

Probabilistic Structure Matching for Visual SLAM with a Multi-Camera Rig

Michael Kaess

CSAIL, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Frank Dellaert

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract

We propose to use a multi-camera rig for simultaneous localization and mapping (SLAM), providing flexibility in sensor placement on mobile robot platforms while exploiting the stronger localization constraints provided by omni-directional sensors. In this context, we present a novel probabilistic approach to data association, that takes into account that features can also move between cameras under robot motion. Our approach circumvents the combinatorial data association problem by using an incremental expectation maximization algorithm. In the expectation step we determine a distribution over correspondences by sampling. In the maximization step, we find optimal parameters of a density over the robot motion and environment structure. By summarizing the sampling results in so-called virtual measurements, the resulting optimization simplifies to the equivalent optimization problem for known correspondences. We present results for simulated data, as well as for data obtained by a mobile robot equipped with a multi-camera rig.

Key words: localization, mapping, mobile robot, multi-camera rig, omni-directional, SFM

1. Introduction

Visual simultaneous localization and mapping (SLAM) is the problem of creating a map of the environment from camera images, while simultaneously using this map to localize the camera. SLAM [1] is essential for many applications in mobile robotics, ranging from search and rescue over reconnaissance to commercial products such as entertainment and household robots. While cameras are cheap, small and have low energy consumption, they only provide bearing information and have higher processing requirements than other sensors. There are many publications on visual SLAM, some of which use monocular [2, 3, 4, 5, 6, 7, 8], stereo-based [9, 10, 11, 12] and omni-directional sensors [13, 14, 15, 16].

Email addresses: kaess@mit.edu (Michael Kaess), dellaert@cc.gatech.edu (Frank Dellaert)



(a)



(b)

Figure 1: (a) Our custom made 8-camera rig, mounted on top of an ATRV-Mini mobile robot platform. The FireWire cameras are distributed equally along a circle and connected to an on-board laptop. (b) A joint image from all eight cameras, providing a 360° view. Note that the camera rig has multiple optical centers.

We propose to use a *multi-camera rig* to combine the advantages of individual cameras with omni-directional sensing. In a multi-camera rig, the cameras are not in any specific stereo configuration, but face in multiple directions potentially without any overlap. While there are some complications from using multiple cameras, such as more difficult calibration and additional wiring, there are several advantages to a multi-camera rig: First, in contrast to single cameras and traditional stereo setups, a multi-camera rig covers a wider field of view and therefore provides better localization constraints. Second, in contrast to omni-directional cameras that distribute the available pixels over the complete scene, a multi-camera rig can focus the available resources on areas of interest depending on the application. Finally, the cameras of a rig do not have to be in a central location and can therefore be placed according to physical constraints of the mobile platform, allowing for new applications, such as placing cameras in the front and the back of a vehicle. A more general configuration, as we use for this work, is shown in Fig. 1, where the cameras are distributed equally along a circle.

In terms of related work, multi-camera rigs appear in the literature in the context of image-based rendering and structure from motion (SFM). A theoretical treatment of multi-camera systems in SFM is presented by Pless [17]. Levin [18] uses the Point Grey Ladybug six-camera omnidirectional rig in combination with a hand drawn map for offline loop closing in the context of visual odometry, which does not create a map. Sola [19] recently presented multi-camera visual SLAM by fusing information from multiple independent monocular cameras.

The problem of SLAM consists of two different components. One is the problem of estimating the structure and motion, which is closely related to SFM in computer vision [20]. We solve this estimation part by standard bundle adjustment [21, 4] as has been applied in several recent SLAM works [22, 23, 7, 24, 25]. The other problem is that of data association, or establishing correspondences between multiple observations, which asks whether two measurements arise from the same structure point or not.

We focus on the data association problem in this paper. Common approaches include nearest neighbor and maximum likelihood assignment. A popular framework for establishing correspondences is random sample consensus (RANSAC) [26], that is frequently used in SFM [27] and visual SLAM [9, 2, 3]. However, selecting a specific correspondence assignment involves the danger of choosing a wrong correspondence, which can lead to catastrophic failure of the system. For dense range data, scan matching [28] and the iterative closest point (ICP) algorithm [29] work well, but they are not suitable for sparse representations as encountered in visual SLAM. A more direct way of dealing with unknown correspondences is to consider multiple hypotheses. This can be based on combinatorial considerations [30], tree search [31], or lazy search that revises decision only if needed [32]. It can also be achieved in the context of particle-based representations [33, 34, 35, 36], where each particle carries its own correspondences, typically based on a per-particle maximum likelihood assignment or random sampling. However, these approaches are directly affected by the underlying combinatorial complexity of the problem, in that they might have to consider exponentially many hypotheses or particles. A Bayesian approach is taken by Dellaert *et al.* [37] that avoids the combinatorial nature of the problem by sampling from probability distributions over correspondences. However, this approach has never been applied to the domain of visual SLAM, where an incremental reconstruction is needed.

In this paper, we present an expectation maximization (EM) based approach to data association, that is based on the idea of Dellaert *et al.* [37]. In contrast to this prior batch processing work, the incoming data is processed incrementally, from one or more cameras mounted on a mobile robot. For each new frame we perform expectation maximization. The initial estimate is based on the estimate obtained for the previous frame. In the E-step, a distribution over the correspondences for the new data is approximated by Markov chain Monte Carlo sampling. In the M-step, the parameters of a density over structure and motion are optimized based on the outcome of the E-step. By using the samples to create a more efficient representation, the so-called virtual measurements, the complexity of the resulting optimization problem is reduced to the same order as that of the equivalent SLAM problem under known correspondences. We evaluate our system on simulated, as well as real data obtained from our 8-camera rig mounted on top of a mobile robot.

Next, we describe how to perform SLAM with a multi-camera rig. In section 3 we discuss our main contribution, probabilistic structure matching, and show how to deal with the distribution over correspondences by Monte Carlo EM. In section 4 we discuss the overall system including computational complexity. We finally present experimental results in section 5.

2. Multi-Camera SLAM

The goal of multi-camera SLAM is to recover the robot motion and the environment structure from images taken by the camera rig and the odometry measurements of the robot. We define the robot *motion* as $M = \{\mathbf{m}_i\}_{i=1}^m$, where \mathbf{m}_i specifies the pose at time i , and the *structure* as $X = \{\mathbf{x}_j\}_{j=1}^n$ which describes n environment features \mathbf{x}_j . The first pose \mathbf{m}_0 is constant and can be chosen arbitrarily. The odometry $O = \{\mathbf{o}_i\}_{i=1}^m$ consists of measurements \mathbf{o}_i of the difference between the robot poses \mathbf{m}_{i-1} and \mathbf{m}_i . The set of image measurements $U = \{\mathbf{u}_k\}_{k=1}^l$ consists of features \mathbf{u}_k extracted from the images by a feature point detector.

The problem can now be formulated probabilistically as finding the structure X^* and motion M^* that best explain the image measurements U and vehicle odometry O :

$$\begin{aligned} X^*, M^* &= \operatorname{argmax}_{X, M} P(X, M | U, O) & (1) \\ &\propto \operatorname{argmax}_{X, M} P(U, O | X, M) P(X, M) \\ &= \operatorname{argmin}_{X, M} (\log P(U, O | X, M) + \log P(X, M)) \end{aligned}$$

where we made use of Bayes law. $P(X, M)$ encapsulates any prior knowledge on structure and motion if available. Below, we describe how the distribution $P(U, O | X, M)$ is defined based on a generative model of the multi-camera rig projection process. To solve this system, we apply Levenberg-Marquardt, a non-linear minimization algorithm that achieves fast convergence to the global minimum based on a good initial estimate derived from the odometry and the result of the previous step.

2.1. Camera Rig Projections

A *multi-camera rig* is a set of c cameras fixed with respect to each other and the robot. Note that we make no assumptions about the orientation of the cameras. They can face in any direction that seems suitable for a specific application, as long as they are statically fixed with respect to each other and the robot. A *joint image* consists of a set of images obtained synchronously from all rig cameras, where the pixel $\mathbf{p} = (u, v)$ in rig camera r is represented by the tuple (r, \mathbf{p}) . Note that this corresponds to a single camera with multiple optical centers, which means that a single point in the environment can have more than one projection in the joint image.

Projecting a 3D world point \mathbf{x} when the robot is at pose \mathbf{m} is a two-step process. First, the world point is transformed to rig coordinates $\mathbf{x}' = R^T(\mathbf{x} - \mathbf{t})$, where the 3D rotation R and translation \mathbf{t} are given by the pose $\mathbf{m} = (R, \mathbf{t})$. Second, this point \mathbf{x}' is projected into rig camera $r \in \{1 \dots c\}$, where c is the number of cameras in the rig. Using standard practices [20], the overall projection $h_r(\mathbf{m}, \mathbf{x})$ is given by

$$h_r(\mathbf{m}, \mathbf{x}) = K_r [R_r | \mathbf{t}_r] R^T (\mathbf{x} - \mathbf{t}) \quad (2)$$

with the *intrinsic* calibration matrix

$$K = \begin{bmatrix} \alpha_x & s & u_0 \\ & \alpha_y & v_0 \\ & & 1 \end{bmatrix} \quad (3)$$



Figure 2: An example of radial distortion removal, comparing the original image (left) with the corrected image (right).

where α_x and α_y are the focal lengths in pixels, s is the skew (typically 0), and $\mathbf{p}_0 = (u_0, v_0)$ is the principal point. The *extrinsic* calibration parameters for camera r consist of the 3×3 rotation matrix R_r and the translation vector \mathbf{t}_r with respect to the center of the robot. The overall rig calibration as summarized by $\{K_r, R_r, \mathbf{t}_r\}_{r=1}^c$ is determined in advance.

Additionally it is necessary to model *radial distortion*. We approximate radial distortion by the quadratic function $r_D = r_U + \kappa r_U^2$ with a single parameter κ , as well as the center of distortion $\mathbf{p}_D = (u_D, v_D)$ that might be different from the principal point \mathbf{p}_0 . r_U and r_D are the radii of the projected point in ideal coordinates before and after distortion, respectively. To remove radial distortion, the incoming images are warped efficiently using a look-up table that is calculated only once. Fig. 2 shows an example image with significant radial distortion and its corrected counterpart.

2.2. Measurements and Generative Models

We define the measurement distribution $P(U, O|X, M)$ from (1) in terms of generative models for image and odometry measurements. The generative model for an image measurement $\mathbf{u} = h_r(\mathbf{m}, \mathbf{x}) + \mathbf{v}$ of 3D point \mathbf{x} in rig camera r at pose \mathbf{m} is given by the geometric prediction $h_r(\mathbf{m}, \mathbf{x})$ from (2) with added measurement noise \mathbf{v} , that we assume to be i.i.d. zero-mean Gaussian with covariance Ξ . Similarly, the odometry measurement $\mathbf{o}_i = d(\mathbf{m}_{i-1}, \mathbf{m}_i) + \mathbf{w}$ is given by the difference $d(\mathbf{m}_{i-1}, \mathbf{m}_i)$ of the two poses \mathbf{m}_{i-1} and \mathbf{m}_i with zero-mean Gaussian noise with covariance Ω added. Assuming known correspondences $J = \{\mathbf{j}_k\} = \{(i_k, j_k, (r_k, \mathbf{u}_k))\}_{k=1}^l$, where the triple $\mathbf{j}_k = (i_k, j_k, (r_k, \mathbf{u}_k))$ describes an image measurement \mathbf{u}_k of map point j_k as observed by rig camera r_k at time i_k , equation (1) simplifies to the minimization of a sum of terms over l correspondences and m odometry measurements

$$\begin{aligned} & \log P(U, O|X, M) \\ &= \sum_{k=1}^l \|\mathbf{u}_k - h_{r_k}(\mathbf{m}_{i_k}, \mathbf{x}_{j_k})\|_{\Xi_k}^2 + \sum_{i=1}^m \|\mathbf{o}_i - d(\mathbf{m}_{i-1}, \mathbf{m}_i)\|_{\Omega_i}^2 \end{aligned} \quad (4)$$

where $\|\mathbf{y}\|_{\Sigma}^2 := \mathbf{y}^T \Sigma^{-1} \mathbf{y}$.

2.3. Features and Templates

We use the Harris corner detector [38] to identify features in the input images. After thresholding on the Harris response, we perform non-maximum suppression,

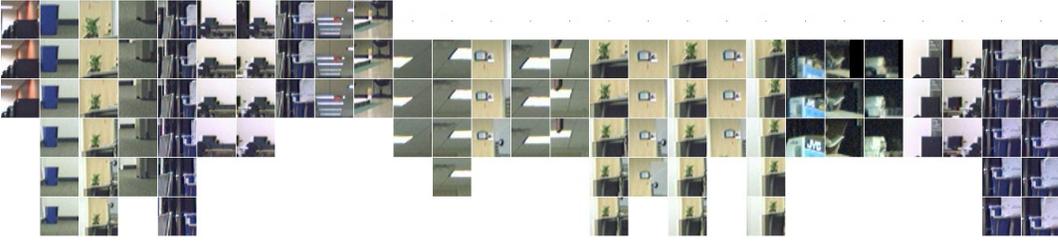


Figure 3: Correspondences are shown as templates of the measurements of different points (columns) over time (rows). To provide more context in the visualization, much larger templates are shown than the ones that are actually used in the matching process. Note that appearance changes over time depending on the motion of the robot and the bearing at which the feature is observed.

selecting the best feature in any 5×5 neighborhood. To prevent an overly large number of features in highly textured environments, only a limited number of features corresponding to the highest Harris responses are accepted.

We preselect potential matches between features in different frames based on the appearance of their local neighborhood. We define a template \mathbf{v} to be the vector of intensity values of a square region centered on the feature. The dissimilarity of two templates \mathbf{v}_a and \mathbf{v}_b is given by their sum of square differences $\|\mathbf{v}_a - \mathbf{v}_b\|^2$. To make the matching process more efficient, we restrict the search region to some area around the odometry-based epipolar prediction. Note that for a camera rig, A and B are joint images that each consist of multiple real images. Matches cannot only occur between images taken by the same rig camera over time, but can also stretch across different rig cameras. Fig. 3 shows an example of matches extracted over time and between rig cameras (based on our reference RANSAC implementation [39]).

3. Probabilistic Structure Matching

In this section we present a Bayesian approach to data association that recovers correspondences not just between subsequent frames, but instead matches against all structure while taking into account the uncertainties of the estimation process. In contrast to the incremental data association in our earlier work [39], this allows for loop closing, ie. establishing correspondences to earlier parts of the trajectory. Furthermore it can re-acquire feature tracks that are interrupted in the incremental matching due to noise or occlusion in an intermediate image. And finally, it automatically matches features across cameras of a multi-camera rig.

We take a probabilistic approach in which *we recover the parameters of a density* over the robot motion and the structure of the environment, based on the image and odometry measurements. We use Θ to denote the parameters of a Gaussian distribution over structure X and motion M , so that $\Theta = (\mu, \Sigma)$ consists of a mean $\mu = (\mathbf{m}_1 \dots \mathbf{m}_m, \mathbf{x}_1 \dots \mathbf{x}_n)^T$ and a covariance Σ for the combined structure and motion.

In order to allow loops to be closed, we need to establish correspondences between measurements and structure. But if we assume for a moment that the data-association problem has been solved, inference on Θ is relatively straightforward.

That is, assume that all *correspondences* $J_{1:m}$ are known, where $J_{1:m} \subseteq J$ is the subset of correspondences relating to only the first m frames, taken from the full set J of correspondences available for the batch problem. In that case we determine a new estimate Θ of the parameters by maximizing the likelihood $P(\Theta|U_{1:m}, O_{1:m}, J_{1:m})$ given all image measurements $U_{1:m}$ and all odometry measurements $O_{1:m}$:

$$\Theta = \arg \max_{\Theta} P(\Theta|U_{1:m}, O_{1:m}, J_{1:m}) \quad (5)$$

However, we typically *do not know* the correspondences $J_{1:m}$. In that case we can rewrite (5) to consider all possible correspondences $J_{1:m}$ by marginalization:

$$\Theta = \arg \max_{\Theta} P(\Theta) \sum_{J_{1:m}} P(U_{1:m}, O_{1:m}, J_{1:m}|\Theta) \quad (6)$$

Unfortunately, it is intractable to sum over all correspondences $J_{1:m}$ for any non-trivial example because of the combinatorial complexity of the correspondence problem.

The key idea in our approach is to circumvent the combinatorics of (6) by using an incremental EM formulation, ie. perform EM for every new frame, which is approximate in both E and M steps:

1. In the E-step, we use Markov chain Monte Carlo (MCMC), an approximate inference method, to sample over the correspondences J_m *in the current time-step only, rather than over all correspondences* $J_{1:m}$.
2. In the M-step, we maximize an approximate expected log-likelihood function where *only the log-likelihood terms dependent on J_m are re-calculated*.

3.1. An EM-based Solution

We find the optimal structure and motion parameters Θ by treating the unknown correspondences J_m of the maximization problem (6) as hidden variables in the expectation maximization (EM) framework, as pioneered by [40]. EM is an iterative algorithm that alternates between an expectation (E) and a maximization (M) step [41], and maximizes the likelihood in an iterative fashion. For a derivation and further details on EM see [41] or [42].

We start iteration $t+1$ of the EM algorithm from the previous estimate Θ^t , where we define the initial estimate $\Theta^0 := \Theta'$ to be the estimation result Θ' of the previous frame (with new variables initialized based on local measurements). In the **E-step** we obtain a distribution over the hidden variables while keeping the parameters fixed. Applied to our problem this yields the posterior distribution $f^t(J_m)$ over all possible correspondences J_m based on the current structure and motion parameter estimate Θ^t

$$f^t(J_m) = P(J_m|U_m, \Theta^t) \quad (7)$$

Note that the correspondences are independent of the odometry measurement given the parameters Θ^t .

In the **M-step** we re-estimate the parameters Θ^{t+1} that maximize the expected log-likelihood, where the expectation is taken with respect to the posterior $f^t(J_m)$.

In other words, we now keep the distribution over correspondences fixed while obtaining a better estimate for the structure and motion parameters. For our maximum likelihood (ML) problem the new parameters Θ^{t+1} are defined by

$$\Theta^{t+1} = \arg \max_{\Theta} Q^t(\Theta) \quad (8)$$

with $Q^t(\Theta)$ the expected log-likelihood of the parameters Θ

$$\begin{aligned} Q^t(\Theta) &= \langle \log P(U_{1:m}, O_{1:m}, J_{1:m} | \Theta) \rangle_{f^t(J_{1:m})} \\ &:= \sum_{J_{1:m}} f^t(J_{1:m}) \log P(U_{1:m}, O_{1:m}, J_{1:m} | \Theta) \\ &= \sum_{J_m} f^t(J_m) \log P(U_m, J_m | \Theta) + \log P(O_m | \Theta) + Q'(\Theta) \end{aligned} \quad (9)$$

where $Q'(\Theta)$ summarizes the log-likelihood terms from the previous step, that do not depend on J_m and do not have to be re-calculated. The resulting algorithm is guaranteed to converge to a local maximum based on the convergence proof of EM. We can also expect fast convergence to a solution close to the global maximum, as we start from a good initial estimate based on the previous parameters Θ' .

3.1.1. E-step and Distribution over Correspondences

We show how to evaluate the posterior distribution $f^t(J_m)$ over correspondences from (7) for given structure and motion parameters Θ^t , as needed in the E-step. By applying Bayes law we obtain the product of a correspondence likelihood and a correspondence prior

$$\begin{aligned} f^t(J_m) &= P(J_m | U_m, \Theta^t) \\ &\propto P(U_m | J_m, \Theta^t) P(J_m | \Theta^t) \end{aligned} \quad (10)$$

The likelihood $P(U_m | J_m, \Theta^t)$ of the correspondences J_m can be factorized, as the individual measurements are independent given the parameters Θ^t :

$$P(U_m | J_m, \Theta^t) = \prod_{k \in K_m} P(\mathbf{u}_k | J_m, \Theta^t) \quad (11)$$

We define the predictive density $P(\mathbf{u}_k | J_m, \Theta^t)$ for an individual image measurement \mathbf{u}_k by a generative model, based on geometrical considerations of the measurement process. The image measurement function $h_{r_k}(\mathbf{m}_m, \mathbf{x}_{j_k})$ as defined in Section 2 predicts the measurement \mathbf{u}_k in rig camera r_k for known point location \mathbf{x}_{j_k} and robot pose \mathbf{m}_m by projection. We approximate the image measurement density in terms of the marginalized covariance $\Xi_k = H_k \Sigma_{j_k} H_k^T + \Xi$ that is derived in the Appendix, where Ξ is the measurement noise, H_k is the linearization of $h_{r_k}(\cdot)$ evaluated at the current parameters \mathbf{m}_m and \mathbf{x}_{j_k} , and $\Sigma_{j_k} = \Sigma_{m_j}$ is the marginal covariance of this pose and point pair $(\mathbf{m}_m, \mathbf{x}_j)$:

$$\Sigma_{m_j} = \begin{pmatrix} P_m & B_{mj} \\ B_{mj}^T & S_j \end{pmatrix} \quad (12)$$

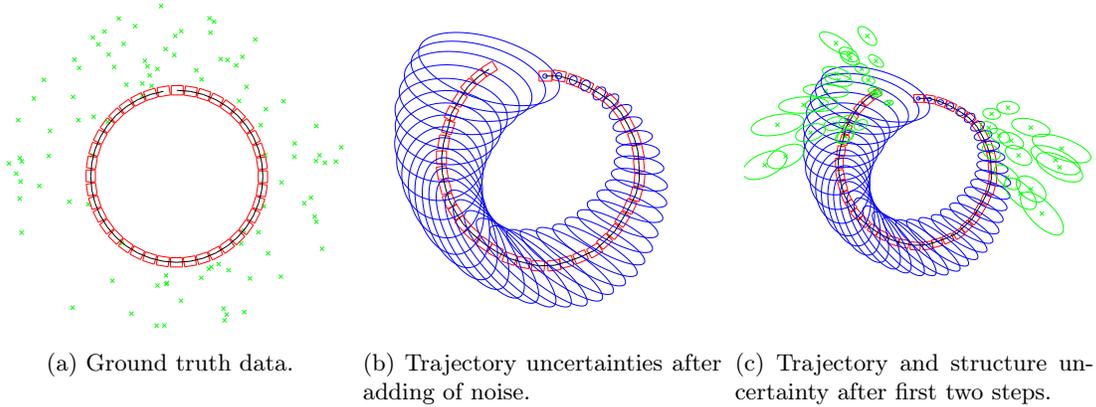


Figure 4: A simulated environment that illustrated the estimation uncertainties. We simulate a rig with one forward and one backward facing camera in an environment with 40 structure points. The robot poses are shown as red rectangles, with the projected uncertainty ellipses in blue. The structure is shown by green crosses with projected uncertainty ellipses. The black arc represents the noisy odometry. a) Ground truth data. b) Significant noise added to odometry and image measurements, resulting in high uncertainties towards the end of the circular trajectory. c) The estimation uncertainty after the first two steps.

which consists of the point uncertainty P_m , structure uncertainty S_j and covariances B_{mj} , which are all components of the overall estimation uncertainty Σ . To avoid recovering the full covariance, we reorder the variables so that the current pose and all structure are in the last columns. By matrix factorization, we obtain the covariance Σ' by back-substitution over those last columns only

$$\Sigma' = \begin{pmatrix} P_m & & B_{mj} & & \\ & \ddots & & & \\ B_{mj}^T & & S_j & & \\ & & & \ddots & \end{pmatrix} \quad (13)$$

An example of the uncertainties is shown in Figure 4. The predictive density is now simply a normal distribution

$$\begin{aligned} & P(\mathbf{u}_k | J_m, \Theta^t) \\ &= P(\mathbf{u}_k | \mathbf{m}_m, \mathbf{x}_{j_k}) \\ &\approx \frac{1}{\sqrt{|2\pi\Xi_k|}} \exp -\frac{1}{2} \|\mathbf{u}_k - h_{r_k}(\mathbf{m}_m, \mathbf{x}_{j_k})\|_{\Xi_k}^2 \end{aligned} \quad (14)$$

Note that we cannot omit the normalization factor, as the covariance Ξ_k does depend on Θ^t through H_k and Σ_{j_k} .

What remains to be defined from the posterior distribution $f^t(J_m)$ is the prior $P(J_m | \Theta^t)$ over correspondences. This prior encodes the mutual exclusion constraint that allows a point or a measurement to appear in at most one correspondence. In other words, the prior $P(J_m | \Theta^t)$ is 0 for any configuration that violates the mutual exclusion constraint.

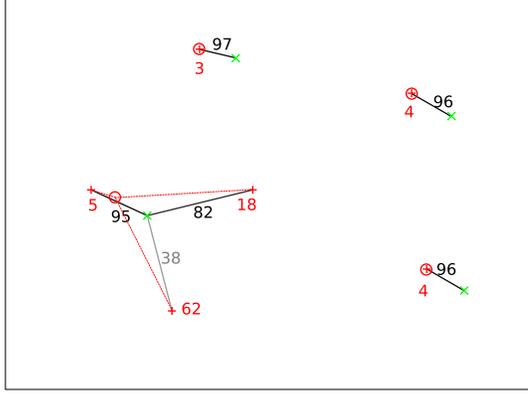


Figure 5: Example of virtual measurements (red 'o') in an image frame with 6 image measurements (red '+') and 4 projected structure points (green 'x'). The numbers represent percentage probabilities of assignments (black) versus spurious (red, near '+') for each original image measurement. In the lower left part of the image, an ambiguous configuration between three image measurements and one projection (in combination with a high measurement uncertainty Ξ) leads to a virtual measurement on that projection that corresponds to a weighted average of the original measurements.

3.1.2. M-step and Virtual Measurements

In the M-step we maximize an approximate expected log-likelihood function to obtain the structure and motion parameters Θ that best explain the observed data given the distribution over correspondences J_m obtained in the E-step. Even though we deal with a distribution over correspondences, we show that the optimization problem has the same number of terms as the standard SFM problem with known correspondences. This is made possible by the introduction of *virtual measurements* [37] as weighted average of the original measurements \mathbf{u}_k

$$\mathbf{v}_{mj}^t := \Xi_{mj} \sum_{k \in K_m} f_{jk}^t \Xi^{-1} \mathbf{u}_k \quad (15)$$

with virtual measurement covariance Ξ_{mj} defined by

$$\Xi_{mj} = \left(\sum_{k \in K_m} f_{jk}^t \right)^{-1} \Xi \quad (16)$$

Here f_{jk}^t is the marginal posterior probability that measurement \mathbf{u}_k of image m corresponds to feature \mathbf{x}_j , defined as

$$\begin{aligned} f_{jk}^t &:= P(j = j_k \in J | U_m, \Theta^t) \\ &= \sum_{J_m} \delta(j, j_k \in J_m) f^t(J_m) \end{aligned} \quad (17)$$

An example of virtual measurements is shown in Figure 5. Except for ambiguous situations such as in the lower left of the example, virtual measurements usually coincide with the actual measurements. In ambiguous situations, however, they provide

weighted averages of possible candidate measurements. Note that the virtual measurements are also associated with an uncertainty, so that more ambiguous matches are automatically discounted. Virtual measurements delay the actual decision, and therefore avoid making a potentially wrong decision, that other methods such as RANSAC might produce under high uncertainty. After a few EM iterations many ambiguous situations will be resolved. Specifically, in this example we can expect that the reprojected structure point moves closer to the top left feature (near '5'), because the other measurements and structure points will pull the estimate in that direction. The virtual measurement will then become more strongly associated to the top left feature, and eventually the links to the other two features will completely disappear.

Note that virtual measurements are used in the context of data association and are not used to represent the underlying SLAM estimation uncertainty. The estimation is performed by the standard maximum likelihood formulation of SLAM given in (8) that correctly deals with the estimation uncertainty, but using the virtual measurements as input to deal with uncertainty in the data association part only.

The objective function (8) to be maximized is the expected log likelihood $Q^t(\Theta)$ from (9). Applying the chain rule to the likelihood $P(U_m, J_m|\Theta)$ of the parameters Θ yields $P(U_m|J_m, \Theta)P(J_m|\Theta)$. In contrast to the E-step, we can now assume that the prior $P(J_m|\Theta)$ does not depend on structure and motion, ie. $P(J_m|\Theta) = P(J_m)$. This is justified as we rewrite the equation in terms of the virtual measurements, and therefore do not have to deal here with the mutual exclusion constraint. We drop the prior $P(J_m)$ since it does not depend on the parameters Θ :

$$Q^t(\Theta) = \log P(O_m|\Theta) + Q'(\Theta) + \underbrace{\sum_{J_m} f^t(J_m) \log P(U_m|J_m, \Theta)}_A \quad (18)$$

where the odometry measurement \mathbf{o}_m is predicted based on the process model $d(\mathbf{m}_{m-1}, \mathbf{m}_m)$ from the previous robot pose \mathbf{m}_{m-1} and the current pose \mathbf{m}_m , adding zero-mean Gaussian measurement noise with covariance Ω :

$$P(\mathbf{o}_m|\Theta) = P(\mathbf{o}_m|\mathbf{m}_m, \mathbf{m}_{m-1}) \propto \exp -\frac{1}{2} \|d(\mathbf{m}_{m-1}, \mathbf{m}_m) - \mathbf{o}_m\|_{\Omega}^2 \quad (19)$$

The key to an efficient implementation is the independence of image measurements for given correspondences J_m and parameters Θ . We apply the resulting factorization of $P(U_m|J_m, \Theta)$ from (11) to the measurement term A of (18)

$$A = \sum_{J_m} f^t(J_m) \sum_{k \in K_m} \log P(\mathbf{u}_k|J_m, \Theta) \quad (20)$$

Using the definition of f_{jk}^t from (17) in (20), we now sum over only n structure points rather than all possible correspondences J_m :

$$A = \sum_{j=1}^n \sum_{k \in K_m} f_{jk}^t \log P(\mathbf{u}_k|j_k, \Theta) \quad (21)$$

Finally substituting the definition of the virtual measurements from (15) and (16) in (21), together with the measurement log likelihood from (14), we obtain a sum of terms that is equivalent to the original SFM problem with known correspondences:

$$\begin{aligned}
A &= \sum_{j=1}^n \sum_{k \in K_m} f_{jk}^t \log P(\mathbf{u}_k | j_k, \Theta) \\
&= \sum_{j=1}^n \sum_{k \in K_m} f_{jk}^t \left(-\frac{1}{2} \|\mathbf{u}_k - h_{r_k}(\mathbf{m}_m, \mathbf{x}_{j_k})\|_{\Xi_k}^2 + \log \frac{1}{\sqrt{|2\pi\Xi_k|}} \right) \\
&= C - \frac{1}{2} \sum_{j=1}^n \sum_{k \in K_m} f_{jk}^t \|\mathbf{u}_k - h_{r_k}(\mathbf{m}_m, \mathbf{x}_{j_k})\|_{\Xi_k}^2 \\
&\approx C - \frac{1}{2} \sum_{j=1}^n \|\mathbf{v}_{mj}^t - h(\mathbf{m}_m, \mathbf{x}_j)\|_{\Xi_{mj}}^2
\end{aligned} \tag{22}$$

Note that A only contains the terms of the current frame m , the others are remembered in $Q'(\Theta)$ in (18).

3.2. Monte Carlo EM

There is no closed form solution available for the expected log likelihood $Q^t(\Theta)$ in (18), due to the mutual exclusion constraint that is contained in the posterior $f^t(J_m)$ (9) [37]. However, for given correspondences J_m , this posterior can easily be evaluated, which suggests the use of a sampling based approximation in the E-step. Replacing the E-step of EM by a sampling approximation is known in the literature as Monte Carlo EM (MCEM) [43]. Its application to our problem is described in this section.

3.2.1. E-step and Sampling

To get a sampling approximation of the posterior distribution $f^t(J_m)$ over all possible correspondences J_m , we use the Metropolis-Hastings [44] algorithm to setup a Markov chain with the correct stationary distribution. The algorithm only evaluates our target distribution $f^t(J_m)$ for a number of different parameters, and generates a sequence of Γ samples $\{J^{t,\gamma}\}_{\gamma=1}^{\Gamma}$ from the distribution as follows:

1. Start with random, but valid initial correspondences $J^{t,0}$.
2. Obtain new correspondences J' according to the proposal density $g(J'; J^{t,\gamma})$.
3. Calculate the acceptance ratio

$$a = \frac{f^t(J') g(J^{t,\gamma}; J')}{f^t(J) g(J'; J^{t,\gamma})} \tag{23}$$

4. $J^{t,\gamma+1} = \begin{cases} J' & \text{if } a \geq 1 \text{ or with probability } \min(1, a) \\ J^{t,\gamma} & \text{otherwise} \end{cases}$
5. Repeat from step 2

There are many possible choices for the proposal distribution g . One possible proposal flips the assignments from two different correspondences. More intelligent proposals that lead to faster convergence are discussed in [37]. Specifically, we employ the ‘‘smart chain flipping’’ algorithm.

Algorithm 1 Summary of our approach. For details please refer to the sections specified in brackets.

For each new frame m :

1. Detect features [2.3].
 2. Iteratively improve estimate Θ^t by Monte Carlo EM, starting from $t = 0$ with Θ^0 based on estimation result from previous frame $m - 1$ [3.2].
 - (a) E-step: Obtain Γ samples from $f^t(J_m)$ [3.1.1] using Metropolis-Hastings algorithm [3.2.1] based on Θ^t
 - (b) M-step: Obtain Θ^{t+1} by non-linear optimization using virtual measurements [3.1.2] based on counting events in samples from $f^t(J_m)$ obtained in E-step [3.2.2].
 3. Add unassigned features temporarily as new structure [4.1].
-

3.2.2. M-step and Virtual Measurements

Our sampling approximation allows evaluation of the marginal posterior distribution, and consequently also the execution of the M-step. The sampling approximation to the marginal posterior distribution f_{jk}^t from (17) is obtained by keeping a count C_{jk} of how often feature j is assigned to measurement \mathbf{u}_k during the sampling process in the E-step:

$$f_{jk}^t \approx \frac{1}{\Gamma} \sum_{\gamma=1}^{\Gamma} \delta(j_k^\gamma, j) = \frac{1}{\Gamma} C_{jk} \quad (24)$$

As a consequence we do not have to store all the samples or a count for every possible correspondence vector J_m , but rather only keep a matrix of $K_m \times n$ entries. Note that each sampled correspondence vector increments several entries of this matrix. Once the sampling finishes, this table is replaced by an even more compact representation of n virtual measurements (15) and covariances (16).

4. Overall System

A summary of our approach is shown in Algorithm 1. Below we still need to discuss how the algorithm is initialized and how new structure is added. We follow up with an illustration based on simulated data.

4.1. Spurious Features and Initialization

So far we have assumed that we have already acquired some structure. However, we need to discuss how this process is initialized, and also how structure is added when exploring new areas. If we already have some structure, then all the points that are currently visible and got detected in the current image will be covered by virtual measurements. Most of those virtual measurements directly correspond to single features, while others are weighted averages of multiple features as shown in Fig. 5. However, typically some features do not contribute at all to any virtual measurement. We call those features spurious.

Spurious features can be random detections due to noise. But spurious features also regularly get detected along boundaries of partial occlusions. And the projections

of previously unseen features are also classified as spurious, even though they should be added to the structure. However, from a single image there is no constraint on the depth of the point, but only on its direction. Fortunately we do have some idea of its depth, though: The point has to be in front of the camera. Inside a building the maximum distance is also restricted. In fact, the depth has some distribution that is initially estimated, and later determined empirically from the data.

Using a Gaussian approximation with variance τ^2 , we find the uncertainty ellipsoid Q in normalized coordinates as

$$Q = \text{diag}(\sigma^2, \sigma^2, \tau^2) \quad (25)$$

By back-projection we obtain the point uncertainty S as

$$S = (H^T Q^{-1} H)^{-1} \quad (26)$$

The projected uncertainty Ξ_{new} in the next image is obtained by combining S with the motion uncertainty P relative to the previous image, projecting it into the image, and adding image measurement noise Ξ

$$\Xi_{new} = H \begin{pmatrix} P & \\ & S \end{pmatrix} H^T + \Xi \quad (27)$$

Using its projected uncertainty, the candidate point is added temporarily and processed as any other feature. There are two possible outcomes for the candidate: Its virtual measurement can either have a high variance, indicating that the point indeed is spurious and should be discarded. Or, in the case of a low variance, it is likely that a new feature has been observed and needs to be added to the map. To make the process more robust, we require more than one confirmation of the point. Three observations correspond to a trifocal constraint which is very unlikely to be caused by wrong matching.

When processing the first image, all features are automatically classified as spurious, as the structure is empty. As a consequence, the structure is automatically initialized from the first image, and no special handling is necessary.

4.2. Illustration on Simulated Data

To test our system with known ground truth, we have simulated a circular environment with a trajectory length of 30 meters, with 40 measurements taken at uniform intervals. The simulated robot has a rig with two cameras, one forward and one backward facing. The environment consists of 100 features that were projected into the rig cameras. Pure point features were used without any appearance information. The ground truth is shown in Figure 4a. We have added realistic noise with standard deviations $\sigma = 1$ pixel for image measurements and $\sigma_x = \sigma_y = 0.05m$ and $\sigma_\alpha = 0.03rad$ for odometry translation and rotation, respectively. Figure 4b shows the resulting noisy trajectory including the projected uncertainties.

As expected, the pose uncertainty towards the end of the trajectory and the structure uncertainty are both large after the first two steps of matching (Figure 4c). After almost half the loop is traversed (Figure 6a), the uncertainty of the initial structure is low, as many more constraints have been added. However, the current

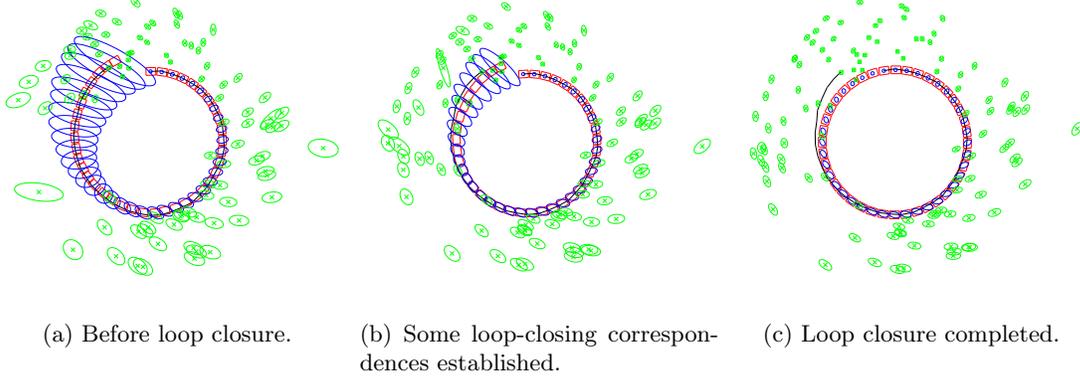


Figure 6: Results of our algorithm applied to the simulated environment from Figure 4 assuming a rig with one forward and one backward facing camera. As before, the robot poses are shown as red rectangles, with the projected uncertainty ellipses in blue. The structure is shown by green crosses with projected uncertainty ellipses. The black arc represents the noisy odometry. a) Uncertainties shortly before loop closing, ie. before the first correspondences are established to structure that was originally observed near the beginning of the trajectory. b) Some observations overlap with those from the start of the trajectory; note that the overall uncertainties decrease. c) After closing the loop the highest uncertainties are on the far end of the trajectory.

pose uncertainty is larger than near the beginning of the loop, as the initial structure cannot be seen from this location, and therefore uncertainties add up along the loop. Once some of the initial structure is re-acquired (Figure 6b), the uncertainty in the system shrinks significantly. When the loop is completely closed (Figure 6c), the largest uncertainties in structure and motion with respect to the starting point are opposite of the starting point, as one would expect.

After our algorithm successfully closes the loop, the reconstruction contains 107 features. All features were reconstructed, but 7 of them have duplicates. Note that while the camera rig has advantages as it provides better constraints, the two cameras with non-overlapping views also add some challenges. The structure created from one camera needs to be matched against the second one later, which in itself is a loop closing problem that is successfully managed by our system.

4.3. Computational Complexity

We discuss the computational complexity of the overall algorithm, starting from the innermost loop. The actual estimation algorithm, the nonlinear optimization with virtual measurements, requires exactly the same calculations as traditional SLAM with known data association. In particular, both the number of variables and the number of measurements in the estimation are the same. We use a fast method for performing this batch estimation called smoothing and mapping (SAM) [23]. Only a small number of iterations (typically 2 to 4) of the Levenberg-Marquardt algorithm are needed as most variables have already been optimized based on previous frames, and only the ones that are visible from the current frame are directly affected. While it is difficult to provide general complexity bounds for SAM, we can provide a bound

for a special case of SLAM problems that have a planar meta-graph structure [45]: In this case, batch solution takes $O(N^{1.5})$ for large-scale problems with arbitrary loops, where $N = O(n + m)$ is the number of variables for structure and motion combined. Note that SLAM with a multi-camera rig inside a building has such a structure and is therefore covered by our analysis.

The EM algorithm in the outer loop requires the nonlinear optimization to be performed multiple times, adding a constant factor to the complexity. As only the data association of the most recent frame is affected by the EM algorithm, the algorithm typically converges within a few iterations (in our example always below 10). Additionally, sampling has to be performed in each iteration. A constant number of samples is obtained, but the complexity depends on the number of visible structure points, and is therefore $O(n)$. Sampling therefore does not increase the computational complexity bound of the overall algorithm. However, sampling requires access to parts of the covariance matrix, which can be very expensive to obtain when calculated naively ($O(N^3)$). Our initial work was severely affected by this problem, leading to the discovery of faster covariance recovery methods as described in Section 3.1.1, and eventually to the work presented in [25]. For the latter one, the complexity depends on a number of factors and is difficult to quantify, but has been shown to work in real-time for large SLAM problems (over 20000 variables, [25]). Note that the same covariances are needed for the combinatorial JCBB data association algorithm [46].

As a summary we can state that the proposed method adds a constant, but not insignificant, factor to the complexity of state-of-the-art full SLAM estimation methods that operate on known data association.

5. Experimental Results and Discussion

We have successfully implemented and applied our system to real world multi-camera rig data. In our implementation, non-linear optimization is performed using the Levenberg-Marquardt algorithm, which achieves fast convergence by adaptively blending between Gauss-Newton and gradient descent. We use a sparse LDL decomposition [47, 48] with approximate minimum degree ordering for efficiency [23]. An automatic differentiation framework [49] allows us to efficiently calculate a Jacobian for a given point, free of numerical instabilities. The minimum track length used for the optimization is 4.

The data was taken with our custom-made multi-camera rig mounted on a mobile robot platform (see Figure 1) traversing an indoor office environment. The camera rig consists of 8 cameras in a circular arrangement to provide a complete 360 degree coverage. The robot’s trajectory is nearly rectangular with side lengths of $10m$ and $20m$. The measurements consist of the odometry provided by the ATRV-Mini mobile robot, as well as 87 joint images taken with variable distances of up to 2 meters between successive views, and an overall trajectory length of about $58m$. An average of 145 Harris features were extracted from each joint image. The unknown poses were modeled as having 6 degrees of freedom (DOF), three translational and three rotational, parametrized as Euler angles yaw, pitch and roll. Even though 3 DOF seems sufficient for a planar indoor office environment, we found that 6 DOF with a prior on the pitch, roll and height is necessary, because floors are typically not completely flat



Figure 7: Camera image with measurements and reprojected structure.

and the robot pitches when changing velocity. From a reference reconstruction based on incremental trifocal RANSAC-based matching [39], the standard deviations were statistically determined on x to be $0.022m$, on y $0.023m$ and on yaw $0.012rad$. Noise on z , pitch and roll was modeled as zero mean normal distributions with standard deviations $0.002m$ and $0.002rad$ respectively.

The camera rig was calibrated in advance by a semi-automatic approach. We placed sets of colored points on three different walls in a small space and recorded joint images at several different locations and for multiple rotations of the rig. We then manually initialized the robot poses by identifying a small number of points in each joint image. The remaining procedure was automated: First, feature points of the calibration pattern were detected. Second, the known geometry of the calibration patterns are projected into the image based on the initial pose estimates. Third, data association is performed by nearest neighbor. And finally, the calibration parameters (see Section 2.1), the relative poses within the camera rig as well as the exact offsets between the three calibration patterns are estimated iteratively.

In order to make full use of visual features, appearance needs to be considered. Appearance can be modeled and included into the EM algorithm. However, it then becomes necessary to recalculate the virtual measurements over all times in each step. Since this is too expensive, we choose an approximation that obtains the appearance from the geometrically closest viewpoint. The reasoning behind this solution is that we can expect the appearance of a point to change more significantly with increased distance and rotation between the viewpoints, and therefore the best available information comes from the geometrically closest pose. Typically this is the previous frame, but in the case of a loop closing it can also originate much earlier in the trajectory. The appearance is extracted from the corresponding virtual measurement in the closest frame by taking a weighted average over the templates around the original feature points, based on the marginals. These so called virtual templates are calculated in each step and stored together with the virtual measurements, so that the full marginals do not need to be remembered.

Figure 8 shows the trajectory based on the noisy odometry measurements (right),



(a) Final trajectory and map (left) and trajectory based on odometry only (right).



(b) Uncertainties before (left) and after loop closing (right).

Figure 8: Results of probabilistic structure matching based on real data from our 8-camera rig mounted on top of an ATRV-Mini platform. (a) The trajectory (red rectangular robot outlines) and the structure (green points) after loop closing are shown on the left, manually aligned with the building map (gray) for comparison. The only input to the algorithm were the features extracted from the images, and the robot's odometry measurements as shown on the right. Note that the structure consists of 3D points, and many of the features are along the ceiling, only apparently in the middle of hallways, see Figure 7. (b) Structure uncertainties (green ellipses) and pose uncertainties (blue ellipses) are shown before and after loop closing.

as well as the result of our algorithm (left), with a successfully closed loop. For reference, the trajectory is manually aligned with the building map. Note that while some structure appears in the middle of hallways, the model is actually three dimensional and features are also extracted from the ceiling as can be seen in Figure 7. The reconstruction contains 593 structure points that are based on 4480 virtual measurements, which corresponds to an average track length of 7.6. In comparison to an average track length of 3.9 for our reference incremental trifocal reconstruction [39], this clearly shows our algorithm’s capability of locally closing the loop by re-acquiring features that were temporarily occluded or dropped because of image noise. The algorithm took an average of 7.3s per step, most of which is used for recovering large parts of the covariance matrix, which can be sped up by using more advanced methods [25]. Without covariance recovery, the algorithm takes an average of 1.6s per step. Timing results were obtained on a laptop with 2.2GHz Core 2 Duo processor.

In terms of the overall loop closing, there are several correct correspondences between features in the last images of the trajectory and structure that was instantiated at the beginning of the loop. However, many features have not been correctly matched, and new structure created instead. We believe that the main problem for this is the insufficient modeling of feature appearance, as the view points are significantly different. What supports this belief is that most of the correctly matched structure appears in the last 3 frames after turning around the corner, where the view point is sufficiently close to the beginning of the trajectory, but at this point the duplicate structure has already been created.

Comparing to other methods, this result could not have been achieved with EKF based methods due to the large number of structure points that are recovered. In particular, MonoSLAM [6] is based on the EKF and uses the covariances directly for efficient data association of a small number of points. However, the EKF only works well up to about 100 points, and the squared computational complexity makes dealing with larger sets difficult. In contrast, our data contains over 500 points. However, the comparison is not completely fair, as we are also dealing with low frame rate data, while MonoSLAM runs at 30 frames per second, but with only small changes between frames. Another method that should be mentioned here is joint compatibility branch and bound (JCBB) [31], which also works well with smoothing based methods [46]. However, under high motion uncertainty, such as when closing large loops, the combinatorial complexity starts to take over at about 20 points per frame, as experienced in [46]. In contrast, our method successfully matched about 50 points per frame. There is of course a tradeoff: while JCBB finds the assignment that is optimal under some metric, our algorithm is not guaranteed to succeed. In our work, in particular, a weak wrong match can get reinforced over multiple EM iterations if not counteracted by other matches, pulling the state estimate in the wrong direction, and eventually leading to a confirmation of the wrong match. Also, EKF and JCBB might be applicable here if we restrict ourselves to recovering less structure points.

6. Conclusion and Future Work

We have presented visual SLAM with multiple cameras in a general, non-stereo setting. Our novel approach to the correspondence problem avoids the combinatorial complexity by using a sampling based approximation for the distribution over correspondences within an incremental expectation maximization framework. We have successfully demonstrated loop closing in a simulated example with significant measurement noise. We have also shown successful loop closing based on data acquired by a multi-camera rig on a mobile robot in an indoor office environment. Our eight camera rig allows robust operation because of the full 360 degree view. Operating with a single forward facing camera for example fails when turning in a hallway with untextured walls, as not enough features will be available. And even though dealing with multiple cameras adds some complications, the individual cameras provide more flexibility for distributed placement, such as in the front and back of a vehicle, than a single omni-direction camera that requires a central mounting point.

We have made use of odometry, but there is no fundamental reason why the proposed method could not work without. However, convergence will be slower if EM is started from a poor initial estimate for new variables. Furthermore, nonlinear optimization is generally susceptible to local minima, and our method makes no exception. However, a reasonable initial estimate might also be generated by a dynamic model of the vehicle.

For larger scale environments a more efficient method to recover the required parts of the structure and motion covariance is needed. We have recently presented this capability in iSAM [25] and plan to combine it with probabilistic structure matching. It then also becomes feasible to obtain positional uncertainty in order to restrict the search region to a manageable size. Furthermore, using a more discriminative feature descriptor would reduce the complexity of the problem by further restricting the search space of possible correspondences, therefore allowing to resolve ambiguities at a larger scale instead. Further generalizations of this work might include multi-robot applications, where the algorithm can be used to merge multiple maps, or to run in a distributed fashion.

Acknowledgments

This work was partially funded by the National Science Foundation under Grant No. IIS - 0448111. The authors would like to thank Drew Steedly for fruitful discussions, and the anonymous reviewers for their valuable comments that lead to a considerable improvement in this work.

Appendix

We show how the predictive measurement density $P(\mathbf{u}_k | J_m, \Theta^t) = P(\mathbf{u}_k | j_k, \Theta^t)$ from (14) is obtained by marginalization. As we are interested in an expression that

only deals with image measurements, we integrate out structure and motion

$$\begin{aligned}
& P(\mathbf{u}|j, \Theta^t) \\
&= \int_{\mathbf{m}\mathbf{x}} P(\mathbf{u}|\mathbf{m}, \mathbf{x})P(\mathbf{m}, \mathbf{x}|j, \Theta^t) \\
&\propto \int_{\mathbf{m}\mathbf{x}} \exp -\frac{1}{2} \left(\|\mathbf{u} - h(\mathbf{m}, \mathbf{x})\|_R^2 + \|\mathbf{s} - \mu\|_\Sigma^2 \right)
\end{aligned} \tag{28}$$

where μ and Σ are the mean and covariance of the parameters of $\mathbf{s} := (\mathbf{m}, \mathbf{x})^T$, which are part of Θ^t , and R is the image measurement uncertainty. In order to solve the integral in (28), we marginalize out \mathbf{s} . The negative log likelihood

$$f(\mathbf{u}, \mathbf{s}) = \frac{1}{2} \|\mathbf{u} - h(\mathbf{s})\|_R^2 + \|\mathbf{s} - \mu\|_\Sigma^2 \tag{29}$$

assumes its maximum for $\mathbf{s} = \mu$ and $\mathbf{u} = H\mu$, where H is the Jacobian of $h(\cdot)$ at \mathbf{s} . This can easily be confirmed from the partial derivatives $\frac{\partial}{\partial \mathbf{u}} f = R^{-1}(\mathbf{u} - H\mathbf{s})$ and $\frac{\partial}{\partial \theta} f = -H^T R^{-1}(\mathbf{u} - H\mathbf{s}) + \Sigma^{-1}(\mathbf{s} - \mu)$. The Hessian W for both variables $(\mathbf{u}, \mathbf{s})^T$ combined is given by

$$W = \begin{pmatrix} R^{-1} & -R^{-1}H \\ -H^T R^{-1} & H^T R^{-1}H + \Sigma^{-1} \end{pmatrix} \tag{30}$$

The Schur complement of the lower right block in W yields the marginal Hessian Ξ^{-1} in \mathbf{u} as

$$\Xi^{-1} = R^{-1} - R^{-1}H (H^T R^{-1}H + \Sigma^{-1})^{-1} H^T R^{-1} \tag{31}$$

where we have made use of $R^{-T} = R^{-1}$ as R is symmetric. By applying the matrix inversion lemma, we obtain a simple expression for the covariance Ξ in image measurement space

$$\Xi = H\Sigma H^T + R \tag{32}$$

Intuitively Ξ is the sum of the image measurement noise R and the combined structure and motion uncertainty Σ projected into the image space using the Jacobian H at $\mathbf{s} = \mu$, where

$$\Sigma = \begin{pmatrix} P & B \\ B^T & S \end{pmatrix} \tag{33}$$

with S the uncertainty in the structure point and P the uncertainty in the motion, and the off-diagonal entry B represents dependencies between structure and motion. Σ is obtained directly from the optimization process as explained in Section 3. We can now approximate the density in measurement space by a normal distribution with covariance Ξ

$$P(\mathbf{u}|\mathbf{s}) \approx \frac{1}{\sqrt{|2\pi\Xi|}} e^{-\frac{1}{2}\|\mathbf{u}-h(\mathbf{s})\|_\Xi^2} \tag{34}$$

References

- [1] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT press, Cambridge, MA, 2005.
- [2] A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Intl. Conf. on Computer Vision (ICCV), 2003, pp. 1403-1410.

- [3] N. Karlsson, E. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, M. Munich, The vSLAM algorithm for robust localization and mapping, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2005, pp. 24–29.
- [4] C. Engels, H. Stewénus, D. Nistér, Bundle adjustment rules, in: *Symposium on Photogrammetric Computer Vision*, 2006, pp. 266–271.
- [5] R. Eustice, H. Singh, J. Leonard, M. Walter, Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters, *Intl. J. of Robotics Research* 25 (12) (2006) 1223–1242.
- [6] A. Davison, I. Reid, N. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, *IEEE Trans. Pattern Anal. Machine Intell.* 29 (6) (2007) 1052–1067.
- [7] E. Eade, T. Drummond, Monocular SLAM as a graph of coalesced observations, in: *Intl. Conf. on Computer Vision (ICCV)*, 2007.
- [8] R. Sim, P. Elinas, J. Little, A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM, *Intl. J. of Computer Vision* 74 (3) (2007) 303–318.
- [9] S. Se, D. Lowe, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *Intl. J. of Robotics Research* 21 (8) (2002) 735–758.
- [10] L. Paz, P. Pinies, J. Tardós, J. Neira, 6DOF SLAM with stereo-in-hand, in: *IROS visual SLAM workshop*, 2007.
- [11] T. Marks, A. Howard, M. Bajracharya, G. Cottrell, L. Matthies, Gamma-SLAM: Stereo visual SLAM in unstructured environments using variance grid maps, in: *IROS visual SLAM workshop*, 2007.
- [12] K. Konolige, M. Agrawal, Frame-frame matching for realtime consistent visual mapping, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2007, pp. 2803–2810.
- [13] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, S. Teller, An Atlas framework for scalable mapping, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2003, pp. 1899–1906.
- [14] B. Mičušík, D. Martinec, T. Pajdla, 3D metric reconstruction from uncalibrated omnidirectional images, in: *Asian Conf. on Computer Vision (ACCV)*, 2004.
- [15] D. Burschka, G. Hager, V-GPS(SLAM): Vision-based inertial system for mobile robots, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2004, pp. 409–415.
- [16] T. Goedemé, M. Nuttin, T. Tuytelaars, L. V. Gool, Omnidirectional vision based topological navigation, *International Journal of Computer Vision. Special Issue: Joint Issue of IJCV and IJRR on Vision and Robotics* 74 (2007) 219–236.
- [17] R. Pless, Using many cameras as one, in: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, 2003, pp. 587–593.
- [18] A. Levin, R. Szeliski, Visual odometry and map correlation, in: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [19] J. Sola, A. Monin, M. Devy, T. Vidal-Calleja, Fusing monocular information in multicamera SLAM, *IEEE Trans. Robotics* 24 (5) (2008) 958–968.
- [20] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [21] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle adjustment – a modern synthesis, in: W. Triggs, A. Zisserman, R. Szeliski (Eds.), *Vision Algorithms: Theory and Practice*, LNCS, Springer Verlag, 1999, pp. 298–375.
- [22] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Real time localization and 3d reconstruction, in: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [23] F. Dellaert, M. Kaess, Square Root SAM: Simultaneous localization and mapping via square root information smoothing, *Intl. J. of Robotics Research* 25 (12) (2006) 1181–1203.
- [24] K. Konolige, M. Agrawal, FrameSLAM: From bundle adjustment to real-time visual mapping, *IEEE Trans. Robotics* 24 (2008) 1066–1077.
- [25] M. Kaess, A. Ranganathan, F. Dellaert, iSAM: Incremental smoothing and mapping, *IEEE Trans. Robotics* 24 (6) (2008) 1365–1378.
- [26] R. Bolles, M. Fischler, A RANSAC-based approach to model fitting and its application to finding cylinders in range data, in: *Intl. Joint Conf. on AI (IJCAI)*, Vancouver, BC, Canada, 1981, pp. 637–643.
- [27] P. Beardsley, P. Torr, A. Zisserman, 3D model acquisition from extended image sequences, in: *Eur. Conf. on Computer Vision (ECCV)*, 1996, pp. II:683–695.
- [28] J.-S. Gutmann, K. Konolige, Incremental mapping of large cyclic environments, in: *IEEE Intl.*

- Symp. on Computational Intelligence in Robotics and Automation (CIRA), 1999, pp. 318–325.
- [29] P. Besl, N. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Machine Intell.* 14 (2).
 - [30] H. Baltzakis, P. Trahanias, Closing multiple loops while mapping features in cyclic environments, in: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003, pp. 717–722.
 - [31] J. Neira, J. Tardos, Data association in stochastic mapping using the joint compatibility test, *IEEE Trans. Robot. Automat.* 17 (6) (2001) 890–897.
 - [32] D. Hähnel, W. Burgard, B. Wegbreit, S. Thrun, Towards lazy data association in SLAM, in: *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, Springer, Sienna, Italy, 2003.
 - [33] D. Hähnel, W. Burgard, D. Fox, S. Thrun, A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements, in: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003, pp. 206–211.
 - [34] M. Montemerlo, S. Thrun, Simultaneous localization and mapping with unknown data association using FastSLAM, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2003.
 - [35] J. Nieto, H. Guivant, E. Nebot, S. Thrun, Real time data association for FastSLAM, in: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2003.
 - [36] A. Eliazar, R. Parr, DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, in: *Intl. Joint Conf. on AI (IJCAI)*, 2003.
 - [37] F. Dellaert, S. Seitz, C. Thorpe, S. Thrun, EM, MCMC, and chain flipping for structure from motion with unknown correspondence, *Machine learning* 50 (1-2) (2003) 45–71, special issue on Markov chain Monte Carlo methods.
 - [38] C. Harris, M. Stephens, A combined corner and edge detector, *Proceedings of the 4th Alvey Vision Conference* (1988) 147–151.
 - [39] M. Kaess, F. Dellaert, Visual SLAM with a multi-camera rig, *Tech. Rep. GIT-GVU-06-06*, Georgia Institute of Technology (Feb 2006).
 - [40] F. Dellaert, The expectation maximization algorithm, *Tech. Rep. GIT-GVU-02-20*, College of Computing, Georgia Institute of Technology (February 2002).
 - [41] G. McLachlan, T. Krishnan, *The EM algorithm and extensions*, Wiley series in probability and statistics, John Wiley & Sons, 1997.
 - [42] T. Minka, Expectation-Maximization as lower bound maximization, tutorial published on the web at <http://www-white.media.mit.edu/~tpminka/papers/em.html> (November 1998).
 - [43] M. Tanner, *Tools for Statistical Inference*, Springer Verlag, New York, 1996, third Edition.
 - [44] W. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* 57 (1970) 97–109.
 - [45] P. Krauthausen, F. Dellaert, A. Kipp, Exploiting locality by nested dissection for square root smoothing and mapping, in: *Robotics: Science and Systems (RSS)*, 2006.
 - [46] M. Kaess, F. Dellaert, Covariance recovery from a square root information matrix for data association, *Journal of Robotics and Autonomous Systems*. To appear.
 - [47] T. Davis, J. Gilbert, S. Larimore, E. Ng, A column approximate minimum degree ordering algorithm, *ACM Trans. Math. Softw.* 30 (3) (2004) 353–376.
 - [48] T. A. Davis, Algorithm 8xx: a concise sparse Cholesky factorization package, *Tech. Rep. TR-04-001*, Univ. of Florida, submitted to *ACM Trans. Math. Software*. (January 2004).
 - [49] A. Griewank, On Automatic Differentiation, in: M. Iri, K. Tanabe (Eds.), *Mathematical Programming: Recent Developments and Applications*, Kluwer Academic Publishers, 1989, pp. 83–108.