2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Representing and Solving Local and Global Ambiguities as Multimodal and Hyperedge Constraints in a Generalized Graph SLAM Framework

Max Pfingsthorn and Andreas Birk

***Abstract*— Graph-based Simultaneous Localization and Mapping (SLAM) has experienced a recent surge towards robust methods. These methods take the combinatorial aspect of data association into account by allowing decisions of the graph topology to be made during optimization. In this paper, the Generalized Graph SLAM framework for SLAM under ambiguous data association is presented, and a formal description of using hyperedges to encode uncertain loop closures is given for the first time. The framework combines both hyperedges and multimodal Mixture of Gaussian constraints to cover all sources of ambiguity in SLAM. An extension of the authors' multimodal *Prefilter* method is developed to find good initial conditions in such a generalized multimodal hypergraph. Experiments on a real world 3D dataset show that the novel multimodal hypergraph *Prefilter* method is both significantly more robust and faster than other robust state-of-the-art methods.**

I. Introduction

Today's robotics research pushes the envelope on where solutions using robots and intelligent systems can be applied. When robots face more complex, unstructured, and dynamic environments while expanding their workspace, the problem of Simultaneous Localization and Mapping (SLAM) becomes even more relevant and significantly harder at the same time. There is a clear and present need for efficient and most of all robust SLAM methods that are able to generate a useful map, even under erroneous data association decisions at any level in the SLAM process.

Graph-based SLAM has been the method of choice in the latest literature on SLAM in dynamic environments [1], portable SLAM systems for humans [2], SLAM with micro aerial vehicles (MAV) [3], [4], as well as underwater SLAM [5]. All of these use cases can benefit significantly from an improved robustness of graph optimization methods for SLAM. For those reasons, robust graph optimization or inference for graph-based SLAM has become a strong research focus very recently [6], [7], [8], [9], [10]. These methods fall into roughly two categories:

In one ([11], [9], [10], [8] and partially [7]), incongruent graph constraints of the traditional unimodal kind are simply discounted during optimization. This category is roughly comparable to iteratively reweighted least squares [12] or least trimmed squares [13], both traditional robust regression techniques.

The other ([6] and partially [7]) allows multiple components per constraint, either as a multimodal Mixture of Gaussians (MoG) [6], or a so-called multimodal *Max-Mixture* [7].

The authors are with the School of Electrical Engineering and Computer Science, Jacobs University, Bremen, Germany. <m.pfingsthorn, a.birk>@jacobs-university.de

This paper presents a general framework for SLAM under ambiguous data associations, called Generalized Graph SLAM. Its description contains the first formal introduction of how uncertain loop closures can be modeled using hyperedges. This is extension is combined with the authors' previous work on multimodal constraints [6].

The hyperedge components allow to cope with global ambiguities, i.e. to represent multiple alternative hypotheses about possible loop closures. The multimodal components in contrast deal with local ambiguities, i.e. they handle different alternative hypotheses about the motion a robot/sensor may have undergone between two subsequent poses. This new framework of Generalized Graph SLAM can hence handle local as well as global ambiguities in a single coherent manner. Furthermore, it is shown that other robust methods found in the literature are conceptually special cases within our Generalized Graph SLAM framework.

To solve the challenges put forth in the Generalized Graph SLAM framework, a novel extension of the authors' *Prefilter* method [6] to generate good initial conditions for selecting globally consistent constraints from such multimodal hypergraphs is presented. Experiments show that the extended *Prefilter* method is both significantly more robust and faster than other robust state-of-the-art methods.

II. Related Work

The robust SLAM methods in [6], [7], [14], [11], [15], [8], [9], [10] improve upon traditional Graph SLAM methods by providing ways to filter out or discount incongruent graph constraints during optimization.

While not specified exactly in the paper, the $g2o$ graph optimization library by Kümmerle *et al.* [16] chooses a traditional robust optimization approach by applying an iteratively reweighted least squares (IRLS) method [12]. Their approach allows weighting the individual terms of the cost function by the computed residuals and reducing the influence of large residuals. Multiple of these robust kernels are implemented, e.g. the Huber or the Cauchy kernel [12]. Thus, incongruent constraints are weighted less, which allows the method to converge to a reasonable result when outliers are present. The latest $g2o$ version[1] was used in the experiments below.

Sünderhauf and Protzel [9], [10] choose a more explicit reweighting scheme were the weight of a cost function term is controlled as part of the state vector during optimization. Instead of using the value of the local residual to scale the impact of it in the total cost function, switching variables are

[1] from https://github.com/RainerKuemmerle/g2o

introduced that are explicitly part of the optimization process. Either a sigmoid function [9] or a linear function [10] is used to weigh individual cost function terms. For the experiments below, an implementation of this method in the $g2o$ library published as open source by Sünderhauf and Protzel is used[2]. This implementation uses the more recently proposed linear switch function [10].

Sünderhauf and Protzel's [9], [10] method was developed into a robust kernel function by Agarwal *et al.* [11] and implemented in $g2o$. Agarwal *et al.* derive an optimal value for the switch variable used by Sünderhauf and Protzel for a specific residual value, thus making an explicit switch variable unnecessary. This method, called *Dynamic Covariance Scaling (DCS)* is generally faster than Sünderhauf and Protzel's method because of the reduced number of variables during optimization.

In the authors' previous work [6], a novel multimodal extension of the traditionally unimodal constraints used in graph-based SLAM was introduced. Standard methods assume that a constraint in the graph is inherently correct and just uncertain due to noise; it can hence be represented by a single underlying normal distribution. [6] uses a multimodal Mixture of Gaussians (MoG) instead that allows multiple mutually exclusive options (modes) in each edge. There, the authors also introduced the concept of local ambiguity vs. global ambiguity, where the presented approach using MoGs is used to solve the local ambiguity problem, i.e. the handling of different motion alternatives between two subsequent robot/sensor poses. Several methods are outlined in [6] to solve SLAM with locally ambiguous registration results that are expressed in multimodal MoGs, including a global graph initialization method called *Prefilter* which is shown to be very robust. It is used to discard incongruent components in the MoG constraints before optimization, which is related to least trimmed squares [13].

Olson and Agarwal [7] take a similar approach by allowing multiple single normal distributions in one constraint in the graph. However, in their method, the decision which of the constraint distributions should be used is reevaluated greedily in each step of the iteration. Instead of using a weighted sum of normal distributions as is the case with MoGs, their approach only uses the component which contributes the maximum probability to the current estimate, thus the name *Max-Mixture*. *Max-Mixtures* have the disadvantage that they are chosen greedily given the current estimate, requiring good initial conditions to allow convergence. Olson and Agarwal [7] also describe the idea that multiple loop closing constraints may be combined into one using a multimodal *Max-Mixture*, effectively representing a hyperedge but not explicitly using the name nor describing the idea in any formal way. Again, the implementation for the $g2o$ library made available by the authors Olson and Agarwal is used in the experiments below[3].

Latif *et al.* [15], [8] present a method called *RRR* to generate clusters of temporally close loop-closing constraints and check these for spatial consistency. The method makes the rather significant assumption that sequential constraints are generated using odometry and are always without outliers. A traditional $\mathcal{X}^2$ error metric is used to identify outliers in each cluster of loop-closing constraints. By explicitly making a binary decision about the inclusion or rejection of individual constraints, the method is highly related to the least trimmed squares method. By using a spatial consistency measure to select outliers, the method is also related to the *Prefilter* method above. The open source implementation for the $g2o$ library by *Latif et al.*[4] only supports 2D graphs, and was not evaluated in the experiments described below.

The idea of using hyperedges to represent global ambiguity (i.e. ambiguous loop detection results) is introduced in this paper, and this idea is combined with the use of multimodal edges to represent local ambiguities in the Generalized Graph SLAM framework. An extension of the multimodal *Prefilter* method that can simultaneously process multimodal MoG constraints as well as ambiguous loop closure constraints in the form of hyperedges is subsequently presented. The following section validates the presented method in comparison to the methods described above with experimental results using a real world dataset. The last section concludes the paper.

III. Generalized Graph SLAM

A. Traditional Graph-Based SLAM

Formally, a pose graph is an undirected graph $G = (V, E)$ consisting of vertices V and edges E. The vertices $v_i \in V$ denote poses where the robot obtained sensor observations z_i. A pose estimate x_i is also associated with the vertex and thus is a tuple $v_i = (x_i, z_i)$. In addition to the vertices it connects, each edge $e_k \in E$ contains a constraint c_k on the pose estimates of the associated vertices, thus $e_k = (v_i, v_j, c_k)$. While the graph itself is undirected, the edge has to declare a sort of *observation direction*, the direction in which the constraint was generated, often also called the reference frame of the constraint. In case the edge is traversed in reverse *observation direction*, the constraint c must be inverted. What exactly that entails is up to the representation of the constraint.

From the formal description in [6], the joint probability of the pose graph G is

$$p(x_{1:t}|G) = \prod_{(v_i,v_j,c_k)\in E} p(x_j \ominus x_i|c_k) \tag{1}$$

where $p(x_j \ominus x_i|c_k)$ is the specific probability distribution of the constraint c_k on edge $e_k \in E$. $\ominus$ is the relative pose operator.

Usually, this is a normal distribution, so

$$p(dx|c_k) = \frac{1}{|2\pi\Sigma_k|^{1/2}} e^{-\frac{1}{2}(dx\ominus\mu_k)^T\Sigma_k^{-1}(dx\ominus\mu_k)} \tag{2}$$

[2]from `http://openslam.org`

[3]from `https://github.com/agpratik/max-mixture`

[4]from `https://github.com/ylatif/rrr`

This results in a very convenient negative log probability formulation that can be directly fed into a general non-linear least squares solver.

$$-\ln p(x_{1:t}|G) \approx \sum_{(v_i,v_j,c_k)\in E} (t_i^j \ominus \mu_k)^T \Sigma_k^{-1} (t_i^j \ominus \mu_k) \quad (3)$$

where $t_i^j = x_j \ominus x_i$.

B. Local Ambiguity vs. Global Ambiguity

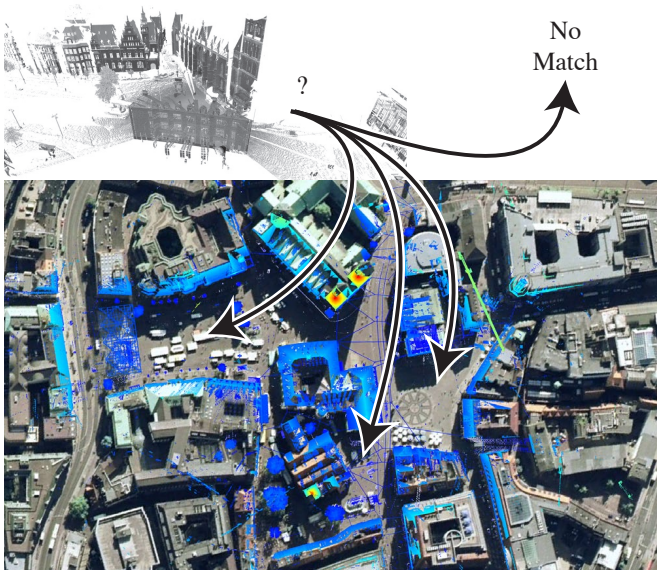


Fig. 1. A visualization of global ambiguity, corresponding to a general loop detection problem. The scan (left) either corresponds to a previously visited location (arrows going towards the map), or represents a newly visited location (arrow leading to "No Match"). In the Generalized Graph SLAM framework, all correspondences relating to previous locations are collected as hypercomponents with corresponding weights, and the option "No Match" is expressed as the null hypothesis in the hyperedge.

The major source of errors in graph-based SLAM is faulty data association. Specifically, two types of data association errors are identified: a) Errors in identifying common data in two consecutive sensor observations (local ambiguity) and b) errors identifying common data in temporally distant sensor observations (global ambiguity). This section describes each of these error sources in detail, laying the foundation for the Generalized Graph SLAM framework.

Specifically, the term ambiguity is used for the situation where a clear global optimum of the respective registration cost function in the local case or the data association metric in the global case can not be found, but multiple local optima are present instead.

Local ambiguity corresponds to the case where two consecutive sensor observations exhibit multiple possible registration results, see [6]. A registration result of such two scans is called locally ambiguous if these ambiguities can not be resolved using only information present in the observations itself. In other words, the registration cost function has multiple optima, resulting in a multimodal Mixture of Gaussian (MoG) probability distribution for the corresponding constraint.

$$p(x_i|x_{i-1}) = \sum_{m=1}^{M_k} \pi_m \mathcal{N}(x_i \ominus x_{i-1}|\mu_m, \Sigma_m) \quad (4)$$

with $\sum \pi_m = 1$. Each mean μ_m corresponds to an optimum in the registration cost function, Σ_m corresponds to the inverse of the hessian at that point, and the weight π_m should be proportional to the value of the registration cost function at μ_m.

Global ambiguity corresponds to the case of uncertain loop closures, where two temporally distant sensor observations may or may not show the same section of the environment. See figure 1 for an example. Formally, there exists a probability mass function (PMF) which is defined over all previously inserted vertices in the graph, and a null hypothesis in case the current vertex is completely new. This PMF can be represented as the weights π_m of a more generalized mixture over all previous poses and an uninformative uniform distribution representing the null hypothesis.

$$p(x_i|x_{1:i-1}) = \pi_0 \mathcal{U}(\mathbb{R}^d) + \sum_{j=1}^{i-1} \pi_j p(x_j \ominus x_i|c_j) \quad (5)$$

with $\sum_0^{i-1} \pi_j = 1$, and where $x_{1:i-1}$ are all poses from vertex v_1 to v_{i-1}, $p(dx|c)$ is any probability distribution representing a registration result, π_0 is the weight of the null hypothesis, and $\mathcal{U}(\mathbb{R}^d)$ is the uniform distribution over all real numbers of the same degree of freedom d as the poses.

Note that local ambiguity may occur also in the registration result referenced in the global ambiguity case. Thus they describe orthogonal problems, both or either may or may not occur in any given SLAM problem.

This means that solutions to both are required, though in the past both have been neglected in favor of a simple traditional unimodal SLAM model. However, many limitations of traditional unimodal SLAM methods that do not use explicit modeling of both sources of faulty data association have been noted, most of which had been attempted to solve by more complex and involved SLAM front ends. These front ends would filter out outliers in both the local and global case and only present verified registration results (local disambiguation) and loop constraints (global disambiguation) to the SLAM backend for optimization. In situations where either local or global ambiguity occurred, these methods would ideally reject all potentially ambiguous results, thus eliminating useful information. In the worst case, these methods would pass on outliers to the optimization backend, resulting in divergence.

C. Modeling Uncertain Loop Constraints as Hyperedges

Modeling the underlying probability mass function from equation 5 exhaustively for all previous poses is wasteful. Most of the weights π_j will be zero or very close to zero. Instead, a more compact representation is needed to exploit this *sparse* connectedness to older vertices.

Graph theory presents a fitting concept in this case, namely a hyperedge. Formally, a hyperedge is a set of vertices that are connected. Thus, instead of every edge $e \in E$ consisting of exactly two endpoints v_i, v_j and the associated constraint c as described in section III-A, a hyperedge in a pose graph is defined as a tuple $e = (v_i, N, \{v_j\}, \{\pi_j\}, \{c_j\})$. The weight of the null hypothesis is implicitly given by $\pi_0 = 1 - \sum \pi_j$, with $\sum \pi_j \leq 1$.

Note that due to the geometric interpretation of the edge, an *observation direction* is still necessary, so v_i is defined as the *reference* of the hyperedge and the base frame of the relative poses represented in the constraints. $\{v_j\}$ is the set of vertices the reference vertex v_i is connected to by this edge, and $N = |\{v_j\}|$. Note that the case where $N = 1$, i.e. there is no ambiguity which older vertex v_j the reference vertex v_i should be connected to, is explicitly covered as well, while still allowing for discounting of this one constraint by reducing the weight π_1.

For the general case, equation 5 becomes

$$p(x_i|G) = \prod_{e \in E} \left[(1 - \sum_{j=1}^{N} \pi_j) \mathcal{U}(\mathbb{R}^d) + \sum_{j=1}^{N} \pi_j p(x_j \ominus x_i | c_j) \right] \tag{6}$$

$$\approx \prod_{e \in E} \sum_{j=1}^{N} \pi_j p(x_j \ominus x_i | c_j) \tag{7}$$

Since $\mathcal{U}(\mathbb{R}^d)$ is practically zero everywhere (technically $\frac{1}{\infty}$), the term corresponding to the null hypothesis is dropped. This means that the expression looks exactly like a regular mixture, with the difference that $\sum_{j=1}^{N} \pi_j \leq 1$ instead of $\sum_{j=1}^{N} \pi_j = 1$.

In the following, each $p(x_j \ominus x_i | c_j)$ is called a *hypercomponent* to distinguish between components in hyperedge and MoG constraint mixtures. π_j will be referred to as the hypercomponent weight.

Without loss of generality, the Generalized Graph SLAM framework assumes that all edges in a generalized pose graph are hyperedges and all the constraints c_j of each edge are multimodal MoG constraints. Here, the cases with $N = 1$ (i.e. no global ambiguity) and $M_k = 1$ (i.e. no local ambiguity) are explicitly included. For this generalized graph, the joint probability then becomes (extended from eq. 14 in [6])

$$p(x_{1:t}|G) = \prod_{e \in E} \sum_{j=1}^{N} \pi_j \sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \tag{8}$$
$$\text{with } e = (v_i, N, \{v_j\}, \{\pi_j\}, \{c_j\})$$
$$\text{and } t_i^j = x_j \ominus x_i$$

Similarly for the joint log probability

$$\ln p(x_{1:t}|G) = \sum_{e \in E} \ln \left[\sum_{j=1}^{N} \pi_j \sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \right] \tag{9}$$

An equivalent formulation moves the hypercomponent weights into the MoG sum, allowing the same vertex multiple times in the set $\{v_j\}$:

$$p(x_{1:t}|G) = \prod_{e \in E} \sum_{l=1}^{L} \pi_l p(t_i^j | \mu_l, \Sigma_l) \tag{10}$$
$$\text{with } e = (v_i, L, \{v_j\}, \{\pi_l\}, \{(\mu_l, \Sigma_l)\})$$

where $L = \sum M_k$ and $\pi_l = \pi_j \pi_m$ for the l-th hypercomponent/MoG component combination. Again, $\pi_0 = 1 - \sum \pi_l$. This formulation, though with $\sum \pi_l = 1$, is implicitly used in Olson's *Max-Mixture* method [7], though not explicitly described. However, eq. 8 is conceptually clearer as it presents a clear separation of global and local ambiguity. Furthermore, there is of course the main challenge not only to represent local and global ambiguities but to find a robust and efficient optimization method for them, which may require separate models for each.

There are now some considerations to be made for computing the natural logarithm of the double sum of weighted Gaussians. In the special case of a simple unimodal edge, where $N = 1$, $\pi_1 = 1$, and $M_k = 1$,

$$\begin{aligned} &\ln \left[\sum_{j=1}^{N} \pi_j \sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \right] \\ &= \ln p(t_i^j | \mu_m, \Sigma_m) \\ &= -\frac{1}{2} \ln \left(|2\pi \Sigma_1| \right) - \frac{1}{2} (t_i^j \ominus \mu_1)^T \Sigma_1^{-1} (t_i^j \ominus \mu_1) \end{aligned}$$

In the following, such an edge will be referred to as *simple*.

In the special case of a purely multimodal edge, where $N = 1$, $\pi_1 = 1$,

$$\begin{aligned} &\ln \left[\sum_{j=1}^{N} \pi_j \sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \right] \\ &= \ln \left[\sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \right] \end{aligned}$$

In the previous two cases, a $\pi_1 < 1$ will result in a simple addition by $\ln \pi_1$ in the log-probability.

In the special case of a pure hyperedge with unimodal subcomponents, where $M_k = 1$,

$$\begin{aligned} &\ln \left[\sum_{j=1}^{N} \pi_j \sum_{m=1}^{M_k} \pi_m p(t_i^j | \mu_m, \Sigma_m) \right] \\ &= \ln \left[\sum_{j=1}^{N} \pi_j p(t_i^j | \mu_m, \Sigma_m) \right] \end{aligned}$$

Additionally, there are a number of methods described in recent literature that can be treated as special cases of the Generalized Graph SLAM framework. The special case where $N = 1$ and c_1 is a unimodal Gaussian constraint (i.e. $M_k = 1$) corresponds to the work done by Sünderhauf and Protzel [9]. In this case, $\pi_1 = \omega_{ij} = sig(s_{ij})$, where s_{ij} is the switch variable between vertices v_i and v_j (see eq. 7

in [9]), or $\pi_1 = \Psi(s_{ij}) = s_{ij}$ for the linear case (cf. eq. 1 in [10]). Similarly, the *RRR* algorithm of Latif *et al.* [15], [8] makes a strictly binary decision where Sünderhauf and Protzel make a fuzzy one, and thus can be modeled the same way in this framework, i.e. $\pi_1 \in 0, 1$.

The special case where the weights π_j and π_m are adjusted at every iteration such that only one π_j and π_m retain their original value, i.e.

$$(j^*, m^*) = \underset{j,m}{\operatorname{argmax}} \pi_j \pi_m p(dx|\mu_m, \Sigma_m) \tag{11}$$

$$\pi_j = \begin{cases} \pi_j & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\pi_m = \begin{cases} \pi_m & \text{if } m = m^* \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

corresponds to the *Max-Mixture* method (see eq. 4 in [7]). Olson and Agarwal aggregate the two conceptually separate weights π_j and π_m into one in their discussion, as in the equivalent formulation presented in eq. 10. As such, they do not offer an implicit null hypothesis choice, but the mixture has to explicitly include a normal distribution defined to be the null hypothesis, which usually has a very large covariance.

The *Prefilter* method [6], and its extension to this Generalized Graph SLAM framework described in the next section, is used to make a similar selection of weights as *Max-Mixture* does. Weights are set such that exactly one π_j and one π_m per edge is one, indicating the component that explains the estimate generated by the *Prefilter* method best:

$$(j^*, m^*) = \underset{j,m}{\operatorname{argmax}} \pi_j \pi_m p(dx|\mu_m, \Sigma_m) \tag{14}$$

$$\pi_j = \begin{cases} 1 & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

$$\pi_m = \begin{cases} 1 & \text{if } m = m^* \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

However it is done once before the optimization starts and this decision is not changed during optimization, allowing the use of standard optimization methods.

D. Good Initial Conditions on Multimodal Hypergraphs

The extension of the original *Prefilter* algorithm [6] to also handle hyperedges in addition to multimodal ones is rather straight forward.

Algorithm 1 shows the pseudocode for the *Prefilter* method extended to hypergraphs. The main difference to the original *Prefilter* is that through choosing the hypercomponent to follow, the underlying graph topology for each sample changes. Therefore, the state of the whole minimum spanning tree traversal has to be kept associated with the corresponding pose sample in a list of traversal states $\mathbf{T}$. For simplicity, each MoG component also gives rise to a new traversal state instance, even though they do not change the graph topology and some data is duplicated. This simplification also allows straightforward parallelization of the algorithm for large values of N and complex graphs.

Algorithm 1: The *Prefilter* algorithm.

Input: MoG Hyper PoseGraph G, maximum number of hypotheses N
Output: $\mathbf{X}$: a set of N sets of vertex poses $X = \{x_i\}$

initialize an empty list $\mathbf{T}$ of traversal states;
let t be a traversal state;
$t.\mathbf{X} = \{x_1\}$;
$t.V_{used} = \{v_1\}$;
$t.E_{used} = \emptyset$;
initialize priority queue $t.P$ to sort by $edgeweight(e)$;
for *all adjacent edges* e *of* v_1 **do**
    enqueue($t.P$, (v_1, e));
    $t.E_{used} = t.E_{used} \bigcup \{e\}$;
end
append t to $\mathbf{T}$;
while $\exists t \in \mathbf{T} : t.P$ *not empty* **do**
    for $\forall t \in \mathbf{T} : t.P$ *not empty* **do**
        (v, e) = dequeue($t.P$);
        if $v = e.v_{start}$ **then**
            for *every hyperedge component* j **do**
                $ExpandMultimodal(\mathbf{T}, t, v, v_j, c_j)$;
            end
        else
            let j be the hyperedge component of e where $v_j = v$;
            $ExpandMultimodal(\mathbf{T}, t, v_j, v, invert(c_j))$;
        end
        if $\sum_{j=1}^{N} e.\pi_j = 1$ **then**
            remove t from $\mathbf{T}$;
        end
    end
    if $|\mathbf{T}| > N$ **then**
        sort $\mathbf{T}$ by joint probability of assigned vertex poses $\mathbf{X}$ of each element;
        truncate $\mathbf{T}$ to contain only the N most probable elements;
    end
end
$\mathbf{X} = \bigcup_{t \in \mathbf{T}} t.\mathbf{X}$;

Algorithm 2: $edgeweight(e)$

Input: MoG Hyper PoseGraph edge $e \in E$
Output: computed edge weight ω

$\omega = 0$;
for *all constraints* c_j *in* e **do**
    $\omega = \omega + M_k$;
end
if $\sum_{j=1}^{N} e.\pi_j = 1$ **then**
    $\omega = \omega + 1$;
end
return ω;

However, the implementation used in the experiments is single threaded to allow a fair comparison.

Algorithm 3: $ExpandMultimodal(\mathbf{T}, t, v, v_{next}, c)$

Input: List of traversal states $\mathbf{T}$, current traversal state t, current vertex v, next vertex v_{next}, multimodal constraint c

Output: Modified list of traversal states $\mathbf{T}$

for *every multimodal component* m *in* c **do**
  make a new traversal state t_{new} as a copy of t;
  x = pose of v in $t.\mathbf{X}$;
  $t_{new}.\mathbf{X} = t_{new}.\mathbf{X} \bigcup (x \oplus \mu_m)$
  for *every edge* e_{adj} *adjacent to* v_{next} *that is not in* $t_{new}.E_{used}$ **do**
    enqueue($t_{new}.P$, (v_{next},e_{adj}));
    $t_{new}.E_{used} = t_{new}.E_{used} \bigcup \{e_{adj}\}$;
  end
  append t_{new} to $\mathbf{T}$;
end

Note that the null hypothesis is never directly referenced in the algorithm. By design of the algorithm, keeping an unmodified version of the current traversal state t in the list $\mathbf{T}$ corresponds to the case where the current edge e is not used, i.e. where the null hypothesis is chosen. This works since edges are marked as used when they are enqueued in the priority queues, and dequeueing an edge from t without using it effectively deletes it from the graph topology for t. Furthermore, calling $ExpandMultimodal(\dots)$ does not change the passed current traversal state, only new traversal states are generated corresponding to all modes. This means that by line 23 in algorithm 1, the current traversal state t is unchanged, and thus corresponding to the null hypothesis. Line 23 checks if the null hypothesis is inadmissible by checking its weight, and if it is not, removes t from $\mathbf{T}$, thus not following the null hypothesis.

The final set of sorted vertex poses $\mathbf{X}$ can be used to select not only components from a MoG, as described in [6], but also hyperedge components in the same way.

IV. Experiments and Results

This real world data is based on 13 scans that were recorded with a Riegl VZ-400 in the center of Bremen, Germany. Each point cloud consists of between 15 and 20 million points with reflectance information. The scanner was mounted on a tripod without a mobile base, thus no odometry information is available. However, markers were placed in the environment beforehand to allow for a comparison with the "gold standard" for geodetic applications, i.e. registration with artificial markers in the Riegl software which requires additional manual assistance in the process like confirming or re-selecting correspondences. This registration based on artificial markers can also be used to seed methods that need a good initial guess, e.g. for ICP based methods like 6D-SLAM [17]. Note that no initial guess, i.e. no initial marker based registration, no motion estimates, no GPS, or anything similar is used in the results presented here, other than as a comparison.

#	1	2	3	4	5	6	7	8	9	10	11	12
0	3	4	3	2	5	3	4	1	1	1	1	1
1	-	1	1	2	1		1	1	2	1	1	1
2		-	1	1	1		1	1	2	1	1	4
3			-	1	1		2		2	2	1	2
4				-	1	3	4	1	2	1	1	1
5					-	1	7	4		2	2	1
6						-	1	9	1		1	1
7							-	1	2	2	4	2
8								-	1	1	2	3
9									-	1	1	1
10										-	1	1
11											-	1

TABLE I

Connectivity matrix between all 13 scans, showing the number of components in the multimodal registration result per pair. A missing number in the upper triangle means that no registration result was found for that pair.

This dataset has been used for multimodal SLAM before, namely in the authors' original paper presenting the *Prefilter* method [6]. The same multimodal plane matching method is used here, though with a slight change. Namely, the absolute minimum overlap parameters $\mathcal{O}_p$ was again lowered to 0.045, allowing even more potential results to be considered.

In the experiments performed in [6], loop closures were added to the map only after validation that one of the reported results actually was the correct one. Here, all scans are exhaustively matched with each other without manual validation, resulting in a very dense graph. Two cases of registration results were treated slightly differently: Registrations between two sequential scans were added to the graph as a regular edge, i.e. not a hyperedge. Registrations between one scan an all its preceding scans without its immediate predecessor were added as a single hyperedge. Any registration result was allowed to be multimodal, also in the sequential case.

Table I shows a connectivity matrix between all pairs. Note that the graph is almost completely connected because of the exhaustive loop generation fashion. However, there are only 23 edges in the graph. These are the 12 sequential MoG edges, and 11 non-sequential MoG hyperedges. For example, the loop closing MoG hyperedge from scan 7 to its predecessors 0 through 5 contains 6 hypercomponents with a total of 19 MoG components. This results in a graph complexity $C(G) = 36.95$, which is large regarding the small size of the graph.

Table II shows the final SSE distances relative to the marker-based ground truth for all tested optimization methods. The implementation of *RRR* by Latif *et al.* [8] currently does not support 3D pose graphs, thus it was not evaluated for this dataset. All methods ran for 150 iterations, using the Gauss-Newton solver in *g2o*. The *Max*, *Max-Mixture*, and *Prefilter* methods used a Cauchy robust kernel implemented in *g2o* with a kernel size of 10. *Switchable Constraints* did not use a robust kernel as the switch variables served the same purpose, and *Dynamic Covariance Scaling* naturally used the DCS kernel in *g2o* with a kernel size (Φ) of 1, as

	Max-Mixture	*SC*	*DCS*
SSE_{xyz}	$4.516 \cdot 10^9$	$2.829 \cdot 10^{37}$	$4.215 \cdot 10^9$
$SSE_{\psi\phi\theta}$	2.501	5.663	2.399
runtime (s)	0.13	0.051	0.03

	Max	*Prefilter*	
SSE_{xyz}	$5.012 \cdot 10^9$	$7.569 \cdot 10^4$	
$SSE_{\psi\phi\theta}$	2.359	0.00006	
runtime (s)	0.009	0.009	

TABLE II

SSE DISTANCE RELATIVE TO THE "GOLD STANDARD" MARKER-BASED REGISTRATION FOR EACH OPTIMIZATION METHOD. *SC* STANDS FOR *Switchable Constraints*, *DCS* STANDS FOR *Dynamic Covariance Scaling*.

recommended in [11].

Switchable Constraints diverged to a result far away from the ground truth, too far to visualize. Interestingly, the result does not improve significantly between the *Max* and *Max-Mixture* methods, even though *Max-Mixture* takes significantly longer. The *Dynamic Covariance Scaling* method performs slightly better than *Max-Mixture*.

Clearly, the *Prefilter* method outperforms all others, both in the quality of the optimization result and efficiency. Note that the mean square errors SSE in the table are in mm^2, so the final SSE_{xyz} of $7.569 \cdot 10^4$ corresponds to a mean distance of $0.27m$ to each ground truth vertex pose. The next best result with $64.92m$ is obtained by the *DCS* method.

Figures 2 and 3 show the maps computed by the different methods. The changes in graph topology induced by the *Prefilter* method is especially visible in figure 2. The changes in graph topology induced by the *Prefilter* method after rejecting incongruent MoG and hyperedge components is especially visible in figure 2 showing the map from the top with an orthographic projection.

V. CONCLUSIONS

In this paper, the Generalized Graph SLAM framework was presented. Especially, a formal description of how to use hyperedges to encode uncertain loop closures was introduced. This method to handle global ambiguities is combined with multimodal edge constraints to cope with local ambiguities. Current state-of-the-art methods in robust graph-based SLAM were shown to be special cases of this Generalized Graph SLAM framework.

A method to generate good initial conditions for the most general case of multimodal hypergraphs was presented and validated with experiments on synthetic graphs. The experiments showed that this extended *Prefilter* method is both significantly more robust and less computationally demanding than current state-of-the-art approaches.

ACKNOWLEDGEMENTS

The research leading to the presented results was supported in part by the European Community's Seventh Framework Programme under grant agreement n. 270350 "Cognitive Robot for Automation Logistics Processes (RobLog)", and grant agreement n. 288704 "Marine robotic system of self-organizing, logically linked physical nodes (MORPH)".

The authors thank Jan Elseberg, Dorit Borrmann and Andreas Nüchter for providing the Bremen City Center data set.

REFERENCES

[1] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. Leonard, "Dynamic pose graph slam: Long-term mapping in low dynamic environments," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 1871 –1878.

[2] M. Fallon, H. Johannsson, J. Brookshire, S. Teller, and J. Leonard, "Sensor fusion for flexible human-portable building-scale mapping," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 4405 –4412.

[3] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 4557 – 4564.

[4] R. Leishman, J. Macdonald, T. McLain, and R. Beard, "Relative navigation and control of a hexacopter," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 4937 –4942.

[5] M. Pfingsthorn, A. Birk, and H. Bülow, "Uncertainty estimation for a 6-dof spectral registration method as basis for sonar-based underwater 3d slam," in *Robotics and Automation, 2012. Proceedings. ICRA '12. IEEE International Conference on*. IEEE Press, 2012.

[6] M. Pfingsthorn and A. Birk, "Simultaneous Localization and Mapping with Multimodal Probability Distributions," *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 143–171, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/2/143.abstract

[7] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[8] Y. Latif, C. C. Lerma, and J. Neira, "Robust loop closing over time," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[9] N. Sunderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 1254 –1261.

[10] ——, "Switchable constraints for robust pose graph slam," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 1879 –1884.

[11] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust Map Optimization using Dynamic Covariance Scaling," in *International Conference on Robotics and Automation (ICRA), 2013*, 2013.

[12] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, 2nd ed. John Wiley & Sons, Inc., March 2009.

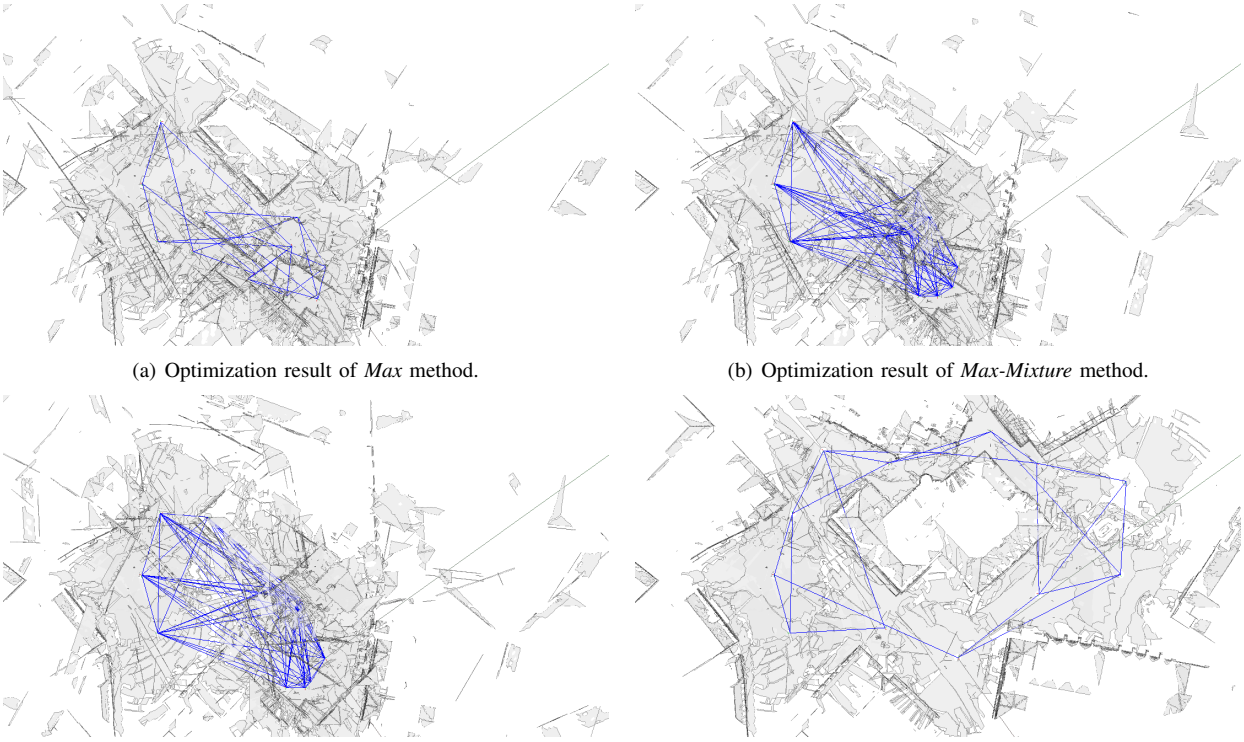
[13] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., 2005. [Online]. Available: http://dx.doi.org/10.1002/0471725382.fmatter

[14] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 826–840, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/7/826.abstract

[15] Y. Latif, C. Cadena, and J. Neira, "Realizing, reversing, recovering: Incremental robust loop closing over time using the irrr algorithm," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct., pp. 4211–4217.

[16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 3607 –3613.

[17] D. Borrmann, J. Elseberg, K. Lingemann, A. Nchter, and J. Hertzberg, "Globally consistent 3d mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, 2008.



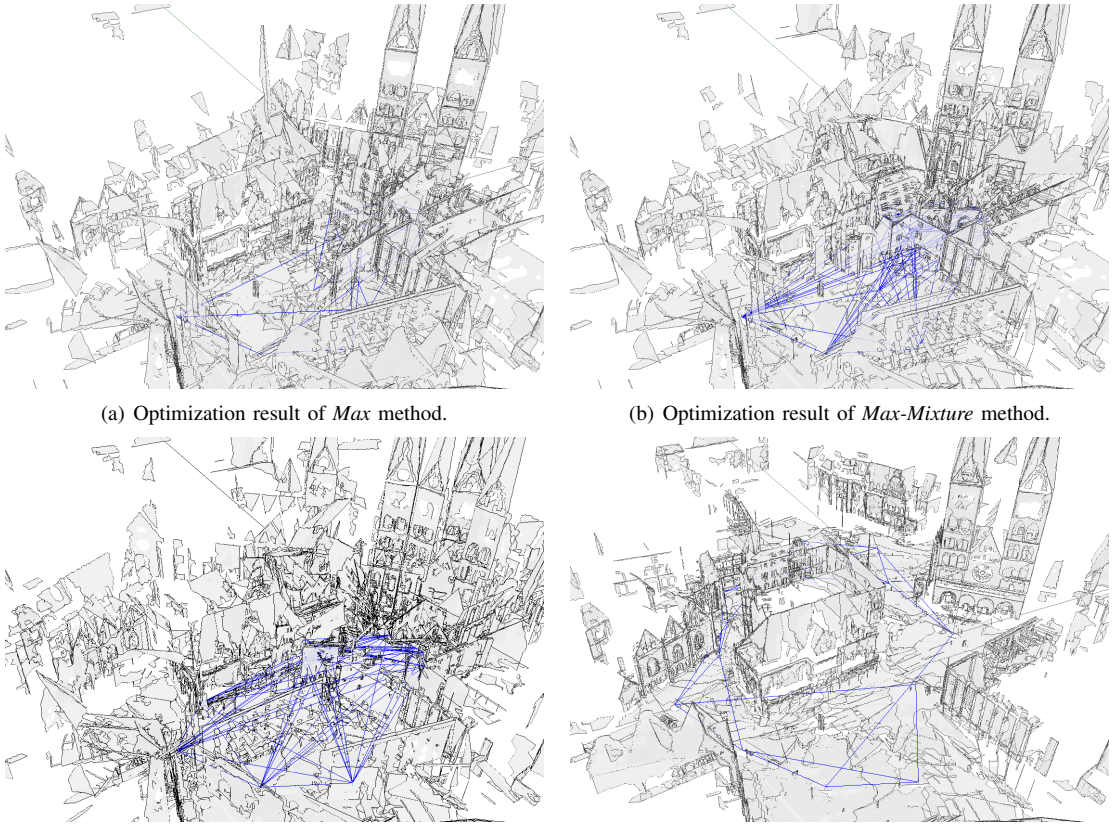
(a) Optimization result of *Max* method.

(b) Optimization result of *Max-Mixture* method.

(c) Optimization result of *DCS* method.

(d) Optimization result of *Prefilter* method.

Fig. 2. Orthographic view of the planar maps generated from the exhaustively matched Bremen City Center dataset after optimization.



(a) Optimization result of *Max* method.

(b) Optimization result of *Max-Mixture* method.

(c) Optimization result of *DCS* method.

(d) Optimization result of *Prefilter* method.

Fig. 3. Perspective view of the planar maps generated from the exhaustively matched Bremen City Center dataset after optimization.