Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 57(3), 810–821(2016)

Diana-Margarita Córdova-Esparza, Juan R. Terven, Hugo Jiménez-Hernández, Alberto Vázquez-Cervantes, Ana-Marcela Herrera-Navarro, Alfonso Ramírez-Pedraza

Multiple Kinect V2 Calibration

DOI 10.7305/automatika.2017.02.1758
UDK 681.772.01-047.58:53.089.6

Original scientific paper

In this paper, we propose a method to easily calibrate multiple Kinect V2 sensors. It requires the cameras to simultaneously observe a 1D object shown at different orientations (three at least) or a 2D object for at least one acquisition. This is possible due to the built-in coordinate mapping capabilities of the Kinect. Our method follows five steps: image acquisition, pre-calibration, point cloud matching, intrinsic parameters initialization, and final calibration. We modeled radial and distortion parameters of all the cameras, obtaining a root mean square re-projection error of 0.2 pixels on the depth cameras and 0.4 pixels on the color cameras. To validate the calibration results we performed point cloud fusion with color and 3D reconstruction using the depth and color information from four Kinect sensors.

Key words: Camera calibration, Kinect V2 system, Multiple cameras

Umjeravanje Kinect V2 sustava s više kamera. U ovom je radu predložena metoda za jednostavno umjeravanje proizvoljnog broja senzora Kinect V2. Izvodi se istovremenim snimanjem objekta s više kamera. Jednodimenzionalan objekt potrebno je snimiti s najmanje 3 različite orijentacije, a dvodimenzionalan s najmanje jedne orijentacije. Istovremeno snimanje s više kamera moguće je zahvaljujući integriranom mapiranju koordinata u Kinect sustavu. Predložena metoda izvodi se u pet koraka: akvizicija slike, predumjeravanje, usklađivanje oblaka točaka, inicijalizacija intrinzičnih parametara i konačno umjeravanje. U radu su modelirani radijalni i distorzijski parametri svih kamera, pri čemu se ostvaruje korijen srednje kvadratične pogreške ponovne projekcije iznosa 0:2 piksela na kamerama dubine i 0:4 piksela na kamerama u boji. Za validaciju rezultata umjeravanja provedena je fuzija oblaka točaka s rekonstrukcijom trodimenzionalnog objekta i boje korištenjem informacije o dubini i boji s četiri Kinect senzora.

Ključne riječi: umjeravanje kamere, Kinect V2, više kamera

1 INTRODUCTION

Multiple calibrated RGB-D cameras have been used in a wide variety of applications such as 3D reconstruction [1–3], people tracking [4–6], motion capture [7–9], object tracking [10], augmented reality [11–13], and telepresence [14–16] among others.

Camera calibration is required in all these applications to obtain metric measurements from the images. It has been extensively studied, and there are several techniques such as the use of 3D calibration objects *e.g.,* orthogonal planes [17], 2D objects such as a single moving plane [18], and 1D objects such as a wand with multiple collinear points [19]. For calibrating multiple cameras, the method using 1D objects is preferred since all the cameras can see the same calibration object at the same time [19–22].

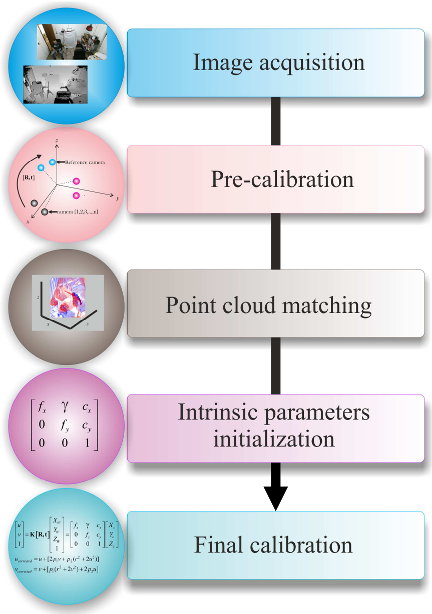
Liu *et al.* [23] describe a method to calibrate the depth and color cameras of a Kinect sensor using a planar model for color camera calibration and a 1D object with three collinear points for depth camera calibration. Svoboda *et al.* [20] proposed a multiple RGB camera calibration using a single moving point. Borguese *et al.* calibrate a stereo system using a rigid bar with two markers, their method rely on accurate known distances between the markers. Pribanic *et al.* extend this method to multi-camera systems using accurate known distances between markers and orthogonality constraints. Auvinet *et al.* [24] calibrate a network of depth cameras using a plane, they obtain the equation of the plane by integrating depth information of all points on each plane and generate virtual points to calibrate using the classical method from [18]. Macknojia *et al.* [25] proposed a calibration technique for multiple Kinect sensors using a regular checkerboard pattern illuminated by external incandescent lamps such that it is visible in IR and color. They also use the classical method from [18] to find the intrinsic parameters; the extrinsic parameters are found between each pair of sensors by looking at the checkerboard pattern in the overlapping region of the two cameras. Avestisyan [26] introduced a calibration method for depth cameras by tracking a rigid body to obtain the pose of a regular checkerboard object located in the overlapping region of two adjacent cameras. Alexiadis *et al.* [2] also calibrate a network of Kinect sensors; for intrinsic parameters estimation, they use a regular checkerboard pattern and the method from [18] and for extrinsic parameters between the different Kinects, they used a wand with two LEDs of different colors and tracked these LEDs on all the cameras followed by a pairwise stereo calibration as described in [27]. Jones *et al.* [13] auto-calibrate a network of Kinect cameras and light projectors by projecting structured light on the scene using the light projectors, the calibration is done without human intervention and in a few minutes. Almazan and Jones [6] introduced a method for calibrating three non-overlapping Kinect sensors using planes as common features. The coordinate systems are recovered from the parameters of at least three mutually orthogonal planes extracted from each pair of sensors. They built a calibration tool to provide such planes.

In this paper, we propose a calibration method for a network of Kinect V2 sensors that uses a 1D or 2D calibration object with color markers to obtain both, intrinsic and extrinsic parameters for all the cameras. In our experiment, we use four Kinect V2 sensors placed with a viewpoint change of approximately 90 degrees as shown in Figure 2. Our method takes advantage of Kinect V2 coordinate mapping between its cameras to find correspondences between color and depth cameras and to estimate the position of the calibration object in camera space. To test the effectiveness of our method we performed colored point cloud fusion and 3D reconstruction using the depth and color data from the four Kinect V2 sensors.

The main contribution of this paper is the method for calibrating one or more Kinect V2 cameras with just three acquisitions using a 1D object or a single acquisition using a 2D object that can be seen by all the cameras. Our method does not impose restrictions on the minimum number of cameras and exploits the Kinect V2 coordinate mapping capabilities.

2 CALIBRATION PROCESS

In this section, we describe the procedure to calibrate multiple Kinect V2 sensors from a single image acquisition. The calibration pipeline is composed of five steps (see Figure 1): (1) Image acquisition, (2) Pre-calibration, (3) Point cloud matching, (4) Intrinsic parameters initialization, and (5) Final calibration. We implemented the whole pipeline in MATLAB using the Kin2 toolbox [28], developed by two of the authors. For brevity, in the rest of the section, we will use *Kinect* to refer to the *Kinect V2*.



Slika 1. Calibration Pipeline. The calibration pipeline follows five steps: Image acquisition, pre-calibration, point cloud matching, intrinsic parameters initialization, and final calibration

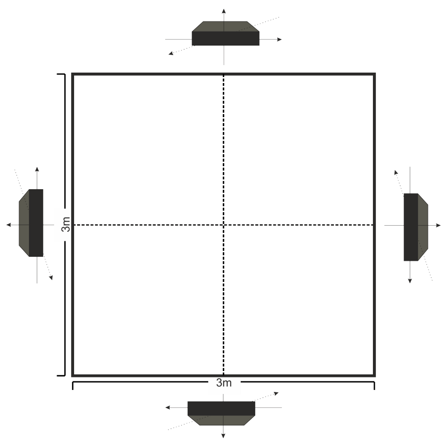
2.1 Image Acquisition

Our experimental setup consists of four Kinect sensors placed at 2 meters high with a viewpoint change of approximately 90 degrees as shown in Figure 2. Each sensor is connected to a computer and the set of computers shared data through a wireless network. The color images have a resolution of 1920×1080 pixels, and the depth and infrared images have a resolution of 512×424 pixels.

We evaluate our calibration procedure with two calibration patterns: a 1D calibration pattern composed of a 60 cm wand with three collinear points of different colors (shown in Figure 3(a)), and a 2D calibration pattern composed of two 60 cm perpendicular sticks with four points of different colors (shown in Figure 3(b)). The calibration pattern must be placed inside the field of view of all the Kinect sensors such that all the sensors can see it at the same time.

One of the computers acts as a server and the other three computers act as clients on a TCP/IP network session. The server synchronizes the data acquisition of the clients, and once the acquisition has finished, it fetches the data from the client sensors.

The goal of the acquisition step is two-fold: (1) to obtain the coordinates of the three collinear points of the calibration pattern in camera space from each Kinect for each acquisition and (2) to obtain a single point cloud from each



Slika 2. Experimental setup. The setup consists of four Kinect sensors placed at 2 meters high with a viewpoint change of approximately 90 degrees.

view. All these data are transferred from the clients to the server.

Camera space refers to the 3D coordinate system used by Kinect. The coordinate system is defined as follows [29]: the origin is located at the center of the infrared sensor on the Kinect; the positive X direction goes to the sensor's left; the positive Y direction goes up; the positive Z goes out in the direction the sensor is facing; and the units are in meters.

The data acquisition procedure follows the next steps:

1. The sensors grab images synchronously. On each sensor, we use the color camera to detect the colored markers. For this, we search for small color blobs that satisfy certain constraints. For example, on the 1D calibration pattern, the color points should lie on a line with the fixed length, and for the 2D pattern, the color points must have fixed distances given by the 2D pattern. The red, green, blue, and yellow markers are defined as $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ respectively. To count as a valid frame, all the cameras must found the three or four markers.

2. Using the coordinate mapping, map the coordinates of $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ from color space to camera space obtaining $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, and $\mathbf{D}_i$ where $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$ $\in \mathbb{R}^3$ and $i \in \{1, 2, 3, 4\}$ is the Kinect id.

3. Using the depth camera and the coordinate mapping, map the complete depth frame (424 $\times$ 512 depth values) to camera space points producing a point cloud $\mathbf{PC}_i$ for each Kinect sensor where $\mathbf{PC}_i \in \mathbb{R}^{217088\times3}$.

4. Finally, transfer the 3D coordinates of the points ($\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$) and the point clouds $\mathbf{PC}_i$ from each client to the server.

2.2 Pre-Calibration

The goal of the pre-calibration step is to obtain an estimate of the extrinsic parameters *i.e.* the pose of all the cameras on a global reference in 3D coordinate space. To accomplish this, we need at least three acquisitions from the 1D object to satisfy the constraints and obtain a better performance or a single acquisition from the 2D object. There are eight cameras in total; a pair of depth and color cameras for each Kinect sensor. We define the depth camera of the first Kinect as the reference. The pose is represented by a 4×4 rigid transformation containing the rotation $\mathbf{R}$ and translation $\mathbf{t}$ that align each of the cameras with the reference. To obtain these transformations we used the camera space points $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$ from each Kinect sensor obtained in the data acquisition step. Setting the first Kinect ($i = 1$) as the reference, the problem of obtaining the pose boils down to obtaining the best rotations $\mathbf{R}_i$ and translations $\mathbf{t}_i$ that align the points from the Kinect sensors ($\mathbf{M}_i = [\mathbf{A}'_i\mathbf{B}'_i\mathbf{C}'_i\mathbf{D}'_i]$, $i \in \{2, 3, 4\}$) to the points in the reference Kinect ($\mathbf{M}_1$). We wish to solve for $\mathbf{R}_i$ and $\mathbf{t}_i$:

$$\mathbf{M}_1 = \mathbf{R}_i \times \mathbf{M}_i + \mathbf{t}_i \tag{1}$$

Where $\mathbf{R}_i$ and $\mathbf{t}_i$ are rotations and translations applied to each set of points $\mathbf{M}_i$, $i \in \{2, 3, 4\}$ to align them with the reference $\mathbf{M}_1$.

Following the method described in [30] for *corresponding point set registration*; first find the centroids of the 3D points $\mathbf{M}_i$:

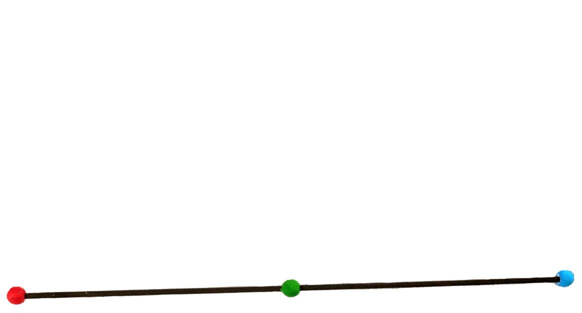
$$centroid_i = \frac{1}{4}\left(\mathbf{A}_i + \mathbf{B}_i + \mathbf{C}_i + \mathbf{D}_i\right), \tag{2}$$

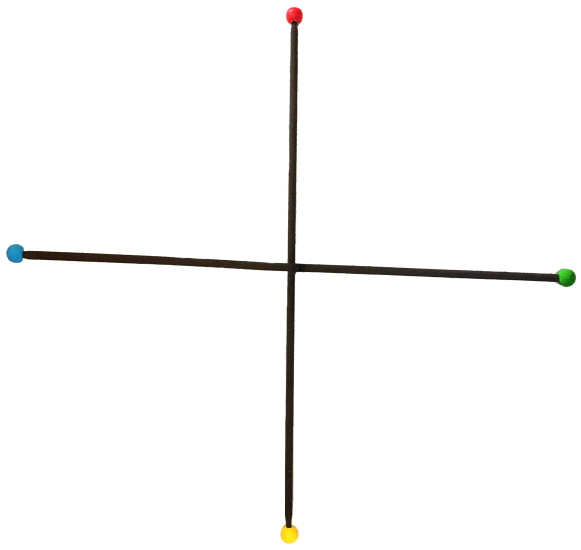
then move the points to the origin and find the optimal rotation $\mathbf{R}_i$ with Equation 3:

$$\begin{aligned} &\mathbf{H}_i = \sum_{j=1}^{3}\left(\mathbf{M}_i^j - centroid_i\right)\left(\mathbf{M}_1^j - centroid_1\right)^T \\ &[\mathbf{U}_i, \mathbf{S}_i, \mathbf{V}_i] = SVD(\mathbf{H}_i) \\ &\mathbf{R}_i = \mathbf{V}_i{\mathbf{U}_i}^T \end{aligned} \tag{3}$$

where $\mathbf{H}_i$ is the covariance matrix of the i^{th} Kinect and SVD denotes the singular value decomposition. Finally, we found the translation $\mathbf{t}_i$ with

$$\mathbf{t}_i = -\mathbf{R}_i \times centroid_i + centroid_1 \tag{4}$$


(a) 1D pattern


(b) 2D pattern

Slika 3. Calibration patterns. (a) 1D calibration pattern with three collinear markers, (b) 2D calibration pattern with 4 markers.

Once we estimate the rigid transformations that align the cameras with the reference, we applied these transformations to the point clouds $\mathbf{PC}_i$, $i \in \{2, 3, 4\}$ to align the 3D points from all the Kinect sensors into a single coordinate frame. However, the alignment is not perfect, so we applied a refining step using Iterative Closest Point (ICP) [31] on each aligned point cloud with the reference, to minimize the difference between them. The aligned point clouds will be denoted as $\hat{\mathbf{PC}}_i$, $i \in \{2, 3, 4\}$.

Figure 4(a) shows the point clouds acquired with each camera plotted on the same coordinate frame before alignment. The points clouds from each camera are displayed with different colors. Figure 4(b) shows the point clouds after alignment. In this figure, we discern the structure of the room from the different cameras viewpoint.

2.3 Point Cloud Matching

To calibrate the cameras with the reference Kinect, we need multiple matching 3D points and their respective 2D projections between the reference and each of the Kinect sensors. The goal of this step is to find these matching points that will be used for intrinsic parameters initialization and final calibration.

Once the cameras have been aligned on the same coordinate frame, we should have points that overlap between point clouds. The objective is to find these overlapped points or matching points between the i^{th} camera and the reference camera. Because the point clouds are relatively well aligned, for each camera point cloud, we apply a linear nearest neighbor search with the reference point cloud. Concretely, taking the point cloud of the first Kinect (reference) $\mathbf{PC}_1$ as query points, for each pair of aligned point clouds: $\mathbf{PC}_1$ and $\hat{\mathbf{PC}}_2$; $\mathbf{PC}_1$ and $\hat{\mathbf{PC}}_3$; and $\mathbf{PC}_1$ and $\hat{\mathbf{PC}}_4$, we search for the nearest point inside a radii of R millimeters of each reference query point.

We say that a point p is an *R-near neighbor* of a point q if the distance between p and q is at most R (see Figure 5). The algorithm either returns the nearest R-near neighbor, or concludes that no such point exists, for some fixed parameter R [32]. Figure 6 shows the matching points in red between (a)$\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_2$(black), (b)$\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_3$(blue), and (c)$\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_3$(magenta).

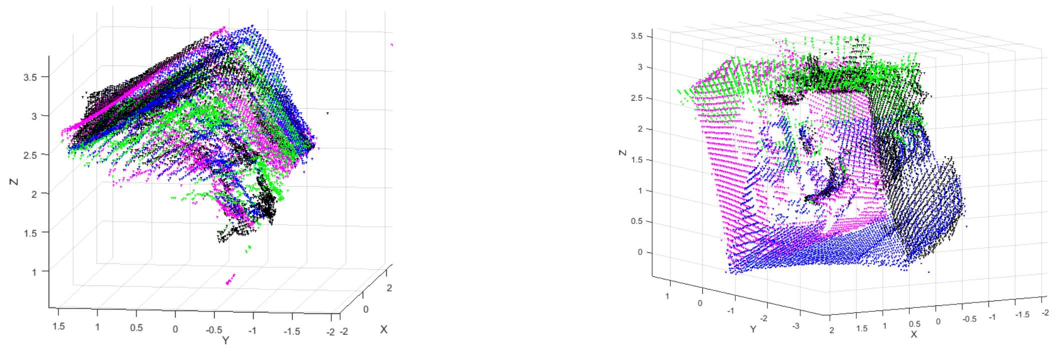
2.4 Intrinsic Parameters Initialization

The matching points between the reference Kinect and the rest (obtained in the point cloud matching step) are the 3D points in the world reference and will be denoted by $\mathbf{P}_{Wi}$ for $i \in \{2, 3, 4\}$, the 2D projections of these points on the image plane are denoted by $\mathbf{u}_i = (u, v)$ and are known from the acquisition step.

In homogeneous coordinates, the mapping between points $\mathbf{P}_W = (x, y, z)$ and their 2D projections $\mathbf{u} = (u, v)$ in the image plane is given by Equation (5).

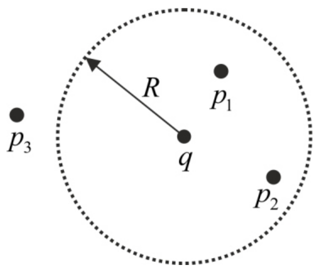
$$\lambda \mathbf{u} = \begin{bmatrix} \mathbf{K} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \mathbf{P}_W \tag{5}$$

where $\mathbf{K}$ is the intrinsic parameters matrix or camera parameters, and $[\mathbf{R}, \mathbf{t}]_{W \to C}$ the extrinsic parameters, $\mathbf{R}$ is a



(a) Original Point Clouds (b) Point Clouds after Pre-calibration

Slika 4. Pre-Calibration. (a) Shows the point clouds obtained from each sensor plotted on the same coordinate axis. (b) Shows the same point clouds after pre-calibration alignment.



Slika 5. R-near neighbor query. The nearest neighbor of the query point q is the point p_1. However, both p_1 and p_2 are R-near neighbors of q.

3x3 rotation matrix that defines the camera orientation and $\mathbf{t}$ is a translation vector that describe the position of the camera in the world. Our goal is to compute the intrinsic parameters $\mathbf{K}$ which contains the focal length (α, β) , a skew factor (γ), and the principal point (u_0, v_0) for fixed extrinsic parameters $[\mathbf{R}, \mathbf{t}]$ obtained with the pre-calibration step. We apply the maximum likelihood method

$$\hat{\mathbf{K}} = \underset{\mathbf{K}}{\text{argmin}} \, |\sum_{j=1}^{J} (\mathbf{u}_j - \text{pinhole}[\mathbf{P}_W, \mathbf{K}, \mathbf{R}, \mathbf{t}])^T \ldots$$
$$(\mathbf{u}_j - \text{pinhole}[\mathbf{P}_W, \mathbf{K}, \mathbf{R}, \mathbf{t}])| \quad (6)$$

Given that Equation (5) is linear with respect to the intrinsic parameters, and can be written as $\mathbf{A}_i\mathbf{h}$ where

$$\mathbf{A}_i = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5) \quad (7)$$

where

$$\mathbf{a}_1 = \begin{bmatrix} \frac{r_{11}x_i + r_{12}y_i + r_{13}z_i + t_x}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \\ 0 \end{bmatrix}$$
$$\mathbf{a}_2 = \begin{bmatrix} \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_x}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \\ 0 \end{bmatrix}$$
$$\mathbf{a}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\mathbf{a}_4 = \begin{bmatrix} 0 \\ \frac{r_{21}x_i + r_{22}y_i + r_{23}z_i + t_y}{r_{31}x_i + r_{32}y_i + r_{33}z_i + t_z} \end{bmatrix}$$
$$\mathbf{a}_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

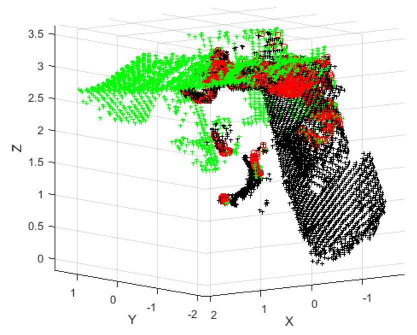
and $\mathbf{h} = [\alpha, \beta, \gamma, u_0, v_o]^T$. Therefore, the problem has the form

$$\mathbf{h} = \underset{\mathbf{h}}{\arg\min} \left[\sum_{i=1}^{J} (\mathbf{A}_i\mathbf{h} - \mathbf{u}_i)^T (\mathbf{A}_i\mathbf{h} - \mathbf{u}_i) \right], \quad (8)$$

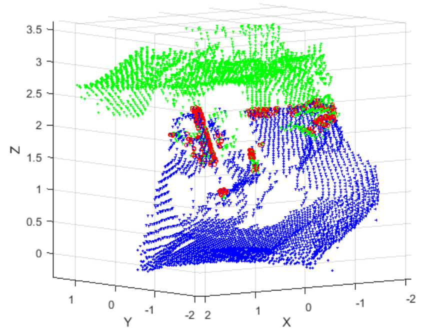
which is a least squares problem that can be solved in closed form.

2.5 Final Calibration

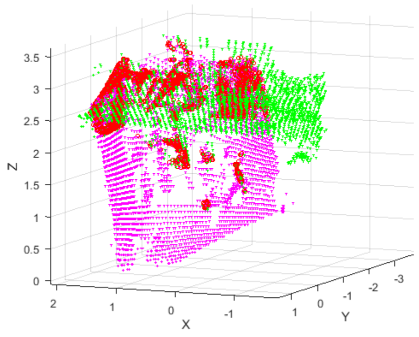
The extrinsic parameters obtained with the *Pre-Calibration* step and the intrinsic parameters obtained with the *Intrinsic Parameters Initialization* may not be optimal. To optimize together these parameters, we apply a non-linear minimization of a cost function f using the Levenberg-Marquard algorithm [33, 34]. To deal with optical distortions, we included radial (Equation 9) and tangential (Equation 10) distortion correction to the model [35,36].



(a) $\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_2$(black)



(b) $\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_3$(blue)



(c) $\mathbf{PC}_1$(green) and $\hat{\mathbf{PC}}_4$(magenta)

Slika 6. Point cloud matching. The matching points are shown in red. (a) Matching points between reference (green) and $\hat{\mathbf{PC}}_2$ *(black). (b) Matching points between reference and* $\hat{\mathbf{PC}}_3$ *(blue). (c) Matching points between reference and* $\hat{\mathbf{PC}}_4$ *(magenta).*

$$\begin{aligned}\delta_u &= u_d(k_1 r_d^2 + k_2 r_d^4 + \ldots + k_{r_0} r_d^{2r_0})\\ \delta_v &= v_d(k_1 r_d^2 + k_2 r_d^4 + \ldots + k_{r_0} r_d^{2r_0}),\end{aligned} \tag{9}$$

$$\begin{aligned}\delta_u &= u_d + \left[2p_1 v_d + p_2(r^2 + 2{u_d}^2)\right]\\ \delta_v &= v_d + \left[p_1(r^2 + 2{v_d}^2 + 2p_2 u_d\right].\end{aligned} \tag{10}$$

where (δ_u,δ_u) are the corrections for geometric lens distortion presented in the distorted image coordinates (u_d,v_d), $r_d^2 = u_d^2 + v_d^2$, with the terms k_i represent the radial distortion parameters, and p_1, p_2 represent the tangential distortion coefficients.

The input parameters of the optimization function are the rotation $\mathbf{R}$ and translation vector $\mathbf{t}$, intrinsic parameters matrix $\mathbf{K}$, radial distortion parameters k_1, k_2, k_3 and tangential distortion parameters p_1 and p_2.

3 EXPERIMENTAL RESULTS

The proposed method has been tested with four Kinect V2 sensors arranged as shown in Figure 2. We calibrated the four sensors using the 1D and 2D pattern with different configurations:

1. *No distortion*: basic model that includes intrinsic parameters (focal length α, skew γ, and principal point (u_0, v_0)) and extrinsic parameters (rotation $\mathbf{R}$ and translation $\mathbf{t}$ with respect to the reference) without distortion parameters.
2. *2 rad*: basic model plus two radial distortion parameters k_1 and k_2.
3. *3 rad*: basic model plus three radial distortion parameters k_1, k_2, and k_3.
4. *2 rad + 2 tan*: basic model plus two radial distortion parameters and two tangential distortion parameters k_1, k_2, p_1, and p_2.
5. *3 rad + 2 tan*: basic model plus three radial distortion parameters and two tangential distortion parameters k_1, k_2, k_3, p_1, and p_2. This will be called *the complete model*.

For the 1D pattern we used three acquisitions, and for the 2D pattern, we used only one. The results obtained between the 1D and 2D pattern are very similar, and we concluded that the only benefit of the 2D pattern is a minimum number of acquisitions with the trade-off of a complex build. For the rest of this section, we reported results using the 1D pattern.

Figure 7 shows the evolution of the root mean square re-projection error (RMSE) of the different configurations during the iterative non-linear optimization (described in Section 2.5) of the eight cameras: four depth cameras and four color cameras of the four Kinect V2 sensors. In all the cases, we see a reduction of the RMSE with the inclusion of the distortion parameters. This reduction is more substantial for the depth cameras (Figures 7(a), (c), (e), and (g)) than for the color cameras (Figures 7(b), (d), (f), and (h)). For the depth camera 1 (Figure 7(a)), we see a small improvement of 7.14% with the inclusion of the third radial distortion parameter k_3 improving the RMSE from 0.29 for the (*2 rad + 2 tan*) model to 0.27 for the (*3 rad + 2 tan*) model. Also, we discern a nearly imperceptible improvement of 0.36% with the inclusion of the tangential distortion parameters p_1 and p_1 improving the RMSE from 0.271 for the (*3 rad*) model to 0.270 for the (*3 rad + 2 tan*) model.

On the other hand, for the color camera 1 (Figure 7(b)), there is no improvement with the inclusion of the third radial distortion parameter with a RMSE of 0.479 for the (*2 rad + 2 tan*) model and a RMSE of 0.479 for the (*3 rad + 2 tan*) model. This may imply that the depth camera has more distortion than the color camera. The rest of the cameras follow a similar pattern: a substantial improvement in the RMSE with the inclusion of the first two radial distortion parameters k_1 and k_2, and a small improvement in the RMSE with the inclusion of the third radial distortion parameter and the tangential parameters.

Table 1 display the values of the calibration results along with the RMSE of each configuration for the depth and color cameras of the first Kinect V2. For the depth camera, the configuration with the complete distortion parameters (*3 rad + 2 tan*) achieved the best result with an RMSE of 0.27. For the color camera, the third radial distortion parameter k_3 did not affected the result, achieving the same RMSE of 0.479 on the last two configurations: (*2 rad + 2 tan*) and (*3 rad + 2 tan*).

The calibration results for the rest of the cameras yielded the best results with the complete model and a point cloud matching distance of 1 mm. Table 2 display the calibration results for the depth and color cameras of the second Kinect V2 sensor. The best RMSE was 0.165 and 0.369 respectively. Table 3 display the same information for third Kinect V2 sensor with a RMSE of 0.183 for the depth camera and 0.331 for the color camera. Finally, table 4 display the calibration results for the fourth Kinect V2 sensor. On the depth camera, the best RMSE was 0.28 and 0.198 for the color camera.

3.1 3D Reconstruction

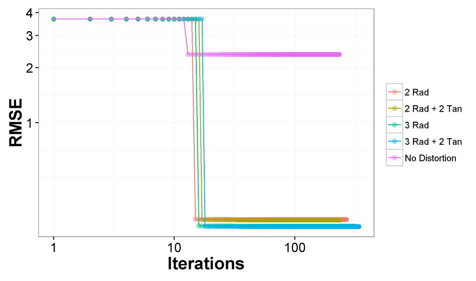
To evaluate the performance of our calibration method, we performed point cloud fusion and 3D reconstruction of two objects: a paperboard box and a toy car. For this, we placed each object inside the field of view of the four cameras and acquired a depth and color frame from each Kinect V2 sensor. Then, using the calibration results with the complete distortion parameters, we un-distorted the depth and color images and obtain the $[x, y, z]$ coordinates of the color markers using equation 5 with $z = depth$. Figures 8(a) and 8(b) show the mean error (with ten acquisitions of the 1D object) in mm between the reconstructed and true wand lengths for the depth and color cameras respectively. These figures, also show the error at increasing target distances (0.5m, 1m, and 2m). As it is known [37], the Kinect accuracy is not very good and degrades with distance. However, our calibration method yields better accuracy than the Kinect's built-in mapping.

To evaluate our calibration results qualitatively, we mapped the $[x, y, z]$ points onto the color frame using the intrinsic and extrinsic parameters of the color camera to obtain the corresponding color of each 3D point. Finally, by merging the colored 3D data from the four Kinect sensors we got a 3D fused point cloud which then we reconstructed into a meshed object using MeshLab. Figure 9 shows the 3D reconstruction of a paperboard box and a toy car.

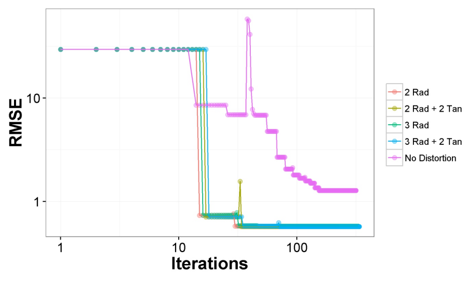
3.2 Comparison with other methods

The calibration methods from Alexiadis *et al.* [2] and Liu *et al.* [23] are similar to our approach for depth camera calibration in the sense that they also used a wand with markers; however, their methodologies are different. Moreover, those methods used a planar object to calibrate the color camera. In contrast, our method uses the same 1D object to calibrate the depth and color cameras.

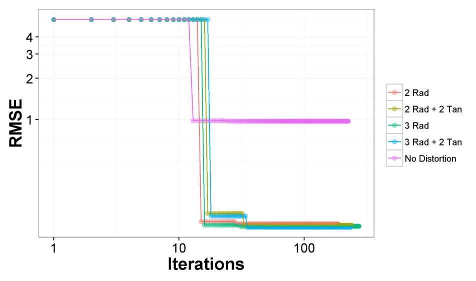
Alexiadis *et al.* [2] reported a mean reprojection error of 0.78 pixels and Liu *et al.* [23] reported a reprojection error of 0.6 pixels on the color camera with 120 calibration points and a metric error < 5 mm with 40 calibration points. Our calibration method yielded minimum reprojection error of 0.16 pixels and 0.33 pixels on the depth and color cameras respectively. On the metric side, we found that the error is proportional to the target distance: at 0.5m we obtained a metric error of 4.6mm and 5.8mm for depth and color cameras, and with the target at 2m from the sensor, we obtained a metric error of 6.0mm and 8.2mm for the depth and color cameras respectively.



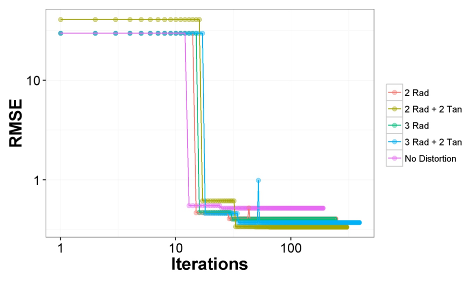
(a) Depth Camera 1



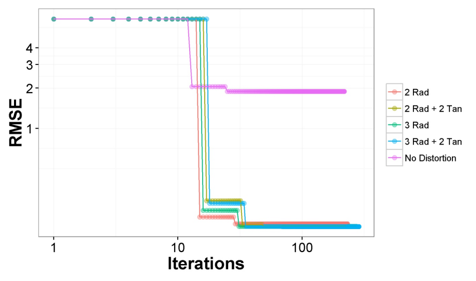
(b) Color Camera 1



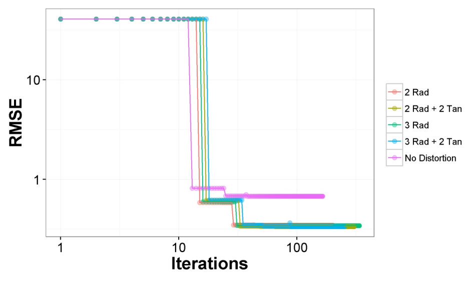
(c) Depth Camera 2



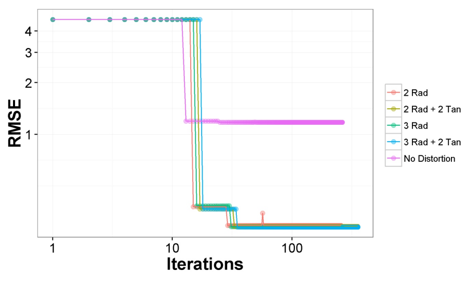
(d) Color Camera 2



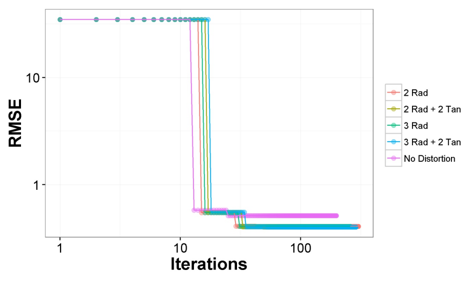
(e) Depth Camera 3



(f) Color Camera 3



(g) Depth Camera 4



(h) Color Camera 4

Slika 7. Non-linear optimization Root Mean Square Error (RMSE) of each camera for the different configurations: No distortion*,* 2 Rad*,* 2 Rad + 2 Tan*,* 3 Rad*, and* 3 Rad + 2 Tan.

Tablica 1. Camera 1

Model	α	γ	u_0	v_0	k_1	k_2	k_3	p_1	p_2	*RMSE*
Depth Camera										
No distortion	391.54	0.33	259.57	215.25						**2.355**
2 rad	367.30	0.11	257.66	213.53	5.12e-02	-1.55e-01				**0.294**
3 rad	366.11	0.13	257.65	213.47	8.54e-02	-2.51e-01	7.59e-02			**0.271**
2 rad + 2 tan	367.20	0.03	257.70	213.26	5.12e-02	-1.55e-01		2.78e-04	-1.61e-04	**0.294**
3 rad + 2 tan	365.85	-0.15	257.69	213.42	8.54e-02	-2.50e-01	7.57e-02	9.39e-05	-2.92e-04	**0.270**
Color Camera										
No distortion	1054.01	-13.07	961.09	551.89						**1.277**
2 rad	1055.73	-12.51	961.02	553.64	1.93e-02	-4.66e-03				**0.484**
3 rad	1055.62	-12.51	961.03	553.63	2.05e-02	-7.93e-03	2.57e-03			**0.484**
2 rad + 2 tan	1061.88	-1.43	959.02	548.15	1.96e-02	-4.71e-03		-2.98e-05	5.87e-04	**0.479**
3 rad + 2 tan	1061.79	-1.42	959.02	548.15	2.06e-02	-7.48e-03	2.22e-03	-2.85e-05	5.85e-04	**0.479**

Tablica 2. Camera 2

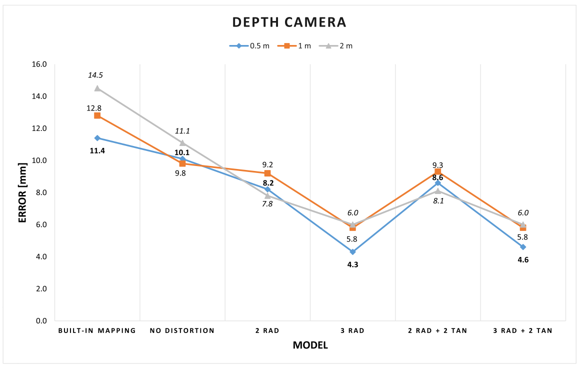
Model	α	γ	u_0	v_0	k_1	k_2	k_3	p_1	p_2	*RMSE*
Depth Camera										
No distort	376.37	-0.01	265.94	218.74						**0.970**
2 rad	364.15	-0.04	258.63	217.04	7.62e-02	-1.86e-01				**0.172**
3 rad	363.55	-0.02	258.38	216.79	9.76e-02	-2.66e-01	8.18e-02			**0.166**
2 rad + 2 tan	364.57	-0.00	257.41	216.64	7.52e-02	-1.80e-01		-3.82e-05	-1.38e-03	**0.170**
3 rad + 2 tan	363.88	0.01	257.63	216.40	9.48e-02	-2.54e-01	7.34e-02	-1.46e-04	-8.65e-04	**0.165**
Color Camera										
No distort	1058.26	0.11	956.99	550.95						**0.500**
2 rad	1061.44	0.52	960.90	552.00	9.81e-03	-7.11e-04				**0.396**
3 rad	1060.89	0.56	960.66	551.86	1.60e-02	-2.26e-02	2.08e-02			**0.394**
2 rad + 2 tan	1063.56	0.34	955.76	555.88	1.27e-02	-4.32e-04		1.81e-03	-2.35e-03	**0.371**
3 rad + 2 tan	1062.91	0.35	955.88	556.11	1.86e-02	-2.11e-02	1.93e-02	1.91e-03	-2.21e-03	**0.369**

Tablica 3. Camera 3

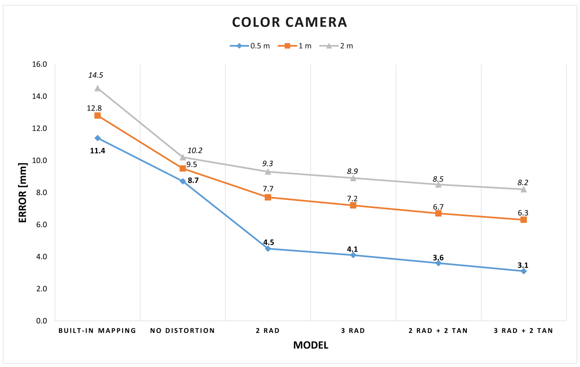
Model	α	γ	u_0	v_0	k_1	k_2	k_3	p_1	p_2	*RMSE*
Depth Camera										
No distort	420.14	11.60	271.60	200.88						**1.863**
2 rad	367.72	0.01	258.87	216.98	5.44e-02	-1.57e-01				**0.191**
3 rad	365.66	-0.11	257.94	215.76	8.04e-02	-2.26e-01	5.32e-02			**0.185**
2 rad + 2 tan	366.64	0.23	259.78	216.72	5.81e-02	-1.59e-01		6.75e-04	6.74e-04	**0.184**
3 rad + 2 tan	365.94	0.12	258.49	216.15	8.01e-02	-2.19e-01	4.73e-02	7.44e-04	1.40e-04	**0.183**
Color Camera										
No distort	1057.07	-1.67	963.37	563.63						**0.672**
2 rad	1064.58	1.15	966.90	558.14	9.76e-03	-1.52e-03				**0.332**
3 rad	1064.20	1.19	967.05	557.92	1.30e-02	-9.22e-03	5.49e-03			**0.332**
2 rad + 2 tan	1064.77	1.23	967.12	558.36	9.96e-03	-1.51e-03		1.17e-04	5.34e-05	**0.332**
3 rad + 2 tan	1064.46	1.54	968.80	558.07	1.51e-02	-1.34e-02	8.61e-03	2.06e-04	4.81e-04	**0.331**

Tablica 4. Camera 4

Model	α	γ	u_0	v_0	k_1	k_2	k_3	p_1	p_2	*RMSE*
Depth Camera										
No distort	384.41	-1.14	244.03	218.42						**1.130**
2 rad	364.80	0.02	256.99	215.25	5.13e-02	-1.52e-01				**0.294**
3 rad	364.08	-0.02	257.17	215.15	6.91e-02	-2.04e-01	4.18e-02			**0.290**
2 rad + 2 tan	364.63	-0.15	256.23	213.89	5.36e-02	-1.58e-01		-1.12e-03	-7.47e-04	**0.291**
3 rad + 2 tan	363.96	-0.11	256.26	214.32	6.77e-02	-2.02e-01	3.69e-02	-7.54e-04	-9.94e-04	**0.288**
Color Camera										
No distort	1058.84	0.66	970.09	536.31						**0.512**
2 rad	1061.85	0.04	965.08	537.48	1.01e-02	-1.01e-04				**0.401**
3 rad	1061.64	0.04	965.16	537.43	1.22e-02	-6.80e-03	5.86e-03			**0.400**
2 rad + 2 tan	1061.90	0.01	965.42	537.07	1.00e-02	5.68e-05		-1.23e-04	1.37e-04	**0.400**
3 rad + 2 tan	1061.70	0.02	965.43	537.08	1.19e-02	-5.98e-03	5.26e-03	-1.08e-04	1.06e-04	**0.400**

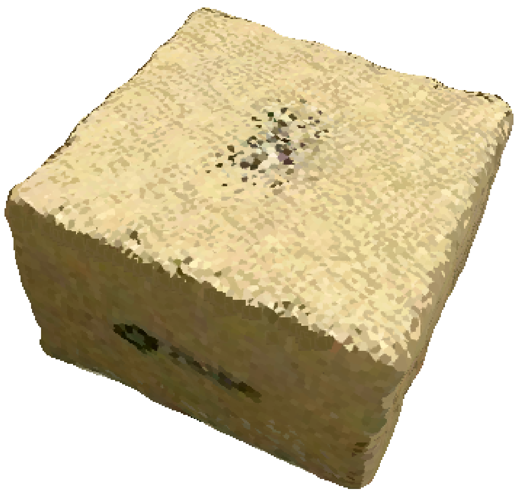


(a) Depth camera metric error

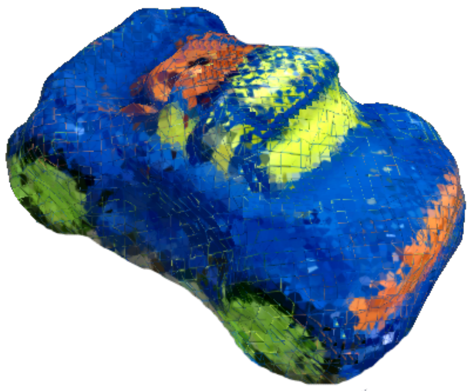


(b) Color camera metric error

Slika 8. Mean error (mm) between reconstructed and true wand lengths for the different calibration models and target distances.



(a) Paperboard box



(b) Toy car

Slika 9. 3D Reconstruction of objects using our calibration method.

4 CONCLUSION

In this paper, we have developed a method to calibrate multiple Kinect V2 sensors easily. The method requires at least three acquisitions of a 1D object from each camera or a single acquisition from a 2D object, and a point cloud from each Kinect V2 obtained with the built-in coordinate mapping capabilities of Kinect V2. The proposed method consists of five steps: (1) Image acquisition, obtains corresponding 3D points from the 1D object on the multiple cameras; (2) Pre-calibration, finds an estimate of the rigid transformation between the Kinect sensors with respect to a reference; (3) Point cloud matching, finds matching points between the reference and the rest of the Kinect sensors; (4) Intrinsic parameters initialization, calculates a closed-form solution of the intrinsic parameters of each camera; and (5) Final calibration, performs a nonlinear refinement of the full model including intrinsic and extrinsic parameters, radial and tangential lens distortion.

Our calibration results yielded a root mean square reprojection of 0.2 pixels on the depth cameras and 0.4 on the color cameras. We implemented the full method in MATLAB using the *Kin2 Toolbox* [28].

Compared with other techniques for multiple camera calibrations, the proposed method do not need a minimum number of cameras nor multiple image acquisitions of a planar target. Moreover, our calibration results improved significantly the accuracy of the Kinect V2 built-in reconstruction at different target distances.

ACKNOWLEDGMENT

This work was supported by CONACYT through Ph.D. scholarships. We also want to thank CIDESI for providing the facilities and support during the development of this project.

Literatura

[1] S. Alexiadis, G. Kordelas, K. C. Apostolakis, J. D. Agapito, J. Vegas, E. Izquierdo, and P. Daras, "Reconstruction for 3d immersive virtual environments," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th International Workshop on*, pp. 1–4, IEEE, 2012.

[2] D. S. Alexiadis, D. Zarpalas, and P. Daras, "Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras," *Multimedia, IEEE Transactions on*, vol. 15, no. 2, pp. 339–358, 2013.

[3] W.-C. Chang and W. C. Chang, "Real-time 3d rendering based on multiple cameras and point cloud," in *Ubi-Media Computing and Workshops (UMEDIA), 2014 7th International Conference on*, pp. 121–126, IEEE, 2014.

[4] M. Luber, L. Spinello, and K. O. Arras, "People tracking in rgb-d data with on-line boosted target models," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 3844–3849, IEEE, 2011.

[5] J. Han, E. J. Pauwels, P. M. De Zeeuw, and P. H. De With, "Employing a rgb-d sensor for real-time tracking of humans across multiple re-entries in a smart environment," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 255–263, 2012.

[6] E. J. Almazan and G. A. Jones, "Tracking people across multiple non-overlapping rgb-d sensors," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pp. 831–837, IEEE, 2013.

[7] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2389–2395, IEEE, 2012.

[8] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, and M. A. Magnor, "Markerless motion capture using multiple color-depth sensors.," in *VMV*, pp. 317–324, 2011.

[9] S. Asteriadis, A. Chatzitofis, D. Zarpalas, D. S. Alexiadis, and P. Daras, "Estimating human motion from multiple kinect sensors," in *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, MIRAGE '13, (New York, NY, USA), pp. 3:1–3:6, ACM, 2013.

[10] F. Faion, S. Friedberger, A. Zea, and U. D. Hanebeck, "Intelligent sensor-scheduling for multi-kinect-tracking," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3993–3999, IEEE, 2012.

[11] A. D. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 273–282, ACM, 2010.

[12] A. Canessa, M. Chessa, A. Gibaldi, S. P. Sabatini, and F. Solari, "Calibrated depth and color cameras for accurate 3d interaction in a stereoscopic augmented reality environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 227–237, 2014.

[13] B. Jones, R. Sodhi, M. Murdock, R. Mehra, H. Benko, A. Wilson, E. Ofek, B. MacIntyre, N. Raghuvanshi, and L. Shapira, "Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units," in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pp. 637–644, ACM, 2014.

[14] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 137–146, IEEE, 2011.

[15] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs, "Enhanced personal autostereoscopic telepresence system using commodity depth cameras," *Computers & Graphics*, vol. 36, no. 7, pp. 791–807, 2012.

[16] A. Maimone and H. Fuchs, "Real-time volumetric 3d capture of room-sized scenes for telepresence," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pp. 1–4, IEEE, 2012.

[17] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint.* MIT press, 1993.

[18] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[19] Z. Zhang, "Camera calibration with one-dimensional objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 7, pp. 892–899, 2004.

[20] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," *PRESENCE: teleoperators and virtual environments*, vol. 14, no. 4, pp. 407–422, 2005.

[21] N. A. Borghese and P. Cerveri, "Calibrating a video camera pair with a rigid bar," *Pattern Recognition*, vol. 33, no. 1, pp. 81–95, 2000.

[22] T. Pribanić, P. Sturm, and M. Cifrek, "Calibration of 3d kinematic systems using orthogonality constraints," *Machine Vision and Applications*, vol. 18, no. 6, pp. 367–381, 2007.

[23] W. Liu, Y. Fan, Z. Zhong, and T. Lei, "A new method for calibrating depth and color camera pair based on kinect," in *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pp. 212–217, IEEE, 2012.

[24] E. Auvinet, J. Meunier, and F. Multon, "Multiple depth cameras calibration and body volume reconstruction for gait analysis," in *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pp. 478–483, IEEE, 2012.

[25] R. Macknojia, A. Chávez-Aragón, P. Payeur, and R. Laganiere, "Calibration of a network of kinect sensors for robotic inspection over a large workspace," in *Robot Vision (WORV), 2013 IEEE Workshop on*, pp. 184–190, IEEE, 2013.

[26] R. Avetisyan, M. Willert, S. Ohl, and O. Staadt, *Calibration of Depth Camera Arrays*. PhD thesis, Master thesis), Dept. of computer science, University of Rostock, Germany, 2013.

[27] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[28] J. R. Terven and D. M. Córdova-Esparza, "Kin2. a kinect 2 toolbox for matlab," *Science of Computer Programming*, vol. 130, pp. 97–106, 2016.

[29] Microsoft, "Coordinate Mapping." `http://tinyurl.com/gvblebj`. Accessed: 2016-10-01.

[30] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.

[31] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International journal of computer vision*, vol. 13, no. 2, pp. 119–152, 1994.

[32] A. Andoni, *Nearest neighbor search: the old, the new, and the impossible*. PhD thesis, Massachusetts Institute of Technology, 2009.

[33] K. Levenberg, "A method for the solution of certain non–linear problems in least squares," 1944.

[34] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[35] D. Brown, "Lens distortion for close-range photogrammetry," *Photometric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

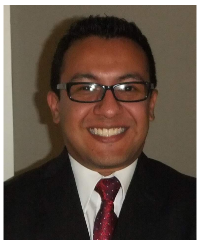
[36] C. B. Duane, "Close-range camera calibration," *Photogrammetric engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[37] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, 2015.

Diana-Margarita Córdova-Esparza received a BS degree in Electronic Engineering and an MS degree in Digital Signal Processing from the Autonomous University of Zacatecas. She holds a Ph.D. in Computer Vision at National Polytechnic Institute in México. She is currently a post-doctoral researcher at CIDESI México. Her research interests include Computer Vision and Signal, Image and Video Processing.

Juan R. Terven received a BS in Electronic Engineering at Mazatlan Institute of Technology (ITM), an MS degree in Computer Science at Autonomous University of Sinaloa and a Ph.D. at National Polytechnic Institute in Mexico. He works as a professor at ITM. He has been a visiting scholar at MIT and a research intern at Microsoft. His research interests include embedded systems, computer vision, and machine learning. He is a member of IEEE.

Hugo Jimenez-Hernández is a professor-researcher at CIDESI. He received his BS in computer science at Querétaro Institute of Technology (ITQ) México, MS degrees in computer Science and Ph.D. in Computer Vision at National Polytechnic Institute in Mexico. He is the author of more than 15 journal papers. His current research interests include Computer Vision, Visual Inference, and Computer Theory.



Alberto Vázquez-Cervantes is a Ph.D. student at CIDESI. He received his BS in Electronic Engineering, at Queretaro Institute of Technology in México, MS degree in Science and Technology in Mechatronics at CIDESI. His current research interests include Computer Vision, Embedded system, and Artificial Intelligence.

Ana M. Herrera-Navarro received a BS degree in Computer Engineering, an MS degree in Computer Science, and Ph.D. from Autonomous University of Querétaro (México). She is currently a professor at the same institution. Her research interests include morphological image processing and computer vision.

Alfonzo Ramírez-Pedraza received a BS Degree in computer science at Autonomous University of Querétaro, and an MS degree at the National Polytechnic Institute. He is a Ph.D. candidate at the National Polytechnic Institute, Mexico. His research interests are Image analysis, 3D segmentation, 3D environment recognition, computer vision and pattern recognition.

AUTHORS' ADDRESSES

Diana-Margarita Córdova-Esparza, Ph.D.
Prof. Hugo Jimenez-Hernández, Ph.D.
Alberto Vázquez-Cervantes, M.Sc.
Centro de Ingenieria y Desarrollo Industrial,
Av. Playa Pie de la Cuesta 702, 76130, Querétaro, México
email: diana_mce@hotmail.com, hugojh@gmail.com, albertovcervantes@gmail.com

Prof. Juan R. Terven, Ph.D.
Alfonzo Ramıirez-Pedraza, M.Sc., CICATA,
Cerro Blanco 141, 76090, Querétaro, México
email: jrterven@hotmail.com, ponchorp.1985@gmail.com

Prof. Ana M. Herrera-Navarro, Ph.D,
Facultad de Informática,
Autonomous University of Querétaro,
Av. de las Ciencias S/N, 75230, Querétaro, México
email: anaherreranavarro@gmail.com

Received: 2016-04-15
Accepted: 2016-12-07