# VideoPoseVR: Authoring Virtual Reality Character Animations with Online Videos

CHENG YAO WANG, Cornell University, USA
QIAN ZHOU, Autodesk Research, Canada
GEORGE FITZMAURICE, Autodesk Research, Canada
FRASER ANDERSON, Autodesk Research, Canada

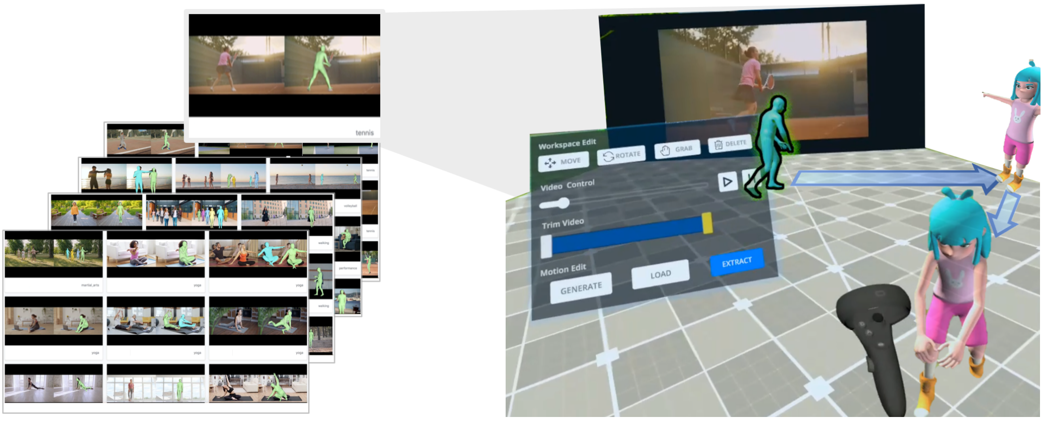


Fig. 1. *VideoPoseVR* reconstructs 3D pose and motion from 2D videos to enable rapid prototyping of character animation in VR. Users without animation experience can import and convert videos into a 3D motion dataset. They can visualize and manipulate the life-size motion to animate 3D characters in VR.

We present *VideoPoseVR* , a video-based animation authoring workflow using online videos to author character animations in VR. It leverages the state-of-the-art deep learning approach to reconstruct 3D motions from online videos, caption the motions, and store them in a motion dataset. Creators can import the videos, search in the dataset, modify the motion timeline, and combine multiple motions from videos to author character animations in VR. We implemented a proof-of-concept prototype and conducted a user study to evaluate the feasibility of the video-based authoring approach as well as gather initial feedback of the prototype. The study results suggest that *VideoPoseVR* was easy to learn for novice users to author animations and enable rapid exploration of prototyping for applications such as entertainment, skills training, and crowd simulations.

CCS Concepts: • **Human-centered computing** → **Virtual reality**; • **Information systems** → **Multimedia content creation**; • **Computing methodologies** → **Motion capture**.

Additional Key Words and Phrases: virtual reality, 3D motion, computer animation, content creation

Authors' addresses: Cheng Yao Wang, Cornell University, New York, USA, cw776@cornell.edu; Qian Zhou, Autodesk Research, Toronto, Canada, qian.zhou@autodesk.com; George Fitzmaurice, Autodesk Research, Toronto, Canada, george.fitzmaurice@autodesk.com; Fraser Anderson, Autodesk Research, Toronto, Canada, fraser.anderson@autodesk.com.

2573-0142/2022/12-ART575 $15.00
https://doi.org/10.1145/3567728

**ACM Reference Format:**
Cheng Yao Wang, Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. VideoPoseVR: Authoring Virtual Reality Character Animations with Online Videos. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 575 (December 2022), 20 pages. https://doi.org/10.1145/3567728

## 1 INTRODUCTION

Recently a growing number of artists, filmmakers and animators have explored using Virtual Reality (VR) for storytelling and filmmaking. Creating 3D animated stories requires professional desktop software such as Maya and Blender. Creators need to manually pose the joints of the 3D character to create the keyframes, which takes a significant part of the time. The keyframe-based animation creation requires extensive learning and exercise to create convincing results, which can be complicated and difficult for novice users. Creating animated VR scenes can be particularly challenging as creators have to alternate between the desktop software and VR environment to validate their creation in VR, adding additional difficulty to their workflow [2, 4].

To address this challenge, several work have explored creating 3D animations directly in Mixed Reality with controllers [18, 56, 73], hand gestures [38], and object motions [53]. Commercial products such as Oculus Quill [18] and AnimVR [56] allow users to create animated cartoons using 3D strokes. Other tools such as AnimationVR [73] and XRDirector [54] use VR devices as puppets to directly manipulate the 3D character. Compare to traditional animation software, these tools make it easier and faster to create animations in particular for novice users as an inexpensive alternative to professional motion capture systems. However, they still require manual input from users to perform the motion or manipulate the joints. The animations that can be created are often limited to what can be performed by the user. Additionally, creators can usually author one animation at a time, which makes it time-consuming to create multiple animations for a group of characters performing different motions in the scene.

While users spend a lot of effort creating animation assets, motions especially human activities are widely available in today's online videos. People record and share their physical movements in various activities such as dancing and fitness. In this work, we introduce *VideoPoseVR* , a novel animation authoring approach that uses online videos to author VR character animations. It leverages the state-of-the-art deep learning approach to reconstruct 3D motions from online videos, caption the motions, and store them in a motion dataset. Creators can import the videos, search in the dataset, modify the motion timeline, and combine the motions from different videos to create new animations in VR. By leveraging the large number of human motions that are present in online videos, creators can quickly animate multiple characters with diverse motions.

To explore the feasibility of utilizing video-based motion to animate characters, we implemented a proof-of-concept system that converted 161 online videos into a 3D motion dataset. We investigated how well users could use the dataset to quickly animate characters in VR by conducting a user study with ten participants. We found that novice users were able to learn and use *VideoPoseVR* to create character animations in VR, while advanced animators might use the reconstructed motion as a starting point to refine the animation and expedite the workflow. Our contributions include: 1) a novel video-based character animation authoring workflow in VR, 2) a proof-of-concept system with a detailed set of methods using the state-of-the-art deep learning approach to reconstruct, classify, and convert the 3D motion from online videos into keyframe-based animations, and 3) the evaluation of the proposed approach that demonstrated the feasibility and provided design implications.

## 2 RELATED WORK

Our work was inspired by research from the domains of 3D human pose estimation using 2D videos, the utilization of human poses in videos, authoring tools that can be used to prototype VR scenes and 3D character animation.

### 2.1 Animation Authoring in AR/VR

Traditionally, creating 3D characters animations requires using professional animation software such as Maya, 3ds Max and Blender. Animators need to manually pose the joints of the 3D character to create the keyframes that are interpolated by the software, which takes a significant part of the time spent in animating characters. The keyframe-based approach also requires extensive learning and exercise to create convincing results of character movement, which is challenging for novice animators. Another widespread performance-based approach uses high-end motion capture (Mocap) systems (e.g. Optitrack) or off-the-shelf equipment such as RBG cameras [35, 79] or depth cameras [13, 24] to capture the performance of skilled actors. These systems may still require extensive instrumentation, skilled actors, and laborious post-processing steps, which makes them inaccessible for many researchers, artists, and content creators.

As VR headsets and smartphone-based AR become more accessible, various work have explored using AR and VR to reduce the complexity of posing the characters in the keyframe-based approach. AR/VR applications and tools such as Tvori [69], VR Blender [7], PoseMMR [57], AnimationVR [73], and AR-Pose [72] allow users to directly manipulate the joints in 3D using VR controllers or smartphones instead of using 2D translation and rotation widgets. Other approaches extend the existing performance-based tools and puppetry interfaces [23, 29] by leveraging the tracking capability of AR and VR devices. Smartphone-based AR allows users to directly turn the phone into a puppet to control the character [55, 81] by tracking the phone. Systems such as Mindshow [51], SpatialProto [53], and XRDirector [54] capture the physical motion of VR devices [51] and gestures [38] and directly apply the motion to characters. Other related applications such as Oculus Quill [18] and AnimVR [56] allow users to create animated cartoons using 3D strokes.

While these tools are intuitive to use, they still require manual input to generate the 3D poses by performing the motion or manipulating the joints. The animations that can be created are often limited to what can be performed by the user. Additionally, users can usually author one animation at a time. The process of creation can be tedious to create multiple animation. We present a solution to generate 3D motion from online videos. While videos have been used to prototype AR experience by capturing the objects and their surrounding environment [42, 43], we focus on the human motions captured in the videos. Compared to existing Mocap dataset [46], the large number of online videos provides a diverse set of human motions that can be potentially used as motion assets for animators to create character animations. Similar to prior work that use a gallery of existing assets (such as motions [16] and pictures [37, 41]) as starting examples, we consider the video-based motion assets as potential starting point for creators to work on. Our work can be combined with existing VR-based animation authoring tools such as MindShow [51] and PoseMMR [57] to further refine the motion in VR and facilitate the workflow of animation creation.

### 2.2 Immersive Authoring Tools for Novices

Prior work has found that the learning affordances of VR helped bridging experience-related gaps between novices and experts for 3D design [77]. Novice and expert users differ in expertise that involves domain knowledge [22, 77] and spatial abilities [3]. The multi-modal feedback in VR promotes meaningful learning for novice to pick up the domain knowledge, while the spatial visualization in VR enables novices to design directly in 3D instead of needing to imagine and mentally manipulate 2D representations in traditional desktop software. Several immersive authoring tools have enabled users to create 3D models or scenes in VR. For example, CaveCAD [30] enabled users to position and transform 3D objects directly in a scene. MagicalHands [1] used direct gestural manipulations for the authoring and modification of animation effects for a 3D object. One Man Movie [21] provided tools for scene layouts, 3D character poses and animations, as well as complex camera rigs. Commercial applications such as Microsoft Maquette [50], Oculus Medium [17] and Google Tilt Brush [26] delivered compelling immersive 3D sketching and sculpting experiences to consumers. The advances of these tools allow casual users to create stunning VR content, rather than exclusively for professionals. Similar to previous work, our work provides an immersive user interface in VR to enable intuitive in-situ character animation creation for casual users.

## 2.3 Multi-person 3D Pose Estimation from Videos

Motion capture from monocular video has many advantages over traditional motion capture because it does not require a complex setup, it is low cost, and it offers a non-intrusive capture process. With the availability of video datasets associated with ground truth motion capture data (e.g., Human3.6M [31] and HumanEva [64]), significant progress has been made in recent years by using data-driven learning-based approaches. These approaches can be classified into two categories: the multi-stage and the one-stage. Most of the multi-stage methods are generally based on a top-down approach, which first detect people using bounding boxes and then apply a single person 3D estimator to each person. These 3D pose estimators [15, 59, 61] lift the 2D keypoints [8, 10, 19] into 3D joints using regression [12, 47] or model fitting [6]. Some work jointly recovers human shape and pose [34, 36, 80]. However, their predictions often fail to deal with truncation, scene occlusion, and person-person occlusion due to overlapping bounding boxes. Besides, they tend to output poor results on in-the-wild datasets like 3DPW [74] or MPI-INF-3DHP [15].

To avoid noisy bounding box prediction, one-stage solutions estimate body joint positions of all people and then group them into individuals. Zanfir et al. [83] proposed a bottom-up approach that simultaneously estimates 2D and 3D poses, and group joints based on 2D pose prediction scores. Mehta et al. [48] proposed occlusion-robust pose-maps and exploited the body part association to resolve the joint grouping problem. Recently, Sun et al. proposed ROMP [66], an one-stage regression network for monocular multi-person 3D mesh regression. They proposed the use of explicit body-center-guided representations to facilitate pixel-level human mesh regression in an end-to-end manner and developed a collision-aware representation to deal with the severe overlapping cases. ROMP achieved state-of-the-art performance on multiple benchmarks including real-time inference speed. In this work, we utilized ROMP to reconstruct 3D multi-person human motions from online videos, enabling rapid prototyping of character animations with various 3D human motion extracted from videos.

## 2.4 Applications Utilizing 3D Human Pose

Human pose estimation from videos have been widely adopted and utilized in several applications such as human behavior analysis, the generation of animations, and for movement learning. In terms of human behavior analysis, prior work has utilized human poses for human action recognition in videos [45, 84]. In addition, Lee et al. developed Skeletonographer, a tool that supports anonymous digital ethnography studies using skeletonized representations of people [39, 40]. ReliveReality [75] utilized various computer vision techniques to reconstruct an experience in 3D by reconstructed avatars, 3D environments, and 3D human poses. The reconstructed 3D experience offered a great opportunity to relive and study human behaviour. To facilitate movement learning, Tharatipyakul et al. [68] developed a web-based application that overlayed the detected 2D skeletons of a user and a teacher in a video on screen, so that the user could realize how his or her pose was different from the teacher. SuppleView [28] enabled coordinate translation-free viewing between an observer and an actor by inferring the 3D poses from trainer videos and creating a virtual 3D character as an actor of the predicted 3D movements. Takahashi et al. [67] also introduced a VR-based batter training system that estimated the 3D positions of a user 's body parts during a swing to visualize and provide real time feedback. Prior research has also leveraged human poses in video to generate animations. For instance, PoseTween [44] leveraged the motion of the subject of a video to create pose-driven tween animations of virtual objects. Park et al. [58] and Willett et al. [76] used reference motions in human action videos to guide the deformation of 2d virtual characters.

Human poses have also been used to control video playback. PoseAsQuery [27] was an interactive browsing system that repeatedly replayed a specific segment of a video by using a person's body movement. Reactive Video [14] used OpenPose [9] to extract user's and instructor's poses and adapt the playback of the video to mimic the user's movements. *VideoPoseVR* is the first prototype that allows users to quickly prototype character animations in VR by leveraging a variety of reconstructed 3D human motions from online videos.

## 3 DESIGN GOALS

After analyzing commercial offerings for animation and identifying the existing gaps, as well as the reviewing technical advances in machine learning and video processing, we developed a set of design goals to guide the development of the system.

**D1. Accessible to Novices:** Content creators without animation experience should be able to intuitively use animation sequences in their scenes. No advanced knowledge of keyframing, rigging, or advanced animation concepts should be necessary for immediate use of the system.

**D2. Enable Medium-fidelity Rapid Exploration:** Manually posing the joints to create keyframes in the professional animation software can be labor-intensive and time-consuming. We believe there are opportunities to develop a system which can automate this process and output medium-fidelity motion to support rapid exploration of character animations for both immediate use and further refinement.

**D3. Support for Multiple Motions per Scene:** Most animation authoring tools are able to create a multi-character scene by editing the animation of each character and manually synchronizing them in a global timeline. Meanwhile, there are data sources such as videos that contain multiple movements in a single scene. An author system should take advantage of these data sources, and enable users to leverage complex scenes with multiple individuals moving.

**D4. Robust to new Videos and Data:** Videos are becoming increasingly accessible and easy to generate and publish online. A system should support capturing and processing these new videos, and adding them into motion datasets for later use.

**D5. Modularized Pipeline** As machine learning and computer vision algorithms are rapidly developing, an animation system should be modular to allow new approaches and developments to be easily integrated into the system.

## 4 VIDEOPOSEVR

Based on the design goals, we designed *VideoPoseVR* , a video-based animation authoring workflow that allows novice users to import videos, extract motions from videos, and intuitively visualize and manipulate the life-size motion to animate characters in VR. *VideoPoseVR* has two parts: a motion reconstruction pipeline based on the state-of-the-art deep learning approaches that outputs a 3D motion dataset, and a VR user interface that allows novice users to intuitively visualize and manipulate the life-size motion to animate characters in VR.

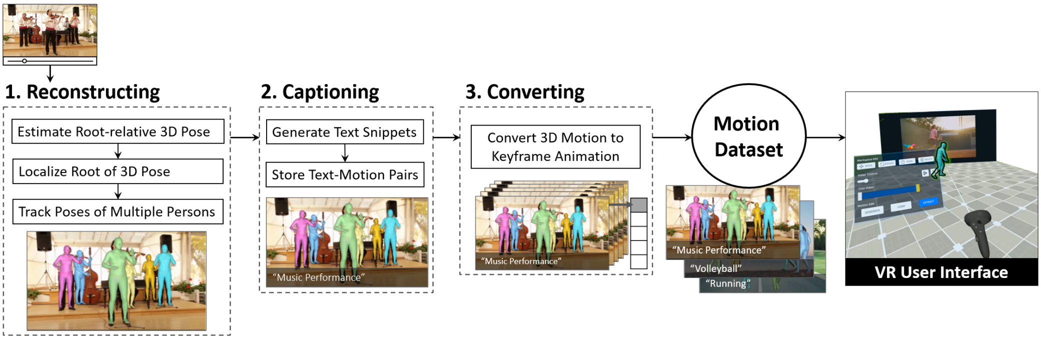


Fig. 2. *VideoPoseVR* system workflow, including an offline pipeline that generates a 3D motion dataset and a VR user interface that allows users to use the reconstructed motion to animate the characters in VR. We create the motion dataset by 1) reconstructing the 3D motion from videos, 2) captioning the motion, and 3) converting the motion to keyframe-based animation ready to use in VR.

## 4.1 Generating 3D Motion Dataset

To generate 3D motion from 2D videos, the system employs a pipeline with three steps: reconstructing 3D pose from 2D videos, captioning the reconstructed motion, and converting the captioned motion to keyframe-based animation (Figure 2).

*4.1.1 Reconstructing 3D motion from videos.* We first estimate the root-relative 3D poses for multiple persons in the video using ROMP [66], a one-stage network that reconstructs root-relative 3D poses for multiple people. Unlike most other approaches that uses a series of stages to handle the multi-person scenes, ROMP regresses all meshes for multiple persons in one single stage, making it robust to truncation, person-person occlusion, and environmental occlusion in the multi-person cases, which is critical for animating characters.

The predicted 3D poses from ROMP are only root-relative (i.e. relative to the pelvis joint). In some cases, it is important to know the absolute 3D poses (i.e. relative to the camera). For example, to animate a group of characters dancing in the scene, it is critical to assign the 3D position of each dancer in the video to the individual character so that when the dancers change their positions the characters can move along the exact same path in the scene. We use RootNet [52] to estimate the camera-centered coordinates of the human pose root by approximating the absolute depth from the camera to the human using the ratio of the human height in the physical space and the height in the image. We apply this approach to recover the root positions relative to the camera for all videos that are recorded with a stationary camera.

To distinguish the motion from multiple persons across frames, we track each person's 3D pose with a unique ID by associating the 3D pose with a tracked 2D pose. We detect 2D poses using AlphaPose [20] and track the keypoints of 2D pose using PoseFlow [78]. The detected 2D poses are tracked across frames with a unique ID for each person in the video. We filter the 2D keypoints and 3D poses with the 1 Euro filter [11] for temporal smoothing. For each 3D pose, we project the 3D joints onto the image and compute the error between the projected 3D joints and detected 2D joints in each frame. We associate each 3D pose with its corresponding 2D pose by finding the nearest joint (Figure 3) and use the tracked 2D pose to track 3D pose across frames for each person.

*4.1.2 Captioning 3D motion.* The reconstructed 3D motion from the video is unclassified without labels. Users have to manually identify and label the motion to use it as animation. This can be tedious when importing a large number of videos. Furthermore, there might be different types of

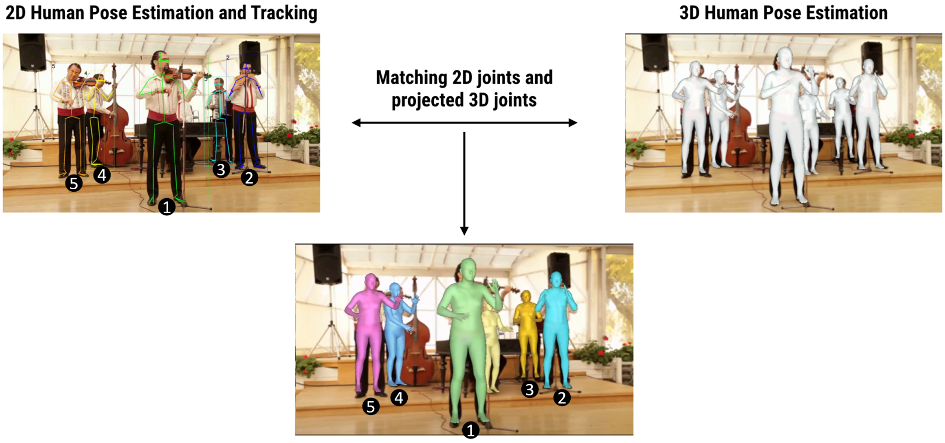

2D Human Pose Estimation and Tracking
Matching 2D joints and projected 3D joints
3D Human Pose Estimation
1
2
3
4
5

Fig. 3. Tracking the 3D poses of multiple persons in the scene. We associate tracked 2D pose (Left) with 3D pose (Right) to distinguish the motion from multiple persons across frames (Mid).

motion presented in one video at different time stamps. Therefore, we caption the 3D motion to allow users to search for a motion from the imported videos without manually annotating them.

We use the Contrastive Language-Image Pre-training (CLIP) [60] model to generate text snippets of the videos. The CLIP model connects text and images. It provides simple textual image descriptions for flexible zero-shot classification on arbitrary image datasets. It was trained on a multimodal dataset of 400 million image-text pairs to associate images with natural language descriptions. To generate captions using CLIP, we first sample one frame of a video every second and use the pre-trained CLIP vision transformer (ViT-B/32) to generate a text snippet for each frame. The text is represented as a feature vector and stored along with its corresponding video and reconstructed motion so that the text is paired with the motion. Generating text per second for each video can result in a large number of captions. To allow users to search efficiently at runtime, we use FAISS [32] to expedite the similarity search by building an index object that encapsulates all the feature vectors of captions. FAISS uses indexing methods built upon product quantization [33] and supports efficient comparisons between the query and stored vectors.

When users search for a motion at runtime, we first use CLIP's text encoder to compute the query vector. Then we use FAISS to get the k nearest neighbors by comparing the query vector and stored vectors. For each video, we compute a mean score of the cosine similarities between the query vector and stored vectors for all the sampled frames in the video (Figure 4). The k most relevant videos are returned by sorting the similarity scores of all videos and choosing the k highest ranked videos.

*4.1.3 Converting raw motion data to keyframe animation.* Similar to the motion captured data, the reconstructed frame-by-frame 3D motion data is difficult to modify. In order to allow users to apply the motion data to characters as animations, we convert the raw motion data into editable keyframe-based animations by sampling the original motion data at 30 fps. To ensure the smoothness of the sampled motion, we add a new keyframe if the difference in the joint rotation and position between the current frame and the last frame is above a fixed heuristic threshold of 0.1% position error and 0.15% rotation error.

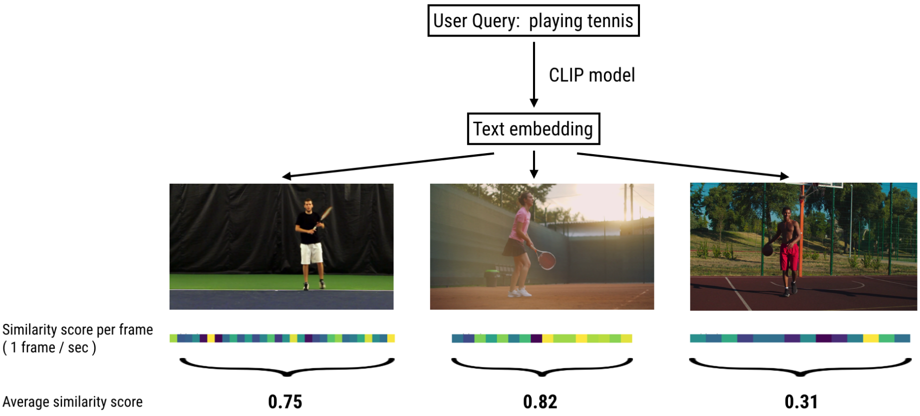


Fig. 4. An example of searching for a motion of playing tennis. We use the text encoder of the Contrastive Language-Image Pre-training (CLIP) model [60] to compute the query vector. For each video, we compute a mean score of the cosine similarities between the query vector and stored vectors for all the frames sampled per second in the video. We get the k most relevant videos by comparing the average similarity scores of all videos.

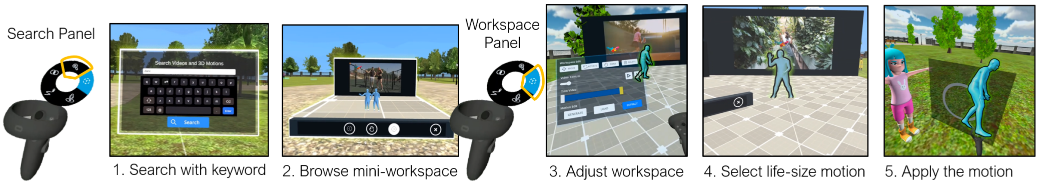


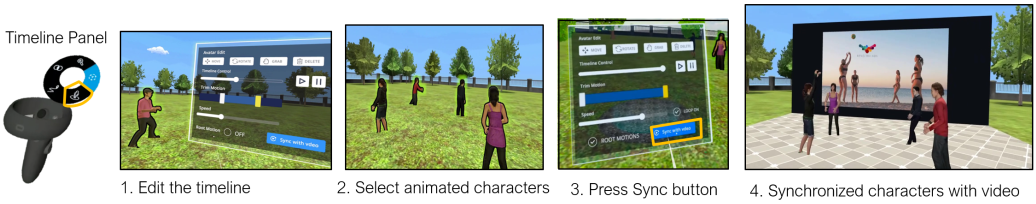
Fig. 5. The search and workspace panel: users can search for the motion with keywords and apply the motion to characters.

## 4.2 VR User Interface

To enable users to quickly prototype VR character animations using the reconstructed motion dataset, we develop a VR user interface that allows users to search, preview, apply, and customize the motion extracted from videos.

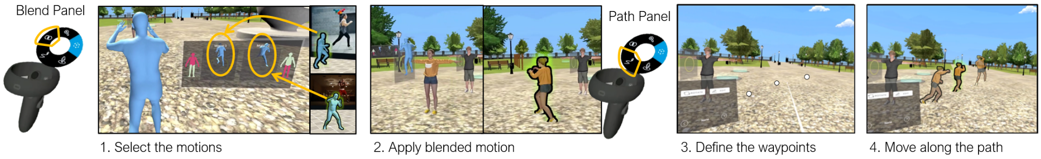
*4.2.1 Search for the motion.* Users can type in the keywords or descriptions of the motion with a virtual keyboard (Figure 5 (1)). Once finding relevant motion, the system renders a miniature workspace that displays both the video and the reconstructed 3D motion for each subject in the video (Figure 5 (2)). The reconstructed motions are synchronized with the video playback and the global movements are scaled to fit the mini-workspace. Users can quickly preview and navigate through different video-motion by clicking the next or previous buttons in the mini-workspace.

*4.2.2 Extract and apply the motion.* After finding a desired motion, users can select and place it in the current environment to visualize a life-size motion from different perspectives. The placement of the life-size workspace can be adjusted to avoid occlusion with the environment (Figure 5 (3)). To apply the motion to a character, users can select the motion from the workspace and place it onto the character (Figure 5 (4-5)). The system will automatically re-target the motion to the skeleton rig of the selected character.



1. Edit the timeline 2. Select animated characters 3. Press Sync button 4. Synchronized characters with video

Fig. 6. The timeline panel: users can edit the timeline of an applied motion and synchronize multiple motions with the video timeline.



1. Select the motions 2. Apply blended motion 3. Define the waypoints 4. Move along the path

Fig. 7. The blend and path panel: users can combine motions from videos and define a motion path of the animated character.

*4.2.3 Customize the motion.* *VideoPoseVR* supports several functions to help users further customize their character animations by trimming the animation and adjusting its playspeed. They can also apply the reconstructed root motion (i.e. the absolute 3D position in the video) to the character if applicable. They can manipulate the character by translating and rotating it. Users can access these functionalities via a virtual window after selecting the character and the button on a ring menu (Figure 6 (1)). The customized characters can be synchronized with the original 2D video to provide the context and details of the activity. The synchronization can be useful when users need to further refine the reconstructed motion by examining the difference and add details such as the facial expression of the persons in the video or the objects involved in the motion. To synchronize the characters with the video, users can select the characters that are animated with the motion from the same video, and then click a button on the virtual window (Figure 6 (3)).

*4.2.4 Combining the motions from different videos.* Users can combine the motions from different videos with a customizable body mask that specifies which parts of the body motion they want to apply. For example, users can create a combat *running* animation by combining a *boxing animation*, performed on the upper body of the character, with a running animation on its lower body (Figure 7 (1)). Users can first search for a *running* motion and define a lower body mask. Then they can search again for a *boxing* motion and define an upper body mask and animate a character with the generated combat running animation.

*4.2.5 Defining the motion path.* For the videos recorded with stationary cameras, *VideoPoseVR* can infer how the persons in the video move within the recorded 3D environment and reconstruct the root motion (i.e. the absolute 3D position in the video). With the root motion applied, the animated character can move in the virtual environment along the same path as in the original 2D video instead of playing at a fixed position. However, for the videos that are not recorded with stationary cameras, we do not have the root motion. In this case, *VideoPoseVR* supports users to create their own motion path by dropping points in the scene to fit a spline curve as the motion path. For example, users can specify the waypoints on the ground to define the motion path for a running

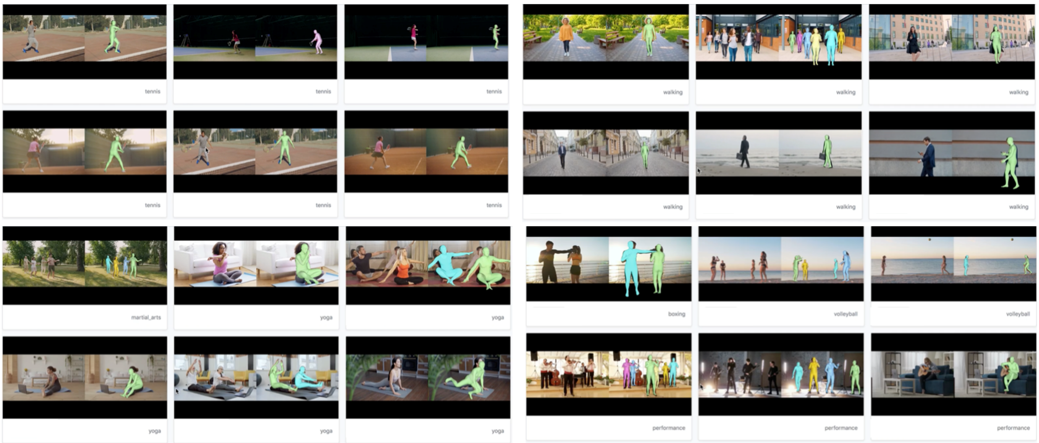


Fig. 9. Examples of the motion dataset with 3D motion reconstructed from 161 online videos consisting of a variety of human activities including tennis, walking, yoga, volleyball, performance etc.

animation (Figure 7 (3-4)). After applying the motion path to the animated character, users can adjust the speed of the movement and rotate the character to refine the animation.

## 4.3 Implementation

Our proof-of-concept prototype consists of three parts: a client-side VR application, a server-side web application, and a motion dataset (Figure 8).

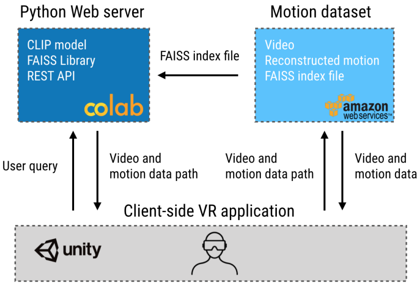


Fig. 8. Implementation of the VideoPoseVR prototype, including a Python web server, a client VR application, and a motion dataset.

*4.3.1 Python Web Server.* We developed a Python web server with Flask on Google Colab. We used open source libraries of ROMP [66], RootNet [52], AlphaPose [20], and PoseFlow [78] to implement the pipeline of creating motion datasets from videos directly on Google Colab. Users can directly run the prototype on their browsers without the need of powerful computers and setting up complex environments. We also integrated the pre-training CLIP model [60] to generate captions for the motion and the FAISS [32] library to enable efficient searching by generating FAISS index files stored in the motion dataset. Client applications can connect to the Python web server through REST API to search and retrieve the video-motion data path. When users send a search query from a client application, the server first uses the CLIP model to compute the embedding of the user query, and then uses FAISS indices to compute the similarity scores with the CLIP embeddings of all videos. After determining the five nearest neighbors (i.e. videos) of a user query, the server returns the five video-motion data paths to the client application, which are used to retrieve the video and motion data from the motion dataset.

*4.3.2 Client VR Application.* We implemented the client VR application with Unity. The client-side VR application connected to the Python web server with REST API. We converted the motion data into Unity's AnimationClip format and used Unity's Mecanim animation system [70] to implement the features to modify the humanoid animations (timeline and body mask). The current implementation supports three different skeleton rigs including the default Mecanim humanoid in Unity, Mixamo, and Microsoft Rocketbox avatar [25]. Within the VR application, users can call out a ring menu (Figure 6) and select functionalities such as searching and editing the motion using the joystick on the left controller. They use laser pointers attached to the controllers for selection and teleportation. They can select content by pressing the index trigger and grab UI panels and models using a grip button on the controllers.

*4.3.3 Motion Dataset.* We created a default motion dataset (Figure 9) with 161 online videos collected from Storyblocks [65] consisting of a variety of human activities including tennis, basketball, walking, yoga, etc. We store the reconstructed motion along with the videos and FAISS index files on an Amazon S3 server. Users can upload their own online videos and run our prototype to expand the motion dataset without annotating the videos.

## 5 EVALUATION

Our interactive *VideoPoseVR* prototype provides a new way for content creators to author character animations in VR. The goal of the study was to evaluate the feasibility of this authoring approach as well as gather initial feedback of the prototype. We collected subjective measures and qualitative observations.

### 5.1 Participants and Procedure

Ten participants (9 males and 1 female) between 29 and 46 years old (averaged 35.2 years) were recruited from within the institution and participated in the study remotely. We asked participants to rate their expertise in VR and 3D animation from 1 (novice) to 5 (expert) respectively. Respondents had an average score of 3.6 for VR, and 1.8 for 3D animation. All participants provided informed consent and were compensated with a $30 USD-equivalent gift card for an 1-hour study. The study was guided and supervised remotely by an experimenter via a video conference call.

Participants used their own VR devices (Oculus Quest 1 or 2) to run a VR application developed in Unity. We used a virtual park scene as the default virtual environment (Figure 10-study scene) and added virtual characters from Mixamo and Microsoft RocketBox avatar [25]. These characters are all in static T-pose in the scene and ready to be animated.

Before the evaluation began, participants were guided to set up the experimental environment by removing potential obstacles within their personal space in a standing position. The evaluation began with an introduction of the prototype. Participants were shown the *VideoPoseVR* interface and basic interactions (pointing, rotating, and teleporting) with a video for 10 minutes. Then they were given four tasks (Figure 10) to complete, each requiring the use of at least one feature in the prototype with an increasing level of difficulty across tasks. Before they conducted each task, participants were shown a video walkthrough illustrating the use of features in the task. The entire process of four tasks with video walkthrough last up to 30 minutes.

After the study, participants completed a questionnaire followed by a semi-structured interview. The questionnaire asked participants to rate individual features and overall experience on a Likert scale from 1 to 7 with 12 questions (Q1-12) shown in Figure 11. In the interview, participants were asked what they liked and disliked about the system, how they perceived the ownership rights of the extracted motions from videos, and any other comments about the overall experience, which lasts approximately 15 minutes.

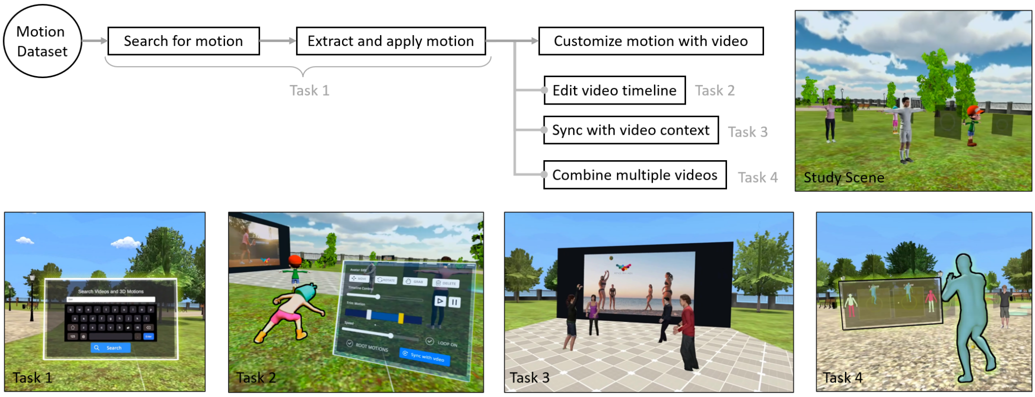


Fig. 10. Tasks used in the study for participants to author character animations in VR. They can search, extract, apply, synchronize, and combine 3D motions from online videos.

## 5.2 Study Tasks

**Search and apply motion.** Participants searched for a *tennis* motion, previewed videos in the mini-workspace, selected and placed a life-size motion in the scene. They created animation by grabbing the life-size motion and applying it to the character by selecting the ring icon beside the character.

**Edit motion timeline.** After applying the motion, participants trimmed the motion with the timeline sliders and adjusted the play speed in the timeline window.

**Synchronize animation with the video.** Participants searched for a video with a keyword of their choice that contains multiple motions and applied motions to characters. They selected and synchronized animated characters with the video background to provide context information of the motion.

**Combine motions from two videos.** Participants searched for a *running* motion and applied it to selected lower body mask in the blending window. They searched again for an upper body motion with a keyword of their choice, applied it to the upper body mask, and used the blended motion to animate a character in the scene.

## 5.3 Results

Overall, *VideoPoseVR* was rated positively (Figure 11) for both the individual features and user experience. Participants found it easy to use and learn (Q8-9). All participants completed the first three tasks and eight participants completed all four tasks within the 30 minutes of time limitation. During the interview, participants provided positive feedback and were impressed with the capability of using online videos to animate VR characters.

**1. Search for motion.** Participants found it easy to find the desired motion (Q1) from the video library: *"when I searched, I think there was a sufficient amount of variety in the videos that I wanted to use (P6)"* They found it useful to have a mini-workspace (Figure 10 (a)) to preview the motion with the video background (Q2). Two participants also explicitly mentioned it as: *"I like when I search and then I can see the video as well as the 3D character that is already performing the actions from the video (P2)".*

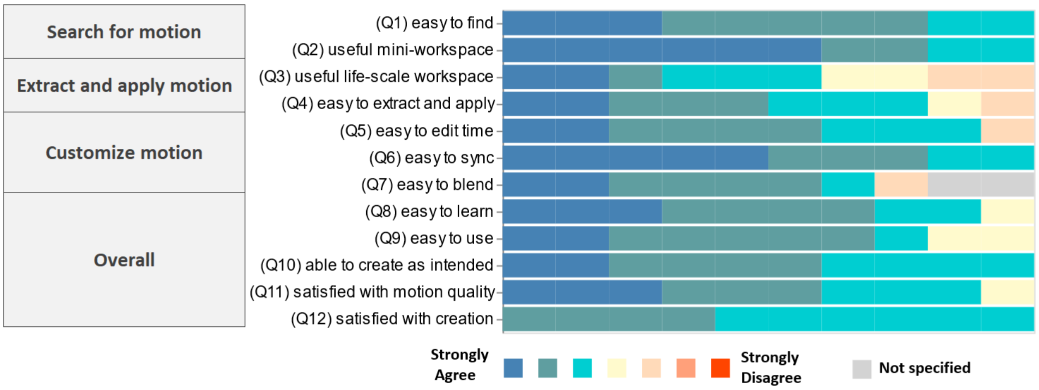


Fig. 11. Participants' rates on the usability of *VideoPoseVR* from -3 "Strongly Disagree" to 3 "Strongly Agree". They reported answers on the questions for features in the *VideoPoseVR* interface, including (Q1-2) searching for motion, (Q3-4) extracting and applying motion, (Q5-7) customizing motion, and (Q8-12) the overall experience. Overall, we found most features were rated positively by participants.

**2. Extract and apply motion.** Interestingly, participants did not tend to value the life-size workspace (Q3) as much as the mini workspace (Q2). Two participants further commented that the life-size workspace may take additional visual real estate and sometimes block their view, causing problems with navigation and applying the motion to VR characters: *"you placed the scene with the video and the avatar down, that sometimes I was trying to figure out, like, how can I move around it? could actually be interesting to avoid blocking the view (P1)." "And it was a little bit confusing because it's like, Oh, I have to make sure that if I placed them, I actually placed them properly (P2)."*

**3. Customize motion with video.** Participants were successful in customizing the motion extracted from the video of their choice, including editing the timeline (Q5) and synchronizing multiple animation timelines with the video (Q6). Two participants mentioned that they liked the blending feature: *"I liked the blending the most. I think actually it's not something that you can do easily today in a 3D digital content creation too (P9)"*. Two participants encountered technical issues in Task 4 (blending motions) in time (app crashed on their devices) and were not required to provide answers for Q7 (Figure 11).

## 6 DISCUSSION

In building *VideoPoseVR* , we strove to meet a set of goals which motivated our design and implementation choices. Overall, the system met these goals, and was able to create an accessible workflow for novice users to find, modify and apply animations. We revisited the design goals and discussed opportunities and concerns raised in the study.

### 6.1 Revisiting the Design Goals

**D1. Accessible to Novices.** One of our design goals is to lower the barrier for non-expert animators (e.g. researchers, content creators) to engage in prototyping VR content with character animations. Thus, instead of letting users pose the virtual character or perform the motion to create animation from scratch, *VideoPoseVR* leveraged reconstructed 3D motions from videos to allow both professionals and non-expert users to quickly prototyping character animation in VR. Participants found it easy to follow the exemplary workflow. Nine out of ten participants mentioned that the system was easy to use and learn. Three of them further commented that the system was friendly to everyone: *“This felt extremely lightweight, intuitive, and it felt like anybody could pick it up (P8)”*. *“This would empower content creation. Everyone can become a 3D creator (P9)”*. Three of them also mentioned this approach has the strength of leveraging the huge amount of videos online, turning them into animation assets available for content creators: *“there are so many videos of people either like explaining how to do certain movements or videos of amateur and professional sports, uh, that you can use to create motions that are more realistic (P1) ”*.

These encouraging findings support our motivation and design goal of enabling content creation in VR for everyone. In addition, four participants mentioned that practitioners could use this technique for quick prototyping without a complex setup to collect animation resources: *“so this is probably very good for creating like a quick prototype. There are like repeated same set of actions. And I don’t want to create, um, just have somebody record a video, and then just do it. But that shouldn’t take you more than 30 minutes or 40 minutes max. Like you have the video and then if your system is able to get those motions out of video, that is way more quicker (P5)”*.

In contrast to the ease of authoring video-based animations, we found that participants struggled with the controller’s button-mapping. We observed participants frequently asked which buttons to press as well as the order of pressing them. Three participants commented that memorizing the controller mapping could be difficult for novice users: *“it was a bit of a mental mapping exercise I had to do with which button do I use for what, and how do I select something and then release it again (P1)." “One of the things that I look at is like, how to assign buttons to certain things and triggers to certain things? And how long does it take for that to become comfortable, but then you may be jumping to another app, and the controls are totally different? (P10)"*. These comments could potentially explain the lower ratings for Q5 and Q7 (Figure 11), which require several buttons being pressed sequentially. While this problem could be possibly alleviated by providing an input mapping diagram overlaid on the controller in the scene, it is still a potential barrier for novices by having to frequently call out the mapping diagram that could interrupt the workflow. We expect such issues could be addressed in the future by bringing standardized desktop input such as keyboard and mouse with user familiarity into the virtual environment.

**D2. Enable Medium-fidelity Rapid Exploration.** In general, we found that participants were mostly satisfied with the motion quality (Q11 in Figure 11). Two participants mentioned the extracted motions could be further improved: *“one thing I found was, the actions may not be one hundred percent accurate. I noticed that in one of those scenes where in playing volleyball, the hands were like a bit separated from each other (P5)”*, and *“ sometimes the avatars’ feet were kind of below the ground. So I guess that if they can react to the ground that would make it more useful (P9)”*. Two participants also mentioned that the extracted motion can serve as a starting point for further refinement: : *“...take a video and then you can concentrate on fine tuning it (P5)”*, and *“I can start from somewhere midway and then edit it (P3)”*.

The goal of *VideoPoseVR* is to support animating characters in the early stage of VR prototyping. Instead of replacing the existing animation tools, users could use *VideoPoseVR* to explore initial animations of their VR experiences and refine important animations later on with their preferable animation tools. We see the potential of combining *VideoPoseVR* with existing VR-based animation authoring tools such as PoseMMR [57] to further refine the motion in VR and expedite the workflow of authoring VR animations.

**D3. Support for Multiple Motions per Scene.** With most of the current animation tools such as Blender [5], users create a multi-character scene (such as team sports) by editing the animation of each character and manually adjust their global timing to synchronize the motion, which makes it time-consuming to prototype a scene containing multi-person human activities. . To overcome this limitation, we adopt the state-of-the-art 3D pose estimation approach in *VideoPoseVR* to extract multi-person motions. The timelines of these motions are inherently synchronized and therefore users do not need to manually synchronize them. During the interview, two participants mentioned that they liked the synchronization feature: *"there was a video playing behind and then all of these four avatars were positioned the same way as it was in the video. They were trying to do the exact same set of movements as it was happening in the video. So that was the best part (P5)"*.

**D4. Robust to new Videos and Data.** Although we created a motion dataset with 112 online videos consisting of a variety of human motions, *VideoPoseVR* allows users to add their own videos and motions to the motion datasets so that they can search and retrieve their preferable motions when animating virtual characters. To achieve this goal, we leveraged open source computer vision techniques and implemented the pipeline of creating motion datasets from videos directly on Google Colab. Thus, users can directly run the prototype on their browsers without the need for powerful computers and setting up deep learning environments. Users can add new motion to the dataset by simply uploading their videos to Google Colab. However, since certain Unity Animation APIs can only be executed in the Unity Editor, we note that users still need to manually complete the step for converting raw motion data to keyframe animation in Unity and uploading them to Google Colab. Although it's possible to realize a full automatic pipeline by implementing custom keyframe animations, it's beyond the scope of the current paper.

**D5. Modularized Pipeline.** As 3D human pose estimation technology continues to evolve, it is important to make the *VideoPoseVR* prototype generalizable and extendable so that the modules in the prototype can be replaced as advances are made. We achieved this by dividing the pipeline of converting video to animation assets into independent modules. With this approach, as new developments are made in computer vision, a single module can be replaced within the pipeline without disrupting the functionality of *VideoPoseVR* .

## 6.2 Motion Ownership

During the study, seven out of ten participants raised concerns on the ownership of copyright on the extracted motions. Within them, four participants felt that video owners would require credits if motions were extracted from the videos: *"Maybe I had to pay them money because they're skilled professionals....the reason why you get training for 15, 20 years in your life when you're young is just to learn those, the physics of the movements of your body parts that makes motion so special. So that's the reason why I would definitely want to credit (P6)". "there's gotta be some sort of connection to the original piece (P10)". "you could actually have like some revenue. Anyone can do it and if they pay me 10 cents or something (P9) "*. Three participants mentioned it would depend on the content and require explicit consent: *"I guess it depends on the video... if it's like a video that's somebody took of me not necessarily for motion extraction, then I'm a little bit iffy about that because I didn't mean to have my video to be extracted (P2)". "it would be good to give explicit consent for that depending on how it's used (P1)"*.

These concerns voiced in the study point to possible abuse of the extracted motions without proper consent and monetary credit to the original video owners. As the techniques of motion reconstruction come closer to practical use, future work is required to discuss what can be extracted with credit and consent and how to ensure the extracted motions can be traced back to the video owners.

## 6.3 VR vs Desktop

While it is possible to use the video-to-animation authoring workflow in the desktop environment, we found the workflow to be more beneficial in VR for two reasons. First, creating animations for VR has been found tedious with excessive context switches between VR and desktop to review the edits [4, 7]. For collaborative virtual world applications with an emphasis on player creation (such as Recroom [63] and Horizon Worlds [49]), casual users without animation expertise need to animate their VR scenes. It would be difficult for them to learn commercial animation software and export their creation into VR. The intuitive workflow of *VideoPoseVR* can be beneficial for creators to author their VR experience with accessible online videos in a single environment. Second, the video-to-animation workflow has the opportunity to enable novel interactions by integrating with the 3D input of VR. In the searching stage, we use a keyword-based approach to demonstrate the workflow. It is possible to query by directly demonstrating the motion [62], which can be useful to specify a particular type of movement, such as a backhand ground-stroke in tennis, resulting in a coarse-to-fine querying approach that users can first use keywords to retrieve a motion type and refine the search results by demonstrating it. It is also possible to combine the demonstrated motion with the motion in the dataset to transfer a dancing style [82] or complement the reconstructed motion with deficiency. In the refinement stage, as two participants suggested that the extracted motion can serve as a starting point for further refinement, *VideoPoseVR* can be combined with existing VR animation tools [7, 73] that allow users to directly manipulate joints in 3D.

## 7 LIMITATIONS AND FUTURE WORK

While most of the motions reconstructed from ROMP [66] have reasonable quality in our dataset, we found that the extracted motions can sometimes become unreliable when people in the video are too small or when the person-person occlusions are severe. Thus, we expect future advancement in 3D motion reconstruction can improve the motion quality. Our prototype reconstructs motions individually without considering the similarity between them. To enable gallery-based animation authoring, we expect future development can merge similar motions (such as running) to complement motions with deficiency caused by the occlusion and refine the motion quality.

The CLIP's zero-shot classifier allows users to create motion datasets without annotation and further enables semantic search. However, it still has poor generalization to scenes not covered in its pre-training dataset and it's sensitive to wording or phrasing. While a formal study to evaluate the performance of our approach in some video benchmarks would be helpful, it's beyond the scope of the current paper. In the future, we plan to extract other metadata from the online video (e.g. titles and categories) to improve the search performance and also explore multimodal techniques to help users find the right motions such as searching by performing the motion.

We developed *VideoPoseVR* to demonstrate a video-to-animation authoring workflow in VR. The interface contains elements from both video and animation interface such as the video playback controls (Figure 6(1)) as well as the body mask (Figure 7(1)) inspired from the Avatar Mask in Unity [71]. For casual users like prosumers, these functionalities allow them to animate their VR scenes without domain knowledge in animation. For professional animators, the animations created by *VideoPoseVR* are still not suitable for final production compared to commercial animation software. In particular, we implemented the timeline editing operations but not the keyframe editing operations for adjusting the position, orientation and scaling of pose. Since there are several VR applications that support keyframe editing [7, 51, 56], we plan to integrate *VideoPoseVR* with existing VR animation tools for further refining motions in the later stage of prototyping such as manipulating the character pose or real-time puppeteering of the character. Future work is required to compare the VR-based authoring workflow with traditional animation software and understand the different needs of novice and expert users.

## 8 CONCLUSION

We presented *VideoPoseVR* , a video-based animation authoring approach using online videos to author character animations in VR. It utilized the state-of-the-art deep learning approach to reconstruct 3D motions from online videos, caption the motions, and store them in a motion dataset. Through *VideoPoseVR* , users can search, extract, apply, synchronize, and combine 3D humanoid motions from existing online videos to animate virtual characters in VR. We conducted a user study to evaluate the feasibility of the video-based authoring approach as well as gather initial feedback of the prototype. The results suggested that *VideoPoseVR* was easy to learn for novice users to author character animations in VR. Participants also suggested using *VideoPoseVR* for rapid exploration of animation ideas and raised concerns on the ownership of motion data extracted from the original videos.

## REFERENCES

[1] Rahul Arora, Rubaiat Habib Kazi, Danny M. Kaufman, Wilmot Li, and Karan Singh. 2019. MagicalHands: Mid-Air Hand Gestures for Animating in VR. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 463–477. https://doi.org/10.1145/3332165.3347942

[2] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. *Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376722

[3] Mayra Donaji Barrera Machuca, Wolfgang Stuerzlinger, and Paul Asente. 2019. The effect of spatial ability on immersive 3d drawing. In *Proceedings of the 2019 on Creativity and Cognition*. 173–186.

[4] Bruna Berford, Carlos Diaz-Padron, Terry Kaleas, Irem Oz, and Devon Penney. 2017. Building an animation pipeline for vr stories. In *ACM SIGGRAPH 2017 Talks*. 1–2.

[5] Blender. 2022. Blender NonLinear Animation. https://docs.blender.org/manual/en/latest/editors/nla/introduction.html. [Online accessed 11-September-2022].

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. 2016. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*. Springer, 561–578.

[7] Alberto Cannavò, Claudio Demartini, Lia Morra, and Fabrizio Lamberti. 2019. Immersive virtual reality-based interfaces for character animation. *IEEE Access* 7 (2019), 125463–125480.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence* 43, 1 (2019), 172–186.

[9] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[10] 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.

[11] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2527–2530.

[12] Ching-Hang Chen and Deva Ramanan. 2017. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7035–7043.

[13] Jiawen Chen, Shahram Izadi, and Andrew Fitzgibbon. 2012. KinÊtre: animating the world with the human body. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 435–444.

[14] Christopher Clarke, Doga Cavdir, Patrick Chiu, Laurent Denoue, and Don Kimber. 2020. Reactive Video: Adaptive Video Playback Based on User Motion for Supporting Physical Activity. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 196–208. https://doi.org/10.1145/3379337.3415591

[15] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. 2018. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 668–683.

[16] Ruta Desai, Fraser Anderson, Justin Matejka, Stelian Coros, James McCann, George Fitzmaurice, and Tovi Grossman. 2019. Geppetto: Enabling semantic design of expressive robot behaviors. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[17] Facebook. 2016. Oculus Medium. https://www.oculus.com/medium/. Accessed: 2021-09-08.

[18] Facebook. 2021. Facebook Quill. https://quill.fb.com/. Accessed: 2021-09-08.

[19] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2334–2343.

[20] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. RMPE: Regional Multi-person Pose Estimation. In *ICCV*.

[21] Quentin Galvane, I-Sheng Lin, Fernando Argelaguet, Tsai-Yen Li, and Marc Christie. 2019. VR as a Content Creation Tool for Movie Previsualisation. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 303–311. https://doi.org/10.1109/VR.2019.8798181

[22] Michael Robert Gardner and Warren W Sheaffer. 2017. Systems to support co-creative collaboration in mixed-reality environments. In *Virtual, augmented, and mixed realities in education*. Springer, 157–178.

[23] Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig animation with a tangible and modular input device. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[24] Mar González-Franco, Zelia Egan, Matt Peachey, A. Antley, Tanmay Randhavane, Payod Panda, Yaying Zhang, Cheng Yao Wang, Derek F. Reilly, Tabitha C. Peck, A. S. Won, A. Steed, and E. Ofek. 2020. MoveBox: Democratizing MoCap for the Microsoft Rocketbox Avatar Library. *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (2020), 91–98.

[25] Mar Gonzalez-Franco, Eyal Ofek, Ye Pan, Angus Antley, Anthony Steed, Bernhard Spanlang, Antonella Maselli, Domna Banakou, Nuria Pelechano, Sergio Orts-Escolano, Veronica Orvalho, Laura Trutoiu, Markus Wojcik, Maria V. Sanchez-Vives, Jeremy Bailenson, Mel Slater, and Jaron Lanier. 2020. The Rocketbox Library and the Utility of Freely Available Rigged Avatars. *Frontiers in Virtual Reality* 1 (2020), 20. https://doi.org/10.3389/frvir.2020.561558

[26] Google. 2016. Tilt Brush. http://www.tiltbrush.com/. Accessed: 2021-09-08.

[27] Natsuki Hamanishi and Jun Rekimoto. 2020. PoseAsQuery: Full-Body Interface for Repeated Observation of a Person in a Video with Ambiguous Pose Indexes and Performed Poses. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 13, 11 pages. https://doi.org/10.1145/3384657.3384658

[28] Natsuki Hamanishi and Jun Rekimoto. 2020. SuppleView: Rotation-Based Browsing Method by Changing Observation Angle of View for an Actor in Existing Videos. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Salerno, Italy) *(AVI '20)*. Association for Computing Machinery, New York, NY, USA, Article 95, 3 pages. https://doi.org/10.1145/3399715.3401952

[29] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D puppetry: a kinect-based interface for 3D animation.. In *UIST*, Vol. 12. Citeseer, 423–434.

[30] Cathleen E. Hughes, Lelin Zhang, Jürgen P. Schulze, Eve Edelstein, and Eduardo Macagno. 2013. CaveCAD: Architectural design in the CAVE. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 193–194. https://doi.org/10.1109/3DUI.2013.6550244

[31] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* 36, 7 (2013), 1325–1339.

[32] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

[33] Herve Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128. https://doi.org/10.1109/TPAMI.2010.57

[34] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5614–5623.

[35] Ning Kang, Junxuan Bai, Junjun Pan, and Hong Qin. 2019. Interactive Animation Generation of Virtual Characters Using Single RGB-D Camera. *Vis. Comput.* 35, 6–8 (June 2019), 849–860. https://doi.org/10.1007/s00371-019-01678-7

[36] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. 2020. VIBE: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5253–5263.

[37] Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 88–1.

[38] Mikko Kytö, Krupakar Dhinakaran, Aki Martikainen, and Perttu Hämäläinen. 2015. Improving 3D character posing with a gestural interface. *IEEE computer graphics and applications* 37, 1 (2015), 70–78.

[39] Bokyung Lee, Michael Lee, Pan Zhang, Alexander Tessier, and Azam Khan. 2019. Semantic Human Activity Annotation Tool Using Skeletonized Surveillance Videos. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers* (London, United Kingdom) *(UbiComp/ISWC '19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 312–315. https://doi.org/10.1145/3341162.3343807

[40] Bokyung Lee, Michael Lee, Pan Zhang, Alexander Tessier, Daniel Saakes, and Azam Khan. 2019. Skeletonographer: Skeleton-Based Digital Ethnography Tool. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing* (Austin, TX, USA) *(CSCW '19)*. Association for Computing Machinery, New York, NY, USA, 14–17. https://doi.org/10.1145/3311957.3359510

[41] Brian Lee, Savil Srivastava, Ranjitha Kumar, Ronen Brafman, and Scott R Klemmer. 2010. Designing with interactive example galleries. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2257–2266.

[42] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmose, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 626–637. https://doi.org/10.1145/3472749.3474774

[43] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376160

[44] Jingyuan Liu, Hongbo Fu, and Chiew-Lan Tai. 2020. PoseTween: Pose-Driven Tween Animation. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 791–804. https://doi.org/10.1145/3379337.3415822

[45] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2018. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[46] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5442–5451.

[47] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2640–2649.

[48] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 82–1.

[49] Meta. 2022. Horizon Worlds | Virtual Reality Worlds and Communities. https://www.oculus.com/horizon-worlds/. [Online accessed 11-September-2022].

[50] Microsoft. 2019. Microsoft Maquette. https://www.maquette.ms/. Accessed: 2021-09-08.

[51] Mindshow. 2020. Mindshow. https://mindshow.com/. Accessed: 2021-09-08.

[52] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. 2019. Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. In *The IEEE Conference on International Conference on Computer Vision (ICCV)*.

[53] Leon Müller, Ken Pfeuffer, Jan Gugenheimer, Bastian Pfleging, Sarah Prange, and Florian Alt. 2021. SpatialProto: Exploring Real-World Motion Captures for Rapid Prototyping of Interactive Mixed Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[54] Michael Nebeling, Katy Lewis, Yu-Cheng Chang, Lihan Zhu, Michelle Chung, Piaoyang Wang, and Janet Nebeling. 2020. XRDirector: A role-based collaborative immersive authoring system. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[55] Michael Nitsche and Sanjeev Nayak. 2012. Cell phone puppets: turning mobile phones into performing objects. In *International Conference on Entertainment Computing*. Springer, 363–372.

[56] nvrmind. 2018. ANIMVR Revolutionizes Your 3D content Production. https://nvrmind.io/. [Online accessed 8-August-2021].

[57] Ye Pan and Kenny Mitchell. 2020. PoseMMR: A Collaborative Mixed Reality Authoring Tool for Character Animation. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 758–759. https://doi.org/10.1109/VRW50115.2020.00230

[58] Min Je Park, Min Gyu Choi, Yoshihisa Shinagawa, and Sung Yong Shin. 2006. Video-Guided Motion Synthesis Using Example Motions. *ACM Trans. Graph.* 25, 4 (Oct. 2006), 1327–1359. https://doi.org/10.1145/1183287.1183291

[59] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7753–7762.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021).

[61] Mir Rayat Imtiaz Hossain and James J Little. 2018. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 68–84.

[62] Roblox. 2022. Animation Capture | Roblox Creation Documentation. https://create.roblox.com/docs/building-and-visuals/animation/animation-capture. [Online accessed 11-September-2022].

[63] Rec Room. 2022. Rec Room - Build and Play Games Together. https://recroom.com/. [Online accessed 11-September-2022].

[64] Leonid Sigal and Michael J Black. 2006. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR* 120 (2006).

[65] Storyblocks. 2017. Storyblocks: Create More Video, Faster Than Ever. https://www.storyblocks.com/video. [Online accessed 8-August-2021].

[66] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. 2021. Monocular, One-stage, Regression of Multiple 3D People. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[67] Kosuke Takahashi, Dan Mikami, Mariko Isogawa, Yoshinori Kusachi, and Naoki Saijo. 2019. VR-based Batter Training System with Motion Sensing and Performance Visualization. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 1353–1354. https://doi.org/10.1109/VR.2019.8798005

[68] Atima Tharatipyakul, Kenny T. W. Choo, and Simon T. Perrault. 2020. Pose Estimation for Facilitating Movement Learning from Online Videos. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Salerno, Italy) *(AVI '20)*. Association for Computing Machinery, New York, NY, USA, Article 64, 5 pages. https://doi.org/10.1145/3399715.3399835

[69] Tvori. 2019. Tvori. http://tvori.co/. Accessed: 2021-09-08.

[70] Unity. 2021. Unity Manual: Animation System Overview. https://docs.unity3d.com/Manual/AnimationOverview.html. [Online accessed 8-August-2021].

[71] Unity. 2022. Unity Avatar Mask. https://docs.unity3d.com/Manual/class-AvatarMask.html. [Online accessed 11-September-2022].

[72] Andreia Valente, Augusto Esteves, and Daniel Lopes. 2021. From A-Pose to AR-Pose: Animating Characters in Mobile AR. In *ACM SIGGRAPH 2021 Appy Hour* (Virtual Event, USA) *(SIGGRAPH '21)*. Association for Computing Machinery, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3450415.3464401

[73] Daniel Vogel, Paul Lubos, and Frank Steinicke. 2018. AnimationVR - Interactive Controller-Based Animating in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 1–1. https://doi.org/10.1109/VR.2018.8446550

[74] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 601–617.

[75] Cheng Yao Wang, Shengguang Bai, and Andrea Stevenson Won. 2020. ReliveReality: Enabling Socially Reliving Experiences in Virtual Reality via a Single RGB camera. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 710–711. https://doi.org/10.1109/VRW50115.2020.00206

[76] Nora S Willett, Hijung Valentina Shin, Zeyu Jin, Wilmot Li, and Adam Finkelstein. 2020. Pose2Pose: Pose Selection and Transfer for 2D Character Animation. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 88–99. https://doi.org/10.1145/3377325.3377505

[77] Wei Wu, Justin Hartless, Aaron Tesei, Venkata Gunji, Steven Ayer, and Jeremi London. 2019. Design assessment in virtual and mixed reality environments: Comparison of novices and experts. *Journal of Construction Engineering and Management* 145, 9 (2019).

[78] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. 2018. Pose Flow: Efficient Online Pose Tracking. In *BMVC*.

[79] 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.

[80] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. 2019. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE International Conference on Computer Vision*. 7760–7770.

[81] Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. 2020. <i>ARAnimator</i>: In-Situ Character Animation in Mobile AR with User-Defined Motion Gestures. *ACM Trans. Graph.* 39, 4, Article 83 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392404

[82] Wenjie Yin, Hang Yin, Kim Baraka, Danica Kragic, and Mårten Björkman. 2022. Dance Style Transfer with Cross-modal Transformer. *arXiv preprint arXiv:2208.09406* (2022).

[83] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. 2018. Deep Network for the Integrated 3D Sensing of Multiple People in Natural Images. In *Advances in Neural Information Processing Systems 31*.

[84] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. 2020. Semantics-Guided Neural Networks for Efficient Skeleton-Based Human Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.