International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2023)
J.-M. Normand, M. Sugimoto, and V. Sundstedt (Editors)

Real-time self-contact retargeting of avatars down to finger level

Mathias DELAHAYE[1], Bruno HERBELIN[1] and Ronan BOULIC[1]

[1]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Figure 1: *Performer pose (left) retargeted onto different avatars*

Abstract

We interact with the world through a body that includes hands and fingers. Likewise, providing an avatar allowing the control of a virtual body with fingers is an important step to improve the user experience in VR. When the user and their avatar skeleton and morphology differ, the direct application of the captured user motion on the avatar's skeleton generally induces self-contact conflicts such as undesired interpenetrations or gaps. Hence it is essential to retarget on-the-fly the user's pose to prevent such mismatches. In this paper, we propose a real-time avatar control with self-contact congruency, including finger mobility. The retargeting approach is evaluated through a subjective evaluation procedure, comparing it against a full-body animation using the user's original movement. Our results show that the retargeted animation approach outperforms the avatar control through the sole captured user movement.

1. Introduction

Our bodies allow us to interact with the environment and to convey semantics through a broad range of poses and gestures. When someone wears an immersive HMD (Head-Mounted Display), the surrounding real world disappears, including the user's physical body, hence requiring an avatar to be introduced for conducting interactions in the virtual environment. Consequently, in a multi-user immersive scenario, failing to translate the original movement onto an avatar might alter the semantics of performed gestures and introduce ambiguities or noise when trying to communicate through gestures. In particular, in [MGB17, BOH*22], the authors highlighted the importance of self-contact in the human perception of poses at the third PV (Person Viewpoint). However, maintaining self-contact consistency on an avatar that was not designed with the morphology of the user (e.g., when the user selects an avatar among a pre-defined list) is not guaranteed when only the raw joint angles are applied to the avatar (e.g., animating an ogre with a large belly when the user is thin). It is, therefore, essential to retarget the user motion for the destination avatar. However, despite the importance of self-contact congruence, many existing approaches proposed in the literature to retarget the user's motion onto the avatar one are either offline or primarily focused on interactions with objects, as reviewed in the next section. Hence, in this article, we propose a retargeting pipeline (section 3) focusing on maintaining self-contact consistency down to the finger level. We assess its ability to convey motion semantics through a subjective evaluation (section 4) and discuss our observations (section 5).

2. Related Works

The process of retargeting (or sometimes remapping) the motion of a performer onto an avatar is covered in the literature through a large panel of techniques [GSC*15,MHLC*22]. Although address-

DOI: 10.2312/egve.20231311

ing the retargeting question, the pioneering work from [Gle98] was offline, and body surfaces were not considered. While online, the approach from [CK00] still presented some instability near singularities, whereas [SLSG01] only considered the end effector locations without a concept of body surface for the retargeting.

[KMA05] introduced a normalized limb representation as a combination of a half-plane containing the three points of its kinematic chain and the root-effector distance normalized by the fully extended limb length. Special care was provided to address the foot contact with a variable floor surface in real-time. A further extension featured a response to external forces applied by the environment [MKHK09]. However, this work focused more on the interaction with the environment rather than preserving the full range of self-contact interactions. Al-Asqhar et al. introduced an approach based on surface descriptors [AAKC13] to tackle the problem of maintaining contact congruency between the skeleton and nearby mesh surfaces (external to the avatar body). Molla et al. combined this approach with elements from [KMA05, MKHK09] to provide a body-independent retargeting animation pipeline that can handle self-contact congruency in real-time [MGB17].

More recently, several new approaches were presented with the new rise of machine learning. In the method from Celikcan et al. [CYC15], equivalent poses between the user and the avatar needed to be calibrated to train the correspondence of poses and the animation of the avatar was performed through a mesh deformation rather than applying rotation on a skeleton. The same also applies to methods relying on skin deformation [JKL18]. This is particularly useful for facial animation or motion retargeting [ZCZ22] when the animation through a skeleton is complex; it broadens the method's applicability on a larger set of non-rigged avatars but drops the internal structure representation. Mesh deformation was also investigated in [BWBM20] to produce convincing avatar poses robust to body shape differences; however, the proposed approach was not real-time compliant, making it impossible to be used in VR for avatar animation. Machine learning was also applied for rigged virtual character motion retargeting with approaches such as the ones from [VYCL18, ALL*20] using neural networks to animate target avatars; however, only the skeleton is considered; hence the difference in morphologies is not fully covered.

Overall, to address the important issue of self-contact congruence between the users and their avatars [BDH*18, BOH*22], an additional process must be put in place to remap the user's motion that not only ensures self-contact consistency (a contact is simultaneously present on the real and the virtual body) but also preserves the relative location of the self contacts on the body as such relative configuration often conveys some semantic information [SLSG01].

Besides, although some approaches addressed interactions with the ground [SLSG01, Gle98, KMA05, MKHK09], with objects [KP16] or self-contact [MGB17], none of these methods appear to address finger-level interactions, which play a crucial role in interacting with the virtual world and to convey a richer palette of meaningful gestures. In this article, we propose integrating the finger animation pipeline from [PDP*19], relying on pre-trained neural networks to predict the hand pose in real-time, with an adapted version of the real-time body animation pipeline from Molla et al. [MGB17] based on a crude approximation of the user's body surface to handle self-contacts. Our approach is specifically designed to use motion distortions at both the limb and finger levels to address the crucial issue of self-contacts consistency in real-time. Then, the contribution from this technique is evaluated against the captured movement, with various avatars presenting different morphologies and sizes.

3. Retargeting animation pipeline

3.1. System overview

Our system relies on a generic motion capture system (here, a combination of Vive trackers and Phasepace) to animate an avatar whose proportions might differ from the user's ones. Compared to the work from [MGB17], where the user's skeleton is calibrated using only gym motion, our skeleton calibration include finger calibration and self-touches to locate more efficiently the different skeleton joint of the user. The resulting data structures are used to determine when contacts are about to occur during the online animation procedure as illustrated in Figure 2.

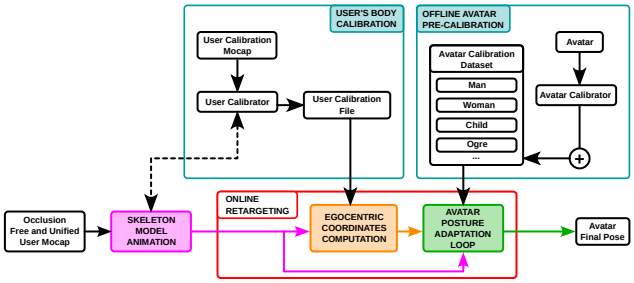


Figure 2: *System overview: the upper stages on the schema represent the user and the avatar calibration. Finally, the online stage computes the instantaneous avatar pose in real time.*

The avatar calibration needs to be performed only once and can be stored in a database. Conversely, the user calibration must be performed each time the user is equipped with the tracking setup, as the trackers are not required to be placed at exact body locations. Given that we also capture the finger motion, we took advantage of the markers present on the user's fingertips to ease the determination of some key joint centers such as elbow or knee as the approach from [MGB17] based on performing calibrating movement to infer the rotation axis showed risks of introducing noise when the movement was not of sufficient rotation magnitude. Likewise, the user hand and finger calibration (c.f. annex) relied on different self-touches to calibrate finger joint locations, widths, or lengths.

Once both the skeleton and body simplified shapes are calibrated, the live performance phase is composed of three steps: the motion capture (to animate the model of the user's structure), the computation of egocentric normalization (to account for a normalized representation of the user pose), and finally, a gradual pose adaptation of the normalized motion onto the avatar character.

The computation of egocentric normalization involves computing the coordinates of key positions (further referred to as "target points") by expressing each position as the sum of the normalized contribution of vectors toward each surface element (constituting

the user's simplified body shape, section 3.2.1). Finally, an adaptation loop progressively attracts each avatar's target points (e.g., effectors, joints) towards its retro-projected egocentric coordinates on the virtual character (section 3.3) to produce the final avatar's pose. Compared to [MGB17], considering the fingers' key positions (effectors and joint centers) in the same way as the other body key positions would result in a too strong influence of the former within the posture adaptation. Hence, we also address how to distinguish and integrate their influence in the posture adaptation stage.

3.2. Online Retargeting

The animation of the internal skeleton model uses traditional IK coupled with specific animation techniques for the hands, including integrating biomechanical constraints.

3.2.1. User Egocentric Coordinates Computation

3.2.1.1. Notations The body's coarse surface structure, including fingers, supports the computation of egocentric coordinates. It consists of 65 surface elements: 26 Triangles (2 per hand and foot, 4 for the face, and 14 for the trunk), 38 Capsules Bones (3 per finger and two per limb), and one sphere for the head. We note the set of surface elements: $(s_i)_{i\in\mathbb{S}}$

The structure also comprises 28 target points (p_j, Figure 3) attached to the skeleton's structure: three per limb effector (hands and feet), one per intermediate limb joint (elbow, knee), and one per fingertip. Those targets are uniquely indexed in a set we note $\mathbb{T}$. $(p_j)_{j\in\mathbb{T}}$ is the notation of the set of targets.

Unless specified differently, i refers to the surface element index and j to the target point's index.

3.2.1.2. Egocentric Coordinates Components A self-contact occurs when the distance between two surface elements becomes null. However, it is equally important to encode the full body posture space to ensure a continuous mapping between contact and non-contact poses. For this reason, we normalize the body pose by computing the relative distance between the coarse surface approximation elements (s_i) and the target points (p_j) defined above. This representation is called Egocentric Coordinates [MGB17].

In such a representation, the distances and directions (i.e., vectors) between a target point p_j and its projection x_i on the surface s_i are noted $\overrightarrow{v_{i,j}}$ (Figure 3 left).

To account for avatars with different sizes (e.g., bones can be longer), x_i are represented in a normalized form, all noted $\hat{x}_i$, regardless of the surface element (barycentric coordinates for triangles, cylindrical coordinates for cylinders and spherical coordinate for the sphere). Furthermore, the contribution of bones on the contributing vector $\overrightarrow{v_{i,j}}$ are also considered with the length of the bones that contribute to the kinematic chain. This overall contribution of the kinematic chain on $\overrightarrow{v_{i,j}}$ is noted $\tau_{i,j}$ (see [MGB17] for its computation).

Finally, to prioritize self-contacts in the pose reconstruction, each contributing vector $\overrightarrow{v_{i,j}}$ is weighted according to its relevance to a self-contact. For instance, if the contribution indicates that the target point is close to a surface element, its weight would be high.

Figure 3: *Decomposition of a target point's position p_j into a surface contact point and a vector (left) re-applied on the target avatar (right).*

We note $\lambda_{i,j}$ such a weight for the importance of the vector $\overrightarrow{v_{i,j}}$ in the reprojection process detailed in paragraph 3.3.2.1.

With such a system of coordinates, the opposite operation to compute the retro-projected target point p'_j is done through Equation 1 with $x'_{i,j}$ the transposition of $\hat{x}_{i,j}$ on the avatar's surface element s'_i, and $\tau'_{i,j}$ the normalization factor computed on the avatars' kinematic chain (Figure 3 right).

$$p'_j = \sum_{i\in\mathbb{S}} \left(x'_{i,j} + \overrightarrow{v_{i,j}} \cdot \frac{\tau'_{i,j}}{\tau_{i,j}} \right) \cdot \lambda_{i,j} \tag{1}$$

In addition to the user's surface, the distance toward the floor is also considered. Here, only the contribution from the vertical component h_j is used.

3.2.2. Computing contribution weights

For each contribution vector $\overrightarrow{v_{i,j}}$, we defined a contribution weight ($\lambda_{i,j}$) such that the retro-projection of the point is performed according to Equation 1. To compute this relative contribution of each vector, we first compute a set of raw weights $\Lambda_{i,j}$ that only relies on the surface element and the target to be computed. Then, a normalization process described below yields the $\lambda_{i,j}$.

3.2.2.1. Removing trivial contributions Prior to computing the weight of the contributions, we already know that the hand, for instance, is always located close to the forearm's extremity. Therefore the importance of the associated contribution vector should be small, not to erase the contribution from other vectors in the weights normalization process. Consequently, the weights from the same limb on which a target p_j is attached are always set to zero, and their associated computations are skipped in the rest of the pipeline.

3.2.2.2. Addressing the integration of fingers' contribution The integration of fingers into a posture normalization scheme has to consider three important properties of the additional set of key positions that are considered compared to the normalization of a fingerless skeleton. First, the total number of key points raises from 18 to 28 when integrating the fingers tips; second the added key

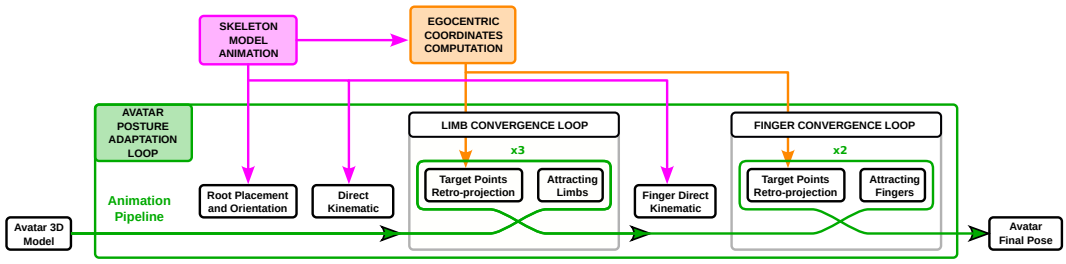


Figure 4: *The posture adaptation pipeline is applied for each frame. It resets the placement of the avatar and pre-orient limbs using the raw captured pose. Limbs are then progressively attracted toward their retro-projected target points (p'_j). Fingers are then initialized using the captured pose and are also progressively attracted toward their retro-projected target points to produce the final avatar pose.*

points are densely localized in only two regions of the skeleton, and third their higher density accounts for a type of activity occurring at a smaller scale than the activities accomplished at the full-body scale. So two sets of weights were computed to prevent artifacts due to an imbalance of the finger tips' contribution in the reprojection process of other target points: one set with the contribution of the fingers' 30 capsule elements set to zero (i.e., as if the finger knuckles were not integrated), and one set with the full contribution of all surface elements to address the fingers' self-contacts.

In both cases, the raw weights $\Lambda_{i,j}$ are normalized into normalized weights ($\lambda_{i,j}$, with $\sum_{i\in\mathbb{S}}\lambda_{i,j}=1$) such that the reprojection yields the targeted p'_j position.

3.3. Avatar Posture Adaptation Loop

The computation of the avatar's pose from the set of egocentric coordinates is composed of several stages illustrated in Figure 4. The rationale for integrating the finger retargeting within the posture adaptation loop is to generalize the "coarse-to-fine" strategy already present in [MGB17] by adding a final finger convergence loop after the 1) trunk initialization and 2) the limb convergence loop.

3.3.1. Avatar Pose Initialization

In this process, the root of the avatar's skeleton is placed at the scaled height of the user's root. With h_{c_u} the height of the user's sacrum calibrated when standing up, h_{a_c} the height of the avatar's sacrum when standing up, the location of the avatar's sacrum is initialized at the height h_a as defined based on the current height of the user's skeleton h_u in Equation 2.

$$h_a = \frac{h_{a_c}}{h_{u_c}} \cdot h_c \tag{2}$$

Once the height is adjusted, the avatar's sacrum is aligned with the user's sacrum, followed by the animation of the spine to account for the twist of the source skeleton. With the trunk placed and animated, the limbs are animated by orienting the anchor bones first to match the orientation of the source skeleton's structure. This process then continues toward the limbs' extremities with the orientation of intermediates bones and, finally, effectors' orientation. The head is also oriented to match the original head orientation.

This constitutes the baseline for the progressive attraction of effectors toward their expected reprojection location.

3.3.2. Limb Animation Convergence Loop

The second stage of the "coarse-to-fine" retargeting strategy is the progressive attraction of effectors' positions towards their final position. It is performed by iteratively updating the surface elements of the avatar and computing the retro-projection of the position of the target points from the current pose (paragraph 3.3.2.1). Limbs are then attracted towards the retro projections, and the new effectors' orientation (paragraph 3.3.2.2) are computed.

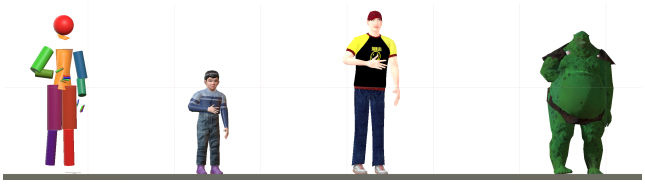
To handle the difference in motion range, this process is performed through two convergence loops, one handling first the limb level and the second one the fingers level, as those are at the extremity of the kinematic chain.

3.3.2.1. Target Point Reprojection Based on the current avatar pose, the calibrated surface elements are updated to match the avatar's skeleton structure. Once the surface elements are placed, the formula from Equation 1 is applied to determine the set of reprojected target points $\left(p'_j\right)_{j\in\mathbb{T}}$ onto the avatar. Once the set of targets positions $\left(p'_j\right)_{j\in\mathbb{T}}$ is computed, target positions $\left(a_j\right)_{j\in\mathbb{T}}$ are progressively attracted towards their reprojected points [MGB17].

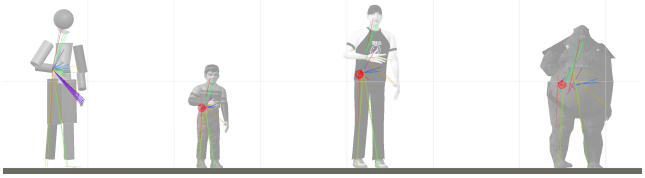
3.3.2.2. Effector Position and Orientations Once the complete set of targets' positions for the avatar $\left(a_j\right)_{j\in\mathbb{T}}$ is computed, the effectors' positions (i.e., wrist and ankles) are then extracted from the rigid body composed of the three reprojected target points from the hand's palms or the feet.

When the effector is far from any surface (i.e., 10cm, corresponding to an average hand's width [NAS95], in future works, this would be the size of the avatar's hand), the avatar's effector orientation is the one from the user's skeleton animation. Then, when the effector gets closer to a surface, its orientation is progressively imposed by the three markers defining a rigid body to ensure the contact surface does not interpenetrate the body.

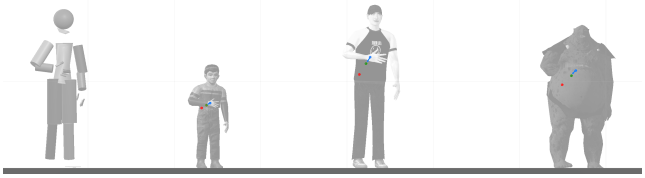
For this reason, the avatar effector orientation is computed as the weighted average between the user effector orientation and the one defined by the three retro-projected target points. The ratio selection uses the maximum weight of the lambdas computed



(a) *Raw angle application from the source skeleton onto the avatar*



(b) *Contribution vectors applied onto the avatar surface elements with the computed target position p'_j illustrated with the red sphere.*



(c) *Attraction process illustrated for the first of the three limb passes. In red, the retro-projected target position p'_j, in blue, the source position a_j before the attraction, and in blue, the same target position a_j after the attraction.*

(d) *Attraction process after the three and two passes for limb and fingers retargeting convergence loops.*

Figure 5: *Illustration of the computation of the contribution vectors and their normalized re-application on the targeted avatar to compute the avatar's target position.*

in section 3.2.2, and the rotation average is performed using the method from [MCCO07], with the selection ratio varying from 0 (the effector orientation is the user's skeleton effector orientation) to 1 (the orientation is entirely defined by the three retro-projected attracted targets). The value chosen for this selection ratio is $\max\left(\left(\lambda_{i,j}\right)_{(i,j)\in\mathbb{S}\times\{a,b,c\}}\right)$, with $(a,b,c)\in\mathbb{T}^3$ the indexes of the three target points attached to the effector, as $\lambda_{i,j}$ represents the importance of the link, and subsequently, relates also the importance of the contribution vector to the orientation.

3.3.3. Finger Animation Convergence Loop

The last stage of the "coarse-to-fine" retargeting strategy is the finger animation pipeline. This stage uses inputs from both the user's body skeleton and the fingers themselves. Each finger possesses its animation pipeline composed of an inverse kinematic with the flexion determined by the effector base joint distance and a realignment based on the yaw-pitch of the target points (p_j). To maximize the similarity between the real finger's pose and the avatar's one, the avatar finger animation is directly transferred from the user's skeleton hand animation when there is no close collision. Otherwise, its animation is progressively driven by the avatar's fingers' IK fed with the retro-projected fingertips input using the greatest $\lambda_{i,j}$ for j the index of the finger's target point as the selection ratio. The value chosen for this selection ratio is $\max\left(\left(\lambda_{i,j}\right)_{j\in\mathbb{S}}\right)$ for j the fingertip's target's index. This finalizes the pose reconstruction of the avatars.

4. Subjective evaluation of the full body animation

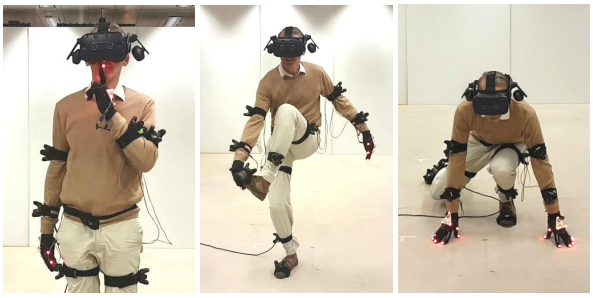
Our movements convey semantics and meanings that are easily perceived by others. Failing to remap the user's movement onto the avatar might lead to failure at the interaction and convey the wrong semantics to the other. In particular, it was shown that the self contact consistency was an important factor in producing a convincing avatar animation ([BOH*22]). Therefore, it is essential to know whether our approach conveys the correct intended semantics.

The long setup required to have a participant calibrated, combined with the variability of morphologies and posture interpretations, made it impractical to have a replicable evaluation involving subjects in a real-time first PV evaluation. However, thanks to the coarse approximation and hardware acceleration, this approach could be used in real-time with a subject facing a mirror looking at himself in an avatar's body (c.f., video).

Consequently, we performed an evaluation at the 3rd PV and compared our approach to a raw captured motion (further labeled as "direct kinematics") through a subjective evaluation. This evaluation consisted of showing three videos to naive observers: one showing a recorded source pose through a camera (always on the left) and two recorded videos using each approach with a similar viewpoint (placed in the middle and on the right, randomly swapped), and then asking the participants to evaluate both animations. This setup provides a 3rd PV where the avatars are seen by another user (here, the subject through a monitor).

4.1. Video dataset

To construct the database used in the comparison, we asked two persons to be equipped with the tracking system, to perform their body calibration, and then to be recorded while performing movements to reach predefined poses. Those poses were chosen to carry semantics (e.g., placing the finger in front of the mouth, placing the hand near the ear, placing the hand in front of the mouth to express surprise Figure 6a), poses with self-contacts (e.g., both hands touching each other, hand foot contact Figure 6b), or poses with contacts with the floor (e.g., crouching with one hand on the floor Figure 6c). The two recorded persons were a man and a woman, on the thinner side of what can be considered regularly shaped. We applied the recorded MoCap using both methods on four avatars: a tall man, a woman, a small child, and an ogre with a large belly. Each of the movements was then presented, at its regular speed, four times to the participant, not necessarily in consecutive order, each time with a different avatar, hence producing a total of 152 individual motion clips. The animation was rendered using an orthographic projection. We framed the view of the camera so that



(a) *Shussing pose* **(b)** *Hand-foot contact* **(c)** *Floor contact*

Figure 6: *Example of semantic poses conveying different types of interactions*

the viewing direction is the same, with the performer at the center of the captured zone.

4.2. Subjective evaluation procedure

Before participating in this study, participants were informed about the task of the study and were asked to give their written informed consent and fill out an anonymous demographic questionnaire. Participants were seated in front of a 57-inch monitor and presented with an interface instructing them to carefully analyze two animation motion clips displayed side by side. Their task was to "Carefully analyzing each animated motion clip and decide with the sliders how faithfully it replicates the performed pose and action in the provided video clip". The retargeted clip and the non-retargeted one were randomly swapped, the continuous sliders were used to collect the evaluation scores, and the models used are listed in Table 1. At the end of the experiment, we asked participants to report which criteria they used, by order of priority, to evaluate the proposed motion clips and received their overall feedback from the experiment. Ultimately, participants collected their financial compensation for their time of *Anonymized*.

4.3. Analysis

To verify the presence of an effect linked to the animation method factor, we performed a pairwise comparison between both samples: the retargeted approach and the direct kinematic one. The analysis pipeline was performed on both the full dataset as a whole and on targeted avatar subsets (152 motion clips). The test used for the comparison was the parametric pair-wised two-sample two-sided Student's t-Test, which evaluates the null hypothesis of the equality of the two means of the samples. Therefore, before applying the test, we verified the required hypothesis for this test to be performed. We first verified that each sample was normally distributed using the Shapiro tests, hence allowing parametric methods to be used, and assessed the homogeneity of the variances through the parametric Hartley's Maximum F-Ratio test. Finally, a test on the direction of the difference was performed post-hoc using the one-sided version of the t-test, and the effect sizes were measured using Cohen's D.

4.4. Results

4.4.1. Demographics

In total, 20 participants (15 women), aged between 20 and 30 years old (average: 21.7, std: 2.43), participated in this study. Those participants were mostly not used to playing video games (75% reported never or rarely playing action video games). Participants often practiced sports such as dance and were in the large majority (17 out of 20) right-handed. On average, the experiment took one hour to be completed.

4.4.2. Evaluation scores

The results from the Shapiro tests indicate that all samples were normally distributed, allowing parametric statistical tests to be performed. The homogeneity test did not reveal any significant difference in the variance distribution across the different paired samples between the retargeted approach and the direct kinematic animation. Therefore t-test comparisons were performed, and the effect sizes were measured using Cohen's D formula. Due to the observed significant differences, the post-hoc analysis was conducted. All of the test values and effect size measurements are reported in Table 1 and illustrated in Figure 7.

Here, we can observe that, among all situations, the scores given by the participants were on the upper side of the range and, in all situations except the ground interaction, the scores for the approach with the retargeting were higher for all the movement clips and across all of the destination avatars.

5. Discussion and future works

5.1. Limitations

5.1.1. Instabilities and Footskate

One of the primary issues observed in the animation pipeline is the instability of the resulting animation. In particular, we observed that the legs were particularly affected, constituting the main highly visible drawback of the current version of the approach. Here, only the vertical height of the feet is constrained; therefore, nothing prevents the characters' feet from sliding on the floor surface. Furthermore, each frame is generated independently of the previous one and only relies on the consistency of transformed input from the MoCap systems for the temporal consistency of the avatar's motion. A low-pass filtering could be introduced for each iteration of retro-projected target points to enforce the animation's stability. Limiting the displacement of each retro-projected target point in regard to the source target's motion would be another possible option to reduce this issue. Concerning the specificity of leg movements, solutions such as the one proposed in [KMA05, LMT07, MHCH22, PYAP22] should be used to increase the overall naturalness of the pose, albeit it permits an offset to exist between the user and the avatar location, as such a difference could easily go unnoticed if the gain in displacement remains reasonable [SBJ*10].

5.1.2. Avatar simplified mesh resolution

The small density of the crude mesh used to animate the avatar can exhibit some slight inter-penetrations, especially for curved surfaces. Increasing the number of triangles would reduce this but induce a higher computational cost. Therefore, addressing this issue

Table 1: *Test and measure results for the scores of the evaluations between the Direct Kinematic and the Retargeting evaluation scores*

Test / Measure	All	Ogre	Child	Man	Woman	Semantics	Self-Contact	Ground
Hartley's maximum F-ratio	0.92	1.00	1.00	1.00	1.00	1.00	0.99	1.00
t-Test (two-sided)	$2.98 \cdot 10^{-25}$	$1.30 \cdot 10^{-12}$	$3.73 \cdot 10^{-11}$	$1.01 \cdot 10^{-2}$	$1.75 \cdot 10^{-6}$	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	$2.28 \cdot 10^{-11}$
t-Test (greater)	$1.49 \cdot 10^{-25}$	$6.50 \cdot 10^{-12}$	$1.87 \cdot 10^{-11}$	$5.07 \cdot 10^{-3}$	$8.75 \cdot 10^{-7}$	$< 2.2 \cdot 10^{-16}$	$< 2.2 \cdot 10^{-16}$	1.00
Cohen's D	0.27	0.37	0.34	0.13	0.25	0.38	0.31	0.63

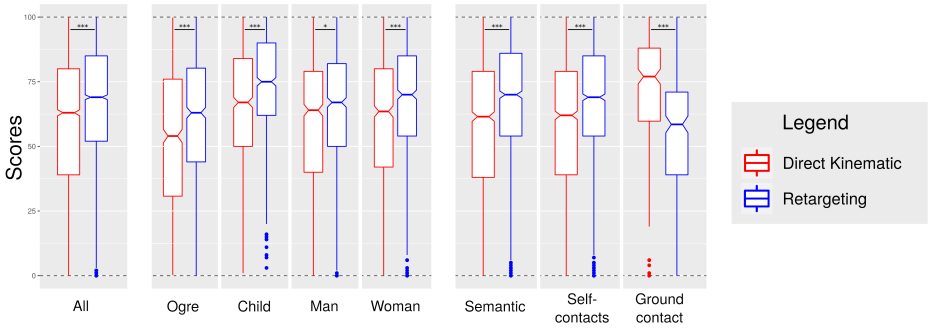


Figure 7: *Normalized boxplots of the participants' evaluation scores.*

through the use of other types of surface elements (e.g., ellipsoid for the belly) might be beneficial in terms of performance.

5.1.3. Setup and calibration

Equipping a user and performing the calibration still requires around 20 minutes despite the semi-automated process. Furthermore, four of the trackers, providing redundant information once the calibration was performed (those placed on forearms and lower legs), could be removed to lighten the setup for the users.

5.2. Subjective evaluation

The subjective evaluation showed that, except for the ground interaction, our retargeting approach was significantly preferred over the direct kinematic animation pipeline. Consequently, and despite the modest effect size, given the limitations of our approach in terms of smoothness, we can expect the observed preference towards our retargeted approach to be driven by the ability of the system to maintain the self-contact consistency known to be a critical point for the animation of a 3D character [BDH*18, BOH*22].

It is also to be noted that the sample population was possibly biased due to their trained body representations and kinematics (e.g., in dances, foot placements are greatly important for balance, and slight movement gaps influence the overall figure), making it likely that the population was better than a more representative sample, at spotting for both foot motion or self-contacts inconsistencies.

Additionally, we can notice a link between the measured effect size (Table 1) and the discrepancy between the performer and the avatar's body: when both the avatar and the user share more or less the same morphology and proportions, the contribution of the retargeting might become less relevant (small effect size for the man and the woman avatars), and those conditions can therefore act as a kind of control condition. However, even in this case, the retargeted approach was preferred over the direct kinematic one, hence suggesting that simply relying on the raw captured motion may not be adequate despite the similarity between the avatar and the user. Conversely, when the avatar's body differed more (for the child or the ogre), the observed effect size was higher.

Finally, this study addresses the evaluation of the animations using a third PV to provide participants with the maximum amount of details on the animation. Future work should be conducted within a virtual environment where multiple users, or a single user with an experimenter confederate, would assess both their own avatar movement in a virtual mirror and also the other avatar's movement quality through well-defined interaction scenarios.

6. Conclusion

In summary, we proposed an animation method, extending the work from [MGB17], taking advantage of human tolerance to motion discrepancies to address the issue of interpenetrations in the animation of an avatar now with both body and finger levels. With pre-calibrated avatars, our approach only requires retrieving the user's skeleton structure and body shape through a calibration process. Once the user's model is calibrated, the posture is stored in a normalized form of joint angles and normalized relative vectors between target points and the body surface. This normalized form is then used to iteratively attract the avatar's posture toward the applied normalized posture on the avatar's structure at the limb level to provide a posture that respects self-contact consistency at the body level, followed by a second iteration loop dealing with the convergence of fingers pose. The repository of the application will be made publicly available upon publication. We conducted a

subjective assessment to compare our retargeting method with direct kinematics in generating believable avatar animations based on the movements of the source character. Participants were asked to evaluate motion clips produced using both approaches, and we performed a statistical analysis of the gathered scores. The results revealed that our approach significantly enhanced the perceived overall quality of the animations compared to direct forward kinematics. The participant feedback emphasized the role of smoothness in the animation, in particular for leg movements, which was not sufficiently addressed in our approach, hence providing an opportunity for improvement. Ultimately, this user evaluation sets the stage for conducting an immersive first PV user evaluation in the future.

References

[AAKC13] AL-ASQHAR R. A., KOMURA T., CHOI M. G.: Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2013), SCA '13, ACM, pp. 45–53. doi:10.1145/2485895.2485905. 2

[ALL*20] ABERMAN K., LI P., LISCHINSKI D., SORKINE-HORNUNG O., COHEN-OR D., CHEN B.: Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 62–1. 2

[BDH*18] BOVET S., DEBARBA H. G., HERBELIN B., MOLLA E., BOULIC R.: The critical role of self-contact for embodiment in virtual reality. *IEEE Transactions on Visualization and Computer Graphics 24*, 4 (2018), 1428–1436. doi:10.1109/TVCG.2018.2794658. 2, 7

[BOH*22] BASSET J., OUANNAS B., HOYET L., MULTON F., WUHRER S.: Impact of self-contacts on perceived pose equivalences. In *Proceedings of the 15th ACM SIGGRAPH Conference on Motion, Interaction and Games* (New York, NY, USA, 2022), MIG '22, Association for Computing Machinery, pp. 1–10. doi:10.1145/3561975.3562946. 1, 2, 5, 7

[BWBM20] BASSET J., WUHRER S., BOYER E., MULTON F.: Contact preserving shape transfer: Retargeting motion from one shape to another. *Computers & Graphics 89* (2020), 11–23. doi:https://doi.org/10.1016/j.cag.2020.04.002. 2

[CK00] CHOI K.-J., KO H.-S.: Online motion retargetting. *The Journal of Visualization and Computer Animation 11*, 5 (2000), 223–235. doi:10.1002/1099-1778(200012)11:5<223::AID-VIS236>3.0.CO;2-5. 2

[CYC15] CELIKCAN U., YAZ I. O., CAPIN T.: Example-based retargeting of human motion to arbitrary mesh models. *Computer Graphics Forum 34*, 1 (2015), 216–227. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12507, doi:10.1111/cgf.12507. 2

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 33–42. doi:10.1145/280814.280820. 2

[GSC*15] GUO S., SOUTHERN R., CHANG J., GREER D., ZHANG J. J.: Adaptive motion synthesis for virtual characters: a survey. *The Visual Computer 31* (2015), 497–512. 1

[JKL18] JIN T., KIM M., LEE S.-H.: Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 311–320. 2

[KMA05] KULPA R., MULTON F., ARNALDI B.: Morphology-independent representation of motions for interactive human-like animation. In *Eurographics* (2005). 2, 6

[KP16] KIM J.-S., PARK J.-M.: Direct and realistic handover of a virtual object. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), IEEE, pp. 994–999. 2

[LMT07] LYARD E., MAGNENAT-THALMANN N.: A simple footskate removal method for virtual reality applications. *The Visual Computer 23* (2007), 689–695. 6

[MCCO07] MARKLEY F. L., CHENG Y., CRASSIDIS J. L., OSHMAN Y.: Quaternion averaging. *Journal of Guidance, Control, and Dynamics* (2007). 5

[MGB17] MOLLA E., GALVAN DEBARBA H., BOULIC R.: Egocentric mapping of body surface constraints. *IEEE Transactions on Visualization and Computer Graphics* (2017), 1–1. doi:10.1109/TVCG.2017.2708083. 1, 2, 3, 4, 7

[MHCH22] MOUROT L., HOYET L., CLERC F. L., HELLIER P.: Underpressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. *arXiv preprint arXiv:2208.04598* (2022). 6

[MHLC*22] MOUROT L., HOYET L., LE CLERC F., SCHNITZLER F., HELLIER P.: A survey on deep learning for skeleton-based human animation. In *Computer Graphics Forum* (2022), Wiley Online Library, pp. 122–157. 1

[MKHK09] MULTON F., KULPA R., HOYET L., KOMURA T.: Interactive animation of virtual humans based on motion capture data. *Computer Animation and Virtual Worlds 20*, 5-6 (2009), 491–500. 2

[NAS95] NASA V. I.: Man-systems integration standards. *Revision B. Section 4* (1995). 4

[PDP*19] PAVLLO D., DELAHAYE M., PORSSUT T., HERBELIN B., BOULIC R.: Real-time neural network prediction for handling two-hands mutual occlusions. *Computers & Graphics: X 2* (2019), 100011. doi:https://doi.org/10.1016/j.cagx.2019.100011. 2

[PYAP22] PONTON J. L., YUN H., ANDUJAR C., PELECHANO N.: Combining motion matching and orientation prediction to animate avatars for consumer-grade vr devices. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 107–118. 6

[SBJ*10] STEINICKE F., BRUDER G., JERALD J., FRENZ H., LAPPE M.: Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics 16*, 1 (2010), 17–27. doi:10.1109/TVCG.2009.62. 6

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Trans. Graph. 20*, 2 (Apr. 2001), 67–94. doi:10.1145/502122.502123. 2

[VYCL18] VILLEGAS R., YANG J., CEYLAN D., LEE H.: Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 8639–8648. 2

[ZCZ22] ZHANG J., CHEN K., ZHENG J.: Facial expression retargeting from human to avatar made easy. *IEEE Transactions on Visualization and Computer Graphics 28*, 2 (2022), 1274–1287. doi:10.1109/TVCG.2020.3013876. 2