Non-Parametric Calibration for Depth Sensors DRAFT

Maurilio Di Cicco, Luca Iocchi, Giorgio Grisetti

Dept. of Computer, Control and Management Engineering
Sapienza University of Rome
Via Ariosto 25, I-00185, Rome, Italy

Abstract

RGBD sensors are commonly used in robotics applications for many purposes, including 3D reconstruction of the environment and mapping. In these tasks, uncalibrated sensors can generate poor quality results. In this article we propose a quick and easy to use approach to estimate the undistortion function of RGBD sensors. Our approach does not rely on the knowledge of the sensor model, on the use of a specific calibration pattern or on external SLAM systems to track the device position. We compute an extensive representation of the undistortion function as well as its statistics and use machine learning methods for approximation of the undistortion function. We validated our approach on datasets acquired from different kinds of RGBD sensors and using a precise 3D ground truth. We also provide a procedure for evaluating the quality of the calibration using a mobile robot and a 2D laser range finder. The results clearly show the advantages in using sensor data calibrated with the method described in this article.

Keywords: Calibration, Mobile Robots, Depth camera

1. Introduction

Depth sensors, like Microsoft Kinect or ASUS Xtion, are major technological achievements and gave a new impulse to a wide range of 3D applications in robotics such as SLAM [1, 2, 3], super-resolution mapping [4], object recognition and many others. Unfortunately, these devices suffer from a large systematic distortion that is hard to model. Neglecting this distortion has substantial effects on the algorithms that use depth data. As shown in Figure 1, uncalibrated sensors produce data with a systematic noise that cannot be easily compensated by standard SLAM algorithms.

Kinect-like RGBD sensors operate on a stereo principle, where one of the cameras is replaced by a light source and the other camera senses the light pattern reflected by the scene. The depth is recovered by a proprietary algorithm developed by PrimeSense. A third camera is then used to augment the depth layer with RGB information. In principle, calibrating this sensor would require to estimate the intrinsic and extrinsic parameters of the three cameras in the system. However, the lack of knowledge about the pattern matching algorithm used to determine the stereo correspondences makes the application of these parameters not straightforward. Even assuming that the matching is always correct, one would have to first determine the disparity by applying the nominal parameters to the measured depths then use the estimated parameters to recover an improved depth estimate.Still it remains unclear how to estimate the intrinsics of the IR projector used as light emitter. Furthermore, we would loose the hardware acceleration provided by the PrimeSense algorithms running on the device.

In this article, we propose an approach to compute an *undistortion function* that is able to convert a distorted depth image onto another one where the systematic distortion is removed. The undistortion function maps every possible depth measurement to a multiplying factor that is the ratio between the true depth and the measured one, as proposed by Teichman *et. al* [5].

In our approach we experimented two different

Email address:
{dicicco,iocchi,grisetti}@dis.uniroma1.it (Maurilio Di Cicco, Luca Iocchi, Giorgio Grisetti)

Figure 1: 3D reconstruction of a home environment from depth data acquired with an ASUS Xtion mounted on a Turtlebot mobile robot. Left: the result using uncalibrated data. Right: the output of the same algorithm when using data calibrated with our procedure. The systematic error in the calibration resulted in noisy open loop estimates that result visible even with correct loop closings.

kinds of function approximators, computed by the corresponding machine learning regression methods: a K-Nearest Neighbor (KNN) and an Artificial Neural Network (ANN). The training data for these models are extracted automatically from a set of RGBD images looking at a scene with a main planar surface at different distances.

The calibration procedure presented in this article is fully automatic, after a set of depth images looking at a planar surface has been taken. The acquisition of these data typically requires less than one minute and the calibration approach is usually implemented off-line. The application of our calibration procedure does not require specific expertise and it can be performed by a non-expert user with just a few recommendations about how to operate the device. Moreover, if the sensor is mounted on a mobile robot, the acquisition procedure can be performed in a fully automatized way by implementing a behavior that slowly moves the robot towards a wall. Being so quick and easy, our approach can be the preliminary step of a mission. Finally, since our approach does not specialize on a specific sensor model, it can be used for any 3D sensor.

Compared to our previous work [6], this paper :

- presents a deeper analysis of imaging depth sensors that allows to better characterize different distortion phenomena;
- extends the previous approach with a routine to adjust for systematic depth dependent noise, thus resulting in increased sensor accuracy;
- proposes a wider set of quantitative experiments.

We tested our approach with different devices, such as Kinect and ASUS Xtion in different versions, and the results of the experimental evaluation shows that our procedure is effective and accurate.

The software implementing the described method is available through the website `http://easy_depth_calibration.dis.uniroma1.it`, which contains also easy-to-follow instructions for its use, data sets and further results, as well as the description of how to reproduce the results reported in this article.

2. Related Work

Calibration of RGBD sensors has been investigated in previous work as described in this section.

Yamazoe *et. al* [7] proposed a traditional approach of parametric calibration of the Kinect by using a checker board. Similarly, Fuchs *et al.* [8] presented a general analytical model to calibrate ToF cameras. In contrast to these approaches, we do not use any external calibration device and we do not rely on a specific sensor model. This is particularly useful since, as Smisek *et al.* stated [9], different Kinect devices shows different radial distortion patterns that may not be well approximated by the same model. The same behavior has been observed also in Xtion devices. Since we do not require a sensor model, our method can be easily applied to a broader range of sensors where distortion is not regular across the image.

While the above mentioned methods are aiming at calibrating the sensor parameters to lessen its systematic distortion, Nguyen *et al.* [10] focus on characterizing the distortion as a function of both distance and angle of a Kinect on the observed surface and derive a noise model to filter depth maps. Our approach instead does not rely on a parametric model to deal with the sensor's distortion.

A notable approach for computing an undistortion function of a depth sensor has been proposed by Teichman *et al.* [5] and it is known as *Calibrating, localizing, and mapping, simultaneously* (CLAMS). This approach relies on a dense representation of the undistortion function. In contrast to our approach, to compute this function the CLAMS system relies on a working SLAM module. By exploiting the fact that the distortion grows with the distance [11] and that it is neglectible for distances below 2 meters, CLAMS repeatedly executes the SLAM algorithm several times. At each run, only the short ranges are used to track the position of the camera and to determine the ground truth of the distances. Thus the overall procedure is iterative and, at each round, SLAM is executed taking into account the most recent calibration. After the SLAM rounds are terminated, the multipliers are estimated for the full range of the sensor. To achieve acceptable results, the approach requires around three minutes of recorded data and one round of calibration, given the SLAM trajectory takes around 10 minutes. The shortcoming of this approach is that it requires a working SLAM module that operates on data acquired on a scene suitable for the calibration and rich in features for SLAM. Executing a full calibration with CLAMS is reported to be an overnight procedure. In contrast to this approach, our method simplifies the extraction of the undistorted depths, by exploiting the fact that pixel in the middle of the image are usually affected by a small systematic error and by fitting a plane in this region. Our method does not require any external software module and it is much faster as it takes a few milliseconds per image. As already mentioned, the input of our method is just a set of depth images capturing a large wall at different distances.

Basso *et al.* [12] proposed a method that is closely related to ours, where they estimate a per-pixel undistortion map *and* recover the calibration parameters of both RGB and Depth camera. The distortion of each pixel is assumed to be locally continuous, and a piecewise linear approximation of this function is computed from a few samples obtained by capturing a planar surface at a set of known depths. Based on this initial undistortion map, the authors calibrate the intrinsics and extrinsic parameters of the depth-RGB pair by means of a checkerboard. Similarly, Canessa *et al.* [13] propose a method for calibrating a depth and RGB pair by using a checkerboard with the RGB and IR camera of the sensor. To this end they disable the IR projector and use a light source that is visible in both images. This allows to accurately recover the intrinsics and the extrinsics of the camera pair. Subsequently they use per pixel depth map consisting of a quadratic function, estimated from four distance samples.

Compared to these methods, our approach focuses on the depth sensor, thus it can be applied to devices that do not expose an RGB channel (e.g. Asus Xtion Pro). Additionally our calibration procedure is simpler, since it does not require a checkerboard and automatically rejects frames when the plane is not sufficiently orthogonal to the sensor.

3. Principles of Depth Imaging Sensors

The devices considered in this paper, are essentially active stereo cameras, where the stereo pair consists of an IR light source, and the other one is an IR camera sensible. The light source projects a fixed pattern onto the scene that is visible in the image captured by the camera. The disparity is computed by a proprietary algorithm running on the sensor that matches the projected and received patterns. This algorithm relies on an estimate of the intrinsics and extrinsics parameters of the camera system. In principle, to get more accurate depth estimates, one could get raw data from the sensor and process the IR images upon known calibration. However this eliminates the computational benefit provided by the sensor on-board processing. Inaccuracies in the parameters are the major source of errors in the camera system. More specifically, a wrong estimate of the baseline between the two cameras results in a systematic error in the computed depth. Inaccuracies in the relative orientation of the two cameras, result in the nonlinear distortion. An example of this distortion for an ASUS Xtion device is visible in Figure 2.

In order to better characterize these kinds of distortion, we have realized a simple simulator that generates distorted data depending on typical calibration inaccuracies. The results of some tests carried out with this simulator are explained in the remainder of this section. Furthermore, we will com-

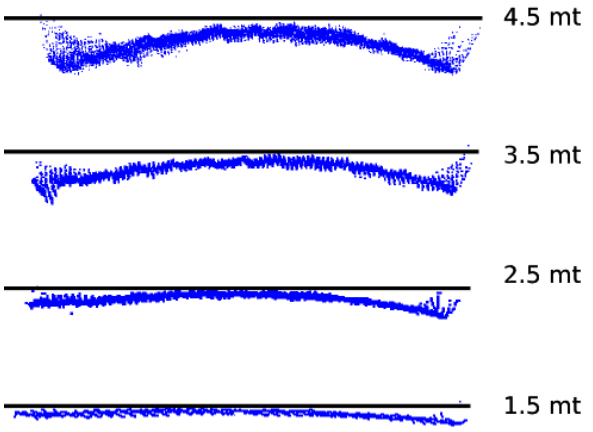


Figure 2: This figure illustrates the typical distortion of an ASUS Xtion sensor. The three images, from bottom to top, illustrate the top view of a wall parallel to the image plane captured respectively at 1.5, 2.5, 3.5 and 4.5 meters.

pare these results with the ones obtained by a real depth sensor to check the validity of the simulator model.

To highlight the effect that wrong parameters have on the depth measurements, we simulated the perception of a set of planes located at different distances from the sensor. In the simulations, the *true parameters* of the simulated device are modified with artificial noise, resulting in *perturbed parameters*. Data are obtained by projecting each point on the image planes of the two cameras, by using the true parameters. Subsequently, we reconstruct the depth of the points based on the perturbed parameters.

Figure 3 illustrates the outcome of this simulation, under different perturbations of the parameters. From the plots we can see that the orientation of the surface in the central region of the image is consistent with the orientation of perceived plane when the sensor faces the surface. However, as the incidence angle between the sensor and the plane increases, the estimate of the surface orientation in the center of the image decreases.

Additionally, errors in the parameters affect the measured depth. We can model this distortion by plotting the ratio between true and measured depth as a function of the depth:

$$d'/d = f(d). \tag{1}$$

here d' is the true depth and d is the measured depth when using a real sensor or the depth obtained with perturbed parameters in the simulations. In the remainder of this paper we refer to

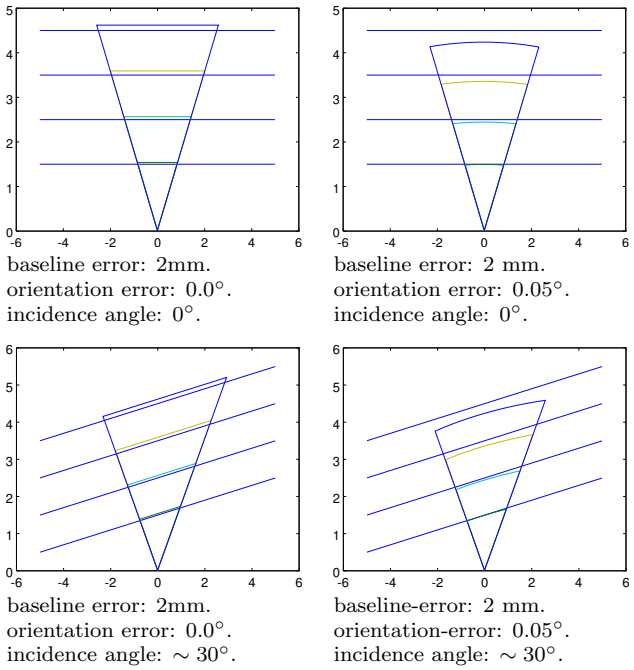


Figure 3: This figure illustrates a synthetic experiment where we reconstruct the depth of points lying at different planes by varying the incidence angle of the camera and the extrinsic parameters of the sensor.

this depth-dependent distortion in the middle of the image as *depth bias*.

This behavior is illustrated in the left part of Figure 4, that illustrates the evolution of such a ratio as the measured distance increases. We then performed the same analysis with a real ASUS Xtion sensor, and we reported the plot in the right part of Figure 4.

Furthermore, the sensor is affected by some highly nonlinear distortion at the corners of the image and in depth estimate at near range. The farther a sensor from a surface the more the corners report distances substantially smaller than the real ones, as shown in Figure 5*a*. This phenomenon

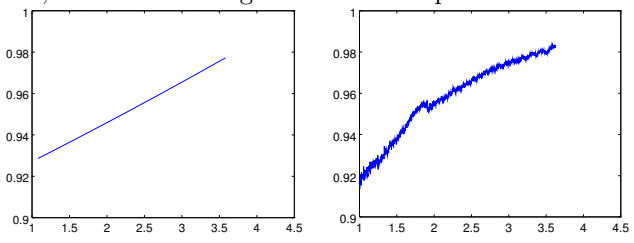


Figure 4: This figure illustrates the evolution of the ratio between true and measured distances as a function of the measured distances. The left plot illustrates the simulated result and the right plot shows the outcome of a real world analysis, from a range of around 1 meter.

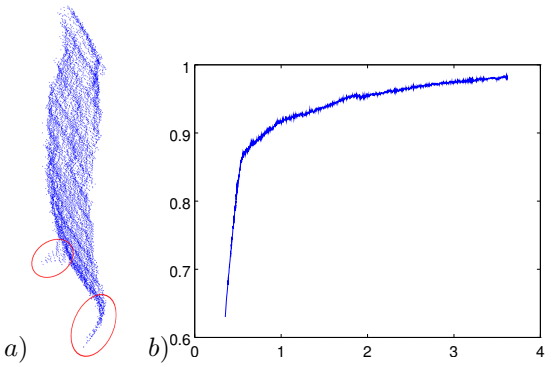


Figure 5: Non modeled effects of rgbd cameras. *a*) Side view of a point cloud reconstructed from a depth image acquired with an ASUS Xtion. The sensor was facing a planar wall at 4.5 meters. The circles highlight the high distortion affecting the corners of the image. *b*) Evolution of the depth bias at short and high range.

is likely to arise from the lens distortion and from boundary conditions of the block matcher at the corners. Whereas in the range of distances between 1 and 4 meters, the theoretical and the practical analysis report similar results at short distances the measured depth does behavior substantially differs from the theoretical one. In particular in the short range the slope is substantially higher, as shown in Figure 5b. Although these phenomena have not been explicitly modeled in our calibration model and thus not explicitly corrected, the non-parametric procedure presented in this article allows for computing a robust and accurate calibration procedure, as shown by the experimental results discussed in Section 5

In summary, when facing a depth sensor to a plane:

- the surface normals in the central region of the sensor image are very close to the plane normals,
- the depth is affected by a systematic bias that has a systematic and monotonic behavior.

4. Non-Parametric Calibration for Depth Sensors

In this section, we describe our calibration approach which is based on the computation of an *undistortion function* from a set of depth images. Our calibration procedure is composed by the following steps: 1) acquire a set of depth images while moving the sensor towards a planar surface (e.g., a wall); 2) label a few images acquired at known distances to compute a polynomial approximation of the depth bias in the central region of the image; 3) from the entire set of depth images, automatically generate samples of the undistortion function used as training data; 4) generalize and smooth the undistortion function through machine learning regression methods by using such training data.

Therefore, operator assistance may be required only for steps 1 and 2, while steps 3 and 4 are carried on automatically. In the remainder of this section, we first describe the estimation of the depth bias, then the distortion model and the computation of the training data, and finally the estimation of the undistortion function.

4.1. Estimating the Depth Bias at the center of the image

As discussed in Section 3, when facing a plane, the measured and true distances reported by a sensor are affected by bias that depends on the distance. Let d_i be a depth measured by the sensor when located at a true distance d'_i from the plane. The ratio between true and measured distances is a function of the depth, according to Eq 1. We approximate Eq 1 with a polynomial $p(d)$ as

$$d'_i/d_i = f(d_i) \simeq p(d_i) \tag{2}$$

To compute an interpolating polynomial of degree N, we need at least $N+1$ samples of measured $d_{1:N+1}$ and ground truth $d'_{1:N+1}$ distances. To this end one can use a straightforward polynomial fitting routine. In our experiments we use $N=3$ and we took $2*(N+1)$ samples uniformly spaced in the range of distances to calibrate.

4.2. Non-parametric Undistortion Model

A non-parametric undistortion model can be represented as a function

$$f(u, v, d_{uv}) \rightarrow m_{uvd} \tag{3}$$

that maps a pixel u, v in the image and a measured depth d_{uv} to a multiplier factor m_{uvd}, such that the undistorted depth d'_{uv} is obtained as $m_{uvd}d_{uv}$. In other words, the undistorted image is obtained by multiplying the depth value of each pixel u, v by $f(u, v, d_{uv})$. We also assume $f(\cdot)$ to be continuous and smooth, as confirmed by experimental evidence, but we do not make any assumption on its model, thus using a non-parametric representation.

As already mentioned, the main goal of our approach is to estimate this function with a regression method. To this end, training data with samples of $\langle u, v, d, m\rangle$ are automatically computed as explained in the next section.

4.3. Generation of the Training Data

In order to develop an automatic calibration procedure, the generation of the training data from the set of depth images must be automatic as well. In this section we describe how a training data containing tuples $\langle u, v, d, m \rangle$ is generated from a set of depth images acquired while moving towards a planar surface.

We exploit the results of the analysis presented in Section 3, namely the property that, when facing a plane, the surface normals in the center of the image are a good approximation of the normals of the plane, and that the depth is affected by a systematic bias.

Knowing an approximation of the depth bias function $p(d)$, we can straightforwardly remove the depth bias and estimate an unbiased depth $\hat{d}$ in that point as:

$$\hat{d} = p(d)d \tag{4}$$

Since, as already mentioned, the normals in the middle of the image are reliable when facing an orthogonal plan, we can fit a plane to these unbiased points $\hat{d}_i$. This plane represents a good approximation of the observed model.

Subsequently, we project all the points of the plane onto the depth image, and determine the association between measured depth values and depths on the plane. We reject outliers by taking into account: i) the Euclidean distance between each depth pixel and the corresponding depth on the plane, ii) the structural information encoded in the surface normal computed at the pixel location.

More formally, let $\mathcal{I} = \{d_{uv}\}$ be a depth image, we define a circular region $\hat{\mathcal{I}} = \{\hat{d}_{uv} \in \mathcal{I} \mid ||(u - u_0, v - v_0)|| < r\}$ around the center of the image (u_0, v_0), where the distortion is smaller. The radius of this region in the image is adapted based on the depth, to capture a region of fixed size in world coordinates. We then project the pixels $\hat{d}_{uv} \in \hat{\mathcal{I}}$ in world coordinates while removing the depth bias. Let $\mathbf{K}$ be the camera matrix, the 3D point after removal of the depth bias $\hat{\mathbf{p}}_{uv}$ is computed as

$$\hat{\mathbf{p}}_{uv} = \mathbf{K}^{-1} p(d_{uv}) d_{uv} \begin{pmatrix} u & v & 1 \end{pmatrix}^T \tag{5}$$

If the angle between the normal of the fitted plane and the z axis of the camera is greater than a threshold τ_n, we drop the image from the calibration set, since the normals of the plane might be poorly estimated. In our implementation $\tau_n = 10°$.

Once the parameters of the planar model are found, we determine how well $\hat{\mathcal{I}} = \{\hat{d}_{uv}\}$ fits the plane by counting the inliers. A point is an inlier if its distance from the plane is below a distance dependent threshold $\tau_d = \alpha_0 + \alpha_1 d_{uv}$. In our implementation $\alpha_0 = 0.1$m and $\alpha_1 = 0.05$. This adaptive threshold serves to capture also the far points that are affected by a large error. If the total number of inliers is below a threshold, we drop the image from the calibration set, since it is probably not capturing a well defined plane. The outcome of this procedure is a plane π^* that we use as reference model.

Finally, we need to determine which points in the original scene belong to the fitted plane. To this end, we consider *all* 3D points $\{\mathbf{p}_{uv}\}$ extracted from the input image $\mathcal{I}$, *without removing the depth bias.* This can be done by applying Eq. 5 with $p(d) = 1$.

In order to measure the correction factor m for each pixel, we first compute $\mathbf{p}'_{uv}$ as the intersection between the fitting plane π^* found with the above procedure and the optical ray (i.e., the ray originated at the optical center of the camera) passing through $(u\, v)$. This point $\mathbf{p}'_{uv}$ is the ideal value that would have been returned by the sensor without distortion. A point is an inlier if the distance between $\mathbf{p}_{uv}$ and $\mathbf{p}'_{uv}$ is smaller than a threshold and the angle between the normal of $\mathbf{p}_{uv}$ and the normal of π is small. For a matching pair $(\mathbf{p}_{uv}, \mathbf{p}'_{uv})$, the corrected depth d'_{uv} is the z coordinate of $\mathbf{p}'_{uv}$. Thus, the multiplier m_{uvd} is determined as d'_{uv}/d_{uv}. These multipliers will also capture the effect of the depth bias that has been used to compute the fitting plane.

Summarizing, after detecting the best fitting plane of a depth image by looking at a small central area of the image where distortion is minimal, we determine a set of pairs $(\mathbf{p}_{uv}, \mathbf{p}'_{uv})$ that represent matched points (distorted vs. undistorted points) and then determine the multipliers that allow to correct the distortion for the matched pairs. This procedure is also illustrated in Figure 6, where the selection of inliers is shown with different colors.

As a results of this first processing step, the collected data

$$\mathcal{C} = \{\langle u, v, d_{uv}, m_{uvd} \rangle^{(i)} \mid \forall\, (\mathbf{p}_{uv}, \mathbf{p}'_{uv})^{(i)}\}$$

are samples of the undistortion function we are looking for and the set of triplets $\mathcal{T} = \{\langle u, v, d_{uv} \rangle^{(i)} \mid \langle u, v, d_{uv}, m_{uvd} \rangle^{(i)} \in \mathcal{C}\}$ represent the values of u, v, d that have been sampled during the acquisition phase. Notice that we do not require $\mathcal{T}$ to be complete (i.e., to contain all the possible values for u, v, d), but we need to have a good coverage of all the values that can be obtained

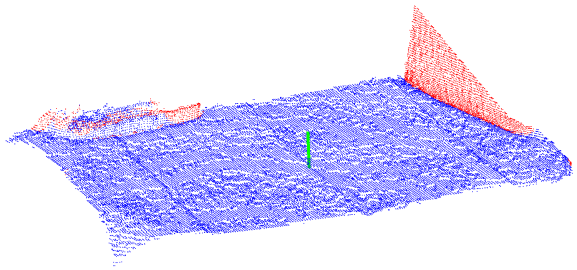


Figure 6: This figure shows the outcome of our inlier selection strategy. We consider the distance of a point as valid if the normal vector of a point is close to the plane normal and the distance of the point to the plane is below a threshold. The green line is the reference plane normal, the blue points are selected as inliers, while the red points are rejected.

by moving the sensor at different distances from a planar surface.

4.3.1. Statistics

The above procedure, applied to all the acquired depth images, returns a set of samples of the function in Eq. 3. Each sample is a quadruple $\langle u, v, d, m_{uvd} \rangle^{(i)}$ and in general multiple values of m_{uvd} can be present for the same triplet $\langle u, v, d \rangle \in \mathcal{T}$. In order to generate a more effective training set for the next approximation phase, we need a further statistical process of the samples computed so far. This procedure is also useful to reduce unavoidable noise and association errors in the extraction procedure.

For each triplet $\langle u, v, d \rangle \in \mathcal{T}$, we first compute the multi-set (i.e., set with repetitions) of values of m_{uvd} associated to the triplet. More specifically, $\mathcal{M}_{uvd} = \{m_{uvd} \mid \langle u, v, d \rangle^{(i)} \in \mathcal{T}\}$. Then, for each triplet $\langle u, v, d \rangle$ we compute and store the following values:

- $N_{[uvd]} = |\mathcal{M}_{uvd}|$, that is the number of samples falling in a bin $(u\, v\, d)$;
- $s_{[uvd]} = \sum_{i \in \mathcal{M}} m^{(i)}_{uvd}$, that represents the sum of the multipliers falling in a bin;
- $s^2_{[uvd]} = \sum_{i \in \mathcal{M}} (m^{(i)}_{uvd})^2$, that represents the sum of the square of the multipliers in a bin.

These information can be updated incrementally as new calibration data are available in constant time. When we need to recover the multiplier for a bin $(u\, v\, d)$ we can obtain the mean μ_{uvd} and the covariance σ^2_{uvd} as:

$$\mu_{uvd} = \frac{s_{uvd}}{N_{uvd}} \qquad \sigma^2_{uvd} = \frac{s^2_{uvd}}{N_{uvd}} - \mu^2_{uvd} \tag{6}$$

Our experiments confirm the intuition that the variance of the estimates grows with the distance of the sensor.

Summarizing, the statistical process described above allows to producing a training set for the undistortion function as

$$\mathcal{D} = \{\langle u, v, d, \mu_{uvd} \rangle^{(j)} \mid \textit{for each } \langle u, v, d \rangle^{(j)} \in \mathcal{T}\}$$

4.4. Function Approximation

Once the training data $\mathcal{D}$ have been generated, a regression method is adopted in order to obtain an estimation of the target function. Although these training data are affected by noise, this is limited by the thresholds and the statistical process we have used in its generation and thus they are suitable for a regression method. The choice on the function approximator depends on the quantity and quality of the data. In this article, we report our experience in using two different approaches: the first one considers the availability of a large quantity of data that may correspond to an off-line setting, while in the second one we assume very limited training data, considering a possible application of the method with lower computational resources.

In many cases the calibration procedure can be done off-line, even after the data for a mapping process have been acquired and stored. In the example reported in the next section, less than 10 seconds of acquisition returned about 45,000,000 samples. In this case, having a large quantity of data allows the use of instance based approaches. In particular, we have experimented a K-Nearest Neighbor (KNN) algorithm, where values of new instances are estimated according to the values of the neighbor instances. For efficiency reasons, we have realized a custom implementation of this algorithm.

In this setting, the undistortion function $f(u, v, d_{uv})$ is discretized in a 3D matrix, where each cell of the array contains the corresponding undistortion multiplier. The assumptions of smoothness and continuity of the function allow us to use a coarse quantization for the array, since the multipliers of neighboring points in the parameter space are close. This has the twofold benefit of reducing the number of parameters that need to be computed from the calibration data and making the estimate more robust. In our experiments, each cell of the matrix covers a region of 8×8 pixels for u, v and 64 millimeters for the depth. The values in each cell are computed by averaging the values in the training data referring to each cell. Then, missing values

are computed as a weighted average of the K values close to the query instance (we used a value of $K = 8$). Finally, a complete discrete representation of the undistortion function (i.e., a lookup table) is generated for on-line undistortion of new images.

The second case considers instead a situation in which a limited data set is available. This may be due to the necessity of performing the described calibration procedure right at the beginning of a task, for example, in the first few seconds of a mapping task. In this case, the RGBD images acquired must be processed in real-time and the computation of the approximated undistortion function must be performed during the same time. For example, assume to use the first minute of a mapping task as follows: 30 seconds to acquire images and 30 seconds to generate the undistortion function, after the first minute, the process will continue by correcting images on-line.

In this case, the generation of the training data must be reduced to analyze only a few pixels per frame (e.g., 100 pixels) randomly distributed in the image. By acquiring images at 10 Hz for 30 seconds, we will have a total number of samples in the training data of 30,000 samples (which is 3 order to magnitude less than the previous case). With this data set, approaches based on discretization are not adequate because of the high sparsity of the data. We thus experimented a function approximator based on Artificial Neural Networks (ANN). With this choice, it is possible to design the structure of the neural network in order to guarantee a reasonable training time (e.g., within less than 10 seconds) and to achieve on-line applicability when needed. More specifically, here we have used the WEKA[1] implementation of ANN and a layout with 8 sigmoid hidden units. Finally, also with ANN, a final lookup table is generated for fast on-line undistortion.

Figure 7 shows an example of function approximation of the two methods for a particular value of d. The top row shows the approximation of a dense set of data (black pixels are unknown values) with KNN, while the bottom row shows the approximation of very sparse data (only a few pixels are non-black) with ANN. The right column shows the results of the approximation where all the pixels have an undistortion value associated.

For a preliminary evaluation of the function approximations, we have performed a cross-validation

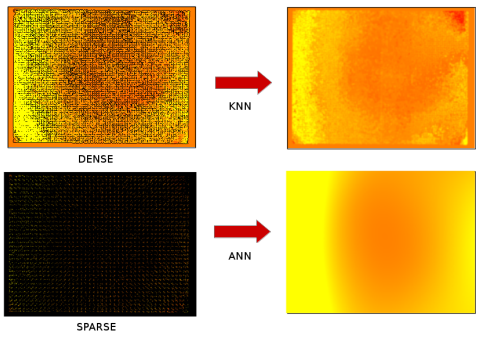


Figure 7: Example of function approximation. The left column shows the training data at different sparsity, while the right column shows the function learned by different machine learning techniques.

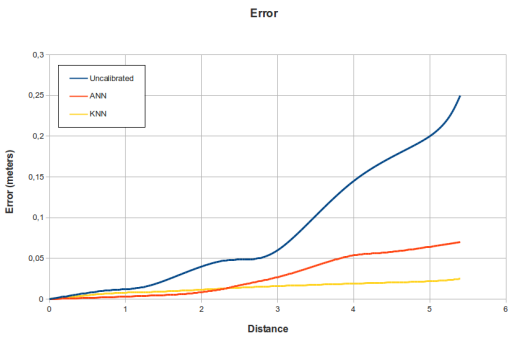


Figure 8: Estimation error of ANN and KNN undistorted measurements.

[1] http://www.cs.waikato.ac.nz/ml/weka/

procedure over the generated training data. More specifically, a set of samples generated with the procedure described in the previous section at different distances have been extracted and not used for training. Then, for each method KNN and ANN, the absolute errors between the estimated values and the generated values (assumed as ground truth) are computed. Finally, these errors have been grouped and averaged according to the distance z, as depicted in Figure 8.

The cross-validation results clearly show the advantage of using a function approximator over uncalibrated data and also that, as expected, KNN provides for better results because of the significantly larger training set. Consequently, the experiments reported in the next section are carried out with KNN method.

5. Implementation and Experimental Evaluation

The calibration method described in this paper has been fully implemented and released as an open-source project. The developed software has been tested on Linux Ubuntu 14.04 and ROS Indigo and does not require any other external software module. We provide not only the calibration software for computing the undistortion model of the RGBD sensor, but also the evaluation method that can be applied by just using a mobile robot and a 2D laser scanner.

In order to provide the users with straightforward instructions to run the proposed method and to replicate the results described in this section, we have developed the already mentioned website `http://easy_depth_calibration.dis.uniroma1.it`. This site contains additional and more detailed information about the implementation and the use of the proposed method, as well as the data sets used for evaluation and the results described here.

In all experiments we used three types of depth sensors: Microsoft Kinect, Asus Xtion Pro and Asus Xtion Pro Live. For each sensor we compare the following calibration methods:

- the approach described in [6], that does not consider the depth bias;
- the approach described in this article that applies the bias removal and the non-parametric undistortion.

We report here two kinds of experiments for evaluation of the accuracy of the calibration methods:

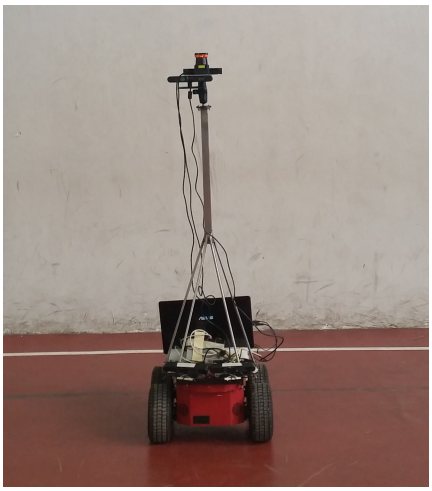


Figure 9: Mobile robot Pioneer 3 AT equipped with an Hokuyo UTM 30LX laser rangefinder used in our experiments.

1) by comparing the depth values with the measurements of a 2D laser scanner mounted on a mobile robot; 2) by comparing the results of 3D models reconstructed with a SLAM method with respect to a model built with a 3D laser.

5.1. Quantitative Evaluation with a Mobile Robot and a 2D Laser Scan

In this section, we present the results of a validation procedure using a mobile robot equipped with a 2D laser scanner and an RGBD sensor. The configuration used in our experiments (shown in Figure 9) comprises a Pioneer Robot equipped with an Hokuyo laser range finder and the RGBD sensor under test.

The core idea is to move the robot with its sensors in a planar environment in front of some vertical walls. Assuming that the scanner is mounted parallel to the ground plane, laser scan lines can be extruded into vertical planes in 3D. Finally, assuming to know the relative position of the depth sensor and the laser, the planes detected by the two sensors can be easily matched. To obtain meaningful results, the relative transformation of the laser and the depth camera has to be known with good accuracy. To this end, we developed a straightforward procedure based on least squares that seeks for the transform that better aligns the planes detected by the depth sensor and the laser, in a sequence of frames.

With this setting we recorded a dataset for each of the depth sensors, while moving forth and back while facing a large flat wall. On each dataset, we

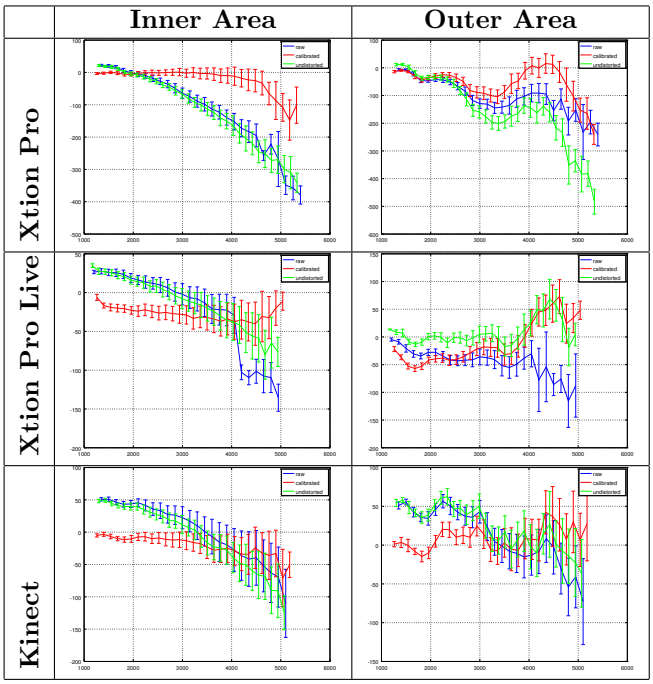


Figure 10: This figure shows the evolution of the error in the calibrated and uncalibrated case for three sensors and three approaches. The x axis reports the distance reported by the laser in mm.

executed the calibration procedure described in this article.

After the application of the calibrated model, for each corrected point lying on a plane, we seek for the distance between this point and the closest point lying on the corresponding vertical plane detected by the laser.

The results of this procedure is thus an error function depending on the distance from the sensor to the reference plane. In particular, as already mentioned, the errors between the points extracted by the RGBD sensor and corrected with the computed calibration model and the points extracted with the laser range finder are computed and used for evaluating the accuracy of the calibration procedure. The lower is the absolute value, the better is the estimate, with an ideal performance being zero.

These results are illustrated in Figure 10, showing average and standard deviation of the distance error depending on the distance from the wall. To highlight the effects of the distortion in different regions of the image we report the results in a central region of 20×20 pixels (inner area) and in a region of 20×20 pixels (outer area) located in the middle of the top left image quadrant.

As shown in the figure, the results with the calibrated sensor (red lines) are always significantly better than the raw data (blue lines), thus having the distance error close to zero. To evaluate the contribution of undistortion removal, we also plot the results obtained when only the correction of undistortion is applied (green line).

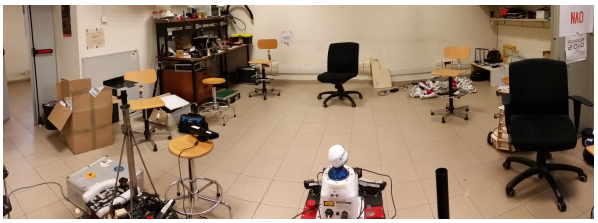


Figure 11: Setup used for the 3D reconstruction experiment. After the acquisition with the laser we moved the rotating chair with the sensor at the position of the laser, and we recorded the data while turning the chair.

5.2. Quantitative Evaluation with a 3D Reference Cloud

The second evaluation procedure described here is based on finding a best matching between the 3D map generated by a 3D SLAM algorithm using data from the RGBD sensor and a 3D ground truth point cloud, acquired by a RE05 3D laser scanner from Ocular Robotics. To align the maps we used Generalized ICP [14].

After the matching, we consider the number of inliers, that are points in the two clouds closer than 0.1 meters and whose normals have an angle closer than 10 degrees. We then compute the residual error of the inliers, which is the sum of squared distance between corresponding points. Ideally, if the two clouds match perfectly, the residual is zero with 100% inliers. Higher values of the residuals and lower inlier ratios denote larger misalignments.

The residual error is not an absolute measure and depends on the path taken, however it is useful to compare the map generated by uncalibrated data and the calibrated data generated from them. To lessen the effect of the path and of the occlusions between the laser cloud and the one obtained with a depth sensor, we placed each sensor on a rotating chair, at the same position of the laser, and we acquired a single sweep by rotating the chair (Figure 11). We could not acquire the datasets for the three sensors simultaneously since we experienced interference effects.

The results of this experiment are summarized in Figure 12. The figure provides for a visual feedback of the residual error, although it only highlights errors visible from a particular view. As shown by both the residual error and the visual feedback, our

approach is always better than the uncalibrated data and than the approach in [6] that does not compensate for the depth bias. In particular, both the error and the number of inliers improve when applying a calibration process. Moreover, the approach proposed in this article further improve with respect to the previous approach.

6. Conclusion

In this article, we have presented a generic approach for automatic calibration of depth sensors that does not require any kind of external device and of external software module. Our method differs from existing approaches in the rapidity and the easiness of the calibration procedure. Full implementation and extensive experimental results show that calibrated sensors can provide significant improvements in 3D reconstruction and mapping of the environment. Moreover, we have provided a method for validating the calibration process using a mobile robot equipped with a 2D laser range finder that can be used to assess the validity of the calibration procedure.

Although the results clearly show improved performance, a more detailed analysis for optimizing the function approximation phase can further increase the benefit of using the proposed method.

From a practical perspective, our work makes the calibration procedure of a sensor a task that requires less than a minute. Both the training phase and the correction phase can be performed either off-line and manually or as an automatic procedure on a mobile robot, just before the start of a task that requires the use of calibrated depth images.

Thus, we believe that the easy procedure described in this article will improve the quality of several robotic applications using RGBD sensors.

References

[1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, , P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon, KinectFusion: Real-time dense surface mapping and tracking, in: Proc. of the Int. Symposium on Mixed and Augmented Reality (ISMAR).

[2] F. Steinbrücker, J. Sturm, D. Cremers, Real-time visual odometry from dense RGB-D images, in: ICCV Workshops.

[3] P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox, Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments, ijrr 31 (2012).

[4] M. Ruhnke, R. Kümmerle, G. Grisetti, W. Burgard, Highly accurate 3d surface models by sparse surface adjustment, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA).

[5] A. Teichman, S. Miller, S. Thrun, Unsupervised intrinsic calibration of depth sensors via SLAM, in: Robotics: Science and Systems.

[6] M. Di Cicco, L. Iocchi, G. Grisetti, Non-parametric calibration for depth sensors, in: Proc. of the 13th International Conference on Intelligent Autonomous Systems. (IAS 13).

[7] H. Yamazoe, H. Habe, I. Mitsugami, Y. Yagi, Easy depth sensor calibration, in: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 465–468.

[8] S. Fuchs, G. Hirzinger, Extrinsic and depth calibration of tof-cameras, in: Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR).

[9] J. Smisek, M. Jancosek, T. Pajdla, 3d with kinect, in: ICCVws.

[10] C. Nguyen, S. Izadi, D. Lovell, Modeling kinect sensor noise for improved 3d reconstruction and tracking, in: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on.

[11] C. Zhang, Z. Zhang, Calibration between depth and color sensors for commodity depth cameras, in: Multimedia and Expo (ICME), 2011 IEEE Int. Conf. on.

[12] F. Basso, A. Pretto, E. Menegatti, Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pp. 6244–6249.

[13] A. Canessa, M. Chessa, A. Gibaldi, S. P. Sabatini, F. Solari, Calibrated depth and color cameras for accurate 3d interaction in a stereoscopic augmented reality environment, Journal of Visual Communication and Image Representation 25 (2014) 227–237.

[14] A. V. Segal, D. Haehnel, S. Thrun, Generalized-ICP, in: Proc. of Robotics: Science and Systems (RSS).

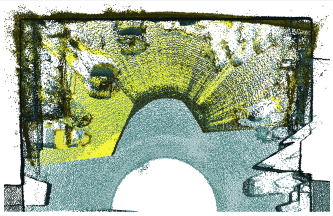
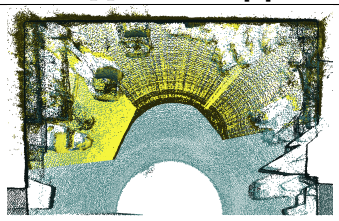
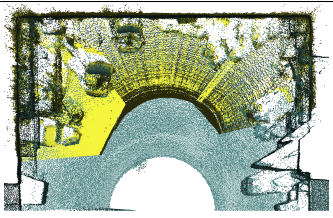
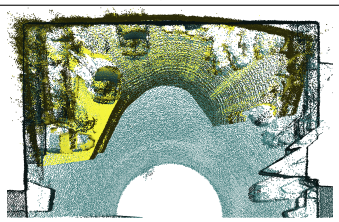
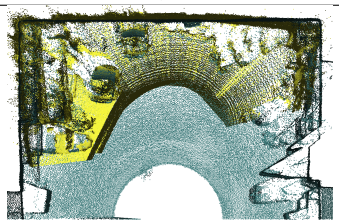
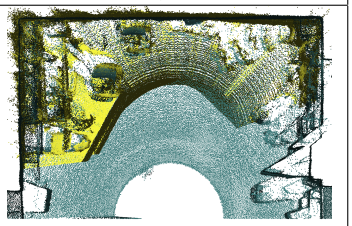
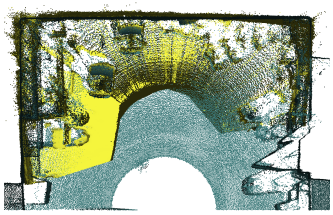
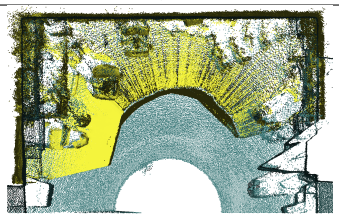
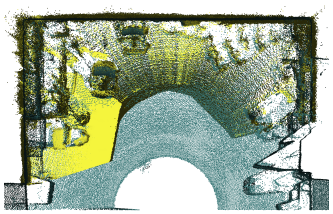
	Uncalibrated	**Approach in [6]**	**Our Approach**
Xtion Pro	 $e = 8.0, i = 82\%$	 $e = 6.1, i = 84\%$	 $e = 5.4, i = 86\%$
Xtion Pro Live	 $e = 10.0, i = 84\%$	 $e = 8.0, i = 85\%$	 $e = 6.6, i = 85\%$
Kinect	 $e = 6.4, i = 82\%$	 $e = 6.2, i = 81\%$	 $e = 5.6, i = 81\%$

Figure 12: This figure illustrates the results of our calibration procedure when employed to feed a 3D reconstruction algorithm. The blue/green point cloud is a reference cloud acquired by a 3D laser scanner. The light yellow cloud is the reconstruction of a scene operated with different sensors, and aligned to the reference through ICP. e represents the residual error in mm per inlier, i is the fraction of inliers found by the matching.