Choreographing Amateur Performers Based on Motion Transfer between Videos

Kenta Mizui[1], Makoto Okabe[2], and Rikio Onai[3]

Department of Informatics, The University of Electro-Communications, 2-11, Fujimicho, Chofu, Tokyo, 182-0033, Japan[1]
Department of Computer Science, The University of Electro-Communications, 2-11, Fujimicho, Chofu, Tokyo, 182-0033, Japan[23]
JST PRESTO[2]
mizui@onailab.com[1], m.o@acm.org[2], onai@cs.uec.ac.jp[3]

Abstract. We propose a technique for quickly and easily choreographing a video of an amateur performer by comparing it with a video of a corresponding professional performance. Our method allows the user to interactively edit the amateur performance in order to synchronize it with the professional performance in terms of timings and poses. In our system, the user first extracts the amateur and professional poses from every frame via semi-automatic video tracking. The system synchronizes the timings by computing dynamic time warping (DTW) between the two sets of video-tracking data, and then synchronizes the poses by applying image deformation to every frame. To eliminate unnatural vibrations, which often result from inaccurate video tracking, we apply an automatic motion-smoothing algorithm to the synthesized animation. We demonstrate that our method allows the user to successfully edit an amateur's performance into a more polished one, utilizing the Japanese sumo wrestling squat, the karate kick, and the moonwalk as examples.

Keywords: shape deformation, video editing.

Fig. 1. Our method uses a pair of videos of a physical activity, such as a karate kick, as input; one is the target video of an amateur performance (a), and the other is a video of a corresponding professional performance (b). We extract timing and pose information from the professional performance and use it to choreograph the amateur performance (c).

1 Introduction

We may imagine ourselves as professional dancers or action stars, but since our physical skills are usually inadequate, the resulting performances are poor.

This situation is commonplace in the movie industry. When an actor's physical abilities are limited, a professional (or stunt) performer is often substituted, but the face must be carefully concealed. To solve this problem, we are interested in developing a technique for editing a performer's motions to make him or her appear more dexterous (Fig. 1).

A number of existing approaches for editing a video of a human performance are based on three-dimensional (3D) reconstruction of the performer. For example, Jain *et al.* proposed a method for reconstructing the shape of a performer's body by fitting it to a morphable body model, which can be edited interactively (e.g., to be thinner or fatter). However, our goal is to edit the speed and pose of a performance more dynamically. The video-based character method is closely related to our technique, but synthesizes a video sequence of a new performance of a pre-recorded character [2]. Although this allows the user to freely edit a 3D human performance and move the camera at will, expensive equipment, including a special studio and multiple cameras, is required.

Since performance editing in 3D space is powerful but expensive, two-dimensional (2D) video editing is a reasonable and popular alternative. Video texture approaches allow an input video or its internal objects to be edited to play in a seamless infinite loop, or with a user-specified temporal behavior [3, 4]. However, since these techniques rearrange only the temporal order of the video frames, it is impossible to change the speed or pose of an object. The method of Scholz *et al.* allows a video object to be moved, rotated, or bent, but only low-dynamic edits have been demonstrated [5]. The method of Weng *et al.* is closely related to our work [6], but the motion of a video object is edited only in user-specified key frames: i.e., the deformations are applied in a sparse set of frames, and the system then propagates the deformations to the rest of frames. Their method requires contour tracking of a video object, a time-consuming task for the user, which our technique avoids. Also, our method is example-based, allowing the user to quickly specify a desired motion.

To make the awkward motions of an amateur performer appear more polished, we propose a method for quickly and easily choreographing a video of the amateur performer by utilizing a video of a corresponding professional performance. Our method allows an amateur performance to be edited by transferring the timings and poses extracted from the professional performance. Our technique is purely 2D, and enables the user to edit human performances at low cost, without the large datasets of 3D models, a special studio, or multiple cameras. We successfully applied our method to amateur performances involving the Japanese sumo wrestling squat, the karate kick, and the moonwalk, making them all look more polished.

2 System Overview

Figure 2 shows an overview of our system. The system utilizes videos of an amateur performance (Fig. 2a) and a corresponding professional performance (Fig. 2b) as inputs. The karate kick is used in this example. In our experiments,

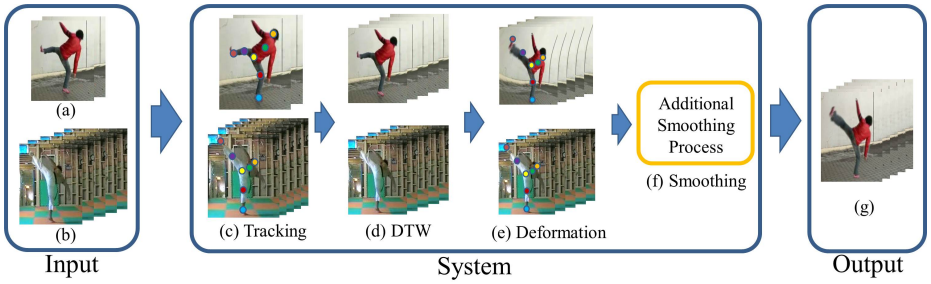


Fig. 2. System overview.

we assume that each video is captured with a single, fixed camera, and the background image is known. Our method consists of the following four stages. First, the user extracts the trajectories from the two input videos (Fig. 2c). Our system tracks a motion and associates its trajectory with a user-specified body part. Since the automatic tracking is incomplete, and often generates incorrect trajectories, our system supports a user interface to interactively adjust them. Second, the system applies dynamic time warping (DTW) to synchronize the amateur's motions with the professional's (Fig. 2d). DTW finds temporal correspondences between the amateur and professional trajectories, and temporally aligns video frames. Third, the system deforms the amateur's pose in every frame, so that the trajectories of both performers are spatially matched (Fig. 2e). Finally, when significant unnatural vibrations appear in the choreographed animation, the user can optionally request the system to apply automatic smoothing. Our optimization algorithm removes the differences between the motions of choreographed and professional animations (Fig. 2f). The tracking and DTW stages are semi-automatic, while the deformation and smoothing stages are fully automatic.

3 Semi-automatic Video Tracking

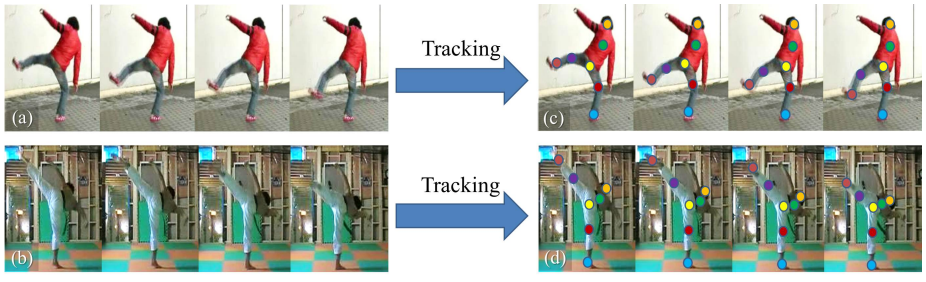


Fig. 3. (a) Amateur frames. (b) Professional frames. (c and d) The colored dots represent the tracking of each body part.

We first analyze the motions of both the amateur and professional, and extract the trajectories of the corresponding body parts (Fig. 3). In this example, we are tracking seven body parts: the head, back, hip, left and right knees, and left and right feet. We rely on semi-automatic video tracking because an automatic algorithm will not always provide correct results. Even the latest techniques, such as the particle video method [7], compute inaccurate, noisy trajectories for fast motions. On the other hand, as we shall demonstrate, only a small amount of user interaction is necessary to efficiently modify such trajectories. Since we track the center of each body part instead of its contour [8], the user's burden is smaller than in methods based on rotoscoping.

3.1 User Interface

Figure 4 shows the flow of our semi-automatic video tracking. The user begins to track the left foot by clicking on it (Fig. 4a). The system automatically computes the trajectory, indicated by the blue dots in the top part of Fig. 4, based on an optical flow method [9]. However, the blue dots gradually slide outside the foot area. To correct this, the user drags and modifies the last tracker position, indicated by the red dot in Fig. 4b. The system then updates the trajectory, indicated by the orange dots in the bottom part of Fig. 4, using our optimization algorithm. In this way, the user can efficiently and accurately extract a trajectory with minimal burden.

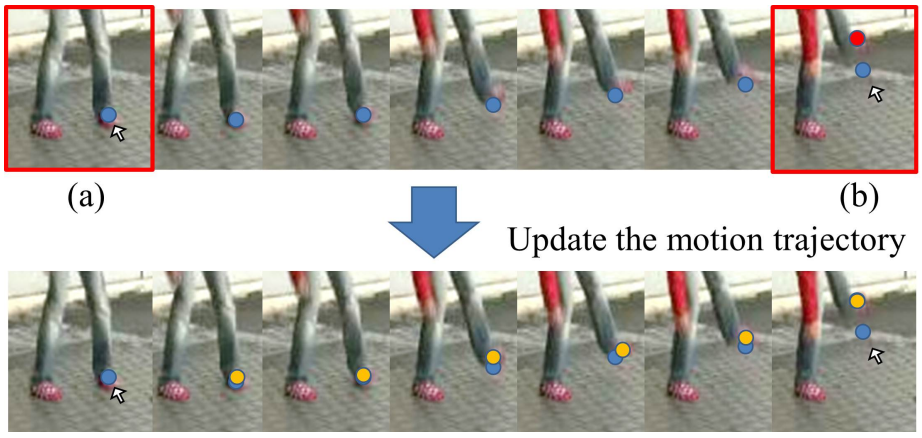


Fig. 4. User interface for the semi-automatic video tracking.

3.2 Algorithm

To develop the user interface described above, we based our algorithm on the following three policies. The resulting trajectory 1) must pass through the user-specified control points, indicated by the red dots in Fig. 4b, 2) must follow the sampled optical flow under the trackers in every frame as much as possible, and

3) must be as smooth as possible. We formulate this optimization problem in a least-squares sense: we calculate the resulting trajectory R from

$$R = \arg\min_{X} E(X), \tag{1}$$

$$E(X) = \lambda \sum_{i \in I^c} (x_i - x_i^c)^2 + \sum_{j \in I^c} \sum_{i}^{n} w_i^j (x_i - t_i^j)^2 + \sum_{i}^{n-1} w_i^a (v_i - v_{i-1})^2, \tag{2}$$

$$v_i = x_{i+1} - x_i, \tag{3}$$

where $X = \{x_1, \ldots, x_n\}$ is the set of tracker positions, I^c is the set of indices of the frames with user-specified control points, $X^c = \{i \in I^c | x_i^c\}$ is the set of user-specified control point positions, and $T^j = \{t_1^j, \ldots, t_n^j\} (j \in I^c)$ is the set of tracker positions, calculated at each control point. The first, second, and third terms correspond to the three policies. The first term relates to the user-specified control points, the second term compels the trackers to try to follow the underlying optical flow, and the third term is the smoothness term. λ, w_i^c, and w_i^a are the respective weights for the terms. λ is set equal to 10,000 in our experiment.

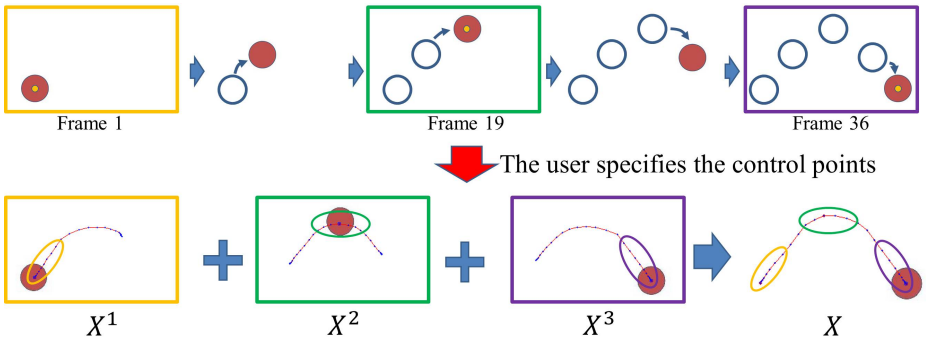


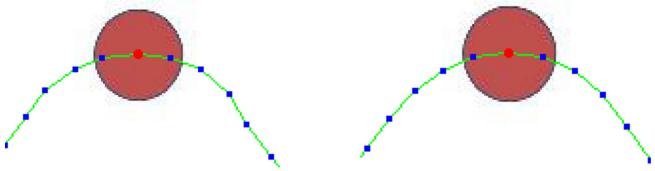
Fig. 5. The idea of our algorithm is to accurately calculate the position of the tracker. When the user specifies the control points, our system calculates the position of the tracker from each control point (bottom). We obtain the precise position of the tracker by mixing these trajectories.

The second term is illustrated in Fig. 5. In this sample video sequence, the red circle moves along an arc, as shown in the top part of Fig. 5. The user specifies three control points, indicated by the orange dots in Frame 1, Frame 19, and Frame 36. Our system then calculates the trajectories T^1, T^{19}, and T^{36}, starting from each control point and sampling the underlying optical flow. The calculated positions of the trajectories are accurate and reliable near the control points around the ellipse in each figure, but inaccurate and unreliable far from the control points. Therefore, we want to merge the reliable parts of those trajectories

and obtain accurate trajectories for the trackers. To achieve this, we set w_i^j to be inversely proportional to the distance from each control point.

Without the third term, the first and second terms introduce unnatural vibrations into the trajectory (Fig. 6a), resulting in awkward motions in the final synthesized animation. Thus, we introduce the third term to smooth the trajectory. We want to keep the accelerations as small as possible, while preserving the significantly large accelerations (Fig. 6b). w_i^a is designed according to Eq. 4.

$$w_i^a = e^{-|(x_{i+1}-x_i)-(x_i-x_{i-1})|^5/10000} \tag{4}$$



(a) Without Smoothing (b) Using Smoothing

Fig. 6. Tracking results w/o smoothness term.

4 Motion Transfer and Rendering

Synchronization of Motion Speed via DTW After extracting the trajectories, we applied DTW between the amateur and professional trajectories and synchronized the amateur's motions with those of the professional [10]. For example, Fig. 7 shows amateur and professional videos of the Japanese sumo wrestling squat, with durations of 84 frames and 96 frames, respectively. Using seven trackers, we obtained seven motion trajectories. Since each tracker represents a 2D position, we obtained (2×7)-dimensional signals for the videos, with durations of 96 and 84 frames, respectively. After applying DTW between these signals, we obtained the lengthened (i.e., slowed down) synchronized amateur video shown in Fig. 7. DTW usually succeeded, but often included poor synchronization in the results. In such a case, we used the "time remapping" user interface of Adobe After Effects CS5 to modify it.

Deformation After synchronizing the motions, we deformed the amateur's pose frame-by-frame to match the professional's pose. We applied feature-based image warping [11] to accomplish this. As Fig. 8 shows, we created a skeleton for each frame by connecting the neighboring trackers, and then applied the algorithm between the amateur and professional skeletons. In the resulting image, the amateur's leg was as high as that of the professional.

We also tried as-rigid-as-possible (ARAP) shape manipulation [12]. ARAP is effective for preserving the original shape and area of the input image. However,

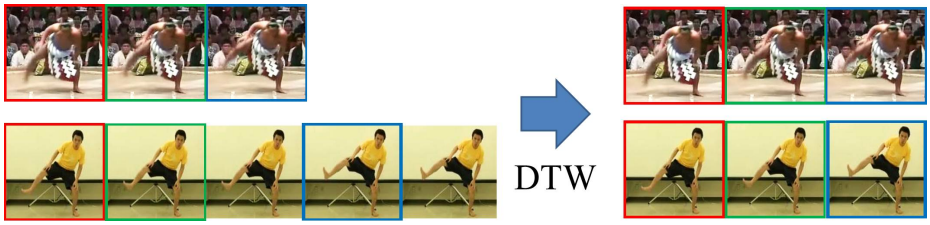


Fig. 7. Synchronization of the amateur's motions with those of the professional.

we often had to deform the image drastically, and in such a case, ARAP sometimes yielded awkward results because of the rigidity constraints (Fig. 8d). On the other hand, feature-based image warping employs several parameters, which enabled us to control the rigidity of the deformation and obtain better results (Fig. 8c).

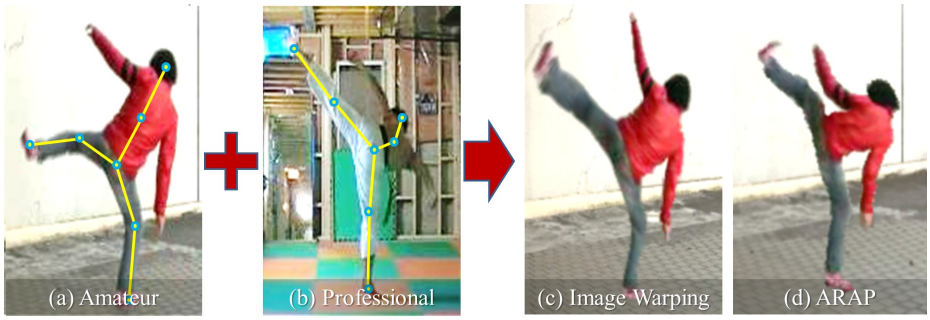


Fig. 8. (a) Amateur frame. (b) Professional frame. (c) Deformation results of feature-based image warping. (d) Deformation results of ARAP.

Smoothing Unnatural Vibration Inaccurate motion trajectories cause visible artifacts, which take the form of unnatural vibrations in the synthesized animation. This effect was especially conspicuous in our moonwalk results. However, it is almost impossible for the user to create perfect trajectories in the semi-automatic video tracking process. Therefore, we introduced an automatic optimization technique to remove such unnatural vibrations and refine the resulting motions. We based our algorithm on the assumption that both the choreographed amateur and professional should have the same motions. Hence, if the choreographed amateur exhibits unnatural vibrations, these can be obtained as motion differences between the choreographed amateur and professional. We therefore calculated these differences, and then deformed each frame of the choreographed amateur video until the difference disappeared. Figure 9 shows the details of our smoothing process. The choreographed amateur's ear

moved downward, but the professional's ear moved to the right (Figs. 9a, 9b, and 9c). The computed optical flow is shown in Fig. 9d. Our algorithm computed the difference between the optical flow vectors (the red arrow in Fig. 9e), and deformed the choreographed amateur's frame in accordance with it (Fig. 9f). We repeated this procedure for every frame, using all the trackers defined in the video tracking process. In this procedure, we used image deformation based on moving least squares (MLS) [13], which is computationally efficient and effective for preserving the original shape.

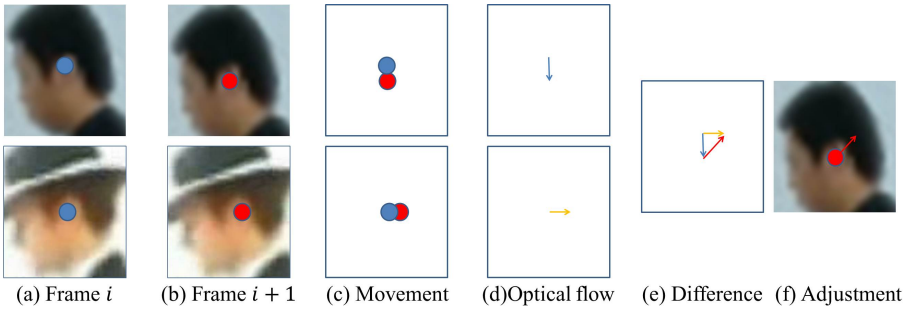


Fig. 9. Smoothing unnatural vibration.

Rendering The deformation algorithm not only deforms the performer but also distorts the background (Fig. 10a). To recover the correct background, we applied the alpha-blending method to create the final composite. Since the background image was given in advance (Fig. 10b), we first performed the background subtraction and obtained a mask for the performer. We then applied the same deformation to the mask (Fig. 10c). Finally, we computed a trimap from the mask and applied the Poisson matting method [14] to create the composite (Fig. 10d).

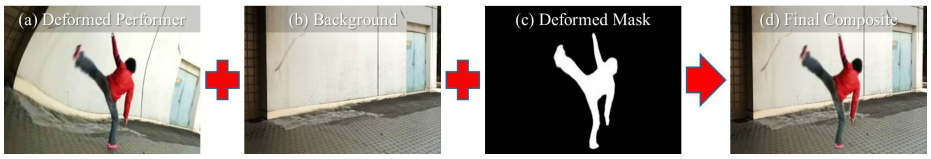


Fig. 10. Rendering the final result with background subtraction and alpha blending.

5 Result

We applied our technique to 1) the Japanese sumo wrestling squat, 2) the karate kick, and 3) the moonwalk. For each of these, we prepared a pair of videos, one

amateur and one professional. We captured the former ourselves, and obtained the latter from YouTube (http://www.youtube.com/). The results are shown in our website (https://sites.google.com/a/onailab.com/mizui/english).

The Japanese sumo wrestling squat The amateur performer could not raise his leg high enough because his hip joint was not as flexible as that of the professional (Fig. 11). Our method successfully deformed the amateur's pose and raised his leg as high as that of the professional. However, we also acquired a visible artifact: the deformed leg grew a bit thicker. This is because our image-warping technique is not effective for preserving area.

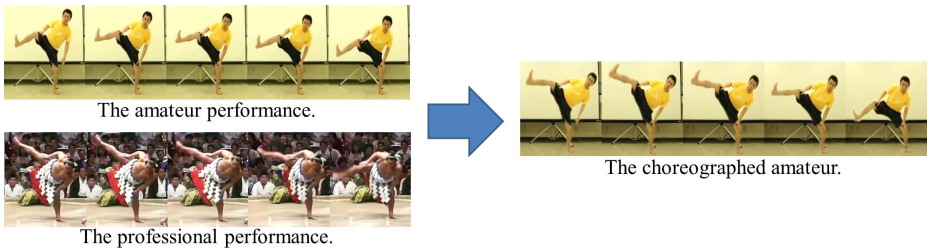


Fig. 11. Results for the Japanese sumo wrestling squat.

The Karate kick Our method successfully edited the low, slow karate kick of the amateur (Fig. 12) to create a high, quick karate kick video. The side-by-side comparison in our supplementary video confirms its similarity to the professional performance.

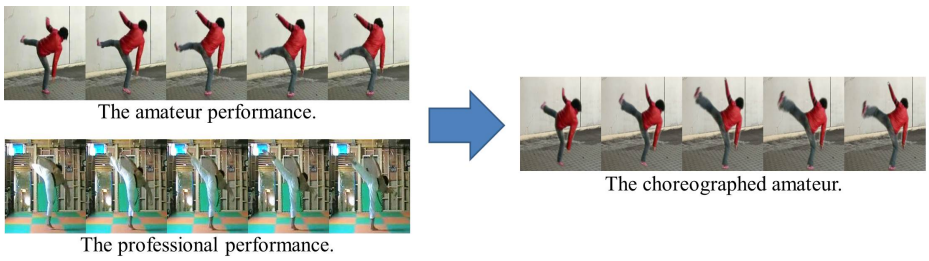


Fig. 12. Results for the karate kick.

The moonwalk Our method was also used to modify an amateur's moonwalk performance, which initially looked as if he was just walking slowly backwards (Fig. 13). After motion synchronization and pose deformation, the resulting animation looked awkward, and visible artifacts were apparent: the choreographed performance had unnatural vibrations throughout the video sequence.

Our smoothing algorithm successfully removed these vibrations and synthesized a smooth animation. The comparison in our website video confirms this.

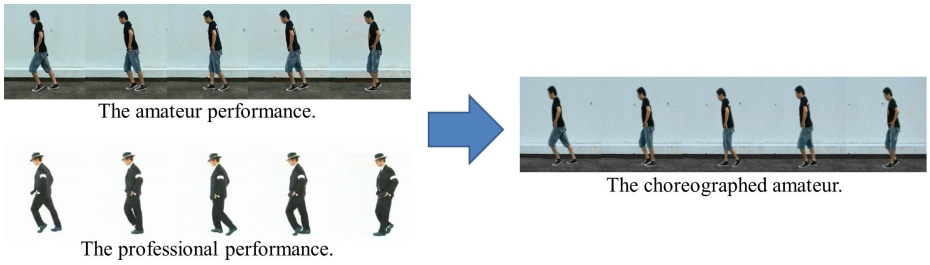


Fig. 13. Results for the moonwalk.

5.1 User Study

We conducted a subjective evaluation to investigate whether or not subjects could find visible artifacts created by our system. We asked 15 subjects to evaluate the system, all of them university students who were not familiar with our technique. We showed them the original videos and choreographed amateur performances, and each subject rated the quantity of visible artifacts via a seven-point Likert scale, where 1 meant "no artifact" and 7 meant "a lot of artifacts". In the case of the karate kick, the original video was rated at 2.20 ± 1.83, and the choreographed video at 5.27 ± 1.44, where $\pm$ denotes the standard deviation. This shows that our finished video still had visual artifacts, but 2 out of 15 subjects rated the choreographed video as having no visible artifacts. 14 out of 15 subjects noticed that the choreographed performer kicked higher and more quickly than in the original video. Also, some subjects pointed out that the leg had grown thicker and the foot was slipping.

5.2 Limitation and Future Work

The quality of the video tracking is important for the success of our method. Several sophisticated 3D video tracking techniques have been proposed recently [15, 16]. Since these might reduce the user burden and also improve the quality of the resulting animation, we intend to try them in the future.

Since our video editing procedure is strictly 2D, we cannot handle occlusions in the video. Figure 14 shows a sample situation. The two legs are overlapped and we must edit each leg independently. However, our method does not support this, resulting in the awkward deformation shown in Fig. 14c. To support such editing, we must extend our technique, e.g., by introducing image segmentation and hole-filling algorithms. Another possibility is to use a depth sensor such as Kinect, which will provide depth information and facilitate image segmentation.

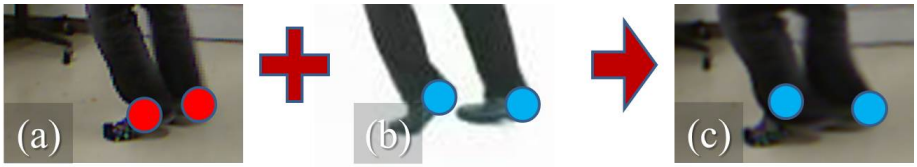


Fig. 14. We cannot separate the two overlapped legs.

References

[1] Jain, A., Thormählen, T., Seidel, H.-P., and Theobalt, C.: Moviereshape: Tracking and reshaping of humans in videos, ACM Trans. Graph. 29, 5, (2010)

[2] Xu, F., Liu, Y., Stoll, C., Tompkin, J., Bharaj, G., Dai, Q., Seidel, H.-P., Kautz, J., and Theobalt. C.: Video-based characters: creating new human performances from a multi-view video database, In ACM SIGGRAPH 2011 papers (2011), 32:1-32:10

[3] Schödl, A., Szeliski, R., Salesin, D., and Essa, I.: Video textures, In Proceedings of ACM SIGGRAPH 2000, Annual Conference Series, ACM SIG- GRAPH (2000), 489-498.

[4] Schödl, A., Essa, I.: Controlled animation of video sprites, In Proc. of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2002), 121-127.

[5] Scholz, V., El-Abed, S., Seidel, H.-P., and Magnor, M. A.: Editing object behavior in video sequences, CGF28, 6 (2009), 1632-1643

[6] Weng, Y., Xu, W., Hu, S., Zhang, J., and Guo, B.: Keyframe based video object deformation, in Proc. International Conference on Cyberworlds 2008 (2008), 142-149

[7] Sand, P., and Teller, S.: Particle video: Long-range motion estimation using point trajectories, In Proc. of CVPR 2006 (2006) 2195-2202

[8] Agarwala, A., Hertzmann, A., Salesin, D. H., and Seitz, S. M.: Keyframe-based tracking for rotoscoping and animation, ACM Trans. Graph. (2004), 23(3) , 584-591

[9] Zach, C., Pock, T., and Bischof, H.: A duality based approach for realtime tv-l1 optical fow, In DAGM-Symposium (2007), 214-223

[10] Rao, C., Gritai, A., Shan, M., and Syeda-mahmood, T.: View-invariant alignment and matching of video sequences, In Proc. of ICCV 2003 (2003), 939-945

[11] Beier, T., and Neely, S.: Feature-based image metamorphosis, In Proc. of SIGGRAPH '92 (1992), 35-42

[12] Igarashi, T., Moscovich, T., and Hughes, J. F.: As-rigid-as-possible shape manipulation, In ACM SIGGRAPH 2005 Papers (2005), 1134-1141

[13] Schaefer, S., Mcphail, T., and Warren, J.: Image deformation using moving least squares, ACM TOG 25, 3, (2006) 533-540

[14] Sun, J., Jia, J., Ang, C.-K., and Shum, H.-Y.: Poisson matting, In ACM SIGGRAPH 2004 Papers (2004), 315-321

[15] Wei, X., and Chai, J.: Videomocap: Modeling physically realistic human motion from monocular video sequences, ACM Transactions on Graphics 29, 4 (2010).

[16] Vondrak, M., Sigal, L., Hodgins, J. K., and Jenkins, O.: Video-based 3D Motion Capture through Biped Control, ACM Transactions on Graphics (Proc. SIGGRAPH) 2012 (2012).