Robotics **2014**, *3*, 438-438; doi:10.3390/robotics3040417

OPEN ACCESS

robotics

ISSN 2218-6581

www.mdpi.com/journal/robotics

Article

Vision-Based Cooperative Pose Estimation for Localization in Multi-Robot Systems Equipped with RGB-D Cameras

Xiaoqin Wang *, Y. Ahmet Şekercioğlu and Tom Drummond

Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, 3800, Australia; E-Mails: ahmet.sekercioglu@monash.edu (Y.A.S.); tom.drummond@monash.edu (T.D.)

* Author to whom correspondence should be addressed; E-Mail: xiaoqin.wang@monash.edu.

External Editor: Wenjie Dong

Received: 3 December 2014; in revised form: 15 December 2014 / Accepted: 15 December 2014 / Published: xx

---

Abstract: We present a new vision based cooperative pose estimation scheme for systems of mobile robots equipped with RGB-D cameras. We first model a multi-robot system as an edge-weighted graph. Then, based on this model, and by using the real-time color and depth data, the robots with shared field-of-views estimate their relative poses in pairwise. The system does not need the existence of a single common view shared by all robots, and it works in 3D scenes without any specific calibration pattern or landmark. The proposed scheme distributes working loads evenly in the system, hence it is scalable and the computing power of the participating robots is efficiently used. The performance and robustness were analyzed both on synthetic and experimental data in different environments over a range of system configurations with varying number of robots and poses.

Keywords: pose estimation; RGB-D camera; self-calibration; cooperative localization; multi-robot coordination

---

1. Introduction

Multi-Robot Systems (MRSs), which were first proposed in early 1980s, are becoming increasingly popular mobile sensor platforms to measure and estimate quantities of interest at spatially distributed locations. Compared to a single-robot operation, MRSs have the advantages on faster task completion, more extensive coverage, increased reliability to sensor failures, and higher estimation

accuracy through sensor fusion. MRSs have been widely used in a variety of tasks such as data collection [1], surveillance [2–4], target tracking [5–7], formation tracking and control [8–10], and visual SLAM [11–13].

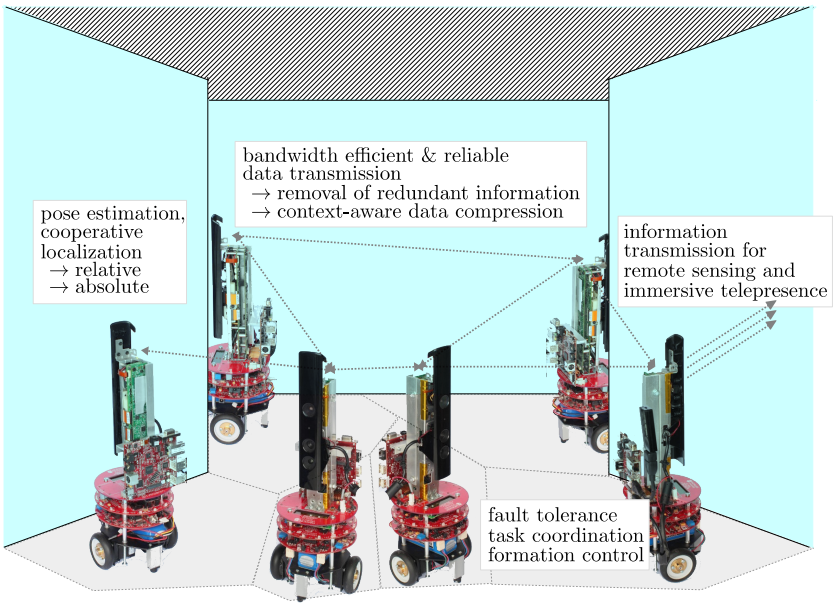
Precise knowledge on locations and orientations (poses) of the robots is a prerequisite for the successful accomplishment of the collaborative tasks. One approach to localize the robots is to equip them with Global Positioning System (GPS) receivers and use the GPS system. However, GPS signals are not available indoors and they cannot directly provide the orientation information. An alternative approach is cooperative localization in which robots work together and use the robot-to-robot measurements to construct a map of their network. Cooperative localization operates in two broad stages: The *initialization process*, which involves *relative pose estimation* (RPE), provides initial location and orientation estimates of the robots. Then the *refinement process* updates the initial estimates iteratively to enhance the accuracy.

In this study, we focus on the initialization process of the cooperative localization in MRSs. The MRSs of interest here are equipped with cameras or visual sensors. In this kind of MRSs, robots' locations and orientations can be estimated through careful calibration of the rigid body transform operations between the cameras mounted on the robots. Generally there are two kinds of vision-based approaches to achieve this goal: (1) manual calibration; or (2) self-calibration. Manual calibration approaches require a special calibration pattern to be visible in all images [14] or the precise pose information of calibration patterns/objects have to be known [15]. Some easily detectable single features which require human interaction, such as moving a LED in a dark room, can also be used to manually calibrate multiple cameras [16–18]. Manual calibration methods, even though provide good results, require special equipment or time consuming manual measurements. Self-calibration algorithms simultaneously process several images captured by different cameras and find the correspondences across images. Correspondences are established through extracting 2D features from images automatically and matching them between different images. Then, based on the established correspondences, cameras' relative poses can be estimated from the essential matrix. Besides the algorithm which uses static features [19–22], Aslan *et al.* [23] detect people walking in the room and use the point on the top of each person's head as the calibration feature. The accuracy of the self-calibration highly depends on the reliability of the relative pose estimates. This problem was first discussed in [24] with the concept of the vision graph. Kurillo *et al.* [25], Cheng *et al.* [26], and Vergés-Llahí *et al.* [27] later used and refined it for this purpose as well. It is becoming a useful general tool for describing the directionality of networked visual sensors. It has been more recently addressed by Bajramovic *et al.* [28–30]. They proposed a graph based calibration method which measures the uncertainty of the relative pose estimation between each camera pair. All of the self-calibration algorithms measure the epipolar structure of the system and suffer from scale ambiguity. If there is not any object or pattern with known geometry in the scene, the orientations and locations between robots are determined up to a scale.

In this paper, we consider the initialization process for localization in a multi-robot system with $N \geq 3$ robots operating in GPS-denied indoor environments (see Figure 1). A RGB-D camera, which provides both color images and per-pixel depth information, is mounted at the top of each robot. A central node with high performance processor is also implemented in the system which can operate computationally expensive computer vision algorithms. We present a novel self-calibration algorithm to determine the

locations and orientations of the robots in this RGB-D camera equipped multi-robot system. The propose scheme can be arranged over an indoor scenarios without imposed constraints for all robots to share a common field-of-view (FoV). Our approaches assume that at least any two given robots have overlapping FoVs and that the cameras on robots have been internally calibrated prior to deployment. Our proposed algorithms consist of the following steps: (1) each robot extracts color features locally and sends the descriptors of these features to the central node; (2) the central node performs feature matching to determine neighboring robots and generates an Initial Pose Matrix (IPM); (3) the central node constructs a robot dependency graph and selects a number of relative poses to connect robots as a calibration tree; (4) after the central node broadcasts the information of the calibration tree, robots work collaboratively to determine the relative poses according to the calibration tree; (5) the determined relative poses are then transmitted to the central node to compute the poses of all the robots in the system. We formulate the selection of relative poses as a shortest path problem, which consists of finding shortest path from a vertex to the other vertices in an edge-weighted graph. The graph represents FoVs of robots as vertices and overlapping FoVs as edges respectively.

Figure 1. An indoor mapping and exploration scenario showing the Monash University's RGB-D equipped experimental mobile robots "eyeBugs". A typical application would be mapping indoors after a disaster such as Fukushima nuclear reactor accident. As shown in the diagram, there are numerous challenges that need to be addressed. In this paper, our focus is initialization problem in cooperative localization.



The main contributions of this paper are:

- Construction of a robot dependency graph based on the overlapping ratio between neighboring robots.
- Development of a procedure to determine the relative pose of multiple RGB-D camera equipped robots.
- By contrast to the conventional approaches that only utilize color information, our approach takes the advantages of the combination of RGB and depth information.

- The locations and orientations of robots are determined up to the real world scale directly without involving scale ambiguity problem.
- Extensive experiments using synthetic and real world data were conducted to evaluate the performance of our algorithms in various environments.

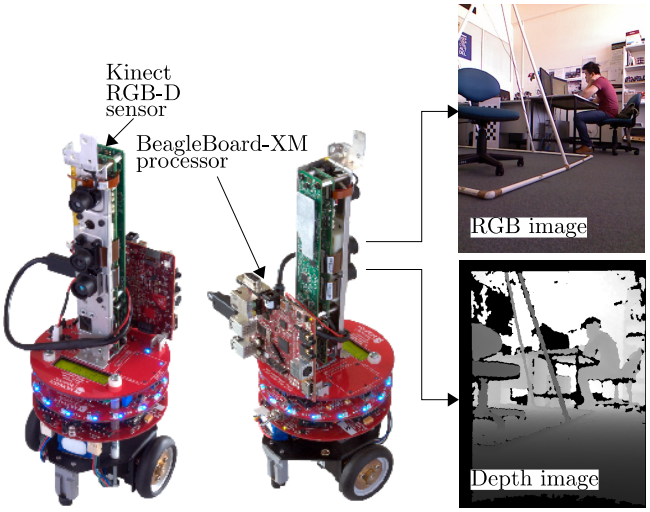
The rest of the paper is organized as follows. In Section II, the characteristics of the RGB-D camera and the multi-robot system used in this paper are introduced. In Section III, we formulate the robot localization problem and propose our solutions. In Section IV, we present the experiments and results, and our concluding remarks can be found in Section V.

2. A Multi-Robot System Using RGB-D Cameras As Visual Sensors

2.1. eyeBug: A Robot Equipped with RGB-D Camera

At the Wireless Sensor and Robot Networks Laboratory (WSRNLab) [31] we have created a multi-robot system consisting of experimental mobile robots called eyeBugs (Figure 2) for computer vision, formation control, visual sensing research activities. A single-board computer, BeagleBoard-xM [32] is the main computational resource of an eyeBug. Each BeagleBoard-xM has an ARM37x 1 GHz processor, a USB hub, and an HDMI video output. Communication service between robots is provided through WiFi links. We run an ARM processor optimized GNU/Linux operating system (Ubuntu Version 11.10) [33]. OpenKinect [34], OpenCV [35] and libCVD [36] libraries were installed to capture and process image information. Microsoft Kinect, which produces color and disparity-based depth images, is mounted vertically at the center of the top plate of each robot. The default RGB video stream provided by Kinect uses 8 bits for each color at VGA resolution (640 $\times$ 480 pixels, 24 bits/pixel). The depth video stream is also in VGA resolution.

Figure 2. eyeBug-the robot developed for the Monash WSRNLab's [31] experimental multi-robot platform. The RGB-D data generated by the Kinect RGB-D sensor is processed on BeagleBoard-xM running GNU/Linux operating system.



2.2. Characteristics of RGB-D Camera

Kinect has an infrared (IR) projector-camera-pair and a RGB camera. The depth sensing of Kinect is based on a fixed structured light source positioned at a known baseline from the IR camera. The depth information is measured through a triangulation process which is based on the detection of transverse shifts of local dot patterns in the IR speckle with respect to its reference patterns at a known distance to the device [37]. This process runs repeatedly on all regions in the IR speckle and generates a disparity-based depth image. It should be noted that the depth signal inevitably degrades when multiple RGB-D cameras are pointing at the same scene. It is because the camera projects a structured light dot pattern onto the scene continuously without modulation and devices interfere with one another [38]. This interference can be eliminated by a "Shake 'n' Sense" approach [38].

The normalized disparity values returned by the Kinect are inversely proportional to their depth [39]. Furthermore, [40,41] show that there is a non-linear relationship between the normalized disparity values and their depth value in Euclidean space. Therefore, it is more suitable to represent the data in inverse depth coordinates. Consider the vector $\mathbf{p}_e = [x\ y\ z\ 1]^T$ which represents a real world point in Euclidean space by using homogeneous coordinates. The relation between a real world point in inverse depth coordinates and its corresponding pixel in the depth image can be established as follows,

$$\mathbf{p}_e \equiv \frac{1}{z}\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \equiv \begin{bmatrix} u & v & 1 & q \end{bmatrix}^T \equiv \begin{bmatrix} \frac{i-i_c}{f_x} & \frac{j-j_c}{f_y} & 1 & \frac{1}{z} \end{bmatrix}^T \tag{1}$$

where (i, j) denotes the pixel coordinates of this real world point projection in the depth image, and z is the corresponding depth value returned by the camera.

3. Self-Calibration Cooperative Pose Estimation

3.1. Overview

Given $N(N \geq 3)$ robots equipped with intrinsically calibrated RGB-D cameras, the goal is to automatically determine the initial pose of each robot in a common coordinate system using only the color and depth data. A central node with a high performance processor is also included in the system which runs the computationally expensive algorithms. The function of this node is explained in Section 3.3.

When two robots a and b have sufficiently overlapping FoVs, the relative pose between two robots can be represented by a transformation matrix, $\mathbf{M}_{ab}$, in SE(3) as follow,

$$\mathbf{M}_{ab} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. $\mathbf{M}_{ab}$ denotes relative pose of robot b with respect to robot a and is the rigid transformation from the coordinate system of robot b to that of

robot a. If there is a robot c and the relative pose between robots c and b is $\mathbf{M}_{bc}$, then the relative pose between robots a and c can be derived via composition as,

$$\mathbf{M}_{ac} = \mathbf{M}_{bc}\mathbf{M}_{ab} \tag{3}$$

This transformation provides a mapping from the coordinate system of c to that of b, then from that of b to that of a. Robot b is the *intermediate node* in this process. This operation is transitive, therefore one robot's pose relative to another can be determined indirectly over an arbitrary number of intermediate poses if they exist.

Thus, the system's topology can be built up from the pairwise relative poses between robots that have common FoVs. In order to achieve this, we first need to determine the robots with sufficiently overlapping FoVs. Secondly, robots are grouped in pairs to determine rough estimations of the relative poses, and a number of relative poses are selected based on the reliability of the pose information. In the final step, we calibrate the overall system based on the selected pairwise relative poses. A general description of the scheme we propose is shown in Figure 3. Each step is described in details in the following sections. The list of symbols used through the paper is given in Table 1.

Figure 3. Operational overview of the proposed self-calibration scheme for cooperative pose estimation.

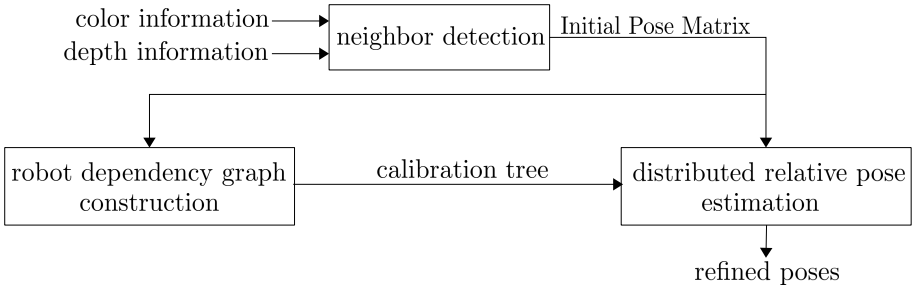


Table 1. Mathematical Notation.

Notation	Description
$\mathbf{Z}_a$	Depth image captured by robot a.
$\mathbf{p}_e$	Vector representing a real world point in Euclidean space.
$(i_{c,a}, j_{c,a})$	Principal point coordinates of the pinhole camera model.
$(f_{x,a}, f_{y,a})$	Focal length of the camera in horizontal and vertical axes.
$\mathbf{M}_{ab}$	Transformation matrix describing the relative pose between robots a and b.
$\mathbf{p}_a^l$	Sampled points on the depth image captured by robot a.
$\mathbf{p}_b^{l*}$	Corresponding points of $\mathbf{p}_a^l$ on the depth image captured by robot b.
N_a	Number of sampled points on $\mathbf{Z}_a$.
P_a	Set of sample points on $\mathbf{Z}_a$.
P_b^*	Set of corresponding points of P_a on $\mathbf{Z}_a$.
$\vec{n}_{l*,b}$	Surface normal at point $\mathbf{p}_b^{l*}$.
$w_{l,a}$	Weight parameter for correspondence established between $\mathbf{p}_a^l$ and $\mathbf{p}_b^{l*}$.
$\mathbf{E}$	Update transformation matrix in each iteration.
α_j	An element of a 6D motion vector.
$\mathbf{G}_j$	6D motion generator matrices.

3.2. Assumptions

We make the following assumptions about the multi-robot system:

- Intrinsic parameters of the RGB-D camera on each robot are calibrated prior to deployment,
- At least two robots in the system have overlapping FoVs;
- The scene is static and the robots do not move during the localization process, and
- The robots can form an ad-hoc network and directly communicate with each other.

3.3. Neighbor Detection and Initial Relative Pose Estimation

We define robots with overlapping FoVs as neighbors. One robot's neighbors can be detected through searching for image pairs sharing a common FoVs. This search can be viewed as a matching of point correspondences that considers the local environment of each feature point set. There are three steps in neighbor detection process: feature detection, feature description, and feature matching. The first two steps are performed on each robot locally. Taking processing speed and accuracy into consideration, we implement FAST [42,43] for feature detection and ORG [44] for feature description on each robot. Instead of transmitting the complete images, each robot sends the feature descriptions to the central node to minimize the transmission load. The corresponding depth information of each feature is also transmitted in conjunction with the feature descriptors.

Associating the feature descriptors with their corresponding depth values, central node can generate feature points in 3D. Central node performs feature matching between every two sets of the feature descriptors. In order to increase the matching reliability and robustness against outliers, we adopt both the symmetrical matching scheme and the geometric constraint to refine the matched points. In the symmetrical matching scheme, the correspondences between two sets of feature descriptors are established bidirectionally. One group of correspondences is generated from matching the first feature set to the second feature set. The other group is produced from matching the second feature set to the first feature set. For a pair of matched features to be accepted, two features must be the best matching candidate of each other in both directions.

Then, we use RANSAC to find a coarse registration, $\mathbf{M}_{ij}^{*}$, between every two matched feature sets. The error metric used to find the best alignment is

$$\mathbf{M}_{ij}^{*} = \arg\max_{\mathbf{M}_{ij}} \left(\sum_{l=1}^{n} |\mathbf{M}_{ij}\mathbf{p}_i^l - \mathbf{p}_j^l|^2 \right) \tag{4}$$

Here, $\mathbf{p}_i^l$ and $\mathbf{p}_j^l$ contain the depth information of two matched feature points as described in Equation (1). Each term in the summation indicates the squared distance between the transformed pose of a feature point $\mathbf{p}_i^l$ in robot i's feature set and the matched feature point $\mathbf{p}_j^l$ in the robot j's feature set. Between every two matched feature sets, the central node samples a number of matched feature point pairs and determine the transformation matrix repeated. The determined transformation in each iteration is evaluated based on the number of inliers in the remaining 3D feature points. Ultimately, only the matched feature points which agree with the optimal transformation matrix are kept as the good

matches. The determined coarse registration between every two matched feature sets are stored as the *initial relative poses*. Initial relative poses are not accurate which require further refinements.

After operating the above process on every two feature sets, an Initial Pose Matrix (IPM) can be constructed. As shown in Table 2, each element, $\mathbf{M}^*_{ij}$, represents the initial relative pose between robot i and robot j. The diagonal elements represent the relative pose with itself, thus they are negligible.

Table 2. Initial Pose Matrix (IPM) and Uncertainty Matrix (UM) of a multi-robot system with four robots.

FMM					UM				
No.	**1**	**2**	**3**	**4**	**No.**	**1**	**2**	**3**	**4**
1	$\times$	$\mathbf{M}^*_{12}$	$\mathbf{M}^*_{13}$	$\mathbf{M}^*_{14}$	1	$\times$	w_{12}	w_{13}	w_{14}
2	$\mathbf{M}^*_{21}$	$\times$	$\mathbf{M}^*_{23}$	$\mathbf{M}^*_{24}$	2	w_{21}	$\times$	w_{23}	w_{24}
3	$\mathbf{M}^*_{31}$	$\mathbf{M}^*_{32}$	$\times$	$\mathbf{M}^*_{34}$	3	w_{31}	w_{32}	$\times$	w_{34}
4	$\mathbf{M}^*_{41}$	$\mathbf{M}^*_{42}$	$\mathbf{M}^*_{43}$	$\times$	4	w_{41}	w_{42}	w_{43}	$\times$

3.4. Selection of Relative Pose

After determining the neighboring robots and initial relative poses, we will show the problem on estimating all robots' poses can be transformed to the all-pair shortest path problem.

The relative pose between two neighboring robots can be estimated by RPE algorithm. In order to calibrate the whole system, we require to select a number of relative poses to link all robots together. Since different overlapping areas in FoVs lead to various uncertainty values in the RPE. This process should choose the relative poses with the minimum overall uncertainty between two robots. Furthermore, it is known that the accuracy of the estimation of the relative pose between two robots may significantly degrade when an increasing number of intermediate nodes are added into the computations. This is mainly due to uncertainty accumulated in each time RPE algorithm operates between two robots. In order to ensure each robot has the reliable knowledge of the other robots' locations and orientations, we require to select the relative poses which introduce the smallest overall amount of uncertainty value to calibration the system.

3.4.1. Robot Dependency Graph Construction

To efficiently consider all possible combinations of robot poses, we suggest the usage of the graph structure *robot dependency graph*. Robot dependency graph consists of a set of vertices representing each view of the scene observed by a robot. The weight on each edge indicates to the degree of uncertainty of the pair of views being connected. Thus, estimating all robots' poses can be transformed to finding the shortest path between every two vertices in the robot dependency graph.

In order to determine the weight on each edge, we need to first derive the uncertainty degree of relative pose estimation between every two neighbors. The relative pose between two neighboring robots can be estimated through aligning the 3D point clouds extracted from the depth images captured by different robots. The motion between two depth images can be estimated by various approaches, such as ICP variants [45–47], feature-based registrations [48,49], and combinational methods [50,51]. We choose to use our previous proposed algorithm [47] among the existing approaches, since it reports more accurate

and robust results in environments with various amount of occlusions than the state-of-the-art works. The performance of the RPE algorithms depends on the overlapping area between two FoVs. In the same circumstance, a larger overlapping area leads to a more accurate estimate. The overlapping area between two neighbors can be estimated by initial relative pose determined in Section 3.3.

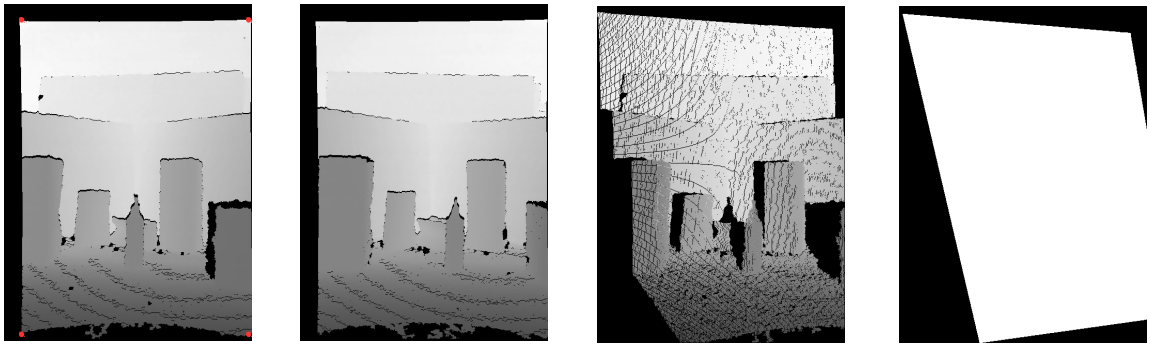
Let $\mathbf{M}_{ab}$ denote the relative pose between robots a and b. The pixels of the depth image captured by robot a can establish a relation with their projections on the depth image captured by robot b as

$$\begin{bmatrix} u_b & v_b & 1 & q_b \end{bmatrix}^T = \mathbf{M}_{ab} \begin{bmatrix} u_a & v_a & 1 & q_a \end{bmatrix}^T \tag{5}$$

$$\begin{bmatrix} \frac{i_b - i_{c,b}}{f_{x,b}} & \frac{j_b - j_{c,b}}{f_{y,b}} & 1 & \frac{1}{z_b} \end{bmatrix}^T = \mathbf{M}_{ab} \begin{bmatrix} \frac{i_a - i_{c,a}}{f_{x,a}} & \frac{j_a - j_{c,a}}{f_{y,a}} & 1 & \frac{1}{z_a} \end{bmatrix}^T \tag{6}$$

Here, (i_a, j_a) and (i_b, j_b) are the pixel coordinates on different depth images, $(i_{c,a}, j_{c,a})$ and $(i_{c,b}, j_{c,b})$ are the principal points of cameras on two robots, $(f_{x,a}, f_{y,a})$ and $(f_{x,b}, f_{y,b})$ are the focal lengths, and z_a and z_b are the depth values of the same real world point's projections in different depth images. Applying Equation (6) on the depth image observed by robot a, we can generate a synthetic view which is virtually taken at robot b's viewpoint. The overlapping area between two robots' FoVs can be determined through comparing the real and synthetic depth images. We define the *overlapping ratio* between two neighbors as the proportion of overlapping area in the observed image. However, in this approach the central node requires the knowledge of complete depth image of robot a. If all the robots have to transmit their observed depth images to the central node, the considerable transmission load will be generated. In order to efficiently estimate the overlapping ratio, robot a can only send the values and coordinates of four pixels in its observed depth image. These four pixels are the nearest pixels with valid depth values to the four corners (top left, top right, bottom left, and bottom right) of a image. After applying Equation (6) on these four pixels, the quadrangle constructed by the reprojections of these four pixels indicates the region observed by robot a in robot b's view. Though the points in the scene lay on different planes and have various range values, this approach can still provide a rough estimate of the overlapping ratio. An example is shown in Figure 4.

Figure 4. Overlapping area estimation. (**a**) Depth image captured by robot a, 4 corner pixels are highlighted in red; (**b**) Depth image captured by robot b; (**c**) Depth image virtually captured at robot b's position. It is generated from (**a**); (**d**) The rough estimate of overlapping area in robot b's view. White region indicates overlapping region.



(a) Robot a's view (b) Robot b's view (c) Synthetic view of robot b (d) Estimated overlapping area

We conducted numerical simulations on the RPE algorithm [47] to explore the relation between the overlapping ratio and the uncertainty in estimation. The equation

$$w_{ij} = \begin{cases} 1 & \text{if } \phi_{ij} \geq 0.7 \\ 1.5 & \text{if } 0.7 > \phi_{ij} \geq 0.6 \\ 2.4 & \text{if } 0.6 > \phi_{ij} \geq 0.5 \\ \infty & \text{if } 0.5 > \phi_{ij} \end{cases} \tag{7}$$

is adopted to quantize the overlapping ratio and uncertainty degree. Here, ϕ_{ij} represents the overlap ratio between two robots i and j, and w_{ij} indicates the uncertainty degree in RPE between two neighboring robots. A larger w_{ij} indicates a larger uncertainty value in the RPE. Based on IPM and criteria in Equation (7), the Uncertainty Matrix (UM) can be generated.

According to the UM, we can generate the robot dependency graph, $G = (V, A)$. There is an edge between any two neighboring robots iff the overlapping ratio is within the range in Equation (7). The weight of the edge linking robots i and j is w_{ij}, which indicates the uncertainty degree. A lower w_{ij} indicate a smaller uncertainty value in relative pose estimation. Then, the problem of relative pose selection is transformed to the all-pairs shortest path problem in the robot dependency graph, which minimizes the uncertainty in the RPE between every two robots.

The shortest path between every two vertices can be determined using Floyd–Warshall algorithm. The central node first generates $Dist$ as a $|V| \times |V|$ array of minimum distance and initializes $Dist$ according to UM. Then, Floyd–Warshall algorithm is used to determine the shortest paths between every pair of robots and update $Dist$. Then, the central node needs to select one robot as the *primary robot* and make all the other robots calibrate their poses according to the primary robot's coordinate system. In order to minimize the uncertainty, the central node selects the robot which has the smallest overall weight on the shortest paths to all the other robots as the primary robot. At last, the robots can be connected as a *calibration tree* with the primary robot as the root. In this method, the RPE algorithm only requires to operate $|V| - 1$ times to connect all the robots. The time complexity of the overall scheme is $O(V)$. Thus, this scheme is scalable to initialize multi-robot systems with a large number of robots.

3.5. Distributed Relative Pose Estimation Algorithm

Though the initial relative poses on the edges of the calibration tree have already be obtained in neighbor detection process, these estimations are too inaccurate to calibrate the overall system. Therefore, after the calibration tree is built, the central node will broadcast the information of the calibration tree and the related initial relative poses to all the robots. Then our earlier work [47] implemented on every robot will operate to refine the initial relative poses. In our earlier work [47], an Iterative Closest Points (ICP) variant was proposed to estimate the relative pose between two RGB-D camera equipped robots. Different from the methods for conventional RGB camera which use feature correspondences to determine the rotation and translation up to a scale, this distributed, peer-to-peer algorithm determines the relative pose in consistent real world scale through explicit registration of surface geometries extracted from two depth images. The registration problem is approached by

iteratively minimizing a cost function in which error metrics are defined based on the bidirectional point-to-plane geometrical relationship.

$\mathbf{Z}_a$ and $\mathbf{Z}_b$ are two depth images captured by robots a and b. The correspondences between $N = N_a + N_b$ pairs of points extracted from $\mathbf{Z}_a$ and $\mathbf{Z}_b$ are established. We can then estimate the transformation matrix $\mathbf{M}_{ab}$ by minimizing the bidirectional point-to-plane error metric, $\mathcal{C}$, expressed in normal least squares form as follows,

$$\mathcal{C} = \sum_{l=1}^{N_a} \left[w_{l,a} (\mathbf{M}_{ab} \mathbf{p}_a^l - \mathbf{p}_b^{l^*}) \cdot \vec{n}_{l^*,b} \right]^2 + \sum_{k=1}^{N_b} \left[w_{k,b} (\mathbf{M}_{ab}{}^{-1} \mathbf{p}_a^{k^*} - \mathbf{p}_b^k) \cdot \vec{n}_{k,b} \right]^2 \tag{8}$$

where $\mathbf{p}_a^l$ and $\mathbf{p}_b^k$ are the sampled points in depth frames $\mathbf{Z}_a$ and $\mathbf{Z}_b$, $\mathbf{p}_b^{l*}$ and $\mathbf{p}_a^{k*}$ are their corresponding points on the other depth image respectively. The variables $w_{l,a}$ and $w_{k,b}$ are weight parameters for correspondences established in opposite directions between pairs. The variables $w_{l,a}$ and $w_{k,b}$ are weight parameters for correspondences established in opposite directions between pairs. Also, $\vec{n}_{l^*,b}$ and $\vec{n}_{k,b}$ are the surface normals at the corresponding points $\mathbf{p}_b^{l^*}$ and $\mathbf{p}_b^k$ in real world coordinates, and

$$\vec{n}_{l^*,b} = \begin{bmatrix} \alpha_{l^*,b} & \beta_{l^*,b} & \gamma_{l^*,b} & 0 \end{bmatrix}^T \tag{9}$$

$$\vec{n}_{k,b} = \begin{bmatrix} \alpha_{k,b} & \beta_{k,b} & \gamma_{k,b} & 0 \end{bmatrix}^T \tag{10}$$

The cost function presented in Equation (8) consists of two parts:

1. the sum of squared distances in the forward direction from depth images $\mathbf{Z}_a$ to $\mathbf{Z}_b$, and
2. the sum of square distances in the backward direction from $\mathbf{Z}_b$ to $\mathbf{Z}_a$.

Algorithm 1 Relative pose refinement procedure

1: Capture a depth image, $\mathbf{Z}_a$, on robot a, and capture a depth image, $\mathbf{Z}_b$, on robot b.
2: Initialize the transformation matrix, $\mathbf{M}_{ab}$, by the initial relative pose.
3: **procedure** REPEAT UNTIL CONVERGENCE
4: Update depth frame $\mathbf{Z}_a$ according to transformation matrix.
5: Randomly sample N_a points from $\mathbf{Z}_a$ to form set P_a,
$P_a = \{\mathbf{p}_a^k \in \mathbf{Z}_a, k = 1, \ldots, N_a\}$,
6: Randomly sample N_b points from $\mathbf{Z}_b$ to form set P_b,
$P_b = \{\mathbf{p}_b^k \in \mathbf{Z}_b, k = 1, \ldots, N_b\}$.
7: Find the corresponding point set, P_b^*, of P_a in $\mathbf{Z}_b$,
$P_b^* = \{\mathbf{p}_b^{k*} \in \mathbf{Z}_b, k = 1, \ldots, N_a\}$;
Find the corresponding point set, P_a^*, of P_b in $\mathbf{Z}_a$,
$P_a^* = \{\mathbf{p}_a^{k*} \in \mathbf{Z}_a, k = 1, \ldots, N_b\}$.
▷ The correspondences are established using the project and walk method with a neighborhood size of 3x3 based on the nearest neighbor criteria
8: Apply the weight function bidirectionally,
$P_a \mapsto P_b^*$, $P_b \mapsto P_a^*$
9: Compute and update transformation matrix based on current bidirectionally weighted correspondences
10: **end procedure**

We can then estimate $\mathbf{M}_{ab}$ by re-weighting the least squares operation in an ICP framework. Details of the criteria for selecting the weight function can be found in [47,52]. An overview of the entire process is presented in Algorithm 1. In the first iteration, $\mathbf{M}_{ab}$ is initialized by the initial relative pose determined

in neighbor detection process. Afterwards, in this coarse-to-fine algorithm, each iteration generates an update $\mathbf{E}$ to the robot's pose which modifies the transformation matrix $\mathbf{M}_{ab}$. $\mathbf{E}$ takes the same form as $\mathbf{M}_{ab}$ which may be parameterized by a 6-dimensional motion vector having the elements $\alpha_1, \alpha_2, \ldots, \alpha_6$ via the exponential map and their corresponding group generator matrices $\mathbf{G}_1, \mathbf{G}_1, \ldots, \mathbf{G}_6$ as

$$\mathbf{E} = \exp\left(\sum_{j=1}^{6} \alpha_j \mathbf{G}_j\right) \tag{11}$$

where

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{G}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{G}_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{G}_6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Here $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{G}_3$ are the generators of translations in x, y and z directions, while $\mathbf{G}_4$, $\mathbf{G}_5$ and $\mathbf{G}_6$ are rotations about x, y and z axes respectively.

Afterwards, the task becomes finding the $\alpha_1, \ldots, \alpha_6$ that describe the relative pose. Through determining the partial derivatives of u_b, v_b and q_b with respect to the unknown elements of the motion vector $\alpha_1, \ldots, \alpha_6$, the Jacobian matrix for each established corresponding point pair can be obtained from

$$\mathbf{J} = \begin{bmatrix} q_a & 0 & -u_a q_a & -u_a v_a & 1 + u_a^2 & -v_a \\ 0 & q_a & -v_a q_a & -1 - v_a^2 & v_a u_a & u_a \\ 0 & 0 & -q_a^2 & -v_a q_a & u_a q_a & 0 \end{bmatrix} \tag{12}$$

The six-dimensional motion vector, which minimizes Equation (8), is then determined iteratively by the least squares solution

$$\mathbf{B} = (\mathbf{K}^T \mathbf{W} \mathbf{K})^{-1} \mathbf{K}^T \mathbf{W} \mathbf{Y} \tag{13}$$

in which $\mathbf{W}$ is a diagonal matrix weighting the bidirectional point-to-plane correspondences. $\mathbf{B}$, $\mathbf{Y}$, and $\mathbf{K}$ are matrices

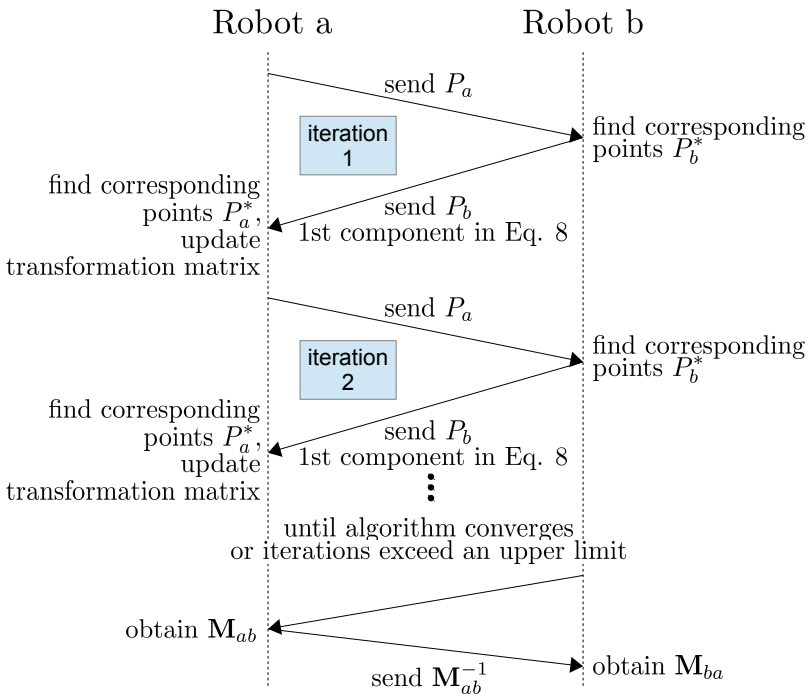
$$\mathbf{B} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} -(\mathbf{p}_a^1 - \mathbf{p}_b^{1^*}) \cdot \vec{n}_{1^*,b} \\ \vdots \\ -(\mathbf{p}_a^{N_a} - \mathbf{p}_b^{N_a^*}) \cdot \vec{n}_{N_a^*,b} \\ -(\mathbf{p}_a^{1^*} - \mathbf{p}_b^1) \cdot \vec{n}_{1,b} \\ \vdots \\ -(\mathbf{p}_a^{N_b^*} - \mathbf{p}_b^{N_k}) \cdot \vec{n}_{N_b,b} \end{bmatrix} \tag{14}$$

$$\mathbf{K} = \left[\vec{n}'_{1^*,b}\mathbf{J}_1 \ ... \ \vec{n}'_{N_a^*,b}\mathbf{J}_{N_a} \ \vec{n}'_{1,b}\mathbf{J}_{1^*} \ ... \ \vec{n}'_{N_b,b}\mathbf{J}_{N_b^*}\right]^T \tag{15}$$

Here, $\vec{n}'_{l,b} = \begin{bmatrix}\alpha_{l,b} & \beta_{l,b} & \gamma_{l,b}\end{bmatrix}^T$ is the surface normal expressed in a slightly different form than the one shown in Equations (9) and (10). To detect the convergence of our algorithm, we use the thresholds for the ICP framework presented in [46]. Once the algorithm converges, the registration is considered as completed and the $\mathbf{M}_{ab}$ is refined based on the initial relative pose.

In reality each robot only has its own captured depth image. In order to efficiently accomplish the centralized working principle of the algorithm described above, we distribute the tasks to two robots. Instead of transmitting a complete depth image from one robot to another, each robot only transmits a number of sampled points on its captured depth image to the other robot. The distributed process is illustrated in Figure 5.

Figure 5. Distributing the tasks to two robots.



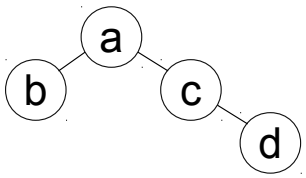
At each iteration, after robot b receives the sampled point set, P_a, from robot a, robot b will find the corresponding point set, P_b^*, on its captured depth frame $\mathbf{Z}_b$. The first component in Equation (8) will be derived. The information representing the first component will be sent along with the sampled point set, P_b, from robot b to robot a. At robot a, P_b's corresponding point set, P_a^*, will be determined. And the second component in Equation (8) will be derived. Thereby, robot a will acquire the information of both first and second component in Equation (8), and the motion parameters can be determined. These procedures will be performed in each iteration. The transformation matrix describing the relative pose between two robots will be obtained by robot a until the algorithm converges. More details and performance evaluations of this approach can be found in [47].

The relative poses that construct the calibration tree are transmitted to the central node after being determined by robots. Then all robots' locations and orientations can be calibrated according to primary robot's coordinate system. A simple example of the working process is shown below.

Figure 6 depicts a calibration tree for initializing a multi-robot system with 4 robots. Robots operate RPE algorithm to derive the relative poses $\mathbf{M}_{ab}$, $\mathbf{M}_{ac}$, and $\mathbf{M}_{cd}$ according to the tree. By using these three pose matrices, the relative pose between every two robots in the network can be derived. For instance, robot d's location and orientation in robot b's coordinate system can be derived as

$$\mathbf{M}_{bd} = \mathbf{M}_{ad}\mathbf{M}_{ba} = \mathbf{M}_{cd}\mathbf{M}_{ac}\mathbf{M}_{ab}^{-1} \tag{16}$$

Figure 6. Example of a calibration tree.



4. Experimental Results and Discussion

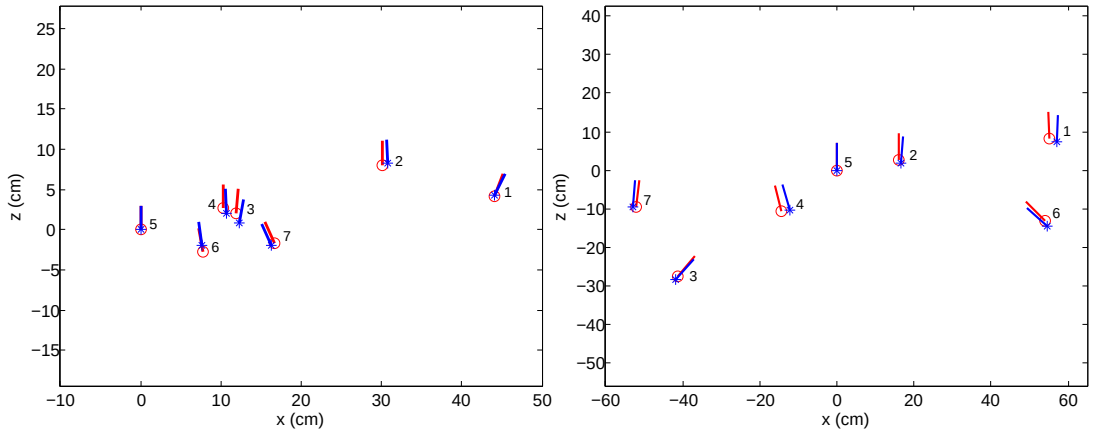
In order to quantitatively evaluate the performance of the proposed method, we conducted a series of experiments both in simulation and with our custom built multi-robot system. Section 4.1 describes the localization experiments that were carried out with our experimental multi-robot system in indoor environments. Section 4.2 presents the results of a set of simulations that were designed to further evaluate the behavior of the method.

4.1. Indoor Experiments

We used the color and depth images captured by our experimental multi-robot system consisting of seven RGB-D camera equipped robots. The images were taken from various locations in and around WSRNLab facility. Color images of collected five scenes are shown in Figure 7. As the robots are deployed on the same plane in this set of experiments, the ground truth locations and orientations can be easily measured manually. The estimated robots' poses are shown in Figure 8, in which the estimated poses are depicted in red circles and the ground truth are represented in blue stars. We derived the average absolute errors accordingly and presented in Table 3. The calibration trees of 5 scenes are illustrated in Figure 9. We can see the pose estimations of Scene 1 have the smallest absolute error, while the estimates in Scene 5 have the largest absolute error. We also measure the average relative error for localization. The relative errors are computed based on the absolute errors and systems' dimensions. It is clear that the pose estimation results in Scene 4 and Scene 5 are the most and least accurate among five scenes respectively. Through analyzing the robots' sensing ranges in different scenes, we find that sensing range is a main factor that affects the performance of our proposed scheme. As reported in [39], the errors in the depth measurements of Kinect increase quadratically from a few millimeters at 0.5 m distance to about 4 cm at the maximum range of the sensor. The inaccurate depth information obtained by the Kinect on each robot influences the performance of the distributed relative pose estimation algorithm, thereby decreasing the accuracy of the overall scheme. Due to this limitation of the RGB-D camera, the robots' sensing ranges should be controlled between around 0.5 m to 3.5 m.

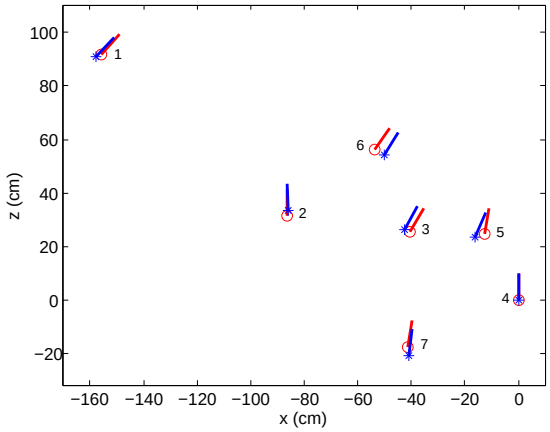
Figure 7. Color images captured by the multi-robot system in 5 scenes.

Figure 8. Estimated and ground truth robots' poses. Estimated locations are depicted in red circles and the ground truth are represented in blue stars. The line segments on different markers indicate orientations. (**a**) Scene 1; (**b**) Scene 2; (**c**) Scene 3; (**d**) Scene 4; (**e**) Scene 5.

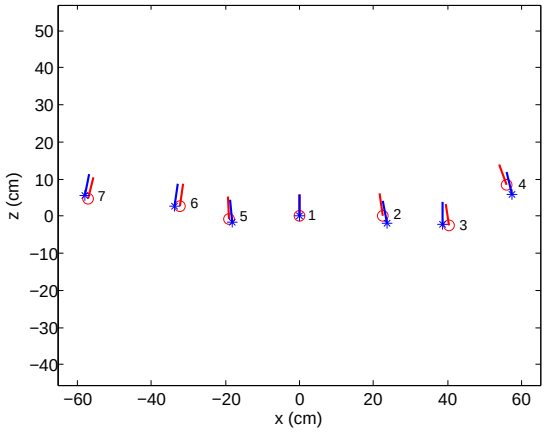


(a)

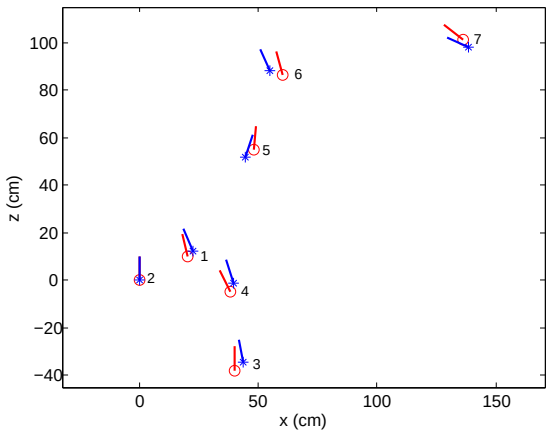
(b)



(c)

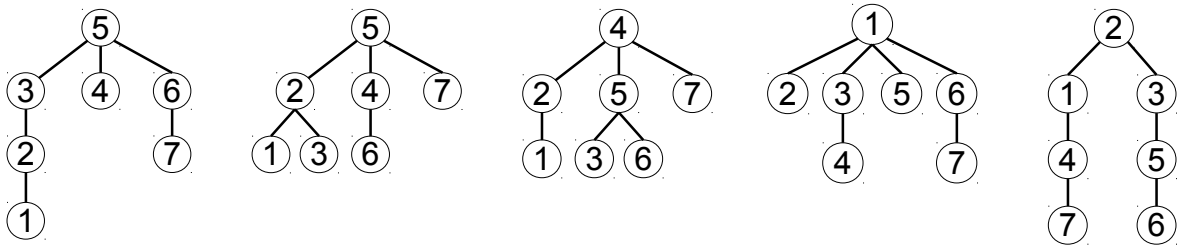


(d)



(e)

Figure 9. Calibration trees of indoor experiments.



(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4 (e) Scene 5

Table 3. Average error between the estimated poses and the ground truth.

Data Set	Sensing Range		Average Absolute Error		Localization Average Relative Error
	Max (m)	Average (m)	Location (mm)	Orientation (°)	
1	1.92	1.47	10.0	1.6	2.26%
2	6.23	1.72	14.8	2.3	1.36%
3	3.95	1.86	25.1	2.7	1.39%
4	1.79	1.41	12.6	2.1	1.12%
5	6.02	4.14	64.7	6.2	3.81%

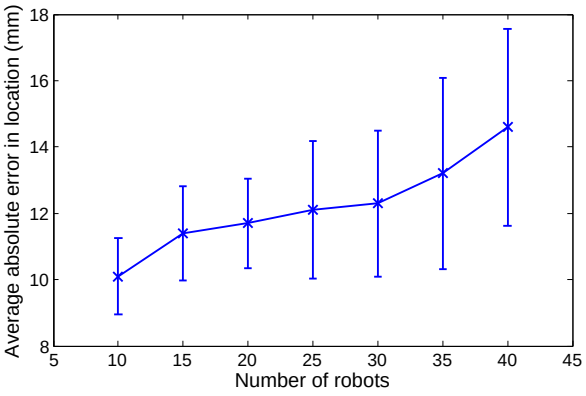
4.2. Simulation Experiments

A series of simulation experiments were conducted to investigate the accuracy and bandwidth consumption of the proposed scheme implemented on systems that were larger and have more complicated topologies than the ones we could construct with our available hardware. When robots are deployed on different planes, the ground truth poses can hardly be measured precisely by the manual methods. Therefore, in this set of simulations we applied 3D image warping technique [53] on the color and corresponding depth images captured in Scene 4 to generate synthetic image sets with known transformation matrices. Image sets are generated for systems consisting of 10, 15 , 20, 25, 30, 35, and 40 robots. In this process, we make sure that each robot has sufficiently overlapping FoV with at least one another robot and can be connected in calibration tree. The results presented in Figure 10 are averaged over 10 runs of the simulations with vertical bars indicating the variance.

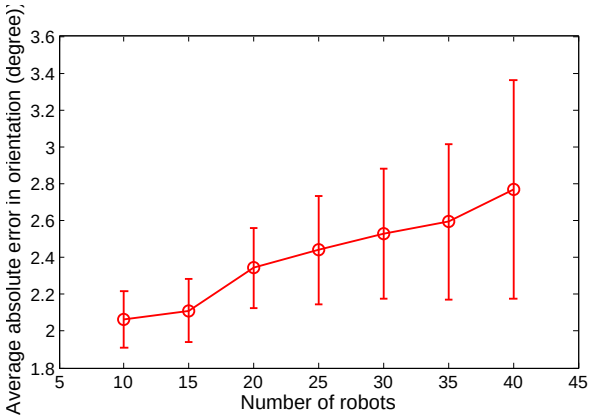
Figure 10a,b, indicate the absolute errors in both location and orientation increase as the number of robots in the system grows. It is because when the number of robots raises, the calibration tree becomes larger and the primary robot requires more intermediate nodes to establish the connection with another robot in the system. Though this effect accumulates small errors, the average absolute errors are still quite small and the average relative error is within 1.3%. The average bandwidth consumption of each robot is presented in Figure 10c. As expected, the bandwidth usage per robot remains consistent approximately in relation to the number of robots in the system. There are two processes in our proposed scheme require communication between robots: (1) Neighbor detection; and (2) distributed relative pose estimation. As the number of times that distributed relative pose estimation runs increase linearly with number of robots, the each robot's transmission load in this process will not be influenced by the number of robots in the system. The only variable that affects the transmission load is the number of the feature points in the neighbor detection process. However, the number of feature points, which depends on the

structure of the captured scene, is irrelative with the number of robots. Therefore, the evenly distributed communication load throughout the system indicates the good scalability of our proposed scheme.

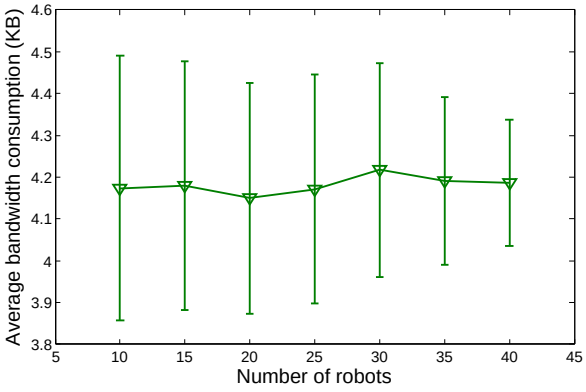
Figure 10. Simulation results. (**a**) Average absolute error in location estimation with increasing number of robots in the system; (**b**) Average absolute error in estimated orientation with increasing number of robots in the system; (**c**) Bandwidth consumption per robot with increasing number of robots in the system.



(a)



(b)



(c)

5. Conclusions

This paper describes the first approach which uses color and depth information to initialize robots' poses in a RGB-D camera equipped multi-robot system. Our scheme first detects each robot's neighbors using robust feature matching. Then the overlapping area between FoVs of neighboring robots are determined to establish the robot dependency graph. A calibration tree is generated through finding the shortest path between the primary robot to all the other robots in the system. At last, a distributed relative pose estimation algorithm is performed to precisely compute relative pose between every two connected robots in the calibration tree. Extensive real world experiments and synthetic dataset simulations have been conducted. The results show that our scheme is robust and accurate in various environments and densities of robots. Importantly, the proposed scheme operates distributively and allows the robots to use the limited wireless bandwidth more efficiently. This calibration algorithm, which provides initial

location and orientation information, has great potentials to be used in a wide range of applications, such as visual SLAM and 3D reconstruction.

In future, we plan to present a distributed solution to enhance the performance of the current semi-centralized framework. In order to achieve this goal, we require to upgrade the hardware of eyeBug. Therefore, eyeBug obtains better computational ability and can operate more complicated computer vision algorithms. Further, we would like to develop an efficient algorithm which can keep updating robots' relative poses in real time. This algorithm can be built on our framework to realize the refinement process in cooperative localization.

Author Contributions

Xiaoqin Wang designed the overall scheme, created the datasets using the multi-robot system developed by WSRNLab researchers, tested the scheme and analyzed the results, and prepared the first draft of the manuscript. Y. Ahmet Şekercioğlu proposed the idea about using depth data in conjunction with color information, created the robot system, provided the experimental facilities and valuable input to the manuscript. Tom Drummond contributed the mathematical deduction and formulas in distributed relative pose estimation algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Oyekan, J.; Hu, H. Ant Robotic Swarm for Visualizing Invisible Hazardous Substances. *Robotics* **2013**, *2*, 1–18.
2. Parker, L.E. Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets. *Auton. Robots* **2002**, *12*, 231–255.
3. Delle Fave, F.; Canu, S.; Iocchi, L.; Nardi, D.; Ziparo, V. Multi-Objective Multi-Robot Surveillance. In Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 10–12 February 2009; pp. 68–73.
4. Karakaya, M.; Qi, H. Collaborative Localization in Visual Sensor Networks. *ACM Trans. Sens. Netw.* **2014**, *10*, 18:1–18:24.
5. Stroupe, A.W.; Martin, M.C.; Balch, T. Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems. In Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 21-26 May 2001; Volume 2, pp. 1092–1098.
6. Soto, C.; Song, B.; Roy-Chowdhury, A.K. Distributed Multi-Target Tracking in a Self-Configuring Camera Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1486–1493.
7. Xu, Y.; Qi, H. Mobile Agent Migration Modeling and Design for Target Tracking in Wireless Sensor Networks. *Ad Hoc Netw.* **2008**, *6*, 1–16.
8. Dong, W. Tracking Control of Multiple-Wheeled Mobile Robots With Limited Information of a Desired Trajectory. *IEEE Trans. Robot.* **2012**, *28*, 262–268.

9. Dong, W.; Chen, C.; Xing, Y. Distributed estimation-based tracking control of multiple uncertain non-linear systems. *Int. J. Syst. Sci.* **2014**, *45*, 2088–2099.
10. Dong, W.; Djapic, V. Leader-following control of multiple nonholonomic systems over directed communication graphs. *Int. J. Syst. Sci.* **2014**, doi: 10.1080/00207721.2014.955553.
11. Samperio, R.; Hu, H. Real-Time Landmark Modelling for Visual-Guided Walking Robots. *Int. J. Comput. Appl. Technol.* **2011**, *41*, 253–261.
12. Gil, A.; Reinoso, Ó.; Ballesta, M.; Juliá, M. Multi-Robot Visual SLAM Using a Rao-Blackwellized Particle Filter. *Robot. Auton. Syst.* **2010**, *58*, 68–80.
13. Chow, J.C.; Lichti, D.D.; Hol, J.D.; Bellusci, G.; Luinge, H. IMU and Multiple RGB-D Camera Fusion for Assisting Indoor Stop-and-Go 3D Terrestrial Laser Scanning. *Robotics* **2014**, *3*, 247–280.
14. Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334.
15. Kitahara, I.; Saito, H.; Akimichi, S.; Ono, T.; Ohta, Y.; Kanade, T. Large-scale virtualized reality. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
16. Chen, X.; Davis, J.; Slusallek, P. Wide Area Camera Calibration Using Virtual Calibration Objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA, 13–15 June 2000.; Volume 2, pp. 520–527.
17. Svoboda, T.; Hug, H.; Gool, L.J.V. ViRoom-Low Cost Synchronized Multicamera System and Its Self-calibration. In Proceedings of the 24th DAGM Symposium on Pattern Recognition, Zurich, Switzerland, 16–18 September 2002; pp. 515–522.
18. Svoboda, T.; Martinec, D.; Pajdla, T. A Convenient Multi-Camera Self-Calibration for Virtual Environments. *Teleoperators Virtual Environ.* **2005**, *14*, 407–422.
19. Hörster, E.; Lienhart, R. Calibrating and Optimizing Poses of Visual Sensors in Distributed Platforms. *Multimed. Syst.* **2006**, *12*, 195–210.
20. Läbe, T.; Förstner, W. Automatic Relative Orientation of Images. In Proceedings of the 5th Turkish-German Joint Geodetic Days, Berlin, Germany, 28–31 March 2006; Volume 29, p. 31.
21. Rodehorst, V.; Heinrichs, M.; Hellwich, O. Evaluation of Relative Pose Estimation Methods for Multi-Camera Setups. *Int. Arch. Photogram. Remote Sens.* **2008**, pp. 135–140.
22. Jaspers, H.; Schauerte, B.; Fink, G.A. Sift-Based Camera Localization Using Reference Objects for Application in Multi-camera Environments and Robotics. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, vilamoura, portugal, 6–8 February 2012; pp. 330–336.
23. Aslan, C.T.; Bernardin, K.; Stiefelhagen, R.; others. Automatic Calibration of Camera Networks Based on Local Motion Features. In Proceedings of the Workshop on Multi-Camera and Multi-Modal Sensor Fusion Algorithms and Applications, Marseille, France, October 2008.
24. Devarajan, D.; Radke, R.J. Distributed Metric Calibration of Large Camera Networks. In Proceedings of the First Workshop on Broadband Advanced Sensor Networks (BASENETS), San Jose, CA, USA, 25 October 2004; Volume 3, pp. 5–24.

25. Kurillo, G.; Li, Z.; Bajcsy, R. Wide-Area External Multi-Camera Calibration Using Vision Graphs and Virtual Calibration Object. In Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras, Stanford, CA, USA, 7–11 September 2008; pp. 1–9.
26. Cheng, Z.; Devarajan, D.; Radke, R.J. Determining Vision Graphs for Distributed Camera Networks Using Feature Digests. *EURASIP J. Appl. Signal Process.* **2007**, *2007*, 220–220.
27. Vergés-Llahı, J.; Moldovan, D.; Wada, T. A New Reliability Measure for Essential Matrices Suitable in Multiple View Calibration. In Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Funchal, Portugal, 22–25 January 2008; pp. 114–121.
28. Bajramovic, F.; Denzler, J. Global Uncertainty-based Selection of Relative Poses for Multi Camera Calibration. In Proceedings of the British Machine Vision Conference, 1–4 September 2008, Leeds, UK; pp. 1–10.
29. Bajramovic, F.; Brückner, M.; Denzler, J. An Efficient Shortest Triangle Paths Algorithm Applied to Multi-Camera Self-Calibration. *J. Math. Imaging Vis.* **2012**, *43*, 89–102.
30. Brückner, M.; Bajramovic, F.; Denzler, J. Intrinsic and Extrinsic Active Self-Calibration of Multi-Camera Systems. *Mach. Vis. Appl.* **2014**, *25*, 389–403.
31. Wireless Sensor and Robot Networks Laboratory (WSRNLab). Available online: http://wsrnlab.ecse.monash.edu.au (accessed on 01/12/2014).
32. Beagleboard-xM System Reference Manual. Available online: http://beagleboard.org/static/ (accessed on 01/12/2014).
33. Ubuntu Server for ARM Processor Family. Available online: http://www.ubuntu.com/download/server/arm (accessed on 01/12/2014).
34. OpenKinect Library. Available online: http://openkinect.org (accessed on 01/12/2014).
35. OpenCV: Open Source Computer Vision Library. Available online: http://opencv.org (accessed on 01/12/2014).
36. libCVD-Computer Vision Library. Available online: http://www.edwardrosten.com/cvd/ (accessed on 01/12/2014).
37. Arieli, Y.; Freedman, B.; Machline, M.; Shpunt, A. Depth Mapping Using Projected Patterns. U.S. Patent 8,150,142, 2012.
38. Butler, D.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Hodges, S.; Kim, D. Shake'n'Sense: Reducing Interference for Overlapping Structured Light Depth Cameras. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, Austin, Texas, 5–10 May 2012; pp. 1933–1936.
39. Khoshelham, K. Accuracy Analysis of Kinect Depth Data. In Proceedings of the ISPRS Workshop Laser Scanning, Calgary, Canada, 29–31 August 2011; Volume 38, p. W12.
40. Stowers, J.; Hayes, M.; Bainbridge-Smith, A. Altitude Control of a Quadrotor Helicopter Using Depth Map from Microsoft Kinect Sensor. In Proceedings of the 2011 IEEE International Conference on Mechatronics, Istanbul, Turkey, 13–15 April 2011; pp. 358–362.
41. Khoshelham, K.; Elberink, S.O. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors* **2012**, *12*, 1437–1454.

42. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In *Computer Vision–ECCV 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
43. Rosten, E.; Porter, R.; Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 105–119.
44. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
45. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; Fitzgibbon, A. Kinectfusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In Proceedings of the UIST, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568.
46. Lui, W.; Tang, T.; Drummond, T.; Li, W.H. Robust Egomotion Estimation Using ICP in Inverse Depth Coordinates. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 1671–1678.
47. Wang, X.; Şekercioğlu, Y.A.; Drummond, T. A Real-Time Distributed Relative Pose Estimation Algorithm for RGB-D Camera Equipped Visual Sensor Networks. In Proceedings of the 7th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2013), Palm Springs, CA, USA, 29 October–1 November 2013.
48. Zou, Y.; Chen, W.; Wu, X.; Liu, Z. Indoor Localization and 3D Scene Reconstruction for Mobile Robots Using the Microsoft Kinect Sensor. In Proceedings of the 10th IEEE International Conference on Industrial Informatics, Beijing, China, 25–27 July 2012; pp. 1182–1187.
49. Wang, H.; Mou, W.; Ly, M.H.; Lau, M.; Seet, G.; Wang, D. Mobile Robot Egomotion Estimation Using RANSAC-Based Ceiling Vision. In Proceedings of the 24th Chinese Control and Decision Conference, Taiyuan, China, 23–25 May 2012; pp. 1939–1943.
50. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663.
51. Endres, F.; Hess, J.; Engelhard, N.; Sturm, J.; Cremers, D.; Burgard, W. An Evaluation of the RGB-D SLAM System. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2012), Saint Paul, MN, USA, 14–18 May 2012; pp. 1691–1696.
52. Holland, P.W.; Welsch, R.E. Robust regression using iteratively reweighted least-squares. *Commun. Stat.-Theory Meth.* **1977**, *6*, 813–827.
53. Mori, Y.; Fukushima, N.; Fujii, T.; Tanimoto, M. View Generation with 3D Warping Using Depth Information for FTV. In Proceedings of the 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, Potsdam, Germany, 4–6 May 2008; pp. 229–232.