INTERACTIVE REGISTRATION METHOD FOR 3D DATA FUSION

Arantxa Casanova, Alba Pujol-Miró, Javier Ruiz-Hidalgo, Josep R. Casas

Image Processing Group
Universitat Politècnica de Catalunya – BarcelonaTech, Spain
ar.casanova.8@gmail.com, alba.pujol@upc.edu, j.ruiz@upc.edu, josep.ramon.casas@upc.edu

ABSTRACT

Commercial depth sensors represent an opportunity for automation of certain 3D production and analysis tasks. One way to overcome some of their inherent limitations is by capturing the same scene with several depth sensors and merging their data, i.e. by performing 3D data fusion, which requires the registration of point clouds from different sensors. We propose a new interactive, fast and user-friendly method for depth sensor registration. We replace the traditional checkerboard pattern used to extract key points in the scene by a finger detector. This provides a main advantage: the method is easier to use and does not require external objects, while the elapsed time and the registration error are similar to those obtained through the classical method. We test the proposed approach with an interactive hand tracking application, improved to use more than a single sensor, and we show the increase in detection area by more than 70%.

Index Terms— 3D Data Fusion, Point Cloud, Calibration, Registration.

1. INTRODUCTION

Since the release of the affordable Kinect depth sensor, 3D data processing has become a strong research area in computer vision, and an attractive tool in the production of special effects. The acquired 3D data is exploited for multimedia computing applications such as gaming, augmented reality, scene segmentation or human pose and body tracking.

Commercial depth sensors, such as Kinect [15], capture both image and depth data. Photometric information comes out from an RGB camera, while geometry (or range data) is measured by active triangulation, using an infrared stereo pair consisting of an emitter and receiver. Depth measurements result in a 16-bit image, where the value of each pixel represents the distance of that pixel to the camera. By registering both the RGB and depth data, the scene can be represented as a set of 3D points, namely, a point cloud.

Thanks to the IR triangulation technology, pose detectors exploiting depth data are more robust than those using solely image data, because scene illumination is not an important factor any longer. The main problem of using a single depth sensor is the limited viewing angle. If we want to enlarge the captured area, several sensors may be placed side by side in order to capture the scene or object.

The fusion of 3D data could serve different purposes: increasing accuracy and resolution, i.e. precision and number of points in overlapped views, or enlarging the overall detection area of the devices, i.e. range and viewing angle in stitched views. This paper focuses and validates a 3D data fusion strategy for the latter purpose.

In the simplest case, 3D data fusion means that two simultaneous captures have to be merged and aligned (registered) in the same coordinate system. The registration problem is solved in two steps: matching some keypoints in both captures and estimating the rigid transformation that translates and rotates the point cloud captured by a given sensor to the chosen reference system. The main problem is found in the matching step, where the keypoints to match may be identified as false correspondences. We propose an interactive, user-friendly and fast method to identify keypoints and perform the matching using a finger detector.

In Figure 1, the registration workflow is depicted. Raw image data from the sensors is represented as point clouds (Section 3). Then, the proposed registration method performs finger detection to match keypoints and align the point clouds (Section 4). Finally, Iterative Closest Point (ICP) [2] is applied to refine the alignment in the fused point clouds (Section 5). Fused 3D data resulting from the registration of two captures is shown to seamlessly improve the detection area in the experiments with a hand tracking application (Section 7).

2. RELATED WORK

The classic registration method for both image and depth sensors is based on Zhang's work [14]. The process is divided into two steps: first, it captures different angles of a known geometrical pattern; in our case, a checkerboard. The size and number of squares of the pattern should be known to obtain the intrinsics matrix. Then, once the instrinsic values are

This research was supported by the Spanish Ministry of Science and Innovation in the framework of the BIGGRAPH project –TEC2013-43935-R– funded by the Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF)

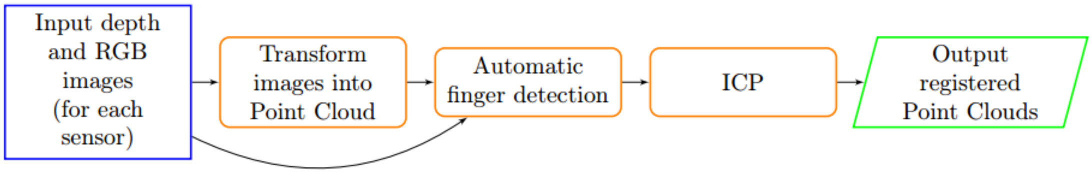

Input depth and RGB images (for each sensor)
Transform images into Point Cloud
Automatic finger detection
ICP
Output registered Point Clouds

Fig. 1. Registration workflow

known, the checkerboard is captured in a still position in order to change the reference system from the camera to a world reference frame. When the reference systems of two sensors have been changed to the same world coordinates, an effective fusion of the point clouds from both sensors can be achieved.

Existing work in multi-depth sensor registration for different applications [13, 3] usually follows the guidelines proposed by Zhang. In the case of RGB cameras, some effortless registration methods are based on the detection of a laser pointer [4, 12], but the application to IR depth sensors is not straightforward. Other registration methods rely on the captured scene and not on physical patterns, by extracting specific keypoints from 2D images and then matching them. In some cases, even by leveraging the depth information from 3D image reconstruction [6].

For point cloud data captured from depth sensors, ICP [2] is widely used as an iterative method to minimize the distance between two point clouds. ICP may use the whole cloud or just keypoint correspondences found in both clouds. Several ICP modifications in any of the ICP steps may result in a more robust algorithm [8, 9].

3. BASELINE REGISTRATION METHOD

Zhang's method [14] has been used as a baseline to evaluate the strategy proposed in this paper. We extract both RGB and infrared sensor parameters, both intrinsics and extrinsics, in order to perform scene registration.

3.1. Intrinsic and Extrinsic Parameters

For the baseline registration, we capture a checkerboard pattern from different angles (Figure 2). The detection of checkerboard corners and the known size of the pattern allows Zhang's algorithm to retrieve the intrinsics matrix and distortion coefficients. This step is done with RGB and IR sensors and allows the transformation of the images into point clouds.

3.2. Classic Registration from Extrinsics

To extract the extrinsic parameters that relate camera and world coordinates, only a single infrared capture of the checkerboard is needed. Once these parameters are extracted for two kinect devices, both point clouds are transformed (registered) to world coordinates and, therefore, fused.

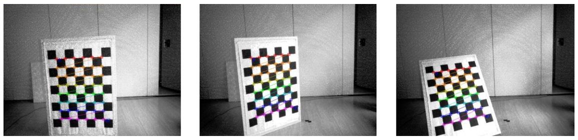


Fig. 2. Examples of different checkerboard captures to extract sensor parameters

Figure 3 illustrates an example of registration. The left image presents the point cloud obtained from sensor 1; the center image, the point cloud from sensor 2; the right image shows the fusion of both point clouds performed by the registration with the extracted extrinsics.

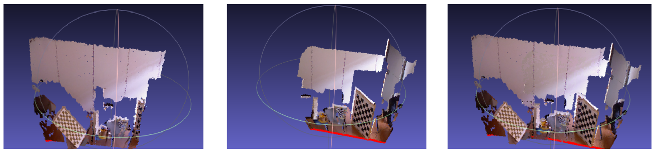


Fig. 3. Scene fusion

4. AUTOMATIC FINGER DETECTION

We propose to replace the extrinsic parameters extraction process and the need of a checkerboard with a detection method in which a human finger draws a path in front of the sensors. In this case, instead of considering a checkerboard corner as the world origin for both captures, the reference system of one device is transformed into the reference system of the other device, estimating a transformation between specific detected keypoints in both captures.

The process consists in identifying a finger in the captured depth data from both depth sensors. The pixel where the finger has been detected is converted to camera coordinates, using intrinsic parameters (depth as the third coordinate). The sequence of 3D points where the fingers have been detected are stored in a point cloud for each sensor. Then, a transformation between clouds is estimated using ICP. Applying this transformation allows the alignment (registration) of the two point clouds.

The ICP algorithm is applied twice, as the initial point clouds are too far apart to achieve satisfying convergence with a single application. First, ICP is applied without a distance threshold between the correspondences, for an initial raw alignment of the detected finger positions in the point clouds. Then, ICP is applied again as a refinement, with a correspondence threshold of 5 cm. This limits the possible correspondences by distance. Note that the captures from the two sensors are not synchronized and this adds uncertainty in the matching performed by ICP when using the finger detector. However, as the hand motion is rather slow, shift in finger position due to lack of synchronization is in the order of the sensor spatial accuracy. Moreover, as the trajectory to be matched from the two sensors has a large number of points, the ICP estimator will average their positions compensating for such imprecision at the individual points.

Finger detections in both sensors are both manually marked and automatically detected, so that we can later evaluate the misalignment introduced by the automatic finger detector.

4.1. Manual Finger Detection

As a first step, the coordinates of the finger tip in the depth maps are manually annotated along the recorded video sequences. A depth frame from both devices with the same timestamp is shown in Figure 4. The point highlighted in red in both images will be treated as a correspondence.

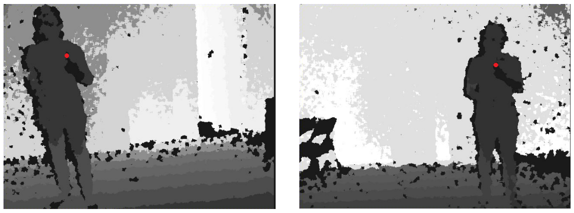


Fig. 4. Manual finger detection in depth maps of two devices

The user is requested to draw a cube with the finger, and 8 points corresponding to the vertices are identified in the image captured by each device, as shown in Figure 5. The orange cloud contains the finger positions in sensor 1, and the red cloud, those in sensor 2. The green points are those from the orange cloud after applying the rigid transformation obtained by ICP, and should coincide with the red points, because both are approximately the same points from the actual 3D scene captured by the two different sensors (consider the noise due to positional sensor accuracy and lack of synchronization). In this step, ICP is used to estimate the transformation between detected point clouds.

For the manually annotated fingertips, the correspondences are already established when constructing the clouds. As a first approach, Singular Value Decomposition (SVD) [1] was used to estimate the rigid transformation. Contrary to SVD, ICP performs trajectory matching (finds correspondences by itself). After comparing the error in the registration between the transformations estimated using SVD and ICP, we concluded that there was no significant difference. Considering that correspondences are not labeled in automatic detection mode, ICP will be used for trajectory matching. Therefore, in order to have a more consistent comparison between manual and automatic detection, we use ICP in both cases.

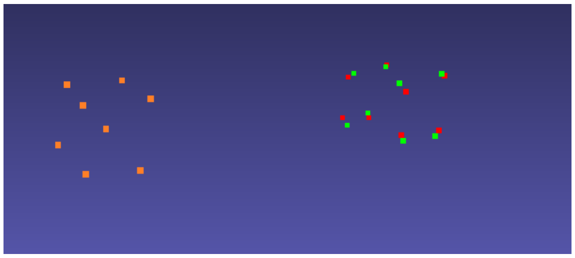


Fig. 5. Finger detection drawing a cube: finger positions from sensor 1 (in orange), and those from sensor 2 (in red) coincide when the first undergo the rigid transformation estimated by ICP (in green)

The rigid transformation estimated by ICP will be later applied to any cloud captured by sensor 1, to keep these points registered to those captured by sensor 2 (provided that sensors do not move). It is important to note that the transformation has to be recalculated if the relative position of the devices changes, as it happens with the classical method.

4.2. Automatic Finger Detection

Manual detection is tedious and consumes a significant amount of time. An automatic finger detector is a better option to register the point clouds without manual intervention.

The detector used in this work locates the fingertips using an ORD feature [11] and exploits a classifier trained with a random forest [7]. In order to train the automatic finger detector, we need a training dataset. Details about the database are explained in Section 6.

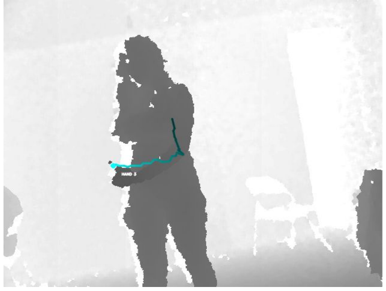


Fig. 6. Detector tracking a human finger

As seen in Figure 6, the detector retrieves the pixel in the depth image where the finger was found, along with a queue of previously detected positions. This visualizes the tracked trajectory for the finger.

The finger positions are detected in all frames, contrarily to the manual case where just the desired vertices of the cube where labeled. The detections in both sensors are separately stored in two different point clouds. As explained in the previous section, the number of points may differ in both clouds, correspondences between the points in the two clouds are not labeled, and it is imperative to use ICP (not SVD) to estimate the transformation.

For the automatic finger detection, different paths have been drawn to test the importance of several finger traces in front of the sensors: a simple square, two squares at different depth distances, and a spiral trajectory (Figure 7). Our experiments show that 5 seconds of captured video provide enough points to estimate the transformation using ICP.

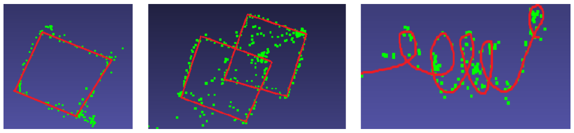


Fig. 7. Different finger paths tested

Although detections are noisy due to the detector's technology, lack of synchronization and the inherent noise in the depth channel, the number of samples in the point clouds and the ICP algorithm itself mitigate the error in the transformation estimation.

5. ITERATIVE CLOSEST POINT

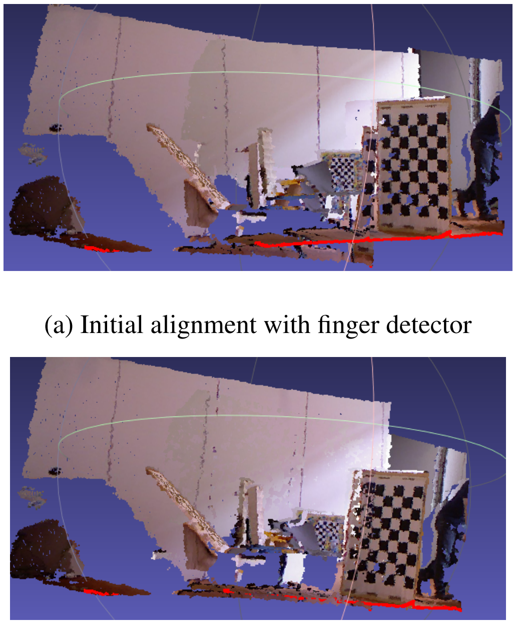
ICP is used again as a refinement method to obtain a better alignment of the whole point cloud. From a good initial alignment performed by the registration with the finger detector explained above, ICP takes less time to converge, and relies on all the points in the scene to improve registration.

6. DATASET

Two different datasets have been used; one for training the finger detector and the other for testing and extracting results. Both datasets are composed of frontal captures.

The training dataset contains 20 minutes of depth video, sectioned into labeled frames. Recorded data shows a person drawing paths with the finger pointing at the ceiling with different orientations, sensor position and different persons. The captures are made by only one sensor; changing the orientation and sensor position tells the detector to find the finger in the two different oriented sensors.

Considering that the sensors face the person, pointing the finger at the ceiling gives the detector more depth discrimination. If the finger was pointing at the sensors, there would be a low depth variation between the finger and the hand. On the contrary, pointing at the ceiling, the surrounding pixels of the finger would be the background or the human body, granting more discrimination in depth. Thus, the detector is more robust and discriminating for the finger detection task.



(a) Initial alignment with finger detector

(b) Final alignment after ICP with the entire cloud

Fig. 8. Example of ICP as a refinement method.

The dataset to test and evaluate the registration is composed by frontal captures of two different sensor configurations, as shown in Figure 9.

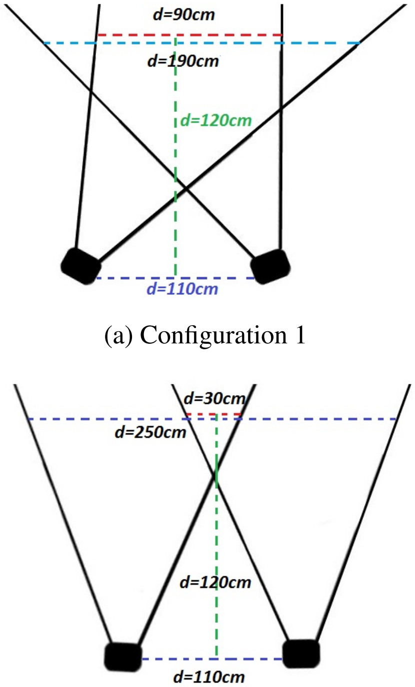
- Configuration 1 - more overlap than extended viewing. Used to increase precision and number of points. Sensors separated 110 cm, 90 cm overlap at 120 cm.
- Configuration 2 - combined viewing area of both sensors larger than configuration 1. Small overlap. More oriented to increase detection area. Sensors separated 110 cm, 30 cm overlap at 120 cm.

For each sensor configuration, three different scene types have been captured: scene full of geometrical objects, scene with two persons, scene with the combination of geometrical objects and persons.

All objects and persons in the scenes are captured in a range of 0.5 to 2m.

7. EXPERIMENTAL RESULTS

In this section, We first explain the evaluation criteria for the quantitative evaluation, along with the results and the comparison with the traditional registration method. Secondly, we describe and discuss the qualitative results obtained with the improvement of a specific application.



(a) Configuration 1

(b) Configuration 2

Fig. 9. Different sensor configurations

7.1. Evaluation criteria

To evaluate the registration results, all outputs of the registration algorithms are compared to a ground truth obtained with manual alignment: a registered pair of point clouds where corresponding points in both clouds are picked manually in pairs and the transformation is estimated. We use a qualitative ground truth because the points in a cloud are not the same than the captured points in the second registered cloud.

The error between the ground truth and the output of each one of the registration algorithms, called target cloud, is calculated by Nearest Neighbor (NN) distance. This metric calculates the Euclidean distance for each point of the ground truth to a corresponding point in the target cloud, found using the NN algorithm [5]. The total error calculated is the average of the distances throughout the entire cloud.

7.2. Registration errors

Table 1 shows the errors, in centimeters, of different registration methods compared to the ground truth specified in the previous section, using the registration dataset explained in Section 6 and the metric defined above. The error has been calculated for the three scene types, but table 1 presents the averaged result for all of them.

The registration methods presented in table 1 are: Zhang's method [14], manual detection to have an evaluation reference, and three different variations of the proposed finger detector, using the paths illustrated in Figure 7.

The errors are calculated considering a manual alignment as the ground truth. This alignment follows a visual quality criteria. If the point cloud is well aligned for the human eye, it is considered as the best registration. However, when the error is calculated, it may appear higher than expected due to specific high point density area where visual alignment has been considered good, such as the walls, but there is a significant numerical error.

Errors in configuration 1 are higher than in configuration 2. This leads to think that the cause is the sensor placement. Configuration 1 is not a stereo setup, contrary to configuration 2; the angle between sensors is higher and the captured objects can be seen from different points of view. This adds up to the typical depth error of approximately 1 cm found in Kinect sensors, decreasing accuracy and increasing the difficulty to register the scenes.

ICP as a refinement method using the rest of the scene cloud always improves the registration. The amount of improvement will depend on the type of scene and the quality of the initial alignment.

Comparing the different finger paths proposed in subsection 4.2, it can be seen that drawing two separated squares achieves the better initial alignment and, therefore, ICP is able to refine it with a better outcome. The paths are drawn between 0.8 and 1.5m from the sensors.

The errors are indeed higher in the automatic detection of two separated squares drawn in the air if we compare with the manual detection. Even if the number of detected points is larger, the noise and location errors introduced by the automatic detector slightly affects the result.

Drawing just a square does not give a good alignment. The lack of resolution in depth results in a major error in the registration. When, on the contrary, two squares are drawn in separate depths, depth indeterminacy disappears.

When drawing a spiral, there is a rotation indetermination which leads to bad registration as well.

Zhang's work [14] together with ICP as a refinement has the lowest error, closely followed by the finger detector using the path with two squares, along with ICP. The errors, from 1.6 to 3.94 cm, are similar to the Kinect depth error of 1 cm.

To sum up, it is safe to say that finger detection to auto-register the sensors is a good registration method and more user-friendly than the method proposed in Zhang's work [14].

7.3. Modified interactive application

To illustrate the purpose of scene registration and validate its result, Exipple Studio [10] has provided an existing hand tracking application. In this case, the registration is used to modify the application to increase the area where the hand can be detected and tracked.

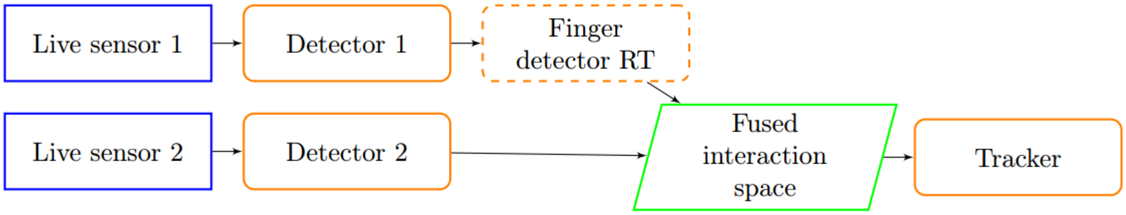
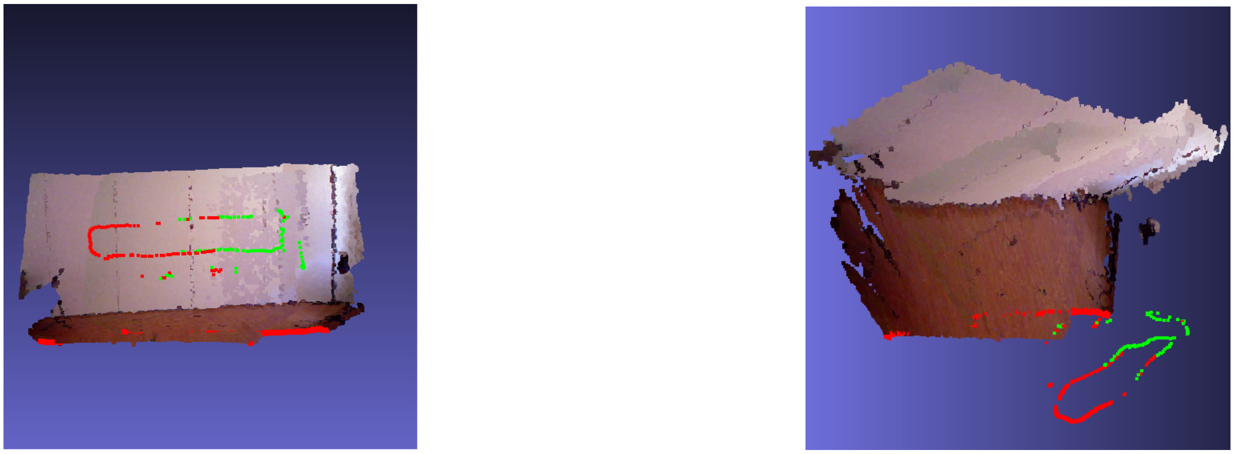


Fig. 10. Application workflow

	Configuration 1 (intersecting)	Configuration 2 (parallel)
Zhang [14]	5.31	1.93
Zhang [14] + ICP	3.47	**1.60**
Manual alignment (path = 2 squares)	4.75	2.12
Manual alignment (path = 2 squares) + ICP	2.76	1.85
Detector(path = square)	59.5	62.7
Detector(path = square) + ICP	46.76	3.96
Detector(path = 2 squares)	6.61	11.55
Detector(path = 2 squares) + ICP	3.94	**2.31**
Detector(path = spiral)	54.81	20.86
Detector(path = spiral) + ICP	4.00	2.39

Table 1. Registration errors in cm


(a) Frontal view

(b) Rotated view

Fig. 11. Application output point cloud

As seen in Figure 10, each Kinect is connected to a different server that detects the hand (Detector 1, Detector 2) and sends 3D detected points to the Tracker. The detector used in this stage is the same used in the automatic finger detection in subsection 4.2, but changing the model to detect the hand instead of the finger, which is a more robust and previously trained model by Exipple studio.

All detected points from the registered sensor to the second sensor have to be transformed with the transformation matrix obtained with the previous registration. After all points from both sensors are registered, they are provided to the tracker as candidates.

The tracker decides, using an internal score, which one of the pool of candidates is better to follow the trajectory of the previous points. When the hand is only captured by one sensor, there will only be one candidate at a time. However, if the hand is in the overlapped viewing area of both sensors, the tracker will consider two candidates and will choose the one with higher score. The final tracked 3D points are stored consecutively in a point cloud.

To test the registration, several figures have been drawn in front of the sensors, with the Configuration 2, illustrated in Figure 9.

In Figure 11, a rectangle has been drawn with the closed hand. The figure shows the captured room as a background and the point cloud with the detections. Green points are the hand detections in sensor 1; the red points, detections in sensor 2.

The transition between both sensors appears smooth and well registered. Considering that a single sensor has a detection distance of 140 cm at 120 cm and the registered sensor has 250 cm, as seen in Figure 9, the distance increase from the registration is 78.5%. Therefore, considering that the viewing height for both registered and not registered captures is the same, the increase in width, or detection distance, leads to the same 78.5% increase of the working (detection) area at 120 cm from the sensors.

8. CONCLUSIONS

The initial objective was to compare our proposed registration method to the traditional registration procedure [14]. In order to validate the registration achieved, a real-time hand tracking application, provided by Exipple Studio, has been modified and adapted to track a hand within the detection area covered by the two registered sensors.

When using the detector, drawing a specific path to register clouds gives a good initial alignment to register scenes. By refining the registration with ICP using the entire scene, the average registration error is between 2.31 and 3.94 cm. This is a good registration, comparable to the results obtained with Zhang's work (1.6 to 3.47 cm). It is also within the range of the 1 cm depth error of Kinect sensors.

The clear advantage of using a detector instead of a pattern is the user-friendly factor and the suppression of additional material needed. Time elapsed for registration in both cases are similar.

The continuation of the tracked hand across the detection area of the sensors is smooth and the application can be used as an improvement of the single sensor version.

A single sensor has a 140 cm effective width of the working area at 120 cm from the sensor. Two registered sensors can achieve 250 cm of working area width. The sensors need a minimum overlap of 30 cm at 120 cm from the sensors, to achieve an increase of more than 70% in the detection distance without affecting the quality of the registration.

9. REFERENCES

[1] K.S. K. S. ARUN *et al.* Arun. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, September 1987.

[2] P. J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[3] T. Beyl, P. Nicolai, J. Raczkowsky, H. Wrn, M. D. Comparetti, and E. De Momi. Multi kinect people detection for intuitive and safe human robot cooperation in the operating room. In *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–6, November 2013.

[4] M. S. Brown and W. K. H. Wong. Laser pointer interaction for camera-registered multiprojector displays. In *2003 International Conference on Image Processing, 2003. ICIP 2003. Proceedings*, volume 1, pages I–913–16 vol.1, September 2003.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[6] Bing Han, Christopher Paulson, and Dapeng Wu. Depth based image registration via 3d geometric segmentation. *J. Visual Communication and Image Representation (JVCIR)*, 22(5):421–431, 2012.

[7] Adolfo López-Méndez and Josep R. Casas. Can our TV robustly understand human gestures? Real-time gesture localization in range data. In *Proceedings of the 9th European Conference on Visual Media Production*, pages 18–25. ACM, 2012.

[8] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings*, pages 145–152, 2001.

[9] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3d registration. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pages 3212–3217, May 2009.

[10] Exipple Studio. Advance user experience design. http://exipple.com. Accessed: 2016-09-26.

[11] Xavier Suau, Marcel Alcoverro, Adolfo López-Méndez, Javier Ruiz-Hidalgo, and Josep R. Casas. Real-time fingertip localization conditioned on hand gesture classification. *Image and Vision Computing*, 32(8):522–532, August 2014.

[12] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence*, 14(4):407–422, 2005.

[13] R. S. Yang, Y. H. Chan, R. Gong, M. Nguyen, A. G. Strozzi, P. Delmas, G. Gimel'farb, and R. Ababou. Multi-Kinect scene reconstruction: Calibration and depth inconsistencies. In *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, pages 47–52, November 2013.

[14] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.

[15] Z. Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia*, 19(2):4–10, February 2012.