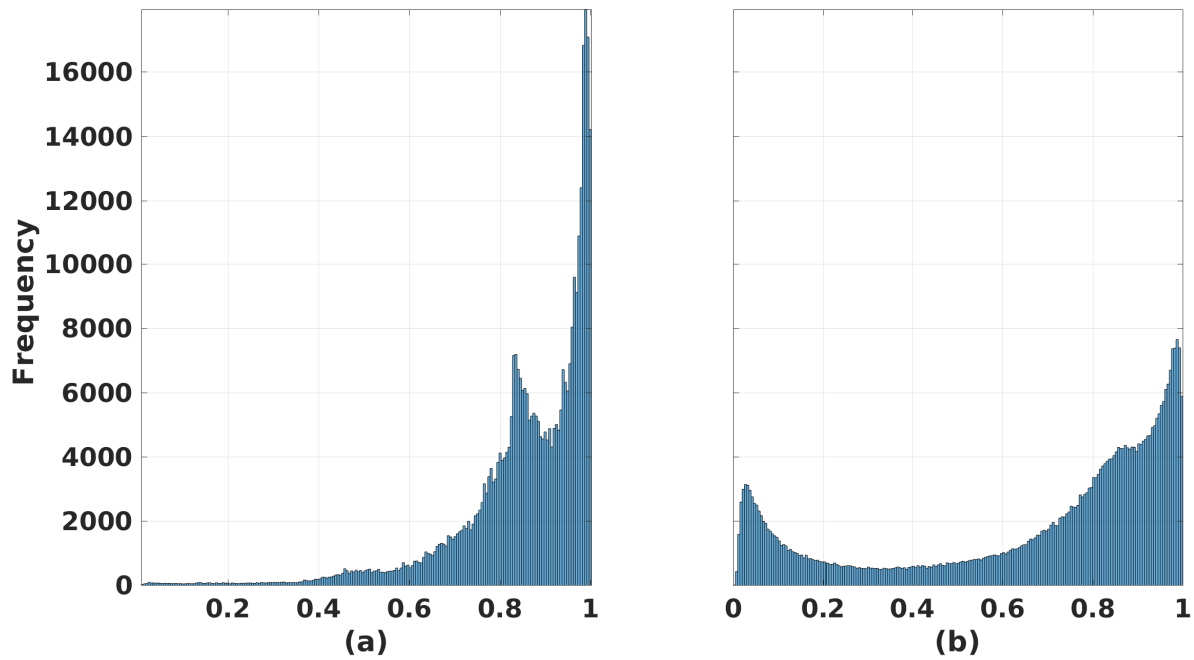


Figure 9: Switchable constraint weighting function distribution for (a) no artificial faults added to the graph and (b) 25% of the pseudorange observables have simulated faults.

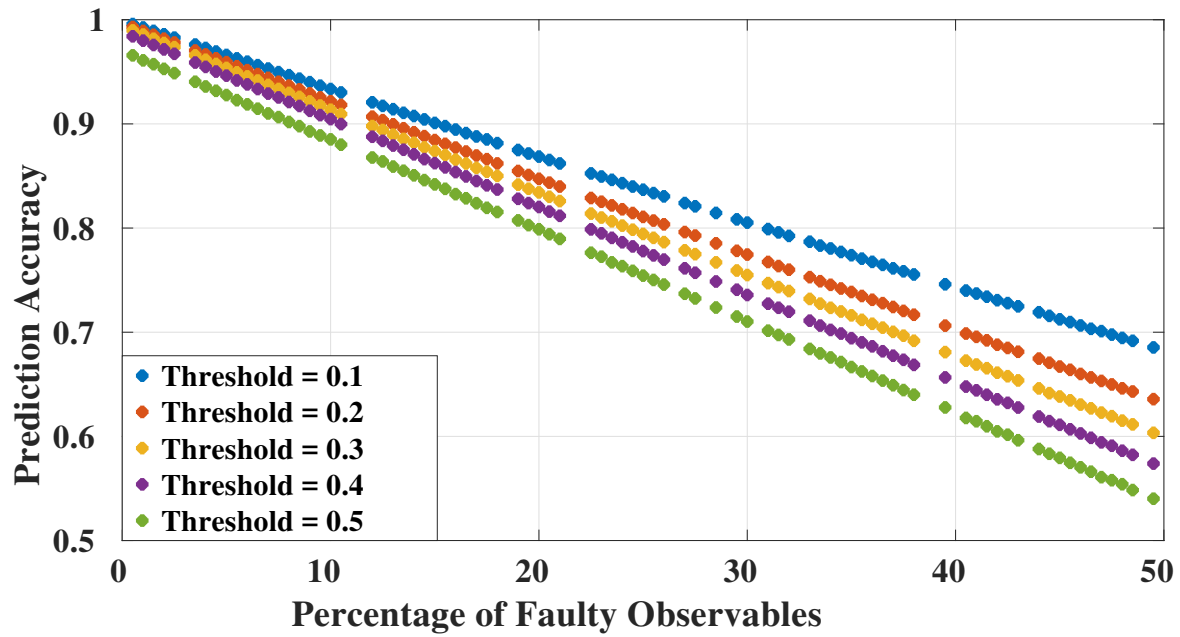


Figure 10: Switchable constraint outlier classification accuracy as percentage of faults increase.

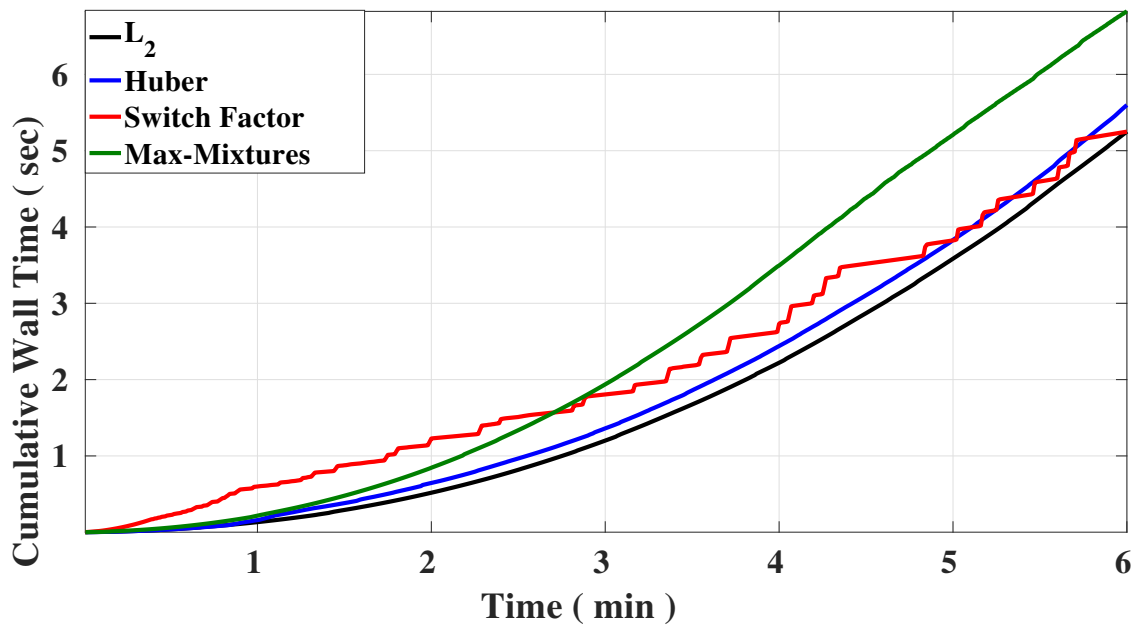


Figure 11: Wall-time comparison for the robust optimization techniques.

3 Conclusion

For autonomous navigation systems, there is a requirement that the system can robustly infer its desired states when confronted with adverse environments. To meet this requirement, we evaluated several robust optimization techniques from various fields for the application of GNSS pseudorange data processing in an urban environment. From this analysis, it has been shown that traditional M-Estimators can aid the optimization scheme when confronted with an adverse environment; however, it has also been shown that including Switchable Constraints in the graph considerably outperforms the other methods implemented in this study, specifically as the number of faulty observables increases.

Additionally, to allow the GNSS community to build upon the advances made within other research communities, the software developed and data sets used for this evaluation has been released publicly at <https://github.com/wvu-navLab/RobustGNSS>, which provides a GNSS front-end integrated with several robust noise models to the GTSAM library.

Acknowledgment

The material in this report was approved for public release on the 26th of June 2017, case number 88ABW-2017-3109.

References

- 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*, 2013.
- H. Black and A. Eisner. Correcting Satellite Doppler Data for Tropospheric Effects. *Journal of Geophysical Research*, 89: 2616, 1984.
- F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- P. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- P. Huber. *Robust Statistics*. Wiley New York, 1981.
- V. Indelmana, S. Williams, M. Kaess, and F. Dellaert. Information Fusion In Navigation Systems Via Factor Graph Based Incremental Smoothing. In *Robotics and Autonomous Systems*, volume 61, 2013.
- M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping. In *International Workshop on the Algorithmic Foundations of Robotics*, 2010.
- M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *The International Journal of Robotics Research*, 31(2), 2012.
- R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- F. Kschischang and B. Frey. Factor Graphs and the Sum-Product Algorithm. In *IEEE Transactions on Information Theory*, volume 42, 2001.
- G. H. Lee, F. Fraundorfer, and M. Pollefeys. Robust pose-graph loop-closures with expectation-maximization. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 556–563. IEEE, 2013.
- M. Lourakis. A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar. 2005.
- P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.
- D. J. Olive. Applied Robust Statistics. *Preprint M-02-006*, 2008.
- E. Olson and P. Agarwal. Inference on Networks of Mixtures for Robust Robot Mapping. 2012.
- N. Sunderhauf and P. Protzel. Switchable Constraints for Robust Pose Graph SLAM. In *Intelligent Robots and Systems*, 2012.
- N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel. Multipath mitigation in gnss-based localization using robust optimization. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 784–789. IEEE, 2012.

- N. Sünderhauf, M. Obst, S. Lange, G. Wanielik, and P. Protzel. Switchable constraints and incremental smoothing for online mitigation of non-line-of-sight and multipath effects. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 262–268. IEEE, 2013.
- T. Takasu. RTKLIB: Open Source Program Package for RTK-GPS. In *FOSS4G 2009 Tokyo, Japan, November 2, 2009*.
- S. Thrun. Particle Filters in Robotics. In *In Proceedings of Uncertainty in AI, 2002*.
- B. Tolman, R. B. Harris, T. Gaussiran, D. Munton, J. Little, R. Mach, S. Nelsen, and B. Renfro. The GPS Toolkit: Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, 2004.
- T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald. Real-Time Large-Scale Dense RGB-D SLAM With Volumetric Fusion. *The International Journal of Robotics Research*, 34(4-5), 2015.