Towards Fully Mobile 3D Face, Body, and Environment Capture Using Only Head-worn Cameras

Young-Woon Cha[§], *Student Member, IEEE*, True Price[§], Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, *Student Member, IEEE*, Zihe Qin, Hyounghun Kim, Zhaoqi Su, Yebin Liu, *Member, IEEE*, Adrian Ilie, *Member, IEEE*, Andrei State, Zhenlin Xu, Jan-Michael Frahm, *Member, IEEE*, and Henry Fuchs, *Life Fellow, IEEE*

Fig. 1. We introduce a head-worn ego-centric capture system capable of reconstructing the wearer and their surrounding environment in 3D. Left: Hardware prototype. Center: An individual using the device. Right: Dynamic reconstruction of the user's body pose and static environment, obtained solely from the prototype's headset-mounted cameras.

Abstract—We propose a new approach for 3D reconstruction of dynamic indoor and outdoor scenes in everyday environments, leveraging only cameras worn by a user. This approach allows 3D reconstruction of experiences at any location and virtual tours from anywhere. The key innovation of the proposed ego-centric reconstruction system is to capture the wearer's body pose and facial expression from near-body views, e.g. cameras on the user's glasses, and to capture the surrounding environment using outward-facing views. The main challenge of the ego-centric reconstruction, however, is the poor coverage of the near-body views – that is, the user's body and face are observed from vantage points that are convenient for wear but inconvenient for capture. To overcome these challenges, we propose a parametric-model-based approach to user motion estimation. This approach utilizes convolutional neural networks (CNNs) for near-view body pose estimation, and we introduce a CNN-based approach for facial expression estimation that combines audio and video. For each time-point during capture, the intermediate model-based reconstructions from these systems are used to re-target a high-fidelity pre-scanned model of the user. We demonstrate that the proposed self-sufficient, head-worn capture system is capable of reconstructing the wearer's movements *and* their surrounding environment in both indoor and outdoor situations without any additional views. As a proof of concept, we show how the resulting 3D-plus-time reconstruction can be immersively experienced within a virtual reality system (*e.g.*, the HTC Vive). We expect that the size of the proposed egocentric capture-and-reconstruction system will eventually be reduced to fit within future AR glasses, and will be widely useful for immersive 3D telepresence, virtual tours, and general use-anywhere 3D content creation.

Index Terms—Telepresence, Ego-centric Vision, Motion Capture, Convolutional Neural Networks.

1 Introduction

As our society grows ever more connected digitally, individuals are increasingly interested in maintaining a connection with reality when they communicate their experiences and ideas with others across the globe. Indeed, modern video (*e.g.*, YouTube) and televideo (*e.g.*, FaceTime or Cisco TelePresence) content-sharing systems are used daily by hundreds of millions of people because they come the closest to relaying a veridical human experience. However, while such systems have grown in popularity as substitutes for witnessing events firsthand or having face-to-face meetings, these technologies fall short of delivering an actual sense of shared physical presence. Recent products such as Google JUMP [24] offer more immersive 360° video experiences but limit the viewer to a fixed position of observation. Prototype 3D capture and telepresence systems such as Microsoft Research's Holo-

- *Young-Woon Cha is with UNC Chapel Hill. E-mail: youngcha@cs.unc.edu*
- *True Price is with UNC Chapel Hill. E-mail: jtprice@cs.unc.edu*
- *Zhen Wei is with UNC Chapel Hill. E-mail: zhenni@cs.unc.edu*
- *Xinran Lu is with UNC Chapel Hill. E-mail: connylu@cs.unc.edu*
- *Nicholas Rewkowski is with UNC Chapel Hill. E-mail: nrewkows@cs.unc.edu*
- *Rohan Chabra is with UNC Chapel Hill. E-mail: rohanc@cs.unc.edu*
- *Zihe Qin is with UNC Chapel Hill. E-mail: zihe@cs.unc.edu*
- *Hyounghun Kim is with UNC Chapel Hill. E-mail: hyounghk@cs.unc.edu*
- *Zhaoqi Su is with Tsinghua University. E-mail: suzq13@tsinghua.org.cn*
- *Yebin Liu is with Tsinghua University. E-mail: liuyebin@mail.tsinghua.edu.cn*
- *Adrian Ilie is with UNC Chapel Hill. E-mail: adyilie@cs.unc.edu*
- *Andrei State is with InnerOptic Technology Inc. and UNC Chapel Hill. E-mail: andrei@cs.unc.edu*
- *Zhenlin Xu is with UNC Chapel Hill. E-mail: zhenlinx@cs.unc.edu*
- *Jan-Michael Frahm is with UNC Chapel Hill. E-mail: jmf@cs.unc.edu*
- *Henry Fuchs is with UNC Chapel Hill. E-mail: fuchs@cs.unc.edu*
- [§]*These authors contributed equally to the paper.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

portation [46] have likewise demonstrated promising steps towards shared 3D presence, but require substantial, expensive, instrumented areas. However, there are still no methods available that enable an everyday user to capture 3D directly.

We envision a future in which passive 3D capture of user experiences is a feature of commonplace head-worn devices. In this future, augmented reality (AR) systems such as the Microsoft HoloLens [41] have shrunk to the form factor of conventional eyeglasses, with fully transparent see-through and wide-field-of-view capabilities, and so can be worn all day just like ordinary eyeglasses. Our ambition is to augment such eyeglasses with a multiplicity of inward- and outward-looking miniature cameras. These cameras will form an ego-centric reconstruction system that will (1) capture its wearer's 3D pose, face, body, and limbs, and (2) map the 3D structure of its surroundings. The resulting dynamic scene can (3) be displayed to other users, using AR or virtual reality (VR) systems to create a shared, immersive 3D experience. Such self-contained, head-worn systems will enable shared presence and virtual touring to occur in any indoor or outdoor location, with no reliance on any instrumentation other than that in the user's headgear.

In this paper, we demonstrate a prototype system for ego-centric capture and reconstruction. Example camera views on the device are shown in Fig. 2. The main challenge of reconstruction from such head-worn cameras is the sparse visibility of body parts, which leads to large gaps in self-reconstruction. To address this problem, we propose to use a deformable-model-based approach to complete the unobserved parts of the wearer: as the generic parametric model still has gaps in the user's appearance, *e.g.*, in clothing, texture, and other detailed characteristics such as hair, we transfer such surface details from a pre-scan of the full body of the user. In the future, when such systems are miniaturized, personalized, and worn for long periods of time, we envision that they will automatically and gradually acquire detailed full-body information of their users and their wardrobes.

Our reconstruction approach is a user-oriented model-based self-reconstruction pipeline that combines parametric body and face models. The model-based incomplete reconstruction is re-targeted to a high-quality pre-scan of the user in a coarse-to-fine manner. The deformable models have two types of parameters: shape-related and pose-related. The shape parameters of the body and face are estimated in the preprocessing stage by fitting the models to the pre-scan. The pose-related parameters, body pose and facial expression, are detected at run-time with our CNN-based pose estimation and through audio- and video-based facial expression estimation.

Our system demonstrates full scene reconstruction, including the user's moving body with audio and their surrounding environment. The environment is reconstructed using structure-from-motion with outward-looking cameras. The trajectory of the user's head is determined using multiple calibrated cameras, which allows the system to localize the reconstructed user within the environment over time. The unified capture can be immersively experienced in a VR system.

The remainder of this paper is organized as follows: The related work is discussed in Section 2. The overall self-reconstruction pipeline is discussed in Section 3. Our ego-centric capture prototype is described in Section 4. Section 5 describes the pre-scanning process. Sections 6 and 7 address CNN-based body pose estimation and CNN-based facial expression estimation from both audio and video. The environment and head pose estimation techniques are detailed in Section 8. After integration considerations in Section 9, we show and discuss our results in Section 10, followed by limitations and future work in Section 11.

2 Related Work

Given its complexity and wide range of challenges, our work is related to a variety of existing research in the areas of 3D reconstruction of static scenes and dynamic objects, head tracking, and motion capture. We will briefly review the most closely related approaches.

Static 3D Reconstruction of an environment from photos and videos has been a long-standing research thrust in computer vision. 3D reconstruction algorithms include structure from motion [1,25,52,58] combined with stereo vision [51], simultaneous localization and mapping (SLAM) [21,22,42,61,64], multi-view vision [53,54], and depth-camera-based algorithms [28,40,43,49] and can be used to reconstruct only static scenes. We build on the progress made by the body of work in these areas to obtain our environment reconstruction and to track the user within the environment. Moreover, we extend the approaches to leverage the constraints provided by our multi-camera setup.

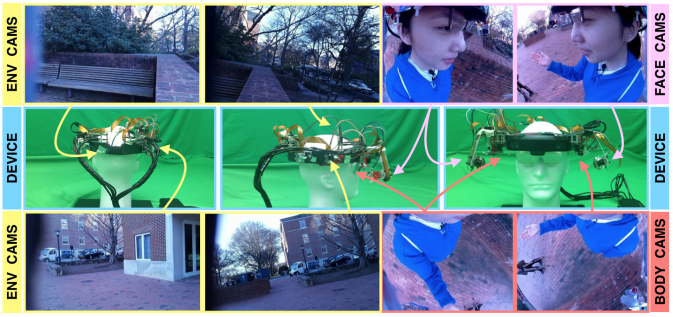

Fig. 2. The eight views from a single time-point of capture on our device. Outward-looking environment cameras (yellow) are placed on the side and rear of the device. Face-oriented cameras (pink) are placed on short arms on either side of the device. Downward-facing body cameras (orange) are located on both sides of the user's forehead. The top row shows the left rear external, left side external, right face, and left face views. The bottom row shows the right rear external, right side external, right body, and left body views.

Dynamic Object Reconstruction. Dynamic object reconstruction has long been an active research area. Most approaches rely on moderate surface deformations or known object shape for reconstructing a 3D model using a video of the object [26,34,35,62,66,70]. Alternatively, motion capture systems [5,15,16,23,59,63,67,69] deliver reliable reconstructions of human bodies from a sequence of color and/or depth videos. These approaches require a pre-scanned body model or template, an instrumented environment, and complicated skinning and rigging preprocessing. These factors prevent their application to reconstructing general shapes in unconstrained environments, which is mandatory for our system.

There has also been a keen interest in parametric body models for reconstruction and tracking. Allen et al. [3] leveraged high-resolution range scans to develop a parametric body shape model. The SCAPE model [4] advanced this approach to not only parameterize body shape but also encode pose deformation. Chen *et al.* [13] further extended the SCAPE model by introducing parameters to explain the deformation from clothing. Their model deformed the overall person model non-rigidly, by applying the composite transformations of the poses, the shapes, and the clothing for each triangle independently. Recently, Loper *et al.* [37] proposed the SMPL model, which provides more realistic deformations and achieves a more accurate representation of the effects of joint motion.

Recent work for template-free dynamic surface fusion [18,19,44,46] has shown promising results for object-level and human reconstruction in outside-in capture scenarios for instrumented environments. However, these methods are not suited to work with passively captured data from our mobile system, which requires reconstruction methods that operate in arbitrary environments. without external instrumentation.

Dynamic Scene Reconstruction from Depth Sensors. There has been significant interest in dynamic scene reconstruction from depth sensors. For example, Maimone and Fuchs constructed a real-time 3D capture system using a dozen Kinects [38,39]. This method adapts the volumetric fusion of Levoy *et al.* [14] to dynamic objects (*i.e.*, people) while incorporating depth and color information. More recent room-size dynamic object reconstruction [17] combines pre-scanning of the static scene parts, data accumulation for dynamic objects, and rigid and nonrigid tracking. However, these approaches rely on successful depth image capture using structured light, which typically fails outdoors. Our system targets both outdoor and indoor use and hence cannot use structured light sensors.

Ego-centric Motion Capture. Ego-centric, body-worn cameras have been used for 3D pose estimation of certain parts of the body

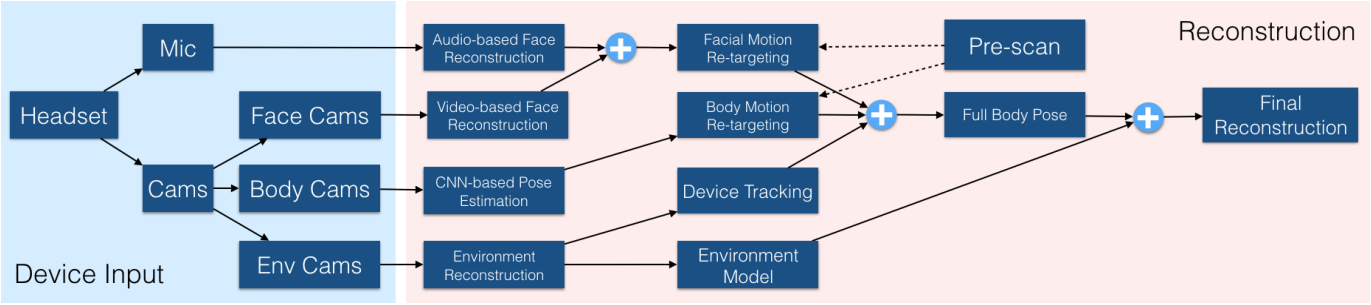

Fig. 3. Functional overview of our system, with HoloLens-mounted capture components at left and off-line reconstruction processing pipeline at right.

such as facial expressions via helmet-mounted cameras [45] or finger motions via wrist-worn sensors [33]. Shiratori *et al*. [56] determined full-body motions based on 16 body-worn cameras with poses estimated through structure-from-motion, assuming a static environment. Jiang and Grauman [30] proposed a learning-based approach to predict full-body poses from a chest-worn camera view to infer invisible poses with very limited accuracy. Zhang et al. [71] used a single outside-in depth camera combined with foot-worn sensors for full-body pose estimation. All of the above approaches only perform skeleton-based motion capture and do not reconstruct the 3D surface of the wearer solely from the body-worn cameras. Rhodin et al. [50] employed two head-mounted fisheye cameras to estimate the full-body skeleton pose. The large field of view allowed the cameras to observe most of the body and to integrate with approaches based on outside-in cameras. However, their system required the head-mounted cameras to be placed on long telescopic arms reaching significantly outward in front of the wearer. This enabled them to perform a stereo-based body reconstruction at the cost of usability. In contrast, our system leverages cameras close to the body, trading a full-body stereo view for broad usability.

3 System Overview

An overview of our mobile capture pipeline is shown in Fig. 3. From a computational perspective, the inputs to our system are individual views from synchronized head-worn cameras, and the output is a posed 3D model of the wearer placed into a reconstructed 3D model of the surrounding environment. Body poses and facial expressions are captured entirely from the on-device camera views, as is the 3D environment model. For visualization, a pre-computed digital human representation ("pre-scan") of the user is posed according to estimated face and body parameters.

Details about the head-worn camera configuration are provided in Sect. 4, and we describe the pre-scan acquisition process in Sect. 5. Our reconstruction approach consists of three processing pipelines, each of which takes in separate camera imagery: body pose estimation, consisting of skeleton joint detection and 3D triangulation (Sect. 6); face reconstruction from video and audio data (Sect. 7); and environment reconstruction, which encompasses both reconstructing the 3D scene and tracking the motion of the user as they move within their surroundings (Sect. 8).

The individual reconstruction results are combined (see Sect. 9 for results). The body pose and face expressions are applied as parametric deformations of their associated pre-scan models; these adjusted face and body pre-scans are then combined to create the momentary digital human representation of the user. This representation is then placed into the scene based on the tracked location of the user within their environment, and the placed (animated) model can be rendered in the context of the reconstructed static scene around the user. The resulting dynamic 3D model can then be utilized for a variety of applications, such as virtual tours (Sect. 10.4 and supplementary material).

4 Mobile Headset Prototype

In our vision for ubiquitous AR/VR systems of the future, an individual will be able to fully capture themselves and their 3D surroundings solely from a lightweight pair of eyeglasses fitted with miniature cameras. We anticipate that these devices, possibly combined with a small backpack computer for processing, will have functionalities for both general capture (*e.g.*, self-created VR content analogous to current online video services) and telepresence (*i.e.*, real-time 3D ego-capture, coupled with AR displays). In this work, we have developed a prototype headset to demonstrate the various camera configurations and reconstruction approaches that such a device would employ.

Our prototype 3D capture unit has been outfitted with 8 Pi V2 miniature cameras (Fig. 2). We divide these cameras into three categories based on their function: four outward-facing cameras capture the environment and track the device's motion, two downward-facing cameras capture the user's body, and two face-oriented cameras capture the wearer's facial expression. The cameras on our headset run individually on Raspberry Pi Zero miniature computers powered by portable battery banks worn in a backpack. The external views are captured using 70° diagonal FoV cameras and are located on the sides and back of the headset. The face and body cameras have 160° diagonal FoV lenses; the body cameras are placed slightly in front of the wearer's forehead, and the face cameras are placed on slightly extended mounts ∼9cm from the user's face. We anticipate that future systems will be able to reduce the outside-in distance of the face cameras even further, to the point where the cameras are mounted directly next to the lenses of the eyeglass frame.

The cameras are synchronized off-line using LED blinking [6] and capture at $25 fps$. (These were design decisions for our prototype; hardware synchronization and faster frame rates are possible in principle.) Anticipating future AR integration capabilities, we have mounted the camera system on a Microsoft HoloLens headset; however, we currently do not use the HoloLens' on-board display or capture technologies. Also note that our capture scenario involves on-line capture and off-line 3D reconstruction – in this work, our motivation is to demonstrate the technologies involved in performing automated, hands-free, use-anywhere 3D capture.

System Calibration. In addition to frame-level camera synchronization, we assume that the intrinsic and relative extrinsic camera parameters for the device are known before capture. This involves estimating the relative rotations between the cameras, the absolute distances between the cameras' centers of projection, and the position of the rig in relation to the wearer's head. Camera intrinsics were computed using standard checkerboard-based camera calibration. To capture the relative camera poses, we set up a small, well-textured scene and moved/rotated the headset by hand (without anyone wearing it) while capturing imagery from the cameras. We then reconstructed this synchronized multi-camera sequence using structure-from-motion (SfM) [52] with a bundle adjustment that estimates a global pose for the device at each time instant while enforcing static relative poses for the cameras in the cluster. Since SfM reconstructions are inherently scale-independent, we recovered the absolute scale of the headset by manually comparing the sizes of reconstructed objects with known real-world measurements. The location of the rig with respect to the wearer was then established by computing the midpoint of the two side external cameras and aligning it with the approximate midpoint of the wearer's temples.

5 Digital Human Pre-scan

Our system integrates motion capture and environment reconstruction. For visualization, however, it is impossible to create a complete model of the wearer from the headset views, because we capture only partial

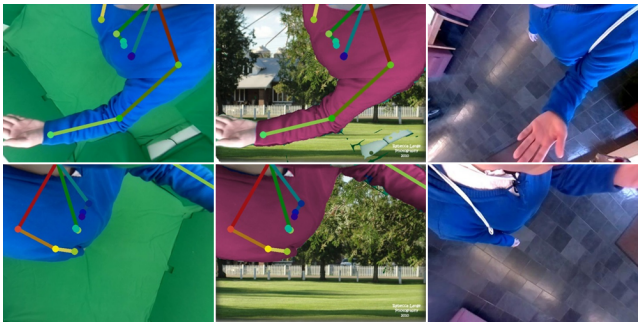

Fig. 4. Example images from the pair of downward-facing body cameras on our device. Left: Training images captured in our green-screen room. Middle: Training images augmented by shirt recoloring and background replacement. Right: Images from our hallway demo. The top and bottom rows show images from the left and right body cameras, respectively. The colored skeleton visualizes the projection of the ground-truth 3D joint positions into the individual views in the original captured and augmented training images.

views of the user's face and of parts of their body, resulting in an incomplete digital human representation. Instead, we obtain an off-device 3D scan ("pre-scan") of the user that fully captures their body shape and clothing. The system localizes body skeleton joints in the two downward-facing views, and parameters for the user's facial expression are computed from the two (non-overlapping) face-oriented views. The pre-scan is deformed to match the skeleton and face parameters and then placed in the 3D environment based on the estimated device pose. Details about the skeletal rigging and skinning of the pre-scan are provided in Sect. 6.4.

To obtain the pre-scan, which is a textured mesh of the entire body, we use itSeez3D [27], a 3D scanning software. The user stands still with their arms extended while another individual moves a small RGB+D camera unit around them to capture the body surface and texture.

In the future, we anticipate that pre-scan acquisitions could be completed entirely on-device, with the wearer capturing their appearance by, *e.g.*, placing the device on a table and walking in front of it, or by wearing the device and standing or turning in front of a mirror. Such on-device processing would not only increase the ease of use, but would also enable on-the-fly representations of new individuals or allow updates of the clothing or appearance of the same individual.

6 Video-based Body Pose Reconstruction

Body pose estimation solely from head-worn cameras is a challenging task. The most closely-related system, EgoCap [50], uses two head-worn fisheye cameras on an extended 'V'-shaped rig. However, they extend 20-30 cm away from the user's head, which is prohibitive for convenient, portable use. Our system is unique in that we target commodity cameras located directly on the compact headgear; this generally results in very restricted viewpoints that provide less reliable measurements for body pose estimation, particularly for the legs, which are far from the cameras and often occluded. To overcome the difficulties in capturing body pose, we leverage deep convolutional neural networks (CNNs) to perform body part detection in the individual downward-facing views, as well as an additional recurrent network module to obtain a final skeleton-based human pose estimation.

6.1 2D Human Body Joint Detection

To solve the initial problem of detecting the device wearer in the downward-facing views, we have extended the convolutional pose machine (CPM) network [11, 65] to detect 2D joint positions in each image independently. CPM incorporates a convolutional neural network into the pose machine framework [48], which enhances image feature extraction (in this case, 2D joint locations) by leveraging inference on image-dependent spatial models. CPM is built upon an end-to-end, multi-stage deep network that enables the learning of both joint appearances and spatial relationships in input imagery. Beyond traditional cascaded networks, CPM is also an interactive sequence framework, with each stage considering the context of previous stages in order to derive an overall set of joint positions for a given image.

A pose machine consists of a hierarchy of 2D joint predictors $g_t(\mathbf{f}_t(x), \psi_t(j, \mathbf{b}_{t-1}))$ that output joint-specific belief values for all positions x the image domain, for each stage t in the hierarchy. $\mathbf{f}_t(x)$ represents a stage-specific feature embedding for the input image, and $\psi_t(\cdot)$ maps the existing volume of belief values $\mathbf{b}_{t-1}$ for all joints across the image into a specific context mapping for joint j. Given the input image, the first stage $g_0(\cdot)$ is an image-space classifier that produces a joint-probability volume $\mathbf{b}_0 = \left\{ b_t^j(X_j = x) \right\}_{j \in 0 \ldots J}$, where X_j is a random variable relating the position of joint j. Later stages $g_t(\cdot)$ update the belief for assigning a location to each part:

$$g_t(\mathbf{f}_\mathbf{t}(x), \psi_t(j, \mathbf{b}_{t-1})) \mapsto \mathbf{b}_\mathbf{t}. \tag{1}$$

The final 2D joint predictions are retrieved as the most probable locations for each X_j after the final belief values are predicted.

The prediction and image feature computation modules of a pose machine can be replaced by a deep convolutional architecture, allowing for both image and contextual feature representations to be learned directly from data. The CPM contains multiple stages of a fully-convolutional network cascaded to characterize both the local features of the input image and the global features across larger receptive fields. By chaining prediction stages, the receptive fields at the output layer of the network are large enough to allow the learning of potentially complex and long-range correlations between body parts.

The cost function at each stage of the CPM minimizes an l_2 distance between the predicted and ideal belief map for each joint:

$$\ell_t = \sum_{j=1}^{J} \sum_{x} \| b_t^j(x) - b_*^j(x) \|_2^2, \tag{2}$$

where $b_*^j(X_j = x)$ represents the ideal belief map for joint j. The overall objective for the full architecture is obtained by adding losses over all T stages and is given by

$$\mathscr{F} = \sum_{t=0}^{T-1} \ell_t. \tag{3}$$

As seen in the views of the downward body cameras shown in Fig. 4, we define our detectable joints as the shoulders, elbows, wrists, hips, and knees. Ankles are generally not visible from our near-body views – for instance, each foot is independently visible for only ∼33% of a gait cycle. – so instead we model them in 3D using motion priors (see Sect. 9). We predict these joint positions via a custom-trained CPM for our input views (see Sect. 6.3). This 2D detection is trained separately from our subsequent 3D pose estimation network. We pad the original images to allow predicting the position of joints that are located outside the images. This enables the fully convolutional network to learn correlations between (and predict 2D locations for) all joints, whether or not they are actually visible in the input views.

6.2 3D Human Pose Sequence Estimation

Given the 2D detection result, we employ a 3D pose sequence module to predict the 3D skeleton joint positions over time. This module leverages a recurrent neural network (RNN) to capture long-term motion trajectories for all observable joints. Compared to general neural networks, RNNs are able to scale to much longer temporal sequences and are practical for sequence-based specialization, such as video processing. This is because in RNNs, each member of the output is a function of the previous member of output, with all outputs being produced by the same update rule. Thus, the temporal motion information between frames can be effectively incorporated into the 3D pose prediction.

For our recurrent 3D human pose network, we take a sequence of 2D positions (x_t^j, y_t^j) in the images of each of the two body-camera views and their corresponding probabilities p_t^j as our network input $X = [X_1, X_2, ..., X_t]^T$, where $X_t = [(x_t^1, y_t^1, p_t^1), \ldots]$ at time step t. (Note that t here refers to the temporal domain of the capture sequence and j

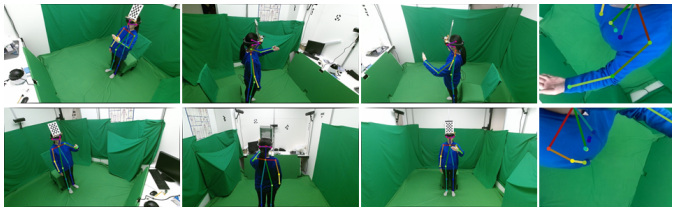

Fig. 5. Images from the six external cameras and two top-down body cameras on our device used for capturing our ground-truth body pose dataset. The colored skeleton depicts the ground-truth 3D joint positions.

the joints over both views.) For training, points and probabilities are generated by random Gaussian perturbations of the ground truth 2D joint position. At run-time, they are generated using our trained CPM.

The network consists of three fully connected layers (512, 1024, and 1024 neurons, respectively), one recurrent layer (2048 hidden states), and finally two fully connected output layers (1024 and 30 neurons) that unilaterally predict all 3D joint positions for a given time step. For each time step t, we have

$$h_t = \sigma(W_{h_1} h_{t-1} + W_{h_2} f_i(X_t) + b_h) \tag{4}$$

$$Y_t = f_o(h_t) \tag{5}$$

where f_i is the function applied on the input before the recurrent part; h_t is the recurrent layer's hidden state at step t; W_{h_1}, W_{h_2}, and b_h are the weights and bias; σ is a non-linear function; and f_o is the function applied after the recurrent layer to obtain the output 3D positions Y_t at time step t.

We minimize the sum of l_2 distances between the ground truth 3D positions and the predictions:

$$E = \sum_t \|Y_t - Y_t^*\|_2^2, \tag{6}$$

where Y_t^* consists of the ground truth 3D body joint positions at frame t. Incorporating the previous 3D pose prediction at each stage allows our network to robustly compute the pose predictions.

6.3 CNN Training and Testing

Capturing the Training Dataset. The key challenge for training our body pose estimation network lies in obtaining ground truth data for the 2D and 3D joint positions. To solve this problem, we constructed a data capture setup for outside-in marker-less motion capture and calibrated headset tracking, and used background subtraction for data augmentation.

The videos for the training dataset and the ground truth positions of 3D human body joints are obtained using a calibrated set of synchronized external cameras. Our training setup consists of a mid-size room with the outside-looking-in cameras placed near the walls. The user wearing our device is standing in the middle of the capture space. Fig. 5 shows an example set of camera images captured at the same time.

In each external view, we apply a pre-trained OpenPose CPM network [11, 65] to detect 2D joint positions. Having pre-computed the positions of the cameras in the room, we are able to triangulate each joint in 3D over time. We also track the 6-DoF pose of the downward-facing cameras using a checkerboard pattern mounted on the device. The relationship between the checkerboard and the device cameras is calculated using hand-eye calibration [55], and the pose of the device within the capture space is determined by recovering the pose of the checkerboard from the external views. Given the triangulated 3D joint positions and the pose of the device, we obtain ground truth 3D joint positions by simply applying the scene-to-device transformation, and 2D joint positions for each camera are then determined via projection using the camera intrinsics.

Network Training. Using data from our capture environment, we trained a new CPM network for our downward-facing views and an RNN to predict the 3D human pose sequence. We used the Caffe deep-learning framework [29] to train both networks. To enhance the generality of our CPM, the surrounding room was made into a "green-screen" environment, and the capture subject was given a blue sweater to wear during training. The training data was then augmented by replacing the green surfaces with random floor/object textures and the blue shirt with randomly adjusted hues. The input images were further augmented using flips, rotations, and translations. In order to obtain sufficient samples for training the 3D pose RNN, we simulated fast-motion speeds by interpolating poses between the frames, and we also subsampled the captured frame sequence into many shorter frame sub-sequences.

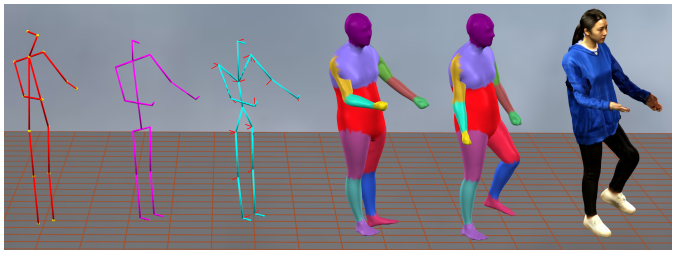

Fig. 6. Body pose re-targeting. From left to right: 1) Detected joint positions. 2) Bone length adjusted skeleton with hand/foot orientation constraints. 3) Rotational skeleton of the model. 4) Deformed body model in which joint angles are estimated by fitting the model skeleton (3) to the canonical positions (2) using joint-limit-constrained inverse kinematics. 5) Walking motion prior. 6) Final textured pre-scan model with blended pose.

Network Execution. At run-time, we use our CPM to hypothesize the most likely 2D joint positions for the input downward-facing imagery. The 3D RNN then takes these points, along with their probabilities, and outputs a hypothesis for the 3D position of each joint relative to the left downward-facing camera. We post-process the 3D joint result using a Kalman filter and basic exponential smoothing, which allows us to robustly account for sporadic mis-predictions of the 3D joint position. The end result is a smooth skeletal motion capture sequence of the user across time.

6.4 Body Motion Re-targeting

Rigged parametric body models [3,4,13,37] can be exploited to deform the pre-scan model constrained by the 3D joint positions output by our RNN. Our proposed approach employs the Simplified-SCAPE parametric body model [47] using linear blend skinning for computational efficiency. During pre-processing, the parametric body model is fit to the pre-scan model for automatic rigging. The predicted 3D posture at each frame is applied to the rigged pre-scan at run-time.

The body model $\mathbf{M}(\theta,\beta)$, which is represented in homogeneous coordinates, is specified by the joint configuration θ and shape parameters β of PCA space $\mathbf{S} \in \mathbb{R}^{4|V|\times|\beta|}$, and is deformed from the mean body shape $\widehat{\mathbf{M}}$:

$$\mathbf{M}(\theta,\beta) = \mathbf{R}(\theta)\widehat{\mathbf{M}} + \mathbf{R}(\theta)\mathbf{S}(\beta). \tag{7}$$

$\mathbf{R} \in \mathbb{R}^{4|V|\times 4|V|}$ is the block diagonal matrix of per-vertex joint transformations. $\mathbf{M}(\theta,\beta)$ is fit to the pre-scan $\mathbf{T}$ to estimate the vertex correspondences by minimizing the following energy w.r.t θ and β:

$$\mathbf{E}_\mathbf{M}(\theta,\beta) = \sum_{i=1}^{|V|} ||v_i(\mathbf{M}(\theta,\beta)) - \mathrm{NN}_i(\mathbf{T})||_F^2, \tag{8}$$

where $||\cdot||_F$ is the Frobenius matrix norm. Each vertex $v_i(\mathbf{M}(\theta,\beta))$ of the model is fit to its closest compatible nearest neighbor vertex $\mathrm{NN}_i(\mathbf{T})$. More details regarding the optimization are given in [47]. Using Eq. (8), the preprocessing shape parameters β_0 with bone lengths and pose parameters θ_0 are determined for the association between the model and the pre-scan. β_0 and bone lengths are fixed for the entire run-time sequence.

The skeletal joint placements Θ_T of the pre-scan $\mathbf{T}$ are transfered from the fit joints $\Theta_M(\theta_0,\beta_0)$ of $\mathbf{M}$. Based on the vertex correspondences from Eq. (8), the skin weights $w(v_i) = \{w_1(v_i),...,w_{|\theta|}(v_i)\}$ of

each model vertex are also transfered to $\mathrm{NN}_i(\mathbf{T})$. The skin weights of remaining pre-scan vertices are interpolated from nearby $\mathrm{NN}_i(\mathbf{T})$. From the transfered joint structure and skinning weights, the captured skeletal animation can be accordingly applied to the pre-scan.

At run-time, the pose parameters θ_t at time t of the body model $\mathbf{M}(\theta_0, \beta_0)$ are estimated from the 3D joint positions output by our RNN. Specifically, the joint positions form a positional skeleton using a pre-defined structure. The joint angles θ_t are estimated from this positional skeleton using joint-limit-constrained inverse kinematics (IK) [20]. To fit the model skeleton to the positional skeleton, the bone lengths of the the positional skeleton are adjusted to match the model skeleton. Using the spine and hip joints, the rigid transform from the model to the positional skeleton is estimated using point-to-point ICP. The remaining joint angles are estimated using the constrained IK.

The joint correspondences between the model skeleton and the positional skeleton are pre-defined. The angular derivative $\dot{\theta}$ of joints are estimated by solving the differential IK $\dot{\theta} = \mathbf{J}^{\#}\dot{x}$. $\dot{x}$ is the change in corresponding joint positions, and $\mathbf{J}^{\#}$ is the pseudo-inverse of Jacobian matrix. The joint angle limit is constrained by transforming the angle derivative $\dot{\theta}$ to the transformed space $\dot{z}$. When $z_t = z_{t-1} + \dot{z}_t$ converges to the joint limit, it regains manipulability by enforcing z_i to move in the other direction [47]. This guarantees that $\theta_i = T(z_i)$ is always a valid joint angle. We use the elbow and knee joint limits to prevent anatomically implausible poses.

The foot and hand orientations are not included in the positional skeleton, however, which can result in an IK result that arbitrarily twists the arms and legs. To prevent this, we added dummy joints at each end effector (hands, feet, and head) to constrain them to valid orientations in IK. The torso normal direction is set according to these dummy joints. Fig. 6 shows the joint fitting result with the joint limits and the orientation constraints.

From the estimated pre-pose θ_0, and current pose θ_t, the pre-scan $\mathbf{T}$ is deformed as:

$$\widehat{\mathbf{T}} = \theta_t\, \theta_0^{-1}\, \mathbf{T}, \tag{9}$$

where θ_0^{-1} is the inverse joint transformation of θ_0, which moves the pre-scan to the neutral pose of $\mathbf{M}$, allowing the current pose θ_t to be applied directly.

7 Audio/Video-based Face Reconstruction

To obtain a high-quality 3D model of the user's face, we adopt a similar pipeline to our body-modeling approach. In the prototype system, two on-device cameras are used to capture each side of the user's face. This is in contrast to most work on face reconstruction that utilizes a single frontal view for face capture. The goal of our setup is to have the cameras capture adequate views of the face without being obtrusive. Similar to state-of-the-art live face capture systems, we use landmarks detected in the individual views to fit a 3D deformable face model that incorporates both face shape and expression. We further enhance reconstruction quality by transferring facial expressions [60] from the deformable face model to a high-quality user model. To compensate for the limited visibility of the face, we leverage an audio-driven deep neural network to enhance the facial expression estimation.

7.1 Video-based Face Reconstruction

Our video-based face reconstruction pipeline takes as input two synchronized images from the downward-facing cameras, as well as a pre-scan model of the user's face. For each captured time instant, we detect 2D landmarks in the two images. We then compute a deformation of the pre-scan that minimizes the reprojection error between the face model's fiducial 3D landmarks and their corresponding 2D detections.

Pre-scan Fitting. As input to our capture process, we fit a morphable model to the high-quality face pre-scan. In general, the face model has three sets of parameters: the pose T (global rotation and translation in relation to the left camera), shape parameters α_s, and expression parameters α_e. First we manually labeled 68 3D landmarks in both the pre-scan and the model, and then we computed the face pose T through rescaling and fitting these correspondences. Following [9], we assume that the pre-scan has a neutral expression α_{e_0} and estimate the shape coefficients α_s by minimizing

$$E_{fPre} = \omega_{lm} E_{lm} + \omega_d E_d + \omega_{reg} E_{reg}, \tag{10}$$

where the first term E_{lm} penalizes errors in the 3D landmark alignments, the second term E_d relates to dense vertex matching between model vertices and their nearest neightbor vertices in the pre-scan, and the final term E_{reg} regularizes the PCA coefficients α_s. The full method and objectives used for shape parameter optimization are described in [9]. In our formulation, we use $\omega_{lm} = 1$, $\omega_d = 2$, and $\omega_{reg} = 1$.

Detecting 2D Landmarks. To compute the face model parameters for the user at a given time instant, we first detect 2D facial landmarks in the side images. The problem of 2D facial landmark localization for frontal face images has largely been solved [7, 8, 10, 68, 72]. However, these methods fail for the profile and oblique views that we target. Recent work [7] has shown good performance on significantly non-frontal 2D and 3D face alignment in difficult illumination conditions; however, we found that this method could not detect landmarks in most of our images. We fine-tuned this neural network with new data captured from our ego-centric viewpoints and provided a rough bounding box to the face detector, which greatly improved the accuracy of the detections. Because the face cameras are fixed in our headset, determining a reliable bounding box for the face is straightforward. Ground-truth landmark positions were obtained by applying the detector to a separate front-facing external view, computing the 3D landmark positions using the approach from [7], and projecting these points into the face-oriented views using the checkerboard tracking method of Sect. 6.3.

3D Model Fitting. Once we obtain the detected 2D facial landmarks, we deform the low-quality face mesh to fit the two side camera images by minimizing the reprojection error of the model's corresponding 3D landmarks. Specifically, for a given time instant, we optimize the pose T, shape α_s, and expression α_e of the morphable model. In practice, the shape and pose of the face are nearly constant in relation to the viewing cameras; however, we found that our facial capture results improved slightly by optimizing these values on a per-frame basis.

For each frame, our optimization iterately minimizes a separate cost function for each parameter type (pose, shape, and expression). The pose cost function is the sum of errors for the left and right cameras (indexed as 1 and 2):

$$E_{pose} = \sum_{i \in L_1} ||y_i - \Pi_1 (TV_i)||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_2 \left(MTV_j\right)||_2^2, \tag{11}$$

where y_i is the i-th detected 2D landmark, V_i is the corresponding labeled vertex, Π_c denotes the projection function of camera c, M is the relative transformation matrix between the two face-oriented cameras, and L_c denotes the set of visible landmarks in camera c. T is thus optimized by minimizing the reprojection errors between each y_i and the projection of its 3D correspondence V_i.

With a fixed M, we found that the pose solution sometimes converged to a local minimum, which lead to inaccurate shape and expression parameters. Thus, we relaxed the M constraint by computing a face pose for each camera separately, and added a term to limit their transformation matrix to be as close as T_r as possible:

$$E'_{pose} = \sum_{i \in L_1} ||y_i - \Pi_1(T_1 V_i)||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_2(T_2 V_j)||_2^2 + ||M' - M||_2^2, \tag{12}$$

where T_1 and T_2 are camera-specific face pose estimates, and $M' = T_2 T_1^{-1}$. After optimization, we set $T := T_1$.

Having computed the pose matrix T, we independently optimize shape and then expression parameters. For the shape parameters, which are initialized according to the pre-scan, the cost function is

$$E_{shape} = w_l E_l + w_{sparse} E_{sparse} + w_{sym} E_{sym} + w_{smooth} E_{smooth}. \tag{13}$$

The first term is similar to the pose cost function, minimizing the reprojection error of corresponding 2D and 3D landmarks:

$$E_l = \sum_{i \in L_1} ||y_i - \Pi_K T(\bar{V} + A_s \alpha_s)_i||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_K T_r T(\bar{V} + A_s \alpha_s)_j||_2^2, \tag{14}$$

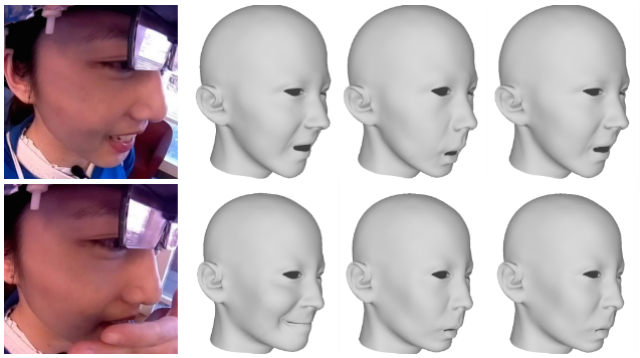

Fig. 7. Two video/audio-based fitting results. The first column shows the original image captured by the right side camera, and the second and third columns respectively show reconstruction results using only video or audio. The last column shows the final result of combining video and audio. The top row shows a result where the face is unoccluded; in this case, the combined result closely matches the video-only result. The second row demonstrates the contribution of audio-based capture when the face is partially occluded.

where $\bar{V}$ is the base shape of the morphable model, and A_s is the model's shape basis matrix. The subscript i denotes the i^{th} deformed vertex. The second term is a regularizing term to constrain the number of active shape parameters:

$$E_{sparse} = \sum_{i=1}^{N_s} |\alpha_s^i|, \tag{15}$$

where N_s is the total number of shape parameters.

The third term enforces vertical symmetry for each left face landmark i with a corresponding right landmark j:

$$E_{sym} = \sum_{(i,j)} \left| (\bar{V} + A_s\alpha_s)_i - (\bar{V} + A_s\alpha_s)_j \right|_y^2, \tag{16}$$

where $|\cdot|_y$ is the distance measured only in the y direction.

The final term smooths the parameters for consecutive frames:

$$E_{smooth} = \sum_{i=1}^{N_s} ||\alpha_s^t - 2\cdot\alpha_s^{t-1} + \alpha_s^{t-2}||_2^2 \tag{17}$$

where α_s^t denotes the frame index for the current frame.

Once the shape parameters have been estimated, we repeat the fitting of expression parameters α_e using the same terms as Eq. (13), and incorporating the shape estimate for Eqs. (14) and (16). We selected $w_l = 1, w_{sp} = 4, w_{sy} = 1, w_{sm} = 1.5$ for the shape cost function and $w_l = 1, w_{sp} = 6, w_{sy} = 1, w_{sm} = 0.8$ for the expression cost function, with α_e initialized to zero each frame. Fig. 14 shows results of 3D face fitting. For each frame, we transfer the fitted parametric model back to the high-quality pre-scanned user model using the approach from [60].

7.2 Audio Enhancement for Face Reconstruction

Full-face reconstruction relying solely on ego-centric views is challenging due to the oblique viewing angles. For example, the model expression parameters are highly influenced by small errors in the landmark detections for the mouth, yet the mouth is only partially visible in each view. Moreover, video-based reconstruction is hindered if the face is (partially) occluded (*e.g.*, see the bottom example in Fig. 7). We address these problems by augmenting the face reconstruction with geometry derived from the captured audio. Recently, Liu et. al. [36] presented a real-time facial tracking and animation approach that uses audio data to augment reconstruction from a single RGB-D camera. We investigated adapting this neural-network-based approach for our ego-centric scenario.

Network Training. We first compute the video-based expression parameters α_e as ground truth from front-facing videos with audio. Then, we extract the corresponding audio features following [31]. For every video frame, we use a 520*ms* audio window consisting of 64 overlapping audio frames, each 16*ms* in length. Audio features consisting of 32 Linear Predictive Coding (LPC) coefficients are calculated for every audio frame. Thus, the input features for each audio window is a 64×32 image, which serves as the input to the neural network.

We used the modified VGG-16 network architecture from [57]. The last 6 convolutional layers and 2 pooling layers are dropped for small-sized input signals, and the output size of the last fully-connected layer is set to 16, which corresponds to the first 16 expression coefficients. We also infer a weight ω_e^a for every time instant, representing the confidence of the audio result, as follows. We first detect silent frames in the data by checking the 600 ms window around each time instant. If all converted wave values in the window are below a threshold, we call it a silent frame. Non-silent frames are assigned a "full-audio" weight $\omega_e^a = 1$, and for silent frames, ω_e^a is determined by the length of time to the nearest non-silent frame.

Fig. 8. Face re-targeting result. The two left images show the input deformed face model and the re-targeted face part of the pre-scan, respectively. The face vertices of the body model are replaced by their deformed counterparts, as shown in the right image.

7.3 Combining Video and Audio

Similar to [36], we combine the audio-estimated expression parameters α_e^a with the video parameters α_e^v to compute the final frame parameters α_e:

$$A_e\alpha_e = WA_e\alpha_e^a + (I - W)A_e\alpha_e^v, \tag{18}$$

where $W \in \mathbb{R}^{3N\times 3N}$ is a diagonal weighting matrix, and N is the number of vertices in the morphable model. Differently from [36], we compute a weight map around the mouth landmarks and multiply it with weights inferred from the audio neural network ω_e^a as the final weights of every vertex. During the combination step, we also consider occlusion of the mouth. If the landmarks detection result has a large difference between two consecutive frames around the mouth, we negate the video-based mouth weights and rely strictly on audio for that region. The result of combining video and audio is shown in Fig. 7.

7.4 Facial Motion Re-targeting

During capture, we estimate pose, shape, and expression coefficients that fit our 3D morphable face model to the observed 2D data. For visualization, we require a way to deform the pre-scan face mesh according to this transformation. Because the face part of the pre-scan is not rigged, we employ a deformation transfer [60] from the face model to the pre-scan. It minimizes the differences of the corresponding triangle deformations between the face-model mesh and the pre-scan mesh (first and second images in Fig. 8, respectively).

Let $\mathbf{S}$ be the face-model mesh after shape-based alignment to the pre-scan; denote its 3D vertices as $\{s_1 \ldots, s_n\}$ and its triangles as $\{(a_1, b_1, c_1), \ldots, (a_m, b_m, c_m)\}$, where the (a_j, b_j, c_j) indexes three vertices. Let $\widetilde{\mathbf{S}}$ denote the deformed face-model mesh using our estimated coefficients for a given frame; it has vertices $\{\widetilde{s_i}\}$ and the same triangles as $\mathbf{S}$.

As outlined in [60], the affine transformation for a triangle j in $\mathbf{S}$ to its corresponding triangle in $\widetilde{\mathbf{S}}$ can be defined as $\mathbf{Q_j} = \widetilde{\mathbf{E}}_\mathbf{j}\mathbf{E_j}^{-1}$. Here, $\mathbf{E_j} \in \mathbb{R}^{3\times 3}$ is the *edge matrix* for triangle j, defined as

$$\mathbf{E_j} = [(s_{b_j} - s_{a_j}) \ (s_{c_j} - s_{a_j}) \ n_j], \tag{19}$$

where n_j is the unit normal for the triangle. $\widetilde{\mathbf{E}}_\mathbf{j}$ is similarly defined.

Now, we wish to deform the pre-scan mesh $\mathbf{T}$ into a new mesh $\widetilde{\mathbf{T}}$ in a manner similar to the transformation of $\mathbf{S}$ into $\widetilde{\mathbf{S}}$. Assume, for the moment, that for each triangle j in $\mathbf{S}$, we know the corresponding

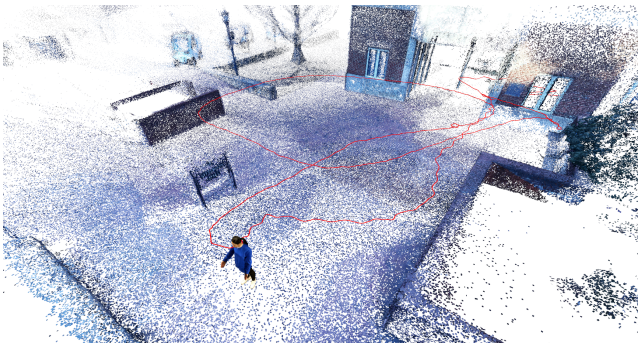

Fig. 9. Integration result. The deformed pre-scan is localized to the reconstructed environment using headset tracking. The entire path of the tracked headset is shown in red.

triangle ℓ in $\mathbf{T}$. (We explain how to obtain these correspondences below.) Using deformation transfer, we optimize for the vertices $\{\tilde{t}_k\}$ of $\widetilde{\mathbf{T}}$ by encouraging the affine transformations $\{\mathbf{Q}'_\ell\}$ of the triangles of $\mathbf{T}$ to match their counterparts $\{\mathbf{Q_j}\}$ of $\mathbf{S}$:

$$\min_{\tilde{t}} \sum_{(j,\ell)\in\mathbf{C}} ||\mathbf{Q_j} - \mathbf{Q}'_\ell||_F^2, \tag{20}$$

where $\mathbf{C}$ is the set of triangle correspondences, and $||\cdot||_F$ denotes the Frobenius matrix norm.

Computing triangle correspondences. The correspondences between the triangles of $\mathbf{S}$ and $\mathbf{T}$ are computed in a pre-processing step that first aligns the 3D landmarks $\mathbf{S}$ with $\mathbf{T}$ while encouraging smoothness of the triangle deformations of $\mathbf{S}$. Once this alignment is achieved, we obtain triangle correspondences based on nearest neighbors. The landmark correspondences in this section are the same as those used for our initial landmark-based model fitting.

Specifically, consider aligning the landmarks of $\mathbf{S}$ and $\mathbf{T}$ by deforming the vertices of $\mathbf{S}$. In a slight abuse of earlier notation, we now state that we optimize the vertices of $\widetilde{\mathbf{S}}$ so that they match $\mathbf{T}$:

$$E_{lm}(\{\tilde{s}_i\}) = \sum_{(i,k)\in\mathbf{L}} ||\tilde{s}_i - t_k||^2, \tag{21}$$

where $\mathbf{L}$ is the set of corresponding landmark-vertex-index pairs for the two meshes.

We regularize Eq. (21) to ensure smooth triangle deformations of $\mathbf{S}$ into $\widetilde{\mathbf{S}}$. To do this, we consider the neighborhoods of the triangles in $\mathbf{S}$ and specify that their deformation be similar:

$$E_{ne}(\{\tilde{s}_i\}) = \sum_{j=1}^{m} \sum_{r\in\text{adj}(j)} ||\mathbf{Q_j} - \mathbf{Q_r}||_F^2, \tag{22}$$

where $\text{adj}(j)$ denotes the set of triangles sharing an edge with triangle j in $\mathbf{S}$, and m is the total number of triangles in $\mathbf{S}$.

Additionally, to avoid over-fitting, we penalize the presence of strong deformations for each triangle:

$$E_{id}(\{\tilde{s}_i\}) = \sum_{j=1}^{m} ||\mathbf{Q_j} - \mathbf{I}||_F^2, \tag{23}$$

where $\mathbf{I}$ is the identity transformation.

The final cost function for the fit is the sum of Eqs. (21-23):

$$E(\{\tilde{s}_i\}) = E_{lm}(\{\tilde{s}_i\}) + E_{ne}(\{\tilde{s}_i\}) + E_{id}(\{\tilde{s}_i\}) \tag{24}$$

Fig. 8 shows an example of our re-targeting result for the face.

8 Device Tracking and Environment Reconstruction

Our device is fitted with four outward-facing cameras that serve to track the motion of the wearer within their environment while simultaneously reconstructing their surroundings. This reconstruction capability is an important component for the overall capture scenario: the wearer's environment provides context for remote observers and greatly contributes to their sense of "being there." While device tracking is ultimately necessary for the system as a *motion capture* unit, the external reconstruction endows the device with the ability for *content capture*.

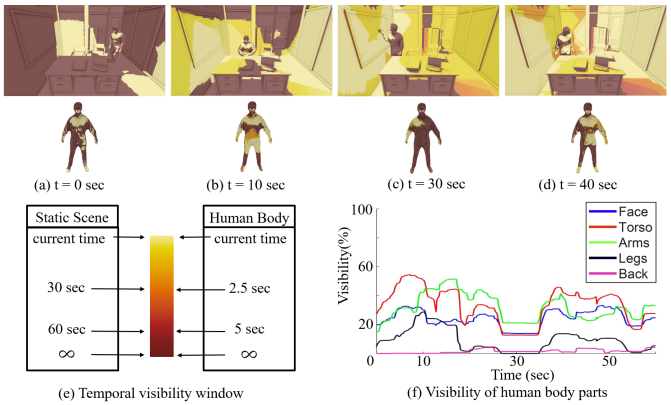

Fig. 10. Environment and body part visibility simulation. (a-d) Temporal visibility heat maps for the static scene (top) and human user's body (bottom). (e) Color coding heat map for (a-d). Surfaces are colored according to how recently they were visible to one of the head-worn cameras. (f) Time plot of visibility percentages for several parts of the simulated user's body.

In our prototype system, we perform environment capture using four synchronized views on the sides and back of the wearer's head, and process them off-line. From this multi-view imagery, we simultaneously estimate the motion of the camera rig and reconstruct the environment using COLMAP, the current state-of-the-art tool for incremental structure-from-motion (SfM) [52] and multi-view stereo (MVS) [53]. The process of SfM has three stages: feature extraction for individual images, feature matching between image pairs, and reconstruction. During reconstruction, images are iteratively registered to each other based on their feature correspondences; here, registration involves computing the rotation and translation of the image relative to the environment, as well as 3D scene points for the individual image features. Since our camera rig is pre-calibrated for both intrinsics and local extrinsics, we are able to obtain a to-scale registration of the cameras to the scene via SfM in an unsupervised fashion. Given these camera registrations, we use MVS to estimate a dense (pixel-wise) depth map for each image, and we then run depth-map fusion and subsequent surface meshing [32] to obtain the final environment model.

The outcome of this off-line processing is a textured 3D mesh depicting the user's environment, as well as information about where the user was standing and where they were looking relative to the environment at each time-point in the capture. When visualizing the capture in, *e.g.*, virtual reality, this information is directly used to place the animated reconstructed body model within the virtual environment.

9 Integration

The resulting face, body, and environment reconstructions are integrated to compose the entire scene (Fig. 9). First, the face vertices in the body pre-scan are replaced using Eq. (20). Then, the pre-scan is deformed using Eq. (9). The deformed pre-scan $\widehat{\mathbf{T}}_{\text{local}}$ in model space is localized to the environment coordinates using the estimated headset pose $C_t \in \mathbb{R}^{4\times4}$ at time t.

$$\widehat{\mathbf{T}}_{\text{global}} = C_t \begin{bmatrix} \mathbf{R}_M^{-1} & \mathbf{R}_M^{-1}\mathbf{J}_{\text{head}} \\ \mathbf{0} & 1 \end{bmatrix} \widehat{\mathbf{T}}_{\text{local}}, \tag{25}$$

where $\mathbf{R}_M$ is the rotation of the body model estimated during the skeleton alignment in Sect. 6.4 and $\mathbf{J}_{\text{head}}$ is the head joint position. $[\mathbf{R}_M^{-1}|\mathbf{R}_M^{-1}\mathbf{J}_{\text{head}}]$ reorients the pre-scan to its head joint at the origin in local space.

Because the feet are often occluded in our downward-facing views, the leg motions of the wearer are rarely detected. – the feet are modeled in 3D as located on the ground, exactly below the knees in Sect. 6.2. To compensate for this, we add a motion prior to the pre-scan deformation based on the norm of average velocity $V_t = ||d/\Delta_t||$ of head-track displacement d. Specifically, we capture a separate walking motion pose sequence $\{\theta_{\text{walk},t}\}$ that captures two full strides of an individual. This step sequence is looped continuously throughout our capture sequence. For a given frame t, the refined pose $\widehat{\theta}_t$ is estimated as

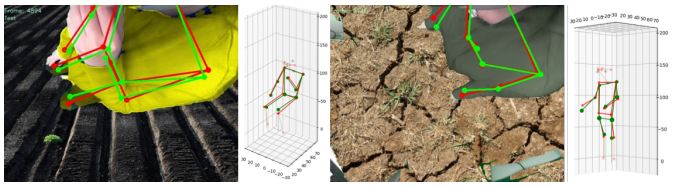

Fig. 11. Example 2D and 3D pose estimation results on our validation dataset. Red points and lines show the ground-truth joints positions and skeleton in the images, while those in green are our prediction results. Left: Sitting pose. Right: Walking pose. In each image, the background and shirt color have been synthetically augmented.

$$\widehat{\theta}_t = \alpha_t\, \theta_{\text{walk},\,t} + (1-\alpha_t)\, \theta_t\,;\ \alpha_t = \min(V_t, 1). \tag{26}$$

The blended pose $\widehat{\theta}_t$ is controlled by velocity V_t. When the user moves quickly, the influence of the walking motion increases. When the user stops walking, the motion becomes negligible. Fig. 6 shows a result of the pose blending.

10 Results

In this section, we present results for our body pose and facial expression estimation pipelines. We additionally showcase a possible use case for our system: virtual tours of a remote place (indoors and outdoors), with the wearer of our device acting as a tour guide.

10.1 Results for Body Visibility in Simulation

To explore our camera modeling approach, we simulated a room-sized environment with static objects such a whiteboard, a desk, and chairs. We then added a simulated user, animated over a 60-second sequence to perform actions such as sitting on a chair, getting up, and writing on the whiteboard. We modeled the cameras in a similar configuration as our physical prototype, with each simulated camera's horizontal field of view set to 90°. We used the method introduced by Chabra *et al.* [12] to model temporal visibility of a surface in the simulation:

$$v_t = \begin{cases} 1 & \text{if } s \text{ is visible from at least one camera at time } t \\ 1 - \frac{\Delta t}{\tau} & \text{if } s \text{ is hidden for a time } \Delta t < \tau \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

In our analysis, we set the temporal visibility threshold interval τ to 5 seconds for dynamic objects and to 60 seconds for static objects. The resulting temporal visibility v_t is shown in Fig. 10 as heat maps at 4 different time instants, with the brightest color representing polygons that were visible most recently. We also show the percentage of visible polygons over time for the virtual person's body. The noticeable drop in visibility around time $t = 30$ corresponds to the interval during which the person was writing on the whiteboard, remaining relatively motionless.

Our simulation results indicate that with our proposed camera arrangement, most of our dynamic scene is visible to at least one camera within reasonable visibility threshold intervals, which gives us confidence that our reconstruction approach can successfully reconstruct a near-static environment. However this simulation's results led us to use larger FoV lenses for body and face capture in our physical prototype (120 degrees horizontal) than in our simulation (90 degrees horizontal), and smaller FoV lenses for environment capture (62 instead of 90 degrees).

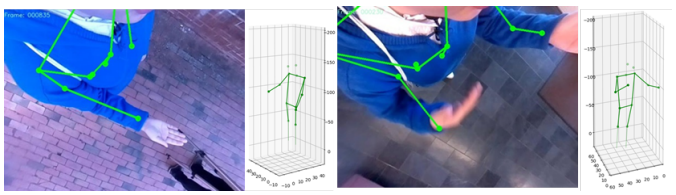

Fig. 12. Example 2D and 3D pose estimation results for our outdoor (left) and indoor (right) video tour scenes. Green points and lines show the predicted skeleton. Note that the mis-predicted right arm in the right image is corrected in 3D using our RNN.

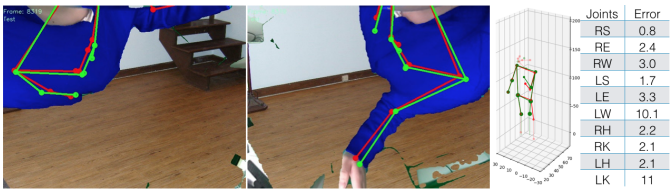

Fig. 13. 2D and 3D pose estimation result where the left wrist cannot be seen from the right-side camera, and the knees are barely visible in either camera. Such a situation can result in large errors for our system: the 3D error of left wrist is 10.13cm, while the error of the right wrist is 2.94cm.

10.2 Results for Body Pose Estimation

In addition to qualitatively evaluating our body pose estimation on demo data, we provide qualitative and quantitative analysis on a validation dataset that was captured in the same environment as the training dataset, independently but using similar motions. Fig. 11 shows results for the 2D joint detection and 3D pose estimation on two typical poses from the validation dataset: walking and sitting while gesticulating. Qualitatively, the results exhibit satisfactory alignment with the ground truth. In Fig. 12, we show qualitative results for the 2D joint detection and 3D pose estimation on our demo test dataset in both outdoor and indoor scenes. The indoor result shows an example where we obtained a reasonable 3D pose despite imperfect 2D detections (right arm in the right image).

Table 1 provides a quantitative analysis for our validation dataset, including 2D errors in joint detection and 3D errors in joint position estimation. For 2D detections, we report mean and standard deviation errors in pixels. The input images are 640×480 px. We generally observe skewed error distributions (not shown due to space restrictions), with the majority of detections closer to the ground truth than the mean. Sporadic large detection errors arise from false-positive maxima in the belief maps output by our CPM. These detection errors are typically corrected during our subsequent 3D prediction and motion smoothing. Regarding 3D skeleton errors, we evaluate the performance of two methods: 1) simple two-view triangulation using the known relative calibration of our downward-facing views, and 2) our proposed RNN approach. Our RNN approach has lower positional error for all joints, with average validation errors between 2cm and 4.7cm in Table 1. The result compares favorably to EgoCap [50], the existing system most similar to our own, for which average 3D joint position errors of 7 ± 1cm were reported. These averages roughly follow the general visibility of the joints in each view, with the hips and elbows having the lowest errors.

10.3 Results for Face Reconstruction

The top row of Fig. 14 shows our face landmark detection and model fitting results for both indoor and outdoor illuminations. Our face model has 66 total landmarks. Due to the limited visibility, in each view we detect the 10 midline landmarks and the 28 additional landmarks for each half of the face.

To quantitatively evaluate our face reconstruction result, we compute the distance between all 66 2D and projected 3D landmarks for the complete set of frames in our indoor (1760 frames) and outdoor (1250

Table 1. 2D and 3D joint estimation errors for our method. 2D: Mean and std. dev. pixel errors for detected 2D joints. 3D: Mean 3D distance (in cm) between the ground-truth and predicted joint positions for two-view triangulation from our body cameras (Tri.) and our recurrent approach (RNN). Notation: Shoulder (S), elbow (E), wrist (W), hip (H), and knee (K). R/L: Right/left joint.

		RS	RE	RW	LS	LE	LW	RH	RK	LH	LK
2D (px)	Avg	11	5.9	7.1	11	7.9	8.5	4.5	7.1	4.5	5.9
	Std	12	7.4	12	12	8.5	14	4.5	11	3.8	9.1
3D (cm)	Tri.	5.9	3.4	4.0	6.2	3.7	6.2	3.7	6.1	3.6	5.9
	RNN	3.7	2.9	3.3	4.3	3.0	4.7	2.1	4.0	2.0	3.9

Fig. 14. 2D face landmark detection and 3D facial fitting. White points in the first (indoor) and third (outdoor) columns show the 66 2D landmarks. The second and fourth columns show the mesh fitting visualization with all mesh vertices (green) projected into the images.

frames) virtual tour capture data. We run our alternating pose, shape, and expression optimization for 5 iterations. Over all frames for both views, the RMS error decreases from an average initial value of 16.38 px to an average final value of 3.57 px. To visualize our 3D fitting, we show the projection of the corresponding mesh onto the input imagery in Fig. 14. We observe that our projection fits the entire face accurately, including the neck and the ears.

Figure 7 provides two examples to demonstrate the final reconstruction result of combining video and audio. The first column shows the original image captured by the side cameras; the second and third column show separately the reconstruction results from video and audio. The final column shows the final result of combining video and audio. The audio result in the first row is unreliable due to silence and is ignored in the combined result. In the second row, the mouth is occluded, causing an unreliable video result, but the audio provides a plausible mouth shape in the combined result.

10.4 Application: Virtual Tour

To demonstrate the potential of our system for ego-capture scenarios, we used our device to record a short VR tour of the UNC Department of Computer Science. Acting as a tour guide, the wearer moves around the capture space and describes her surroundings. Our system then reconstructs the wearer's motions and environment, creating a dynamic 3D representation that remote users can experience in VR, as if they were getting an in-person tour. Fig. 15 shows example frames from the indoor portion of our tour, and a view of our outdoor portion is shown in Fig. 9. Videos from our virtual tour displayed in an HTC Vive are available in the supplementary material.

For real-time visualization, the animated sequence of per-frame body poses is built into an Alembic geometry cache [2] using Autodesk Maya 2018, which is then represented as an animated non-skeletal 3D mesh in Unreal Engine 4. The viewer wearing the head-mounted display is provided controller-based locomotion in addition to physical locomotion to walk with the reconstructed tour guide in a reconstructed virtual environment that is larger than the available physical environment.

11 System Limitations and Future Work

Our current system captures the raw data in real-time and processes it off-line. With advances in computing power and GPU-enabled parallelization, we can reasonably expect that our processing techniques will be accelerated to real-time in the near future. We also aim to integrate the capture and processing components of our system into a wearable package, *e.g.*, a backpack connected to the headset, in order to allow telepresence-type interactions. To that aim, we plan to employ new and existing techniques to cull, compress, and transmit the reconstructed data. We also plan to explore reconstruction and meshing techniques to obtain better 3D meshes for VR visualization.

A key limitation in our current evaluation is that our pose estimation approach is user-specific. Training the CNN and RNN with data from multiple users will make our approach more broadly applicable, and we anticipate that increasing the amount of and variation in training data will improve the modeling of unseen joints, such as the left arm in Fig. 13 or the ankles. Increasing the user base will also allow us to account for a variety of outfits, compared to the two shirts (long- and short-sleeved, with color augmentation) used in our experiments.

With respect to device tracking and environment reconstruction, the main direction for future work is to have the system run in real time and reconstruct dynamic environments. Our plan is to use our rolling-shutter high-frequency tracking approach [6] combined with SLAM

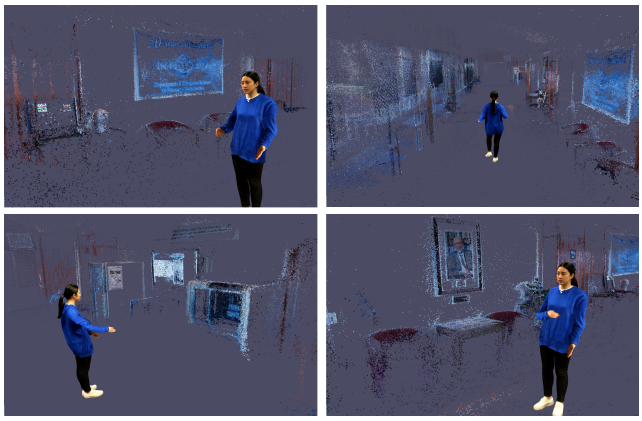

Fig. 15. Four frames from the indoor section of our virtual tour.

for long-term stabilization and reconstruction [21, 61]. One currently unavoidable limitation is that we can only reconstruct the part of the scene observed at some time by the headset-mounted cameras. Adding more cameras will help: in particular, front-facing external cameras would improve user comfort, since it is easier for the user to know what parts of the environment have been captured when those views line up with their line of sight.

The user reconstruction part of our system also offers directions for future research. Our current body re-targeting technique uses a body model with limited degrees of freedom. We plan to employ a state-of-the-art model such as SMPL [37] to obtain more natural body movements. Similarly, our current face re-targeting approach uses deformation transfer, which results in limited facial expressions. We expect a rigged face model to yield more-natural-looking facial animations. Another research direction is to fully model hand and finger motions, and enable capture and reconstruction of arbitrary objects being carried or manipulated. Finally, we would like to explore using mirrors to allow reconstruction of the user's body model directly from images captured using the headset-mounted cameras, rather than requiring a separate body pre-scan process.

12 Conclusion

We proposed a new approach for the 3D capture of an individual and their environment that relies not on any instrumented environment, but only on cameras and sensors worn by the individual. This approach allows for the reconstruction and communication of experiences from any location, indoors or out. With a vision of the fully mobile capture systems of tomorrow, we have outlined the key technological advances necessary for capturing the wearer's body pose, facial expression, and limbs—entirely from near-body views—and we have shown how the surrounding environment can be reconstructed using outward-facing views, which enables completely ego-centric content capture. Our results demonstrate workable methods that leverage state-of-the-art machine learning approaches to overcome the profound problems of poor visibility for body capture from head-worn cameras.

We envision our prototype device to one day shrink to the size of everyday prescription eyeglasses and be worn as such. This all-in-one form factor is key to enabling the ubiquitous use of user-centric 3D content capture, virtual tours, and 3D telepresence for a large variety of users and scenarios.

Acknowledgments

The authors thank Jim Mahaney for his extensive assistance with the physical set-up of the prototype capture system. This research is partially supported by National Science Foundation (NSF) Grants CNS-1405847, CHS-1718313, IIS-1423059, and IIS-1405847, by a gift from CISCO Systems, and by the BeingTogether Centre, a collaboration between Nanyang Technological University (NTU) Singapore and University of North Carolina (UNC) at Chapel Hill. The BeingTogether Centre is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centres in Singapore Funding Initiative.

REFERENCES

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Alembic computer graphics interchange framework. `http://www.alembic.io/`. Accessed: 2017-03-15.

[3] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM transactions on graphics (TOG)*, vol. 22, pp. 587–594. ACM, 2003.

[4] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 408–416. ACM, 2005.

[5] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *3DPVT*. Atlanta, GA, USA, June 2008.

[6] A. Bapat, E. Dunn, and J.-M. Frahm. Towards kilo-hertz 6-dof visual tracking using an egocentric cluster of rolling shutter cameras. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2358–2367, 2016.

[7] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[8] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43, 2014.

[9] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[10] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[11] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[12] R. Chabra, A. Ilie, N. Rewkowski, Y.-W. Cha, and H. Fuchs. Optimizing placement of commodity depth cameras for known 3d dynamic scene capture. In *Virtual Reality (VR), 2017 IEEE*, pp. 157–166. IEEE, 2017.

[13] X. Chen, Y. Guo, B. Zhou, and Q. Zhao. Deformable model for estimating clothed and naked human shapes from a single image. *The Visual Computer*, 29(11):1187–1196, 2013.

[14] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[15] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 98. ACM, 2008.

[16] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *Proc. CVPR*, 2007.

[17] M. Dou and H. Fuchs. Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In *IEEE Virtual Reality (VR) 2014, Best short paper award*, pp. 39–44. IEEE, 2014.

[18] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)*, 35(4):114, 2016.

[19] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 3d scanning deformable objects with a single rgbd sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 493–501, 2015.

[20] D. A. Drexler and I. Harmati. Joint constrained differential inverse kinematics algorithm for serial manipulators. *Periodica Polytechnica. Electrical Engineering and Computer Science*, 56(4):95, 2012.

[21] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Mar. 2018.

[22] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pp. 834–849. Springer, 2014.

[23] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1746–1753. IEEE, 2009.

[24] Google JUMP VR. `https://vr.google.com/jump/`. Accessed: 2017-03-15.

[25] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Computer Vision and Pattern Recognition (CVPR)*, pp. 3287–3295, 2015.

[26] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In A. Fitzgibbon, ed., *European Conf. on Computer Vision (ECCV)*, LNCS 7577, Part IV, pp. 242–255. Springer-Verlag, Oct. 2012.

[27] itseez3d scanning app. `https://itseez3d.com/`. Accessed: 2017-03-15.

[28] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, and A. Davison. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568. ACM, 2011.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678. ACM, 2014.

[30] H. Jiang and K. Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. *arXiv preprint arXiv:1603.07763*, 2016.

[31] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):94, 2017.

[32] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013.

[33] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 167–176. ACM, 2012.

[34] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph.*, 32(6):187, 2013.

[35] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *Proc. ICCV*, pp. 167–174. IEEE, 2009.

[36] Y. Liu, F. Xu, J. Chai, X. Tong, L. Wang, and Q. Huo. Video-audio driven real-time facial animation. *ACM Transactions on Graphics (TOG)*, 34(6):182, 2015.

[37] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[38] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, oct. 2011.

[39] A. Maimone and H. Fuchs. Reducing interference between multiple structured light depth sensors using motion. In *IEEE Virtual Reality (VR) 2012, Best short paper award, 2012 IEEE*, pp. 51–54. IEEE, 2012.

[40] Matterport commerical room scanning system. `http://matterport.com/`. Accessed: 2017-03-15.

[41] Microsoft Hololens. `https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details`. Accessed: 2017-03-15.

[42] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[43] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pp. 127–136. IEEE, 2011.

[44] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 343–352, 2015.

[45] K. Olszewski, J. J. Lim, S. Saito, and H. Li. High-fidelity facial and speech animation for vr hmds. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2016)*, 35(6), December 2016.

[46] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, and M. Dou. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 741–754. ACM, 2016.

[47] L. Pishchulin, S. Wuhrer, T. Helten, C. Theobalt, and B. Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 67:276–286, 2017.

[48] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*, pp. 33–47. Springer, 2014.

[49] Reconstructme realtime scanning software. `http://reconstructme.net/`. Accessed: 2017-03-15.

[50] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: Egocentric marker-less motion capture with two fisheye cameras. In *ACM Transactions on Graphics (TOG)*. ACM, 2016.

[51] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[52] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

[53] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518. Springer, 2016.

[54] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 519–528. IEEE, 2006.

[55] M. Shah, R. D. Eastman, and T. Hong. An overview of robot-sensor calibration methods for evaluation of perception systems. In *Workshop on Performance Metrics for Intelligent Systems*, pp. 15–20. ACM, 2012.

[56] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins. Motion capture from body-mounted cameras. In *ACM Transactions on Graphics (TOG)*, vol. 30, p. 31. ACM, 2011.

[57] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[58] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, vol. 25, pp. 835–846. ACM, 2006.

[59] J. Starck and A. Hilton. Surface capture for performance-based animation. *Computer Graphics and Applications*, 27(3):21–31, 2007.

[60] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, 23(3):399–405, 2004.

[61] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *IEEE Trans. on Visualization and Computer Graphics*, 18(4), 2012.

[63] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 97. ACM, 2008.

[64] R. Wang, M. Schwörer, and D. Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *International Conference on Computer Vision (ICCV), Venice, Italy*, 2017.

[65] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4732, 2016.

[66] A. Weiss, D. A. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *Int. Conf. on Computer Vision (ICCV)*, 2011.

[67] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM Transactions on Graphics (TOG)*, 32(6):161, 2013.

[68] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 532–539, 2013.

[69] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance capture of interacting characters with handheld kinects. In *Proc. ECCV*, pp. 828–841. Springer, 2012.

[70] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 145–152. IEEE, 2013.

[71] P. Zhang, K. Siu, J. Zhang, C. K. Liu, and J. Chai. Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *ACM Transactions on Graphics (TOG)*, 33(6):221, 2014.

[72] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 146–155, 2016.