# C-KLAM: Constrained Keyframe-Based Localization and Mapping

Esha D. Nerurkar[†], Kejian J. Wu[‡], and Stergios I. Roumeliotis[†]

***Abstract*—In this paper, we present C-KLAM, a Maximum A Posteriori (MAP) estimator-based keyframe approach for SLAM. As opposed to many existing keyframe-based SLAM approaches, that discard information from non-keyframes for reducing the computational complexity, the proposed C-KLAM presents a novel, elegant, and computationally-efficient technique for incorporating most of this information, resulting in improved estimation accuracy. To achieve this, C-KLAM projects both proprioceptive and exteroceptive information from the non-keyframes to the keyframes, using marginalization, while *maintaining the sparse structure of the associated information matrix*, resulting in fast and efficient solutions. The performance of C-KLAM has been tested in both simulations and experimentally, using visual and inertial measurements, to demonstrate that it achieves performance comparable to that of the computationally-intensive batch MAP-based 3D SLAM, that uses all available measurement information.**

## I. Introduction and Related Work

For mobile robots navigating in large environments over long time periods, one of the main challenges in designing an estimation algorithm for Simultaneous Localization and Mapping (SLAM) is its inherently high computational complexity. For example, the computational complexity of the Minimum Mean Squared Error (MMSE) estimator for SLAM, i.e., the Extended Kalman filter [15], is $O(N^2)$ at each time step, where $N$ is the number of landmarks in the map. Similarly, for the batch Maximum A Posteriori (MAP) estimator-based SLAM (smoothing and mapping) [1], the worst-case computational complexity is $O([K+N]^3)$, where $K$ is the number of robot poses in the trajectory. While existing batch MAP-based SLAM approaches such as the $\sqrt{SAM}$ [1], $g^2o$ [10], and SPA [9] generate efficient solutions by exploiting the sparsity of the information matrix, for large-scale SLAM with frequent loop closures, this cost eventually prohibits real-time operation.

The approximate solutions developed to reduce MAP-based SLAM's computational complexity can be classified into three main categories. The first category of approaches such as iSAM [4] and iSAM2 [5] *incrementally* optimize over all robot poses and landmarks, using *all* available measurement information. However, for trajectories with frequent loop closures, (i) fill-ins are generated between periodic batch updates for iSAM, when the number of constraints is greater than five times the number of robot poses [4], and (ii) many nodes in the Bayes tree used by iSAM2 have to be relinearized, hence degrading the performance of these approaches.

The second category includes fixed-lag smoothing approaches such as [12, 14] that consider a constant-size, sliding-window of recent robot poses and landmarks, along with measurements only in that time window. Here, old robot poses and landmarks are *marginalized* and the corresponding measurements are discarded. However, marginalization destroys the sparsity of the information matrix, and the cost of this approach becomes $O(R^3)$, hence limiting the number of poses, $R$, in the sliding window. Moreover, this approach is unable to close loops for long trajectories.

The third category consists of *keyframe*-based approaches, such as PTAM [6], FrameSLAM [7], and view-based maps (pose graphs) [8], [2], [3] that process measurement information from only a *subset* of all available views/keyframes/robot poses. Here, information from non-keyframes is *discarded* (as opposed to marginalized) in order to retain the sparsity of the information matrix, hence trading estimation accuracy for reduced computational cost.

In this paper, we present the Constrained Keyframe-based Localization and Mapping (C-KLAM), an approximate batch MAP-based algorithm, which estimates only keyframes (key robot poses) and key landmarks while also exploiting information (e.g., visual observations and odometry measurements) available to the non-keyframes. In particular, this information is projected onto the keyframes, by generating pose constraints between them. Our main contributions are as follows:

- In contrast to existing keyframe methods, C-KLAM utilizes both proprioceptive [e.g., inertial measurement unit (IMU)] and exteroceptive (e.g., camera) measurements from non-keyframes to generate pose constraints between the keyframes. This is achieved by marginalizing the non-keyframes along with the landmarks observed from them.
- In contrast to sliding-window approaches, C-KLAM incorporates information from marginalized frames and landmarks *without* destroying the sparsity of the information matrix, and hence generates fast and efficient solutions.
- The cost of marginalization in C-KLAM is cubic, $O(M^3)$, only in the number of non-keyframes, $M$, between consecutive keyframes, and *linear* in the number of

[†]E. D. Nerurkar, and S. I. Roumeliotis are with the Department of Computer Science and Engineering, Univ. of Minnesota, Minneapolis, USA {nerurkar,stergios}@cs.umn.edu

[‡]K. J. Wu is with the Department of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, USA kejian@cs.umn.edu

This work was supported by the University of Minnesota (UMN) through the Digital Technology Center (DTC) and AFOSR (FA9550-10-1-0567). E. D. Nerurkar was supported by the UMN Doctoral Dissertation Fellowship. The authors thank Chao X. Guo and Dimitrios G. Kottas for their help with the experimental datasets.

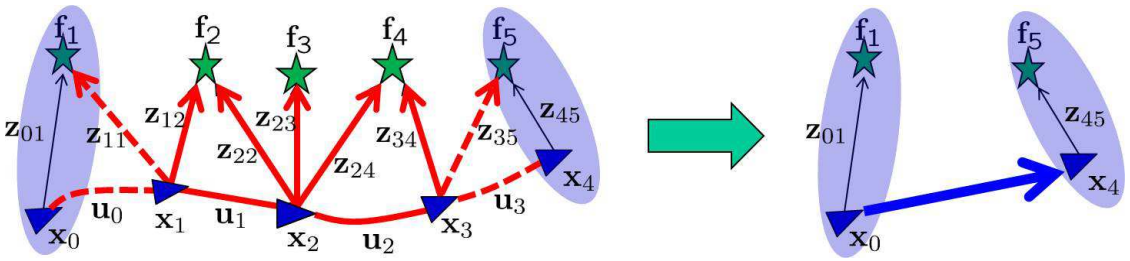


Fig. 1. An example of the exploration epoch before (left) and after (right) the approximation employed in C-KLAM. $\mathbf{x}_0$, $\mathbf{x}_4$ are the keyframes to be retained, and $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are the non-keyframes to be marginalized. Similarly, $\mathbf{f}_1$, $\mathbf{f}_5$ are key landmarks (observed from the keyframes) to be retained, while $\mathbf{f}_2$, $\mathbf{f}_3$, and $\mathbf{f}_4$ are non-key landmarks (observed exclusively from the non-keyframes) to be marginalized. In the left figure, the arrows denote the measurements between different states. In the right figure, the blue arrow represents the pose constraint generated between the keyframes using C-KLAM.

landmarks, $F_M$, observed exclusively from the $M$ non-keyframes, where $M \ll F_M$.

- The keyframes and the associated landmark-map are maintained over the entire robot trajectory, and thus C-KLAM enables efficient loop closures, necessary for ensuring accurate and consistent long-term navigation.

## II. Algorithm Description

In this section, we first present a brief overview of batch MAP-based SLAM, followed by the details of the proposed C-KLAM algorithm. Moreover, to facilitate the description of these estimation algorithms, we will use the specific example scenario depicted in Fig. 1. Note, however, that C-KLAM is a general approach that can be used for any number of key and non-key poses and landmarks.

### A. Batch MAP-based SLAM

Consider a robot, equipped with proprioceptive (e.g., IMU) and exteroceptive (e.g., camera) sensors, navigating in a 3D environment. The motion model for the robot is given by:

$$\mathbf{x}_{i+1} = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i - \mathbf{w}_i) \tag{1}$$

where $\mathbf{f}$ is a general nonlinear function[1], $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ denote the robot poses at time-steps $i$ and $i+1$, respectively, $\mathbf{u}_i = \mathbf{u}_{i_t} + \mathbf{w}_i$, is the measured control input (linear acceleration and rotational velocity), where $\mathbf{u}_{i_t}$ denotes the true control input, and $\mathbf{w}_i$ is the zero-mean, white Gaussian measurement noise with covariance $\mathbf{Q}_i$. The measurement model for the robot at time-step $i$, obtaining an observation, $\mathbf{z}_{ij}$, to landmark $\mathbf{f}_j$ is given by:

$$\mathbf{z}_{ij} = \mathbf{h}(\mathbf{x}_i, \mathbf{f}_j) + \mathbf{v}_{ij} \tag{2}$$

where $\mathbf{h}$ is a general nonlinear measurement function[2] and $\mathbf{v}_{ij}$ is the zero-mean, white Gaussian measurement noise with covariance $\mathbf{R}_{ij}$.

Consider the current exploration epoch shown in Fig. 1, consisting of five robot poses, $\mathbf{x}_i$, $i = 0, 1, \ldots, 4$, and of five point landmarks' positions, $\mathbf{f}_j$, $j = 1, 2, \ldots, 5$, observed from these poses. The batch MAP estimates, $\hat{\mathbf{x}}_{0:4}^{MAP}, \hat{\mathbf{f}}_{1:5}^{MAP}$, of *all* robot poses, $\mathbf{x}_{0:4}$, and *all* landmark positions, $\mathbf{f}_{1:5}$, using *all* available proprioceptive, $\mathbf{u}_{0:3}$, and exteroceptive, $\mathcal{Z}_{0:4}$, measurements is given by:

$$\hat{\mathbf{x}}_{0:4}^{MAP}, \hat{\mathbf{f}}_{1:5}^{MAP} \triangleq \arg\max_{\mathbf{x}_{0:4}, \mathbf{f}_{1:5}} p(\mathbf{x}_{0:4}, \mathbf{f}_{1:5} | \mathcal{Z}_{0:4}, \mathbf{u}_{0:3}) \tag{3}$$

where $\mathcal{Z}_i$ denotes the set of all exteroceptive measurements obtained at robot pose $\mathbf{x}_i$, $i = 0, 1, \ldots, 4$. Under the Gaussian and independent noise assumptions in (1) and (2), (3) is equivalent to minimizing the following nonlinear least-squares cost function:

$$\begin{aligned}
&\mathcal{C}(\mathbf{x}_{0:4}, \mathbf{f}_{1:5}; \mathcal{Z}_{0:4}, \mathbf{u}_{0:3}) \\
&= \frac{1}{2}||\mathbf{x}_0 - \hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}} + \sum_{i=0}^{3} \frac{1}{2}||\mathbf{x}_{i+1} - \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i)||^2_{\mathbf{Q}'_i} \\
&\quad + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:4}} \frac{1}{2}||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{x}_i, \mathbf{f}_j)||^2_{\mathbf{R}_{ij}} \\
&\triangleq \mathcal{C}_P(\mathbf{x}_0; \hat{\mathbf{x}}_{0|0}) + \sum_{i=0}^{3} \mathcal{C}_M(\mathbf{x}_{i+1}, \mathbf{x}_i; \mathbf{u}_i) \\
&\quad + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{0:4}} \mathcal{C}_O(\mathbf{x}_i, \mathbf{f}_j; \mathbf{z}_{ij})
\end{aligned} \tag{4}$$

where $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0})$ denotes the prior for the robot pose, $\mathbf{Q}'_i = \mathbf{G}_i \mathbf{Q}_i \mathbf{G}_i^T$, and $\mathbf{G}_i$ is the Jacobian of $\mathbf{f}$ with respect to the noise $\mathbf{w}_i$. In what follows, we denote the cost terms arising from the prior, the robot motion, and the landmark measurements by $\mathcal{C}_P$, $\mathcal{C}_M$, and $\mathcal{C}_O$, respectively [see (4)].

A standard approach for minimizing (4) is to employ the Gauss-Newton iterative minimization algorithm [16] with computational complexity up to $O([K+N]^3)$, where $K$ and $N$ denote the number of robot poses and landmarks, respectively. Note that, as the robot explores the environment and observes new landmarks, the size of the optimization problem (both $K$ and $N$) in (4) continuously increases. Therefore, for long trajectories with many features and frequent loop closures, the cost of solving (4) may prohibit real-time operation.

### B. C-KLAM Algorithm

In order to reduce the computational complexity of MAP-based SLAM and ensure accurate and real-time navigation over long time durations, the proposed C-KLAM approach

[1]The details of the IMU motion model can be found in [12].

[2]The details of the camera measurement model for point features, used in our experiments, can be found in [12].

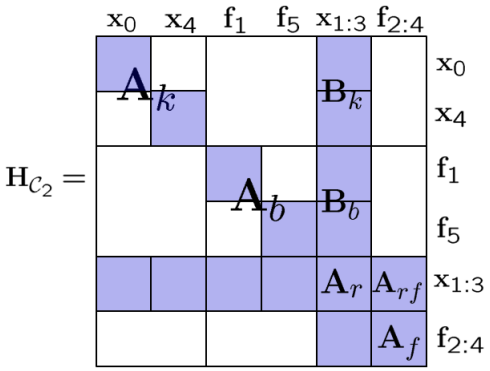


Fig. 2. Structure of the sparse symmetric positive definite information (Hessian) matrix corresponding to the cost function $\mathcal{C}_2$ in (5) (measurements shown with red arrows in Fig. 1). The colored blocks denote non-zero elements. The block-diagonal sub-matrices $\mathbf{A}_k$ and $\mathbf{A}_b$ correspond to key poses and key landmarks, respectively. $\mathbf{A}_r$ and $\mathbf{A}_f$ correspond to non-key poses and non-key landmarks to be marginalized, respectively. Here $\mathbf{A}_k$ and $\mathbf{A}_r$ are, in general, block tri-diagonal, while $\mathbf{A}_b$ and $\mathbf{A}_f$ are block diagonal.

(i) builds a sparse map of the environment consisting of *only* the key robot poses and the distinctive landmarks observed from these key poses[3], and (ii) uses measurement information from non-key poses to create constraints between the key poses, in order to improve estimation accuracy.

Specifically, for the example in Fig. 1, let us assume that we retain: (i) $\mathbf{x}_0$ and $\mathbf{x}_4$ as key poses, and (ii) landmarks, $\mathbf{f}_1$ and $\mathbf{f}_5$, observed from these key poses as key landmarks[4]. In this case, (4) can be split into two parts as follows:

$$\mathcal{C} = \underbrace{\mathcal{C}_P(\mathbf{x}_0; \hat{\mathbf{x}}_{0|0}) + \mathcal{C}_O(\mathbf{x}_0, \mathbf{f}_1; \mathbf{z}_{01}) + \mathcal{C}_O(\mathbf{x}_4, \mathbf{f}_5; \mathbf{z}_{45})}_{\mathcal{C}_1(\mathbf{x}_0, \mathbf{x}_4, \mathbf{f}_1, \mathbf{f}_5; \hat{\mathbf{x}}_{0|0}, \mathbf{z}_{01}, \mathbf{z}_{45})} + \underbrace{\sum_{i=0}^{3} \mathcal{C}_M(\mathbf{x}_{i+1}, \mathbf{x}_i; \mathbf{u}_i) + \sum_{\mathbf{z}_{ij} \in \mathcal{Z}_{1:3}} \mathcal{C}_O(\mathbf{x}_i, \mathbf{f}_j; \mathbf{z}_{ij})}_{\mathcal{C}_2(\mathbf{x}_{1:3}, \mathbf{f}_{2:4}, \mathbf{x}_0, \mathbf{x}_4, \mathbf{f}_1, \mathbf{f}_5; \mathcal{Z}_{1:3}, \mathbf{u}_{0:3})} \quad (5)$$

The first part of the cost function, $\mathcal{C}_1$, depends only upon the key poses, key landmarks, and the measurements between them (denoted by thin black arrows in Fig. 1). This part consists of cost terms arising from the prior term and from the two exteroceptive measurements, $\mathbf{z}_{01}$ and $\mathbf{z}_{45}$, obtained at the key poses $\mathbf{x}_0$ and $\mathbf{x}_4$, respectively. The second part of the cost function, $\mathcal{C}_2$, contains all cost terms that involve non-key poses and non-key landmarks. Specifically, these correspond to two types of cost terms: (i) terms that involve *only* non-key poses and non-key landmarks (corresponding to measurements denoted by solid red lines in Fig. 1), e.g., $\mathcal{C}_O(\mathbf{x}_1, \mathbf{f}_2; \mathbf{z}_{12})$, and (ii) terms that involve *both* key and non-key elements (corresponding to measurements denoted by dashed red lines in Fig. 1), e.g., $\mathcal{C}_O(\mathbf{x}_1, \mathbf{f}_1; \mathbf{z}_{11})$ and $\mathcal{C}_M(\mathbf{x}_1, \mathbf{x}_0; \mathbf{u}_0)$.

Before we proceed, we note that keyframe-based approaches optimize only over $\mathcal{C}_1$ in order to reduce the computational complexity, i.e., the cost terms in $\mathcal{C}_2$ and the corresponding measurements are discarded, resulting in significant information loss. An alternative and straightforward approach that retains part of the information in $\mathcal{C}_2$, is to *marginalize* the non-key poses and landmarks, $\mathbf{x}_{1:3}$ and $\mathbf{f}_{2:4}$, respectively. Mathematically, this is equivalent to approximating $\mathcal{C}_2$ as follows (see Fig. 2):

$$\begin{aligned} \mathcal{C}_2 &\simeq \mathcal{C}_2'(\mathbf{x}_0, \mathbf{x}_4, \mathbf{f}_1, \mathbf{f}_5; \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_4, \hat{\mathbf{f}}_1, \hat{\mathbf{f}}_5) \\ &= \alpha' + \mathbf{g}_{\mathcal{C}_2'}^T \begin{bmatrix} \mathbf{x}_0 - \hat{\mathbf{x}}_0 \\ \mathbf{x}_4 - \hat{\mathbf{x}}_4 \\ \mathbf{f}_1 - \hat{\mathbf{f}}_1 \\ \mathbf{f}_5 - \hat{\mathbf{f}}_5 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{x}_0 - \hat{\mathbf{x}}_0 \\ \mathbf{x}_4 - \hat{\mathbf{x}}_4 \\ \mathbf{f}_1 - \hat{\mathbf{f}}_1 \\ \mathbf{f}_5 - \hat{\mathbf{f}}_5 \end{bmatrix}^T \mathbf{H}_{\mathcal{C}_2'} \begin{bmatrix} \mathbf{x}_0 - \hat{\mathbf{x}}_0 \\ \mathbf{x}_4 - \hat{\mathbf{x}}_4 \\ \mathbf{f}_1 - \hat{\mathbf{f}}_1 \\ \mathbf{f}_5 - \hat{\mathbf{f}}_5 \end{bmatrix} \end{aligned} \quad (6)$$

with,

$$\mathbf{H}_{\mathcal{C}_2'} = \begin{bmatrix} \mathbf{A}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_b \end{bmatrix} - \begin{bmatrix} \mathbf{B}_k & \mathbf{0} \\ \mathbf{B}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A}_r & \mathbf{A}_{rf} \\ \mathbf{A}_{fr} & \mathbf{A}_f \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{B}_k^T & \mathbf{B}_b^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (7)$$

$$\mathbf{g}_{\mathcal{C}_2'} = \begin{bmatrix} \mathbf{g}_k \\ \mathbf{g}_b \end{bmatrix} - \begin{bmatrix} \mathbf{B}_k & \mathbf{0} \\ \mathbf{B}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{A}_r & \mathbf{A}_{rf} \\ \mathbf{A}_{fr} & \mathbf{A}_f \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{g}_r \\ \mathbf{g}_f \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{g}_{\mathcal{C}_2',k} \\ \mathbf{g}_{\mathcal{C}_2',b} \end{bmatrix} \quad (8)$$

where $\hat{\mathbf{x}}_0$, $\hat{\mathbf{x}}_4$, $\hat{\mathbf{f}}_1$, and $\hat{\mathbf{f}}_5$ are the estimates of $\mathbf{x}_0$, $\mathbf{x}_4$, $\mathbf{f}_1$, and $\mathbf{f}_5$, respectively, at the time of marginalization, $\alpha'$ is a constant term independent of the optimization variables, and $\mathbf{g}_k, \mathbf{g}_b, \mathbf{g}_r$, and $\mathbf{g}_f$ are the gradient vectors of $\mathcal{C}_2$ with respect to $\{\mathbf{x}_0, \mathbf{x}_4\}, \{\mathbf{f}_1, \mathbf{f}_5\}, \{\mathbf{x}_{1:3}\}$, and $\{\mathbf{f}_{2:4}\}$, respectively. Also, $\mathbf{g}_{\mathcal{C}_2'}$ and $\mathbf{H}_{\mathcal{C}_2'}$ denote the Jacobian and Hessian matrix, respectively. Lastly, we note that $\mathbf{H}_{\mathcal{C}_2'}$, as expected, is the Schur complement of the diagonal block, corresponding to non-key poses and non-key landmarks, of the Hessian, $\mathbf{H}_{\mathcal{C}_2}$, of the original cost function, $\mathcal{C}_2$ (see Fig. 2).

However, as expected, the marginalization of non-key poses, $\mathbf{x}_1$ and $\mathbf{x}_3$, due to their observations of key landmarks $\mathbf{f}_1$ and $\mathbf{f}_5$, creates additional constraints between the key poses and the key landmarks. This directly translates into fill-ins in the reduced Hessian matrix, $\mathbf{H}_{\mathcal{C}_2'}$, hence destroying the sparse structure of the original Hessian, $\mathbf{H}_{\mathcal{C}_2}$, and increasing the computational cost of obtaining a solution to the corresponding minimization problem.

In order to address this problem and maintain the sparse structure of the Hessian (information) matrix while incorporating information from $\mathcal{C}_2$, C-KLAM carries out an additional approximation step, i.e., it further approximates $\mathcal{C}_2'$ by a quadratic cost term, $\mathcal{C}_2''(\mathbf{x}_0, \mathbf{x}_4; \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_4)$ that constraints *only* the key poses $\mathbf{x}_0$ and $\mathbf{x}_4$. Specifically, along with the non-key poses/landmarks, C-KLAM *marginalizes the key landmarks* $\mathbf{f}_1$ *and* $\mathbf{f}_5$, but *only* from $\mathcal{C}_2$. At this point, we should note that these key landmarks still appear as optimization variables in $\mathcal{C}_1$ [see (5)]. Moreover, marginalizing $\mathbf{f}_1$ and $\mathbf{f}_5$ from $\mathcal{C}_2$, while retaining them in $\mathcal{C}_1$, implies that we ignore their data

[3]The terms key poses and keyframes are used interchangeably in this paper.

[4]Note that we retain only two key poses/landmarks in this example, in order to simplify the explanation. However, C-KLAM can be used to retain any number of key poses/landmarks. Furthermore, for the example in Fig. 1, we assume that the depth to the features is available (e.g., from an RGB-D camera), in order to reduce the number of measurements and poses required. However, if a regular camera is used, at least two observations of a key feature and the corresponding poses will need to be retained.

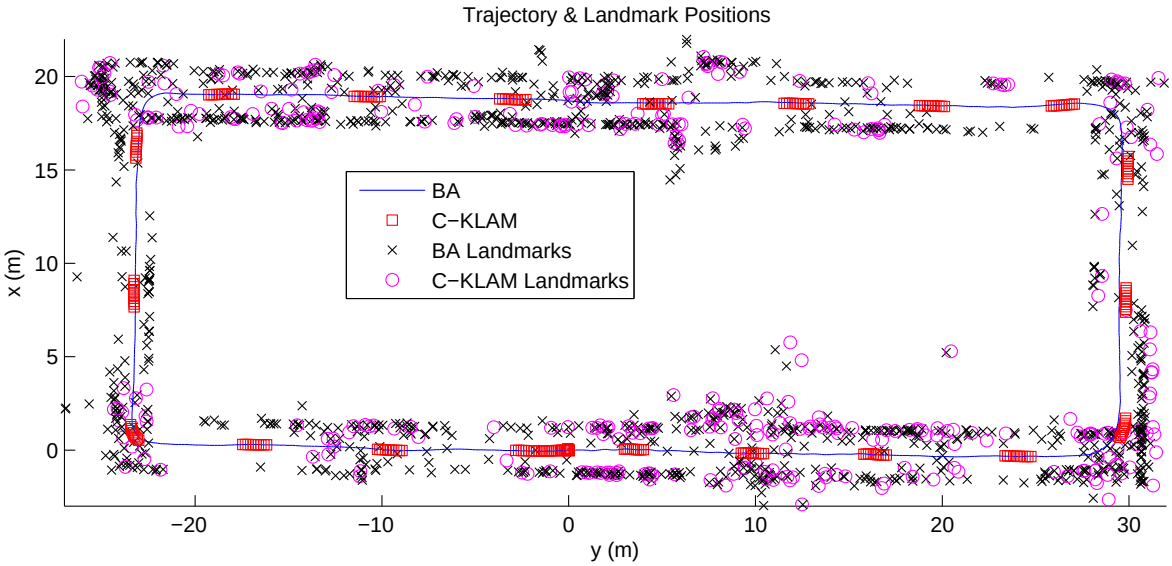


Fig. 3. Overhead $x-y$ view of the estimated 3D trajectory and landmark positions. The C-KLAM estimates only keyframes (marked with red squares) and key features (marked with magenta circles), while BA estimates the entire trajectory (marked by black line) and all features (marked by black x-s).

association[5] and treat them as different features (say $\mathbf{f}_1'$ and $\mathbf{f}_5'$) in $\mathcal{C}_2$. Under this implicit assumption, C-KLAM approximates $\mathcal{C}_2$ by [see (6) - (8), and Fig. 2]:

$$\begin{aligned}\mathcal{C}_2 &\simeq \mathcal{C}_2'(\mathbf{x}_0,\mathbf{x}_4,\mathbf{f}_1',\mathbf{f}_5';\hat{\mathbf{x}}_0,\hat{\mathbf{x}}_4,\hat{\mathbf{f}}_1,\hat{\mathbf{f}}_5) \simeq \mathcal{C}_2''(\mathbf{x}_0,\mathbf{x}_4;\hat{\mathbf{x}}_0,\hat{\mathbf{x}}_4)\\ &= \alpha'' + \mathbf{g}_{\mathcal{C}_2''}^T \begin{bmatrix}\mathbf{x}_0-\hat{\mathbf{x}}_0\\ \mathbf{x}_4-\hat{\mathbf{x}}_4\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\mathbf{x}_0-\hat{\mathbf{x}}_0\\ \mathbf{x}_4-\hat{\mathbf{x}}_4\end{bmatrix}^T \mathbf{H}_{\mathcal{C}_2''}\begin{bmatrix}\mathbf{x}_0-\hat{\mathbf{x}}_0\\ \mathbf{x}_4-\hat{\mathbf{x}}_4\end{bmatrix}\end{aligned} \tag{9}$$

with,

$$\mathbf{H}_{\mathcal{C}_2''} = \mathbf{A}_k - \mathbf{B}_k(\mathbf{D}-\mathbf{B}_b^T\mathbf{A}_b^{-1}\mathbf{B}_b)^{-1}\mathbf{B}_k^T \tag{10}$$

$$\begin{aligned}\mathbf{g}_{\mathcal{C}_2''} &= \mathbf{g}_{\mathcal{C}_2',k} + \mathbf{B}_k\mathbf{D}^{-1}\mathbf{B}_b^T\\ &\quad\cdot(\mathbf{A}_b^{-1}+\mathbf{A}_b^{-1}\mathbf{B}_b(\mathbf{D}-\mathbf{B}_b^T\mathbf{A}_b^{-1}\mathbf{B}_b)^{-1}\mathbf{B}_b^T\mathbf{A}_b^{-1})\mathbf{g}_{\mathcal{C}_2',b}\end{aligned} \tag{11}$$

and

$$\mathbf{D} = \mathbf{A}_r - \mathbf{A}_{rf}\mathbf{A}_f^{-1}\mathbf{A}_{fr}. \tag{12}$$

where $\alpha''$ is a constant, independent of the optimization variables, and $\mathbf{g}_{\mathcal{C}_2''}$, $\mathbf{H}_{\mathcal{C}_2''}$ denote the Jacobian and Hessian matrix, respectively.

After this approximation, the final C-KLAM cost function becomes:

$$\begin{aligned}\mathcal{C}_{CKLAM} &= \mathcal{C}_1(\mathbf{x}_0,\mathbf{x}_4,\mathbf{f}_1,\mathbf{f}_5;\hat{\mathbf{x}}_{0|0},\mathbf{z}_{01},\mathbf{z}_{45})\\ &\quad+\mathcal{C}_2''(\mathbf{x}_0,\mathbf{x}_4;\hat{\mathbf{x}}_0,\hat{\mathbf{x}}_4)\end{aligned} \tag{13}$$

whose corresponding Hessian would be the same as that of $\mathcal{C}_1$ (and thus sparse) with an additional link between $\mathbf{x}_0$ and $\mathbf{x}_1$ due to $\mathcal{C}_2''$. In summary, by approximating $\mathcal{C}_2$ by $\mathcal{C}_2''$, C-KLAM is able to incorporate most of the information from the non-key poses/landmarks, while maintaining the sparsity of the Hessian matrix. Moreover, the part of the cost function, $\mathcal{C}_1$, corresponding to the key poses/landmarks, remains intact.

Lastly, we show that the approximation (marginalization) described above can be carried out with cost cubic in the number of marginalized non-key poses, and only linear in the number of marginalized non-key landmarks. For computing the Hessian, $\mathbf{H}_{\mathcal{C}_2''}$, note that both $\mathbf{A}_b$ and $\mathbf{A}_f$ [see (10), (12)] are block diagonal and hence their inverses can be calculated with cost linear in the number of corresponding landmarks. The most computationally-intensive calculation in (10) is that of $(\mathbf{D}-\mathbf{B}_b^T\mathbf{A}_b^{-1}\mathbf{B}_b)^{-1}$, which is cubic in the number of non-key poses currently being marginalized. Since this size is bounded, the marginalization in C-KLAM can be carried out with minimal computational overhead. Note that the analysis for the cost of computing the Jacobian, $\mathbf{g}_{\mathcal{C}_2''}$, is similar.

## III. Experimental and Simulation Results

### A. Experimental Results

The experimental setup consists of a PointGrey Chameleon camera and a Navchip IMU, rigidly attached on a light-weight (100 g) platform. The IMU signals were sampled at a frequency of 100 Hz while camera images were acquired at 7.5 Hz. SIFT features [11] were detected in the camera images and matched using a vocabulary tree [13]. The experiment was conducted in an indoor environment where the sensor platform performed a 3D rectangular trajectory, with a total length of 144 m and returned back to the initial position in order to provide an estimate of the final position error.

In the C-KLAM implementation, the corresponding approximate batch MAP optimization problem was solved every 20 incoming camera frames. The exploration epoch was set to 60 camera frames, from which the first and last 10 con-

[5]Besides the inability to relinearize marginalized states, ignoring this data association is the main information loss incurred by C-KLAM as compared to the batch MAP-based SLAM.

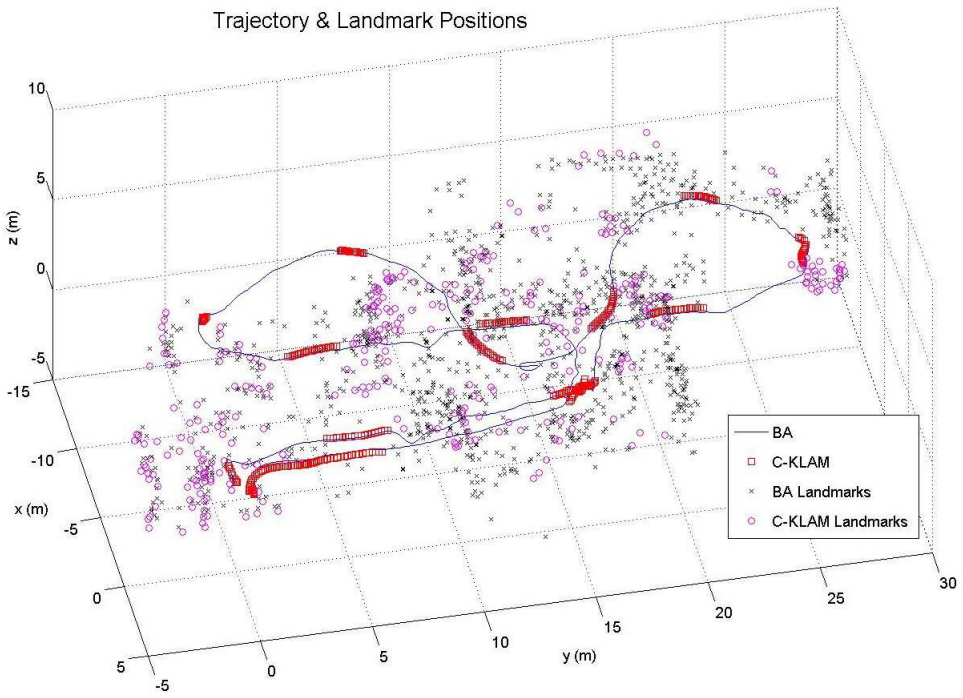


Fig. 4. 3D view of the estimated trajectory and landmark positions for the AR.Drone experiment. C-KLAM estimates only keyframes (marked with red squares) and key features (marked with magenta circles) while BA estimates the entire trajectory (marked by black line) and all features (marked by x-s).

secutive camera frames were retained as keyframes, while the rest were marginalized using the C-KLAM algorithm. We compared the performance of C-KLAM to that of the computationally-intensive, batch MAP-based SLAM [bundle adjustment (BA)], which optimizes over all camera poses and landmarks, using all available measurements, to provide high-accuracy estimates. In the BA implementation, the batch MAP optimization problem was solved every 20 incoming camera frames.

Fig. 3 shows the $x - y$ view of the estimated trajectory and landmark positions. As evident, the estimates of the robot trajectory and landmark positions generated by C-KLAM are almost identical to those of the BA. Loop closure was performed and the final position error was 7 cm for C-KLAM, only 5% more than that of the BA.

In terms of speed, the C-KLAM algorithm took only 4% of the time required for the entire BA. At the end of this experiment, C-KLAM retained 238 keyframes and 349 key landmarks, while BA had 1038 camera frames and 1281 landmarks. This significant reduction in the number of estimated states in C-KLAM led to substantial improvement in efficiency. Moreover, by using information from non-keyframes to constrain the keyframes, C-KLAM was able to achieve estimation performance comparable to that of the BA.

Another experiment was conducted using the same IMU-camera sensor package mounted on a Parrot AR.Drone quadrotor, flying in an indoor environment with a total trajectory length of 126 m. However, in this experiment, the drone did not return to the exact starting position and there were no loop closures. In the C-KLAM implementation, the resulting optimization problem was solved every 20 incoming camera frames. The exploration epoch was set to 100 camera frames, from which the first and last 20 consecutive camera frames were retained as keyframes, while the rest were marginalized using the C-KLAM algorithm. At the end of the experiment, C-KLAM retained 330 keyframes and 348 key landmarks, compared to 1110 camera poses and 1083 landmarks in BA.

Fig. 4 shows the estimated 3D trajectory and landmarks for both BA and C-KLAM. From the figure, we see that, similar to the previous experiment, the estimates of the robot trajectory and landmark positions generated by C-KLAM almost coincide with those generated by the BA, although no loop closure was performed in either C-KLAM or BA. Since the quadrotor did not return to the exact starting position, the final position error cannot be determined for this experiment. However, the difference between the final position estimates of BA and C-KLAM was 0.4% of the length of the total trajectory.

### *B. Simulation Results*

The performance of C-KLAM was extensively tested in simulations for a variety of conditions. The simulation results corroborate our experimental results, both in terms of the accuracy and speed of C-KLAM. However, due to space limitations, we present Monte-Carlo results for a single simulation setup. In particular, the IMU-camera platform traversed a helical trajectory of radius 5 m at an average velocity of 0.6 m/s and the camera observed features distributed on the

interior wall of a circumscribing cylinder with radius 6 m and height 2 m. The camera had a 90 deg field of view, with measurement noise standard deviation of 1 pixel, while the IMU was modeled with MEMS quality sensors. The C-KLAM approximate batch MAP optimization problem was solved every 10 incoming camera frames. The exploration epoch was set to 20 camera frames, from which 4 consecutive camera frames were retained as keyframes, while the rest were marginalized using the C-KLAM algorithm. In the BA implementation, the batch MAP optimization problem was solved every 10 camera frames.

TABLE I
RMSE RESULTS FOR BA AND C-KLAM.

| | BA | C-KLAM |
|---|---|---|
| Robot Orientation (rad) | 3.92e-4 | 5.02e-4 |
| Robot Position (m) | 2.24e-2 | 2.75e-2 |
| Landmark Position (m) | 2.78e-2 | 5.31e-2 |

Table I shows the Root Mean Square Error (RMSE) for the platform's position and orientation, and for the landmarks' position (averaged over all key landmarks). From the table, we see that, as expected, the performance of C-KLAM, in terms of accuracy, is comparable to that of the BA.

## IV. CONCLUSIONS

In this paper, we presented C-KLAM, an approximate MAP estimator-based SLAM algorithm. In order to reduce the computational complexity of batch MAP-based SLAM, C-KLAM estimates only the keyframes and key landmarks, observed from these keyframes. However, instead of discarding the measurement information from non-keyframes and non-key landmarks, C-KLAM uses most of this information to generate pose constraints between the keyframes, resulting in substantial information gain. Moreover, the approximations performed in C-KLAM retain the sparsity of the information matrix, and hence the resulting optimization problem can be solved efficiently. We presented both simulation and experimental results for validating the performance of C-KLAM and compared it with that of the batch MAP-based SLAM (bundle adjustment). Our results demonstrated that C-KLAM not only obtains substantial speed-up, but also achieves estimation accuracy comparable to that of the batch MAP-based SLAM that uses all available measurement information.

## REFERENCES


[1] F. Dellaert and M. Kaess. Square root SAM: Simultaneous Localization and Mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, Dec. 2006.

[2] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, Dec. 2006.

[3] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 54–61, Karlsruhe, Germany, May 6–10 2013.

[4] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, Dec. 2008.

[5] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *International Journal of Robotics Research*, 21:217–236, Feb. 2012.

[6] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan, Nov. 13–16 2007.

[7] K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, Oct. 2008.

[8] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *International Journal of Robotics Research*, 29(29):941–957, Jul. 2010.

[9] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent. Efficient Sparse Pose Adjustment for 2D mapping. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 22–29, Taipei, Taiwan, Oct. 18–22 2010.

[10] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3607–3613, Shanghai, China, May 9–13 2011.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[12] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, Apr. 2009.

[13] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, New York, NY, Jun. 17–22 2006.

[14] G. Sibley, L. Matthies, and G. Sukhatme. Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608, Sep./Oct. 2010.

[15] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Dec. 1986.

[16] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment A Modern Synthesis. *Lecture Notes in Computer Science*, 1883:298–372, Jan. 2000.