BodySLAM: Opportunistic User Digitization in Multi-User AR/VR Experiences

Karan Ahuja
Carnegie Mellon University
Pittsburgh, PA, United States
kahuja@cs.cmu.edu

Mayank Goel
Carnegie Mellon University
Pittsburgh, PA, United States
mayank@cs.cmu.edu

Chris Harrison
Carnegie Mellon University
Pittsburgh, PA, United States
chris.harrison@cs.cmu.edu

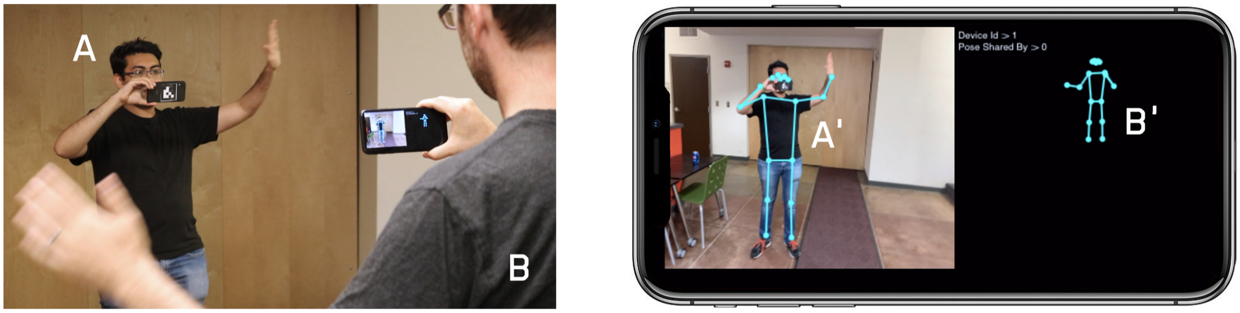

Figure 1: In this example scene (left image), two users (A and B) face one another, perhaps playing a mobile AR game where full-body tracking could be valuable for expressive input. Unfortunately, neither phone is able to see – and thus digitize – its owner's body. However, User B's phone (right image) can see User A (and vice versa) through its rear facing camera. BodySLAM uses this view to capture and digitize the body, hands and mouth of User A and shares the data (visualized as A'). User A does the same for User B, providing ad hoc full-body tracking (B') without having to instrument either the user or environment.

ABSTRACT

Today's augmented and virtual reality (AR/VR) systems do not provide body, hand or mouth tracking without special worn sensors or external infrastructure. Simultaneously, AR/VR systems are increasingly being used in co-located, multi-user experiences, opening the possibility for opportunistic capture of other users. This is the core idea behind BodySLAM, which uses disparate camera views from users to digitize the body, hands and mouth of other people, and then relay that information back to the respective users. If a user is seen by two or more people, 3D pose can be estimated via stereo reconstruction. Our system also maps the arrangement of users in real world coordinates. Our approach requires no additional hardware or sensors beyond what is already found in commercial AR/VR devices, such as Microsoft HoloLens or Oculus Quest.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; *Interaction techniques*; Gestural input.

SUI '20, October 31-November 1, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-7943-4/20/10.
https://doi.org/10.1145/3385959.3418452

KEYWORDS

Virtual Reality, Augmented Reality, Mixed Reality, Body Pose, Motion Capture, Hand Gestures, Facial Expression.

ACM Reference Format:
Karan Ahuja, Mayank Goel, and Chris Harrison . 2020. BodySLAM: Opportunistic User Digitization in Multi-User AR/VR Experiences. In *Symposium on Spatial User Interaction (SUI '20), October 31-November 1, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3385959.3418452

1 INTRODUCTION

Handheld controllers, offering several buttons and six degree-of-freedom tracking, are the most common input approach seen in today's augmented and virtual reality (AR/VR) systems (e.g., HTC Vive [42], Oculus Rift [33]). Of course, there are many other facets that could be valuable to digitize, including user body pose, facial expression, skin tone and apparel. Unfortunately, very few AR/VR systems capture these dimensions, and when they do, it is most often via special worn sensors (e.g., instrumented gloves [19], additional cameras mounted on the headset [23]). Alternatively, external infrastructure can be deployed (e.g., multiple room-mounted cameras) that capture body pose without having to instrument the user [33, 42].

In this work, we take advantage of an emerging use case: co-located, multi-user AR/VR experiences. Although nascent, the application space is already diverse (see Video Figure), ranging from co-located 3D modeling [7, 38] and AR-augmented collaborative spaces [29, 36], to tele-medicine and multi-player games [3, 8].

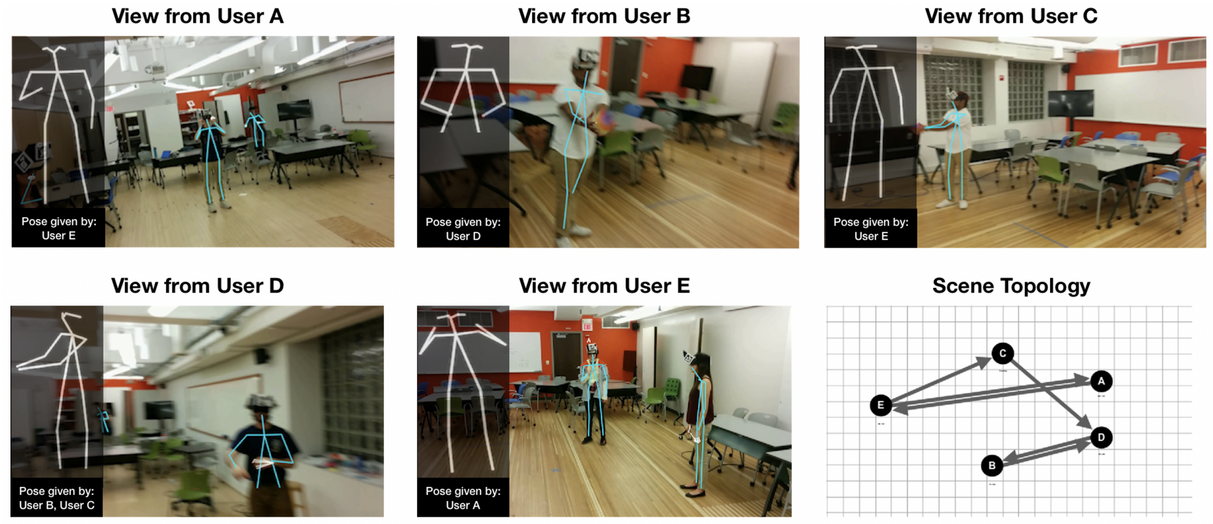

Figure 2: Example camera views (A-E) from co-located users wearing AR headsets, which BodySLAM uses to digitize the body pose, hand gesture, mouth state and apparel of participants. This data is then relayed back to users (shaded insets), offering advanced functionality without the need for worn instrumentation. BodySLAM also generates a scene topology (bottom right).

Our system makes it possible to have shared and ad-hoc extended reality experiences in dynamic environments without the need for any additional hardware or setup (Figure 1. In such contexts, participants are often able to see each other's bodies, hands, mouths, apparel, and other visual facets, even though they generally cannot see their own bodies. Using the existing outwards-facing cameras on smartphones and AR/VR headsets (e.g., Microsoft HoloLens, Oculus Quest), these visual dimensions can be opportunistically captured and digitized, and then relayed back to their respective users in real time (enabling e.g., full-body motion capture and hand gesture recognition) without any special instrumentation. This is the key insight that motivated our work on BodySLAM.

Our system name was inspired by SLAM (simultaneous localization and mapping) techniques for mapping unknown environments [10, 13]. In these systems, many viewpoints are used to reconstruct the geometry of the environment. In a similar vein, BodySLAM uses disparate camera views from many participants (Figure 2, User views A-E) to map the geometric arrangement of other users in the environment (Figure 2, bottom right), as well localize the capturing user's position in the scene. Our system also captures fine-grained details, such as hand and mouth pose, as well as visual attributes such as apparel. When a person is seen by two or more users (Figure 2, views of User D from Users B and C), we also estimate 3D pose data (Figure 4).

After reviewing related work, we then describe our proof-of-concept implementation. We evaluate our system in a multipart user study, incorporating two tasks and two group sizes. To explore how BodySLAM might scale to larger numbers of people, we ran software simulations in virtual rooms. Although we did not build any example applications, we note that digitization of bodies has been well motivated in prior work, including uses in social VR and telepresence [29], entertainment and gaming [3], and 3D manipulation [38].

2 RELATED WORK

There has been significant prior work in motion capture and reconstruction of scenes. Thus, we first review the literature in body digitization – sensed using both external and on-body techniques. We then move to multiview geometry and similar reconstruction techniques, concluding with efforts in the domain of creating shared AR/VR experiences.

2.1 Body Digitization

The desire to digitize bodies is not new and there are popular commercial systems (such as Vicon [41] and OptiTrack [30]) that provide very accurate tracking of the body, hands and face using optical markers. In VR settings, the use of external trackers is prevalent, including systems such as HTC Vive's Lighthouse tracking system and the external sensors used by the Oculus Rift[33, 42]. Camera-based approaches have made tremendous recent progress, driven by advances in deep learning, e.g., OpenPose [11], PoseNets [31] and V-NECT [25]. More specifically in the area of virtual reality, FaceVR [39] uses an external RGB and depth camera to capture face and mouth pose. Non-optical methods have also been considered for external pose tracking, including radio frequency (RF) [47] and capacitive [46] sensing.

Alternatively, instead of relying on room-borne infrastructure, the user can be instrumented, which has advantages in terms of mobility. For whole body capture, suits using IMU's [14], body-mounted cameras capturing optical flow [37], head-mounted fish-eye cameras viewing the body [32, 40, 44] and mechanical linkages to the limbs [26] have all been considered. There is also a plethora of research on first-person hand pose and gesture recognition, including vision [22], acoustic [4] and electrical methods [45, 49]. For mouth pose estimation, Li et al. [23] used a camera mounted to the underside of the headset to capture mouth pose. EyeSpyVR [2] uses the front-facing camera of the smartphone (placed into a low-cost

VR headset) to capture eye movements and blinks. MeCap [1] used the outward-facing camera of a VR headset — in concert with a hemispherical mirror attachment — to capture the body, hand, and mouth pose of the wearer.

2.2 Scene Reconstruction

There has been a plethora of research in the robotics and computer vision community on scene and multiview reconstruction. These include the aforementioned simultaneous localization and mapping (SLAM) based techniques that help place objects in an unknown environment [13]. Visual SLAM approaches can rely on multiview stereo [10], visual odometry [28] and structure from motion [20]. Similarly, BodySLAM uses multiple views to reconstruct user topology, and multi-view stereo to estimate 3D pose information.

2.3 Collocated AR/VR

The HCI community has lightly explored collocated AR/VR experiences through different enabling technologies. For instance, Side-by-Side [43] used handheld projectors (emitting both visible and infrared light) to create multi-user experiences, such as gaming. SynchronizAR [17] demonstrated collaborative gaming through a smartphone AR app with indoor localization. HoloRoyale [34] provides a comprehensive design space exploration of large-scale high-fidelity collaborative AR experiences. Commercially, Apple's ARKit allows for shared AR experiences, where many users can join using their own smartphones. Finally, Imaginary Reality Games [5] creates virtual playgrounds for participants to play sports.

3 SYSTEM ARCHITECTURE

We now describe the three high-level components that form the basis of our system.

3.1 Exemplary AR/VR Prototypes

We created three exemplary prototypes that cover different AR/VR modalities. These include an iPhone XR smartphone for mobile (i.e., "passthrough") AR (Figure 3, left), a HoloKit headset [16] for AR (center), and a Google Cardboard [12] for VR (right). We fitted each of these devices with printed 7x7 cm ArUco tag [27] for spatial tracking. For our two headsets, unique tags are placed on all four sides for user identification from any viewpoint. For our mobile AR prototype, a single tag is placed on the back of the phone.

3.2 Client Software

BodySLAM employs a client-server paradigm. We created two client implementations, with different approaches for computation. First is a smartphone app that streams camera data using RTSP over WiFi to our backend server for computer vision processing. Although this architecture can leverage greater CPU and GPU resources, it incurs a latency penalty. On average, it takes 110 ms for video to reach the server, followed by 154 ms of processing, and a further 10 ms for processed data to be returned to the client. In total, our system latency is roughly 300 ms at a frame rate of 9 Hz.

Our second client implementation runs all computer vision on the smartphone. For this, we use PoseNets [31], a lightweight pose estimator that reports body keypoints. Although missing hand and face tracking capabilities, it nonetheless offers a useful proof-of-concept for a more self-sufficient system. ArUco detection and tracking also runs locally, using JSArUco [18]. Processed data is then sent to the server for final processing (e.g., 3D pose via stereo correspondence) and distribution to other clients. End-to-end latency is around 150ms, and the full stack runs at 12Hz on an iPhone XR (see demo in Video Figure). Given the tremendous strides phone manufacturers are making with hardware accelerated deep learning, a fully featured and full framerate pose tracking pipeline should be possible in the near future.

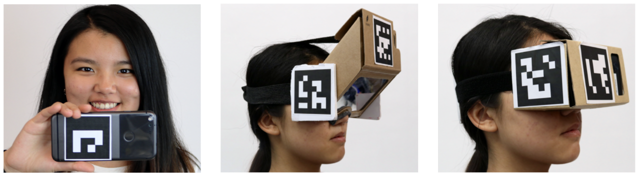

Figure 3: BodySLAM works across different AR/VR modalities, including mobile "passthrough" AR (left), head-worn AR (center; HoloKit), and VR (right; Google Cardboard).

For our VR client software and our two AR clients, we sync the overlaid pose of in-view people with the gyroscope of the smartphone, so that the latency of the real-time view (limited by camera FPS in our VR and Mobile AR implementations) is decoupled from the FPS of pose tracking. For demonstration, our interface is chiefly a passthrough view, on top of which we superimpose the wearer's body, hand, face, skin and apparel if provided by others.

3.3 Backend

Our backend (i.e., cloud) server is a 12-core Intel Core i7 with three Nvidia 1080 Ti GPUs. The server software is a multi-threaded listener, which listens for connections and data from clients. The server maintains global state and multicasts it back to registered clients. For our AR and VR headset prototypes, the client smartphones stream camera frames over WiFi to our backend for computer vision processing. For our mobile VR prototype, all computer vision processing happens on the smartphone and only pose data is transmitted to the server.

4 PROCESSING PIPELINE

We now describe the core processes that form our main computer vision and machine learning pipeline.

4.1 Identifying and Localizing Users

The first stage of our computer vision pipeline is to find and track participants in each camera view. As noted in Apparatus, users are assigned unique ArUco tags. For our headsets, four tags are used, which identify the user and the side of the head. The front-facing ArUco tag is considered the user's origin, and the left, right, and rear tags have known rotational and spatial offsets to this origin. If multiple tags from one user are visible, we use the tag most-frontal to the viewer, as this provides the most stable tracking result. In our self-contained mobile AR client we use JSArUco for tracking; for our backend computer vision pipeline we use OpenCVs ArUco

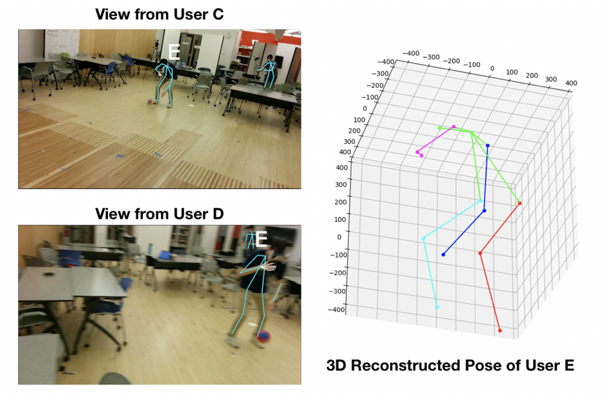

Figure 4: Users C and D see a common user E. These two views can be used to estimate E's 3D pose via stereo correspondence.

marker implementation [9]. In both cases, we use a time-to-live of one second to provide stability against momentary losses in tracking (e.g., occlusion, motion blur). The result of this process is the relative 3D position and orientation of every user that is seen by others, with wearers as the origin.

4.2 Scene Topology

As users cannot see themselves, it means everyone is given their position and pose data by someone else. This pairwise information is used to create a directional graph, with distance and angle information. Note that users that see no one can still be added to the graph if at least one person sees them. Also, a user that is not seen by anyone can still be added to the graph if they see at least one person, as show-cased in Figure 2.

To create a "global" scene topology, our software finds the most complete graph with a depth-first search. Although links are directional in reality (i.e., one person sees another), the origins can be inverted to make them functionally bi-directional. We move from one person to another, building a unified 3D coordinate system by multiplying individual user's transformation matrices as they are added. This way the whole operation is limited to the order of O(V+E) where V is the number of people and E is the number of people seen by each person.

4.3 Body, Hand and Mouth Pose

Once participants are localized in a common coordinate system, our next step is to extract fine-grained body details. For this, we use PoseNets [31] on our mobile AR client and OpenPose [11] running on our backend server for our worn VR and AR clients. Both packages provide body keypoints, however only OpenPose provides hand and face keypoints.

4.4 2D Pose via Multi-Viewpoint Selection

If a user is seen by exactly one person, we must make best use of this limited keypoint data. Unfortunately, this view is rarely frontal, and so we attempt to transform it to a frontal view using the algorithm in Kostrikov et al. [21], which estimates 3D keypoints given 2D keypoints. With this estimated 3D output, we can rotate to a synthetic frontal view.

In multiuser settings, where a user might be seen by several people, our software selects the most frontal of the views for this transformation. Although other views might contain more tracked keypoints, we found in practice that the more frontal the view, the truer the pose following rotation.

4.5 3D Pose via Stereo Correspondence

In cases where users are seen by two or more people, we can estimate 3D body pose. For this, we take the pair that minimizes the reprojection error of the 3D pose back onto the image. For each selected pair, we move all our cameras to a homogeneous coordinate frame using the 6 DOF data from users' ArUco tags. We then run a 3D point triangulation [15] to estimate the 3D position for each body keypoint. An example output from this process is shown in Figure 4. As before, we rotate this view to be frontal when presenting the data to the wearer in the client app.

4.6 Skin and Apparel Color Estimation

For skin color, we extract patches from the neck, hands and lower face (which are the least likely to be occluded by clothing) and compute the median color. For shirts, we take an image patch between the hips and torso keypoints, and similarly compute the median the color. For pants, we extract a patch from above the left and right knees of a user.

4.7 Hand Gesture Recognition

As a proof-of-concept of a downstream use of pose data, our system performs real-time hand gesture recognition. We support five gestures: okay, thumbs up, high five, fist and peace sign (Figure 5). For machine learning features, we calculate unit direction vectors from all hand keypoints to the wrist. These converts our data from a higher-dimensional pixel-space to a scale invariant feature space. These are fed to a standard multiclass classifier (MLP; sklearn, default parameters).

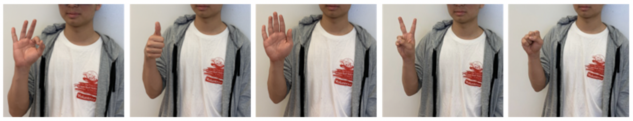

Figure 5: Hand gestures from left to right: okay, thumbs up, high five, peace sign and fist.

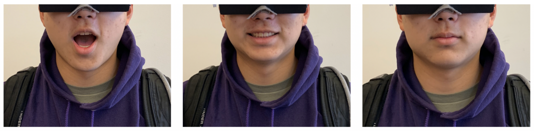

Figure 6: Mouth states from left to right: mouth open, smile and neutral.

4.8 Mouth State Recognition

We also perform mouth state recognition, supporting neutral, smile and mouth open gestures (Figure 6). Similar to our hand gesture process, we compute unit direction vectors for all mouth keypoints using the left mouth corner as the origin. As before, we pass this feature vector to a MLP classifier (sklearn, default parameters) for prediction.

5 USER STUDY

To better evaluate the performance of BodySLAM in different use contexts, our study procedure varied the activity (static vs. dynamic), distance between participants (2 or 4 meters) and group size (one-on-one vs. small group). As a proof-of-concept apparatus we used our Holokit AR prototype, so that participants could see each other and BodySLAM data. Our backend software saved processed camera frames for later evaluation. All classifiers were pre-trained on five independent users and ran live during data collection.

5.1 Static Activity

We chose a meeting context as an exemplary static activity. In our one-on-one condition, this was equivalent to a standing, face-to-face discussion. For this, we recruited three pairs of participants, and had them repeat our study procedure at 2 and 4 meters apart. At each distance, we asked both participants to perform the following actions sequentially, in a random order:

- *10 body gestures*: hands by side, right hand raised, left hand raised, arms crossed, hands behind head, arms stretched horizontally, hands on hips, sitting down with arms at rest, sitting down with arms crossed, and sitting down with chin resting on hands.
- *5 hand gestures* (Figure 5): okay, thumbs up, high five, fist and peace sign.
- *3 mouth states* (Figure 6): neutral, smile and mouth open.

Participants saw their live pose and gesture classification results in the AR overlay, which they judged and verbal stated to be correct or incorrect, which was recorded by the experimenter. Participants pairs repeated the two distance conditions three times each, for a total of six collection sessions. For our small group condition, we mimicked a conference room setting. We recruited two groups of 5 participants, who were seated around a table. Participants completed the same 10 body gestures, 5 hand gestures and 3 face gestures using the same procedure above.

During the study, BodySLAM also captured shirt, pant and skin color. After completing all sessions, we had users select their own skin color on a printed Fitzpatrick scale [35], which we later compared to BodySLAM's estimate. To assess the quality of shirt and pant color estimation, the extracted colors were shown to participants, who judged it as either accurate or inaccurate.

5.2 Dynamic Activity

As an exemplary dynamic activity, we used a ball game (Figure 2), where participants were told to achieve the highest number of consecutive passes without dropping the ball. After fitting participants with our Holokit AR prototype, we let them play for five minutes, during which BodySLAM ran continuously on all headsets. Distance between participants varied as they moved around the space, as did their head direction and pose, especially when having to pick up dropped balls. For our one-on-one condition, we recruited three new pairs of participants to play our game. For groups, we recruited two sets of five players.

6 RESULTS

BodySLAM tracks many user dimensions, and as such, we break the discussion of our results into four parts: user registration in the scene topology, keypoint tracking, gesture recognition, and skin/apparel extraction. As noted previously, all classifiers were trained before the study, and thus all accuracy numbers reported here are cross-user results.

6.1 User Identification and Tracking

Across all conditions, the average percentage of participants captured and registered in the scene topology was 95.2%. In our one-on-one conditions, both static and dynamic activities, registration was 100%. Even when one user had to bend down to pick up a dropped ball, the other user was able to capture them and add them to the global scene topology. Surprisingly, this performed better than our static small group condition (89%). In the latter setting, users were sitting fairly close, and often only captured one or two other participants in the camera's field of view. Depending on the group foci, this sometimes led to disjoint graphs (e.g., two participants looking at each other, but not seen by anyone else). The group ball game had the worst registration performance (87%), which we found was chiefly due to high motion blur causing ArUco tracking to fail.

6.2 Body, Hand and Mouth Keypoints

We considered capturing 3D ground truth keypoints for users' bodies, hands and mouths using a professional optical tracking system. This required dozens of markers to be worn by our participants, which was cumbersome to setup and prone to breakage in our ball game condition. We also found that the markers interfered with our computer vision pipeline, especially on the hands and mouth. We instead decided to avoid instrumenting our participants and use human annotators to post hoc code the live tracking output from our system.

For our one-on-one conditions, we extracted one frame every second (image with keypoints overlaid), yielding 1184 frames. For our small group conditions, we randomly sampled 1000 frames. These frames were split equally between two annotators, who coded each frame as correctly or incorrectly registered to all users visible in the view. A correct registration required 1) more than 80% of visible keypoints in the frame to be detected and 2) that all keypoint centers intersected with their respective body joint. Overall registration accuracy was 95%, 93% and 83% accuracy for body, hand and mouth keypoint registration. Figure 7 breaks this out per condition.

6.3 Self-Assessed Body Pose Quality

As described previously, participants were shown their body pose (provided by other participants via BodySLAM) in their AR headsets. In our static activity conditions, participants were asked to perform one of 10 possible body poses. The experimenter then verbally

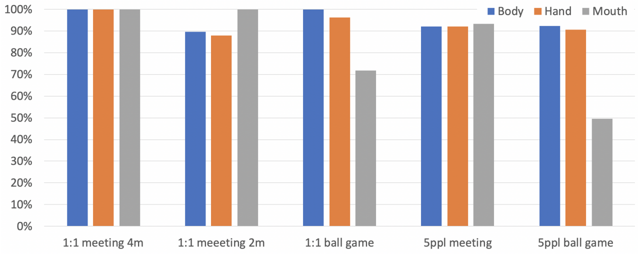

Figure 7: Percentage of body, hand and mouth keypoints correctly registered (all study conditions combined).

asked, "does the pose you see on your screen match the pose you are holding." In 94% of instances, participants agreed. Failure cases were usually due to gross keypoint registration errors.

6.4 Hand and Mouth Gesture Recognition

BodySLAM uses hand and mouth keypoints for multiclass gesture classification (Figures 5 and 6). In our static activity conditions, we found a mean hand classification accuracy of 88% and a mean mouth state recognition of 91%. Unlike body pose, which benefits from greater distance between participants (such that the whole body is visible), hand and mouth classification benefit from being closer in order to provide sufficient camera resolution to resolve fine details (Figure 8).

6.5 Skin Color and Apparel Detection

As described above, participants selected their own skin color from a printed Fitzpatrick scale [35] during the study. We compared this number to BodySLAM's skin color estimate and found a mean error of 0.7 (SD=1.0). For pant and shirt color, all of our participants rated BodySLAM's estimated colors as accurate.

7 SIMULATION STUDY

To explore how BodySLAM might scale to larger spaces and numbers of people (e.g., conferences, stadiums), we ran software simulations in virtual rooms. We tested different virtual room sizes: 3x3, 10x10, 30x30 and 100x100 meters). We also varied the number of people in the room, as well as their orientation behavior (random body orientations or common focus body orientations – see Figure 9 for examples).

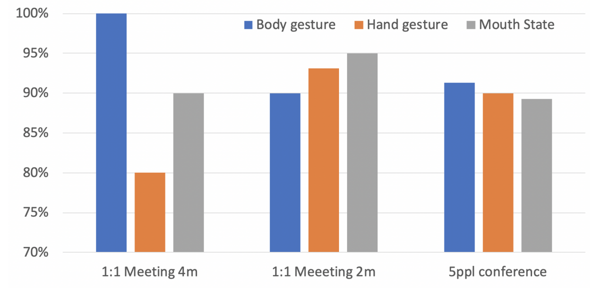

Figure 8: Participant-assessed accuracy of their body pose, along with automatic classification accuracy of hand and mouth gestures.

We used a 90° virtual camera field of view, matching an average mid-tier smartphone camera. Maximum body registration range was set to 14 meters, which we found to be the practical limit of ArUco tag detection at HD camera resolution. We model users as 50 cm circles that cannot be closer than 50 cm to one another, though we note that we do not model occlusion. Each combination of parameters was simulated 100 times, with detection statistics averaged. Results can be seen in Figure 10 broken out by random (A) and common (B) focus body orientations.

8 LIMITATIONS

The most immediate limitation of our approach is its heavy computational requirement. Running computer vision (pose modes and ArUco detection) on mobile hardware is taxing. Our process runs at around 12 FPS on an iPhone XR, much slower than native camera frame rate, and more critically, has a noticeable lag for users (150ms; see Section 3.2). Note that this lag applies to BodySLAM data only, and not the AR/VR graphics, which can run at full frame rate. Nonetheless, a mismatch between e.g., a user seen in AR and their body data could induce discomfort. Processing latency can also produce incorrect 3D pose estimations due to out of sync stereo correspondences. Fortunately, smartphone manufacturers are increasingly including hardware accelerated deep learning capabilities, which should improve performance in the coming years. For example, BlazePose [6] makes use of such optimizations and can run at above 30 Hz on modern smartphones.

We also note that BodySLAM is limited by the field of view of the rear-facing camera and occlusion in the environment. A user can lose pose tracking if they are not in the field of view of any other user. The likelihood of this occurring can be decreased by the use of ultra-wide-angle cameras, which are becoming more common in the market (e.g., Samsung S10 has 123° FOV, iPhone 11 has 120°).

Another issue with our current design is that the apparel color estimation can only work for apparels with single colors. Thus large printed designs and multi-colored apparels will lead to decrease in accuracy. In such cases a texture based approach might work better

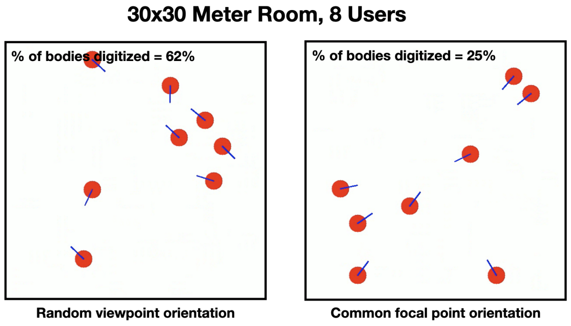
30x30 Meter Room, 8 Users
% of bodies digitized = 62%
% of bodies digitized = 25%
Random viewpoint orientation
Common focal point orientation

Figure 9: Two example simulated rooms, one with random body orientations (left) and common focus body orientations (right). See also Figure 10 for accuracy results across rooms of different sizes.

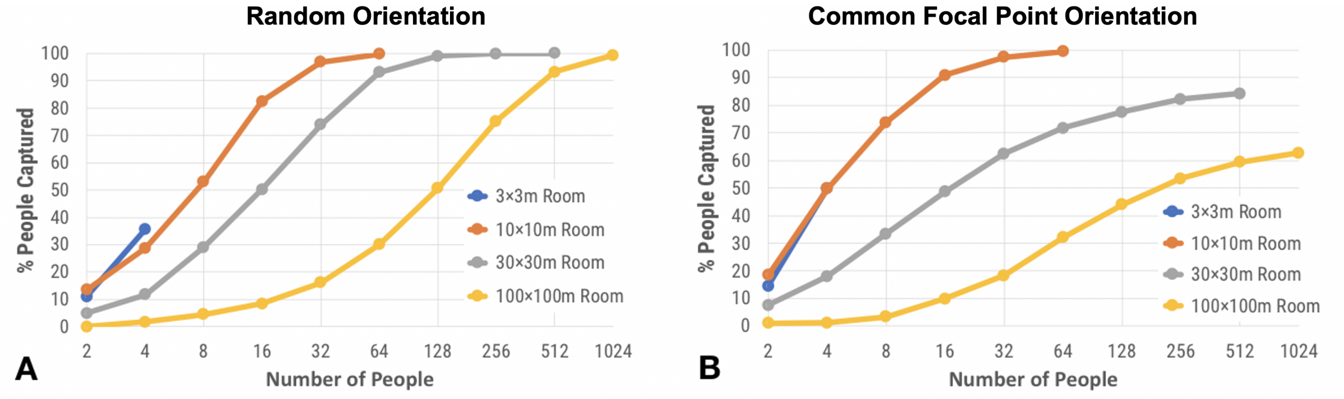

Figure 10: Percentage of bodies captured for random (A) and common focus body orientations (B) in simulated rooms of different size and occupant count.

than color estimation. Furthermore, since the color of the apparel is extracted based on joint locations, it is more likely to work on garments that fully cover the wearers' body (and fail on e.g., short shorts and crop tops).

As noted in Section 6.2, we could not not use an optical tracking system for ground truth as it interfered with our computer vision pipeline. Instead, we had human annotators subjectively rate each keypoint as correctly placed or not (i.e., a binary rating), as opposed to a continuous spatial accuracy metric such as euclidean error. This experimental compromise permitted us to run a user study, but at the expense of reporting precision. For some insight into per-joint error, accuracy benchmarks on public datasets can be found in [11, 31].

We also acknowledge that our use of ArUco for person detection and relative 3D spatial localization is inelegant, requiring instrumentation of headsets or smartphones. Additionally, although we have a tracker and time-to-live associated with each detected ArUco marker, in cases of extreme occlusion and motion blur, this process can fail in registering users. In such events markerless based ID approaches could be used. For our mobile AR use case, face recognition could be used to dispense with the need for an ArUco tag. In our two headset form factors, the face is mostly occluded, and so matching would have to occur based on body biometrics, apparel and other person attributes as seen in deep learning based person re-identification [24, 48].

Finally, our current implementation relies on the cloud to collect and disseminate pose data, and also run advanced features such as 3D pose estimation using views from several users. This requires internet connectivity and contributes almost half of our system's total latency. In the future, it could be that proximate devices create ad hoc wireless networks to share pose information more directly with others.

9 CONCLUSION

We have described our work on BodySLAM, which shows that it is possible to use the existing cameras in AR/VR experiences to opportunistically capture the bodies, hands, mouths and appearance of participants in multiuser settings. This offers functionality that would otherwise have to be achieved with additional special-purpose sensors, either worn or installed in the environment. We combine user studies with software simulations to evaluate how well our system scales across rooms and group sizes. While there are innate limitations of our ad hoc approach, its software-only nature makes it unique in the literature.

ACKNOWLEDGMENTS

We especially thank NVIDIA for hardware donations and Andey Ng for her help with the user studies and participant recruitment. We also thank our reviewers for their valuable feedback and comments.

REFERENCES

[1] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 453–462.

[2] Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–10.

[3] Dimitrios S Alexiadis, Philip Kelly, Petros Daras, Noel E O'Connor, Tamy Boubekeur, and Maher Ben Moussa. 2011. Evaluating a dancer's performance using kinect-based skeleton tracking. In *Proceedings of the 19th ACM international conference on Multimedia*. 659–662.

[4] Brian Amento, Will Hill, and Loren Terveen. 2002. The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. 724–725.

[5] Patrick Baudisch, Henning Pohl, Stefanie Reinicke, Emilia Wittmers, Patrick Lühne, Marius Knaust, Sven Köhler, Patrick Schmidt, and Christian Holz. 2013. Imaginary reality gaming: ball games without a ball. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 405–410.

[6] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. 2020. BlazePose: On-device Real-time Body Pose tracking. *arXiv preprint arXiv:2006.10204* (2020).

[7] Hrvoje Benko, Edward W Ishak, and Steven Feiner. 2004. Collaborative mixed reality visualization of an archaeological excavation. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 132–140.

[8] Mark Billinghurst, Ivan Poupyrev, Hirokazu Kato, and Richard May. 2000. Mixing realities in shared space: An augmented reality interface for collaborative computing. In *2000 IEEE international conference on multimedia and expo. ICME2000. Proceedings. Latest advances in the fast changing world of multimedia (Cat. No. 00TH8532)*, Vol. 3. IEEE, 1641–1644.

[9] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* 32, 6 (2016), 1309–1332.

[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.

[12] Google Cardboard. 2014. Google Cardboard. Retrieved 2014 from https://vr.google.com/cardboard

[13] MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. 2001. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation* 17, 3 (2001), 229–241.

[14] Sehoon Ha, Yunfei Bai, and C Karen Liu. 2011. Human motion reconstruction from force sensors. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 129–138.

[15] Richard I Hartley and Peter Sturm. 1997. Triangulation. *Computer vision and image understanding* 68, 2 (1997), 146–157.

[16] HoloKit. 2019. HoloKit. Retrieved 2019 from https://holokit.io

[17] Ke Huo, Tianyi Wang, Luis Paredes, Ana M Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018. Synchronizar: Instant synchronization for spontaneous and spatial collaborations in augmented reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 19–30.

[18] JSArUco. 2018. JSArUco. Retrieved 2018 from https://github.com/jcmellado/js-aruco

[19] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. 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*. 167–176.

[20] Jan J Koenderink and Andrea J Van Doorn. 1991. Affine structure from motion. *JOSA A* 8, 2 (1991), 377–385.

[21] Ilya Kostrikov and Juergen Gall. 2014. Depth Sweep Regression Forests for Estimating 3D Human Pose from Images.. In *BMVC*, Vol. 1. 5.

[22] Cheng Li and Kris M Kitani. 2013. Pixel-level hand detection in ego-centric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3570–3577.

[23] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. 2015. Facial performance sensing head-mounted display. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–9.

[24] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. 2019. Improving person re-identification by attribute and identity learning. *Pattern Recognition* 95 (2019), 151–161.

[25] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

[26] METAmotion. 2018. METAmotion. Retrieved 2018 from http://metamotion.com/gypsy/gypsy-motion-capture-system.htm

[27] Rafael Munoz-Salinas. 2012. Aruco: a minimal library for augmented reality applications based on opencv. *Universidad de Córdoba* (2012).

[28] David Nistér, Oleg Naroditsky, and James Bergen. 2004. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 1. Ieee, I–I.

[29] TechCrunch Oculus. 2020. TechCrunch - Oculus. Retrieved 2020 from https://techcrunch.com/2016/10/06/facebook-social-vr/

[30] OptiTrack. 2020. OptiTrack. Retrieved 2020 from http://optitrack.com

[31] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 269–286.

[32] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–11.

[33] Rift. 2006. Oculus Rift. Retrieved 2006 from https://www.oculus.com/rift/

[34] Damien Constantine Rompapas, Christian Sandor, Alexander Plopski, Daniel Saakes, Joongi Shin, Takafumi Taketomi, and Hirokazu Kato. 2019. Towards large scale high fidelity collaborative augmented reality. *Computers & Graphics* 84 (2019), 24–41.

[35] Silonie Sachdeva et al. 2009. Fitzpatrick skin typing: Applications in dermatology. *Indian Journal of Dermatology, Venereology, and Leprology* 75, 1 (2009), 93.

[36] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L Miguel Encarnaçao, Michael Gervautz, and Werner Purgathofer. 2002. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments* 11, 1 (2002), 33–54.

[37] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K Hodgins. 2011. Motion capture from body-mounted cameras. In *ACM SIGGRAPH 2011 papers*. 1–10.

[38] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 2013. 3D with Kinect. In *Consumer depth cameras for computer vision*. Springer, 3–25.

[39] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Facevr: Real-time facial reenactment and eye gaze control in virtual reality. *arXiv preprint arXiv:1610.03151* (2016).

[40] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. 2019. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision*. 7728–7738.

[41] Vicon. 2020. Vicon. Retrieved 2020 from https://vicon.com/

[42] Vive. 2006. HTC VIVE. Retrieved 2006 from https://www.vive.com/

[43] Karl DD Willis, Ivan Poupyrev, Scott E Hudson, and Moshe Mahler. 2011. SideBySide: ad-hoc multi-user interaction with handheld projectors. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 431–440.

[44] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. 2019. Mo 2 cap 2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2093–2101.

[45] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 167–173.

[46] Yang Zhang, Chouchang Yang, Scott E Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++ room-scale interactive and context-aware sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15.

[47] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7356–7365.

[48] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. 2019. Pose-invariant embedding for deep person re-identification. *IEEE Transactions on Image Processing* 28, 9 (2019), 4500–4509.

[49] Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. 2016. AuraSense: enabling expressive around-smartwatch interactions with electric field sensing. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 81–86.