

Switchable Constraints and Incremental Smoothing for Online Mitigation of Non-Line-of-Sight and Multipath Effects

Niko Sünderhauf, Marcus Obst, Sven Lange, Gerd Wanielik and Peter Protzel

Abstract—Reliable vehicle positioning is a crucial requirement for many applications of advanced driver assistance systems. While satellite navigation provides a reasonable performance in general, it often suffers from multipath and non-line-of-sight errors when it is applied in urban areas and therefore does not guarantee consistent results anymore. Our paper proposes a novel *online* method that identifies and excludes the affected pseudorange measurements. Our approach does not depend on additional sensors, maps, or environmental models. We rather formulate the positioning problem as a Bayesian inference problem in a factor graph and combine the recently developed concept of switchable constraints with an algorithm for efficient *incremental* inference in such graphs. We furthermore introduce the concepts of auxiliary updates and factor graph pruning in order to accelerate convergence while keeping the graph size and required runtime bounded. A real-world experiment demonstrates that the resulting algorithm is able to successfully localize despite a large number of satellite observations are influenced by NLOS or multipath effects.

I. INTRODUCTION

Accurate and reliable positioning is a key prerequisite for many applications of advanced driver assistance systems (ADASs) in the automotive domain, intelligent transportation systems (ITSs) or location based services (LBSs) for pedestrians. Such applications, e.g. road charging [1] or dangerous good transports, rely on stable, precise, and up to date estimates of the current position of the vehicle or user. Consumer-class receivers for satellite navigation systems, e.g. GPS or GLONASS, are inexpensive and highly integratable and provide road-level accurate position estimates under good conditions. However, especially in urban areas, non-line-of-sight and multipath effects remain a constant challenge and lead to heavily biased and unreliable position estimates. *Multipath* effects occur when satellite signals are received multiple times, e.g. directly and via reflections on buildings or the ground. If the signal is solely received via reflections, the term *non-line-of-sight* (NLOS) observation is used instead. Fig. 1 illustrates both situation. Since both effects cause highly erroneous position estimates, it is necessary to remove the affected pseudorange observations before attempting to solve for the receiver’s position.

In this paper we therefore propose a novel online method to identify and remove multipath or NLOS measurements. Our method uses *only* the pseudorange data from the GNSS-receiver. No additional information (e.g. velocity or yaw rate) or models (e.g. a map containing the road layout or information on the building structure) are needed.

All authors are with the Faculty of Electrical Engineering and Information Technology, Chemnitz University of Technology, 09111 Chemnitz, Germany
niko.suenderhauf@etit.tu-chemnitz.de

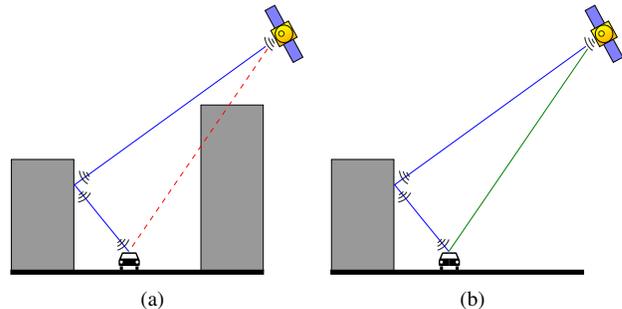


Fig. 1. The non-line-of-sight (left) and the multipath (right) problem are two common challenges for urban satellite-based positioning. In both cases, the resulting position estimate will be heavily biased.

We model the GNSS-based positioning problem as a factor graph problem and show how it can be solved incrementally, i.e. in an *online* fashion. Our paper applies the concept of *switchable constraints* [2], [3] that were recently introduced in the domain of SLAM (Simultaneous Localization and Mapping) in autonomous robotics as a tool for solving factor graph problems in the presence of outlier measurements.

This paper also extends our previous work [4] on multipath and NLOS mitigation where we applied a *batch* solver with our switchable constraints and pointed out that in order to be qualified for real-world applications, the method has to be transferred to an online, *incremental* solver.

We will begin our paper with a short review of related work, factor graphs, inference methods and the concept of switchable constraints. We then explain how to formulate the positioning problem using the factor graph framework and describe the extensions to iSAM2 [5], the factor graph inference algorithm utilized here. The evaluation of the algorithm on a real-world dataset and discussions conclude the paper.

II. RELATED WORK

Different approaches for multipath mitigation are known to the literature, besides expensive hardware-related approaches like using special antenna designs (e.g. choke ring) or antenna arrays. For instance [6] discusses the application of RAIM (Receiver Autonomous Integrity Monitoring) that is however limited to exclude only single satellites from the solution. [7] proposes to actively determine occluded satellites with the help of an omnidirectional infrared camera mounted on the vehicle which is disadvantageous because an additional sensor is required. [8] proposes to identify multipath observations by using information about the local

building structure, i.e. a database of building positions and dimensions. Although this approach worked well, it requires a large knowledge base of accurate 3D building structure information that has to be kept up to date.

III. FACTOR GRAPHS AND SWITCHABLE CONSTRAINTS

Factor graphs are bipartite undirected graphs and have been proposed by [9] as a general tool to model factorizations of large functions with many variables into smaller local subsets. The idea can be applied to general probabilistic estimation problems where the joint conditional probability distribution one wants to estimate can be expressed as a product over several single *factors*. These factors are formed according to the dependency structure between the hidden variables \mathcal{X} and the given evidence \mathcal{Z} (e.g. measurements or a-priori knowledge):

$$P(\mathcal{X}|\mathcal{Z}) = \prod_i P_i(\bar{\mathcal{X}}_i|\bar{\mathcal{Z}}_i) \quad (1)$$

where $\bar{\mathcal{X}}_i \subseteq \mathcal{X}$ and $\bar{\mathcal{Z}}_i \subseteq \mathcal{Z}$ are arbitrary subsets of \mathcal{X} and \mathcal{Z} respectively. If we assume that \mathcal{Z} contains the pseudorange measurements from the observed satellite and \mathcal{X} represents the vehicle poses we immediately see that the GNSS-based positioning problem can be formulated and solved within the general factor graph framework.

Factor graphs are bipartite by definition, i.e. they contain two sets of nodes: one for the hidden variables and the other for the probabilistic relations (the factors) between them.

A. Finding the Maximum a Posteriori Solution

The maximum a posteriori (MAP) estimate of the distribution $P(\mathcal{X}|\mathcal{Z})$, i.e. the most likely variable configuration \mathcal{X}^* given the data \mathcal{Z} , is formalized as an optimization problem of the form

$$\mathcal{X}^* = \operatorname{argmax}_{\mathcal{X}} P(\mathcal{X}|\mathcal{Z}) = \operatorname{argmax}_{\mathcal{X}} \prod_i P_i(\bar{\mathcal{X}}_i|\bar{\mathcal{Z}}_i) \quad (2)$$

If the single factors P_i are Gaussian, they are of the general form

$$P_i(\bar{\mathcal{X}}_i|\bar{\mathcal{Z}}_i) = \eta \exp -\frac{1}{2} \|\mathbf{e}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{Z}}_i)\|_{\Sigma_i}^2 \quad (3)$$

where $\mathbf{e}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{Z}}_i)$ is a problem-specific error function. Using this relation and taking the negative logarithm, we can transform (2) into

$$\mathcal{X}^* = \operatorname{argmin}_{\mathcal{X}} \sum_i \|\mathbf{e}_i(\bar{\mathcal{X}}_i, \bar{\mathcal{Z}}_i)\|_{\Sigma_i}^2 \quad (4)$$

which is a least squares optimization problem, since we seek the minimum over a sum of squared terms.

Such problems can be solved using a variety of methods like Levenberg-Marquardt, Gauss-Newton or Powell's Dog-Leg. These approaches iteratively solve the problem by repeatedly linearizing it and updating the current estimate of \mathcal{X}^* until convergence. At their heart, these methods rely on a factorization (either QR or Cholesky) of the Jacobian associated with the factor graph. Specialized solvers that exploit the sparse nature of the factorization (i.e. the sparse structure of the Jacobians) can solve typical problems with

thousands of variables very efficiently. Examples for convenient C++ frameworks that contain such solvers and can be easily applied to a number of different problem domains are g^2o [10] or GTSAM [11].

B. iSAM2 - Incremental Inference in Factor Graphs

In general, factor graph problems can be either solved in *batch* mode or *incrementally*. The difference is that a batch solver uses all available measurements and solves the complete graph at once. In contrast, incremental methods are able to efficiently update the graph (i.e. incorporate new measurements) and calculate a new estimate online, after each update step. Such incremental solvers have been explored by Kaess et al. who introduced the iSAM [12] algorithm (*incremental smoothing and mapping*) and more recently iSAM2 [5].

Their key insight was that the sparse QR or Cholesky factorization that lies at the heart of batch solvers like Levenberg-Marquardt or Gauss-Newton is equivalent to converting the factor graph into a Bayes net via an *elimination* algorithm. The resulting Bayes net can be further converted into a new data structure coined the *Bayes tree* [13] by discovering the cliques in the Bayes net. This tree structure allows particularly easy and efficient incremental updates and inference. That is, new factors can be added and a new updated estimate \mathcal{X}^* is calculated where only the necessary parts of the tree are re-evaluated.

For more details we have to refer the reader to [5] which gives an elaborate description of iSAM2 and the Bayes tree.

C. Outliers in Factor Graphs

Let us now consider the case that some of the terms \mathbf{e}_i that constitute the error functions in (4) are *outliers*. In general, outliers are caused by observations or measurements that violate the assumed underlying error model (e.g. a zero-mean Gaussian noise model) of the sensor or the measurement process. In the context of GNSS-based localization, such outlier constraints are caused by the non-line-of-sight or multipath observations that frequently occur in urban areas.

It is well known that outliers in least squares optimization problems cause biased and erroneous solutions. Different techniques have been developed to detect and remove outliers before the optimization step. A commonly known example is RANSAC that can be best applied in model-fitting or regression but is unsuitable for the factor graph problems that are in the focus of our paper.

D. Switchable Constraints

In earlier work [2], [3] we developed the *switchable constraints*, a specialized approach to detect and remove outlier constraints from factor graph problems. We also successfully applied it to the domain of GNSS-based positioning [4]. This section shortly summarizes the key ideas, but we refer the reader to our previous publications for more details.

The general idea behind the switchable constraints is to make the topology of the factor graph partially variable and subject to the optimization. This way, factors can be

removed from the problem formulation as part of the inference process. This is achieved by *augmenting* the original optimization problem (4) by a new set of hidden variables. In addition to \mathcal{X}^* the augmented problem also estimates a set of so called *switch variables* S .

Each switch variable $s_i \in \mathbb{R}$ is associated with a constraint e_i that could potentially be an outlier. Depending on its value, the switch variable can downweight, i.e. suppress or completely remove its associated constraint from the factor graph via a multiplicative *switch function* $\Psi : \mathbb{R} \rightarrow [0, 1]$.

The original optimization problem in (4) in its augmented form is given as:

$$\mathcal{X}^*, S^* = \underset{\mathcal{X}}{\operatorname{argmin}} \underbrace{\sum_i \|\Psi(s_i) \cdot e_i\|_{\Sigma_i}^2}_{\text{Switchable Constraints}} + \underbrace{\sum_i \|e_i^{\text{SP}}\|_{\Xi_i}^2}_{\text{Switch Prior Constraints}} \quad (5)$$

Different switch functions Ψ can be defined, e.g. a step function, or a sigmoid. However, our earlier work showed that a simple linear function is a suitable choice and superior to the previously proposed sigmoid function [3].

To summarize, the idea behind the switchable constraints is that single factors can be removed during the inference process by driving the associated switch variable s_i to a value so that $\Psi(s_i) \approx 0$. This way, the inference algorithm can exclude outlier measurements and converge towards a correct solution.

IV. GNSS POSITIONING AS A FACTOR GRAPH PROBLEM

Estimating the receiver position from a set of satellite observations is essentially a least squares optimization problem that can be expressed in a factor graph formulation. Following the notation from above, the measurements \mathcal{Z} will be the pseudorange readings from individual satellites and the unknown variables \mathcal{X} comprise at least the receiver position and the clock bias.

This section explains in detail how we modelled the positioning problem using factor graph notation. Fig. 2 illustrates the factor graph, showing the variables in large vertices and the factors between them with small vertices. The optimization problem expressed by the graph is solved *incrementally* and *online* using the iSAM2 algorithm [5] which was shortly explained in the previous section.

A. Involved Variables

The state space of the GNSS-based positioning problem contains at least the 3D position $(x, y, z)^T$ of the vehicle and the receiver clock error δ , leading to a state space that is at least 4-dimensional.

In the work described in this paper we extended this state space by jointly estimating the vehicle velocity v , and the clock error drift $\dot{\delta}$, leading to an 6-dimensional state space vector $\mathbf{x} = (x, y, z, v, \delta, \dot{\delta})^T$.

To address a subset of the complete state space, we will use a superscript notation in the following. E.g. we will write $\mathbf{x}_t^{x,y,z}$ to address the vehicle position at time t .

B. The Switchable Pseudorange Factor

A number of satellites are observed from every vehicle state \mathbf{x}_t , each providing a pseudorange measurement ρ_{tj} . Given the receiver position $\mathbf{x}_t^{x,y,z}$ and the position of the observed satellite $\mathbf{x}_{tj}^{\text{SAT}}$, the expected pseudorange measurement is given by the measurement function

$$h(\mathbf{x}_t, j) = \|\mathbf{x}_{tj}^{\text{SAT}} - \mathbf{x}_t^{x,y,z}\| + \delta^{\text{EarthRotation}} + \delta^{\text{Atm}} + \mathbf{x}_t^\delta \quad (6)$$

The terms $\delta^{\text{EarthRotation}}$ and δ^{Atm} correct ranging effects caused by the earth's rotation and atmosphere (ionospheric and tropospheric propagation errors).

If we assume the measured pseudorange ρ_{tj} is given by the measurement function $h(\mathbf{x}_t, j)$ plus a zero-mean Gaussian error term, then the (switchable) error function of a single pseudorange factor is given as

$$\|e_{tj}^{\text{SPR}}\|_{\Sigma_{tj}}^2 = \|\Psi(s_{tj}) \cdot (h(\mathbf{x}_t, j) - \rho_{tj})\|_{\Sigma_{tj}}^2 \quad (7)$$

Since every pseudorange measurement is a potential outlier (it could be affected by NLOS or multipath errors), we make each pseudorange factor switchable as described in section III-D. Notice in Fig. 2 that an additional *switch prior factor* f^{SP} is used as a soft constraint that tries to anchor each switch variable in its original ‘‘on’’ position at a value of $\gamma_{tj} = 1$.

C. The Motion Model Factor

A variety of motion models can be applied in the context of vehicle localization or motion estimation. For instance, [14] lists and evaluates six different types. Although a *constant velocity and turn rate* model (CTRV) seems to be a suitable choice to describe the behaviour of a moving vehicle, close inspection and practical experience reveals that a *constant velocity* (CV) model is more suitable in our scenario. The main reason for this is that no odometry information (i.e. velocity and yaw rate) from the vehicle are available to support the estimation of heading, velocity and yaw rate. The only sensor information used in our work are the pseudorange readings from the observed satellites.

Although even under these conditions, a CTRV model is still able to estimate all of velocity, heading and turn rate well while the vehicle is moving, the model breaks when the vehicle stops, e.g. at traffic lights. In this case, estimating the turn rate and heading from the pseudoranges alone is extremely error-prone and a random walk, i.e. a constant position (CP) model, describes the estimated behaviour of the vehicle better than CTRV or CV. We expect the best method to be using a switching model or mixture approach. In the context of factor graph inference, [15] proposed a max-mixture approach that could be of use here.

To summarize these considerations, we apply a constant velocity (CV) model, that is a good compromise between the CTRV and CP models and works well while the vehicle is moving and stopped. The error function for the constant velocity motion model factor is thus given as:

$$\|e_t^{\text{CV}}\|_{\Sigma_t^{\text{CV}}}^2 = \left\| \left(\sqrt{\|\mathbf{x}_t^{x,y} - \mathbf{x}_{t+1}^{x,y}\|^2} \right) - \begin{pmatrix} \mathbf{x}_t^v \Delta t \\ \mathbf{x}_t^v \end{pmatrix} \right\|_{\Sigma_t^{\text{CV}}}^2 \quad (8)$$

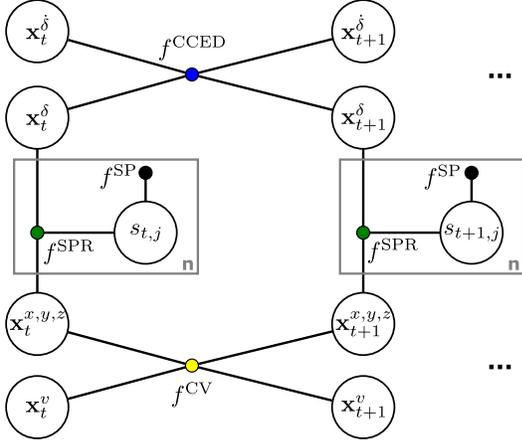


Fig. 2. The factor graph layout used in this work. Large vertices represent unknown variables, while small vertices illustrate the different factors for the switchable pseudorange (f^{SPR}), switch prior (f^{SP}), constant clock error drift (f^{CCED}), and constant velocity (f^{CV}) models. Notice that the plate representation (the rectangle around f^{SPR} and $s_{t,j}$) implies that all factors and variables inside the plate are present n times. These correspond to the n pseudorange measurements taken at time t .

D. The Clock Error Drift Factor

To account for the drift in the receiver clock error, we apply a constant drift model. Therefore, the error function associated with the *constant clock error drift* factor is

$$\|e_t^{\text{CCED}}\|_{\Sigma_t^{\text{CCED}}}^2 = \left\| \begin{pmatrix} \mathbf{x}_t^\delta + \mathbf{x}_t^\delta \Delta t \\ \mathbf{x}_t^\delta \end{pmatrix} - \begin{pmatrix} \mathbf{x}_{t+1}^\delta \\ \mathbf{x}_{t+1}^\delta \end{pmatrix} \right\|_{\Sigma_t^{\text{CCED}}}^2 \quad (9)$$

$\Sigma_t^{\text{CCED}} = \text{diag}(\sigma_t^{\text{Clock}}, \sigma_t^{\text{ClockDrift}})$ is the covariance matrix associated with the state transition factor at time t .

V. USING iSAM2 WITH SWITCHABLE CONSTRAINTS FOR ONLINE PROCESSING

This section describes two novel adaptations of the original iSAM2 algorithm we conducted in order to perform incremental inference in the context of GNSS-based positioning. These adaptations comprise the pruning of the Bayes tree and a strategy to perform additional or *auxiliary* updates of the factor graph and the Bayes tree to accelerate convergence. Notice that both ideas can be applied whenever iSAM2 is used to perform sensor fusion with factor graphs. We used the iSAM2 implementation available as part of GTSAM [11] as the basis for our own code.

A. Auxiliary Factor Graph Updates

In its original description in [5] and its implementation in GTSAM [11], the iSAM2 algorithm performs only a *single* update on the estimate \mathcal{X} whenever new factors are added. That is, \mathcal{X} is *not* repeatedly updated until convergence, but rather only a single update step $\mathcal{X} + \Delta$ is performed. While this behaviour is suitable for the context of SLAM in robotics iSAM2 was developed for, it is clearly suboptimal for sensor fusion applications which are in the focus of this paper.

The reasons for that suboptimality become clear if we understand that in the context of sensor fusion, subsequent

algorithms and applications depend on the quality of the estimate of the *most recent* variable \mathbf{x}_t . Therefore, we require the estimate of \mathbf{x}_t to have converged before we move on and add measurements taken at the next time step $t + 1$. With the original iSAM2 algorithm, \mathbf{x}_t usually converges only after several update steps, i.e. at time $t + n$. How many additional time steps n are necessary until convergence of \mathbf{x}_t occurs, depends on the general convergence properties of the problem, nonlinearities, initial guesses and so on.

To achieve immediate convergence of \mathbf{x}_t , we perform a number of auxiliary updates after \mathbf{x}_t was added. Each auxiliary update adds an auxiliary variable to the factor graph and performs the normal iSAM2 update step. We abort this process when the current estimate of $\mathbf{x}_t^{x,y,z}$ does not differ by more than 0.5 m compared to the estimate of the previous auxiliary update or when a maximum number (set to 15) of auxiliary updates has been performed. The added auxiliary variables are removed during the next update cycle.

B. Pruning of the Factor Graph and Bayes tree

In order to keep the depth of the Bayes tree and thus the required update time bounded, we prune old variable vertices and their associated factors. Not doing so would result in a Bayes tree growing linearly over time as new vertices are added with every measurement. Although iSAM2 would not necessarily re-evaluate old vertices in the tree during inference, the required update time still scales linearly with the variables in the tree.

Vehicle position vertices ($\mathbf{x}_t^{x,y,z}$) in the factor graph and Bayes tree are pruned depending on their age and on their spacial distance to the most current variable. We remove all vehicle variables that are older than 35 measurement cycles or further away than 25 meters. Other vertices, i.e. those involving the variables $s_{t,j}$, \mathbf{x}_t^δ , \mathbf{x}_t^δ and \mathbf{x}_t^v are pruned when the vehicle variable they are connected to is deleted.

Notice that both parameters mainly influence the runtime of the algorithm, but not so much the quality of the results. They were set empirically without exhaustively searching for the best values to avoid overfitting.

VI. EVALUATION

The previous sections described how the GNSS-based positioning problem can be formulated using the factor graph framework and how switchable constraints are in general able to identify outliers during inference in such graphs. We furthermore described iSAM2, an algorithm for *incremental* inference and two novel adaptations to it which enable the application of factor graphs in online sensor fusion problems.

This section will present results from an evaluation conducted on a real-world dataset.

A. The Chemnitz City Dataset

For the evaluation of our proposed approach to robust NLOS and multipath mitigation, we used the Chemnitz City Dataset which already served as a benchmark in our earlier work [4]. The dataset consists of pseudorange measurements collected by a consumer-class GPS device (u-blox LEA4-T)

TABLE I
PARAMETERS USED IN THE EVALUATION.

Parameter	Value	Description
γ_{t_j}	1.0	switch prior value
Ξ_{t_j}	1.0	switch prior covariance
Σ_{t_j}	$(10 \text{ m})^2$	cov. of pseudorange measurements
Σ_t^{CCED}	$\text{diag}(0.001 \text{ s}, 0.25 \frac{\text{s}}{\text{s}})^2$	clock error drift factor covariance
Σ_t^{CV}	$\text{diag}(0.01 \text{ m}, 8 \frac{\text{m}}{\text{s}})^2$	CV motion model covariance

that was mounted in a vehicle. Data was recorded during a 36 minutes long drive while the vehicle crossed a large road junction in the city center many times. Due to the many tall buildings nearby, massive multipath and NLOS effects occur over most parts of the trajectory. Notice that no velocity or yaw rate information from the vehicle were used to support the positioning. However, as usually done, all satellites with an elevation of $< 15^\circ$ above the horizon have been removed. Except for this, no other preprocessing has been conducted.

Highly accurate ground truth data was provided by a centimeter-precision NovAtel SPAN GPS/INS with RTK support running at 50 Hz update rate.

B. Used Parameters

Table I lists the different parameters that were used in the implementation. The values in the upper part of the table correspond to the same parameters of the robust back-end we encountered in the SLAM (Simultaneous Localization and Mapping) context where the concept of switchable constraints was originally developed. They were chosen to have the same values as in our earlier work [2], which underlines that the proposed approach is generic and domain-independent. The values for the parameters in the lower part of the Table I are problem specific and were chosen in correspondence to the values used in our earlier work [4] describing the batch solution.

C. Results

1) *Trajectory Errors:* The trajectory estimated by our proposed system of iSAM2 with switchable constraints is plotted in Fig. 3 along with the ground truth trajectory and the results of the non-robust naive least squares solution. The latter is massively biased from NLOS and multipath effects while our robust approach is able to maintain a good estimate of the vehicle's position over the whole course of the experiment.

The trajectory errors, i.e. the distance between the estimated vehicle position and ground truth position, are compared in Table II for the different approaches. Notice that the blue and red plots in Fig. 3 correspond to the first and third row in Table II that respectively use the same color.

In the top row (blue) the median, mean, and maximum errors are given for the naive least squares approach that uses merely the pseudoranges to estimate the vehicle position. Due to the various NLOS and multipath effects in the chosen scenario, the errors are extremely high. The second row repeats the results gained by our previous work [4] that

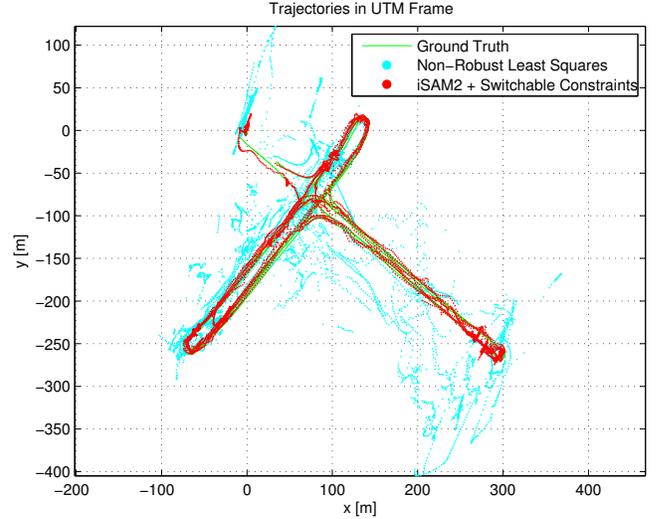


Fig. 3. The ground truth trajectory (green) overlaid with the estimated trajectory of our proposed robust approach using iSAM2 with switchable constraints (red). The blue trajectory is the solution of the non-robust naive least squares result that is heavily biased due to multipath and NLOS effects. The benefit of our method compared to the non-robust solution is clearly visible.

TABLE II
TRAJECTORY ERRORS AND CONVERGENCE TIME FOR DIFFERENT METHODS ON THE CHEMNITZ CITY DATASET.

Method	Median [m]	Mean [m]	Max [m]	Time [s]
Pseudorange only	21.79	33.37	171.64	1.2
Robust batch [4]	2.45	2.96	16.31	66.9
Proposed approach	2.55	3.21	21.04	845
Control experiment 1	22.02	33.39	172.32	516
Control experiment 2	2.88	12.51	310.25	665

used a batch optimizer based on g^2_o [10] extended by the proposed switchable constraints. The third row (in red) finally lists the results obtained with the incremental online approach proposed in this paper.

It is apparent that the incremental solution proposed here is only slightly less accurate than the robust batch solution we reported in earlier work [4]. This is a remarkable result, since it proves that the robustness against outliers gained by the switchable constraints is not dependent on performing a batch optimization. It can rather be applied in an incremental fashion as well.

The last two rows in Table II show the results obtained by two control experiments. The first one used the same factor graph layout (i.e. pseudorange, motion model and clock bias drift factors) except for the switchable constraints. This experiment was conducted to show that the motion model or clock bias drift factors alone are not able to mitigate the various multipath and NLOS effects and that it is indeed the switchable constraints that are responsible for the good results from the third row. The second control experiment corresponds to the factor graph with the switchable constraints, but without the motion model. Here we see that

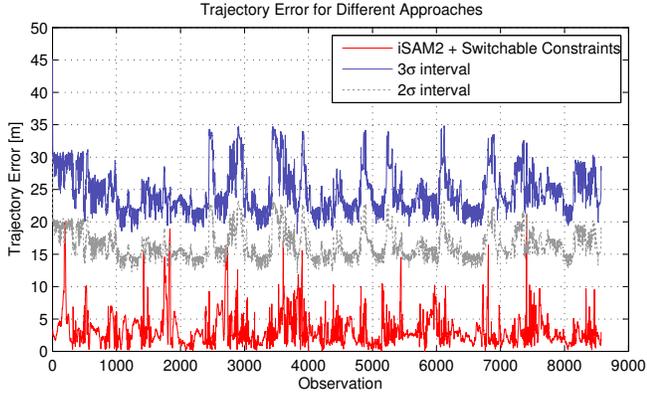


Fig. 4. Trajectory errors along with the estimated 2σ and 3σ bounds. 99.56% of all position estimates are within the 2σ boundary.

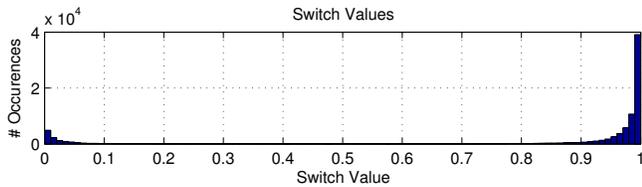


Fig. 5. Histogram over the values of the switch variables s_{tj} . Notice that most switch variables are either approximately 1 or 0, but almost no intermediate values exist. This means that the optimizer very clearly “decided” whether a satellite observation should be regarded as an outlier or an inlier.

while the median error is almost equal to the proposed method (red), without the motion model there are a number of bad estimates that dramatically increase the mean and maximum errors. Thus we conclude that the motion model has a stabilizing effect on the estimation problem and helps identifying the satellite observations influenced by NLOS and multipath effects.

2) *Runtime*: All experiments were conducted on the same Intel Core2-Duo machine in order to compare the required runtime. From Table II we see that the proposed incremental approach takes much longer than both the naive and the robust batch solution. However, we see that the achieved frame rate is still high enough to be feasible for real-world application since it is two times faster than real-time.

3) *Position Estimate Integrity*: Fig. 4 shows the position errors over time, along with the estimated 2σ and 3σ bounds. It is apparent that the error bounds are estimated rather conservatively, since 95.8% of all position estimates are within the 1σ boundary, 99.7% are within the 2σ boundary, and 100% lie inside the estimated 3σ interval.

4) *Switch Variables*: A histogram over the estimated values of the switch variables s_{tj} is shown in Fig. 5. From the clear bimodal nature of the histogram we can conclude that the proposed system is able to clearly distinguish outliers (with a value of ≈ 0) from inliers (value ≈ 1). Using a threshold of 0.5, 15.4% of all satellite observations have been declared to be an outlier. We already observed the

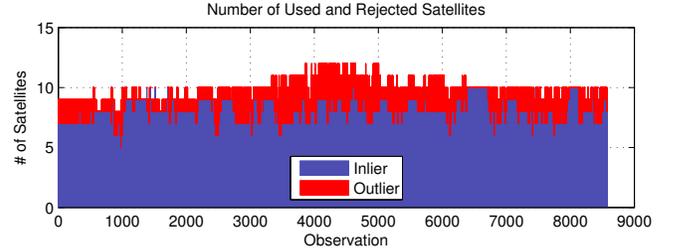


Fig. 6. Comparison of the satellite observations rejected as outliers (i.e. influenced by NLOS or multipath effects) vs. the number of inliers, i.e. valid observations. It is apparent that at almost every position along the trajectory at least one observation has been rejected by the switchable constraints.

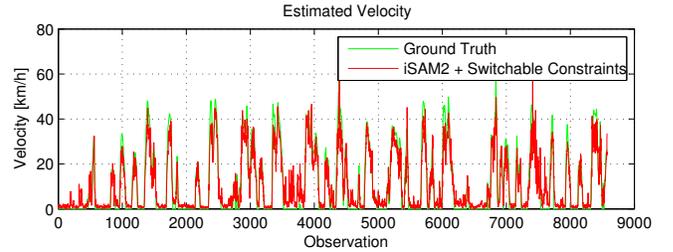


Fig. 7. Estimated (red) and ground truth velocity (green). Although the estimate fits the ground truth well in general, the peak velocity is systematically slightly underestimated. The reasons for this behaviour remain to be explored in future work.

clear bimodal behaviour in our previous work using a batch solver [4]. As we see from Fig. 5 the incremental approach presented here was able to reproduce these beneficial results.

Fig. 6 compares the number of inlier and outlier satellite observations for each vehicle position.

5) *Velocity Estimate*: The vehicle velocity is estimated due to the constant velocity motion model factor that is part of the factor graph layout. These velocity estimates are compared to the ground truth in Fig. 7. As one can see, the estimates follow the ground truth closely. However, the peak velocity is constantly underestimated slightly. The reasons for this behaviour are yet unknown and the issue has to be clarified in future work.

6) *Bayes Tree Pruning*: The effects of the pruning strategy can be seen in Fig. 8. The size of the tree was bounded to less than 700 nodes throughout the experiment. The time required for a single update fluctuates below 0.2 seconds, except for a few spikes. Notice that the plotted update time includes the time required for the auxiliary updates.

Close inspection and comparison with the velocity plot in Fig. 7 reveals that the tree is smallest when the vehicle moves fast. In this case the prune-by-distance strategy described in section V-B removes vehicle variables that are estimated to be further than 25 meters away from the current vehicle position. Associated factors and variables are removed as well. In other cases, i.e. when the vehicle moves slowly or stops at traffic lights, the tree size is bounded to contain only factors and variables from the 35 most recent measurement cycles and therefore grows quickly.

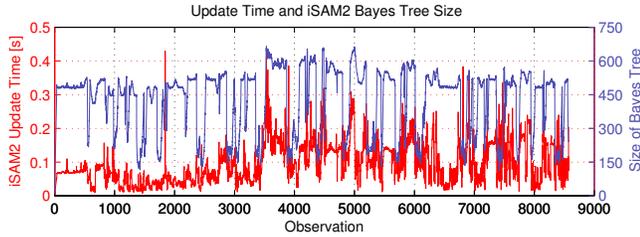


Fig. 8. The blue curve illustrates how the size of the Bayes tree fluctuates due to the pruning scheme described in section V-B. The required update time (including the auxiliary updates) is shown in red.

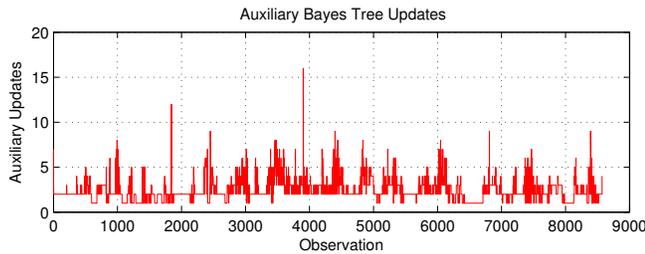


Fig. 9. Number of performed auxiliary updates of the Bayes tree. See section V-A for an explanation.

7) *Auxiliary Update Steps*: Fig. 9 illustrates the number of auxiliary update steps performed over time. As explained in section V-A, the number of necessary auxiliary updates is determined adaptively by our algorithm in order to foster the convergence of the current variable’s estimate. While the number of necessary auxiliary steps is rather low for most of the dataset, a few spikes in the plot mark especially difficult situations where the solver had to iterate more often in order to converge.

VII. CONCLUSIONS AND FUTURE WORK

We presented a novel approach for *online* robust GNSS-based positioning in urban areas and demonstrated that reliable positioning can be reached despite the presence of massive non-line-of-sight and multipath effects.

We demonstrated how GNSS-based positioning can be formulated as a factor graph problem and solved efficiently *online*, i.e. in real-time, by applying an incremental inference algorithm (iSAM2 [5]). We furthermore demonstrated that by using *switchable constraints*, it is possible to detect and remove outlier pseudorange observations that are caused by multipath and non-line-of-sight effects. In addition, we developed and applied two ideas for factor graph pruning and auxiliary updates that foster fast online convergence and keep the graph size and therefore the computation time bounded.

In this paper, we successfully advanced our earlier work [4] from pure *batch* processing to *online* applicability. The approach presented here, i.e. combining switchable constraints with iSAM2, auxiliary updates and graph pruning, can therefore now be applied to real-world applications that require online precise positioning information in urban

areas, such as advanced driver assistance systems, intelligent transportation systems or location based services.

One of the advantages of the factor graph formulation is its easy expandability. It is worthwhile to include additional sensor information such as velocity and yaw rate measurements from the vehicle’s internal sensors. Such additional sensor information can simply be incorporated by adding new variables and prior factors to the factor graph used here.

Another very promising direction of future work is to overcome the limitations of Gaussian noise models by enabling the factor graph framework to perform inference over non-Gaussian distributions. Recent work [16] demonstrated first steps into this direction and showed that e.g. the heavy-tailed Cauchy distribution can be easily integrated in iSAM2. We will explore these possibilities in future work. Although an elaborate analysis has yet to be conducted, we expect that GNSS-based positioning in urban areas can benefit from such more realistic error models than the ubiquitous Gaussian.

REFERENCES

- [1] R. Søråsen and O. Lykkja, “GNSS-Based Tolling – Possibilities, Challenges and Opportunities,” in *19th ITS World Congress*, 2012.
- [2] N. Sünderhauf and P. Protzel, “Switchable Constraints for Robust Pose Graph SLAM,” in *Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, 2012.
- [3] N. Sünderhauf, “Robust Optimization for Simultaneous Localization and Mapping,” Ph.D. dissertation, Chemnitz University of Technology, 2012. [Online]. Available: <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-86443>
- [4] N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel, “Multipath Mitigation in GNSS-Based Localization using Robust Optimization,” in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, 2012.
- [5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree,” *Intl. Journal of Robotics Research*, 2012.
- [6] Z. Qiang, Z. Xiaolin, and C. Xiaoming, “Research on RAIM Algorithm under the Assumption of Simultaneous Multiple Satellites Failure,” in *Proc. of ACIS Intl. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 2007.
- [7] J.-i. Meguro, T. Murata, J.-i. Takiguchi, Y. Amano, and T. Hashizume, “GPS Multipath Mitigation for Urban Area Using Omnidirectional Infrared Camera,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 22–30, march 2009.
- [8] M. Obst, S. Bauer, and G. Wanielik, “Urban Multipath Detection and Mitigation with Dynamic 3DMaps for Reliable Land Vehicle Localization,” in *IEEE/ION PLANS*, 2012.
- [9] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A General Framework for Graph Optimization,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [11] “GTSAM – The Georgia Tech Smoothing and Mapping Library,” <https://collab.cc.gatech.edu/borg/gtsam/>.
- [12] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental Smoothing and Mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, 2008.
- [13] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, “The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping,” in *Proc. of Intl. Workshop on the Algorithmic Foundations of Robotics*, 2010.
- [14] R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt, and G. Wanielik, “Empirical evaluation of vehicular models for ego motion estimation,” in *Intelligent Vehicles Symposium (IV)*, 2011.
- [15] E. Olson and P. Agarwal, “Inference on Networks of Mixtures for Robust Robot Mapping,” in *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.
- [16] D. Rosen, M. Kaess, and J. Leonard, “Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2013.