ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

An efficient 3-D environment scanning method

Kin Hong Wong Ho Chuen Kam Ying Kin Yu* Sheung Lai Lo
Department of Computer Science and Engineering
The Chinese University of Hong Kong
*Hong Kong
{khwong, hckam}@cse.cuhk.edu.hk, {ykyu.hk, lester2345}@gmail.com

Kwan Pang Tsui
Department of Mechanical and Automation Engieering
The Chinese University of Hong Kong
Hong Kong
warrentsui@outlook.com

Hing Tuen Yau
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong
billyauhk@gmail.com

ABSTRACT

In this paper, we discuss an idea of a system that can capture the 3-D model of a large area using only one single Kinect 3-D range sensor plus a stationary master camera. In operation, the Kinect is placed at different key positions to capture the local 3-D models, while a stationary master camera is situated behind the Kinect to find the current pose of the Kinect range sensor. Traditionally, a large scene can be scanned by moving the Kinect sensor across the whole area. Then the models obtained can be combined using motion capturing and pattern matching methods. However, the accuracy deteriorates when the area is too large or the environment does not provide enough features for registration. In our proposal, we place the Kinect at different key positions to obtain a number of local models. A dual-face checkerboard is placed on the top of the Kinect sensor in a way that the pattern can be seen from both the front and rear sides but not blocking the view of the Kinect. The pose of the board and the Kinect is estimated by a pose estimation algorithm using the images captured by the master camera. Since the embedded RGB-camera in the Kinect cannot see the checkerboard, a method based on a mirror is devised to determine the relative pose between the board and the embedded RGB-camera. Finally, we can combine all the 3-D local models and the pose information obtained to build up the complete global model. Various parts of the idea have been tested. We plan to integrate all parts and build a complete system for building the 3D map of a shopping mall or a museum in the future.

Keywords

Rotation averaging; mirrors; camera calibration; virtual reality development

1 INTRODUCTION

Obtaining the 3-D model of a small area can be achieved by a low cost 3-D scanner such as the Kinect camera. There is also a huge demand on the 3-D digitization of larger environment for virtual reality or 3-D navigation applications. Currently, a popular method is to scan the scene by moving the sensor manually for a distance to obtain the model by the software called Kinfu [Pir11]. However, the known problem of this approach is that the result may deteriorate if the scanning region is too large. Moreover, in order to achieve a reasonably good result, the scanning process is best to be performed by experienced technicians who are able to handle the scanner steadily for a long time. This creates a problem in deployment and execution. In this paper, we report a simple but yet effective method to solve the problem.

The setup is illustrated in Figure 1. In the proposed system, we employ only one Kinect sensor. It is to be placed at different key positions at different times to cover a large target area. One extra static camera, called the master camera in the world coordinate system, is needed to determine the current pose of the Kinect sensor. Pose estimation is achieved by placing a dual-face checkerboard on the top of the Kinect sensor. A standard pose estimation method [CW05] is used to obtain the position and orientation of the

checkerboard with respect to the master camera.
Even the poses of the Kinect sensor can be determined by the method explained above, however, there is an extra problem. Since the checkerboard and the Kinect cannot be aligned at the same position, we need to determine this small pose difference. This gives a certain difficulty to the overall system. Firstly, the checkerboard has to be observable from behind the Kinect by the master camera. It cannot be placed in front of the Kinect to block its view. The solution is that we fix the checkerboard on top of the Kinect so it can be seen on both (front and rear) sides. The board has two faces and its images are the same on both sides. Hence it is called the dual-face checkerboard. Using this scheme, we need to determine the pose between the dual-face board and the Kinect. Traditional extrinsic parameter calibration methods cannot be employed because the Kinect camera cannot observe the checkerboard directly. To tackle the task under this special arrangement, we propose to make use of a mirror together with the method by [LKL+15] to solve the problem. The procedure is discussed in the Section 3.
With the equipment setup mentioned above, the 3-D reconstruction process can be carried out as follows: Firstly, we need to choose a number of key positions. So by combining the models obtained at these key positions, the whole area of the target 3-D space can be covered. Then, at each key position, the Kinect sensor is operated to obtain the local 3-D model accurately by some existing approaches [Pir11] hence the localized 3-D structure can be calculated. Since a checkerboard is attached to the top of the Kinect sensor, the master camera is used to obtain the 3-D pose of the Kinect in the master (or world) coordinate system. Finally, we can combine all these local results into the world coordinate frame and we get the global 3-D model of the large environment.
The major contributions of this work are:

- The setup is low cost; only a single Kinect sensor and a normal digital camera are needed.
- The result is accurate compared to the traditional handheld scanning method, especially if the area is large or lacking features in some areas.

Our paper is organized as follows. The background of the research is discussed in section 2. The theories used in this work are discussed in section 3. The experimental result is shown in section 4. Section 5 concludes the work.

2 BACKGROUND

2.1 Structure From Motion

Structure from Motion (SfM) is an important problem in the field of computer vision. It has been studied for more than a decade, for examples in the literature [JAP99], [Nis00], [HZ03], [FLP04], [YWC05], [YWC06] and [SSS08] etc.. The target is to find the 3-D model of an object from 2-D pictures. SfM is still an ongoing research topic. Most techniques rely on the correspondences between 3-D points of the object and its 2-D projection on the images. In the process, both the camera pose and the 3-D structure are computed. In order to find the correspondences among the input images, interest points are first located and extracted by the feature detectors like [HS88] and [Low04]. Features such as points, corners, edges and even planes can be tracked and extracted. Based on the features, the 3-D strcutrue can be estimated using bundle adjustment, for example [CW05].

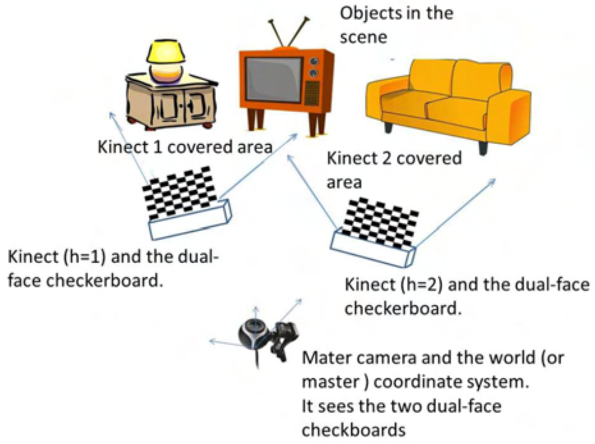

Figure 1: Overall setup - master camera and kinects

2.2 Reconstruction by Kinect

Nowadays, there are lots of range sensors available in the market. They are suitable for performing computer vision tasks such as 3-D reconstruction. The Microsoft Kinect [Zha12] is a popular consumer grade range sensor for games and digital entertainment. Since it is inexpensive and its precision is high enough, it has been widely adopted by researchers in the field [CPF+12], [SHBS11], [TJRF13]. The underlying technology of a 3-D scanner is the use of Laser or Infra-red beam. The output is a point cloud consisting of the depth information that represents the structure of the target object. Triangles, polygon planes or curvature structures, can be constructed based on the obtained points through the process of surface reconstruction. One of the most straight-forward method is Triangle Strip devised by Zhang et. al. [ZZC+13]. It tackles the task by filling up the gaps along the two neighboring rows of horizontal pixels with multiple consecutive triangle plane surfaces. Triangle Strip offers an efficient way to reconstruct the object surface but the output is of relatively low quality. Another algorithm having a better performance is Poisson Surface Reconstruction [KBH06]. It inserts hundreds of extra pixels in between the direct neighboring points. The insertion of pixel points is based on the original neighboring point curvature and tangential level to build up smooth object surfaces. In this way, it can produce high quality 3-D models. The main disadvantage is its long compuatation time, thus not suitable for real-time processing. Besides, there are other traditional algorithms to reconstruct 3-D surfaces, such as Ball Pivoting [BMR+99] and Power Crust [ACK01].

2.3 Long sequence reconstruction

There are some studies on the 3-D reconstruction of a large indoor environment. Kinect Fusion [NIH+11] is one of the most popular system for real-time surface mapping and tracking [IKH+11]. Depth information generated from the Kinect is used for pre-processing. After getting surface vertices and normal maps, a pose estimation algorithm is executed to calculate the camera pose in the scene. With Iterative Closest Point (ICP) procedure [BM92], a 6-DOF motion of the sensor is found by aligning the points in the current frame with respect to the previous ones. There is a major limitation of the Kinect Fusion system. As it relies on the ICP procedure for point matching, the model scanning process fails if it is applied to a plane with few features or a shiny surface [CKN+14].

3 THEORY

3.1 Over view of the system

The idea of our approach is illustrated in Figure 2. First we need to calibrate the pose between the dual-face checkerboard and the Kinect. It cannot be achieved by the usual camera calibration methods because the checkboard is not observable from the Kinect. To recover the pose between the Kinect and the checkerboard, we can employ a mirror-based technique similar to the one described in [LKL+15]. After this procedure, the rotation and translation between the dual-face checkerboard and the Kinect sensor can be found. Then we can use our setup to find the complete 3-D model of the environment. Since the dual-face checkerboard is very thin and the patterns on both sides are the same, the model and the image of the dual-side checkerboard are the same no matter which side you are looking at it. The procedure for 3-D reconstruction of the environment is described as follows. We first place the Kinect at a key position $h = 0$. The camera obtains the image of the dual-face checkerboard at the back of the Kinect. Using the pose estimation algorithm, the pose parameters (R_h, T_h) can be obtained. At the same time, the local 3-D model $M^{(h)}$ is also captured by the Kinect. We repeat the process for all h. After all models $M^{(h)}$ are obtained, we can put them into the coordinate system of the master camera (or world coordinate frame). In the following sessions, we will describe the details of (1) the pose estimation method between the master camera and the dual-face checkerboard and (2) a mirror-based pose determination approach to find the pose of the dual-face checkerboard relative to the Kinect.

3.2 Pose estimation between the camera and the dual-face checkerboard

Pose estimation is to determine the pose ($R = 3 \times 3$ rotation matrix and $T = 3 \times 1$ translation vector) of a rigid body when the 3-D model feature points $M_{i=1,2,...I}$ are given, where I is 3 or above and each 3-D feature M_i is a 3×1 vector. The method is discussed in [CW05] and is a well-known approach using the Gauss-Newton least-squares scheme.

Input: Models $M_{i=1,2,...N}$ with N 3-D feature points at rotation R=I_3, translation $T = [0,0,0]^T$; N 2-D image points $x_{i=1,2,...N}$ of the 3-D features at time t
Output: Pose $= \theta = [\phi_x, \phi_y, \phi_z, T_x, T_y, T_z]^T$ where $[\phi_x, \phi_y, \phi_z]$ = rotation angles, and $[T_x, T_y, T_z]$ = Translations
$x_i = p(M_i, \theta)$, where $p(M_i, \theta)$ is the projection of model M_i of feature i with pose θ to image x_i
Loop until error is small or too many times

```
Initialse θ_{k=0} and find p(M, θ_{k=0})
for (k = 1; k < K; k++) do
    Find image error e_i ← ||p(M_i, θ_k) − p(M_i, θ_{k−1})||
    E_k ← [e_1, e_2, ..., e_N]^T
    and Jacobian J ← ∂E_k/∂θ
    Δθ_k ← J^{−1} * E_k
    Break if Δθ is small enough
    θ_{k+1} ← θ_k + Δθ_k
end for
Return θ_k
```

3.3 Pose computation between the dual-face checkerboard and the Kinect

Since the dual-face checkerboard is required to determine the pose of the Kinect with respect to the master camera, there is a need to find out the relative pose (rotation=R_b, translation=t_b) between the dual-face checkerboard and the Kinect because they cannot be aligned perfectly. The idea is illustrated in Figure 3. This can be achieved using a mirror and the method is shown in Figure 4. Users are required to perform this procedure just once since only one Kinect sensor is used in the scanning operation. There are two steps in this procedure.

ISSN 2464-4617 (print) WSCG 2016 - 24th Conference on Computer Graphics, Visualization and Computer Vision 2016
ISSN 2464-4625 (CD-ROM)

3.3.1 Step 1: Pose estimation through a mirror

In this step, the Kinect and the checkerboard are stationary. The user places the mirror at n different positions. At each mirror position, the user takes a picture of the dual-face checkerboard through the mirror using the Kinect-RGB-camera. After this procedure, we have n pictures for pose estimation. For examples, the camera calibration toolbox [Bou11] or the pose estimation algorithm [CW05] can be applied to determine the pose of the checkerboard relative to the master camera. Then $R_{i=1,2,..n}$ rotations are obtained. However, please be noted that the rotations $R_{i=1,2,..n}$ obtained through capturing the checkerboard through the mirror are needed to be converted back to the corresponding improper rotations $\tilde{R}_{i=1,2,..n}$ using the formulas (equation 5) found in [LKL+15]. For example, an easy test to see if a rotation is improper or not is to see if $det(R) = -1$.

3.3.2 Step 2: Weiszfeld algorithm

Assuming the rotation between the Kinect and the dual-face checkerboard is R_b, the Weiszfeld algorithm [HAT11] is able to find this from $\tilde{R}_{i=1,2,..n}$ obtained in the above step. The procedure is shown in Algorithm 1.

Algorithm 1 Weiszfeld algorithm to find R_b

Input: $\tilde{R}_{i=1,2,..,N}, R_{init}$
Output: R_b
$R_{b(k=0)} = R_{b(init)}$
Define: $V = eig\_vector = V, D = eigen\_value$

1: **repeat**
2: $[V,D] \leftarrow eig(R^T_{b(k)}\tilde{R}_i)$
3: Select index c such that $D(c)$ == -1
4: $n_i \leftarrow V(:,c)$
5: $w \leftarrow log(R^T_{b(k)}\tilde{R}_i(I - 2n_in_i^T))$
6: $\delta \leftarrow \frac{\sum_{i=1}^{i=n}\frac{w}{\|w\|}}{\sum_{i=1}^{i=n}(1/\|w\|)}$
7: $R_{b(k+1)} \leftarrow exp(\delta)R_{b(k)}$
8: **until** diff. between $R_{b(k+1)}$ and $R_{b(k)}$ is very small
9: **return** $R_b \leftarrow R_{b(k)}$

The above algorithm is based on the method proposed by [LKL+15] and [HTDL13]. It is a rotation averaging scheme to find the optimal rotation. In the algorithm, the inputs are the rotations collected from the mirror images of the checkerboard. Rotation averaging has been studied in [HAT11], [LKL+15], [HTDL13], [KIFP08], [Huy09], [CG13] and [HTDL13]. Since the board is not directly observed by the camera but only its reflected image, the rotation obtained $\tilde{R}_i$ is required to be transformed back to the normal view by the formulation $R_b^T\tilde{R}_i(I - 2n_in_i^T)$. It is computed by steps 2 to 5

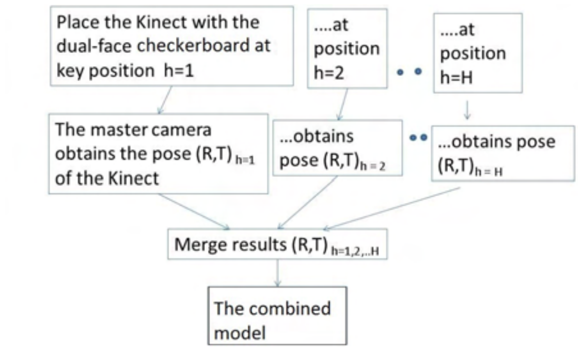

Figure 2: Overview of the approach

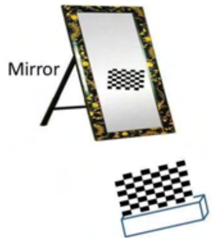

Figure 3: The Kinect camera can see the checkerboard through the mirror

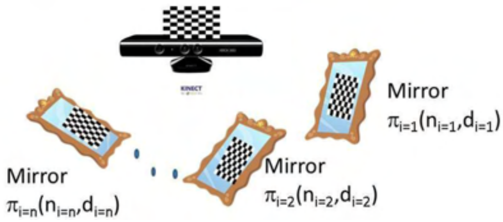

Figure 4: Calibration of the dual-face checkerboard and camera by repositioning the mirror

of Algorithm 1 and the method is described in [LKL+15] . Since the relative rotation of the pose R_b between the board and the camera is fixed and unchanged, the only change is the mirror orientation which will affect $\tilde{R}_i$. If we have enough samples of $\tilde{R}_i$, we can find R_b using an iterative scheme. So we need to identify a metric for evaluating the similarity between two rotations. A recent formulation is to use the metric proposed by Huynh [Huy09]. Together with the Weisfield algorithm, this method [LKL+15] can find the pose between the board and the camera efficiently and accurately. After R_b of the pose is found, the translation t_b can also be found by a simple linear formula using Equation(13) of [LKL+15].

4 EXPERIMENTS

4.1 Simulation for rotation averaging

The rotation averaging algorithm is used to find the rotation component of the pose between the dual-face checkerboard and the Kinect camera. We have carried out a simulation test to evaluate the performance of the rotation averaging method. The test was implemented in MATLAB 7.11 on a desktop computer.

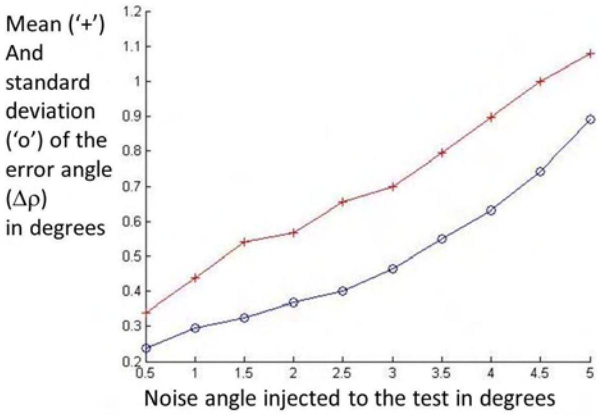

Figure 5: Simulation test result: The red +_line is mean error angle $\Delta\rho$ in degrees. The blue o_line is standard deviation of error angle $\Delta\rho$ in degrees.

In each test, we created a certain ground-truth rotation $R_{b\_ground\_truth}$ of the pose between the dual-face checkerboard and the Kinect-camera. This rotation was used to create 15 mirrored rotations $\tilde{R}_i$ based on the mirror reflection formula in Equation 4 in [LKL+15]. Noise in terms of rotation angles was injected into each of the rotation axis of input rotation $\tilde{R}_i$ with a standard deviation of from 0.5 to 5 degrees. Then we used the rotation averaging algorithm in Algorithm 1 to find the rotation matrix R_b. 1000 tests were carried out for each level of noise injected. To show and analyze the performance of the system, we plot the mean and standard deviation of the error angle $\Delta\rho$ against noise injected in Figure 5 .The error angle $\Delta\rho$ is the angle of the axis-angle representation of ΔR_{error}, where $\Delta R_{error} = R^T_{b\_found} * R_{b\_ground\_truth}$. The mean and standard deviation of the results are shown in Figure 5. As we can see from Figure 5, the error angle $\Delta\rho$ is still small even under noisy conditions. It shows that the rotation averaging method can compute the rotation of the pose accurately.

4.2 Mirror simulation toolbox

To let the readers further investigate into the mathematical properties of a mirror and the process of virtual object creation, we have developed a mirror simulation toolbox based on the formulas in [RBN10]. In the simulation test, we can create 3-D model points and a planar mirror in arbitrary positions. Then the corresponding virtual points and images can be formed and displayed. The 2-D virtual image points are captured by the real camera (in the simulation) and are then passed to a pose estimation algorithm [CW05], [Bou00] to find the pose of the virtual object with respect to the real camera. It is noted that the pose found should contain the improper rotation matrix since it is calculated from the virtual object. However, the camera does not know whether

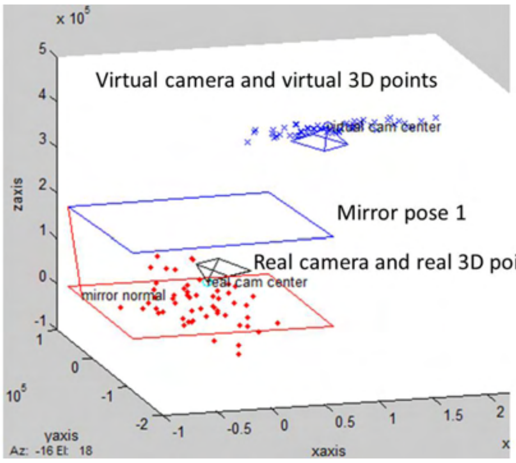

Figure 6: The First Sample Case of Our Mirror Simulation

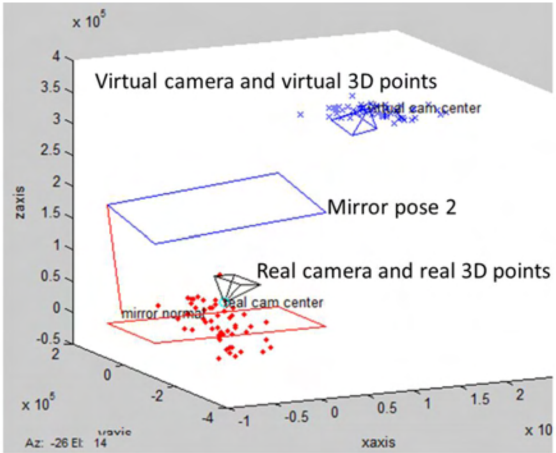

Figure 7: The Second Sample Case of Our Mirror Simulation

the object is a real one or from the mirror. So we need to convert the rotation (from proper to improper rotation) using the function described by Equation 5 in [LKL+15]. R_i is the converted rotation and is used in Algorithm 3.2. In this way, mirrors at different positions can be generated, resulting in a set of images of the virtual object. R_i can be calculated from these images using a pose estimation algorithm. If we can have enough R_i for $i \leq 3$, Algorithm 3.2 can be applied to find the rotation R_b between the real camera and the object. Screen shots of the toolbox are shown in Figures 6 and 7 . The toolbox visualizes the mirroring process. Files related to this research can be found at
https://appsrv.cse.cuhk.edu.hk/~khwong/www2/conference/2016/WSCG2016/WSCG2016.html

4.3 Real image experiment

We tested the proposed idea shown in Figure 1 using two Kinects at two different key positions, which are about 60 cm apart. The pose between the two Kinects are then calibrated as described in Section 3.

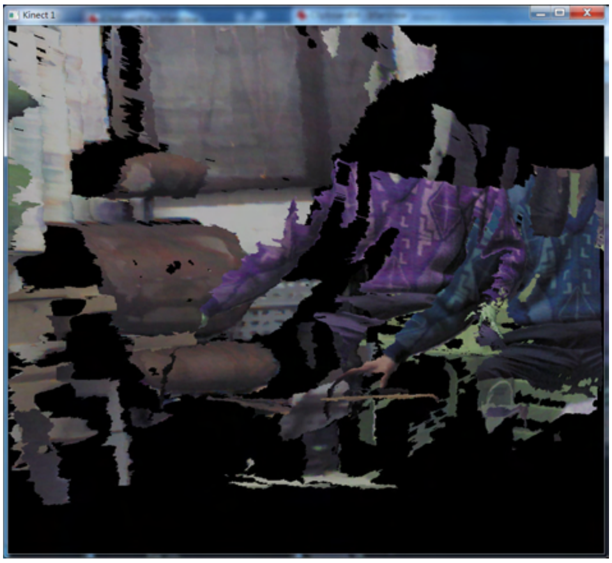

Figure 8: The 3-D point clouds before merging and they are separated

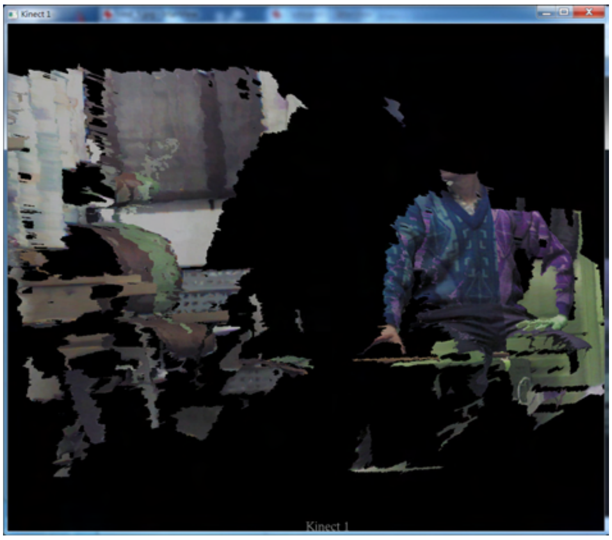

Figure 9: The 3-D point cloud combined using the computed pose parameters

The 3-D point clouds captured by the two Kinects are merged to generate a large 3-D model with the pose computed. The results before and after merging are shown in Figure 8 and Figure 9, respectively. It is demonstrated that the idea is to be feasible. Our next plan is to build a complete system to make it become a working system.

4.4 Comparison with an existing method

Our method is compared to an existing method called KinFu [Pir11] in terms of stability and usability. KinFu [Pir11] is a popular public domain software. It can be used to scan a large indoor area. The main problem of using Kinfu is that the operator is required to move the Kinect sensor very slowly. In our experience, the translation and rotation motion of the Kinect should be smaller than 0.5 meters or 10 degrees per second during operation, respectively. As our method requires the Kinect to be placed at a few stationary positions in the environment, it is convenient to use and the performance is relatively stable. Figure 10 shows the 3-D point cloud of a scene accquired by KinFU. It fails to capture the 3-D model of the environment correctly. It is because during scanning the sensor is rotated slightly for about 10 degrees, objects with vertical edges in the point cloud are mistakenly rotated. KinFu is unstable and not easy to be handled.

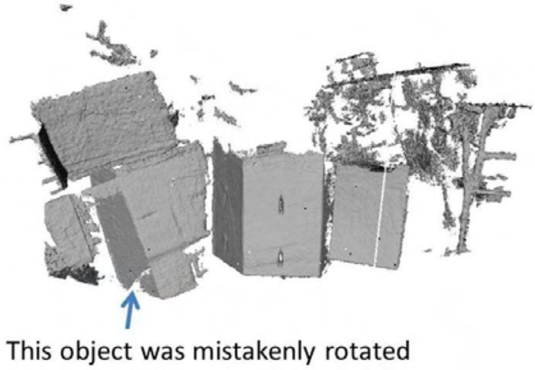

Figure 10: This is a case when Kinfu fails. An object in the scene was mistakenly rotated.

Method	Cost	Easy to use	Accuracy
Ours	Low, a common PC will do the job	Easy	High
KinFu	High, requires GPU	Not easy, needs the user to operate with care	High but can become inaccurate when handled incorrectly.

5 CONCLUSION

In this research, we have dicussed a system that can capture a large environment based on a two-level approach. At the first level, a Kinect sensor is placed at different positions of a large environment to obtain a number of local 3-D models. A dual-face checkerboard is attached to the Kinect so its pose relative to a stationary master camera can be estimated and recorded. Finally, all local models, each obtained by the same Kinect placed at different positions, are combined to become the complete wider view global model. We have also adopted a method using mirrors and rotation averaging to calibrate the pose between a camera and an object that the camera cannot observe directly; the object can only be seen by the camera through a mirror. The pose information is computed using a rotation averaging algorithm called the Weisfield Algorithm [HAT11]. A toolbox of the mirror image formation process and rotation averaging algorithm is also developed to help us use the mirror-based techniques. Unlike existing approaches based on motion tracking methods, our system is easy to deploy, relatively stable and low cost. A test is carried out to show that we can combine local models to become a larger model. The system can be used in many applications such as virtual and augmented reality. Although it is not a complete system yet, we are confident that the idea is feasible and we will work on it to build a complete system in future. For example, we will apply the proposed techniques to the reconstruction of a large shopping mall or a museum.

REFERENCES

[ACK01] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry*, 19(2):127–153, 2001.

[BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[BMR+99] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359, 1999.

[Bou00] Jean-Yves Bouguet. Matlab camera calibration toolbox. *http:\ www. vision. caltech. edu\ bouguetj\ calib doc*, 2000.

[Bou11] Jean-Yves Bouguet. Camera calibration toolbox for matlab, h ttp. *www. vision, caltech, edu*, 2011.

[CG13] Avhishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 521–528. IEEE, 2013.

[CKN+14] S Chumplue, T Kondo, I Nilkhamhang, P Bunnun, and M Sato. 3d room reconstruction using visual odometry guided kinectfusion with rgb-d camera. 2014.

[CPF+12] Ross A Clark, Yong-Hao Pua, Karine Fortin, Callan Ritchie, Kate E Webster, Linda Denehy, and Adam L Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & posture*, 36(3):372–377, 2012.

[CW05] Michael Ming-Yuen Chang and Kin Hong Wong. Model reconstruction and pose acquisition using extended lowe's method. *Multimedia, IEEE Transactions on*, 7(2):253–260, 2005.

[FLP04] Olivier Faugeras, Quang-Tuan Luong, and Theo Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004.

[HAT11] Richard Hartley, Khurrum Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3041–3048. IEEE, 2011.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. Citeseer, 1988.

[HTDL13] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[Huy09] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[JAP99] Tony Jebara, Ali Azarbayejani, and Alex Pentland. 3d structure from 2d motion. *Signal Processing Magazine, IEEE*, 16(3):66–84, 1999.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[KIFP08] Ram Krishan Kumar, Adrian Ilie, Jan-Michael Frahm, and Marc Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008.

[LKL+15] Gucan Long, Laurent Kneip, Xin Li, Xiaohu Zhang, and Qifeng Yu. Simplified mirror-based camera pose computation via rotation averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1247–1255, 2015.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[NIH+11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[Nis00] David Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Computer Vision-ECCV 2000*, pages 649–663. Springer, 2000.

[Pir11] Michele Pirovano. Kinfu–an open source implementation of kinect fusion+ case study: implementing a 3d scanner with pcl. *Project Assignment, 3D structure from visual motion, University of Milan*, 2011.

[RBN10] Rui Rodrigues, Joao P Barreto, and Urbano Nunes. Camera pose estimation using images of planar mirror reflections. In *Computer Vision–ECCV 2010*, pages 382–395. Springer, 2010.

[SHBS11] John Stowers, Michael Hayes, and Andrew Bainbridge-Smith. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE International Conference on*, pages 358–362. IEEE, 2011.

[SSS08] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[TJRF13] Yasuhiro Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5182–5189. IEEE, 2013.

[YWC05] Ying Kin Yu, Kin Hong Wong, and Michael Ming-Yuen Chang. Recursive three-dimensional model reconstruction based on kalman filtering. *IEEE Transactions on Systems, Man and Cybernetics: Part B (Cybernetics)*, 35(3):587–592, 2005.

[YWC06] Ying Kin Yu, Kin Hong Wong, and Michael Ming-Yuen Chang. Merging artificial objects with marker-less video sequences based on the interacting multiple model method. *IEEE Transactions on Multimedia*, 8(2):521–528, 2006.

[Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10, 2012.

[ZZC+13] Zhou Zhang, Mingshao Zhang, Yizhe Chang, El-Sayed Aziz, Sven K Esche, and Constantin Chassapis. Real-time 3d model reconstruction and interaction using kinect for a game-based virtual laboratory. In *ASME 2013 International Mechanical Engineering Congress and Exposition*, pages V005T05A053–V005T05A053. American Society of Mechanical Engineers, 2013.