Joint depth and color camera calibration with distortion correction

Daniel Herrera C., Juho Kannala, and Janne Heikkilä

Abstract—We present an algorithm that simultaneously calibrates two color cameras, a depth camera, and the relative pose between them. The method is designed to have three key features: accurate, practical, and applicable to a wide range of sensors. The method requires only a planar surface to be imaged from various poses. The calibration does not use depth discontinuities in the depth image which makes it flexible and robust to noise. We apply this calibration to a Kinect device and present a new depth distortion model for the depth sensor. We perform experiments that show an improved accuracy with respect to the manufacturer's calibration.

Index Terms—camera calibration, depth camera, camera pair, distortion, Kinect

1 Introduction

Color and depth information provide complementary cues about a scene. Many applications need to capture both simultaneously, like scene reconstruction and image based rendering. This requires at least two sensors as no single sensor is able to capture both. A basic device for scene reconstruction is a depth and color camera pair, which consists of a color camera rigidly attached to a depth sensor (e.g. time-of-flight (ToF) camera, laser range scanner, structured light scanner). The increasingly popular Kinect device is an example of such a camera pair.

In order to reconstruct a scene from the camera pair measurements the system must be calibrated. This includes internal calibration of each camera as well as relative pose calibration between the cameras. Color camera calibration has been studied extensively [2], [3]. For depth sensors, different calibration methods have been developed depending on the technology used. ToF cameras simultaneously produce an intensity and a depth image from the same viewpoint, which simplifies calibration because color discontinuities can be accurately localized [4]. Most structured light systems can calibrate the projector and camera separately. However, if the internals of the device are not open, we might not have access to the original intensity images. The Kinect device, for example, uses an infrared camera to detect a projected dot pattern. However, it returns a processed image that is not aligned with the original infrared image.

There is a particular need to calibrate the Kinect device because it delivers depth information in *Kinect disparity units* (kdu) whose conversion to metric units changes for each device. Furthermore, independent calibration of the cameras may not yield the optimal system parameters, and a comprehensive calibration of the system as a whole could improve individual calibration as it uses all the available information.

Depth cameras have been observed to suffer from complicated geometric distortions due to the processing performed and the inevitable tolerances in their manufacturing. Whereas a radial and tangential distortion model is usually sufficient to correct the 2D pixel positions in color cameras, depth cameras require a more complicated model to correct the 3D measurement volume.

1.1 Previous work

A standard approach is to calibrate the cameras independently and then calibrate only the relative pose between them [5], [6], [7]. This may not be the optimal solution as measurements from one camera can improve the calibration of the other camera. Moreover, the independent calibration of a depth camera may require a high precision 3D calibration object that can be avoided using joint calibration.

Fuchs and Hirzinger [4] propose a multi-spline model for time-of-flight (ToF) cameras. Their model has a very high number of parameters and it requires a robotic arm to know the exact pose of the camera. Lindner and Kolb [8] use a high resolution color camera to determine the pose of the camera, removing the need for a robotic arm. Lichti [9] proposes a calibration method for an individual laser range scanner using only a planar calibration object. It performs a comprehensive calibration of all parameters. However, it relies on the depth camera delivering radiometric intensity and range for each pixel. This is not directly applicable to a camera pair because the color and depth information are not taken from the same reference frame. Zhu et al. [10] describe a method for fusing depth from stereo and ToF cameras. Their calibration uses the triangulation from the stereo cameras as ground truth. This may not be optimal as it ignores the possible errors in stereo triangulation and measurement uncertainties.

Authors are with the Center for Machine Vision Research, University of Oulu, Finland.
E-mail: {dherrera, jkannala, jth}@ee.oulu.fi

ToF cameras are known to present distortions both on the optical ray direction and on the measured depth. Kim et al. [11] show a principled approach to correcting these distortions. Cui et al. [12] show that the depth distortion of ToF cameras is radially symmetric and scene dependant. Thus they estimate new distortion correction parameters for each image. The Kinect device has also shown radially symmetric distortions [13]. However, being a structured light sensor, the nature of the distortions is different.

Kinect devices are calibrated during manufacturing with a proprietary algorithm. The calibrated parameters are stored in the device's internal memory and are used by the official drivers to perform the reconstruction. This is adequate for casual use, but we have observed that the manufacturer's calibration does not correct the depth distortion. Other calibration algorithms have been developed by the Kinect community. The first algorithms (e.g. [14]) calibrated only the intrinsics (focal length and principal point) of the infrared camera but did not calibrate the parameters to convert *kinect disparity units* to meters. In our previous work [15], we make a comprehensive calibration of all parameters of the camera pair. However, depth distortion was not corrected. Using a similar formulation, Zhang and Zhang [16] augment our previous work with correspondences between the color and depth images, but still do not address distortion of the depth values. Smíšek et al. [13] include a depth distortion correction component as the average of the residuals in metric coordinates. We propose a disparity distortion correction that depends on the observed disparity which further improves accuracy.

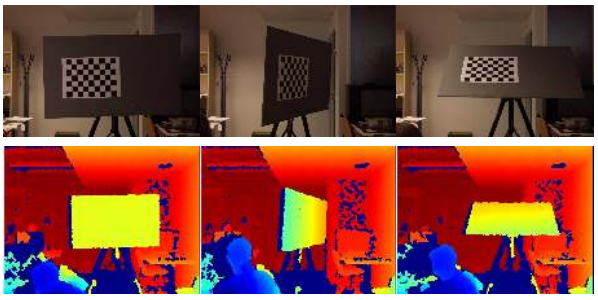


Fig. 1. **Top:** Sample calibration images from the external camera. **Bottom:** Disparity images. Note the inaccuracies at the edges and that the checkerboard is not visible.

1.2 Motivation

As a motivation for our work, we propose three requirements that an optimal calibration algorithm must have. To the best of our knowledge, no available calibration algorithm for a depth and color camera pair fulfills all three criteria.

Accurate: The method should provide the best combination of intrinsic and extrinsic parameters that minimizes the reprojection error for both cameras over all calibration images. This may seem like an obvious principle but we stress it because partial calibrations, where each camera is calibrated independently and the relative pose is estimated separately, may not achieve the lowest reprojection error.

Practical: It should be practical to use with readily available materials. A high precision 3D calibration object is not easy/cheap to obtain and a robotic arm or a high precision mechanical setup to record the exact pose of the camera pair is usually not practical, whereas a planar surface is usually readily available.

Widely applicable: To be applicable to a wide range of depth sensors, one cannot assume that color discontinuities are visible on the depth image. Moreover, some depth sensors, like the one used for our experiments, may not provide accurate measurements at sharp depth discontinuities. Thus, neither color nor depth discontinuities are suitable features for depth camera calibration. The method should use features based on depth measurements that are most reliable for a wide range of cameras (e.g. planes).

Finally, the increasing popularity of the Kinect devices provides an additional motivation for our research. For example, the work from Shotton et al. [17] based on the Kinect was selected as the best paper in CVPR 2011. We believe that many applications would benefit from improved accuracy. We have previously released a Kinect calibration toolbox [15] that has been well received by the developer community. With this work we aim to provide an improved calibration algorithm for the Kinect community.

1.3 Overview of the approach

We use a planar checkerboard pattern for calibration which can be constructed from any readily available planar surface (e.g. a flat table, a wall). We use multiple views of the calibration plane and for each view all cameras take an image. The checkerboard corners provide suitable constraints for the color images, while the planarity of the points provides constraints on the depth images. The pixels at the borders of the calibration object can be ignored and thus depth discontinuities are not needed. Figure 1 shows sample images from the external and depth cameras used for calibration. The three orientations shown constrain the three dimensions of the relative translation between depth and color cameras.

In the color images, the checkerboard corners are extracted. In the depth images, the area containing the plane is located. To initialize the depth camera intrinsics, the user also selects the corners of the calibration plane in the depth images. A standard calibration based on user selected corners [3] is performed on each image individually to initialize the calibration. An iterative non-linear bundle adjustment is then

performed so that our formulation allows for a closed-form solution of the disparity distortion parameters.

2 CALIBRATION MODEL

Our setup consists of a depth and color camera pair with an optional high resolution color camera rigidly attached. Although the camera pair is sufficient for calibration, in the case of a Kinect device, the internal color camera has low quality. Therefore, if one needs high quality color images with depth maps, an external camera is very useful. Our system calibrates both color cameras simultaneously. We will refer to the high resolution camera as the external camera, while the color camera from the camera pair will be referred to simply as the color camera. Our implementation and experiments use the Kinect sensor, which consists of a projector-camera pair as the depth sensor that measures per pixel disparity. The external camera is a Canon EOS 5D Mark II.

2.1 Color camera intrinsics

We use a similar intrinsic model as Heikkilä [2] which consists of a pinhole model with radial and tangential distortion correction. The projection of a point from color camera coordinates $\mathbf{x}_c = [x_c, y_c, z_c]^\top$ to color image coordinates $\mathbf{p}_c = [u_c, v_c]^\top$ is obtained through the following equations. The point is first normalized by $\mathbf{x}_n = [x_n, y_n]^\top = [x_c/z_c, y_c/z_c]^\top$. Distortion is then performed:

$$\mathbf{x}_g = \begin{bmatrix} 2k_3 x_n y_n + k_4(r^2 + 2x_n^2) \\ k_3(r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix} \quad (1)$$

$$\mathbf{x}_k = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6)\mathbf{x}_n + \mathbf{x}_g \quad (2)$$

where $r^2 = x_n^2 + y_n^2$ and $\boldsymbol{k}_c = [k_1, \ldots, k_5]$ is a vector containing the distortion coefficients.

Finally, the image coordinates are obtained:

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} f_{cx} & 0 \\ 0 & f_{cy} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} u_{0c} \\ v_{0c} \end{bmatrix} \quad (3)$$

where $\boldsymbol{f}_c = [f_{cx}, f_{cy}]$ are the focal lengths and $\boldsymbol{p}_{0c} = [u_{0c}, v_{0c}]$ is the principal point. The same model applies to the color and external cameras. The model for each camera is described by $\mathcal{L}_c = \{\boldsymbol{f}_c, \boldsymbol{p}_{0c}, \boldsymbol{k}_c\}$.

2.2 Depth camera intrinsics

In our experiments we used the Kinect as a depth camera. Yet, the method allows different kinds of depth sensors by replacing this intrinsic model. The Kinect's depth sensor consists of an infrared projector that emits a constant pattern and an infrared camera that measures the disparity between the observed pattern and a pre-recorded image at a known constant depth. The output consists of an image of scaled disparity values in Kinect disparity units.

The transformation between depth camera coordinates $\mathbf{x}_d = [x_d, y_d, z_d]^\top$ and depth image coordinate $\mathbf{p}_d = [u_d, v_d]^\top$ follows a similar model to that used for the color camera. Eq. (3) is used with the respective parameters $\boldsymbol{f}_d$ and $\boldsymbol{p}_{0d}$. However, whereas the color camera's distortion is defined in terms of the forward model (world to image), we define the geometric distortion of the depth camera in terms of the backward model (image to world). This is computationally convenient in our case because our formulation of the bundle-adjustment in Section 3.3 backward projects optical rays for the depth camera but forward projects optical rays for the color camera. Further, previous studies have shown that the lens distortion model defined by Eqs. (1) and (2) works well in both ways [2], [18]. Thus, the geometric distortion model for the depth camera is obtained by simply switching the role of $\boldsymbol{x}_n$ and $\boldsymbol{x}_k$ in Eqs. (1) and (2).

The relation between the obtained disparity value d and the depth z_d contains two parts: a scaled inverse and a distortion correction. The scaled inverse has been observed by most previous calibration approaches to fit the observations, it is modeled by the equation:

$$z_d = \frac{1}{c_1 d_k + c_0} \quad (4)$$

where c_1 and c_0 are part of the depth camera intrinsic parameters to be calibrated and d_k is the undistorted disparity (i.e. after distortion correction). Note that for all Kinect devices c_1 is negative, which means that higher disparity values correspond to points farther away from the camera (the opposite of traditional stereo disparity units).

When calibrated using only Equation (4) (i.e. without distortion correction) the Kinect displays a fixed error pattern in the measurements (Figure 2). Because the internal algorithms of the Kinect are proprietary and closed, it is not possible to pinpoint the exact nature of this distortion. However, we can correct it based on observations. It was suggested in [13] that this distortion could be corrected by applying a spatially varying offset $\boldsymbol{Z}_\delta$ to the calculated depth:

$$z_{dd} = z_d + \boldsymbol{Z}_\delta(u, v) \quad (5)$$

and it was observed that this usually reduces the reprojection error. However, we have found that a more accurate calibration is made by correcting for the distortion directly in disparity units.

The shape of the error pattern is constant but its magnitude decreases as the distance from the object increases. To demonstrate this decay we took the errors from planes at several distances and normalized them (dividing all images by Figure 2). Figure 3 shows the resulting median values for each measured disparity. The normalized error fits well to an exponential decay. This led us to construct a distortion model that has per-pixel coefficients and decays exponentially with increasing disparity.

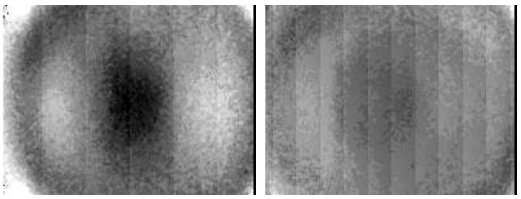


Fig. 2. Error residuals (kdu) without distortion correction of a plane at 0.56m (left) and 1.24m (right).

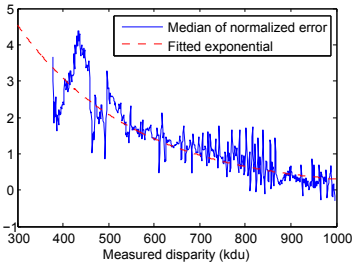


Fig. 3. Distortion magnitude with increasing disparity.

We use a spatially varying offset that decays as the Kinect disparity increases:

$$d_k = d + \boldsymbol{D_\delta}(u,v) \cdot \exp(\alpha_0 - \alpha_1 d) \tag{6}$$

where d is the distorted disparity as returned by the Kinect, $\boldsymbol{D_\delta}$ contains the spatial distortion pattern, and $\boldsymbol{\alpha} = [\alpha_0, \alpha_1]$ models the decay of the distortion effect.

Note that this models does not enforce any smoothness on the spatial distortion pattern. To properly constrain this pattern it is enough to include some (four) images of a flat surface that spans the entire depth image. We add images of an empty wall at several depths. These images do not need the checkerboard pattern since they are only needed to constrain the distortion pattern. This ensures that all pixels in the depth image have samples to estimate their coefficients $D_\delta(u,v)$.

Although this disparity distortion model was developed with the Kinect in mind, it bears similarities with the model of a ToF camera. Kim et al. [11] obtained results similar to Figure 3, except that they fit a 6^{th} degree polynomial instead of an exponential. Furthermore, the calibration of this ToF camera model is simpler because they don't use per-pixel coefficients.

Equations (4) and (6) are used when measured disparities are transformed to metric coordinates, also known as the backward model. The inverse of these functions, the forward model, is also needed to compute the reprojection errors. The inverse of Equation (4) is straightforward:

$$d_k = \frac{1}{c_1 z_d} - \frac{c_0}{c_1} \tag{7}$$

But the inverse of Equation (6) is a bit more involved because of the exponential. We perform two variable substitutions to isolate the exponential product:

$$\begin{aligned}
y &= \exp(\alpha_0 - \alpha_1 d_k + \alpha_1 \boldsymbol{D_\delta}(u,v) y) \\
&\text{where } y = \frac{d_k - d}{\boldsymbol{D_\delta}(u,v)} \\
y &= \exp(\alpha_1 \boldsymbol{D_\delta}(u,v) y) \exp(\alpha_0 - \alpha_1 d_k) \\
\frac{-\tilde{y}}{\alpha_1 \boldsymbol{D_\delta}(u,v)} &= \exp(-\tilde{y}) \exp(\alpha_0 - \alpha_1 d_k) \\
&\text{where } \tilde{y} = -y \alpha_1 \boldsymbol{D_\delta}(u,v) \\
\tilde{y} \exp(\tilde{y}) &= -\alpha_1 \boldsymbol{D_\delta}(u,v) \exp(\alpha_0 - \alpha_1 d_k)
\end{aligned}$$

The product can be solved using the Lambert W function [19]. The Lambert W function is the solution to the relation $W(z)\exp(W(z)) = z$.

$$\begin{aligned}
\tilde{y} &= W(-\alpha_1 \boldsymbol{D_\delta}(u,v) \exp(\alpha_0 - \alpha_1 d_k)) \\
(d - d_k)\alpha_1 &= W(-\alpha_1 \boldsymbol{D_\delta}(u,v) \exp(\alpha_0 - \alpha_1 d_k)) \\
d &= d_k + \frac{W(-\alpha_1 \boldsymbol{D_\delta}(u,v) \exp(\alpha_0 - \alpha_1 d_k))}{\alpha_1}
\end{aligned} \tag{8}$$

Although the Lambert W function is a trascendental function, there are many accurate approximations in the literature [19] and modern mathematical packages include implementations of it (e.g. Matlab).

The model for the depth camera is described by $\mathcal{L}_d = \{\boldsymbol{f}_d, \boldsymbol{p}_{0d}, \boldsymbol{k}_d, c_0, c_1, \boldsymbol{D_\delta}, \boldsymbol{\alpha}\}$, where the first 3 parameters come from the model described in section 2.1 and the last 4 are used to transform disparity to depth values.

2.3 Extrinsics and relative pose

Figure 4 shows the different reference frames present in a scene. Points from one reference frame can be transformed to another using a rigid transformation denoted by $\mathcal{T} = \{\boldsymbol{R}, \boldsymbol{t}\}$, where $\boldsymbol{R}$ is a rotation and $\boldsymbol{t}$ a translation. For example, the transformation of a point $\mathbf{x}_w$ from world coordinates $\{W\}$ to color camera coordinates $\{C\}$ follows $\mathbf{x}_c = {}^W\boldsymbol{R}_C \mathbf{x}_w + {}^W\boldsymbol{t}_C$.

Reference $\{V_i\}$ is anchored to the corner of the calibration plane of image i and is only used for initialization. The relative poses (${}^D\mathcal{T}_C$ and ${}^E\mathcal{T}_C$) are constant, while each image has its own world to camera pose ${}^{W_i}\mathcal{T}_C$. By design, the table and the checkerboard are coplanar but the full transformation between $\{V\}$ and $\{W\}$ is unknown.

3 Calibration Method

A block diagram of our calibration method is presented in Figure 5. The individual steps are described in the following sections.

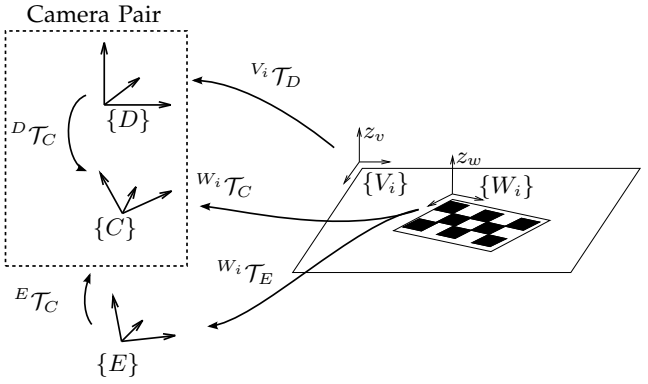


Fig. 4. Reference frames and transformations. $\{D\}$, $\{C\}$, and $\{E\}$ are the depth, color, and external cameras. For image i, $\{V_i\}$ is attached to the calibration plane and $\{W_i\}$ is the calibration pattern.

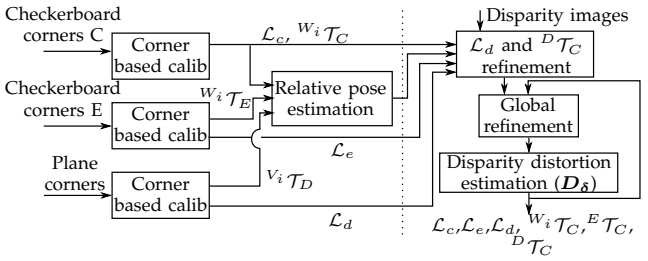


Fig. 5. Calibration algorithm. **Before dashed line:** initialization. **After dashed line:** non-linear minimization.

3.1 Corner based calibration

The calibration of a color camera is a well studied problem, we use Zhang's method [3] to initialize the camera parameters. Briefly, the steps are the following. The checkerboard corners are extracted from the intensity image. A homography is then computed for each image using the known corner positions in world coordinates $\{W_i\}$ and the measured positions in the image. Each homography then imposes constraints on the camera parameters which are solved with a linear system of equations. The distortion coefficients are initially set to zero.

The same method is used to initialize the depth camera parameters. However, because the checkerboard is not visible in the depth image, the four corners of the calibration plane are extracted (the gray plane in Figure 1). These corners are very noisy and are only used here to obtain an initial guess. The homography is thus computed between $\{V_i\}$ and $\{D\}$ also using Zhang's method. This initializes the focal lengths, principal point, and the transformation $^{V_i}\mathcal{T}_D$. Using these initial parameters we obtain an estimate for the expected depth of each selected corner. With this expected depth and the measured disparity an overdetermined system of linear equations is built using (4), which gives an initial guess for the depth parameters (c_0 and c_1).

3.2 Relative pose estimation

The relative pose between the external and color cameras can be obtained directly because their pose with respect to the same reference frame $\{W\}$ is known. For the depth camera, however, only the pose with respect to $\{V\}$ is known, which is not aligned to $\{W\}$. To obtain the relative pose $^{C}\mathcal{T}_D$ we take advantage of the fact that $\{V\}$ and $\{W\}$ are coplanar by design. We extract the plane equation in each reference frame and use it as a constraint. We define a plane using the equation $\boldsymbol{n}^\top \mathbf{x} - \delta = 0$ where $\boldsymbol{n}$ is the unit normal and δ is the distance to the origin.

If we divide a rotation matrix into its colums $\boldsymbol{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ and choose the parameters of the plane in both frames as $\boldsymbol{n} = [0, 0, 1]^\top$ and $\delta = 0$, the plane parameters in camera coordinates are:

$$\boldsymbol{n} = \mathbf{r}_3 \quad \text{and} \quad \delta = \mathbf{r}_3^\top \boldsymbol{t} \tag{9}$$

where we use $^{W_i}\boldsymbol{R}_C$ and $^{W_i}\boldsymbol{t}_C$ for the color camera and $^{V_i}\boldsymbol{R}_D$ and $^{V_i}\boldsymbol{t}_D$ for the depth camera.

As mentioned by Unnikrishnan and Hebert [6] the relative pose can be obtained in closed form from several images. The plane parameters for each image are concatenated in matrices of the form: $\boldsymbol{M}_c = [\boldsymbol{n}_{c1}, \boldsymbol{n}_{c2}, ..., \boldsymbol{n}_{cn}]$, $\boldsymbol{b}_c = [\delta_{c1}, \delta_{c2}, ..., \delta_{cn}]$, and likewise for the depth camera to form $\boldsymbol{M}_d$ and $\boldsymbol{b}_d$. The relative transformation is then:

$$^{C}\boldsymbol{R}'_D = \boldsymbol{M}_d \boldsymbol{M}_c^\top \tag{10}$$

$$^{C}\boldsymbol{t}_D = (\boldsymbol{M}_c \boldsymbol{M}_c^\top)^{-1} \boldsymbol{M}_c (\boldsymbol{b}_c - \boldsymbol{b}_d)^\top \tag{11}$$

Due to noise $^{C}\boldsymbol{R}'_D$ may not be orthonormal. We obtain a valid rotation matrix through SVD using: $^{C}\boldsymbol{R}_D = UV^\top$ where $USV^\top$ is the SVD of $^{C}\boldsymbol{R}'_D$.

3.3 Non-linear minimization

The calibration method aims to minimize the weighted sum of squares of the measurement reprojection errors over all parameters ($\mathcal{L}_c$,$\mathcal{L}_d$,$\mathcal{L}_e$,$^{E}\mathcal{T}_C$,$^{D}\mathcal{T}_C$, and ,$^{W_i}\mathcal{T}_C$ for all images i). The error for the color camera is the Euclidean distance between the measured corner position $\hat{\boldsymbol{p}}_c$ and its reprojected position $\boldsymbol{p}_c$ (the same for the external camera with $\hat{\boldsymbol{p}}_e$ and $\boldsymbol{p}_e$ respectively). Whereas for the depth camera it is the difference between the measured disparity $\hat{d}$ and the predicted disparity d. The predicted disparity is obtained by calculating the distance along the optical ray of the calibration plane and transforming to disparity using Eqs. (7) and (8). Because the errors have different units, they are weighted using the inverse of the corresponding measurement variance (σ_c^2, σ_d^2, σ_e^2). The resulting cost function is:

$$c = \frac{\sum \|\hat{\boldsymbol{p}}_c - \boldsymbol{p}_c\|^2}{\sigma_c^2} + \frac{\sum (\hat{d} - d)^2}{\sigma_d^2} + \frac{\sum \|\hat{\boldsymbol{p}}_e - \boldsymbol{p}_e\|^2}{\sigma_e^2} \tag{12}$$

Note that (12) is highly non-linear and depends on a lot of parameters ($\boldsymbol{D_\delta}$ contains 307200 entries). To

separate the optimization of the disparity distortion parameters from the rest we make a slight modification to Equation (12). Instead of comparing the reprojection in measured disparity space, we calculate the residuals in undistorted disparity space:

$$c = \frac{\sum \|\hat{\boldsymbol{p}}_c - \boldsymbol{p}_c\|^2}{\sigma_c^2} + \frac{\sum (\hat{d}_k - d_k)^2}{\sigma_d^2} + \frac{\sum \|\hat{\boldsymbol{p}}_e - \boldsymbol{p}_e\|^2}{\sigma_e^2} \tag{13}$$

It is also possible to optimize Eq. (12) by inverting the roles of Eqs. (6) and (8). However, including the Lambert W function in the backward camera model would make it cumbersome to use for transforming measurements into 3D points. We tested both approaches and found no practical advantage of minimizing Eq. (12) over (13).

The optimization has three steps as shown in Fig. 5. The initialization gives a very rough guess of the depth camera parameters and relative pose, whereas the color camera parameters have fairly good initial values. Thus, the first step optimizes only $\mathcal{L}_d$ and ${}^D\mathcal{T}_C$ with all other parameters fixed. The optimization then continues iteratively with two alternating steps. In the first step $\boldsymbol{D_\delta}$ is kept constant and Equation (13) is minimized using the Levenberg-Marquardt algorithm over all other parameters. In the second step the spatial disparity distortion pattern $\boldsymbol{D_\delta}$ is optimized independently for each pixel. The initial values of the depth distortion model ($\boldsymbol{\alpha}$ and $\boldsymbol{D_\delta}$) are not critical and initially assuming zero for both has proven to yield accurate results. The algorithm iterates until the residuals are stable.

3.4 Disparity distortion estimation

Optimizing $\boldsymbol{D_\delta}$ separately is more efficient because the entries in $\boldsymbol{D_\delta}$ are independent from each other and the estimation of $\boldsymbol{D_\delta}(u,v)$ takes into account only measurements obtained from pixel (u,v). Moreover, when the other parameters are fixed we can solve for $\boldsymbol{D_\delta}(u,v)$ in closed-form.

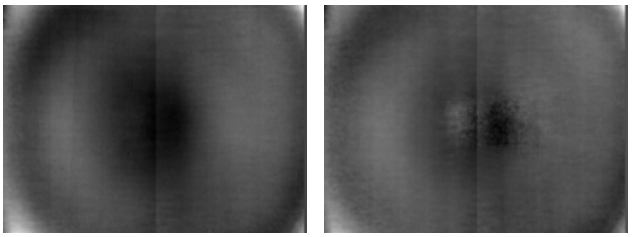
Each disparity measurement $\hat{d}$ is first undistorted using Equation (6). We compute a predicted disparity d_k using the distance to the plane and Equation (7). We minimize the following cost function to obtain the distortion parameters:

$$c_d = \sum_{\text{images}} \sum_{u,v} (\hat{d} + \boldsymbol{D_\delta}(u,v) \cdot \exp(\alpha_0 - \alpha_1 \hat{d}) - d_k)^2 \tag{14}$$

This minimization is straightforward because Equation (14) is quadratic in each $\boldsymbol{D_\delta}(u,v)$ and hence the optimal value of each $\boldsymbol{D_\delta}(u,v)$ is obtained by solving a linear equation.

For comparison we also calibrated using the model of Smíšek et al. [13]. The value of $\boldsymbol{Z}_\delta(u,v)$ is calculated as the mean difference between measured depth $\hat{z}_d$ and expected depth z_d:

$$\boldsymbol{Z}_\delta(u,v) = \frac{\sum_N z_d - \hat{z}_d}{N} \tag{15}$$



(a) Our method: $\boldsymbol{D_\delta}$ (b) Smíšek et al.: $\boldsymbol{Z}_\delta$

Fig. 6. Obtained distortion spatial patterns.

TABLE 1
Calibration with different distortion models. Std. deviation of residuals with a 99% confidence interval.

		Color ±0.02 px	**Depth** ±0.002 kdu	**External** ±0.05 px
A1	No correction	0.42	1.497	0.83
	Smíšek [13]	0.32	1.140	0.72
	Our method	0.28	0.773	0.64
A2	No correction	0.36	1.322	0.83
	Smíšek [13]	0.33	0.884	0.85
	Our method	0.38	0.865	0.79
B1	No correction	0.56	1.108	0.97
	Smíšek [13]	0.62	1.300	0.91
	Our method	0.57	0.904	0.85

4 Results

The color camera of the Kinect delivers images with a resolution of 1280x1024, whereas the resolution of the external camera was 2784x1856. Three different data sets were captured (A1, A2, and B1). Two of them were captured with the same Kinect (A1 and A2) and one with a different Kinect (B1). For each set, the captured images were divided into calibration and validation groups with 60 and 14 images respectively. The calibration images were used to estimate all the parameters in the model, then the intrinsic parameters were kept fixed to estimate only the rig's pose (${}^W\mathcal{T}_C$) for the validation images. All results presented here were obtained from the validation images.

We implemented our algorithm in Matlab. The code has been released as a toolbox for the research community. It can be found in the same website as our previous work [15]. We used 60 plane orientations for calibration. However, we found that comparable accuracy is achieved with only 20 positions: 4 for each orientation shown in Fig. 1 at different depths, and 4 of a flat surface that covers the entire depth image. The calibration with 60 positions takes 15min on a 2.4GHz computer, but only 3min with 20 positions.

Normally, the different calibrations (A1, A2, and B1) would produce slightly different error variances (σ_c,σ_d, and σ_e). To compare the data sets the variances were kept constant ($\sigma_c = 0.18$px, $\sigma_d = 0.9$kdu, and $\sigma_e = 0.30$px).

TABLE 2
Calibration without the external camera. Std. deviation of residuals with a 99% confidence interval.

	Color ± 0.02 px	Depth ± 0.002 kdu
A1	0.26	0.765
A2	0.36	0.873
B1	0.60	0.902

4.1 Calibration accuracy

Figure 6 shows the obtained spatial patterns for the distortion correction using Eqs. (6) and (5). We can observe a very similar pattern in both images. Table 1 shows a comparison of the calibration results obtained using both types of distortion correction and no correction. The three models were calibrated using the same data sets and the table shows the results of validation against the respective validation set.

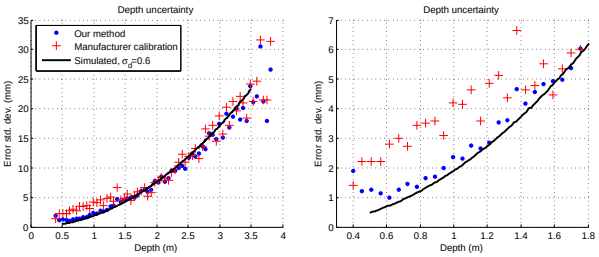
The distortion correction proposed by Smíšek et al. [13] improves the reprojection error for data sets A1 and A2. However, because it reduces the error in metric space it increases the reprojection error for B1. In contrast, our approach produces the best results for all cases. The standard deviation of the reprojection errors for all sets were both very low (< 1px and < 1kdu), which demonstrates an accurate calibration. Also note that even though no spatial smoothness was enforced for the spatial distortion pattern, the obtained pattern is smooth, proving that the procedure provides enough constraints.

Table 2 shows the results of calibration without the external camera. The calibration accuracy remains the same. The external camera is thus not necessary for an accurate calibration. Still, its joint calibration is a useful feature for many applications that need a high quality external camera.

4.2 Comparison with manufacturer calibration

The drivers provided by the manufacturer (Primesense) use factory calibrated settings to convert the disparity measurements to 3D points. We used these calibration parameters and compared their performance to that of our calibration. Using the disparity images from the A2 data set, both calibrations were used to obtain 3D points. The calibration of our method was done with the A1 data set to avoid any bias. A plane was fitted to each cloud of points and the measured depth was compared to the expected based on the plane's depth at the given pixel. The error measurements are shown in Figure 7 for both calibrations. The measurements were grouped by depth in 64 bins from 0.4m to 3.7m. For each bin, the standard deviation of the error was plotted.

The uncertainty was also simulated using the calibrated model. For a given depth, the expected disparity for each pixel was calculated using Equations (7) and (8). Gaussian noise ($\mu = 0$ and $\sigma = 0.6$) was applied to this disparity and the corrupted depth is obtained through Equations (4) and (6). The standard deviation of the error between the initial and corrupted depths is plotted as a solid line in Figure 7. We can see that these virtual results are very close to our experimental results.



(a) Full calibrated range (b) Zoom in

Fig. 7. Measurement uncertainty for varying depths.

Like any stereo camera, the Kinect is expected to be more accurate the closer the object is. Due to the inverse relation between disparity and depth, zero mean noise with constant variance on the measured disparity will result in higher depth uncertainty as the depth increases. This relation is shown in Figure 7. Our method clearly outperforms the manufacturer calibration in ranges up to 1.5m. At 1m distance the manufacturer calibration has twice the uncertainty. After 1.5m the distortion correction has a smaller influence in the reconstruction and both calibrations have similar accuracies.

It is suspected that the infrared image is not distortion corrected before the depth estimation algorithm is applied, which produces the depth distortion pattern. This is why the disparity distortion has the same spatial distribution as a radial distortion (i.e. concentric circles). It is unclear why the depth distortion decays with depth. The depth estimation algorithm locates known point patterns in the infrared image. At far distances, the point patterns might be closer to the factory memorized position and the distortion in the infrared image could have less impact.

4.3 Variability of Kinect devices

To justify the need for calibrating the Kinect we used the calibration of one data set on the validation images of another data set. The external camera was not used for the validation here because its relative pose is different between the different data sets. The results are presented in Table 3. They show that the reconstruction using the calibration of another Kinect is highly inaccurate and increases the reprojection error considerably, both for color and depth camera. Thus supporting the idea that each Kinect must be individually calibrated to achieve maximum accuracy.

TABLE 3
Variability of Kinects. Std. dev. of residuals using different sets. Dark cells indicate a device mismatch.

	CalibA1	CalibA2	CalibB1
ValidA1	0.26px, 0.77kdu	0.29px, 0.83kdu	1.35px, 1.55kdu
ValidA2	0.45px, 0.76kdu	0.36px, 0.86kdu	1.77px, 1.80kdu
ValidB1	1.81px, 1.52kdu	1.72px, 1.49kdu	0.56px, 0.89kdu

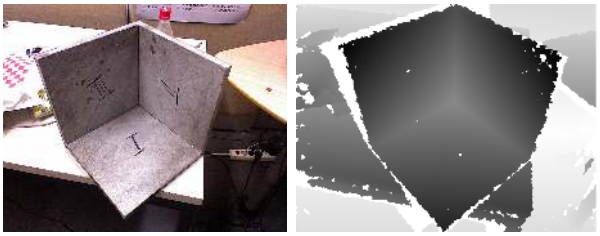


Fig. 8. 3D reference cube. Color and disparity images.

4.4 3D ground truth

We also compared the accuracy of calibration by reconstructing a hollow cube whose sides are known to be at $90°$ from each other. Figure 8 shows the reference cube. A point cloud was obtained from the disparity image and planes were fitted to the points from each side. Table 4 shows how much the angle between the obtained planes deviates from $90°$. Our method clearly achieves a better reconstruction accuracy.

5 Conclusion

We have presented a calibration algorithm for a depth and color camera pair that is optimal in the sense of the postulated principles. The algorithm takes into account color and depth features simultaneously to improve calibration of the camera pair system as a whole. It requires only a planar surface and a simple checkerboard pattern.

The results show that our algorithm achieved a better calibration for the Kinect than that provided by the manufacturer. The disparity distortion correction model considerably improved reconstruction accuracy, better than previously proposed models. At 1m distance our calibration showed twice the reconstruction accuracy than the manufacturer's calibration. Moreover, we have released our code as a Matlab toolbox to the research community.

The extension of the calibration to several external color cameras is straighforward and is already implemented in the released toolbox. In addition, we believe that our algorithm is flexible enough to be used with other types of depth sensors by replacing the intrinsics model of the depth camera. The constraints used can be applied to any type of depth sensor. Future work can include the calibration of a ToF and color camera pair.

TABLE 4
Angular error between reconstructed planes. $\angle_{ab}$ is the angle between planes a and b.

	Manufacturer	Smíšek [13]	Our method
$90° - \angle_{12}$	-1.4	1.2	0.6
$90° - \angle_{13}$	-1.1	1.2	-0.2
$90° - \angle_{23}$	1.0	-1.0	0.1

Acknowledgements

This project has been funded by the Academy of Finland's project #127702.

References

[1] S. G. Latta, Gesture keyboarding, uS 2010/0199228 A1 (Aug. 2010).
[2] J. Heikkilä, Geometric camera calibration using circular control points. IEEE Transcations on Pattern Analysis and Machine Intelligence 22(10), 1066-1077.
[3] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, in: ICCV, 1999, pp. 666–673.
[4] S. Fuchs, G. Hirzinger, Extrinsic and depth calibration of ToF-cameras, in: CVPR, 2008, pp. 1–6.
[5] Q. Zhang, R. Pless, Extrinsic calibration of a camera and laser range finder (improves camera calibration), in: IROS, Vol. 3, 2004, pp. 2301–2306.
[6] R. Unnikrishnan, M. Hebert, Fast extrinsic calibration of a laser rangefinder to a camera, Tech. Rep. CMU-RI-TR-05-09, Robotics Institute, Pittsburgh (2005).
[7] D. Scaramuzza, A. Harati, R. Siegwart, Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes, in: IROS, 2007, pp. 4164–4169.
[8] M. Lindner, A. Kolb, Calibration of the intensity-related distance error of the PMD TOF-Camera, in: SPIE: Intelligent Robots and Computer Vision XXV, Vol. 6764, 2007.
[9] D. Lichti, Self-calibration of a 3D range camera, ISPRS 37 (3).
[10] J. Zhu, L. Wang, R. Yang, J. Davis, Fusion of time-of-flight depth and stereo for high accuracy depth maps, in: CVPR, 2008, pp. 1–8.
[11] Y. Kim, D. Chan, C. Theobalt, S. Thrun, Design and calibration of a multi-view ToF sensor fusion system, in: IEEE CVPR Workshop on Time-of-flight Computer Vision, 2008.
[12] Y. Cui, S. Schuon, D. Chan, S. Thrun, C. Theobalt, 3d shape scanning with a time-of-flight camera, in: Proc. of IEEE CVPR, 2010, pp. 1173–1180.
[13] J. Smíšek, M. Jančošek, T. Pajdla, 3D with Kinect, in: IEEE Workshop on Consumer Depth Cameras for Computer Vision, 2011.
[14] N. Burrus, Kinect calibration (Nov. 2011).
URL http://nicolas.burrus.name/index.php/Research/KinectCalibration
[15] D. Herrera C., J. Kannala, J. Heikkilä, Accurate and practical calibration of a depth and color camera pair, in: CAIP 2011, Part II, LNCS 6855, 2011, pp. 437–445.
URL http://www.ee.oulu.fi/~dherrera/kinect/
[16] C. Zhang, Z. Zhang, Calibration between depth and color sensors for commodity depth cameras, in: International Workhsop on Hot Topics in 3D, in conjunction with ICME, 2011.
[17] J. Shotton, A. Fitzgibbon, M. Cook, A. Blake, Real-time human pose recognition in parts from single depth images, in: CVPR, IEEE, 2011, pp. 1297–1304.
[18] D. Brown, Close-range camera calibration, Photogrammetric Engineering 37 (8) (1971) 855–866.
[19] D. Barry, J.-Y. Parlange, L. Li, H. Prommer, C. Cunningham, F. Stagnitti, Analytical approximations for real values of the Lambert W-function, Mathematics and Computers in Simulation 53 (1-2) (2000) 95–103.