# ReMagicMirror: Action Learning Using Human Reenactment with the Mirror Metaphor

Fabian Lorenzo Dayrit[1], Ryosuke Kimura[3], Yuta Nakashima[1], Ambrosio Blanco[2], Hiroshi Kawasaki[3], Katsushi Ikeuchi[2], Tomokazu Sato[1], and Naokazu Yokoya[1]

[1] Nara Institute of Science and Technology
Takayamacho 8916-5, Ikoma, Nara, 630-0101 Japan
[2] Microsoft Research Asia
Building 2, No. 5 Dan Ling Street, Haidian District, Beijing, 100080, China
[3] Kagoshima University
Korimoto 1-21-24, Kagoshima 890-8580 Japan

**Abstract.** We propose ReMagicMirror, a system to help people learn actions (e.g., martial arts, dances). We first capture the motions of a teacher performing the action to learn, using two RGB-D cameras. Next, we fit a parametric human body model to the depth data and texture it using the color data, reconstructing the teacher's motion and appearance. The learner is then shown the ReMagicMirror system, which acts as a mirror. We overlay the teacher's reconstructed body on top of this mirror in an augmented reality fashion. The learner is able to intuitively manipulate the reconstruction's viewpoint by simply rotating her body, allowing for easy comparisons between the learner and the teacher. We perform a user study to evaluate our system's ease of use, effectiveness, quality, and appeal.

**Keywords:** 3D human reconstruction · human reenactment · RGB-D sensors

## 1 Introduction

When people want to learn an action, e.g., a martial arts performance or a dance, one of the most intuitive ways is to watch a teacher performing the action and imitate it. This can be done in person, e.g., how most martial arts are traditionally learned, or from a recording, such as a video of teacher performing the action. Both ways have advantages and disadvantages: imitating a teacher in-person is limited by the availability of the teacher regarding time and place, but the learner is free to observe the action from any point of view. In contrast, the video may be watched anytime and anywhere, but is limited to the original capturing point of view, which may pose problems for difficult or hard-to-understand actions.

There are several technical remedies that try to provide convenience for learners without spoiling the capability to view the action from an arbitrary viewpoint


Fig. 1: Our ReMagicMirror system. The learner is mirrored on the left in the screen, and the reenactment of the teacher is shown on the right.

as in-person learning does. Most of them make use of augmented reality (AR), which is the technique of visualizing virtual objects in the real world. Several systems [11, 12] provide visual guidance for tasks by overlaying, e.g., arrows and labels on key objects. We especially draw inspiration, however, from systems that allow easy comparison with the learner's own body by using a mirror [1, 4].

This paper proposes a system that is able to be replayed like a video and allows viewing from an arbitrary viewpoint, combining these with the mirror metaphor to help learners comprehend difficult actions. Our system is centered around *reenactments* of motion, i.e. a sequence of novel views of a person in motion, and the system is called ReMagicMirror (Reenactment-based Magic Mirror). This system captures and displays reenactments of the teachers' motions in order to aid learner comprehension. It consists of a large screen that mirrors the learner, upon which the system then overlays the teacher's reenactment as shown in Figure 1. The reenactment should be rendered from an intuitive viewpoint for the learner, i.e., matching the learner's own body orientation. To view the action from the side, for example, the learner has only to turn his/her own body to the side, and by doing so can easily see the differences between his/her side view and the teacher's. This makes it easy to compare two actions.

Our system acquires the reenactments by fitting a parametric model to two RGB-D sequences: First, we capture the teacher's motion sequence using two RGB-D cameras, acquiring the entire shape of the teacher. Next, the system generates a 3D mesh sequence by fitting a parametric model, such as [3, 6], to the scans. Finally, the motion sequence is overlaid on top of a screen mirroring our learner (Fig. 1).

The main contributions of this paper are summarized as follows:

1. A novel method to synthesize views of humans in motion, called reenactments, using two RGB-D streams.

2. ReMagicMirror, an end-to-end reenactment display system, that helps learners by overlaying a mirror of them with teachers' reenactments with easily-controlled viewpoints.
3. User study results to show how well the proposed reenactment display system helps learners.

## 2 Related works

Our system has its roots in two main fields of research: human shape reconstruction and augmented reality for learning.

### 2.1 Human shape reconstruction

For our system, we render novel views of the teacher in motion by first reconstructing the teacher's shape and motion from two depth cameras, rendering these from the desired viewpoint, and finally displaying it to the learner. Shape reconstruction is an active research field and techniques here are mainly divided into *model-free* and *model-based* approaches, referring to the usage or non-usage of a human shape model.

The model-free approach requires no prior data on human shape and makes no assumptions on the person captured. Most recent methods of this type employ variants of the signed distance field technique, which is basically a registration problem of multiple depth maps and represents 3D shape using the zero-level iso-surface of the signed distance field. This approach was originally designed for rigid scenes, and one of the more well-known examples is KinectFusion [15]. This approach was later extended to handle non-rigid objects by describing the deformation of objects with transformations of signed distance field [9, 13, 14]. These methods can generate surprisingly high quality 3D shapes, but may lack tracking stability with regards to, e.g., occlusions.

In contrast to the model-free approach the *model-based* approach uses prior knowledge on the object to be captured to facilitate entire object reconstruction, and most existing methods that take this approach are designed for human body reconstruction. Most methods of this type use a parametric model of human body shape, such as SCAPE [3] and TenBo [6]. These methods in particular describe plausible body shapes using pose and shape parameters that control the human body's attributes like weight, height, etc.

Existing methods that adopt the model-based approach basically fit the parametric model to a point cloud of depth observations. For example, Weiss et al. [17] proposed a system using SCAPE, where the fitting process is initialized with skeletal tracking results. Bogo et al. [5] use SCAPE with several modifications including multi-resolution mesh fitting and using displacement maps for finer details. Due to the modification of multiple resolution meshes, their system no longer relies on skeletal tracking, which is error-prone.

Since we know we are targeting humans, our system uses a model-based approach for stability. We fit a TenBo model [6] to our input depth sequences, applying temporal constraints for smoothness. This has the following advantages.

1. Our input depth maps give no guarantee that they cover the entire body of the teacher due to, e.g., self-occlusion. Fitting a body model to the visible regions gives us a plausible estimate of the unobserved ones.
2. Our fitting process is constrained in two ways to increase stability. Firstly, since our system is designed for capturing a single person, i.e., a teacher, we keep the same shape parameters of the TenBo model for the entire sequence. Also, we can assume that each frame in a sequence is captured a few milliseconds (at most 33ms for a Kinect) after the previous one, which let us add temporal smoothness constraints to the fitting process.
3. The results of fitting include pose parameters, from which we can derive the relative camera view. We use this during the viewing stage in order to display the most appropriate viewpoint to the learner.

### 2.2 Augmented reality for learning

Using AR technology for learning has become a popular topic in recent years. AR visualizes virtual objects in the real world, which can range from placing labels on top of real objects to rendering entire virtual objects in real environments. Several systems implement AR techniques for the purpose of learning. For example, Henderson and Feiner [11] proposed a system to help users learn a procedure of actions by attaching a sequence of 3D arrows to key objects. Another system [12] generates virtual targets for rehabilitating users who have had a stroke. Dayrit et al. [8] proposed a similar system that renders a teacher through a handheld device, such as a tablet, using AR technology. The main technical difference of our work is that we adopt a parametric 3D human shape model to improve visual quality.

The mirror metaphor in particular is well-suited to AR, as it is instantly understandable. Mirracle [4] is a system for anatomy education that simulates a mirror and projects virtual organs in the appropriate place on top of the learner's body in the mirror. Here, the mirror metaphor allows users to easily learn about their own bodies. Another system employing the mirror metaphor, YouMove [1], projects a teacher's motions as stick figures on a half-silvered mirror, allowing learners to align their body with the stick figures in the mirror. In their user study, participants were asked to imitate motions either from videos or using the system, and those using the system were noticeably closer to the teacher's motion. We thus believe that using the mirror metaphor to let learners compare their own motion to the virtual teacher's motion will be helpful for action learning.

## 3 ReMagicMirror System

Figure 2 shows an overview of our ReMagicMirror system. The system has three offline stages and one online one. The offline portion captures and builds the reenactment, and is composed of the capturing, fitting, and texturing stage. In the capturing stage, a teacher captures his/her action with two RGB-D sensors

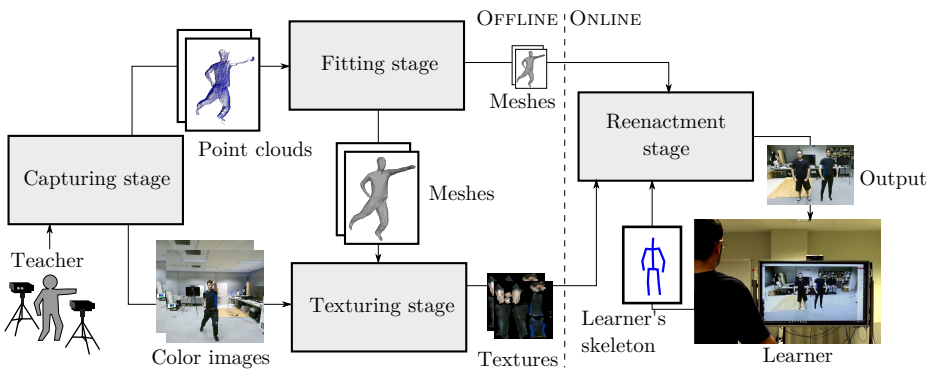



Fig. 2: System overview.

with known relative poses. Our system merges the point clouds in the two RGB-D streams from the two sensors, feeding the merged point clouds into the fitting stage. Since the point clouds usually have unobserved regions due to, e.g., occlusion, we fit the parametric 3D human model [6] to each merged point cloud to obtain a mesh sequence. The texturing stage extracts the texture applied to each triangle of the every frame from RGB images. In the online reenactment stage, our system presents, to the learner, his/her flipped image for the mirror metaphor. The system also provides the teacher's reenactment as shown in Fig. 1 so that the learner can easily imitate the teacher's action. For each playback, the learner can intuitively determine the orientation of the reenactment, observing it from a desired direction. The following sections detail each stage.

### 3.1 Capturing stage

In the capturing stage, our system records an action of the teacher using a pair of RGB-D sensors facing each other. The relative pose between these two sensors is calculated, and they are manually synchronized. Since we require the depth and color pixels belonging to the teacher, separate from the background, we extract the teacher's region using such a method as [16]. After extracting the teacher's region, we regain the 3D position of each depth pixel to form a point cloud. We merge the two point clouds from the pair of sensors using the relative pose calculated above.

We denote the $f$-th frame point cloud with $N_f$ points, by

$$Z_f = \{\mathbf{z}_{fn} | n = 1, \ldots, N_f\}, \tag{1}$$

and the RGB images from first and second sensors as $I_f^1$ and $I_f^2$, respectively.

### 3.2 Fitting stage

Figure 3 (a, top) shows examples of merged point clouds. Generally, even though we capture the teacher from both his front and back, the point cloud can be

(a) (b)

Fig. 3: (a) Top: Example input point clouds. Middle: Examples of fit meshes. Bottom: Textured meshes. (b) Segmented reference mesh, front and back. Each color in (b) represents one of the 13 body parts: head, shoulders, upper arms, lower arms, torso, abdomen, upper legs, and lower legs.

incomplete because of occlusion or difficult-to-capture regions such as hair. In addition, some body parts can partially be out of the sensors' field of view. To reconstruct the complete shape of his body, we fit the TenBo model [6], which is a state-of-the-art statistical human shape model, to each point cloud.

Training a statistical human shape model, usually requires a large amount of registered meshes of multiple subjects in various poses. We used the MPII dataset [10], which contains over 500 registered meshes. For stable fitting, we selected a mesh and reduced the number of vertices in it from 6,449 to 502 using the quadric edge collapse decimation algorithm [7]. From here we treat the decimated mesh as the reference.

This decimation is transferred to all other meshes in the dataset as they are registered, i.e., we keep the same vertices in a mesh as the reference and use the edges in the reference instead of the original ones. We refer to the reference as

$$M_{\mathrm{X}} = \{X, E\}, \tag{2}$$

where $X = \{\mathbf{x}_j | j = 1, \ldots, J\}$, $\mathbf{x}_j$ being the $j$-th vertex, and $E$ contains the pairs of vertex indices that form the edges of the reference. The TenBo model also requires segmenting the mesh into body parts so that each body part is not subjected to excessive deformation. Instead of using an automatic approach, such as [2], we manually segmented the mesh as in Fig. 3 (b).

The TenBo model, like other parametric shape models such as [3], regresses a deformation matrix of each triangle in the reference given the body part poses

$\Theta$ and shape parameter $\mathbf{v}$, where $\Theta = \{\boldsymbol{\theta}_l | l = 1, \ldots, L\}$ is a set of rotation representations for all body parts. Letting $\mathbf{D}_k(\Theta, \mathbf{v})$ be the deformation matrix and $\mathbf{R}(\boldsymbol{\theta}_l)$ be the rotation matrix obtained from $\boldsymbol{\theta}_l \in \Theta$ for body part $l$, the deformed triangle $k$'s edges, which are called triangle vectors, $\triangle\mathbf{y}_{k1}$ and $\triangle\mathbf{y}_{k2}$ can be given by

$$\triangle\mathbf{y}_{k1} = \mathbf{R}(\boldsymbol{\theta}_l)\mathbf{D}_k(\Theta, \mathbf{v})\triangle\mathbf{x}_{k1}$$
$$\triangle\mathbf{y}_{k2} = \mathbf{R}(\boldsymbol{\theta}_l)\mathbf{D}_k(\Theta, \mathbf{v})\triangle\mathbf{x}_{k2},$$

where $\triangle\mathbf{x}_{km} = \mathbf{x}_{km} - \mathbf{x}_{k0}$ and $\mathbf{x}_{km}$ $(m = 0, 1, 2)$ is in $X$ and forms a triangle of the mesh. In the above equation, $l$ is the body part that triangle $k$ belongs to.

The fitting algorithm tries to find the body part poses $\Theta$ and the shape parameter $\mathbf{v}$. We modify the fitting algorithm in [6] to take advantage of the temporal continuity of meshes in successive frames. More specifically, we apply an additional smoothness term for the pose parameters that penalizes pose differences between adjacent frames, as well as modifying the shape parameter fitting to simultaneously take multiple frames into account. The optimization involves three terms: the model error term $\mathcal{M}$, the point cloud error term $\mathcal{P}$, and the temporal pose smoothness term $\mathcal{R}$.

The model error term penalizes the difference between the TenBo model-based body shape prediction and the deformed mesh $Y_f$ in frame $f$. $\triangle\mathbf{y}_{fkt}$ is triangle vector $t \in \{1, 2\}$ of triangle $k$ in frame $f$, the term is given by

$$\mathcal{M}(Y_f, \Theta_f, \mathbf{v}) = \sum_{k=1}^{K} \sum_{t} \|\mathbf{R}(\boldsymbol{\theta}_{fl})\mathbf{D}_k(\Theta_f, \mathbf{v})\triangle\mathbf{x}_{kt} - \triangle\mathbf{y}_{fkt}\|^2. \tag{3}$$

The point cloud error term $\mathcal{P}$ for frame $f$ is the difference between the deformed mesh $Y_f$ and the point cloud $Z_f$. As there are no explicit correspondences between the deformed mesh and the point cloud, we first use the rigid iterative closest point (ICP) algorithm to bring the mesh into rough alignment, then assign correspondences by nearest neighbor. Using $\tilde{\mathbf{y}}_f(\mathbf{z}_{fn})$ as the nearest vertex in $Y_f$ to point cloud point $\mathbf{z}_{fn}$, the point cloud error term is

$$\mathcal{P}(Y_f) = \sum_{n} \|\tilde{\mathbf{y}}_f(\mathbf{z}_{fn}) - \mathbf{z}_{fn}\|^2. \tag{4}$$

The pose smoothness term $\mathcal{R}$ for frame $f$ penalizes large differences in pose between frames. Due to our assumption of fitting depth image sequences, we do not want subsequent frames to vary wildly. This term increases fitting robustness. The term is defined as the sum of squared Frobenius norms:

$$\mathcal{R}(\Theta_f, \Theta_{f+1}) = \sum_{l} \|\mathbf{R}(\boldsymbol{\theta}_{fl}) - \mathbf{R}(\boldsymbol{\theta}_{(f+1)l})\|_{\text{fro}}^2. \tag{5}$$

The final meshes $M_{\text{Y},f} = \{Y_f, E\}$ can be found by minimizing the following objective with respect to $Y_f$ and $\Theta_f$ for $f = 1, \ldots, F$ as well as $\mathbf{v}$:

$$\sum_{f=1}^{F} [\mathcal{M}(Y_f, \Theta_f, \mathbf{v}) + w_z\mathcal{P}(Y_f)] + w_r \sum_{f=1}^{F-1} \mathcal{R}(\Theta_f, \Theta_{(f+1)}). \tag{6}$$

We cannot handle all frames at once because of memory requirements. We instead use a sliding window of three frames at a time with the second and third frames' parameters being updated (frames 1 and 2 are independently minimized). The minimization is done using coordinate descent. In each iteration, we first minimize with respect to $Y_f$, and then $\Theta_f$. Assuming that shape parameter $\mathbf{v}$ does not change along with the sequence, we deal with $\mathbf{v}$ only in the minimization for frame 1, and use the value for the rest of minimization processes. Figure 3 (a, middle) shows examples of fit meshes.

### 3.3 Texturing stage

Our system extracts textures from RGB images $I_f^1$ and $I_f^2$ from the first and second sensors using $M_{\mathrm{Y},f}$ $(f = 1, \ldots, F)$. For each triangle in frame $f$, we project its vertices $\mathbf{y}_{km}$ to $I_f^1$ and $I_f^2$. Since the image region corresponding to a triangle may not necessarily be visible (e.g., an arm may be occluding the body), we must detect and handle such regions.

To do this, we generate a depth map of $M_{\mathrm{Y},f}$ for each sensor that captures $I_f^1$ and $I_f^2$, and project a vertex to them. If the depth component of one of the vertices in a triangle is inconsistent with the corresponding depth value by a threshold $T$, we deem the triangle not visible. If the triangle is not visible from both sensors, we use the averaged texture calculated over corresponding visible triangles in the entire sequence. Figure 3 (a, bottom) shows some examples of textured meshes. We create a $1024 \times 1024$ texture per frame.

### 3.4 Reenactment stage

In the reenactment stage, the system reenacts the captured action and presents it to the learner through our interface with the mirror metaphor. This section describes reenactment generation and the interface in detail.

Figure 1 shows the configuration of our system's learning interface. The interface has one RGB-D sensor to capture the learner and the environment as well as a screen to present the captured live video stream from the sensor and the reenactment of the teacher. The RGB image in the live video stream is flipped before it is presented to the learner so that it appears like a mirror. Note that the image is not a true mirror image as the RGB-D sensor is on top of the screen. We however consider it similar enough to the learner's mental model of a mirror.

One key aspect of our system is that it can present the teacher's action from any direction that the learner wants. For this, we use a skeleton tracker (e.g., [16]) to obtain the learner's shoulders' position and compute the learner's direction. After a fixed amount of time, the system fixes the rotation of the teacher's reenactment and starts playing the action.

## 4 Evaluation

To implement our system, we used two Microsoft Kinect v2s as our RGB-D sensors. We used Kinect v2 SDK for extracting the teacher's region in depth

maps and for skeleton tracking. The fitting stage is implemented on a Windows PC with 3.20GHz CPU and 32GB memory. Optimization process (Eq. (6)) takes around 5 minutes per frame. We use $w_z = 1$, $w_r = 0.05$, and $T = 10$ cm. For the reenactment stage, the screen is $165 \times 97$ cm. The system was implemented on a Windows PC with 3.40GHz CPU and 8GB memory. It runs at 20FPS.

We conducted an objective evaluation to demonstrate how well our system helps users learn actions and a survey to subjectively evaluate our system in terms of ease of use, effectiveness, graphics quality, and appeal.

### 4.1 Objective Evaluation of Effectiveness

We compared the system against the process of learning by imitating a video. We recorded four Taekwondo actions (A, B, C, and D) for this purpose, ranging from 4-12 seconds long[4]. We divided the actions into two groups: Group 1, consisting of actions A and B, where the teacher mainly faced forward, and group 2, consisting of actions C and D, with no restriction. Users learned one action from each group using the system, and the other with the video.

For this evaluation, we recruited 14 users with ages ranging from 20-30, with 3 female and 11 male users. The process of learning an action is as follows: First, we show a video of the action to the user. Next, we establish a baseline by having the user perform the action and recording it, while the video plays again. After that, the user learns the action by practicing it over and over. The practice is accompanied either with a video of the action looping repeatedly, or with our system looping the reenactment repeatedly. For our system, the user can freely change the viewing direction before every repetition. Finally, we test the user's learning by playing the video or the reenactment one last time and recording, comparing it to the baseline.

We measured the error by recording the users' motion using a Kinect v2. Since we play the video or the reenactment at the same time that the users perform the action, we are able to match body pose frames up one to one and compare each frame directly. We compare body part orientations, normalizing all orientations relative to the spine.

Figure 4 summarizes the results of our experiment. For all sequences, those using our system were able to follow our teacher's motions more closely compared to the pre-test and those learning from a video. In fact, those learning from the video barely changed from the pre-test. We consider that this is due to the fact that the user is not able to see their mistakes, while our system makes it easy to do so, allowing users to adjust their motions to better copy the teacher's by observing the teacher from desired directions.

### 4.2 Survey

We asked the same users to try out 2 other reenactment methods: the untextured full mesh, and the skeleton of the teacher (Fig. 5). Finally, our users answered a

[4] The videos of the actions may be viewed at https://db.tt/qupIZ91a

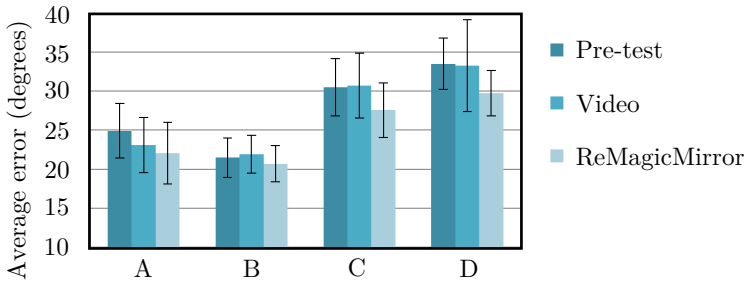
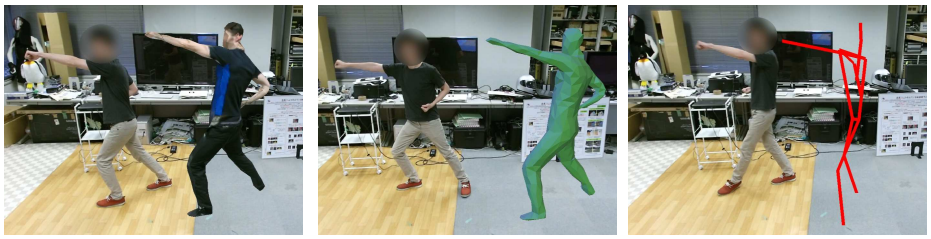



Fig. 4: Average error in degrees per joint, per frame, between the user and the teacher, for action sequences A, B, C, and D.



(a) (b) (c)

Fig. 5: (a) Textured full mesh reenactment. (b) Untextured full mesh reenactment. (c) Teacher skeleton reenactment.

survey consisting of 8 questions with the goal of evaluating the system's perceived ease of use, effectiveness, quality, and appeal (Fig. 6).

Table 1 summarizes our users' responses. Most users preferred the reenactment with a fully textured mesh for all questions, even for the equivalent video questions. This means that users found our system easy to use, effective at helping them learn actions, having high output quality, and most would use a similar system given the chance. Many users also appreciated the mirroring as it was more difficult to tell left from right by watching the video.

## 5 Conclusion

We have proposed and implemented an augmented reality system for helping users learn actions. The actions are performed by a teacher, and the system reconstructs the body and motion of the teacher using two RGB-D sensors. Using the reconstruction, the system overlays *reenactments*, which are novel views of the actions, onto a screen which also mirrors the learner. Learners are then are able to control the viewpoint intuitively by moving their own body. We conducted a user study, and found that this system allows for easy comparisons between learner and teacher, and users were able to perform more accurate motions using the system than with video. They appreciated the ability to intuitively control

**Part 1. For each reenactment (Full mesh with full textures, untextured full mesh, skeletons only) and the video:**

Q1 Was the reenactment/video comprehensible?
Q2 Was it easy to learn the motions using the system/video?

**Part 2. For each reenactment (Full mesh with full textures, untextured full mesh, skeletons only):**

Q3 Did the reenactment have good quality?
Q4 Did the reenactment resemble the original video?
Q5 Was it easy to manipulate the viewpoint to your desired one?
Q6 Were the differences between yourself and the reenactment clear?
Q7 Did changing the viewpoint help you learn the action?
Q8 Would you use this system in the future?

Fig. 6: Questions asked in our user study. Users answered from 1 (strongly disagree) to 5 (strongly agree).

Table 1: Users' averaged answers for the survey in Fig. 6, for full mesh with full textures (R1), untextured full mesh (R2), skeletons only (R3), and video (V). Users answered from 1 (strongly disagree) to 5 (strongly agree).

| | R1 | R2 | R3 | V |
|---|---|---|---|---|
| Q1 | $\mathbf{4.11} \pm 0.66$ | $3.86 \pm 0.77$ | $2.29 \pm 1.07$ | $3.86 \pm 0.77$ |
| Q2 | $\mathbf{4.29} \pm 0.73$ | $3.71 \pm 0.91$ | $2.29 \pm 0.83$ | $2.93 \pm 0.83$ |
| Q3 | $\mathbf{4.00} \pm 0.68$ | $3.71 \pm 0.99$ | $2.64 \pm 1.22$ | — |
| Q4 | $\mathbf{4.50} \pm 0.65$ | $3.64 \pm 1.08$ | $2.50 \pm 1.16$ | — |
| Q5 | $\mathbf{3.93} \pm 1.00$ | $\mathbf{3.93} \pm 1.00$ | $3.14 \pm 1.29$ | — |
| Q6 | $\mathbf{4.07} \pm 1.21$ | $3.50 \pm 1.22$ | $2.29 \pm 1.33$ | — |
| Q7 | $\mathbf{4.00} \pm 0.96$ | $3.79 \pm 0.89$ | $2.93 \pm 1.14$ | — |
| Q8 | $\mathbf{4.43} \pm 0.85$ | $3.57 \pm 1.02$ | $2.00 \pm 1.11$ | — |

the point of view while comparing motions, which to our knowledge is unique to our system at the time of writing. In general, our users preferred learning using the system over watching a video.

From here, we have several possible avenues of improvement. One way is to further develop the application, for example by developing an automatic feedback system. Another way is to make capturing easier, for example by using only a single RGB-D sensor. Finally, the texture quality can also be improved by using a higher-resolution mesh.

## References

1. Anderson, F., Grossman, T., Matejka, J., Fitzmaurice, G.: YouMove: Enhancing movement training with an augmented reality mirror. In: Proc. ACM Symposium

on User Interface Software and Technology. pp. 311–320 (2013)
2. Anguelov, D., Koller, D., Pang, H.C., Srinivasan, P., Thrun, S.: Recovering articulated object models from 3D range data. In: Proc. Conf. on Uncertainty in Artificial Intelligence. pp. 18–26 (2004)
3. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: Shape completion and animation of people. ACM Transactions on Graphics pp. 408–416 (2005)
4. Blum, T., Kleeberger, V., Bichlmeier, C., Navab, N.: mirracle: An augmented reality magic mirror system for anatomy education. In: Proc. IEEE Virtual Reality Workshops. pp. 115–116 (2012)
5. Bogo, F., Black, M.J., Loper, M., Romero, J.: Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In: Proc. IEEE Int. Conf. on Computer Vision. pp. 2300–2308 (2015)
6. Chen, Y., Liu, Z., Zhang, Z.: Tensor-based human body modeling. In: Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. pp. 105–112 (2013)
7. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: Meshlab: an open-source mesh processing tool. In: Eurographics Italian Chapter Conf. pp. 129–136 (2008)
8. Dayrit, F.L., Nakashima, Y., Sato, T., Yokoya, N.: Increasing pose comprehension through augmented reality reenactment. Multimedia Tools and Applications pp. 1–22 (2015), doi: 10.1007/s11042-015-3116-1
9. Dou, M., Taylor, J., Fuchs, H., Fitzgibbon, A., Izadi, S.: 3D scanning deformable objects with a single RGBD sensor. In: Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. pp. 493–501 (2015)
10. Hasler, N., Stoll, C., Sunkel, M., Rosenhahn, B., Seidel, H.P.: A statistical model of human pose and body shape. Computer Graphics Forum pp. 337–346 (2009)
11. Henderson, S.J., Feiner, S.K.: Augmented reality in the psychomotor phase of a procedural task. In: Proc. IEEE Int. Symposium on Mixed and Augmented Reality. pp. 191–200 (2011)
12. Hondori, H.M., Khademi, M., Dodakian, L., Cramer, S.C., Lopes, C.V.: A spatial augmented reality rehab system for post-stroke hand rehabilitation. In: Proc. Medicine Meets Virtual Reality Conf. pp. 279–285 (2013)
13. Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., Stamminger, M.: VolumeDeform: Real-time volumetric non-rigid reconstruction (2016), arXiv Preprint, arXiv:1603.08161
14. Newcombe, R., Fox, D., Seitz, S.: DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. pp. 343–352 (2015)
15. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., David Kim, A.J.D., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: Proc. IEEE Int. Symposium on Mixed and Augmented Reality. pp. 127–136 (2011)
16. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM pp. 116–124 (2013)
17. Weiss, A., Hirshberg, D., Black, M.J.: Home 3D body scans from noisy image and range data. In: Proc. IEEE Int. Conf. on Computer Vision. pp. 1951–1958 (2011)