Multi-Camera Parallel Tracking and Mapping with Non-overlapping Fields of View

Michael J. Tribou[a,*], Adam Harmat[b], David W. L. Wang[c], Inna Sharf[b], Steven L. Waslander[a]

[a] *Department of Mechanical and Mechatronics Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1.*
[b] *Department of Mechanical Engineering, McGill University, 845 Sherbrooke Street West, Montreal, QC, Canada, H3A 0G4.*
[c] *Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1.*

Abstract

A novel real-time pose estimation system is presented for solving the visual SLAM problem using a rigid set of central cameras arranged such that there is no overlap in their fields-of-view. A new parameterization for point feature position using a spherical coordinate update is formulated which isolates system parameters dependent on global scale, allowing the shape parameters of the system to converge despite the scale remaining uncertain. Furthermore, an initialization scheme is proposed from which the optimization will converge accurately using only the measurements from the cameras at the first time step. The algorithm is implemented and verified in experiments with a camera cluster constructed using multiple perspective cameras mounted on a multirotor aerial vehicle and augmented with tracking markers to collect high-precision ground-truth motion measurements from an optical indoor positioning system. The accuracy and performance of the proposed pose estimation system are confirmed for various motion profiles in both indoor and challenging outdoor environments, despite no overlap in the camera fields-of-view.

Keywords: Localization, Mapping, Multi-camera cluster, Non-overlapping FOV, SLAM

*Corresponding author at: University of Waterloo, E3X-4118 – 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1. Telephone: +1-519-635-8971

Email addresses: mjtribou@aeryon.com (Michael J. Tribou), adam.harmat@mail.mcgill.ca (Adam Harmat), dwang@uwaterloo.ca (David W. L. Wang), inna.sharf@mcgill.ca (Inna Sharf), stevenw@uwaterloo.ca (Steven L. Waslander)

1. Introduction

The use of vision as the primary localization sensor for robotics applications provides many inherent advantages over either GPS or laser scanners. Modern vision systems can be extremely light-weight, low-cost and yet high-resolution, and provide valuable colour channel information for the surroundings that can be used for additional purposes such as place recognition, object detection, and motion tracking. The localization accuracy and availability can far exceed GPS, while 3D laser scanners remain expensive and bulky. Recent examples of the rich abilities of vision based autonomy include autonomous driving (Ziegler et al., 2014), mining/planetary exploration (Furgale and Barfoot, 2010) and aerial vehicle flight control (Fraundorfer et al., 2012). {I-1}

In recent years, the trend in visual pose estimation has been to move away from single perspective cameras performing localization and mapping to using more complicated imaging systems. Clusters of central projection cameras have been used which are fixed rigidly with respect to each other, as shown in Figure 1. The individual cameras can be arranged into any configuration, including those with no spatial overlap in the camera field-of-view (FOV){II-5} to make the most effective use of available camera pixels. Even without FOV overlap, non-zero baselines between the individual camera centres allow for full global scale recovery. Additionally, the cameras can be configured such that small translation-rotation motion ambiguities (Fermuller and Aloimonos, 2000) in one camera are compensated for by other cameras facing in orthogonal directions. With the large collective FOV and increased sensitivity, localization accuracy is dramatically improved when compared with monocular and stereo configurations. This is possible even when there is no inter-camera feature correspondence throughout the entire motion sequence.

Some limitations remain, however, with existing methods for localization using multi-camera clusters. Current approaches, described in detail in Section 2, are unable to solve the multi-camera SLAM problem in a unified way, using completely non-overlapping camera FOV or without the requirement for known fiducial markers or any additional sensors with global scale information, such as IMUs or odometry.{II-5}

This work presents a novel visual SLAM formulation to address the current limitations of camera cluster localization techniques, based on ⊞-manifolds (pronounced "box plus manifolds") (Hertzberg et al., 2013), for the specific case of calibrated multi-camera clusters, including configurations in which there is no overlap in FOV and requiring no additional sensors or scale information{II-5}. The use of ⊞-manifolds

Figure 1: An example camera cluster in which the three component cameras are rigidly-fixed with respect to each other.

avoids issues{II-5} with orientation representation{II-5} singularities and enforcement of{II-5} constraints, as they act as a real vector space locally, but can encode a more complex global topology. This is advantageous for representing{II-5} the space of 3D orientations in the group $SO(3)$, as well as 3D rigid-body motions in $SE(3)$. Further, a novel parameter update step for point feature positions in the target model based on spherical coordinates is proposed to isolate the effect of global scale uncertainty. A new initialization scheme is also detailed that allows the pose estimation system to accurately converge even in the case of completely non-overlapping FOV camera clusters. The proposed formulation requires no explicit boot-strap process, immediately starting both tracking and mapping operations, resulting in a simple unified algorithm. Additionally, the global scale of the SLAM solution will naturally converge as sufficient information becomes available in the image point measurements.{II-5} Finally, a real-time open-source{II-5} implementation for the proposed algorithm is detailed, based on the MCPTAM software (Harmat et al., 2014), and demonstrated to produce accurate pose estimates on a robotic aerial platform.

The remainder of this paper is arranged as follows: Section 2 provides a detailed review of the existing multi-camera cluster pose estimation methods in the literature; Section 3 presents the proposed problem formulation and Section 4 describes the novel SLAM algorithm; experimental verification of the proposed method is provided in Section 5 for both indoor and outdoor environments using a cluster onboard a multirotor aerial vehicle; and conclusions are drawn in Section 6.

2. Related Work

For non-overlapping FOV camera cluster motion estimation methods, two general strategies exist{I-2}. A *decoupled* algorithm is one in which each camera's motion is

resolved separately, after which the results are combined to find the cluster motion. A *coupled* method, on the other hand, solves for a single rigid-body motion using images from all cameras concurrently.

There are several decoupled non-overlapping FOV visual odometry (VO) methods in the literature (Kim et al., 2007, 2010b; Clipp et al., 2008; Kazik et al., 2012; Gupta et al., 2013). Due to the limited FOV of each camera, the accuracy of the local motion estimates will suffer more acutely since each camera must individually avoid translation-rotation ambiguities inherent in monocular VO. Consequently, the accuracy of the cluster motion increment will be affected whenever any one of the cameras approaches these ambiguities.

To address the limitations of decoupled approaches, Li *et al.* presented a coupled VO method using the General Epipolar Constraint (GEC) (Pless, 2003) to linearly solve for the cluster motion (Li et al., 2008), similar to the eight-point algorithm (Longuet-Higgins, 1981) for a single camera. This approach provides improved accuracy compared to the decoupled methods by using all of the images from the cameras at once. However, critical motions still remain for the entire non-overlapping FOV cluster motion estimation problem, and as a result, all VO methods which rely only on relative motion estimation are particularly vulnerable to scale drift, which continues to accumulate throughout the estimation sequence.

Instead, the non-overlapping FOV cluster motion estimation problem can be more accurately solved by a SLAM approach, in which the algorithms build a map of point features and localize the cluster pose with respect to the map. As the cluster motion becomes sufficient to accurately resolve the global scale of the map and the motion of the cluster, the point feature positions are updated to account for the new scale information and remove the accumulated drift. This is further refined by using one of the many existing loop-closure methods (Kümmerle et al., 2011; Strasdat et al., 2011) to constrain the solution and produce more accurate SLAM estimates.

Ragab and Wong present a decoupled SLAM method in which they mount two back-to-back camera pairs on a robot (Ragab and Wong, 2010). Each camera tracks its own motion using a separate extended Kalman filter (EKF), which are later combined to find global motion. However, their approach shares the same disadvantages as the decoupled VO methods, namely inherent sensitivity to degenerate motions by either camera pair.

There are several coupled SLAM methods, but most utilize at least some overlap in the cluster camera FOV and are not directly applicable to the problem of non-overlapping FOV cluster SLAM. In (Kaess and Dellaert, 2006, 2010), Kaess and Dellaert provide SLAM algorithms using eight perspective cameras mounted on a robot in a ring facing outwards with FOV overlap in adjacent cameras. Harmat

et al. present a modified version of Parallel Tracking and Mapping (PTAM) (Klein and Murray, 2007), called Multi-Camera PTAM (MCPTAM) (Harmat et al., 2014), able to track the pose of the cluster in real-time while simultaneously computing the Bundle Adjustment (BA) solution in parallel. The system requires at least two of the component cameras to have FOV overlap during the initialization phase. Both Solà *et al.* and Kim *et al.* present coupled recursive SLAM systems which consider a stereo camera setup as two monocular cameras with some FOV overlap (Solà et al., 2008; Kim et al., 2010a). The methods use an EKF to estimate the structure of the environment, the relative motion, and, for Solà *et al.*, even the orientation parameters of the cluster extrinsic calibration.

Mouragnon *et al.* propose a coupled SLAM method for a Generalized Camera Model (GCM) (Mouragnon et al., 2009). In order to initialize the algorithm, a linear solution to the GEC is used to find an initial estimate which is then refined by the subsequent BA. However, solving the GEC using the seventeen point algorithm is not possible for non-overlapping FOV multi-camera clusters as this configuration leads to degeneracy of the solution (Kim and Kanade, 2010). As a result, this method cannot be directly applied to non-overlapping FOV camera systems.

There are{I-2} only a small number of existing{I-2} coupled SLAM methods capable of operating with no overlap in the FOV of the cluster cameras. Tribou *et al.* propose a non-overlapping FOV coupled recursive SLAM system using a single EKF (Tribou et al., 2014b). However, the system is only capable of operating in small environments due to the increasing computational requirements of the EKF as the number of tracked point features grows. Yang *et al.* present another modified version of PTAM to work with multiple cameras with non-overlapping FOV (Yang et al., 2014). A forward-facing and a downward-facing camera are mounted to an aerial robot, and each generates its own sub-map of point features which are only observed by that camera. As a result, areas of the environment which are seen by both cameras must be redundantly mapped by each sensor and valuable constraints on the solution are ignored. Furthermore, the solution scale is initialized by observing a fiducial marker of known dimensions in the downward-facing camera. Subsequently, the forward camera's map is initialized only after the scale is resolved by observing the synthetic target.

The coupled SLAM framework described in this work is implemented as a generalization of the MCPTAM algorithm to allow it to initialize immediately and operate successfully with no FOV overlap in any of the component cameras, using only natural image features. As a result, the new algorithm is able to fully exploit any multi-camera cluster configuration, with or without overlap, and provide accurate, real-time SLAM estimates in a previously unknown environment.

3. Non-overlapping FOV Multi-camera Cluster SLAM

3.1. Calibrated Multi-camera Cluster

Collectively, the calibrated camera cluster is modelled as a set of n_c central projection cameras with known relative coordinate transformations between each camera coordinate frame. Accordingly, a homogeneous point in the projective space $\tilde{\mathbf{p}}^{C_h} \in \mathbb{P}^3$ within the camera frame C_h, can be transformed into any other camera frame C_i by,

$$\tilde{\mathbf{p}}^{C_i} = \mathbf{T}_{C_h}^{C_i} \tilde{\mathbf{p}}^{C_h} \tag{1}$$

where $\mathbf{T}_{C_h}^{C_i} \in SE(3)$, $\forall i, h = 1, 2, \ldots, n_c$. Without loss of generality, the coordinate frame for the camera cluster is chosen to coincide with the first camera frame, C_1. The transformation from camera h to the cluster frame can be written in shortened form as $\mathbf{T}_{C_h} \equiv \mathbf{T}_{C_h}^{C_1}$, where the cluster frame C_1 is implied when the superscript is neglected.

3.2. Taylor Omnidirectional Camera Model

Each individual camera is modelled using the Taylor omnidirectional camera (Scaramuzza, 2007), which allows the use of cameras with greater than 180 degrees FOV. The projection model consists of two steps. First, the 3D point $\mathbf{p} \in \mathbb{R}^3$ is mapped to the i^{th} camera's x-y plane, accounting for the radial distortion of the lens. Second, an affine transformation is applied to represent any misalignment between the lens axis and image sensor.

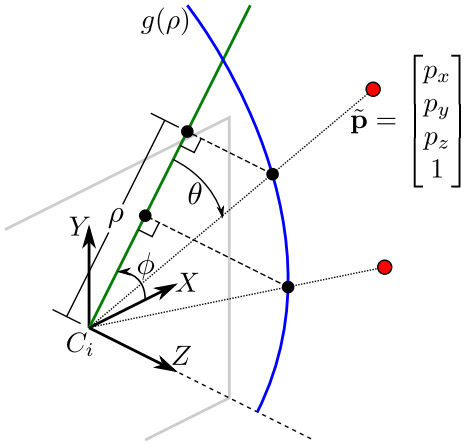


Figure 2: Taylor omnidirectional camera model. The point $\mathbf{p}$ is projected into the camera x-y plane.

Figure 2 shows the first step, mapping the point onto the camera x-y plane. The mapping function is represented as a fourth-order polynomial as in (Harmat et al., 2014),

$$g(\rho) = a_0 + a_2\rho^2 + a_3\rho^3 + a_4\rho^4, \tag{2}$$

where ρ is the radial distance in the camera x-y plane to the projection of the point $\mathbf{p}$, and a_i are the coefficients for the terms of increasing order. Note that $a_1 = 0$ to ensure that the derivative is continuous at $\rho = 0$.

The altitude angle, θ, is the angle between the camera x-y plane and the point feature position,

$$\tan(\theta) = \frac{g(\rho)}{\rho} = \frac{p_z}{\sqrt{p_x^2 + p_y^2}}. \tag{3}$$

An inverse mapping for the projection can be found as,

$$\rho = h(\theta) = b_0 + b_1\theta + b_2\theta^2 + \cdots + b_n\theta^n, \tag{4}$$

where b_i are the coefficients found by solving (2) for a range of θ values and fitting the polynomial to the resulting ρ values. The degree of the polynomial (4) is selected to achieve a desired error bound.

After computing the value of ρ, the image coordinates of the point can be calculated as,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \boldsymbol{\kappa}_i(\mathbf{p}) = \mathbf{A}_i \begin{bmatrix} \rho\cos(\phi) \\ \rho\sin(\phi) \end{bmatrix} + \mathbf{c}, \tag{5}$$

where ϕ is the azimuth angle bearing to the point feature such that,

$$\tan(\phi) = \frac{p_y}{p_x}, \tag{6}$$

$\mathbf{c} \in \mathbb{R}^2$ is the image centre offset, and $\mathbf{A}_i \in \mathbb{R}^{2\times 2}$ is the affine transformation matrix with elements determined at calibration time.

The inverse of the projection mapping ${\boldsymbol{\kappa}_i}^{-1}$ takes image coordinates and determines the unit vector, $\hat{\mathbf{p}} \in \mathbb{R}^3$, representing the bearing to a point feature,

$$\hat{\mathbf{p}} = {\boldsymbol{\kappa}_i}^{-1}\left(\begin{bmatrix} u \\ v \end{bmatrix}\right). \tag{7}$$

3.3. Target Object Model

The target object or environment is modelled as a set of point features organized into n_k keyframes. Each keyframe is a six degree of freedom (DOF) pose estimate with respect to the target model reference frame M, along with the n_c images from the cluster cameras captured at that location, as in (Klein and Murray, 2007) for a single camera. Each keyframe contains a set of point features that are said to be *anchored* within the respective camera coordinate frame at that keyframe. The coordinate frame of camera h at keyframe k is denoted C_hF_k {I-3}.

The position of a point feature is expressed with respect to the camera coordinate frame at its anchor keyframe – the first keyframe in which it is observed. It is assumed in this work that the point features are a finite distance from the camera cluster at all time steps. This assumption excludes tracking point features such as stars or distant points on the horizon, which are effectively at infinite depth from a practical viewpoint.

Since the relative position and orientation of each component camera within the cluster is fixed at all times, the k^{th} keyframe pose is parameterized by the single homogeneous transformation for the cluster coordinate frame at the keyframe, C_1F_k, with respect to the target model reference frame, M, resulting in $\mathbf{T}^M_{C_1F_k} \in SE(3)$. The C_1 and M frames are implied in this keyframe pose definition, and therefore, the transformation will be written simply as $\mathbf{T}_{F_k} \equiv \mathbf{T}^M_{C_1F_k}$. The pose of camera h at keyframe k is easily found as $\mathbf{T}^M_{C_hF_k} = \mathbf{T}_{F_k}\mathbf{T}_{C_h}$.

An example system with a camera cluster composed of $n_c = 2$ cameras and $n_k = 3$ keyframes is shown in Figure 3. The cameras in this example are arranged back-to-back with the optical axes looking outwards along the z-axes of the associated coordinate frames. The j^{th} point feature is anchored in the second camera at the second keyframe, C_2F_2, and its position with respect to this coordinate frame is represented as $\mathbf{p}_j^{C_2F_2}$.

Since the target model is initially unknown, the set of keyframes is accumulated as the estimation proceeds and new observations of the target object are made. At the beginning, the model consists of only one keyframe from the initial observation. As the algorithm continues through the sequence, it will determine when to add a new keyframe at the current cluster pose. This process grows the set of keyframes to cover the entire target object.

The parameters representing the set of keyframe poses together with the positions of the point features observed within them, compose the target model, as well as the full system state, $\mathbf{x} \in \mathcal{S}$, where $\mathcal{S}$ is the state manifold. These parameters are estimated using the point feature image measurements within the cluster cameras. The next section derives this relationship.

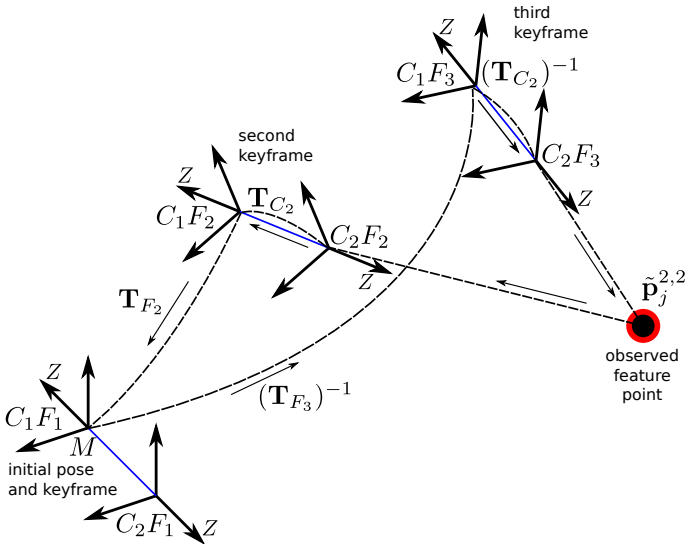


Figure 3: An example target object model with the transformations between the coordinate frames shown as dashed lines. The point feature j is anchored within the C_2F_2 frame and measured at the C_2F_3 frame.

3.4. Camera Cluster Measurement Model

The measurement model, relating the observed point feature locations in the camera image planes, to the system states, can be written as a series of coordinate transformations. Suppose that the j^{th} point feature, anchored in the coordinate frame C_hF_k, is measured by camera i at C_iF_ℓ. An example of this chain of transformations is shown for the simple back-to-back two-camera cluster system with three keyframes in Figure 3. In this particular case, the point feature j is anchored in C_2F_2 and observed in C_2F_3.

The feature point position parameters give the location of the j^{th} feature in its anchor keyframe and camera frame C_hF_k, resulting in $\mathbf{p}_j^{h,k}$. The homogeneous representation of this point is denoted $\tilde{\mathbf{p}}_j^{h,k} = \left[\left(\mathbf{p}_j^{h,k}\right)^\top \; 1 \right]^\top$. This point feature is first transformed into the target model coordinate frame by $\tilde{\mathbf{p}}_j^{M} = \mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j^{h,k}$, which are transformations provided by the known cluster calibration and the estimated keyframe pose.

Next, the point is transformed into the coordinate frame of the observing keyframe and camera C_iF_ℓ using the observing keyframe pose states and the cluster calibration,

$$\tilde{\mathbf{p}}_j^{i,\ell} = (\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j^{h,k}. \tag{8}$$

Finally, the point is projected onto the image plane of camera C_i using the corresponding projection function,

$$\begin{bmatrix} u_j^{i,\ell} \\ v_j^{i,\ell} \end{bmatrix} = \boldsymbol{\kappa}_i \left(\boldsymbol{\pi}_3(\tilde{\mathbf{p}}_j^{i,\ell}) \right), \tag{9}$$

known from the intrinsic calibration of the individual cluster cameras and the operator $\boldsymbol{\pi}_3$ which extracts the first three elements of the homogeneous point, assuming that the feature is not infinitely far from the camera.

Each of the four intermediate homogeneous transformation matrices in (8) are formed by either the system states, or the known cluster camera configurations from extrinsic calibration. Therefore, the measurement equation for feature j as seen in the observing keyframe and camera is,

$$\mathbf{z}_j^{i,\ell} = \mathbf{g}_j^{i,\ell}(\mathbf{x}) + \boldsymbol{\gamma}_j^{i,\ell} \tag{10}$$

where

$$\mathbf{g}_j^{i,\ell}(\mathbf{x}) = \boldsymbol{\kappa}_i \left(\boldsymbol{\pi}_3 \left((\mathbf{T}_{C_i})^{-1} (\mathbf{T}_{F_\ell})^{-1} \mathbf{T}_{F_k} \mathbf{T}_{C_h} \tilde{\mathbf{p}}_j^{h,k} \right) \right) \tag{11}$$

and $\boldsymbol{\gamma}_j^{i,\ell} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_j^{i,\ell}\right)$. The Gaussian noise model has been shown to be a good approximation of the actual image measurement noise from the feature extraction process (Madhusudan, 1992).

The full system measurement vector $\mathbf{z} \in \mathcal{M}$ is made up of all of the individual point feature observations at all of the keyframes. It is modelled as a stacked column vector of measurements of the form (10). The measurement function, $\mathbf{g} : \mathcal{S} \to \mathcal{M}$, maps the current state of the system, $\mathbf{x} \in \mathcal{S}$, to the deterministic portion of the system measurement manifold, $\mathcal{M}$. The complete system measurement model is, $\mathbf{z} = \mathbf{g}(\mathbf{x}) + \boldsymbol{\gamma}$, where $\boldsymbol{\gamma}$ is the measurement noise vector formed by stacking all of the individual noise vectors for feature observations into a column vector.

3.5. Optimization

The optimization for the camera tracking and target mapping processes is performed using the Levenberg-Marquardt (LM) algorithm (Hartley and Zisserman, 2003). The system presented uses the $\boxplus$-manifold (Hertzberg et al., 2013), which encapsulates the global topology of the multi-camera cluster system state consisting of the target model keyframe poses and point feature positions. The novel application of this concept to the particular case of non-overlapping FOV multi-camera clusters is claimed. Furthermore, a new parameterization and state update method for the point feature positions is proposed.

3.5.1. State Representation Using ⊞-Manifolds

The state space for the BA system is composed of a set of keyframe poses and point feature positions. There are many ways to parameterize these states as real-valued vectors and estimation methods have been proposed using a wide variety of parameterizations, which for rotation include the minimal representations such as, Euler angles (Wilson et al., 1996), unit quaternions (Davison, 2003), or modified Rodrigues parameters (Crassidis and Markley, 1996). However, all of the minimal representations for the $SO(3)$ group as a flat vector in $\mathbb{R}^3$ or $\mathbb{R}^4$ have singularities or extra constraints on the parameters which present unnecessary challenges to the optimization process. For example, Euler angles suffer from the gimbal lock singularity when the second angle goes to $\pm\frac{\pi}{2}$ rad, and unit quaternions must maintain the unit-length constraint. The least-squares optimization methods cannot enforce this constraint without requiring the addition of extra equations or an explicit normalization step (Hertzberg et al., 2013).

In this work, the state space $\mathcal{S}$ is represented as a ⊞-manifold, as proposed by Hertzberg *et al.* (Hertzberg, 2008; Hertzberg et al., 2013)[{II-2}]. The formal definition and properties of these objects are found in the cited works. These manifolds act as a real vector space locally, but can encode a more complex global topology, such as that of the space of 3D orientations in the group $SO(3)$, as well as 3D rigid-body motions in $SE(3)$. The method is a formalism of other work on parameterizing and linearizing specific manifolds for iterative optimization schemes, e.g. (Furgale, 2011).[{II-2}]

The general idea behind the ⊞-manifold is to change the iterative update and error calculations to replace the $+$ and $-$ operators with the more general operators $\boxplus$ and $\boxminus$, which respect the underlying topology of the state space, but interface with optimization algorithms using the real vector space (Hertzberg et al., 2013). The operator $\boxplus$ is used to apply updates to the state manifold, $\mathbf{x} \in \mathcal{S}$, given a perturbation vector using a selected minimal representation for the update vector, $\boldsymbol{\delta} \in \mathbb{R}^n$,

$$\mathbf{x} \mapsto \mathbf{x} \boxplus \boldsymbol{\delta} \in \mathcal{S}, \tag{12}$$

producing an updated $\mathbf{x}$ that maintains the global topology of the manifold $\mathcal{S}$. This allows the state to be a (possibly over-parameterized) representation free of singularities, manipulated by small-magnitude perturbations $\boldsymbol{\delta} \in \mathbb{R}^n$. Since the iterative optimization methods apply small refinements to the state, the update vector $\boldsymbol{\delta}$ is a minimal representation kept sufficiently far from the respective singularities.

A further strength of the ⊞-manifold approach is that the update equation (12) can be selected using prior knowledge of the problem. For the point feature states,

a non-overlapping FOV camera cluster has difficulties recovering the properly-scaled depth to the feature, particularly in the initial time steps when there is little information due to small relative motions. Therefore, even though the point position parameters are represented as the Cartesian coordinates in $\mathbb{R}^3$, the proposed update step treats the point as if it were on the surface of a sphere, centred at the camera coordinate frame, and updates the bearing along the surface separate from the update to the radial distance. This allows the bearing to converge{II-1} despite the depth remaining uncertain.

3.5.2. Keyframe Pose Manifold

The $\boxplus$-manifold $\mathcal{F}$ for the keyframe pose states is a $\boxplus$-manifold on the group $SE(3)$. The pose of each keyframe in the system is represented directly as a homogeneous transformation and there is no need to use a minimal vector representation in the real vector space, except for the update operation.

An{I-2} exponential map from a minimal representation on $\mathbb{R}^6$ to the special Euclidean group $SE(3)$, is defined as,

$$\mathbf{T}_{\mathcal{F}} : \mathbb{R}^6 \rightarrow SE(3), \tag{13}$$

and takes the increment vector $\boldsymbol{\delta}_F = [\mathbf{v}^\top \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^6$ into $SE(3)$ using the exponential on $SO(3)$ (Lee, 2013) for the orientation and calculates the transformation matrix as (Agrawal, 2006),

$$\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_F) = \begin{bmatrix} \boldsymbol{\mathcal{R}}_{\mathcal{F}}(\boldsymbol{\omega}) & \left(\mathbf{I}_{3\times 3} + \left(\frac{1-\cos\theta}{\theta}\right)[\hat{\boldsymbol{\omega}}]_\times + \left(1 - \frac{\sin\theta}{\theta}\right)[\hat{\boldsymbol{\omega}}]_\times^2\right)\mathbf{v} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}, \tag{14}$$

where (Hartley and Zisserman, 2003),

$$\boldsymbol{\mathcal{R}}_{\mathcal{F}}(\boldsymbol{\omega}) = \mathbf{I}_{3\times 3} + (\sin\theta)\,[\hat{\boldsymbol{\omega}}]_\times + (1-\cos\theta)\,[\hat{\boldsymbol{\omega}}]_\times^2\,. \tag{15}$$

and $\boldsymbol{\omega} = \theta\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$ where $\theta = \|\boldsymbol{\omega}\| \in \mathbb{R}$ is the rotation angle, $\hat{\boldsymbol{\omega}}$ is the unit-length rotation axis, and $[\mathbf{a}]_\times$ is the skew-symmetric matrix such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$, for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$.

The operator $\boxplus_{\mathcal{F}}$ to modify the state estimates of the keyframe poses is then defined:

$$\boxplus_{\mathcal{F}} : SE(3) \times \mathbb{R}^6 \rightarrow SE(3) \tag{16}$$

such that for a keyframe $\mathbf{T}_{F_i} \in SE(3)$ and update vector $\boldsymbol{\delta}_F \in \mathbb{R}^6$,

$$\mathbf{T}_{F_i} \boxplus_{\mathcal{F}} \boldsymbol{\delta}_F = \mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_F)\mathbf{T}_{F_i}. \tag{17}$$

Note that in the BA optimization algorithm the $\boxminus$ operator is not used since the elements of the state space are never measured directly. It is only through the measurements that information is gained regarding the state. This means that the keyframe poses do not ever need to be reduced to the flattened vector in the real vector space. Instead, the system state consists of a set of 4×4 coordinate transformation matrices.

3.5.3. Point Position Manifold

The $\boxplus$-manifold $\mathcal{P}$ for the point feature position states is defined as the vector space $\mathbb{R}^3$ along with the operator, $\boxplus_{\mathcal{P}} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$. As with the keyframe state manifold, the point positions are not measured directly and the $\boxminus_{\mathcal{P}}$ operator is not used in the optimization algorithm.

In defining the $\boxplus_{\mathcal{P}}$ operator, this section will present two alternatives. For a point feature, $\mathbf{p}_j \in \mathbb{R}^3$ and an increment $\boldsymbol{\delta}_P \in \mathbb{R}^3$. The first, and most obvious option is to define it as the usual addition operator in $\mathbb{R}^3$,

$$\mathbf{p}_j \boxplus_{\mathcal{P}} \boldsymbol{\delta}_P = \mathbf{p}_j + \boldsymbol{\delta}_P. \tag{18}$$

This is the simplest operator to implement and there is no difference in this representation and the classical flat vector representation of the model point feature positions.

The second option is a novel state update based on spherical coordinates and inspired by the Inverse Depth Parameterization (IDP) (Civera et al., 2008). In the calibrated multi-camera cluster tracking system with non-overlapping camera FOV, the image measurements are quite insensitive to the global scale of the solution, particularly when the relative motion of the cluster and the target object is (near) degenerate (refer to Section 3.6). As a result, the directions of all of the position vectors in the system (keyframe translations and point positions), as well as the keyframe orientations can be accurately determined, while the global scale may be ambiguous or inaccurate.

This new update treats the feature position as a point on a sphere centred at the anchor camera coordinate frame. The point is moved on the surface of the sphere by the angle increments, δ_α and δ_β, and then moved in the radial direction by a scaling factor increment δ_r. The angle increments move the point in the local (X', Y', Z') coordinate system along the surface of the sphere, while the scale increment changes the radial distance to the point.

Using this update parameterization isolates the parts of the point position that can be estimated precisely using a camera, the bearing to the point on the sphere, from the part which is difficult to determine without sufficient motion, the radial

depth to the point. This separation allows for a point feature position estimate, which is accurate in bearing but may not be in depth, to converge{II-1} to the proper scale when there is sufficient information from non-degenerate relative motion.

The $\boxplus_{\mathcal{P}}$ operator is thus defined, for $\mathbf{p}_j, \boldsymbol{\delta}_P \in \mathbb{R}^3$,

$$\mathbf{p}_j \boxplus_{\mathcal{P}} \boldsymbol{\delta}_P = \boldsymbol{\pi}_3 \left(\mathbf{T}_{\mathcal{P}}(\boldsymbol{\delta}_P, \mathbf{p}_j) \tilde{\mathbf{p}}_j \right) \tag{19}$$

where the operator $\mathbf{T}_{\mathcal{P}}$ forms the transformation matrix, $\mathbf{T}_{\mathcal{P}} : \mathbb{R}^3 \times \mathbb{R}^3 \to SE(3)$ such that if $\boldsymbol{\delta}_P = [\delta_\alpha, \delta_\beta, \delta_r]^\top$,

$$\mathbf{T}_{\mathcal{P}}(\boldsymbol{\delta}_P, \mathbf{p}_j) = \begin{bmatrix} (1+\delta_r)\boldsymbol{\mathcal{R}}_{\mathcal{P}}(\mathbf{p}_j)\boldsymbol{\mathcal{R}}_{\mathcal{F}}([\delta_\alpha,\ \delta_\beta,\ 0]^\top)\boldsymbol{\mathcal{R}}_{\mathcal{P}}(\mathbf{p}_j)^\top & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{20}$$

where a prerotation $\boldsymbol{\mathcal{R}}_{\mathcal{P}}(\mathbf{p}_j)^\top$ aligns the vector $\mathbf{p}_j$ with the camera frame z-axis, and is calculated by the operator, $\boldsymbol{\mathcal{R}}_{\mathcal{P}} : \mathbb{R}^3 \to SO(3)$, according to $\boldsymbol{\mathcal{R}}_{\mathcal{P}}(\mathbf{p}_j) = \boldsymbol{\mathcal{R}}_{\mathcal{F}}(\theta\hat{\boldsymbol{\omega}})$, with $\theta = -\arcsin\|\boldsymbol{\omega}\|$ and $\boldsymbol{\omega} = \hat{\mathbf{p}}_j \times [0,\ 0,\ 1]^\top$.

Next, the vector is perturbed by the two incremental angles δ_α and δ_β, which preserves the length of the vector. This changes the direction of the vector using the local coordinate system, (X', Y', Z'), on the sphere. Finally, the vector is rotated back to the original neighbourhood using $\boldsymbol{\mathcal{R}}_{\mathcal{P}}(\mathbf{p}_j)$ and then scaled by the increment $(1+\delta_r)$.

Compared with IDP, this parameterization has the advantage of maintaining the state representation of the point feature position in the Cartesian coordinates, while updating the position in a similar spherical manner. Additionally, the parameterization is not vulnerable to the singularity present in IDP associated with the altitude angle going to $\pm\frac{\pi}{2}$ rad. As a result, it can accommodate component cameras with greater than 180 degree FOV.

3.5.4. Jacobian Calculation

At each iteration of the LM optimization algorithm, the state estimate $\breve{\mathbf{x}} \in \mathcal{S}$ is modified by the update vector $\boldsymbol{\delta}$, calculated using the measurement Jacobian matrix, $\mathbf{J}$. This matrix represents the change in the predicted point feature measurements for a change in the update vector,

$$\mathbf{J} = \left. \frac{\partial \mathbf{g}(\breve{\mathbf{x}} \boxplus_{\mathcal{S}} \boldsymbol{\delta})}{\partial \boldsymbol{\delta}} \right|_{\boldsymbol{\delta}=\mathbf{0}}. \tag{21}$$

The Jacobian is formed by vertically stacking the $2\times n$ Jacobians for the individual

point feature observations,

$$\mathbf{J}_j^{i,\ell} = \left.\frac{\partial \mathbf{g}_j^{i,\ell}(\breve{\mathbf{x}} \boxplus_{\mathcal{S}} \boldsymbol{\delta})}{\partial \boldsymbol{\delta}}\right|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbf{H}_j^{i,\ell}\mathbf{G}_j^{i,\ell}, \tag{22}$$

where

$$\mathbf{H}_j^{i,\ell} = \left.\frac{\partial \boldsymbol{\kappa}_i(\boldsymbol{\pi}_3(\tilde{\mathbf{p}}_j^{i,\ell}))}{\partial \tilde{\mathbf{p}}_j^{i,\ell}}\right|_{\boldsymbol{\delta}=\mathbf{0}}, \quad \text{and} \quad \mathbf{G}_j^{i,\ell} = \left.\frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \boldsymbol{\delta}}\right|_{\boldsymbol{\delta}=\mathbf{0}}. \tag{23}$$

The term $\mathbf{H}_j^{i,\ell}$ is a 2×4 matrix dependent on the camera structure and the point position in the observing camera and keyframe coordinate frame. For the Taylor camera model the matrix is (Harmat et al., 2014),

$$\mathbf{H}_j^{i,\ell} = \mathbf{A}_i \begin{bmatrix} \frac{d\rho}{d\theta}\cos(\phi) & -\rho\sin(\phi) \\ \frac{d\rho}{d\theta}\sin(\phi) & \rho\cos(\phi) \end{bmatrix} \begin{bmatrix} \frac{-\tan(\theta)p_x}{p_x^2+p_y^2+p_z^2} & \frac{-\tan(\theta)p_y}{p_x^2+p_y^2+p_z^2} & \frac{\sqrt{p_x^2+p_y^2}}{p_x^2+p_y^2+p_z^2} & 0 \\ \frac{p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 & 0 \end{bmatrix}, \tag{24}$$

where

$$\frac{d\rho}{d\theta} = \frac{\rho^2 + g(\rho)^2}{-a_0 + a_2\rho^2 + 2a_3\rho^3 + 3a_4\rho^4}, \tag{25}$$

with $\tilde{\mathbf{p}}_j^{i,\ell} = [p_x,\ p_y,\ p_z,\ 1]^\top$, and θ, ρ, and ϕ are calculated using (3), (4), and (6), respectively.

The term $\mathbf{G}_J^{i,\ell}$ is a $4 \times (6n_k + 3n_f)$ matrix, where n_k and n_f are the number of keyframes and point features in the target model, respectively. This matrix represents the change of the feature position within the observing camera and keyframe coordinate frame for a change in the update vector. The fourth row of this matrix is all zeros since the points are restricted to lie in $\mathbb{R}^3$.

Assume the point feature $\mathbf{p}_j^{h,k}$, anchored in camera C_h of keyframe F_k, is observed at keyframe F_ℓ by camera C_i. The coordinates of the point feature position in the coordinate frame C_iF_ℓ are given in (8). The update vector consists of components for all of the keyframes and points in the system.

$$\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_{F_1}{}^\top & \dots & \boldsymbol{\delta}_{F_{n_k}}{}^\top & \boldsymbol{\delta}_{P_1}{}^\top & \dots & \boldsymbol{\delta}_{P_{n_f}}{}^\top \end{bmatrix}^\top \in \mathbb{R}^{(6n_k+3n_f)} \tag{26}$$

The position of the point feature in the observing coordinate frame subject to the state perturbations is written,

$$\tilde{\mathbf{p}}_j^{i,\ell} = \mathbf{T}_{C_i}{}^{-1}(\mathbf{T}_{F_\ell} \boxplus_{\mathcal{F}} \boldsymbol{\delta}_{F_\ell})^{-1} (\mathbf{T}_{F_k} \boxplus_{\mathcal{F}} \boldsymbol{\delta}_{F_k})\, \mathbf{T}_{C_h} \begin{bmatrix} \left(\mathbf{p}_j \boxplus_{\mathcal{P}} \boldsymbol{\delta}_{P_j}\right)^\top & 1 \end{bmatrix}^\top. \tag{27}$$

Applying the operators for the various $\boxplus$-manifolds in the state space,

$$\tilde{\mathbf{p}}_j^{i,\ell} = \mathbf{T}_{C_i}{}^{-1}(\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_\ell})\mathbf{T}_{F_\ell})^{-1}\left(\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_k})\mathbf{T}_{F_k}\right)\mathbf{T}_{C_h}\left(\mathbf{T}_{\mathcal{P}}(\boldsymbol{\delta}_{P_j})\tilde{\mathbf{p}}_j\right) \tag{28}$$

$$= \mathbf{T}_{C_i}{}^{-1}\mathbf{T}_{F_\ell}{}^{-1}(\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_\ell}))^{-1}\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_k})\mathbf{T}_{F_k}\mathbf{T}_{C_h}\mathbf{T}_{\mathcal{P}}(\boldsymbol{\delta}_{P_j})\tilde{\mathbf{p}}_j. \tag{29}$$

The only non-zero blocks in $\mathbf{G}_j^{i,\ell}$ are those corresponding to the anchor and observing keyframes, as well as the feature position,

$$\mathbf{G}_j^{i,\ell} = \begin{bmatrix} \mathbf{0} & \dfrac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_k}} & \mathbf{0} & \dots & \mathbf{0} & \dfrac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_\ell}} & \mathbf{0} & \dots & \mathbf{0} & \dfrac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{P_j}} & \mathbf{0} \end{bmatrix} \tag{30}$$

Accordingly, these blocks will be investigated separately in the following.

The Jacobian block associated with the update vector for the anchor keyframe, F_k, is (Sibley et al., 2009),

$$\frac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_k}} = \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right)\frac{\partial\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_k})}{\partial\boldsymbol{\delta}_{F_k}}\left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right) \tag{31}$$

$$= (\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\begin{bmatrix}\mathbf{I}_{3\times3} & -\left[\mathbf{p}_j^M\right]_\times \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix}. \tag{32}$$

Similarly, for the Jacobian block associated with the observing keyframe update vector,

$$\frac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_\ell}} = \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right)\frac{\partial\left(\mathbf{T}_{\mathcal{F}}(\boldsymbol{\delta}_{F_\ell})^{-1}\right)}{\partial\boldsymbol{\delta}_{F_\ell}}\left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right) \tag{33}$$

$$= \left((\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\right)\frac{\partial\mathbf{T}_{\mathcal{F}}(-\boldsymbol{\delta}_{F_\ell})}{\partial\boldsymbol{\delta}_{F_\ell}}\left(\mathbf{T}_{F_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j\right) \tag{34}$$

$$= (\mathbf{T}_{C_i})^{-1}(\mathbf{T}_{F_\ell})^{-1}\begin{bmatrix}-\mathbf{I}_{3\times3} & \left[\mathbf{p}_j^M\right]_\times \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix}. \tag{35}$$

In the event that the anchor and observing keyframes are the same, $k = \ell$, updating the pose will have no effect on the feature measurement since the transformation matrices will always combine to identity in (11). As a result, the block in $\mathbf{G}_j^{i,\ell}$ for that keyframe will be,

$$\frac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_\ell}} = \frac{\partial\tilde{\mathbf{p}}_j^{i,\ell}}{\partial\boldsymbol{\delta}_{F_k}} = \mathbf{0}_{4\times6}. \tag{36}$$

Finally, for the novel point feature position update, the Jacobian block has the form,

$$\frac{\partial \tilde{\mathbf{p}}_j^{i,\ell}}{\partial \boldsymbol{\delta}_{P_j}} = \left\|\mathbf{p}_j\right\| (\mathbf{T}_{C_i})^{-1} (\mathbf{T}_{F_\ell})^{-1} \mathbf{T}_{F_k} \mathbf{T}_{C_h} \begin{bmatrix} \boldsymbol{\mathcal{R}}_{P_j}(\mathbf{p}_j) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{0}_{1\times 3} \end{bmatrix}. \tag{37}$$

The full Jacobian $\mathbf{J}$ for the general BA optimization can be calculated given the current estimate of the state and used to find the new update vector $\boldsymbol{\delta}$ for the next iteration. Provided that the initial estimate is in the neighbourhood of the true solution the optimization will converge to the global minimum and produce an accurate target model.

3.6. Solution Degeneracies

In the case of a non-overlapping FOV camera cluster configuration, it is possible to recover the motion parameters when disjoint sets of point features are tracked by each individual camera and not seen by any other in the camera cluster. However, there are relative motions of the camera cluster and the target for which the motion and structure cannot be uniquely estimated using the image measurements. These are called critical motions (Clipp et al., 2008), and can potentially cause an estimator to diverge or converge to an incorrect solution, if proper care is not taken.

A detailed analysis of the solution degeneracies associated with non-overlapping FOV multi-camera cluster SLAM was performed by Tribou *et al.* (Tribou et al., 2014a). It was demonstrated that providing there are measurements of more than the minimal number of point features on the target object, the measurement Jacobian $\mathbf{J}$ is rank-deficient and, therefore, the solution is degenerate only when all of the optical centres of the component cameras move in parallel between two keyframes. This includes the cases of concentric circle motion for two-camera clusters, or pure translation for a cluster of any number of cameras. Under these motions, the solution is ambiguous in the global scale of the relative motion and target model. Furthermore, as the relative motion approaches critical, the measurements become increasingly insensitive to the global scale metric. In the presence of image measurement noise, the solution will converge to a consistent, but incorrect scale value.

4. Non-overlapping FOV MCPTAM Algorithm

4.1. Algorithm Strategy

The modelling and optimization mechanisms described in the previous sections are now combined into a real-time relative motion and target model estimation al-

gorithm using the non-overlapping calibrated camera cluster. The proposed parameterization and optimization method, along with the novel initialization scheme, presented in Section 4.1.3, allow the solution to accurately converge despite the lack of prior knowledge of the target object model or relative motion.

Similar to PTAM (Klein and Murray, 2007) and MCPTAM (Harmat et al., 2014), the tasks of real-time motion tracking and accurate reconstruction of the target model structure are divided into parallel tasks running as distinct processes. The full optimization, as described in Section 3.5, is implemented in the BA module to accurately determine the poses of the keyframes and positions of the point features within them. The included keyframes must be carefully selected to sufficiently constrain the target model solution, while limiting the total number of keyframes in the target model to keep the computational requirements low. Concurrently, a pose tracking process localizes the current camera cluster coordinate frame within the most recent target model generated by the BA module.

The separation of the motion tracking and BA optimization tasks alleviates the real-time requirement on the full nonlinear optimization. The challenging part is to boot-strap the process successfully despite no overlap in the cluster camera FOV and no prior knowledge of the target object.

4.1.1. Pose Tracking

In the pose tracking process, the current position and orientation of the camera cluster are localized using the most recent target model provided by the BA process. The target model parameters are held fixed, and only the current pose of the cluster coordinate frame, denoted U, with respect to the target model frame, M, is optimized given the measurements of the existing target model point features in the current set of component camera images. As a result, the tracking system state is simply,

$$\mathbf{x} = \mathbf{T}_U^M \equiv \mathbf{T}_U = \begin{bmatrix} \boldsymbol{\mathcal{R}}_U & \mathbf{t}_U \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in SE(3), \tag{38}$$

while the system measurement vector consists of all of the image-plane measurements of all of the currently visible point features from within the target model. The relative pose is then estimated using an iterative nonlinear least squares optimization method. At the conclusion of the optimization, the current pose estimate is made available to the BA process for adding new keyframes to the model.

Since both the camera cluster and the target object or environment are able to move, the relative motion dynamics are approximated as a simple constant-velocity model. Between time steps, the two previous relative pose estimates are combined to predict the current pose of the cluster with respect to the target. However, this only

serves as the initial condition for the pose estimate and the optimization proceeds using just the current image measurements to constrain the tracking solution.

The low dimension of this state space allows this process to run at the frame rate of the cluster cameras to maintain a real-time pose estimate for the camera cluster with respect to the target model. The accuracy of the resulting estimates is dependent on the accuracy of the most recent target model from the BA process. In practice, this methodology has proven to work effectively once the target model is refined over a number of BA optimization runs. The most critical portion is the initialization phase when the generated target model is uncertain due to the limited information available.

4.1.2. Bundle Adjustment

A dedicated process generates and continually refines the model of the target object or environment using the full nonlinear BA framework detailed in Section 3.5. The target model consists of a set of permanent keyframes, in which all of the corresponding point feature positions are parameterized, and a single temporary keyframe representing the most recently acquired cluster pose from the tracking thread. The distinction between permanent and temporary keyframes is only their persistence in the target model past the convergence of the optimization.

Permanent keyframes are, as the name suggests, permanently part of the target model and can be used to anchor point feature positions. A temporary keyframe is added to the target model at the most recent pose estimate from the tracking thread, for the purposes of triangulating point features within the permanent keyframes and constraining the poses of the permanent keyframes. When the optimization converges, the temporary keyframe may be discarded or, if the algorithm determines that it significantly improves the target model by observing a set of new point features or has a long baseline from the neighbouring permanent keyframes, the temporary keyframe can be promoted to a permanent keyframe in the target model. As a result, new point features are anchored in this frame and added to the target model.

The criteria for selecting when to promote the temporary keyframe to permanent are similar to the keyframe selection process in PTAM, but with some additional metrics to accommodate the multi-camera configuration. A new permanent keyframe is added to the map when the distance from the temporary keyframe to the nearest neighbour permanent keyframe exceeds a preset threshold. This distance between two keyframes is given as the smallest distance between any two of their respective component camera coordinate frames.

The distance between two component camera coordinate frames aims to encode that two cameras may be considered “close” if their optical centres are physically

located near one another or if they are looking at the same scene from different viewpoints. Therefore, it is defined as the sum of the Euclidean distance between the two camera frame origins and the Euclidean distance between two virtual points located at the mean scene depth along each camera's z-axis. Finally, this sum is divided by the mean depth of the features within the permanent keyframe. This division accounts for the fact that when the scene depth is large, a large amount of motion is necessary before tracking is lost.{II-3}

As the BA process finishes the first few optimization runs, the second permanent keyframe is added at the cluster pose when the maximum trace of the point feature covariance matrices falls below a selected threshold. This metric is selected to use the feature position uncertainty as a guide for the keyframe placement but{II-4} keep the number of permanent keyframes low during the initial operation and avoid adding keyframes before the solution is sufficiently-constrained. With the small keyframe count, the BA thread is able to supply the tracking thread with an updated version of the target model at or near the frame rate of cluster cameras.

When the BA process completes, the updated target model is sent to the tracking thread for use in localizing the current relative pose of the camera cluster. The BA process will then retrieve the latest cluster pose estimate from the tracking process, including the images from the individual cameras, and begin the optimization again with this pose as a new temporary keyframe.

It is possible for the target model to change significantly between time steps of the pose tracking process, particularly if recently-added keyframes lead to significant corrections to the global scale metric in the model generated by the BA process. As a result, the tracking process could potentially fail if its previous pose estimate becomes a poor initial condition for the optimization with respect to the new model. However, the large scale corrections usually occur towards the beginning of the motion trajectory as the relative motion becomes sufficiently large to constrain the global scale metric. Assuming that the update rate of the BA process is fast enough given the magnitude of the relative motion, the point feature measurements will still be relatively insensitive to an incorrect scale metric (refer to Section 3.6). As a result, the pose tracking optimization will converge to the new current estimate despite a poorly-scaled initial condition.

4.1.3. Initialization

Different from the PTAM approach, the proposed tracking and BA processes operate in parallel right from the start of the motion sequence. Further, compared to the previous MCPTAM, it adds the ability to successfully initialize and operate even when there is no overlap in the FOV of the cluster cameras. To achieve

this, the tracking thread requires that a suitable target model be available immediately at start-up. This novel initialization method provides an initial target model constructed using the information available in the first set of cluster camera images.

Upon capturing an initial set of images from all of the component cameras in the cluster, a permanent keyframe is added to the target model and fixed collocated at the target model reference frame, M. The observed feature points are initialized using the image space measurements to determine the bearing to the points in space at this keyframe. Since there is no overlap in the FOV of the cameras, there is no information about the depth of the point features in the scene, except that the points must have a positive non-zero depth value.

For a measurement of the j^{th} point feature $[u_j, v_j] \in \mathbb{R}^2$ first observed, and subsequently anchored in camera h at the new keyframe being added to the model, the ray along which it lies can be found using inverse mapping of the camera projection, ${\boldsymbol{\kappa}_h}^{-1}$ from (7),

$$\hat{\mathbf{q}}_j = {\boldsymbol{\kappa}_h}^{-1} \left(\begin{bmatrix} u_j \\ v_j \end{bmatrix} \right) \in \mathbb{R}^3. \tag{39}$$

Subsequently, the initial estimate of the point feature position can be calculated using this direction ray,

$$\mathbf{p}_j = d_0 \hat{\mathbf{q}}_j \tag{40}$$

where the initial depth along the feature ray d_0 can be set using even poor a priori information regarding the target model, such as expected average feature point depth. If none is available, a reasonable nominal value can be selected, such as $d_0 = 1$. This forms a hemisphere of point features around each camera coordinate frame. An example initial keyframe for a four-camera cluster is shown in Figure 4. This rough target model is immediately available to the tracking thread to localize the camera cluster frame against.

The tracking and BA processes now run in parallel as described in the previous sections. Initially, the model is inaccurate, particularly in the global scale, due to the incorrect point feature depths, the small triangulation baselines, and the motion being close to degenerate. However, results have shown that the tracking thread is able to consistently localize with respect to the poor target model with a small amount of steady-state error and incorrect global scale. In fact, the small translational baselines between cluster poses are beneficial, in this case, since the incorrect point feature depth estimates do not produce large errors in the reprojection of those features onto the camera image plane when the motion is small. Therefore

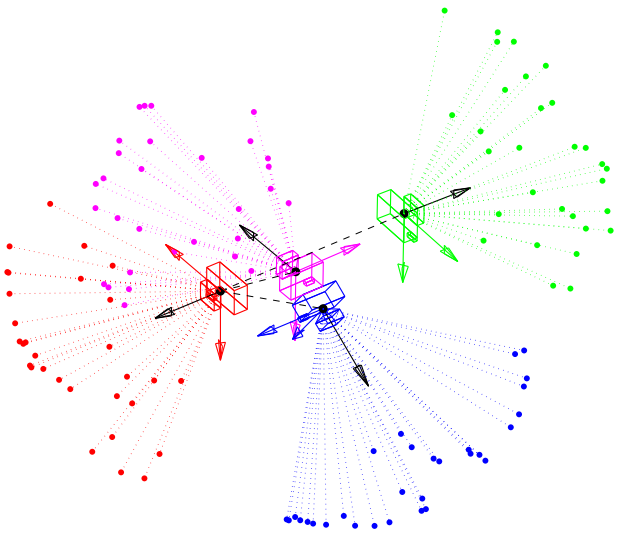


Figure 4: The initial permanent keyframe generated using the first set of camera images for an example four-camera cluster. The point estimates are generated using the bearing from the image measurement at an uncertain depth estimate, resulting in a set spheres of point feature estimates centred about the camera coordinate frame origins.

the initial tracking solution is not very sensitive to errors in the depth of the point features, in the local neighbourhood around the initial cluster pose. In the event that the camera cluster undergoes an extremely fast motion that makes the measurements more sensitive to scale, such as large rotations, the tracking thread may become lost if the target model is not updated in time.

The proposed algorithm continues in this manner as more keyframes are added and refined to improve the pose tracking process. As more permanent keyframes are accumulated, the optimization time required for the BA thread will begin to grow approximately cubically. This constrains the application of this method to tracking target objects and environments of a certain size, such that they are sufficiently modelled using a moderate number of keyframes. For general motion trajectories, the global scale of the target model will converge and the tracking thread is able to successfully and accurately localize the camera cluster despite only receiving the low-frequency refinements from the BA process.

5. Experimental Results

A set of experiments were carried-out to evaluate the performance of the new MCPTAM framework operating with image data collected from a multi-camera cluster mounted on an Unmanned Aerial Vehicle (UAV) in different operating environments and using different camera configurations.

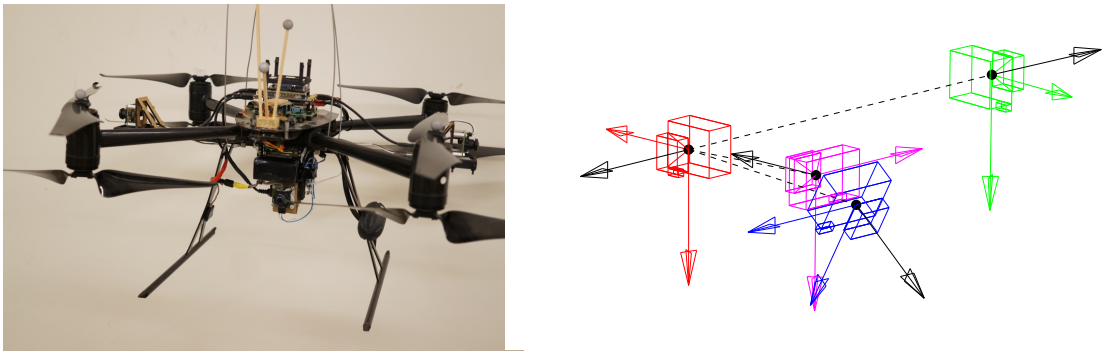
5.1. Setup

Four cameras were rigidly attached to a small multirotor aerial robot, shown in Figure 5a. The selected cluster configuration is shown in Figure 5b. Two spherical cameras with ultra-wide FOV of greater than 180 degrees are mounted on either side of the octocopter looking outwards. Two more cameras are mounted under the helicopter body, one facing directly forwards and the other facing down and back. The second two cameras have a smaller FOV of approximately 150 degrees. The individual cameras are intrinsically calibrated using the Taylor camera model (Scaramuzza, 2007). The extrinsic calibration for the camera cluster was performed using the method described in Harmat *et al.* (Harmat et al., 2014).

All four cameras have a pixel resolution of 752×480, and are synchronously triggered to capture images at the same instant in time. They have global shutters, as well as auto-exposure, auto-white balance, and hardware gamma correction enabled.{I-4} Each camera is connected to the onboard computer via a USB 2.0 connection. While the cameras are capable of capturing images at up to 30 Hz, the bandwidth of the USB 2.0 bus only allows the four cameras to deliver frames at a rate of 7 Hz. In the current implementation of the MCPTAM algorithm, this is the bottleneck as the tracking thread will comfortably run at a faster rate. The experiments were performed using a desktop with a quad-core 2.8 GHz processor and 4 GB of memory. Previously, the tracking process of MCPTAM has been run onboard the octocopter equipped with an Atom computer board with a 1.6 GHz processor and 1 GB of memory. The system was shown to achieve a tracking rate of 7 Hz using two cameras, or 2 Hz with four cameras (Harmat et al., 2014).

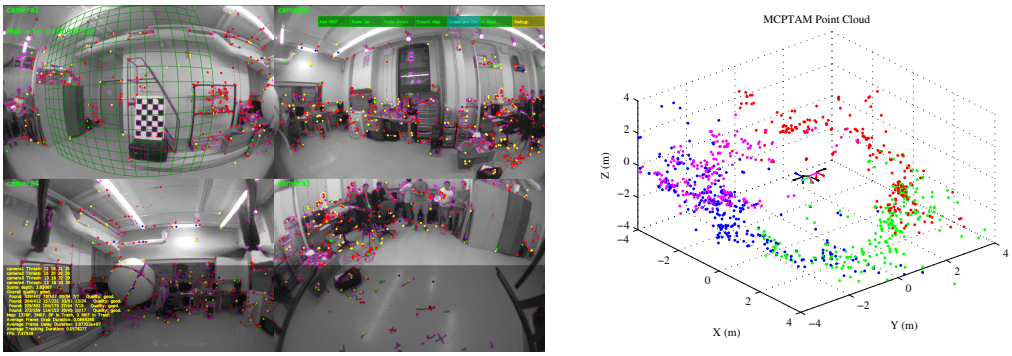
An example frame from the modified MCPTAM algorithm running with the four-camera cluster in the indoor lab environment, along with a generated map point cloud,{I-6} is shown in Figure 6. The coloured dots in the camera images{I-6} represent the point features detected and tracked at different image pyramid levels, while those in the point cloud represent the features anchored in the different camera frames{I-6}.

The four cameras within the cluster have significant overlap in their FOV simply because the viewing angle of the lenses is so extreme. In particular, almost all of the FOV in the front and rear-facing cameras can been seen in the side-mounted wide-angle cameras. In order to demonstrate that the proposed algorithm works with or



(a) Front view (b) Camera configuration

Figure 5: Four cameras are rigidly attached to an octocopter with cameras 1 (red) and 2 (green) facing outwards, camera 3 (blue) looking down and back, and camera 4 (magenta) forward.



(a) MCPTAM View (b) Map Point Cloud

Figure 6: A screenshot (a) of the MCPTAM algorithm running on a cluster of four cameras with FOV overlap, and the generated map point cloud (b) with coloured points corresponding to the camera frames in which they are anchored: 1 (red), 2 (blue), 3 (magenta), and 4 (green).{I-6}

without overlapping FOV, two configurations will be used in the experimental runs that follow:

Overlap (4 cameras) All four cameras use their entire FOV. Some point features will be seen by more than one camera at one time.

Non-overlap (3 cameras) The two side-facing cameras use their entire view, while the rear facing camera only uses a triangle at the bottom which does not overlap

with the sides. Point features are only seen by one camera at one time.

Videos of the experiments to follow, as well as other applications of MCPTAM, can be viewed online at http://wavelab.uwaterloo.ca/?q=multicamera.{I-7}

5.2. Computational Analysis

The measurements of timing for various per-frame functions within the tracking thread have been classified into three categories: detection; matching; and localization. Detection includes the items related to image processing for each set of camera frames: constructing the image pyramid; and extracting feature points. The matching category includes operations related to corresponding detected image features with existing map points, such as creating the set of potentially-visible features and determining correspondences with the measured image points. Finally, the localization category includes the nonlinear localization optimization tasks.

The detection category is a constant-time process for each set of camera images, while the matching timing is a linear function of the number of points in the map. The localization timing is also linear in the number of map points, but the process is limited to a maximum number of points for each camera. Graphs of the time required for each tracking phase for each set of camera images are presented in Figure 7 as a function of the number of point features in the map. In this example, a three camera configuration is used. The detection timing is the total over the three cameras and, as a result, is linear in the number of cameras within the cluster.

As detailed previously, the map-building thread runs in parallel with the tracking thread and performs local and global BA optimizations. The required execution time for each iteration of the BA optimization is shown in Figure 8 as a function of the number of keyframes in the map. From the indicated polynomial fittings, it can be seen that the iteration time is related to the number of keyframes as approximately $O(n^{2.4})$.{I-7}

The polynomial growth of the computational requirements places a limit on the number of keyframes that can feasibly be added to the target map. This complexity is due to the direct application of the LM algorithm to perform the BA optimization, and restricts the workspace volume in which the MCPTAM algorithm can be effective. The previously mentioned videos show the algorithm is able to map and track the environment through motions in a volume sized 30 m $\times$ 20 m $\times$ 10 m.

However, if large scale BA is required, several methods exist to which the camera configurations, state parameterizations, and initialization techniques proposed in this work can be readily applied, with the expectation of similar benefits to localization accuracy and robustness (Strasdat et al., 2011; Sibley et al., 2009; Konolige, 2010).{I-8}

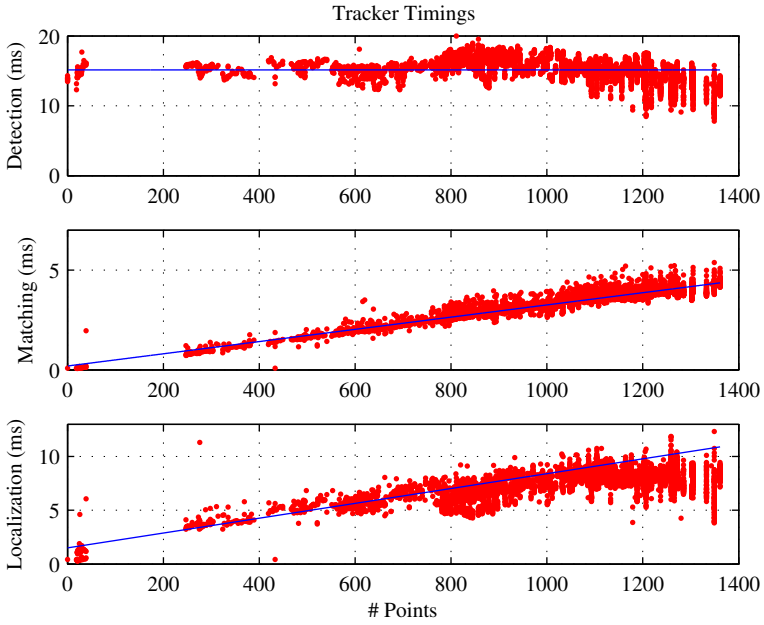


Figure 7: Computational time requirements for the three phases of the tracking process against the number of point features in the target map, for an example three-camera system.{I-7}

5.3. Procedure

While the modified MCPTAM algorithm is able to run on live image streams captured from the cluster cameras in real-time, for these experiments the images from the cameras during the octocopter flight were recorded, along with the measured pose of the robot within the Vicon indoor positioning system. Subsequently, the MCPTAM algorithm was run on the recorded camera images and the resulting pose trajectory from the tracker is captured and compared to the ground truth data from the Vicon system measurements for the indoor tests, and against one another for the outdoor tests. This is done to compare the different cluster configurations on the exact same image data to isolate their effect on the resulting pose estimates.

For the following tests, MCPTAM was run from initialization on the image sequence, generating the target model, and recording the tracked cluster poses (labelled 'initial'). Subsequently, MCPTAM continues to run and the image sequence was restarted. MCPTAM was able to relocalize with respect to the previously observed keyframes and continued to track the cluster pose within the target model. As the tracker localizes the cluster pose, it determines whether any of the previously

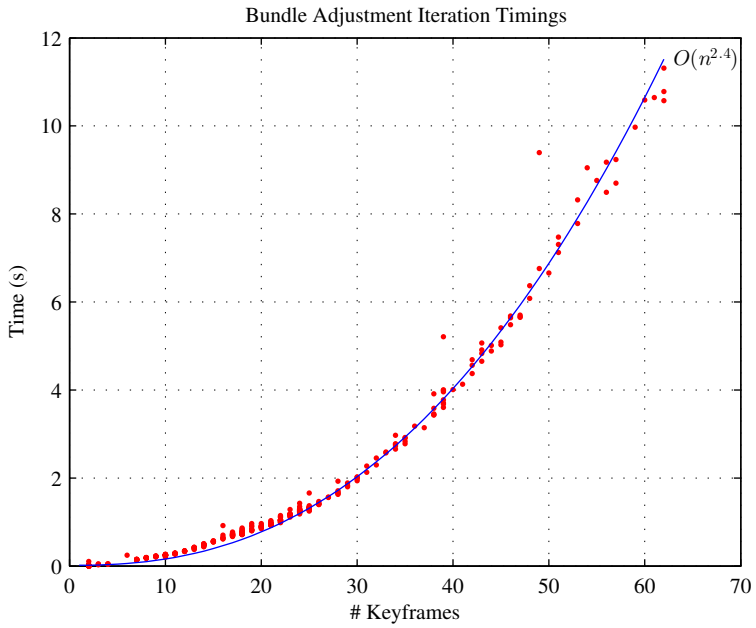


Figure 8: Computational time requirements for an iteration of the BA optimization against the number of keyframes in the target map.{I-7}

observed point features cannot be consistently relocated, and will discard them from the model. The resulting target model is stable and provides good localization constraints to track a high-accuracy pose estimate. The image sequence from the motion trajectory was then run for a third time and the pose estimates from the tracker were captured (labelled 'stable') for comparison with the Vicon ground truth data.

The system is tested in this way to demonstrate the performance of the tracker during initialization and afterwards once the map has stabilized. This simulates the performance of a system when the camera cluster stays in the same area of the environment long enough to reach a steady-state with regards to adding keyframes and refining the target model that it has generated. It also isolates the effects of initialization and map stability on the resulting pose estimates since the same image sequence was used.{I-5}

5.4. Indoor Tests

The Vicon system measurements are used to verify the accuracy of the MCPTAM algorithm both with and without overlapping FOV in the component cameras. It will be confirmed that with sufficient rotational motion, both configurations are able

to track the cluster pose accurately, including recovery of the global scale metric. Additionally, a degenerate pure translational motion identified in Section 3.6 was used to show situations where the non-overlap configuration is able to accurately recover an up-to-scale solution for the relative motion and structure. However, it is shown that the configuration with overlap is still able to accurately reconstruct the correctly-scaled motion in the degenerate case. This result is used to justify the comparison between the overlap and non-overlap configurations for the later outdoor tests where the Vicon system is unavailable. Finally, a hand-held motion sequence was used where the operator moves the camera cluster by hand and at times they occupy large portions of multiple camera FOVs and occludes parts of the target map. This test demonstrates the robustness of the camera configuration and algorithm to moving objects within the environment.{I-6}

5.4.1. Non-Degenerate Case – Large Rotation

The first trajectory tested was a general motion in which there are large translations and rotations within the full workspace of the Vicon system. The estimated pose from the stable run{I-5} and the Vicon ground truth measurement trajectories were aligned to compare the true error magnitudes of the position and orientation estimates from the two cluster configurations. The resulting aligned trajectories exhibit excellent agreement with the ground truth and are shown in Figure 9.

The magnitudes of the error in the position and orientation estimates, in millimetres and degrees, respectively, for both cluster configurations are shown in Figure 10. The Root Mean Squared Error (RMSE) of each configuration is shown as the dashed constant line on each graph. The overlap configuration shows an RMSE for position and orientation magnitudes of 5.5 mm and 0.42 deg, respectively. The non-overlap configuration has RMSE values of 9.9 mm and 0.47 deg.

To see how much of the error is associated with an incorrect global scale estimate, the estimated pose trajectories for both cluster configurations are aligned with the Vicon measurements using the trajectory alignment optimization with the scale factor set as variable. This produces a normalized trajectory and isolates the scale factor between the MCPTAM and Vicon pose measurements. The scale factor between the estimated pose and Vicon trajectories for the two cluster configurations were found to be 1.003 and 1.012 for the overlap and non-overlap configurations, respectively. These scale factors represent the ratio of the position magnitudes of the Vicon measurements to the MCPTAM position estimates. Accordingly, a scale factor greater than one means that the actual positions are larger than the MCPTAM estimates.

With the error due to incorrect scale removed from the estimates, the RMSE for the overlap configuration are 5.1 mm and 0.42 deg, while the RMSE of the non-

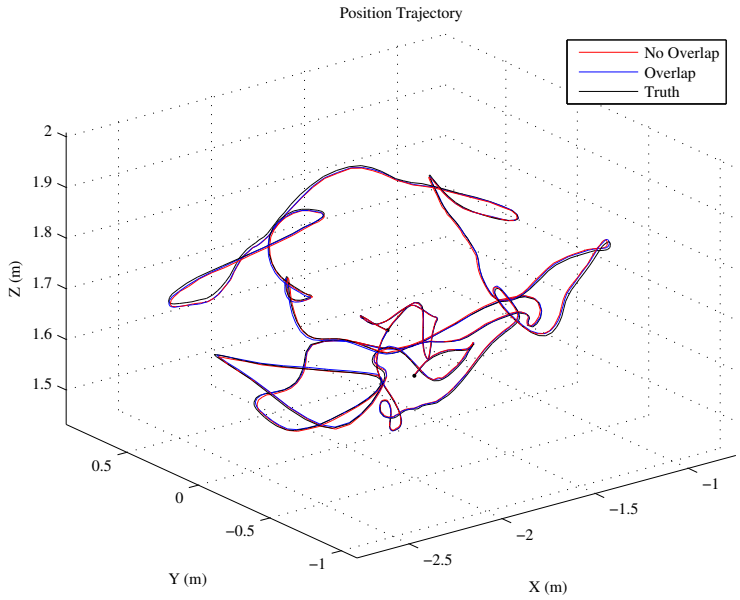


Figure 9: The position trajectories of the estimated pose during the stable map run,{I-5} from the overlap (blue) and non-overlap (red) configurations compared against the measured position from Vicon (black).

overlap configuration falls to 4.6 mm and 0.47 deg. It is clear that the orientation error magnitudes are almost identical to those with the scale error included, in Figure 10. This strongly indicates that both of the camera cluster configurations are able to estimate the up-to-scale solution effectively, even though there may be small errors in the scale recovery. This experiment confirms that when there is a large amount of rotation in the relative motion, both cluster configurations are able to recover accurate estimates of the cluster pose through the motion sequence.

In the above tests, the target model was initialized and{I-5} allowed to stabilize over two runs of the image sequence to allow it to produce the most accurate map possible to test the pose estimation. With the large rotational and translational motion, the previous results show{I-5} the BA thread was able to converge to an accurate, correctly-scaled solution for the map model. To test the performance of the system during the initial{I-5} run when keyframes are being added and the solution is uncertain, the pose estimates were collected immediately after start-up for the the two cluster configurations. The magnitudes of the estimated and measured positions are shown in Figure 11, along with the ratio of the estimated magnitudes over the

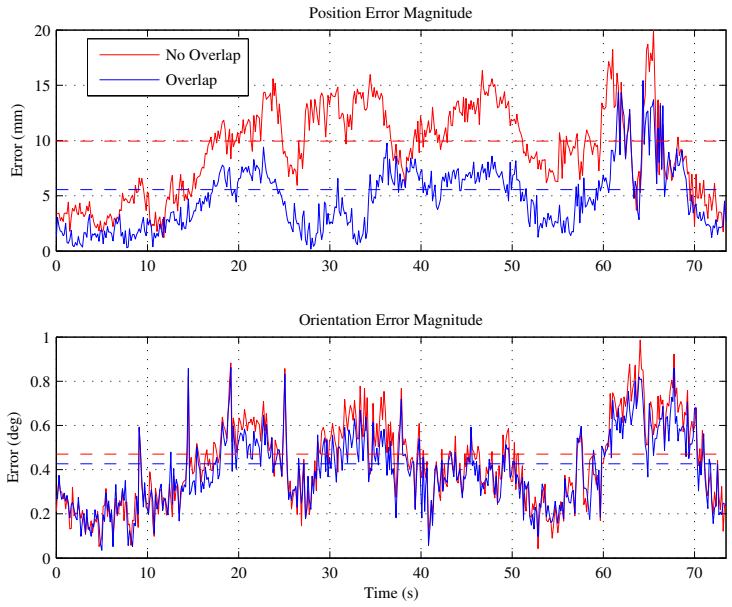


Figure 10: The magnitudes of the position and orientation errors during the stable map run,{I-5} for the overlap (blue) and non-overlap (red) configurations. The RMSE for each configuration are shown as the dashed constant lines.

Vicon measured position magnitudes. A correct scale value is indicated by a unity ratio.

The abrupt changes in the position estimates are caused by the BA thread providing a newly updated map model at that time step. Initially, both configurations provide poorly-scaled position estimates when the relative translation and orientation magnitudes are small compared to the initial cluster pose. As the motions evolve, the scale estimates vary but eventually converge to close to unity.

There is not any significant rotation until approximately 14 seconds into the trajectory. Therefore, the amount of rotation between any keyframes collected up to that point, is small and the motion is near-degenerate. After that time, the algorithm must choose to place a keyframe and complete the subsequent BA optimization before the proper scale is resolved. For the non-overlap configuration, an initial scale correction is observed at 14 seconds, followed by another at approximately 23 seconds that leads to greater accuracy due to further rotation between the permanent keyframes in the target model. With the overlap configuration, the positions are more accurate from the start, but the significant scale correction occurs around 18

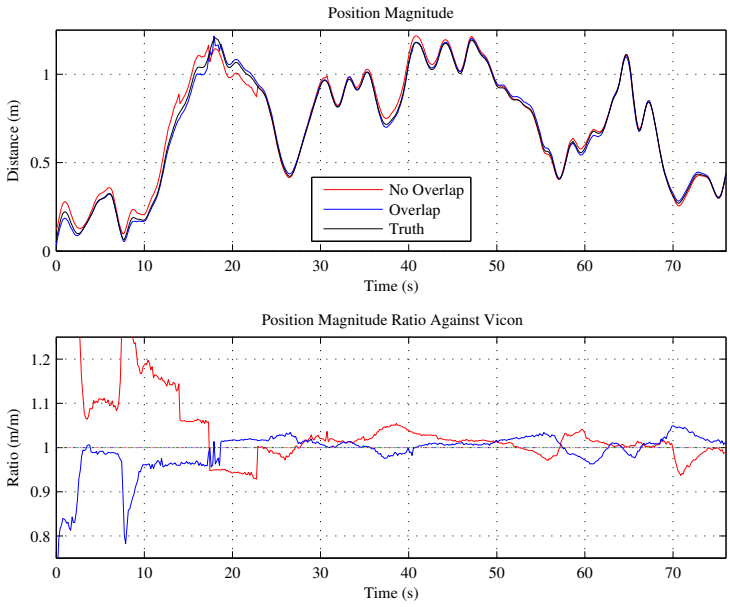


Figure 11: The position magnitudes (top) for the two cluster configurations against ground truth during the initial map run{I-5}. The ratio of the magnitudes (bottom) compared to Vicon position magnitude.

seconds. As more of the environment is explored and more permanent keyframes are added to the target model, both solutions become more accurate as the initial{I-5} run progresses.

Finally, the pose errors for the non-overlap configuration, collected during the initial and stable runs, are compared in Figure 12 to show the effects of map stability on tracking accuracy. At approximately 35 seconds, the pose estimates generated during the initial run have an accuracy close to that of the stable run. The RMSE over the initial run for the position and orientation estimates are 68 mm and 0.71 deg, respectively. The large position error magnitude is due to the poorly-scaled estimates upon initialization. Furthermore, if the average is found starting from 40 seconds, the RMSE on the position estimates falls to 16 mm. The results also demonstrate that the orientation estimates are accurate to within one degree immediately upon initialization, and only improve slightly after the map has stabilized.{I-5}

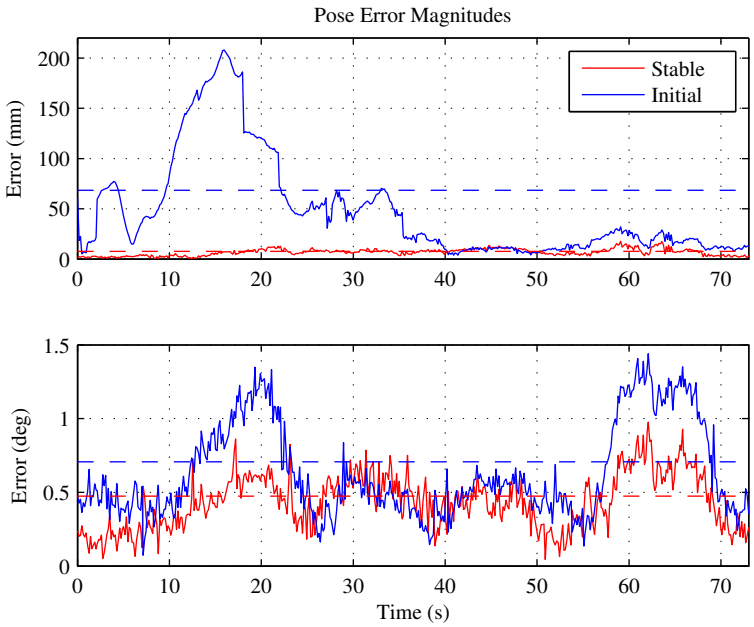


Figure 12: The position and orientation error magnitudes from the non-overlap configuration for the initial (blue) and stable (red) map runs against ground truth. The RMSE for each run are shown as the dashed constant lines.{I-5}

5.4.2. Near-Degenerate Case – Small Rotation

The second motion profile selected was one in which the amount of relative rotation was kept minimal. The octocopter vehicle was flown with a constant heading angle in yaw and the only rotation through the trajectory was due to subtle pitching and rolling motions to generate the translational motion. The maximum rotation angle relative to the initial cluster pose was approximately 9 degrees, while the translations were again large enough to cover the entire Vicon workspace.

Motion with small rotation is a scenario close to degeneracy for the non-overlap configuration, as discussed in Section 3.6, and the solution, even after the map stabilizes,{I-5} will be insensitive with respect to global scale. In the presence of measurement noise, the global scale will still converge, but the scale metric recovered will likely be incorrect.

The same processing procedure is carried out as with the previous motion trajectory, and the aligned trajectories during the stable map run{I-5} with the Vicon measurements are shown in Figure 13. It is immediately apparent that the non-overlap configuration is unable to recover the correct global scale of the solution.

However, the overlap configuration, due to the use of inter-camera correspondences, is still able to resolve the scale despite the small rotational motion. It is noteworthy that the recovered trajectory from the overlap configuration does not agree with the Vicon measurements as well as with the previous motion with large rotation. It is likely because the previously large rotational motion placed additional constraints on the solution which helped to more accurately recover the map model scale even when the FOV overlap is taken into account.

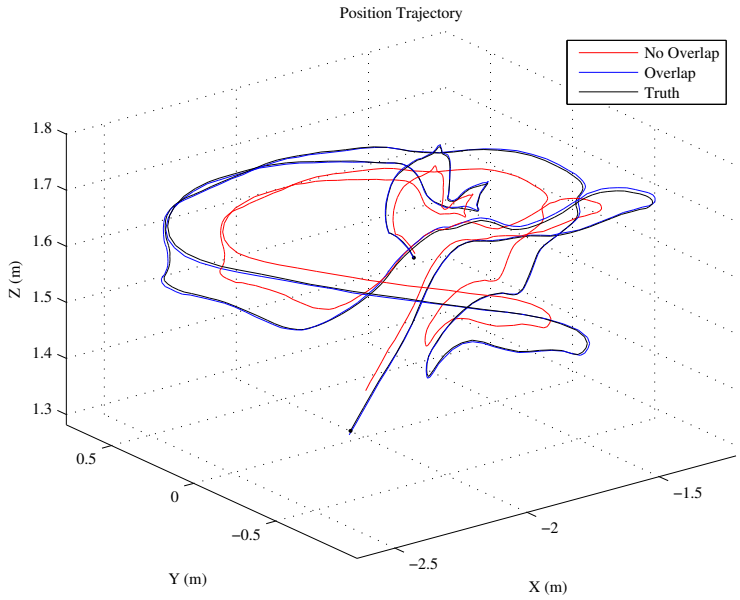


Figure 13: The position trajectories of the estimated pose from the overlap (blue) and non-overlap (red) configurations during the stable map run{I-5} compared against the measured position from Vicon (black).

As expected, the magnitudes of the position and orientation errors are large for the non-overlap configuration, and still relatively accurate, by comparison, for the overlap case. The RMSE for the non-overlap configuration are 125 mm and 0.21 deg, while those of the overlap configuration are 8.5 mm and 0.20 deg. It is clear that the overlap configuration is still able to recover an accurate solution when the non-overlap configuration experiences degenerate motion. As a result, the overlap scenario will be used to verify the solution of the non-overlap configuration for the outdoor test cases that follow in Section 5.5.

Most of the position error generated using this motion profile is associated with

an incorrect scale estimate. When the trajectories are aligned with the scale factor variable, the resulting position trajectories show good agreement, and the scale factors identified for the non-overlap and overlap configurations are 1.276 and 0.9874, respectively. The remaining position errors are reduced after the removal of the component related to scale and indicate that both configurations recover an accurate up-to-scale solution. As before, the orientation error magnitudes are similar to those found previously, but the position errors are significantly reduced, particularly for the non-overlap case, to result in an RMSE of 4.9 mm. The overlap configuration has a position RMSE of 4.3 mm.

5.4.3. Robustness Case – Moving Object

The final indoor test case was one in which the camera cluster was manually moved around within the Vicon workspace. Since the collective FOV of the cluster is large, the human operator appears in large sections of the images throughout the entire motion sequence. An example frame is shown in Figure 14 where the human operator is clearly visible in two different cameras. Because the operator was not stationary within the target environment, any detected image features on them could not be used to localize the cluster pose and must be ignored.

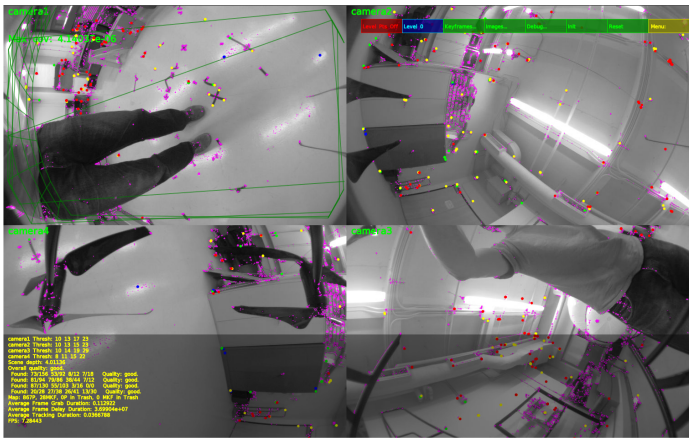


Figure 14: A screenshot of MCPTAM running with the overlap cluster configuration with the operator in view of two cameras, manually moving the cluster around the Vicon workspace.{I-6}

This test demonstrates the ability of MCPTAM to detect and reject unstable point features measured in the cameras, as well as to maintain accurate pose estimates of the cluster within the target map despite significant portions of some camera images containing few trackable point features. Despite the presence of the operator

throughout the entire image sequence, the error magnitudes in the pose estimates were small, as shown in Figure 15 for the stable map run of the overlap configuration. The RMSE for the position and orientation estimates are 10 mm and 0.40 deg, respectively, which do not exhibit significant degradation relative to unobstructed sequences presented previously.

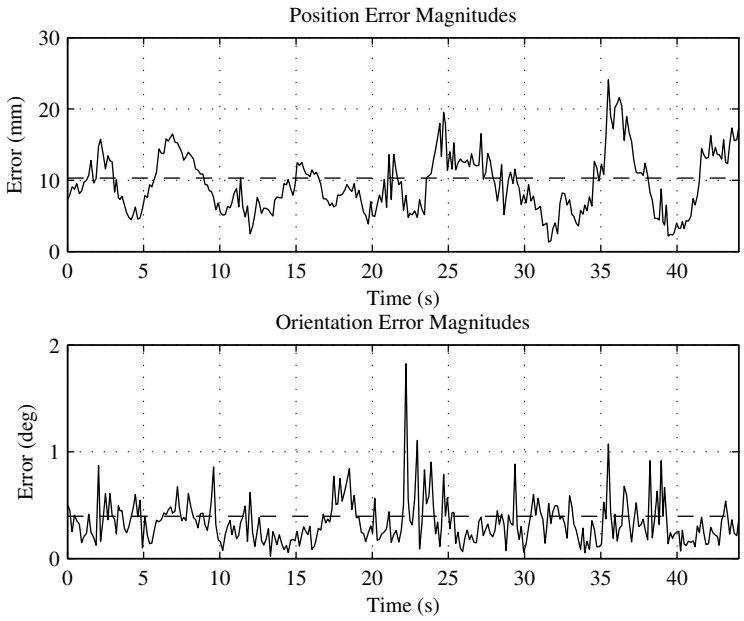


Figure 15: The magnitudes of the position and orientation errors with occlusions caused by the moving human operator throughout the trajectory. The RMSE are shown as the dashed constant lines.{I-6}

This test case confirms the robustness of both the large FOV cluster configuration, as well as the MCPTAM algorithm to non-stationary objects within the operating environment. A subsequent test in Section 5.5 further demonstrates the robustness of the large FOV configurations when significant sections of the environment lack suitable image texture for feature detection.{I-6}

The results from all of the indoor test cases are summarized in Table 1.

5.5. Outdoor Tests

The previous indoor tests allowed the MCPTAM system performance to be confirmed using the Vicon system providing ground truth pose measurements. To

Table 1: Indoor Test Results

Motion	Configuration	Scale	RMSE	
			Pos. (mm)	Rot. (deg)
Large Rotation	Overlap stable{I-5}	1	5.5	0.42
		1.003	5.1	0.42
	Non-overlap stable{I-5}	1	9.9	0.47
		1.012	4.6	0.47
	Non-overlap initial{I-5}	1{I-5}	68{I-5}	0.71{I-5}
Small Rotation	Overlap stable{I-5}	1	8.5	0.20
		0.9874	4.3	0.20
	Non-overlap stable{I-5}	1	125	0.21
		1.276	4.9	0.21
Moving Object{I-6}	Overlap stable{I-6}	1{I-6}	10{I-6}	0.40{I-6}

demonstrate the applicability of the proposed algorithm and implementation in real-world applications, the MCPTAM system was tested in an outdoor roof environment to verify that the system was capable of operating in a larger workspace with natural lighting conditions and difficult visual landmarks. Unfortunately, the Vicon system cannot operate outside due to the sunlight overpowering the reflections from the passive IR markers. As a result, the non-overlap cluster configuration was directly compared to the overlap configuration to determine the quality of the pose tracking algorithm when using non-overlapping FOV. The previous indoor test cases showed that the overlap configuration was able to recover an accurate relative motion trajectory, including the proper global scale, even when the non-overlap configuration experienced degenerate motion and could only recover an accurate up-to-scale solution.

In this test, the octocopter vehicle was flown around the large roof area and the cameras observed point features on the surrounding walls and ground. The yaw angle heading of the octocopter was varied during the flight to ensure there was sufficient orientation change in the cluster trajectory and allow the non-overlap configuration to avoid degenerate motions.

The environment can be seen in the example frame from MCPTAM during the execution with the overlap configuration, shown in Figure 16. Some of the walls were quite smooth and provided poorly textured surfaces which were devoid of any usable point features. This is a challenging scenario for any vision algorithm, particularly when the collective FOV is narrow, since the track can easily be lost when not enough features are visible in the camera images. However, the large collective FOV for the proposed cluster system allows the estimator to track any available point features since they are visible in at least one of the component cameras.

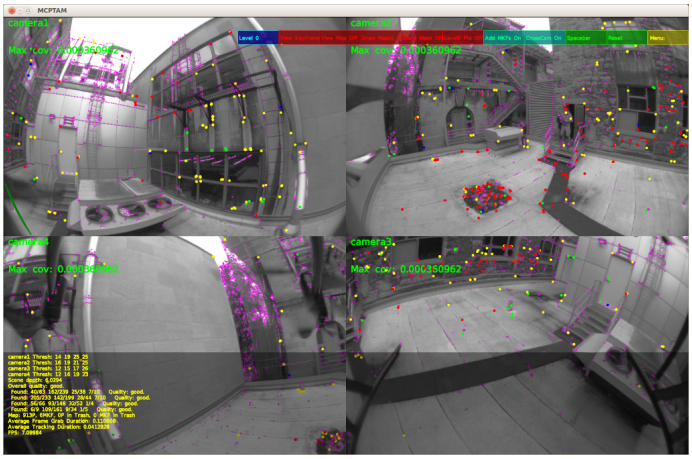


Figure 16: A screenshot of MCPTAM running with the overlap cluster configuration in the outdoor roof environment. The set of available point features is sparse in certain directions due to lack of texture. However, the large collective FOV of the cluster should be able to track any features available to prevent tracking failure.

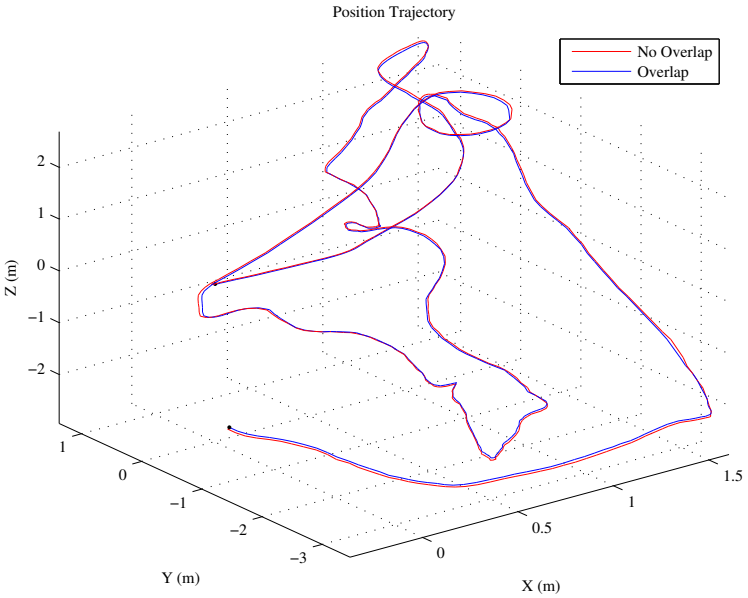


Figure 17: The position trajectories estimated by the non-overlap (red) and overlap (blue) configurations during the stable map run{I-5} for the outdoor roof flight.

Table 2: Outdoor Test Results

Scale	RMSE	
	Pos. (mm)	Rot. (deg)
1	24	0.15
0.9899	9.0	0.15

Both cluster configurations were able to track the relative pose consistently through the motion and the resulting aligned trajectories for the stable map run{I-5} show good agreement, as shown in Figure 17. The estimated global scale from non-overlap configuration is slightly larger than that recovered by the overlap case. Both configurations recover similar pose trajectories, and the RMSE between them for the stable map run{I-5} are 24 mm and 0.15 deg.

When the pose estimates from the non-overlap configuration are aligned with those from the overlap case and the scale factor is allowed to vary, it is observed that most of the position error is due to a small disagreement in the estimated scale. The scale factor between the two configurations was found to be 0.9899. With the overlap configuration assumed to recover the correct scale metric, the non-overlap configuration is able to recover the correct scale within approximately 1% of this value. The magnitudes of the pose errors after removing the scale-error component are significantly reduced to an RMSE of 9.0 mm for the non-overlap configuration.

The outdoor test results are summarized in Table 2. These results show that the two camera configurations produce consistent estimates of the pose trajectory of the camera cluster as the octocopter moves through the outdoor roof-top environment despite the challenging visual environment.

6. Conclusions

In this work, a novel visual SLAM framework based on $\boxplus$-manifolds was proposed for multi-camera clusters with non-overlapping FOV. Additionally, a new point feature position update operation was presented to isolate the estimation of the bearing and scaling effects, such that under critical motions, the shape of the motion and structure solution can converge, while the scale remains uncertain. Finally, a novel initialization scheme, specifically tailored to non-overlapping FOV cluster is provided.

The MCPTAM algorithm was modified to use the parameterization and initialization scheme from this work, allowing for multi-camera clusters to successfully track and model the target object or environment after being initialized using only the information in the first set of camera images. The proposed algorithm is able to run at real-time rates on camera images collected during motions

in both indoor and outdoor environments using natural image features. The new MCPTAM code is available as open-source software and can be downloaded at https://github.com/aharmat/mcptam.

The accuracy of the modified MCPTAM algorithm both during initialization and after map stabilization{I-5} was demonstrated by comparing the pose estimates from cluster configurations mounted on a multirotor aerial vehicle, both with and without FOV overlap, to ground truth pose measurements from a Vicon system. It was confirmed that providing the degenerate motions from Section 3.6 were avoided, the non-overlap configuration was able to estimate the relative pose of the cluster and target model to sub-centimetre and sub-degree accuracy. The performance of the estimator was presented for a challenging outdoor roof-top environment where sections of the environment were sparsely populated by usable point features. It was shown that the large collective FOV of the cluster configurations allowed the algorithm to maintain observations of the available point features and successfully track the cluster pose trajectory.

Future research into suitable keyframe selection processes for the case of multi-camera clusters is of particular importance. Given the unique degenerate configurations for these camera systems, it is vital to place the sparse set of keyframes at places within the image sequence that provide good constraints on the SLAM solution, such that an accurate properly-scaled estimate is recovered.

Acknowledgements

This work was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. CRDPJ 397768-10 and supported through the NSERC Canadian Field Robotics Network (NCFRN){AC}. Partial funding also comes from the NSERC through the Alexander Graham Bell Canada Graduate Scholarship - Doctoral (CGS-D) award.

Agrawal, M., 2006. A Lie algebraic approach for consistent pose registration for general Euclidean motion. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1891–1897.

Civera, J., Davison, A. J., Montiel, J. M. M., 2008. Inverse depth parametrization for monocular SLAM. IEEE Transactions on Robotics 24 (5), 932–945.

Clipp, B., Kim, J. H., Frahm, J. M., Pollefeys, M., Hartley, R., Jan. 2008. Robust 6DOF motion estimation for non-overlapping, multi-camera systems. In: Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV). pp. 1–8.

Crassidis, J., Markley, F., 1996. Attitude estimation using modified Rodrigues parameters. In: Proceedings of the Flight Mechanics/Estimation Theory Symposium. pp. 71–83.

Davison, A. J., 2003. Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Vol. 2. pp. 1403–1410.

Fermuller, C., Aloimonos, Y., 2000. Observability of 3D motion. International Journal of Computer Vision 37 (1), 43–63.

Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., Pollefeys, M., October 2012. Vision-based autonomous mapping and exploration using a quadrotor MAV. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4557–4564.

Furgale, P., Barfoot, T. D., 2010. Visual teach and repeat for long-range rover autonomy. Journal of Field Robotics 27, 534560.

Furgale, P. T., 2011. Extensions to the visual odomoetry pipeline for the exploration of planetary surfaces. Ph.D. thesis, University of Toronto.

Gupta, P., da Vitoria Lobo, N., Lariola Jr., J. J., 2013. Markerless tracking and gesture recognition using polar correlation of camera optical flow. Machine Vision and Applications 24 (3), 651–666.

Harmat, A., Trentini, M., Sharf, I., 2014. Multi-camera tracking and mapping for unmanned aerial vehicles in unstructured environments. Journal of Intelligent & Robotic Systems, 1–27.

Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge University Press.

Hertzberg, C., 2008. A framework for sparse, non-linear least squares problems on manifolds. Master's thesis, University of Bremen.

Hertzberg, C., Wagner, R., Frese, U., Schröder, L., 2013. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. Information Fusion 14 (1), 57–77.

Kaess, M., Dellaert, F., Feb. 2006. Visual SLAM with a multi-camera rig. Tech. Rep. GIT-GVU-06-06.

Kaess, M., Dellaert, F., 2010. Probabilistic structure matching for visual SLAM with a multi-camera rig. Computer Vision and Image Understanding 114 (2), 286–296.

Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., Siegwart, R., 2012. Real-time 6D stereo visual odometry with non-overlapping fields of view. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1529–1536.

Kim, J. H., Chung, M. J., Choi, B. T., 2010a. Recursive estimation of motion and a scene model with a two-camera system of divergent view. Pattern Recognition 43 (6), 2265–2280.

Kim, J. H., Hartley, R., Frahm, J. M., Pollefeys, M., 2007. Visual odometry for non-overlapping views using second-order cone programming. In: Proceedings of the Asian Conference on Computer Vision. Vol. 2. pp. 353–362.

Kim, J. H., Li, H., Hartley, R., Jun. 2010b. Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and L_∞ geometric solutions. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (6), 1044–1059.

Kim, J. S., Kanade, T., May 2010. Degeneracy of the linear seventeen-point algorithm for generalized essential matrix. Journal of Mathematical Imaging and Vision 37 (1), 40–48.

Klein, G., Murray, D., 2007. Parallel tracking and mapping for small AR workspaces. In: Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR). pp. 225–234.

Konolige, K., Aug. 2010. Sparse sparse bundle adjustment. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 102.1–102.11.

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

Lee, J. M., 2013. Introduction to smooth manifolds, Second Edition. Springer.

Li, H., Hartley, R., Kim, J. H., Jun. 2008. A linear approach to motion estimation using generalized camera models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8.

Longuet-Higgins, H. C., 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135.

Madhusudan, C., 1992. Error analysis of the Kalman filtering approach to relative position estimation using noisy vision measurements. Master's thesis, University of Waterloo.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P., 2009. Generic and real-time structure from motion using local bundle adjustment. Image and Vision Computing 27 (8), 1178–1193.

Pless, R., Jun. 2003. Using many cameras as one. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 2. pp. II–587–593.

Ragab, M. E., Wong, K. H., Sep. 2010. Multiple nonoverlapping camera pose estimation. In: Proceedings of the IEEE International Conference on Image Processing (ICIP). pp. 3253–3256.

Scaramuzza, D., 2007. Omnidirectional vision: from calibration to robot motion estimation. Ph.D. thesis, ETH Zurich.

Sibley, G., Mei, C., Reid, I., Newman, P., 2009. Adaptive relative bundle adjustment. In: Robotics: Science and Systems Conference (RSS). pp. 1–8.

Solà, J., Monin, A., Devy, M., Vidal-Calleja, T., Oct. 2008. Fusing monocular information in multicamera SLAM. IEEE Transactions on Robotics 24 (5), 958–968.

Strasdat, H., Davison, A. J., Montiel, J. M. M., Konolige, K., 2011. Double window optimisation for constant time visual SLAM. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2352–2359.

Tribou, M. J., Wang, D. W. L., Waslander, S. L., May 2014a. Degeneracies in multicamera cluster SLAM with non-overlapping fields of view, submitted to *Image and Vision Computing*.

Tribou, M. J., Waslander, S. L., Wang, D. W. L., 2014b. Scale recovery in multicamera cluster SLAM with non-overlapping fields of view. Computer Vision and Image Understanding 126 (0), 53–66.

Wilson, W. J., Hulls, C. C. W., Bell, G. S., 1996. Relative end-effector control using Cartesian position based visual servoing. IEEE Transactions on Robotics and Automation 12 (5), 684–696.

Yang, S., Scherer, S. A., Zell, A., 2014. Visual SLAM for autonomous MAVs with dual cameras. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C., Kaus, E., Herrtwich, R., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., Ghanaat, M., Braun, M., Joos, A., Fritz, H., Mock, H., Hein, M., Zeeb, E., 2014. Making Bertha drive? An autonomous journey on a historic route. Intelligent Transportation Systems Magazine 6 (2), 8–20.