Fast and Accurate Calibration of a Kinect Sensor

Carolina Raposo
Institute of Systems and Robotics
Dept. of Electrical and Comp. Eng.
University of Coimbra
3030-290 Coimbra, Portugal
carolinasraposo@gmail.com

João Pedro Barreto
Institute of Systems and Robotics
Dept. of Electrical and Comp. Eng.
University of Coimbra
3030-290 Coimbra, Portugal
jpbar@isr.uc.pt

Urbano Nunes
Institute of Systems and Robotics
Dept. of Electrical and Comp. Eng.
University of Coimbra
3030-290 Coimbra, Portugal
urbano@isr.uc.pt

***Abstract*—The article describes a new algorithm for calibrating a Kinect sensor that achieves high accuracy using only 6 to 10 image-disparity pairs of a planar checkerboard pattern. The method estimates the projection parameters for both color and depth cameras, the relative pose between them, and the function that converts *kinect disparity units (kdu)* into metric depth. We build on the recent work of Herrera *et. al* [8] that uses a large number of input frames and multiple iterative minimization steps for obtaining very accurate calibration results. We propose several modifications to this estimation pipeline that dramatically improve stability, usability, and runtime. The modifications consist in: (i) initializing the relative pose using a new minimal, optimal solution for registering $3D$ planes across different reference frames; (ii) including a metric constraint during the iterative refinement to avoid a drift in the disparity to depth conversion; and (iii) estimating the parameters of the depth distortion model in an open-loop post-processing step. Comparative experiments show that our pipeline can achieve a calibration accuracy similar to [8] while using less than $1/6$ of the input frames and running in $1/30$ of the time.**

***Keywords*-Kinect; Camera Calibration, RGB-Depth Camera Pair**

I. Introduction

Nowadays, the joint information provided by cameras and depth sensors has applications in areas including scene reconstruction, indoor mapping, and mobile robotics. The Kinect is a camera pair capable of providing such information. Its depth sensor consists of a projector that emits a dot pattern which is detected by an infrared (IR) camera. The Kinect has been used for multiple purposes including 3D modeling of indoor environments [6], and Structure from Motion [12]. Most of these applications require the camera pair to be calibrated both intrinsically and extrinsically. The intrinsic calibration consists in determining the parameters that enable to convert measurement units into metric units. The extrinsic calibration consists in locating the sensors in a common coordinate frame, for them to function as a whole.

The literature about color camera calibration is vast with the methods that use a planar checkerboard pattern being specially popular because they are stable, accurate, and the calibration rig is easy to build [15], [1]. For depth sensors, calibration methods depend on the technology used, whether they are time-of-flight (ToF) cameras, laser range scanners, or structured light scanners. Methods which use color discontinuities [4], or planar surfaces [14], [12], [7], [8], [13] have been developed.

In this work, we build on the recent work of Herrera *et. al* [8] that uses image-disparity map pairs of planes to accurately calibrate a Kinect device. They use tens of images to estimate the intrinsic parameters of the color and depth cameras, as well as their relative pose. The method relies on multiple iterative optimization steps that take minutes to complete. We propose several modifications to this calibration pipeline that improve stability, and dramatically decrease the number of input images and runtime. The experiments show that our method is able to accomplish similar accuracy to [8], using as few as 6-10 images, as opposed to 60 images, and running in 20-30 sec, instead of 15 min.

A. Related work

Kinect is a device for the consumer market of games and entertainment. The intrinsic parameters of both depth and color cameras, as well as their relative pose, are pre-calibrated in factory and recorded in the firmware. Average values for these parameters are known by the community and commonly used in robotic applications [3]. However, it is well known that these parameters vary from device to device, and that the factory presets are not accurate enough for many applications [6], [12]. This justifies the development of calibration methods for the Kinect, or of methods to refine and improve the accuracy of the factory presets.

Authors have tried to independently calibrate the intrinsics of the depth sensor and color camera, and then register both in a common reference frame [11], [13]. As pointed out by Herrera *et. al* [8], the depth and the color camera must be calibrated together both because the accuracy in the color camera propagates to the depth camera, and because all available information is being used.

Depth sensors may present depth distortions which decreases their accuracy. This is the case of the Kinect device which has shown radially symmetric distortions [12] that are not corrected in the manufacturer's calibration. Herrera

et. al [7] firstly proposed an algorithm that calibrates not only the cameras' intrinsics, but also the parameters that convert kdu into meters. Zhang and Zhang extend this work by considering point correspondences between color and depth images, showing improved accuracy. However, neither methods deal with the depth distortion.

Smisek *et. al* [12] observed that the Kinect exhibited residuals for close range measurements, and were the first to propose considering both distortion in the projection and in the depth estimation. The depth distortion was estimated for each pixel by averaging the metric error, after carrying the intrinsic and extrinsic calibrations of the device. More recently, another depth distortion correction procedure was proposed by Herrera *et. al* [8], which leads to improved accuracy. They initially estimate the intrinsics and the plane pose, from homographies computed using plane-to-image correspondences. The extrinsic calibration is carried by registering the 3D planes estimated in color and depth camera coordinates. Due to the high number of parameters to be optimized, they use an iterative refinement step that optimizes the parameters alternately. Unfortunately, in order to effectively model the depth camera parameters, including the distortion term, it requires many images ($\geq$ 20). Also, its iterative optimization step is highly time consuming.

B. Overview of the approach

Since the recent work of Herrera *et. al* [8] is the one that reports better accuracies, we build on their contribution and downsize the calibration pipeline to improve the usability.

As in [8], the color camera is calibrated from plane-to-image homographies which enable to know the pose of the calibration plane in the color camera reference frame. Concerning the depth camera, we use the preset values to reconstruct 3D points, and compute the calibration plane pose using a standard fitting algorithm. Computation of the extrinsic calibration is accomplished by performing plane registration. While Herrera carries the registration using a sub-optimal linear algorithm, we describe a new method based on [10]. It is a minimal solution that is included in a random sample consensus (RANSAC) framework. This provides better initializations of the relative pose, which facilitate the subsequent steps. All parameters are optimized in a bundle adjustment step which considers metric information in order to avoid a drift in the disparity to depth conversion. We use the depth distortion model presented in [8], which has shown to yield good accuracy. However, unlike Herrera's method, we estimate the model's parameters in an open-loop, making our approach much less time consuming.

This pipeline leads to improvements in usability without sacrificing the final accuracy. The improvements are both in terms of decreasing the number of input images by a factor of 6, and reducing the computational time by a factor of 30. Our method, as Herrera's method, can be used with more than one color camera. However, in this work, we only consider the Kinect's color camera in the experiments.

Notation: Scalars are represented by plain letters, *e.g.* x, vectors are indicated by bold symbols, *e.g.* $\mathbf{x}$, and matrices are denoted by letters in sans serif font, *e.g.* R. Planes are represented by a 4D homogeneous vector that is indicated by an uppercase Greek letter, *e.g.* Π. Sets of intrinsic parameters are defined by uppercase calligraphic letters, *e.g.* $\mathcal{I}$. Subscripts c and d attached to symbols refer to the color and depth cameras, respectively.

II. Background

A. Projection model

In this work, the intrinsic parameters of the color camera are modeled as in [5], where both radial and tangential distortions are considered. Let $\mathbf{X}_c = [X_c, Y_c, Z_c]^\mathsf{T}$ be a 3D point in the camera reference frame. The normalized image projection of $\mathbf{X}_c$ is $\mathbf{x}_n = [x_n, y_n]^\mathsf{T}$, with $x_n = X_c/Z_c$ and $y_n = Y_c/Z_c$. Including lens distortion, it comes that

$$\mathbf{x}_k = (1 + k_{c1}r^2 + k_{c2}r^4 + k_{c5}r^6)\mathbf{x}_n + \mathbf{d}_x, \qquad (1)$$

where $r^2 = x_n^2 + y_n^2$ and $\mathbf{d}_x$ is the tangential distortion. The pixel coordinates $\mathbf{x}_c = [x_c, y_c]^\mathsf{T}$ of the projection of $\mathbf{X}_c$ on the image plane are

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} f_{cx} & 0 \\ 0 & f_{cy} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} c_{cx} \\ c_{cy} \end{bmatrix}, \qquad (2)$$

where $\mathbf{f}_c = [f_{cx}, f_{cy}]$ are the focal lengths, and $\mathbf{c}_c = [c_{cx}, c_{cy}]$ is the principal point. We refer to the set of intrinsic parameters of the color camera by $\mathcal{I}_c = \{\mathbf{f}_c, \mathbf{c}_c, \mathbf{k}_c\}$.

The pixel coordinates of the projection of the 3D point $\mathbf{X}_d$ in depth camera coordinates can be obtained using a model similar to the color camera's one. The parameters $\mathbf{f}_d$ and $\mathbf{c}_d$ are the focal length and the principal point of the depth camera, respectively. Considering the distortion in the depth camera does not improve accuracy significantly. Thus, $\mathbf{k}_d$ is set to zero.

B. Depth measurements

The Kinect's depth sensor consists of an IR camera which detects a constant pattern emitted by a projector. It delivers depth information in disparity units (kdu) which must be converted into metric units (meters). This can be done by using a scaled inverse of the format

$$z = \frac{1}{c_1 d_u + c_0}, \qquad (3)$$

where c_0 and c_1 are part of the depth camera's intrinsics. Depth z is obtained from d_u, which is the undistorted disparity, *i.e.*, after performing distortion correction. The Kinect's depth sensor presents a depth distortion which has been modeled by Herrera *et. al* [8]:

$$d_u = d + \mathsf{D}(x_d, y_d)e^{\alpha_0 - \alpha_1 d}, \qquad (4)$$

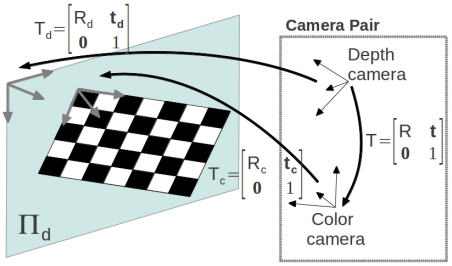


Figure 1. The color and depth cameras are related by a rigid transformation $(\mathsf{R}, \mathbf{t})$. Both sensors observe the same planar surface, allowing the computation of the extrinsic calibration.

where d is the disparity returned by the Kinect in pixel $[x_d, y_d]$, D contains the spatial distortion pattern, and $\alpha = [\alpha_0, \alpha_1]$ models the decay of the distortion effect. We refer to the set of intrinsic parameters of the depth camera by $\mathcal{I}_d = \{\mathbf{f}_d, \mathbf{c}_d, \mathbf{k}_d, c_0, c_1, \mathsf{D}, \alpha\}$.

C. Herrera's method

Herrera *et. al* [8] recently proposed a new method for calibrating a color-depth camera pair, as well as a new explicit distortion correction term for the Kinect device, which significantly improves accuracy. They use a setup identical to Figure 1, where all cameras observe a planar checkerboard pattern from multiple views, which are used for calibrating the sensors. In this section, we review Herrera's method for which we show a diagram in Figure 2. It can be seen that the initialization steps can be performed by two different methods, yielding two versions of the method to which we refer by Herrera and Herrera I. The remaining steps do not depend on how the initial estimate was obtained, and constitute the non-linear minimization.

1) Initial estimation: The color camera intrinsics can be initialized using Zhang's method [15]. The checkerboard corners are extracted from the intensity images and, using known corner positions in the checkerboard reference frame, both the intrinsic parameters and the plane to image homographies can be estimated. This leads to the initialization of $\mathcal{I}_c$ and $\mathsf{T}_c^{(i)}$, for all input images i.

The same method can be applied to estimate the depth camera parameters and homographies using plane corners [8]. From these initial parameters, it is possible to obtain an estimate for the expected depth of each selected corner. The corresponding measured disparities can be used for determining an initial guess for c_0 and c_1, using equation 3. Thus, by setting D and α to zero, an initialization of $\mathcal{I}_d$ and $\mathsf{T}_d^{(i)}$ is obtained. This initialization procedure is used in Herrera's method, as depicted in Figure 2. However, it produces a very rough initial estimate, especially if the number of calibration planes is small. Thus, since there exist publicly available values for the intrinsics of the Kinect device, in method Herrera I these are used, and the extrinsic calibration step is skipped since estimates for $\mathcal{I}_d$ and T are known.

2) Extrinsic calibration: In method Herrera, it is necessary to explicitly compute the relative pose between the sensors T. From Figure 1, it is evident that the checkerboard and calibration plane reference frames are not aligned, and thus there is not a common reference frame between the two sensors. This means that is it not possible to find T by simply chaining transformations $\mathsf{T}_c^{(i)}$ and $\mathsf{T}_d^{(i)}$. However, T can be found through plane registration, since it is known that both planes are coplanar.

Given $\mathsf{T}_c^{(i)}$, $\Pi_c^{(i)}$ can be obtained by computing

$$\Pi_c^{(i)} = \begin{bmatrix} \mathbf{r}_{c3} \\ \mathbf{r}_{c3}^{\mathsf{T}} \mathbf{t}_c \end{bmatrix}, \tag{5}$$

where $\mathbf{r}_{c3}$ is the third column of matrix $\mathsf{R}_c^{(i)}$. For finding $\Pi_d^{(i)}$, we proceed similarly. The registration problem is the one of estimating R and $\mathbf{t}$ such that

$$\Pi_d^{(i)} \sim \underbrace{\begin{bmatrix} \mathsf{R} & \mathbf{0} \\ -\mathbf{t}^{\mathsf{T}}\mathsf{R} & 1 \end{bmatrix}}_{\mathsf{T}^{-\mathsf{T}}} \Pi_c^{(i)}, i = 1, 2, 3 \tag{6}$$

verifies. Herrera *et. al* use a linear sub-optimal algorithm to carry this estimation.

3) Non-linear minimization: The non-linear minimization of Herrera's method consists of 3 steps, as shown in the diagram of Figure 2. It aims to minimize the weighted sum of squares of the measurement reprojection errors over all parameters ($\mathcal{I}_c$, $\mathcal{I}_d$, T, and T_c for all calibration images). For the color camera, the error is the Euclidean distance between the measured corner position $\hat{\mathbf{x}}$ and its reprojected position $\mathbf{x}$ (first term of equation 7). For the depth camera it is the difference between the measured disparity $\hat{d}$ and the predicted disparity d. The errors are normalized using each measurement variance σ^2. It comes that the cost function is

$$c = \frac{\sum ||\hat{\mathbf{x}}_c - \mathbf{x}_c||^2}{\sigma_c^2} + \frac{\sum (\hat{d} - d)^2}{\sigma_d^2}. \tag{7}$$

The optimization process is divided into three steps: firstly, only $\mathcal{I}_d$ and T are optimized to account for the fact that they are poorly initialized; secondly, equation 7 is minimized over all the parameters, except for D; lastly, D is optimized independently for each pixel. The two last steps are repeated until convergence is reached.

III. Calibration method

We propose a new calibration method that consists of four main consecutive steps: an initial estimation of the intrinsic and extrinsic parameters, a non-linear minimization, and a depth distortion model estimation. Figure 2 shows a block diagram of our method, which presents a simpler framework than Herrera's. Our optimization procedure consists of only one step, and a depth distortion model is estimated using the optimized parameters.

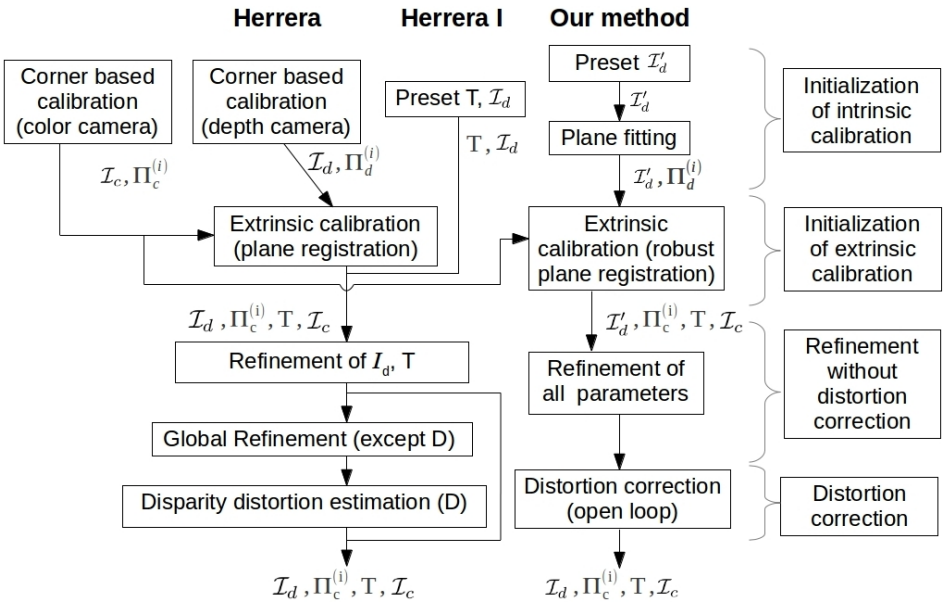


Figure 2. Calibration algorithms: two versions of Herrera's method (named Herrera and Herrera I), and our method.

A. Initialization of intrinsic calibration

For the color camera, the initial estimation of $\mathcal{I}_c$ and $\mathsf{T}_c^{(i)}$ for all calibration images is done as described in section II-C1, for which we use Bouguet's toolbox [2]. We redefine the intrinsic parameters of the depth camera as $\mathcal{I}'_d = \{\mathbf{f}_d, \mathbf{c}_d, \mathbf{k}_d, c_0, c_1\}$ because we do not consider depth distortion terms. They are initialized using preset values, which are publicly available for the Kinect [3].

B. Initialization of extrinsic calibration

For each input disparity map i, the plane corners are extracted, defining a polygon. For each point $\mathbf{x}_d$ inside the polygon, the corresponding disparity d is used for computing a depth value z_d using equation 3, where $d = d_u$ since the measured disparities are used. The correspondences (x_d, y_d, z_d) are used for computing 3D points $\mathbf{X}_c$ originating a 3D point cloud. To each 3D point cloud, a plane is fitted using a standard total least squares algorithm.

Plane registration in the dual space

Consider two sets of three planes $\Pi_c^{(i)}$ and $\Pi_d^{(i)}, i = 1, 2, 3$, in color and depth camera reference frames, respectively, in homogeneous representation $\Pi_c^{(i)} \sim [\mathbf{n}_{ci} \quad 1]^\mathsf{T}$ (and equivalent for $\Pi_d^{(i)}$). Knowing that points and planes are dual entities in 3D - a plane in the projective space $\mathcal{P}^3$ is represented as a point in the dual space $\mathcal{P}^{3*}$, and vice-versa - equation (6) can be seen as a projective transformation in $\mathcal{P}^{3*}$ that maps points $\Pi_c^{(i)}$ into points $\Pi_d^{(i)}$. Figure 3 illustrates the problem in the dual space $\mathcal{P}^{3*}$ with origin $\mathcal{O}^*$. The transformation $\mathsf{T}^{-\mathsf{T}}$ can be factorized into a rotation

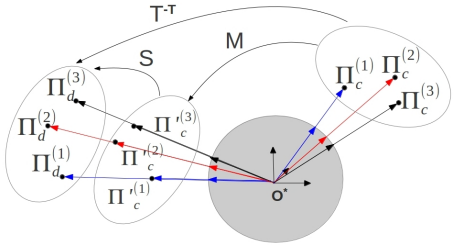


Figure 3. Interpretation of the registration problem in the dual projective space $\mathcal{P}^{3*}$. The factorization $\mathsf{T}^{-\mathsf{T}} \sim \mathsf{SM}$ allows the rotation and translation components to be computed separately.

transformation M, mapping points $\Pi_c^{(i)}$ into points $\Pi'^{(i)}_c$, and a projective scaling S that maps points $\Pi'^{(i)}_c$ into points $\Pi_d^{(i)}$:

$$\lambda_i \Pi_d^{(i)} = \underbrace{\begin{bmatrix} \mathsf{I}_3 & \mathbf{0} \\ -\mathbf{t}^\mathsf{T} & 1 \end{bmatrix}}_{\mathsf{S}} \underbrace{\begin{bmatrix} \mathsf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathsf{M}} \Pi_c^{(i)}, \tag{8}$$

where I_3 is the 3×3 identity matrix and λ_i is an unknown scale factor. M can be computed from $N = 2$ point-point correspondences, but S requires $N = 3$ point-point correspondences to be estimated. An easy two-step process to perform the registration is presented:

1) Since the length of a vector is not changed by rotation, we normalize $\mathbf{n}_{ci}$ and $\mathbf{n}_{di}$, obtaining $\vec{\mathbf{n}}_{ci}$ and $\vec{\mathbf{n}}_{di}$. The normalized vectors are represented by the vectors in Figure 3 inside the sphere of radius 1. Next, we apply the algorithm from [9] for computing a transformation between two sets of unitary vectors.

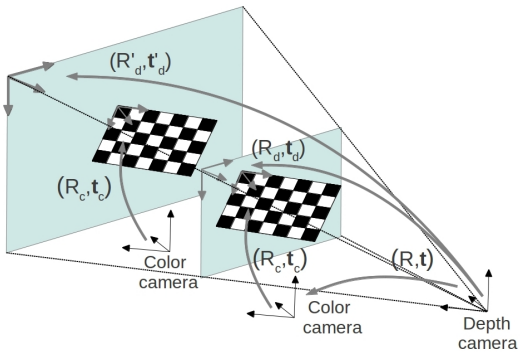


Figure 4. The problem of occurring a drift in scale. The pose of grid in the color camera reference frame is fixed, while the depth camera may observe the calibration plane at different depths.

2) From equation 8 we can write

$$\mathbf{n}_{di}^{\mathsf{T}}\mathbf{n}_{di}\mathbf{n}_{ci}^{\mathsf{T}}\mathsf{R}^{\mathsf{T}}\mathbf{t} - \mathbf{n}_{di}^{\mathsf{T}}\mathbf{n}_{di} + \mathbf{n}_{di}^{\mathsf{T}}\mathsf{R}\mathbf{n}_{ci} = 0. \quad (9)$$

Each pair $\Pi_c^{(i)}$, $\Pi_d^{(i)}$ gives rise to a linear constraint in the entries of the translation vector $\mathbf{t}$, which can be computed by $\mathbf{t} = \mathsf{A}^{-1}\mathbf{b}$ with

$$\mathsf{A} = \begin{bmatrix} \mathbf{n}_{d1}^{\mathsf{T}}\mathbf{n}_{d1}\mathbf{n}_{c1}^{\mathsf{T}} \\ \mathbf{n}_{d2}^{\mathsf{T}}\mathbf{n}_{d2}\mathbf{n}_{c2}^{\mathsf{T}} \\ \mathbf{n}_{d3}^{\mathsf{T}}\mathbf{n}_{d3}\mathbf{n}_{c3}^{\mathsf{T}} \end{bmatrix} \mathsf{R}^{\mathsf{T}}, \mathbf{b} = \begin{bmatrix} \mathbf{n}_{d1}^{\mathsf{T}}\mathbf{n}_{d1} - \mathbf{n}_{d1}^{\mathsf{T}}\mathsf{R}\mathbf{n}_{c1} \\ \mathbf{n}_{d2}^{\mathsf{T}}\mathbf{n}_{d2} - \mathbf{n}_{d2}^{\mathsf{T}}\mathsf{R}\mathbf{n}_{c2} \\ \mathbf{n}_{d3}^{\mathsf{T}}\mathbf{n}_{d3} - \mathbf{n}_{d3}^{\mathsf{T}}\mathsf{R}\mathbf{n}_{c3} \end{bmatrix}. \quad (10)$$

This plane registration algorithm provides the extrinsic calibration of a camera and a depth sensor in the case of $N = 3$ correspondences. For $N > 3$ pairs of planes, each triplet of plane-plane correspondences gives rise to one solution, and the best estimation can be found using an hypothesize-and-test framework:

1) For each possible triplet of pairs of planes $\Pi_c^{(i)}$, $\Pi_d^{(i)}$, find transformation T.
2) For each solution T, compute the depth camera coordinates Π_{dj} for all $\Pi_c^{(i)}$ using (6), and determine the euclidean distance l_j in the dual space between the computed Π_{dj} and $\Pi_d^{(i)}$.
3) Each T is ranked by $rank(\mathsf{T}) = \sum_j \max(t, l_j)$, where t is a predefined threshold. The correspondences for which $l_j < t$ are considered as inliers and T for which $rank(\mathsf{T})$ is minimum is the pose estimation.

Only the inlier correspondences are used for optimization.

C. Non-linear minimization

We observed experimentally that under poor initialization and a small number of images, Herrera's method tends to drift in depth. After careful analysis, we came up with an hypothesis for this observation. Figure 4 depicts the problem. From equation 3, it can be seen that if c_0 and c_1 are affected by a scale component, this will reveal in a depth scaling, which consequently originates a shift in the z component of $\mathbf{t}_d$. Note that the rotation component is not affected, *i.e.*, $\mathsf{R}'_d = \mathsf{R}_d$ in Figure 4. This does not change the reprojection error in a given pixel because the expected disparity in that pixel is the same. Since T_c remains unchanged, the translation between the two sensors $\mathbf{t}$ is also shifted, originating an error in the extrinsic calibration.

Thus, we change the cost function 7 by adding a term that accounts for the difference between the Euclidean distances between points of an object λ and the measured distances between those points $\hat{\lambda}$. Our objective function is, then,

$$\min_{\mathcal{I}_c, \mathcal{I}'_d, \mathsf{T}, \mathsf{T}_{ci}} e = \frac{\sum ||\hat{\mathbf{x}}_c - \mathbf{x}_c||^2}{\sigma_c^2} + \frac{\sum (\hat{d} - d)^2}{\sigma_d^2} + \beta|\hat{\lambda} - \lambda|^2, \quad (11)$$

where β is a weighting factor which should be sufficiently high. This information could be included as a hard constraint. However, since we do not know how accurate the measurements are, we decided to include it as a penalty term. This means that our algorithm requires at least one image of an object with known dimensions, for avoiding the calibration to drift in scale.

D. Depth distortion model estimation

The optimized intrinsic and extrinsic calibrations can be used for estimating the depth distortion model of equation 4. Note that it can be rewritten as

$$d_u = d + \mathsf{W}(x_d, y_d)e^{-\alpha_1 d}, \quad (12)$$

where $\mathsf{W}(x_d, y_d) = \mathsf{D}(x_d, y_d)e^{\alpha_0}$.
For a pair of disparity maps where a given pixel $\mathbf{x}_d$ belongs to the calibration plane in both maps, there are two correspondences $(\tilde{d}_1, d_1)$ and $(\tilde{d}_2, d_2)$, where d is the measured disparity and $\tilde{d}$ is the expected disparity computed by knowing the plane equation. Using the two correspondences, we can write the system of equations

$$\begin{cases} \tilde{d}_1 - d_1 = \mathsf{W}(x_d, y_d)e^{-\alpha_1 d_1} \\ \tilde{d}_2 - d_2 = \mathsf{W}(x_d, y_d)e^{-\alpha_1 d_2} \end{cases} \quad (13)$$

and find α_1 by

$$\alpha_1 = \frac{\ln \frac{\tilde{d}_1 - d_1}{\tilde{d}_2 - d_2}}{d_2 - d_1}. \quad (14)$$

For every possible pair of correspondences, we compute an estimate for α_1 and consider their average as the final result.

Knowing α_1, W can directly be estimated for the pixels which belong to a known plane. For pixel (x_d, y_d), if more than one value is found, the average of all values is considered. Although it is not possible to find individual estimates for α_0 and D, this method allows to recover the whole depth distortion function. Like Smisek *et. al* [12], we perform the estimation in open-loop. However, since we use Herrera's model, we obtained better accuracy.

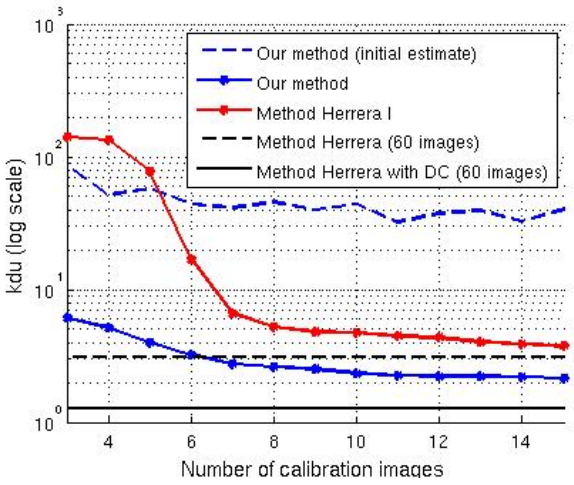
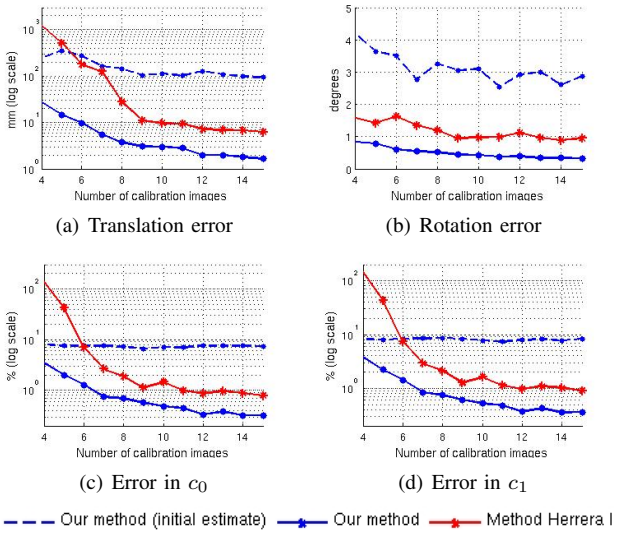


Figure 5. Average RMS reprojection errors in kdu obtained with the validation set of 10 images. All calibrations were performed without distortion correction (DC), except for one using a data set with 60 plane poses (pseudo ground truth).



(a) Translation error (b) Rotation error

(c) Error in c_0 (d) Error in c_1

Figure 6. Errors relative to the pseudo ground truth obtained without performing distortion correction.

IV. Experimental results

Two sets of experiments were conducted in order to compare the accuracy of Herrera's method, which has been released as a toolbox, and our method. The first one uses the data set included in the toolbox, and shows extensive results with a varying number of calibration images. The second set uses a small number of images acquired by another Kinect, in order to further validate the results.

A. Herrera's data set

The dataset comprises image-disparity map pairs for 70 distinct plane poses, with the images being both acquired by the Kinect's color camera and an external high resolution camera. We selected 10 image-disparity map pairs acquired by the Kinect (validation set) and used the rest of the data as input to the original Herrera algorithm, that was executed with and without distortion correction (DC). Figure 5 shows the reprojection error measured for the validation set, where it can be seen that the latter is substantially more accurate than the former. We will consider this last calibration result as being close to the ground truth, and refer to it as *pseudo ground truth*, given the large amount of data and the use of a high resolution camera. However, it is merely indicative, since we do not know how exact the calibration is. The estimations by the different methods are compared against this one. From the test set we selected 20 image-disparity map pairs acquired by the Kinect. These pairs were grouped in sets of $K = 4, 5, \ldots, 15$, and we randomly picked 50 sets for each value of K. For each group of input images, we ran the calibration using Herrera I and our method. The initial values were sampled from a uniform distribution with amplitude equal to 5% of the original value. The idea was to evaluate the robustness to poor initialization.

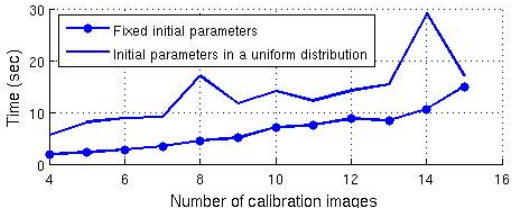
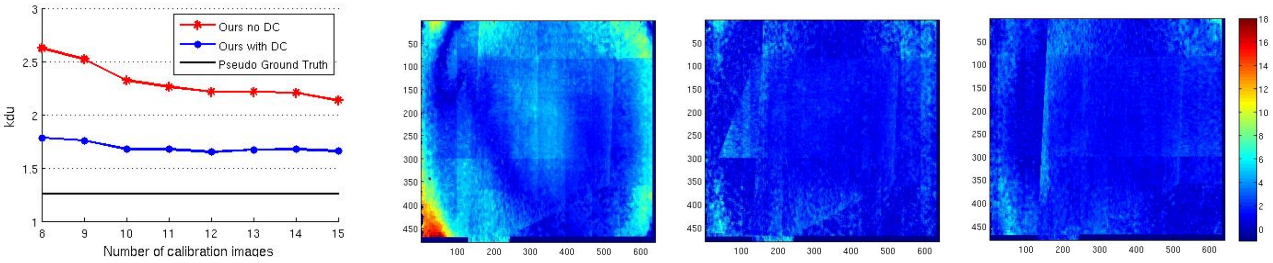


Figure 7. Average run times obtained with our method, for increasing number of calibration images.

For each trial, we evaluated the result in terms of reprojection error, using the 10 validation images, and in terms of extrinsics, by comparing with the pseudo ground truth. Figures 5 and 6 show the average errors for increasing number of K input images. Results clearly show that under the same conditions, our method systematically outperforms Herrera's method, which is not capable of producing acceptable results with small data sets (≤ 8 calibration images). Our method, on the other hand, yields good results with only 6 calibration images. Although the initial estimates are very poor, both methods are capable of converging during the optimization phase. Figure 7 shows the average run times of our method, when using fixed initial parameters or parameters sampled from a uniform distribution. When the parameters are not fixed, a poorer initial estimation may be obtained, leading to higher run times in the optimization step. However, this is a low time consuming method since it never exceeds 30 seconds.

Using the results obtained with calibration sets of more than 7 images, we estimated the depth distortion model with 2 images of a wall at different depths. The average



(a) Average results (b) Our without DC (c) Our with DC (d) Pseudo Ground Truth

Figure 8. Results obtained with Herrera's data set. (a) Average and per-pixel RMS reprojection errors obtained with the validation set for (b) our method without distortion correction (DC), (c) our method with DC, and (d) the pseudo groud truth.

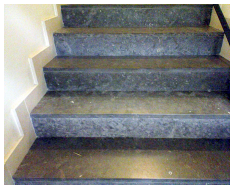


Figure 9. Image (acquired with our Kinect) of the stairs used for 3D reconstruction.

	Our method	Method Herrera I
No DC	0.495°	0.743°
DC	0.369°	0.602°

Table I
AVERAGE ANGULAR ERROR BETWEEN ALL POSSIBLE PAIRS OF 10 RECONSTRUCTED PLANES.

	Our method	Method Herrera I
No DC	1.54 kdu	4.08 kdu
With DC	1.20 kdu	3.61 kdu

Table II
AVERAGE RMS REPROJECTION ERRORS OBTAINED WITH THE VALIDATION SET OF 6 IMAGES ACQUIRED BY OUR KINECT.

RMS reprojection errors for the validation images are shown in Figure 8(a). It can be seen that the model was correctly estimated since the reprojection errors significantly decreased. This can be confirmed in Figure 8 where the average reprojection errors obtained in each pixel, for the 10 validation images, are shown. It can be seen that before correcting the distortion, a radial pattern of the residuals is observed. After applying the distortion correction, the reprojection errors significantly decrease, and the pattern obtained becomes very similar to the pseudo ground truth's. The estimation of the model with 2 images takes about 10 seconds, so that the overall run time is of about 30 seconds for 15 calibration images. Herrera's method, however, is much more time consuming, taking about 3 minutes with 20 images.

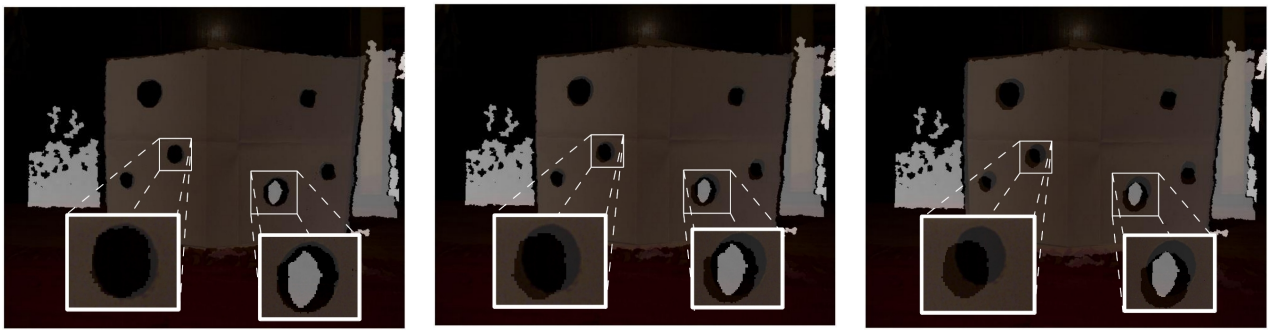
B. Our data set

In this set of experiments, we acquired a data set of 14 images, of which 8 were used for calibration and 6 for validation. We used the 8-image data set for calibrating the camera pair with ours and Herrera I method, both with and without distortion correction. Note that our estimation of the depth distortion model is done with the 8 images of the calibration set. The quality of the depth camera's intrinsic calibration is assessed by reconstructing the planes of a flight of perpendicular stairs (Figure 9), and computing the angles between all possible pairs of planes. These are compared with 90° if the planes are orthogonal, and 0° if they are parallel. Results in Table I show that, although both methods perform well, ours yields smaller angular errors in average. Applying distortion correction leads to a more accurate reconstruction in both cases.

Average RMS reprojection errors were computed for the validation set and results are shown in Table II. As expected, our method outperforms Herrera's since the calibration set is not large enough for it to produce good results. Although using distortion correction leads to an improvement in the accuracy for both methods, Figure 10 shows that in Herrera's method, it leads to a poorer extrinsic calibration. The 3D points computed from the disparity image are represented in color camera coordinates, to which colors are assigned. A correct calibration should align the intensity image with the depth map. Results with our method show that the misalignment is very slight, while for Herrera's method it is significant, and is larger when using distortion correction. This indicates that Herrera's method is not able to properly model the depth distortion with small data sets.

V. CONCLUSION

We present a new method for calibrating a color-depth camera pair which outperforms one of the state-of-the-art methods, when using small data sets. Our main contributions are a new optimization step that prevents the calibration to suffer from a drift in scale, as well as a method for estimating a depth distortion model which significantly improves the calibration accuracy. Since this model is estimated in open-loop, the overall method has low run times ($\approx$ 30sec for 15 images). Moreover, we present a new minimal solution for the problem of registering two sets of corresponding planes.



(a) Our method with DC (b) Herrera's method without DC (c) Herrera's method with DC

Figure 10. Image of an object with holes acquired by our Kinect. The RGB image is overlaid with the depth map to show the misalignment.

ACKNOWLEDGMENT

Carolina Raposo is funded by the Portuguese Foundation for Science and Technology (FCT) through the PhD grant SFRH/BD/88446/2012. This work has also been supported by the Portuguese Foundation for Science and Technology (FCT) and COMPETE program (co-funded by FEDER) under Project PTDC/EEA-AUT/113818/2009.

REFERENCES

[1] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca. Automatic camera calibration applied to medical endoscopy. In *Proceedings of the 20th British Machine Vision Conference, London, UK*, 2009.

[2] J. Y. Bouguet. Camera calibration toolbox, 2010.

[3] N. Burrus. Kinect calibration, 2011.

[4] S. Fuchs and G. Hirzinger. Extrinsic and depth calibration of tof-cameras. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.

[5] J. Heikkilä. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10), 2000.

[6] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research (IJRR)*, 31(5):647–663, April 2012.

[7] C. D. Herrera, J. Kannala, and J. Heikkilä. Accurate and practical calibration of a depth and color camera pair. In *Proceedings of the 14th international conference on Computer analysis of images and patterns - Volume Part II*, CAIP'11, Berlin, Heidelberg, 2011.

[8] D. Herrera C., J. Kannala, and J. Heikkilä. Joint depth and color camera calibration with distortion correction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10), 2012.

[9] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4), Apr 1987.

[10] T. Lozano-Pérez and W. E. L. Grimson. Model-based recognition and localization from sparse range or tactile data, 1983.

[11] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007.

[12] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011.

[13] H. Yamazoe, H. Habe, I. Mitsugami, and Y. Yagi. Easy depth sensor calibration. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012.

[14] C. Zhang and Z. Zhang. Calibration between depth and color sensors for commodity depth cameras. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 2011.

[15] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, 1999.