Volumetric Video Capture Using Unsynchronized, Low-Cost Cameras

Andrea Bönsch[1], Andrew Feng[2], Parth Patel[2] and Ari Shapiro[2]
[1] *Visual Computing Institute, RWTH Aachen University, Aachen, Germany*
[2] *Institute for Creative Technologies, University of Southern California, Los Angeles, United States of America*
boensch@vr.rwth-aachen.de, {feng|shapiro}@ict.usc.edu, parth@usc.edu

Keywords: avatar, video capture, photogrammetry, low-cost, animation

Abstract: Volumetric video can be used in virtual and augmented reality applications to show detailed animated performances by human actors. In this paper, we describe a volumetric capture system based on a photogrammetry cage with unsynchronized, low-cost cameras which is able to generate high-quality geometric data for animated avatars. This approach requires, inter alia, a subsequent synchronization of the captured videos.

1 MOTIVATION

Recent advances in virtual and augmented reality technology has increased the demand for authentically animated 3D avatars as personalized, digital doubles of human subjects. To generate photo-realistic avatar models in a fast and cost-efficient manner, photogrammetry cages with low-cost cameras and a traditional 3D pipeline for the avatar mesh generation proved to be beneficial. Extending the pipeline by rigging, skinning, and animation allows to use the resulting avatar models as realistic and interactive doubles in 3D environments (Andrew et al., 2017). However, seeing a virtual double of a known human subject raises expectations with respect to precise body movements, mimics, and overall behavior. Thus, the avatar needs to be animated authentically with respect to its real-life counterpart. Capturing and processing sufficient data in order to recreate such a detailed and authentically animated human model in 3D is a non-trivial and time-consuming task, including the capture and processing of numerous facial expressions, and simulation of clothing and hair. By contrast, volumetric video can preserve detailed performances, including facial expressions, hair movements, and clothing folds without the need to understand and thus to model and simulate such systems.

Photogrammetry cages designed to capture human bodies typically use over 100 cameras in close proximity to the captured subject in order to maximize geometric and texture quality from the reconstruction process. We explored whether such a cage can be utilized for volumetric video capture of a human subject's performance as well. In this work, we present our resulting volumetric capture pipeline derived from a low-cost, Raspberry Pi-based photogrammetry cage that uses unsynchronized cameras. Our results indicate, that the animated reconstruction geometric quality is high, and the overall visual quality is sufficient to show detailed facial expressions of the captured subject (see Fig. 1) including hair movement and clothing folds.

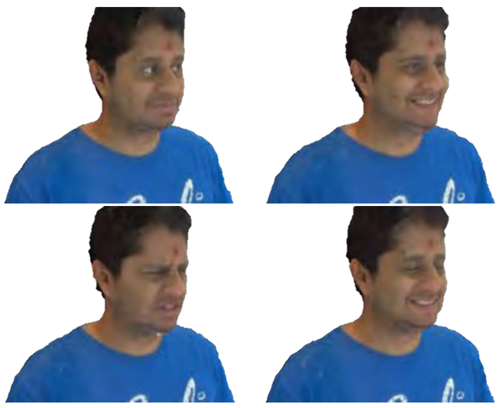


Figure 1: 3D volumetric reconstruction results showing different facial expressions, gaze lines, and skin wrinkles.

2 BACKGROUND

Creating free viewpoint video of humans has been an important yet challenging research problem. Such capture systems usually involve multi-camera setups with carefully designed capture environments. The pioneer work by Carranza et al. (Carranza et al., 2003) utilized a deformable body model and fit the model into multiple silhouette images of an actor to reconstruct the 3D performance. The work by Vlasic et al. (Vlasic et al., 2008) also applied a similar skeletal deformable model to fit into a multi-view visual hull. Casas et al. (Casas et al., 2014) proposed a novel data representation to efficiently store and render the model with view-dependent textures. The work by Collet et al. (Collet et al., 2015) is a capture system using infrared and synchronized cameras. They also deal with the topology changes of the captured sequences by tracking and encoding the texture meshes at suitable keyframes. In a recent approach, Achenbach et al. (Achenbach et al., 2017) propose a multi-view stereo reconstruction process using separate scanners for the subject's body and face, each equipped with single-shot, synchronized cameras. Based on nine manually placed landmarks, the captured data is then fitted to a human template model in order to animate the obtained avatar based on separately captured motion data.

More recently, with the ubiquity of RGB-D sensors from Microsoft Kinect, Occipital Structure Sensor, or iPhone X, it becomes much easier to obtain a 3D reconstruction of a static object. However, to capture a dynamic scene, multiple sensors are usually required. Dou et al. (Dou et al., 2013) utilize eight Kinect sensors and a deformable template model to capture dynamic human performances. Their follow up work (Dou et al., 2016) tracks the mesh across different cameras and frames to reconstruct a temporally consistent mesh without using a template model. The work by Wang et al. (Wang et al., 2016) used a novel pairwise registration method to reconstruct the dynamic textured models of a moving subject with only a small number of handheld RGB-D sensors. Recent research has developed novel methods to reconstruct the dynamic 3D performances without the requirements for multiple sensors. DynamicFusion (Newcombe et al., 2015) first fuses a detailed template model on-the-fly from multiple frames. It then utilizes the non-rigid motion sequences obtained through the non-rigid registration to reconstruct the dynamic performance with detailed geometry. The work by Guo et al. (Guo et al., 2017) improves upon this work by also extracting the albedo and low-frequency lighting from the reconstruction to allow relighting and albedo editing during real-time rendering of 4D video.

Photogrammetry can be used to reconstruct 3D models of real objects, ranging from the very large, such as entire buildings or cityscapes, to small, such as personal objects like shoes and pursues. Many photogrammetric algorithms rely on the assumption that a capture object is rigid, stationary and unmovable. It is difficult for people to stand completely still for any period of time; bodies regularly sway slightly as a consequence of balancing, eyelids blink, and the upper body can change in volume as a consequence of breathing. For this reason, the most accurate 3D reconstruction results come from capture systems that simultaneously capture from multiple locations, such as capture cages.

In contrast to most approaches mentioned before, capture cages do not utilize depth sensors while being build based on low-cost hardware setups (Garsthagen, 2014). Nevertheless, these photogrammetry cages produce results suitable for immersive uses (Andrew et al., 2017). Several studies have used avatars of that quality, including whether game performance is affected by using one's own doppelganger (Lucas et al., 2016) and whether there is a gender difference when using self-similar avatars (Wauck et al., 2018).

In order to reuse existing and approved hardware setups, we extended a low-cost photogrammetry capture cage allowing it to serve two purposes:

1. Traditional Photogrammetry
   Generating realistic, but static 3D avatars as digital doubles of human subjects (Andrew et al., 2017).
2. Volumetric Video
   Obtaining animated 3D avatars reflecting all details of a captured human subject's performance, based on the approach described in this work.

3 SYSTEM OVERVIEW

Our goal is to derive a volumetric capture system tailored to the photogrammetry cage shown in Figure 2. The cage consisting of 104 unsynchronized Raspberry Pi 2 Modules B+, each connected to a Raspberry Pi RGB camera module rev 1.3.

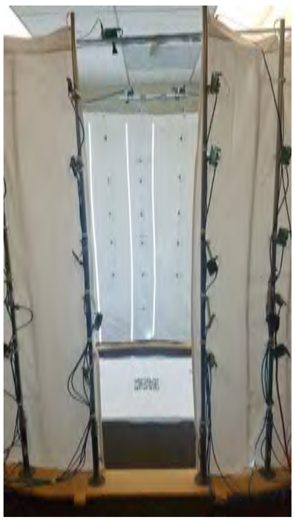
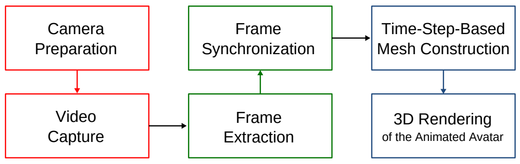


Figure 2: Photogrammetric capture cage consisting of 104 unsynchronized Raspberry Pi cameras; Designed for human body capture and used for volumetric video reconstruction.


Camera Preparation
Frame Synchronization
Time-Step-Based Mesh Construction
Video Capture
Frame Extraction
3D Rendering of the Animated Avatar
Figure 3: Overview of the volumetric capture system pipeline.

Challenges include the video quality, the capture frame rates, as well as the synchronization of the captured videos. We address these issues by dedicated steps in our system's pipeline, which is schematically illustrated in Figure 3. Thus, the proposed system ensures (a) qualitatively sufficient videos with a reasonable frame rate (20-24 fps) from different perspectives (Fig. 3, red steps), (b) a subsequent video synchronization based on two camera flashes (Fig. 3, green steps), and (c) a time-step-based mesh construction using photogrammetry with an ensuing 3D rendering of the meshes to display the subject's animated avatar in a VR or AR environment (Fig. 3, blue steps).

The technical advancement of our approach is surely limited, as we re-use and partially improve existing techniques to configure a proper pipeline. However, the insight gained in our work is valuable for future approaches used to digitalize a human subject's performances.

4 VOLUMETRIC CAPTURE SYSTEM

4.1 Capturing the Subject

Our goal is to record a human subject inside a capture cage to generate a detailed, animated, and personalized avatar for use in 3D environments. To ensure qualitatively sufficient videos for this process, we carefully adjust the Raspberry Pi camera parameters by means of *PiScanner* (Jones, 2013). Thereby, we use identical configurations for all cameras, providing a consistent visual quality and allowing a subsequent video synchronization.

Camera Preparation

In order to generate matching textures for the single time steps of the animated avatar, we first ensure an adequate and constant brightness per video. Therefore, the automatic brightness adjustment is deactivated by disabling the automatic gain of the cameras' video ports. As a consequence, a preparation step is required to adjust the digital and analog gain as they cannot be set directly. Without this preparation, the captured videos will appear black as the automatic camera adjustment uses initially low gain values. Testing indicates that capturing two photos on the video port prior to the video recording results in a suitable brightness and color configuration.

Video Capture

Besides the previously ensured constant brightness, additional demands are placed on the captured videos to allow an animated avatar generation: (i) Using uncompressed file formats prevents handling compression artifacts and issues arising due to the quality reduction of the videos. (ii) A high video resolution with a denoised data stream simplifies the construction process and provides high-quality textures, improving the visual quality of the resulting avatar meshes. (iii) A high frame rate guarantees to capture a complete performance sequence, avoiding discontinuities in the constructed animation. (iv) Synchronization markers support the subsequent video synchronization. However, as some of these parameters mutually influence each other, we had to make several compromises.

Using an uncompressed video format leads to a large amount of data to be stored, limiting the frame rate to about five fps. As this is insufficient for capturing a human's performance, *i* cannot be met. Using the lossy compression h264 instead, reduces the data amount allowing higher frame rates while maintaining a sufficient video quality. As the captured frame rate is not stored here, it has to be manually computed by dividing the recording time by the number of frames captured.

Choosing a high resolution (2592×1944) meeting *ii*, again, reduces the frame rate to an insufficient low amount. Thus, we set the resolution to 1920×1080. Based on the documentation, this allows a frame rate up to 30 fps. However, we observed strong differences in the fps between the single cameras, while none of it reached the 30 fps. By defining the frame rate to be 25 fps for the capture process, we could solve this issue while ensuring a stable and sufficiently high frame rate meeting *iii*.

The cameras' video ports do not use aggressive denoising algorithms. Thus, the video quality obtained by the h264 encoder has to be specified explicitly. Indicated by several tests, 20 is a suitable quality value to meet *iii*. This quality assurance allows reconstructing geometric details from the videos.

For the volumetric reconstruction, the video feeds need to be synchronized among the multiple cameras. Common approaches include hardware synchronization or sound synchronization. However,

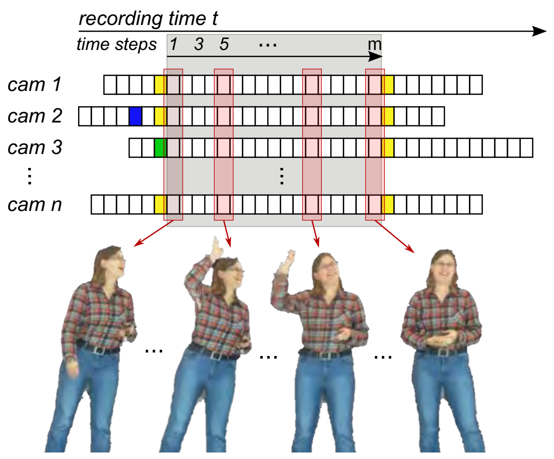


Figure 4: Video synchronization and mesh generation: Two flashes are detected per video (yellow frames). Based on the gathered information on the time span between both flashes via the other videos, false detections are ignored (blue frame) and missing flashes are added (green frame). After the alignment, one mesh is constructed for all m time steps.

since the cameras have no hardware synchronization[1] and no microphones, other synchronization markers have to be introduced. Here, two flashes from a professional photographer's flash are used, denoting the start and end of a subject's performance during the video capture phase. Consequently, the videos are subsequently synchronized based on their frames' luminance histograms, explained next.

4.2 Processing the Videos

After the volumetric capture, the videos' frames are extracted and a computer-vision-based synchronization is used to collect the single frames across all videos per animation time step.

Frame Extraction

An FFmpeg routine is used to extract the frames per video as jpg images. As the chosen file format h264 does not provide the video's actual frame rate, we provide it based on our fps computation. Furthermore, specifying the use of the image file demuxer and setting an appropriate image quality level enhances the resulting images and thus the construction result.

[1]Although the signals to start and end a video capture are sent simultaneously to all cameras, we observed different timings for the actual recording events caused by, e.g., slight delays due to network travel or by different processing times per camera module.

Frame Synchronization

For the video synchronization, two camera flashes during the video capture are used. Besides being the synchronization markers, the flashes denote the start and end of the subject's performance. Short flash times are chosen to minimize their occurrence in the videos to single frames. Aligning these frames, yield the required synchronization, indicated in Figure 4.

The flash detection is based on a computer vision approach presented by Shrestha et al. (Shrestha et al., 2006): a flash is detected when the luminance histograms between two consecutive frames differ significantly. Although the proposed approach was successful in many tests, we experienced false or missing detections in some occasions. To address this issue, we improved the approach by an automatic flash adaption, utilizing information about the time span between both flashes gathered from the videos, in which the correct amount of flashes had been detected (see Fig. 4). Three cases are handled: (1) If three flashes are detected, the most unlikely one is ignored. (2) If only one flash is detected, the frame number of the missing flash's occurrence is estimated based on the gathered information. The flash is added if the respective frame exists. Otherwise, the video is excluded from the reconstruction process. (3) If no flash is detected, the respective video is discarded. Although this extension is simple and straightforward, it enhances the final results.

Based on the adapted flash detection, we can now specify which frames contribute to which time step of the avatar animation (see Fig. 4). Next, the frames in-between the flashes are resorted, all other frames are ignored. The resorting is done time-step-wise by collecting the respective frame per video in a common directory, preparing the avatar generation process.

4.3 Generating the Animated Avatar

Time-Step-Based Mesh Construction

After the video frames are synchronized and resorted, a photogrammetry software, e.g., *Agisoft Photoscan* (Agisoft, 2006), is used for a time-step-based mesh construction (see Fig. 4). Thereby, an embedded mesh clean-up automatically eliminates the captured environment, i.e., the capture cage and the floor, from the final meshes. In addition, the construction step is parallelized across 50 nodes, accelerating the output generation.

In order to facilitate the subsequent avatar rendering, an identical transformation of all meshes is required. Hereto, a camera alignment for the mesh construction is defined by the initial time step. This

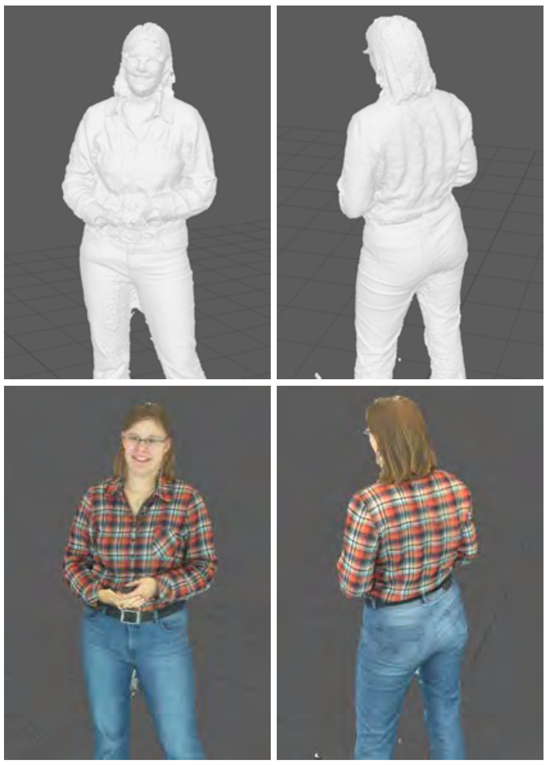


Figure 5: Resulting geometry (top) and fully textured models (bottom) of a typical animated frame of volumetric rendering.

alignment is then shared to reconstruct the meshes of the remaining time steps.

3D Rendering

Finally, all generated meshes are rendered in their chronological order in 3D to display the animated avatar of the recorded human subject, demonstrated in the supplemental material.

5 RESULTS

We tested the volumetric capture system in our capture cage. For the synchronization, we used a Neewer NW670 E-TTL Flash with a wireless trigger. By this, the flashes are triggered from outside the capture cage, reducing the video content to the cage and the subject and thus facilitating the mesh generation of the human subject's performance. A suitable flash configuration in our setup is a 24mm zoom and 0.5 as brightness value of the speedlite in M-Mode.

The output quality of the volumetric data is exemplarily shown in Figure 5. The geometry and the textured renderings have a high level of detail regarding mimics (see also Fig. 1), hair and clothing folds.

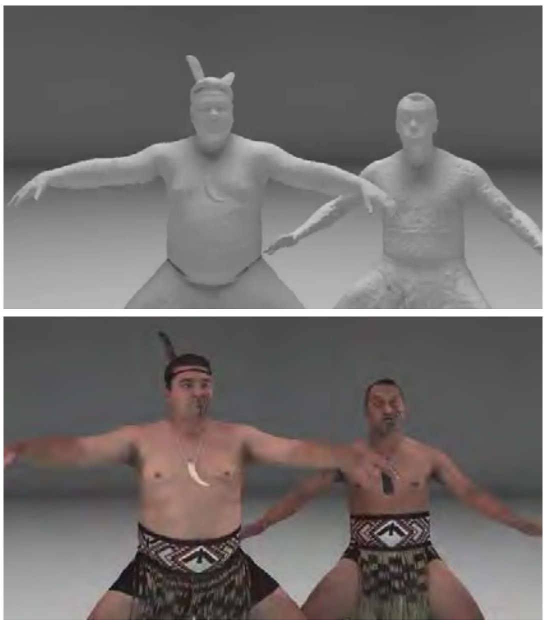


Figure 6: Resulting geometry (top) and fully textured models (bottom) from the volumetric capture in (Collet et al., 2015).

Particularly, when comparing our meshes to those generated by high-quality approaches, our pipeline is a promising low-cost variant. As example, data presented by Collet et al. is given in Figure 6. Their approach uses 106 synchronized RGB and IR video cameras on a greenscreen stage for the volumetric capture, while the mesh generation pipeline is optimized for high-quality, coherent mesh output by an adaptive meshing technique which is guided by automatic detection of perceptually salient areas as well as by a mesh tracking (Collet et al., 2015). As the mesh quality between Collet's and our approach are comparable, we are convinced that volumetric capture systems based on photogrammetry cages with unsynchronized, low-cost cameras provide suitably detailed, animated avatars.

The generation time for an animated avatar, however, is a shortcoming of our system. While the camera preparation step of 20 seconds per volumetric capture is

Table 1: Durations to generate an animated avatar based on 104 videos (fps: 24, length: 30 sec).

#	Description	Duration	
1	camera preparation	20 sec	
2	video recording	30 sec	
3	frame extraction	8.7 min	
4	video synchronization		durations
	a) flash detection	90 min	depend
	b) frame sorting	30 min	on #2
5	mesh generation	120 min	
	Total Duration	4.16 h	

neglectable, the duration of the remaining steps heavily depend on the captured videos' lengths. Table 1 gives the figures of the individual pipeline steps, based on 104 captured videos with 24 fps and a length of 30 seconds. Furthermore, it is assumed that all 720 frames per video are used in the photogrammetry reconstruction, resulting in a total pipeline duration of 4.16 hours. Per time-step of the animation, the mesh generation based on 104 images takes 8.7 minutes, resulting in approximately 104.4 hours on a single computer. As each frame of the video can be reconstructed independently based on the same camera alignment, we parallelized this step on a 50 node cluster. As a consequence, we were able to significantly reduce the mesh generation time to approximately 0.17 seconds per frame, resulting in 2 hours for the complete video, as shown in step 5 of Table 1.

6 LIMITATIONS

The capture space is confined by the photogrammetry cage's footprint, limiting the type of movements which can be recorded. In our cage, subjects can only move within a narrow volume of about $0.8m^3$. Although this is sufficient for animations required in virtual face-to-face conversations or interactions, expanding the available space would enhance the system's use case.

As the capture frame rate of the low-cost camera used is limited, fast movements result in partially blurred frames. As a consequence, rapid moving body parts, such as hands, e.g., during waving, or aspects like flying hair during head movements, can be missing in the final mesh, preventing a detailed digitization of the subject per frame. Thus, the visual quality and naturalness of the resulting, animated avatar might decrease and mesh discontinuities between succeeding frames might occur. However, for most gestures and mimics required in face-to-face interactions (see Fig. 1), the achieved recording speed is sufficient. To use our pipeline also for faster performances two methods are reasonable: (a) An increased capture frame rate would improve the quality of the reconstructed imagery. However, the specific setup of our capture cage restricts the potential frame rates: The placement of the cameras within the cage was designed around a 4:3 image aspect ratio, but the Raspberry Pi cameras used are not capable of capturing in 4:3 at a high frame rate, so a 16:9 aspect ratio resolution was used instead. This results in less overlap between neighboring imagery, thereby negatively affecting the final quality of the reconstructed video. (b) A mesh tracking over the single time steps should be embedded to improve the generated mesh quality. By using the mesh information of previous and succeeding frames, temporally coherent meshes can be created. As a consequence, missing information on body parts or dynamics in hairs and clothes, normally leading to mesh discontinuities, can be supplemented.

Besides enabling a high-quality and detailed reconstruction of fast performances of a captured subject, accelerating the pipeline itself will improve our work. A small speed-up can be achieved by parallelizing the third step of our pipeline, the frame extraction routine (cf. Table 1). Furthermore, and even more essential with respect to a speed-up is the frame sorting (see step 4b in Table 1). In our current design, the mesh generation routine requires one distinct directory with all frames from the different perspectives per time frame. This structure is obtained by the frame sorting routine, copying the single frames based on the detected flashes to the distinct directories. This brute force method should be replaced by a more advanced approach using a designated data structure working, e.g., with index shifts to group the single frames into the individual time steps.

7 SUMMARY

Based on photogrammetry cages utilized to generate a realistic, digital double of a captured human body, we configured a 3D pipeline allowing for volumetric video capture of a human subject's performance. Our results indicate that the pipeline is a first valuable step towards fast and cost-efficient generation approaches of authentic, high-quality animation recording for digital doubles. For a successful future implementation, the pipeline duration has to be further shortened, while an additional mesh tracking step should be embedded to enhance the mesh quality per animation time step.

REFERENCES

Achenbach, J., Waltemate, T., Latoschik, M. E., and Botsch, M. (2017). Fast generation of realistic virtual humans. In *Proc. 23rd ACM Symp. Virtual Real. Softw. Technol. - VRST '17*, pages 1–10.

Agisoft (2006). Agisoft PhotoScan. `http://www.agisoft.com/`. [Online; last-accessed 17-October-2018].

Andrew, F., Suma, R. E., and Ari, S. (2017). Just-in-time, viable, 3-d avatars from scans. *Computer Animation and Virtual Worlds*, 28(3-4):e1769. e1769 cav.1769.

Carranza, J., Theobalt, C., Magnor, M. A., and Seidel, H.-P. (2003). Free-viewpoint video of human actors. In *ACM SIGGRAPH 2003 Papers*, pages 569–577.

Casas, D., Volino, M., Collomosse, J., and Hilton, A. (2014). 4D Video Textures for Interactive Character Appearance. *Computer Graphics Forum*.

Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., and Sullivan, S. (2015). High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4):69:1–69:13.

Dou, M., Fuchs, H., and Frahm, J.-M. (2013). Scanning and tracking dynamic objects with commodity depth cameras. *2013 IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pages 99–106.

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., and Izadi, S. (2016). Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13.

Garsthagen, R. (2014). An open source, low-cost, multi camera full-body 3d scanner. In *Proc. of 5th Int. Conf. on 3D Body Scanning Technologies*, pages 174–183.

Guo, K., Xu, F., Yu, T., Liu, X., Dai, Q., and Liu, Y. (2017). Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Trans. Graph.*, 36(3).

Jones, D. (2013). Picamera. `https://picamera.readthedocs.io/en/release-1.10/`. [Online; last-accessed 17-October-2018].

Lucas, G., Szablowski, E., Gratch, J., Feng, A., Huang, T., Boberg, J., and Shapiro, A. (2016). The effect of operating a virtual doppleganger in a 3d simulation. In *Proc. of the 9th Int. Conf. on Motion in Games*, pages 167–174. ACM.

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352.

Shrestha, P., Weda, H., Barbieri, M., and Sekulovski, D. (2006). Synchronization of multiple video recordings based on still camera flashes. In *Proc. of the 14th ACM Int. Conf. on Multimedia*, pages 137–140.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):97:1–97:9.

Wang, R., Wei, L., Vouga, E., Huang, Q., Ceylan, D., Medioni, G., and Li, H. (2016). Capturing dynamic textured surfaces of moving targets. In *Computer Vision – ECCV 2016*, pages 271–288, Cham.

Wauck, H., Lucas, G., Shapiro, A., Feng, A., Boberg, J., and Gratch, J. (2018). Analyzing the effect of avatar self-similarity on men and women in a search and rescue game. In *Proc. of the 2018 CHI Conf. on Human Factors in Computing Systems*, pages 485:1–485:12.