

Expectation-Maximization for Adaptive Mixture Models in Graph Optimization

Tim Pfeifer and Peter Protzel

Dept. of Electrical Engineering and Information Technology

TU Chemnitz, Germany

Email: {firstname.lastname}@etit.tu-chemnitz.de

Abstract—Non-Gaussian and multimodal distributions are an important part of many recent robust sensor fusion algorithms. In difference to robust cost functions, they are probabilistically founded and have good convergence properties. Since their robustness depends on a close approximation of the real error distribution, their parametrization is crucial.

We propose a novel approach that allows to adapt a multimodal Gaussian mixture model to the error distribution of a sensor fusion problem. By combining expectation-maximization and non-linear least squares optimization, we are able to provide a computationally efficient solution with well-behaved convergence properties.

We demonstrate the performance of these algorithms on several real-world GNSS and indoor localization datasets. The proposed adaptive mixture algorithm outperforms state-of-the-art approaches with static parametrization. Source code and datasets are available under <https://mytuc.org/libRSF>.

I. INTRODUCTION

Robotic systems as well as autonomous vehicles require a reliable estimation of their current state and location. The algorithms that compute this information from sensor data are typically formulated as least squares optimization and solved with frameworks like Ceres [1] or GTSAM [2]. Usually, they rely on the assumption of Gaussian distributed measurement errors, which is correct for many sensors and comes with advantageous mathematical consequences for the estimation process. However, there is a broad range of sensors, like global navigation satellite systems (GNSS), wireless range measurements, ultrasonic range finders or vision-based systems, that violate this assumption. Even simple wheel odometry can slip on difficult grounds and cause non-Gaussian error distributions. These violations can heavily distort the estimation process and lead to false estimates. A variety of robust approaches exist to provide robustness against non-Gaussian errors, but many of them require an exact parametrization and therefore knowledge about the expected error distribution. Previous evaluations [3, 4] have shown that the optimal set of parameters is hard to find and small deviations can lead to fatal errors.

With this paper, we want to introduce a novel approach to estimate the sensors' error distribution during the least squares optimization process. As improvement of the ideas in

The project is funded by the "Bundesministerium für Wirtschaft und Energie" (German Federal Ministry for Economic Affairs and Energy).

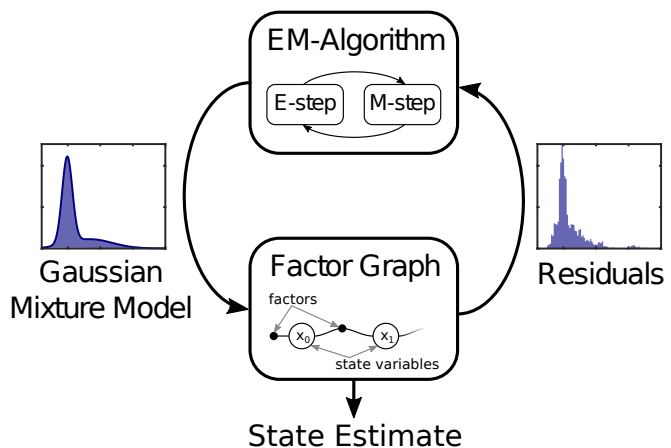


Fig. 1. The main concept of our proposed adaptive mixtures algorithm. Residuals of the optimization problem (represented as factor graph) are used to estimate a multimodal error model. Expectation-Maximization (EM) is applied for a well-behaved convergence.

[5], we show how to construct an expectation-maximization (EM) based algorithm that adapts a Gaussian mixture model (GMM) to the sensor properties. Fig. 1 shows how the least squares optimization and the error model estimation are connected. Due to its graphical representation, the least squares problem is also referred to as factor graph. In difference to previous work, we are able to overcome the limitations of the Max-Mixture approximation and implement an exact GMM inside the least squares problem. We also increase the robustness of the GMM's estimation process without extensive parametrization. This results in a sensor fusion algorithm that is robust against non-Gaussian errors without prior knowledge. Through extensive evaluation with real world localization datasets from [6, 7, 8], we are able to demonstrate its performance in comparison to several state-of-the-art approaches. Our datasets represent different scenarios from GNSS localization in urban canyons to centimeter-level wireless ranging.

II. PRIOR WORK

Motivated by the simultaneous localization and mapping (SLAM) problem, several algorithms exist that try to achieve robustness against non-Gaussian outliers. We want to give a brief overview of them and show why existing solutions are often difficult to apply to real world problems.

Several methods use generic mechanisms to reduce the influence of outliers without knowledge about the true sensor distribution. A typical example are M-estimators, that apply non-convex cost functions to the residual term of the optimization problem. They are suited for many problems, but the right parametrization is hard to find and global convergence is not guaranteed. A different approach is Switchable Constraints (SC) [9] that introduced a set of additional weights, each assigned to one measurement. The weights are optimized to weight possible outliers down, which works on SLAM [10] and GNSS localization problems [6]. However, the introduced tuning parameter is not directly connected to probabilistic metrics and has to be set manually. This can be very difficult as shown in [3] and [4]. Dynamic Covariance Scaling (DCS) [11] was introduced as improvement of SC and is basically an M-estimator. In difference to SC, the weights are not longer part of the incremental optimization process. Instead, they get optimized analytical, which leads to an overall faster convergence. However, they keep the disadvantage of difficult parametrization [3, 8]. Although there is a usable parameter window for many SLAM applications, this does not necessarily apply to general sensor fusion. To overcome the parametrization of SC/DCS for non-SLAM applications, we introduced Dynamic Covariance Estimation (DCE) in [8]. Due to the non-convex optimization surface, this approach is still limited to problems with a good initialization and moderate outliers. Agamennoni et al. proposed an approach to tune the parameters of some M-estimators in [12]. It is limited to a subset of M-estimators that can be described as elliptical distributions and cannot be applied to more robust ones like DCS or the Tukey M-estimator.

Another approach to handle non-Gaussian measurements, is to consider their true distribution during the estimation process. In difference to previously mentioned methods, this allows to handle asymmetric or multimodal distributions probabilistically correct. Max-Mixture (MM) [13] describes the expected distribution with an approximation of a GMM. Rosen et al. introduced a method that allows arbitrary non-Gaussian distributions [14]. A drawback of both approaches is the lack of concepts to get the required parametrization of the GMM or any other non-Gaussian distribution. Existing approaches [15, 16] estimate it in advance which is not possible if the error distribution depends on the environment and varies over time. This paper is build on top of both ideas from Olson and Agarwal [13] / Rosen et al. [14], so we provide more details about their concepts in Sec. III. A recently published algorithm [17] allows interference of non-parametric distributions based on kernel densities. This offers new possibilities for non-Gaussian distributions but cannot solve the parametrization problem.

Our recently published approach of self-tuning GMMs [5] aims to overcome the burden of parametrization by introducing a self-optimizing version of Olson's Max-Mixture. By enabling the optimizer to change the model's parameters, the GMM can be adapted to the most likely error distribution. Although the convergence on a real world GNSS dataset is shown, the direct optimization of the error model comes with

several drawbacks:

- 1) The algorithm requires a good initial guess of the true distribution since it can not recover from a local minimum.
- 2) Tight bounds for the optimized GMM parameters are required to work around numerical issues.
- 3) The applied Max-Mixture model is just an approximation of a GMM which is not correct for GMMs with strongly overlapping components.
- 4) The used model is fixed in the number of mixture components and there is no possibility to detect if more or fewer components are required.

Therefore, [5] is just a proof-of-concept and real applications require further steps. With this paper we relax limitation (1) and overcome (2) and (3) completely. We want to introduce a sensor fusion algorithm that adapts its error model during runtime without more than minimal knowledge about the true distribution. As far as we know, this is the first algorithm which implements a robust self-optimizing GMM as error model of a realtime capable state estimation algorithm.

III. GAUSSIAN MIXTURE MODELS

For the representation of multimodal distributions, Gaussian mixture models are a state-of-the-art approach with several advantages. With a weighted sum of multiple Gaussians, corresponding to (1), non-Gaussian properties like asymmetry or multiple modes can be represented easily. Due to the variable number of components, distributions with different complexity can be described. With expectation-maximization exists a powerful algorithm to estimate the model's parameters from distributed data. We want to show in this section how GMMs can be used for least squares. Although this is nothing new, it is important for the further construction of the proposed algorithm.

$$\mathbf{P} \sim \sum_j w_j \cdot \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \quad (1)$$

A. Multivariate Gaussians and Least Squares

The least squares problem, which is solved for state estimation, arises from the formulation of a maximum-likelihood problem (2) where \mathbf{Z} is a set of measurements \mathbf{z}_i and \mathbf{X} is the set of estimated states \mathbf{x}_i .

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmax}} \mathbf{P}(\mathbf{X}|\mathbf{Z}) \quad (2)$$

To estimate the optimal set of states \mathbf{X}^* , the problem can be described as product of conditional probabilities:

$$\mathbf{P}(\mathbf{X}|\mathbf{Z}) \propto \prod_i \mathbf{P}(\mathbf{z}_i|\mathbf{x}_i) \quad (3)$$

By applying the negative logarithm, the optimization problem can be rewritten as sum:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i -\ln(\mathbf{P}(\mathbf{z}_i|\mathbf{x}_i)) \quad (4)$$

The estimated $\hat{\mathbf{X}}$ is the maximum-likelihood estimator of \mathbf{X} .

For multivariate Gaussians, the corresponding least squares problem is defined as:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \frac{1}{2} \left\| \mathcal{I}^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}) \right\|^2 \quad (5)$$

The estimation error $\mathbf{e}(\mathbf{x}_i, \mathbf{z}_i)$ is defined as non-linear function of a measurement and a corresponding subset of states. Instead of the covariance Matrix $\boldsymbol{\Sigma}$, we use the square root information matrix $\mathcal{I}^{\frac{1}{2}}$ that is defined by $\mathcal{I}^{\frac{1}{2}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}$. It can be computed from $\boldsymbol{\Sigma}^{-1}$ using Cholesky decomposition. For a sum of Gaussians with n components, the conditional probability is defined as:

$$\mathbf{P}(\mathbf{z}_i | \mathbf{x}_i) \propto \sum_{j=1}^n c_j \cdot \exp \left(-\frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \right) \quad (6)$$

with $c_j = w_j \cdot \det \left(\mathcal{I}_j^{\frac{1}{2}} \right)$

Due to the summation, the logarithm cannot be pushed inside and the log-likelihood has to be calculated differently. Olson and Agarwal [13] respectively Rosen et al. [14] provided two possible solutions.

B. Approximated Solution (Max-Mixture)

In [13] the summation of a GMM was replaced by the maximum operator, which leads to:

$$-\ln(\mathbf{P}) = -\ln \max_j \left(c_j \cdot \exp \left(-\frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \right) \right) \quad (7)$$

This approximation is valid as long as the Gaussian components are well separated. The maximum becomes a minimum when the logarithm is pushed inside:

$$-\ln(\mathbf{P}) = \min_j \left(-\ln c_j + \frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \right) \quad (8)$$

In difference to other implementations, we keep the log-normalization term $-\ln c_j$ to preserve a consistent optimization surface. Hence, we introduce a normalization constant γ_m to construct a well-behaved least squares problem:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \frac{1}{2} \min_j \left\| \sqrt{-2 \cdot \ln \frac{c_j}{\gamma_m}} \cdot \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \quad (9)$$

with $\gamma_m = \max_j c_j$

The additional term $-\ln \frac{c_j}{\gamma_m}$ is treated as separate dimension of the vectorized error function. A positive expression under the square root is guaranteed by γ_m . The separation of log-normalization and quadratic error leads to better convergence, since the partial derivative regarding the state value \mathbf{x}_i is identical to the original problem (5).

C. Exact Solution (Sum-Mixture)

The exact implementation of a GMM inside a least squares problem is possible with the approach proposed in [14]. Rosen

et al. demonstrated that arbitrary distributions can be applied to weight the estimation error with:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \left\| \sqrt{-\ln \left(\frac{\mathbf{P}(\mathbf{z}_i | \mathbf{x}_i)}{\gamma_s} \right)} \right\|^2 \quad (10)$$

with $\gamma_s \geq \max_i (\mathbf{P}(\mathbf{z}_i | \mathbf{x}_i))$

It requires a normalization constant γ_s to keep the negative log-likelihood positive. Based on (6) the new normalized probability of a GMM is defined as:

$$\frac{\mathbf{P}(\mathbf{z}_i | \mathbf{x}_i)}{\gamma_s} = \sum_j \frac{c_j}{\gamma_s} \cdot \exp \left(-\frac{1}{2} \left\| \mathcal{I}_j^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}_j) \right\|^2 \right) \quad (11)$$

Since it has to satisfy $\mathbf{P}(\mathbf{z}_i | \mathbf{x}_i) / \gamma_s \leq 1$, the normalization can be defined as summation of all constant terms $\gamma_s = \sum_j c_j$. The resulting least squares problem is:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \left\| \sqrt{-\ln \left(\sum_j \frac{c_j}{\gamma_s} \cdot \exp(\dots) \right)} \right\|^2 \quad (12)$$

IV. EXPECTATION-MAXIMIZATION FOR MIXTURE DISTRIBUTIONS

The EM-algorithm [18] is a maximum likelihood approach in which a set of parameters $\boldsymbol{\theta}$ is estimated from a set of observed variables \mathbf{O} , but also depends on a set of hidden ones \mathbf{H} . Since $\boldsymbol{\theta}$ and \mathbf{H} are unknown at the beginning, $\boldsymbol{\theta}$ cannot be estimated directly. Instead, an alternating sequence of E-steps and M-steps is performed. E-steps estimate the hidden variables based on an initial guess of $\boldsymbol{\theta}$ and M-steps compute $\boldsymbol{\theta}$ based on the previously estimated \mathbf{H} . Both steps are performed iteratively until a maximum number of iterations or a different convergence criterion is reached. For GMM estimation it is defined as:

$$\begin{aligned} \mathbf{O}^{GMM} &= \{\mathbf{e}_i\} & \mathbf{H}^{GMM} &= \{\alpha_{ij}\} \\ \boldsymbol{\theta}^{GMM} &= \{w_j, \boldsymbol{\mu}_j, \mathcal{I}_j^{\frac{1}{2}}\} \end{aligned} \quad (13)$$

The observed variable is the measurement error $\mathbf{e}_i = \mathbf{z}_i - f(\mathbf{x}_i)$, defined by the measured value \mathbf{z}_i and the true value of the state \mathbf{x}_i . Hidden variable is the probability α_{ij} of each measurement \mathbf{z}_i to belong to component j of the GMM. The parameters $w_j, \boldsymbol{\mu}_j$ and $\mathcal{I}_j^{\frac{1}{2}}$ define the mixture distribution. For mathematical details please see Appendix A.

V. SELF-TUNING MIXTURES

To solve the problem of simultaneous state and error model estimation, we define them together as a nested EM-algorithm. At first, we assume that we are able to estimate a state that is a coarse approximation of the true state \mathbf{X} . This can be done with a simple initialization using accumulated odometry measurements or non-robust least squares optimization. It implies that the estimated measurement error $\hat{\mathbf{e}}_i$ is also close to its true value:

$$\hat{\mathbf{x}}_i \approx \mathbf{x}_i \Rightarrow \hat{\mathbf{e}}_i \approx \mathbf{e}_i \quad (14)$$

Based on this assumption, we can redefine the observed variable for the GMM estimation problem as (15) and the EM problem of state estimation as (16).

$$\mathbf{O}^{GMM} = \{\hat{\mathbf{e}}_i = \mathbf{e}(\hat{\mathbf{x}}_i, \mathbf{z}_i)\} \quad (15)$$

$$\mathbf{O}^{SE} = \mathbf{Z} \quad \mathbf{H}^{SE} = \boldsymbol{\theta}^{GMM} \quad \boldsymbol{\theta}^{SE} = \mathbf{X} \quad (16)$$

Algorithm 1 summarizes how both stages of EM are connected. After the initialization of \mathbf{X} and $\boldsymbol{\theta}^{GMM}$, an outer EM-algorithm for state estimation (SE) is combined with an inner one for the Gaussian mixture model (GMM). The E-step of the SE problem estimates the hidden parameter $\boldsymbol{\theta}^{GMM}$ and is composed of the E- and the M-step of the GMM estimation problem. The outer M-step is the least squares optimization according to the Max-Mixture (9) or Sum-Mixture (12) formulation from Sec. III. To keep the algorithm real-time capable, only the measurements and states in a sliding window with the length t_{SW} are considered in the estimation problem.

Result: $\hat{\mathbf{X}}, \hat{\boldsymbol{\theta}}$

Initialize states \mathbf{X} randomly

Estimate $\mathbf{X}_{t=0}$ with (5)

Initialize $\boldsymbol{\theta}_{t=0}^{GMM}$ (e.g with (17))

foreach time step t **do**

 Add $\mathbf{z}_i, \mathbf{x}_i$ at t to $\mathbf{P}(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}^{GMM})$

 Remove $\mathbf{z}_i, \mathbf{x}_i$ older than $t - t_{SW}$ from $\mathbf{P}(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}^{GMM})$

 // E-step state estimation

begin estimate $\boldsymbol{\theta}_t^{GMM}$

 Compute all \mathbf{e}_i from \mathbf{X}, \mathbf{Z}

 Initialize $\boldsymbol{\theta}_t^{GMM} \leftarrow \boldsymbol{\theta}_{t=0}^{GMM}$

repeat

 // E-step GMM

 Compute α_{ij} from $\mathbf{e}_i, \boldsymbol{\theta}_t^{GMM}$ with (18)

 // M-step GMM

 Compute $\boldsymbol{\theta}_t^{GMM}$ from $\mathbf{e}_i, \alpha_{ij}$ with (19)

until convergence;

end

 Update $\mathbf{P}(\mathbf{X}|\mathbf{Z}, \boldsymbol{\theta}^{GMM})$ with $\boldsymbol{\theta}_t^{GMM}$

 // M-step state estimation

 Estimate \mathbf{X}_t with (9) or (12)

end

Algorithm 1: Nested EM for Self-tuning Mixtures

Since the outer EM-algorithm is computed only once per time step, its convergence have to be achieved over time. For sensor fusion applications, the difference between each time step is small, so it is called often enough to converge until the distribution parameters change. The convergence of the algorithm is hard to prove, since it depends on the overall structure of the estimation problem and the non-linearity in the error function \mathbf{e} . It has to be assumed that the optimized surface of the problem has several local minima, which is a common problem in non-linear state estimation. Nevertheless, we would like to mention a few arguments in favor of convergence:

- 1) The resulting log-likelihood of Max-Mixture and Sum-Mixture is mostly convex which supports convergence.
- 2) The inner EM-algorithm for the GMM will converge for sure. [19, p. 450]
- 3) Since the algorithm is solved iteratively, every time step starts close to the minima that the previous one found.

The crucial question is, if a good initial starting point for the state values can be found without knowing the exact distribution. As we will show in our experimental evaluation, this is possible at least for different localization problems.

VI. LOCALIZATION PROBLEMS AS FACTOR GRAPH

Since we evaluate the proposed adaptive GMM approach on different localization datasets, we want to explain what makes them challenging and how the estimation problems are composed. Our adaptive mixtures algorithm is applied to reduce the estimation error that non-line-of-sight (NLOS) measurements cause. Hence, the estimation error \mathbf{e}_i , that is described with the Gaussian mixture, is related to the one-dimensional (pseudo) range measurements.

A. UWB-Radio Localization

In [8], we introduced a dataset that includes range measurements as well as wheel odometry of a small robot. While driving through a labyrinth, the distance to fixed points is measured with wireless ultra-wide-band (UWB) modules. They are able to provide accurate range measurements, but artificial metallic obstacles were added to provoke NLOS effects. A top-view camera system provides a centimeter-level precise ground truth and allows to evaluate the accuracy of estimation algorithms. As Fig. 2 shows, the error distribution is asymmetric and right skewed with a mean of 0.12 m and outliers up to 1 m. Since the robot's motion is restricted in a plane, the system's state is simply a 2D-pose. In difference to earlier work [8], we omit the estimation of a common offset for all range measurements. The maximum-likelihood problem is composed of one range and odometry measurement for each time step. Further details can be found in [8]. Tab. I summarizes the noise properties that are used for the non-robust factors.

TABLE I
NOISE PROPERTIES OF THE UWB INDOOR DATASET

Factor	Square Root Information $\mathcal{I}^{\frac{1}{2}}$
Range	$(0.1 \text{ m})^{-1}$
Odometry	$\text{diag} \begin{pmatrix} 0.01 \text{ m s}^{-1} \\ 0.01 \text{ m s}^{-1} \\ 0.01 \text{ m s}^{-1} \end{pmatrix}^{-1}$

B. GNSS Localization

Besides the robotic example, we evaluate the proposed algorithm on several real world GNSS datasets. Along with the older Chemnitz City dataset from [6], we use the four smartLoc datasets [7] which are recorded in the major cities

TABLE II
NOISE PROPERTIES OF THE GNSS DATASETS

Factor	Square Root Information $\mathcal{I}^{\frac{1}{2}}$	
	Chemnitz City	smartLoc
Pseudorange	$(10 \text{ m})^{-1}$	*
Odometry	$\text{diag} \begin{pmatrix} 0.05 \text{ m s}^{-1} \\ 0.03 \text{ m s}^{-1} \\ 0.03 \text{ m s}^{-1} \\ 0.006 \text{ rad s}^{-1} \end{pmatrix}^{-1}$	$\text{diag} \begin{pmatrix} 0.05 \text{ m s}^{-1} \\ 0.03 \text{ m s}^{-1} \\ 0.03 \text{ m s}^{-1} \\ 0.002 \text{ rad s}^{-1} \end{pmatrix}^{-1}$
	$\text{diag} \begin{pmatrix} 0.1 \text{ m} \\ 0.009 \text{ m s}^{-1} \end{pmatrix}^{-1}$	$\text{diag} \begin{pmatrix} 0.05 \text{ m} \\ 0.01 \text{ m s}^{-1} \end{pmatrix}^{-1}$

* individually estimated by the GNSS receiver

Frankfurt and Berlin. Due to the urban environment, they contain a high proportion of NLOS measurements, as their error histogram in Fig. 2 shows. All datasets combine the wheel odometry of a driving car with a set of pseudorange measurements from a mass-market receiver. As ground truth, a NovAtel differential GNSS receiver supported by a high-end inertial measurement unit is used.

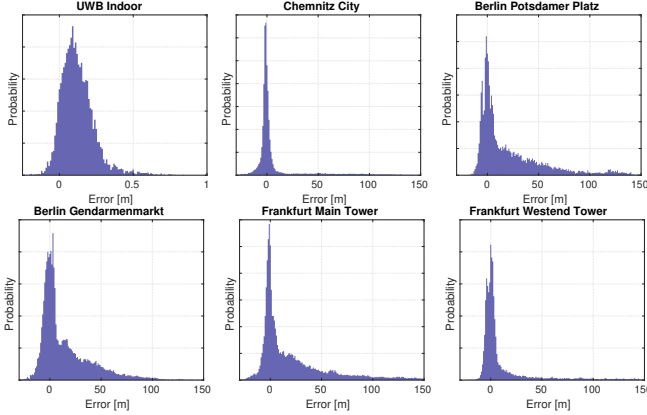


Fig. 2. Distributions of the (pseudo) range error of the UWB and GNSS datasets. Please note the differently scaled first x-axis. All data sets show a more or less distinct right skewness, which is caused by NLOS effects.

The vehicle's pose is estimated according to the Cartesian ECEF coordinate system and the rotation around its upright axis. Since the pseudorange measurements are biased by the receivers drifting clock error, it is estimated along with its derivation. In difference to the UWB dataset, there are multiple (pseudo) range measurements for each time step. Their number depends on the number of observable satellites. The dynamic of the clock error is described with a constant clock error drift model (CCED). Pseudorange, wheel odometry and CCED factor are described in detail in [5]. Tab. II summarizes the noise properties of all GNSS datasets. Since they are recorded with different sensors, the values for odometry and the CCED model differ. While the pseudorange noise of the Chemnitz City dataset is fixed, the smartLoc datasets include an individual value for each measurement that is provided by the GNSS receiver.

VII. EVALUATION

This section gives an overview over the performance of the proposed adaptive mixture approach. Beside the estimation accuracy, we want to demonstrate the good convergence properties of the adapted error model. We compare our approach against static mixture models as well as the state-of-the-art algorithm DCS [11] and cDCE [8]. We use the absolute trajectory error (ATE) in the XY-plane as error metric, since it can be applied to both estimation problems. The source code as well as the datasets are online available as part of our robust sensor fusion C++ library libRSF¹.

A. Implementation Details

The least squares optimization of the state estimation problem is implemented with the Ceres solver [1]. To keep the runtime bounded, we use a simplistic sliding window approach that removes factors older than $t_{SW} = 60 \text{ s}$. Even when we use recorded datasets, we process the data as it were in real time. Hence, the estimation problem is solved in every time step without using future measurements.

B. Parametrization of the GMM

The initial parametrization $\theta_{t=0}^{GMM}$ of the GMM can be determined in different ways. It can be estimated with a simple clustering approach like k-means or defined with prior knowledge about the estimation problem. We assume that the (pseudo) range measurements are corrupted by outliers that have a larger spread than the valid measurements. Therefore, we define the initial GMM with equal-weighted, zero-mean components that differ only in their square root information. As defined with (17) for a n-component GMM, we scale each matrix $\mathcal{I}_j^{\frac{1}{2}}$ with factor 10^{1-j} based on the square root information for the non-robust Gaussian case $\mathcal{I}^{\frac{1}{2}}$. For the smartLoc datasets, we also use a value of $\mathcal{I}^{\frac{1}{2}} = (10 \text{ m})^{-1}$ to be consistent with the Chemnitz City dataset. The algorithms with a static GMM rely on the same values.

$$\left. \begin{aligned} \mathcal{I}_j^{\frac{1}{2}} &= \mathcal{I}^{\frac{1}{2}} \cdot 10^{1-j} \\ \mu_j &= 0 \\ w_j &= \frac{1}{n} \end{aligned} \right\} \forall j = 1 \dots n \quad (17)$$

For the automatic selection of the number of GMM components, we examined classical static criteria such as BIC [20] and AIC [21]. Since the sensor data is strongly multimodal, both lead to higher numbers like $n = 5, 6$. Regarding the trajectory error, our evaluation in Fig. 3 has shown no advantages for higher numbers, we set $n = 2$ for all datasets. Although the proposed parametrization is applied in our final evaluation, we want to demonstrate that the self-tuning algorithms are robust regarding their initialization. Therefore, we evaluated the mean ATE of adaptive MM and SM for different standard deviations of the second GMM component. Fig. 4 shows that there is a wide basin of convergence for both algorithms, although Sum-Mixture converges slightly better. Both algorithms surpass their static variants significantly for any parametrization.

¹<https://mytuc.org/libRSF>

TABLE III
RESULTS OF THE FINAL EVALUATION.

Algorithm	Indoor UWB		Chemnitz		Berlin PP		Berlin GM		Frankfurt MT		Frankfurt WT	
	ATE [m]	Time [s]	ATE [m]	Time [s]	ATE [m]	Time [s]	ATE [m]	Time [s]	ATE [m]	Time [s]	ATE [m]	Time [s]
Gaussian	0.1212	26.0	30.0	56.7	29.2	9.84	13.38	47.3	30.97	54.8	23.54	32.2
DCS	0.1443	55.9	4.403	52.6	25.04	15.7	19.11	62.3	13.25	65.0	11.39	35.8
cDCE	0.1210	53.7	4.326	52.3	17.91	15.2	14.59	64.6	14.93	68.8	11.12	35.5
Static MM	0.1569	24.6	4.127	55.3	27.25	15.2	15.73	63.1	18.1	64.4	9.467	37.9
Static SM	0.1185	39.3	4.463	55.1	18.36	19.8	12.52	63.8	20.15	69.6	10.82	37.2
Adaptive MM	0.0666	41.0	2.562	88.4	26.65	28.0	16.8	124.0	15.39	111.0	6.995	67.9
Adaptive SM	0.0651	46.4	2.559	81.0	12.96	24.2	15.67	110.0	13.89	96.9	6.669	58.6



Fig. 3. Position error for the smartLoc GNSS datasets depending on the number of components of the GMM. Using more than two components has no advantage regarding the error. Please note the shared axes.

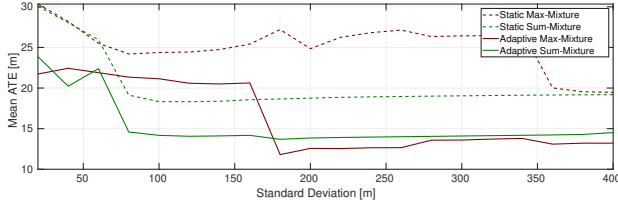


Fig. 4. Mean position error for the dataset “Berlin Potsdamer Platz” depending on the initialization of the second GMM component’s standard deviation. The exact sum-mixture approach converges for a wider range of values, but both adaptive approaches outperform their static variants.

C. Position Accuracy

Tab. III summarizes the mean ATE as well as the runtime of all algorithms. The smallest trajectory error is marked in green. For 5 out of 6 datasets, adaptive mixture models are able to reduce the impact of erroneous measurements over static ones. They are the only ones that reduce the ATE of the indoor dataset significantly. This can be explained by the shifted distribution, which cannot be compensated by other robust methods. The proposed adaptive SM algorithm achieved the best results in 4 cases. DCS achieved a slightly better result on the “Frankfurt WT” dataset, which has a relatively low number of NLOS measurements. However, it performs significantly worse on other datasets. A special case is the “Berlin GM” dataset, where only the static SM algorithm were able to reduce the ATE. We have to investigate further which characteristic caused this result. The adaptive MM algorithm achieves comparable but slightly worse results

than the SM variant. Since there is no significant difference in runtime, we would prefer the exact adaptive SM algorithm.

D. Runtime

The computation times in Tab. III are measured on an Intel i7-7700 system. As a rule of thumb, the adaptive algorithms require twice as much time as the other robust ones. Nevertheless, both proposed algorithms are significantly faster than the recording time of the datasets.

VIII. CONCLUSION

In this paper, we introduced a novel approach to combine multimodal and self-tuning error models to improve least squares optimization. Based on the EM-algorithm, we are able to efficiently adapt a Gaussian mixture model during the state estimation process. Therefore, the burden of parametrization could be relaxed for Max-Mixture [13] and Sum-Mixture [14] based algorithms. We compared our work to a set of state-of-the-art algorithms on real world datasets and showed their improved performance. Especially for the adaptive Sum-Mixture algorithm, the well-behaved convergence could be demonstrated experimentally.

Still open for future research is the question after the optimal number of Gaussian components. Although “two” is a sufficient compromise between flexibility and runtime, this does not necessarily apply to scenarios beside GNSS. Interesting could also be the transfer to SLAM problems that are defined by relative rather than absolute measurements.

APPENDIX

A. Expectation-Maximization for GMMs

The expectation step is defined by:

$$\alpha_{ij} = \frac{\mathbf{P}(\mathbf{e}_i | w_j, \boldsymbol{\mu}_j, \mathcal{I}_j^{\frac{1}{2}})}{\sum_{k=1}^n \mathbf{P}(\mathbf{e}_i | w_k, \boldsymbol{\mu}_k, \mathcal{I}_k^{\frac{1}{2}})} \quad (18)$$

with $\mathbf{P}(\mathbf{e}_i | w, \boldsymbol{\mu}, \mathcal{I}^{\frac{1}{2}}) =$

$$w \cdot \det(\mathcal{I}^{\frac{1}{2}}) \cdot \exp\left(-\frac{1}{2} \left\| \mathcal{I}^{\frac{1}{2}} (\mathbf{e}_i - \boldsymbol{\mu}) \right\|^2\right)$$

The maximization step is defined by:

$$w_j = \frac{1}{m} \sum_{i=1}^m \alpha_{ij} \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^m \alpha_{ij} \mathbf{e}_i}{\sum_{i=1}^m \alpha_{ij}} \quad (19)$$

$$\mathcal{I}_j^{\frac{1}{2} \top} \cdot \mathcal{I}_j^{\frac{1}{2}} = \left(\frac{\sum_{i=1}^m \alpha_{ij} (\mathbf{e}_i - \boldsymbol{\mu}_j) (\mathbf{e}_i - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^m \alpha_{ij}} \right)^{-1}$$

REFERENCES

- [1] S. Agarwal, K. Mierle, and Others, “Ceres Solver,” <http://ceres-solver.org>.
- [2] F. Dellaert and Others, “GTSAM,” <http://research.cc.gatech.edu/borg/gtsam>.
- [3] Y. Latif, C. Cadena, and J. Neira, “Robust graph SLAM back-ends: A comparative analysis,” in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.
- [4] T. Pfeifer, P. Weissig, S. Lange, and P. Protzel, “Robust factor graph optimization – a comparison for sensor fusion applications,” in *Proc. of Intl. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2016.
- [5] T. Pfeifer and P. Protzel, “Robust sensor fusion with self-tuning mixture models,” in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.
- [6] 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 *Proc. of Intelligent Vehicles Symposium*, 2013.
- [7] P. Reisdorf, T. Pfeifer, J. Breßler, S. Bauer, P. Weissig, S. Lange, G. Wanielik, and P. Protzel, “The problem of comparable gnss results – an approach for a uniform dataset with low-cost and reference data,” in *Proc. of Intl. Conf. on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR)*, 2016.
- [8] T. Pfeifer, S. Lange, and P. Protzel, “Dynamic covariance estimation – a parameter free approach to robust sensor fusion,” in *Proc. of Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017.
- [9] N. Sünderhauf and P. Protzel, “Switchable constraints for robust pose graph slam,” in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.
- [10] N. Sünderhauf and P. Protzel, “Switchable constraints vs. max-mixture models vs. rrr – a comparison of three approaches to robust pose graph SLAM,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2013.
- [11] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2013.
- [12] G. Agamennoni, P. Furgale, and R. Siegwart, “Self-tuning m-estimators,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2015.
- [13] E. Olson and P. Agarwal, “Inference on networks of mixtures for robust robot mapping,” in *Proc. of Robotics: Science and Systems (RSS)*. Sydney, Australia: Robotics: Science and Systems Foundation, 2012.
- [14] D. M. Rosen, M. Kaess, and J. 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.
- [15] R. Morton and E. Olson, “Robust sensor characterization via max-mixture models: GPS sensors,” in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013. estimation for learning noise distributions in mobile robotics,” in *1st Workshop on Robust and Multimodal Inference in Factor Graphs, ICRA*, 2013.
- [16] D. M. Rosen and J. J. Leonard, “Nonparametric density approaches to robust pose graph SLAM,” in *Proc. of Intl. Conf. on Robotics and Automation (ICRA)*, 2013.
- [17] D. Fourie, J. Leonard, and M. Kaess, “A nonparametric belief solution to the bayes tree,” in *Proc. of Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.
- [18] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, 1977.
- [19] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer: New York, 2006.
- [20] G. Schwarz *et al.*, “Estimating the dimension of a model,” *Annals of Statistics*, 1978.
- [21] H. Akaike, “A new look at the statistical model identification,” *IEEE transactions on automatic control*, 1974.