FingerPuppet: Finger-Walking Performance-based Puppetry for Human Avatar

Ching-Wen Hung
National Taiwan University
Taipei, Taiwan
ching-wenh@sigchi.org

Chung-Han Liang
National Taiwan University
Taipei, Taiwan
r09922a02@ntu.edu.tw

Bing-Yu Chen
National Taiwan University
Taipei, Taiwan
robin@ntu.edu.tw

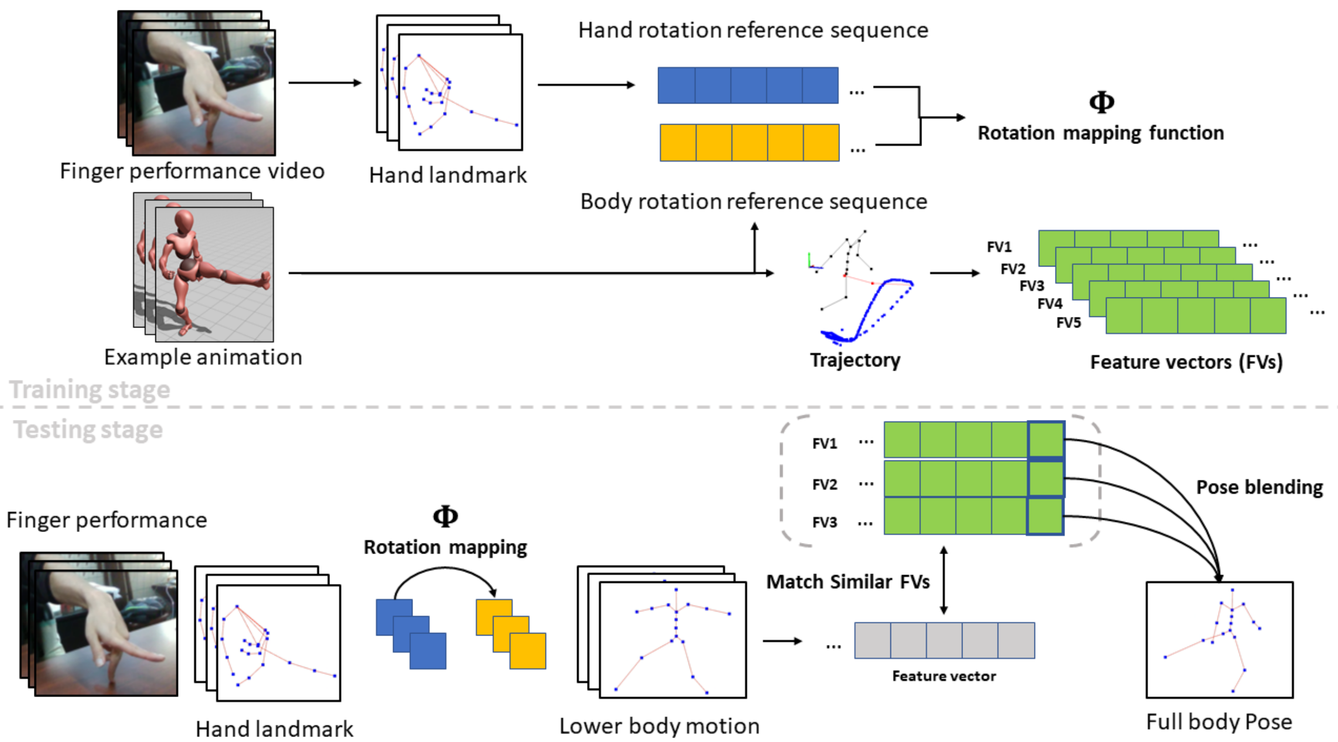

Figure 1: Illustration of the proposed method for finger-walking-based puppetry. A rotation mapping function and multiple feature vectors are constructed during the training stage. In the testing stage, the mapping function maps finger rotation to lower body rotation. Then, feature vectors generated from the rotation-mapped lower body motion and the training stage are matched to blend a final full-body pose.

ABSTRACT

Performance-based digital puppetry has gained widespread popularity in various fields, including gaming, storytelling, animation editing, etc. Human hands are well-suited for manipulating digital avatars with their skill and ability to perform multiple movements. In this work, we adopted the finger-walking technique, a natural and intuitive method of performance, as an interface for controlling human avatars. We first conducted a preliminary study to explore the range of finger-walking movements preferred by casual users and identified several general types of finger-walking performances. Based on the study results, we selected five common example animations from a database suitable for finger-walking performance. To manipulate human avatars, we developed a method that maps finger-walking motions to leg motions using rotation mapping and matches similar leg motions in the example animations to generate expressive full-body motions. We also implemented a prototype interactive storytelling application to demonstrate the effectiveness of our system in developing responsive and reliable human avatar motions.

CHI EA '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0331-7/24/05
https://doi.org/10.1145/3613905.3650840

CCS CONCEPTS

• **Human-centered computing → Gestural input**.

KEYWORDS

finger-walking, motion retargeting, performance-based input

ACM Reference Format:

Ching-Wen Hung, Chung-Han Liang, and Bing-Yu Chen. 2024. FingerPuppet: Finger-Walking Performance-based Puppetry for Human Avatar. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3613905.3650840

1 INTRODUCTION

Computer puppetry, or digital puppetry, is a rising research area, and its technique has been adopted in various applications, such as gaming, storytelling, animation editing, etc. Keyboard and mouse are well-known conventional interfaces for digital puppetry in animation editing. A joystick with buttons and an analog stick are common interfaces for interacting with digital avatars in gaming. With the improvement and increased affordability of tracking devices, many low-cost, commodity devices such as Microsoft Kinect, Leap Motion, Nintendo Wii, and Nintendo Switch utilize the human body's motion as an interface for interacting with virtual environments, have become available. Moreover, significant research has been done into novel interaction interfaces for animation editing and storytelling, including those that use tangible or user specified input devices [3, 4].

Performance-based interfaces, which utilize full-body or partial-body movements to achieve impressive character control, have gained popularity in virtual avatar manipulation. This is due to the greater relevance between the user's and the avatar's movements. Besides that, the increasing popularity of virtual reality (VR) and augmented reality (AR) devices and applications have enabled users to experience the movements of a 3D avatar in an immersive virtual world. The embodiment of the interaction interface is critical to the illusion of controlling the avatar in the virtual environment, as typical interfaces such as mouse, keyboard, and joystick cannot provide an intuitive control, potentially disrupting the illusion.

The hand performance interface is one type of performance-based interface, which offers the advantage of requiring minimal performance space, little physical effort, and no additional particular device compared to hand-held device interfaces while preserving a partial sense of embodiment compared to full-body performance interfaces. Finger-walking, a type of hand performance, is a naturally chosen and widely accepted performance scheme used as a character manipulation interface. According to a user study in [9], participants used the middle and index fingers of their dominant hand to mimic the leg movement of the corresponding full-body motion. The finger-walking performance was found to be an intuitive way of manipulating human avatars, with the embodiment being fulfilled through the relevance of the movement of the fingers to the leg movement.

Previous work explored the hand performance interface, such as proposing customized hand gestures for wearable devices [13] and augmented reality [6, 12] to improve system performance. PuppetPhone [2] also utilized smartphones as a virtual character control interface, providing a real-time interactive experience. Specifically, they proposed MotionStick, which offers a more intuitive control mechanism allowing users to control the virtual character at a certain distance in front of the phone's camera. However, users employing this mechanism indirectly control the character, as the relevance between the user's gesture and character motion is low, leading to a decreased sense of embodiment. Hand Interface [10] explored various hand interaction designs, and HandAvatar [7] focused on hand interfaces embodying non-human avatars. Still, they mainly focused on objects, tools, or animal control instead of human avatar manipulation. User-defined gestures for character manipulation are explored in Puppeteer [5]. The authors propose a control interface that utilizes finger joint features to recognize the user's hand gesture, which aids in selecting the appropriate character animation to display. However, their method requires the user to complete the entire hand gesture before the animation can be shown, which reduces the manipulation experience and introduces asynchrony between the user's gesture and character motion. The real-time finger-walking interface of expressive human avatar control still needs to be explored.

To achieve finger-walking performance in character manipulation tasks, a method that can transfer the performance to avatar body motion is necessary. However, transferring motion between two different articulated structures is a non-trivial task. While a gesture recognition approach can generate elegant motion similar to an example animation, it can suffer from noticeable delays that can cause leg movement to become asynchronous with finger movement. This limitation renders the approach unsuitable for certain applications such as gaming and storytelling. The first idea that comes to mind to address the issue of asynchronization is direct mapping, which can be achieved by using either contact data from the fingertips or the rotation of the finger joints. While this approach can achieve full synchronization between the finger and leg movements, it can also result in unrealistic leg motion. Furthermore, since finger-walking only mimics leg movement, users may desire to see upper body movement that is relevant to the leg motion, which cannot be generated by direct mapping alone. To address the aforementioned issues, we propose a method that utilizes an example animation to adjust the direct mapping method and make the resulting leg movement feasible. Additionally, we use the example animation to generate corresponding full-body motion that is realistic and expressive, making it suitable for various applications. The full process of the proposed method is illustrated in Figure 1.

This work aims to design and develop a method to achieve real-time finger-walking performance-based puppetry for a human avatar. The key contribution of our method lies in its ability to retarget the motion of two distinct, articulated structures, i.e., the hand motion to the entire body motion, thereby resulting in a seamless, synchronous movement of the avatar in response to the user's finger movements. Moreover, the intuitive and expressive method, which requires only a commonly available RGB camera for motion capture, is suitable for various applications such as gaming, storytelling, animation editing, etc. To explore more feasible finger-walking performances and corresponding full-body motion for human avatar puppetry, we first conducted a user study and categorized the collected finger-walking performances. Through the study, we discovered several classes of finger-walking movements that casual users prefer to use for human avatar puppetry and implemented these movements in our proposed method. We have also designed and developed a simple storytelling application to showcase the effectiveness of our proposed performance-based puppetry method. The application allows users to select an action from a 2D user interface and manipulate a 3D virtual avatar through finger performance to create a digital narrative. This demonstrates the ease of use and versatility of the method.

In summary, this work has three main contributions:

- Introduced a motion retargeting method for different articulated structures, focusing on the finger and human body movements.
- Proposed a finger-walking performance-based puppetry method for manipulating a 3D human avatar, ensuring that the avatar's motion accurately synchronizes with the user's performance while not losing the elegant and skillful movement in the example animation.
- A user study investigates preferences of the finger-walking performances for mimicking most of the movements in real-life

2 PRELIMINARY STUDY

2.1 Apparatus and Procedure

We conducted a preliminary study to investigate additional types of finger-walking performances and determine which body movements casual users prefer to manipulate using finger-walking performances. Five participants (four males and one female) were recruited for the study. They were first shown a video of a single movement from an Olympic game and then asked to perform the movement using finger-walking and another hand movement with the same hand without restriction. These performances were intended to capture the best expression of the movement from the video while being within the camera's range to capture their full motion. During the study, participants were allowed to replay the video and perform the movement multiple times. After completing the movement using both finger-walking and another hand movement, participants were asked to choose the best way to achieve the movement between the hand and finger performances they had just completed and performance using their entire body. When selecting, participants were reminded to consider the criteria of goodness, ease of performance, and fatigue. Additionally, a brief survey was conducted to probe the reason for choosing a specific performance as the best expression of the movement.

2.2 Result and Discussion

Each participant performed 28 finger-walking motions, each of which corresponded to a movement in a video of an Olympic sport. We categorized these motions into eight primitive finger movements, excluding the global movements of the entire hand. The categorized result is listed in Table 1. Furthermore, we counted the number of sports in each category. Most participants preferred finger-walking to mimic the movement (more than three participants), which is illustrated in Table 2.

According to Table 2, actions belonging to "single step" and "stand" are not suitable to perform by finger-walking. We found body actions related to other categories in a well-known animation database, Mixamo, except for the action related to "cross finger." Finally, we chose five body motions related to each category to participate in our proposed system, which are listed in Table 3.

Table 1: Sports in Olympic Games Tokyo 2020 and the corresponding categories of finger-walking movement. Participants choose to use finger-walking to perform the movement in the sport, which is in bold text.

Run	**Cycling**, Sport climbing, **Athletics**, Boxing, Table tennis, Tennis, **Wrestling**
Kick	**Football**, **Taekwondo**, **Karate**
Cross fingers	**Skateboarding**
Stand	Shooting, Archery, Golf
Front and side split	**Artistic gymnastics**, **Judo**, **Rhythmic gymnastics**
Single step	Badminton, Fencing, Softball, Handball
Crouch	Equestrian, **Hockey**, Beach volleyball, **Trampoline gymnastics**, **Weightlifting**, **Rugby sevens**

Table 2: Number of sports in each finger-walking category and number of sports that most participants preferred to use finger-walking to perform.

Run	All: 7, Finger Walking: 3
Kick	All: 3, Finger Walking: 3
Cross fingers	All: 1, Finger Walking: 1
Stand	All: 3, Finger Walking: 0
Front and side split	All: 3, Finger Walking: 3
Single step	All: 4, Finger Walking: 0
Crouch	All: 7, Finger Walking: 4

3 SYSTEM IMPLEMENTATION

We proposed a finger-walking performance-based puppetry system that allows users to control virtual avatars in real-time with straightforward hand gestures. From the user's perspective, the system's usage can be divided into two steps: the training stage and the testing stage. During the training stage, the user must mimic the animation by fingerwalking performance. Then, the replay of the performance was shown to the user, and the time that the movement started and ended needed to be labeled. Also, the time that motion begins and ends in the example animation must be marked. In the testing stage, the user's intended animation must be specified. After that, the user can perform the animation by hand, and the proposed system can show an avatar motion synchronized to the user's hand performance and similar to the desired animation. Figure 2 illustrates the system diagram.

3.1 Data Preprocessing in the Training Stage

Two kinds of data were used as reference information in the proposed hand performance-based animating system: Hand performance video and character animation. A simple, easy-to-get RGB camera Realsense from Intel was used to capture hand performance video. The video is recorded in resolution (848, 480), and the frame rate was 60 frames per second (FPS). The Realsense camera performs a depth estimation result, but it was not used in this study since our hand pose estimation method does not require depth

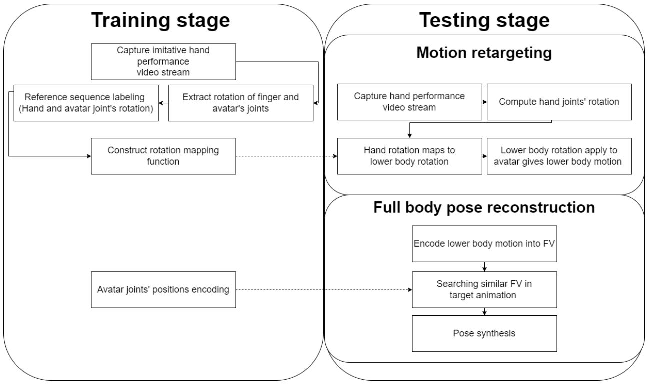

Figure 2: System diagram.

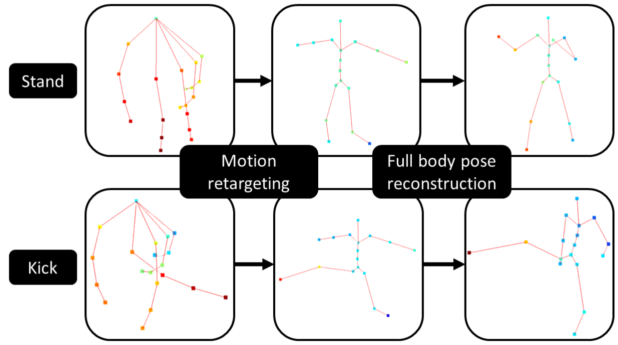

Figure 3: Hand and body pose in each computation step of the testing stage. The figure depicts hand articulation, lower body position after rotation mapping, and whole body pose after pose blending from left to right. The left-hand pose mimics the stand-up state in the front kick animation, while the right-hand pose mimics a kick with the right leg in the front kick animation by raising the index finger.

Table 3: Example animations were used in this study, as well as the corresponding finger-walking category.

Example animation	Finger-walking category
Run	Run
Joyful jump	Crouch
Double leg jump	Crouch
Front kick	Kick
Side kick	Front and side split

information. We used the Google Mediapipe [9] framework to get the estimated 3D hand pose of 21 finger joints. Since finger-walking is trying to mimic two legs motions by the index finger and middle finger, only joints of the wrist and joints of the index and middle finger are used in this study, and they are paired with corresponding avatar joints of two legs. The rotation of those joints was computed from the estimated pose. The other reference information for the proposed method is the character animation that the user intends to perform. Five animations collected from Mixamo are utilized in this study, including three movements: running, jumping, and kicking. The used animations are listed in Table 3.

3.2 Hand Motion Retargeting to Lower Body Motion

A mapping that maps finger joints' rotation to lower body joints' rotation is constructed at the training stage since they belong to two different kinematic chain systems, and the range of rotation values is diverse. The mapping method proposed in [8] is referenced, but part is modified due to our needs. [8] suggested using an Euler angle along a single axis of the avatar's joint and an Euler angle along a single axis of the finger's joints to construct a mapping function. As extracting an Euler angle along a single axis can result in undesired rotation, and the Euler angle is unsuitable for interpolation, we adopt quaternion to represent the rotation of the avatar's joint. In contrast, we used the Euler angle to describe the rotation of finger joints, as the performance of the hand is an imprecise imitation of the target animation. Moreover, the rotation of finger joints is used exclusively for controlling the avatar and is not interpolated for display to the user.

In the section on data preprocessing, we mentioned that a specific motion's starting and ending times were labeled in the training stage, and two reference sequences, HRRS and ARRS, were constructed. Each reference sequence represents rotations through the time of a lower body joint or a finger joint. HRRS is a sequence $X = [x_1, x_2, ..., x_n]$, which includes the elements representing the rotation of a finger joint at a specific frame. And n represents the number of frames in HRRS, which is set to 100 in this study. ARRS is a sequence $Y = [y_1, y_2, ..., y_n]$, which includes the elements representing the rotation of a lower body joint at a specific frame. The frame count of HRRS and ARRS are the same, as we interpolated the same number of frames in the training stage. Then a set of rotation pairs $M = (x_1, y_1), ..., (x_n, y_n)$ can be constructed, which is used for mapping $\Phi : x \longrightarrow y$. During the testing stage, a finger joint rotation, denoted as $\hat{x}$, is estimated in real-time, representing the finger joint's bending state. Mapping is done by identifying a x_t in X, which exhibits the highest degree of similarity to x, and outputs the corresponding y_t in Y.

After rotation mapping, the result rotation was applied to the lower body joints of an avatar. By utilizing forward kinematics, lower body joints were moved by those rotations. Two legs were considered two different two-bone kinematic chains, each consisting of three joints: upper leg, lower leg, and foot. Since hip joints do not rotate by hand rotation, the positions of both left and right upper leg joints remain fixed. However, the mapped rotations rotate the upper and lower leg. Thus, lower leg and foot joints are moved and generate a temporal trajectory. While hand movements typically approximate an avatar's motion, the animations used in our study feature rich movements in both legs. Therefore, we searched for similar motions in the example animation and the motion generated by applying mapped rotations using a single-foot trajectory to minimize the computational load. As the foot is an end effector of a leg, its position is influenced by the rotation of both the lower and upper leg joints, implying the foot's position encompasses the information about both rotations. Additionally, when using hand performance to control the avatar, the user is sensitive to synchronizing between the fingertips and the foot. In summary, the foot position of a specific leg is utilized to search for the desired motion within an animation that the user intends to perform.

3.3 Full Body Pose Reconstruction

We used the position sequence of an end effector encoded to an FV to identify the proper full-body pose. The mapped rotation of a joint is denoted as $\hat{y}$. The position of the end effector is denoted as $\hat{z}$. Rotations of finger joints are streaming data in the testing stage, and the proposed system maps them to lower body joint rotations and then converts them to positions of lower body joints. Thus, mapped rotations of a joint can be denoted as $\hat{Y} = [\hat{y_{t-m+1}}, ..., \hat{y_{t-1}}, \hat{y_t}]$, where $\hat{Y}$ is a sequence consisting of data from m frames. In addition, the position sequence of an end effector can be denoted as $\hat{Z} = [\hat{z_{t-m+1}}, ..., \hat{z_{t-1}}, \hat{z_t}]$. After retargeting, the lower body performs movements that synchronize with finger motion, but the upper body remains still. Position sequence is concatenated into a one-dimensional vector $\hat{w} = [\hat{z_{t-m,x}}, ..., \hat{z_{t-1,x}}, \hat{z_{t,x}}, \hat{z_{t-m,y}}, ..., \hat{z_{t-1,y}}, \hat{z_{t,y}}, \hat{z_{t-m,z}}, .., \hat{z_{t-1,z}}, \hat{z_{t,z}}]$, $\hat{z_{t,x}}$ represents the position of the foot along the x axis at frame t. The m in the vector represents the number of frames utilized for encoding a feature vector, and in this study, it is set to a value of 10. Unlike the FV encoding method in [1], we did not use velocity and acceleration as features. The speed of the lower body motion after retargeting is the same as the finger motion, which might differ significantly from the more down-body motion in the example animation. Therefore, including velocity and acceleration within the feature vector could be more helpful and decrease the search speed for similar FV. As mentioned earlier in the section on data and preprocessing, we augment the feature vector by incorporating six different speed ratios to address the significant discrepancy in the speed of movement between user performance and animation. End effector positions were encoded into FVs in the training stage in the same way, and it is indicated by a sequence of FVs $S = [f_1, f_2, ..., f_u]$. The scalar t denotes the t-th frame in animation, while the variable u represents the number of feature vectors that have been encoded. To find the top k f_t that are the most similar to $\hat{w}$ with l2 distance as a similarity measurement method, we implemented a K-Nearest Neighbor based on KDTree data structure via Scipy [11]. The scalar k is set to 5 in this study. Each FV f_t corresponds to a whole body pose at the t-th frame in the animation. k entire body poses in animation relevant to the top k f_t are weighted averages, and the weights are inversely proportional to the distance between f_t and $\hat{w}$. Figure 3 took the stand and kick actions as examples to show the computing pose in the motion retargeting and pose reconstruction process.

4 APPLICATION

A simple storytelling application has been developed to demonstrate the feasibility of the proposed finger-walking performance-based interface. The user interacts with a digital 3D scene and 3D avatar displayed on a monitor, with an RGB camera in front of the monitor and a keyboard for system configuration. Users can use the keyboard to choose the action intended to perform and manipulate the 2D translation of the virtual avatar. In addition to the virtual scene and avatar controlled by the user, a group of 2D user interfaces (UI) are also displayed on the screen Figure 4. The group of UI consists of six panels; the first panel shows the current action intended to be performed, and the other panels are labeled with action names, which can be performed, and a corresponding number. All the actions are listed in Table 3. Users can select the desired action to be performed by the avatar by pressing the corresponding key on the keyboard; for example, press 1 to complete the first action or press 2 to perform the second action.

The storytelling application provides a scene where the user plays the role of a human protagonist attempting to get an apple from a tree and return home. During the digital storytelling experience, the user must manipulate the protagonist's movements and interact with objects in the virtual environment. In the initial scene, the protagonist is positioned near a tree and must approach it, followed by kicking the trunk to cause an apple to fall from the branches (Figures 5a and 5b). In the second scene, the protagonist must cross a road with cars passing by. The user must quickly assess the situation and choose the jump action, deftly dodging the oncoming vehicles one by one 5c. After that, the protagonist finally walks home and gives a joyful jump 5d and 5e. This application demonstrates the proposed interface's feasibility. It allows users to manipulate an avatar and have it interact with a virtual environment to complete a digital narrative using hand performance. The interface's capability to control avatar-environment interactions opens the potential for creating gaming applications.

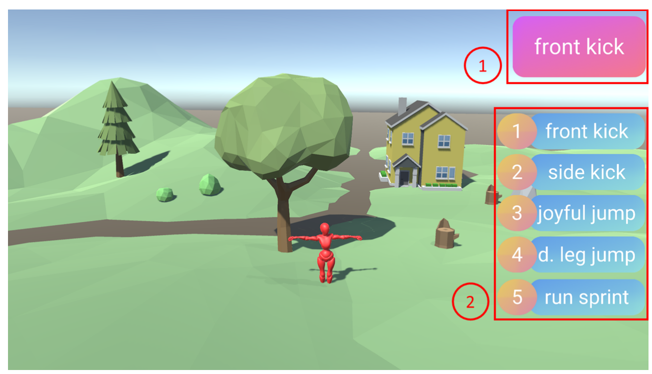

Figure 4: Graphical User Interface in the Storytelling application. The top right panel, labeled with 1, shows the current action. Below the first panel, five panels show the possible action that can be achieved, which the user can choose by pressing the corresponding number on the keyboard. This interface allowed users to interactively control the actions of a virtual character in the storytelling application.

5 CONCLUSION

We proposed a real-time digital puppetry method that uses finger-walking performance to manipulate a human avatar. The technique features a novel interface that is responsive and expressive, enabling users to interact with a 3D virtual environment using hand performance captured by a consumer-level RGB camera. Also, a preliminary study explores finger-walking movements that novice users prefer to perform to represent corresponding body motions.

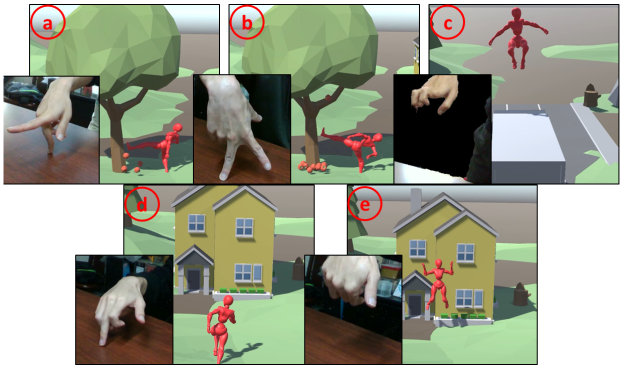

Figure 5: Examples of storytelling scenes and corresponding finger-walking performances. Subfigures (a) and (b) depict the user manipulating an avatar to kick a tree. Subfigure (c) demonstrates double-leg jumps to avoid passing cars. Subfigures (d) and (e) show the user attempting to return home and celebrating with a joyful jump, respectively.

We propose a motion retargeting technique for different articulated structures. The method generates full-body poses by retargeting the user's hand to the avatar's lower body motion. Then, it matches the motion trajectory of the retargeted motion to the motions in an example animation to reconstruct the full-body pose. Finally, it blends checked motions from the example animation to give users an intuitive and expressive agency interface over the avatar. The proposed method is useful for 3D gaming and storytelling applications requiring an immersive virtual world experience. Additionally, a storytelling application has been implemented and demonstrates the usefulness and effectiveness of the proposed method.

ACKNOWLEDGMENTS

This research was supported in part by the National Science and Technology Council of Taiwan (NSTC112-2218-E-002-029, 111-2221-E-002-145-MY3, 112-2634-F-002-006), and National Taiwan University (NTU113L894001, 113L8514, 112L900902).

REFERENCES

[1] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D. Wilson. 2021. CoolMoves: User Motion Accentuation in Virtual Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 52 (jun 2021), 23 pages. https://doi.org/10.1145/3463499

[2] Raphael Anderegg, Loïc Ciccone, and Robert W. Sumner. 2018. PuppetPhone: Puppeteering Virtual Characters Using a Smartphone. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games* (Limassol, Cyprus) *(MIG '18)*. Association for Computing Machinery, New York, NY, USA, Article 5, 6 pages. https://doi.org/10.1145/3274247.3274511

[3] Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig Animation with a Tangible and Modular Input Device. *ACM Trans. Graph.* 35, 4, Article 144 (jul 2016), 11 pages. https://doi.org/10.1145/2897824.2925909

[4] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D Puppetry: A Kinect-Based Interface for 3D Animation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 423–434. https://doi.org/10.1145/2380116.2380170

[5] Ching-Wen Hung, Ruei-Che Chang, Hong-Sheng Chen, Chung Han Liang, Liwei Chan, and Bing-Yu Chen. 2022. Puppeteer: Exploring Intuitive Hand Gestures and Upper-Body Postures for Manipulating Human Avatar Actions. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology* (<conf-loc>, <city>Tsukuba</city>, <country>Japan</country>, </conf-loc>) *(VRST '22)*. Association for Computing Machinery, New York, NY, USA, Article 13, 11 pages. https://doi.org/10.1145/3562939.3565609

[6] Rahul Jain, Jingyu Shi, Runlin Duan, Zhengzhe Zhu, Xun Qian, and Karthik Ramani. 2023. Ubi-TOUCH: Ubiquitous Tangible Object Utilization through Consistent Hand-object interaction in Augmented Reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (<conf-loc>, <city>San Francisco</city>, <state>CA</state>, <country>USA</country>, </conf-loc>) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 12, 18 pages. https://doi.org/10.1145/3586183.3606793

[7] Yu Jiang, Zhipeng Li, Mufei He, David Lindlbauer, and Yukang Yan. 2023. HandAvatar: Embodying Non-Humanoid Virtual Avatars through Hands. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Hamburg</city>, <country>Germany</country>, </conf-loc>) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 309, 17 pages. https://doi.org/10.1145/3544548.3581027

[8] Wai-Chun Lam, Feng Zou, and Taku Komura. 2004. Motion Editing with Data Glove. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology* (Singapore) *(ACE '04)*. Association for Computing Machinery, New York, NY, USA, 337–342. https://doi.org/10.1145/1067343.1067393

[9] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. arXiv:1906.08172 [cs.DC]

[10] Siyou Pei, Alexander Chen, Jaewook Lee, and Yang Zhang. 2022. Hand Interfaces: Using Hands to Imitate Objects in AR/VR for Expressive Interactions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>New Orleans</city>, <state>LA</state>, <country>USA</country>, </conf-loc>) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 429, 16 pages. https://doi.org/10.1145/3491102.3501898

[11] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[12] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 552–567. https://doi.org/10.1145/3472749.3474769

[13] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>New Orleans</city>, <state>LA</state>, <country>USA</country>, </conf-loc>) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 496, 19 pages. https://doi.org/10.1145/3491102.3501904