A. Lejeune, D. GROGNA, M. VAN DROOGENBROECK and J. VERLY. **Evaluation of pairwise calibration techniques for range cameras and their ability to detect a misalignment.** In *International Conference on 3D Imaging (IC3D)*, Liège, Belgium, December 2014.

EVALUATION OF PAIRWISE CALIBRATION TECHNIQUES FOR RANGE CAMERAS AND THEIR ABILITY TO DETECT A MISALIGNMENT

Antoine Lejeune, David Grogna, Marc Van Droogenbroeck and Jacques Verly

Department of Electrical Engineering and Computer Science, University of Liège, Belgium

ABSTRACT

Many applications require the use of multiple cameras to cover a large volume. In this paper, we evaluate several pairwise calibration techniques dedicated to multiple range cameras. We compare the precision of a self-calibration technique based on the movement in front of the cameras to object based calibration. While the self-calibration technique is less precise than its counterparts, it yields a first estimation of the transformation between the cameras and permits to detect when the cameras become mis-aligned. Therefore, this technique is useful in a practical situations.

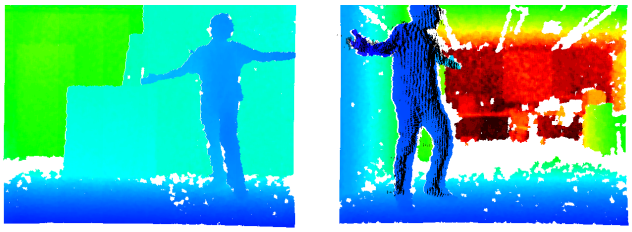
1. INTRODUCTION

Commodity range cameras have recently permitted to develop new applications by providing dense depth maps at real-time frame rate ($>$ 15fps). Some computer vision problems can be solved with such cameras, such as human pose estimation [20], augmented reality, and virtual reality. Indeed, there is no longer a need to infer 3D information from colorimetric information.

Any application that operates in large environments requires multiple cameras to work robustly over the whole volume. Examples of such applications include immersive virtual environment, telepresence set-ups, and gait analysis of humans over long distances [17].

Range cameras capable of recording at real-time frame rate operate only for a short distance range. The Microsoft Kinect can measure depths up to 10 meters but provide a precision on the order of the centimeter only for depth up to a few meters [14]. Time-of-flight cameras, such as the PMD CamCube 2.0, have a physical limit on the maximum depth that can be measured unambiguously. This limit is usually well below 10 meters. Therefore, multiple cameras have to be used to correctly cover areas with a size larger than 10 meters.

To aggregate the information captured from all the cameras, their relative positions need to be estimated. This takes the form of pairwise rigid body transformations (translation and rotation between two cameras). The difficulties for estimating this transformation originate from establishing point correspondences between the images of the cameras. For robust correspondences, a calibration object such as a chessboard is often used. The calibration object also provides Euclidean constraints, which are required for a "metric" calibration of the cameras.



(a) (b)

Fig. 1: Human seen by two range cameras. Black dots in image (b) represent the projection of pixels also found in image (a).

In this paper, we evaluate several methods for pairwise calibration of range cameras, including one self-calibration technique based on movement detection, which can also be used to detect when the cameras become mis-aligned.

In Section 2, we first review existing calibration techniques and the characteristics of range cameras. In Section 3, we describe the pairwise calibration methods used in this paper. In Section 4, we compare the results of the techniques.

The results presented in Section 4 show that techniques directly based on the depth values are less precise than a technique based on the detection of a calibration object in a color image. However, they permit to obtain a first estimation of the rigid body transformation and to detect a misalignment of the cameras by only using the movement observed in the scene.

2. RELATED WORK

2.1. Calibration and self-calibration

The problem of camera calibration has been extensively analyzed for color cameras. Most methods require the use of a calibration pattern, such as a chessboard [3], and perform the pairwise calibration as well as the intrinsic calibration of each individual camera. The calibration object ensures that

(a) Microsoft Kinect (b) PMD CamCube 2.0

Fig. 2: Range cameras used in this work.

the correspondences between the images are correct.

More recently, several authors proposed calibration techniques based on a laser pointer [6, 21]. The cameras record the successive positions of the light to form a virtual calibration pattern which can be used to perform the pairwise calibration of the cameras. A variation of this technique has already been used for the calibration of a network of Microsoft Kinect cameras [1].

Azarbayejani *et al.* [2] proposed a self-calibration technique for wide baseline stereo system. It is based on the detection of the face and the hands of the user in front of the camera. The estimated parameters are thus strongly constrained by the imposed position of the cameras and the content of the scene.

For laser range finders, Glas *et al.* [9] proposed an automatic self-calibration technique in crowded environment. They used an elliptical shape model to estimate, track, and match the positions of the users for each sensor. The field of view of all sensors are within the same plane, meaning that there is only a single angle of rotation to estimate.

Kaenchan *et al.* [13] proposed an iterative technique for Microsoft Kinect cameras based on an initial guess of the transformation. The technique uses the limbs' locations given by the OpenNI framework for point correspondences. However, they don't evaluate the performance of their calibration procedure.

For cameras close to each other, there exists various techniques dedicated to range data that yield accurate results. The iterative closest point algorithm [18] estimates the rigid body transformation by iteratively matching the closest 3D points together. Kinect Fusion [12] is a nonlinear minimization of the reprojection error between two depth maps.

Most techniques developed for color cameras use the reprojection error as the evaluation metric. While this metric is good for RGB and gray-level images, it isn't adequate for sensors measuring geometric information. In this paper, we evaluate the self-calibration accuracy according to the translation and angular error between the cameras.

2.2. Range cameras

A range camera records, for each pixel $\mathbf{p}$, the depth $Z(\mathbf{p})$ of the corresponding element of the scene. If the intrinsic parameters of the camera (focal length and optical center) are known, one can convert the point $(\mathbf{p}, Z(\mathbf{p}))$ to a 3D real-world coordinate in a reference frame located at the optical center of the camera [10]. Thus, these cameras permit a direct metric reconstruction of the scene.

In this paper, we use two different models of range cameras: the Microsoft Kinect (version 1) and the PMD CamCube 2.0 (see Figure 2). These cameras are based on different acquisition techniques. The Microsoft Kinect uses a structured-light technique to estimate the depth [8] and the PMD CamCube 2.0 is a time-of-flight (ToF) camera which estimates the depth by measuring the phase shift between the transmitted and received modulated infrared signals [22].

We note that these cameras can have significant noise or no data at all for some pixels. This happens when not enough light is reflected back to the sensors. This can be due to bad reflective properties or a large angle of incidence with the intercepted surface in the scene. Also, the amount of noise can be different for each pixel of an image. Several models have been proposed to characterize this noise. The Microsoft Kinect has a noise that increases quadratically with the depth [14], and ToF cameras have a noise that increases with the inverse of the amplitude of the signal [7].

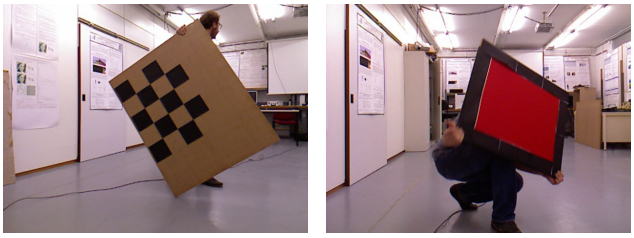
There can be significant artifacts when using more than one camera of the same model: the transmitted signals can interfere with each other. Two Microsoft Kinect can fail to measure depth when their fields of view largely overlap. Schröder *et al.* [19] proposed fast rotating discs to create a time division multiple access scenario. Butler *et al.* [4] reduced the interference between the cameras by using an electric motor which make them vibrate. For ToF cameras, the modulation frequency of the signal can be changed in some cameras. In this case, there is no problem in using more than one camera. When two cameras use the same modulation frequency, Castabeda proposed a technique which still permits the use of several cameras in a wide baseline stereo set-up [5]. In this paper, we don't use any such technique. We limit our study to the pairwise calibration of the cameras.

Here, we assume that the intrinsic parameters of the range cameras are already calibrated. We note that these parameters have to be determined only once and that the Microsoft Kinect camera is factory calibrated. For ToF cameras, there exists dedicated calibration techniques, e.g. [15].

3. PAIRWISE CALIBRATION TECHNIQUES

The goal of the pairwise calibration is to find the rigid body transformation that brings points of one camera to the reference coordinate frame of the second camera,

$$\mathbf{P}^{(2)} = R\mathbf{P}^{(1)} + t \tag{1}$$



(a) Chessboard (b) Colored plane object

Fig. 3: Calibration objects used in this work.

where R and t are the rotation matrix and translation vector of the rigid body transformation, $\mathbf{P}^{(1)}$ is a point expressed in the coordinate frame of the first camera, and $\mathbf{P}^{(2)}$ is the same point expressed in the coordinate frame of the second camera.

The rigid body transformation can be computed from two corresponding 3D point clouds $\left\{\mathbf{P}_i^{(1)}, \mathbf{P}_i^{(2)}\right\}$ using the Kabsch algorithm [16]. This technique minimizes the least-square loss function

$$\sum_i w_i \left\| \mathbf{P}^{(2)} - R\mathbf{P}^{(1)} - t \right\|^2 . \tag{2}$$

We use this algorithm, combined to a RANSAC procedure, to estimate the rigid body transformation.

3.1. Technique 1: chessboard pattern

Our first pairwise calibration technique is the one implemented in the OpenCV library [3]. This technique is based on gray-level images and uses a calibration pattern such as a chessboard (see Figure 3a) to establish point correspondences between the images. Note that we use the RGB image of the Microsoft Kinect and the intensity image of the PMD CamCube 2.0 to detect the corners of the chessboard.

This technique is based on epipolar geometry [10] and performs a non-linear minimization of the reprojection error of the detected corners. The final reprojection error is usually smaller than one pixel.

3.2. Technique 2: plane

The second technique is based on a colored plane (see Figure 3b). We use both the texture and the depth information to perform the pairwise calibration. The pixels belonging to the plane are detected in the color image by a floodfill algorithm using the color of the plane.

We obtain the 3D point correspondences $\left\{\mathbf{P}_i^{(1)}, \mathbf{P}_i^{(2)}\right\}$ by estimating the center of the planes in both images using

$$\mathbf{P}_i^{(k)} = \frac{\sum_i Z^{(k)}\left(\mathbf{p}_i\right)^2 \mathbf{P}^{(k)}\left(\mathbf{p}_i\right)}{\sum_i Z^{(k)}\left(\mathbf{p}_i\right)^2} , \tag{3}$$

where $Z^{(k)}\left(\mathbf{p}_i\right)$ is the depth value of the pixel $\mathbf{p}_i$ for camera k and $\mathbf{P}^{(k)}\left(\mathbf{p}_i\right)$ the 3D point corresponding to the pixel $\mathbf{p}_i$. By weighting each point with its squared depth, we take into consideration the fact that a larger distance between a pixel and the camera correspond to a larger surface encompassed by the pixel in the scene. In other words, the density of the 3D points decreases with their depth.

3.3. Technique 3: movement detection

The third technique is based on movement detection. We track the objects in movement in the two cameras and establish the point correspondences from the center of mass of the observed objects. The processing pipeline used for movement detection and matching is summarized in Figure 4. We perform a background subtraction on the images to retrieve the various objects in the scene. We model them using a multivariate normal distribution. We track the objects by using a similarity measure and their last position. The objects are matched between the two cameras using a similarity measure.

3.3.1. Background subtraction

For background subtraction, we base our technique on a simple background model where we record the largest depth seen at each pixel of the image. The depth background $B_Z^{(k)}\left(\mathbf{p}\right)$ of camera k at pixel $\mathbf{p}$ can be expressed as

$$B_Z^{(k)}\left(\mathbf{p}\right) = \max_j \left(Z_j^{(k)}\left(\mathbf{p}\right) \right), \quad j < N_{BG} , \tag{4}$$

where $Z_j^{(k)}$ is j-th range image captured by camera i and N_{BG} is the number of images used to learn the background. The foreground $F^{(k)}$ is obtained by simply comparing the current range image with the background model,

$$F^{(k)}\left(\mathbf{p}\right) = \begin{cases} \text{true} & \text{if } Z^{(k)}\left(\mathbf{p}\right) \text{ is valid and} \\ & \left| B^{(k)}\left(\mathbf{p}\right) - Z^{(k)}\left(\mathbf{p}\right) \right| > \lambda \sigma^{(k)}\left(\mathbf{p}\right) \\ \text{false} & \text{otherwise,} \end{cases} \tag{5}$$

where $\sigma^{(k)}$ is an image indicating the relative level of noise at each pixel, λ is a parameter of the technique. For the Microsoft Kinect, we use $\sigma_{kinect}^{(k)}\left(\mathbf{p}\right) = \left(Z^{(k)}\left(\mathbf{p}\right)\right)^2$ and $\lambda_{kinect} = 0.05$. For the PMD CamCube 2.0, we use $\sigma_{tof}^{(k)}\left(\mathbf{p}\right) = \left(A^{(k)}\left(\mathbf{p}\right)\right)^{-1}$ and $\lambda = 70$, where A denotes the amplitude image given by the camera.

There can be noise in the foreground mask, especially around the edges in the case of the Microsoft Kinect. To remove false positive and segment the foreground in its different part, we use a connected component analysis algorithm on the foreground mask and the depth. We then reject the smallest components. Our connected component algorithm is based on the floodfill algorithm: two neighboring pixels which are in the foreground mask and whose depth difference is below a

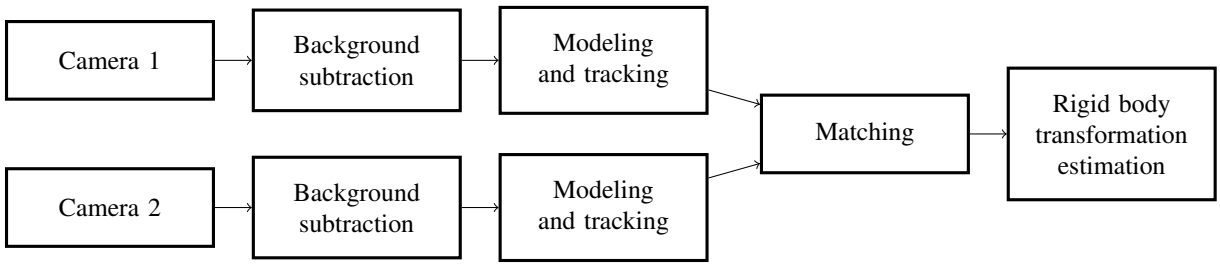


Fig. 4: Processing pipeline for pairwise calibration based on movement detection.

certain threshold are in the same component. The connected component analysis also permits to correctly separate two objects crossing each other.

3.3.2. Modeling and similarity measure

We model an object $\mathcal{O}_i$ by its center of mass μ_i and its covariance Σ_i. These parameters constitute a multivariate normal distribution. As a single image is only able to capture the front side of the object and not its whole volume, the estimated parameters will be biased towards the camera. To reduce this bias, we compute μ_i only using the border pixels of the object. Indeed, for cylinder shaped object, the borders will be symmetrically distributed around the center of mass. The center of mass and the covariance are estimated by weighting pixels according to their local density as in Equation (3).

To track an object within a single camera, we use an estimation of the speed of the object to predict its distribution in the next image, and we use the Kullback-Leibler divergence as a similarity measure.

As the position of the objects in the two cameras can be significantly different, we perform an inter-camera matching by comparing only the covariance of the objects. The centers of mass of the objects matched between the two cameras are used to make a new point correspondence for pairwise calibration.

3.3.3. Misalignment detection

During the operation of the cameras, they can become misaligned. For example, this can happen when the cameras or their support are bumped into or when they are moved inadvertently. Our third calibration technique can then be used as a basis for misalignment detection.

We constantly add the new detected correspondences while we remove those which are outdated. Then, we compare the last ground truth rigid body transformation to the one estimated using the detected movement. To do so, we use the translation and angular error between the transformations. Denoting by $(R_{GT},\, t_{GT})$ the transformation between the cameras, and by $(R^\star,\, t^\star)$ the transformation estimated using the movement, a mis-alginment is detected if

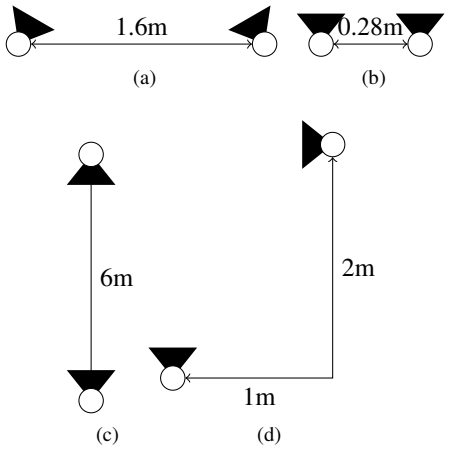


Fig. 5: Configurations of the cameras for the four tested scenarios.

$$t_{err} = \|t_{GT} - t^\star\| > t \tag{6}$$

or

$$R_{err} = \left\|\log\left(R_{GT}^T R^\star\right)\right\|_F > \theta\,, \tag{7}$$

where t_{err} is the distance between the translations, and R_{err} the angular error. The angular error is a geodesic distance between the two rotation matrices [11]. Its value lies between 0 and π. We discuss the possible value of the thresholds t and θ in Section 4.

4. EVALUATION

We have tested the techniques for several configurations of cameras (see Figure 5). Configuration (a) is a usual stereoscopic setting with converging cameras; (b) is a close parallel stereoscopic setting; in configuration (c), the cameras face each other; in (d), they are rotated by 90° from each other. Table 1 presents the results for a configuration with two Microsoft Kinect cameras, and Table 2 presents the results when two different cameras are used (a Microsoft Kinect and a PMD CamCube 2.0 camera). The results were obtained

Configuration	(a)	(b)	(c)	(d)
	Translation error (in meter)			
Technique 1	0	0	0	0
Technique 2	0.094	0.057	0.047	0.155
Technique 3	0.076	0.069	0.128	0.189
	Angular error (in degree)			
Technique 1	0	0	0	0
Technique 2	2.59	1.34	0.45	5.53
Technique 3	2.49	1.37	1.96	3.58

Table 1: Errors obtained with two Microsoft Kinect cameras compared to a pairwise calibration with a chessboard pattern.

Configuration	(a)	(b)	(c)	(d)
	Translation error (in meter)			
Technique 1	0	0	0	0
Technique 2	0.176	0.042	0.044	0.064
Technique 3	0.10	0.223	0.169	0.339
	Angular error (in degree)			
Technique 1	0	0	0	0
Technique 2	4.42	4.94	0.54	1.61
Technique 3	3.46	2.88	1.95	11.28

Table 2: Errors obtained with one Microsoft Kinect and one PMD CamCube 2.0 compared to a pairwise calibration with a chessboard pattern.

on sequences with duration comprised between one and two minutes.

With two Microsoft Kinect cameras, techniques 2 and 3 have comparable results. We see that configuration (d) yields the worse estimation. We believe that this occurs because the overlap between the fields of vision of the cameras is the smallest in this scenario.

With different cameras, technique 2 performs better than technique 3, which leads to larger translation errors. This may be explained by the fact that the movement extraction algorithm failed more often with the PMD CamCube 2.0, partly because areas of the scene are located at a depth larger than the camera unambiguous measurement range, and partly because the angle of view of this camera is smaller (the objects weren't fully contained inside its field of view). It is also probable that a better background subtraction algorithm and preprocessing techniques for the PMD CamCube 2.0 camera would improve the results.

We have attempted to increase the performance of our technique by using an iterative closest point algorithm after a first estimation was obtained. However, our attempts didn't yield significant improvements. In the case of cameras facing each other (configuration (c)), the results were always worse. This can be explained by the lack of corresponding points between the two cameras.

For misalignment detection, the thresholds on the translation and angular error should be chosen larger than the errors reported in Table 1 and 2, according to the configuration of the cameras.

5. CONCLUSION

In this paper, we have just reviewed several techniques for pairwise calibration of two range cameras. The common technique based on a chessboard pattern was compared to a technique based on a colored plane calibration object and a self-calibration technique based on movement detection in the scene. We have shown that the techniques that use the range values for calibration fail to reach the same level of precision as that of the chessboard technique. However, they are able to provide a good approximation in some cases.

We recommend to use the self-calibration technique that uses the observed movement only when it is impossible to calibrate the cameras with a more accurate technique. However, this technique can be used for misalignment detection and can provide a temporary calibration when a misalignment is detected.

Acknowledgment. Antoine Lejeune and David Grogna are under a contract funded by the European Regional Development Fund (ERDF) program of the Walloon Region, Belgium.

6. REFERENCES

[1] D. Alexiadis, D. Zarpalas, and P. Daras. Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Transactions on Multimedia*, 15(2):339–358, February 2013.

[2] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 3, pages 627–632, August 1996.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim. Shake'n'sense: Reducing interference for overlapping structured light depth cameras. In *SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1933–1936, New York, NY, USA, May 2012. ACM.

[5] V. Castaneda. *Constructive interference for Multi-view Time-of-Flight acquisition*. PhD thesis, Technische Universitat München, 2012.

[6] X. Chen, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 520–527, June 2000.

[7] M. Frank, M. Plaue, H. Rapp, U. Kothe, B. Jahne, and F. Hamprecht. Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras. *Optical Engineering*, 48(1), January 2009.

[8] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns, 2010. US Patent Application 20100118123.

[9] D. Glas, T. Miyashita, H. Ishiguro, and N. Hagita. Automatic position calibration and sensor displacement detection for networks of laser range finders for human tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2938–2945, Taipei, Taiwan, October 2010.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[11] D. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[12] S. Izadi, R. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time dynamic 3D surface reconstruction and interaction. In *ACM SIGGRAPH Talks*, Vancouver, Canada, 2011.

[13] S. Kaenchan, P. Mongkolnam, B. Watanapa, and S. Sathienpong. Automatic multiple kinect cameras setting for simple walking posture analysis. In *International Computer Science and Engineering Conference (ICSEC)*, pages 245–249, September 2013.

[14] K. Khoshelham and S. Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, February 2012.

[15] M. Lindner, I. Schiller, A. Kolb, and R. Koch. Time-of-Flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, 2010.

[16] L. Markley. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, 36(3):245–258, July 1988.

[17] S. Piérard, R. Phan-Ba, V. Delvaux, P. Maquet, and M. Van Droogenbroeck. GAIMS: a powerful gait analysis system satisfying the constraints of clinical routine. *Multiple Sclerosis Journal*, 19(S1):359, October 2013. Proceedings of ECTRIMS/RIMS 2013 (Copenhagen, Denmark), P800.

[18] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 145–152, Quebec City, Canada, June 2001.

[19] Y. Schröder, A. Scholz, K. Berger, K. Ruhl, S. Guthe, and M. Magnor. Multiple Kinect studies. Technical Report 09-15, ICG, October 2011.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, Providence, Rhode Island, USA, June 2011.

[21] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[22] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck. Smart pixel-photonic mixer device (PMD): New system concept of a 3D-imaging camera-on-a-chip. In *International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pages 259–264, Nanjing, China, September 1998.