International Conference on Real-time Computing and Robotics
Changsha, China, June 23-26, 2015

RESEARCH PAPER

3D Modeling of Indoor Environments Using Multiple Kinects

Ruibin Guo[1*], Dongxiang Zhou [1], Keju Peng [1] and Yunhui Liu[2]

Abstract

This paper proposes a novel 3D modeling system to reconstruct dense and full indoor environments by the registration of different-region reconstruction volumes obtained by using single or multiple Kinect sensors simultaneously. We highlight two key techniques for applying the extension of KinectFusion to the modeling problem: (i) 2 GPU-based volumes' cyclical reconstruction for large indoor environments using only single Kinect in real time; (ii) dynamic calibration for Multiple Kinects by combining the features of sparse key-frames and Simplified-ICP algorithm. Experimental results show the effectiveness and scalability of our proposed approach.

Keywords: cyclical reconstruction; 3D registration; real time; GPU

1 INTRODUCTION

With the prevalence of Kinect sensors, it is convenient to obtain RGB-D image databases. The availability of such rich images presents enormous opportunities for mapping or 3D modeling. Many 3D scene reconstruction systems by using Kinect have been published in recent years. Henry[1,2] etc. resent a RGB-D mapping method, a framework for using RGB-D cameras to generate dense 3D models of indoor environments, which is a full 3D mapping system that utilizes a joint optimization algorithm combining visual features and shape-based alignment. The KinectFusion algorithm[3,4] introduced by Newcombe and Izadi can be used for accurate real-time mapping of complex and arbitrary indoor scenes in variable lighting conditions, it's a frame-to-global method and it can reduce error propagation compare to frame-to-frame ones, it maintained the single scene model with a global volumetric, truncated signed distance function(TSDF) representation. All of the methods mentioned above use only one Kinect, and there are lots of extended methods based on RGB-D mapping and KinectFusion. Considering to the view limitation of single Kinect, some relevant works had been published to achieve a complete 3D model by multiple Kinects in different views. Wang [5] etc. proposed a novel plane-sweeping based algorithm to handle interference caused by multiple cameras in the projected light overlap regions. Alexiadis[6] implemented a real-time, full 3D reconstrction of moving foreground objects from multiple consumer depth cameras.

Despite these methods have shown lots of encouraging results, some shortcomings still exist. RGB-D Mapping only uses two consecutive frames to estimate the motion of the camera, this method is always used in SLAM, while it isn't real time and its accuracy is not very high. The KinectFusion system works well for mapping medium sized room, however, the reconstruction of large-scale models require too much memory and the drift of very large exploratory sequences is inevitable. The multiple Kinect 3D reconstruction systems[7,8] published usually get surfaces measurement under different views, but the extrinsic parameters between different Kinects have been pre-aligned.

In this paper we use 2 GPU-based volumes' cyclical reconstruction for large indoor environments to extend the bound-limitation of KinectFusion, the spatially extended mapping is real time by one volume performs

*correspondence:15116245356@163.com
[1] College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, Hunan, China.
Full list of author information is available at the end of the article.

dense tracking&mapping while the other performs the raycasting. To achieve a full 3D model of indoor environment efficiently, we use multiple Kinects to perform the modeling process simultaneously, and these Kinects have not been pre-calibrated, their extrinsic parameters are achieved by computing its key frames' location in the relative Kinect's coordinate system dynamically, we get the transformation matrix between different region reconstructions built by multiple Kinects, and realize 3D registration.

2 CYCLICAL RECONSTRUCTION USING 2 VOLUMES

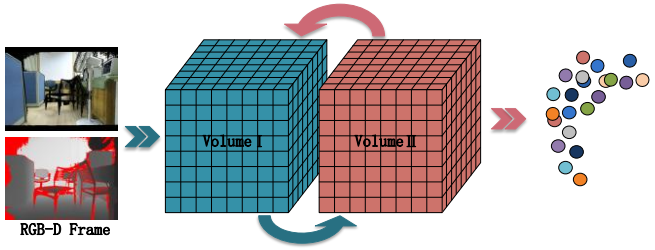
In this section we will introduce KinectFusion system briefly, and describe our 2 GPU-based volumes' cyclical reconstruction algorithm(shown in Fig.1), which is the extension of KinectFusion, this improved algorithm works well for large-scale models in real time.

2.1 Background

KinectFusion is a real-time 3D reconstruction and interaction system using a moving standard Kinect. There are four main stages in the system pipeline: a) Depth Map Conversion; b) Camera Tracking; c) Volumetric Integration; d) Raycasting. The incoming depth map from the camera is registered incrementally into the global TSDF[9] using ICP algorithm. The TSDF is a volumetric data structure that encodes implicit surface by storing the signed distance to the closest surface at each voxel up to a given truncation distance from the actual surface position. The raycasting process extract views of the implicit surface for rendering and tracking. According to the project of KinectFusionExplorer, each voxel in volume occppies 4 bytes memory of GPU, if the size of volume is $384\times384\times384$, the whole volume needs 256MB GPU memory. The commodity graphics usually have not enough memory for large-scale indoor environments.

2.2 Cyclical reconstruction

Camera pose estimation and surface reconstruction in KinectFusion is restricted to the region that pre-defined volume, we use cyclical reconstruction of 2 GPU-based volumes to solve the restriction on pre-defined boundaries. We represent the camera pose at time i by P_i, composed of a rotation $R_i \in SO_3$ and a



(a) (b) (c)

Fig.1 Cyclical reconstruction process: (a) the input RGB-D frame; (b) 2 volumes for cyclical reconstruction, if the camera pose's movement exceeds threshold Δp, the function of these 2 volumes changed; (c) the calculated 3D point cloud by Raycasting.

translation $t_i \in R^3$. We define 2 cubic volumes in the same size, each volume's side length in voxels v_s and the resolution of the reconstruction is 128 voxelPerMeter, the dimension in meters v_d is available in Eq. 1. If we set the voxelsPerMeters as 128, thus the TSDF volume represents a physical volume of space, e.g. usually 3m cube.

$$v_d = \frac{v_s}{128} \tag{1}$$

In the beginning stage, initially $R_0 = I$ and $t_0 = (0, 0, 0)$, only volume Ⅰ performs camera tracking and volumetric integration, when the camera pose P_j exceeds a movement threshold Δp at time j, volume Ⅰ performs raycasting and we can get the point cloud which will be stored in a CPU memory, we set the initial camera pose of volume Ⅱ as P_j, and volume Ⅱ performs camera tracking and volumetric integration simultaneously. At time k, when $D(P_k - P_j) > \Delta p$, volume Ⅱ stops camera tracking and start raycasting, calculate point cloud to be stored in a CPU memory, volume Ⅰ do the camera tracking and volumetric integration again by setting camera pose as P_k. The cyclical reconstruction algorithm is shown as follows.

Algorithm 1. Cyclical reconstruction using one Kinect

Initialization: TSDF volume Ⅰ & Ⅱ in GPU memory
P_0 camera pose at time 0
Δp movement threshold
$P_{pre\_ref} = P_0$ reference camera pose

Input: rgb_i RGB image
d_i depth image
for each rgb_i-d_i frame at time i
if ($D(P_i - P_{pre\_ref}) > \Delta p$)
$P_{pre\_ref} = P_i$
raycast to extract implicit surface in volume Ⅰ(Ⅱ)
set initial camera pose of volume Ⅱ(Ⅰ) as P_i
camera tracking&volumetric integration TSDF in volume Ⅱ(Ⅰ)
else
camera tracking&volumetric integration TSDF in volume Ⅰ(Ⅱ)
raycast to extract implicit surface in volume Ⅱ(Ⅰ)

The movement threshold Δp , composed of a rotation ΔR and a translation $\Delta t = (\Delta x, \Delta y, \Delta z)$. In the cyclical reconstruction, we only used the accumulated translation threshold in each axis.

3 Dynamic Calibration of Multiple Kinects

It is more efficient to achieve a full 3D model of indoor environment by using multiple Kinects simultaneously. Unlike the proposed methods, in which the positions of multiple Kinects are fixed and the calibration procedure need to be done only once. Our presented method does not have the restriction on Kinects' positions, we estimate the extrinsic parameters of these Kinects by computing their similar keyframes' location in the relative Kinect's coordinate system dynamically, and this extrinsic parameters will be used in 3D registration for reconstructions built by different Kinects, Fig.2 outlines our proposed method using 2 Kinects, three or more models' registration can be extended.

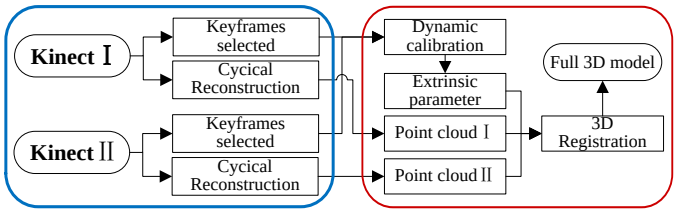


Fig.2 Full 3D models' building using two Kinects.

The key to realize the registration of multiple models is the computation for transformation matrix. To achieve this purpose, we can use point clouds' registration directly, i.e. ICP algorithm. While it's well known that the coarse registration of different point clouds is hard to estimate if the models' positions have not been pre-calibrated. So we use both depth and color images to align the point clouds obtained by multiple Kinects.

3.1 Sparse Keyframes selected

Our cycical reconstruction algorithm is the extension of KinectFusion, which is a real-time technology, if we choose every n-th frame as the keyframe simply, the computation required for the matching between frames grows quickly. Considering to the fact that we can get the 'groundtruth' Kinect pose in the reconstruction procedure, our method utilizes the Kinect's spatial movement to select keyframes.

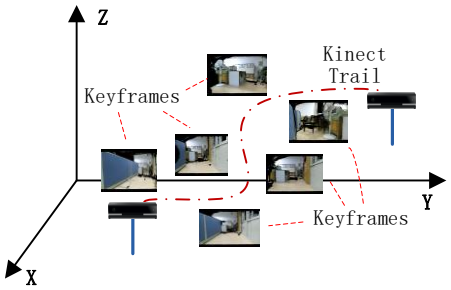


Fig.3 Keyframes selected sample.

Unlike the keyframes selected using visual features' match in RGB-D mapping, we select a frame as keyframe whenever the accumulated rotation or translation of Kinect pose is above a threshold, and we record its current Kinect Pose at the same time. The accumulated rotation threshold measured by Euler angles $(\Delta\alpha, \Delta\beta, \Delta\gamma)$ and accumulated translation $(\Delta x, \Delta y, \Delta z)$.

In this section, we will describe the details about dynamic calibration for N Kinects. Supposed that we use N Kinects to reconstruct the environment, and obtained N models separately. We set the 1st model built by Kinect 1 as the reference, to achieve a full model of the indoor environment, the rest N-1 models should be aligned to the reference one. The keyframes of model 1 defined as $S_1 = \{f_{11}, f_{12}, ... f_{1N_1}\}$, f_{1n} represents the $n\text{-}th$ keyframe, contains a RGB image and a depth image, its corresponding camera pose p_{1n} , N_1 is the number of keyframes in model 1. Similarly, the keyframes of model k is defined as $S_k = \{f_{k1}, f_{k2}, ... f_{kN_k}\}$, N_k is the number of keyframes

in model k, p_{kn} represents the corresponding camera pose of keyframe f_{kn}.

To estimate the extrinsic parameter between model 1 and model k, composed of a rotation R_{1k} and a translation T_{1k}, we measure the similarity of each frame in S_1 and S_2 by their visual feature.

3.2 Similarity measurement

In order to differentiate the similarity between f_{1i} and f_{kj}, we use SIFT features to measure it. SIFT features are invariant to image scale and rotation, which perform reliable matching between different views of an object or scene, first presented by David G.Lowe. The SIFT features' extraction contains:(a) Detection of scale-space extrema; (b) Accurate keypoint localization; (c) Orientation assignment; (d) The local image descriptor.

After getting the SIFT descriptors of two images, we use KNN match method, where we set k=2. For a feature point in image f_{1i}, several candidate feature points in image f_{kj} are similar to it, assuming that the smallest distance between this point and one of its candidate feature points is d_1, the second smallest distance is d_2, when it meets:

$$\frac{d_1}{d_2} < Th\,(Th = 0.8) \tag{2}$$

We define this pair of points are matched feature points temporally, shown in Fig.4.

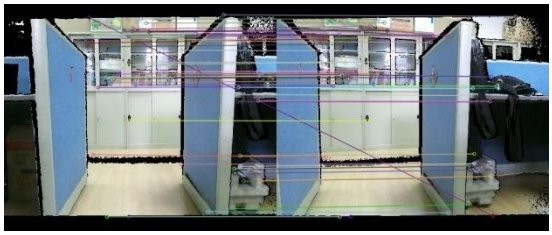


Fig.4 Keypoint Matching using 2NN method

Note that there are some wrong matched points, we add the restriction of the third smallest distance d_3 and the fundamental matrix F, x_{src} and x_{dst} are the pre-matched points' coordinates, the optimization result shown in Fig.5.

$$\begin{cases} \dfrac{d_1}{d_3} < 0.85 \times Th(Th = 0.8) \\ \left\| x_{dst} - F \bullet x_{src} \right\| < Th' \end{cases} \tag{3}$$

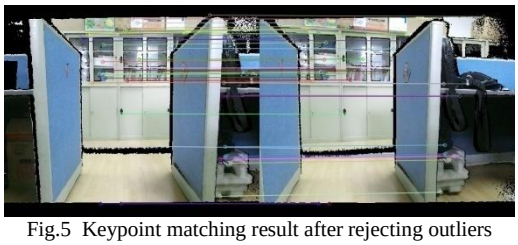


Fig.5 Keypoint matching result after rejecting outliers

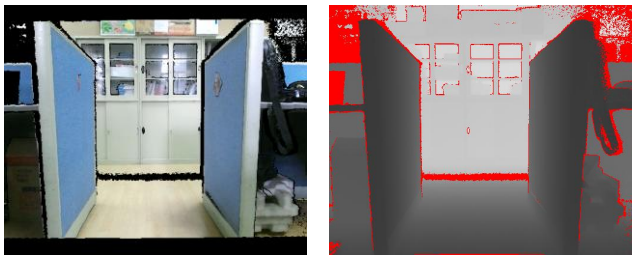
Thus, the matched points' number N_{1k}^{mn} is defined as the similarity measurement for frame m in S_1 and frame n in S_k.

3.3 Extrinsic parameter

After the similarity measurement between S_1 and S_k, we can get the most similar frames f_{1m} and f_{kn} by

$$N_{1k}^{mn} = \max_{\substack{i=1,2...N_1 \\ j=1,2...N_k}} (N_{1k}^{ij}) \tag{4}$$

The keyframe contains a RGB image and a depth image, but their initial image size are not the same, we need adjust color to the same space as depth, shown in Fig.6, so that we can get the depth information of per-pixel in RGB image.



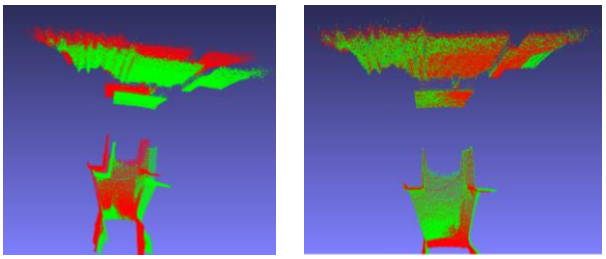
(a) (b)

Fig.6 Alignment for RGB image and depth image.(a) the aligned RGB image; (b) the depth image, the red pixels are the invalid depth information

Assuming that the coordinate of SIFT feature point in RGB image f_{1m} plane is $\vec{x}_{1m}^{\,i}$, its corresponding 3D point is $\vec{X}_{1m}^{\,i}$, the matched point's coordinate in image f_{kn} represents as $\vec{x}_{kn}^{\,i}$ and the corresponding 3D point is $\vec{X}_{kn}^{\,i}$, so $\vec{X}_{1m}^{\,i}$ and $\vec{X}_{kn}^{\,i}$ are the corresponding 3D points

for point clouds $Model_{1m}$ and $Model_{kn}$ generated by the frame f_{1m} and f_{kn} separately, thus we obtain N_{1k}^{mn} pairs of corresponding 3D points.

For the registration of $Model_{1m}$ and $Model_{kn}$, $Model_{1m}$ and $Model_{kn}$ are the sub-part of $Model_1$ and $Model_k$, we use the simplified Iterative Closet Point (ICP) algorithm to achieve the transformation matrix $[R|T]$, shown in Fig.7, the 'simplified' means that we have skipped the regular step 1: Finding the closet point in the reference point cloud, for the reason that we have obtained the closet point pairs by matching their visual feature.



(a) (b)

Fig.7 Registration for point clouds. (a) Pre-registration; (b) Registration result. The point cloud in red color corresponds to the frame f_{1m}, the green one corresponds to the frame f_{kn}.

From Section 3.1, we know that the camera pose of frame f_{1m} in $Model_1$ is p_{1m}, and the camera pose of frame f_{kn} in $Model_k$ is p_{kn}, after registration, the camera pose of frame f_{kn} in $Model_1$ is computed as follows:

$$p_{1k}^{n} = [R|T] \bullet p_{1m} \tag{5}$$

The extrinsic parameter to realize the registration for point clouds $Model_1$ and $Model_k$ is:

$$\begin{aligned} [R_{1k} \mid T_{1k}] &= Invert(p_{kn}) \bullet p_{1k}^{n} \\ &= Invert(p_{kn}) \bullet [R|T] \bullet p_{1m} \end{aligned} \tag{6}$$

$Invert(p_{kn})$ represents the pose of $Model_k$'s origin in the coordinate system of local camera f_{kn}. Thus, the registration of $Model_1$ and $Model_k$ is as follows:

$$Model_{fin} = Model_1 + [R_{1k} \mid T_{1k}] \bullet Model_k \tag{7}$$

For N Kinects, the full model of indoor environment is:

$$Model_{fin} = Model_1 + \sum_{k=2,3,....N} [R_{1k} \mid T_{1k}] \bullet Model_k \tag{8}$$

4 EXPERIMENTAL RESULTS

To verify the effectiveness of our algorithm, we performed experiment in laboratory environment. We realized the modeling for large scale indoor environment by using cyclical reconstruction in real time with single Kinect, and the movement threshold $\Delta t = (\Delta x, \Delta y, \Delta z) = (1,1,3)$. We used 4 Kinects to reconstruct our laboratory environment, and defined the 1st Kinect as the reference one, the rest modeling results were aligned to the reference one. The experimental program running on Inter(R) Core(TM) i7 CPU 2.80GHz Window8 platform, 2G GPU memory, the compiler is VS2012, and we used Kinect 2.0.

In the process of cyclical reconstruction, we selected keyframes of $Model_1$ and recorded its corresponding pose, shown in Fig.8. The red line is the sensor's trail, and the black nodes represents the keyframes' pose. We have done some test for dynamic calibration, the left of matched images is the keyframe of $Model_1$, and the right ones are arbitrarily captured images.

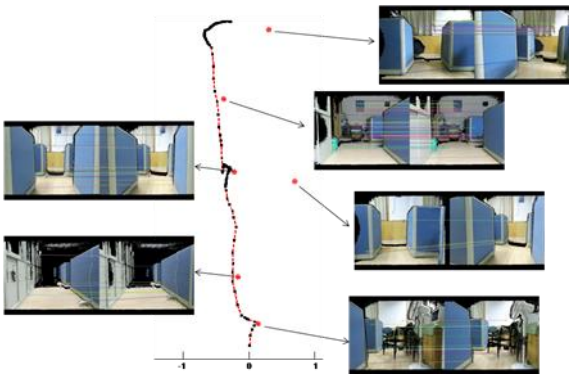
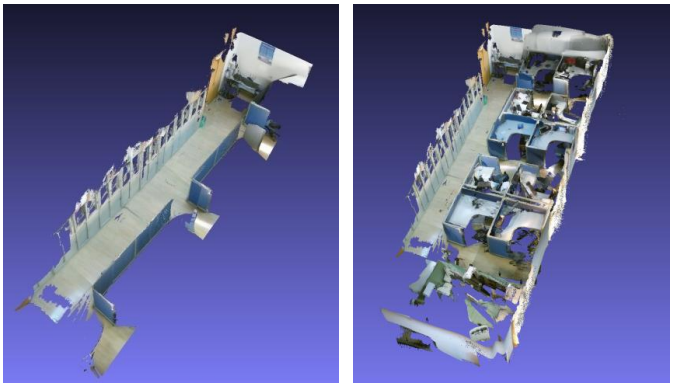


Fig.8 The trail of keyframes and dynamic calibration with arbitrary images.

Besides the Kinect 1 performed the reconstruction of $Model_1$, the rest 3 Kinects do the reconstruction for the desks and sub-corridor, we used the proposed method to do the registration of the 4 models, the result

is shown in Fig.9. The ground truth size of our reconstruction area is $18m \times 6m$.



(a) (b)

Fig.9 Modeling results. (a) Cyclical reconstruction result using Kinect 1;(b) The full model of indoor environment using 4 Kinects

5 CONCLUSIONS

This paper presents a cyclical reconstruction for indoor environment using only one Kinect in real time, and we introduce dynamic calibration for Multiple Kinects by combining the sparse key-frames features' matching and Simplified-ICP algorithm. It effectively overcomes the shortcoming of bounding box limitation in KinectFusion, and it is scalability in laboratory environment. In the future work, we will try to realize the communication between different Kinects' modeling process, and realize the real time registration for the modeling of indoor environment.

Author's contributions
The authors declare that they have no competing interests.

Acknowledgements
The support of National University of Defense Technology through project ZDYYJCYJ20140601 is gratefully acknowledged.

Author details
[1] College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, Hunan, China.
[2] Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China.

References

1. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments [C]. Proceedings of the International Symposium on Experimental Robotics (ISER), 2010.
2. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environments [J]. International Journal of Robotics Research(IJRR), 2012.
3. Newcombe R A, Davison A, Davison A J, Izadi S, et al. Kinectfusion：Real-time dense surface mapping and tracking [C]. Mixed and augmented reality(ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011: 127-136.
4. Izadi S, Newcombe R A, Kim D, et al. KinectFusion: real-time dynamic 3D surface reconstruction and interaction [C]. In: ACM SIGGRAPH 2011 Talks, ACM(2011):23.
5. Wang J, Zhang C, Zhu W,et al. 3D scene reconstruction by multiple structured-light based commodity depth cameras [C]. Acoustics, Speech and Signal Processing(ICASSP), 2012 IEEE international Conference on. IEEE, 2012:549-5432.
6. Alexiadis D S, Zarpalas D, Daras P. Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras [J]. Multimedia, IEEE Transactions on, 2013,15(2):339-358.
7. Lim Y W, Lee H Z, Yang N E, et al. 3-D reconstruction using the Kinect sensor and its application to a visualization system [C]. Systems, Man, and Cybernetics(SMC), 2012 IEEE International Conference on. IEEE, 2012: 3361-3366.
8. Nakazawa M, Mitsugami I, Makihara Y, et al. Dynamic scene reconstruction using asynchronous multiiple kinects [C]. Pattern Recognition(ICPR), 2012 21st International Conference on. IEEE, 2012:469-472.
9. B. Curless and M. Levoy. a volumetric method for building complex models from range images. ACM Trans. Graph. 1996.
10. Whelan T, Johannsson H, Kaess M, et al. (2013a) Robust real-time visual odometry for dense RGB-D mapping. In: IEEE internationalconference on robotics and automation, ICRA, Karlsruhe,Germany.
11. Whelan T, Kaess M, Fallon M, et al. (2012) Kintinuous: Spatially extended KinectFusion. In: RSS workshop on RGB-D: Advanced reasoning with depth cameras, Sydney, Australia.
12. Whelan T, Kaess M, Leonard J, et al. (2013b) Deformation-based loop closure for large scale dense RGB-D SLAM. In: IEEE RSJ international conference on intelligent robots and systems, IROS, Tokyo, Japan.
13. Handa A, Newcombe R, Angeli A, et al. (2012) Real-time camera tracking: When is high frame-rate best? In: ECCV 2012 (Lecture Notes in Computer Science, vol. 7578), pp. 222–235.
14. Henry P, Fox D, Bhowmik A, et al. (2013a) Patch volumes: Multiple fusion volumes for consistent RGB-D modeling. In: RSS workshop on RGB-D: Advanced reasoning with depth cameras, Berlin, Germany.
15. R. Macknojia, A. Chavez-Aragon, P. Payeur, and R. Laganiere, "Calibration of a network of kinect sensors for robotic inspection over a large workspace," in Robot Vision (WORV), 2013 IEEE Workshop on, Jan 2013, pp. 184–190.
16. Kinect for windows sensor components and specifications. Access: 13 March, 2014. [Online]. Available: http://msdn.microsoft.com/enus/library / jj131033.aspx
17. Zeng M, Zhao F, Zheng J, et al. (2012) A memory-efficient KinectFusion using octree. In: Computational Visual Media (Lecture Notes in Computer Science, vol. 7633). New York, NY: Springer, pp. 234–241.
18. Zhou Q and Koltun V (2013) Dense scene reconstruction with points of interest. In: SIGGRAPH 2013, Anaheim, CA, USA.
19. " KinectFusion extensions to large scale environments." http://www.pointclouds.org/blog/srcs/fheredia/index.php, August 10th 2012.