A Real-Time Distributed Relative Pose Estimation Algorithm for RGB-D Camera Equipped Visual Sensor Networks

Xiaoqin Wang, Y. Ahmet Şekercioğlu and Tom Drummond
Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia
Email: {Xiaoqin.Wang|Ahmet.Sekercioglu|Tom.Drummond}@monash.edu

***Abstract*—In mobile visual sensor networks, relative pose (location and orientation) estimation is a prerequisite to accomplish a wide range of collaborative tasks. In this paper, we present a distributed, peer-to-peer algorithm for relative pose estimation in a network of mobile robots equipped with RGB-D cameras acting as a visual sensor network.**

Our algorithm uses the depth information to estimate the relative pose of a robot when camera sensors mounted on different robots observe a common scene from different angles of view. To create the algorithm, we first developed a framework based on the beam-based sensor model to eliminate the adverse effects of the situations where two views of a scene each are partially seen by the sensors. Then, in order to cancel the bias introduced by the beam-based sensor model, we developed a scheme that allows the algorithm to symmetrize across the two views.

We conducted simulations and also implemented the algorithm on our mobile visual sensor network testbed. Both the simulations and experimental results indicate that the proposed algorithm is fast enough for real-time operation and able to maintain a high estimation accuracy. To our knowledge, it is the first distributed relative pose estimation algorithm that uses the depth information captured by multiple RGB-D cameras.

I. Introduction

The latest advances in video technology, inexpensive camera sensors, and distributed processing allow the wide utilization of image sensors. It has resulted in a new paradigm– visual sensor network [1]. Visual sensor networks observe and process image/video data give rich information on situation awareness. Replacing the conventional RGB cameras with RGB-D camera sensors (e.g., Microsoft Kinect [2]) which can capture color image along with per-pixel depth information, visual sensor networks promise a wider range of the innovative applications, such as 3D reconstruction, object localization, etc.

In this paper, we consider mobile visual sensor networks of robots equipped with RGB-D camera sensors to observe the environment. We treat each robot as a mobile RGB-D sensor. The goal is to enable each mobile RGB-D sensor to obtain the precise location and orientation information of other sensors. In order to achieve this goal, we present a peer-to-peer, distributed depth image registration algorithm estimating the relative pose between multiple sensors when two or more sensors observe a common scene from different angles.

Many research works have been proposed to determine the pose of a single RGB-D sensor in the last few years. The most of the pose estimation methods operate in a frame-to-frame tracking manner via estimating the motion between every two consecutive frames. The interframe motion, between a pair of RGB-D images, is normally estimated through the explicit matching of surface geometry. These methods can be classified into three main categories: (1) Iterative Closest Point (ICP) variants ([3], [4], [5], [6]), (2) feature-based registrations ([7], [8]), and (3) hybrid approaches ([9], [10], [11], [12], [13]).

The ICP variants approach the registration problem by iteratively minimizing a cost function whose error metrics are defined based on the point-to-point, point-to-plane or other geometrical relationships. ICP [14] was made popular following its successful application in the registration of highly accurate range data from laser rangefinders with a wide field-of-view (FoV). However, due to the narrow FoV of RGB-D sensors, occlusion and limited overlapping region between two views of a scene can easily lead to the failure of the ICP variants.

Unlike ICP variants that match the randomly sampled points on two consecutive depth frames, the feature based methods first detect and match texture feature points on consecutive color frames. Then the corresponding depth information of the matched feature points are used to determine the motion parameters. As the feature detection requires sufficient visual contents and consistent illumination in the scene, these approaches have limitations in many situations, such as the dark environments.

In order to improve the estimation accuracy, the hybrid approaches combine ICP variants and feature-based registration algorithms together. In this kind of approaches, the feature registration is usually adopted to provide a rough estimation of the motion parameters. Then, ICP variants are used to refine the result. These approaches inherit the limitations of ICP and feature registration. Furthermore, these approaches are computationally expensive and require GPU to operate in real time.

All of these approaches focus on estimating the egomotion of a single RGB-D sensor. The disadvantages of these egomotion estimation algorithms prevent them from being directly applied to estimate the relative pose between multiple computationally constrained sensors.

Our proposed novel algorithm fills the gap existing in the area of relative pose estimation between multiple RGB-

D sensors. In this algorithm, a maximum likelihood framework based on beam-based sensor model [15] is devised and incorporated with ICP framework to enhance the limited performance of ICP variants in relative pose estimation. The algorithm was implemented and tested both on a laptop and our visual senor network testbed comprised of mobile RGB-D sensors. Extensive experiments using existing datasets and real world data are conducted to examine the performance of the proposed algorithm under different 3D scenes.

The rest of the paper is organized as follows. The problem and task are presented in Section II. Section III discusses our proposed work and algorithm for relative pose estimation in details. Experimental setup and summarize of our key findings are described in Section IV. The final section draws the conclusion and identifies the directions of future work.

II. Problem Statement

As an RGB-D sensor can provide a continuous measurement of the 3D structure within the environment, the relative pose between two RGB-D sensors can be estimated through explicit matching of surface geometry. The relative pose between two sensors a, b can be represented by a transformation matrix, $\mathbf{M}_{ab}$, in SE(3),

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

where $\mathbf{R}$ is a 3×3 matrix indicating the relative orientation, and $\mathbf{t}$ is a 3×1 vector representing the relative position. The subscript ab indicates the sensor a's pose relative to sensor b's pose.

Let $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images of the same scene captured by two separated RGB-D sensors (see Fig. 1). For a depth image, $\mathbf{Z}_a$, is made up of N pixels where each pixel contains the corresponding depth information. After calibrating the RGB-D sensor, the depth pixel, $\mathbf{p}_a^k$, which contains the range information and the pixel coordinates in frame $\mathbf{Z}_a$, can find its relationship to a corresponding world point. It can be expressed as

$$\begin{aligned} \mathbf{p}_a^k \equiv 1/z_a^k[x_a^k, y_a^k, z_a^k, 1]^T &= [u_a^k, v_a^k, 1, q_a^k]^T \\ &= [\frac{i_a^k - i_{c,a}}{f_{x,a}}, \frac{j_a^k - j_{c,a}}{f_{y,a}}, 1, 1/z_a^k]^T. \end{aligned} \tag{2}$$

(x_a^k, y_a^k, z_a^k) denotes the corresponding point in Euclidean space represented using homogeneous coordinates. (i_a^k, j_a^k) denotes the pixel coordinates in the image, $(i_{c,a}$,$j_{c,a})$ is the principal point offset and $(f_{x,a}$,$f_{y,a})$ is the focal length of the RGB-D sensor a. It is more convenient to solve for pose using this formulation because (u, v) are a linear function of pixel position. And it preserves the linear relationship with the normalized disparity values and avoids conversion to 3D Euclidean space which has non homogeneous and anisotropic noise characteristics. With the accurate information of the transformation matrix, each pixel with coordinates (i_a^k, j_a^k) in depth image $\mathbf{Z}_a$ can find its corresponding pixel in $\mathbf{Z}_b$ at (i_b^k, j_b^k), which represents the same world point, by applying the rigid body transformation as

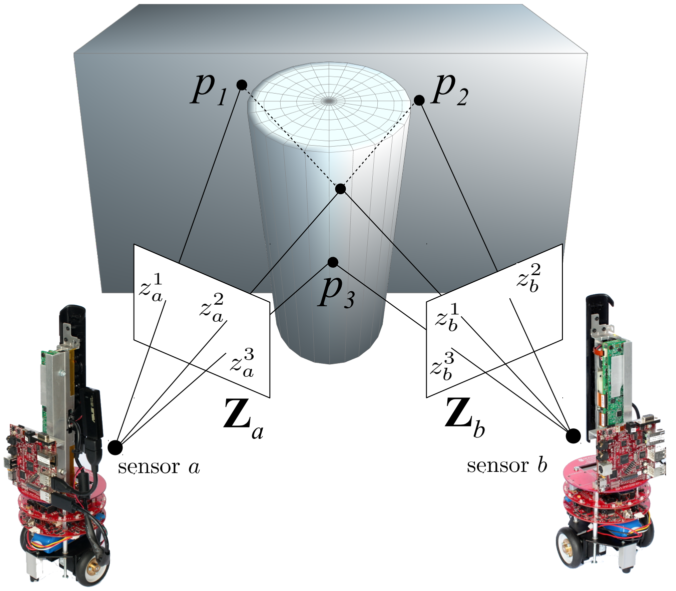


Fig. 1. A scene with occlusion: $\mathbf{Z}_a$ and $\mathbf{Z}_b$ denote a pair of depth images. p_3 is a world point which can be seen by both sensors a and b. p_1 and p_2 are world points which can only be seen by either sensor a or sensor b.

$$[\frac{i^k - i_c}{f_x}, \frac{j^k - j_c}{f_y}, 1, 1/z^k]_b^T = \mathbf{M}_{ab}[\frac{i^k - i_c}{f_x}, \frac{j^k - j_c}{f_y}, 1, 1/z^k]_a^T. \tag{3}$$

Conversely, if we can establish the correspondences between two depth images and put the corresponding pixel pairs in Eq. 3, the transformation matrix denoting the relative pose between two RGB-D sensors can be determined. And all the pixels in $\mathbf{Z}_a$ can be warped to generate a virtual depth image which matches $\mathbf{Z}_b$.

However, when there is occlusion in the scene (see Fig. 1), some world points may only be seen by sensor a and cannot be seen by sensor b. Therefore, the pixels representing these points in $\mathbf{Z}_a$ are not able to find their correct corresponding pixels in $\mathbf{Z}_b$. If the incorrect correspondences are established, an virtual depth image which cannot match $\mathbf{Z}_b$ will be generated. And a wrong transformation matrix is provided according to Eq. 3.

III. Proposed Work and Algorithm

In the proposed work, a maximum likelihood framework is presented to deal with the effect of occlusion. The proposed algorithm is inspired by the working principle of the beam-based sensor model [15] and is incorporated into our ICP solver as a robust weighting function. We will first review the beam-based sensor model which distinguishes the points on occlusion from the scene using a maximum likelihood framework. And then we present an approach for eliminating the bias introduced by beam-based sensor model.

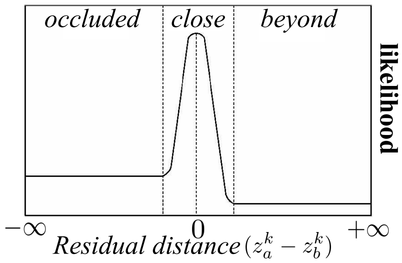


Fig. 2. Piecewise function used in the beam model.

A. Beam-based Sensor Model

Let $\mathcal{D}_a$ and $\mathcal{D}_b$ denote the depth measurements returned by the depth sensor a and b. Each set of the depth measurements is made up of N pixel elements where each pixel in the image contains the corresponding depth value, z_a^k, such that $\mathcal{D}_a = \{z_a^1, ..., z_a^N\}$. In this model, the depth information in $\mathcal{D}_b$ is treated as the expected surface, and the depth information in $\mathcal{D}_a$ is treated as the measurements. The relative motion which best aligns the measurements to the expected surface, described by a 6DoF motion matrix $\mathbf{M}$, can be estimated by formulating this as a maximum likelihood problem as,

$$\mathbf{M} = \arg\max_{\tilde{\mathbf{M}}} p(\mathcal{D}_a|\mathcal{D}_b, \tilde{\mathbf{M}}). \tag{4}$$

The conditional probability $p(\mathcal{D}_a|\mathcal{D}_b, \tilde{\mathbf{M}})$ can be approximated by the product of the individual measurement probabilities as

$$p(\mathcal{D}_a|\mathcal{D}_b, \tilde{\mathbf{M}}) = \prod_k p(z_a^k|\mathcal{D}_b, \tilde{\mathbf{M}}), \tag{5}$$

where $p(z_a^k|\mathcal{D}_b, \tilde{\mathbf{M}})$ can be modeled according to the beam model which describes the probability distribution of a measurement z_a^k to lie in front (occluding surface), close to, or beyond the surface given the expected measurements in the depth measurements $\mathcal{D}_b$ and the motion matrix $\mathbf{M}$. The beam model is illustrated in Fig. 2 and this can be represented using a piecewise function. There are three parts in this piecewise function:

- Case 1: when $z_a^k \ll z_b^k$: describes the probability of a depth measurement z_a^k to be an occluded surface. This is described by a uniformly distributed function.
- Case 2: when $z_a^k \approx z_b^k$: describes the probability of a depth measurement z_a^k to be closely aligned to its expected value. This is described by a Gaussian distribution centered at 0 with standard deviation σ_z for $z_a^k \approx z_b^k$
- Case 3: when $z_a^k \gg z_b^k$: describes the probability of a depth measurement z_a^k to lie beyond its expected value. This is described by a uniformly distributed function with very low probability.

The beam model when applied to motion estimation within a maximum likelihood framework explicitly deals with occlusion. However, as the piecewise model used to distinguish various points is unsymmetrical, the beam model creates a bias. It tends to push the sample points away from the camera that senses the surface so that they are more likely to lie behind it, which would produce less robust and inaccurate results.

B. Bidirectional Beam Model

To remove this bias introduced by bean model, we propose to use the beam model bidirectionally, and we name it bidirectional beam model. In a maximum likelihood framework, this can be formulated as

$$\mathbf{M}_{ab} = \arg\max_{\tilde{\mathbf{M}}}[p(\mathbf{Z}_a|\mathbf{Z}_b, \tilde{\mathbf{M}})p(\mathbf{Z}_b|\mathbf{Z}_a, \tilde{\mathbf{M}}^{-1})]. \tag{6}$$

$\mathbf{Z}_a$, $\mathbf{Z}_b$ are the depth images captured by sensor a and sensor b respectively. Similar to the unidirectional beam model, we can assume the conditional independence for each depth measurement such that

$$p(\mathbf{Z}_a|\mathbf{Z}_b, \tilde{\mathbf{M}}) = \prod_k p(z_a^k|\mathbf{Z}_b, \tilde{\mathbf{M}}), \tag{7}$$

$$p(\mathbf{Z}_b|\mathbf{Z}_a, \tilde{\mathbf{M}}^{-1}) = \prod_k p(z_b^k|\mathbf{Z}_a, \tilde{\mathbf{M}}^{-1}). \tag{8}$$

According to the piecewise function of beam model, an occlusion determined by $p(\mathbf{Z}_a|\mathbf{Z}_b, \tilde{\mathbf{M}})$ is treated as the measurement beyond the expected surface by $p(\mathbf{Z}_b|\mathbf{Z}_a, \tilde{\mathbf{M}}^{-1})$. When the first probability component is maximized as points are being pushed to the front of the reference surface, the second probability component will become smaller, which can prevent the transformation matrix $\mathbf{M}_{ab}$ being incorrectly estimated. Eq. 6 can only be maximized when the balance between the two probability components is reached. Eq. 6, can be converted into negative log likelihoods,

$$\mathbf{M}_{ab} = \arg\min_{\tilde{\mathbf{M}}} \sum_k [\log p(z_a^k|\mathbf{Z}_b, \tilde{\mathbf{M}}) + \log p(z_b^k|\mathbf{Z}_a, \tilde{\mathbf{M}}^{-1})]. \tag{9}$$

To estimate the 6DoF motion that best aligns a pair of depth images in this form would require partial derivatives of $\tilde{\mathbf{M}}$ with respect to the 6 motion parameters to be derived. This is not a trivial task and even if this is achievable, it is not computationally efficient. This forms the basis of our motivation to incorporate the bidirectional beam model as a robust weighting function into the ICP algorithm.

C. Motion Estimation Using ICP with Bidirectional Beam Model

We now describe how the bidirectional beam model is incorporated into the ICP algorithm. We approach this problem by using an asymmetric weighting function in the least squares component of our ICP solver. As reported in [16], different weighting functions lead to various probability distributions. For a weighting function $w(x)$, the probability density function is expressed as,

$$p(x) = \frac{1}{k}\exp\left(-\lambda \int_0^x x' w(x')\mathrm{d}x'\right), \tag{10}$$

in which $k = \int_{-\infty}^{+\infty} \exp\left(-\lambda \int\limits_{0}^{x} x' w(x') \mathrm{d}x'\right)$ is the normalization factor. To achieve the probabilistic model for beam model in Fig. 2, we find a piecewise weighting function as follows,

$$w(z) = \begin{cases} c/[c + (z^* - z)] & \text{if } z \leq z^* \\ c/[c + (z^* - z)^2] & \text{if } z > z^* \end{cases}, \tag{11}$$

where z^* is the expected depth value, and z is the measured value. c is the mean of deviation between expected depth values and measured depth values.

As shown in Fig. 2, the likelihood of one correspondence is directly related to the residual distance between the measurement and the expected surface. Therefore, the maximum likelihood framework of the bidirectional beam model can be converted and solved as a novel least squares approach which operates in a bidirectional way with the weighting function presented above. Suppose that correspondences between $N = N_a + N_b$ pairs of points from two depth images $\mathbf{Z}_a$ and $\mathbf{Z}_b$ are established, we can then estimate the transformation matrix $\mathbf{M}_{ab}$ by minimizing

$$\mathcal{C} = \sum_{k=1}^{N_a} [w_{k,a} (\tilde{\mathbf{M}} \mathbf{p}_a^k - \mathbf{p}_b^{k*}) \cdot \vec{n}_{k,b}]^2 + \sum_{k=1}^{N_b} [w_{k,b} (\tilde{\mathbf{M}}^{-1} \mathbf{p}_b^k - \mathbf{p}_a^{k*}) \cdot \vec{n}_{k,a}]^2, \tag{12}$$

in which $\mathbf{p}_a^k$ and $\mathbf{p}_b^k$ are the sampled points on different depth images, $\mathbf{p}_b^{k*}$ and $\mathbf{p}_a^{k*}$ are their corresponding points respectively. $\vec{n}_{k,b}$,$\vec{n}_{k,a}$ are the surface normals at corresponding points $\mathbf{p}_b^{k*}$ and $\mathbf{p}_a^{k*}$ respectively. $w_{k,a}$ and $w_{k,b}$ are weight parameters for various correspondences established in different directions. The cost function in Eq. 12 includes two parts: (a) the sum of squared distances in the forward direction from depth images $\mathbf{Z}_a$ to $\mathbf{Z}_b$, (b) the sum of square distance in the backward direction from $\mathbf{Z}_b$ to $\mathbf{Z}_a$. Eq. 12 can be minimized by reweighting the least squares operation in an ICP framework. Our algorithm is outlined in Algorithm 1.

In each iteration of this coarse-to-fine algorithm, the corresponding pixel pairs between two depth images are established. And the transformation matrix which can warp the pixels from one image to their corresponding pixels' coordinates is updated. Therefore, the pixels in one depth image can be warped to generate a virtual depth image matching the other depth image iteratively. Once the algorithm converges, the registration is accomplished and the transformation matrix describing the relative pose between two sensors is determined.

D. Distributing the Algorithm to Two RGB-D Sensors

In reality each sensor only has its own captured depth frames. In order to accomplish the centralized working principle of the algorithm described above, we distribute the tasks to two sensors.

Considering the limited bandwidth of the network, instead of transmitting a complete depth image from one sensor to another, each sensor only transmits a number of sampled points to the other sensor. For example, at each iteration, after sensor b receives the sampled point set, P_a, from sensor a, sensor b will find the corresponding point set, P_b^*, on its captured depth frame $\mathbf{Z}_b$. The first component in Eq. 12 will be derived. The information representing the first component will be sent with the sampled point set, P_b, from sensor b to sensor a. At sensor a, P_b's corresponding point set, P_a^*, will be determined. And the second component in Eq. 12 will be derived. Thereby, sensor a will acquire the information of both first and second component in Eq. 12. And the motion parameters can be determined.

Algorithm 1 The concept of ICP with bidirectional beam model (centralized)

```
Capture a depth frame, $\mathbf{Z}_a$, on sensor $a$, and capture a depth frame, $\mathbf{Z}_b$, on sensor $b$.
Initialize the transformation matrix, $\mathbf{M}_{ab}$, as the identity transformation.
procedure REPEAT UNTIL CONVERGENCE
    Update depth frame $\mathbf{Z}_a$ according to transformation matrix.
    Randomly sample $N_a$ points from $\mathbf{Z}_a$ to form set $P_a$, $P_a = \{\mathbf{p}_a^k \in \mathbf{Z}_a, k = 1, \ldots, N_a\}$,
    Randomly sample $N_b$ points from $\mathbf{Z}_b$ to form set $P_b$, $P_b = \{\mathbf{p}_b^k \in \mathbf{Z}_b, k = 1, \ldots, N_b\}$.
    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
    Apply the weight function in Eq. 11 bidirectionally, $P_a \mapsto P_b^*$, $P_b \mapsto P_a^*$
    Compute and update transformation matrix based on current bidirectionally weighted correspondences
end procedure
```

These procedures will be performed in each iteration. The transformation matrix describing the relative pose between two sensors will be obtained by sensor a until the algorithm converges. And sensor a will send the inverse transformation matrix to sensor b. Then both sensors will obtain the information of the other sensor's location and orientation. The distributed process is illustrated in Fig. 3.

IV. Experiment Results

In order to justify the proposed algorithm towards relative pose estimation between multiple RGB-D sensors, we have conducted extensive tests to evaluate the performance. We have implemented our algorithm (ICP-BD) in C++ using the libCVD and OpenKinect libraries on a laptop with an Intel i7 M620 processor and our mobile visual sensor network testbed to evaluate its fast-processing as well as robust performance. To verify the superiority of our algorithm in relative pose estimation, we compared it with the benchmark ICP algorithm

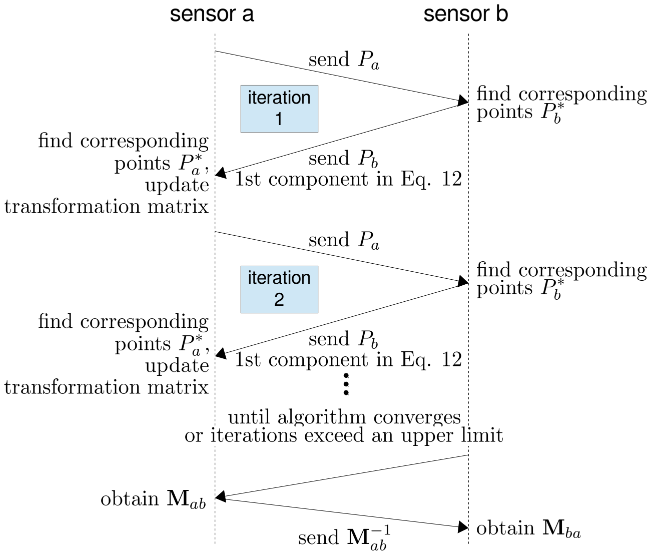

sensor a
sensor b
send P_a
iteration 1
find corresponding points P_b^*
find corresponding points P_a^*, update transformation matrix
send P_b
1st component in Eq. 12
send P_a
iteration 2
find corresponding points P_b^*
find corresponding points P_a^*, update transformation matrix
send P_b
1st component in Eq. 12
until algorithm converges or iterations exceed an upper limit
obtain $\mathbf{M}_{ab}$
send $\mathbf{M}_{ab}^{-1}$
obtain $\mathbf{M}_{ba}$

Fig. 3. Distributing the tasks to two mobile sensors.

[14] and ICP in inverse depth coordinates (ICP-IVD) [5] using point-to-plane error metric.

A. Dataset Simulations

This set of experiments is conducted on the laptop by using the datasets *Cabinet, Large cabinet, Plant, and Structure-no-texture* provided in [17]. Each dataset is a sequence of Kinect video frames capturing one scene from different angles of view. In order to simulate situations including different amounts of occlusion between two sensors' views, we extracted 4 new sequences from each dataset by taking one frame out of every 5, 10, 20, and 30 frames. For each trial we treat two consecutive frames in the new sequence as the depth images captured by two separated sensors. We deem a trial to be successful if the error between the estimated pose and ground truth pose is within 10 centimeters in translation. The percentages of successful relative pose estimation for different algorithms are presented in Table I. Table I clearly indicates that

- As the frame is sampled at an incremental interval, each algorithm's successful percentage decreases. When frame interval is greater than 10, more occlusion and differences exist between two sensors' views. As proposed ICP-BD reports higher successful estimation percentages, it outperforms other algorithms in environments with heavy occlusion.
- When the frame interval is 5, three algorithms have similar performances. Therefore, all of them can be used to handle small motion in the presence with minus occlusion.
- When the frame interval is 30, the occlusion between two views is too heavy and the two consecutive depth images are largely different from each other. As a result of this, the performance of three algorithms drops down significantly.

TABLE I
PERCENTAGE OF SUCCESSFUL RELATIVE POSE ESTIMATION IN VARIOUS DATASETS

Dataset	Frame interval	ICP	ICP-IVD	ICP-BD
Cabinet	5	1.00	1.00	1.00
	10	0.98	1.00	1.00
	20	0.81	0.90	0.98
	30	0.43	0.43	0.55
Large cabinet	5	0.99	0.99	0.99
	10	0.89	0.92	0.95
	20	0.61	0.69	0.81
	30	0.22	0.31	0.37
Plant	5	1.00	0.99	1.00
	10	0.77	0.82	0.91
	20	0.59	0.75	0.82
	30	0.31	0.40	0.47
Structure no texture	5	1.00	1.00	1.00
	10	0.94	0.97	0.97
	20	0.71	0.81	0.90
	30	0.38	0.40	0.53

An intuitive example of successful trail using ICP-BD in dataset "Cabinet" is illustrated in Fig. 4. Furthermore, through adjusting the number of sample points on the depth frames, our proposed algorithm can process up to 30Hz while still hold the estimation accuracy on a standard laptop without GPU implementation, which is faster than the other ICP variants.

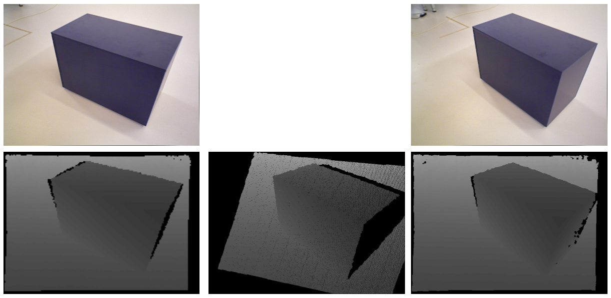
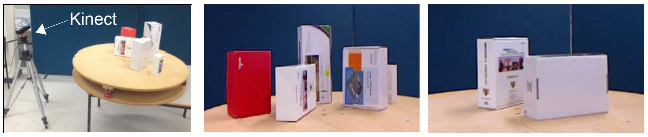


Fig. 4. A successful trail in dataset "Cabinet" with a frame interval at 20. The captured depth images and their corresponding color images are displayed on left and right side. The depth image in the middle is warped from the left depth image to register to the right depth image.

B. Turntable Simulations

In order to control the occlusion ratio in two sensors' views precisely, we generated our own datasets to evaluate the performance of our proposed algorithm for heavily occluded situations. A turntable was used to obtain ground truth. Several objects were placed on the center of the turntable, and the images were captured by a Kinect that was mounted on a tripod. We generated our dataset from the two scenes illustrated in Fig. 5 and in each scene the turntable was rotated clockwise incrementally at an interval of 5° up to 90°.

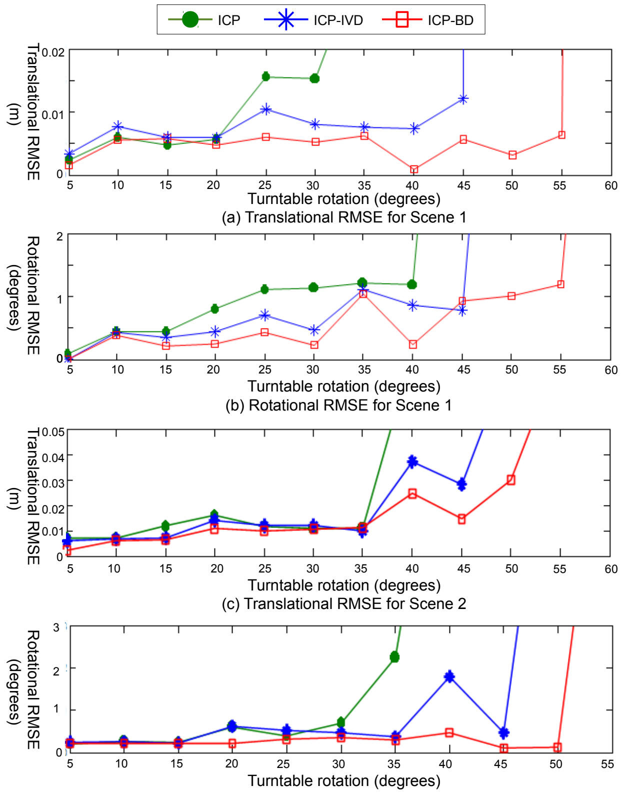
The main difference of this simulation in comparison to the previous set is that the ground truth is known exactly at every 5° interval which is precisely controlled. Whereas in the previous simulation, the motion between two depth images is quite random. And in this simulation, we can determine



(a) Experimental setup (b) Scene 1 (c) Scene 2

Fig. 5. Experimental setup and two scenes with different occlusion.

when the algorithms fail to provide the accurate estimation. The performance of the different algorithms is evaluated based on the rotational and translational RMS error, as illustrated in Fig. 6.



(a) Translational RMSE for Scene 1

(b) Rotational RMSE for Scene 1

(c) Translational RMSE for Scene 2

(d) Rotational RMSE for Scene 2

Fig. 6. Rotational RMSE and translational RMSE for variant algorithms. Turntable rotation indicates turntable rotation interval between two frames in degrees.

The graphs in Fig. 6 clearly indicate that

- When angular interval is greater than 15 degrees, more occlusion exist between two sensors' views. And the proposed algorithm outperforms other variants as it reports much lower translational and rotational RMSE.
- Standard ICP has the poorest performance across the experiments. ICP-IVD can provide similar accuracy in pose estimation before it diverges. However, as the scene becomes more occluded as the turntable is being rotated, ICP-IVD would fail to converge sooner than our proposed method.
- For small angular interval, the relative accuracy between the three algorithms are small. Therefore, all of them can be used to handle small motion in the presence with minus occlusion.

C. Mobile Visual Sensor Network Testbed Experiments

In this set of the experiments, we implemented the proposed algorithm on our mobile visual sensor network testbed. This testbed consists of multiple mobile RGB-D sensors named "eyeBug" (see Fig. 7) [18].

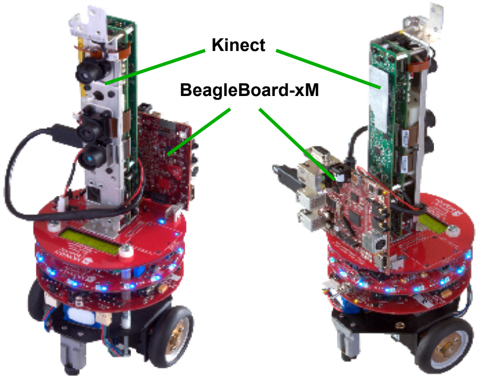
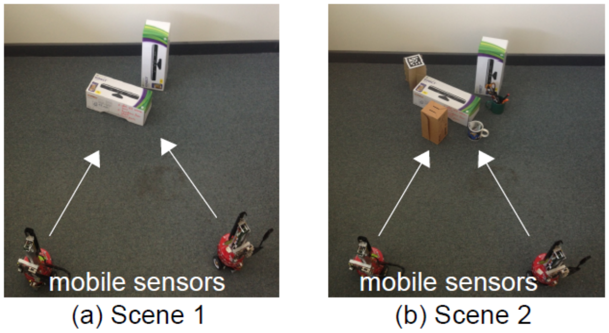


Fig. 7. eyeBug– the mobile RGB-D sensor in our mobile visual sensor network platform.

On each mobile sensor, a Microsoft Kinect is mounted at the center and a BeagleBoard-xM single-board computer is built at the rear. Equipped with an Atmel Cortex processor, a USB hub, and a HDMI video output, the BeagleBoard-xM can run an ARM-processor-optimized Linux kernel mounted on a micro-SD card and act as if it were a desktop computer. Despite the BeagleBoard only running at a maximum 800 MHz processor cycles, it can easily interface with both the Kinect sensor and the WiFi dongle through its USB ports, facilitating the streaming system. In this experiment, instead of using the whole network, we only used two mobile sensors to test the algorithm.

We generated two different scenes illustrated in Fig. 8. In the following experiments, we performed 50 trails per scene. And in each trail, we placed two eyeBugs at different locations while kept them looking at the scene from different angles. Two eyeBugs are able to communicate with each other directly through wireless channel. And the proposed algorithm was implemented on each eyeBug and worked in a distributed manner. As we did not have the precise ground truth information of each eyeBug's location and orientation in this set of experiments, we programed the first eyeBug to keep stable and programed the second eyeBug to move to the first eyeBug's position after it obtained the relative pose information. We deem a trail to be successful if the second eyeBug can move to the place within the range of 10 centimeters to first eyeBug's position.

In Fig. 9 we present the frequency of successful trail that one eyeBug moves to the other eyeBug's position. When the amount of occlusion and clutters increases, our algorithm performs 10% to 16% better than ICP-IVD and far more better than the benchmark ICP. Due to the computational



(a) Scene 1

(b) Scene 2

Fig. 8. Two scenes with varying amount of occlusions and clutters.

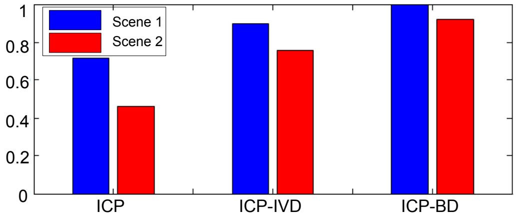


Fig. 9. Frequency of successful relative pose estimation

constraint of our mobile RGB-D sensor, the algorithm requires an average 1.21 second to provide the relative pose estimation.

V. Conclusion and Future Work

In this paper, we propose the first relative pose estimation algorithm for multiple RGB-D sensors. The proposed approach can operate in real time on depth images captured by the Microsoft Kinect. And it has been implemented on a mobile visual sensor network to enable the sensor to obtain the location and orientation information of the other sensor in the network.

The main contribution of this paper is the development of a novel maximum likelihood model named bidirectional beam model which can deal with the effect of occlusion in the views of different sensors. And we incorporated this model into the ICP framework in order to determine the motion parameters. Different from the existing works aligning 3D point clouds to estimation inter-frame motion, the proposed algorithm directly registers two depth images. We conducted three sets of experiments to evaluate the accuracy and robustness of our proposed algorithm in environments with various amount of occlusion. Results of the experiments indicate that the proposed ICP with bidirectional beam model is robust and accurate in different environments for relative pose estimation.

Further research will concentrate on enhancing the scalability of the algorithm to estimate the relative pose among a large number of RGB-D sensors. And this algorithm will be used as the initialization for Cooperative Localization (CL) in the near future. Moreover, our algorithm is constrained to operate in static scenes and it is not able to deal with dynamic objects in the environment. As future work, we intend to address such limitation as well.

References

[1] W. H. Stanislava Soro, “A survey of visual sensor networks,” *Advances in Multimedia*, vol. 2009, 2009.

[2] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli, *Depth mapping using projected patterns*, 2008.

[3] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, “A review of recent range image registration methods with accuracy evaluation,” *Image and Vision Computing*, vol. 25, pp. 578–596, 2007.

[4] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *In Proc. UIST*, 2011, pp. 559–568.

[5] W. L. D. Lui, T. J. J. Tang, T. Drummond, and W. H. Li, “Robust egomotion estimation using icp in inverse depth coordinates,” in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1671 –1678.

[6] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, “Tracking a depth camera: Parameter exploration for fast icp,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 3824–3829.

[7] Y. Zou, W. Chen, X. Wu, and Z. Liu, “Indoor localization and 3d scene reconstruction for mobile robots using the microsoft kinect sensor,” in *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, July, pp. 1182–1187.

[8] H. Wang, W. Mou, M. H. Ly, M. Lau, G. Seet, and D. Wang, “Mobile robot ego motion estimation using ransac-based ceiling vision,” in *Control and Decision Conference (CCDC), 2012 24th Chinese*, 2012, pp. 1939–1943.

[9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” in *The International Journal of Robotics Research*, 2012.

[10] C. K. Schindhelm, “Evaluating slam approaches for microsoft kinect,” in *ICWMC 2012, The Eighth International Conference on Wireless and Mobile Communications*, 2012.

[11] Y. Takeda, N. Aoyama, T. Tanaami, S. Mizumi, and H. Kamata, “Study on the indoor slam using kinect,” in *Advanced Methods, Techniques, and Applications in Modeling and Simulation*, ser. Proceedings in Information and Communications Technology, 2012, vol. 4, pp. 217–225.

[12] W.-T. Huang, C.-L. Tsai, and H.-Y. Lin, “Mobile robot localization using ceiling landmarks and images captured from an rgb-d camera,” in *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, July, pp. 855–860.

[13] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the rgb-d slam system,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 1691–1696.

[14] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[15] M. Krainin, K. Konolige, and D. Fox, “Exploiting segmentation for robust 3d object matching,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 4399–4405.

[16] P. W. Holland and R. E. Welsch, “Robust regression using iteratively reweighted least-squares,” *Communications in Statistics - Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[18] N. D’Ademo, W. L. D. Lui, A. Sekercioglu, and T. Drummond, “ebug - an open robotics platform for teaching and research,” in *Robotics and Automation, 2011 Australasian Conference on*, 2011.