The International Journal of Robotics Research
http://ijr.sagepub.com/

Inference on networks of mixtures for robust robot mapping

Edwin Olson and Pratik Agarwal

The International Journal of Robotics Research 2013 32: 826
DOI: 10.1177/0278364913479413

The International Journal of
Robotics Research
32(7) 826–840
© The Author(s) 2013
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364913479413
ijr.sagepub.com

Inference on networks of mixtures for robust robot mapping

Edwin Olson[1] and Pratik Agarwal[1,2]

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, rather than characterizing loop closures as being "right" or "wrong", we propose characterizing the error of those loop closures in a more expressive manner that can account for their non-Gaussian behavior. Our approach leads to an fully integrated Bayesian framework for dealing with error-prone data. 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 data sets that motivated this work.

Keywords

SLAM, robust inference, mixture models

1. 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 (Newman, 1999; Dellaert, 2005; Frese, 2005), and the computational convenience afforded by assuming that error models are simple unimodal Gaussian (Smith et al., 1988).

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 bimodal 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 identifying 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 10 minutes ago (it has "closed a loop"), with an uncertainty of 1 m. Suppose, however, that the robot was somewhere else entirely: a full 10 m away. The back-end's role is to compute the maximum likelihood map, and an error of 10 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 Figure 1).

[1] Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA
[2] University of Freiburg, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany

Corresponding author:
Edwin Olson, Computer Science and Engineering, University of Michigan, 2260 Hayward Street, Ann Arbor, MI 48109, USA.
Email: ebolson@umich.edu

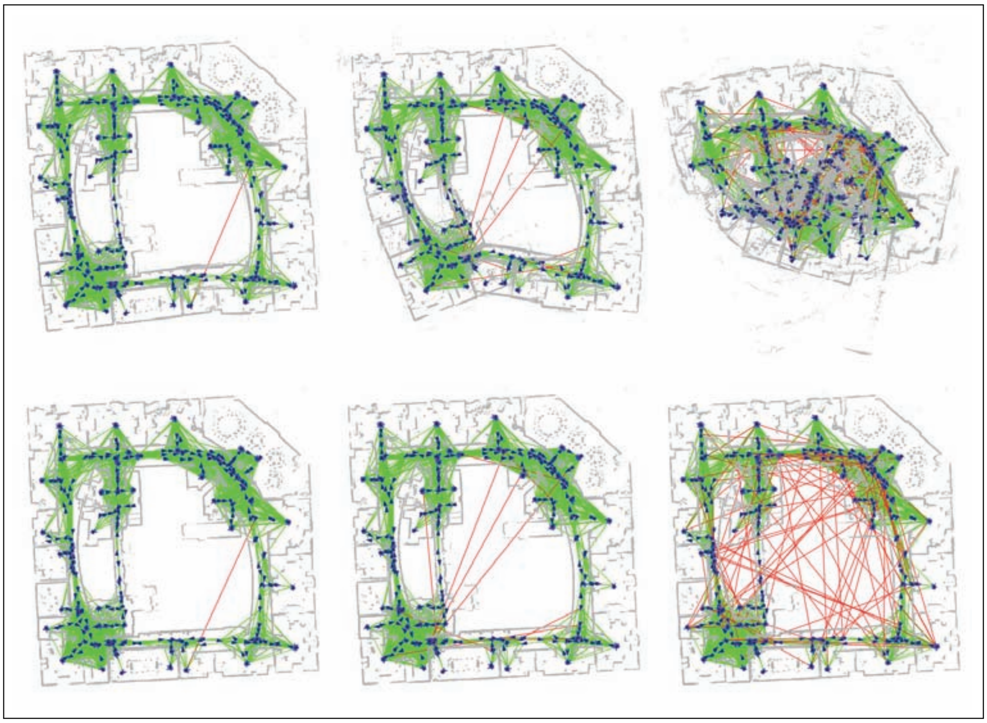

Fig. 1. Recovering a map in the presence of erroneous loop closures. We evaluated the robustness of our method by adding erroneous loop closures to the Intel data set. The top row reflects the posterior map as computed by a state-of-the-art sparse Cholesky factorization method with 1, 10, and 100 bad 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.

The conventional strategy is to build better front-end systems. Indeed, much effort has been devoted to creating better front-end systems (Neira and Tardos, 2001; Bailey, 2002; Olson, 2009b), and these approaches have succeeded in vastly reducing the rate of errors. However, 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 (Olson et al., 2012) 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 (Crossman et al., 2012), errors were still evident (see Figure 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 unimodal 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 (Lerner and Parr, 2001). 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 as follows:

- 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 (such as error rate and overall graph degree) and their impact. We show that the basin of convergence is large for a variety of benchmark 2D and 3D data sets over a range of plausible parameter values.
- We evaluate our algorithm on real-world data sets to demonstrate its practical applicability both in terms of the quality of results and the computation time required.

2. 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 FastSLAM (Montemerlo, 2003), approximate arbitrary probability distributions through a finite number of samples. Particle filters attempt to explicitly (and 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 (Hähnel et al., 2003; Stachniss et al., 2005; Kwak et al., 2007). 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 (Grisetti et al., 2005; Bailey et al., 2006).

MHT approaches (Durrant-Whyte et al., 2003; Blackman, 2004) provide an alternative approach more closely related to mixture models. These explicitly represent the posterior using an ensemble of Gaussians that collectively encode a mixture. However, the size of the ensemble also grows rapidly: the posterior distribution arising from N observations each with c components is a mixture with c^N components. As with particle filters, this exponential blow-up quickly becomes intractable, forcing approximations that cause information loss and ultimately lead to errors.

In the special case where errors are modeled as unimodal 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 (Thrun and Liu, 2003; Walter et al., 2005; Eustice et al., 2006), efforts to exploit that sparsity resulted in rapid improvements to map inference by leveraging sparse factorization and good variable-ordering heuristics (Dellaert and Kaess, 2006; Kaess et al., 2008; Konolige, 2010; Agarwal and Olson, 2012). 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 (Bosse et al., 2004; Olson, 2008). It is highly desirable for new methods to be able to leverage the same insights that made these approaches so effective.

Sum-mixtures of Gaussians have been recently been explored (Pfingsthorn and Birk, 2012). The mixtures are converted into unimodal Gaussians via a “pre-filtering” step, yielding a problem that can be approximately solved using standard sparse methods. Another approach for increasing robustness is to use the χ^2 of individual measurements in order to identify clusters of mutually consistent loop closures (Latif et al., 2012b). This mutual consistency can be re-evaluated as new information arrives (Latif et al., 2012a). The “max-mixture” approach described in this paper differs from these approaches in that the challenging process of approximating a sum-mixture is avoided, and that the set of activated modes is intrinsically re-evaluated at every iteration.

One method similar to our own explicitly introduces switching variables whose value determines whether or not a loop closure is accepted (Sunderhauf and Protzel, 2012). This work is notable for being the first to propose a practical way of dealing with front-end errors. In comparison to our approach, they penalize the activation/deactivation of a loop closure through a separate cost function (as opposed to being integrated into a mixture model), and must assign initial values to these switching variables (as opposed to our implicit inference over the latent variables). Our approach does not introduce switching variables, instead explaining poor quality data in the form of a non-Gaussian probability density function which can be arbitrarily complex (including having multiple maxima).

Robustified cost functions Hartley and Zisserman (2004) provide resilience to errors by reducing the cost associated with outliers, and have been widely used in the vision community Sibley et al. (2009); Strasdat et al. (2010). Our proposed max-mixture model can approximate arbitrary probability distributions, including those arising from robustified cost functions.

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 unimodal approaches.

3. 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

$$p(x|z) \propto \prod_i p(z_i|x) \tag{1}$$

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

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

Further, while $f_i(x)$ is generally nonlinear, 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 Equation (1), differentiating with respect to x, then solving for x. This classic least-squares approach leads to a simple linear system of the form $Ax = b$. Critically, this is possible because the logarithm operator can be "pushed" inside the product in Equation (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:

$$p(z_i|x) = \sum_i w_i N(\mu_i, \Lambda_i^{-1}) \tag{3}$$

In this case, each $N(\mu_i, \Lambda_i^{-1})$ 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. More complicated distributions, including those with multiple maxima, can also be represented.

The problem with a sum-mixture is that the maximum likelihood solution is no longer simple: the logarithm can no longer be pushed all of the way into the individual Gaussian components: the summation in Equation (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.

3.1. Max-mixtures

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

$$p(z_i|x) = \max_i w_i N(\mu_i, \Lambda_i^{-1}) \tag{4}$$

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.

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 Figure 2). Note, however, that when fitting a mixture to a desired probability distribution, different components will result for sum- and max-mixtures. Assuring that the distributions integrate to one is also handled differently: for a sum-mixture, $\sum w_i = 1$ 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 one. Specifically, suppose that some normalization factor γ is required in order to ensure that a max-mixture integrates to one. Since γ is used to scale the distribution, the log of the resulting max-mixture is simply the log of the unnormalized 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 γ.

3.2. 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 unimodal 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 nonetheless 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

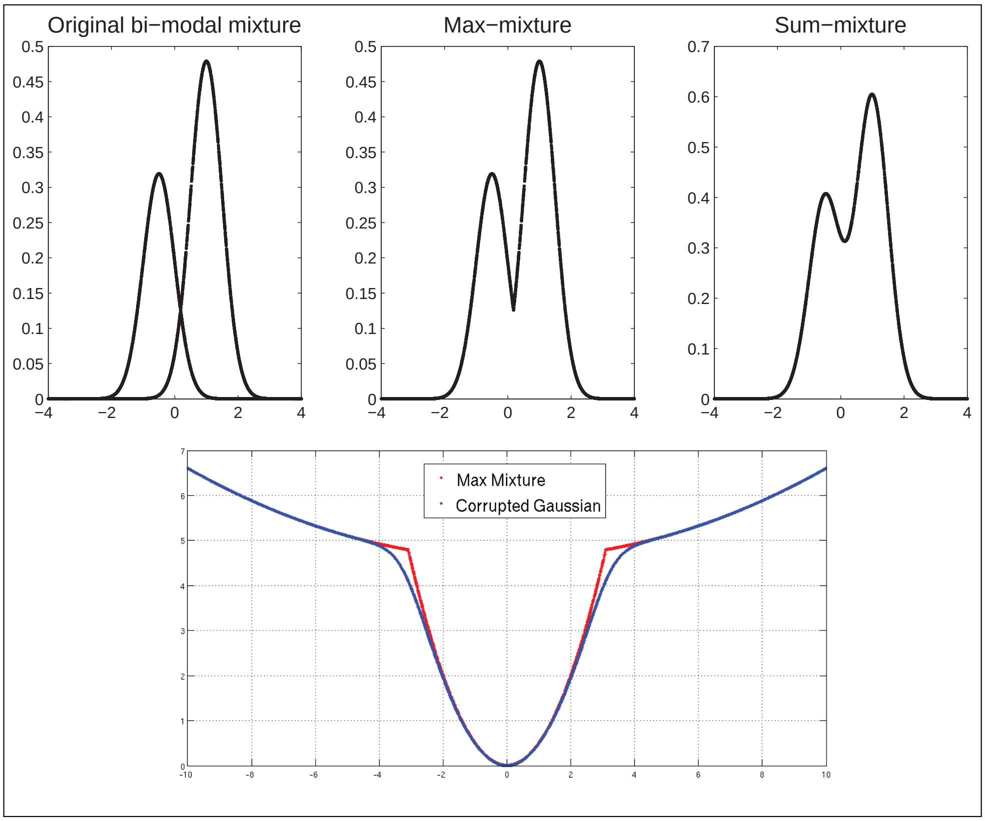

Fig. 2. Mixture overview. Given two mixture components (top left), the max- and sum-mixtures produce different distributions. In both cases, arbitrary distributions can be approximated. A robustified cost function (in this case a corrupted Gaussian, bottom) can be constructed from two Gaussian components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

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. 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 using Gauss–Newton steps, in a manner similar to that described by Dellaert (2005). 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. However, 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.

4. 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 run time performance.

4.1. 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. In our experiments, we set the mean of the null-hypothesis component equal to that of the other component. Weights and variances 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 (for example, 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 reaffirm 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 (Hartley and Zisserman, 2004) (see Figure 2).

To evaluate the performance of our approach, we added randomly generated loop closures to two standard benchmark data sets: the 3500 node Manhattan set (Olson, 2008) and the Intel data set (Howard and Roy, 2003). These were processed in an online fashion, adding one pose at a time and potentially one or more loop closures (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 with the global optimal solution (Olson, 2011), and listed in Table 1.

Our proposed method achieves dramatically lower mean squared errors (MSEs)[1] 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.[2]

We can evaluate the quality of the accepted edges by comparing the error distribution of the true positives and false positives (see Figure 3). 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.

4.1.1. Extension to six degrees of freedom While many important domains can be described in terms of planar motion (with 3D factor potentials reflecting translation in *x*, translation in *y*, and rotation), there is increasing interest in six-degree-of-freedom problems. Rotation is a major source of non-linearity in SLAM problems, and full six-degree-of-freedom problems can be particularly challenging.

To evaluate the performance of our method on a six-degree-of-freedom problem, we used the benchmark Sphere2500 data set (Kaess et al., 2008). This data set does not contain incorrect loop closures, and so we added additional erroneous loop closures. In Figure 4, we show the results of a standard Cholesky solver and our max-mixture approach applied to corrupted Sphere2500 data set with an additional 1, 10, and 100 erroneous edges. As in previous examples, the maps produced by a standard method quickly deteriorate. In contrast, the proposed method produces posterior maps that are essentially unaffected by the errors. In this experiment, each loop-closure edge in the graph (both correct and false) was modeled as a two-component max-mixture in which the second component had a large variance (10^7 times larger than the hypothesis itself) and a small weight (10^{-7}). The method is relatively insensitive to the particular values used: the critical factor is ensuring that, if the hypothesis is incorrect, the null hypothesis provides a higher probability explanation than the putative (incorrect) hypothesis, and that the null hypothesis is sufficiently weak so as to not distort the final solution. The impact of the relative strength of the null hypothesis on the basin of convergence is explored experimentally in Section 4.5.

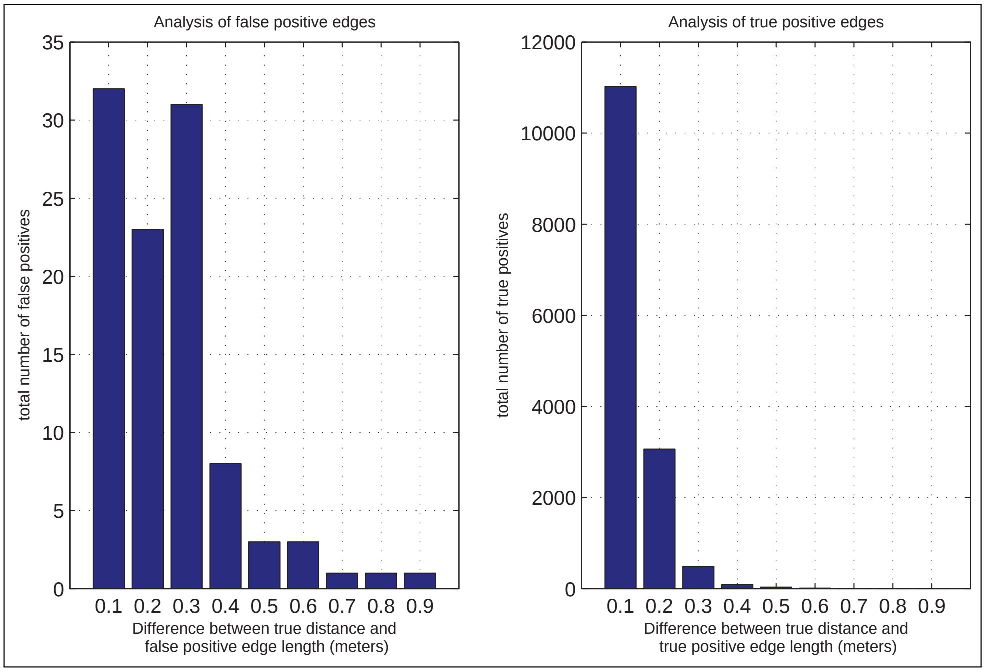

Fig. 3. Error distribution for true and false positives. Our method accepts some randomly-generated “false positives”, but an analysis of the error of those edges indicates that they (left) are only slightly worse than the error of true edges (right).

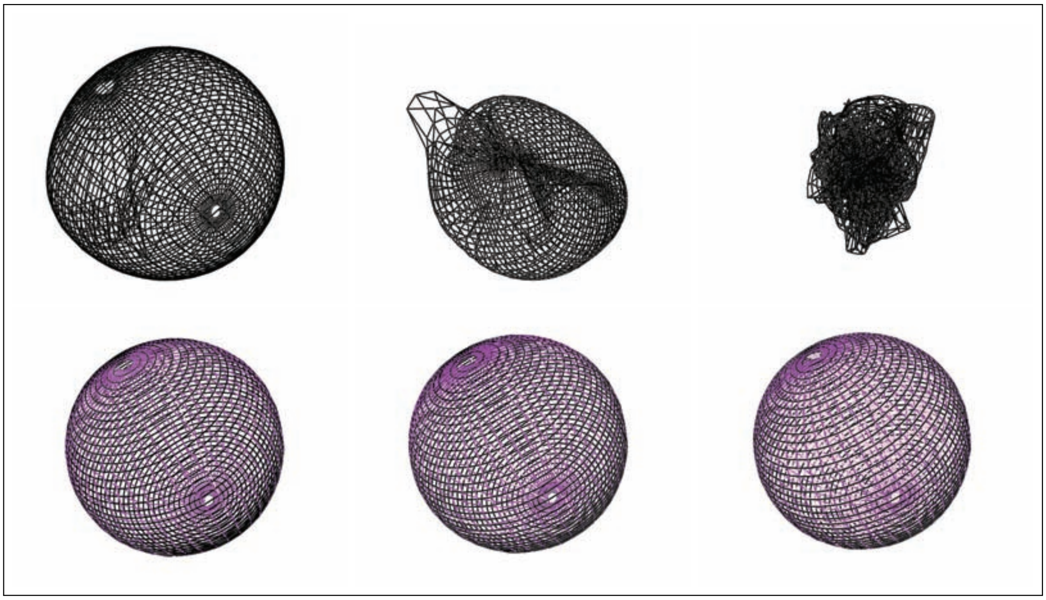

Fig. 4. Recovering a map in the presence of outliers. We evaluated the robustness of our method by adding erroneous loop closure edges to the Sphere2500 data set, a data set with a full six degrees of freedom. The top row reflects the posterior of the map with a standard least-squares Cholesky solver with 1, 10, and 100 wrong edges. The bottom row shows the corresponding map for the same data set using max-mixtures method.

4.2. Multi-modal distributions

In the previous sections, we demonstrated that our method can be used to encode null hypotheses, or equivalently, implement robustified cost functions: a capability similar to earlier work (Sunderhauf and Protzel, 2012). 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.

Table 1. 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.

Manhattan Data Set

True edges	False edges	True positive	False positive	Average FP error	MSE (our method)	MSE (non-mixture)
2,099	0	2,099	0	NaN	0.6726	0.6726
2,099	10	2,099	0	NaN	0.6713	525.27
2,099	100	2,099	1	0.0208	0.6850	839.39
2,099	200	2,099	2	0.5001	0.6861	874.67
2,099	500	2,099	3	0.6477	0.6997	888.82
2,099	1,000	2,099	10	0.7155	0.7195	893.98
2,099	2,000	2,099	22	0.5947	0.7151	892.54
2,099	3,000	2,099	36	0.5821	0.7316	896.01
2,099	4,000	2,099	51	0.6155	0.8317	896.05

Intel Data Set

True edges	False edges	True positive	False positive	Average FP error	MSE (our method)	MSE (non-mixture)
14,731	0	14,731	0	NaN	7.122×10^{-10}	1.55×10^{-9}
14,731	10	14,731	0	NaN	7.123×10^{-10}	0.044
14,731	100	14,731	2	0.1769	4.431×10^{-6}	2.919
14,731	200	14,731	9	0.1960	5.583×10^{-6}	8.810
14,731	500	14,731	19	0.1676	1.256×10^{-5}	34.49
14,731	1,000	14,731	29	0.1851	5.840×10^{-5}	71.86
14,731	2,000	14,731	64	0.1937	2.543×10^{-4}	86.37
14,731	3,000	14,731	103	0.1896	3.307×10^{-4}	91.04
14,731	4,000	14,217	146	0.1699	0.014	95.36

4.2.1. Slip or grip problem 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 (Olson, 2009a) 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 "slip" mode is more likely than the "grip" mode, and recovers the correct map (see Figure 5).

As part of our earlier multi-robot mapping work (Ranganathan et al., 2010; Olson et al., 2012), 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 Figure 6).

4.2.2. Simplifying the front end with "one-of-k" formulation 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?

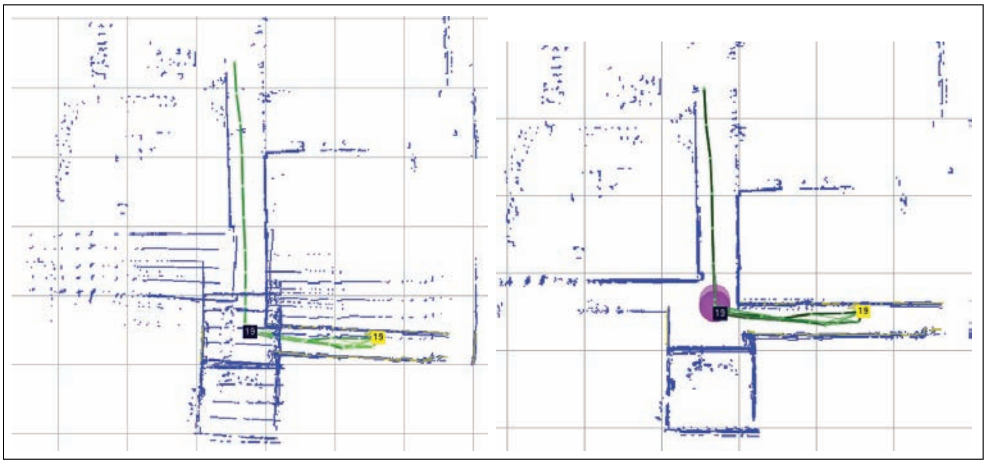

Fig. 5. Slip or grip example. We evaluate the ability of our algorithm to recover a good map in the presence of catastrophically slipping wheels. In this case, the robot is obtaining loop closures using a conventional laser scanner front-end. These loop closures are of high quality, but the odometry 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).

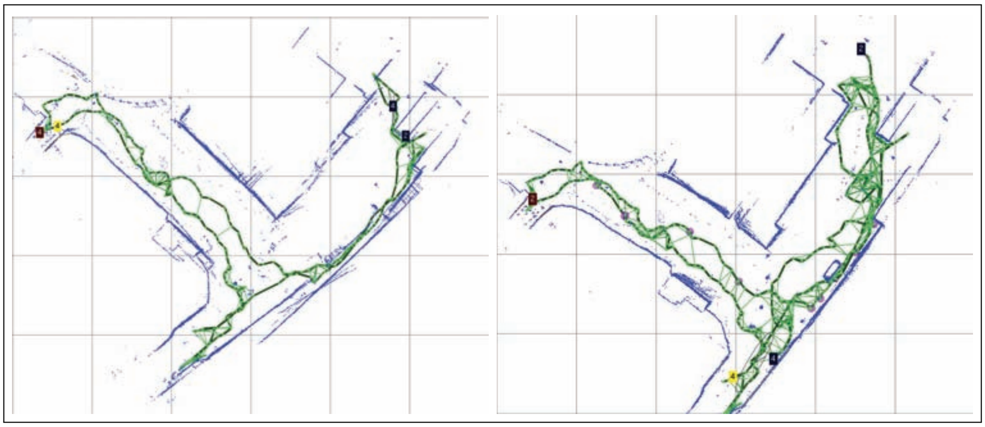

Fig. 6. Online results using odometry mixture model. The left figure shows a map of a 30 m × 25 m area in which our multi-robot urban mapping team produced a poor map due to wheel slippage and the ensuing inability to find loop closures. With our odometry mixture model (right), the wheel slippage is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

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 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.

An example of one mixture with $k = 4$ putative matches is shown in Figure 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 Figure 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.

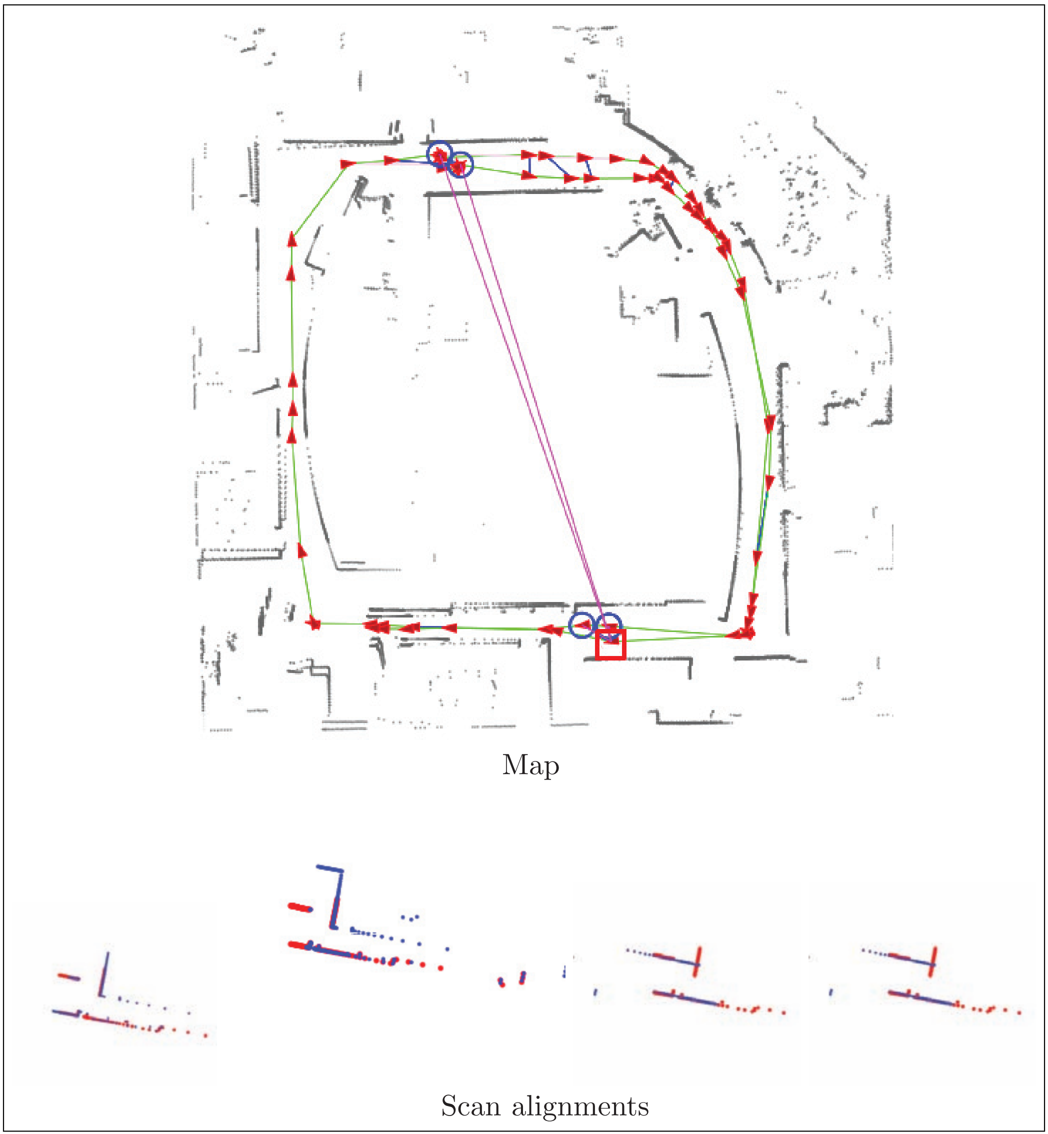

Fig. 7. Data association as a mixture. Given a query pose (red square at bottom of map), we perform a brute-force scan matching operation to all previous poses. The best four scan match results, based on overlap, are added to a max-mixture model that also includes a null hypothesis. The position of the 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.

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.

4.2.3. Performance impact of uncertainty modeling In the previous section, uncertain data associations were modeled as “one-of-k” mixtures, in which multiple candidate loop closures were grouped together in a single edge. Alternatively, each candidate loop closure could be encoded as a two-component mixture in a “null-hypothesis” style mixture; this approach is well-suited to the case where little is known about alternatives to a putative loop closure, while still allowing for the possibility that it is incorrect. (It is also possible that the mixture components have no obvious semantic meaning: the mixture model could simply be approximating a more complex distribution. For example, a max-mixture could be fit to an empirically derived cost function from a correlation-based scan matcher (Olson, 2009a).)

In this section, we explore the performance impact of “one-of-k” mixtures versus “null-hypothesis” mixtures. Consider a “one-of-k” mixture consisting of three candidate loop closures plus a null hypothesis: $\{L_1, L_2, L_3, \text{null}\}$. This can be transformed into three “null-hypothesis” mixtures: $\{L_1, \text{null}\}$, $\{L_2, \text{null}\}$, and $\{L_3, \text{null}\}$. These two formulations

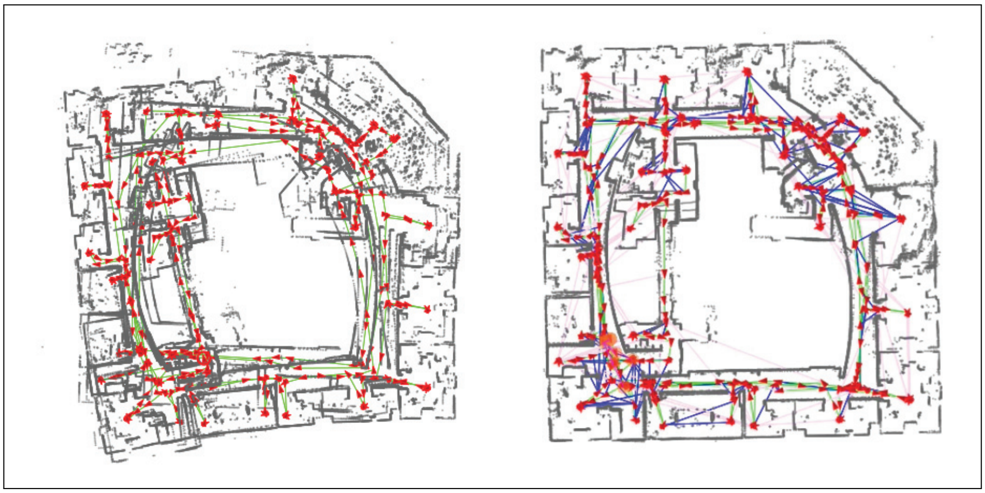

Fig. 8. Intel without front-end loop validation. Our system can identify correct loop closures and compute a posterior map from within a single integrated Bayesian framework (right); the typical front-end loop validation has been replaced with a $k + 1$ mixture component containing the k best laser scan 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.

are not exactly equivalent: the "one-of-k" encodes mutual-exclusion between the hypotheses, whereas the k separate "null-hypotheses" would permit solutions in which more than one of the loop closures was accepted. In many practical situations, however, the semantic difference is relatively minor. In this section, we show that the performance impact of this choice can be dramatic.

In Table 2, we show results from an experiment in which both formulations were used. We consider the case where loop hypotheses are generated in pairs and in triples; this leads to "one-of-k" mixtures with three and four components respectively once a null hypothesis is added. For the "one-of-k" formulation, the null hypotheses has a mean chosen randomly from one of the k constraints and a large variance roughly the size of the whole map. An alternative graph, constructed from "null-hypothesis" mixtures is constructed from the same sets of loop-closure hypotheses; naturally, each of these has two components.

An obvious difference between the two formulations is the number of edges in the graph: the "null-hypothesis" approach creates many more edges. That alone can be expected to increase computational time versus a "one-of-k" encoding. However, a more critical scaling issue becomes apparent: the "null-hypothesis" formulation leads to dramatically higher fill-in due to the fact that more nodes are connected to factor potentials. In contrast, a "one-of-k" edge does not contribute the same fill-in, since only one of the components in the mixture has any effect during a single Cholesky iteration. In other words, the max operator in the max-mixture formulation effectively severs edges corresponding to sub-dominant mixture components, improving the sparsity of the information matrix.

The difference in fill-in leads to significant increases in run time: on the Manhatten-3500 data set with groups of three candidate hypotheses, moving from a "one-of-k" to a "null-hypothesis" formulation causes an increase in non-zero entries from 0.17% to 4.3%, with a corresponding increase in computation time from 0.13 s to 1.2 s.

Table 2 also reports run times for Sunderhauf and Protzel's switchable constraints approach (Sunderhauf and Protzel, 2012), which adds an additional "switching" variable for every edge. In this way, it is semantically comparable with the "null-hypothesis" approach, although the formulation is somewhat different. The run time of the switchable constraints approach, 1.5 s, is somewhat worse than "null-hypothesis" approach and much worse than the "one-of-k" approach. (Note, for this comparison, all methods were implemented in the g2o (Kummerle et al., 2011) framework using *CHOLMOD* with a *COLAMD* variable ordering.)

These results suggest that, when semantically reasonable to do so, it is preferable to use "one-of-k" mixtures rather than either "null-hypothesis" mixtures or switchable constraints.

4.3. 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 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

Table 2. Run time comparison between switchable constraints, “null-hypothesis”, “one-of-*k*” formulations. Groups of related hypotheses were generated and either grouped as a single set of mutually exclusive edges (one-of-*k*), individually associated with a null hypothesis, or individually associated with a switching variable (Sunderhauf and Protzel, 2012). Using the one-of-*k* formulation reduces the effective connectivity in the graph, reducing fill-in, and resulting in faster computation time.

Data set		Switchable constraints	Bimodal MM	*k*-modal MM
Manhattan	Iteration time (s)	0.90 s	0.74 s	0.13 s
with $k = 2$	Fill-in (%)	1.50%	2.89%	0.17%
outliers = 2099	Number of loop edges	4,198	4,198	2,099
	Number of components	—	2	3
Manhattan	Iteration time (s)	1.5 s	1.2 s	0.13 s
with $k = 3$	Fill-in (%)	1.70%	4.30%	0.17%
outliers= 4198	Number of loop edges	6,277	6,277	2,099
	Number of components	—	2	4

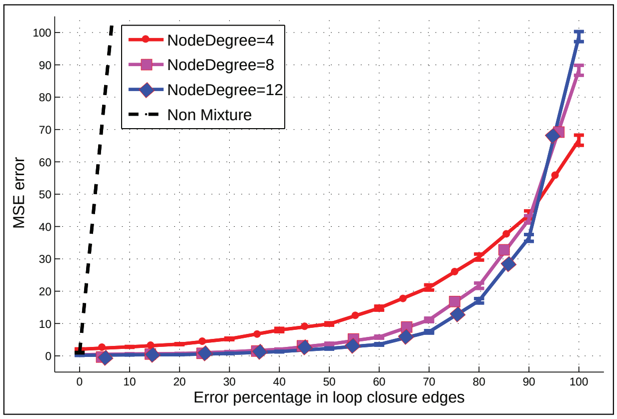

Fig. 9. Effect of error rate and node degree on robustness. We evaluate the quality of posterior maps (in terms of mean squared error) as a function of the percentage of bad edges and the node degree of the graph. Each data point represents the average of 3,000 random trials; the standard error is also 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 outperform non-mixture methods (black dotted line).

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 Figure 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.

4.4. *Run time performance*

The performance of our method is comparable to existing state-of-the-art sparse factorization methods (see Figure 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.

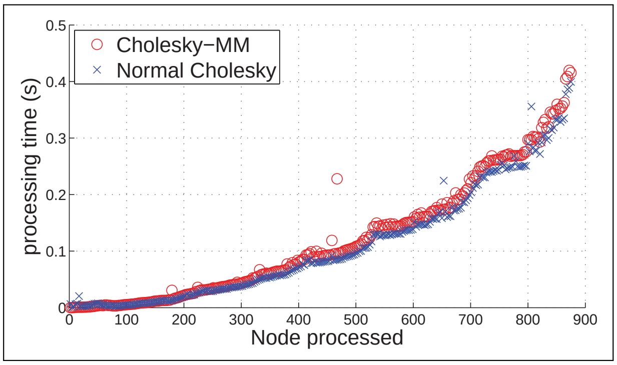

Fig. 10. Run time performance. Using the Intel data set, we plot the time required to compute a posterior map after every pose, using a batch solver. Our Intel data set contains 875 nodes and 15,605 edges, and each edge is modeled as a two-component max-mixture with a null hypothesis. The additional cost of handling mixtures is quite small in comparison with the total computation time. Run times were computed using the Java-based april.graph library, which is slower than g2o, but exhibits the same scaling behavior as other methods.

4.5. *Basin of convergence*

A key issue in non-linear optimization methods is whether the globally optimal solution will be found, or whether the optimization process will get stuck in a local minimum. This is a function of the initial solution as well as the parameters of the problem. In this section, we describe the effects of these parameters on the robustness of our method, as well as an experiment to empirically evaluate the magnitude of these effects.

The effect of the initial solution is relatively straightforward: some initial solutions provide a better path for the optimization system to follow. In high-noise cases, some initial solutions may be far from the desired solution and in a different basin of convergence, leading to a poor solution. We consider two initializations: (1) open-loop odometry (well-suited to online optimization) and (2) an approach which initializes each node relative to its oldest neighbor (a heuristic used by TORO (Grisetti et al., 2007) and implemented within g2o (Kummerle et al., 2011), most useful in batch processing).

The type of errors that occur also affect the robustness of the method. In this analysis, we consider four types of erroneous loop closures based on the Vertigo package (Sünderhauf, 2012): (1) random errors, (2) locally clustered (but not mutually consistent) errors, (3) randomly grouped errors in which the groups are mutually consistent (group size = 10), and (4) locally grouped errors (group size = 10).

The relative "strength" of the null hypothesis in comparison to the putative hypothesis also has an effect on the optimization. This strength can be described in terms of two parameters. The weight parameter (w) is the mixing parameter associated with the null-hypothesis component. Larger values of w increase the likelihood of the null hypothesis and cause the system to reject more of the putative hypotheses. If w is too small, and the system will accept incorrect hypotheses.

The second parameter, s, is the scale factor used in generating the information matrix associated with the null hypothesis from the information matrix associated with the putative hypothesis. When $s = 1$, both mixture components are identical and thus no robustness from the method can be expected. Smaller values (closer to zero) of s yield null hypotheses that are less certain than the putative hypothesis. This is equivalent to increasing its covariance, which pushes more probability mass away from the mean. This not only allows the null-hypothesis to produce a higher probability explanation of observed data, but also results in less curvature in the cost function. As s gets smaller, the cost function becomes increasingly flat, decreasing any influence of the mixture on the posterior solution.

We explore the effect of these parameters in Figure 11. The columns represent the four different outlier generation strategies and rows represent different data sets and initializations (not all combinations are presented for space reasons). Within each cell, the parameters w and s are swept resulting in a 2D grid of map scores. At each data point, a graph was constructed and solved using the max-mixture method and the log of the mean squared error (evaluated with respect to ground truth) is plotted according to color.

The data in Figure 11 shows graphically how to tune the free parameters w and s to maximize the quality of the resulting map. Across virtually all of the experiments, the best performance is generally found in the lower right corner of the parameter sweep. This area corresponds to null-hypotheses with relatively large weights and low information (equivalently large covariances). However, it is also evident that the region of good performance (which we subjectively appraise to be mean squared errors less than about -1) is quite large in almost all cases. From these results, we conclude that the method is robust across many orders of magnitude of w and s, and that in general, w should be made relatively close to one and s relatively close to zero.

5. 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 run time performance of our algorithm is similar to that of existing state-of-the-art maximum likelihood systems. In comparison to other robust formulations, including those based on switching constraints, the ability of our method to encode "one-of-k" mixtures provides a significant performance advantage. Further, while we have explored the case of batch solvers, our method can be equally well adapted to incremental systems (Kaess et al., 2008). An open-source implementation can be downloaded from Agarwal et al. (2012)

Funding

This research was partially supported by the U.S. Department of the Air Force, grant FA2386- 11-1-4024, BMBF contract number 13EZ1129B-iView and European Commission under contract number ERC-267686-LifeNav.

Notes

1. We report MSEs based on translational error, i.e. MSE_{xy} for three-degree-of-freedom and MSE_{xyz} for six-degree-of-freedom problems.
2. 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.

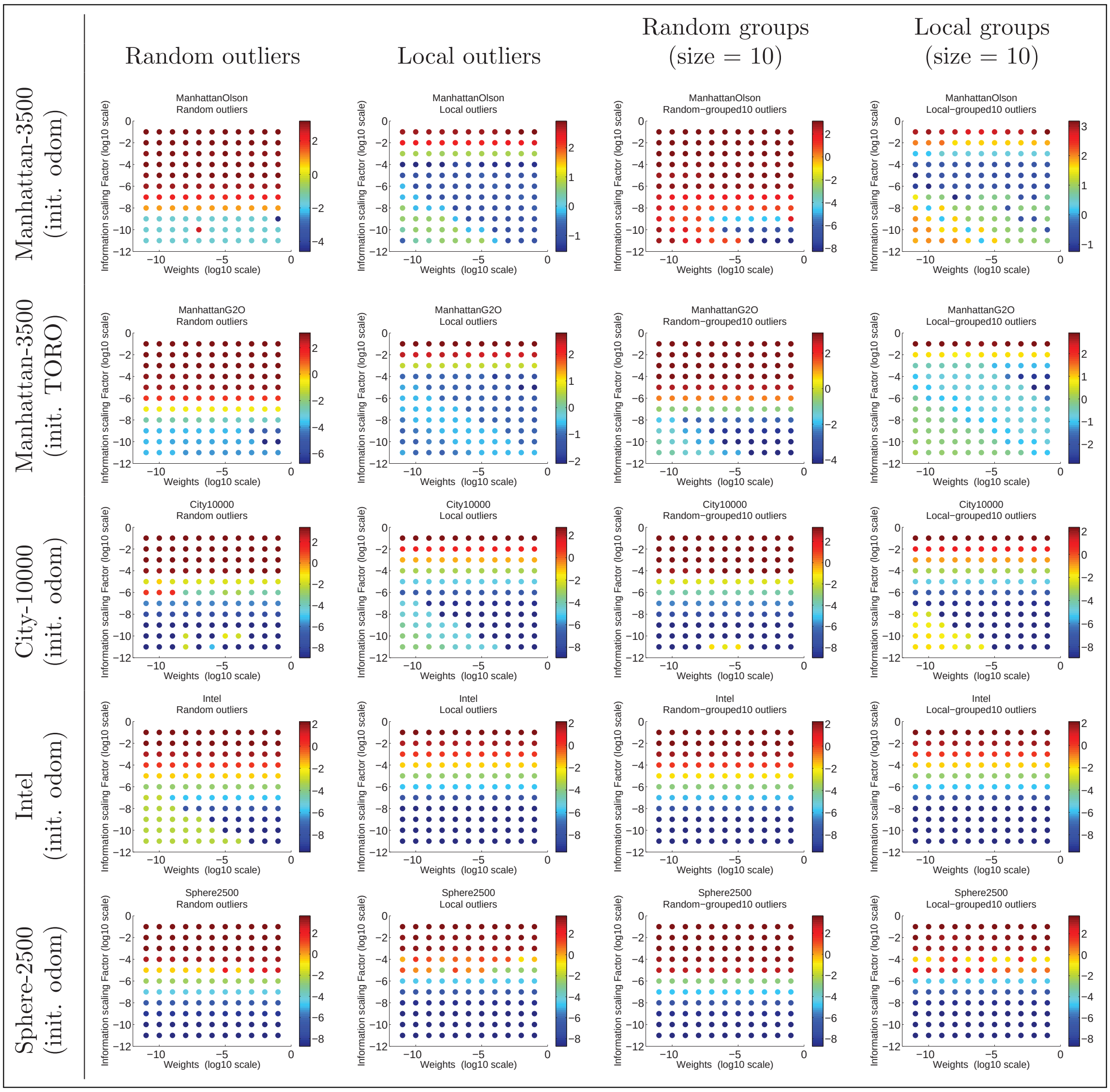

Fig. 11. Robustness of method over a range of parameters. We consider five different data set + initial conditions (rows), four data association error generation methods (columns), and a parameter sweep over *w* and *s* within each grid cell. Colors correspond to the log of the MSE; maps less than −0.1 are relatively good and maps less than −1 are excellent. These plots show that the basin of convergence for the max-mixture method is quite large. A total of 1,000 outliers of each error type was added for each data set. For the grouped errors it resulted in 100 groups, each with 10 mutually consistent outliers.

References

Agarwal P and Olson E (2012) Variable reordering strategies for slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Agarwal P, Olson E and Burgard W (2012) Max-mixture - open source implementation with g2o. http://openslam.org/maxmixture.html

Bailey T (2002) *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. Ph.D. thesis, Australian Centre for Field Robotics, University of Sydney.

Bailey T, Nieto J and Nebot E (2006) Consistency of the FastSLAM algorithm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 424–429.

Blackman SS (2004) Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine* 19: 5–18.

Bosse M, Newman P, Leonard J and Teller S (2004) Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research* 23: 1113–1139.

Crossman J, Marinier R and Olson E (2012) A hands-off, multi-robot display for communicating situation awareness to operators. In *Proceedings of the International Conference on Collaboration Technologies and Systems*, pp. 109–116.

Dellaert F (2005) Square root SAM. In: *Proceedings of Robotics: Science and Systems (RSS)*. Cambridge, MA.

Dellaert F and Kaess M (2006) Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research* 25: 1181–1203.

Durrant-Whyte H, Majumder S, Thrun S, de Battista M and Scheding S (2003) A Bayesian algorithm for simultaneous localisation and map building. In Jarvis R and Zelinsky A (eds), *Robotics Research* (*Springer Tracts in Advanced Robotics*, vol. 6). Berlin: Springer, pp. 49–60.

Eustice R, Singh H, Leonard J and Walter M (2006) Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *The International Journal of Robotics Research* 25: 1223–1242.

Frese U (2005) A proof for the approximate sparsity of SLAM information matrices. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, pp. 331–337.

Grisetti G, Stachniss C and Burgard W (2005) 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, pp. 2432–2437.

Grisetti G, Stachniss C, Grzonka S and Burgard W (2007) A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In: *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA.

Hähnel D, Burgard W, Fox D and Thrun S (2003) A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 206–211.

Hartley R and Zisserman A (2004) *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge: Cambridge University Press.

Howard A and Roy N (2003) The robotics data set repository (radish). http://radish.sourceforge.net/.

Kaess M, Ranganathan A and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transations on Robotics* 24: 1365–1378.

Konolige K (2010) Sparse sparse bundle adjustment. In: *British Machine Vision Conference*, Aberystwyth, Wales.

Kummerle R, Grisetti G, Strasdat H, Konolige K and Burgard W (2011) g2o: A general framework for graph optimization. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai.

Kwak N, Kim I-K, Lee H-C and Lee BH (2007) Analysis of resampling process for the particle depletion problem in fastSLAM. In: *IEEE International Symposium on Robots and Human Interactive Communications (RO-MAN)*, pp. 200–205.

Latif Y, Cadena C and Neira J (2012a) Realizing, reversing, recovering: incremental robust loop closing over time using the IRRR algorithm. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4217.

Latif Y, Lerma CC and Neira J (2012b) Robust loop closing over time. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia.

Lerner U and Parr R (2001) Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*. San Francisco, CA: Morgan Kaufmann, pp. 310–331.

Montemerlo M (2003) *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Neira J and Tardos JD (2001) Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation* 17: 890–897.

Newman PM (1999) *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. Ph.D. thesis, University of Sydney, Australia.

Olson E (2008) *Robust and Efficient Robotic Mapping*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Olson E (2009a) Real-time correlative scan matching. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan.

Olson E (2009b) Recognizing places using spectrally clustered local matches. In: *Robotics and Autonomous Systems*.

Olson E (2011) Evaluating back-ends: metrics. In *Automated SLAM Evaluation Workshop, Robotics Science and Systems*, Los Angeles, CA.

Olson E, Strom J, Morton R, et al. (2012) Progress towards multi-robot reconnaissance and the MAGIC 2010 competition. *Journal of Field Robotics* 29: 762–792.

Pfingsthorn M and Birk A (2012) Simultaneous localization and mapping (SLAM) with multimodal probability distributions. *International Journal of Robotics Research* 32: 143–171.

Ranganathan P, Morton R, Richardson A, et al. (2010) Coordinating a team of robots for urban reconnaisance. In: *Proceedings of the Land Warfare Conference (LWC)*.

Sibley G, Mei C, Reid I and Newman P (2009) Adaptive relative bundle adjustment. In: *Robotics Science and Systems Conference*, Seattle, WA.

Smith R, Self M and Cheeseman P (1988) A stochastic map for uncertain spatial relationships. In: Faugeras O and Giralt G (eds), *Proceedings of the International Symposium of Robotics Research (ISRR)*, pp. 467–474.

Stachniss C, Grisetti G and Burgard W (2005) 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, pp. 667–672.

Strasdat H, Montiel J and Davison A (2010) Scale drift-aware large scale monocular SLAM. Robotics: Science and Systems 2(3): 5.

Sünderhauf N (2012) Vertigo: versatile extensions for robust inference using graphical models. http://http://www.tu-chemnitz.de/etit/proaut/forschung/robustSLAM.html.en

Sunderhauf, N. and Protzel, P. (2012). Switchable constraints for robust pose graph slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Thrun S and Liu Y (2003) Multi-robot SLAM with sparse extended information filters. In: *Proceedings of the International Symposium of Robotics Research (ISRR)*, Sienna, Italy.

Walter M, Eustice R and Leonard J (2005) A provably consistent method for imposing exact sparsity in feature-based SLAM information filters. In: Thrun S, Brooks R and Durrant-Whyte H (eds.), *Proceedings of the International Symposium of Robotics Research (ISRR)*, San Francisco, CA. Berlin: Springer, pp. 214–234.