CALIBRATION BETWEEN DEPTH AND COLOR SENSORS FOR COMMODITY DEPTH CAMERAS

Cha Zhang and Zhengyou Zhang

Communication and Collaboration Systems Group, Microsoft Research
{chazhang, zhang}@microsoft.com

ABSTRACT

Commodity depth cameras have created many interesting new applications in the research community recently. These applications often require the calibration information between the color and the depth cameras. Traditional checkerboard based calibration schemes fail to work well for the depth camera, since its corner features cannot be reliably detected in the depth image. In this paper, we present a maximum likelihood solution for the joint depth and color calibration based on two principles. First, in the depth image, points on the checkerboard shall be co-planar, and the plane is known from color camera calibration. Second, additional point correspondences between the depth and color images may be manually specified or automatically established to help improve calibration accuracy. Uncertainty in depth values has been taken into account systematically. The proposed algorithm is reliable and accurate, as demonstrated by extensive experimental results on simulated and real-world examples.

Index Terms— depth camera, calibration

1. INTRODUCTION

Recently, there has been an increasing number of depth cameras available at commodity prices, such as those from 3DV systems[1] and Microsoft Kinect[2]. These cameras can usually capture both color and depth images in real time. They have created a lot of interesting new research applications, such as 3D shape scanning [1], foreground/background segmentation [2], facial expression tracking [3], etc.

For many applications that use the color and the depth images jointly, it is critical to know the calibration parameters of the sensor pair. Such parameters include the intrinsic parameters of the color camera, its radial distortion parameters, the rotation and translation between the depth camera and the color camera, and parameters that help determine the depth values (e.g., in meters) of pixels in the depth image. Although color camera calibration has been thoroughly studied in the literature [4, 5], the joint calibration of depth and color images presents a few new challenges:

[1]3DV Systems, http://www.3dvsystems.com/.
[2]Microsoft, http://www.xbox.com/en-US/kinect/.

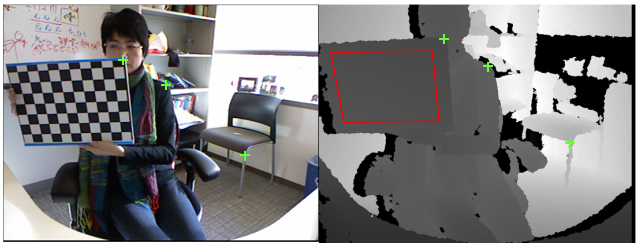

Fig. 1. The calibration pattern used in this paper. The color image is shown on the left; and the depth image is shown on the right. The depth pixels inside the red rectangle shall lie on the model plane surface, though point correspondence is difficult to obtain. A few manually specified corresponding point pairs are also shown in the figure.

- Feature points such as the corners of checkerboard patterns are often indistinguishable from other surface points in the depth image, as shown in Fig. 1.
- Although depth discontinuity can be easily observed in the depth image, the boundary points are usually unreliable due to unknown depth reconstruction mechanisms used inside the depth camera.
- One may use the infrared image co-centered with the depth image to perform calibration. However, this may require external infrared illumination (e.g., the Kinect camera). In addition, the depth mapping function (Eq. (22)) of the depth image may not be calibrated with such a method.
- Most commodity depth cameras produce noisy depth images. Such noises need to be accurately modeled in order to obtain satisfactory results.

In this paper, we propose a maximum likelihood solution for joint depth and color calibration using commodity depth cameras. We use the popular checkerboard pattern adopted in color camera calibration (Fig. 1), thus no extra hardware is needed. We utilize the fact that points on the checkerboard shall lie on a common plane, thus their distance to the plane shall be minimized. Point correspondences between the depth and color images may be further added to help improve calibration accuracy. A maximum likelihood framework is presented with careful modeling of the sensor noise, particularly

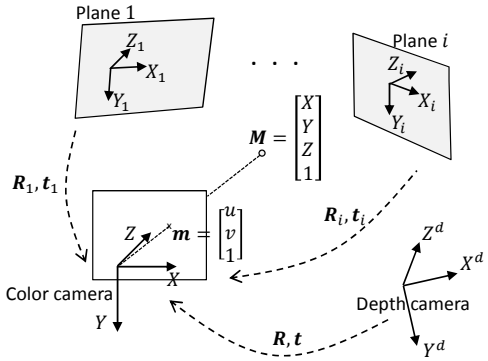

Fig. 2. Illustration of the notations used in the paper.

in the depth image. Extensive experimental results are presented to validate the proposed calibration method.

2. NOTATIONS

Fig. 2 illustrates the notations used during our calibration procedure. We assume the color camera's 3D coordinate system coincides with the world coordinate system. In the homogeneous representation, a 3D point in the world coordinate system is denoted by $\mathbf{M} = [X, Y, Z, 1]^T$, and its corresponding 2D projection in the color image is $\mathbf{m} = [u, v, 1]^T$. We model the color camera by the usual pinhole model, i.e.,

$$s\mathbf{m} = \mathbf{A}[\mathbf{I}\ \mathbf{0}]\mathbf{M}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{0}$ is the zero vector. s is a scale factor. In this particular case, $s = Z$. $\mathbf{A}$ is the camera's intrinsic matrix, given by [5]:

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where α and β are the scale factors in the image coordinate system, (u_0, v_0) are the coordinates of the principal point, and γ is the skewness of the two image axes.

The depth camera typically outputs an image with depth values, denoted by $\mathbf{x} = [u, v, z]^T$, where (u, v) are the pixel coordinates, and z is the depth value. The mapping from $\mathbf{x}$ to the point in the depth camera's 3D coordinate system $\mathbf{M}^d = [X^d, Y^d, Z^d, 1]^T$ is usually known, denoted as $\mathbf{M}^d = \mathbf{f}(\mathbf{x})$. The rotation and translation between the color and the depth cameras are denoted by $\mathbf{R}$ and $\mathbf{t}$, i.e.,

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M}^d. \tag{3}$$

3. JOINT DEPTH/COLOR CAMERA CALIBRATION

3.1. Problem Statement

During calibration, we assume the user moves a planar calibration board in front of the depth camera, similar to that in [5]. In total there are n image pairs (color and depth) captured by the depth camera. The positions of the calibration board in the n images are different, as shown in Fig. 2. We set up local 3D coordinate system (X_i, Y_i, Z_i) for each position of the calibration model plane, such that the $Z_i = 0$ plane coincides with the model plane. In addition, we assume the model plane has a set of m feature points. Usually they are the corners of a checkerboard pattern. We denote these feature points as $\mathbf{P}_j, j = 1, \cdots, m$. Note the 3D coordinates of these feature points in each model plane's local coordinate system are identical. Each feature point's local 3D coordinate is associated with the world coordinate as:

$$\mathbf{M}_{ij} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{P}_j. \tag{4}$$

where $\mathbf{M}_{ij}$ is the j^{th} feature point of the i^{th} image in the world coordinate system, $\mathbf{R}_i$ and $\mathbf{t}_i$ are the rotation and translation from the i^{th} model plane's local coordinate system to the world coordinate system. The feature points are observed in the color image as $\mathbf{m}_{ij}$, which are associated with $\mathbf{M}_{ij}$ through Eq. (1).

Given the set of feature points $\mathbf{P}_j$ and their projections $\mathbf{m}_{ij}$, our goal is to recover the intrinsic matrix $\mathbf{A}$, the model plane rotations and translations $\mathbf{R}_i, \mathbf{t}_i$, and the transform between the color and the depth cameras $\mathbf{R}$ and $\mathbf{t}$. It is well-known that given the set of color images, the intrinsic matrix $\mathbf{A}$ and the model plane positions $\mathbf{R}_i$, $\mathbf{t}_i$ can be computed [5]. It is unclear, however, whether the depth images can be used to reliably determine $\mathbf{R}$ and $\mathbf{t}$ automatically.

3.2. A Maximum Likelihood Solution

The calibration solution to the color image only problem is well known [5]. Due to the pinhole camera model, we have:

$$s_{ij}\mathbf{m}_{ij} = \mathbf{A}[\mathbf{R}_i\ \mathbf{t}_i]\mathbf{P}_j \tag{5}$$

In practice, the feature points on the color images are usually extracted with automatic algorithms, and may have errors. Assume that $\mathbf{m}_{ij}$ follows a Gaussian distribution with the ground truth position as its mean, i.e.,

$$\mathbf{m}_{ij} \sim \mathcal{N}(\bar{\mathbf{m}}_{ij}, \mathbf{\Phi}_{ij}). \tag{6}$$

The log likelihood function can be written as:

$$L_1 = -\frac{1}{2nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\epsilon_{ij}^T\mathbf{\Phi}_{ij}^{-1}\epsilon_{ij}, \tag{7}$$

where

$$\epsilon_{ij} = \mathbf{m}_{ij} - \frac{1}{s_{ij}}\mathbf{A}[\mathbf{R}_i\ \mathbf{t}_i]\mathbf{P}_j. \tag{8}$$

We next study terms related to the depth images. There are a set of points in the depth image that correspond to the model plane, as those inside the red quadrilateral in Fig. 1. We

randomly sample K_i points within the quadrilateral, denoted by $\mathbf{M}^d_{ik_i}, i = 1, \cdots, n; k_i = 1, \cdots, K_i$. If the depth image is noise free, we shall have:

$$[0\ 0\ 1\ 0] \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M}^d_{ik_i} = 0, \tag{9}$$

which states that if we transform these points to the local coordinate system of each model plane, the Z_i coordinate shall be zero.

Since the depth images are usually noisy, we assume $\mathbf{M}^d_{ik_i}$ follows a Gaussian distribution as:

$$\mathbf{M}^d_{ik_i} \sim \mathcal{N}(\bar{\mathbf{M}}^d_{ik_i}, \mathbf{\Phi}^d_{ik_i}). \tag{10}$$

The log likelihood function can thus be written as:

$$L_2 = -\frac{1}{2\sum_{i=1}^n K_i} \sum_{i=1}^n \sum_{k_i=1}^{K_i} \frac{\varepsilon^2_{ik_i}}{\sigma^2_{ik_i}}, \tag{11}$$

where

$$\varepsilon_{ik_i} = \mathbf{a}_i^T \mathbf{M}^d_{ik_i} \tag{12}$$

where

$$\mathbf{a}_i = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ \mathbf{t}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i & \mathbf{0} \\ -\mathbf{t}_i^T \mathbf{R}_i & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \tag{13}$$

and

$$\sigma^2_{ik_i} = \mathbf{a}_i^T \mathbf{\Phi}^d_{ik_i} \mathbf{a}_i. \tag{14}$$

It is sometimes helpful to have a few corresponding point pairs in the color images and the depth images, as shown in Fig. 1. We denote such point pairs as $(\mathbf{m}_{ip_i}, \mathbf{M}^d_{ip_i}), i = 1, \cdots, n; p_i = 1, \cdots, P_i$. These point pairs shall satisfy:

$$s_{ip_i} \mathbf{m}_{ip_i} = \mathbf{A}[\mathbf{R}\ \mathbf{t}] \mathbf{M}^d_{ip_i}. \tag{15}$$

Whether the point correspondences are manually labeled or automatically established, they may not be accurate. Assume:

$$\mathbf{m}_{ip_i} \sim \mathcal{N}(\bar{\mathbf{m}}_{ip_i}, \mathbf{\Phi}_{ip_i}); \mathbf{M}^d_{ip_i} \sim \mathcal{N}(\bar{\mathbf{M}}^d_{ip_i}, \mathbf{\Phi}^d_{ip_i}), \tag{16}$$

where $\mathbf{\Phi}_{ip_i}$ models the inaccuracy of the point in the color image, and $\mathbf{\Phi}^d_{ip_i}$ models the uncertainty of the 3D point from the depth sensor. The log likelihood function can be written as:

$$L_3 = -\frac{1}{2\sum_{i=1}^n P_i} \sum_{i=1}^n \sum_{p_i=1}^{P_i} \xi_{ip_i}^T \tilde{\mathbf{\Phi}}^{-1}_{ip_i} \xi_{ip_i}, \tag{17}$$

where

$$\xi_{ip_i} = \mathbf{m}_{ip_i} - \mathbf{B}_{ip_i} \mathbf{M}^d_{ip_i}, \tag{18}$$

where

$$\mathbf{B}_{ip_i} = \frac{1}{s_{ip_i}} \mathbf{A}[\mathbf{R}\ \mathbf{t}], \tag{19}$$

and

$$\tilde{\mathbf{\Phi}}_{ip_i} = \mathbf{\Phi}_{ip_i} + \mathbf{B}_{ip_i} \mathbf{\Phi}^d_{ip_i} \mathbf{B}^T_{ip_i}. \tag{20}$$

Combining all the information together, we maximize the overall log likelihood as:

$$\max_{\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{R}, \mathbf{t}} \rho_1 L_1 + \rho_2 L_2 + \rho_3 L_3, \tag{21}$$

where $\rho_i, i = 1, 2, 3$ are weighting parameters. The above objective function is a nonlinear least squares problem, which can be solved using the Levenberg-Marquardt method [6].

3.3. Estimate Other Parameters

There may be a few other parameters that need to be estimated during calibration. For instance, the color camera may exhibit significant lens distortions, thus it is necessary to estimate them based on the observed model planes. Another set of unknown parameters may be in the depth mapping function $\mathbf{f}(\cdot)$. For instance, the structured light-based Kinect depth camera may have a depth mapping function as:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} (\mu z + \nu)(\mathbf{A}^d)^{-1}[u, v, 1]^T \\ 1 \end{bmatrix}, \tag{22}$$

where μ and ν are the scale and bias of the z value, and $\mathbf{A}^d$ is the intrinsic matrix of the depth sensor. Usually $\mathbf{A}^d$ is pre-determined. The other two parameters μ and ν can be used to model the depth sensor's decalibration due to temperature variation or mechanical vibration, and can be estimated within the same maximum likelihood framework.

3.4. Solutions for Initialization

Since the overall likelihood function in Eq. (21) is nonlinear, it is very important to have good initialization for the unknown parameters. For the parameters related to the color camera, i.e., $\mathbf{A}$, $\mathbf{R}_i$ and $\mathbf{t}_i$, we may adopt the same initialization scheme as in [5]. In the following, we discuss methods to provide the initial estimation of the rotation $\mathbf{R}$ and translation $\mathbf{t}$ between the depth and color sensors. During the process, we assume $\mathbf{A}$, $\mathbf{R}_i$ and $\mathbf{t}_i$ of the color camera are known.

3.4.1. Initialization with Model Plane Matching

For most commodity depth cameras, the color camera and the depth camera are positioned very closely. It is therefore simple to automatically identify a set of points in each depth image that lies on the model plane. Let these points be $\mathbf{M}^d_{ik_i}, i = 1, \cdots, n; k_i = 1, \cdots, K_i$. For a given depth image i, if $K_i \geq 3$, it is possible to fit a plane to the points in that image. That is, given:

$$\mathbf{H}_i \begin{bmatrix} \mathbf{n}^d_i \\ b^d_i \end{bmatrix} = \begin{bmatrix} (\mathbf{M}^d_{i1})^T \\ (\mathbf{M}^d_{i2})^T \\ \vdots \\ (\mathbf{M}^d_{iK_i})^T \end{bmatrix} \begin{bmatrix} \mathbf{n}^d_i \\ b^d_i \end{bmatrix} = 0, \tag{23}$$

where $\mathbf{n}_i^d$ is the normal of the model plane in the depth sensor's 3D coordinate system, $\|\mathbf{n}_i^d\|^2 = 1$; and b_i^d is the bias from the origin. $\|\mathbf{n}_i^d\|$ and b_i^d can be easily found through least squares fitting.

In the color sensor's coordinate system, the model plane can also be described by plane equation

$$[0\ 0\ 1\ 0] \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} \mathbf{M} = 0. \quad (24)$$

Since $\mathbf{R}_i$ and $\mathbf{t}_i$ are known, we represent the plane's normal as $\mathbf{n}_i$, $\|\mathbf{n}_i\|^2 = 1$, and bias from the origin b_i.

We first solve the rotation matrix $\mathbf{R}$. Denote:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}. \quad (25)$$

We minimize the following objective function with constraint:

$$J(\mathbf{R}) = \sum_{i=1}^{n} \|\mathbf{n}_i - \mathbf{R}\mathbf{n}_i^d\| + \sum_{j=1}^{3} \lambda_j(\mathbf{r}_j^T \mathbf{r}_j - 1) + 2\lambda_4 \mathbf{r}_1^T \mathbf{r}_2 + 2\lambda_5 \mathbf{r}_1^T \mathbf{r}_3 + 2\lambda_6 \mathbf{r}_2^T \mathbf{r}_3. \quad (26)$$

This objective function can be solved in close form [7]. Let:

$$\mathbf{C} = \sum_{i=1}^{n} \mathbf{n}_i^d \mathbf{n}_i^T. \quad (27)$$

The singular value decomposition of $\mathbf{C}$ can be written as:

$$\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \quad (28)$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{D}$ is a diagonal matrix. The rotation matrix is simply:

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T. \quad (29)$$

The minimum number of images to determine the rotation matrix $\mathbf{R}$ is $n = 2$, provided that the two model planes are not parallel to each other.

For translation, we have the following relationship:

$$(\mathbf{n}_i^d)^T \mathbf{t} + b_i^d = b_i. \quad (30)$$

Thus three non-parallel model planes will determine a unique $\mathbf{t}$. If $n > 3$, we may solve $\mathbf{t}$ through least squares fitting.

3.4.2. *Initialization with Point Pair Matching*

Another scheme to estimate the initial rotation $\mathbf{R}$ and translation $\mathbf{t}$ is through the knowledge of a set of point correspondences between the color images and the depth images. Denote such point pairs as $(\mathbf{m}_{ip_i}, \mathbf{M}_{ip_i}^d), i = 1, \cdots, n; p_i = 1, \cdots, P_i$. We have the relationship:

$$s_{ip_i} \mathbf{m}_{ip_i} = \mathbf{A}[\mathbf{R}\ \mathbf{t}]\mathbf{M}_{ip_i}^d. \quad (31)$$

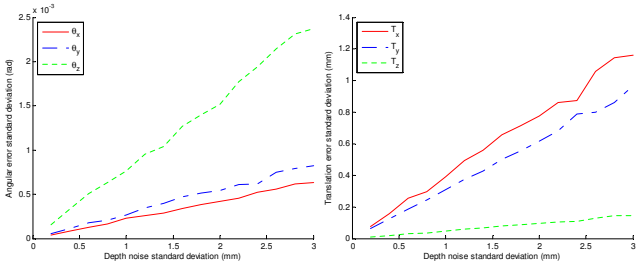

Fig. 3. Calibration accuracy vs. depth camera noise level.

Note the intrinsic matrix $\mathbf{A}$ is known. Such a problem has been studied extensively in the literature [8, 9]. It has been shown that given 3 point pairs, there are in general four solutions to the rotation and translation. When one has 4 or more non-coplanar point pairs, the so-called POSIT algorithm [10] can be used to find the initial value of $\mathbf{R}$ and $\mathbf{t}$.

4. EXPERIMENTAL RESULTS

The maximum likelihood solution in Eq. (21) can be used to calibrate all unknown parameters for the depth and color sensors. Due to space limitations, in this paper we focus our attention on the parameters related to the depth sensor only, i.e., $\mathbf{R}, \mathbf{t}$ and $\mathbf{f}(\cdot)$, and assume $\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i$ are known (or obtained separately from, say, maximizing Eq. (7) only [5]).

4.1. Simulated Results

The simulated depth/color camera has the following parameters. For the color camera, $\alpha = 750, \beta = 745, \gamma = 0, u_0 = 315, v_0 = 245$. The image resolution is 640×480. The rotation and translation from the depth camera to the color camera is represented by vector $[\theta_x, \theta_y, \theta_z, t_x, t_y, t_z]^T = [0.05, -0.01, 0.02, 25, 2, -2]^T$, where $[\theta_x, \theta_y, \theta_z]^T$ in radians represents rotation, which can be converted to $\mathbf{R}$ through the well-known Rodrigues' rotation formula, and the last three elements represent translation $\mathbf{t} = [t_x, t_y, t_z]^T$ in millimeters.

4.1.1. *Performance w.r.t. the Noise Level*

In this experiment we examine the impact of the depth camera's noise level to the calibration accuracy. Three model planes are used in the experiment. The checkerboard pattern has 10×7 corners on a regular grid. The distance between neighboring corners is 37. The three model planes are located at $[\frac{\pi}{8}, 0, 0, -300, 25, 750]^T$, $[0, \frac{\pi}{8}, -\frac{\pi}{18}, -110, -100, 1150]^T$ and $[-\frac{\pi}{36}, 0, \frac{\pi}{6}, 120, -200, 800]^T$, respectively. Only the plane fitting likelihood term (Eq. (11)) is maximized to determine $\mathbf{R}$ and $\mathbf{t}$, where $K_i = 1000$. The covariance of the depth noise is assumed to be independent of the depth values (which is an acceptable assumption for time-of-flight based depth cameras). At each noise level, 500 trials were run and the standard deviations (STDs) of the errors are reported, as

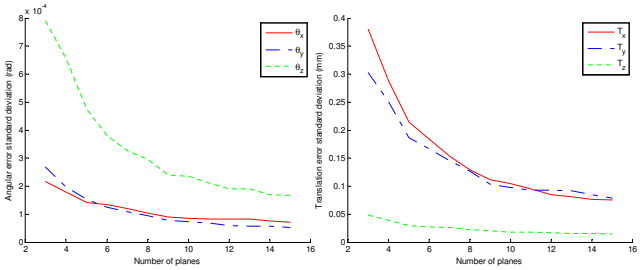

Fig. 4. Calibration accuracy vs. number of model planes.

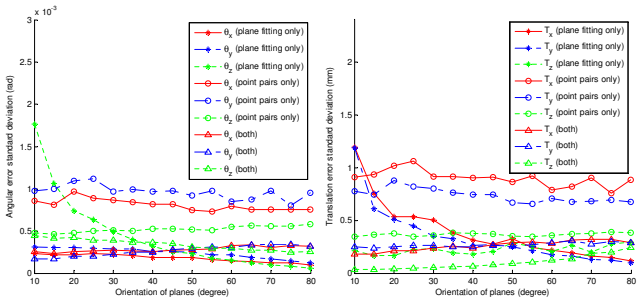

Fig. 5. Calibration accuracy vs. plane orientations.

shown in Fig. 3. It can be seen that the STDs of the angular errors and translation errors increase linearly with the noise level. The mean of the errors are generally very close to zero (not shown in Fig. 3), and the STDs are very small, indicating satisfactory calibration accuracy and algorithm stability.

4.1.2. Performance w.r.t. the Number of Planes

In the second experiment we examine whether increasing the number of model planes could improve calibration accuracy. We tested between 3 and 15 planes, as shown in Fig. 4. The first 3 planes are the same as those in Section 4.1.1. From the fourth image on, we randomly generate a vector on the unit sphere, and apply a rotation with respect to the vector for an angle of $\frac{\pi}{6}$. Again only the plane fitting likelihood term (Eq. (11)) is maximized to determine $\mathbf{R}$ and $\mathbf{t}$. The covariance of the depth noise is set to 1 throughout the experiment. For a given number of planes, we run 500 trials and report the STDs of the errors. From Fig. 4, it is obvious that increasing the number of planes leads to smaller STDs of the errors, thus better calibration accuracy. We recommend at least 8-10 planes to achieve sufficient accuracy during calibration.

4.1.3. Performance w.r.t. the Plane Orientations

Next we study the impact of the model plane orientations. We again use 3 planes for calibration. The planes are generated as follows. We first randomly choose three vectors on the unit circle in the color camera's imaging plane, and we make sure the smallest angle between the three vectors is greater than $\frac{\pi}{9}$. We then apply a rotation with respect to the three vectors for a varying angle between 10° and 80° to generate 3 model planes for calibration. A total of 500 trials were run for each configuration in Fig. 5. It contains three groups of curves.

In the first group, only the plane fitting likelihood term (Eq. (11)) is maximized to determine $\mathbf{R}$ and $\mathbf{t}$. It can be seen that when the plane orientations are small, the calibration errors are big. This is intuitive, since according to Section 3.4.1, parallel planes would not be effective in determining the rotation/translation between the depth and color sensors.

In the second group, we use the point pair likelihood term (Eq. (17)) to determine $\mathbf{R}$ and $\mathbf{t}$. For this purpose, we assume the 4 corners of each model plane is known in both the color and the depth images, thus we have a total of 12 point pairs. Noise of covariance 0.25^2 is added to the position of the point pairs to mimic the real-world scenario. It can be seen from Fig. 5 that with so few point pairs, the calibration error STDs are generally bigger than those generated by planing fitting. An exception is the the cases of very small plane orientations, where the plane fitting only solution performs very poorly.

In the third group, we determine $\mathbf{R}$ and $\mathbf{t}$ based on the combination of plane fitting and point pair likelihood terms. We use $\rho_2 = 1$ and $\rho_3 = 0.2$. It can be seen that for small plane orientations, combining the two likelihood terms results in better performance than using only either. When the plane orientations are large, however, the plane fitting likelihood term alone seem to perform better. In practice, we also need to consider the calibration accuracy of the color camera parameters. It has been shown in [5] that color camera calibration will perform poorly if the model planes are near perpendicular to the color image's imaging plane. Therefore, we recommend to use model planes oriented about 30-50 degrees with respect to the color camera's imaging plane for better overall calibration quality.

4.1.4. Performance w.r.t. the Correct Noise Model

So far we have assumed depth-independent noises in the depth image. This is acceptable for time-of-flight depth cameras such as the ZCam from 3DV systems. However, for triangularization based depth cameras such as the Kinect camera, the noise level is a quadratic function of the depth [3]. Both types of noises can be accommodated with our maximum likelihood solution in Section 3. On the other hand, applying the correct noise model would generally improve calibration performance. Here we use the same setup as Section 4.1.1, except that the depth noise follows the formula in [3]:

$$\sigma \propto Z^2, \tag{32}$$

and we assume the noise level at $Z = 1000$ is known (adapting as the horizontal axis of Fig. 6). We ran two sets of experiments. In the first set, we assume the user is unaware of the sensor noise type, and blindly assume that the noise is depth-independent. In the second set, the correct noise model is applied. It can be seen from Fig. 6 that using the correct noise model results in better calibration performance.

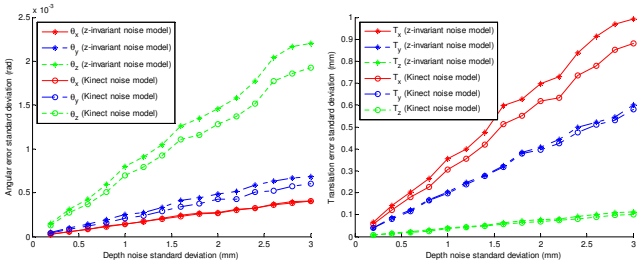

Fig. 6. Calibration accuracy vs. correct noise model.

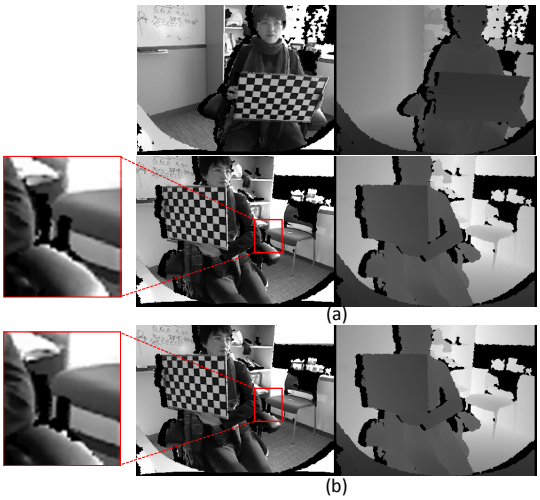

Fig. 7. Calibration results for a real scene. (a) Use plane fitting only. (b) Use both plane fitting and point pairs.

4.2. Real-World Results

Finally, we test the proposed method on a real Kinect sensor. A set of 12 model planes at different positions and orientations are captured. One of the image pairs has been shown in Fig. 1. The color sensor's intrinsic parameters are first estimated with [5] as $\alpha = 528.32, \beta = 527.03, \gamma = 0, u_0 = 320.10, v_0 = 257.57$. We then apply the proposed technique to calibrate $\mathbf{R}, \mathbf{t}$ and $\mathbf{f}(\cdot)$, where $\mathbf{f}(\cdot)$ contains unknown parameters such as the depth scale and bias μ, ν. The depth camera's intrinsic matrix $\mathbf{A}^d$ is pre-set to $\alpha^d = 575, \beta^d = 575, \gamma = 0, u_0^d = 320, v_0^d = 240$.

The calibration results with plane fitting alone are $[0.0058, -0.0049, 0.0057, -16.4411, 21.077, 5.4074]^T$ for rotation and translation, $\mu = 0.9771, \nu = 16.1883$ for depth scale and bias. To demonstrate the calibration accuracy, we warp the color images based on the calibrated parameters, and overlay them onto the depth images to examine how well they align with each other, as shown in Fig. 7 (a). It can be seen that the alignment is very accurate.

As shown in Section 4.1.3, adding additional point correspondences between the color and the depth images may help improve calibration performance when the model planes do not have sufficient variations in orientation. Another benefit of adopting point correspondences is to expand the calibration effective zone. It is well known that for calibration to work well, the checkerboard shall be placed across the whole workspace. In Fig. 7 (a), we notice the chair region is not aligned well, since no checkerboard was placed there in the 12 images. We manually add 3-5 point correspondences for each image pair, some of them lying on the background of the scene (see Fig 1). After using both plane fitting and point correspondences to calibrate, the results are $[-0.0089, -0.0071, 0.004, -15.7919, 8.2644, 12.5897]^T$ for rotation and translation, $\mu = 0.9948, \nu = -6.5470$ for depth scale and bias. A warped result is shown in Fig. 7 (b). The improvement in the chair area is very obvious.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel algorithm to calibrate color and depth sensors jointly. The method is reliable and accurate, and it does not require additional hardware other than the easily available checkerboard pattern. Future work includes better modeling of the depth mapping function $\mathbf{f}(\cdot)$, and better understanding of the depth cameras' noise models.

6. REFERENCES

[1] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3D shape scanning with a time-of-flight camera," in *CVPR*, 2010.

[2] R. Crabb, C. Tracey, A. Puranik, and J. Davis, "Real-time foreground segmentation via range and color imaging," in *CVPR Workshop on ToF-Camera based Computer Vision*, 2008.

[3] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3d deformable face tracking with a commodity depth camera," in *ECCV*, 2010.

[4] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lens," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 323–344, 1987.

[5] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[6] J. More, "The levenberg-marquardt algorithm: implementation and theory," *Numerical Analysis, Lecture Notes in Mathematics*, vol. 630/1978, pp. 105–116, 1978.

[7] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. PAMI*, vol. 9, no. 5, pp. 698–700, 1987.

[8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] J. S.-C. Yuan, "A general photogrammetric method for determining object position and orientation," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 2, pp. 129–142, 1989.

[10] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1, pp. 123–141, 1995.