Accurate Intrinsic Calibration of Depth Camera with Cuboids

Bingwen Jin, Hao Lei, and Weidong Geng

State Key Lab. of CAD&CG,
College of Computer Science and Technology, Zhejiang University,
Hangzhou, 310027, China

Abstract. Due to the low precision, the consumer-grade depth sensor is often calibrated jointly with a color camera, and the joint calibration sometimes presents undesired interactions. In this paper, we propose a novel method to carry out the high-accuracy intrinsic calibration of depth sensors merely by the depth camera, in which the traditional calibration rig, checker-board pattern, is replaced with a set of cuboids with known sizes, and the objective function for calibration is based on the length, width, and height of cuboids and its angle between the neighboring surfaces, which can be directly and robustly calculated from the depth-map. We experimentally evaluate the accuracy of the calibrated depth camera by measuring the angles and sizes of cubic object, and it is empirically shown that the resulting calibration accuracy is higher than that in the state-of-the-art calibration procedures, making the commodity depth sensors applicable to more interesting application scenarios such as 3D measurement and shape modeling etc.

Keywords: intrinsic calibration, depth camera, 3D measurement, depth map, cuboids.

1 Introduction

Nowadays there has been an increasing number of depth cameras available at commodity prices, such as Microsoft Kinect. Although it was primarily designed for natural interaction for video game, the low cost, reliability and speed of the measurement promises of Kinect-type cameras have created a lot of interesting new research applications [26,16], such as 3D scanning and modeling [10,22,30,13,9]. Unfortunately Kinect-type 3D sensors are usually low accuracy, low precision devices. Diverse studies [14,16] show that their accuracy decreases with myopic intrinsic parameters [27] when the distance from sensor increases. This level of accuracy is quite satisfactory in human interaction applications. However, it is definitely insufficient in many exciting applications (e.g., indoor navigation, 3D measurement or fine manipulation) that require a relatively high accuracy of depth sensors.

Depth cameras (Kinect etc.) are usually pre-calibrated with a proprietary algorithm. The calibrated parameters are stored in the device during manufacturing and are used by the official drivers to calculate the 3D point clouds. The

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 788–803, 2014.

manufacturer's calibration does not correct the depth distortion, and accuracy can be improved by software correction of sensor outputs later on. The correction is based on a specific calibration model whose intrinsic parameters are identified during the calibration process. In general, the calibration of a sensor measuring a quantity is the estimation of the relationship between the measured quantity and the actual quantity (also called the ground truth) [3].

The depth data from depth cameras are usually in low precision, and therefore it is often calibrated jointly with a color camera, which has a potential of improving the accuracy of optimal solution. However the joint calibration sometimes presents undesired interactions [14], e.g. in [11], it shows that a refinement of depth model can paradoxically lead to enlargement of reprojection errors of RGB camera.

In this paper, we aim at providing an intrinsic calibration algorithm with higher accuracy for the Kinect community merely by depth camera. Cuboids with known size, instead of traditional checker-board, are chosen as the reference calibration rig, since the depth camera itself can robustly measure the length, width, and height of cuboids and its angle between the neighboring surfaces. Therefore the known sizes and angles in reference cuboids could be registered as the ground truth for the calibration of depth sensor, and the resulting sizes and angles in 3D shape modeling could also have a higher accuracy potentially. The work here focused on a calibration and analysis of accuracy of Kinect sensor, but the results and the approach are applicable to other similar sensors.

The paper proceeds with a short review of representative works on calibration of Kinect-type sensors in section 2. In section 3 the calibration model and its algorithmic steps are presented. Section 4 analyzed the accuracy of the resulting intrinsic calibration. Section 5 summarizes conclusions on the novel approach.

2 Related Work

Kinect-type 3D sensors considered in this work operate as structured light sensors. Accuracy of them can be improved by intrinsic calibration, which can be classified into two categories: supervised calibration and unsupervised calibration, by the principle of whether the dimensions of the calibration rig are known in advance or not.

2.1 Supervised Calibration

Calibration of Kinect-type 3D sensors is naturally split into two parts: identification of parameters of sensor cameras and identification of parameters of depth measurement model. The early works from Burrus [2] and Zhang and Zhang [29] identified intrinsic parameters of sensor cameras using checker-board pattern. Later, researchers propose methods to achieve calibration of sensor depth measurement model, e.g. Khoshelham and Elberink [16], Smisek et al. [26], Kim et al [17], Herrera. et al. [11] and its enhanced method in [24]. In these work, approaches are proposed for depth map conversion from disparity maps provided

by the sensor. As a nominal factory model in OpenNI and Microsoft Kinect SDK already can convert disparity data into depth, a reformulation of depth calibration model was proposed in [14,15], utilizing a linear relationship between actual and sensor-provided depth data.

An important issue in depth model calibration is how to get the accurate depth data as it normally requires either a special 3D calibration rig or an external measuring tool. In [16], depth was measured using a simple measuring tape. In [11], correspondence between depth map and RGB camera image was established using external corners of calibration table. A similar approach was proposed by Draelos et al. [4]. Geiger et al. [6] employed multiple checker-boards as reference objects. Shibo and Qing [25] designed a specific planar board with regularly-spaced drilled holes allowing their easy identification in both RGB images and depth maps.

In our work, cuboids with known sizes are employed as a depth calibration rig for subsequent calibration of the depth model based on ground-truth angles and dimensions externally measured in advance.

2.2 Unsupervised Calibration

The accuracy problem will arise occasionally in depth measurement, due to the myopic property of Kinect-type depth camera. In order to facilitate this problem, the calibration without the pre-measured ground-truth, called unsupervised calibration, is preferred. One of the popular approaches to carry out unsupervised calibration is to couple the calibration procedure with specific application scenarios such as 3D mapping or registration of depth map [21]. Yamazoe et al. [28] estimate the intrinsic parameters by showing a planar board with unknown size to the depth camera with different poses and distances with the objective function of minimizing the plane-fitting errors. Kummerle et al. [19] embed the calibration procedure into the application scenarios and solve the objective function with simultaneous localization and mapping (SLAM). Using SLAM, one can capture relatively accurate close range data and build a map. Then inaccurate long range data could be compared with the expected measurements from the map. Based on this idea, Teichman et al. [27] further presented a generic approach to make RGBD intrinsic calibration with CLAMS: (Calibrating, Localizing, and Mapping, Simultaneously) by recording a few minutes of data from natural environments, and neither special calibration targets nor measurements of the environment is required.

From the point of view of manipulation, the unsupervised calibration will be much easy and convenient for the end user or a dedicated application scenario, however the estimated parameters will usually be less accuracy than that in supervised calibration [5].

3 Calibration Method

In Kinect-type 3D sensors, depth is calculated by sensor software on the basis of disparity of reflected patterns with respect to the reference patterns obtained

for a plane placed at a known distance from the sensor. Its intrinsic calibration is in essence to collect sensor outputs and compare them to reference data, using a special calibration rig. In our approach, it is a cubic object of precisely known dimensions from a high-precision external measurement.

Our intrinsic calibration of depth sensor can be viewed as fine-tuning the intrinsic parameters by minimizing the errors of angles and sizes of reference cuboids in depth images. The proposed calibration procedure consists of an iterative optimization loop of calibration of sensor's IR cameras, calibration of depth measurement model and disparity distortion correction.

3.1 IR Camera Calibration

To calibrate IR camera in depth sensor, we use the pinhole model [7,8]. Given a point P whose coordinates in the camera coordinate system are $(x, y, z)^T$, it is firstly normalized as $\mathbf{x}_n$:

$$\mathbf{x}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \end{bmatrix}. \tag{1}$$

We have the image coordinates:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \tag{2}$$

where $[f_x, f_y]$ is the focal length in the x and y axis respectively, and $[c_x, c_y]$ is the principal point. Given these notations, the camera model can be described by $[f_x, f_y, c_x, c_y]$.

3.2 Depth Measurement Calibration

Several models have been proposed for depth calibration [12,2,4], transforming disparity d_k into depth value z. We use the calibration function proposed in [12,2] since it strongly resembles the functional form of the relationship between depth and pixel offset given on the ROS.org wiki page [18]. The function is defined as:

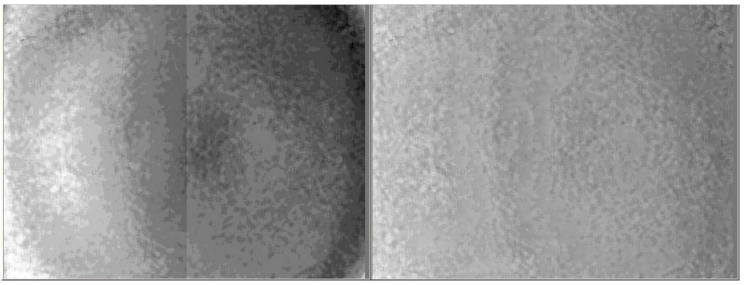
$$z(u, v) = \frac{1}{\gamma_1 d_k(u, v) + \gamma_0} \tag{3}$$

where $d_k(u, v)$ is the disparity value at image coordinate $[u, v]$, γ_0 and γ_1 are part of the intrinsic parameters in depth sensor to be estimated.

3.3 Disparity Distortion Correction

We take the same distortion model in as [11] that has per-pixel coefficients and decays exponentially with increasing disparity:

$$d_k(u, v) = d(u, v) + D(u, v) \cdot \exp(\alpha_0 - \alpha_1 d(u, v)) \tag{4}$$



(a) (b)

Fig. 1. Depth images before (a) and after (b) disparity distortion correction

where $d(u, v)$ is the disparity returned by the device, D is a matrix containing the per-pixel coefficients, $[\alpha_0, \alpha_1]$ models the decay of the error pattern, and $d_k(u, v)$ is calculated by Equation 3 with measured distance. An example of disparity distortion correction is given in Fig. 1.

It raises the problem of how to get the ground truth of depth value, as it is difficult for us to accurately measure the absolute distance between the depth camera and a fronto-parallel surface. However we observe that it is much easier for us to precisely measure the relative distance between two camera positions, with the assistance of software toolkit that can approximately determine whether the current sensor is vertical to the planar surface or not. Therefore we developed a toolkit to help acquire the desired depth data, and compute the ground truth of distance in an optimal sense by taking $d_k(u, v)$ as the internal variable in the iterative optimization. Its initial value is the approximate distance measured between the camera and the planar surface in the first reference camera position. The ground truth distance of the other camera positions are accordingly calculated by the precisely measured relative distances to the same reference camera position.

3.4 Iterative Optimization

The overall objective function for intrinsic parameter estimation has two components - the distance error and the angle error computed from the reference cuboids. The calculation of the distances between parallel surfaces and the angles of neighboring surface of cuboids is easy. It firstly builds up the plane equations of the relevant surfaces by fitting the converted 3D point clouds from the visible disparity image after the correction of IR camera and disparity distortion. Then the length, width and height of the cuboids are computed from the plane equations of the corresponding parallel surfaces, and the angles between the neighboring surfaces are calculated from the plane normals of neighboring surfaces in cuboids.

Formally, the overall objective function for intrinsic calibration is defined as the linear combination of angle and distance errors:

$$E = S \sum_{\text{view}} (\lambda E_a + (1 - \lambda) E_d), \tag{5}$$

where $S = 1/(\sum_{\text{view}} \sum_{i,j} 1)$, E_a is the angle errors, E_d is the distance errors, and λ is the weight of E_a. An instance of E_a and E_d is defined as

$$E_a = \sum_{i,j} \left(\frac{\theta_{i,j} - \bar{\theta}_{i,j}}{\bar{\theta}_{i,j}} \right)^2, \tag{6}$$

$$E_d = \sum_{i,j} \left(\frac{\delta_{i,j} - \bar{\delta}_{i,j}}{\bar{\delta}_{i,j}} \right)^2, \tag{7}$$

where $\theta_{i,j}$ and $\delta_{i,j}$ are computed angle and distance respectively, whose externally measured ground truth values are $\bar{\delta}_{i,j}$ and $\bar{\theta}_{i,j}$ respectively. More definitions for E_a and E_d are discussed and evaluated in Sec 4.3.

$\delta_{i,j}$ and $\theta_{i,j}$ are calculated by following equations:

$$\delta_{i,j} = \Big(\frac{1}{|Q_i|} \sum_{\mathbf{q}_k \in Q_i} \frac{|\mathbf{P}_j \cdot \hat{\mathbf{q}}_k|}{|\mathbf{n}_j|} + \frac{1}{|Q_j|} \sum_{\mathbf{q}_k \in Q_j} \frac{|\mathbf{P}_i \cdot \hat{\mathbf{q}}_k|}{|\mathbf{n}_i|} \Big) / 2, \tag{8}$$

$$\theta_{i,j} = \arccos \Big(\frac{\mathbf{n}_i \cdot \mathbf{n}_j}{|\mathbf{n}_i||\mathbf{n}_j|} \Big), \tag{9}$$

where $\mathbf{P}_i = (a_i, b_i, c_i, d_i)$ and $\mathbf{P}_j = (a_j, b_j, c_j, d_j)$ represent two planes equations fitted from two 3D point sets Q_i and Q_j respectively, $\hat{\mathbf{q}}_k = (x_k, y_k, z_k, 1)$, $\mathbf{n}_i$ and $\mathbf{n}_j$ are the normals of $\mathbf{P}_i$ and $\mathbf{P}_j$ respectively. Q_i and Q_j are converted from two depth pixel sets $\tilde{Q}_i$ and $\tilde{Q}_j$ respectively. Each depth pixel set corresponds to a manually marked region. For a depth pixel $(u, v, d(u, v))$, we convert it into a 3D point using

$$x = \frac{z(u - c_x)}{f_x}, \tag{10}$$

$$y = \frac{z(v - c_y)}{f_y}, \tag{11}$$

$$z = \frac{1}{r_1(d(u,v) + D(u,v) \cdot \exp(\alpha_0 - \alpha_1 d(u,v))) + r_0}. \tag{12}$$

Equations (10), (11), and (12) are derived from Equation (1), (2), (3), and (4). Accordingly, Q can be defined as

$$Q = g_{\text{cvt}}(\tilde{Q}, \mathbf{M}), \tag{13}$$

where $\mathbf{M} = (f_x, f_y, c_x, c_y, \gamma_0, \gamma_1, \alpha_0, \alpha_1, D)$ represents a vector composed of depth camera intrinsic parameters.

Given a 3D point set Q, the plane $\mathbf{P} = (a, b, c, d)$ are fitted through

$$\mathbf{P} = g_{\text{fit}}(Q) \tag{14}$$

$$= \arg\min_{a,b,c,d} \sum_{\mathbf{q}_i \in Q} (ax_i + by_i + cz_i + d)^2. \tag{15}$$

In our implementation, we employ VCG library [23] to solve it.

Given Equations (8), (9), (13), and (14), we can see $\delta_{i,j}$ and $\theta_{i,j}$ are functions of intrinsic parameters $\mathbf{M}$ and two depth pixel sets $\tilde{Q}_i$ and $\tilde{Q}_j$:

$$\delta_{i,j} = g_{\text{dis}}(\tilde{Q}_i, \tilde{Q}_j, \mathbf{M}), \tag{16}$$

$$\theta_{i,j} = g_{\text{ang}}(\tilde{Q}_i, \tilde{Q}_j, \mathbf{M}). \tag{17}$$

According to Equation (5), (6), (7), (16), and (17), we rewrite the overall objective function explicitly depending on the intrinsic parameters as

$$E = S \sum_{\text{view}} \left(\lambda \sum_{i,j} \left(\frac{g_{\text{ang}}(\tilde{Q}_i, \tilde{Q}_j, \mathbf{M}) - \bar{\theta}_{i,j}}{\bar{\theta}_{i,j}} \right)^2 + (1-\lambda) \sum_{i,j} \left(\frac{g_{\text{dis}}(\tilde{Q}_i, \tilde{Q}_j, \mathbf{M}) - \bar{\delta}_{i,j}}{\bar{\delta}_{i,j}} \right)^2 \right). \tag{18}$$

The iterative optimization for minimizing E follows a two step procedure as that in [11]. During the first step we optimize parameters $(f_x, f_y, c_x, c_y, \gamma_0, \gamma_1)$ over the cost E using the Levenberg Marquardt algorithm [20]. The initial value of these parameters are from the standard calibration of IR camera as that in [26], where the IR images for calibration are captured with libfreenect [1] (all raw data from the camera is captured with libfreenect in out implementation). The disparity correction parameters, i.e. D, α_0 and α_1 are initially set to 0. In the second step the optimized parameter values from the first step are fixed, and α_0 and α_1 are further optimized using the Levenberg Marquardt algorithm with a set of fronto-parallel surfaces spanning the entire view. Optimized α_0 and α_1 are then used to linearly solve each coefficient in D. The two steps iterate till the residual error is stable.

4 Evaluation of Sensor Accuracy

We have implemented the calibration algorithm, and in this section we will validate our approach by similar principles in [16].

4.1 Calibration Setting-Up

In our work, at least 3 well-manufactured cuboids are required for calibration, as 2 distances and 3 angles are minimally needed for an acceptable calibration. Sizes of cuboids are externally measured, $399.574 \times 249.596 \times 299.436mm$,

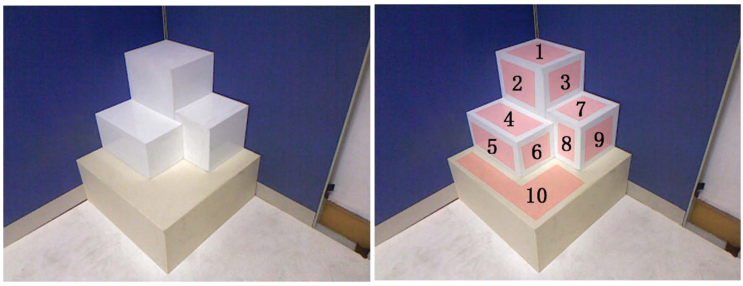


Fig. 2. Cuboids setting-up to calibrate the depth camera. Plane annotations are given in the right image

Table 1. Ground truth for calibration. Cuboids and plane annotations are shown in Fig. 2

Angle Between Planes	Degree	Distance Between Planes	Millimeter
$\bar{\theta}_{1,2}$	89.97	$\bar{\delta}_{1,10}$	399.57
$\bar{\theta}_{1,3}$	89.97	$\bar{\delta}_{4,10}$	180.25
$\bar{\theta}_{2,3}$	89.89	$\bar{\delta}_{5,2}$	199.19
$\bar{\theta}_{4,5}$	89.96	$\bar{\delta}_{7,10}$	219.43
$\bar{\theta}_{4,6}$	89.67	$\bar{\delta}_{9,3}$	198.95
$\bar{\theta}_{5,6}$	89.91		
$\bar{\theta}_{7,8}$	89.99		
$\bar{\theta}_{7,9}$	90.00		
$\bar{\theta}_{8,9}$	89.99		

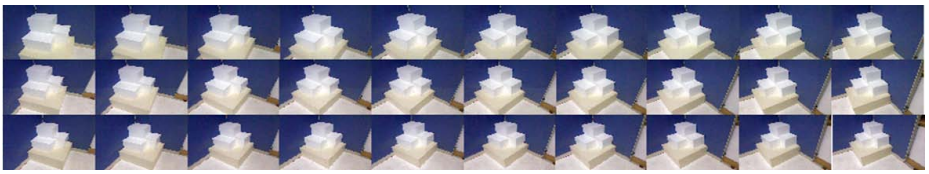


Fig. 3. 10 views of captured calibration data in 3 different distances. The 1st, 2nd, and 3rd rows are captured with the distances from camera to cuboids as $75-115cm$, $90-135cm$, and $115-155cm$ respectively.

$349.326 \times 180.252 \times 199.192mm$, and $198.948 \times 249.378 \times 219.434mm$ respectively. The combination and its 10 numbered planar regions are illustrated in Fig. 2. Table 1 gives the ground truth measurements among these planar regions used for calibration. We capture data roughly at three different distances: 75-115cm, 90-135cm, and 115-155cm. In each distance, 10 frames of depth images are captured from 10 different views (see Fig. 3). For each frame, we first convert the raw depth image into a normal map by calculating the normal of each point.

Secondly we manually marked the visible surface on cuboid based on the normal map, as shown in Fig. 4. Each marked region has a corresponding plane annotated in Fig. 2. It is worth noting that there are invalid depth pixels in the object boundaries. These invalid pixels should be avoided in the estimation of normals. In Fig. 4, we can clearly see that boundaries are not included while manually marking regions of interest. In case a marked region still includes invalid pixels, we can remove them easily since their disparity values all equal to a constant value far from valid ones'.

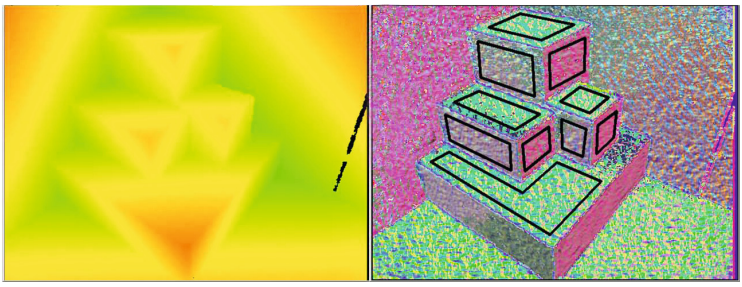


Fig. 4. The disparity map (left), the normal map, and manual segmentation (right)

In order to correct the disparity distortion, the depth data of fronto-parallel surface spanning the entire view are captured from 1m to 1.9m. However, it is difficult to manually align camera direction with the normals of fronto-parallel surfaces. Therefore, we develop a tool to semi-automatically help user make the decision. When camera is online, this tool lays a uniform grid over the depth image of each frame, and computes an averaged depth value in every rectangle. Since all depth pixels are valid except the ones near the right edge of the depth image, the grid is generated to be smaller than the depth image to exclude invalid depth pixels. As distortion pattern in depth image is centrosymmetric (see Fig. 1 (a)), the tool will alert user and automatically save current frame as a depth image for the fronto-parallel surface, when each averaged depth value is almost the same as the one computed in the symmetric box with respect to image center. With this tool, we can easily capture depth data for disparity distortion correction.

4.2 Calibration Accuracy

To evaluate the calibration accuracy, we firstly use a "hollow cube" as that in [11]. Three neighboring planes are fitted to the 3D point cloud from disparity image. The computed angles errors between neighboring planes are 0.4°, 0.05°, and 0.4°. It achieves a comparable accuracy in their evaluation experiment [11].

Combo 1 Combo 2

Fig. 5. Cuboids setup for the validation experiment. Plane annotations are also shown in this Fig.

However, with a "hollow cube", only angles can be estimated. For a more general evaluation which includes both angles and distances, we employ a set of combos of solid cubes.

Depth images are captured from two combos of cuboids (see Fig. 5) different from those in calibration. Table 2 summarizes the ground truth measurements among planar regions used for evaluation. The height of the cuboid refers to the distance between plane 1 and 4 (see Fig. 5) of a cuboid. We compare the accuracy of our calibration method and the-state-of-the-art method proposed by Herrera et al. [11]. The depth data for evaluation are also captured in three different distances: 75-115cm, 90-135cm, and 115-155cm. We assign indexes 1, 2, and 3 to the three distances for convenience. At each distance, 20 frames of depth images are captured from 20 different views respectively. Given a depth image, the evaluation error for angles is defined as $\hat{\theta} = (|\theta_{1,2} - \bar{\theta}_{1,2}| + |\theta_{1,3} - \bar{\theta}_{1,3}| + |\theta_{2,3} - \bar{\theta}_{2,3}|)/3$. The evaluation error for height is defined as $\hat{\delta} = |\delta_{1,4} - \bar{\delta}_{1,4}|$.

Firstly, we empirically set calibration parameter λ to 0.75, and the calibration result is $[f_x, f_y, c_x, c_y, \gamma_0, \gamma_1, \alpha_0, \alpha_1] = [580.083, 585.555, 311.101, 238.108, 3.0582, -0.00280854, -19.0559, 0.0023218]$ (matrix D is too large to be shown here). The resulting evaluation is presented in Fig. 6. To evaluate Herrera et al.'s method [11], we use their matlab toolbox for calibration. In Herrera et al.'s matlab toolbox (see doc\doc_2_1.pdf in their latest toolbox), they recommend capturing 30 images for calibration. Therefore, 30 images are captured for both methods while calibrating them (in fact 10 images are sufficient for our calibration). The resultant precision of our method at all distances are higher than that in the peer method [11]. When sufficient images are given, Raposo et al. [24] achieves similar results to that in Herrera et al [11]. Hence, in this situation, our method can deliver better results than Raposo et al. [24]. In current implementation, we use the method of Smisek et al. [26] for initialization, while some other methods [29][2] can also be used for initialization. In principle, calibration accuracy of Zhang and Zhang's method [29] will not be higher than that

of Smisek et al. [26], who also stated that their method is better than Burrus' one [2]. Fig. 6 also shows resultant evaluation of Smisek's method [26], which are obviously worse than ours. Therefore, our calibration accuracy is also higher than that in methods [2][29].

Table 2. Cube measurements for evaluation. Cuboids and plane annotations are shown in Fig. 5.

Angle Between Planes	Degree	Distance Between Planes	Millimeter
Combo 1			
$\theta_{1,2}$	89.99	$\delta_{1,4}$	349.25
$\bar{\theta}_{1,3}$	90.02		
$\bar{\theta}_{2,3}$	90.02		
Combo 2			
$\theta_{1,2}$	90.02	$\delta_{1,4}$	300.08
$\bar{\theta}_{1,3}$	89.99		
$\bar{\theta}_{2,3}$	90.02		

To evaluate the role of weight λ in the objective function, we experiment five different values from 0 to 1 (see Fig. 7). To clearly demonstrate the effect of λ, for each test value of λ, we average the angle and height errors as

$$\varepsilon_a = \frac{1}{60} \sum_{j=1}^{3} \sum_{i=1}^{20} \hat{\theta}_{i,j}, \tag{19}$$

$$\varepsilon_d = \frac{1}{60} \sum_{j=1}^{3} \sum_{i=1}^{20} \hat{\delta}_{i,j}, \tag{20}$$

where i is the index of frame and j is the index of distance from camera to cuboids. We clearly see that minimum errors of angle and height are both achieved when λ is roughly from 0.5 to 0.75. Hence, both the angle and distance constraints in cuboids should be taken into consideration when building up the objective functions for intrinsic calibration.

The distance from depth camera to the cuboids is also an important factor of evaluation error. Hence, we average the angle and height errors as $\frac{1}{20}\sum_{i=1}^{20}\hat{\theta}_{i,j}$ and $\frac{1}{20}\sum_{i=1}^{20}\hat{\delta}_{i,j}$ respectively, where i is also the index of frame. Fig. 8 shows the errors computed with 3 different capturing distances, when $\lambda = 0.75$. We observe from the averaged errors that 90-135cm is the best working distance.

4.3 Discussion of Objective Function

There are alternative approaches to calculate the errors of angle and distance in a combo of solid cubes. With relative and absolute metrics, more definitions on

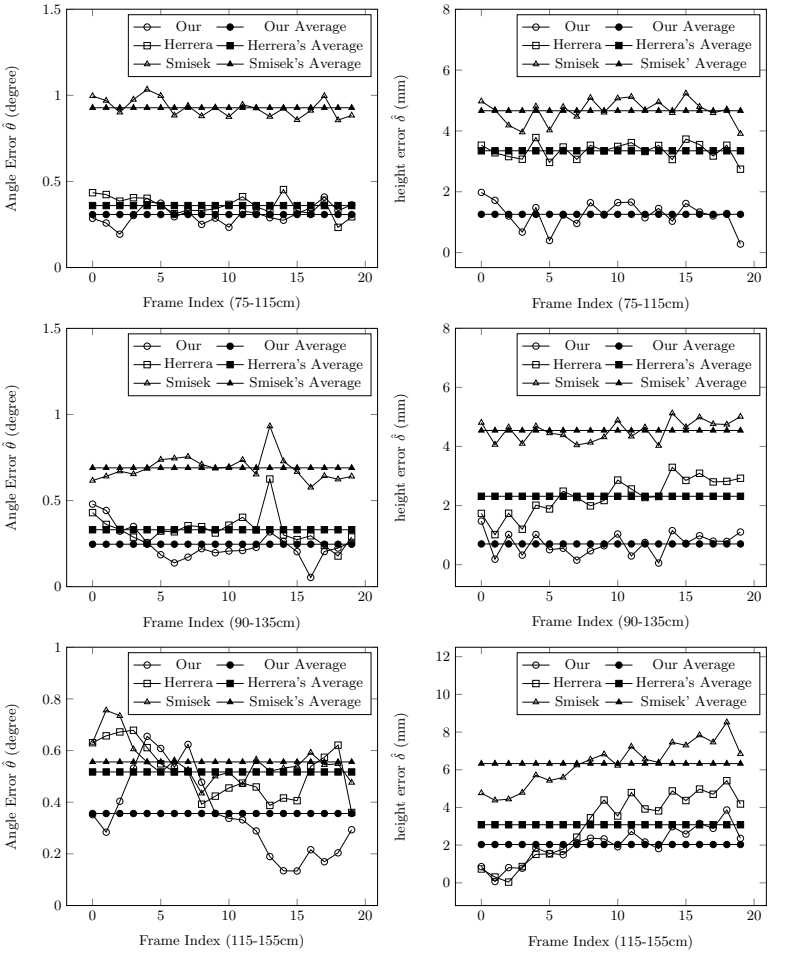


Fig. 6. Errors of evaluation across frames.

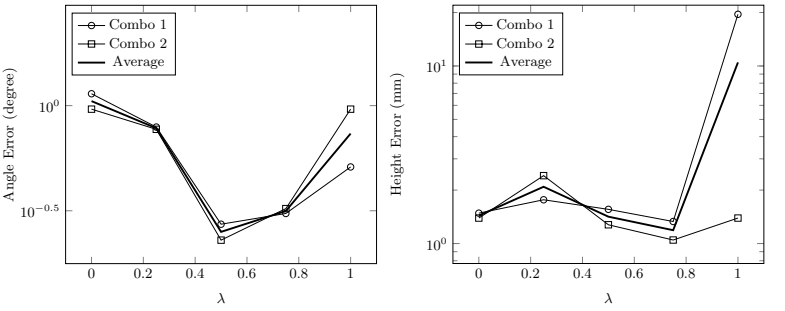


Fig. 7. Comparing errors of the calibrated depth camera with different λ values.

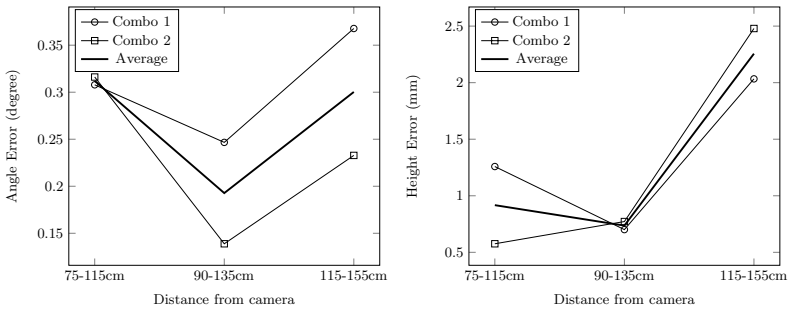


Fig. 8. Comparing errors of the calibrated depth camera with evaluation data captured in three different distances.

the objective function are given below:

$$E_a^1 = \sum_{i,j} \left(\frac{\theta_{i,j} - \bar{\theta}_{i,j}}{\bar{\theta}_{i,j}} \right)^2, E_d^1 = \sum_{i,j} \left(\frac{\delta_{i,j} - \bar{\delta}_{i,j}}{\bar{\delta}_{i,j}} \right)^2, \tag{21}$$

$$E_a^2 = \sum_{i,j} \left| \frac{\theta_{i,j} - \bar{\theta}_{i,j}}{\bar{\theta}_{i,j}} \right|, E_d^2 = \sum_{i,j} \left| \frac{\delta_{i,j} - \bar{\delta}_{i,j}}{\bar{\delta}_{i,j}} \right|, \tag{22}$$

$$E_a^3 = \sum_{i,j} (\theta_{i,j} - \bar{\theta}_{i,j})^2, E_d^3 = \sum_{i,j} (\delta_{i,j} - \bar{\delta}_{i,j})^2, \tag{23}$$

$$E_a^4 = \sum_{i,j} |\theta_{i,j} - \bar{\theta}_{i,j}|, E_d^4 = \sum_{i,j} |\delta_{i,j} - \bar{\delta}_{i,j}|, \tag{24}$$

$$E_a^5 = \omega_a^5 E_a^1 + (1 - \omega_a^5) E_a^2, E_d^5 = \omega_d^5 E_d^1 + (1 - \omega_d^5) E_d^2, \tag{25}$$

$$E_a^6 = \omega_a^6 E_a^1 + (1 - \omega_a^6) E_a^3, E_d^6 = \omega_d^6 E_d^1 + (1 - \omega_d^6) E_d^3, \tag{26}$$

$$E_a^7 = \omega_a^7 E_a^1 + (1 - \omega_a^7) E_a^4, E_d^7 = \omega_d^7 E_d^1 + (1 - \omega_d^7) E_d^4, \tag{27}$$

$$E_a^8 = \omega_a^8 E_a^2 + (1 - \omega_a^8) E_a^3, E_d^8 = \omega_d^8 E_d^2 + (1 - \omega_d^8) E_d^3, \tag{28}$$

$$E_a^9 = \omega_a^9 E_a^2 + (1 - \omega_a^9) E_a^4, E_d^9 = \omega_d^9 E_d^2 + (1 - \omega_d^9) E_d^4, \tag{29}$$

$$E_a^{10} = \omega_a^{10} E_a^3 + (1 - \omega_a^{10}) E_a^4, E_d^{10} = \omega_d^{10} E_d^3 + (1 - \omega_d^{10}) E_d^4, \tag{30}$$

where $\omega_a^i, \omega_d^i \in (0, 1)$ with $i = 5, 6, 7, 8, 9, 10$. Note E_a^1 and E_d^1 have already been utilized in all the experiments presented in Section 4.2. In the experiment for evaluating the aforementioned objective functions, we set $\lambda = 0.75$, and use the same calibration data and Combo 1's evaluation data used in Section 4.2. ω_a^i, ω_d^i are empirically assigned in terms of $\frac{\omega_a^i E_a^j}{(1-\omega_a^i)E_a^k} \approx 1$, $\frac{\omega_d^i E_d^j}{(1-\omega_d^i)E_d^k} \approx 1$ in the first 100 computations of E_a^i and E_d^i, where $(i, j, k) = (5, 1, 2), (6, 1, 3), (7, 1, 4), (8, 2, 3), (9, 2, 4), (10, 3, 4)$. The resulting average errors of angle and height are calculated with Equation (19) and (20) respectively and shown in Fig. 9 (Equation (21-30) are numbered as $1, 2, \cdots, 10$ respectively in Fig. 9). Due to the difficult tradeoff between the errors of angle and distance, we will not recommend which one is the best objective function. However, we can clearly see that all the average

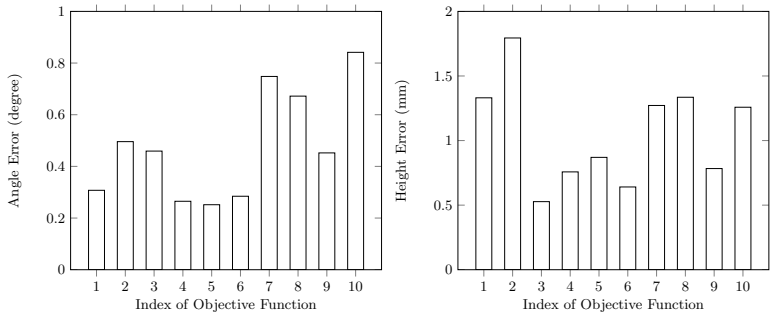


Fig. 9. The resulting errors evaluated with the candidate objective functions

angle errors are less than 1 degree, and all the average height errors are around 1mm. Such a calibrated precision is applicable to most 3D measurement and shape modeling.

5 Conclusion

The major contributions in our depth camera calibration system are 1) providing an alternative approach to estimate the intrinsic parameters of depth sensors merely by the depth data itself; 2) a relatively high-accuracy calibration of the low-precision depth sensors, better-fitted to the application of 3D measurements and shape modeling where high geometrical accuracy is usually required. What's more, our method is also useful in popularizing high-precise 3D modelling for CAD and 3D printing, since Kinect is much cheaper than other accurate 3D scanners.

The limitation of our approach is also obvious. Tens of thousands of 3D points are involved in the iterative plane fitting and non-linear optimization, and the calibration process is time-consuming. It takes around 5 minutes to accomplish the iterative optimization on a PC equipped with 4GB memory and an AMD Phenom^{TM}II X4 955 CPU 3.20GHz. In the future, we plan to develop a GPU-based parallel algorithmic pipeline for iterative plane fitting and non-linear optimization. Also we plan to extend our calibration system with an automatic or semi-automatic segmentation process, and develop a flexible extension to other types of depth cameras.

References

1. Blake, J., et al.: Openkinect, `http://openkinect.org`
2. Burrus, N.: Kinect calibration, `http://nicolas.burrus.name/index.php/Research/KinectCa`
3. Dal Mutto, C., Zanuttigh, P., Cortelazzo, G.M.: Time-of-Flight Cameras and Microsoft Kinect^{TM}. Springer (2012)

4. Draelos, M., Deshpande, N., Grant, E.: The kinect up close: Adaptations for short-range imaging. In: 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 251–256. IEEE (2012)
5. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the rgb-d slam system. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1691–1696. IEEE (2012)
6. Geiger, A., Moosmann, F., Car, O., Schuster, B.: Automatic camera and range sensor calibration using a single shot. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 3936–3943. IEEE (2012)
7. Heikkila, J.: Geometric camera calibration using circular control points. PAMI 22(10), 1066–1077 (2000)
8. Heikkila, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. PAMI, 1106–1112 (1997)
9. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) Experimental Robotics. Springer Tracts in Advanced Robotics, vol. 79, pp. 477–491. Springer, Heidelberg (2012)
10. Herbst, E., Henry, P., Ren, X., Fox, D.: Toward object discovery and modeling via 3-d scene comparison. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2623–2629. IEEE (2011)
11. Herrera, C., Kannala, J., et al.: Joint depth and color camera calibration with distortion correction. PAMI 34(10), 2058–2064 (2012)
12. Herrera, D., Kannala, J., Heikkilä, J.: Accurate and practical calibration of a depth and color camera pair. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part II. LNCS, vol. 6855, pp. 437–445. Springer, Heidelberg (2011)
13. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Andrew, Davison, o.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: UIST, pp. 559–568. ACM (2011)
14. Karan, B.: Accuracy improvements of consumer-grade 3d sensors for robotic applications. In: 2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY), pp. 141–146. IEEE (2013)
15. Karan, B.: Calibration of depth measurement model for kinect-type 3d vision sensors (2013)
16. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors 12(2), 1437–1454 (2012)
17. Kim, J.-H., Choi, J.S., Koo, B.-K.: Calibration of multi-kinect and multi-camera setup for full 3d reconstruction. In: 2013 44th International Symposium on Robotics (ISR), pp. 1–5. IEEE (2013)
18. Konolige, K., Mihelich, P.: kinect calibration/technical.ros.org, `http://www.ros.org/wiki/kinectcalibration/technical`
19. Kummerle, R., Grisetti, G., Burgard, W.: Simultaneous calibration, localization, and mapping. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3716–3721. IEEE (2011)
20. Lourakis, M.: levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++, `http://users.ics.forth.gr/%7elourakis/levmar/`
21. Macknojia, R., Chávez-Aragón, A., Payeur, P., Laganière, R.: Calibration of a network of kinect sensors for robotic inspection over a large workspace. In: 2013 IEEE Workshop on Robot Vision (WORV), pp. 184–190. IEEE (2013)

22. Menna, F., Remondino, F., Battisti, R., Nocerino, E.: Geometric investigation of a gaming active device. In: SPIE Optical Metrology. pp. 80850G–80850G. International Society for Optics and Photonics (2011)
23. Paolo Cignoni, F.G.: The visualization and computer graphics library, http://vcg.isti.cnr.it/~cignoni/newvcglib/html/index.html
24. Raposo, C., Barreto, J.P., Nunes, U.: Fast and accurate calibration of a kinect sensor. In: 2013 International Conference on 3DTV-Conference, pp. 342–349. IEEE (2013)
25. Shibo, L., Qing, Z.: A new approach to calibrate range image and color image from kinect. In: 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 2, pp. 252–255. IEEE (2012)
26. Smisek, J., Jancosek, M., Pajdla, T.: 3d with kinect. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1154–1160. IEEE (2011)
27. Teichman, A., Miller, S., Thrun, S.: Unsupervised intrinsic calibration of depth sensors via slam. In: Robotics: Science and Systems, RSS (2013)
28. Yamazoe, H., Habe, H., Mitsugami, I., Yagi, Y.: Easy depth sensor calibration. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 465–468. IEEE (2012)
29. Zhang, C., Zhang, Z.: Calibration between depth and color sensors for commodity depth cameras. In: 2011 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2011)
30. Zollhöfer, M., Martinek, M., Greiner, G., Stamminger, M., Süßmuth, J.: Automatic reconstruction of personalized avatars from 3d face scans. Computer Animation and Virtual Worlds 22(2-3), 195–202 (2011)