FREE-VIEWPOINT AR HUMAN-MOTION REENACTMENT BASED ON A SINGLE RGB-D VIDEO STREAM

Fabian Lorenzo Dayrit, Yuta Nakashima, Tomokazu Sato, Naokazu Yokoya

Nara Institute of Science and Technology
{fabian-d,n-yuta,tomoka-s,yokoya}@is.naist.jp

ABSTRACT

When observing a person (an actor) performing or demonstrating some activity for the purpose of learning the action, it is best for the viewers to be present at the same time and place as the actor. Otherwise, a video must be recorded. However, conventional video only provides two-dimensional (2D) motion, which lacks the original third dimension of motion. In the presence of some ambiguity, it may be hard for the viewer to comprehend the action with only two dimensions, making it harder to learn the action. This paper proposes an augmented reality system to reenact such actions at any time the viewer wants, in order to aid comprehension of 3D motion. In the proposed system, a user first captures the actor's motion and appearance, using a single RGB-D camera. Upon a viewer's request, our system displays the motion from an arbitrary viewpoint using a rough 3D model of the subject, made up of cylinders, and selecting the most appropriate textures based on the viewpoint and the subject's pose. We evaluate the usefulness of the system and the quality of the displayed images by user study.

***Index Terms*—** Augmented reality, free-viewpoint image generation, human motion capture

1. INTRODUCTION

When viewers observe a person, hereafter referred to as an actor, performing or demonstrating an action (e.g., gymnastics or athletics) for the purpose of training or learning that action, it is best for them to directly observe the action at the time and place. Conventional video can provide more opportunities for training and learning by capturing and recording the action, so that the viewers can replay it whenever and wherever they want. However, conventional video cannot wholly capture and record the three-dimensional (3D) motion of the action. The viewers are sometimes able to make a good guess from the video, but, if there is ambiguity in the motion, it becomes harder to follow.

There exist systems that can help users learn these sorts of actions. Free-viewpoint image generation systems are able to generate an arbitrary view of a dynamic scene [1, 2, 3]. A viewer can then choose a viewpoint that is easier to comprehend, if he or she wishes. However, these kinds of systems usually require multiple cameras or sensors. These are difficult to set up for an ordinary user, and nearly impossible for outdoor or mobile use.

Fig. 1. Top: a frame from the 2D video used in the user study. Bottom: the proposed system in action.

In this paper, we propose a system that uses just a single RGB-D sensor for reenacting the motion of an actor who performs or demonstrates a specific action, for the purpose of training or learning, as shown in Fig. 1. In our system, a user captures and records the actor with an RGB-D sensor, and the captured video frames, corresponding depth images, etc., are stored in a database. Upon a viewer's request, the stored data is downloaded to his or her device, such as a tablet

computer, in order to playback or reenact the actor's action from an arbitrary viewpoint. Only requiring a single RGB-D sensor allows ordinary users to use our system; however, a single RGB-D sensor cannot capture a scene from multiple viewpoints simultaneously. The challenge is showing the viewer a different viewpoint, even when we are only able to capture from a single one. Our ideas for overcoming this limitation are that (i) the actor's rough 3D model can be generated based on the depth image and the pose of the actor in that image, and that (ii) the actor may, over the course of the entire recording, expose different angles of himself or herself to the camera, which we may then use to color the rough 3D model. To demonstrate the effectiveness of the system, we conducted a user study regarding the system's ease of comprehension, visual quality, and applicability.

2. RELATED WORK

Learning motion with AR Some systems have attempted to use augmented reality (AR) technology in order to make motion easier to comprehend. Hondori et al. [4] propose a system for rehabilitation of users who have suffered a stroke. The system shows instructions to users using AR for the purpose of doing repetitive motions (e.g., reaching for colored dots). Henderson and Feiner [5] attach AR labels to real objects, which makes the actions more concrete. This means, however, that every action must be in the context of some object, so actions not linked to objects, e.g., dances, cannot be taught. "Just follow me" [6] superimposes a trainer's motions on the user's body. Since it does so in a first-person view, upper-body movements are easier to follow, but full-body movements may be harder to comprehend. YouMove [7] is an augmented reality system that uses stick figures projected onto a mirror so that a user can compare prerecorded motions to his. In contrast to these systems, our proposed system uses a free-viewpoint image generation technique which can make motion more comprehensible as it provides detailed textures.

Free-viewpoint image generation Free-viewpoint image generation can make the motion easier to comprehend and solve the realism problem by rendering the scene from arbitrary viewpoints. Würmlin et al. [1] make use of image-based visual hull (IBVH) in order to generate free-viewpoint image sequences of a moving object. The original IBVH was developed for static objects, captured from several viewpoints. Würmlin's method adapts it by utilizing multiple cameras simultaneously. Zitnick et al. [2] segment frames into foreground and background layers, and then blend the layers from different cameras in order to render a virtual viewpoint between the cameras. Kainz et al. [3] capture and stream free-viewpoint video of moving objects by integrating the output of multiple RGB-D sensors.

Noticeably, however, all these systems require simultaneous capture from multiple viewpoints. This necessitates the use of camera arrays consisting of multiple cameras or RGB-D sensors, which are more difficult to set up and deploy than a system that only uses a single camera or RGB-D sensor.

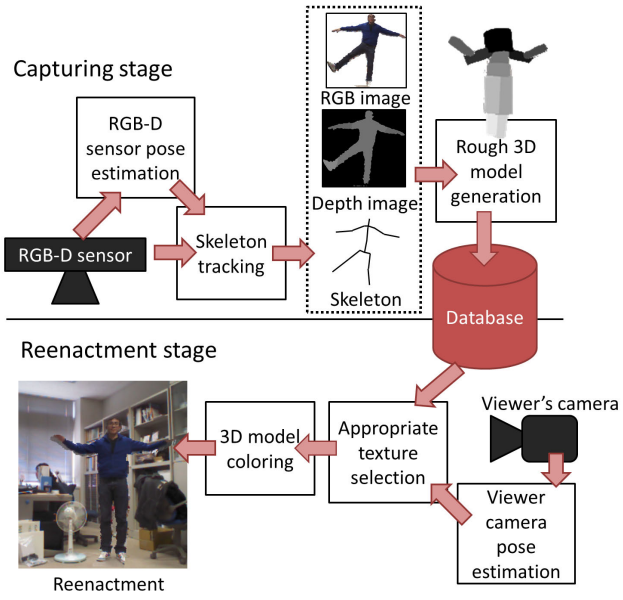

Fig. 2. Overview of the proposed system. It consists of two stages: the capturing stage and the reenactment stage.

3. OVERVIEW OF AR REENACTMENT SYSTEM

As shown in Fig. 2, the proposed AR reenactment system consists of two stages: the capturing and the reenactment stages.

During the capturing stage, a user of the proposed system captures an actor in a video stream consisting of N_{V} video frames with a single RGB-D sensor, such as Microsoft Kinect. The pose of the RGB-D sensor is estimated using a visual-SLAM technique [8], in an arbitrarily set world coordinate system (Fig. 3). Each video frame consists of an RGB image, a depth image, and the actor's estimated pose represented by the positions of a predefined set of joints, called a skeleton, in the world coordinate system. We then prepare a rough 3D model of the actor based on the skeletons and the depth images. The video stream and the 3D model are stored in a database for the later use in the reenactment stage together with the 3D map information obtained from the SLAM technique, so as to use a shared world coordinate system with the reenactment stage.

During the reenactment stage, we synthesize a novel viewpoint image of the actor from the stored video stream for the viewpoint of a viewer's camera, which we call the actor's reenactment. The pose of the viewer's camera is again estimated using the SLAM technique [8] using the 3D map stored in the database. To synthesize the reenactment of the actor, we build the rough 3D model of the actor that was

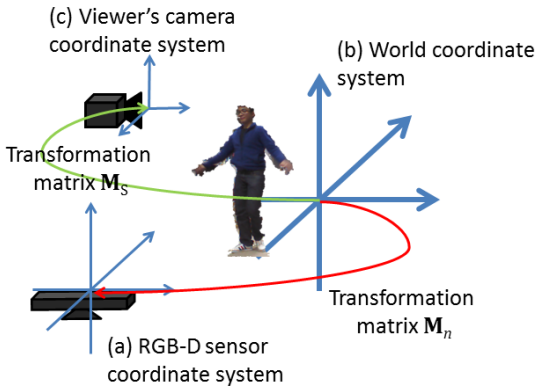

Fig. 3. (a) RGB-D sensor camera coordinate. (b) World coordinate. (c) Viewer's camera coordinate.

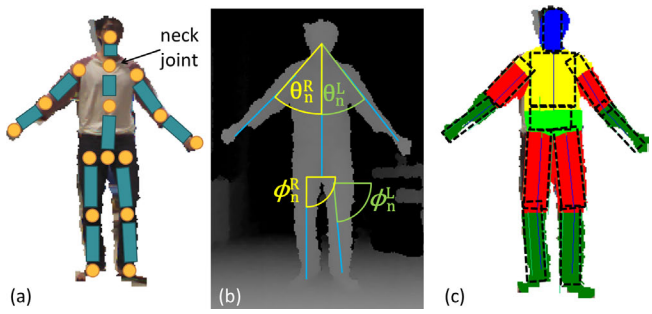

Fig. 4. (a) The skeleton representation. Orange dots are joints, and blue segments are body parts. (b) Corresponding depth image with definitions of some angles. (c) Rectangles fitted to each body part.

prepared during the capturing stage. The n-th frame of the reenactment is synthesized using the 3D model applied to the n-th skeleton in the video stream and a video frame that is the most appropriate for coloring the 3D model in the sense of a certain criterion. The proposed system then presents the reenactment superimposed on the real-time RGB image captured by the viewer's camera.

4. METHODS OF AR REENACTMENT SYSTEM

4.1. Capturing stage

For the n-th video frame of the captured video stream, we first estimate the RGB-D sensor's pose as extrinsic camera parameters $\mathbf{M}_n$ with respect to the world coordinate system using the RGB image with a SLAM technique. The capturing stage then extracts and tracks the skeleton, based on the depth image obtained from the RGB-D sensor. A rough 3D model of the actor body is prepared based on the captured video stream. This section describes the skeleton tracking and rough 3D model building in detail.

Skeleton tracking Figure 4(a) shows the N_{J} joints that compose a skeleton, where N_{BP} vectors identified by specific pairs of the joints are referred to as body parts. Each body part can be viewed as a vector formed by the pair of the joints in a specific order. The skeleton of the actor's body in the n-th frame can be extracted and tracked using an existing technique [9]. Assuming a single actor in the scene, we denote the skeleton in the n-th frame by

$$\mathbf{S}_n = \{\mathbf{s}_{n,i} | i = 1, \ldots, N_{\mathrm{J}}\}, \tag{1}$$

where $\mathbf{s}_{n,i}$ is the 3D position of the i-th joint of the skeleton in the RGB-D sensor's coordinate system shown in Fig. 3.

Using the inverse of $\mathbf{M}_n$, which transforms the 3D coordinates in the world coordinate system to the RGB-D sensor's one, we transform the 3D joint positions in $\mathbf{S}_n$ by $\mathbf{s}'_{n,i} = \mathbf{M}_n^{-1}\mathbf{s}_{n,i}$ for all i in $\mathbf{S}_n$ and define the skeleton in the world coordinate system as $\mathbf{S}'_n = \{\mathbf{s}'_{n,i} | i = 1, \ldots, N_{\mathrm{J}}\}$, so as to store the skeleton in the world coordinate system.

We store the n-th video frame, i.e., skeleton $\mathbf{S}_n$, the RGB image I_n, and depth image D_n in the database.

Rough 3D model preparation To render the reenactment of the actor, we prepare a 3D model for generating a novel viewpoint image of the actor. We use a cylinder to represent each body part. Since the heights of the cylinders are trivially determined from the length of the body part vector, all we need to determine the cylinders are their radii. For this, we first find the index of a single representative frame $\hat{n}$ from the recorded video stream and then fit rectangles to the actor's region in the depth image of the representative frame $D_{\hat{n}}$, which can viewed as a projection of the cylinders onto the image plane of the RGB-D sensor.

To obtain radii and heights of the cylinders based on the rectangles that are their projection, the directions of their heights must be perpendicular to the optical axis of the RGB-D sensor. This means that the representative frames should contain the actor's appearance that meet the following requirements: (i) both arms should be away from the body, (ii) the line segments formed by the joints corresponding to both hands should be parallel to the image plane as possible, and (iii) the legs should be uncrossed. These requirements ensure that the representative frame has body parts that are separate from each other as shown in Fig. 4(a), making it easier to build an accurate model of the actor's body. Such a pose may be specifically requested of the actor, but it may also be captured during the normal course of recording. We find such a pose by inspecting the angles formed by the body parts.

As shown in Fig. 4(b), we denote the angles between the torso and the left and right arms in $\mathbf{S}'_n$ by θ_n^{L} and θ_n^{R}, respectively. We also define term $g(\phi_n^{\mathrm{R}}, \phi_n^{\mathrm{L}})$ that gives a positive value when legs are uncrossed as

$$g(\phi_n^{\mathrm{R}}, \phi_n^{\mathrm{L}}) = \begin{cases} 1 & \text{if } \phi_n^{\mathrm{R}} > \phi_n^{\mathrm{L}} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where ϕ_n^{L} and ϕ_n^{R} are the angles between $[1\ 0\ 0]^T$ and the vectors of the left leg and right leg, respectively. This representative frame selection is done in the RGB-D sensor's coordinate system, assuming that the user who capture the video stream does not rotate it very much. The above requirements can be empirically encoded in the criterion

$$E(n) = \theta_n^{\mathrm{L}} a_n^{\mathrm{L}} + \theta_n^{\mathrm{R}} a_n^{\mathrm{R}} + \lambda g(\phi_n^{\mathrm{R}}, \phi_n^{\mathrm{L}}), \tag{3}$$

where a_n^{L} and a_n^{R} are the x-components of the left and right arm vectors, whose lengths are normalized to 1 and λ is an empirically-defined constant. The first and second terms ensure that the arms are lifted away from the torso and that they are parallel to the x-axis of the RGB-D sensor's coordinate system. We obtain the index of the most appropriate frame in sense of the above criterion by maximizing E, i.e.,

$$\hat{n} = \arg\max_n E(n). \tag{4}$$

We then find the rectangle that fit to each body part in $D_{\hat{n}}$, as in Fig. 4(c). The radius r of the cylinder for the body part is then given as the length of the line segments perpendicular to the body part segment. For compensating the slight differences in the body part segment length from frame to frame, we store in the database the radius rate given by r/l for each body part, where l is the length of the body part segment.

4.2. Reenactment stage

In the reenactment stage, we capture an RGB video stream consisting of only RGB images from the viewer's camera. We synthesize the reenactment sequentially for this real-time stream. For each frame, we estimate extrinsic camera parameter $\mathbf{M}_{\mathrm{S}}$ of the viewer's camera again by the SLAM technique, based on the 3D map stored in the database. The reenactment stage then transforms the skeleton to the viewer's camera coordinate system using $\mathbf{M}_{\mathrm{S}}$, builds a rough 3D model based on the radius ratios stored in the database, and colors it based on the appropriate RGB frame in the database. Finally, the reenactment is superimposed on the frame of real-time RGB video stream to be displayed on the viewer's mobile device. This section describes appropriate texture selection and 3D model coloring.

Appropriate texture selection Since our 3D model of the actor is very rough and no color is assigned to it as in Fig. 5 (a), we apply textures to our 3D model so as to improve its visual quality. For a static scene, view-dependent texture mapping proposed by Debevec et al. [10] works well for this purpose by assigning as textures those images which were captured from the viewpoint close to that of the novel image to be synthesized. However, we cannot adopt it naively because the proposed system captures a moving actor and uses only a single RGB-D sensor and thus there are no video frames that capture the same scene at the same time from different viewpoints. Our idea for solving this problem is based on our observation that there still are several video frames that capture a similar actor's pose, which means that we can select a frame such that the joint positions in the selected frame are close to those in the novel image to be synthesized.

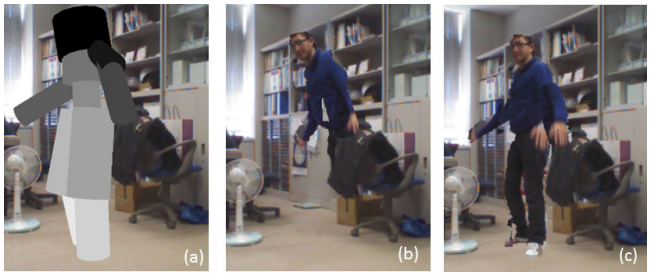

Fig. 5. (a) The cylinder model. Cylinders are colored for visibility. (b) The colored cylinders, without an individual mapping for each cylinder. (c) The colored cylinders corrected to have an individual mapping for each cylinder.

When reenacting the actor's appearance from the k-th skeleton, $\mathbf{S}'_k$, we first transform joint $\mathbf{s}'_{k,i}$ in the world coordinate system into the viewer camera's coordinate system using $\mathbf{M}_{\mathrm{S}}$, giving us $\mathbf{S}^*_k$. We also transform $\mathbf{S}'_n$ for all n into its original RGB-D sensor's coordinate system using $\mathbf{M}_n$, giving us $\mathbf{S}_n$. Since the position of the actor in the world coordinate system varies frame by frame, to make the selection translation invariant, the position of a specific joint is subtracted from the all joint's position so that the specific joint coincide the origin. In this work, we choose the neck joint shown in Fig. 4(a) as the origin. We select the appropriate video frame, of which associated skeleton $\mathbf{S}_n$ in the original RGB-D sensor's coordinate system is closest to the $\mathbf{S}^*_k$ in the viewer camera's coordinate system. To summarize, we find the appropriate frame index $\bar{n}$ by

$$\bar{n} = \arg\min_n \sum_{i=1}^{N_{\mathrm{J}}} \|(\mathbf{s}^*_{k,i} - \mathbf{s}^*_{k,\mathrm{neck}}) - (\mathbf{s}_{n,i} - \mathbf{s}_{n,\mathrm{neck}})\|, \tag{5}$$

where $\mathbf{s}^*_{k,\mathrm{neck}}$ and $\mathbf{s}_{n,\mathrm{neck}}$ are the neck joint positions of $\mathbf{S}^*_k$ and $\mathbf{S}_n$, respectively.

The limitation of this texture selection is its inability to preserve the facial expression of the actor because the selected texture is not always the frame associated with frame index k. However, we consider that it is sufficient to make the actor's motion comprehensible.

3D model coloring Although we selected the appropriate frame for coloring the cylinder, since the poses represented by $\mathbf{S}^*_k$ and $\mathbf{S}_{\bar{n}}$ are not exactly the same, naively projecting the cylinder to the selected RGB frame can lead to inconsistency between the cylinders and the frame as shown in Fig. 5(b). We thus find a projection individually for each cylinder that compensates the actor's poses in $\mathbf{S}^*_k$ and $\mathbf{S}_{\bar{n}}$, and use the projection to determine the color on each 3D point on that cylinder (Fig. 5(c)). Finally, we superimpose the reenactment on the real-time RGB video frame from the viewer's camera.

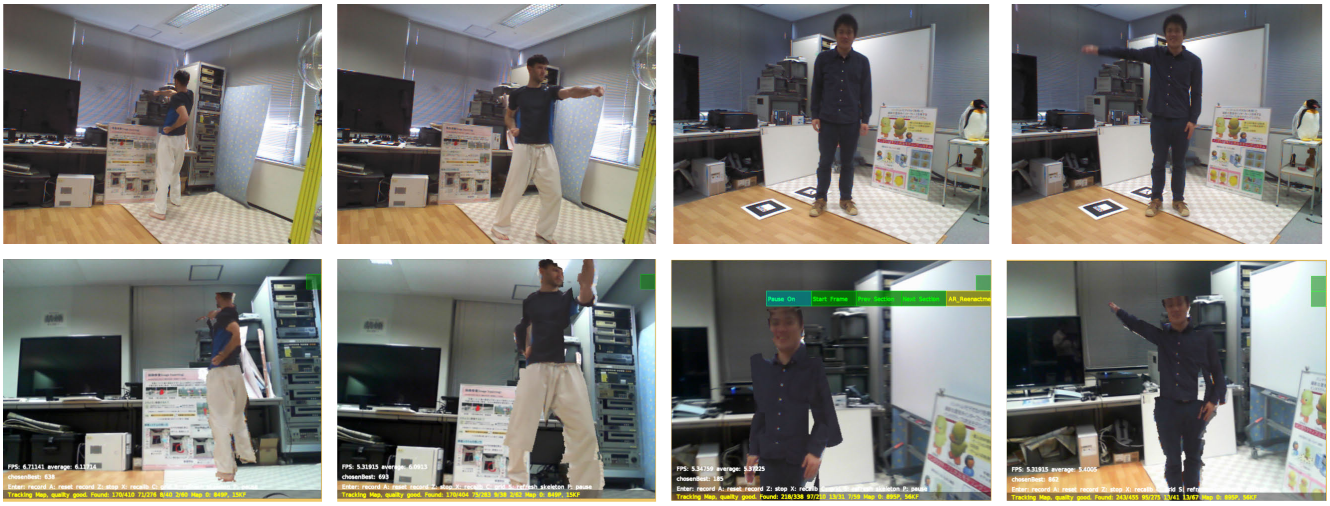

Fig. 6. Top: frames from the videos. Bottom: the same frames from a different angle, reenacted by our proposed system.

5. IMPLEMENTATION

For evaluating our system by user study, we implemented its prototype. We employed Microsoft Kinect as an RGB-D sensor for the capturing stage. For the reenactment stage, our prototype employed Microsoft Surface 2 with an Intel Core i5-4200U processor and 4GB of RAM as a display device and used its embedded camera as the viewer's camera. The intrinsic parameters for Kinect and Surface's cameras are preliminarily calibrated. To obtain the extrinsic camera parameters in both capturing and reenactment stages, we employed PTAMM [8], which is capable of storing the 3D map for later uses, so that we can use the shared world coordinate system in these stages. We set the world coordinate system according to PTAMM. In order to continuously track the actor's joints, we made use of OpenNI and NiTE skeleton tracking. With this skeleton tracker, skeletons contain 16 joints and 11 body parts. Since the coordinate system in which skeletons from OpenNI and NiTE lie and that of PTAMM are different, we calculated the mapping to compensate this difference and apply it to all joint positions in obtained skeletons. We empirically determined the value of λ in Eq. 3 to be 2000.

6. USER STUDY

In the capturing stage for the user study, we captured two videos. For the first video, we captured a Taekwondo form, and for the second, we captured a simple motion. We used different actors for each video in order to verify that the system handles different body types equally. We captured the motions with a fixed-position Kinect, considering that users who capture an RGB-D video stream generally do not move while capturing. Before capturing, we made a 3D map with the PTAMM system. For comparison between the proposed system and conventional video, we also compiled the RGB component of our captured stream into a separate video. Figure 6 shows some example frames from the RGB images, compared with the AR reenactment generated by our proposed system. We employed 15 subjects.

The user study was divided into three parts: the first video, the second video, and the general questions. For each part, subjects were asked to answer the corresponding questions in Fig. 7. The questions asked for both videos were identical. For the first five questions, subjects were asked to watch the 2D conventional video first, and then the reenactment generated by the proposed system. Both were presented on the Microsoft Surface's display. Subjects were allowed to move around while watching the proposed system's reenactment. This part investigates whether the proposed systems aids user comprehension of 3D motion. For the next two questions, a real person attempted to copy the motion of the recorded actor. The subjects first simultaneously watched the conventional video and the motions of the real person; afterward, they simultaneously watched the reenactment and the motions of the real person. Subjects were again allowed to move freely while watching the reenactment. The general questions were designed to show the applicability of the proposed system. The answers were multiple choice and free entry. The multiple choice answers were on a scale of 1 (I strongly think not) to 5 (I strongly think so).

Figure 8 shows the results. The results for the first and second videos are similar, which means that the system handled the difference in actors and motions well. They also demonstrate an improvement in real-person comparison between the conventional video and the proposed system (Q6 and Q7). A possible explanation for this result is because the proposed system can present both the real person and the reenactment in the same display. The weak points of the

For each video:

Q1 Were you able to comprehend the position of the actor in the environment with the conventional video?

Q2 Were you able to comprehend the position of the actor in the environment with the proposed system?

Q3 Did it become easier to comprehend the recorded motion on the proposed system as you moved the perspective?

Q4 Was it easier to comprehend the recorded motion on the proposed system, than on the conventional video?

Q5 Were you satisfied by the quality of the image generated by the proposed system, compared to the conventional video?

Q6 Were you easily able to compare the motions of the real person with the recorded motion on the conventional video?

Q7 Were you easily able to compare the motions of the real person with the recorded motion on the proposed system?

General:

Q1 Is the proposed system more fun to use than conventional video?

Q2 Do you think that this system would be useful for learning specific motions?

Q3 Do you think that this system would be useful for watching performances?

Fig. 7. Questions asked in our user study.

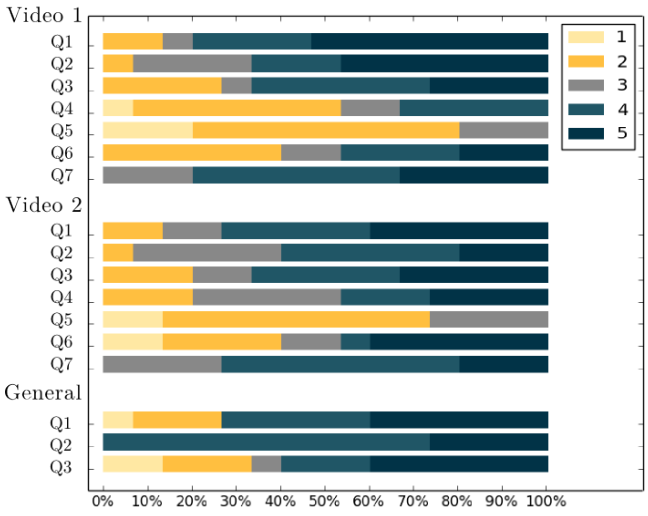

Fig. 8. Evaluation results.

system may be seen in the quality of the rendered images (Q5). Some comments from the subjects were that the output was jittery, that the sillhouette was not smooth, and that if they moved drastically, the image would become corrupted. This also negatively affected the comprehension (Q1 and Q2). However, subjects answered positively regarding the applicability of the proposed system, e.g. for training or learning.

7. CONCLUSION

In this paper, we have presented a novel AR reenactment system using free-viewpoint image generation specifically for human motion. The benefit of our system compared to other similar systems is the requirement for only one single camera during the capturing stage. The system was implemented on a tablet computer, taking advantage of mobile AR. Our user study demonstrated that the proposed system is helpful for better comprehension of an actor's motion as well as for finding differences in the motion of a real human and the reenactment. However, the generated images were not of desired quality. Thus, a good direction for future work would be to improve the image generation process. Another interesting direction would be the incorporation of a human pose estimator that works on an RGB video stream, which would enable us to use the proposed system without depth sensors.

Acknowledgements This work was partially supported by JSPS Grant-in-Aid for Scientific Research Nos. 23240024 and 25540086.

8. REFERENCES

[1] S. Würmlin, E. Lamboray, O. Staadt, and M. Gross, "3D video recorder," in *Proc. Pacific Conference on Computer Graphics and Applications*, 2002, pp. 325–334.

[2] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.

[3] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg, "OmniKinect: Real-time dense volumetric data acquisition and applications," in *Proc. ACM Symposium on Virtual Reality Software and Technology*, 2012, pp. 25–32.

[4] H. Hondori, M. Khademi, L. Dodakian, S. Cramer, and Cristina V. Lopes, "A spatial augmented reality rehab system for post-stroke hand rehabilitation," in *Proc. Conference on Medicine Meets Virtual Reality*, 2013, pp. 279–285.

[5] S. Henderson and S. Feiner, "Augmented reality in the psychomotor phase of a procedural task," in *Proc. IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 191–200.

[6] U. Yang and G. Kim, "Implementation and evaluation of "just follow me": An immersive, VR-based, motion-training system," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 3, pp. 304–323, 2002.

[7] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice, "YouMove: Enhancing movement training with an augmented reality mirror," in *Proc. ACM Symposium on User Interface Software and Technology*, 2013, pp. 311–320.

[8] R. Castle, G. Klein, and D. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *Proc. IEEE International Symposium on Wearable Computers*, 2008, pp. 15–22.

[9] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[10] P. Debevec, C. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. ACM SIGGRAPH*, 1996, pp. 11–20.