Depth Correction for Depth Cameras From Planarity

Amira Belhedi[123]
amira.belhedi@cea.fr
Adrien Bartoli[2]
http://isit.u-clermont1.fr/~ab/
Vincent Gay-Bellile[1]
vincent.gay-bellile@cea.fr
Steve Bourgeois[1]
Steve.BOURGEOIS@cea.fr
Patrick Sayd[1]
Patrick.SAYD@cea.fr
Kamel Hamrouni[3]
kamel.hamrouni@enit.rnu.tn

[1] CEA, LIST, LVIC,
F-91191 Gif-sur-Yvette, France.

[2] Clermont Université, Université d'Auvergne, ISIT, BP 10448, F-63000 Clermont-Ferrand, France.

[3] Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LR-SITI Signal Image et Technologie de l'Information, BP-37, Le Belvédère, 1002 Tunis, Tunisia.

Abstract

Depth cameras open new possibilities in fields such as 3D reconstruction, Augmented Reality and video-surveillance since they provide depth information at high frame-rates. However, like any sensor, they have limitations related to their technology. One of them is depth distortion. In this paper, we present a method to estimate depth correction for depth cameras. The proposed method is based on two steps. The first one is a non-planarity correction that needs depth measurement of different plane views. The second one is an affinity correction that,contrary to state of the art approaches, requires a very small set of ground truth measurements. Thus, it is more easy to use compared to other methods and does not need a large set of accurate ground truth that is extremely difficult to obtain in practice. Experiments on both simulated and real data show that the proposed approach improve also the depth accuracy compare to state of the art methods.

1 Introduction

Real time depth measurement is an important requirement for many applications such as collision prevention and motion interpretation. Until recently, available systems have been limited mainly to laser scanners and stereo vision. The former does not work in real time or is limited to a small part of the scene while the latter might drop accuracy on low textured scenes. Recently, promising active sensors have been developed that outperform past technologies. Two families of active sensors can be distinguished: Time-of-Flight (TOF) sensor and structured light sensor such as the Microsoft's Kinect. They provide depth images in real time even for poorly textured scenes. However, they have a limitation: they are subject to depth measurement distortion. In this article, we propose a new depth correction method, that do not impose a physical model and thus can be used for any depth camera. In the following, we focus on TOF camera since it has been used for the experiments. Some works, *e.g.* [2, 4, 11] have been devoted to understanding the causes of depth distortion related to TOF technology and measuring them. The most critical cause, that we called *systematic error*, is due to the fact that the modulation signal (see [5] for TOF principle) is not a perfect sinusoidal. There are other causes such as the reflectivity of the objects in the scene, the integration time, the incident light with respect to the objects' orientation and the camera's low resolution (lack of accuracy at depth discontinuities).

In this article we present a new model for the correction of the *systematic error*. Some models has been presented in literature. Simple model [10] try to describe the depth distortion by a linear function that depends on the depth measured and the pixel position in the image. More complex models use a look-up table [3] or B-spline function [6] for a global distortion correction in addition with a simpler model for a per-pixel distortion correction. A non-parametric model [1] describes the depth distortion by a discrete function.

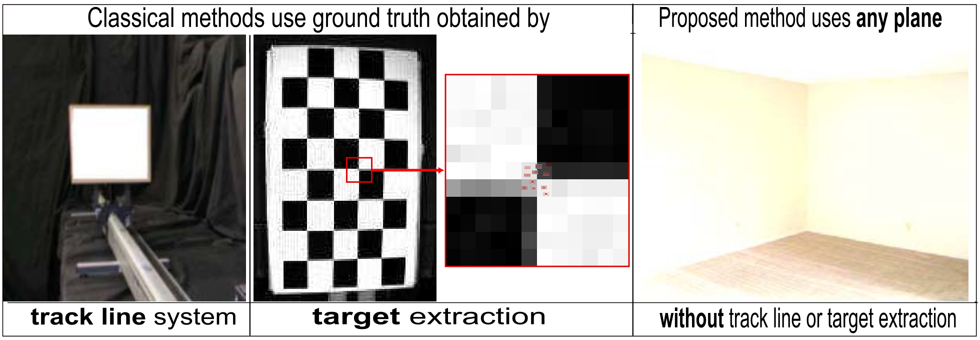

Classical methods use ground truth obtained by
Proposed method uses **any plane**
track line system
target extraction
without track line or target extraction

Figure 1: Classical approaches require a set of accurate ground truth that are obtained by track line system or target extraction approach. The first system is expensive. The second approach does not provide accurate ground truth: it is not feasible to extract accurate point due to the camera's low resolution (lack of accuracy at transition area): the red crosses represent the different possibilities of a corner localization. Our approach uses planar views and does not need a large number of ground truth.

The main limitation of these methods is that they require an accurate ground-truth for each depth-pixel (*e.g.* from 0.75m to 7.5m with a step of 10cm for each pixel in [6]). However, acquiring these reference depths is extremely difficult for several reasons. In fact, an additional system is required, *i.e.* high accuracy track line as in [3, 6] or a calibrated color camera as in [1, 7, 10] (see Figure 1). The former system is expensive and is only used to provide the measurement of a single pixel per image. The latter one requires an accurate stereo calibration that can not be reached due to the low accuracy of targets extraction caused by the ToF images' characteristics (low-resolution, noise, *etc*). In contrast, the proposed method is more easy to use. It is based on two steps. The first one is a non-planarity correction where depth measurements of different plane views are required which is easy to set up. The second one is an affinity correction where a small set of ground truth is needed. 10 reference measurements is used in the implementation that are computed by target extraction approach. Thus, the advantage of our calibration approach is the ability to use it easily, and to be able to recalibrate on-line. Simulated data are used to validate our approach and to compare the obtained results with two methods from the literature. We also present experimental results on real data.

Plan. This article is structured as follows. A description of our method is presented in Section 2. The solution of the main step of the presented approach is detailed in Section 3. The results obtained using both simulated and real data and a comparison with state of the art approaches are presented in Section 4. Finally, conclusions and further work are discussed in Section 5.

Notation. $\mathbf{q}$ is the depth-pixel (3-vector) and $\mathbf{Q}$ is the corresponding 3D point in the camera coordinate frame defined as: $\mathbf{q}^\mathsf{T} = (u \quad v \quad d) \quad \mathbf{Q}^\mathsf{T} = (X \quad Y \quad Z)$, where $(u \; v)$ the pixel coordinates and d the associated depth. Homogeneous coordinates are written as $\check{\mathbf{Q}} = (\mathbf{Q}^\mathsf{T} \quad 1)$.

2 Depth Correction From Planarity

The proposed approach operates on the 3D space (correction of the 3D point $\mathbf{Q}$), in contrast to the most existing methods which operate in the 2.5D space (correction of $d(u,v)$). We demonstrate in the appendix that it is equivalent since it exists a transformation (Equation (14) between the two spaces. Our approach is composed of two steps:

Non-planarity correction (NPC): estimates a correction function $F : \psi \to \mathbb{R}, \psi \subset \mathbb{R}^3$ such that, $F(\mathbf{Q}) = C_Z$ where ψ is a subset of $\mathbb{R}^3$: $\psi = [X_{min}; X_{max}] \times [Y_{min}; Y_{max}] \times [Z_{min}; Z_{max}]$ and C_Z is a scalar that represents the Z correction. F lies in $L^2(\psi)$ [1]. NPC is based on training F: collecting a massive set of different views (different orientations and different distances) of a plane that intersect to cover all the 3D calibrated space (see Figure 4(a)), which is easy to set up. The 3D points of each view are not coplanar. This is caused by the depth distortion. The NPC principle is to train F such that the corrected points of each view tend toward coplanar points. This will constraint F up to a global 3D affine transformation A.

Affine correction (AC): estimates an affine transformation A. Any affine transformation of the corrected space will keep the planarity constraints. Estimating A (12 parameters) requires to collect a small set of ground truth measurements. AC will end up as linear least squares constraints and can be easily solved. In the following, we focus on the NPC step, since AC is trivial. The NPC is first formulated as a variational problem and it is then resolved using an iterative process.

2.1 Variational Problem Statement of The NPC

Let our input data be n views with m points: $\mathbf{Q}_{ij} \overset{\text{def}}{=} (X_{ij} \;\; Y_{ij} \;\; Z_{ij})^\mathsf{T}$, where i is the plane view number and j is the position of pixel in the image. The goal is to undistort the points $\{\mathbf{Q}_j\}_{j=1}^m$ so that they become coplanar. The constraints are :

$$\mathbf{S}_{ij,\;\; j=1,\ldots,m} \in \Pi_{i,\;\; i=1,\ldots,n},$$

where $\mathbf{S}_{ij}$ is expressed from Equation (14) as $\mathbf{S} = \left(\; X + \frac{1}{Z}XC_Z \quad Y + \frac{1}{Z}YC_Z \quad Z + C_Z \; \right)$ is the corrected point and can be rewritten as $\mathbf{Q} + \frac{1}{Z}F(\mathbf{Q})\mathbf{Q}$ and Π_i is an unknown 3D plane in the corrected space, corresponding to view i. We represent a plane Π_i by a vector $\mathbf{P}_i \in \mathbb{R}^3$ whose norm is the inverse of the distance to the camera center and that is collinear to the

[1] The Hilbert space of square-integrable functions

plane's normal. The set of the unknown planes $\mathcal{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_n\} \in \mathbb{R}^3$ are latent variables. The problem statement is formulated as the minimization of a cost functional C:

$$\min_{F \in L^2, \mathcal{P}} C[F, \mathcal{P}], \tag{1}$$

which is composed of a data term C_d, a regularization term C_s and a regularization weight λ:

$$C[F, \mathcal{P}] \overset{\text{def}}{=} C_d[F, \mathcal{P}] + \lambda C_s[F]. \tag{2}$$

Data term. It minimizes the distance between the corrected points S and the plane Π:

$$C_d[F, \mathcal{P}] \overset{\text{def}}{=} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\mathbf{S}_{ij}, \Pi_i), \tag{3}$$

where d corresponds to the point-to-plane Euclidean distance:

$$d^2(\mathbf{S}, \Pi) \overset{\text{def}}{=} \frac{(\mathbf{S}^{\mathsf{T}}\mathbf{P} + 1)^2}{\|\mathbf{P}\|_2^2}. \tag{4}$$

Regularization term. It is the 'bending' energy defined as:

$$C_s[F] \overset{\text{def}}{=} \int_{\psi} \left\| \frac{\partial^2 F}{\partial \mathbf{Q}^2} \right\|_2^2 d\mathbf{Q}. \tag{5}$$

2.2 Iterative Resolution of The NPC

The objective is to estimate F minimizing C (Equation (1)). It corresponds to a non-convex and non-linear optimization problem where both $\mathcal{P}$ and F have to be estimated. Since it is difficult to simultaneously estimate $\mathcal{P}$ and F, an iterative process is adopted, which alternates the estimation of $\mathcal{P}$ and the estimation of F. The different steps of the iterative process is resumed in (Algorithm 1). Initially, F is set to no correction (F^0), then it is updated at each iteration (k is the iteration number) getting closer and closer to the solution. It is convenient to view the algorithm as two alternatively minimization steps. The first one consists on fixing F^k to estimate $\mathcal{P}^k$ (EP step). The second one consists on fixing $\mathcal{P}^k$ to estimate F^{k+1} (EF step). For the EP step, $\mathcal{P}^k$ that minimizes $C_d[F^k, \mathcal{P}]$ is estimated. First, the corrected points $\{\mathbf{S}_{ij}\}_{j=1}^{m}$ are computed from F^k. After that, the plane Π_i that best fits these points by minimizing point-to-plane distance is estimated. This must be done for all views to obtain $\mathcal{P}^k$. The EP step corresponds to a linear least square minimization problem which can be easily solved. Now, $\mathcal{P}^k$ is fixed and F^{k+1} that minimizes $C[F, \mathcal{P}^k]$ is estimated (EF step). This step depends on the used correction model. We choose a 3D smoothing spline, known as a 3D Thin-Plate-Spline to model the correction function. A solution of the EF step with this model is presented in Section 3.

3 Correction Function Modeling and Estimation

The depth correction depends on X, Y and Z position. These variations cannot be well modelled by a simple linear function. A more complex model is needed. It must also provide continuity, since it must gives for each 3D point the associated Z correction. A 3D Thin-Plate-Spline function is therefore chosen, since it verifies all these conditions. In this section, the 3D TPS model is first presented and the solution for the EF step (of the NPC process) is then detailed.

Algorithm 1 Iterative resolution of the NPC

```
$k \leftarrow 0$ : first iteration.
$F^k \leftarrow 0$ : initialize $F$ to no correction (0 is the 'zero' function).
repeat
    $\mathcal{P}^k \leftarrow \arg\min_{\mathcal{P}} C_d[F^k, \mathcal{P}]$ (EP step).
    $F^{k+1} \leftarrow \arg\min_{F \in L^2} C[F, \mathcal{P}^k]$(EF step).
    $k \leftarrow k+1$.
until $F^{k+1} \approx F^k$ or the preset iteration number (20 in the implementation) is achieved.
```

3.1 Correction Function Modeling with 3D TPS

A 3D-TPS $\mathbb{R}^3 \to \mathbb{R}$ is a smooth function [8] known to be an efficient approximation to many types of deformation and that minimizes the 'bending energy'. It is flexible, controlled by l 3D centres $\mathbf{c}_k$ ($\mathbf{c}_k \in \mathbb{R}^3, k = 1, \ldots, l$) that may be placed anywhere in the space. It is usually parametrized by an $l+4$ coefficient vector $\mathbf{h}^\top = \begin{pmatrix} \mathbf{w}^\top & \mathbf{a}^\top \end{pmatrix}$. There are l coefficients in $\mathbf{w}$ and four coefficients in $\mathbf{a}$. The coefficients in $\mathbf{w}$ must satisfy $\check{\mathsf{P}}^\top \mathbf{w} = \mathbf{0}$, where the kth row of $\check{\mathsf{P}}^\top$ is given by $\begin{pmatrix} \mathbf{c}_k^\top & 1 \end{pmatrix}$. These four 'side-conditions' ensure that the TPS has square integrable second derivatives. Let $\ell_{\mathbf{Q}}^\top = ((d(\mathbf{Q}, \mathbf{c}_1)) \ \cdots \ (d(\mathbf{Q}, \mathbf{c}_l)) \ \mathbf{Q}^\top \ 1)$, 3D-TPS at point $\mathbf{Q}$ is:

$$\omega(\mathbf{Q}, \mathbf{h}) = \ell_{\mathbf{Q}}^\top \mathbf{h} = \left(\sum_{k=1}^{l} \mathbf{w}_k d(\mathbf{Q}, \mathbf{c}_k) \right) + \mathbf{a}^\top \check{\mathbf{Q}}. \tag{6}$$

We use a 3D-TPS to model the correction function F by defining a set of l centres positioned throughout the working volume (Figure 4(a)): $F(\mathbf{Q}) \overset{\text{def}}{=} \ell_{\mathbf{Q}}^\top \mathbf{h}$. This parametric function is chosen for many reasons. First, it efficiently approximates the Z correction being considered as a deformation. Second, it limits the memory requirement, in fact, only the $l+4$ parameters and the l centers have to be saved. l is the unknown hyper-parameter included in this model that have to be estimated (see Section 4.2). The corrected point $\mathbf{S}$ can be expressed by:

$$\mathbf{S} = \begin{pmatrix} X + \frac{1}{Z} X \ell_{\mathbf{Q}}^\top \mathbf{h} & Y + \frac{1}{Z} Y \ell_{\mathbf{Q}}^\top \mathbf{h} & Z + \ell_{\mathbf{Q}}^\top \mathbf{h} \end{pmatrix}. \tag{7}$$

3.2 Correction Function Estimation

Data term. The data term (Equation (3)) can be rewritten from (Equation (7) and (4)) as:

$$C_d(\mathbf{h}, \mathcal{P}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{\frac{1}{Z_{ij}} \mathbf{P}_i{}^\top \mathbf{Q}_{ij} \ell_{\mathbf{Q_{ij}}}^\top}{\|\mathbf{P}_i\|_2} \mathbf{h} + \frac{\mathbf{P}_i{}^\top \mathbf{Q}_{ij} + 1}{\|\mathbf{P}_i\|_2} \right)^2 = \|\mathsf{D}\mathbf{h} - \mathbf{b}\|_2^2, \tag{8}$$

where $\mathsf{D} \in \mathbb{R}^{nm \times (l+4)}$ and $\mathbf{b} \in \mathbb{R}^{nm \times 1}$. The four 'side-conditions' (see Section 3.1) are necessary for each coefficient of $\mathbf{w}$ to ensure that (8) has a solution. Let $\mathsf{V}_2 \in \mathbb{R}^{(l+4) \times l}$ be the matrix corresponding to the l vectors of the $(\check{\mathsf{P}}^\top \ \ \mathbf{0})$ null space basis. The null space is calculated numerically by SVD. When these conditions are incorporated into (Equation (8)), replacing $\mathbf{h}$ by $\mathsf{V}_2 \mathbf{h}_1$ ($\mathbf{h}_1 \in \mathbb{R}^l$), we obtain:

$$C_d(\mathbf{h}, \mathcal{P}) = \|\mathsf{D}\mathsf{V}_2 \mathbf{h}_1 - \mathbf{b}\|_2^2. \tag{9}$$

Regularization term. It is defined (for 3D TPS) as:

$$C_s(\mathbf{h}) \overset{\text{def}}{=} \mathbf{w}^\mathsf{T} \mathsf{K} \mathbf{w}. \tag{10}$$

To rewrite this equation as a matrix norm depending on $\mathbf{h}_1$, the matrix K can be expressed as $\mathsf{K} = \mathsf{B}^\mathsf{T}\mathsf{B}$ with $\mathsf{B} \in \mathbb{R}^{l \times l}$ and $\mathbf{w}$ can be rewritten as: $\mathbf{w} = \mathsf{V}_3 \mathbf{h}_1$, where $\mathsf{V}_3 \in \mathbb{R}^{l \times l}$ corresponds to the l first lines of V_2. Thus (Equation (10)) can be write as:

$$C_s(\mathbf{h}) = \|\mathsf{B}\mathsf{V}_3\mathbf{h}_1\|_2^2. \tag{11}$$

EF Solving. Minimizing C is equivalent to solving the following system obtained by combining (Equation (9)) and (Equation (11)):

$$C(\mathbf{h}, \mathcal{P}) = \left\| \begin{pmatrix} \mathsf{D}\mathsf{V}_2 \\ \sqrt{\lambda}\mathsf{B}\mathsf{V}_3 \end{pmatrix} \mathbf{h}_1 - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2^2. \tag{12}$$

The matrix D is a very large system of linear equations. In fact, a large number of 3D points is used ($n \times m = 46 \times (204 \times 204)$ more details in Section 4.1). A part of the calibrated space ranged approximately from 1m to 2.5m is shown in the Figure 4(a). We cannot save D in memory, thereafter, in our implementation, we compute directly the matrix $\mathsf{T} = \mathsf{D}^\mathsf{T}\mathsf{D}$, $\mathsf{T} \in \mathbb{R}^{(l+4)\times(l+4)}$ without forming the D matrix (the computation of T is very time consuming). Thus, the system Equation (12) can be solved by the pseudo-inverse technique: $\mathbf{h}_1 = \left(\mathsf{V}_2{}^\mathsf{T}\mathsf{T}\mathsf{V}_2 + \lambda \mathsf{V}_3{}^\mathsf{T}\mathsf{K}\mathsf{V}_3\right)^{-1} \mathsf{V}_2{}^\mathsf{T}\mathsf{D}^\mathsf{T}\mathbf{b}$. Finally, the 3D-TPS parameters $\mathbf{h}$ is obtained by:

$$\mathbf{h} = \mathsf{V}_2 \mathbf{h}_1. \tag{13}$$

4 Experimental Results

In this section, we evaluate the accuracy of the proposed method and compare it with two methods from the literature on simulated data. A second experience on real data is also performed to evaluate the NPC. In the following, the experimental protocol is first presented and the results of the two experiences are then discussed.

4.1 Experimental Protocol

The data set used for the two experiences corresponds to different views of a plane that intersect to cover all the calibrated 3D space. It is divided into three parts: training, test and validation data set. The training set is used to perform different calibrations (NPC and AC estimation) by varying the set of hyper-parameters (l,λ). These calibrations are then evaluated on the test set to select the optimal (l,λ) values (that minimize the RMS of the point-to-plane Euclidean distance). The final results used to evaluate our method are obtained on the validation set using the optimal set of (l,λ). The simulated data are obtained by distorting a plane with a distortion function that varies according to the distance and increases from the image centre to the image boundaries. They cover a distance range from 1m to 3m. A training set of 36 plane views is used (Figure 4(a)). For the real data, the TOF camera used is a PMD CamCube2 with a resolution of 204×204 pixels [9] and the observed object corresponds to a white wall. The data cover a distance range from 2m to 7m. A training set of 46 plane views is used. The AC is performed using a set of 10 reference data. These data represent the central pixel of 10 reference plane (shown in Figure 4(f)). For both experiences, the test and validation set contain each 10 plane views.

4.2 Experience with Simulated Data

Calibration. The calibration steps are shown in Figure 4. Initially ($k = 0$), the point-to-plane distance is large (Figure 4(c)), it decreases at the next iteration (Figure 4(d)) and become very small in the last iteration (Figure 4(e)). After NPC step, the points are coplanar but not aligned with ground truth Figure 4(g). The AC step is then performed. It is shown in Figure 4(h) that after AC the obtained data are very close to the ground truth.

Optimal set of hyper-parameters (l,λ). We have to determine the optimal values of (l,λ). Different calibrations are performed by varying (l,λ) and then tested. The results shown in Figure 2 represent the RMS error computed after NPC. They are represented by a mesh with $x-$ and $y-$ coordinates are respectively λ and l. It shows that, from $l = 5^3$ (5 centres over $X-$, $Y-$ and $Z-$), the system is stable over a large range of λ (blue part). Note that l must be chosen as small as possible to limit computation time, while not degrading the results. For this reason, $l = 5^3$ is chosen, since it is the smallest value that gives a small RMS. Now, l is fixed, we have to determine the optimal λ value. We propose to use the golden section search. It is a fast and robust minimum search method. The optimal λ value computed with this method is equal to 0.12 (Rms = 1.36 mm).

Results. The proposed method is evaluated on the validation set. An example of a fronto-parallel view is chosen in Figure 3. This view is situated at a distance of 1.20m from the camera. The distortion increases from the image centre to the image boundaries (the maximum distortion is 60mm). After the NPC, a perfect plane is obtained (Figure 3(c)): all points are coplanar and situated at 1.22m; however, they are not aligned with the ground truth (1.20m). It is shown in the Figure 3(d) that after the AC the final corrected plane is very close to the ground truth. To evaluate the proposed method over all the pixels of the validation data set, we compare the point-to-plane distance before and after correction. For the NPC validation, the distance to the plane that best fits the points ($d(S,\Pi_S)$) is considered, while for the AC validation, the distance to the ground truth plane $d(S,\Pi_{GT})$ is used. The results are presented in Table 1. We observe an important improvement. After NPC, the RMS error is equal to 1.36 mm whereas it is equal to 15.64 mm before correction. After AC, the RMS error is equal to 2.27 mm whereas is equal to 24.22 mm before correction. An improved depth image accuracy is obtained with the proposed approach.

Comparison. In order to evaluate the proposed approach, a comparison with two methods from the literature is performed. The depth calibration methods proposed in [6] and in [3] have been implemented and tested on the same validation data set. The distance to the ground truth plane $d(S,\Pi_{GT})$ is used to compare results (see Table 2). The obtained results are as good as [6]: the RMS error remaining after depth correction is equal to 2.27 mm whereas it is equal to 2.99 mm with [6]. We observe an accuracy improvement with our approach compared to [3] (remaining RMS is equal to 6.59 mm). In fact, the global distortion is not well estimated (just the pixel at the image center is considered) with this method.

4.3 Experience with Real Data

The proposed method is evaluated on the validation set of real data. The optimal values of l,λ are respectively $6^3, 0.4$ (determined by the golden section search method). An example of the wall view is chosen in Figure 5. After the NPC step, the 3D points are coplanar. A comparison of the point-to-plane distance before and after correction is also performed to evaluate the accuracy of the proposed method over all the pixels. Before correction, the

RMS is equal to 27.63 mm, while after correction it is equal to 8.03 mm. The remaining RMS is considered sensor noise. In fact, an evaluation of the sensor noise is performed, the standard deviation of the wall is computed out of 100 measurements in every pixel. It varies from 5 mm to 18 mm. In spite of noise, a good results are obtained.

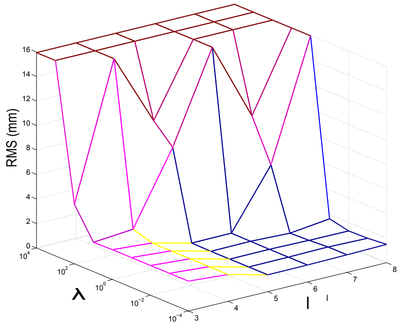
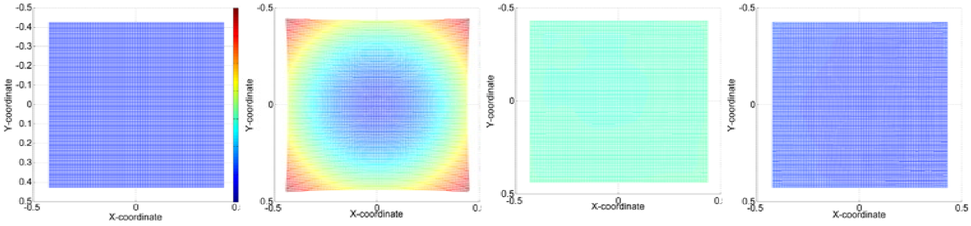


Figure 2: Point-to-plane distance measured after NPC by varying hyper-parameters values. Mesh representation with l range from 3 to 8 and λ range from $10^{-}4$ to 10^4 in logarithmic scale. The optimal set of parameters is $\lambda = 0.12$ and $l = 5^3$ (represented by diamond)



(a) (b) (c) (d)

Figure 3: Fronto-parallel view of the plane (a) undistorted (ground truth) (b) before correction, (c) after NPC and (d) after AC

	before NPC	**after NPC**
$d(S, \Pi_S)$	15.64	1.36
	before AC	**after AC**
$d(S, \Pi_{GT})$	24.22	2.27

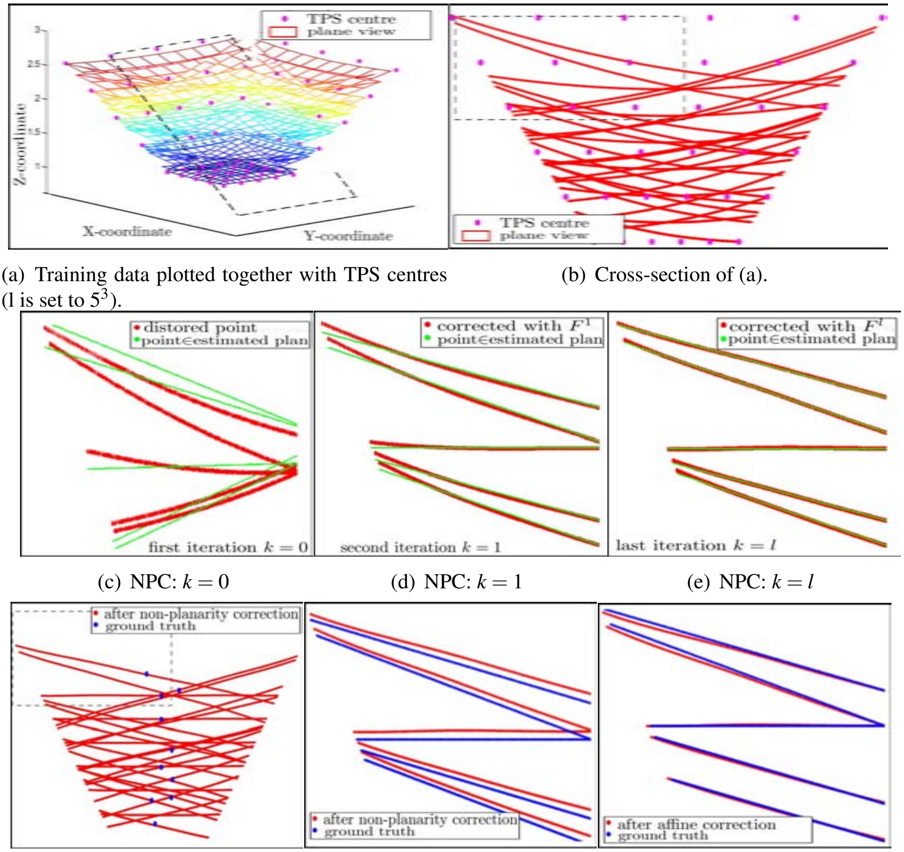
Table 1: RMS error (mm)

	Depth errors (mm)
Uncalibrated	29.17
Our method	**2.27**
[6]	2.99
[3]	6.59

Table 2: Comparison with two approaches from the literature

5 Conclusion

We presented a depth correction method based on a non-planarity correction that requires a large set of plane views and an affinity correction that needs a very small set of ground truth measurements. A 3D TPS is used to model the correction function. This type of model is used, to our knowledge, for the first time in this context. Our method has the advantage to be easy to use compared to the most methods from the literature. In fact, it does not need a ground truth for each depth measurement. Experimental results on both simulated and real data demonstrate the validity and accuracy of the proposed approach. Future work will improve the results by applying a more robust denoising filter [8] to the depth image before correction estimation. It would also be interesting to test the proposed approach for depth images provided by the Kinect sensor.



(a) Training data plotted together with TPS centres (l is set to 5^3).

(b) Cross-section of (a).

(c) NPC: $k = 0$

(d) NPC: $k = 1$

(e) NPC: $k = l$

(f) After NPC

(g) Before AC

(h) After AC

Figure 4: Simulated data results during depth correction process. (a)A part of calibrated space ranged from 1m to 2.5m. A part of (b) is considered to show obtained results at (c) first iteration, (d) second one and (e) last one of NPC. (f)A small set of reference data used to compute A plotted together with the corresponding section of training data obtained after NPC. Comparison of results (g) before and (h) after AC.

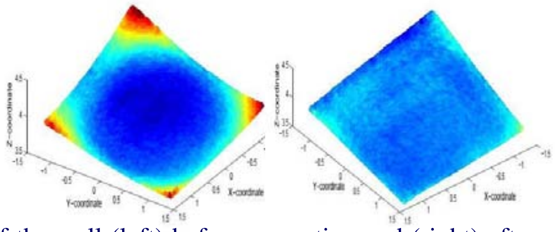


Figure 5: A view of the wall (left) before correction and (right) after correction (the color represents the point-to-plane distance)

APPENDIX

Equivalence Between Correction Functions f and F

The depth correction is generally modelled by a function $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^3$ that associates to each point $\mathbf{q}$ of the space the corresponding correction such that: $f(\mathbf{q}) = c$ where Ω is a

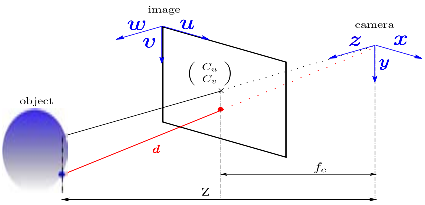


Figure 6: Distance d versus 3D point $\mathbf{Q}$

subset of $\mathbb{R}^3$: $\Omega = [u_{min}; u_{max}] \times [v_{min}; v_{max}] \times [d_{min}; d_{max}]$ and c is a scalar that represents the depth correction. In other word, $d + c$ is the corrected depth. The proposed method estimates an equivalent function F (see Section 2) that corrects $\mathbf{Q}$. We show in the following that it exists a transformation between the two function spaces. We assume that the camera's intrinsics parameters are known, thus, the transformation from $\mathbf{q}$ to $\mathbf{Q}$ in metric space can be estimated (as shown in the Figure 6). We call $(c_u \quad c_v)$ the optical center on the sensor array, f_c the camera focal length, $(d_u \quad d_v)$ the pixel pitch in the u (resp. v) direction. Neglecting lens distortion, the transformations between $\mathbf{q}$ and $\mathbf{Q}$ are given by:

$$\begin{cases} X = Z\frac{(u-c_u)d_u}{f_c} \qquad Y = Z\frac{(v-c_v)d_v}{f_c} \\ Z = d\frac{f_c}{\sqrt{f_c^2+((u-c_u)d_u)^2+((v-c_v)d_v)^2}} \end{cases} \qquad \begin{cases} u = \frac{f_c}{d_u}\frac{X}{Z} + c_u \qquad v = \frac{f_c}{d_v}\frac{Y}{Z} + c_v \\ d = Z\frac{\sqrt{f_c^2+((u-c_u)d_u)^2+((v-c_v)d_v)^2}}{f_c} \end{cases} \tag{14}$$

Finding f is equivalent to find F and c can be obtained from $F(\mathbf{Q})$ by simple transformation:

$$c = f(\mathbf{q}) = F(\mathbf{Q})\frac{f_c}{\sqrt{f_c^2 + ((u-c_u)d_u)^2 + ((v-c_v)d_v)^2}}. \tag{15}$$

References

[1] A. Belhedi, S. Bourgeois, V. Gay-Bellile, P. Sayd, A. Bartoli, and K. Hamrouni. Non-parametric depth calibration of a tof camera. In *ICIP*, 2012.

[2] S. Á. Guömundsson, H. Aanæs, and R. Larsen. Environmental effects on measurement uncertainties of Time-of-Fight cameras. In *ISSCS*, 2007.

[3] T. Kahlmann, F. Remondino, and H. Ingensand. Calibration for increased accuracy of the range imaging camera SwissRanger $^{\text{Tm}}$. In *IEVM*, 2006.

[4] W. Karel, P. Dorninger, and N. Pfeifer. In situ determination of range camera quality parameters by segmentation. In *Opt. 3D Meas. Tech.*, 2007.

[5] R. Lange. *3D Time-of-Flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology*. PhD thesis, University of Siegen, Germany, 2000.

[6] M. Lindner and A. Kolb. Lateral and depth calibration of PMD-distance sensors. In *ISVC*, 2006.

[7] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD TOF-camera. In *IRCV*, 2007.

[8] S. May, D. Droeschel, and S. Fuchs. D. Holz. Three-dimensional mapping with time-of-flight cameras. *J. Field Robot.*, 26:934–964, 2009.

[9] PMDTechnologies. Pmd[vision]camcube 3.0. www.pmdtec.com/products-services/pmdvisionr-cameras/pmdvisionr-camcube-20/, 2009.

[10] I. Schiller, C. Beder, and R. Koch. Calibration of a PMD-camera using a planar calibration pattern together with a multi-camera setup. In *ISPRS*, 2008.

[11] C. A. Weyer, K. H. Bae, K. Lim, and D. D. Lichti. Extensive metric performance evaluation of a 3D range camera. In *ISPRS*, 2008.