

# Inference on networks of mixtures for robust robot mapping

Edwin Olson

Computer Science and Engineering,  
University of Michigan,  
2260 Hayward Street,  
Ann Arbor, Michigan  
Email: ebolson@umich.edu

Pratik Agarwal

Computer Science and Engineering,  
University of Michigan,  
2260 Hayward Street,  
Ann Arbor, Michigan  
Email: pratikag@umich.edu

**Abstract**—The central challenge in robotic mapping is obtaining reliable data associations (or “loop closures”): state-of-the-art inference algorithms can fail catastrophically if even one erroneous loop closure is incorporated into the map. Consequently, much work has been done to push error rates closer to zero. However, a long-lived or multi-robot system will still encounter errors, leading to system failure.

We propose a fundamentally different approach: allow richer error models that allow the probability of a failure to be explicitly modeled. In other words, we optimize the map while simultaneously determining which loop closures are correct from within a single integrated Bayesian framework. Unlike earlier multiple-hypothesis approaches, our approach avoids exponential memory complexity and is fast enough for real-time performance.

We show that the proposed method not only allows loop closing errors to be automatically identified, but also that in extreme cases, the “front-end” loop-validation systems can be unnecessary. We demonstrate our system both on standard benchmarks and on the real-world datasets that motivated this work.

## I. INTRODUCTION

Robot mapping problems are often formulated as an inference problem on a factor graph: variable nodes (representing the location of robots or other landmarks in the environment) are related through factor nodes, which encode geometric relationships between those nodes. Recent Simultaneous Localization and Mapping (SLAM) algorithms can rapidly find maximum likelihood solutions for maps, exploiting both fundamental improvements in the understanding of the structure of mapping problems [1], [2], [3], and the computational convenience afforded by assuming that error models are simple uni-modal Gaussian [4].

Despite their convenience, Gaussian error models often poorly approximate the truth. In the SLAM domain, perceptual aliasing can lead to incorrect loop closures, and the resulting error can lead to divergence of the map estimate. Similarly, the wheels of a robot may sometimes grip and sometimes slip, leading to a bi-modal motion model. Similar challenges arise throughout robotics, including sonar and radar (with multi-path effects), target-tracking (where multiple disjoint hypotheses may warrant consideration), etc.

In the specific case of SLAM, it has become standard practice to decompose the problem into two halves: a “front-end” and “back-end”. The front-end is responsible for iden-

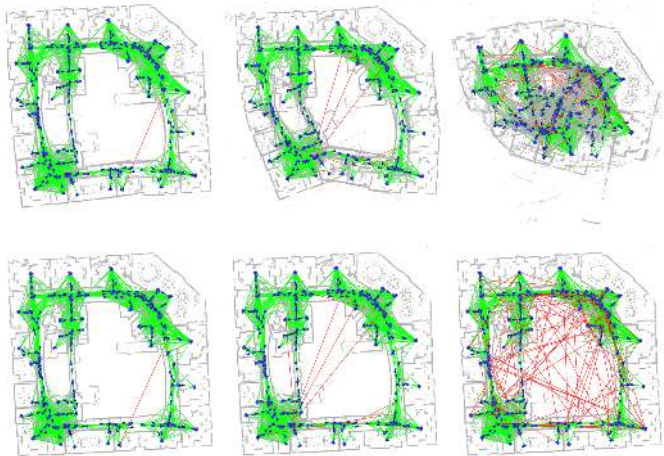


Fig. 1. Recovering a map in the presence of erroneous loop closures. We evaluated the robustness of our method by adding erroneous loop closures to the Intel data set. The top row reflects the posterior map as computed by a state-of-the-art sparse Cholesky factorization method with 1, 10, and 100 bad loop closures. The bottom row shows the posterior map for the same data set using our proposed max mixture method. While earlier methods produce maps with increasing global map deformation, our proposed method is essentially unaffected by the presence of the incorrect loop closures.

tifying and validating loop closures and constructing a factor graph; the back-end then performs inference (often maximum likelihood) on this factor graph. In most of the literature, it is assumed that the loop closures found by the front-end have noise that can be modeled as Gaussian.

For example, the front-end might assert that the robot is now at the same location that it was ten minutes ago (it has “closed a loop”), with an uncertainty of 1 meter. Suppose, however, that the robot was somewhere else entirely— a full 10 meters away. The back-end’s role is to compute the maximum likelihood map, and an error of ten standard deviations is so profoundly unlikely that the back-end will almost certainly never recover the correct map: it is compelled to distort the map so as to make the erroneous loop closure more probable (see Fig. 1).

The conventional strategy is to build better front-end systems. Indeed, much effort has been devoted to creating better front-end systems [5], [6], [7], and these approaches have succeeded in vastly reducing the rate of errors. But

for systems that accumulate many robot-hours of operation, or robots operating in particularly challenging environments, even an extremely low error rate still results in errors. These errors lead to divergence of the map and failure of the system.

Our recent efforts at building a team of robots that can cooperatively explore and map an urban environment [8] illustrate the challenges, and motivated this work. At the time, we modeled the uncertainty of odometry and loop closing edges with simple Gaussians, but despite extensive validation of these edges prior to optimization, some of these edges had large errors that were virtually impossible given their noise model. Even with a novel interface allowing a human to help untangle the resulting map [8], errors were still evident (see Fig. 6). Our subsequent analysis revealed that odometry edges were often to blame. We had assumed a 15% noise model, but our robots, driving under autonomous control, would occasionally get caught on small, unsensed obstacles. As a result, the robot actually encountered 100% error—five standard deviations given our prior noise model. The resulting error in our position estimates exacerbated the perceptual aliasing problem: our incorrect position prior would argue against correct loop closure hypotheses, and would favor some incorrect hypotheses.

In this paper, we propose a novel approach that allows efficient maximum-likelihood inference on factor graph networks that contain arbitrarily complex probability distributions. This is in contrast to state-of-the-art factor graph based methods, which are limited to uni-modal Gaussian distributions, and which suffer from the real-world problems described above. Specifically, we propose a new type of mixture model, a *max*-mixture, which provides similar expressivity as a sum-mixture, but avoids the associated computational costs. With such a mixture, the “slip or grip” odometry problem can be modeled as a multi-modal distribution, and loop closures can be accompanied by a “null” hypothesis. In essence, the back-end optimization system serves as a part of the front-end— playing an important role in validating loop closures and preventing divergence of the map.

We will demonstrate our system on real data, showing that it can easily handle the error rates of current front-end data validation systems, allowing robust operation even when these systems produce poor output. We will also illustrate that, in extreme cases, *no* front-end loop validation is required at all: all candidate loop closures can simply be added to the factor graph, and our approach simultaneously produces a maximum likelihood map while identifying the set of edges that are correct. This is an interesting development, since it provides a fully integrated Bayesian treatment of both mapping and data association, tasks which are usually decoupled.

It has previously been shown that exact inference on even poly-trees of mixtures is NP-hard [9]. Our method avoids exponential complexity at the expense of guaranteed convergence to the maximum likelihood solution. In this paper, we explore the robustness of our method, and characterize the error rates that can be handled.

In short, the contributions of this paper are:

- We formulate a new mixture model that provides significant computational advantages over the more traditional sum-of-Gaussians mixtures, while retaining similar expressive power.
- We develop an algorithm for fast maximum-likelihood inference on factor graph networks containing these max-mixtures.
- We demonstrate how robot mapping systems can use these methods to robustly handle errors in odometry and loop-closing systems.
- We characterize the robustness of our method to local minima, identifying factors (like error rate and overall graph degree) and their impact.
- We evaluate our algorithm on real-world datasets to demonstrate its practical applicability both in terms of the quality of results and the computation time required.

## II. RELATED WORK

We are not the first to consider estimation in the presence of non-Gaussian noise. Two well-known methods allow more complex error models to be used: particle filter methods and multiple hypothesis tracking (MHT) approaches.

Particle filters, perhaps best exemplified by Fast-SLAM [10], approximate arbitrary probability distributions through a finite number of samples. Particle filters attempt to explicitly (though non-parametrically) describe the posterior distribution. Unfortunately, the posterior grows in complexity over time, requiring an ever-increasing number of particles to maintain the quality of the posterior approximation. This growth quickly becomes untenable, forcing practical implementations to employ particle resampling techniques [11], [12], [13]. Unavoidably, resampling leads to a loss of information, since areas with low probability density are effectively truncated to zero. This loss of information can make it difficult to recover the correct solution, particularly after a protracted period of high uncertainty [14], [15].

Multiple Hypothesis Tracking approaches provide an alternative approach more closely related to mixture models [16]. These explicitly represent the posterior using an ensemble of Gaussians that collectively encode a mixture. However, the number of potential hypotheses tends to grow exponentially, forcing pruning of hypotheses and thus information loss. A similar problem occurs when explicitly performing inference on mixtures; Whyte et al approximate terrain using mixtures [17] but must resample the terrain in order to prevent an intractable number of mixture components. In both cases, information is lost that can result in lower-quality solutions.

In the special case where errors are modeled as uni-modal Gaussians, the maximum likelihood solution of the factor graph network can be found using non-linear least squares. Beginning with the observation that the information matrix is sparse [18], [19], [20], efforts to exploit that sparsity resulted in rapid improvements to map inference by leveraging sparse factorization and good variable-ordering heuristics [21], [22], [23]. While the fastest of these methods generally provide only maximum-likelihood inference (a shortcoming shared by our proposed method), approximate marginal estimation

methods are fast and easy to implement [24], [25]. It is highly desirable for new methods to be able to leverage the same insights that made these approaches so effective.

One method similar to our own explicitly introduces switching variables whose value determines whether or not a loop closure is accepted [26]. This work is notable for being the first to propose a practical way of dealing with front-end errors during inference. In comparison to our approach, the activation/deactivation of a loop closure is penalized through a separate cost function (as opposed to being integrated into a mixture model), they must assign initial values to these variables (as opposed to our implicit inference over the latent variables), and are limited to either “on” or “off” (as opposed to being able to model mixture models with multiple distinct modes).

Robustified cost functions [27] provide resilience to errors by reducing the cost associated with outliers, and have been widely used in the vision community [28], [29]. We show that robustified cost functions are subsumed by our mixture approach.

Our proposed method avoids the exponential growth in memory requirements of particle filter and MHT approaches by avoiding an explicit representation of the posterior density. Instead, like other methods based on sparse factorization, our method extracts a maximum likelihood estimate. Critically, while the memory cost of representing the posterior distribution grows exponentially, the cost of storing the underlying factor graph network (which implicitly encodes the posterior) grows only linearly with the size of the network. In other words, our method (which only stores the factor graph) can recover solutions that would have been culled by particle and MHT approaches. In addition, our approach benefits from the same sparsity and variable-ordering insights that have recently benefited uni-modal approaches.

### III. APPROACH

Our goal is to infer the posterior distribution of the state variable (or map)  $x$ , which can be written in terms of the factor potentials in the factor graph. The probability is conditioned on sensor observations  $z$ ; with an application of Bayes’ rule and by assuming an uninformative prior  $p(x)$ , we obtain:

$$p(x|z) \propto \prod_i p(z_i|x) \quad (1)$$

Prior to this work, it is generally assumed that the factor potentials  $p(z_i|x)$  can be written as Gaussians:

$$p(z_i|x) = \frac{1}{(2\pi)^{n/2} |\Lambda_i^{-1}|^{1/2}} e^{-\frac{1}{2}(f_i(x)-z_i)^T \Lambda_i (f_i(x)-z_i)} \quad (2)$$

Further, while  $f_i(x)$  is generally non-linear, it is assumed that it can be approximated using a first-order Taylor expansion such that  $f_i(x) \approx J_i \Delta x + f_i(x_0)$ .

The posterior maximum likelihood value can be easily solved in such cases by taking the logarithm of Eqn. 1, differentiating with respect to  $x$ , then solving for  $x$ . This classic least-squares approach leads to a simple linear system

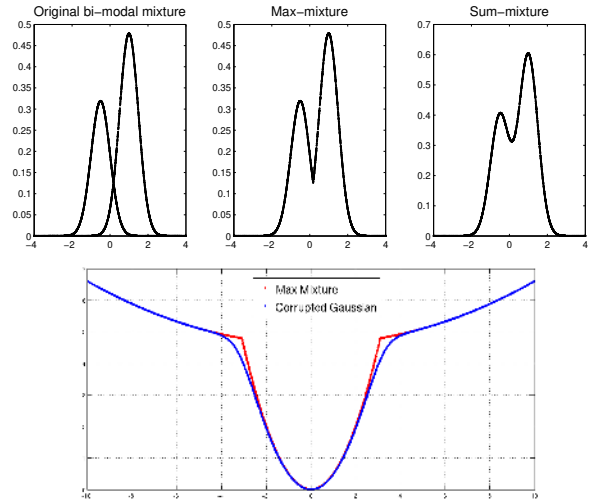


Fig. 2. Mixture Overview. Given two mixture components (top left), the max- and sum- mixtures produce different distributions. In both cases, arbitrary distributions can be approximated. A robustified cost function (in this case a corrupted Gaussian, bottom) can be constructed from two Gaussian components with equal means but different variances.

of the form  $Ax = b$ . Critically, this is possible because the logarithm operator can be “pushed” inside the product in Eqn. 1, reducing the product of  $N$  terms into a sum of  $N$  simple quadratic terms. No logarithms or exponentials remain, making the resulting expression easy to solve.

We might now consider a more complicated function  $p_i(x|z)$ , such as a sum-mixture of Gaussians:

$$p(z_i|x) = \sum_j w_j N(\mu_j, \Lambda_j^{-1}) \quad (3)$$

In this case, each  $N(\mu_j, \Lambda_j^{-1})$  represents a different Gaussian distribution. Such a sum-mixture allows great flexibility in specifying the distribution  $p(z_i|x)$ . For example, we can encode a robustified cost function by using two components with the same mean, but with different variances. Or, we can encode a bi-modal distribution.

The problem with a sum-mixture is that the maximum likelihood solution is no longer simple: the logarithm can no longer be pushed all the way into the individual Gaussian components: the summation in Eqn. 3 prevents it. As a result, the introduction of a sum-mixture means that it is no longer possible to derive a simple solution for  $x$ .

#### A. Max-Mixtures

Our solution to this problem is a new mixture model type, one based on a max operator rather than a sum:

$$p(z_i|x) = \max_j w_j N(\mu_j, \Lambda_j^{-1}) \quad (4)$$

While the change is relatively minor, the implications to optimization are profound. The logarithm *can* be pushed inside a max mixture: the max operator acts as a selector, returning a single well-behaved Gaussian component. The optimization process will be more thoroughly described in the following section.

A max mixture has much of the same character as a sum mixture and retains a similar expressivity: multi-modal distributions and robustified distributions can be similarly represented (see Fig. 2). Note, however, that when fitting a mixture to a desired probability distribution, different components will result for sum- and max- mixtures. Assuming that the distributions integrate to one is also handled differently: for a sum mixture,  $\sum w_j = 1$  is a necessary and sufficient condition; for a max mixture, proper normalization is generally more difficult to guarantee. Usefully, for maximum likelihood inference, it is inconsequential whether the distribution integrates to 1. Specifically, suppose that some normalization factor  $\gamma$  is required in order to ensure that a max mixture integrates to one. Since  $\gamma$  is used to scale the distribution, the log of the resulting max mixture is simply the log of the un-normalized distribution plus a constant. The addition of such a constant does not change the solution of a maximum-likelihood problem, and thus it is unnecessary for our purposes to compute  $\gamma$ .

### B. Cholesky-MM

We now show how max mixture distributions can be incorporated into existing graph optimization frameworks. The principle step in such a framework is to compute the Jacobian, residual, and information matrix for each factor potential. As we described previously, these are trivial to compute for a uni-modal Gaussian distribution.

For the max-mixture case, it might seem that computing the needed derivatives for the Jacobian is difficult: the max-mixture is not actually differentiable at the points where the maximum-likelihood component changes. While this makes it difficult to write a closed-form expression for the derivatives, they are none-the-less easy to compute.

The key observation is that the max operator serves as a selector: once the mixture component with the maximum likelihood is known, the behavior of the other mixture components is irrelevant. In other words, the solver simply iterates over each of the components, identifies the most probable, then returns the Jacobian, residual, and information matrix for that component scaled according to the weight  $w_j$  of that component. If the likelihood of two components are tied—an event which corresponds to evaluating the Jacobian at a non-differentiable point—we pick one of the components arbitrarily. However, these boundary regions comprise an area with zero probability mass.

The resulting Jacobians, residuals, and information matrices are combined into a large least-squares problem which we subsequently solve with a minimum-degree variable ordering heuristic followed by sparse Cholesky factorization, in a manner similar to that described by [3]. With our modifications to handle max-mixtures, we call our system Cholesky-MM.

It is often necessary to iterate the full least-squares problem several times. Each time, the best component in each max-mixture is re-evaluated: in essence, as the optimization proceeds, we dynamically select the best mixture component as an integral part of the optimization process.

Even in the non-mixture case, this sort of non-linear optimization cannot guarantee convergence to the global optimal solution. It is useful to think of a given inference problem as having a “basin of convergence”—a region that contains all the initial values of  $x$  that would ultimately converge to the global optimal solution. For most well-behaved problems with simple Gaussian distributions, the basin of convergence is large. Divergence occurs when the linearization error is so severe that the gradient points in the wrong direction.

The posterior distribution for a network with  $N$  mixtures, each with  $c$  components, is a mixture with as many as  $c^N$  components. In the worst-case, these could be non-overlapping, resulting in  $c^N$  local minima. The global optimal solution still has a basin of convergence: if our initial solution is “close” to the optimal solution, our algorithm will converge. But if the basin of convergence is extremely small, then the practical utility of our algorithm will be limited.

In other words, the key question to be answered about our approach is whether the basin of convergence is usefully large. Naturally, the size of this basin is strongly affected by the properties of the problem and the robustness of the algorithm used to search for a solution. One of the main results of this paper is to show that our approach yields a large basin of convergence for a wide range of useful robotics problems.

## IV. APPLICATIONS AND EVALUATION

In this section, we show how our approach can be applied to several real-world problems. We include quantitative evaluations of the performance of our algorithm, as well as characterize its robustness and runtime performance.

### A. Uncertain loop closures

We first consider the case where we have a front-end that produces loop closures with a relatively low, but non-zero, error rate. For each uncertain loop closure, we introduce a max-mixture consisting of two components: 1) the front-end’s loop closure and 2) a null hypothesis. The null hypothesis, representing the case when the loop closure is wrong, is implemented using a mixture component with a very large covariance. Weights are assigned to these two components in accordance with the error rate of the front-end.

In practice, the behavior of the algorithm is not particularly sensitive to the weights associated with the null hypotheses. The main benefit of our approach arises from having a larger probability associated with incorrect loop closures, as opposed to the exponentially-fast decreasing probability specified by the loop closer’s Gaussian. Even if the null hypothesis has a very low weight (we used  $10^{-5}$ ), it will provide a sufficiently plausible explanation of the data to prevent a radical distortion of the graph. Second, once the null hypothesis becomes dominant, its large variance results in a very weak gradient for the edge. As a result, the edge plays virtually no role in subsequent optimization. We set the mean of the null hypothesis equal to that of the front-end’s hypothesis so that the small amount of gradient that

remains produces a slight bias back towards the front-end’s hypothesis. If subsequent observations re-affirm the front-end’s hypothesis, it can still become active in the future. Unlike particle filter or MHT methods which must eventually truncate unlikely events, no information is lost.

A two-component mixture model in which both components have identical means but different variances can be viewed as a robustified cost function. In particular, parameters can be chosen so that a two-component max mixture closely approximates a corrupted Gaussian [27] (see Fig. 2).

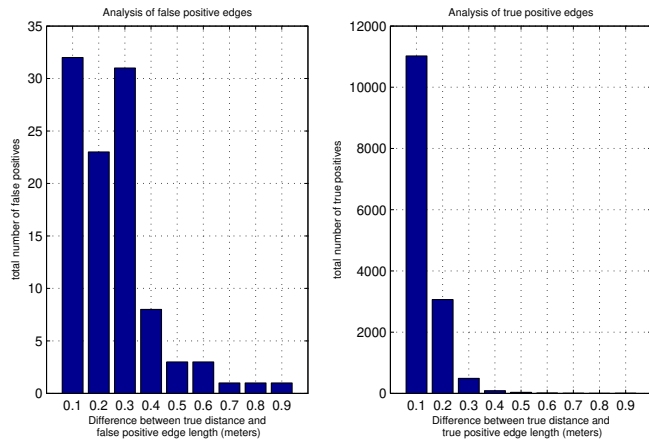


Fig. 4. Error distribution for true and false positives. Our method accepts some randomly-generated “false positives”, but an analysis of the error of those edges indicates that they (left) are only slightly worse than the error of true edges (right).

To evaluate the performance of our approach, we added randomly-generated loop closures to two standard benchmark datasets: the 3500 node Manhattan set [25] and the Intel data set. These were processed in an online fashion, adding one pose and its edges (both correct and incorrect). This mimics real-world operation better than a batch approach, and is more challenging due to the fact that it is easier to become caught in a local minimum since fewer edges are available to guide the optimization towards the global optimum.

For a given number of randomly-generated edges, we compute the posterior map generated by our method and a standard non-mixture method, using a laser-odometry solution as the initial state estimate. The mean-squared error of this map is compared to the global optimal solution [30], and listed in Fig. 3.

Our proposed method achieves dramatically lower mean-squared errors than standard non-mixture versions. While the true positive rate is nearly perfect in both experiments, some randomly-generated edges (labeled false positives) *are* accepted by our system. However, since the false positives are randomly generated, some of them (by chance) are actually close to the truth. Such “accidentally correct” edges *should* be accepted by our algorithm<sup>1</sup>.

<sup>1</sup>We favor generating “false positives” in a purely random way, even though it leads to “accidentally” correct edges. Any filtering operation to reject these low-error edges would introduce a free parameter (the error threshold) whose value could be tuned to favor the algorithm.

We can evaluate the quality of the accepted edges by comparing the error distribution of the true positives and false positives (see Fig. 4). As the histogram indicates, the error distribution is similar, though the error distribution for the false positives is slightly worse than for the true positives. Still, no extreme outliers (the type that cause divergence of the map) are accepted by our method.

### B. Multi-modal distributions

In the previous section, we demonstrated that our method can be used to encode null hypotheses, or equivalently, implement robustified cost functions—a capability similar to earlier work [26]. In that case, the probability distributions being approximated by each mixture have only a single maximum. Our use of a mixture model, however, also allows multi-modal distributions to be encoded. The ability to directly represent multi-modal distributions is a feature of our approach.

For example, one of the original motivating problems of this work was dealing with the “slip or grip” problem: the case where a robot’s wheels occasionally slip catastrophically, resulting in near zero motion. With a typical odometry noise model of 10-20%, such an outcome would wreak havoc on the posterior map.

Our approach to the “slip or grip” problem is to use a two-component mixture model: one component (with a large weight) corresponds to the usual 15% noise model, while the second component (with a low weight) is centered around zero. No changes to our optimization algorithm are required to handle such a case. However, since the distribution now has multiple local maxima, it poses a greater challenge in terms of robustness.

Of course, without some independent source of information that contradicts the odometry data, there is no way to determine that the wheels were slipping. To provide this independent information, we used a state-of-the-art scan matching system [31] to generate loop closures. We manually induced wheel slippage by pulling on the robot. Despite the good loop closures, standard methods are unable to recover the correct map. In contrast, our method determines that

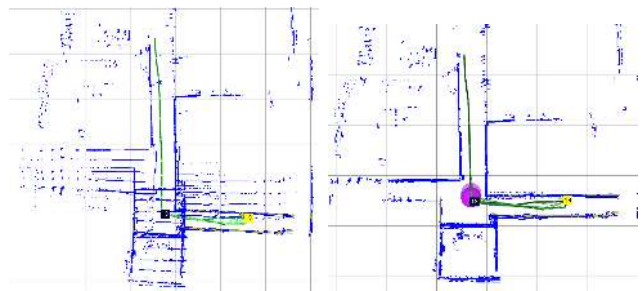


Fig. 5. Slip or Grip Example. We evaluate the ability of our algorithm to recover a good map in the presence of catastrophically slipping wheels. In this case, the robot is obtaining loop closures using a conventional laser scanner front-end. These loop closures are of high quality, but the odometry edges still cause significant map distortions when using standard methods (left). When a small probability is added to account for slippage, our mixture approach recovers a much improved map (right).

Manhattan Data Set						
True Loop Edges	False edges	True Positive	False Positive	Average FP Error	MSE (Our method)	MSE (Non-mixture)
2099	0	2099	0	NaN	0.6726	0.6726
2099	10	2099	0	NaN	0.6713	525.27
2099	100	2099	1	0.0208	0.6850	839.39
2099	200	2099	2	0.5001	0.6861	874.67
2099	500	2099	3	0.6477	0.6997	888.82
2099	1000	2099	10	0.7155	0.7195	893.98
2099	2000	2099	22	0.5947	0.7151	892.54
2099	3000	2099	36	0.5821	0.7316	896.01
2099	4000	2099	51	0.6155	0.8317	896.05

Intel Data Set						
True Loop Edges	False Edges	True Positive	False Positive	Average FP Error	MSE (Our method)	MSE (Non-mixture)
14731	0	14731	0	NaN	$7.122 \times 10^{-10}$	$1.55 \times 10^{-9}$
14731	10	14731	0	NaN	$7.123 \times 10^{-10}$	0.044
14731	100	14731	2	0.1769	$4.431 \times 10^{-6}$	2.919
14731	200	14731	9	0.1960	$5.583 \times 10^{-6}$	8.810
14731	500	14731	19	0.1676	$1.256 \times 10^{-5}$	34.49
14731	1000	14731	29	0.1851	$5.840 \times 10^{-5}$	71.86
14731	2000	14731	64	0.1937	$2.543 \times 10^{-4}$	86.37
14731	3000	14731	103	0.1896	$3.307 \times 10^{-4}$	91.04
14731	4000	14217	146	0.1699	0.014	95.36

Fig. 3. Null-hypothesis robustness. We evaluate the robustness of our method and a standard Gaussian method to the presence of randomly-generated edges. As the number of randomly-generated edges increases, the mean squared error (MSE) of standard approaches rapidly degenerates; our proposed method produces good maps even when the number of randomly-generated edges is large in comparison to the number of true edges. Our approach does accept some randomly-generated edges (labeled “false positives” above), however the error of these accepted edges is comparable to that of the true positives. In each case, the initial state estimate is that from the open-loop odometry.

“slip” mode is more likely than the “grip” mode, and recovers the correct map (see Fig. 5).

As part of our earlier multi-robot mapping work [8], we employed a team of 14 robots to explore a large urban environment. Wheel slippage contributed to a poor map in two ways: 1) the erroneous factor potentials themselves, and 2) the inability to identify good loop closures due to a low quality motion estimate. By using a better odometry model, our online system produced a significantly improved map (see Fig. 6).

### C. Eliminating the Front End

In current approaches, front-end systems are typically responsible for validating loop closures prior to adding them to the factor graph network. However, if the back-end can recover from errors, is it possible to omit the filtering entirely?

In certain cases, our inference method can eliminate the need for loop validation by the front-end. This is desirable from a conceptual standpoint: in principle, a better map

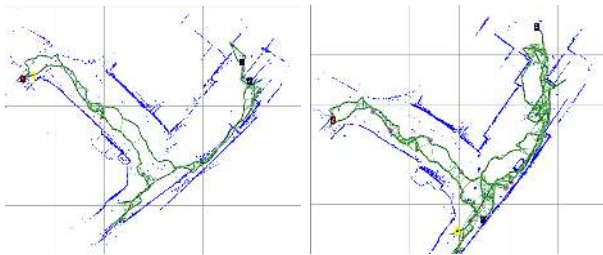


Fig. 6. Online results using odometry mixture model. The left figure shows a map of a  $30\text{m} \times 25\text{m}$  area in which our multi-robot urban mapping team produced a poor map due to wheel slippage and the ensuing inability to find loop-closures. With our odometry mixture model (right), the wheel slippage is (implicitly) detected, and we find additional loop closures. The result is a significantly improved map.

should result from handling loop closing and mapping from within an integrated Bayesian framework. The conventional decoupling of mapping into a front-end and back-end, while practical, prevents a fully Bayesian solution.

We evaluated this possibility using the Intel data set. At every pose, a laser scan matcher attempts a match to *every* previous pose. The top  $k$  matches (as measured by overlap of the two scans) are formed into a mixture containing  $k + 1$  components. (The extra component remains a null hypothesis to handle the case where all  $k$  matches are incorrect.) To push the experiment as far as possible, no position information was used to prune the set of  $k$  matches. Larger values of  $k$  provide robustness against perceptual aliasing, since it increases the likelihood that the correct match is present somewhere within the set of  $k$  components.

Note that this approach is significantly different than adding  $k$  conventional edges to the graph. When edges are added “in parallel”, the posterior reflects a weighted average of all of them. In our mixture formulation, only the selected mode has an effect on the solution: the others are effectively “switched off”.

An example of one mixture with  $k = 4$  putative matches is shown in Fig. 7. The weight of the components is set in proportion to the score of the scan matcher.

Running our system in an online fashion, we obtain the final map shown in Fig. 8. Online operation is more difficult than batch operation, since there is less information available early on to correct erroneous edges. Our system recovers a consistent global map despite the lack of any front-end loop validation.

The quality of the open-loop trajectory estimate plays an important role in determining whether the initial state estimate is within the basin of convergence. In this case, our open-loop trajectory estimate is fairly good, and our method is able to infer the correct mode for each mixture despite the

lack of any front-end loop validation.

The robustness of our method is amplified by better front-end systems: with better quality loop closures, the basin of convergence is enlarged, allowing good maps to be computed even when the open-loop trajectory is poor.

#### D. Robustness

We have identified two basic factors that have a significant influence on the success of our method: the number of incorrect loop closures and the node degree of the graph. The node degree is an important factor because it determines how over-determined the system is: it determines the degree to which correct edges can “overpower” incorrect edges.

To illustrate the relationship between these factors and the

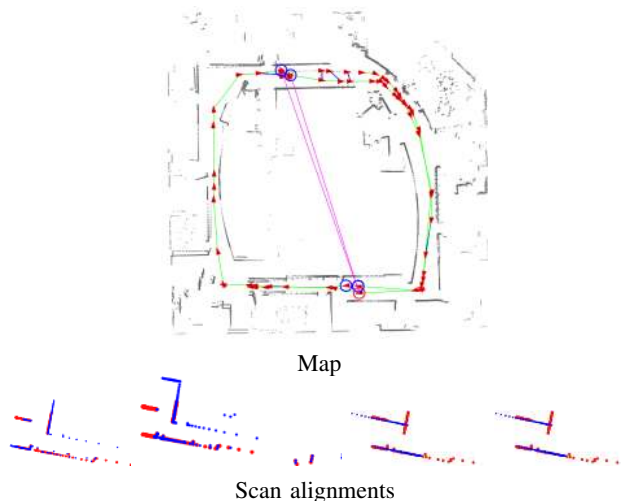


Fig. 7. Data-association as a mixture. Given a query pose (red circle at bottom of map), we perform a brute-force scan matching operation to all previous poses. The best 4 scan match results, based on overlap, are added to a max-mixture model that also includes a null hypothesis. The position of the best matches are shown as blue circles, and the corresponding scan matches shown at the bottom. The similarity in appearance between the blue poses represents a significant degree of perceptual aliasing. The scan matcher finds two correct matches and two incorrect matches. The two correct matches are the two blue circles at the bottom of the map and the first two scan alignments.



Fig. 8. Intel without front-end loop validation. Our system can identify correct loop closures and compute a posterior map from within a single integrated Bayesian framework (right); the typical front-end loop validation has been replaced with a  $k + 1$  mixture component containing the  $k$  best laser scan matches (based purely on overlap) plus a null hypothesis. In this experiment, we used  $k = 5$ . For reference, the open-loop trajectory of the robot is given on the left.

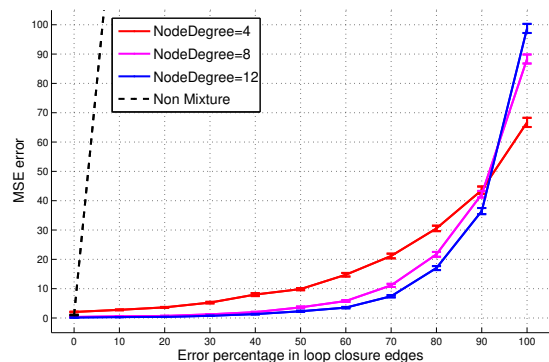


Fig. 9. Effect of error rate and node degree on robustness. We evaluate the quality of posterior maps (in terms of mean squared error) as a function of the percentage of bad edges and the node degree of the graph. Each data point represents the average of 3,000 random trials; the standard error is also plotted showing that the results are significant. The quality of the posterior graph is robust to even high levels of error, and is improved further by problems with a high node degree. Our methods, regardless of settings, dramatically out-perform non-mixture methods (black dotted line).

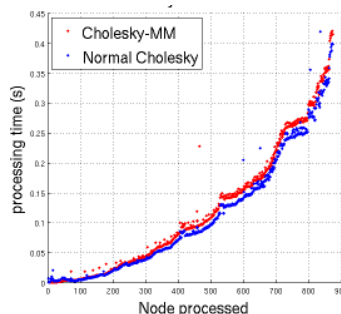


Fig. 10. Runtime performance. Using the Intel dataset, we plot the time required to compute a posterior map after every pose, using a batch solver. Our Intel dataset contains 875 nodes and 15605 edges, and each edge is modeled as a two-component max-mixture with a null hypothesis. The additional cost of handling mixtures is quite small in comparison to the total computation time.

resulting quality of the map (measured in terms of mean squared error), we considered a range of loop-closing error rates (ranging from 0% to 100%) for graphs with an average node degree of 4, 8, and 12. Note that an error rate of 80% means that incorrect loop closures outnumber correct loop closures by a ratio of 4:1. In each case, the vehicle’s noisy odometry is also provided. For each condition, we evaluate the performance of our method on 100,000 randomly-generated Manhattan-world graphs (see Fig. 9). Our method produces good maps even when the error rate is very high, and the performance improves further with increasing node degree. In contrast, a standard non-mixture approach diverges almost immediately.

#### E. Runtime Performance

The performance of our method is comparable to existing state-of-the-art sparse factorization methods (see Fig. 10). It takes additional time to identify the maximum likelihood mode for each mixture, but this cost is minor in comparison to the cost of solving the resulting linear system.

## V. CONCLUSION

We have described a method for performing inference on networks of mixtures, describing an application of our method to robot mapping. Our method consists of a novel mixture model based on a “max” operator that makes the computation of the Jacobian and residual fast, and we show how existing sparse factorization methods can be extended to incorporate these mixtures. We believe that such an approach is necessary for long-lived systems, since any system that relies on a *zero* error rate will fail.

We demonstrate how the mixture model allows null hypotheses and robustified cost functions to be incorporated into a maximum likelihood inference system. We show that our system is robust to a large error rates far in excess of what can be achieved with existing front-end loop validation methods. We further demonstrate a multi-modal formulation, addressing the “slip or grip” problem and showing that our system can make loop validation unnecessary in some cases.

Our algorithm cannot guarantee convergence to the global optimum, but we characterized the basin of convergence, demonstrating the relationship between error rate, node degree, and convergence to a good solution.

Finally, we demonstrate that the performance of our algorithm is similar to that of existing state-of-the-art maximum likelihood systems, and can be easily integrated into state-of-the-art methods [22].

## REFERENCES

- [1] P. M. Newman, “On the structure and solution of the simultaneous localisation and map building problem,” Ph.D. dissertation, University of Sydney, Australia, 1999.
- [2] U. Frese, “A proof for the approximate sparsity of SLAM information matrices,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005, pp. 331–337.
- [3] F. Dellaert, “Square root SAM,” in *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA, June 2005.
- [4] R. Smith, M. Self, and P. Cheeseman, “A stochastic map for uncertain spatial relationships,” in *Proceedings of the International Symposium of Robotics Research (ISRR)*, O. Faugeras and G. Giralt, Eds., 1988, pp. 467–474.
- [5] J. Neira and J. D. Tardos, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, December 2001.
- [6] T. Bailey, “Mobile robot localisation and mapping in extensive outdoor environments,” Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, August 2002.
- [7] E. Olson, “Recognizing places using spectrally clustered local matches,” *Robotics and Autonomous Systems*, 2009.
- [8] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Mariner, “Progress towards multi-robot reconnaissance and the MAGIC 2010 competition,” *Journal of Field Robotics*, September 2012.
- [9] U. Lerner and R. Parr, “Inference in hybrid networks: Theoretical limits and practical algorithms,” in *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*. San Francisco, CA: Morgan Kaufmann, 2001, pp. 310–331.
- [10] M. Montemerlo, “FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association,” Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.
- [11] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, “A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 206–211.
- [12] N. Kwak, I.-K. Kim, H.-C. Lee, and B. H. Lee, “Analysis of resampling process for the particle depletion problem in fastslam,” in *IEEE International Symposium on Robots and Human Interactive Communications (RO-MAN)*, 2007, pp. 200–205.
- [13] C. Stachniss, G. Grisetti, and W. Burgard, “Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005, pp. 667–672.
- [14] T. Bailey, J. Nieto, and E. Nebot, “Consistency of the FastSLAM algorithm,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, 2006, pp. 424–429.
- [15] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, April 2005, pp. 2432–2437.
- [16] S. S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, pp. 5–18, 2004.
- [17] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding, “A Bayesian algorithm for simultaneous localisation and map building,” in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, R. Jarvis and A. Zelinsky, Eds. Springer Berlin / Heidelberg, 2003, vol. 6, pp. 49–60.
- [18] S. Thrun and Y. Liu, “Multi-robot SLAM with sparse extended information filters,” in *Proceedings of the International Symposium of Robotics Research (ISRR)*, Sienna, Italy, 2003.
- [19] M. Walter, R. Eustice, and J. Leonard, “A provably consistent method for imposing exact sparsity in feature-based SLAM information filters,” in *Proceedings of the International Symposium of Robotics Research (ISRR)*, S. Thrun, R. Brooks, and H. Durrant-Whyte, Eds. San Francisco, CA: Springer, October 2005, pp. 214–234.
- [20] R. Eustice, H. Singh, J. Leonard, and M. Walter, “Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters,” *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1223–1242, December 2006.
- [21] F. Dellaert and M. Kaess, “Square root SAM: Simultaneous localization and mapping via square root information smoothing,” *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, December 2006.
- [22] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [23] K. Konolige, “Sparse sparse bundle adjustment,” in *British Machine Vision Conference*, Aberystwyth, Wales, August 2010.
- [24] M. Bosse, P. Newman, J. Leonard, and S. Teller, “Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework,” *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.
- [25] E. Olson, “Robust and efficient robotic mapping,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.
- [26] N. Sündenhauf and P. Protzel, “Towards a robust back-end for pose graph slam,” in *Proc. of IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.
- [27] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [28] H. Strasdat, J. Montiel, and A. Davison, “Scale drift-aware large scale monocular slam,” *Proceedings of Robotics: Science and Systems (RSS)*, vol. 2, no. 3, p. 5, 2010.
- [29] G. Sibley, C. Mei, I. Reid, and P. Newman, “Adaptive relative bundle adjustment,” in *Robotics Science and Systems Conference*, Seattle, USA, June 2009.
- [30] E. Olson, “Evaluating back-ends: Metrics,” in *Automated SLAM Evaluation Workshop, Robotics Science and Systems*, Los Angeles, USA, 2011.
- [31] —, “Real-time correlative scan matching,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, June 2009.