iPose: Interactive Human Pose Reconstruction from Video

Jingyuan Liu
The University of Tokyo, Japan
jliucb@connect.ust.hk

Li-Yi Wei
Adobe Research, USA
liyiwei@acm.org

Ariel Shamir
Reichman University, Israel
arik@runi.ac.il

Takeo Igarashi
The University of Tokyo, Japan
takeo@acm.org

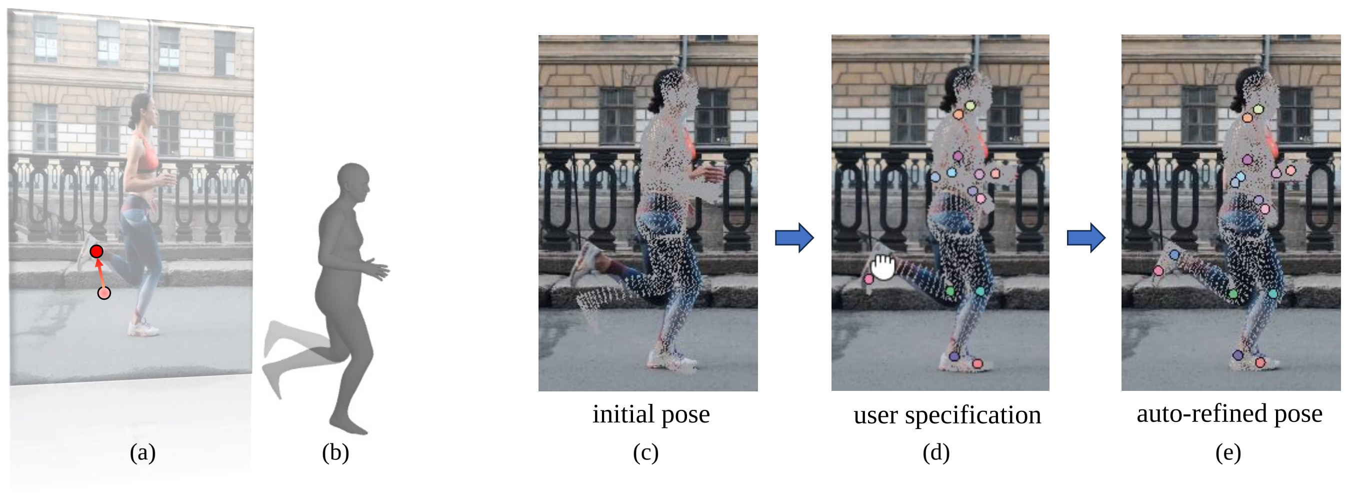

Figure 1: We present *iPose*, a system for interactively reconstructing 3D human poses (b) via simple 2D operations on top of video frames (a). Given an initial 3D pose (with errors) from a computer vision approach (c), a user can easily manipulate the 3D pose via 2D joint handles (d), and our algorithm will automatically refine the 3D pose by "snapping" it to the video subject (e).

ABSTRACT

Reconstructing 3D human poses from video has wide applications, such as character animation and sports analysis. Automatic 3D pose reconstruction methods have demonstrated promising results, but failure cases can still appear due to the diversity of human actions, capturing conditions, and depth ambiguities. Thus, manual intervention remains indispensable, which can be time-consuming and require professional skills. We thus present *iPose*, an interactive tool that facilitates intuitive human pose reconstruction from a given video. Our tool incorporates both human perception in specifying pose appearance to achieve controllability, and video frame processing algorithms to achieve precision and automation. A user manipulates the projection of a 3D pose via 2D operations on top of video frames, and the 3D poses are updated correspondingly while satisfying both kinematic and video frame constraints. The pose updates are propagated temporally to reduce user workload. We evaluate the effectiveness of *iPose* with a user study on the 3DPW dataset and expert interviews.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; **Graphical user interfaces**; *Graphics input devices.*

KEYWORDS

Monocular reconstruction; human pose estimation; video processing; user interface.

ACM Reference Format:
Jingyuan Liu, Li-Yi Wei, Ariel Shamir, and Takeo Igarashi. 2024. iPose: Interactive Human Pose Reconstruction from Video. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3613904.3641944

1 INTRODUCTION

Reconstructing 3D human poses from monocular video provides a more accessible means to studying human movements than sensor-based 3D motion capture (MoCap), which is intrusive and requires specific equipment. Diverse and realistic human movements captured by videos can also improve digital character animation workflows.

Nonetheless, obtaining accurate 3D poses from video is still a challenge. In recent years, advancements in computer vision have shown promising results in automatic 3D human pose reconstruction from video [6, 32, 35]. However, pose reconstruction can still fail in many scenarios, such as under self-occlusions, unusual poses, and the inherent depth ambiguity. In addition, the limited diversity of pose and body shape variations in training datasets often do not generalize well to in-the-wild videos. Improving reconstruction accuracy by designing more sophisticated automatic methods or by collecting datasets with greater pose and shape diversity is difficult and might still produce results with artifacts. In contrast, both identifying errors in pose reconstruction results and specifying corrections are relatively easier tasks for human perception than for automatic video processing algorithms. We thus propose to introduce human interventions in the reconstruction process to eliminate artifacts and achieve better results.

On the other hand, manual reconstruction of 3D poses using commercial software [16] is complex and tedious, requiring professional 3D pose editing skills. Some animation workflows [29, 40] incorporate human pose priors to simplify the editing while retaining the realism and naturalness of human movements. However, pose priors obtained from general human movements might not be applicable for every individual (e.g., stroke patients with gait asymmetry). Although some mismatches between the reconstructed poses and the video subject's poses can be acceptable in animation workflows, such pose reconstruction methods cannot support kinesiological analysis scenarios.

In this work, we aim to bridge the gap between full manual pose reconstruction and automatic methods by designing an intelligent interface, *iPose*, for interactive human pose reconstruction from video. As shown in Fig. 1(a-b), the input to our system includes video frames and initial 3D poses estimated by an automatic computer vision method, which may contain errors. A user identifies errors through mismatches between the projection of 3D pose and the subject in the video frame and manually corrects them. Since the video is the only reference for pose reconstruction, we postulate that editing the human pose within the visible viewplane (i.e., video frame) is easier than in 3D space. We thus design 2D operations to make the 3D pose editing task simple. In particular, the user can locally align a specific body part by dragging 2D joint handles (the colored circles in Fig. 1(d)) on the projection to modify the 3D pose.

At the core of our method is a mapping from 2D position changes of joint handles to 3D pose parameters. This 2D to 3D mapping is ambiguous and we incorporate both kinematic and video frame constraints to address this ambiguity. In addition, because the precision and efficiency of humans' manual operations are limited, we compensate them with algorithms. Specifically, after the user specifies the pose of a local body part, our algorithm utilizes video frames to automatically refine the user's specification by aligning the projection of the 3D body part to its segmentation in the video frame (Fig. 1(e)). Upon such specifications at one frame, *iPose* propagates the changes to future frames using automatic video processing to reduce the user workloads.

We evaluate the effectiveness and accuracy of *iPose* via a user study on the ground-truth 3D poses in the 3DPW dataset [38] and interviews with domain experts in sports analysis and rehabilitation. To the best of our knowledge, our system is the first approach to combine interactive and intelligent tool in pose reconstruction from videos and utilize human factors to make video-based pose reconstruction more accurate and controllable.

2 RELATED WORK

Video-based MoCap. 3D human pose reconstruction methods in computer vision are implemented by either regressing 3D joint positions from images [5, 30] or recovering 3D joint rotations by utilizing a 3D body model [33, 41]. In both types the accuracy can be affected by many factors, such as viewpoints, motion blurs, and pose and shape variations in the training datasets. Research aiming at improving reconstruction accuracy includes collecting datasets with greater pose diversity [8, 14, 20], introducing physical constraints [31, 35, 39], and designing better neural network architecture [13]. Our approach builds upon such automatic methods, but addresses their errors with human interventions. In the annotation of the Motion-X dataset [26], the initial poses from automatic computer vision methods are refined by fitting to detected 3D joint positions via optimization. However, such automatic refinement still results in errors in the optimized poses and requires human verification to exclude videos with artifacts. The limitation in such optimization-based solutions motivates us to seek analytical [34] mappings between user operations and human pose parameters to retain full controllability in users' interventions.

Interactive Character Posing. Interactive character posing has gained significant attention in computer graphics and animation, driven by the growing demand for user-friendly tools in various creative industries such as gaming, film, and virtual reality. When editing without a reference motion, multiple constraints have been proposed to indicate how the remaining parts of the body are updated with respect to a moving joint. Examples of constraints include human pose prior captured with latent space [19, 29, 37], inverse kinematics [1] and user-specified fixed joints [40]. However, in monocular pose reconstruction, such pose prior might not match the video subject's pose due to individual movement differences. In contrast, the reference motion from the video provides stronger cues of pose constraints than other priors. We make use of the cues on body part contours in video frames to ~~in return~~ reduce the ambiguity of IK. A tool closest to our purpose is DeepMotion's Rotoscope Pose Editor [12], which only allows per-joint position specification without any constraints, easily resulting in distorted poses.

Inverse kinematics. IK is a fundamental technique in computer graphics and animation, enabling the realistic and efficient manipulation of articulated structures. In general, IK techniques can be classified into numerical [1], analytical [34], data-driven [29, 37] and hybrid solutions [25]. Please refer to a comprehensive survey [2] for details on the four types. We build our approach on analytical IK to support live interactions. IK has been well-recognized for its ambiguity as multiple solutions exist. The most common probabilistic approaches [19, 29, 37] that suggest the pose closest to mass distribution do not fit our case-specific goal. Some existing work solves the ambiguity by incorporating animators in the loop, such as selecting from possible pose suggestions [10]. We adopt a similar idea to [21] that uses images as constraints in inverse kinematics. While they only penalized bone segments fall out of the body silhouette, our method explicitly fits the body surface to silhouette to constrain poses at a finer level.

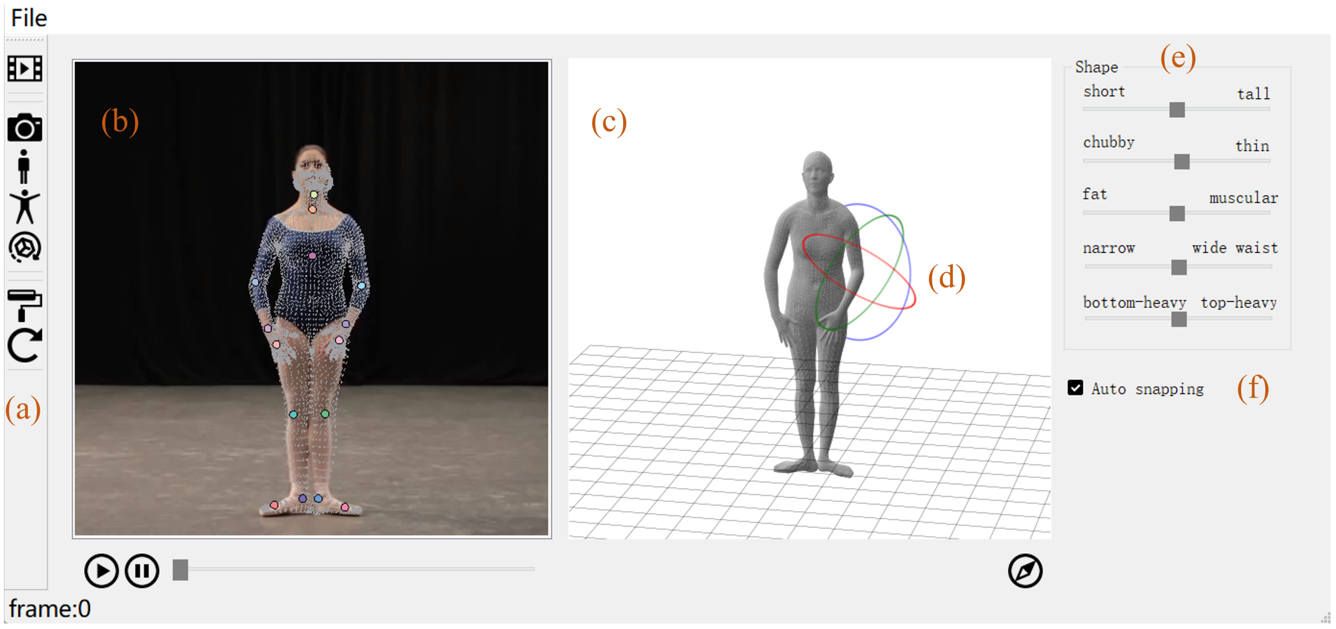

Figure 2: The user interface in the pose mode. (a) Panel buttons for loading videos, switching among modes, and controls for temporal propagation and undo; (b) a video viewer for supporting 2D operations on top of video frames; (c) 3D pose viewer for allowing 3D pose manipulations with a traditional rotation widget (d); (e) shape sliders for fine-tuning body shape parameters; (f) control option to disable the automatic snapping of body parts to video frames.

3 PRELIMINARIES

We adopt the SMPL model [27] as the 3D pose representation in our pipeline due to its efficient computation of realistic shapes under different poses. The SMPL model is parameterized by 72-dimensional pose parameters θ, which are local axis-angle 3D joint rotations for each of the 24 joints, and 10-dimensional shape parameters β, which are coefficients of blend shapes obtained by Principle Component Analysis (PCA). The SMPL model $M(\beta, \theta)$ maps shape and pose parameters to vertices of a 3D mesh. The model also defines a regressor function J that regresses 3D joint positions from mesh vertices, and body part segmentations that label each vertices with a part ID.

We adopt a weak perspective camera in projecting the 3D SMPL model to fit the 2D landmarks as in most automatic fitting methods [6, 33, 41]. Specifically, the camera is parameterized by the translation (denoted as t_x and t_y) and scale (denoted as s_x and s_y) along the x and y dimensions. A 2D joint detection method is first applied to compute the landmarks of the n joints in an image, denoted as $(x_i, y_i), i = 1, 2, ..., n$. The automatic fitting solves for the optimal camera and SMPL parameters that minimize the distances between the projection of 3D joints $(X_i, Y_i, Z_i), i = 1, 2, ..., n$ and the 2D landmarks.

4 USER INTERFACE

Our main user interface (Fig. 2) follows a conventional 2D-to-3D modeling layout [17, 36] that contains a video viewer (Fig. 2(b)) and a 3D body model viewer (Fig. 2(c)) for providing synchronized 2D and 3D previews. The user interface works in three modes: camera, shape, and pose mode, supporting the manipulation of the three factors affecting the alignments between the projection and the video subject.

Camera mode. We adopt a bounding box as a widget for manipulating camera parameters in the video space. As shown in Fig. 3(a), a user directly drags and scales the bounding box to align the projected 2D point cloud to the contour of the video subject at a global level. The translation and scale of the bounding box modify the underlying parameters of the weak perspective camera.

Shape mode. We follow the conventional solution that tunes shapes with sliders. Among the ten SMPL body shape parameters, we retain 5 with visual semantics (Fig. 2(e)). Although there are more advanced semantic sliders (e.g., Semantify [18]), since they modify shape globally, we adopt the original SMPL coefficients for simplicity.

Pose mode. Obtaining users' specifications on poses is the core component of the interface. In commercial software, such as Unity and Blender, 3D character poses are often manipulated via 3D gizmos (Fig. 4(a)), which separate the operations on each dimension to reduce the ambiguity of positioning in 3D with 2D operations (Fig. 4(b)). However, in our system where the input video frame is the only reference, the space for editing is reduced to a 2D screen space instead of the 3D space. We thus provide 2D joint handles that allow a user to specify the desirable key joint positions within the visible viewplane (i.e., video frame) in the 3D space. For simplicity, among the 24 joints we visualize joint handles for 15 mobile joints (e.g., wrist and ankle) and hide passive joints (e.g., shoulders and spine) from users. When a 2D joint handle is being dragged, only kinematic constraints (Section 5.2) are applied to provide a live

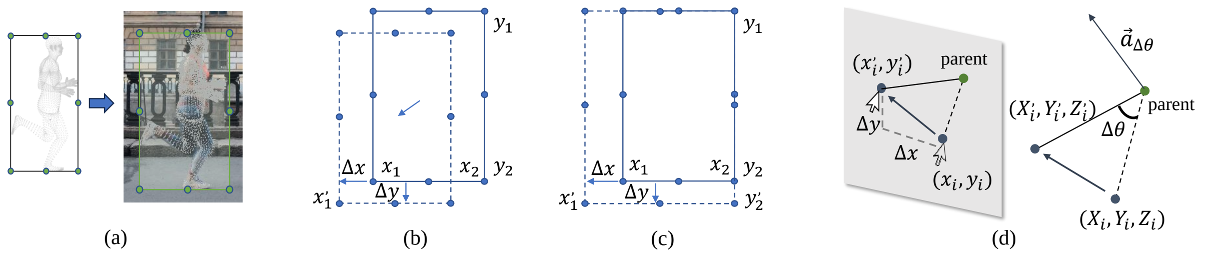

Figure 3: When a user globally aligns the projection of the 3D pose to the video subject with a bounding box (a), the 2D operation in the screen space produces a displacement vector $(\Delta x, \Delta y)$ in both dragging (b) and scaling (c) of the bounding box. Similarly, when dragging a joint handle, the 2D operation produces a displacement vector $(\Delta x, \Delta y)$ that maps to the change in 3D joint angle (d).

preview. After a joint handle is released, our system automatically starts to parse the video frame to refine the pose (Section 5.3).

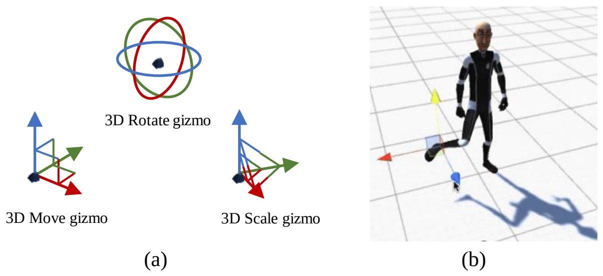

Figure 4: (a) common gizmos for manipulating 3D objects [3]; (b) an example of 3D character pose editing in Unity (figure credit: Luis Bermudez).

5 METHODOLOGY

This section introduces the algorithms for mapping from user operations in the screen space to the camera parameters (Section 5.1) and pose parameters (Section 5.2), utilizing video frames to refine poses (Section 5.3), and temporal propagation of parameter changes to reduce user interventions (Section 5.4).

5.1 Camera

When a user modifies the camera via the bounding box (Fig. 3(a)), the corresponding operation produces a displacement vector $(\Delta x, \Delta y)$ in the screen space (Fig. 3(b)(c)). Δx and Δy produce new camera parameters (t_x', s_x') and (t_y', s_y') along both dimensions independently. Without loss of generality, we give the deduction of x dimension of the left bottom control point. The deduction processes for the y dimension and other control points on the bounding box are similar, and thus omitted here.

Dragging. When a user drags the bounding box (Fig. 3(b)), Δx only modifies the translation parameter t_x. Specifically, given the weak perspective projection $x_i = [(X_i + t_x) \cdot s_x + 1] \cdot w/2$, where w is the width of the video frame, the new projection after dragging is: $x_i + \Delta x = [(X_i + t_x') \cdot s_x + 1] \cdot w/2$. Jointly solving these two equations gives the new translation parameter t_x':

$$t_x' = t_x + \frac{2}{w} \times \frac{\Delta x}{s_x} \tag{1}$$

Scaling. When a user scales the bounding box by dragging a control point (Fig. 3(c)), Δx modifies both the translation parameter t_x and scale parameter s_x. According to the weak perspective projection, the projection of two points $X1$ and $X2$ after scaling in the case of Fig. 3(c) is $x_1 + \Delta x = [(X_1 + t_x') \cdot s_x' + 1] \cdot w/2$ and $x_2 = [(X_2 + t_x') \cdot s_x' + 1] \cdot w/2$, respectively. Jointly solving these two equations gives the new scale parameter s_x':

$$s_x' = \frac{x_2 - x_1'}{x_2 - x_1} \cdot s_x \tag{2}$$

Denote the coefficient in Equation (2) $\frac{x_2 - x_1'}{x_2 - x_1}$ as r. Then according to the equivalence of a random point X_i: $x_2 + r \cdot (x_i - x_2) = [(X_i + t_x') \cdot s_x' + 1] \cdot w/2$, the new translation parameter is:

$$t_x' = t_x + \frac{1}{s_x} + \frac{(1 - r) \cdot x_2}{r \cdot s_x \cdot \frac{w}{2}} - \frac{1}{r \cdot s_x}$$

5.2 Pose

In the pose mode, the change in 2D position $(\Delta x, \Delta y)$ of the i-th joint is mapped to its change in 3D rotation $\Delta\theta$. Specifically, as shown in Fig. 3(d), the new position of a joint in the screen space (x_i', y_i') is first inversely projected into (X_i', Y_i', Z_i') in the 3D world space under joint kinematic constraints. Then the angle between the vectors of the old and new joint positions with the parent joint position, along with their normal directions, formed the incremental axis-angle 3D rotation $\Delta\theta$. To model kinematic constraints, we categorize the joints on the SMPL model into the following four types according to mobility (see Fig. 5(a)).

Fixed-based joints are those considered less movable due to their stability requirements, e.g., shoulder collars and hips. They can be indirectly manipulated through global translations or orientations of the torso, which helps in maintaining overall rigidity.

First-level joints are joints that directly connect to fixed-based joints, e.g., elbows, knees, and chest. A fixed-based joint, a first-level joint, and a second-level joint form a two-bone structure (Fig. 5(b)), connected with two bone segments. When a user drags a first-level joint handle on the video frame, the positions of the fixed-based and second-level joints remain unchanged. The trajectory of the

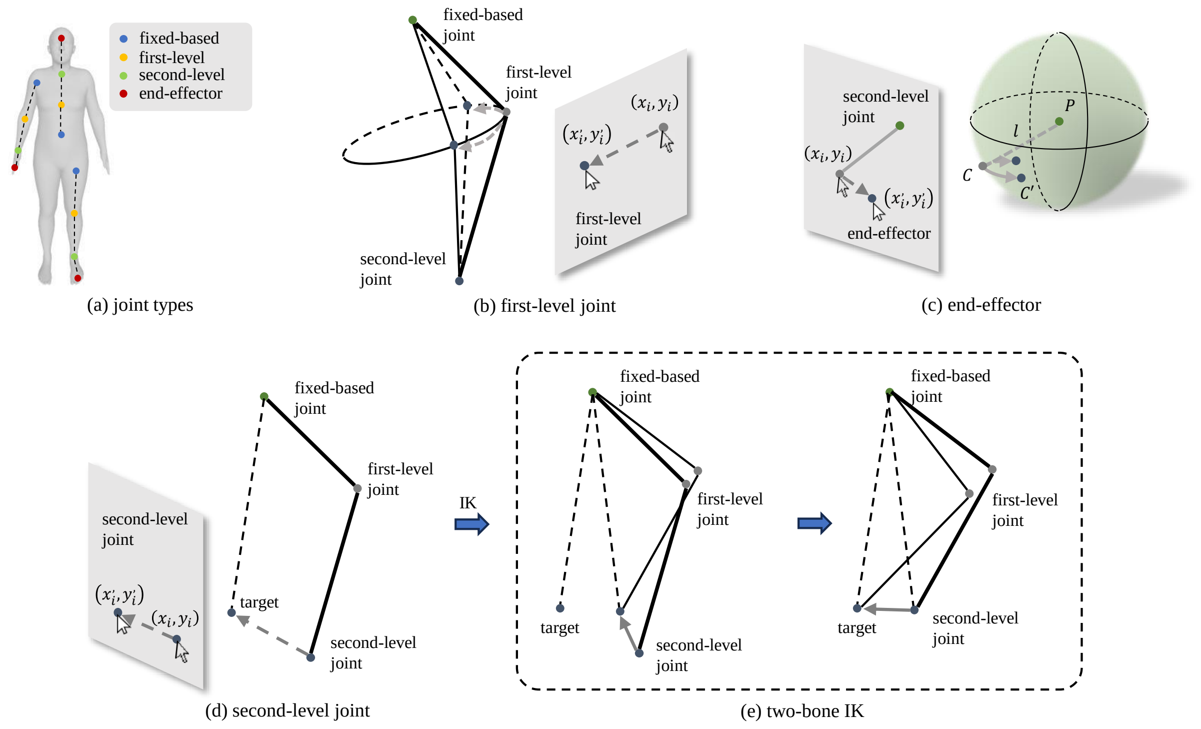

Figure 5: The manipulation of a 2D joint handle produces different effects according to joint types (a). Manipulating a 3D body model is a special case of general 3D manipulation in that it only involves deciding the relative depths of a joint to its parent joint, and this task is much more constrained when the lengths of limbs are fixed. With parent joints serving as anchors in the 3D space, we construct auxiliary proxies (i.e., 3D ring for first-level joints (b), 3D sphere for end-effectors(c), and minimum skew plane for second-level joints in (d)) at parent joints, and then the relative depth is found by a ray casting to these auxiliary proxies. Second-level joints and first-level joints are formulated with two-bone kinematics chains. When the target position of a second-level joint is set (d), the first-level joint is first rotated so that the two dotted lines have the same length (e left), and then the kinematics chain is rotated at the base joint to align the second-level joint with the target (e right).

first-level joint in 3D is thus constrained by a 3D ring to maintain constant bone lengths.

Second-level joints are those connected to a fixed-based joint via first-level joints, e.g., the wrist, ankle, and neck. Their position changes often largely influence the kinematic chain. When a user specifies the target position of a second-level joint on the video frame, its target position in 3D is determined by inverse projecting the 2D position into a minimum skew viewplane [11](Fig. 5(d)). Then the 3D positions of the first-level and second-level joints are both updated according to the two-bone IK algorithm [9] (Fig. 5(e)).

End-effectors are joints whose movements are relatively more independent and do not largely propagate back along kinematics chains, such as head, hand, and foot. End-effectors connect to their parent joints with a single bone segment (Fig. 5(c)) and are thus constrained by a 3D sphere to maintain constant bone length.

There still exist ambiguities despite the above-mentioned geometry constraints. Specifically, for both the first-level joints and end-effectors, the 3D joint can move towards two possible directions along the 3D ring or sphere [9]. We address this ambiguity with a heuristic that a user's manual intervention fine-tunes the initial pose and should choose a side that retains the adduction or abduction movement tendency. The algorithm thus checks the relative depth between the manipulated joint and its parent joint in the initial pose, and selects a side that makes the absolute relative depth increase. In case the heuristic fails in specific scenarios, the user can seek to the traditional 3D rotation widget (Fig. 2(d)) to fix the errors. For the second-level joints, an inherent ambiguity for two-bone IK is the uncertain orientation of the first-level joint during the update. We empirically set the forward orientation as the transformed forward axis of a first-level joint under T-pose. A user can overwrite the forward orientation via the first-level joint handle, which explicitly specifies the orientation of the two-bone kinematics chain.

5.3 Pose Refinement

This section introduces the method for using video frame constraints to automatically refine the 3D pose. To maintain controllability, the refinement is conducted locally at body-part level instead of globally to the whole body pose. The main idea is to first find the transformation via the pixel-level mismatch between the projection of a body part and its segmentation in the background video frame, and then use the transformation to refine the foreground 3D pose.

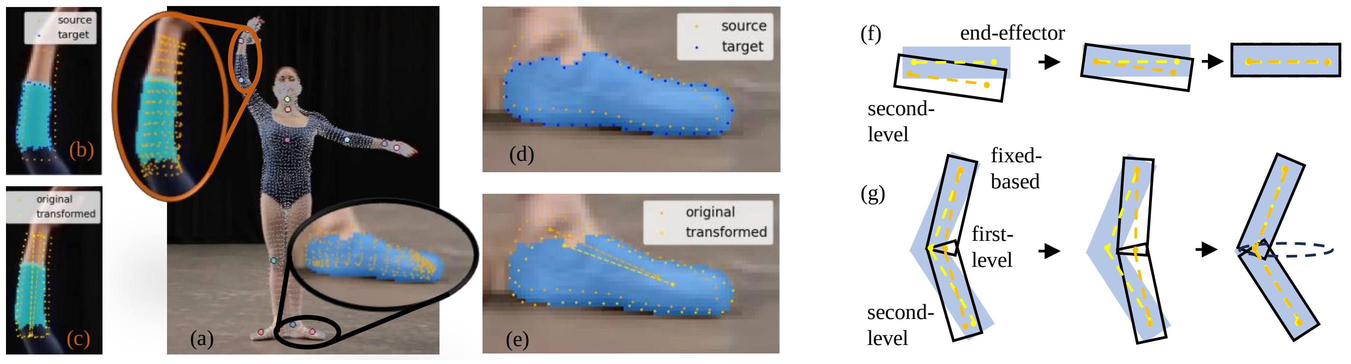

Figure 6: An illustration of using video frames as constraints to refine 3D pose. (a) Projection (orange point clouds) and segmentation masks (blue and cyan masks) of body parts. (b)(d) Re-sampled contours of the projections and segmentation masks. (c)(e) Transformed contours of body parts to align with the mask contours. (f)(g) Refinement processes of the end-effectors and two-bones, respectively.

We first use the projected foreground body parts to parse video frames. As shown in Fig. 6(a), the projection of the 3D vertices of the body part (the orange point clouds) is used as an initial hint to retrieve the segmentation of the body part in the background video frame (the blue and cyan masks). We implement the video frame body part segmentation with Segment Anything [23], using the bounding box of projected vertices and the 2D position of the bone center as prompts. However, the shapes of masks can be noisy due to the influence of clothes and uncertain body part boundaries, and thus they do not perfectly match the shapes of projected vertices (Fig. 6(b)). We thus seek to use a robust alignment method to find an optimal alignment that best snaps the body part projection to the mask. Specifically, the projected body part vertices and the segmentations are both converted to contour representations (Fig. 6(b)(d)). Then we run a RANSAC [15] on both contours to compute robust alignment with inliers identified by the algorithm. The results are the optimal 2D translation and rotation, which are applied to the projected joint positions to produce two displacement vectors at both the parent joint and the user-specified joint (Fig. 6(c)(e)).

Then the displacements are used as constraints from video frames to update the pose (Fig. 6(f)(g)) and meanwhile satisfying the kinematic constraints using the same method as in Section 5.2. The only difference is that the 2D displacements are automatically computed instead of specified by a user. Note that the pose refinement process includes the video frame segmentation and the optimal alignment that are not fully controllable. In case the algorithm produces errors, a user can decline the snapping result via the undo button (Fig. 2(a)) and disable the automatic snapping (Fig. 2(f)) to retain the manual correction results.

5.4 Temporal Propagation

The goal of temporal propagation is to automatically update future τ frames upon a user's specification at frame t_f by using the user's specifications and the temporal consistency of video frames. Since the errors in the camera and pose parameters do not follow temporal consistency and are random in future frames, we formulate temporal propagation as an optimization problem that allows constraints to influence each other. We use user-specified 2D joint handles (x_{t_f}, y_{t_f}) as prompts to predict the landmark positions in the future frames $(x_p, y_p), p \in [t_f+1, t_f+\tau]$ using a point tracking method (coTracker [22] in our implementation). Denote the new pose as $\theta'_p = \theta_p + \Delta\theta_p$. The objective is to find pose corrections $\Delta\theta_p, p \in [t_f+1, t_f+\tau]$ and camera corrections $(\Delta s_p, \Delta t_p), p \in [t_f+1, t_f+\tau]$ in future frames, so that the projections of the 3D joints under new parameters best fit (x_p, y_p). The objective function also includes a temporal smoothness term on joint rotations, and constraints on the magnitude of Δs and Δt:

$$\begin{aligned}\underset{\Delta s, \Delta t, \Delta\theta}{\arg\min} \sum_{p=t_f+1}^{t_f+\tau} \sum_{i=1}^{n} \|\Pi\left(J\left[M(\beta, \theta'_p)\right]_i\right) - (x_{p,i}, y_{p,i})\|^2 \\ + \lambda_{smooth} \sum_{p=t_f+1}^{t_f+\tau} \sum_{i=1}^{n} \|\theta'_{p,i} - \theta'_{p+1,i}\|^2 \\ + \lambda_s \sum_{p=t_f+1}^{t_f+\tau} \|\Delta s_p\|^2 + \lambda_t \sum_{p=t_f+1}^{t_f+\tau} \|\Delta t_p\|^2\end{aligned} \tag{3}$$

Both joint position tracking and the optimization can also produce errors, where users can intervene manually again at specific errors.

6 RESULTS

We implemented *iPose* with Python3.6 and PyQt5. See Appendix A for implementation details. This section presents the results of a user study evaluating the accuracy of pose reconstruction and the usability of *iPose* (Section 6.1) and expert interviews (Section 6.2). See Appendix B.2 and Appendix B.3 for more qualitative results.

6.1 User Study

Procedure. We recruited 8 participants (A1~A8, aged 21~33, three female). A1 and A3 had experience in 3D modeling, A2 had background in monocular human pose reconstruction, A4, A7 and A8 had background in pose analysis, and A5 and A6 were novice in all of these fields. In order to quantitatively measure the human pose reconstruction accuracy by *iPose*, we use the test set of the 3DPW dataset [38], which provides ground-truth 3D poses for 24 in-the-wild videos, with duration ranging from 387 to 1,824 frames. See Appendix B.1 for details on dataset processing. We obtained the initial parameters with a state-of-the-art automatic human pose reconstruction method (ROMP [32]) and let participants correct errors with *iPose*. We did not compare our method against traditional pose editing methods (e.g., using Blender) because they have a significantly steeper learning curve.

In each session, a participant was first trained on two sample videos outside the 3DPW dataset, and then corrected three videos in a think-aloud manner. There was no limitation on editing time for each video; the participants ended editing when they were satisfied with the results. We recorded user intervention statistics, such as editing time and usage of functionalities, as well as participants' comments. See Table 2 for detailed statistics of the 24 videos. At the end of the session, the participant completed a standard System Usability Scale (SUS) questionnaire [7]. Each session took about four hours, and the training session took about 20 minutes on average.

Accuracy. We adopt the conventional metrics of pose accuracy, including mean per joint position error (MPJPE), Procrustes-aligned MPJPE (PA-MPJPE), and mean per vertex position error (MPVPE). Table 1 shows the accuracy achieved by *iPose* within the editing time as listed in Table 2. We compared with automatic computer vision methods, whose performance is derived from the original papers.

Method	ROMP [32]	PyMAF [41]	VIBE [24]	CycleAdapt [28]	*iPose*
MPJPE ↓	85.5	92.8	82.9	85.8	**77.89**
PA-MPJPE ↓	53.3	58.9	51.9	53.9	**47.52**
MPVPE ↓	103.1	110.1	99.1	102.1	**95.51**

Table 1: A comparison of pose reconstruction accuracy on the 3DPW [38] test set. (unit: mm)

Our method achieved significant accuracy improvements over existing automatic methods. In addition, we observed scenarios where human perception is advantageous at addressing issues challenging for automatic methods. For example, as shown in Fig. 7(a), when a subject is walking towards the camera, the aperture problem makes the poses of limbs difficult to recover, but can be addressed with human's prior knowledge of walking movements. The video subject's left arm was heavily occluded in the Fig. 7(b). However, with the understanding of the action that the video subject is pointing to a distant area, A1 easily recognized that the left arm should be straight instead of curled. Fig. 7(c) contains challenging lighting conditions with heavy occlusion, which is solved by a hybrid of partial body parts observation and the knowledge of the getting off car action. Fig. 7(d) is an example of camera perspective. A7 explicitly commented that it is difficult to compromise the right foot to align with the frame, because it would result in the toe pointing upwards and not matching reality. The camera distortion also caused the video subject's left leg to appear longer than the right one. A7 edited referring to the 3D viewer instead of fixing the alignment on the video frame.

We also inspected video frames with large errors after corrections with *iPose* and have identified the following main sources of errors. First of all, the 3DPW dataset [38] in essence provides "pseudo" ground truth 3D poses, and thus the precision of ground truth is also limited. We observed many artifacts in ground truth 3D poses, such as the left arm in Fig. 7(c) and the unstable leg poses in Fig. 9(d). In these cases, the poses resulting from *iPose* are better than ground truth regarding physical plausibility. Besides, a general type of error is caused by the mismatch between the estimated body shape and the actual video subjects' body shape. Even though *iPose* supports adjusting body shapes via sliders, there is an ambiguity in editing the global scale of video subjects. As illustrated in Fig. 8, when there is a misalignment between the projection and the video subject, a user can either modify the camera scale parameter (Fig. 8(d)) or the body height parameter (Fig. 8(e)). Despite the alignment on the video frame, there remains a near-constant error in joint positions in the model space. In order to verify the influence of shape difference, we applied the ground-truth poses in the 3DPW test set onto SMPL with body shapes after editing with *iPose*, resulting in an average MPJPE 40.15mm, PA-MPJPE 20.71mm, and MPVPE 51.83mm (corresponding to 51.55%, 43.58%, and 54.27% of the total errors in Table 2). The third type of error comes from camera models. We adopt a weak perspective camera model to be consistent with computer vision methods, but this introduces an approximation error with the perspective projection [42]. Besides, the orientation of the camera often causes failures in recovering the video subject's global orientation (e.g., Fig. 9(b)). Finally, the depth ambiguity and occlusions remain challenging to solve when no precise semantics exist. For example, the bending of the upper body in Fig. 9(a) and the right hand in Fig. 9(c) remain uncertain.

Usability. The average SUS score was 80.93 (*SD=11.38*). While all participants found *iPose* easy to learn, they agreed that it is specific-purpose and not designed for the general public users. All participants agreed or strongly agreed that functions in *iPose* are well-integrated, and we observed that all functions were covered during their editing. For each video, apart from correction operations, the participants spent large portion of editing time previewing the action, identifying the errors and considering about the corrections. Since all frame contain certain amount of errors, the participants focused on major errors to trade-off between accuracy and editing time. Participants switched from 2D joint handles to the 3D rotation widget mainly to move joints perpendicular to the video frame (e.g., limbs in Fig. 7(a)) and adjust the orientation of the head, indicating using one 2D joint handle to control the head is insufficient. Participants disabled auto-snapping for end-effectors because subjects in the 3DPW dataset wore hats and IMUs, making the snapping of heads and hands unstable. Participants often used the temporal propagation to reuse the correction results mainly when the issues in the following frames were similar, and when the corrections at a frame would be relatively complex but propagated from a previous frame and then refined were easier. The temporal propagation failed when movements were perpendicular to video frames, such as the walking towards camera action (e.g., Fig. 7(a)).

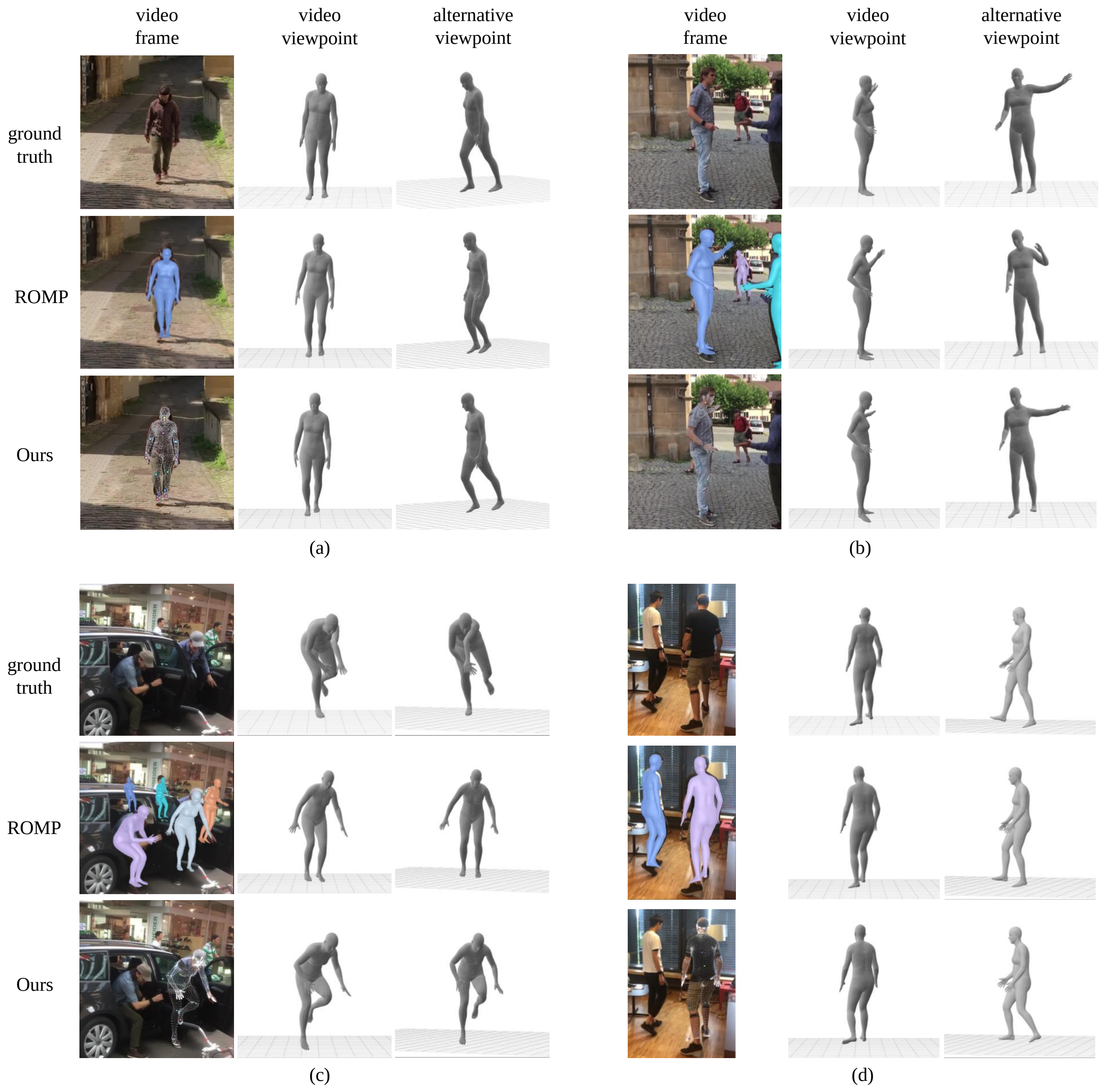

Figure 7: Examples of accuracy improvements due to human perception. (a) The aperture problem of limb movement when the video subject is walking towards the camera is addressed by the prior knowledge of walking. (b) The occluded left hand is recovered by the general understanding of the pointing action. (c) A challenging case with heavy occlusion and extreme lighting conditions was addressed by aligning visible body parts and the knowledge of the getting off-car action. (d) An example shows the influence of camera perspective and distortion, where the participant edited mainly according to the perception.

Design Reflections. Through the user study, we identified the following limitations in the current design of *iPose*. Even though the 3D viewer provides alternative viewpoints to videos, the participants skipped checking under other viewpoints when there were no large misalignments on the video. This is problematic when the depth ambiguity is hardly noticeable. It is thus helpful to simultaneously visualize a few viewpoints to help identify the errors. Besides, we provide one joint handle for manipulating the head following the SMPL model definition, but this is often insufficient, so participants had to use the 3D rotation widget for finer adjustment. It

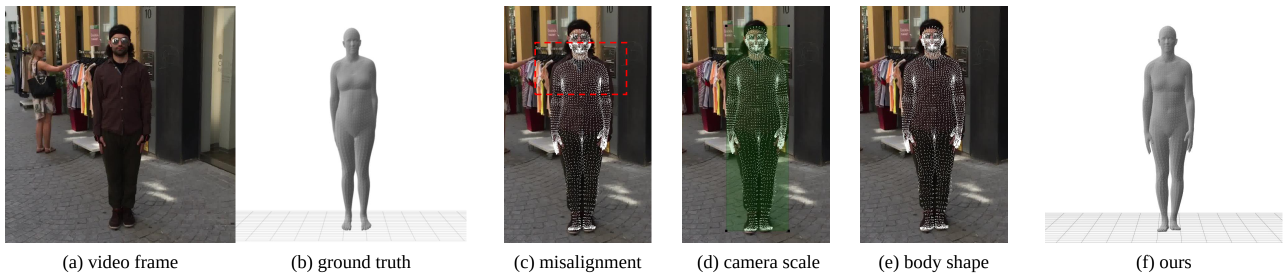

Figure 8: An example illustrating the ambiguity in editing the global scale of the video subject in (a). When there is a misalignment (c) in the global scale between the projection and the video subject, a user can either adjust the camera scale (d) or body shape (e). Although either way can fix the alignment in the video space, the difference in body shapes between (b) and (f) causes a near-constant error in joint positions.

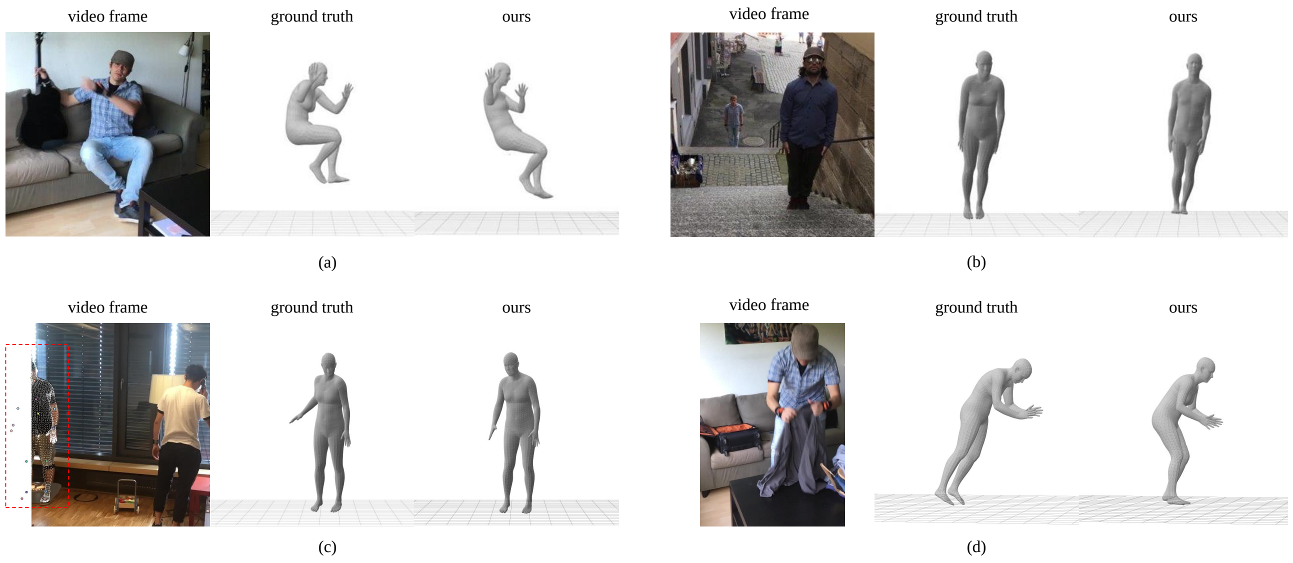

Figure 9: Examples showing the sources of errors after corrections. (a) The bending of the upper body remains uncertain when there is no prior knowledge of how the subject should sit on the sofa. (2) Failure in reconstructing a subject's global orientation due to the influence of camera orientation. (3) An example of partial observation where the right arm pose is hard to infer when this action has minor semantics. (d) An example of artifacts in the ground truth data of the 3DPW dataset [38].

is worth a special design for the head joint, such as manipulating look-at. Finally, the temporal propagation automatically corrects a fixed number of future frames. However, since there are large variations in video subjects' action pace, it would be more desirable to adopt a flexible temporal propagation scheme, where a user can dynamically decide how many frames to propagate according to action complexity and number of consecutive frames with similar errors.

6.2 Expert Interviews

To understand the effectiveness of *iPose* in practice, we conducted semi-structured interviews with a researcher in rehabilitation (**E1**) and a practitioner in sports performance analysis (**E2**). We prepared video footage of rehabilitation exercises and sports actions. During the interview, we first demonstrated our interface and let the experts try it freely. Then we discuss revolving questions such as how they feel about *iPose*, how they position its role in their practice, and suggestions on improvements. Each session lasted about one hour.

Both experts confirmed the usefulness of *iPose* in their practice. **E1**: "*Current clinical gait analysis relies heavily on marker-based capture. Markerless methods are very sensitive to lighting conditions. This tool demonstrates a promising future trend.*" **E2**: "*iPose will be especially useful in two cases: the training of action reproducibility for beginners and the analysis of an elite athlete template [4]. These two cases require accurate poses but currently they can only be conducted empirically by biomechanical analysts.*" **E1** suggests that it would be useful to enable pose correction with events since the requirements for accuracy differ with focus. For example, the range of motion is a key attribute in evaluating a gait cycle, and it would be desirable if a user can quickly locate the timing when a thigh reaches its extreme position and correct the pose at that moment. **E2** mentions that when target users of the system are non-professional athletes, it is recommended to trade some accuracy for speed in displaying pose correction results.

7 DISCUSSIONS AND FUTURE WORK

In this paper, we present an interactive and intelligent tool *iPose* for reconstructing accurate 3D poses from videos. We designed a user interface that allows users to manipulate 3D poses with respect to the video subject with simple 2D operations. The specified 3D pose is automatically refined and the changes are propagated to future frames. We currently use SMPL model, but our approach is generalizable to other 3D body models.

The current prototype contains two main limitations. First, we utilize low-level video frame features (i.e., contour of body parts) as video frame constraints, which is not robust to occlusions and video variations, such as clothes and hairs. In the future, we will explore more robust solutions, such as introducing individualized high-level motion prior of video subject to achieve globally coherent constraints. Second, currently *iPose* only supports modify body shapes globally via the SMPL shape parameters. This is not compatible with unusual video subjects' body shapes, such as an athlete with long legs. It thus worth exploring a local body shape manipulation method to better facilitate the alignment between SMPL model and video subjects.

ACKNOWLEDGMENTS

We thank anonymous reviewers for their valuable comments and suggestions. This work was partially supported by JST AdCORP, Grant Number JPMJKB2302, Japan, and by Israel Science Foundation Grant No. 1390/19.

REFERENCES

[1] Andreas Aristidou, Yiorgos Chrysanthou, and Joan Lasenby. 2016. Extending FABRIK with Model Constraints. *Comput. Animat. Virtual Worlds* 27, 1 (Jan. 2016), 35–57. https://doi.org/10.1002/cav.1630
[2] Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Inverse kinematics techniques in computer graphics: A survey. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 35–58.
[3] AutoCAD. 2022. About Using 3D Gizmos. https://help.autodesk.com/view/ACD/2022/ENU/?guid=GUID-7BD066C9-31BA-4D47-8064-2F9CF268FA15
[4] Roger Bartlett. 2014. *Introduction to sports biomechanics: Analysing human movement patterns.* Routledge.
[5] 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).
[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 *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer, 561–578.
[7] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
[8] Xin Chen, Anqi Pang, Wei Yang, Yuexin Ma, Lan Xu, and Jingyi Yu. 2021. SportsCap: Monocular 3D Human Motion Capture and Fine-Grained Understanding in Challenging Sports Videos. *International Journal of Computer Vision* (Aug 2021). https://doi.org/10.1007/s11263-021-01486-4
[9] John J Craig. 2006. *Introduction to robotics.* Pearson Educacion.
[10] James Davis, Maneesh Agrawala, Erika Chuang, Zoran Popović, and David Salesin. 2006. A sketching interface for articulated figure animation. In *Acm siggraph 2006 courses.* 15–es.
[11] Chris De Paoli and Karan Singh. 2015. SecondSkin: sketch-based construction of layered 3D models. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.
[12] DeepMotion. 2022. Rotoscope Pose Editor. https://blog.deepmotion.com/2022/09/30/rotoscopeposeeditor/.
[13] Moritz Einfalt, Katja Ludwig, and Rainer Lienhart. 2023. Uplift and Upsample: Efficient 3D Human Pose Estimation with Uplifting Transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 2903–2913.
[14] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. 2021. Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 9919–9928.
[15] Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.
[16] CG Geek. 2020. How to Animate 3D Characters in 1 Minute. https://www.youtube.com/watch?v=TjJLIuFKA20&ab_channel=CGGeek.
[17] Yotam Gingold, Takeo Igarashi, and Denis Zorin. 2009. Structured annotations for 2D-to-3D modeling. In *ACM SIGGRAPH Asia 2009 papers.* 1–9.
[18] Omer Gralnik, Guy Gafni, and Ariel Shamir. 2023. Semantify: Simplifying the Control of 3D Morphable Models using CLIP. arXiv:2308.07415 [cs.CV]
[19] Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. 2004. Style-based inverse kinematics. In *ACM SIGGRAPH 2004 Papers.* 522–531.
[20] Christian Keilstrup Ingwersen, Christian Mikkelstrup, Janus Nørtoft Jensen, Morten Rieger Hannemose, and Anders Bjorholm Dahl. 2023. SportsPose – A Dynamic 3D sports pose dataset. arXiv:2304.01865 [cs.CV]
[21] Antoni Jaume-i Capó, Javier Varona, Manuel González-Hidalgo, and Francisco J. Perales. 2010. Adding Image Constraints to Inverse Kinematics for Human Motion Capture. *EURASIP J. Adv. Signal Process* 2010, Article 4 (jan 2010), 13 pages. https://doi.org/10.1155/2010/142354
[22] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. 2023. CoTracker: It is Better to Track Together. *arXiv:2307.07635* (2023).
[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *arXiv:2304.02643* (2023).
[24] 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.
[25] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. 2021. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 3383–3393.
[26] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. 2023. Motion-X: A Large-scale 3D Expressive Whole-body Human Motion Dataset. *arXiv preprint arXiv: 2307.00818* (2023).
[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.
[28] Hyeongjin Nam, Daniel Sungho Jung, Yeonguk Oh, and Kyoung Mu Lee. 2023. Cyclic Test-Time Adaptation on Monocular Video for 3D Human Mesh Reconstruction. In *International Conference on Computer Vision (ICCV).*
[29] Boris N Oreshkin, Florent Bocquelet, Felix G Harvey, Bay Raitt, and Dominic Laflamme. 2021. ProtoRes: Proto-Residual Network for Pose Authoring via Learned Inverse Kinematics. In *International Conference on Learning Representations.*
[30] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR).*
[31] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. 2020. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)* 39, 6 (2020), 1–16.
[32] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. 2021. Monocular, One-stage, Regression of Multiple 3D People. In *ICCV.*
[33] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. 2022. Putting people in their place: Monocular regression of 3d people in depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 13243–13252.
[34] Deepak Tolani and Norman I Badler. 1996. Real-time inverse kinematics of the human arm. *Presence: Teleoperators & Virtual Environments* 5, 4 (1996), 393–401.
[35] Shashank Tripathi, Lea Müller, Chun-Hao P. Huang, Taheri Omid, Michael J. Black, and Dimitrios Tzionas. 2023. 3D Human Pose Estimation via Intuitive Physics. In *Conference on Computer Vision and Pattern Recognition (CVPR).* 4713–4725. https://ipman.is.tue.mpg.de
[36] Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (2011), 90.
[37] Vikram Voleti, Boris Oreshkin, Florent Bocquelet, Félix Harvey, Louis-Simon Ménard, and Christopher Pal. 2022. SMPL-IK: Learned Morphology-Aware Inverse Kinematics for AI Driven Artistic Workflows. In *SIGGRAPH Asia 2022 Technical Communications.* 1–7.
[38] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in The Wild Using

IMUs and a Moving Camera. In *European Conference on Computer Vision (ECCV)*.

[39] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. 2021. Physics-based human motion estimation and synthesis from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11532–11541.

[40] Katsu Yamane and Yoshihiko Nakamura. 2003. Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on visualization and computer graphics* 9, 3 (2003), 352–360.

[41] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. 2021. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11446–11456.

[42] Zhengyou Zhang. 2014. *Weak Perspective Projection*. Springer US, Boston, MA, 877–883. https://doi.org/10.1007/978-0-387-31439-6_115

A IMPLEMENTATION DETAILS

iPose runs on a PC running Ubuntu20.04 (Intel i7 @3.6GHz, 12GB RAM) with an Nvidia GeForce RTX 3080Ti GPU. The videos used in the user study are of resolution 1080p, with a frame rate of 25 fps. Video frames are cropped and scaled according to the initial projection of poses to best fit the video viewer window in the interface. Video frame processing with Segment Anything [23] ran at 0.5 fps on this PC in a separate thread from the main interface. To avoid drifting, we empirically set temporal propagation within five future frames, with duration about 10 seconds. λ_{smooth}, λ_s and λ_t in Equation (3) are all set to 100.

B RESULTS

B.1 User Study Statistics

Table 2 shows the meta data of videos in the 3DPW test set and statistics of user interventions in the user study.

Thirteen out of 24 videos in the 3DPW test set contain two subjects. We randomly picked one of the subjects for annotation for editing time consideration for these videos. "#frames" is the number of valid frames of each video subject as indicated by the 3DPW benchmark. We roughly balance the number of frames for each participant. Participants only correct poses on valid frames since invalid frames often correspond to outliers such as missing or heavily occluded video subjects. Accuracy metrics are also only computed at valid frames, following the convention of the 3DPW benchmark. All the accuracy metrics are under *Protocol 1*, i.e., without using any ground truth during inference [24]. We only used the ground-truth 2D poses in the pre-processing of dataset to filter an identity from the initial poses when there are multiple subjects in a scene (e.g., Fig. 7(c)). The accuracy metrics (MPJPE, PA-MPJPE, and MPVPE) under "Meta Data" are the scores of the initial poses of selected video subjects on valid frames estimated by ROMP [32].

Besides the factors discussed in Section 6, the resulting performance is also highly dependent on the difficulty of corrections. For example, in videos "downstairs 00" and "walkBridge 01", there are long segments where the video subject's legs are hidden due to occlusions and out of video frame boundary. In these videos the hidden legs follow walking movements, and without video frame backgrounds as references, the correction became a traditional editing of walking motion. The participant thus skipped the corrections on the leg movements. In contrast, in videos "bus 00" and "flat packBags 00", the video subject's legs are also hidden due to occlusions. But since in these cases the video subject is in a standing pose the participant could relatively easily fix the errors.

B.2 Qualitative Results

We compare the reconstruction results from *iPose* with state-of-the-art automatic computer vision methods, including three image-based methods (ROMP [32], PyMAF [41], and BEV [33]) and two video-based methods (VIBE [24] and CycleAdapt [28]). We used video footage from the Internet with challenging sports poses. The results are shown in Fig. 10. For automatic pose reconstruction, there is no significant difference in temporal consistency nor regularities in artifacts between image-based and video-based methods. While no automatic computer vision method can reconstruct human poses perfectly, *iPose* can effectively address the visible artifacts.

B.3 Comparison with Alternative Approach

3D body landmark detection is an orthogonal and complementary task with respect to pose reconstruction. Specifically, more accurate landmark detection could provide better handles for users to manipulate poses. To explore the possibility of an alternative approach that first regresses 3D joint positions and then fits a 3D body model to the joint positions with inverse kinematics, we compare the results from *iPose* with a state-of-the-art image-based 3D landmark detection method MediaPipe BlazePose [5]. As shown in Fig. 11, except for the occluded legs Fig. 11(d), BlazePose can accurately locate the joint positions from the video viewpoint. However, it still suffers from depth ambiguity, such as the errors in the elbow pose in Fig. 11(e) and the right knee in Fig. 11(f). The misinterpretation of the video subject's center of gravity in Fig. 11(a) and (f) also makes the poses hard to refine with manual interventions. Thus, there is still a gap in incorporating 3D joint positions in the pipeline.

	Meta Data				**User Intervention Statistics**						
Video name	*#frames*	*MPJPE*	*PA-MPJPE*	*MPVPE*	*time*	*#keyframes*	*#temporal*	*#3D widget*	*MPJPE*	*PA-MPJPE*	*MPVPE*
arguing 00	898	83.95	58.94	118.30	109	151	49	32	69.31	40.17	97.55
bar 00	1,252	96.35	66.00	115.22	61	118	97	4	81.86	55.67	101.87
bus 00	1,066	91.56	56.14	118.52	46	70	113	5	82.62	48.76	112.10
cafe 00	934	79.75	51.77	96.80	55	84	102	24	73.27	44.61	91.98
car 00	735	98.28	57.89	116.90	51	63	87	11	92.45	53.82	111.51
crossStreets 00	508	83.18	57.50	98.24	89	90	62	23	71.29	47.70	88.19
downstairs 00	638	86.74	48.06	93.05	36	72	56	0	81.63	45.20	91.41
enterShop 00	1,385	77.79	41.62	92.73	45	182	161	28	70.05	37.25	89.79
rampAndStairs 00	980	80.12	54.99	93.49	92	216	117	54	69.93	49.60	84.35
runForBus 00	656	93.78	54.65	108.55	73	203	187	52	74.80	46.07	91.84
runForBus 01	692	111.43	65.17	135.11	106	70	101	18	89.83	61.47	113.81
sitOnStairs 00	1,329	94.79	62.71	109.61	95	229	126	63	78.59	56.32	94.71
stairs 00	1,197	85.83	58.12	105.33	26	58	32	0	83.95	57.16	102.73
upstairs 00	825	64.15	42.81	76.96	52	143	62	29	62.43	39.98	74.39
walkBridge 01	1,182	84.51	45.14	100.02	53	74	77	31	82.27	43.09	96.84
walking 00	1,264	83.51	52.43	99.19	45	137	129	38	76.94	46.98	90.42
walkUphill 00	387	73.13	49.68	91.06	67	91	104	29	69.04	45.10	88.65
warmWelcome 00	568	112.87	61.75	131.89	39	78	84	16	103.23	56.15	115.27
weeklyMarket 00	1,055	80.03	46.41	96.12	31	88	59	10	77.74	44.83	93.65
windowShopping 00	1,824	73.95	41.91	84.61	52	103	79	23	67.65	38.13	80.09
flat guitar 01	748	91.61	53.55	106.86	82	86	54	12	88.33	45.16	102.35
flat packBags 00	1,273	75.95	54.33	92.75	40	59	91	6	69.70	50.06	85.68
office phoneCall 00	789	73.41	47.67	97.45	54	65	49	37	64.01	40.86	89.55
outdoors fencing 01	939	93.52	53.27	109.62	32	152	24	5	88.47	46.39	103.50

Table 2: Statistics of the 24 videos in the 3DPW test set. "Meta Data" includes the number of valid frames and accuracy metrics (MPJPE, PA-MPJPE, and MPVPE) before corrections (unit: mm). "User Intervention Statistics" includes editing time (unit: min), the number of frames on which the participant makes corrections ("#keyframes"), the number of times the participant uses the temporal propagation functionality ("#temporal"), the number of times a participant switch to the 3D widget to fix a specific issue ("#3D widget"), and accuracy metrics after corrections (unit: mm).

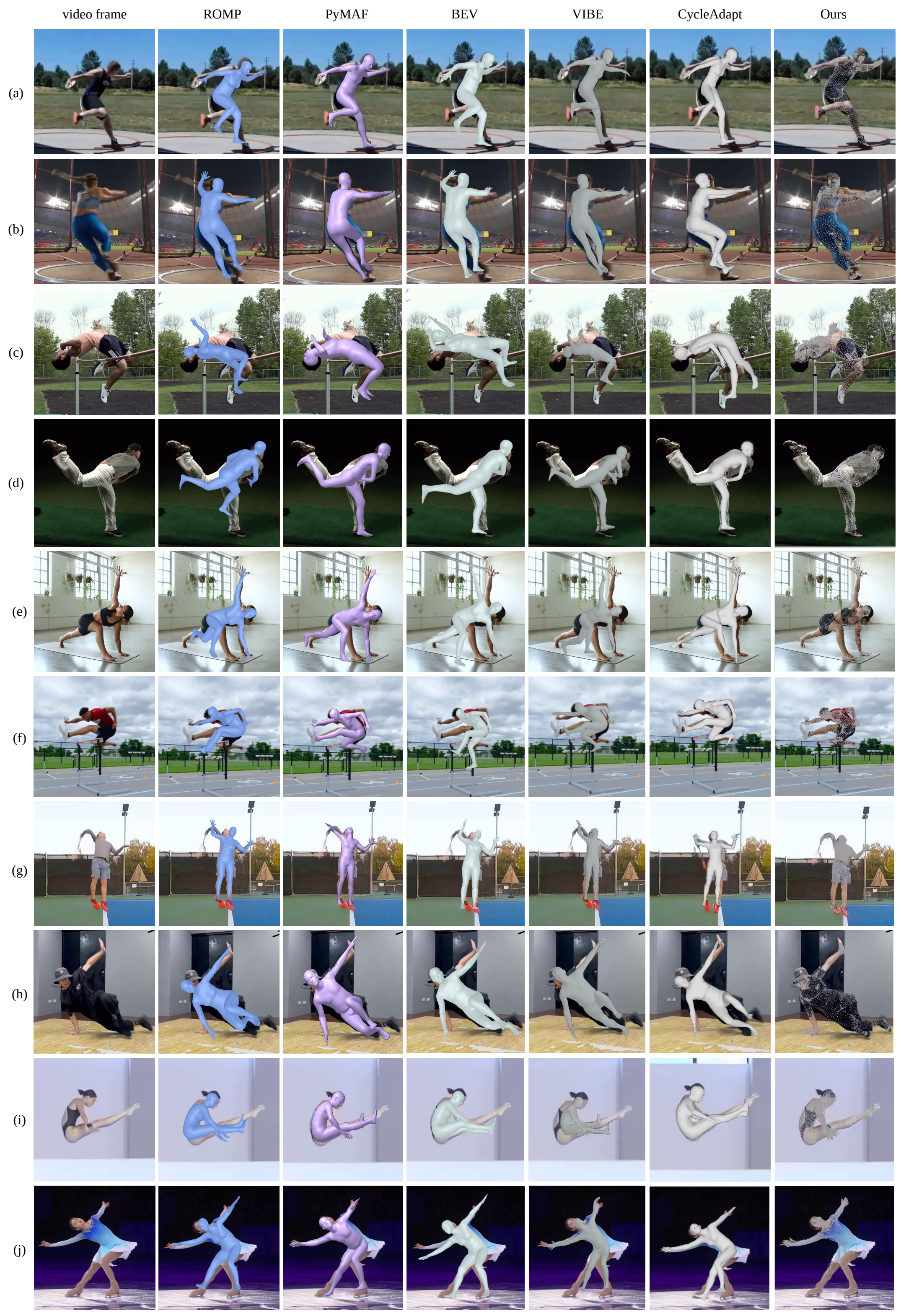

Figure 10: Qualitative comparisons of 3D pose reconstruction results between ROMP [32], PyMAF [41], BEV [33], VIBE [24], CycleAdapt [28], and ours.

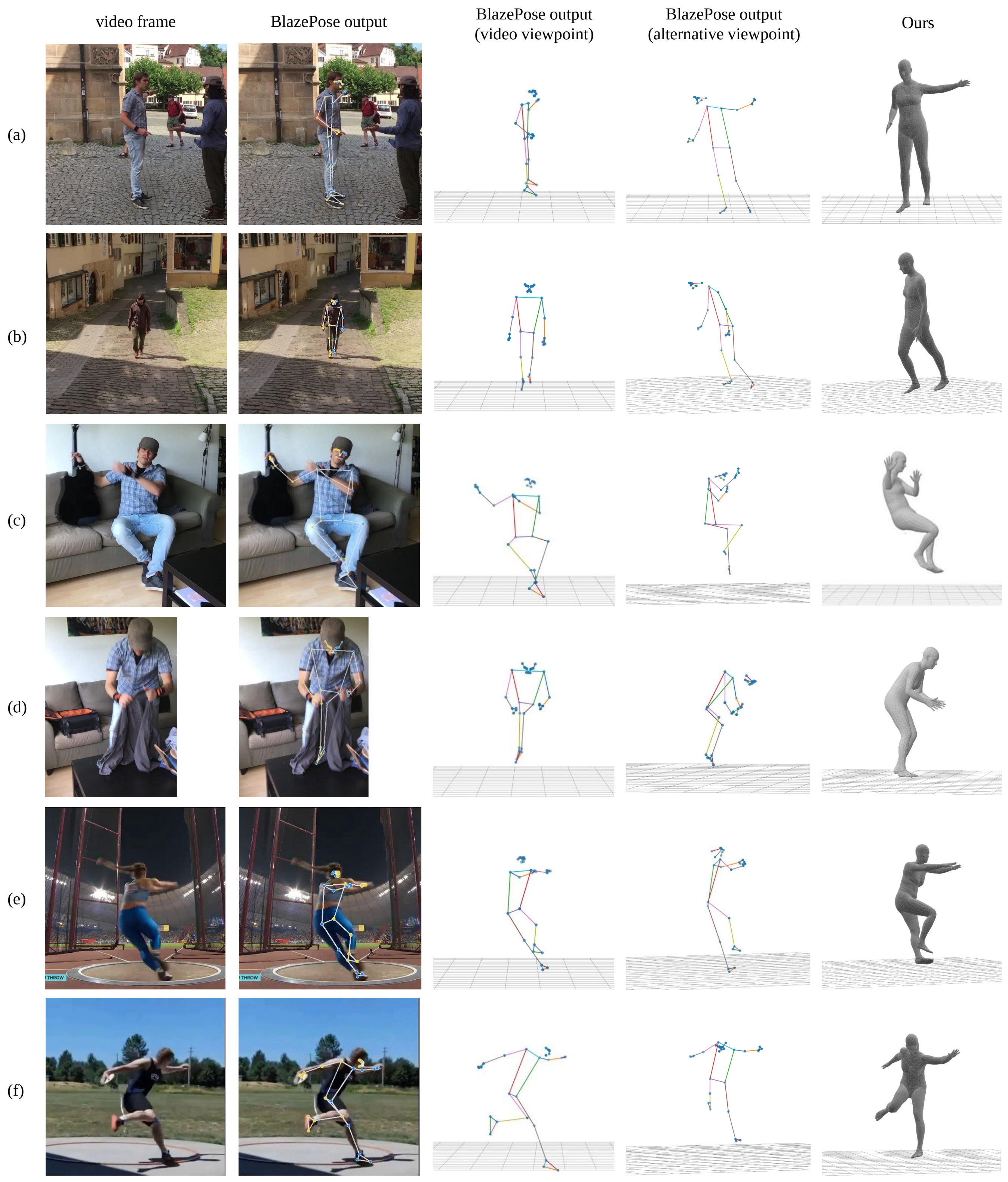
video frame
BlazePose output
BlazePose output (video viewpoint)
BlazePose output (alternative viewpoint)
Ours
(a)
(b)
(c)
(d)
(e)
(f)

Figure 11: Qualitative comparison with 3D body landmark detection from BlazePose [5].