REGISTRATION OF MULTIPLE RGBD CAMERAS VIA LOCAL RIGID TRANSFORMATIONS

Teng Deng[1], *Jean-Charles Bazin*[2], *Tobias Martin*[2], *Claudia Kuster*[2], *Jianfei Cai*[1], *Tiberiu Popa*[3] *and Markus Gross*[2]

[1]School of Computer Engineering, Nanyang Technological University, Singapore
[2]Department of Computer Science, ETH Zurich, Switzerland
[3]Department of Computer Science and Software Engineering, Concordia University, Canada

ABSTRACT

RGBD cameras, such as the Kinect, have recently revolutionized the field of real-time geometry and appearance acquisition. While impressive 3D reconstruction results have been obtained, combining data acquired by multiple RGBD cameras constitutes a technical challenge. Several methods have been proposed to estimate the internal parameters of each RGBD camera (such as depth mapping function and focal length). Despite that the textured geometry obtained by each RGBD camera individually is visually attractive, even state-of-the-art methods have difficulties in correctly combining the textured geometries obtained by several RGBD cameras via a rigid transformation. Based on this observation, our approach registers the RGBD cameras by a *smooth field* of rigid transformations, instead of a *single* rigid transformation. Experimental results on challenging data demonstrate the validity of the proposed approach.

Index Terms— RGBD cameras, virtual view rendering

1. INTRODUCTION

RGBD cameras, such as the popular Microsoft Kinect device, can simultaneously acquire the appearance and the geometry of a scene, leading the way to new exciting applications such as real-time content acquisition for 3D telepresence [1, 2] and real-time novel view synthesis [3], among many others. However, a key limitation of such devices is that they provide only 2.5D geometry (depth maps) and cannot reconstruct an entire scene particularly in the presence of occlusions, a critical practical requirement for many applications. At first glance, this major limitation can be easily mitigated by using several RGBD cameras simultaneously. The data streams from these devices can be, in principle, registered in the same coordinate frame using optical [4] or geometric [5] methods. They compute for each camera a global rigid transformation (rotation and translation) aligning the textured geometries into a common coordinate frame. However, this registration process is noted by many to be a very delicate and sensitive procedure [3, 2].

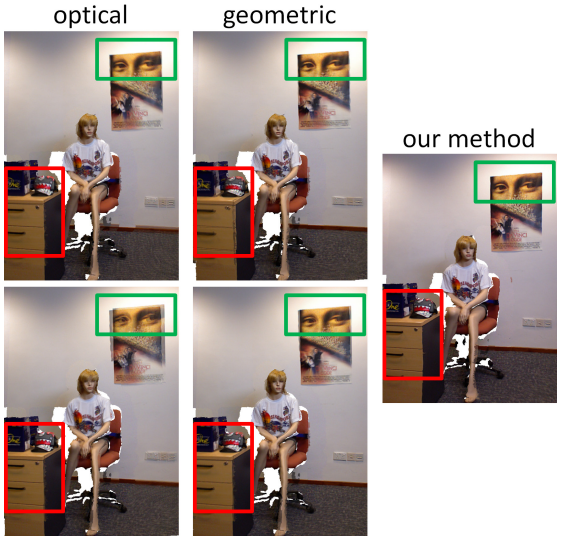

Fig. 1. Single rigid transformation methods for multiple RGBD cameras obtained with the optical calibration [4] and with the geometric ICP algorithm [5] generally provide a correct registration only for a certain region, such as the background (green, top row) or the foreground (red, bottom row), but not all the regions. In contrast, our method (right) provides a correct registration for the whole captured scene.

Our experiments support this observation and show that even with a very carefully executed registration procedure using state-of-the-art methods, the geometry and color do not align consistently everywhere. Indeed in some areas the alignment might be satisfying but not in others, as illustrated in Figure 1. This is an important observation as it tends to suggest that there might not exist a (single) rigid transformation that can correctly align the data streams from multiple RGBD devices.

The reasons why this is the case are complex. First, multi-RGBD camera setups do not require only geometry alignment but combined geometry and texture alignment which makes the task challenging. Second, current RGBD cameras, such as the Kinect, consist of a color camera, an IR camera (called depth

Fig. 2. Registration of two textured geometries acquired from two RGBD cameras (Kinects) by a single 3D rigid transformation (e.g., obtained by the popular ICP algorithm) with (a) the internal parameters of the RGBD cameras provided by the manufacturer, and with (b) the internal parameters computed by Herrera's calibration toolbox [6]. (c) The proposed 3D registration field with the internal parameters provided by the manufacturer. This comparison shows that, in contrast to single 3D rigid transformation-based methods, our proposed approach provides a visually appealing alignment even when using low accuracy internal (e.g., manufacturer's) parameters.

camera) and an IR projector. They require a careful calibration, but calibrating such a complex device is a complicated task and thus highly error prone. Small calibration errors can result in erroneous 3D data that, in turn, cannot be rigidly aligned to 3D data acquired by another device, as illustrated in Figure 1. While these calibration errors can be visually acceptable when only one RGBD camera is used, they become immediately apparent in multi-RGBD camera setups resulting in severe visual artifacts. Third, the proprietary implementation of these cameras contains additional processing, especially on the depth stream, that are unknown to the user. Therefore, even though several noise modeling methods have been proposed [6], it is difficult to define and fit a realistic physical model that could be used to obtain consistent 3D data.

In this paper, we propose an approach to register data obtained from a multi-hybrid camera setup that uses a non-rigid alignment strategy making the method more robust and independent of technology and manufacturer. Our approach locally computes the parameters that can accurately register the geometry and color streams and interpolates these values across the entire captured volume. This approach provides a more visually appealing alignment than competing techniques (see Figure 2), is easy to perform, and efficient to use in practice.

2. RELATED WORK

Current RGBD cameras output a color image and the associated depth map. To obtain a textured geometry, internal calibration must be conducted for each RGBD camera. First, the depth map values must be converted to metric units and the 3D points computed using the depth camera parameters [6]. This process is related to intrinsic calibration. Second, the rigid transformation between the depth camera and the color camera must be estimated to align the geometry to the texture. This step corresponds to extrinsic calibration.

Both intrinsic and extrinsic parameters of RGBD cameras can generally be obtained from the manufacturer via the drivers/API as each device is factory calibrated. In practical terms, the drivers/API can provide a depth map that is directly aligned with the color image. However the manufacturer parameters are generally not accurate and many methods have been proposed to estimate them more precisely. We apply the state-of-the-art method of Herreta et al. [6] that can perform a complete device calibration.

In order to use data acquired by a multi-device setup, it is also necessary to register the data streams from all devices in a common coordinate system. This is typically achieved by following either an optical approach (in a similar way to extrinsic calibration using the color streams) or a geometric approach (using the depth streams). The optical approach uses a calibration pattern (typically a checkerboard) that is observed by the cameras of the setup. A very popular tool for stereo-camera systems is the Camera Calibration Toolbox [4]. This optical approach has been followed, for example, by Maimone and Fuchs [7] to register their multi-Kinect setup. The geometric approach aims to find the rigid transformation that aligns the 3D point clouds obtained by each RGBD camera. The most popular technique is the Iterative Closest Point (ICP) algorithm [5] and many variants have been proposed (see [8] for a recent review). In case some correspondences of 3D points are known, one can directly estimate the rigid transformation [9]. This geometric approach has been followed by Izadi et al. [10] in the context of a moving Kinect.

In this work, we compare our method to the (single) rigid transformations obtained by optical and geometric approaches, and also demonstrate experimentally that we outperform state-of-the-art methods.

3. PROPOSED APPROACH

While in theory, a single global rigid transformation for each device should be sufficient to register multiple data streams

to the same coordinate system, in practice, even using state-of-the-art calibration methods, the alignment from this global registration is inaccurate. Therefore, rather than assuming a global rigid transformation, we model the registration in a spatially variant way. Therefore, transforming a point $\mathbf{x}$ from camera A into camera B is a function of $\mathbf{x}$, i.e.,

$$\mathbf{x}' = \mathbf{R}_\mathbf{x}\,\mathbf{x} + \mathbf{T}_\mathbf{x}. \tag{1}$$

We represent the field of translations, $\mathbf{T}_\mathbf{x}$, and the field of rotations, $\mathbf{R}_\mathbf{x}$, by first placing a rectangular regular grid of $n \times m \times o$ vertices within camera A. Then, for each vertex $\mathbf{v}_{ijk}$, where $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $k = 1, \ldots, o$, we estimate a rotation coefficient $\mathbf{r}_{ijk}$ and a translation coefficient $\mathbf{t}_{ijk}$, which are tuned to accurately transform $\mathbf{v}_{ijk}$ into camera B. The computation of these transformation coefficients is conducted as a preprocess and is discussed in more detail in Section 3.1. Then, a linear basis is imposed on the grid to reconstruct the smooth rotation and translation field, both of which are $C^{(0)}$ continuous.

Due to the linear basis, the evaluation of the amount of rotation $\mathbf{r} := \mathbf{R}_\mathbf{x}$ and translation $\mathbf{t} := \mathbf{T}_\mathbf{x}$ boils down to first identifying the voxel which contains $\mathbf{x}$. Then, a parameter (u, v, w) is computed from the eight corresponding voxel vertices $\mathbf{v}_{ijk}$. Note that u, v, and w vary between 0 and 1. By referring to Figure 3, $\mathbf{r}$ and $\mathbf{t}$ are computed by first interpolating the respective transformation coefficients pair-wise along u. The resulting values are then interpolated along v. And finally, the resulting two values are interpolated along w, yielding $\mathbf{r}$ and $\mathbf{t}$, respectively.

While trilinear interpolation is used to interpolate translation quantities, rotations have to be interpolated over the sphere to achieve constant-speed motion. This is achieved by the *slerp* operation, defined as

$$slerp(t, \mathbf{r}_0, \mathbf{r}_1) = \frac{\sin((1-t)\alpha)}{\sin(\alpha)}\mathbf{r}_0 + \frac{\sin(t\alpha)}{\sin(\alpha)}\mathbf{r}_1 \text{ with } t \in [0, 1] \tag{2}$$

which linearly interpolates between the two quaternions $\mathbf{r}_0$ and $\mathbf{r}_1$, respectively, and where $\cos(\alpha) = \mathbf{r}_0 \cdot \mathbf{r}_1$. More information on the *slerp* operation is provided in [11].

The next section discusses the computation of the two 3D lattices of rotations and translations, and proposes an algorithm which guarantees that our method is always at least as accurate and in most cases significantly more accurate in registering the geometry of multi device data as methods that use a global rigid transformation.

3.1. Computation of Transformation Coefficients

The choice of the coefficients $\mathbf{r}_{ijk}$ and $\mathbf{t}_{ijk}$ significantly affects the quality of alignment of the camera data, hence, accurate computation of these coefficients is key. We first sample the scene with a checkerboard that is observed by both cameras. A 3D location is determined for each checkerboard corner which is local to the respective camera, i.e., for each checkerboard corner we have a pair of points $(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}$ is the corner as seen in camera A, and $\mathbf{x}'$ is its corresponding corner as seen in camera B. We propose an adaptive scheme to compute $\mathbf{r}_{ijk}$ and $\mathbf{t}_{ijk}$ as follows:

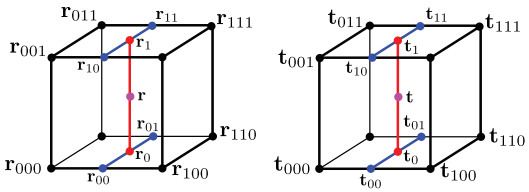

Fig. 3. Given parameter (u, v, w) for sample point, $\mathbf{x}$, 7 *lerp* operations are performed to compute the amount of translation at $\mathbf{x}$ from the eight translation coefficients $\mathbf{t}_{ijk}$. The amount of rotation is computed accordingly, by performing 7 *slerp* operations given the quaternions $\mathbf{r}_{ijk}$.

1. Consider a set of nested cubic regions with increasing sizes centered all around $\mathbf{v}_{ijk}$.
2. For each region l, construct a set $\mathcal{C}^l$ containing checkerboard corner correspondences, $(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}$ is located within the respective region.
3. For each of these regions, estimate the rotation $\mathbf{r}^l_{ijk}$ and translation $\mathbf{t}^l_{ijk}$ using the rigid transformation estimation method in [9].
4. Pick $\mathbf{r}^l_{ijk}$ and $\mathbf{t}^l_{ijk}$ that produces the smallest reprojection error. The reprojection errors are evaluated on the pairs of corners in the respective set $\mathcal{C}^l$.

By construction, the reprojection error obtained with our method is never higher than the reprojection error obtained using the transformation computed from all the samples. In fact, in the result section we demonstrate that our reprojection error is frequently much lower.

4. RESULTS AND DISCUSSION

4.1. Setup

Our acquisition setup is composed of two rigidly attached Kinects. To reduce the interference due to the IR projectors, we attach a small vibrator to each Kinect [12]. As discussed in Sections 1 and 2, each RGBD camera must be internally calibrated to obtain appropriate textured geometry, and then the transformations between the RGBD cameras are needed to register the textured geometries in a common coordinate system. For the internal calibration, we use the parameters provided by the manufacturer as well as obtained from Herrera's state-of-the-art toolbox [6]. In our experiments, the toolbox provides very low reprojection/residual errors in both the color images and depth space, namely, less than 0.7 pixel and less than 0.8 kdu (Kinect disparity units, see [6]) for each Kinect.

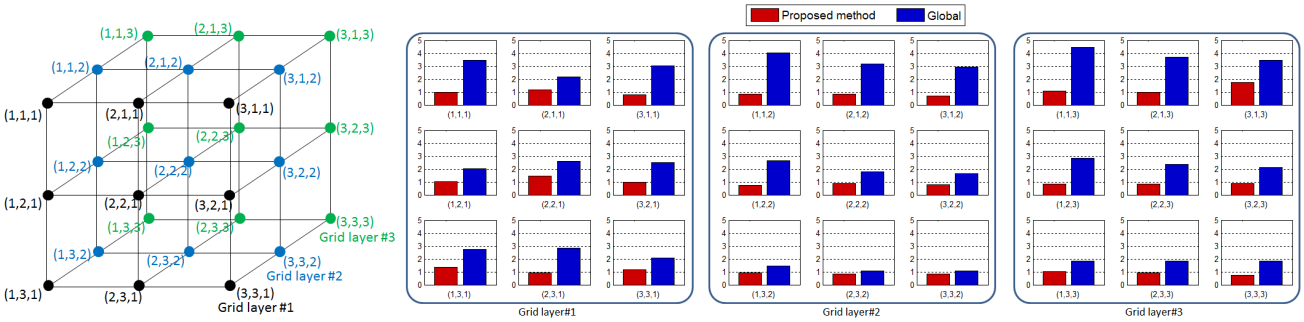

Fig. 4. Comparison of the average reprojection error (in pixels) in both color images obtained by a global registration and the proposed method at $3 \times 3 \times 3$ locations uniformly distributed in the scene. For an easier visualization, the (x, y, z) locations are illustrated on a 3D grid.

We compare our registration approach against both optical and geometric approaches (see Section 2). For the optical approach, we apply the popular Camera Calibration Toolbox for Matlab [4], in a similar way to Maimone and Fuchs [7]. A checkerboard is observed by both cameras, and the checkerboard corners as well as their correspondences between the cameras are extracted semi-automatically (see [4]). To avoid biasing of the registration for a certain part of the scene (see discussion in Section 1), we move the checkerboard at many different orientations and locations in the whole volume of interest. For the geometric approach, we apply ICP [5] and acquire several 3D geometries by moving the setup at different poses. Since ICP depends on the initial solution, we give the best possible initial alignment manually and then run the ICP method available in Point Cloud Library. To apply our approach, we use the same checkerboard data as acquired for the optical approach and run the algorithm presented in Section 3.1. The grid covers the volume of interest and its size is set to $3m \times 3m \times 3m$ with $20 \times 20 \times 20$ regular voxels.

4.2. Rendering

To render the scene from a virtual camera position, the depth map of each RGBD camera is tesselated and vertices of the triangles are transformed by either a single rigid transformation, or by the proposed approach. These triangle meshes are textured accordingly. The textured geometries of each RGBD camera are projected into the virtual view, and the color of the pixel in the final view is the average of the projected textures. If a pixel is visible by only one camera, we use the color observed by that camera. This type of blending is chosen intentionally to better visualize the quality of the alignment.

The proposed method is computationally inexpensive and runs in real-time on the GPU, taking about 18ms per frame for two RGBD cameras at full resolution (i.e., 640×480 depth maps). The execution time linearly scales with the number of devices by pairwise registration. An important observation is that the execution time depends on neither the volume size nor the grid resolution: First, the regularity of the grid allows for efficient lookup of the coefficients for a given query point, where a lookup is performed in constant time. Second, the time to compute the transformation of a query point only depends on the eight coefficients and thus runs in constant time as well.

4.3. Evaluation and Discussion

In the following, the global transformation is computed by ICP only, as it yields comparable quality to an optical method. A visual comparison between a global 3D rigid transformation and the proposed rigid transformation field is shown in Figure 2. It can be seen that our approach leads to visually appealing images in which the textures are correctly aligned.

To quantitatively measure the results, we consider the reprojection error in the images. We prefer this measurement over the 3D registration error simply because the reprojection reflects the visual quality. We apply the following procedure to compute the reprojection error. The N corners of a checkerboard are observed by each of the M RGBD cameras. Their 2D coordinates are obtained by corner extraction in each color image and their 3D coordinates are provided by the corresponding depth image. These 3D corners can be reprojected in the image plane of any RGBD camera. In the case where noise is absent, these reprojected points lie at the exact same location as the extracted corner in the image. However, due to noise, these points do not perfectly coincide and the distance between them is not zero. We compute the registration error in image space as the mean distance between the corresponding corners. The reprojection error from the k-th RGBD camera to the j-th RGBD camera is:

$$\frac{\sum_{i=1}^{N} \| p_i^j - p_i^{k \to j} \|}{N} \tag{3}$$

where p_i^j is the 2D location of the i-th corner point extracted in the color camera of the j-th RGBD camera and $p_i^{k \to j}$ represents p_i^k which is reprojected in the color image of the j-th RGBD camera. If $k = j$, the error is zero, so we do not consider self-reprojection. Summing over each pair of RGBD

Fig. 5. Novel viewpoints rendered by a virtual camera. The full motion of the virtual camera is available in the accompanying video.

cameras, the overall reprojection error is computed as:

$$\frac{\sum_{j=1}^{M}\sum_{k=1,k\neq j}^{M}\sum_{i=1}^{N}\|p_i^j - p_i^{k\to j}\|}{M(M-1)N} \quad (4)$$

A single rigid transformation obtains an average reprojection error of 2.5 px (with a std. dev. of 0.8). In contrast, our method provides an average of 0.8 px (with a std. dev. of 0.2), which is an improvement of about 68%.

We further analyze the error distribution within the volume of interest. We compute the average reprojection error (see Eq. 4) of the checkerboard corners located inside a local volume of size $20 \times 20 \times 20$ cm and centered at each grid vertex, using both the proposed method and a global rigid transformation. Results are illustrated in Figure 4. For better visibility, we display the error computed on a uniform $3 \times 3 \times 3$ subsampling of the grid. It shows that our reprojection always outperforms the global registration approach. It also illustrates the fact that the error obtained by the global registration greatly varies with respect to the location in the scene, in agreement with the results in Figure 1.

In addition, we study the reprojection accuracy with respect to the grid resolution. The average reprojection error is 1.3 px for a $5 \times 5 \times 5$ grid, 1.1 px for a $10 \times 10 \times 10$ grid, 0.8 px for a $20 \times 20 \times 20$ grid and 0.8 px for a $40 \times 40 \times 40$ grid. This indicates that the accuracy increases with the grid resolution until converging to 0.8 px. For all the results shown in this paper, we use a $20 \times 20 \times 20$ grid (same accuracy as a $40 \times 40 \times 40$ grid but with fewer coefficients in memory).

Fig. 6. Representative results on a dynamic scene.

4.4. Additional Results

In the context of free-viewpoint video, the pose (location and orientation) of the virtual camera can be controlled interactively by the user or via a head/eye tracking system, for example in combination with autostereoscopic displays [7]. To study the quality of the results in this context, we move the virtual camera between the physical cameras and render the scene at these new viewpoints. Still images of the novel viewpoints are available in Figure 5. We invite the readers to view the full camera motion in the accompanying video. It shows that our method provides visually appealing results for a range of virtual camera poses. Note that our goal is not to compete with systems dedicated to free-viewpoint video such as [3, 13], but to provide a visually appealing alignment that can then be combined with free-viewpoint video techniques such as silhouette refinement and advanced blending (see [13]).

Our approach can also be applied to dynamic scenes. To illustrate this, we acquire sequences with moving subjects. We process each frame sequentially by directly applying the method described in Section 3. Figure 6 illustrates a representative result obtained with a person walking. This example demonstrates that our method can correctly deal with dynamic scenes. In an additional experiment, we capture a dynamic scene with an acquisition setup installed on a mobile platform. Results are available in Figure 7 and show that our approach can also be applied in a mobile acquisition context.

Our approach can deal with more than two RGBD cameras. In practice, we apply it pairwise with respect to a given reference camera. We use a single calibration grid and run the method of Section 3.1 for each pair of devices, i.e., the coefficients of the grid vertices encode the rotation and translation associated with each pair. Figure 8 shows a representative result obtained by a setup composed of three Kinects.

4.5. Limitations

While a single rigid transformation does not modify the captured geometry (it just "moves" it), rigid transformation fields

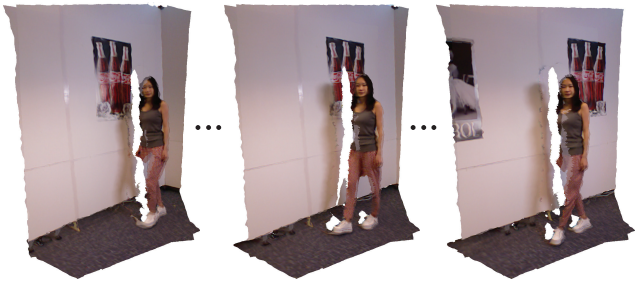

Fig. 7. Representative results on a dynamic scene acquired by a moving setup.

Fig. 8. Left: Representative result with three Kinects. Right: color coded visualization of the overlap of the three Kinects.

might change the geometry. In practice, we have not observed any visual distortion of the geometry resulting from our smooth transformations in the experiments. For quantitative evaluation, we considered the two most distant corners of each checkerboard and measured the distance between their 3D points for all the captured checkerboards in the entire space. The average distance obtained by a single rigid transformation and the proposed approach are both 72.6cm, which shows the distortion is quasi-inexistent.

Our current method follows a pairwise registration with respect to a reference camera. While in theory the contents of the auxiliary cameras might not perfectly align, experiments show that the results are visually appealing. The quality could be further refined by a global registration procedure in post-processing, which considers all the cameras simultaneously in a way similar to bundle adjustment.

5. CONCLUSION

Registration of multi RGBD camera setups is a notoriously sensitive and delicate procedure despite the variety of available methods. In particular, the registration between RGBD cameras which typically consists of a (single) rigid transformation often fails in practice to faithfully align all the depth and color streams consistently everywhere inside the capture volume, as observed in many experiments. We address this issue by proposing a practical and general approach to register the depth and color streams of multi RGBD camera setups. Our approach estimates a smooth field of rigid transformations between RGBD cameras with $C^{(0)}$ continuity on a regular grid within the captured volume. In terms of both visual quality and measurable reprojection errors, we demonstrate that our method provides better results than state-of-the-art methods. In future work, we plan to investigate our proposed methodology on other range sensors, especially time-of-flight cameras.

Acknowledgement. This research, which is carried out at BeingThere Centre, is supported by Singapore MoE AcRF Tier-1 Grant RG30/11 and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

6. REFERENCES

[1] A. Maimone and H. Fuchs, "Real-time volumetric 3D capture of room-sized scenes for telepresence," in *3DTV-CON*, 2012.

[2] C. Kuster, N. Ranieri, Agustina, H. Zimmer, J.-C. Bazin, C. Sun, T. Popa, and M. Gross, "Towards next generation 3D teleconferencing systems," in *3DTV-CON*, 2012.

[3] C. Kuster, T. Popa, C. Zach, C. Gotsman, and M. Gross, "FreeCam: a hybrid camera system for interactive free-viewpoint video," in *VMV*, 2011.

[4] J.-Y. Bouguet, "Camera calibration toolbox for Matlab," `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *PAMI*, 1992.

[6] D. Herrera, J. Kannala, and J. Heikkila, "Joint depth and color camera calibration with distortion correction," *PAMI*, 2012.

[7] A. Maimone, J. Bidwell, K. Peng, and H. Fuchs, "Enhanced personal autostereoscopic telepresence system using commodity depth cameras," *Computers & Graphics*, 2012.

[8] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, 2013.

[9] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, 1987.

[10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *ACM Symposium on User Interface Software and Technology*, 2011.

[11] K. Shoemake, "Animating rotation with quaternion curves," in *SIGGRAPH*, 1985.

[12] D. 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*, 2012.

[13] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *SIGGRAPH*, 2004.