2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Unsupervised Intrinsic and Extrinsic Calibration of a Camera-Depth Sensor Couple

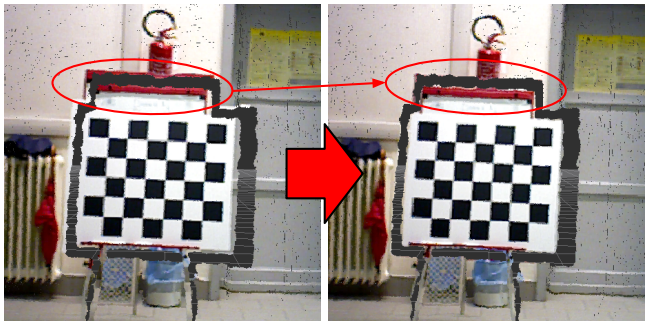
Filippo Basso, Alberto Pretto and Emanuele Menegatti.

***Abstract*— The availability of affordable depth sensors in conjunction with common RGB cameras (even in the same device, e.g. the Microsoft Kinect) provides robots with a complete and instantaneous representation of both the appearance and the 3D structure of the current surrounding environment. This type of information enables robots to safely navigate, perceive and actively interact with other agents inside the working environment. It is clear that, in order to obtain a reliable and accurate representation, not only the intrinsic parameters of each sensors should be precisely calibrated, but also the extrinsic parameters relating the two sensors should be precisely known. In this paper, we propose a human-friendly and reliable calibration framework, that enables to easily estimate both the intrinsic and extrinsic parameters of a camera-depth sensor couple. Real world experiments using a Kinect show improvements for both the 3D structure estimation and the association tasks.**

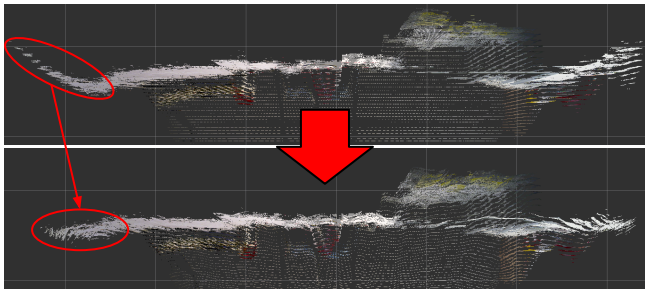
I. INTRODUCTION

Typical robotic tasks like SLAM, navigation, object recognition and many others, highly benefit from having color and depth information fused together. While color information is almost always provided by RGB cameras, there are plenty of sensors able to provide depth information: time-of-flight (ToF) cameras, laser range scanners and sensors based on structured light. Even if there are some devices able to provide both color and depth data (e.g. the popular low-cost Microsoft Kinect, composed by two very close sensors), as far as we know, there are no integrated sensors able to provide both color and depth information yet. In this paper we focus on Kinect-like devices (among others, the Asus Xtion Pro Live). These sensors provide colored point clouds that suffer from a non accurate association between depth and RGB data, due to a non perfect alignment between the camera and the depth sensor. Moreover, depth images suffer from a geometric distortion, typically irregular and position dependent. Finally, we have noticed that for increasing distances, there is an increasing bias (i.e., a systematic error) in depth measurements. These devices are factory calibrated, so each sensor is sold with its own calibration parameter set stored inside a non-volatile memory. However, the quality of this calibration is only adequate for gaming purposes.

This research has been partially supported by Telecom Italia SPA with the grant "Service Robotics", by University of Padova with the grants "DVL-SLAM" and "TIDY-UP: Enhanced Visual Exploration for Robot Navigation and Object Recognition", and by the European Commission under FP7-600890-ROVINA. Basso, Pretto and Menegatti are with the Department of Information Engineering, University of Padova, Italy. Email: {filippo.basso, emg}@dei.unipd.it. Pretto is also with the Department of Computer, Control, and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Italy. Email: pretto@dis.uniroma1.it



(a)



(b)

Fig. 1. Some results of our calibration procedure: (a) The non perfect alignment between the camera and the depth sensor produce inaccuracies in the depth-color association (left point cloud). A better alignment obtained with our calibration procedure results in a more accurate association (right point cloud). (b) A point cloud of a planar surface (a wall) without depth distortion correction (top) and the same cloud after the application of the proposed undistortion map (bottom).

Moreover, the depth distortion is not modeled in the factory calibration. A proper calibration method for robust robotics applications should precisely estimate the misalignment and both the systematic and distortion errors.

We propose a novel calibration method that employs a simple data-collection procedure, that only needs a minimally structured environment, and that does not require any parameters tuning or a great interaction with the calibration software. Moreover, even if the principal targets of the method are the Kinect-like devices mentioned above, it is thought to be used also with, even non-close, heterogeneous camera-depth sensor couples.
Given a calibrated camera and an uncalibrated depth sensor, the proposed method automatically infers the intrinsic parameter set of the depth sensor and the alignment between the two sensors, i.e. the rigid body transformation that relates the two sensor frames.

For the depth sensor, we employ an error model that

includes a "distance space" distortion error along with a variable systematic error (in the following also called "global error"). We propose to represent the undistortion map by means of a set of functions (quadratics for the Kinect), iteratively fitted to the acquired data during a first calibration stage. We include the systematic error (that our experiments show it is quadratic) and the sensors alignment in a second stage of the calibration: at this point, we exploit the plane-to-plane constraints between color and depth data to align the two sensor and to infer the systematic error inside a non-linear optimization framework.

Our main contribution are:

- An easy-to-implement calibration protocol, that provides the input data used for both the undistortion map and alignment estimation processes.
- A spatial/parametric undistortion map that models in a compact and efficient way the lens distortion effect in Kinect-like depth sensors.
- A novel optimization framework that aims to estimate the camera-depth sensor alignment along with a parametric model that well describes the systematic error of the depth measurements.

The paper is structured as follows: in Section II we explain in details how we model the distortion error of the depth sensor while in Section III we describe an algorithm to estimate the intrinsic parameters of the depth sensor. Section V is dedicated to the experiments and the evaluation of the proposed approach, and Section IV gives an overview on a real implementation of the algorithm.

A. Related work

Literature is full of calibration methods for fusing color and depth data, however most of them deal with depth data generated by 2D/3D laser range finders [1], [2] or time-of-flight cameras [3]. Such methods are not directly applicable to Kinect-like devices. In fact, since they are structured-light based sensors, the nature of the error is different from other sensor types, so also the introduced error pattern is different. In one of the first works on the Kinect, Smisek et al. [4] showed that Kinect devices are affected by a sort of radially symmetric distortions: this fact was later confirmed in [5], [6], [7]. In [5], we can find a first attempt to take into account the distortion during the calibration process. However, as stated by the authors, their approach is not suitable for large distances between the camera and the depth sensor. Zhang et al. [8] realize that the depth value z provided by the depth sensor of the Kinect was a linear function of the real one z^*, that is $z = \mu z^* + \eta$.

Teichman et al. [6] recently proposed a calibration approach for Kinect-like devices where it has been observed that the error introduced by such depth sensors is *myopic*, that is, if the depth value increases, the error increases as well. This is probably the work most related to our approach. In fact, they estimate the undistortion map by means of a SLAM framework, in an unsupervised way.

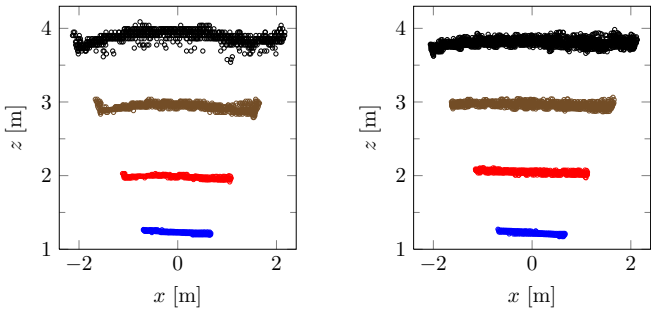


Fig. 2. (a) Point clouds of a planar surface (a flat wall) for increasing, known distances. The distortion effect increases for greater distances; (b) After the application of the proposed undistortion map to the point clouds in (a), the depth distortion becomes negligible.

B. Notations

We use non-bold characters x to represent scalars, bold lower case letters $\mathbf{x}$ to represent vectors with no distinction between cartesian coordinates and homogeneous coordinates. Bold upper case letters $\mathbf{M}$ represent matrices. Note that matrices can be seen as ordered lists of vectors, one for each column. A coordinate frame belonging to a body B is denoted by $\mathcal{B}$. The coordinates of a point $\mathbf{p}$ with respect to the coordinate frame $\mathcal{F}$ are denoted by ${}^{\mathcal{F}}\mathbf{p}$. ${}^{\mathcal{B}}_{\mathcal{A}}\mathbf{T}$ denotes the homogeneous transformation matrix from $\mathcal{A}$ to $\mathcal{B}$, such that ${}^{\mathcal{A}}\mathbf{p} = {}^{\mathcal{B}}_{\mathcal{A}}\mathbf{T}\,{}^{\mathcal{B}}\mathbf{p}$.

II. DEPTH ERROR MODEL

A. State of the Art

Kinect-like depth sensors *intrinsic* parameters has been termed *myopic* [6], that means that an incorrect parameter set results in an absolute error that increases with distance. We can easily see this property analyzing the data coming from one of these depth sensor when positioned in front of a flat wall. As we can see in Fig. 2(a), while at close ranges the data is highly reliable, for greater distances the distortion strongly affect the depth accuracy.

To model the effects of this distortion, in some recent works [5], [6], [7] authors try to estimate depth correction functions in a per-pixel basis. That is, given a pixel $(u, v) \in \Omega \subset \mathbb{Z}^2$ and the corresponding depth value d, the real depth d^* is estimated as

$$d^* = \mathrm{f}_{(u,v)}(d). \tag{1}$$

This function is estimated using a camera as a reference [5], [7] or running a SLAM system while inferring the function parameters [6]. In the former two methods, the authors assume that the camera and the depth sensor are extrinsically calibrated (i.e., they assume to know the rigid body transformation that relates the two sensors) and they use the plane defined by the checkerboard as a ground truth for the depth sensor. Actually, if the extrinsic calibration is correct, the depth sensor's intrinsics can be easily estimated but, as introduced above (e.g., see Fig. 1(a)), this assumption is partially violated also when using a factory calibrated

device as the Kinect. On the other hand, if no extrinsic calibration is provided, as in the case of general camera-depth sensor couples, these methods cannot be applied.

B. Our Proposal

We propose to separate the error into two different parts and treat them separately. We call *distortion* the error responsible of the local alteration of an object shape (e.g., see Fig. 2(a)) and *global error* the *systematic* wrong estimation of the average depth. We substantially express each $\mathrm{f}(\cdot)$ in (1) as a composition of two functions: $\mathrm{u}(\cdot)$ that takes into account the local distortion and $\mathrm{g}(\cdot)$ that instead makes a global correction of the depth values. That is, the real depth d^* is estimated as

$$d^* = \mathrm{f}_{(u,v)}(d) = (\mathrm{g} \circ \mathrm{u})_{(u,v)}(d). \tag{2}$$

In this context, we define an *undistortion map* $\mathbb{U}$ as the set of all possible $\mathrm{u}(\cdot)$. This separation allows us to estimate the intrinsic parameters in two steps. We can first compute the undistortion map without worrying about the real depth and, at a later stage, estimate the systematic error along the z-axis.

C. Conventions

An image pixel (u, v) along with its depth value d represent the projection of an unique 3D scene point $(x, y, z)^\mathsf{T}$. In other words, the same scene point $\mathbf{p} = (x, y, z)^\mathsf{T}$ can be represented also as (u, v, d), i.e.

$$(u, v, d) \Leftrightarrow (x, y, z)^\mathsf{T} = \mathbf{p}.$$

Therefore, in the following we will use indifferently any of the two forms. Considering for example Eq. (1), we can rewrite it in terms of a point $\mathbf{p} = (x, y, z)^\mathsf{T}$, obtaining

$$d^* = \mathrm{f}_{(u,v)}(d) \Leftrightarrow \mathrm{f}(\mathbf{p}) = \mathbf{p}^*.$$

III. CALIBRATION APPROACH

For a given sensor couple, there is a close relationship between intrinsic and extrinsic calibration parameters. A first solution could be to estimate everything together, inside an unique optimization process. That is, starting from an initial guess, one can try to find the parameters that minimize a suitable error function. However, even if theoretically plausible, due to the huge number of parameters (e.g. 32,000 in [6]) the problem is really hard to solve.

Starting from the considerations we made in Section II, we decided to follow a different approach. Exploiting the continuity of the error, we can assume that given two close 3D points $\mathbf{p}$ and $\mathbf{q}$ along the same direction, i.e $\mathbf{q} = (1+\epsilon)\mathbf{p}$ with $\epsilon \simeq 0$,

$$\mathbf{q}^* = \mathrm{f}(\mathbf{q}) = \mathrm{f}((1+\epsilon)\mathbf{p}) \simeq (1+\epsilon)\mathrm{f}(\mathbf{p}) = (1+\epsilon)\mathbf{p}^*.$$

where $\mathrm{f}(\cdot)$ has been defined in Eq. 2. In particular, when dealing with the distortion error

$$\hat{\mathbf{q}} = \mathrm{u}(\mathbf{q}) \simeq (1+\epsilon)\mathrm{u}(\mathbf{p}).$$

It means that, if we know how to undistort a point $\mathbf{p}$, we can undistort close points with a good approximation. This assumption is the basis of our algorithm to estimate the undistortion map $\mathbb{U}$.

A. Undistortion Map Estimation

A depth sensor D provides a discrete representation of the scene by means of a point cloud $\mathbf{C}$. Supposing that the sensor is pointing a planar surface (e.g., a wall), we define $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2 \dots \mathbf{p}_N\} \subseteq \mathbf{C}$ as the subset of points belonging to this planar surface. Using standard least squares methods we can easily fit a plane π to the given subset $\mathbf{P}$ and therefore define $\mathrm{D}_\pi(\mathbf{p})$ as the distance from a point $\mathbf{p}$ to the plane π.

The input of the algorithm is a list of point clouds taken when the sensor is pointing a planar surface at different distances. For each cloud, we extract the distance from the fitted plane, we sort the list for increasing distances, then we process the ordered list $\{\mathbf{P}_0, \mathbf{P}_1 \dots \mathbf{P}_M\}$ as follows.
Firstly, initialize $\mathbb{U}_0$ with a set of identity functions. That is, $\mathrm{u}_{(u,v)}(d) = d$, $\forall\ (u, v) \in \Omega$.
Then, for $i = 1 \rightarrow M$:

1) Undistort $\mathbf{P}_i$ with the previously estimated undistortion map $\mathbb{U}_{i-1}$.
2) Fit a new plane π_i to the undistorted data $\hat{\mathbf{P}}_i = \{\mathrm{u}(\mathbf{p}) \mid \mathbf{p} \in \mathbf{P}_i\}$.
3) Compute the distances $\mathrm{D}_{\pi_i}(\mathrm{u}(\mathbf{p}))$ of all points from the plane.
4) For each point $\mathbf{p} \in \mathbf{P}_i$ that meets the condition $\mathrm{D}_{\pi_i}(\mathrm{u}(\mathbf{p})) < K$ (K is a threshold we use to reject outliers):
   a) Project $\mathbf{p}$ on π_i along its direction, obtaining a new point $\mathbf{p}'$ (in this way $(u, v) = (u, v)'$).
   b) Add the pair (d, d') to the sample set $\mathbb{S}_{(u,v)}$ of (u, v).
   c) Update the estimation of the parameters of $\mathrm{u}_{(u,v)}(d)$ fitting a new curve to the whole sample set $\mathbb{S}_{(u,v)}$.

At the end we have the undistortion map $\mathbb{U} = \mathbb{U}_M$ that better corrects the input data.

B. Global Error Correction

In order to estimate the global, systematic error, we need to introduce some constraints on the input data. A common way in this case is to use some measurements taken with the camera C. However, as stated in section II, we *cannot* keep the extrinsic calibration of the two sensors (i.e., the transformation between them) separated from the intrinsic calibration of the depth sensor (i.e., the global error). The idea here is to find a non-optimal initial guess and then optimize the intrinsic and the extrinsic parameters together. We use the same input dataset we used during the undistortion map estimation (Sec. III-A), corrected using $\mathbb{U}$. We also assume that in the planar surface we frame with the sensor, a checkerboard is also present (e.g., a checkerboard attached to a wall). Using an already calibrated camera, we can estimate the pose of the checkerboard B in its field of view, while using a (partially calibrated) depth sensor it is possible to fit

planes to the input (planar) point clouds. We can therefore use the Unnikrishnan method [9] to estimate an initial guess for the transformation ${}^{\mathcal{D}}_{\mathcal{C}}\overline{\mathbf{T}}$ between the two sensor frames. We can also estimate an initial guess for the error correction function $\overline{\mathrm{g}}$ by using all the points in $\mathbf{P}$ and their line-of-sight projection [10] on the planes defined by the checkerboards as sample pairs for the curve fitting procedure.

We can now estimate the correct transformation ${}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}^*$ and the correct global function g^* as:

$$({}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}^*, \mathrm{g}^*) = \arg\min_{{}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}, \mathrm{g}} e({}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}, \mathrm{g}) \tag{3}$$

where $\mathrm{e}(\cdot)$ is the error function we want to minimize and that takes into account both the re-projection error of the checkerboard corners and the alignment between the planes in the images and the planes in the depth maps. We use ${}^{\mathcal{D}}_{\mathcal{C}}\overline{\mathbf{T}}$ and $\overline{\mathrm{g}}$ as an initial guess for the optimization.

C. Optimization Function

Given a checkerboard B framed by the camera and its corner set ${}^{\mathcal{B}}\mathbf{B} = \{{}^{\mathcal{B}}\mathbf{b_1}, {}^{\mathcal{B}}\mathbf{b_2} \dots {}^{\mathcal{B}}\mathbf{b_N}\}$, we can define:

- ${}^{\mathcal{I}}\mathbf{B}_i = \{{}^{\mathcal{I}}\mathbf{b_1}_i, {}^{\mathcal{I}}\mathbf{b_2}_i \dots {}^{\mathcal{I}}\mathbf{b_N}_i\}$ as the corner set in pixel coordinates (i.e., the 2D image points) in the image i, $i = 0 \dots M$.
- ${}^{\mathcal{C}}\mathbf{B}_i$ as the corner set of the checkerboard (i.e., the 3D scene points) extracted from image corners ${}^{\mathcal{I}}\mathbf{B}_i$ (using one of the *PnP* methods).
- ${}^{\mathcal{D}}\mathbf{P}_i$ as the 3D point set belonging to the checkerboard plane extracted from the i^{th} *undistorted* depth map.

We also define $\mathrm{repr}(\cdot)$ as the re-projection function that returns the pixel coordinates of a scene point $\mathbf{p}$.
We define our error function as

$$e_{\text{repr}} = \sum_{i=0}^{M}\sum_{j=1}^{N} ||\mathrm{repr}({}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}^{-1}{}^{\mathcal{D}}\mathbf{b_j}_i) - {}^{\mathcal{I}}\mathbf{b_j}_i||^2.$$

where $||\cdot||$ is the L^2-norm. At each step s of the optimization, for each image i, we can compute ${}^{\mathcal{D}}\mathbf{b_j}_i$ as following:

1) Correct the depth points using the current estimation $\mathrm{g_s}(\cdot)$ of $\mathrm{g}(\cdot)$, i.e. ${}^{\mathcal{D}}\hat{\mathbf{P}}_i = \mathrm{g_s}({}^{\mathcal{D}}\mathbf{P}_i)$.
2) Fit a plane ${}^{\mathcal{D}}\hat{\pi}_i$ to the corrected points.
3) Transform ${}^{\mathcal{D}}\hat{\pi}_i$ in the camera coordinates using the current estimation ${}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}_s$ of ${}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}$.
   Note that, if the two sensors are perfectly calibrated, ${}^{\mathcal{C}}\hat{\pi}_i = {}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}_s^{-1}{}^{\mathcal{D}}\hat{\pi}_i$ coincides with the plane ${}^{\mathcal{C}}\pi_i$ defined by the checkerboard in the camera frame.
4) Project the corners ${}^{\mathcal{C}}\mathbf{B}_i$ (blue points in in Fig. 3) onto the plane ${}^{\mathcal{C}}\hat{\pi}_i$ extracted using the depth sensor (green points in in Fig. 3). We just need to intersect the optical ray of the center $\mathbf{c}_i$ of the checkerboard ${}^{\mathcal{C}}\mathbf{B}_i$ with the plane ${}^{\mathcal{C}}\hat{\pi}_i$ (red points in in Fig. 3) and set the pose of the corners ${}^{\mathcal{D}}\mathbf{b_j}_i$ imposing the dimension constraints, as depicted in Fig. 3.

The minimization of e_{repr} converges to accurate results in presence of good initial guesses ${}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}$ and g, while usually converges to wrong local minima with initial guesses far

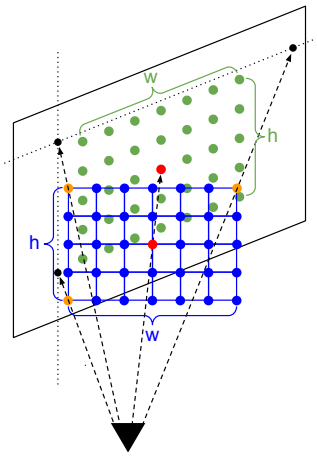


Fig. 3. Overview of the projection procedure of the checkerboard corners onto the plane extracted using the depth sensor.

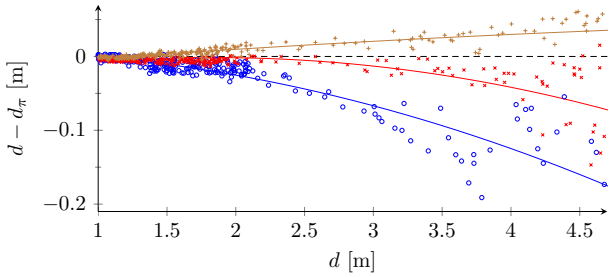


Fig. 4. Signed distortion errors $d - d_\pi$ relative to 3 different pixels (u, v). They are clearly non-linear.

from the optimal solution. For this reason we add a penalty factor

$$e_{\text{pen}} = \prod_{i=0}^{M} \exp\left(\frac{1}{N}\sum_{j=1}^{N}\frac{||{}^{\mathcal{C}}_{\mathcal{D}}\mathbf{T}^{-1}{}^{\mathcal{D}}\mathbf{b_j}_i - {}^{\mathcal{C}}\mathbf{b_j}_i||}{||{}^{\mathcal{C}}\mathbf{b_j}_i||}\right)$$

to take into account the distance between the two planes ${}^{\mathcal{C}}\hat{\pi}_i$ and ${}^{\mathcal{C}}\pi_i$. Eq. (3) therefore becomes

$$({}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}^*, \mathrm{g}^*) = \arg\min_{{}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}, \mathrm{g}} \left[e_{\text{pen}}({}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}, \mathrm{g}) \cdot e_{\text{repr}}({}^{\mathcal{D}}_{\mathcal{C}}\mathbf{T}, \mathrm{g})\right].$$

D. Real Data Analysis

We conducted some experiments to estimate the function types that better fit to the two above-mentioned errors. To have an idea of how the distortion evolved with the depth, we positioned a Kinect in front of a wall and fit a plane to the depth data. Given a pixel (u, v) we compared its depth value d with the value d_π the pixel should have had to lie on the fitted plane. We repeated this procedure at increasing distances from the wall. As we can see in Fig. 4, the error $d - d_\pi$ is non-linear and well fitted by a second degree polynomial.

We then estimated the evolution of the global error for the Kinect sensor. We positioned a Kinect in front of a big checkerboard and extracted the relative plane (function `solvePnP()` from `http://opencv.org`). Using the

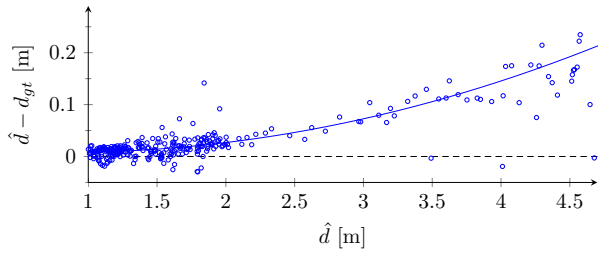


Fig. 5. Signed global error $\hat{d} - d_{gt}$ and correspondent fitted second degree polynomial.

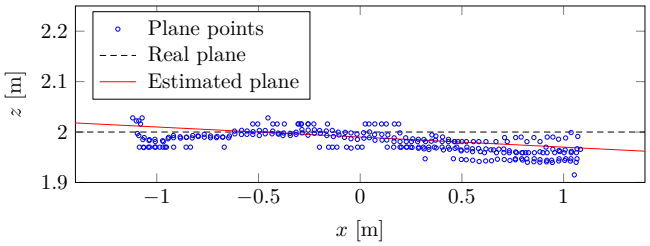


Fig. 6. Comparison between the real plane and the one fitted to the point cloud. They are not parallel.

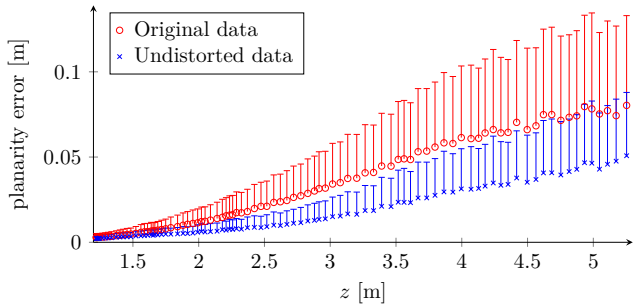


Fig. 7. Planarity error. The plot shows both the mean error and the its standard deviation. As we can see also in Fig. 2(b) the planarity of the cloud is greatly improved.

previously computed undistortion map, we undistorted the depth data and computed, for each pixel, the difference between its value $\hat{d}$ and the one it should have to lie on the checkerboard plane d_{gt}. We then repeated the procedure at different distances. The results are visible in Fig. 5. As we can see, also in this case the error is not linear in d.

IV. IMPLEMENTATION DETAILS

We distribute the source code of our calibration system[1] implemented inside the well known robotic framework ROS[2]. Our software takes less than an hour to complete the calibration in a modern quad-core PC when dealing with a dataset of nearly 400 image-point cloud pairs at half resolution.

Using the provided software, the user is asked to collect a dataset of point cloud-image pairs of the same scene containing one or more checkerboards. Obviously, with bigger planar surface containing the checkerboard, it is possible to obtain a more accurate undistortion map. Moreover, to be able to estimate checkerboard surfaces, the software needs a rough initial transformation between the frames of the two sensors (e.g. for Kinect-like devices the identity matrix is a very good initial guess). The transformation can be set in a configuration file or estimated with the help of a simple tool that asks the user to select some (less than 10) planes that contains the checkerboards. During the undistortion map estimation, the checkerboard is only used as a marker to discriminate the calibration plane in the scene. The software keeps trace of the z-distance between the center of the checkerboard and its projection on the fitted plane. These values let the program to incrementally model the systematic error and therefore have a better estimation of the location of the calibration plane with respect to the depth sensor. At this point the calibration procedure analyzes all the pairs and estimates both the intrinsic parameters of the depth sensor and the transformation between the two sensors.

Even if the proposed global error model works well in many circumstances, it is based on a wrong assumption: the fitted planes are parallel to the real planes. In fact, as depicted in Fig. 6, this assumption is wrong. Therefore, instead of using a single second degree polynomial to correct the global error, we estimate 4 polynomials, one for each corner of the depth map. This operation enables us to model non-parallel fitted planes and greatly improves our calibration results.

V. EXPERIMENTAL EVALUATION

We evaluate our calibration procedure with a test set consisting of more than 100 image-depth map pairs. We mounted an Asus Xtion Pro on a mobile robot and collected the data while moving towards a flat wall with a checkerboard attached.

A. Quantitative Results

We firstly evaluated the *planarity* of the original depth maps compared with the planarity of the same cloud undistorted with the computed undistortion map. We fit a plane π to the clouds and computed both the mean of the line-of-sight error [10] of the points and its variance. As visible in Fig. 7, using the estimated undistortion map planarity is substantially improved.

We then estimated the results of the whole calibration procedure by computing the planarity error with respect to the plane defined by the checkerboard, namely *calibration error*. We extracted the plane by means of the above-mentioned `solvePnP()` function and transformed it to the depth sensor reference frame. We computed the error of the original depth map, the "partially" undistorted one (no global error correction applied) and the fully undistorted one. As expected (Fig. 8), the original clouds and the partially undistorted ones show roughly the same error distribution, while the fully undistorted clouds are definitely much more reliable.

[1] `https://github.com/iaslab-unipd/rgbd_calibration`
[2] `http://ros.org`

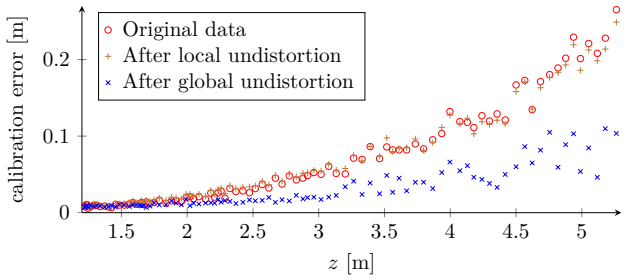


Fig. 8. Calibration error.Absolute error with respect to the ground truth d_{gt} after different stages of the algorithm.

TABLE I

RESULTING CALIBRATION PARAMETERS.

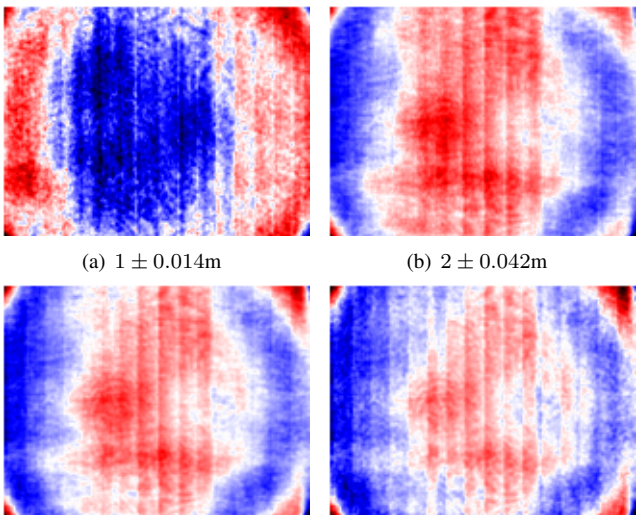
Translation [m]	Rotation
x: 0.0222252	x: -0.00337542
y: -0.00159009	y: 0.00425487
z: 0.00780007	z: 0.00190883
	w: 0.999983

For what concerns the extrinsic parameters, the resulting values (Tab. I) are close to the factory provided ones.

B. Qualitative Results

We cannot evaluate the estimated undistortion map directly, we can instead show (Fig. 9) that the one estimated with our algorithm is comparable with those visible in other works, especially [4], [6], [7]. The maps visible in Fig. 9 are obtaining by applying the local undistortion function to a synthetic point cloud at a defined distance and printing the resulting depth values with a color scale from deep red to deep blue.

The reported undistortion map is obtained by discretizing the depth images into bins of 8×8 pixels. We successfully tested the algorithm even with a full resolution map without any significant improvement in performances.



(a) 1 ± 0.014m (b) 2 ± 0.042m

(c) 3 ± 0.110m (d) 4 ± 0.237m

Fig. 9. Undistortion map applied to synthetic flat point clouds. Note that the color scale is different for each map.

We also evaluated the calibration procedure in real world scenarios. Fig. 1(a) shows that the estimated calibration outperforms the factory one in terms of point cloud-image alignment. Moreover, as visible in both Fig. 2(b) and 1(b), the scene geometry is improved.

VI. CONCLUSIONS

In this paper we presented a novel method to calibrate both the intrinsic and the extrinsic parameters of a Kinect-like RGBD sensor. The proposed calibration procedure only requires to collect data in a minimally structured environment (e.g., a wall with attached a checkerboard). We propose to model the depth sensor error by means of two different components, a distortion error and a global, systematic error. The distortion error is modeled using a per-pixel parametric undistortion map, estimated in the first stage of the algorithm. The camera-depth sensor alignment along with the depth systematic error are then estimated in the second stage of the algorithm, inside a robust optimization framework [11]. Reported results show the possibility to improve the accuracy of a low cost RGBD sensor with very simple procedure. As a future work we plan to compare our algorithm with other recent approaches (e.g., [6]).

ACKNOWLEDGMENT

Authors would like to thank Edmond So for the fruitful discussions and all the valuable suggestions.

REFERENCES

[1] C. Mei and P. Rives, "Calibration between a Central Catadioptric Camera and a Laser Range Finder for Robotic Applications," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 532–537.

[2] 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, pp. 4164–4169.

[3] Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun, "Design and Calibration of a Multi-view TOF Sensor Fusion System," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2008. CVPRW '08. IEEE Computer Society Conference on*, 2008, pp. 1–7.

[4] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect," in *Computer Vision Workshops (ICCV Workshops), 2011. ICCVW 2011. IEEE International Conference on*, 2011, pp. 1154–1160.

[5] 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*, vol. 34, no. 10, pp. 2058–2064, 2012.

[6] A. Teichman, S. Miller, and S. Thrun, "Unsupervised Intrinsic Calibration of Depth Sensors via SLAM," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[7] A. Canessa, M. Chessa, A. Gibaldi, S. P. Sabatini, and F. Solari, "Calibrated Depth and Color Cameras for Accurate 3D Interaction in a Stereoscopic Augmented Reality Environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 227 – 237, 2014.

[8] 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, pp. 1–6.

[9] R. Unnikrishnan and M. Hebert, "Fast Extrinsic Calibration of a Laser Rangefinder to a Camera," Carnegie Mellon University, Tech. Rep., 2005.

[10] E. So, F. Basso, and E. Menegatti, "Calibration of a Rotating 2D Laser Range Finder using Point-Plane Constraints," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 7, no. 2, pp. 30–38, 2013.

[11] S. Agarwal and K. M. et al., "Ceres Solver." [Online]. Available: https://code.google.com/p/ceres-solver/