A QUALITY EVALUATION OF SINGLE AND MULTIPLE CAMERA CALIBRATION APPROACHES FOR AN INDOOR MULTI CAMERA TRACKING SYSTEM

Michele Adduci*, Konstantinos Amplianitis* and Ralf Reulke

Humboldt Universität zu Berlin
Computer Science Department, Computer Vision Group
Rudower Chaussee 25, 12489 Berlin, Germany
(michele.adduci, konstantinos.amplianitis, reulke)@informatik.hu-berlin.de

Commission V, WG V/1

KEY WORDS: Calibration, Single Camera Orientation, Multi Camera System, Bundle Block Adjustment, ICP, 3D human tracking

ABSTRACT:

Human detection and tracking has been a prominent research area for several scientists around the globe. State of the art algorithms have been implemented, refined and accelerated to significantly improve the detection rate and eliminate false positives. While 2D approaches are well investigated, 3D human detection and tracking is still an unexplored research field. In both 2D/3D cases, introducing a multi camera system could vastly expand the accuracy and confidence of the tracking process. Within this work, a quality evaluation is performed on a multi RGB-D camera indoor tracking system for examining how camera calibration and pose can affect the quality of human tracks in the scene, independently from the detection and tracking approach used. After performing a calibration step on every Kinect sensor, state of the art single camera pose estimators were evaluated for checking how good the quality of the poses is estimated using planar objects such as an ordinate chessboard. With this information, a bundle block adjustment and ICP were performed for verifying the accuracy of the single pose estimators in a multi camera configuration system. Results have shown that single camera estimators provide high accuracy results of less than half a pixel forcing the bundle to converge after very few iterations. In relation to ICP, relative information between cloud pairs is more or less preserved giving a low score of fitting between concatenated pairs. Finally, sensor calibration proved to be an essential step for achieving maximum accuracy in the generated point clouds, and therefore in the accuracy of the produced 3D trajectories, from each sensor.

1 INTRODUCTION

In recent years, computer vision and machine learning groups started developing algorithms for detecting people in an organized 3D point cloud which can be quite promising considering the metric informations that can be extracted from the detected people. In (Luber et al., 2011) a machine learning technique for detecting people in a point cloud using features trained in RGB and Depth images was developed. Another approach consists in an on-line boosted target models for detection and tracking (Spinello and Arras, 2011). These aforementioned techniques are applied using a single sensor; these approaches might suffer of typical 2D related problems which might occur in some scenarios, such as occlusions, fast illumination changes, natural or physical scene constraints. Therefore it is important to investigate and research on techniques applied in a multi camera system, which can enhance the quality of informations retrieved from the environment and overcome the limitations of single camera systems. In this publication, we evaluate the calibration approaches for single and multiple camera, which are used in an indoor configuration for human detection and tracking: knowing the absolute and relative position of the cameras, we are able to reconstruct a full 3D scene which can be analysed by tracking algorithms.

1.1 Related Work

In this work, we used a RGB-D sensor, a Microsoft Kinect, a low cost device which disrupted the computer vision's world, enabling a new series of studies, techniques and results. One of the most widely used approaches to calibrate a single camera, modeled as a pinhole camera model is the Brown model (Brown, 1971), and also the technique explained in (Zhang and Zhang, 1998) which estimate intrinsic camera parameters using a planar pattern. An analysis on the accuracy of data acquired using a Microsoft Kinect was discussed extensively in (Khoshelham, 2011) and, for indoor mapping applications, in (Khoshelham and Elberink, 2012).

Since the RGB and the Infrared sensor must be co-registered, a cross calibration of both sensors must be performed. In (Almazan and Jones, 2013), they proposed an interesting approach to calibrate Infrared and RGB sensors using normals on a plane is described.

In a multiple camera system, it is fundamental to know the position of each camera respect to an object; some techniques concerning single camera pose estimation and solution to the Perspective Three Point problem are discussed in (Gao et al., 2003) and (Lepetit et al., 2009). More refined approaches to compute intrinsic and extrinsic camera parameters, pose estimation and scene reconstructions are based on bundle adjustment, as reported in (Hartley and Zisserman, 2004) and (Luhmann, 2011). In general, the reconstructed 3D scene obtained from bundle adjustment or other techniques might require a better alignment and refinement: the best choice is to run the Iterative Closest Point (Besl and McKay, 1992) or one of its variants.

We mentioned above that human detection and tracking is an active topic: some interesting results are presented in (Luber et al., 2011) and (Spinello and Arras, 2011), which use a trained classifier based on RGB and Depth images, using Histograms of Oriented Gradients (HOG) and Histograms of Oriented Depths based detectors.

The approach used in this paper was inspired from our previous work (Amplianitis et al., 2014), where each point cloud is analyzed independently from others and the active foregrounds are detected and tracked.

*These authors contributed equally to this work

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-5-9-2014

2 THEORETICAL BACKGROUND

This section provides the reader with an overview of theoretical and mathematical disciplines intended for understanding the proposed approach.

2.1 Calibration

Calibration involves finding the parameters that will correct the position of an image point due to lens distortions. Within this work, calibration was done by combining the 10-parametric model introduced by (Brown, 1971) and the calibration approach introduced by (Zhang and Zhang, 1998). Employing the pinhole camera model, this is an ideal mathematical assumption for cameras that often deviate from the ideal perspective imaging model. Also, the pinhole model comes along with lens distortions parameters which can compensate lens errors and give correct image points. Most important type of distortion is the *radial distortion* which causes an inward or outward position of the image point from its ideal position and is expressed by:

$$\begin{aligned}\Delta x_{rad} &= x[k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots] \\ \Delta y_{rad} &= y[k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots]\end{aligned} \tag{1}$$

where x, y are the distorted image points, r is the euclidean distance of every point in the image with respect to the principal point (also known as *image radius*) and $k_1, \dots, k_n$ are the radial distortion coefficients used for modelling the radial distortion curve. For most standard types of lenses, going further than the third order parameters could be neglected without any significant loss in the accuracy of the points.

Second form of distortion is the *decentering* (or tangential) distortion which is caused by physical elements in a lens not being perfectly aligned to the image plane and it's existance is mostly due to manufacturing defects and can be compensated by the following function:

$$\begin{aligned}\Delta x_{dec} &= p_1(r^2 + 2x^2) + 2p_2 xy \\ \Delta y_{dec} &= p_2(r^2 + 2y^2) + 2p_1 xy\end{aligned} \tag{2}$$

where p_1, p_2 are the decentering parameters. This lens correction part can give large values for low cost cameras (such as surveillance cameras) and smaller quantity distortion values for high quality lenses.

Finally, the *affinity* and *shearing* parameters are used to describe deviations of the image coordinate system with respect to the non orthogonality and uniform scaling of the coordinate axes. This is mathematically expressed as:

$$\begin{aligned}\Delta x_{aff} &= b_1 x + b_1 y \\ \Delta y_{aff} &= 0\end{aligned} \tag{3}$$

where b_1, b_2 express the affinity and shearing parameters respectively. It is noteworthy that in most cameras used in close range application, b_1 and b_2 are set to zero.

Overall, these individual terms used to model the imaging errors could be summarized as follow:

$$\begin{aligned}\Delta x' &= \Delta x_{rad} + \Delta x_{dec} + \Delta x_{aff} \\ \Delta y' &= \Delta y_{rad} + \Delta y_{dec} + \Delta y_{aff}\end{aligned} \tag{4}$$

For the calibration method introduced by (Zhang and Zhang, 1998), this is considered a well suited technique for finding the pose (position and rotation) of a camera by observing a planar pattern object (such as a chessboard) shown from at least two different points of view. It is a close form solution, followed by a nonlinear refinement based on a maximum likelihood criterion. For further technical information refer to (Zhang and Zhang, 1998).

2.2 Single camera pose estimation

Single camera pose estimation is a topic extensively investigated for many decades and it has several applications not only in computer vision but also in relevant fields such as robotics and augmented reality fields. Within this work, a quality evaluation on a recent iterative and non-iterative approach is made by checking the quality of the camera pose through its reprojection error. Subsequently, these pose parameters, together with the lens distortion coefficients computed in the calibration step for each camera, are given as initial values to a photogrammetric bundle block adjustment algorithm. Relative information between camera pairs is crucial for determining the initial rigid transformation guess required as an input to the ICP algorithm.

As was previously mentioned, two different solutions to the PnP problem are evaluated. The first method is a non-iterative solution proposed by (Lepetit et al., 2009), who tried to express the *n* 3D points as a weighted sum of four virtual control points. In this way, according to the author, time complexity (which is linearly growing with *n*) is significantly reduced to a $O(n)$ time. In the second approach, non linear Levenberg-Marquardt optimization algorithm was used for minimizing the sum of square distances between the observed points and calculated projected points. From the $2D \leftrightarrow 3D$ correspondences, 3D points are considered to be error-free while 2D points should be compensated for lens distortions. If that was not true, camera pose would be computed based on undistorted points which is not trivial.

2.3 Bundle Block Adjustment

Bundle is defined as a bundle of rays that span in 3D space starting from the center of the cameras, going through the image points and intersecting in space. The problem that occurs from the intersection of rays in space is that they do not meet at an optimal point (best intersection of the corresponding rays). Thus, bundle adjustment deals with the rearrangement of the camera positions and 3D points in order to achieve an optimal intersection of the rays in 3D space. In practice, that is interpreted by trying to minimize the distance between the measured points on the images and the ones that are back projected from the rearrangement of the cameras and reconstructed points (Figure 1). This is mathematically expressed as:

$$\min_{\hat{P}_i, \hat{X}_j} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\| x_{ij}, \hat{P}_i \hat{X}_j \right\|^2, \hat{x}_{ij} = \hat{P}_i \hat{X}_j \tag{5}$$

where x_{ij} is the image point j on image i, $\hat{P}_i$ is the 3×4 projection camera matrix corresponding to the i^{th} image and $\hat{X}_j$ is the corresponding 3D point. In case of planar 3D objects, such as an ordinal chessboard, the position of the 3D points is precise and error free. Each projection camera in general has 11DOF and every 3D point has 3DOF. Within this work, a general form projection camera matrix is used ($>$ 11 DOF) in which the calibration matrix also incorporates the lens distortion parameters. Fixing the position of the 3D points, minimization is over $15m$ parameters (6 exterior and 9 interior) where m the number of views.

Let $f(p)$ be a function that relates a parameter vector p, with an estimated measurement vector $\hat{x} = f(p), \hat{x} \in \Re^2$. The estimated measurement vector contains the corrected pose parameters defined from the optimization of the bundle of arrays and $f(p)$ is a function that has as arguments of p the parameters of the cameras. Therefore p is of the form:

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-5-9-2014

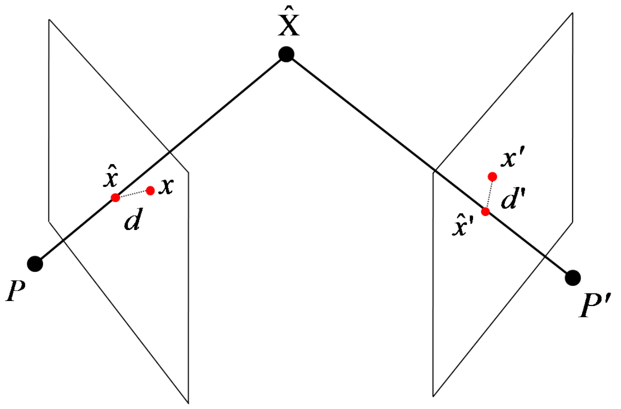

Figure 1: The reprojection error

$$p = [P_1, P_2, \ldots, P_n, X_1, X_2, \ldots, X_m] \tag{6}$$

where P_i is the 3 × 4 projection camera matrix over n number of cameras and X_i are the 3D ground control points (chessboard points) which remain invariant throughout all the bundle process. An initial parameter vector p_0 and measurement vector x (observed image points) are required for finding a vector that best satisfies the functional relation, meaning finding the values of this vector that will locally minimize the objective function. As a stopping criterion, we define the error between the observed and calculated values, which has to be minimized (refer to relation 5). Using the $Gauss - Newton$ non linear least square approach, the mathematical model is defined as follows:

$$\left(J^T J\right) \delta_P = -J\varepsilon_p \tag{7}$$

where J is the Jacobian matrix defined as $J = \frac{\partial f(p)}{f(p)}$, δ_p is the corrected measurement vector and ϵ_p is the error between the observed and calculated measurement vectors $x - \hat{x}$. For further reading refer to (Hartley and Zisserman, 2004) and (Luhmann, 2011).

2.4 Iterative Closest Point (ICP)

The Iterative Closest Point (ICP) algorithm is a widely used algorithm for registering two 3D datasets. Within the algorithm, one point cloud acts as the reference cloud (also known as target), while the other one, known as source, is transformed to best match the reference. That involves an iteration process in which the source cloud would undergo a rigid transformation (that implies translation and rotation) for minimizing the distance to the reference point cloud. In mathematical notation, ICP could be given in the form of a pseudo code as seen in algorithm 1. For further details of the algorithm refer to (Besl and McKay, 1992). Using the same notation as in the pseudo algorithm for the source dataset(P) and target dataset(M), the rigid relative transformation R_{MP} given as an initial guess for the ICP is defined as:

$$\begin{aligned} R_{PM} &= {R_M}^T R_P \\ T_{PM} &= {R_M}^T (T_P - R_M) \end{aligned} \tag{8}$$

where R_P, R_M express the rotation matrices that rotate the source and target clouds to the world system and T_P, T_M express the position of the clouds within the world system.

Algorithm 1 Iterative Closest Point (ICP) algorithm

Require: $\mathcal{P}$ (source) and $\mathcal{M}$ (target) datasets
Ensure: Transformation $(R', \boldsymbol{t}')$, Error ε

```
$R' \leftarrow I, \boldsymbol{t}' \leftarrow \boldsymbol{0}, \varepsilon \leftarrow \infty$
while ($\varepsilon > threshold$) do
    $\mathcal{Y} \leftarrow \{\boldsymbol{m} \in \mathcal{M} \mid \boldsymbol{p} \in \mathcal{P} : \boldsymbol{m} = cp(\boldsymbol{p})\}$
    $(R, \boldsymbol{t}, \varepsilon) \leftarrow \min_{R,\boldsymbol{t}} \sum_{k=1}^{n_p} |\boldsymbol{y_k} - R\boldsymbol{p_k} - \boldsymbol{t}|^2$
    $\mathcal{P} \leftarrow R \cdot \mathcal{P} + \boldsymbol{t}$
    $R' \leftarrow R \cdot R'$
    $\boldsymbol{t}' \leftarrow R \cdot \boldsymbol{t}' + \boldsymbol{t}$
end while
```

2.5 People Detection and Tracking

Detecting people in a point cloud is a current research topic for many scientists. Within this work, detection of people is based on previous work introduced in (Amplianitis et al., 2014), where current foreground is extracted by detecting the spatial differences between the computed OcTree of the static background and the OcTree of the current cloud.

In the tracking part, Kalman filter was used to predict the next target location of the person. Initialization of the tracking was done by extracting the convex hull of the body silhouette and computing the mean from it. Every person was detected and tracked independently by every sensor.

3 EXPERIMENTAL RESULTS

The experimental part is divided into three sections: First section is a description of the room used for testing and evaluating the detection and tracking algorithm. Second section provides results coming from the internal calibration process of each Kinect sensor. In the third section, an evaluation of the bundle adjustment accuracy and the ICP algorithm are given based on the initial camera pose parameters computed by the algorithms explained in section 2.2. Finally in the fourth section, we demonstrate some tracking results of moving people in the scene and discuss the accuracies.

For chariness within this section, additional tables and graphs have been appended in an Appendix section.

3.1 Camera configuration and hardware

Our experimental setup consists of four RGB-D Microsoft Kinect sensors mounted on a aluminium construction as depicted in figure 2. The space within the aluminium construction has dimensions of approximately $4.5 \times 2.2 \times 2.3$ depth, width and height respectively. Data acquisition was done from all cameras in parallel with an acquisition rate of $\approx$ 19 fps. One of the main drawbacks of using multiple structured light sensors is the drastic reduction of the quality of the depth images due to the intersection of their near-infrared light in space. Therefore, all sensors where oriented looking towards the center of the scene and the amount of overlapping was restricted only in the lower part of the field of view (FOV).

Giving some technical characteristics of the Kinect sensor, it looks like a horizontal elongated bar, connected on a base with a motorized pivot and consists of two cameras and one infrared emitter.

The depth is produced by projecting a known pattern onto the scene by the infrared emitter and then infer depth from the deformation of that pattern recorded by the infrared image. The resolution of the RGB and depth images is 640 × 480 pixels and so the generated point cloud has an organized form of 307200 points. Best working limit without introducing artifacts is within the range of 1.2 and 3.5 meters.

Concerning hardware performance, a computer with an Intel Core i7-3770 processor, 16GB RAM and a Samsung 840 Pro SSD was used. Although good hardware is essential for these kinds of applications, real time detection and tracking would require extra hardware performance and software optimization.

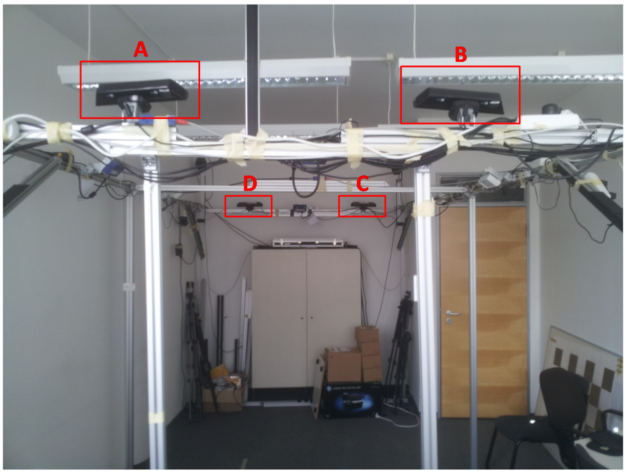

Figure 2: Aluminium construction with mounted Kinects sensors

3.2 Calibration

There are several libraries (eg. OpenNI, FreeKinect) which provide out-of-the-box calibration parameters of the Kinect sensor. Nevertheless, for achieving maximum possible accuracy of the generated point clouds (implying better 3D tracking results), a more precise calibration is required. Main advantage of the Kinect sensor is that it uses low distortion lenses with faintly apparent displacement errors around the corners/edges of the images. The calibration process was conducted in the following manner: as a 3D object, a well defined chessboard pattern was used with a chessboard size of $2cm$ and inner dimensions of 5 × 7 rows and columns respectively. Since infrared and RGB sensors cannot work simultaneously, they were switched on and off continuously (a switch lasts 0.5 seconds), in order to acquire roughly the same chessboard data, from different perspectives. Detection and acquisition of chessboard points was done in a live mode using OpenCV's chessboard corner detector, which also provides subpixel accuracy. To avoid any disturbances of the speckles coming from the infrared emitter in the infrared camera, the emitter was covered with tape and a external light was used for detecting the chessboard corners. A total amount of 100 images was acquired and splitted using a random selection algorithm in 10 different sets of 24 images each, in which the calibration was performed independently from other sets. Applying (Zhang and Zhang, 1998) algorithm stated in section 2.1, internal calibration results are given in tables 3 and 4 of Appendix A. The overall RMS for the infrared and RGB camera respectively for each camera is given in table 1. As was expected, the RMS is within the range of almost a quarter of a pixel for every sensor significantly outperforming in quality the RMS default results provided by the default Kinect parameters (0.34px for IR and 0.53px for RGB camera). Figures 6 and 7 from Appendix A show the radial symmetric curve of the infrared and RGB lenses of every Kinect sensor expressed by the principal point. It is clear that the radial distortion almost follows a constant zero effect for most of the distances and starts effecting the points' displacement only towards the edges of the image (less of half a pixel in the extreme regions in both infrared and rgb sensor). Moreover, the decentering distortions parameters provided by table 4 in Appendix A have very small values clearly proving the quality and stability of both infrared and RGB lenses.

Sensor Type	A	B	C	D
IR	0.1711	0.2171	0.2079	0.3357
RGB	0.2284	0.2409	0.2162	0.2157

Table 1: Calibration RMS error for IR and RGB sensor (in pixels)

3.3 Single camera orientation and bundle adjustment

Solving a bundle block adjustment system requires good initial approximated values for internal and external parameters of all sensors. As was mentioned in section 3.1, all four RGB-D sensors are mounted in a way that would be quite complicated for a person to empirically provide sufficient initial guess. One solution to the problem was to use the well known *Direct Linear Transformation (DLT)* algorithm but for planar objects such as the chessboard, DLT would fail due to its coplanarity constrain. Therefore, approaches such as $PnP - LM$ and $EPnP$ that are designed to retrieve the pose of a camera from planar objects where considered (refer to section 2.2).

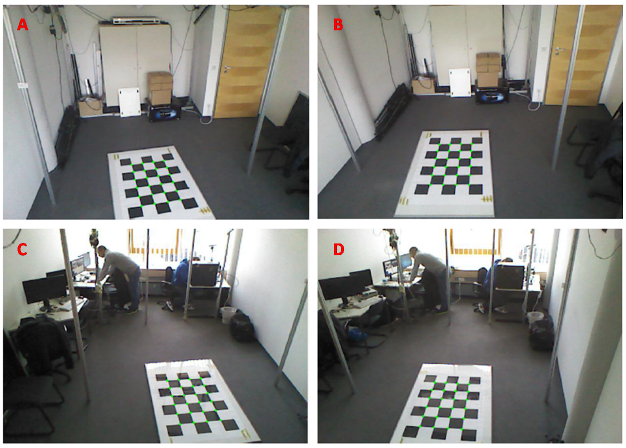

Figure 3: Optimized reprojected image points

ID	A	B	C	D
PnP - LM	0.240	0.247	0.255	0.231
EPnP	0.463	0.432	0.389	0.372

Table 2: Kinects reprojection error for initial pose estimation (in pixels)

Full ground control points were generated from the chessboard with their Z value set to zero.

Making use of this form of a reference object has its advantages and disadvantages. Main advantages are that it is easily portable in indoor environments, can easily be used as a reference object for orienting a set of cameras and also provides error free ground control points. On the other hand, coplanar objects lack of spatial

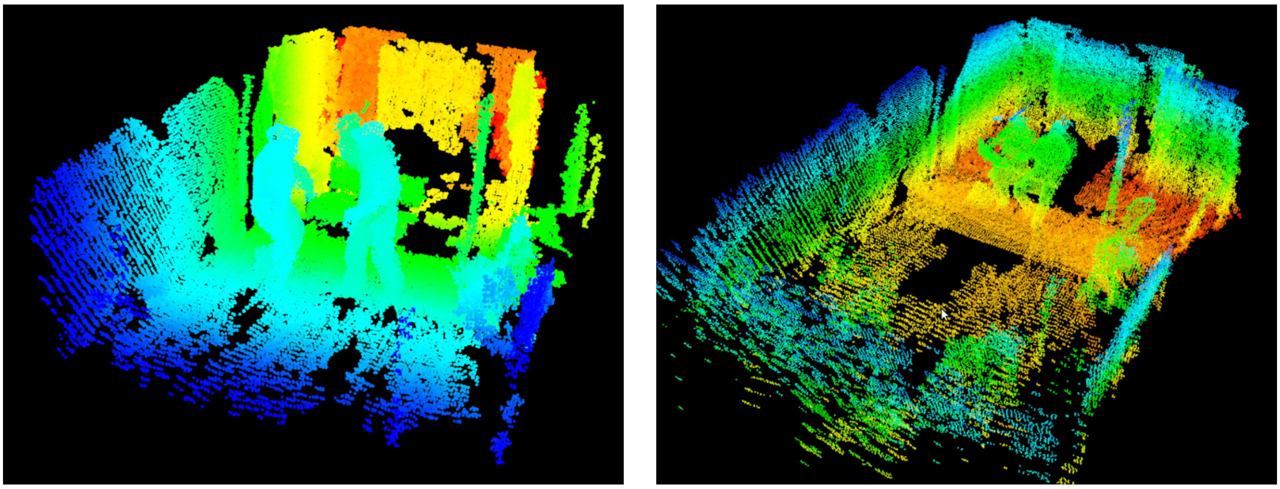

Figure 4: Final result of ICP fusion. Left figures shows two people standing and iteracting whears in the right figure two are sitting and chatting

distribution information and introduce several geometrical constrains. Acquiring the pattern from all four sensors simultaneously and running the algorithms discussed in section 2.2 the resultant reprojection errors are given in table 2 which can also be seen in figure 3.

The generation of point clouds is performed using Kinects' depth and RGB image: the depth information is used to produce the 3D points, which are then textured using RGB information.

Mathematically this is expressed by finding the rotation and translation parameters (3D rigid transformation) between the two camera systems which can be computed through a classical stereo calibration procedure. Knowing the focal length, principal point and radial distortion parameters, every 3D point (expressed now in the RGB camera system) can be projected on the RGB image, interpolate the color and assign it to the point (Khoshelham, 2011). Therefore, it is clear that the rest of the working pipeline will be based on point clouds orientated with respect to the RGB camera system rather than the infrared.

Considering all exterior and interior parameters as unknown for each sensor, a bundle block adjustment was applied for refining these parameters so that the condition 5 is met. Setting a convergence condition of $0.0001m$ as a minimum distance between observed and calculated points, both $PnP - LM$ and $EPnP$ converged after 3 iterations and returned a σ_o of 0.00029 and 0.00032 respectively. It is clear that both methods provide similar pose parameters and therefore explains the minimal corrections bundle adjustment did on the initial values. Results of point clouds translated and oriented with respect to the world chessboard system are given in figure 4. Left figure shows the complete fused cloud together with the presence of two people interacting with each other. As can be seen, both human are represented as a solid body. Equivalent results are also given by the right figure with two people sitting on a chair.

Given as initial guess the relative transformation between two pair-wise clouds, AB and CD, based on equation 8, ICP was performed only once for the first frames of a scene and remained fixed for the rest. The number of iterations given as a converging conditions were 500, with an epsilon of 1×10^{-8} and maximum correspondences' distance of $5cm$. For the first pair, AB, the score of convergence was $0.017m$, whereas for the second pair, CD, the convergence was $0.048m$. In both cases, the fusion was done quite fast and with a very good error fit.

3.4 Human detection and tracking

As was stated in section 2.5, moving objects are extracted from all sensors independently and in a parallel process. Knowing the refine relationship between the sensor pairs $A - B$ and $C - D$ (coming from ICP), all foreground points extracted independently from each sensor are now transformed in the same coordinate system. As a result, when a person is currently in the scene, all points corresponding to him will move in the same direction.

Pay notice that random error and point density play a significant role in the accuracy of the foreground. The further away a person is from a sensor, the larger the random error is generated from that sensor. Also, the density of a point is inversely proportional to the square distance from the sensor, which also explains the reduction of density resolution in larger depths. Thus, people being closer to a sensor will contain points from that sensor with a higher density and accuracy and vice versa.

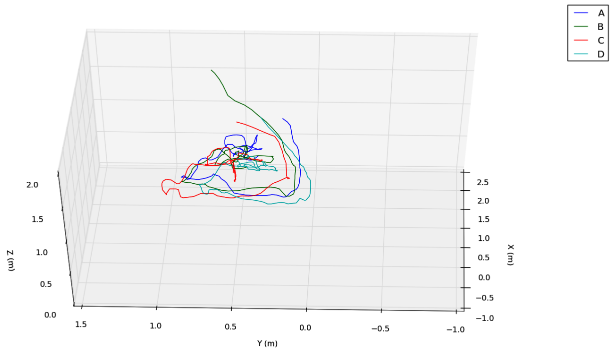

Figure 5: Trajectories of a person as produced by each sensor

For every sensor, an independent trajectory of a human was produced. An approximated center of gravity of a person was derived by extracting the convex hull from all points corresponding to him and computing the mean as proposed in section 2.5. This was also important for initializing the Kalman filter. Figure 5 shows the trajectory of a person generated from all four RGB-D sensors. As can be seen, all trajectories follow the same root and direction of the body with some deviation between them. That is due to noisy foregrounds affecting the size of the convex hull and therefore the position of the center of gravity. Projecting the trajectory coordinate elements into three separate planes, an analysis of each dimension is accomplished individually. Figures 8, 9 and 10 from Appendix B show the amount of variation for each coordinate component independently with the larger offsets visible in the Z axis. That clearly shows that the fluctuation in the size of the convex hull mostly effects the Z direction and less the X and Y.

4 CONCLUSIONS

We performed a processing chain for demonstrating how calibration, bundle adjustment and ICP can affect a multiple RGB-D human detection and tracking system. Pre-calibration of every Kinect sensor is an essential step for rectifying the RGB and IR images and producing high quality point clouds. Although the correction of infrared and RGB images is mostly in the extrema regions, this could significantly affect the position of a moving object in the scene, depending on the distance of the object from each sensor. Planar single pose camera estimators have shown that although they have small orientation differences, they don't lack of accuracy and that was immediately proven by performing a bundle block adjustment. Given as initial values the results coming from the single camera pose estimators and setting all internal and external parameters as unknowns to the bundle (except ground control points) the algorithm would converge after very few iterations achieving a σ_o of less than a mm. Giving as initial guess a 3D rigid transformation to ICP for every pair of point clouds, the fitting score for both pairs was within cm accuracy, clearly proving the necessity of all aforementioned procedures. In the detection and tracking part, generated trajectories of moving people were shifted one another due to the accuracy of the foreground and therefore the tracked center of gravity.

5 DISCUSSIONS

Presented work could serve well as a solid foundation of evaluating multi RGB-D camera systems. Human detection and tracking in a point cloud is still an unexplored area with not much work introduced by the research community. Therefore, improving our current implementation and refining foreground masks could significantly improve the accuracy of the tracks. Optimal goal involves having a unique trajectory for every person in the scene produced by the confidence of the intermediate tracks generated from each sensor.
Time performance is also a very crucial factor for our algorithm. Using GPU programming could significantly improve our algorithmic workflow and bring it closer to a more real time application. On the other hand, RT applications would require having one computer per sensor, due to the amount of computational power needed to manage all devices simultaneously.

REFERENCES

Almazan, E. J. and Jones, G. A., 2013. Tracking people across multiple non-overlapping rgb-d sensors. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Amplianitis, K., Adduci, M. and Reulke, R., 2014. Calibration of a multiple stereo and rgb-d camera system for 3d human tracking. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-3/W1, pp. 7–14.

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14(2), pp. 239–256.

Brown, D. C., 1971. Close-range camera calibration. Photogrammetric Engineering 37(8), pp. 855–866.

Gao, X.-S., Hou, X.-R., Tang, J. and Cheng, H.-F., 2003. Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. 25(8), pp. 930–943.

Hartley, R. I. and Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Second edn, Cambridge University Press, ISBN: 0521540518.

Khoshelham, K., 2011. Accuracy analysis of kinect depth data. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-5/W12, pp. 133–138.

Khoshelham, K. and Elberink, S. O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors 12(2), pp. 1437–1454.

Lepetit, V., F.Moreno-Noguer and P.Fua, 2009. Epnp: An accurate o(n) solution to the pnp problem. International Journal Computer Vision.

Luber, M., Spinello, L. and Arras, K. O., 2011. People tracking in rgb-d data with on-line boosted target models. In: Proc. of The International Conference on Intelligent Robots and Systems (IROS).

Luhmann, T.; Robson, S. S. K. H. I., 2011. Close Range Photogrammetry: Principles, Methods and Applications. Second edn, Whittles Publishing of Caithness, Caithness, Scotland, UK.

Spinello, L. and Arras, K. O., 2011. People detection in rgb-d data. In: Proc. of The International Conference on Intelligent Robots and Systems (IROS).

Zhang, Z. and Zhang, Z., 1998. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, pp. 1330–1334.

APPENDIX

Appendix A: Calibration results

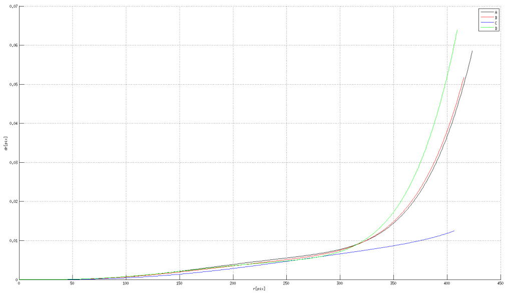

Figure 6: Radial symmetric distortion curves for all IR sensors

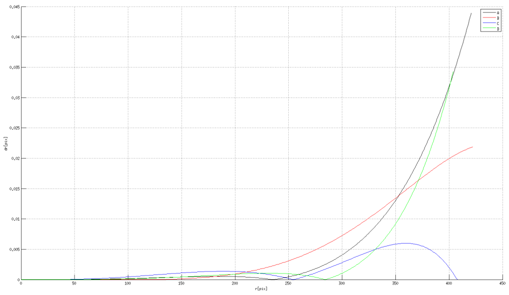

Figure 7: Radial symmetric distortion curves for all RGB sensors

	ID	c_x	c_y	x_o	y_o
RGB	A	569.833	569.117	335.038	259.227
	B	569.220	569.198	323.839	260.600
	C	586.786	585.971	327.520	240.808
	D	591.379	591.084	330.742	241.404
Infrared	A	511.908	510.640	330.890	259.780
	B	497.488	497.568	326.354	267.325
	C	516.212	516.059	324.874	246.470
	D	521.307	520.709	323.213	242.130

Table 3: Infrared and RGB camera internal parameters (in pixels)

	ID	k_1	k_2	p_1	p_2	k_3
Infrared	A	-0.1648	0.7673	0.0069	0.0064	-1.3135
	B	-0.1480	0.6950	0.0079	0.0017	-1.2501
	C	-0.0992	0.2648	0.0022	-0.0036	-0.2833
	D	-0.1933	1.1570	0.0002	0.0039	-2.4079
RGB	A	0.0288	-0.1249	0.0070	0.0056	-0.0506
	B	0.0093	-0.1718	0.0124	0.0048	0.1517
	C	0.0803	-0.4595	0.0072	-0.0008	0.5304
	D	0.0187	0.0259	-0.0009	0.003	-0.2990

Table 4: Infrared and RGB camera lens distortion parameters (in pixels)

ID	T_x	T_y	T_z	R_x	R_x	R_x
A	-2.16	0.01	0.78	0.9370	0.2804	-0.3862
B	-2.68	0.41	-0.63	0.8422	0.0489	0.1004
C	-1.84	-0.73	-0.09	0.536	-2.6761	-0.2824
D	-2.01	0.11	-0.19	-0.1178	0.4829	0.2939

Table 5: Translation (cm) and rotation (deg) parameters between RGB and IR camera

Appendix B: Detection and tracking

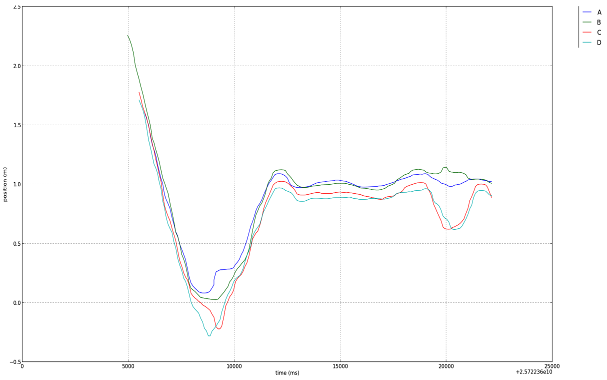

Figure 8: Variation of the trajectories along the X axis

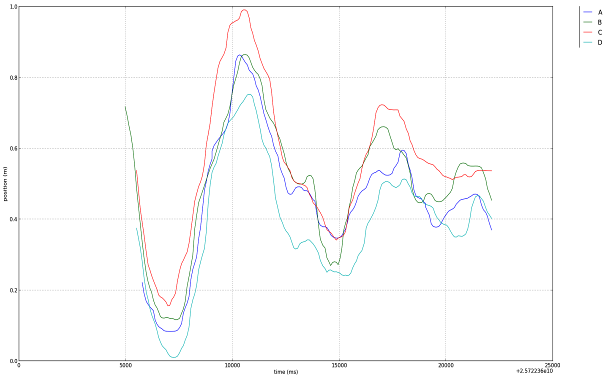

Figure 9: Variation of the trajectories along the Y axis

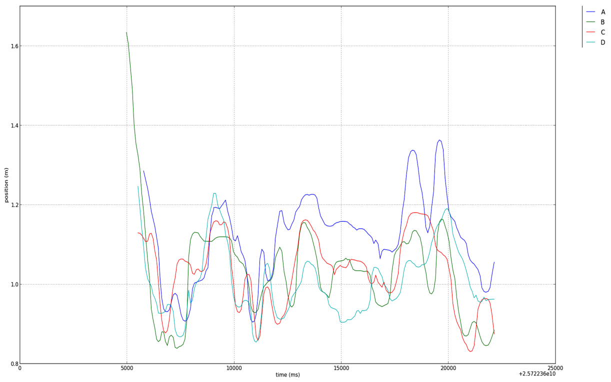

Figure 10: Variation of the trajectories along the Z axis