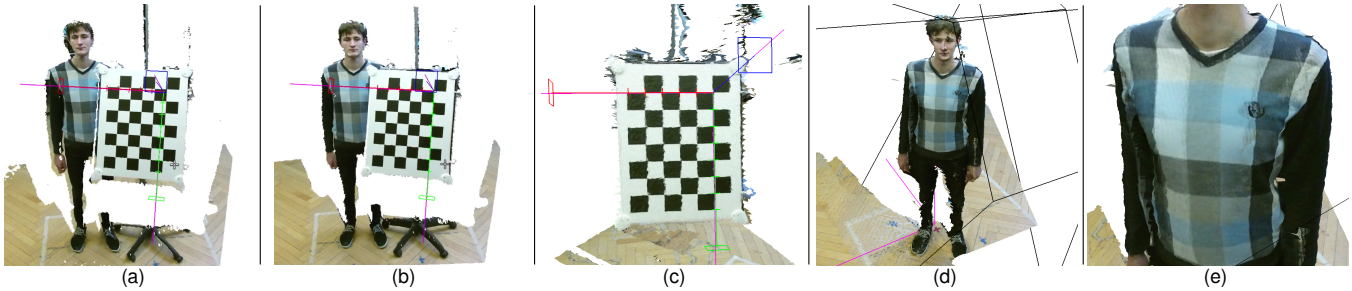
Volumetric Calibration and Registration of Multiple RGBD-Sensors into a Joint Coordinate System

Stephan Beck* Bernd Froehlich†

Virtual Reality and Visualization Research Group at Faculty of Media, Bauhaus-Universität Weimar



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

Figure 1: Calibration and registration results of our proposed calibration method visualized with the real-time 3D reconstruction from [3]: (a) without and (b) with our calibration applied. (c) - (e) Two overlapping Kinect V2 sensors are positioned at an angle of about 90 degrees, about 45 degrees left and right of the user and the checkerboard. The achieved accuracy of our volumetric calibration allows for precise matching of the sensor contributions as well as a precise registration into a joint coordinate system. (a) - (c) The rgb-colored coordinate system is tracked with our calibrated multi-Kinect-V2 setup and precisely coincides ((b) and (c)) with the magenta coordinate system which is tracked with an optical tracking system [1] – our joint coordinate system.

ABSTRACT

We present an integrated approach for the calibration and registration of color and depth (RGBD) sensors into a joint coordinate system. Our application domain is 3D telepresence where users in front of a three-dimensional display need to be captured from all directions. The captured data is used to virtually reconstruct the group of people at a remote location. One key requirement of such applications is that contributions from different color and depth cameras match, as closely as possible, in spatially overlapping or adjacent regions. Our method employs a tracked checkerboard to establish a number of correspondences between positions in color and depth camera space and in world space. These correspondences are used to construct a single calibration and registration volume per RGBD sensor which maps raw depth sensor values in a single step into a joint coordinate system and to their associated color values. This approach considerably reduces reconstruction latency by omitting expensive image rectification processes during runtime. Furthermore, our evaluation demonstrates a high measurement accuracy with an average 3D error below 3 mm and an average texture deviation smaller than 0.5 pixels for a space of about 1.5 m x 1.8 m x 1.5 m.

Keywords: Telepresence, 3D capturing, camera calibration, registration, depth camera, Kinect.

1 INTRODUCTION

3D capturing systems are used in many interactive applications in order to serve as a basis for real-time 3D reconstruction of humans, pose estimation, 3D user interfaces, skeleton tracking, or more general 3D measuring tasks. Our application domain is immersive 3D telepresence, which has been a research topic for more than 20 years [9]. While early work relied on a set of color cameras for capturing users in 3D (e.g. [9, 13]), the recent availability of increasingly better, inexpensive depth and color (RGBD) sensors has revived interest in this topic (e.g. [15, 4, 3]). Fuchs [10] states that the main challenges for 3D telepresence remain in the field of 3D displays and the acquisition and reconstruction of the participants. Basic requirements for the latter tasks are low latency processing and transmission of the depth and color cameras' RGBD values and their accurate mapping into an application's world coordinate system.

We developed a volumetric calibration and registration approach which directly maps raw depth sensor values to 3D positions in world space and to their corresponding texture coordinates of the associated color camera image. A tracked checkerboard is placed at various positions in our capturing volume to establish correspondences between raw depth values of an RGBD sensor and the associated positions in world space. A depth camera's infrared image of the checkerboard is used to also establish correspondences between the raw depth values and the texture coordinates of the associated color camera. These correspondences are entered into a 3D lookup table of a typical size of $128 \times 128 \times 256$. Empty cells are filled by scattered data interpolation. This process is performed once for each RGBD sensor. During runtime, this 3D lookup table can be used on the CPU or GPU to directly map the raw depth values to 3D positions in world space, whereas the color information is retrieved through the looked-up texture coordinates.

In virtual reality systems, optical tracking systems are often employed to track the users' head and hand positions or even more body parts. The tracking system's coordinate system and the world coordinate system of the application are typically linked together by a rigid body transformation. Thus, the tracking system's accuracy –or inaccuracy– is inevitably transferred into the application. As a

*e-mail: stephan.beck@uni-weimar.de
†e-mail: bernd.froehlich@uni-weimar.de

Table 1: Comparison of characteristics of state-of-the-art multi-sensor calibration methods. Intrinsic relates to the identification of intrinsic parameters of the color and depth sensor. Depth calibration relates to an explicit correction of the sensors' depth measurements. Extrinsic relates to the calibration between the involved depth and color sensors. Registration relates to whether multiple RGBD-sensors are externally registered to a (joint) reference coordinate system or in a camera-to-camera fashion (inter-camera). Geometric preservation relates to the ability to preserve shapes, lengths, and angles of the captured scene geometry.

Method	Intrinsic	Depth calibration	Extrinsic	Registration	Geometric preservation
Maimone et al. [15]	optical	none	optical	reference	no
Maimone et al. [16]	optical	none	optical	reference + inter-camera	no
Kainz et al. [12]	optical	none	optical	reference + inter-camera	no
Beck et al. [3]	optical	yes	optical	reference	yes
Deng et al. [8]	optical	none	optical + geometric	inter-camera	no
Avetisyan et al. [2]	optical	yes	optical	reference	yes
Our	simultaneous optical + geometric color and depth calibration to a joint reference				yes

consequence, even a precisely calibrated RGBD-sensor cannot be mapped into the application's coordinate system by a simple rigid body transformation. In fact, absolute accuracy is less important. More important is that the contributions of different RGBD sensors are precisely registered into the tracking space so that they match, as closely as possible, in spatially overlapping or adjacent regions of a captured object. In our approach, this is ensured by using the optical tracking system as a reference for acquiring the correspondences between depth camera space and a virtual world coordinate system for all involved depth sensors. The spatially varying correspondences of the associated color image to the depth image is captured by the 3D lookup table, as well, which leads to smooth transitions along seams where contributions of multiple RGBD sensors are blended or stitched together.

The main properties of our novel approach are

- a low-latency single-step mapping of raw depth sensor values to positions in world space and to texture coordinates in an associated color image,
- no reliance on any specific lens or camera model and
- an accuracy close to the resolution of the sensors throughout the capturing area.

We compare our approach to the state-of-the art calibration method described by Beck et al. [3] which reveals significant improvements in accuracy. Furthermore, we evaluated different scattered data interpolation approaches for constructing the 3D lookup table and recommend the use of natural neighbor interpolation.

2 Related Work

The Microsoft Kinect is one of the most popular RGBD-sensors. It is used in many applications, either to serve as an input device or as a 3D capturing device. While the Kinect's depth and color sensor are integrated into a single device, it is also possible to combine a color camera with a pure depth sensor like the Asus Xtion Pro™, or time-of-flight sensors like the CamCube™. Regardless of which type of sensor is used, its calibration involves the identification of all parameters of an underlying projection model. In particular, the camera's intrinsic parameters, which describe its projection and rectification model, have to be identified. The intrinsic parameters of Zhang's [19] established calibration model consist of the camera's principal point, focal length, and coefficients for radial and tangential distortion. In addition, the depth sensor itself either reports disparity values at each pixel, which then have to be converted by a parameterized function or an explicit mapping to metric distance, or metric values, depending on its underlying technology. For most applications, a rigid body transformation that defines the relation between the color and depth reference frame, has to be identified, too, which is often termed as extrinsic calibration in literature. Raposo et al. [17] give a detailed definition of all involved parameters and the camera's projection model. Although the parameters are factory pre-calibrated, the accuracy is limited and improvements have been investigated [18, 11, 20].

State-of-the-art camera calibration methods are based on capturing a planar checkerboard for several poses and using the detected checkerboard crossing points to find the camera's intrinsic parameters [19]. Herrera et al. [11] presented a depth distortion model for the Kinect sensor and an algorithm that jointly calibrates the color and depth camera. They show that their method yields higher accuracy than separately calibrating the color and depth sensor. Raposo et al. [17] improved the joint calibration method of [11] by replacing the refinement step of the initial calibration guess with a non-linear optimization. Their method further improves accuracy and results in a speed-up of the process since it relies on fewer reference images.

While the aforementioned methods mostly focus on the calibration of a single sensor, various solutions for the specific challenge of calibrating 3D capturing systems have been investigated more recently [15, 16, 12]. Capturing typically requires the use of multiple RGBD sensors. Therefore a large set of matching intrinsic and extrinsic parameters need to be identified.

In 2011 Maimone et al. [15] introduced a telepresence system that uses an array of Kinects for 3D capturing people and a surrounding scene in real time. Their system was not designed for accuracy in terms of 3D reconstruction but rather for perceived visual quality. In order to match the contributions from overlapping sensors, they proposed a quality-based fusion model for a screen space-based merging process which incorporates the depth measurement error that increases with distance. As a result, their method ensures that the depth camera contributions with the highest available accuracy are blended and displayed. In their evaluation, they measured a 3D positional error of around 1.8 cm at a distance 0.7 m and about 3 cm at a distance of 1.8 m from the cameras, which was mainly caused by the depth sensors' inaccurate distance measurements.

In order to improve the 3D matching of several sensors throughout a larger capturing space, Maimone et al. [16] proposed an inter-camera-based calibration method. They placed a calibration target at several positions inside the capturing space to obtain a set of 3D correspondences which were then used to fit an affine transform for each sensor that minimized the distances between the measured correspondences. As a result, they were able to reduce the 3D positional error from about 3 cm to about 1cm. In 2012 Kainz et al. [12] presented a similar approach to registering a setup consisting of multiple Kinects. Like [16] they simultaneously captured a calibration sphere from all involved depth cameras to obtain a set of 3D correspondences. For each Kinect, they fit a three-dimensional

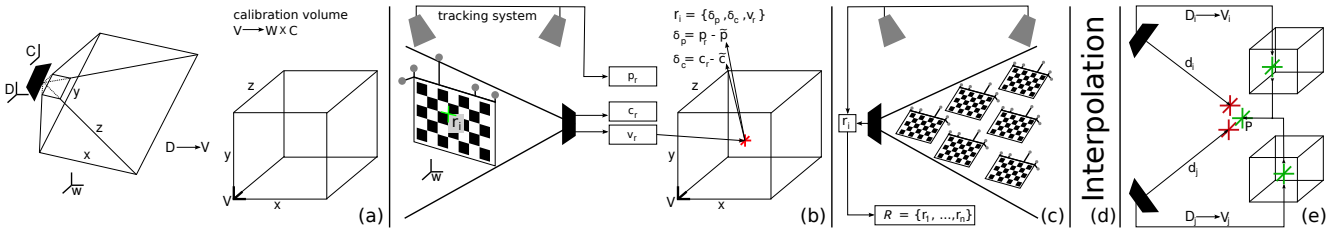


Figure 2: Overview of our proposed method: (a) We transform the depth sensor space D into a normalized calibration volume space V. Every point $\mathbf{v} \in V$ maps to a point $\mathbf{p}$ in our joint coordinate system W and to a coordinate $\mathbf{c}$ in the sensor's color texture space C. In the first step, the calibration volume is initialized at each voxel with values $\widetilde{\mathbf{p}} \in W$ and $\widetilde{\mathbf{c}} \in C$. These values are computed from an initial calibration Υ of the sensor. At this point the volume is a valid mapping with the accuracy of the initial calibration. The central idea of our method is to correct the calibration volume based on a set of references $\{R\}$ which we sample at the crossing points of a tracked checkerboard as depicted in (b). For each reference $r_i \in \{R\}$ its texture coordinate $\mathbf{c_r} \in C$ in the sensor's color image is detected and its normalized volume coordinate $\mathbf{v_r} \in V$ is computed using z from the depth and x, y from infrared image of the sensor. The tracking system monitors the checkerboard and reports the measured 3D world position $\mathbf{p_r} \in W$ of $\mathbf{r_i}$. For each r_i the offsets δ_p and δ_c are computed and stored. (c) Calibration is performed by taking references at various checkerboard locations in our area of interest. Note that the volume is not updated in this phase. (d) In a final interpolation phase the volume is locally corrected at each voxel based on the offsets at its neighbors in $\{R\}$. (e) During runtime, a point $\mathbf{p} \in W$ is reported by, e.g., two sensors to be at local positions $d_i \in D_i$ and $d_j \in D_j$. The correct location of $\mathbf{p}$ can be reconstructed from both sensors by lookups in their calibration volumes.

polynomial function to the correspondences, which was used to map depth values to world space positions at runtime. However, they did not provide any quantitative results.

Beck et al. [3] suggested a volumetric approach for the metric correction of individual depth sensors. They used an optical tracking system as a reference to obtain a mapping from a sensor's raw depth values to metric values. Their method significantly improved the depth measuring accuracy compared to the standard approach for raw depth-to-meter conversion. The registration of the sensors into a global reference system was achieved by a geometric registration using a custom box-shaped tracked calibration target. However, the overall registration of multiple depth sensor contributions was still not perfect throughout the volume of interest. In particular, using only a single rigid body transform for the extrinsic registration per sensor produces good results in some areas, whereas in other areas the contributions match poorly. Deng et al. [8] suggested a smooth field of rigid transforms to improve on this. Their method is able to pairwise match RGBD-sensors. They first capture a set of correspondences at different locations in the scene using a checkerboard that is simultaneously seen by two sensors. Based on these correspondences, they construct a 3D grid of rigid transforms which is then used to locally interpolate transformations during runtime. As a result, the video textures, as well as the captured 3D scene for two or more Kinects, matches with higher accuracy compared to methods which only use a single rigid transform per camera.

However, the limitations of inter-camera based approaches as presented by [16, 12, 8] are that at least two sensors have to overlap and that the registration results in small geometrical distortions of the captured scene. Our approach preserves geometrical consistency by an implicit correction of the depth sensors metric measurement and, as we register each RGBD-sensor individually into a joint reference coordinate system, the sensors do not have to overlap.

More recently, Avetisyan et al. [2] presented an approach for depth sensor calibration to overcome depth measurement inaccuracies. The depth calibration is performed by sweeping a tracked checkerboard through the capturing space in front of the sensors and constructing a 3D look up table that maps depth values identified at the checkerboard crossing points to metric distances which are measured by an optical tracking system. Their method achieves a slightly better metric depth accuracy than the similar approach by [3] (0.8-1.2 cm vs. 1-2 cm). However, the extrinsic calibration (relation between color and depth), as well as the registration to an external reference system, is achieved by a single rigid transform. Therefore, their proposed method might still contain the problem of varying fusion quality throughout a larger capturing volume. To address the problem of varying fusion quality, our approach performs the calibration of intrinsic parameters and the external registration as an integrated process which ensures the best possible fusion of multiple contributions throughout a large capturing space.

A more general limitation of depth correction approaches like [3, 2], which are based on sweeping, is the missing synchronizing between the depth sensor and the tracking system. This can lead to interferences between the different sampling processes and, therefore, inaccuracies in the depth calibration. We therefore prefer a calibration method that operates with a static target.

Table 1 shows a comparison of the characteristics of the methods that are most related to our work. All these methods focus on the registration of multiple RGBD-sensors and aim for a perfect fusion of multiple RGBD-sensors throughout a large capturing volume. While most methods either apply an explicit depth calibration or use an inter-camera approach, they all depend on the accuracy of a large set of interdependent and error-prone parameters. In contrast, our integrated process makes the resulting calibration independent of any specific lens or camera model and it is independent of the real type of involved distortions. We are convinced that 3D capturing systems, as well as 3D skeleton tracking systems, can benefit from the accuracy and simplicity of our calibration method.

3 Calibration Method

Notations: Our method uses several coordinate systems: The 3D joint world coordinate system W in Euclidean space where all sensors will be calibrated and registered to. The 3D depth sensor coordinate system D with (x,y) image coordinates and the sensors z coordinate and its corresponding reference frame in Euclidean space D'. The 2D color camera coordinate system C and its corresponding reference frame in Euclidean space C'. A normalized 3D volume space V where our calibration is performed.

We start from an initial calibration Υ of an RGBD-sensor. Υ is defined by the intrinsic and extrinsic parameters of the sensor and the rigid transformation that maps from the 3D depth reference frame D' to the 3D color reference frame C'. Suppose that Υ is calibrated such that the captured values from depth space D are registered to a joint coordinate system W. We are now interested in the accuracy of Υ and measure it as follows: A tracking system monitors the 3D position of a tracked checkerboard and its crossing points in W. The crossing points can also be detected in the sensor's color space C and depth space D, using the infrared image that is

provided by the sensor. For a single crossing point, we then know its correct position $\mathbf{p} \in W$, $\mathbf{c} \in C$ and $\mathbf{d} \in D$. If we compute the location of the same crossing point that is located at $\mathbf{d}$ using Υ its positions, we will end up at a slightly different location $\widetilde{\mathbf{p}}$ and $\widetilde{\mathbf{c}}$. The difference δ_p between $\mathbf{p}$ and $\widetilde{\mathbf{p}}$, as well as the difference δ_c between $\mathbf{c}$ and $\widetilde{\mathbf{c}}$, is caused by the inaccuracy of Υ. In particular, δ_p and δ_c define the local error of Υ at the corresponding value $\mathbf{d}$.

By sampling a set of references R at different locations in W, we are able to locally correct Υ and obtain our new calibration Ω. The correction of Υ is performed by an interpolation step in normalized volume space V, which we denominate calibration volume. In summary, our proposed calibration method involves the following steps (cf. Figure 2):

1. Computation of an initial calibration Υ
2. Initialization of the calibration volume
3. Reference sampling of R at multiple locations
4. Correction of the calibration volume based on interpolation

In the following sections, we will describe the above steps in detail.

3.1 Initial Calibration

The intrinsic parameters of the sensor's color and depth camera are computed using OpenCV [6]. The extrinsic calibration $T_{D'\to W}$ which registers the sensor into our joint coordinate system W is computed using the algorithm and the reference calibration cube from [3].

With Υ we can compute the position $\widetilde{\mathbf{p}}$ for each value $\mathbf{d}$ by first computing its position d' in the 3D reference frame of the depth sensor:

$$\begin{aligned} d'_x &= \mathbf{d_z} \cdot \frac{\mathbf{d_x} - p_{d_x}}{f_{d_x}} \\ d'_y &= \mathbf{d_z} \cdot \frac{\mathbf{d_y} - p_{d_y}}{f_{d_y}} \\ d'_z &= \mathbf{d_z} \end{aligned} \tag{1}$$

where f_d is the focal length and p_d the principal point of the depth camera. d' is then transformed by $T_{D'\to W}$ resulting in:

$$\widetilde{\mathbf{p}} = T_{D'\to W} \cdot d' \tag{2}$$

In addition, the texture coordinate $\widetilde{\mathbf{c}}$ can be computed by, first transforming d' into the 3D color camera reference frame:

$$c' = T_{D'\to C} \cdot d' \tag{3}$$

then $\widetilde{\mathbf{c}}$ is obtained computing:

$$\begin{aligned} \widetilde{\mathbf{c}}_u &= p_{c_x} + \frac{f_{c_x} \cdot c'_x}{c'_z} \\ \widetilde{\mathbf{c}}_v &= p_{c_y} + \frac{f_{c_y} \cdot c'_y}{c'_z} \end{aligned} \tag{4}$$

where f_c is the focal length and p_c the principal point of the color camera.

3.2 Calibration Volume

As our calibration is performed in volume space, we transform the depth sensor reference space D into a normalized volume space V having its origin at the lower left front crossing point. The x, y coordinates are normalized in relation to the width and height of the depth image. The z component (the raw depth) is normalized to a range inside the near and far plane of the sensor:

$$z_{norm} = \frac{z - s_{near}}{s_{far} - s_{near}} \tag{5}$$

We now initialize the calibration volume based on the initial calibration Υ. For every voxel $\mathbf{v} \in V$ we compute its position $\widetilde{\mathbf{p}}$ using (1) and (2) and the corresponding texture coordinate $\widetilde{\mathbf{c}}$ using (3) and (4). At this point the calibration volume is a valid mapping with the accuracy of Υ. A lookup of a position $\widetilde{\mathbf{p}}$ or a texture coordinate $\widetilde{\mathbf{c}}$ can be performed by a tri-linear interpolation in the calibration volume. We denote these lookups as $\Upsilon_W(\mathbf{v})$ and $\Upsilon_C(\mathbf{v})$.

The size of the calibration volume can be asymmetric, corresponding the higher resolution of the depth sensor. For example, we have chosen a size of $128 \times 128 \times 256$.

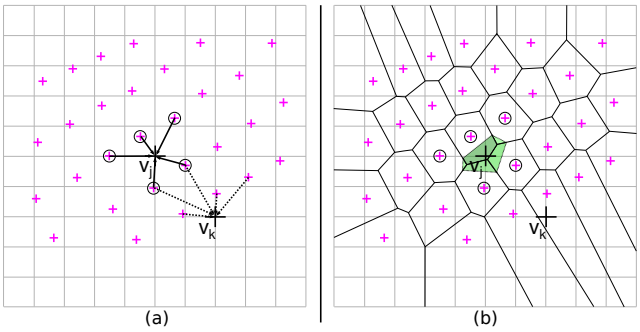


Figure 3: Illustration of the interpolation at a voxel $v_j \in V$, the reference set R is shown in purple. (a) For *IDW* the offsets from the marked 5 nearest references are interpolated. The weights are defined by the inverse distances. (b) For *NNI* the corresponding Voronoi-diagram is depicted. The weights are schematically illustrated by the intensity of the color of the (temporary) Voronoi-subcells. Note that, for a voxel $v_k \in V$, which is outside the convex hull of R, *IDW* extrapolates from its neighbors, illustrated by dashed arrows. In contrast *NNI* is not defined at voxel $v_k \in V$.

3.3 Reference Sampling

Reference sampling is performed by placing a tracked checkerboard at different locations in the area of interest inside the camera frustum of the sensor. In this phase the rectification of the sensor images is ignored because our method implicitly compensates the image distortions inside the calibration volume. For a reference sample $\mathbf{r_i} \in V \times W \times C$, the texture coordinate $\mathbf{c_r} \in C$ of the actual checkerboard crossing point is detected in the sensor's color image and its volume coordinate $\mathbf{v_r} \in V$ is computed using the normalized z from the depth and x, y from the detected checkerboard crossing point of the infrared image. The tracking system monitors the checkerboard and reports the measured reference $\mathbf{p_r} \in W$. In summary, a reference sample $\mathbf{r_i}$ consists of the correction tuple $(\mathbf{v_r}, \delta_p, \delta_c)_i$ with:

$$\begin{aligned} \delta_p &= \mathbf{p_r} - \Upsilon_W(\mathbf{v_r}) \\ \delta_c &= \mathbf{c_r} - \Upsilon_C(\mathbf{v_r}) \end{aligned} \tag{6}$$

As a result of the sampling step, we obtain a set R which then becomes the input for the interpolation of the calibration volume. Note that, at the moment of sampling the image frames of the sensor and the tracking system, the checkerboard must not move. We furthermore reduce noise in the measurements of the sensors by filtering and averaging the acquired values over a period of 30 frames.

3.4 Interpolation

The purpose of this step is to correct the calibration volume (meaning $\widetilde{\mathbf{p}}$ and $\widetilde{\mathbf{c}}$) at each voxel $\mathbf{v} \in V$ by interpolating the offsets from references in the local neighborhood of R. The type of interpolation in our case is a problem similar to scattered data interpolation. After the interpolation, the calibration volume contains our new calibration Ω with a higher accuracy compared to Υ. The method operates on individual voxels one after another.

In the following, we denote a voxel which is corrected as v_j. The position $\Upsilon_W(v_j)$ and texture coordinate $\Upsilon_C(v_j)$ at v_j are corrected by interpolating the offsets $(\delta_p, \delta_c)_i \in R$ to:

$$\delta_{p_{correction}} = \frac{1}{\sum_{i=1}^{k} \Phi(\mathbf{v_{r_i}})} \cdot \sum_{i=1}^{k} \Phi(\mathbf{v_{r_i}}) \cdot \delta_{p_i}, \tag{7}$$

and

$$\delta_{c_{correction}} = \frac{1}{\sum_{i=1}^{k} \Phi(\mathbf{v_{r_i}})} \cdot \sum_{i=1}^{k} \Phi(\mathbf{v_{r_i}}) \cdot \delta_{c_i}, \tag{8}$$

with a weighting function Φ and a neighborhood of k reference samples. Both, Φ and the neighborhood k depend on the interpolation method. $\widetilde{\mathbf{p}}$ and $\widetilde{\mathbf{c}}$ are then corrected by adding the corresponding interpolated offsets:

$$\begin{aligned} \widetilde{\mathbf{p}} &= \widetilde{\mathbf{p}} + \delta_{p_{correction}} \\ \widetilde{\mathbf{c}} &= \widetilde{\mathbf{c}} + \delta_{c_{correction}} \end{aligned} \tag{9}$$

As a result, the calibration Υ is corrected at voxel v_j and the calibration volume is updated.

We investigated two different interpolation methods, which both address Φ and k: inverse distance weighting (*IDW*) and natural neighbor interpolation (*NNI*). Figure 3 illustrates both schemes. *NNI* is state-of-the-art in scattered data interpolation and it is known to have $C^{(1)}$ property inside the convex hull of its underlying delaunay triangulation [14].

In a basic *IDW* interpolation, the neighborhood k is fixed and the function Φ_{IDW} computes the weight of each neighbor as the inverse distance in normalized volume coordinates between the current voxel v_j and the reference $\mathbf{r_i} \in R$ with correction tuple $(\mathbf{v_r}, \delta_p, \delta_c)_i$:

$$\Phi_{IDW}(\mathbf{v_{r_i}}) = \frac{1}{distance(v_j, \mathbf{v_{r_i}})} \tag{10}$$

Natural neighbor interpolation is based on the construction of a Voronoi-diagram for the volume positions $\mathbf{v_{r_i}} \in V$ from the set R. For a detailed explanation of Voronoi space decomposition, we refer to [14, 7]. The weight for each natural neighbor $r_i \in R$ of a voxel v_j is computed by temporarily inserting its position into the Voronoi-diagram. This insertion leads to a new Voronoi-cell υ_j which covers a part the volume from each neighboring cell υ_{r_i}. Let Θ be a function that computes the volume of a Voronoi-cell. The weighting function Φ_{NNI} then is:

$$\Phi_{NNI}(r_i) = \frac{\Theta(\upsilon_{r_i})}{\Theta(\upsilon_j)} \tag{11}$$

4 Evaluation

Our evaluation was performed for the Microsoft Kinect V1 and the developer release of the Microsoft Kinect V2, but our method also applies to any combination of depth and color cameras which outputs an infrared image of the depth sensor. For a detailed specification of the Kinect V1, we refer to the literature [18]. The depth sensor of the Kinect V2 uses time-of-flight for depth measuring with a resolution of 512×424, a field of view of 70×60 degrees, and a depth precision of 14bit in the range of 0.5 to 4.5 meters. The color camera is full HD, but we cropped the color image to a resolution of 1280×1080 to match the field of view of the depth sensor. We implemented our proposed method in C/C++ using OpenCV [6] for checkerboard crossing point detection and CGAL [7] for natural neighbor interpolation *NNI* and an alpha version of the open source driver [5] for the Kinect V2. For reference measurements of the set R, we used the tracking system from A.R.T. [1]. The accuracy of the tracking system is in the range of 1-2 mm throughout a large space of about 4 m x 3 m x 2.5 m. We used a checkerboard of size 7×5 crossing points which we printed onto a warp-resistant board mounted on a custom stand (Figure 4 (left)). The initial calibration for volume initialization was performed using the method described in [3].

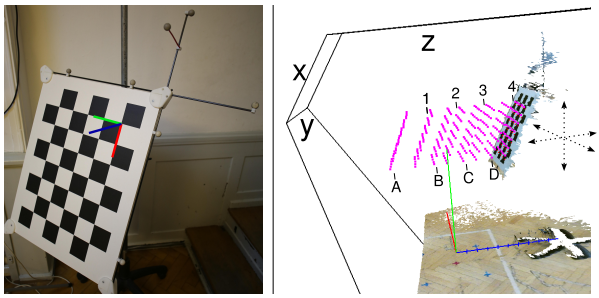


Figure 4: (left) The custom checkerboard with attached tracking markers and the local coordinate system linked to them. The crossing point distance is 7.5 cm. (right) Illustration of a small set of subsequent reference samples (marked in purple). Per checkerboard location, 35 reference samples are captured. In this example the input sets would be $R_{dense} = \{A,B,C,D\}$ and $R_{sparse} = \{A,C\}$ and the accuracy would be evaluated at intermediate checkerboard locations forming the reference set $R_m = \{1,2,3,4\}$.

4.1 Evaluation Approach

We were interested in the accuracy of our calibration volume at locations between the actual sample location used for generating the volume. We also wanted to investigate the effect of sparse and dense reference sampling. Therefore we split the set R after reference sampling into subsets grouped by board locations as follows: Reference samples are inserted into the disjoint sets R_{calib} and R_m, alternating per board location. We evaluated the accuracy of the calibration with a dense input set ($R_{dense} = R_{calib}$) and a sparse set R_{sparse}, a subset of R_{dense}. Reference samples in R_m are used for evaluation only (Figure 4 (right)). The evaluation of the accuracy of our calibration was performed in the following steps:

1. Initialization of the calibration volume.
2. Reference sampling of R.
3. Split of R into the sets R_{dense} and R_{sparse} as input for interpolation and R_m for evaluation.
4. Correction of the calibration volume using interpolation schemes *NNI* and *IDW* for different neighborhoods k.
5. Evaluation of the achieved accuracy at the references from R_m.

5 Results and Discussion

5.1 Calibration Accuracy

We measured the accuracy of our method and calibrated both Kinect versions into our joint coordinate system. For the Kinect V1, we applied the initial calibration with the method from [3]. For the Kinect V2, we computed the initial calibration for the intrinsic calibration using OpenCV [6] and the method from [3] for the extrinsic

Table 2: Average absolute errors for the Kinect V1 in 3D world space and 2D texture space compared to [3]; measured for R_m based on the input sets R_{dense} and R_{sparse} for different interpolation schemes and volume resolutions (upper row $64 \times 64 \times 128$ lower row $128 \times 128 \times 256$) per method and row. For *IDWk*, only positions inside the convex hull of the input set were evaluated. 3D errors are in mm, 2D errors in texels, standard deviations in parentheses, and maximum errors in brackets.

Method	3D dense	3D sparse	2D dense	2D sparse
Beck et al. [3]	12.8 (5.8)[35.6]	-	1.8 (0.4)[4.0]	-
IDW5	3.5 (2.2)[12.6]	4.1 (2.2)[12.8]	0.5 (0.3)[2.1]	0.5 (0.3)[2.0]
	3.5 (2.1)[12.7]	3.9 (2.3)[12.7]	0.5 (0.3)[2.0]	0.5 (0.3)[2.0]
IDW10	3.4 (2.2)[12.3]	3.7 (2.3)[14.5]	0.5 (0.3)[2.1]	0.5 (0.3)[2.0]
	3.4 (2.1)[12.3]	3.8 (2.3)[14.4]	0.5 (0.3)[2.0]	0.5 (0.3)[2.0]
IDW20	3.4 (2.2)[13.2]	3.9 (2.2)[14.0]	0.5 (0.3)[2.1]	0.5 (0.3)[2.1]
	3.4 (2.2)[13.1]	3.8 (2.2)[13.8]	0.5 (0.3)[2.2]	0.5 (0.3)[2.0]
NNI	3.2 (2.1)[11.2]	3.5 (2.2)[13.8]	0.5 (0.3)[2.1]	0.5 (0.3)[2.0]
	3.1 (2.1)[11.1]	3.6 (2.2)[13.7]	0.5 (0.3)[2.1]	0.5 (0.3)[2.0]

Table 3: Average absolute errors for the Kinect V2 in 3D world space and 2D texture space compared to a rough initial calibration; measured for R_m based on the input sets R_{dense} and R_{sparse} for different interpolation schemes and volume resolutions (upper row $64 \times 64 \times 128$ lower row $128 \times 128 \times 256$) per method and row. For *IDWk*, only positions inside the convex hull of the input set were evaluated. 3D errors are in mm, 2D errors in texels, standard deviations in parentheses, and maximum errors in brackets.

Method	3D dense	3D sparse	2D dense	2D sparse
Initial	35.7 (13.0)[75.0]	-	24.2 (1.0)[28.0]	-
IDW5	3.2 (2.1)[14.0]	4.2 (2.9)[18.5]	0.3 (0.2)[1.1]	0.3 (0.2)[1.2]
	3.2 (2.1)[13.6]	4.1 (2.9)[17.0]	0.3 (0.2)[1.9]	0.3 (0.2)[1.2]
IDW10	3.1 (2.0)[14.7]	4.3 (2.8)[16.0]	0.3 (0.2)[1.2]	0.3 (0.2)[1.2]
	3.1 (2.1)[13.9]	4.2 (2.8)[15.8]	0.3 (0.2)[1.2]	0.3 (0.2)[1.2]
IDW20	3.0 (2.0)[17.7]	4.5 (2.7)[15.8]	0.3 (0.3)[5.0]	0.4 (0.2)[1.1]
	3.0 (2.1)[16.5]	4.4 (2.8)[16.5]	0.3 (0.2)[1.0]	0.4 (0.2)[1.1]
NNI	1.7 (1.0)[5.0]	2.0 (1.2)[5.7]	0.2 (0.2)[1.5]	0.3 (0.2)[1.5]
	1.7 (1.1)[5.8]	2.0 (1.3)[6.9]	0.2 (0.2)[1.3]	0.3 (0.2)[1.9]

registration. The capturing volume was about 1.2 m x 1.8 m x 1.0 m for the Kinect V1 and about 1.5 m x 1.8 m x 1.5 m for the Kinect V2. Depending on the actual mounting of the sensors, the capturing volumes differ, which is mainly due to the different fields of view of the two Kinect versions. We took approximately 2000 reference samples and divided these into R_m of size 1000, R_{dense} of size 1000 and R_{sparse} of size 500. We were interested in the errors in terms of the absolute distance to the ground truth. For the 3D error the ground truth is our tracking system and for the 2D error the ground truth is the detected crossing point in image space at the reference sample $r_i \in R_m$. The average results for *NNI*, as well as *IDW* for different neighborhoods k (5, 10 and 20) and for different resolutions of the calibration volume, are listed in Table 2 for the Kinect V1 and in Table 3 for the Kinect V2.

Our evaluation clearly shows that our method is able to significantly improve the calibration accuracy of the method from Beck et al. [3] for the Kinect V1. Our method reduced the average absolute 3D error from 12.8 mm to 3.1 mm and the average absolute 2D error from 1.8 texel to 0.5 texel. The basic initial calibration of the Kinect V2 resulted in relatively high errors compared to the initial calibration of the Kinect V1. However, we were able to achieve a very high accuracy with our calibration method. Note that the quantitative error evaluation reported in [3] focused on the Kinect V1's z-error only whereas our evaluation is a distance measurement in our joint 3D coordinate system. The relatively high 3D error of 12.8 mm for the method from [3] listed in Table 2 is mainly caused by the rotation and translation inaccuracies due to the rigid body transformation which registers a sensor into the application's coordinate system. In particular, using a single rigid body transform per sensor is one of the main drawbacks of the calibration method for the multi-sensor setup of [3]. In contrast, our proposed method registers each sensor into the application's coordinate system by a 3D-lookup.

It turns out that natural neighbor interpolation produces a much higher accuracy than inverse distance weighted interpolation inside the convex hull of R. In addition, the calibration benefits from the C^1 continuity of *NNI*. On the other hand, *IDW* can extrapolate values at voxels that lie outside the convex hull of R. For practical scenarios it is important that the capturing space is sampled as widely and densely as possible. However, e.g. it is not always feasible taking reference samples near the borders of a sensor's frustum and close to the floor. A modified version of our interpolation phase could blend between *NNI* and *IDW* at such critical borders. Furthermore, our method scales with the density of the reference set (R_{dense} vs. R_{sparse}) and with the resolution of the calibration volume. In our experiments we also tested additional volume sizes – $32 \times 32 \times 64$ produced very poor results whereas a very dense volume of $256 \times 256 \times 512$ did not further improve accuracy.

The most important part of our method is the reference sampling step. First, it is critical in terms of synchronization. We therefore ensure that all involved data streams (images, checkerboard target, and tracked checkerboard pose) are stable. This is an important difference to the sweeping approaches of [3, 2], because we avoid artifacts due to interferences of the different sampling frequencies and times. Second, good lighting conditions are critical for the reference sampling step because it relies on an accurate and stable crossing point detection in the involved images. The crossing point detection in the color image of both sensor types works well if appropriate room lighting is ensured. The infrared image of the Kinect V2 is generally also of sufficient quality. However, the infrared image of the Kinect V1 sees the structured light pattern of its infrared projector. In order to make the crossing point detection more reliable and stable, we apply a 5 x 5 median filter to the infrared image as suggested by [2] (Figure 5).

The interpolation phase took about 2 minutes for the natural neighbor interpolation and 3 to 6 minutes for the inverse distance methods, depending on the volume resolution and number of neighbors. The use of an acceleration structure for the neighbor search could speed up this phase. Of course, the most time-consuming part of our method is the reference sampling. For the Kinect V1, it takes up to 30 minutes and for the Kinect V2, about 20 minutes to take enough samples for a capturing space of about 1.5 m x 1.8 m x 1.5 m. The difference is caused by the fact that the driver is not able to simultaneously read infrared and depth images from the Kinect V1, but rather has to switch between the streams. The ideal case would be a sweeping-based reference sampling phase. However, current hardware does not support synchronization and the frame rates of the current RGBD-sensors are not high enough to capture sharp images of a moving target.

5.2 Dense Reference Sampling

It is obvious that our method depends on the density of the acquired reference samples. We sampled a very dense sequence of checkerboard locations in order to find an upper limit of the accuracy for our method. We set the size of our calibration volume to $128 \times 128 \times 256$ and sampled a distance range of about 0.9 m to 2.1 m in front of the Kinect V1 and of about 0.9 m to 2.4 m in front of the Kinect V2. The subsequent board locations had a distance of about 5 cm and we captured 800 samples for the Kinect V1 and 1000 samples for the Kinect V2. The evaluation was performed

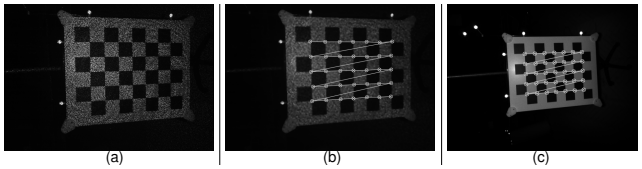


Figure 5: Checkerboard crossing point detection in the infrared image of the depth sensor. (a) Kinect V1 uses structured light, crossing point detection fails. (b) Successful crossing point detection with a 5 x 5 median filter applied. (c) Stable crossing point detection for Kinect V2.

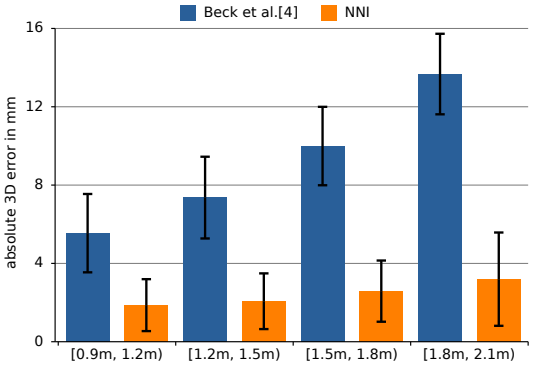


Figure 6: Absolute 3D errors and standard deviations in mm with increasing distance for a densely sampled range of about 0.9 - 2.1 meters in front of the Kinect V1, clustered into 4 subranges.

in the same way as before, i.e. the odd checkerboard locations were used as reference samples for construction of the calibration volume while the samples at the even checkerboard locations were used for evaluation (Figure 4 (right)). The resulting accuracy for the Kinect V1 is depicted in Figure 6. For the Kinect V1, the initial calibration was performed using the method from [3] and the rigid transform for the calibration cube was captured at a distance of 1.1m in front of the sensor, which corresponds to the range $[0.9\ m, 1.2\ m)$ in Figure 6. One can see that the single rigid transform produces only good results in the proximity of the calibration samples while the error increases with distance. In contrast, our method is able to achieve a much higher accuracy throughout the whole sampling range. However, the accuracy still decreases with distance. For the Kinect V2, our method is able to achieve a high accuracy of around 1.5 mm at a distance of 0.9 m and 2.5 mm at a distance of 2.4 m.

5.3 Multiple Sensors

In a multi-RGBD-sensor setup, we individually calibrate each sensor and thus our calibration method does not incorporate information from other sensors. Methods such as [16, 12, 8] incorporate inter-camera information to minimize distances between the measurements of individual sensors. The amount of captured correspondences that are used by these methods theoretically improve the calibration result. However, these methods do not register the sensors into an application's coordinate system but rather into a warped space, which can be a drawback because the preservation of angles and distances is not guaranteed. In contrast, our proposed method is designed to directly map depth values from sensor space to 3D positions in an Euclidean space with very high accuracy. The accuracy scales with the amount of reference samples which are captured during the calibration. However, our calibration process relies heavily on a precise mounting of the attached tracking target since it links the checkerboard crossing points to the coordinate system of the tracked checkerboard. As a consequence, rotation and translation errors result in a misalignment between the registration and the ground truth. Thus, the accurate registration of the checkerboard's local coordinate system is a very important precondition for the overall accuracy in a multi-sensor setup. More accurate manufacturing and measuring facilities could potentially improve our results.

We evaluated the registration of multiple sensors by measuring pairwise 3D distances of the reconstructed checkerboard crossing points from two sensors. The checkerboard was positioned at a distance of approx. 1.8 m from the sensors. The sensors were positioned about 45 degrees left and right of the checkerboard. The average pairwise crossing point distance was 5.8 mm with a standard deviation of 1.2 mm (cf. Figure 7). This overall error is affected by the sensor resolution and noise, as well as the mechanical alignment of markers for both involved sensor systems. We performed our measurements running both sensors simultaneously and also sequentially. However, inter-sensor interference did not have any significant influence on the accuracy in our setup. The pairwise average 3D error of about 5.8 mm is within the limits of the accuracy of our method, as the spatial extent of a depth pixel covers 6.2 mm on the surface of the checkerboard at that distance.

The motivation for our research is the development of a 3D capturing system for 3D telepresence. We calibrated a setup of three Kinect V2 with our proposed method taking approximately 2000 reference samples for each sensor. The capturing space was about 1.5 m x 1.8 m x 1.5 m and the sensors were positioned at an angle of about 90 degrees, allowing for capturing from three sides. However, our calibration method also allows other sensor configurations. In particular, the amount of overlap can be chosen to fit the individual application. Figure 1 shows screen shots and close-ups of real-time reconstructions based on the reconstruction pipeline from [3]. As one can see, our calibration results in a very good fusion of the two overlapping sensors.

In addition, we tracked our calibration target with our calibrated multi-Kinect V2 setup. We therefore used the depth values at the checkerboard crossing points to look up 3D positions in the respective calibration volumes in order to calculate the 3D pose of the checkerboard in real-time. Figure 1 illustrates the quality of registration in terms of the coincidence between two coordinate systems, one using the reference tracking from [1] and the other using our method. Please also refer to the video figure.

5.4 Application Runtime

We measured the influence of our method for real-time scenarios on a PC workstation equipped with an Intel® Core™ i7 X980 six-core processor running at 3.33 GHz and a GeForce™ GTX 680 graphics card with 2 GByte VRAM. The calibration volume serves as a 3D lookup table during runtime and is stored in the memory of the graphics card. Our application is implemented in OpenGL/GLSL. As the maximal number of channels for a 3D texture is four, the volume is split into two volumes, one for the positions $\mathbf{p} \in W$ (three channels, GL_RGB) and one for texture coordinates $\mathbf{c} \in C$ (two channels, GL_RG), both of type GL_FLOAT. This introduces a considerable amount of additional memory. E.g. for a calibration volume of size $128 \times 128 \times 256$ the memory overhead is 82 MByte for one RGBD-sensor, resulting in 410 MByte for a 3D capturing system which consists of five sensors. However, this overhead is not critical for current graphics cards.

The lookups require two tri-linear interpolations per depth value (one for the position and one for the texture coordinate) and are hardware accelerated on the GPU via built-in shader functions. While it is not possible to measure the performance impact of the lookups directly, we measured the rendering times of a typical real-time reconstruction from three RGBD-sensors with and without lookups. It turns out that the additional costs for tri-linear inter-

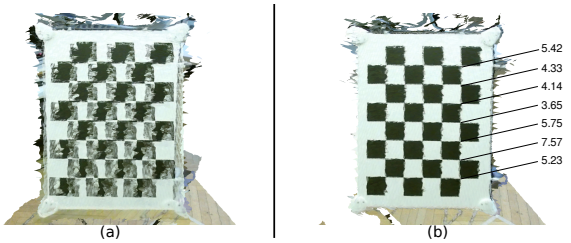


Figure 7: Reconstruction of our checkerboard without (a) and with (b) our volumetric calibration for two overlapping Kinect V2 sensors. (b) We measured a pairwise checkerboard crossing point distance of 5.8 mm on average for all 35 checkerboard crossing points. Seven pairwise distances in mm are illustrated.

polation are below 0.5 milliseconds.

Our approach can reduce the overall runtime latency of a 3D capturing system compared to a model-based calibration method, because no image rectification processing must be performed. We measured image rectification costs of 4 milliseconds for an RGB color image of size 1280×1080 and 2 milliseconds for a depth image of size 512×424 on average, using cvRemap from OpenCV [6] as it is needed by the methods from [16, 3]. At first glance, the time savings of our method might seem small and the rectification could also be accomplished on the GPU. However, if a 3D capturing system has to process the image streams of multiple sensors, the costs for image rectification become increasingly noticeable.

6 Conclusion and Future Work

The calibration and registration of multiple RGBD-sensors is a challenging task. In order to obtain a perfect fusion of all involved cameras in three dimensional space, most approaches have to identify a large set of intrinsic and extrinsic parameters to fit an underlying mathematical model with high accuracy. We realized an integrated method for the accurate calibration and registration of multiple RGBD-sensors into a joint coordinate system which does not rely on a precise identification of these parameters. Our approach starts with an initial sensor calibration that is then locally fitted based on a set of references which are sampled by placing a tracked checkerboard at different locations inside the capturing space. Each reference is defined by the world space positions of the crossing points of the checker board pattern and their corresponding positions in the color and depth camera spaces. The reference set is then used to construct a single calibration and registration volume per RGBD-sensor which implicitly integrates all the intrinsic and extrinsic parameters. As a result, our calibration volume maps raw depth sensor values in a single step into a joint coordinate system and to their associated color values.

Our evaluation shows that we are able to register the sensors with an average accuracy of about 4-6 mm for the Kinect V1 and 2-3 mm for the Kinect V2 into our joint coordinate system. We also achieved a texture coordinate deviation smaller than 0.8 pixel for the Kinect V1's color camera and smaller than 0.5 pixel in the Kinect V2's color camera. We identified natural neighbor interpolation to be a robust and high quality interpolation scheme when the acquired reference samples are densely distributed inside the capturing space. In addition, real-time applications can benefit from our approach because image rectification processes can be avoided and the end-to-end latency of 3D capturing systems can be reduced.

The main constraints of our method are that the accuracy can only be achieved inside the convex hull of the reference samples and that it relies on dense sampling. Furthermore, the accuracy of the involved tracking system and the precise calibration of the tracked checkerboard are also of significant influence. Our calibration and registration process could be accelerated by sweeping the checkerboard through the capturing space. However, this would require that the tracking system and the RGBD-sensors are in perfect sync, which might be possible with next generation hardware.

Acknowledgements

The authors wish to thank the CGAL-development community.

References

[1] ART. Advanced realtime tracking gmbh, 2014. http://www.ar-tracking.com/home/.

[2] R. Avetisyan, M. Willert, S. Ohl, and O. Staadt. Calibration of depth camera arrays. In *Proc. of the 13th SIGRAD 2014 Conference of the Swedish Eurographics Chapter, Eurographics Association*, 2014.

[3] S. Beck, A. Kunert, A. Kulik, and B. Froehlich. Immersive group-to-group telepresence. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):616–625, April 2013.

[4] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of CHI 2012*, pages 199–208, New York, NY, USA, 2012. ACM Press.

[5] F. Blake, J. Echtler and C. Kerl. Openkinect/libfreenect2: Driver for kinect for windows v2 (k4w2) devices. https://github.com/OpenKinect/libfreenect2.

[6] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[8] T. Deng, J. C. Bazin, T. Martin, C. Kuster, J. Cai, T. Popa, and M. H. Gross. Registration of multiple rgbd cameras via local rigid transformations. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2014.

[9] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, H. F. G. Bishop, R. Bajcsy, S. W. Lee, H. Farid, and T. Kanade. Virtual space teleconferencing using a sea of cameras. In *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery (MRCAS 94)*, pages 161–167, 1994.

[10] H. Fuchs, A. State, and J.-C. Bazin. Immersive 3d telepresence. *Computer*, 47(7):46–52, July 2014.

[11] D. Herrera C, J. Kannala, and J. Heikkila. Joint depth and color camera calibration with distortion correction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10):2058–2064, Oct. 2012.

[12] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg. Omnikinect: real-time dense volumetric data acquisition and applications. In *Proc. of VRST 2012*, VRST '12, pages 25–32, New York, NY, USA, 2012. ACM Press.

[13] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, 1997.

[14] H. Ledoux and C. Gold. An efficient natural neighbour interpolation algorithm for geoscientific modelling. In *Proc. 11th Int. Symp. Spatial Data Handling*, pages 23–25, 2004.

[15] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Proc. of ISMAR 2011*, pages 137–146, Washington, DC, USA, 2011. IEEE Computer Society.

[16] A. Maimone and H. Fuchs. A first look at a telepresence system with room-sized real-time 3d capture and large tracked display. In *Proc. of ICAT 2011*, New York, NY, USA, 2011. ACM Press.

[17] C. Raposo, J. Barreto, and U. Nunes. Fast and accurate calibration of a kinect sensor. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 342–349, June 2013.

[18] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain*, pages 1154–1160, 2011.

[19] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, Nov. 2000.

[20] Q.-Y. Zhou and V. Koltun. Simultaneous localization and calibration: Self-calibration of consumer depth cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 454–460, June 2014.