# Scale Recovery in Multicamera Cluster SLAM with Non-overlapping Fields of View

Michael J. Tribou[a,*], Steven L. Waslander[a], David W. L. Wang[b]

[a] *Department of Mechanical and Mechatronics Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1.*
[b] *Department of Electical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1.*

**Abstract**

A relative pose and target model estimation framework using calibrated multicamera clusters is presented. It is able to accurately track up-to-date relative motion, including scale, between the camera cluster and the (free-moving) completely unknown target object or environment using only image measurements from a set of perspective cameras. The cameras within the cluster may be arranged in any configuration, even such that there is no spatial overlap in their fields-of-view. An analysis of the set of degenerate motions for a cluster composed of three cameras is performed. It is shown that including the third camera eliminates many of the previously known ambiguities for two-camera clusters. The estimator performance and the degeneracy analysis conclusions are confirmed in experiment with ground truth data collected from an optical motion capture system for the proposed three-camera cluster against other camera configurations suggested in the literature.

*Keywords:* Localization, Mapping, Multicamera cluster, Non-overlapping FOV, SLAM, Degeneracy analysis, Critical motions

*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@uwaterloo.ca (Michael J. Tribou), stevenw@uwaterloo.ca (Steven L. Waslander), dwang@uwaterloo.ca (David W. L. Wang)

## 1. Introduction

Many researchers have studied the use of visual feedback for environment mapping and camera localization, principally within two areas; mobile/industrial robotics, and computer vision. The robotics community refers to the problem as visual Simultaneous Localization and Mapping (SLAM) and frames it in terms of online tracking of a position and orientation state estimate (pose) with respect to an unknown environment through an image sequence [1]. On the other hand, the Structure From Motion (SFM) problem in computer vision focuses primarily on offline reconstruction of the 3D scene geometry using 2D image measurements over a (sparse) set of discrete camera frames [2].

Recently, the trend has been to move away from using monocular or conventional-stereo cameras to perform localization and structure estimation, to using more sophisticated imaging systems, including camera clusters [3]. A camera cluster is composed of any number of simple perspective cameras mounted rigidly with respect to each other. An example configuration is shown in Fig. 1.

The use of a multicamera cluster for localization and target modeling has several advantages over other camera systems. The individual cameras can be arranged into any configuration, including those with no spatial overlap in the camera fields-of-view (FOV) to make the most effective use of available camera pixels. Even without FOV overlap, non-zero baselines between the individual camera centres allows for full global motion scale recovery. Additionally, the cameras can be configured such that small translation-rotation motion ambiguities [4]{I-7} 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.

The intuition behind how a calibrated cluster is able to track its relative motion without finding correspondence between point features across cameras is as follows. Each individual camera is able to estimate its own local motion increment up-to-scale in its own frame using its image sequence. Using the known cluster calibration, these local motions are combined to find the unique cluster translation and rotation which includes the proper scale value. Therefore, the world scale is embedded in the camera cluster extrinsic

Figure 1: Multiple perspective cameras are arranged to form a camera cluster such that they cover as large an FOV as possible, even with no spatial overlap. This cluster consists of four cameras, two forward, one sideways and one backward-facing, at known positions and orientations.

calibration.{I-7}

This work presents an estimation system capable of tracking the relative position and orientation of a calibrated camera cluster with respect to an unknown target object or environment. Additionally, it is demonstrated through analysis and experimentation that a cluster composed of three or more non-collinear cameras is able to overcome the estimation degeneracies of two-camera systems identified in the literature [5, 6]. The nonlinear estimation problem can be solved using a single recursive filter that is initialized using only the first set of image measurements from the cluster cameras. Finally, the tracked target object or environment is free to move independently of the cluster motion since only relative measurements are used for the estimation, thereby solving a superset of the SLAM problem.

An ideal application of this algorithm would be one in which the camera cluster is surrounded by a moving target which is visible in the non-overlapping FOV of more than one of the component cameras at the same point in time. One example would be an aerial robot maneuvering in close

proximity to a moving ship during a docking or inspection task. In this case, both the camera cluster and the target are in free motion, and auxiliary sensors such as Global Positioning Satellite (GPS) receivers or Inertial Measurement Units (IMU) mounted on the robot cannot be used to estimate the relative motion since they provide a measure of only the cluster motion in an Earth-fixed coordinate frame.{I-5} Applicability of these multicamera clusters will increase with the continued miniaturization of robotic systems and sensors as it will be more likely that the vehicle will need to operate in or around a moving target.

The remainder of the paper is arranged as follows: Section 2 contains a detailed review of existing techniques using camera clusters for motion and structure estimation; Section 3 presents the proposed multicamera cluster pose estimation framework, including parameterizations and the initialization process; a novel degeneracy analysis for a three-camera cluster is presented in Section 4 to identify motion sets leading to solution scale ambiguity; experimental results demonstrating and evaluating the performance of the proposed method with a variety of cluster configurations are provided in Section 5; finally, conclusions are drawn in Section 6.

## 2. Related Work

The use of calibrated non-overlapping camera clusters for motion and structure estimation has largely been inspired by the work of Fermuller *et al.* [4]. In that work, it is demonstrated that a single perspective camera with a limited FOV will have fundamental problems estimating 3D motion due to confusion between some translations and rotations. However, the ambiguities disappear when the FOV is increased to cover the entire viewing sphere. Accordingly, Baker *et al.* [3] propose the Argus Eye camera cluster composed of six perspective cameras placed in back-to-back pairs on each of the Cartesian axes. Since the cameras have their optical centres displaced from each other, the system allows for full metric reconstruction of the relative pose and target model, including scale, without using traditional stereo point correspondence techniques. The advantages of this configuration were further confirmed by Pless [7] using the Fisher Information matrix for a variety of camera setups.

There are two important requirements for successful scale recovery with any multicamera setup. First, the distance between the camera centres must be sufficiently large compared to the depth of the point features. If the

features are too far away for a particular camera baseline length, the scale information is lost in the image measurement noise. This observation was used by Kim *et al.* in [8] to justify their assumption that a camera cluster can be approximated as a spherical camera when the features were far enough away relative to the distance between the camera centres. In that case, their assumption does not allow for the scale to be determined. If scale recovery with non-overlapping FOV is required, the ratio of the cluster baseline length to point feature depth must be kept sufficiently large. As a result, the method in the current work is best suited for applications such as docking, grasping, or close-pursuit. When this ratio is not suitable for scale recovery, a camera cluster estimator with the parameterization presented in Section 3 will recover an accurate up-to-scale solution, but the scale of the solution would remain uncertain.

The second requirement is that the relative motion of the cluster and target object or environment must not fall into the set of critical motions [5], for which the solution becomes degenerate and the global scale is ambiguous. The set of critical motions has previously been identified for a cluster consisting of two cameras [5, 6]. The two-camera cluster experiences critical motion whenever the two camera centres move in concentric circles with a common centre on the line through the optical centres of the cameras before and after the motion. This includes many common motions such as pure translations (circles of infinite radius), pure rotations, and constant radius turns. These profiles represent a large set of common motions and make scale recovery difficult.

Previous motion estimation algorithms using camera clusters can be categorized using two criteria. The first criterion relates to the point at which the camera images are combined to find the global cluster motion:

**decoupled** – Individual cameras in the cluster estimate their own local motion increment and global cluster motion is subsequently found by combining these motion estimates.

**coupled** – All camera image measurements are considered concurrently when generating the cluster global motion estimate.

Methods using the decoupled strategy, combining local motion estimates from individual cameras, suffer from the same issues of reduced accuracy and sensitivity to motion ambiguities as in monocular techniques, primarily stemming from limited FOV. Additionally, the combination step that resolves the

global motion and scale does not properly account for camera image measurement noise. Finally, all cameras must each observe a sufficient number of feature points in order to estimate their own local motion. A preferable solution is the coupled strategy which considers all image measurements across all cameras to determine the global motion directly.

The second criterion relates to whether the global cluster motion is accumulated through increments or determined relative to a generated target environment model:

**visual odometry (VO)** – Considers only a small set of current and previous camera frames to estimate the cluster motion increment, which is then accumulated into the global motion trajectory.

**localization** – Builds a model of the unknown target object or environment through the image sequence, and the cluster position and orientation is localized with respect to the model.

Because the sensitivity of global solution scale is low due to the baseline-to-feature depth ratio requirement, and the prevalence of critical motions, the scale of each motion increment in the VO methods will accumulate error in the global estimate which will not be corrected through the sequence. For this reason, it is beneficial to maintain a model of the target as in the localization methods. If there is an error in the estimated scale of the solution, it will be reflected in the generated model and can be corrected over time when the relative motion is not critical.

There are several decoupled VO methods in the literature [9, 10, 5, 11], and all suffer from the limitations of both decoupled and VO approaches. Li *et al.* improve on these by introducing a coupled VO method that uses the General Epipolar Constraint (GEC) [7] to linearly solve for the global motion of a non-overlapping camera cluster [12], similar to the eight-point algorithm [13] for a single camera.

Ragab and Wong [14] present a decoupled localization method in which they mount two back-to-back camera pairs on a robot. The camera cluster has non-overlapping FOV but each camera tracks its own motion using a separate extended Kalman filter (EKF), which are later combined to find global motion. There are also several coupled localization methods. In [15, 16], Kaess and Dellaert mount eight perspective cameras to a robot in a ring facing outwards. The system solves the SLAM problem, but requires either odometry, a dynamics model, or a good initial estimate to proceed. Both

Sola *et al.* [17] and Kim *et al.* [6] present coupled recursive SLAM systems which consider a stereo camera setup as two monocular cameras with some FOV overlap. The methods use an EKF to estimate the structure of the environment, the relative motion, and, for Sola *et al.*, even the orientation parameters of the cluster extrinsic calibration. The systems are similar to the one proposed in Section 3, however, these methods use the overlap in the FOV of the cameras and explicitly match point features across the cameras. This allows them to operate despite the critical motions of the two-camera cluster, but limits the collective FOV, and therefore, the accuracy of the estimation. The effect of the narrow FOV will be demonstrated in Section 5.

In this work, it is shown by analysis that adding a third camera to the system avoids almost all of the two-camera cluster critical motions. This mitigates the requirement for the small FOV overlap from previous systems, which in turn, improves the accuracy of the solution. In the next section, the coupled framework using a single estimator is presented. The system is able to resolve global scale, without FOV overlap, and is simple to implement and initialize for completely unknown target environments or moving objects from the first set of measurements.

## 3. Multicamera Cluster Pose Estimation

The calibrated multicamera cluster estimation in this work considers a set of one or more (preferably, three or more) rigidly-connected perspective cameras as a single vision sensor. This allows the estimation to proceed using a single recursive filter.The formulation used is similar to that found in the work of Sola *et al.* [17], as well as Kim *et al.* [6].

### *3.1. Pin-hole Camera Model*{I-8}

An individual perspective camera is modeled as a simple pin-hole imaging device, which maps 3D points onto a 2D plane called the image plane [18]. An example is shown in Fig. 2. The 3D point $\mathbf{p}^{C_i}$, expressed in the $i^{\text{th}}$ camera coordinate frame, $C_i$, is mapped to a particular camera pixel in the image plane, $I_i$, at the intersection of the line through $\mathbf{p}^{C_i}$ and the camera frame origin, $\mathbf{o}^{C_i}$, the optical centre.

It is assumed that each camera has been intrinsically calibrated using one of the many existing offline techniques [19] and the camera projection matrix, $\mathbf{K}_i$, is known.{I-8} Using homogeneous coordinates [20], $\mathbf{K}_i$ transforms the 3D

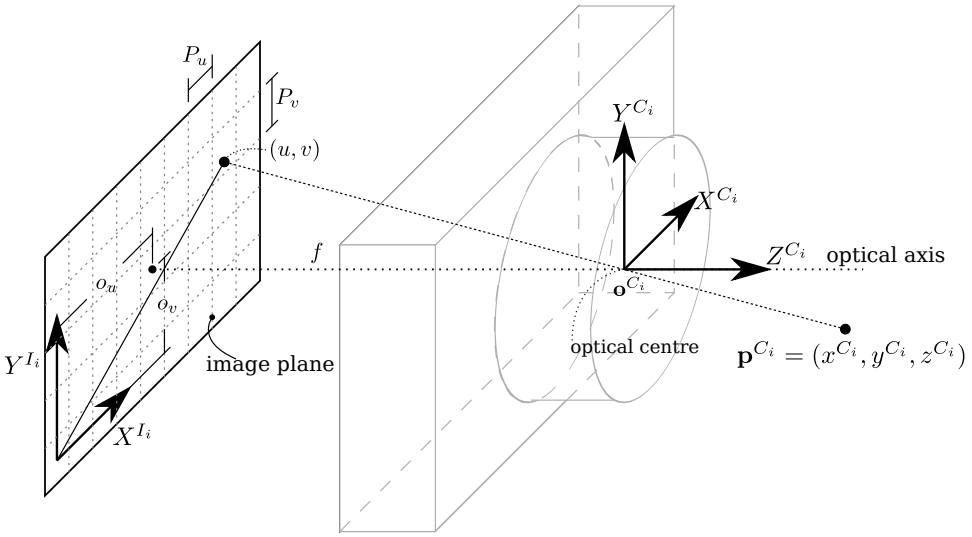


Figure 2: A simple pin-hole camera measurement model is used to relate the camera frame coordinates to the camera image plane coordinates for a feature point.{I-8}

point, $\mathbf{p}^{C_i}$, onto the image plane using the intrinsic calibration parameters,

$$\tilde{\mathbf{p}}^{I_i} = \mathbf{K}_i \tilde{\mathbf{p}}^{C_i} \tag{1}$$

$$= \begin{bmatrix} -\frac{f}{P_u} & 0 & o_u & 0 \\ 0 & -\frac{f}{P_v} & o_v & 0 \end{bmatrix} \begin{bmatrix} x^{C_i} \\ y^{C_i} \\ z^{C_i} \\ 1 \end{bmatrix}, \tag{2}$$

where $\tilde{\mathbf{p}}^{C_i} = \begin{bmatrix} (\mathbf{p}^{C_i})^\top 1 \end{bmatrix}^\top$ are the homogeneous coordinates of $\mathbf{p}^{C_i}$, $f$ is the focal length, $P_u$ and $P_v$ are the inter-pixel spacing of the camera sensor in the x and y axes, and $(o_u,\ o_v)$ are the coordinates on the pixel x-y plane at the intersection with the optical axis. The homogeneous coordinates of the point on the image plane are mapped to the corresponding pixel coordinates through $\boldsymbol{\pi} : \mathbb{R}^3 \to \mathbb{R}^2$ such that,

$$\boldsymbol{\pi}(\tilde{\mathbf{p}}^{I_i}) = \begin{bmatrix} u \\ v \end{bmatrix} \overset{\{I\text{-}10\}}{=} \begin{bmatrix} \frac{-f}{P_u}\frac{x^{C_i}}{z^{C_i}} + o_u \\ \frac{-f}{P_v}\frac{y^{C_i}}{z^{C_i}} + o_v \end{bmatrix}. \tag{3}$$

Collectively, the camera cluster is modeled as a set of $n_c$ pin-hole cameras with known relative coordinate transformations between each camera

coordinate frame. Accordingly, a point $\mathbf{p}^{C_h}$ in the camera frame $C_h$, can be transformed into any other camera frame $C_i$ by,

$$\tilde{\mathbf{p}}^{C_i} = \mathbf{T}_{C_h}^{C_i} \tilde{\mathbf{p}}^{C_h} \tag{4}$$

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.{I-7}

### *3.2. Target Model Representation*

The tracked target object or environment, henceforth referred to simply as the target, is assumed to be a rigid body which contains a set of visible point features. A point feature is a visually distinguishable point on the tracked physical target that corresponds to a unique 3D position in a local target coordinate frame and is measurable in a set of camera images through a relative motion sequence. Image measurements of these point features are extracted from the images using image processing techniques, including feature extraction algorithms like the Scale-Invariant Feature Transform (SIFT) [21], or Speeded-Up Robust Features (SURF) [22].

The locations of the point features are initially unknown to the estimation system, but are constrained to be fixed with respect to each other. This allows for the relative target pose to be fully characterized by six parameters representing the position and orientation of a local target model frame, $M$, with respect to the camera cluster frame, $C_1$.

The target model consists of a set of $n_k$ keyframes, each containing $n_c$ sets of $n_{f_{i,j}}$ point features, where $i = 1, \ldots, n_c$ and $j = 1, \ldots, n_k$. A keyframe consists of 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 captured at that location, as in [23] for a single camera. Each keyframe contains a set of features for which their positions are defined in that keyframe's coordinate frame.

To specify a particular keyframe coordinate frame, it is necessary to reference both the keyframe and the camera within the cluster. For example, the coordinate frame associated with camera $h$ at keyframe $k$ will be labeled $C_h K_k$. When expressing this camera coordinate frame at the current time step, the label $C_h K_r$ is used. The position of a point in the coordinate frame

of camera $h$ and keyframe $k$ is denoted $\mathbf{p}^{h,k}$. Additionally, the transformation from the coordinate frame associated with camera $h$ at keyframe $k$ to the coordinate frame of camera $i$ at keyframe $\ell$, will be written $\mathbf{T}^{i,\ell}_{h,k}$.

Since the relative position and orientation of each component camera within the cluster is fixed at all times, the $k^{\text{th}}$ keyframe pose is parameterized by the single homogeneous transformation for the cluster coordinate frame at the keyframe, $C_1K_k$, with respect to the target model reference frame, $M$, resulting in $\mathbf{T}^M_{C_1K_k} \in SE(3)$. The $C_1$ and $M$ frames are applied universally in this keyframe pose definition, and therefore, the transformation will be written simply as $\mathbf{T}_{K_k} \equiv \mathbf{T}^M_{C_1K_k}$ for notational clarity. The pose of camera $h$ at keyframe $k$ is easily found as,

$$\mathbf{T}^M_{C_hK_k} = \mathbf{T}_{K_k}\mathbf{T}_{C_h}. \tag{5}$$

{I-7}

The position of each point feature within a keyframe is represented using a refinement of the Inverse Depth Parameterization (IDP) introduced by Civera *et al.* in [24]. IDP encapsulates the relative position uncertainty arising from bearing-only measurement systems in a single inverse depth parameter instead of distributing it over all parameters used to define a feature location. Civera *et al.* demonstrate that the uncertainty in the inverse depth parameter is well-approximated by a Gaussian distribution for low-parallax motion. Additionally, the position is more effectively estimated using the EKF since the resulting measurement model has better linearity properties than when the feature parameters are the Cartesian coordinates. Since, the linearizations are valid over a larger region, propagating the Gaussian state estimates through the process and measurement equations results in more Gaussian-like distributions and the filter is able to provide more accurate estimates and converge over a larger set of initial conditions [24].{I-1}

In typical IDP, the feature point position in the local target model frame is the sum of an initial observation point and an observation ray. In the system presented here, the IDP is modified{I-7} such that the initial observation point for the point feature is the camera frame origin of the anchor keyframe in which it is first observed, as shown in Fig. 3. The position of the $j^{\text{th}}$ point feature in its anchor coordinate frame $C_hK_k$ is written,

$$\mathbf{p}^{h,k}_j = \frac{1}{\rho_j}\begin{bmatrix} \cos\alpha\sin\beta \\ -\sin\beta \\ \cos\alpha\cos\beta \end{bmatrix}_j \tag{6}$$

where $\alpha_j$, $\beta_j$ are the azimuth and altitude angles to the point feature, respectively, and $\frac{1}{\rho_j}$ is the distance along those bearings to the point feature position. By allowing many features to share this common point, the number of parameters is significantly reduced, from six per feature to three per feature plus six per keyframe, and the system is better constrained as a result.

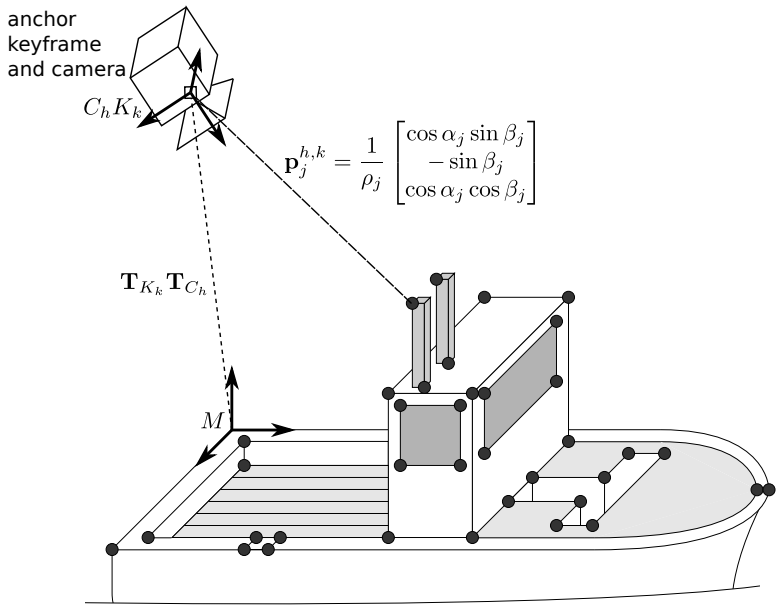


Figure 3: The position of each feature point is represented by the modified IDP and parameterized by the bearing and inverse depth within the coordinate frame of the camera and keyframe in which it was first observed.

At the start of the estimation, the initial cluster coordinate frame $C_1K_r$ is superimposed with the target model coordinate frame $M$, and is also selected as the first keyframe, $C_1K_1$,

$$\mathbf{T}_M^{1,r} = \left(\mathbf{T}_{1,1}^M\right)^{-1} = \mathbf{I}_{4\times 4} \tag{7}$$

$$M \equiv C_1K_1 \equiv C_1K_r, \quad \text{at } t = 0, \tag{8}$$

where $\mathbf{I}$ is the identity matrix of the specified dimensions.

*3.3. System Representation*

The full relative pose system can be represented as a nonlinear discrete-time state-space system by choosing an appropriate set of states, $\mathbf{x} \in \mathbb{R}^n$,

controllable inputs, $\mathbf{u} \in \mathbb{R}^p$, and measured outputs, $\mathbf{z} \in \mathbb{R}^m$, as well as suitable process and measurement models to represent the system dynamics and output, $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $\boldsymbol{\pi} : \mathbb{R}^n \to \mathbb{R}^m$, at time step $t$,

$$\mathbf{x}_{t+1} = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t) + \boldsymbol{\eta}_t, \tag{9}$$

$$\mathbf{z}_t = \boldsymbol{\pi}(\mathbf{x}_t) + \boldsymbol{\gamma}_t, \tag{10}$$

where $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}_{n\times 1}, \mathbf{Q}_t)$, $\mathbf{Q}_t \in \mathbb{R}^{n\times n}$, and $\boldsymbol{\gamma}_t \sim \mathcal{N}(\mathbf{0}_{m\times 1}, \mathbf{R}_t)$, $\mathbf{R}_t \in \mathbb{R}^{m\times m}$, are vectors of zero-mean Gaussian disturbance and measurement noise, respectively.

In this combined state and parameter estimation problem, the full system state vector is formed by augmenting the relative position and orientation motion states of the cluster and target, with the target model parameters. The state vector,

$$\mathbf{x} = \begin{bmatrix} \mathbf{w}^\top & \mathbf{k}^\top & \mathbf{f}^\top \end{bmatrix}^\top, \tag{11}$$

consists of three components: current camera cluster relative pose; keyframe poses; and feature parameters. The first set of states represent the current pose of the target model frame, $M$, with respect to the current cluster frame, $C_1K_r$.

$$\mathbf{w} = \begin{bmatrix} \mathbf{t}_w{}^\top & \boldsymbol{\omega}_w{}^\top \end{bmatrix}^\top \in \mathbb{R}^6, \tag{12}$$

where $\mathbf{t}_w \in \mathbb{R}^3$ is the relative translation vector and,

$$\boldsymbol{\omega}_w = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^\top \in \mathbb{R}^3 \tag{13}$$

is the relative orientation vector, which is expressed using modified Rodrigues parameters (MRP) [25].

The camera-centric convention used here is the inverse of the usual parameterization in SLAM algorithms, but offers better measurement model linearity with respect to the state variables [26]. Accordingly, the homogeneous transformation from the target model frame to the current cluster frame $C_1K_r$, is,

$$\mathbf{T}_M^{1,r} \equiv \mathbf{T}_{K_r}{}^{\{\text{I-7}\}} = \begin{bmatrix} \boldsymbol{\mathcal{R}}(\boldsymbol{\omega}_w) & \mathbf{t}_w \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}, \tag{14}$$

where $\boldsymbol{\mathcal{R}} : \mathbb{R}^3 \to SO(3)$ is the rotation matrix formed by the MRP.

The second set within the states contains the keyframes associated with the camera cluster and are represented by six parameters for each keyframe, expressing the pose of the camera cluster frame with respect to the target model frame when the keyframe is captured. Accordingly, the keyframes vector is composed of the parameters for all of the keyframes,

$$\mathbf{k} = \begin{bmatrix} \mathbf{k}_1{}^\top & \mathbf{k}_2{}^\top & \dots & \mathbf{k}_{n_k}{}^\top \end{bmatrix}^\top, \tag{15}$$

with the $\ell^{\text{th}}$ keyframe parameterized by six parameters,

$$\mathbf{k}_\ell = \begin{bmatrix} \mathbf{t}_k{}^\top & \boldsymbol{\omega}_k{}^\top \end{bmatrix}_\ell^\top \in \mathbb{R}^6, \tag{16}$$

where $\mathbf{t}_k \in \mathbb{R}^3$ is the relative translation vector for the keyframe, and $\boldsymbol{\omega}_k \in \mathbb{R}^3$ is the relative orientation vector, which together represent the transformation $\mathbf{T}_{K_\ell}$.

The final set of states are the point feature parameters. Each point feature in the target model is anchored in its associated keyframe within which its local position is estimated. Thus, the $j^{\text{th}}$ feature point is represented by its three parameters in the system state,

$$\mathbf{f}_j^{h,k} = \begin{bmatrix} \alpha & \beta & \rho \end{bmatrix}_j^\top \in \mathbb{R}^3, \tag{17}$$

which describe a ray and depth to feature point $j$ in the anchor keyframe $C_h K_k$ using the modified IDP.

### *3.4. Constant-Velocity Motion Model*

When using a recursive filter, such as the EKF, it is necessary to provide a motion model which describes the dynamics of the system states. The relative motion dynamics are approximated by the generic constant velocity process model [1].{I-5} This model assumes that the time-derivatives of the six pose parameters are subject to random walk [27], and integrates them into the position states accordingly. At time step $t$,

$$\mathbf{w}_{t+1} = \mathbf{G}_w \mathbf{w}_t + \boldsymbol{\eta}_w, \tag{18}$$

where the pose vector is augmented with the time-derivatives of the motion states so that $\mathbf{w}_t \in \mathbb{R}^{12}$, and $\boldsymbol{\eta}_w \sim \mathcal{N}\left(\mathbf{0}_{12\times 1}, \mathbf{N}_w\right)$ is a vector of disturbance noise,

$$\mathbf{G}_w = \begin{bmatrix} \mathbf{I}_{6\times 6} & \delta_t \mathbf{I}_{6\times 6} \\ \mathbf{0}_{6\times 6} & \mathbf{I}_{6\times 6} \end{bmatrix}, \tag{19}$$

while $\mathbf{0}$ is the zero matrix of the specified dimensions, and $\delta_t$ is the sampling period, given by the frame rate of the cameras.

The target model parameters are modeled as random constants,

$$\mathbf{k}_{t+1} = \mathbf{k}_t \tag{20}$$

$$\mathbf{f}_{t+1} = \mathbf{f}_t, \tag{21}$$

which converge to the correct value given some initial uncertainty and are not subject to disturbance noise.

The resulting system dynamics are linear and for the $t^{\text{th}}$ time step,

$$\mathbf{x}_{t+1} = \mathbf{G}_t \mathbf{x}_t + \boldsymbol{\eta}_t, \tag{22}$$

where $\boldsymbol{\eta}_t = \left[ \begin{array}{cc} \boldsymbol{\eta}_w{}^\top & \mathbf{0}_{(6n_k+3n_f)\times 1}{}^\top \end{array} \right]^\top$ is the system disturbance noise,

$$\mathbf{G}_t = \boldsymbol{\Psi}\left(\mathbf{G}_w, \mathbf{I}_{6n_k \times 6n_k}, \mathbf{I}_{3n_f \times 3n_f}\right), \tag{23}$$

where $\boldsymbol{\Psi}\left(\right)$ creates a block-diagonal matrix using the arguments, and $n_f = \sum_{i,j} n_{f_{i,j}}$ is the total number of point features in the model.

### *3.5. Camera Cluster Measurement Model*

The measurement model, relating the observed feature point locations in the camera image planes, to the system states, can be written as a series of coordinate transformations. Suppose that at the current time step, the feature $\mathbf{f}_j^{h,k}$ is observed in the image plane of camera $C_i$. The feature parameters give the location of the $j^{\text{th}}$ feature at its anchor keyframe, $K_k$, in camera, $C_h$, resulting in $\mathbf{p}_j^{h,k}$ from (6).

This point is first transformed into the target model coordinate frame by,

$$\tilde{\mathbf{p}}_j^M = \mathbf{T}_{h,k}^M \tilde{\mathbf{p}}_j^{h,k} \tag{24}$$

$$= \mathbf{T}_{K_k} \mathbf{T}_{C_h} \tilde{\mathbf{p}}_j^{h,k}, \tag{25}$$

which are transformations provided by the known cluster calibration and the keyframe parameters.

Next, the point is transformed into the frame of camera $C_i$ at the current cluster pose $K_r$ through the primary camera coordinate frame, using the cluster relative pose states and the cluster calibration,

$$\tilde{\mathbf{p}}_j^{i,r} = \mathbf{T}_M^{i,r} \tilde{\mathbf{p}}_j^M \tag{26}$$

$$= \left(\mathbf{T}_{C_i}\right)^{-1} \mathbf{T}_{K_r} \tilde{\mathbf{p}}_j^M. \tag{27}$$

Finally, the point is projected onto the image plane of camera $C_i$ using the corresponding projection matrix, $\mathbf{K}_i$,

$$\tilde{\mathbf{p}}_j^{I_i} = \mathbf{K}_i \tilde{\mathbf{p}}_j^{i,r}, \tag{28}$$

known from the intrinsic calibration of the individual cluster cameras.

Each of the four intermediate transformation matrices are formed by either the system states, or the known cluster camera configurations from extrinsic calibration. Therefore, the full measurement equation for this $j^{\text{th}}$ feature is,

$$\mathbf{z}_j^{i,r} = \begin{bmatrix} u_j^{i,r} \\ v_j^{i,r} \end{bmatrix}^{\{\text{I-10}\}} = \boldsymbol{\pi}_j(\tilde{\mathbf{p}}_j^{I_i}) + \boldsymbol{\gamma}_j \tag{29}$$

$$\tilde{\mathbf{p}}_j^{I_i} = \mathbf{K}_i (\mathbf{T}_{C_i})^{-1} \mathbf{T}_{K_r} \mathbf{T}_{K_k}^{\{\text{I-11}\}} \mathbf{T}_{C_h} \tilde{\mathbf{p}}_j^{h,k}, \tag{30}$$

where $\boldsymbol{\gamma}_j \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \mathbf{R}_j)$. An example of this chain of transformations is shown for a simple back-to-back two-camera cluster in Fig. 4.

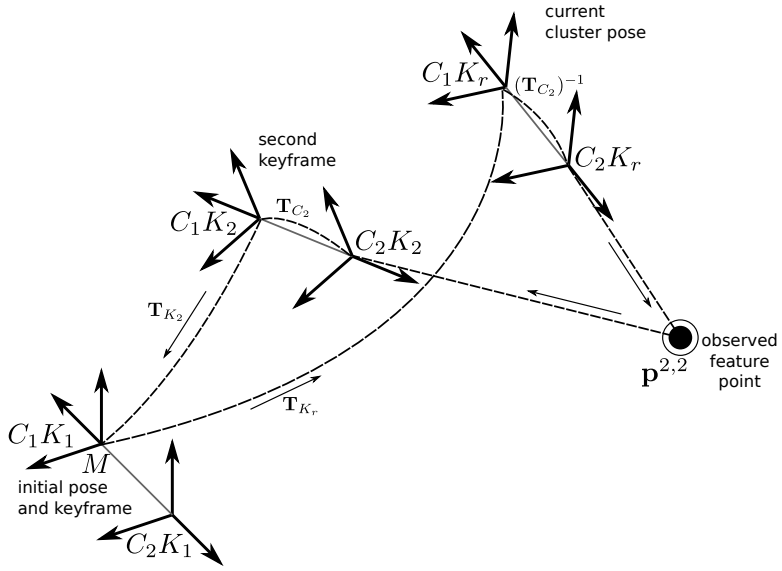


Figure 4: The transformations, for a two-camera back-to-back cluster, involved in finding the predicted image plane measurements of a point $\mathbf{p}^{2,2}$ anchored in $C_2$ of keyframe $K_2$, observed in $C_2$ at the current cluster pose, $K_r$, with respect to the target model frame, $M$.{I-12}

The full system measurement vector $\mathbf{z}$ is made up of all of the individual point feature observations at the current time step. It is modeled as a stacked column vector of measurements of the form (29).

With this state-space representation, it is now a matter of applying the recursive filter to the set of camera images as they are retrieved at each time step to produce an estimate of the full system state.

*3.6. Initialization*

An advantage of the proposed formulation is the ability to initialize the estimator system and successfully track the relative pose and target model structure through a trajectory from the first set of cluster image measurements with no prior knowledge of the target or motion. At start-up, the initial cluster pose is fixed coincident with the target model frame,

$$\breve{\mathbf{w}}_0 \equiv \mathbf{0}_{6\times 1}, \tag{31}$$

where $\breve{\mathbf{x}}_t$ denotes the estimate of the vector $\mathbf{x}_t$ at time step, $t$. This choice is arbitrary but fixes six degrees of freedom in the solution since it is only relative motion that is being estimated.

The initial cluster pose is also chosen as the first keyframe pose in the target model,

$$\breve{\mathbf{k}}_0 \equiv \mathbf{0}_{6\times 1} \tag{32}$$

This keyframe pose is also known with perfect certainty and the parameters are not subject to disturbance noise. Technically, this first keyframe pose does not need to be included in the state vector since it has zero uncertainty and is not subject to disturbance noise. However, it is included here for clarity and consistency.

Finally, all of the parameter estimates for the point features observed in each of the first set of camera images are initialized using their image plane measurements,

$$\breve{\mathbf{f}}_0 = \begin{bmatrix} \begin{bmatrix} \breve{\mathbf{f}}_1^{1,1} \\ \breve{\mathbf{f}}_2^{1,1} \\ \vdots \\ \breve{\mathbf{f}}_{n_{f_{1,1}}}^{1,1} \end{bmatrix}^\top & \cdots & \begin{bmatrix} \breve{\mathbf{f}}_1^{n_c,1} \\ \breve{\mathbf{f}}_2^{n_c,1} \\ \vdots \\ \breve{\mathbf{f}}_{n_{f_{n_c,1}}}^{n_c,1} \end{bmatrix}^\top \end{bmatrix}^\top. \tag{33}$$

Individually, the feature parameters are initialized as,

$$\breve{\mathbf{f}}_j^{i,1} = \begin{bmatrix} \alpha_0 & \beta_0 & \rho_0 \end{bmatrix}^\top \tag{34}$$

by calculating $\alpha_0$ and $\beta_0$ as the bearing to the image coordinate in the camera frame and choosing $\rho_0$ according to the following method. Suppose that the designer is confident that the features will have a depth on the interval $[d_{min}, d_{max}]$, then the associated inverse depth estimate is set to,

$$\rho_0 = \frac{1}{2}\left(\frac{1}{d_{max}} + \frac{1}{d_{min}}\right). \tag{35}$$

Notably, this parameterization can handle features infinitely far from the camera, $d_{max} \to \infty$.

The estimation is now ready to proceed iteratively or recursively until the algorithm determines, through some set of heuristics on the current state (e.g. normed distance of cluster pose parameters to set of keyframe poses, minimum number of estimated point features are observed, threshold on current pose uncertainty, etc.) that it should add another keyframe to the model. At this time, the estimated state vector is augmented with another set of six keyframe pose parameters extracted from the current relative pose transformation through $(\mathbf{T}_{K_r})^{-1}$. Subsequently, parameters for the set of newly observed point features in each camera image are augmented to the state vector and initialized as previously described for start-up.

## 4. Degenerate Configurations

For a camera cluster in general motion, the scale of the environment can be determined using the known extrinsic calibration of the cluster cameras. However, there exist a set of critical motions, dependent on the geometry of the camera cluster, for which the solution to the estimation problem is degenerate and the system scale cannot be recovered. In other words, the shape of the motion and target object are found, but not the absolute size.

The critical motions of a cluster consisting of two cameras have been studied by Clipp *et al.* [5] and Kim *et al.* [6]. It was demonstrated that the scale cannot be determined when the cluster motion causes the camera centres to move in concentric circles, as shown in Fig. 5. This comprises a large portion of typical robotic or vehicular motion (e.g. a robot arm with revolute joints, a car travelling in a straight line or in a constant-radius turn). During these motions the scale of the environment is unrecoverable.

In this section, it is shown that adding a third camera to the cluster significantly reduces the set of critical motions when compared with the two-camera cluster case. Consider the combined state and parameter estimation

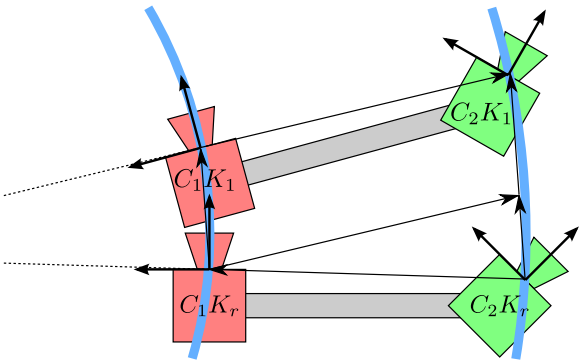


Figure 5: The motion of a two-camera cluster is critical when the centres of the cameras move in concentric circles.{I-7}

for a non-overlapping three-camera cluster observing a set of point features over two keyframes, as shown in Fig. 6. Each camera has its own mutually exclusive set of point features which it observes at both keyframes. That is, if a point feature is observed by camera $i$ at the first keyframe, it is observed by only camera $i$ at the current keyframe and no other cameras. The correspondence of point features is only performed across keyframes within the same camera.

### *4.1. System Parameterization*

As shown in (30), the system measurements are parameterized as a set of four transformations, followed by a projection. Before the projection, the position of the point feature defined in camera $h$ and keyframe $k$, transformed into the coordinate frame of the observing camera $i$ and keyframe $r$ is written,

$$\tilde{\mathbf{p}}_j^{i,r} = (\mathbf{T}_{C_i})^{-1}\mathbf{T}_{K_r}\mathbf{T}_{K_k}\mathbf{T}_{C_h}\tilde{\mathbf{p}}_j^{h,k}. \quad \{\text{I-7}\} \tag{36}$$

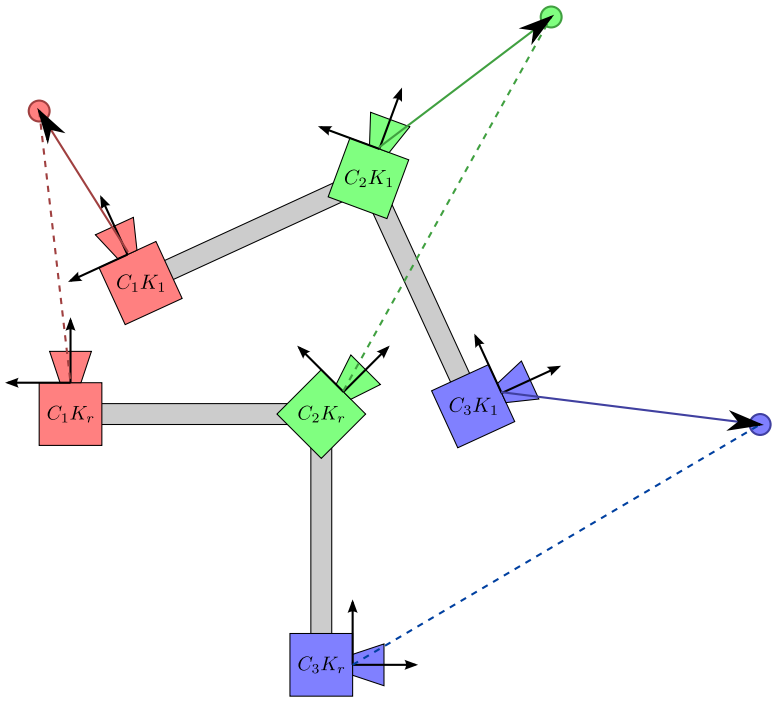


Figure 6: The three-camera cluster observes point features over two keyframes.

Each of these intermediate transformations can be represented by a rotation matrix and translation vector,

$$\mathbf{T}_{C_h} = \begin{bmatrix} \boldsymbol{\mathcal{R}}_{C_h} & \mathbf{t}_{C_h} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{37}$$

$$\mathbf{T}_{K_k} = \begin{bmatrix} \boldsymbol{\mathcal{R}}_{K_k} & \mathbf{t}_{K_k} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{38}$$

$$\mathbf{T}_{K_r} = \begin{bmatrix} \boldsymbol{\mathcal{R}}_{K_r} & \mathbf{t}_{K_r} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{39}$$

$$\mathbf{T}_{C_i} = \begin{bmatrix} \boldsymbol{\mathcal{R}}_{C_i} & \mathbf{t}_{C_i} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{40}$$

When these values are substituted into equation (36), it becomes,

$$\begin{aligned} \mathbf{p}_j^{i,r} = \boldsymbol{\mathcal{R}}_{C_i}^\top \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{K_k} \boldsymbol{\mathcal{R}}_{C_h} \mathbf{p}_j^{h,k} + \boldsymbol{\mathcal{R}}_{C_i}^\top \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{K_k} \mathbf{t}_{C_h} \\ + \boldsymbol{\mathcal{R}}_{C_i}^\top \boldsymbol{\mathcal{R}}_{K_r} \mathbf{t}_{K_k} + \boldsymbol{\mathcal{R}}_{C_i}^\top \mathbf{t}_{K_r} - \boldsymbol{\mathcal{R}}_{C_i}^\top \mathbf{t}_{C_i}. \end{aligned} \tag{41}$$

This is the most general form of the transformation chain. In this section it is assumed that each point feature is observed by only one camera, and therefore,

$$\boldsymbol{\mathcal{R}}_{C_i} = \boldsymbol{\mathcal{R}}_{C_h} \tag{42}$$

$$\mathbf{t}_{C_i} = \mathbf{t}_{C_h}. \tag{43}$$

Furthermore, the point feature is assumed, without loss of generality, to be parameterized in the first keyframe coordinate frame,

$$\boldsymbol{\mathcal{R}}_{K_k} = \boldsymbol{\mathcal{R}}_{K_1} = \mathbf{I}_{3\times 3} \tag{44}$$

$$\mathbf{t}_{K_k} = \mathbf{t}_{K_1} = \mathbf{0}_{3\times 1}. \tag{45}$$

When these constraints are applied to (41), it simplifies to,

$$\mathbf{p}_j^{i,r} = \boldsymbol{\mathcal{R}}_{C_h}^\top \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{C_h} \mathbf{p}_j^{h,k} + \boldsymbol{\mathcal{R}}_{C_h}^\top \boldsymbol{\mathcal{R}}_{K_r} \mathbf{t}_{C_h} + \boldsymbol{\mathcal{R}}_{C_h}^\top \mathbf{t}_{K_r} - \boldsymbol{\mathcal{R}}_{C_h}^\top \mathbf{t}_{C_h} \tag{46}$$

$$= \boldsymbol{\mathcal{R}}_{C_h}^\top \left( \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{C_h} \mathbf{p}_j^{h,k} + \mathbf{t}_{K_r} + (\boldsymbol{\mathcal{R}}_{K_r} - \mathbf{I}_{3\times 3})\, \mathbf{t}_{C_h} \right). \tag{47}$$

For the analysis to follow, the position of the $j^{\text{th}}$ point feature, first observed in the $h^{\text{th}}$ camera at keyframe 1, is written as a unit vector bearing, $\hat{\mathbf{p}}_j^{h,1}$, and let the depth along this bearing to the point feature be $s_j \in \mathbb{R}^+$, such that the point feature position in the camera frame is,

$$\mathbf{p}_j^{h,1} = s_j \hat{\mathbf{p}}_j^{h,1}. \tag{48}$$

*4.2. System Degeneracies*

A typical least-squares optimization method, including the EKF, will seek to refine a parameter vector estimate, $\breve{\mathbf{x}}_i \in \mathbb{R}^n$, at the $i^{\text{th}}$ iteration, using a first-order Taylor-series expansion about the current estimated operating point,

$$\breve{\mathbf{x}}_{i+1} = \breve{\mathbf{x}}_i + \boldsymbol{\delta}_i, \tag{49}$$

where the parameter update, $\boldsymbol{\delta}_i \in \mathbb{R}^n$, is found by solving the system,

$$\mathbf{J}\boldsymbol{\delta}_i = \bar{\mathbf{z}}_i, \tag{50}$$

with the measurement error,

$$\bar{\mathbf{z}}_i = \mathbf{z} - \boldsymbol{\pi}(\breve{\mathbf{x}}_i), \tag{51}$$

and the associated measurement Jacobian matrix,

$$\mathbf{J} = \left.\frac{\partial \boldsymbol{\pi}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\check{\mathbf{x}}_i}. \tag{52}$$

Solving for $\boldsymbol{\delta}_i$ depends critically on the column rank of matrix $\mathbf{J}$. Therefore, when rank $(\mathbf{J}) < n$, one cannot find a unique solution for $\boldsymbol{\delta}_i$ given the measurement error and the system is considered degenerate. Accordingly, the rank of the Jacobian $\mathbf{J}$ is studied to identify the set of critical motions for the three-camera cluster when using an estimator based on this approach.

### *4.3. Scale Degenerate Motions*

In this section, the following assumptions will be made regarding the system:

1. The three cameras in the system are rigidly fixed in a configuration such that their centres are unique and not collinear.
2. Each of the three cameras observe a disjoint set of at least three non-collinear point features at the two keyframes. The point features are not on the camera x-y plane at either keyframe, and are not infinitely far from the camera.{I-2}
3. The five up-to-scale motion parameters describing relative orientation and translation direction of the individual cameras within the cluster are determined *a priori* from the single camera ego-motion problem, for which numerous techniques exist (e.g. [19, 28, 1, 23]).{I-2}

The purpose of Assumption 3 is to simplify the system such that the scale-degenerate motions can be identified in the subsequent analysis. The full set of degeneracies associated with the coupled system state (11) are difficult to determine due to the high dimensionality of the system. However, Assumption 3 allows for the identification of those motions which, despite the extra information from the up-to-scale motion estimate, are unable to recover the global scale of the cluster motion and target structure. Therefore, the critical motions identified in this section are a subset of those for the system in Section 3 but are sufficient conditions for scale degeneracy. {I-2}

Of interest are{II-1} the conditions when the measurements from the camera cluster allow for estimation of the final degree of freedom, corresponding to the translation magnitude and therefore, global system scale. The cluster translation is written as,

$$\mathbf{t}_{K_r} = s_t \hat{\mathbf{t}}_{K_r}, \tag{53}$$

where the translation unit vector $\hat{\mathbf{t}}_{K_r}$ is assumed known using Assumption 3, but the magnitude, $s_t$, is estimated and included in the state vector.

One of the point features observed in each of the three cameras is selected{I-3} and the resulting state vector $\mathbf{x} \in \mathbb{R}^4$, is written,

$$\mathbf{x} = \begin{bmatrix} s_1 & s_2 & s_3 & s_t \end{bmatrix}^\top, \tag{54}$$

where $s_1$, $s_2$, and $s_3$ are the depths to the selected point features in the three cameras, respectively, at the first keyframe.

In order to simplify the subsequent expressions, the positions of the feature points in the camera frames at the current keyframe, $K_r$, are written using a set of intermediate vectors,

$$\begin{bmatrix} x^{i,r} \\ y^{i,r} \\ z^{i,r} \end{bmatrix} = \boldsymbol{\mathcal{R}}_{C_i}^\top \left( s_i \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{C_i} \hat{\mathbf{p}}_i^{i,1} + s_t \hat{\mathbf{t}}_{K_r} + (\boldsymbol{\mathcal{R}}_{K_r} - \mathbf{I}) \mathbf{t}_{C_i} \right) \tag{55}$$

$$= \boldsymbol{\mathcal{R}}_{C_i}^\top \left( s_i \hat{\mathbf{e}}_i + s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i \right), \tag{56}$$

where the vectors $\hat{\mathbf{e}}_i = \boldsymbol{\mathcal{R}}_{K_r} \boldsymbol{\mathcal{R}}_{C_i} \hat{\mathbf{p}}_i^{i,1}$, $\hat{\mathbf{f}} = \hat{\mathbf{t}}_{K_r}$, $\mathbf{g}_i = \boldsymbol{\mathcal{R}}_{K_r} \mathbf{t}_{C_i}$, and $\mathbf{h}_i = -\mathbf{t}_{C_i}$. An example configuration of a three-camera cluster observing features over two keyframes is shown in Fig. 7 with the intermediate vector quantities labeled.

The measurement vector for this system, $\mathbf{z} \in \mathbb{R}^6$, consists of the image plane measurements for the point features from the three cameras, and has the following structure,

$$\begin{bmatrix} u^{1,r} \\ v^{1,r} \\ u^{2,r} \\ v^{2,r} \\ u^{3,r} \\ v^{3,r} \end{bmatrix} = \begin{bmatrix} \dfrac{x^{1,r}}{z^{1,r}} & \dfrac{y^{1,r}}{z^{1,r}} & \dfrac{x^{2,r}}{z^{2,r}} & \dfrac{y^{2,r}}{z^{2,r}} & \dfrac{x^{3,r}}{z^{3,r}} & \dfrac{y^{3,r}}{z^{3,r}} \end{bmatrix}^\top \tag{57}$$

which uses the positions of the three features in the coordinate frames of the respective cameras at the second keyframe, from (56).

The measurement Jacobian, $\mathbf{J}$, is composed of the partial derivatives of the measurement equations with respect to the system states. These partials are formed using the partial derivatives of the $i^{\text{th}}$ point feature position in

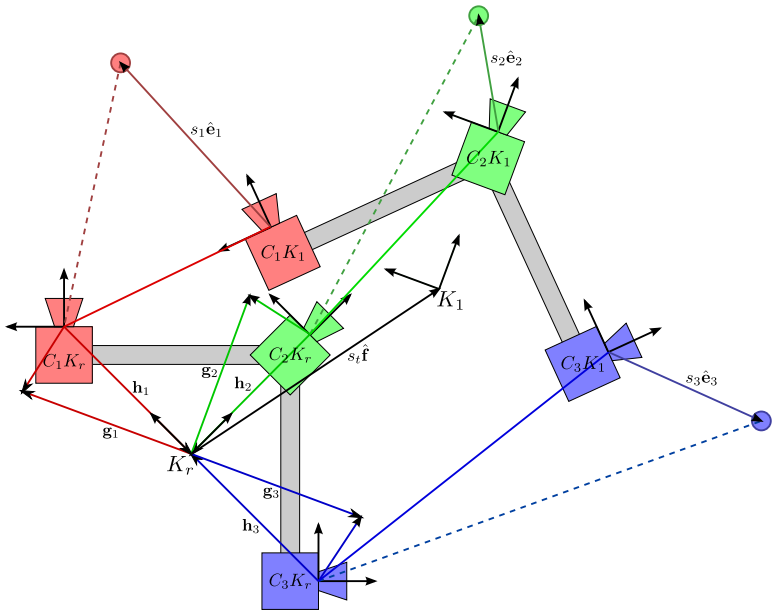


Figure 7: The three-camera cluster system is shown with the intermediate motion and structure quantities labeled.{I-3}

camera $i$ at keyframe $r$, with respect to the states. An individual observation Jacobian matrix has the structure,

$$\mathbf{J}^{i,r} = \begin{bmatrix} \dfrac{\partial u^{i,r}}{\partial \mathbf{x}} \\ \dfrac{\partial v^{i,r}}{\partial \mathbf{x}} \end{bmatrix} = \frac{1}{(z^{i,r})^2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \left[\mathbf{p}_i^{i,r}\right]_\times \frac{\partial \mathbf{p}_i^{i,r}}{\partial \mathbf{x}}, \tag{58}$$

where $z^{i,r}$ is the non-zero z-axis coordinate of the point in the observing keyframe, and the operator $[\cdot]_\times$ maps a vector in $\mathbb{R}^3$ to a skew-symmetric $3 \times 3$ matrix such that $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$, $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^3$.

The full measurement Jacobian is formed by stacking the three observation Jacobians,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}^{1,r} \\ \mathbf{J}^{2,r} \\ \mathbf{J}^{3,r} \end{bmatrix}, \tag{59}$$

and the partial derivatives of the point positions with respect to the system parameters are,

$$\frac{\partial \mathbf{p}_i^{i,r}}{\partial s_i} = \boldsymbol{\mathcal{R}}_{C_i}^{\top} \hat{\mathbf{e}}_i. \tag{60}$$

$$\frac{\partial \mathbf{p}_i^{i,r}}{\partial s_t} = \boldsymbol{\mathcal{R}}_{C_i}^{\top} \hat{\mathbf{f}} \tag{61}$$

Accordingly, the solution is degenerate when the Jacobian, which has the structure,

$$\mathbf{J} = \begin{bmatrix} j_{1,1} & 0 & 0 & j_{1,4} \\ j_{2,1} & 0 & 0 & j_{2,4} \\ 0 & j_{3,2} & 0 & j_{3,4} \\ 0 & j_{4,2} & 0 & j_{4,4} \\ 0 & 0 & j_{5,3} & j_{5,4} \\ 0 & 0 & j_{6,3} & j_{6,4} \end{bmatrix}, \tag{62}$$

where $j_{i,k}$ is element in the $i^{\text{th}}$ row and $k^{\text{th}}$ column, has less than full column rank,

$$\text{rank}\left(\mathbf{J}\right) < 4. \tag{63}$$

The first three columns of $\mathbf{J}$ describe the change in the point feature image location with respect to a change in the respective feature depth. A column will be all zeros only when the motion of the camera is collinear with the initial bearing to the point feature in the respective camera frame at the first keyframe,

$$\hat{\mathbf{e}}_i \times \left( s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i \right) = \mathbf{0}_{3\times 1}, \tag{64}$$

or when $\hat{\mathbf{e}}_i$ lies on the $i^{\text{th}}$ camera x-y plane at the first keyframe – which is impossible by Assumption 2. Also by Assumption 2, since there are at least two points in each camera which are not collinear with the camera centre at the first keyframe, there is at least one point which is not collinear with the position of the camera centre at the current keyframe, $\left( s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i \right)$, and including that point in the estimation ensures that each of the first three columns has at least one non-zero element.

If one of the elements in the $i^{\text{th}}$ column of $\mathbf{J}$, where $i = 1, 2, 3$, at rows $2i-1$ or $2i$ is zero, the matrix $\mathbf{O}_1$ swaps the two corresponding rows so that the non-zero element is in the $2i-1$ row. The matrix $\mathbf{O}_1$ is full rank and therefore, the matrix $\mathbf{K} = \mathbf{O}_1\mathbf{J}$ has the same rank as the matrix $\mathbf{J}$.

Next, the row operations matrix $\mathbf{O}_2$ is applied to rearrange the structure of the matrix $\mathbf{K}$ with,

$$\mathbf{O}_2 = \begin{bmatrix} o_{1,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & o_{1,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & o_{1,3} & 0 \\ o_{2,1} & k_{1,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & o_{2,2} & k_{3,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & o_{2,3} & k_{5,3} \end{bmatrix}, \tag{65}$$

where $k_{i,j}$ is the element of the matrix $\mathbf{K}$ in the $i^{\text{th}}$ row and $j^{\text{th}}$ column. If column $i = 1, 2, 3$ of the matrix $\mathbf{J}$ contained a zero in the $2i-1$ or $2i$ row, the corresponding elements in $\mathbf{O}_2$ are given by,

$$o_{1,i} = 1 \tag{66}$$

$$o_{2,i} = 0, \tag{67}$$

otherwise the elements from the matrix $\mathbf{K}$ are used,

$$o_{1,i} = k_{2i,i} \tag{68}$$

$$o_{2,i} = -k_{2i,i}. \tag{69}$$

By construction, the matrix $\mathbf{O}_2$ also has full rank and therefore,

$$\mathbf{M} = \mathbf{O}_2\mathbf{K} = \begin{bmatrix} m_{1,1} & 0 & 0 & m_{1,4} \\ 0 & m_{2,2} & 0 & m_{2,4} \\ 0 & 0 & m_{3,3} & m_{3,4} \\ 0 & 0 & 0 & m_{4,4} \\ 0 & 0 & 0 & m_{5,4} \\ 0 & 0 & 0 & m_{6,4} \end{bmatrix}. \tag{70}$$

has the same rank as $\mathbf{J}$. The diagonal elements $m_{1,1}$, $m_{2,2}$, and $m_{3,3}$ are all non-zero and it is apparent that the rank of $\mathbf{M}$, and by extension $\mathbf{J}$, is not

full rank and the system is degenerate, if and only if,

$$\mathbf{m}_4 = \begin{bmatrix} m_{4,4} \\ m_{5,4} \\ m_{6,4} \end{bmatrix} = \begin{bmatrix} \frac{1}{z^{1,r}} \left( \left( \hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_1 + \mathbf{h}_1) \right) \cdot \hat{\mathbf{e}}_1 \right) \\ \frac{1}{z^{2,r}} \left( \left( \hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_2 + \mathbf{h}_2) \right) \cdot \hat{\mathbf{e}}_2 \right) \\ \frac{1}{z^{3,r}} \left( \left( \hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_3 + \mathbf{h}_3) \right) \cdot \hat{\mathbf{e}}_3 \right) \end{bmatrix} = \mathbf{0}_{3\times 1}. \tag{71}$$

Each individual element in the vector $\mathbf{m}_4$ goes to zero in the following conditions:

**(i)** $\mathbf{g}_i + \mathbf{h}_i = \mathbf{0}_{3\times 1}$ **:** implies that $\boldsymbol{\mathcal{R}}_{K_r} = \mathbf{I}_{3\times 3}$ – there is no relative rotation in the motion, or that the camera centre is at the cluster frame origin. By Assumption 1, there can only be one camera with a centre at the cluster frame origin.

**(ii)** $\left( \hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i) \right) \cdot \hat{\mathbf{e}}_i = 0$ **:** the relative translation $\hat{\mathbf{f}}$, baseline change $(\mathbf{g}_i + \mathbf{h}_i)$, and point feature position $\hat{\mathbf{e}}_i$, are coplanar.

**(iii)** $\hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i) = \hat{\mathbf{f}} \times (\mathbf{g}_i + \mathbf{h}_i) = \mathbf{0}_{3\times 1}$ **:** the relative translation, $\hat{\mathbf{f}}$, is collinear with the vector to the camera centre at keyframe $K_r$.

Therefore, all of the elements of $\mathbf{m}_4$ are zero when one or more of the above cases are true for each of the point features in the system. When there is no relative rotation, the system is always degenerate. If there is some non-zero rotation, condition (i) can only be satisfied for at most one of the cameras with its centre at the cluster frame origin.

When the translation of a camera centre is coplanar with the cluster translation and the point feature position in the first keyframe, the corresponding $\mathbf{m}_4$ element is zero by condition (ii). This condition is unlikely to occur in a practical system as each camera will observe several point features at any keyframe such that there is a non-empty subset of features that are not coplanar with the translation and baseline change.

Condition (iii) is independent of the locations of the point features and depends solely on the motion of the camera cluster. For all of the elements of $\mathbf{m}_4$ to be zero, the cluster translation is collinear with the translations of the centres of all cameras,

$$\hat{\mathbf{f}} \times (s_t \hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i) = \mathbf{0}_{3\times 1}, \quad \forall i \in \{1, 2, 3\}. \tag{72}$$

This leads to the necessary condition for degeneracy that all of camera centres move in parallel,

$$(s_t\hat{\mathbf{f}} + \mathbf{g}_i + \mathbf{h}_i) \times (s_t\hat{\mathbf{f}} + \mathbf{g}_j + \mathbf{h}_j) = \mathbf{0}_{3\times 1}, \quad \forall i, j \in \{1, 2, 3\}. \tag{73}$$

In the two-camera cluster case, condition (72) leads to the concentric circle critical motions which cause the camera centres to move in parallel. For the non-collinear three-camera case studied here, the centres move in parallel in a much smaller set of motions. With non-zero rotation, it is only possible for the system to be in a critical motion when the axis of rotation of the keyframe orientation $\boldsymbol{\mathcal{R}}_{K_r}$, is completely in the plane defined by the three non-collinear camera centres. This plane is spanned by the vectors between the three cameras, $(\mathbf{h}_i - \mathbf{h}_k)$ and $(\mathbf{h}_j - \mathbf{h}_k)$ for $i, j, k \in \{1, 2, 3\}$, $i \neq j \neq k$, and since the camera centres are assumed to be non-collinear, the normal $(\mathbf{h}_i - \mathbf{h}_k) \times (\mathbf{h}_j - \mathbf{h}_k) \neq \mathbf{0}_{3\times 1}$.

The relative rotation can be parameterized as a rotation angle, $\theta = \|\mathbf{a}\|$ about an axis, $\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$. The axis can be written with components tangential and orthogonal to the camera centre plane,

$$\mathbf{a} = c_1(\mathbf{h}_i - \mathbf{h}_k) + c_2(\mathbf{h}_j - \mathbf{h}_k) + c_3(\mathbf{h}_i - \mathbf{h}_k) \times (\mathbf{h}_j - \mathbf{h}_k). \tag{74}$$

If there is any component of the keyframe rotation axis that is perpendicular to the camera centre plane, $c_3 \neq 0$, the system is not degenerate. The relative rotation can be written using the Rodrigues rotation formula [19],

$$\boldsymbol{\mathcal{R}}_{K_r} = \mathbf{I}_{3\times 3} \cos\theta + \sin\theta \, [\hat{\mathbf{a}}]_\times + (1 - \cos\theta)\hat{\mathbf{a}}\hat{\mathbf{a}}^\top. \tag{75}$$

For the system to be degenerate by (72), the translation vector $\hat{\mathbf{f}}$ must be collinear with $(\mathbf{g}_m + \mathbf{h}_m)$ for $m \in \{1, 2, 3\}$, requiring $(\mathbf{g}_m + \mathbf{h}_m) \times (\mathbf{g}_n + \mathbf{h}_n) = \mathbf{0}_{3\times 1}$ for $m, n \in \{1, 2, 3\}$. However,

$$(\mathbf{g}_m + \mathbf{h}_m) \times (\mathbf{g}_n + \mathbf{h}_n) = ((\boldsymbol{\mathcal{R}}_{K_r} - \mathbf{I}_{3\times 3})\, \mathbf{h}_m) \times ((\boldsymbol{\mathcal{R}}_{K_r} - \mathbf{I}_{3\times 3})\, \mathbf{h}_n) \tag{76}$$

$$= 2\frac{(1 - \cos\theta)}{\theta} \left((\mathbf{h}_m \times \mathbf{h}_n) \cdot \mathbf{a}\right) \hat{\mathbf{a}}, \tag{77}$$

which is zero only when the triple scalar product $(\mathbf{h}_m \times \mathbf{h}_n) \cdot \mathbf{a} = 0$ for $m, n \in \{1, 2, 3\}$. When the axis of rotation has a non-zero perpendicular component, the triple scalar product resolves to a non-zero value,

$$c_3 \, \|(\mathbf{h}_i - \mathbf{h}_k) \times (\mathbf{h}_j - \mathbf{h}_k)\|^2 \neq 0. \tag{78}$$

Therefore, for the system to be degenerate, the rotation axis for the relative keyframe orientation must be completely in the camera centre plane, and the resulting motion must cause all three camera centres to move in parallel. This is a much smaller set of critical motions than with the two-camera cluster. It still requires rotational motion to resolve the scale, but any component of the rotational axis that is perpendicular to the camera centre plane prevents the system from becoming degenerate. For example, an automobile with a three-camera cluster mounted with the centre plane parallel to the road would avoid the critical motions during a turning maneuver since the rotation axis is always perpendicular to the camera centre plane.

## 5. Experimental Results

The objectives of the experiments presented in this section are as follows: to show that an estimator using the parameterization detailed in this work is able to recover the relative pose of an initially unknown target accurately even without spatial overlap between the FOV of the cluster cameras; to demonstrate the flexibility of the parameterization for handling all varieties of cluster configurations; and to demonstrate the benefits of using the third camera against the two-camera cluster with some FOV overlap as in [6, 17]. It will be shown that the third camera allows the non-overlapping three-camera cluster to overcome the critical motion degeneracies of the two-camera cluster that necessitated the intersection of the FOV. By lifting that constraint and using a larger collective FOV, the accuracy of the pose estimates is improved.

### *5.1. Experiment Setup*

A camera cluster was constructed, as shown in Fig. 1, as well as a target object consisting of easily detectable point features. Black dots on a white background are used in order to simplify the feature extraction module, which can be considered the front-end to any relative pose estimation system. Even though the artificial black dot features are more reliably detected than more general image features, their 3D locations are still initially completely unknown to the estimator. The design of the front-end feature extraction algorithm is not the focus of this work, so by eliminating the variability introduced by the feature extraction and correspondence phase, these experiments confirm the performance of the estimator and allow for a more direct comparison of cluster configurations.

An optical motion capture system was used to provide accurate ground truth data with which to evaluate the estimates from the algorithm using the various camera configurations. The NaturalPoint OptiTrack system serves a dual purpose in these tests: to facilitate the mutual extrinsic calibration of the individual cluster cameras; and to measure the camera cluster motion to provide accurate ground truth data for the comparison.

The camera cluster, along with the rest of the experimental apparatus, is shown in Fig. 8. The rig is composed of four internally calibrated NaturalPoint V100:R2 USB cameras, as well as a set of infrared (IR) reflecting markers which are tracked with the OptiTrack system. The positioning system is listed as having sub-millimetre accuracy [29].

Figure 8: The camera cluster constructed for these experiments is mounted on a hand-held rig and augmented with IR reflecting markers for ground truth data collection using the motion capture system (shown on tripods in the background). The target object consists of the black dots on the white background.

To relate the ground truth motion measurements with the estimates from the cluster algorithm, the pose of each cluster camera with respect to the IR markers must be known. This extrinsic calibration was performed for each camera with respect to the IR markers on the rig via an eye-in-hand calibration [30], which also provides the relative poses of each of the cameras in the cluster with respect to each other.

Each camera captures greyscale images of 640 $\times$ 480 pixels at 22 Hz,

and the shutters are automatically synchronized not only with each other, but also with the motion capture measurements from the OptiTrack system. The four cameras on the rig are separate from the five used in the actual OptiTrack system.

The cameras within the cluster are arranged to allow for comparisons between common configurations – two forward-facing, one rear-facing, and one looking to the side. The two forward-facing cameras have some overlap in their FOV as in Kim *et al.* [6] and Sola *et al.* [17], and have a baseline length of approximately 10 cm. In these experiments, three different cluster configurations were used under the same relative motion profile: a divergent stereo cluster with some FOV overlap and inter-camera point correspondence when available; a forward-backward-sideways three-camera cluster with non-overlapping FOV as studied in the analysis in Section 4; and a cluster consisting of all four cameras. The estimates from each camera set were then compared to the ground-truth to determine the performance in terms of pose estimate accuracy.

For the test run, the camera rig was moved by hand around the workspace and the image frames from each camera were collected, along with the motion capture ground truth data for each time step. The target object point features were between 1 and 2 m away from the the camera cluster, simulating a close maneuvering operation. Although only the camera cluster is moved in these tests, the algorithm is estimating relative motion of the cluster and target environment. Since only camera image measurements are used in the estimation, the rigid target environment is also free to move in the world frame. However, the environment is kept static in these tests so that the comparison to the ground truth cluster motion data is valid. The ground-truth relative motion profile used for these tests is shown in Fig. 9.

For these tests, an iterated EKF (IEKF) [31] was used to estimate the system state at each time step. The estimation algorithm was implemented in MATLAB and run sequentially on the image frames collected by the camera cluster using the three configurations specified. Videos of the estimator in operation can be viewed at http://wavelab.uwaterloo.ca/?q=multicamera.

The experiments were not implemented in real-time, but instead serve to show the accuracy gains of including additional cameras in the cluster for solving the local SLAM problem. The computational requirements for this method are on the order of similar recursive filter SLAM algorithms (e.g. [1]) which have been demonstrated to operate in real-time with several hundred point features included in the state vector, but are known

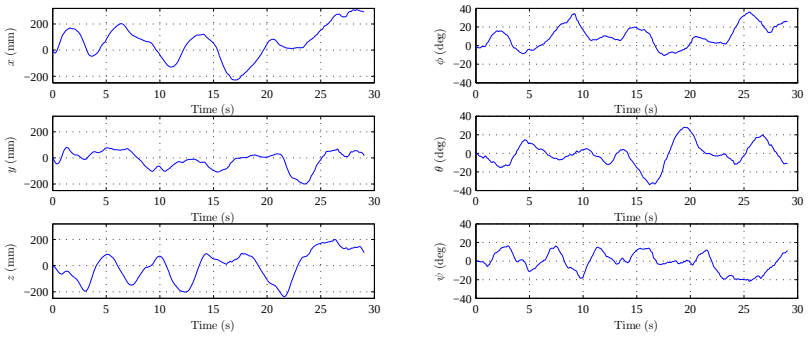


Figure 9: The ground-truth trajectories of the six relative pose parameters for the camera cluster as measured by the motion capture system. The orientation is shown in Euler angles for convenience.

to grow approximately cubically with the number of features included in the model. While the relatively small environment and time span used in these experiments are sufficient to demonstrate the improved accuracy offered by the camera cluster configurations, for large-scale exploration and model-building operations, the Kalman-type recursive filters are less appropriate. However, there exist several filter adaptations and nonlinear optimization Bundle Adjustment methods to which this formulation can be readily applied [32, 33, 34, 35, 36]{I-4, I-14, II-2}.

### *5.2. Three-Camera Cluster Results*

The estimation algorithm was run using the three-camera cluster to estimate the relative pose and target model parameters. The absolute values of the estimation errors for the six pose parameters are shown in Fig. 10.

It is evident that despite the non-overlapping FOV, the reduction in the set of critical motions by using the three-camera cluster allows the estimator to readily find the scale of the motion and environment. The resulting position estimates are accurate to less than one percent of the ground-truth distance from the initial position, once the global scale converges. The scale metric, shown in Fig. 11, is calculated as the ratio of the norms of estimated to actual position, and expressed as a percentage.

Moreover, the wide collective FOV provides improved localization constraints resulting in improved overall accuracy of the estimation. These results confirm the conclusions in Section 4 that this framework is able to

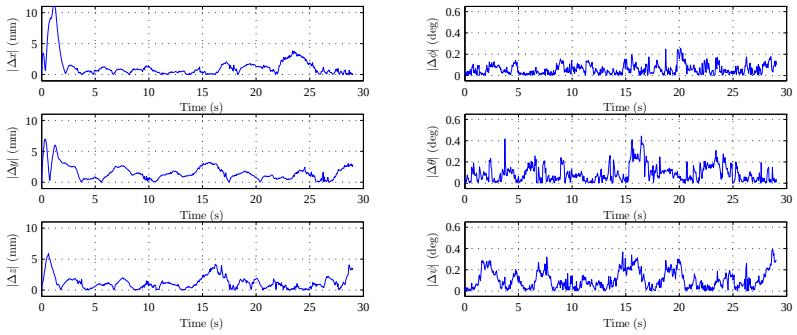


Figure 10: Magnitudes of estimation errors in the six relative pose parameters as estimated using the three-camera cluster. The rotation parameters are converted to Euler angles and expressed in degrees for readability.

accurately localize the cluster and model the target object, even in the absence of FOV overlap or inter-camera correspondence by adding the third camera to the cluster.

### *5.3. Comparison to Other Configurations*

To see how the three-camera results compare to other common camera configurations, two more tests were run using the same relative motion profile: divergent stereo cameras with FOV overlap; and all four cameras together. The framework presented in this work is able to easily accommodate each of these configurations.

When the divergent stereo configuration is used, the stereo correspondence is solved in a passive manner by matching the image measurements to the existing point features in the target model instead of explicitly matching points between the two camera images and triangulating to find the depth. This alleviates the requirement that the motion of the two-camera cluster be non-critical, since the system scale is resolved by the implicit triangulation of the depth by the estimator using the corresponding point features seen in both cameras at the same time step. However, as discussed previously, this configuration limits the collective FOV of the system and results in poor localization estimates, similar to monocular and binocular camera systems [4].

The resulting magnitude of estimated position errors for the different camera configurations are shown in Fig. 12. The three plots show that the large collective FOV on the three-camera cluster (3-cluster) and the four-camera

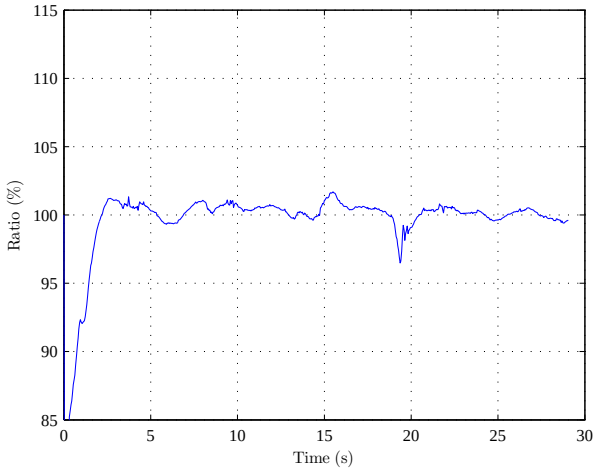


Figure 11: The scaling between the estimated and actual motions using the three-camera cluster, calculated as the ratio of the estimated position norm to the actual position norm as measured by the motion capture system and expressed in percent. The downward spike just prior to 20 seconds is a numerical artifact in the scale ratio calculation caused by division by the actual position magnitude, which is very close to zero at that point.

Table 1: RMSE For Cluster Configurations

| Configuration | RMSE (mm) |
|---|---|
| stereo | 11.8 |
| 3-cluster | 3.3 |
| 4-cluster | 2.5 |

cluster (4-cluster) improve the accuracy of the estimates well beyond the results of the divergent-stereo configuration (stereo), which has a narrow FOV. Additionally, the 4-cluster is able to converge to the proper global scale faster than the 3-cluster because of the immediate feature correspondence between the front-facing cameras. The 3-cluster must wait for sufficient motion in order to have the global scale converge to the proper value. However, once this occurs, the two wide-FOV cluster configurations show similar performance. Table I summarizes the root-mean-squared error (RMSE) of the errors in translation magnitude of the three cluster configurations.

These results confirm that the three-camera cluster outperforms the divergent stereo case as predicted in the critical motion analysis. The large FOV of the three-camera cluster lends itself to stability in the estimates as more

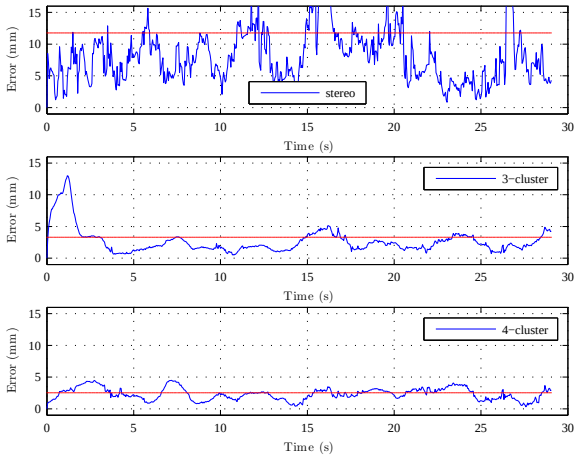


Figure 12: Magnitude of translation error for the estimator using the divergent stereo camera pair (top), 3-cluster (middle), and 4-cluster (bottom) configurations through the relative motion trajectory. The RMSE for each case is drawn as the red constant line.

features are visible at any point in the relative motion, and the decreased set of critical motions allows the the scale metric to be readily determined without the need for feature correspondence between cameras.

## 6. Conclusions

A framework for calibrated multicamera clusters is presented which accurately estimates both relative motion and model structure of the cluster and an unknown rigid target object or environment. An estimator based on this parameterization is able to maintain an online estimate of the relative pose when both the camera cluster and the target are in free motion. The framework is also easily extendable when other measurements, such as GPS, IMUs, or robot motion dynamics are available and the target is stationary in the inertial frame.

Any configuration of perspective cameras may be used, including arrangements with spatially non-overlapping FOV, and the estimator is able to resolve the global motion and model scale as soon as the information is available in the image measurements, assuming the relative motion is not strictly critical or degenerate, and the ratio of camera baseline distance to point feature depth is sufficiently large. A novel analysis of the degenerate configurations

was performed to identify the set of critical motions present when using a camera cluster consisting of three cameras arranged with non-overlapping FOV. The set of critical motions, compared with that of the two-camera cluster is significantly reduced with the addition of the third camera, enabling scale-recovery in more generic relative motion profiles.

Finally, experimental results demonstrate the accuracy of the estimates using a variety of camera cluster configurations, against a set of high-precision motion tracking system measurements. The degeneracy analysis conclusions are confirmed for the three-camera clusters with the large collective FOV, showing reliable stability and accuracy in the estimates, including the correct scale in general motion profiles. It is shown that because the critical motion set is reduced, it is not necessary to maintain overlap in the FOV, and the non-overlapping three-camera cluster provides more accurate motion estimates compared to the divergent stereo configuration, and is a better choice for high-precision robotic applications.

## Acknowledgements

This work was partially funded by the National Sciences and Engineering Research Council of Canada (NSERC) under Grant No. CRDPJ 397768-10. Partial funding also comes from the NSERC through the Alexander Graham Bell Canada Graduate Scholarship - Doctoral (CGS-D) award.

## References

[1] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052–1067.

[2] D. A. Forsyth, J. Ponce, Computer vision: a modern approach, Prentice-Hall, 2003.

[3] P. Baker, C. Fermuller, Y. Aloimonos, R. Pless, A spherical eye from multiple cameras (makes better models of the world), in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2001, pp. 576–583.

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

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

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

[7] R. Pless, Using many cameras as one, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2003, pp. II–587–93.

[8] J. S. Kim, M. Hwangbo, T. Kanade, Spherical approximation for multiple cameras in motion estimation: Its applicability and advantages, Computer Vision and Image Understanding 114 (10) (2010) 1068–1083.

[9] J. H. Kim, R. Hartley, J. M. Frahm, M. Pollefeys, Visual odometry for non-overlapping views using second-order cone programming, in: Proceedings of the 8th Asian Conference on Computer Vision, Vol. 2, 2007, pp. 353–362.

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

[11] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, R. Siegwart, 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), 2012, pp. 1529–1536.

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

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

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

[15] M. Kaess, F. Dellaert, Visual SLAM with a multi-camera rig, Tech. Rep. GIT-GVU-06-06.

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

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

[18] Y. Lu, J. Z. Zhang, Q. M. J. Wu, Z. N. Li, A survey of motion-parallax-based 3-D reconstruction algorithms, IEEE Transactions on Systems, Man, and Cybernetics 34 (4) (2004) 532–548.

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

[20] M. W. Spong, M. Vidyasagar, Robot dynamics and control, Wiley, 1989.

[21] D. G. Lowe, Object recognition from local scale-invariant features, Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2 (1999) 1150–1157.

[22] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding 110 (3) (2008) 346–359.

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

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

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

[26] D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti, Monocular SLAM with inverse scaling parametrization, in: Proceedings of the British Machine Vision Conference (BMVC), 2008, pp. 945–954.

[27] A. Gelb, Applied optimal estimation, The MIT Press, 1999.

[28] E. Eade, T. Drummond, Scalable monocular SLAM, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 469–476.

[29] NaturalPoint, Inc., V100:R2 data sheet, accessed: 2013-08-26.
URL `http://www.naturalpoint.com/optitrack/static/documents/V100-R2 Data Sheet.pdf`

[30] K. H. Strobl, G. Hirzinger, Optimal hand-eye calibration, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2006, pp. 4647–4653.

[31] D. Simon, Optimal state estimation: Kalman, $H_\infty$ and nonlinear approaches, John Wiley and Sons, 2006.

[32] M. R. Walter, R. M. Eustice, J. J. Leonard, Exactly sparse extended information filters for feature-based SLAM, The International Journal of Robotics Research 26 (4) (2007) 335–359.

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

[34] S. Agarwal, N. Snavely, S. M. Seitz, R. Szeliski, Bundle adjustment in the large, in: Proceedings of the European Conference on Computer Vision (ECCV), Vol. 2, 2010, pp. 29–42.

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

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

## Vitae

*Michael J. Tribou*

Michael J. Tribou received the B.A.Sc. degree from Queen's University, Kingston, ON, Canada, in 2007, and the M.A.Sc. and Ph.D. degrees from the University of Waterloo, Waterloo, ON, Canada, in 2009 and 2014, respectively.

In 2010, he joined the Waterloo Autonomous Vehicles Lab (WAVELab) and is currently working as a Postdoctoral Fellow in the Department of Mechanical and Mechatronics Engineering at the University of Waterloo in Waterloo, ON, Canada. His research interests include vision-based pose estimation and control of autonomous vehicles.

*Steven L. Waslander*

Steven L. Waslander received his Ph.D. from Stanford University in 2007, his M.S. from Stanford University in 2002, both in Aeronautics and Astronautics, and his B.Sc.E. in 1998 from Queen's University.

In 2008, he joined the Department of Mechanical and Mechatronics Engineering at the University of Waterloo in Waterloo, ON, Canada, as an Assistant Professor. He is the Director of the Waterloo Autonomous Vehicles Laboratory (WAVELab). His research interests are in the areas of multi-agent control and coordination, air traffic management and autonomous mobile robotics, specifically quadrotor helicopters.

*David W. L. Wang*

David W. L. Wang received the B.E. degree from the University of Saskatchewan, Saskatoon, SK, Canada, in 1984, and the M.A.Sc. and Ph.D. degrees from the University of Waterloo, Waterloo, ON, Canada, in 1986 and 1989, respectively.

In 1990, he joined the Department of Electrical and Computer Engineering, University of Waterloo, and was promoted to Full Professor in 1999. He was Associate Chair for Graduate Studies from 1998 to 2000. His research interests include nonlinear control, flexible manipulators/structures, shape memory alloy actuators, and haptic interfaces.