Motion Sculptures: Automating Artistic Visualization of Shape and Time

by

Xiuming Zhang

B.Eng., National University of Singapore (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author .....
Department of Electrical Engineering and Computer Science
July 13, 2018

Certified by .....
William T. Freeman
Thomas and Gerd Perkins Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Motion Sculptures: Automating Artistic Visualization of Shape and Time

by

Xiuming Zhang

Submitted to the Department of Electrical Engineering and Computer Science
on July 13, 2018, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We present a method for automatically visualizing complex human and object motion via 3D *motion sculptures* – a representation that conveys the 3D structure swept by the human's body as it moves through space. Given only a monocular RGB video as input, our algorithm computes the motion sculpture by estimating the object's 3D geometry over time, and then renders the motion sculpture under different rendering styles such as sculpture material, scene lighting, and floor reflections. We develop a 3D-aware image-based rendering approach to insert motion sculptures into a synthetic scene or back into the original video. This results in high-quality artistic visualizations of motion. Because motion sculptures are 3D, they can be viewed from arbitrary viewpoints and even physically printed. As such, they may reveal space-time information that is undetected by the naked eyes and allow the viewer to interpret how different parts of the object interact over time. Our method automates the process of motion sculpture creation, making this manual process typically done only by professional artists accessible to novice users and applicable to standard videos. We show results on various scenes involving complex motions such as sports actions and dancing.

Thesis Supervisor: William T. Freeman
Title: Thomas and Gerd Perkins Professor of Electrical Engineering and Computer Science

Acknowledgments

First of all, I would like to thank my advisor, Prof. William T. Freeman, for his unstinting support and invaluable guidance. I could not ask for a better advisor: on the one hand, Bill has been granting me the freedom to explore research that I found exciting; on the other hand, Bill also constantly inspired me with his crazy creative ideas during our meetings. It is Bill's "I don't like magic" that continually motivated me to trace why things worked (or, for most of the time, did not work). I am truly grateful.

This thesis would not have been possible without my collaborators' wisdom and efforts. I would like to thank Dr. Tianfan Xue, Dr. Tali Dekel, Dr. Andrew Owens, Jiajun Wu, and Prof. Stefanie Mueller for their valuable input into this thesis work. Thanks also go to Zhoutong Zhang, Chengkai Zhang, and Prof. Joshua B. Tenenbaum for their contributions to the work not presented in this thesis.

I am grateful to the many people who have made the Vision and Graphics Neighborhood (VGN) such a fun place to work in. Especially, I would love to thank, in no particular order, Vickie Ye, Dr. Katie Bouman, Dr. Donglai Wei, Dr. Jun-Yan Zhu, Dr. Guha Balakrishnan, Prafull Sharma, Ruizhi Liao, Maz Abulnaga, Rujian Chen, Dr. Dian Yu, Dr. Adrian Dalca, Yu Wang, Prof. Miaomiao Zhang, Xavier Puig, David Bau, Wei-Chiu Ma, Hang Zhao, Dr. Bolei Zhou, Dr. Wenzhen Yuan, Shaoxiong Wang, Dr. Randi Cabezas, Sue Zheng, Tao Du, Jie Xu, Jimmy Wu, Hunter Lang, and Dr. Shaiyan Keshvari, among others. I would love to extend my gratitude to other friends outside the VGN and outside MIT for making my life colorful.

I would also like to express my gratitude and appreciation to Prof. B.T. Thomas Yeo, Prof. Mert R. Sabuncu, and Prof. Elizabeth C. Mormino for supervising my bachelor's thesis and getting me started on research.

Finally, I thank my parents and my family for their wise words, tender care, unconditional love, and unwavering support of all my decisions. Thank you.

To my beloved 奶奶 and 姥爷, whom I miss dearly.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Complicated actions, such as athletic events or ballet, can be difficult to convey in a static image. Methods for motion visualization, such as chronophotography, stroboscopic photography, and multi-exposure photography [4, 5], have had a long history in photography and art. These techniques, however, operate entirely in 2D and are therefore unable to convey the underlying 3D geometry of motion. As such, they tend to create results that are cluttered in the presence of occlusion, because the depth ordering among objects is not preserved (see Figure 6-5). Moreover, they often require special capturing procedures (*e.g.*, working only with a plain, black background) or lighting equipment.

In this thesis, we present an algorithm that automatically visualizes human and object motion captured in a monocular RGB video. Our visualization is based on rendering *motion sculptures*—the spatial-temporal structure carved by the target as it moves through space. Our algorithm allows us to transform a standard video into a physical sculpture of motion (Figure 1-2d) with minimal user input, or to render a virtual motion sculpture (Figure 1-2b and c). For these virtual renderings, we can composite the sculpture either with the source video contents or apply stylistic effects, such as choosing the sculpture material, scene lighting, and background (Figure 1-2c). Combining our renderings with the source video contents results in an artistic visualization that conveys the target's trajectory and reveals how its 3D shape evolves over time. Furthermore, because of their 3D nature, motion sculptures can be directly

Figure 1-1: Motion sculptures from professional artists [1, 2] inspired our work.

explored in 3D – a user can navigate around a motion sculpture and view it from alternative viewpoints, revealing information about the motion that is inaccessible from the original viewpoint.

Our approach is inspired by recent artistic work that visualizes 3D motion trails [2, 6, 7] (see Figure 1-1). These renderings, however, are produced by professional artists and rely on special recording procedures (such as motion capturing). While our method results in similar effects as these artistic methods, it is applicable to standard RGB videos and requires minimal user input, which makes it accessible to novice users. Furthermore, because our approach is based on automatic measurements of the motion and shape, rather than impressions of an artist, we can thoroughly evaluate the resulting motion sculptures. That is, we study the methodology and challenges in creating motion sculptures, which to the best of our knowledge, has not been documented so far.

Motion sculpture also relates to traditional video summarization techniques such as image montage [8, 9], generally stitching together foreground objects captured at different timestamps. As in stroboscopic photography, these summarization methods do not preserve the actual depth ordering among objects, and therefore cannot illustrate the 3D trajectory of objects.

Depth-based summarization methods overcome some of these limitations using

geometric information provided by depth sensors. Shape-time photography [3], for example, conveys occlusion relationships by showing, at each pixel, the color of the surface that is the closest to the camera over the entire video sequence. More recently, Klose *et al*. introduced a video processing method that uses per-pixel depth layering to create action shot summaries [10]. While these methods are useful for presenting the 3D relationships in a small number of sparsely sampled images, such as by showing where an object moved over the course of the video, they are not well suited to visualizing continuous motion. Moreover, these methods are based on depth maps, and thus provide only a "2.5D" reconstruction that cannot be easily viewed from multiple viewpoints as in our case.

Automatic generation of motion sculptures poses two major challenges. First, we need to solve the inverse 2D-3D problem, *i.e*., recover the object's shape, pose, and motion from the source video. In Chapter 3, we describe a novel joint optimization formulation that exploits the temporal coherency of human motion as constraints to reduce ambiguities. Solving the optimization provides us with an initial motion sculpture.

The second challenge is to blend the estimated motion sculpture with the original video contents in a visually pleasing way. This requires careful analysis, because every error in the sculpture may show up as visual artifacts in the final rendering. To obtain high-quality, artifact-free results, in Chapter 4, we develop an image-based rendering (IBR) technique that refines the sculpture and inserts it into the source video, while preserving proper depth orderings.

Our generated motion sculptures on diverse videos gracefully reveal the beauty and vividness of human motion in sports and dancing. In Chapter 6, we present qualitative and quantitative studies to validate our technical innovations and to justify our design choices. While we focus on human motion in this thesis, we also show that our approach can be extended to general objects with a parameterized shape model.

This thesis draws on a collaborative work, currently under review, in which I am the lead author. The other collaborators are Dr. Tianfan Xue, Dr. Tali Dekel, Dr. Andrew H. Owens, Jiajun Wu, Prof. Stefanie Mueller, and Prof. William T. Freeman.

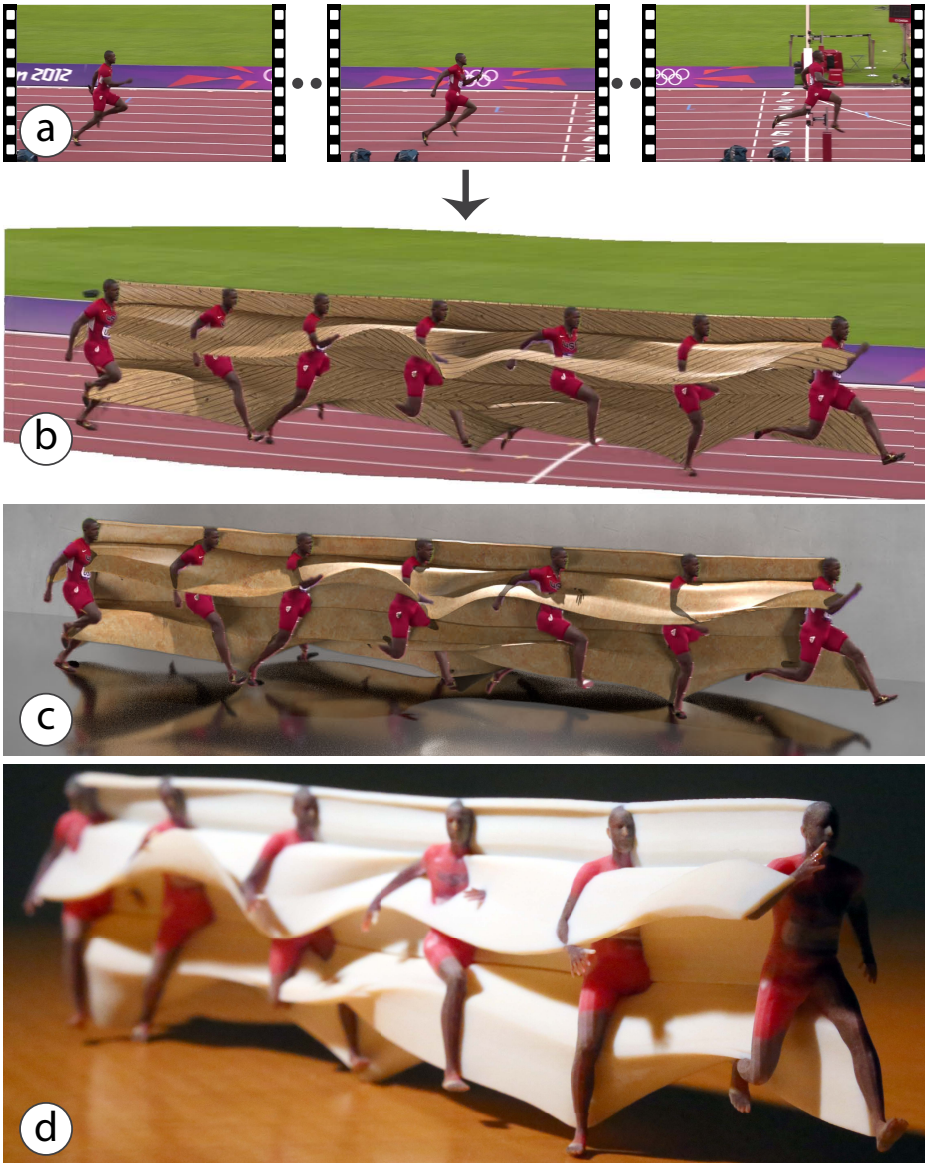


Figure 1-2: We present a method for visualizing motion in a video via *motion sculptures*—an artistic rendering of the 3D path that an object traces as it moves through space. Our algorithm transforms a standard RGB video depicting a complex action, captured by a static or moving camera (a), into a physical motion sculpture (d) with minimal user input, or renders it with the source video contents (b) in many styles, *e.g.*, different sculpture materials, scene lighting and background (c). Our image-based rendering approach seamlessly blends the sculpture with the moving object, producing an artistic visualization. The 3D nature of motion sculptures reveals information about the motion, such as the sinusoidal motion of the arms, which is not readily visible in the input video frames. This can be seen in sharper relief when the motion sculpture is 3D-printed or viewed from alternative viewpoints.

Chapter 2

Related Work

Besides video summarization methods mentioned above, our work is also related to manually produced artistic renderings and physical visualizations.

2.1 Automating Artistic Renderings

A range of tools have been developed to aid users in creating artist-inspired motion visualizations [11, 12, 13, 14]. DemoDraw [12], for instance, allows users to generate drawing animations by physically acting out the action, then motion capturing them, and finally applying different stylizing filters.

Our work continues along this line of work; *i.e.*, we provide an interactive system that facilities the creation of motion sculptures. While our work is inspired by artistic work that visualizes 3D motion trails [2, 6, 7, 15], these renderings are produced by professional artists and require special recording procedures (such as motion capturing) or advanced computer-generated imagery (CGI) skills. In this thesis, we opt to lower the barrier to entry—our system is applicable to standard videos, which makes the production of motion sculptures less cost-intensive and more accessible to novice users.

The most closely related work to ours in this category is ChronoFab [15], a system for creating motion sculptures from 3D animations. However, a key difference is that ChronoFab requires both the subject's full 3D shape and its motion as input,

while our system directly takes a video as input and estimates the shape and motion. Capturing real human motion is challenging, which limits ChronoFab's practical uses. In comparison, our model can estimate real human motion well and is applicable to natural videos.

2.2 Motion Effects into 2D Images

Illustrating motion in a single image has also been studied in the context of non-photorealistic renderings. For example, there are methods for creating stroboscopic copies of a moving object and motion lines [16, 17] for CGI animations and cartoons [18]. Schmid *et al.* designed programmable motion effects as part of the rendering pipeline by aggregating triangle meshes over time and using this data structure to produce stylized blurring and stroboscopic images [13].

A system for adding motion effects using only a single image was proposed by Teramoto *et al.* [19]. Similar effects have also been used by Baudisch *et al.* for creating animated icon movements [20]. For time-lapse videos, Bennett *et al.* developed a method for simulating a virtual shutter, which adds motion tails (motion paths) into the frames [21].

2.3 Physical Visualizations

Recent research has shown great progress in physical visualizations and has demonstrated their benefits in allowing the user to efficiently access information along all dimensions [22, 23, 24]. MakerVis [25] is a tool that allows users to quickly convert their digital information into physical visualizations. ChronoFab [15], in contrast, addresses some of the challenges in converting digital data to physical, for example, connecting parts that would otherwise float in midair. Our motion sculptures can be physically printed as well; however, our focus is in rendering and seamlessly compositing them into the input videos, rather than optimizing the procedure for physically printing them.

Chapter 3

Generating Motion Sculptures

Our algorithm turns a video depicting a complex human action into a motion sculpture summary. An overview of our pipeline is illustrated in Figure 3-2: given a monocular RGB video, we first estimate the object's shape in all the frames. This involves tracking annotated 2D keypoints using an off-the-shelf keypoint detector (Figure 3-2a), then using these keypoints in a joint optimization to recover the object's shape, pose, and trajectory over time (Figure 3-2b). Given the predicted shape, we create an initial estimate of the 3D motion sculpture by sweeping a predefined surface contour of the object across space (Figure 3-2c).

3.1 2D Keypoint Detection

We use 2D keypoints as a mid-level representation for video-level shape estimations. We first detect annotated pose keypoints in each frame independently using OpenPose [26]. While per-frame detections are typically accurate, the inherent ambiguity in the problem sometimes leads to temporal inconsistency, such as the flipping of the left and right shoulders between adjacent frames. We address this by imposing smoothness across all frames using a Hidden Markov Model (HMM), computing the maximum marginal likelihood estimate of joint i's location at time t, x_i^t:

$$\arg\max_{x_i^t} \int_{x_i^{t-1}} p\left(x_i^{t-1}, y_i^1, \ldots, y_i^{t-1}\right) p\left(y_i^t \mid x_i^t\right) p\left(x_i^t \mid x_i^{t-1}\right), \tag{3.1}$$

where the emission probability $p\left(y_i^t \mid x_i^t\right)$ is the heatmap predicted locally at frame t, and the transition probability $p\left(x_i^t \mid x_i^{t-1}\right)$ is a bivariate Gaussian centered at x_i^{t-1} with a standard deviation of three pixels. In cases where some of the person's joints are not detected locally, we linearly interpolate their locations from neighboring frames before running the HMM.

3.1.1 User Interaction

The most common type of errors in human keypoint detection is that the left-right pair of joints flip. When such errors occur in multiple consecutive frames, the smoothness prior cannot filter them out, in which case, we use minimal inputs from the user. Specifically, a window of our graphical user interface (GUI; to be elaborated in Chapter 5) is dedicated to collecting sparse *binary* user responses to whether the detected joints are *all* correct or not for a given frame, and runs a very similar version of the HMM inference that enforces smoothness over the clip. Approximately three or four user clicks per 100 frames are enough to obtain accurate detections in all our results.

3.2 Reconstructing Humans in Motion

With the detected 2D keypoints in hand, we turn to the problem of recovering the parametric 3D model, namely, the human's shape, pose, and trajectory over time. We formulate the problem as a joint optimization problem using keypoint reprojection losses and imposing shape, pose, and smoothness priors. Our formulation can be seen as an extension of SMPLify [27], a single-image 3D human pose and shape estimation algorithm, to videos.

We use SMPL [28] for the parametric 3D model and jointly solve for the human's shape $\beta \in \mathbb{R}^{10}$, per-frame pose $\theta^t \in \mathbb{R}^{72}$, and per-frame translation $T^t \in \mathbb{R}^3$ for each

of the N frames. The loss function is

$$\mathcal{L}\left(\{T^t\},\{\theta^t\},\beta\right) = \sum_{t=1}^{N} \mathcal{L}_{\text{data}}\left(T^t,\theta^t,\beta\right) + \alpha_1 \mathcal{L}_{\text{spatial}}\left(\theta^t,\beta\right) + \alpha_2 \sum_{t=1}^{N-1} \mathcal{L}_{\text{temporal}}\left(T^t,T^{t+1},\theta^t,\theta^{t+1},\beta\right), \tag{3.2}$$

where α_i are constant weights, $\mathcal{L}_{\text{data}}$ is the local evidence, defined to be the sum of squared keypoint reprojection distances, and $\mathcal{L}_{\text{spatial}}$ is a per-frame spatial prior. This prior, defined in [27], contains a human pose prior (a mixture of Gaussians of the pose vectors), penalty for mesh interpenetration, and joint bending priors $\mathcal{L}_{\text{bend}} = \sum_i \mathbb{1}_i e^{\theta_i} + (1-\mathbb{1}_i)e^{-\theta_i}$, where θ_i is the bending angle of joint i, and indicator function $\mathbb{1}_i = 1$ when $\theta_i < 0$ corresponds to natural bending ($= 0$ otherwise).

Finally, $\mathcal{L}_{\text{temporal}}$ encourages the motion sculpture reconstruction to be smooth; it penalizes changes in the human's global translations (Equation 3.4), local vertex locations (Equation 3.5), and pose parameters (Equation 3.6). More specifically,

$$\mathcal{L}_{\text{temporal}} = \lambda_1 \mathcal{L}_{\text{global}} + \lambda_2 \mathcal{L}_{\text{local}} + \lambda_3 \mathcal{L}_{\text{rotation}}, \tag{3.3}$$

$$\mathcal{L}_{\text{global}}\left(T^t,T^{t+1}\right) = \left\|T^t - T^{t+1}\right\|_2^2, \tag{3.4}$$

$$\mathcal{L}_{\text{local}}\left(\theta^t,\theta^{t+1},\beta\right) = \left\|V\left(\theta^t,\beta\right) - V\left(\theta^{t+1},\beta\right)\right\|_F^2, \tag{3.5}$$

$$\mathcal{L}_{\text{rotation}}\left(\theta^t,\theta^{t+1}\right) = \left\| \begin{bmatrix} \cos\left(\theta^t\right) - \cos\left(\theta^{t+1}\right) \\ \sin\left(\theta^t\right) - \sin\left(\theta^{t+1}\right), \end{bmatrix} \right\|_2^2, \tag{3.6}$$

where $V(\cdot)$ are the vertices' local 3D coordinates given the pose and shape, and the λ's are constants that set the three losses to be roughly the same order of magnitude.

Intuitively, $\mathcal{L}_{\text{global}}$ requires the motion sculpture's global trajectory to be smooth; $\mathcal{L}_{\text{local}}$ further requires vertices of the human mesh to translate smoothly; $\mathcal{L}_{\text{rotation}}$ imposes additional rotational smoothness in the parameter space, which is necessary for producing natural pose evolution. The relative weights of these losses are given by hand-chosen constants α_1 and α_2.

3.2.1 Optimization

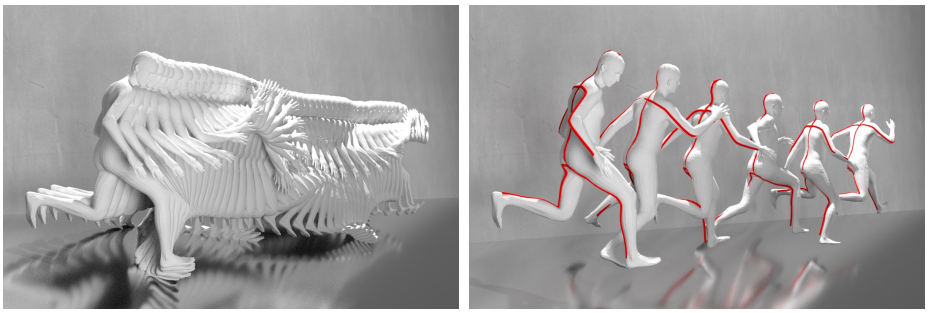
We use a two-stage optimization procedure, in which we first ignore the temporal losses, optimizing only the per-frame losses: $\mathcal{L}_{\text{data}} + \alpha_1 \mathcal{L}_{\text{spatial}}$. This provides a good initialization and allows us to address the "pose-flipping problem": the tendency of the joint optimization getting stuck with flipped facing directions when the person is captured in a side view, often due to the inherent ambiguity (as in *Run, Forrest, Run!* shown in Figure 6-2b). To avoid such local minima, the algorithm tries both directions that the model could face, when the left-right shoulder or hip joints are less than 100 pixels apart (a heuristic whose value does not matter too much, as a lower-than-necessary threshold merely costs some extra time without affecting the final results), and then initializes the joint optimization with the pose that gives a lower $\mathcal{L}_{\text{bend}}$. We minimize this loss function using a non-linear least squares approach [29].

Our reconstruction offers a human mesh for each frame, whose pose smoothly evolves across space and time (Figure 3-1a). To generate the initial motion sculpture, we extract a surface contour (Figure 3-1b) from the mesh by projecting the 3D skeleton onto the back surface and connecting the surface contour across all frames (Figure 3-1c). Here we use surface contours, instead of skeletons, because they better align with the occlusion boundary in 2D images (or can be easily made so, as will be shown in Section 4.1).

3.3 Handling Camera Motion

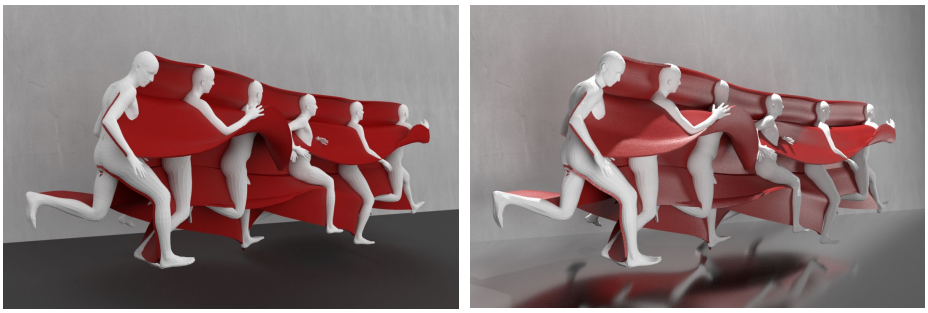
Many interesting human actions unfold over a large area (*e.g.*, Figure 1-2), so we extend our algorithm to take camera motion into account while summarizing an action.

One approach to handle camera motion is to stabilize the background, *e.g.*, by registering each frame to the panoramic background before applying our algorithm to the stabilized video. This works well when the background is mostly planar, and the registration can be well modeled by homography (see Figure 1-2). However, when the background is complex, and the objects' depths vary, we see significant artifacts



(a) Estimated 3D shapes

(b) Sample contours



(c) Motion sculpture without localized lighting or reflections

(d) Motion sculpture with localized lighting and reflections

Figure 3-1: (a) A collection of estimated 3D geometries for *Olympic* (see Figure 1-2). (b) 3D contours (marked in red over representative shapes) are obtained by projecting the 3D skeleton onto the back surface. (c) An initial motion sculpture is generated by joining the estimated 3D contours from all frames into a single surface. (d) The motion sculpture is rendered with shading and reflections to effectively convey the 3D structure.

due to motion parallax.

Looking back, we notice that $\{T^t\}$ in Equation 3.2 are essentially the human's relative translations w.r.t. the camera; $\{T^t\}$ only become the human's global translation with a static camera. Thus, for general cases, we can first estimate the human's translations relative to the moving camera by solving the same optimization, and then compute the camera trajectory with an off-the-shelf structure-from-motion (SfM) software [30]. Finally, we offset $\{T^t\}$ by the camera trajectory to obtain the human's global translations.

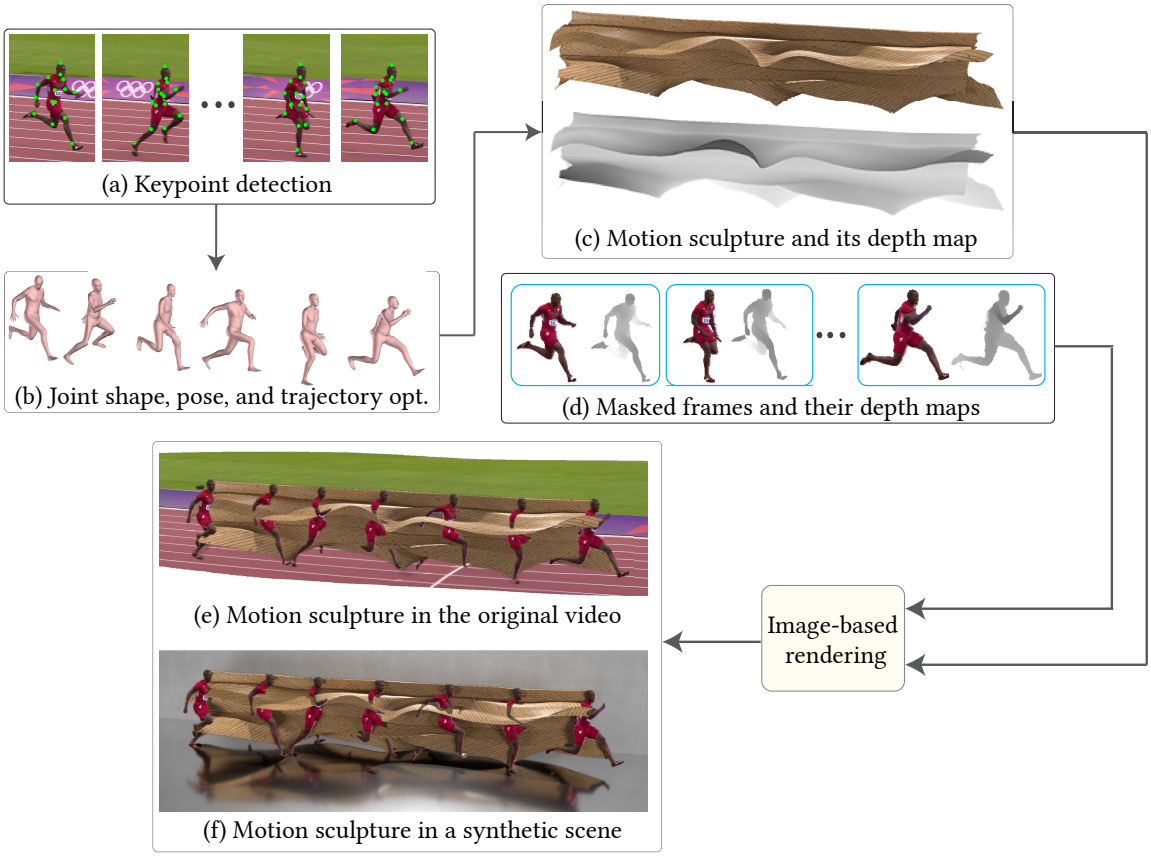


Figure 3-2: Generating motion sculptures: given an input video, we first extract 2D keypoints for each input frame. The detected keypoints are then used as input into the optimization step, in which we jointly solve for the shape, pose, and trajectory of the human over time (b). An initial motion sculpture is generated from the estimated 3D geometries (c), which is then refined to better align with the masked frames (d). Finally, we render the motion sculpture with the moving object while preserving depth orderings. The sculpture can be embedded either into the original video (e) or in a synthetic scene (f); our renderings combine reflections, shading, and different materials to convey the underlying 3D geometry of motion captured by the sculpture.

Chapter 4

Augmenting Videos with Sculptures

Our goal is to generate a high-quality, rich, and vivid visualization of the estimated motion sculpture. Meanwhile, we are also interested in maintaining fidelity to the source video, *i.e.*, preserving the visual appearance of the human in motion. While motion sculptures can be fully rendered and visualized in 3D, the estimated 3D geometries provide only rough outlines of the human shape and lack fine structural details. Furthermore, full 3D rendering requires texture mapping—a challenging problem that becomes intractable when parts of the human body are not covered by the 3D model (*e.g.*, the ballerina's hair and skirt in Figure 4-1a).

Instead, we take an image-based rendering (IBR) approach to preserve the richness and high-frequency information in the video (*e.g.*, object texture). Clearly, superimposing naively the rendered sculpture image onto the video results in a cluttered visualization that completely disregards the 3D spatial relationships between the sculpture and the object (Figure 4-1b). In contrast, our method produces a depth-preserving composite (Figure 4-1c; Figure 3-2e and f).

Despite the significant progress made in 3D reconstruction from monocular videos, high-quality blending still remains challenging, as it requires pixel-level alignment between the projection of the estimated geometry and the input video. We next demonstrate how we achieve high-quality blendings with noisy 3D estimations.

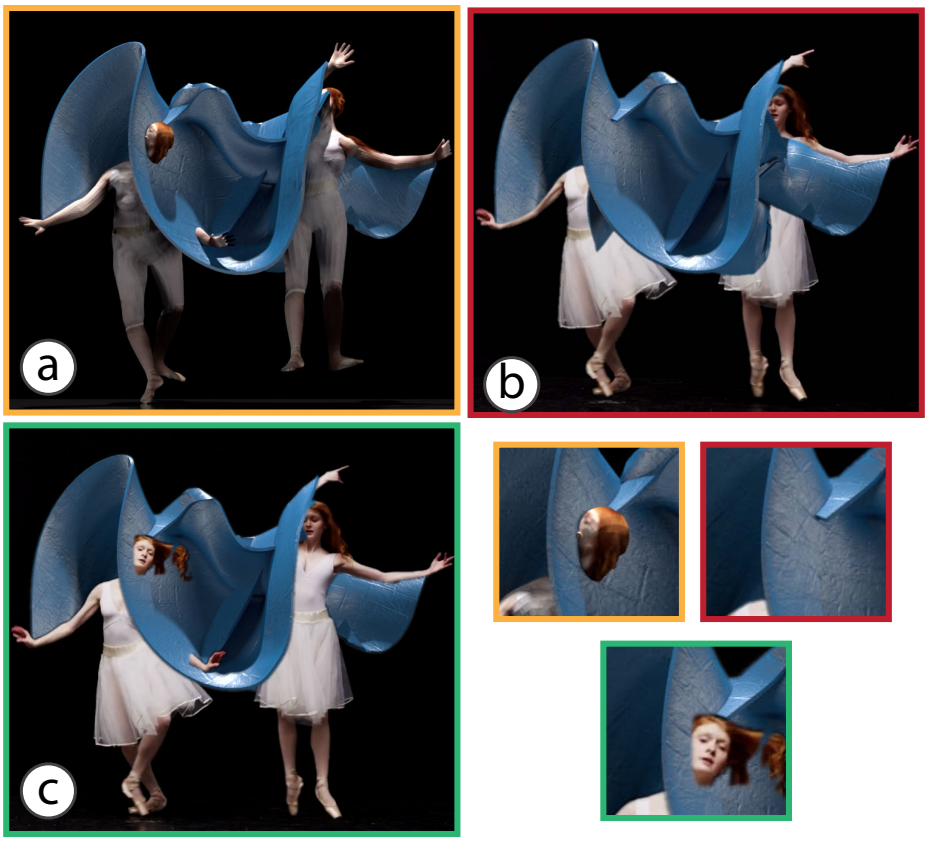


Figure 4-1: (a) Full 3D rendering of the reconstructed human body; this visualization lacks important appearance information, *e.g.*, the subject's hair and dress. (b) Simple composite of the sculpture back onto the scene; this approach discards information about depth ordering. (c) Our IBR-based method reveals accurate 3D relationships and rich appearance information, while not requiring full texture mapping (c).

4.1 Aligning 3D Sculptures with Input Videos

Figure 6-3 shows the importance of the joint optimization in producing smooth and accurate 3D motion sculptures (Figure 6-3b) in contrast to per-frame optimizations (Figure 6-3a). In *Ballet-1* (Figure 4-2), however, joint optimization alone is insufficient for obtaining an artifact-free composite. Figure 4-2a shows that the sculpture generated from the jointly optimized results, albeit smooth, still fail to achieve pixel-level alignment with the original images, and such small misalignment errors show up as visual artifacts. To eliminate such artifacts, we refine our initial 3D sculptures as follows.

We first compute dense optical flow [31] between the foreground mask and the

projected, rendered 3D silhouette and then use it to warp the image coordinates of the 3D surface contours (that form the sculpture). We then back-project this warped surface contour to 3D, assuming the same depth as before editing. Essentially, we are editing the sculpture in the 3D x- and y-axes such that its boundary, when projected to 2D, aligns well with the original 2D images. To compensate for some minor jittering introduced by this editing, we smooth each dimension with a kernel of form $[\ldots, 2^{-1}, 2^{0}, 2^{-1}, \ldots]$ and width 15. As for the sculpture color, it is either user-specified or the average of the original pixel values across time.

To automatically extract foreground masks of sufficient quality for this work, we first run Mask R-CNN [32] on each frame to produce loose foreground masks, which we then erode to produce the corresponding overtight masks. Combining the loose and overtight masks produces trimaps, which are then fed to kNN matting [1] to produce the final masks.

4.2 Approximating Objects' Depth Maps

We need the object's depth maps to respect the spatial relationships between the sculpture and the object in blending. However, the imperfect 3D geometry estimation of human shapes provides us with only rough depth maps, again often misaligned with the original images. More importantly, they may not fully cover the foreground object, *e.g.*, the ballerina's hair and skirt in Figure 4-2c.

To resolve this issue, we use the same flow-based method described above to warp these rendered depth maps to match their corresponding foreground masks. When a pixel from a foreground mask has no depth value after warping, we copy the depth of its nearest neighbor. By doing so, we generate an approximated depth map for each foreground human. We show an example in Figure 4-2c, where the corrected depth maps respect the occlusion boundary in the original images and provide depth values for her hair and skirt, which are not covered by the 3D human model. This allows them to appear from the back of the sculpture (compare the hair in Figure 4-2a and c).

4.3 Rendering and Compositing

We wish to emphasize the structure of the sculpture when it is embedded in a video or an image. Working in 3D grants us the freedom to render the sculpture in a synthetic 3D scene with appealing lighting, shadows, and reflections. We then composite these rendered passes with the original video according to the refined depth maps.

More specifically, we first render the RGB projection and depth image of the sculpture using the recovered camera (Figure 3-2c), along with depth maps for the human (Figure 3-2d). We then composite together the sculpture's rendered image and the original image by selecting, for each pixel, the value that is the closest to the camera. Due to the noisy nature of the object's depth maps, we use a simple Markov random field with Potts potentials to enforce some smoothness during this composition.

We also provide an alternative, artistic rendering styles, where an artificial background wall and a reflective floor are used. To improve the viewer's ability to perceive the sculpture's shape, we render the background with shadows cast by the object and sculpture, and we show reflections on the floor (as can be seen in Figure 1-2c). We achieve these effects by coarsely texturing the 3D human with UV mapping computed by simple ray casting. By using IBR for the human and coarse texturing for its reflections, we produce high-quality 3D rendering without the need for actually solving the challenging problem of texture mapping.

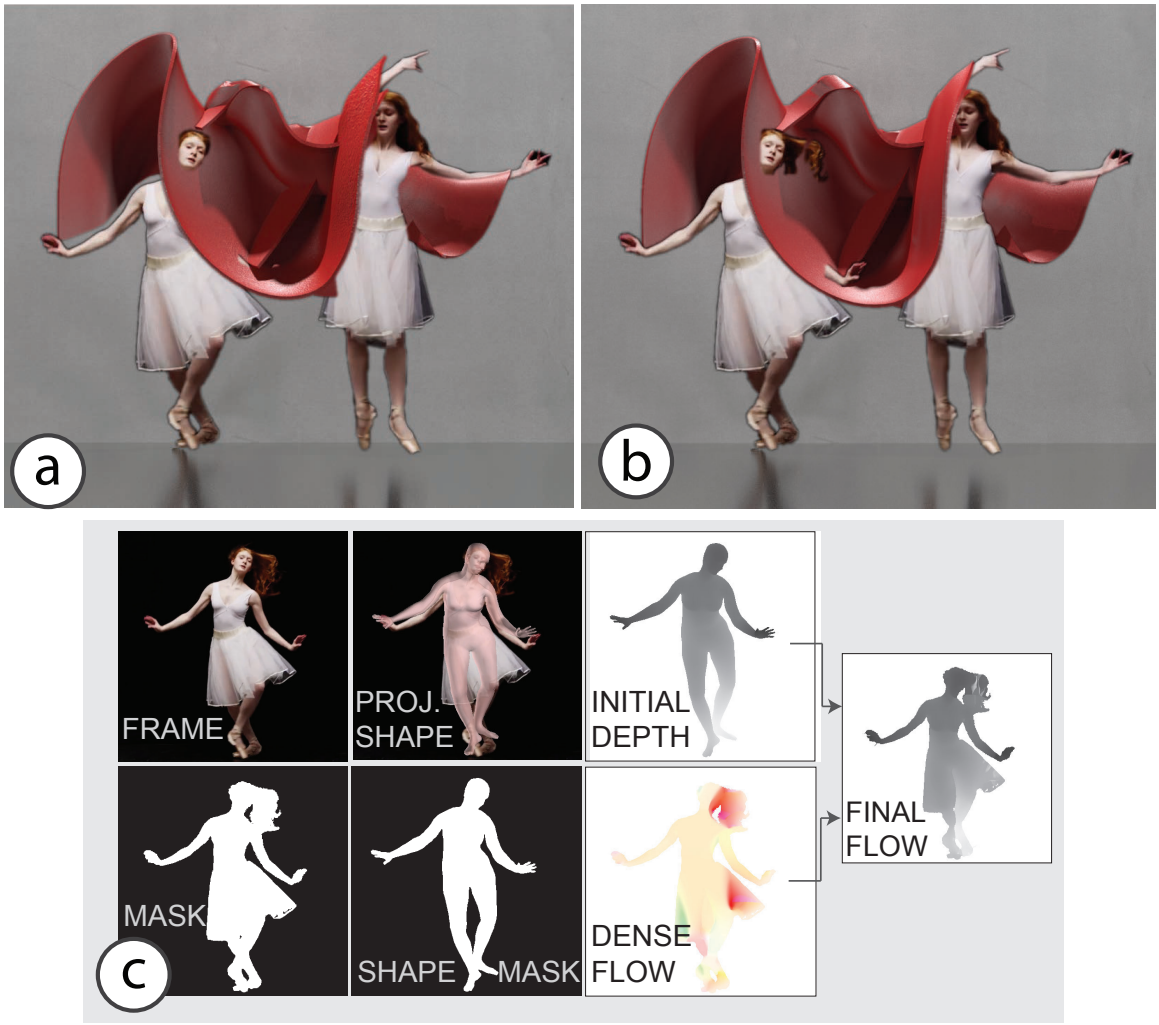


Figure 4-2: (a) Rendering generated using our joint optimization (shape, pose, and translation are jointly estimated over time). (b) Our results with flow-based refinement (c): we compute a dense flow field to align between the 2D silhouette, (c)-left, and the projected 3D silhouette. We refine the initial depth and the 3D sculpture using the computed flow to make them consistent with foreground images. For example, using flow we propagate the depth values to the skirt, although it is not modeled by the initial 3D shape (c).

Chapter 5

Graphical User Interface

We design and implement a graphical user interface (GUI) to facilitate the user in producing and exploring the motion sculptures. To generate a motion sculpture, users start by loading a standard RGB input video into the system. After loading the video, the system automatically detects the human body's pose in the video and displays it as 2D keypoints overlaid on the video frames (Figure 5-1a). The user then browses the detection results and confirms, on a few randomly selected frames, that the keypoints are correct by clicking the "left/right correct" button; these labeled frames serve as anchors to our algorithm and are used to filter out incorrect detections. After labeling, the user hits "Done Annotating," which triggers the system to generate the motion sculpture. This is an offline process that includes estimating the human's shape and poses across the frames.

5.1 Available Options

After processing, users load the motion sculpture into the system, virtually explore it in 3D, and customize its design by controlling various appearance settings, illustrated in the following subsections.

5.1.1 3D Model

The user loads the motion sculpture mesh into the system, which is then displayed in the second window in our GUI (Figure 5-1b). The user can navigate around the motion sculpture in 3D, *i.e.*, view it from alternative novel viewpoints. This often reveals information about shape and motion that are not available from the original camera viewpoint and facilities the understanding of how different body parts interact across space and time.

5.1.2 Appearance

Users can customize the following settings for rendering the motion sculpture in different styles (Figure 5-1c):

- **Scene settings.** The user can choose to render the sculpture in a synthesized scene or embed it back into the original video contents. This is controlled by the "Artistic Background" button. For synthetic scenes (*i.e.*, "Artistic Background" is on), we use a glossy floor and a simple background that is lightly textured for realism. To improve the viewer's ability to perceive the sculpture's shape, we render the background with shadows cast by the object and sculpture, and show reflections on the floor (as can be seen in Figure 1-2c).

- **Lighting.** Our lighting settings include one area light on each side of the scene and one point light from the top. The user may turn on any combination of the three light sources (shown on the left menu).

- **Material.** Users can control the texture and color of the sculpture by choosing one of six different materials: leather, marble, metal, tarp, tiles, and wood. See the radio buttons under "Sculpture Material" on the left side menu.

- **Human figures.** In addition to the 3D shape, our motion sculptures include a number of images of the human in motion (similar to sparse, stroboscopic photographs). This allows the viewer to associate the 3D structure of our model with the corresponding parts of the human. We allow the user to customize the

number of frames (which we call "figures") that get inserted into the sculpture via the "Keyframe Density" slider. These frames are sampled at a uniform rate from the input video.

- **Body parts.** Users can decide which parts of the body should be used to form the motion sculpture. The user can choose to render the left/right arm, left/right leg, or any combination of them. By default, the complete skeleton will be rendered.

- **Transparency.** A slider is dedicated to controlling the transparency of the motion sculpture, hence allowing the user to see through the sculpture and avoid self-occlusion.

These tools grant users the ability to customize their visualizations and select the rendering settings that best convey the space-time information captured by motion sculptures.

5.2 Example Explorations

To demonstrate the applicability of our approach to a wide range of different motions and input videos, we collected video material for complex actions including ballet, tennis, running, and dancing. We downloaded most of the videos from the web (*e.g.*, YouTube, Vimeo, and Adobe Stock), and captured two videos ourselves using a Canon 6D (*Jumping* and *U-Walking*).

For each example, we embedded the motion sculpture back into the source video as well as into a synthetic background. We also rendered the sculpture from a novel viewpoint, which often reveals information that is not perceptible from the captured viewpoint.

In *Jumping* of Figure 6-1, for example, the novel-view rendering shows the slide-like structure carved out by the arms during the jump.

An even more complex action, *i.e.*, a cartwheel, is presented in *Cartwheel* of Figure 6-1. For this example, we make use of the "Body Parts" option in our GUI and

decide to only visualize the legs to avoid clutter. Viewing the sculpture from a top view reveals that the girl's legs cross and switch their depth ordering—a complex interaction that is hard to comprehend even by repeatedly playing the original video.

In the *U-Walking* sequence of Figure 6-1, the motion sculpture depicts the person's motion in depth; this can be perceived also from the original viewpoint thanks to the shading and lighting effects that we selected from the different rendering options.

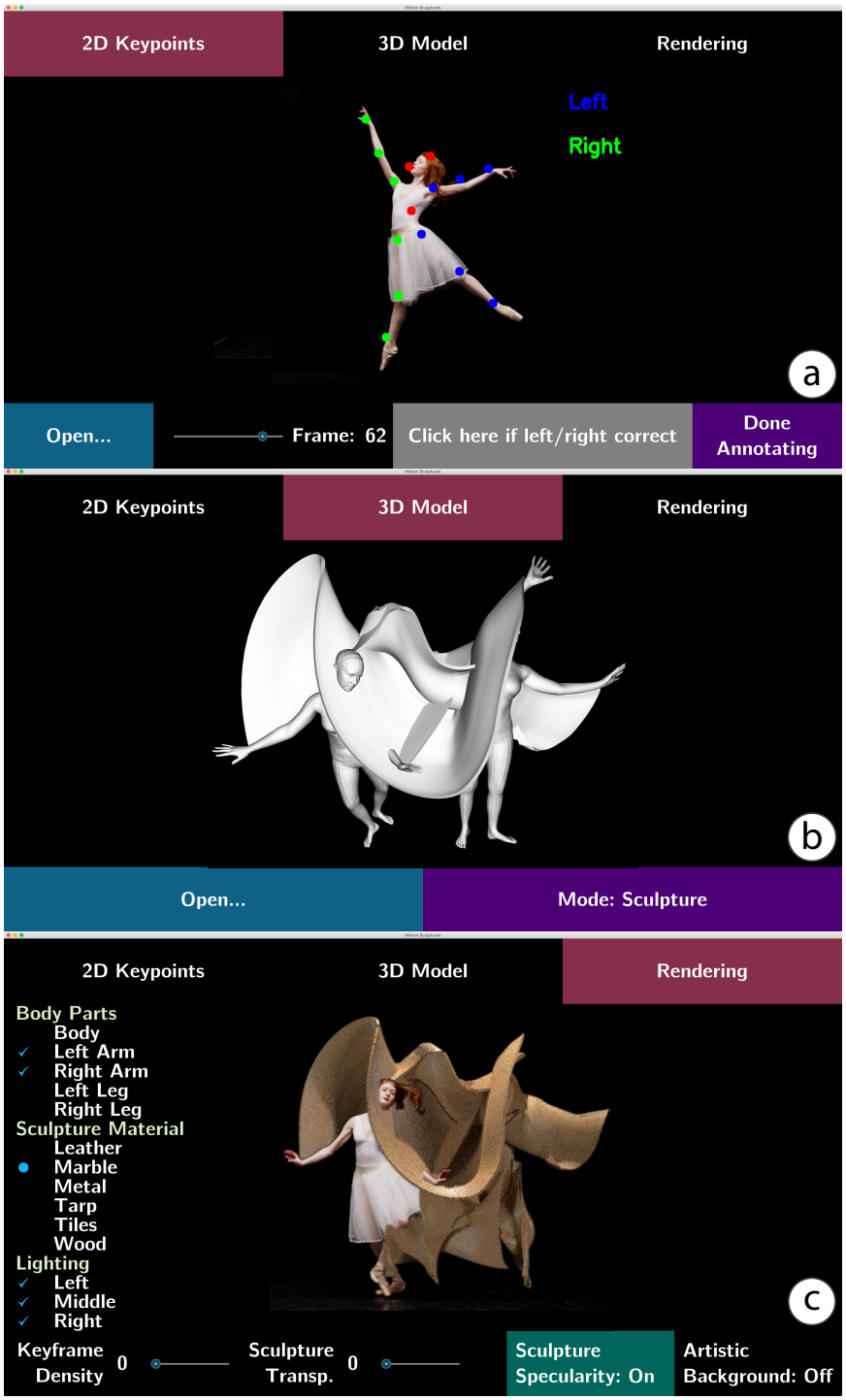


Figure 5-1: Motion sculpture user interface. Our interface allows the user to fix keypoint detection errors with a few clicks (a). After generating the motion sculpture, the user can navigate around it in 3D (b), and customize the rendering by selecting body parts, lighting, keyframe density, sculpture materials, transparency, specularity, and the scene background (c).

Chapter 6

Results

We evaluated our method on a variety of videos involving diverse and complex human motion, *e.g.*, ballet, tennis, running, and fencing. Most of the videos were downloaded from the internet (*e.g.*, YouTube, Vimeo, and Adobe Stock), and two of them were captured by us using a Canon 6D (*Jumping* and *U-Walking*).

Our motion sculptures come with six different materials (downloaded from Poliigon[1]) and two possible backgrounds—the original video background or a synthetic scene (see Section 4.3). For the latter, we used a glossy floor with a small amount of roughness and a simple background that is lightly textured for realism. For lighting, we placed one area light on each side and one point light from the top. It is left to the user to decide, per sequence, which parts of the object to render and which frames to insert. If the user prefers a fully automatic rendering procedure, the complete skeleton will be rendered, and evenly spaced frames will be inserted. We rendered our scenes using Cycles in Blender, and for 3D printing, we used a Stratasys J750 printer with a matte surface finish.

6.1 Main Results

Sampled results are shown in Figure 1-2 and Figure 6-1, where the motion sculptures are visualized in images that correspond to the final frames of the sequences.

[1] `http://poliigon.com`

Our approach consistently produces high-quality, artifact-free renderings.

In Figure 6-1, we show rendering examples with different materials, where motion sculptures are embedded into the original video contents or to a synthetic background (Figure 6-1b and c). We also rendered the scene from a novel viewpoint (Figure 6-1d); this often reveals information that is not perceptible from the captured viewpoint.

For example, in *Jumping*, the novel-view rendering shows the slide-like structure carved out by the arms during the jump. An even more complex action is presented in *Cartwheel*, where we visualize only the legs to avoid clutter. Viewing the sculpture from a top view reveals that the girl's legs cross in midair and switch their depth ordering – a complex interaction that is hard to comprehend even by repeatedly playing the original video. In the *U-Walking* sequence, the motion sculpture depicts the person's motion in depth more clearly than the original video does. In *Tennis*, the sculpture highlights the bending of the arm during this tennis serve, which is invisible from 2D or 2.5D visualizations shown in Figure 6-5. Similarly, in the *Ballet-2* sequence, a sinusoidal 3D surface emerges from the motion of the ballerina's right arm, again absent in the 2D or 2.5D visualizations shown in Figure 6-5.

6.2 Clips with Moving Cameras

We show motion sculpture results on three videos with moving cameras—*Run, Forrest, Run!*, *Olympic*, and *Dunking*—in Figure 1-2 and Figure 6-2. Figure 1-2 and Figure 6-2a show Justin Gatlin racing in 2012 London Olympics and LeBron James slamdunking, respectively, both with a moving camera. For these two sequences, we first obtained a panoramic image of the background [33], registered each frame to this panoramic background using homography, and finally applied our pipeline to reconstruct the motion sculptures. Our method is robust to these internet videos, conveying the rapid motion of Gatlin's limbs and the trajectory of James's leap.

In *Run, Forrest, Run!* (Figure 6-2b), there is large variation in the scene depth, violating the planar assumptions of homography. Thus, we estimated the camera trajectory using SfM [30], which we then compensated for (see Section 3.3). Our method

also works well on this challenging video, producing a motion sculpture spanning a long distance.

6.3 Evaluating Pipeline Components

We now present quantitative and qualitative evaluations of our model's two key components: joint temporal-geometry estimation (Chapter 3) and flow-based refinement (Chapter 4).

6.3.1 Estimating Geometry over Time

Figure 6-3a bottom shows the motion sculpture generated by replacing our joint optimization with per-frame pose and shape estimations. The errors in the per-frame estimates and the lack of temporal consistency result in a jittery, disjoint sculpture. Our optimization solves for a single set of shape parameters for the entire sequence while imposing motion smoothness priors, and hence significantly improves upon the per-frame results (Figure 6-3b bottom).

To further quantitatively verify the superiority of our joint optimization, we plot the object pose over time in 2D by running the principal component analysis (PCA) on the 72D pose vectors. In the per-frame optimization (Figure 6-3a), we observe significant discrepancy between poses in frames 25 and 26: the human body abruptly swings to the right side. As a result, the first two principal components can explain only 69% of the variance. In contrast, our joint optimization (Figure 6-3b) produces a smooth evolution of poses, which roughly lies on a 2D manifold with the first two principal components explaining 93% of the variance.

6.3.2 Flow-Based Refinement

Because the shape and pose are encoded as low-dimensional basis vectors, perfect alignment between the projected shape and the 2D image is unattainable (see Section 4.1). As shown in Figure 4-2b, such artifacts can be significantly reduced with

	Tennis	Fencing	Ballet-1	Ballet-2	Jumping	Walking	Olympic	Avg
Raw	.56	.87	.54	.60	.57	.68	.65	.64
Warp	.97	.93	.93	.93	.98	.95	.86	.94
Warp+HF	**.98**	**.99**	**.96**	**.96**	**.99**	**.96**	**.92**	**.97**

Table 6.1: IoU between human silhouettes and binarized human depth maps before warping, after warping, and after additional hole filling (HF). Flow-based refinement leads to better alignment with the original images, and improves the final renderings.

our flow-based refinement.

To quantify the contribution of the refinement step, we compute the intersection-over-union (IoU) between the 2D human silhouette (extracted in Section 4.1) and the projected silhouette of the estimated 3D shape. Table 6.1 shows the computed average IoU for all our sequences, before and after flow refinement. As expected, the refinement step significantly improves the 3D-2D alignment, increasing the average IoU from 0.64 to 0.94. After filling the "depth holes" (Section 4.2), the average IoU further increases to 0.97.

6.3.3 Stylistic Design Choices

We conducted user studies on Amazon Mechanical Turk (AMT) to evaluate two key stylistic components in our rendering: (i) the use of reflections and shadows, and (ii) our choice of localized lighting. The raters were requested to choose which visualization they prefer (see an example in Figure 6-4): with *vs*. without reflections and shadows (Ours *vs*. A), and using localized *vs*. ambient lighting (Ours *vs*. B).

The table of Figure 6-4 shows the results collected from 20-35 responses for each sequence, after filtering out workers who failed our qualification task (specifically, a worker has been disqualified if he/she voted differently for the same sequence appearing twice). Most raters preferred our rendering with reflections and shadows (82%) and localized lighting (84%).

6.3.4 Comparing with Other Summarization Methods

In Figure 6-5, we compare our visualizations with shape-time photography [3] and stroboscopic photography. Because shape-time photography works on RGB-depth image pairs, we fed our approximated depth maps to the algorithm in addition to the original videos. Directly applying the method in [3] led to a considerable number of artifacts, especially near depth boundaries, perhaps due to the complexity of the scenes. We therefore adapted the model of Freeman and Zhang [3] to normal videos (rather than low-frame-rate videos or hand-selected frames, as in [3]) by augmenting it with the texture smoothness prior of [34] and Potts smoothness terms. This removes artifacts resulting from depth errors and adaptively selects a sparser set of output frames, making results significantly easier to interpret.

Qualitatively, Figure 6-5 shows that our method can reveal interesting spatial-temporal interactions over time. For instance, in *Ballet-1* (Figure 6-5 bottom), our motion sculpture visualizes the out-of-plane sinusoidal curve swept out by the ballerina's arm, whereas only in-plane motions can be seen in both shape-time and stroboscopic photography. Furthermore, our visualization shows the interactions between the left and right arms. By looking at our visualization, the viewer can read out that the ballerina's left arm penetrates the space once traversed by her right arm.

Quantitatively, we conducted another AMT user study, where the raters were asked to evaluate how well 3D is perceived in motion sculpture *vs*. shape-time. Our setup is similar to that in Section 6.3. After filtering out inconsistent users, we collected 234-312 responses for each sequence. On average, 78% of the users preferred our visualization to the shape-time visualization, as shown in the table of Figure 6-5.

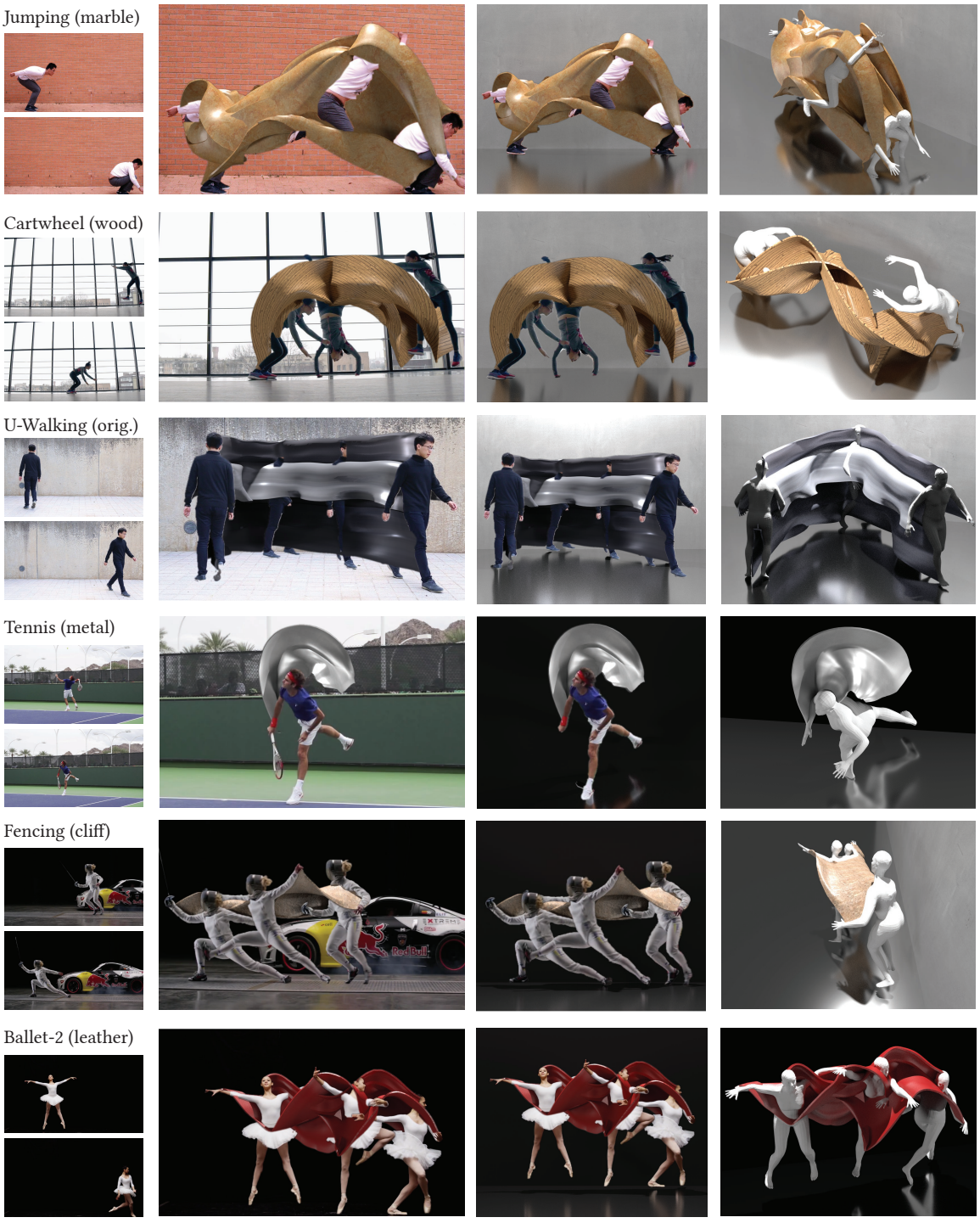
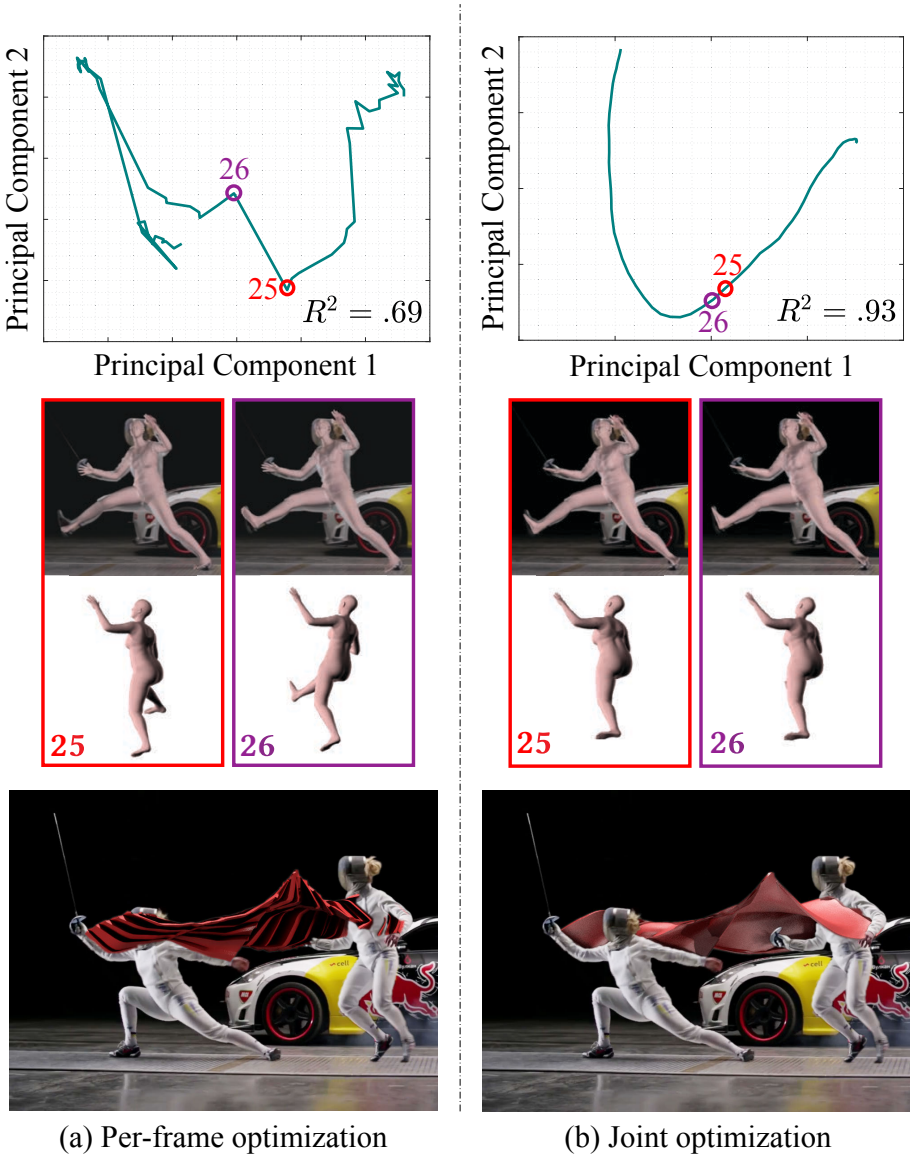


Figure 6-1: Motion sculptures generated by our algorithm on standard videos. (a) The first and last frames of each input video. Our motion sculpture composed with the source video contents (b), and rendered with a synthetic background (c); the material of each sculpture is mentioned next to its sequence name. (d) Full 3D rendering of the motion sculpture from a novel viewpoint.

Figure 6-2: Motion sculptures of videos captured by moving cameras.



(a) Per-frame optimization (b) Joint optimization

Figure 6-3: (a) Pre-frame optimizations produce drastically different poses between neighboring frames (*e.g.*, from frame 25 [red] to frame 26 [purple]). The first two principal components explain only 69% of the pose variance. (b) On the contrary, the joint optimization produces temporally smooth poses across the frames. The same PCA reveals that the pose change is gradual, lying on a 2D manifold with 93% of the variance explained.

A Ours B

	Tennis	Ballet-1	Ballet-2	Jumping	Walking	Olympic	Dunking	**Avg**
Prefer Ours to A	93%	63%	86%	83%	83%	93%	73%	**82%**
Prefer Ours to B	78%	94%	84%	78%	91%	78%	79%	**84%**

Figure 6-4: We conducted human studies to justify our artistic design choices. Top: sample stimuli used in the studies; our rendering (middle) with two variants: (A) without reflections or shadow and (B) without localized lighting. Bottom: users' responses. Most of the users agreed with our choices.

(a) Stroboscopic photos (b) Shape-time photos (c) Motion sculptures

	Tennis	Ballet-1	Ballet-2	Jumping	Walking	Olympic	Dunking	**Avg**
Prefer Ours to [3]	82%	78%	84%	80%	71%	74%	78%	**78%**

Figure 6-5: We compare with two summarization methods: (a) the standard, depth-ignorant stroboscopic photography and (b) shape-time photography [3]. We have also conducted a human study to compare our visualization with [3], where most of the users supported that ours conveys more 3D information.

Chapter 7

Discussions and Conclusion

We presented an algorithm for generating motion sculptures, the spatial-temporal structures traced by objects in motion, from natural videos. Producing such sculptures requires precise shape estimations in the presence of rapid motion – a challenging problem that we addressed using motion and pose analysis. We showed how to insert these sculptures back into the input videos in a way that occlusion relationships are respected. Finally, we showed that our motion sculptures gracefully revealed the beauty and vividness of motion through a variety of examples.

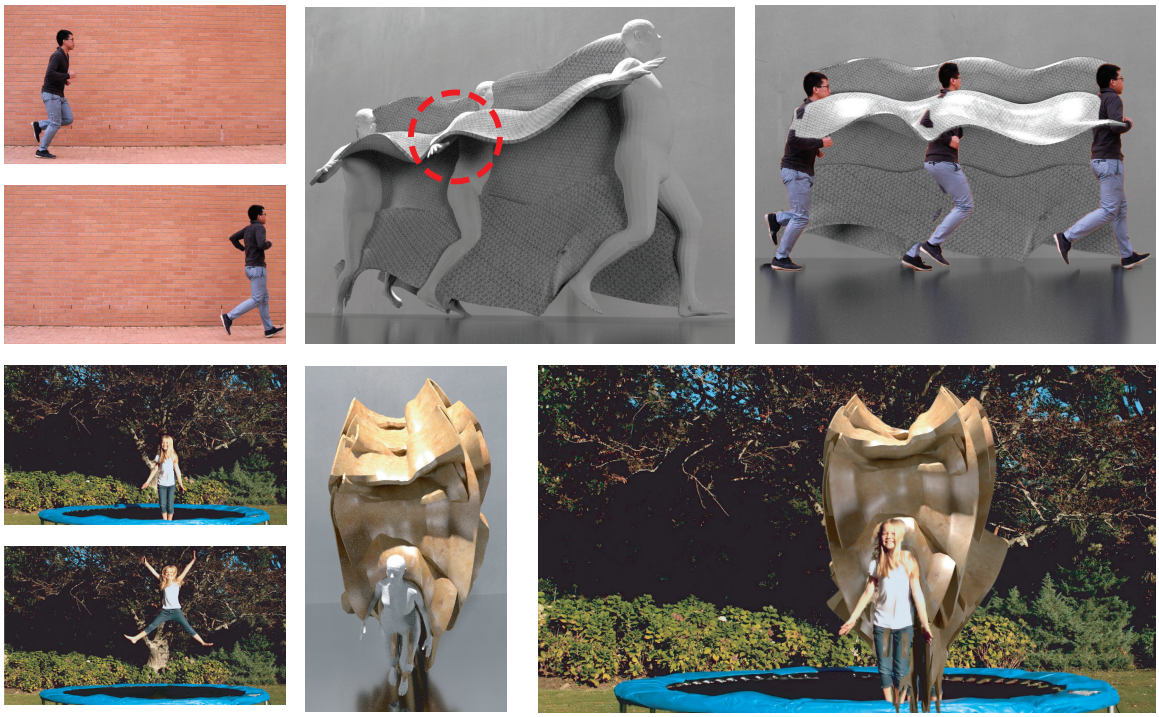
While we have focused on visualizing human motion, our algorithm can also be applied to other objects, as long as they can be reliably represented by a parametric 3D model – an idea that we explored with the following two examples. Figure 7-1a shows the motion sculpture generated for a running horse, where we visualized the two back legs (using the horse's whole body results in significant self-occlusion). To do so, we estimated the horse's pose in all the frames with the per-frame method by Zuffi *et al.* [35], smoothed the estimated poses and translations, and finally applied our 3D-aided IBR algorithm.

In Figure 7-1b, we visualize how a basketball interacts in space and time with the person dribbling it. We tracked the ball (which can be represented with a translation and radius) in 2D, assigning the hand's depth to the ball whenever they are in contact (linearly interpolating the values in between). With these depths, camera parameters, and masks, we inserted a 3D ball into the scene.

(a) Horse gaits (b) Human-ball interaction

Figure 7-1: Motion sculptures for non-human objects. (a) We visualize the leg motion of a horse gait, and (b) we sculpt the interaction between a human and a basketball.

As for limitations, our algorithm relies on reliable pose estimations, which are sometimes unattainable due to the inherent ambiguity of the 2D-to-3D inverse problem. One such example is the person being captured in a near-perfect side view (Figure 7-2a top), where his right forearm has the freedom to swing towards or away from the body without affecting the keypoint reprojection losses or pose priors significantly. This ambiguity leads to pose estimation errors highlighted in Figure 7-2b top. Nevertheless, when our algorithm blends the imperfect sculpture back into the video in the original view, these errors are no longer noticeable (Figure 7-2c top). Also, while our algorithm works well with large motions that traverse through the space, repetitive motion within a certain spatial volume can lead to significant self-occlusion in its motion sculpture (Figure 7-2 bottom).



(a) Sample frames (b) Motion sculpture (c) Motion sculpture in the original view

Figure 7-2: Failure cases. Top: when this person is captured in a near-perfect side view (a), there are multiple possible arm poses that satisfy the objective function equally well (b). Nonetheless, these pose errors are not noticeable in the original camera view (c). Bottom: when the girl's motion remains local instead of spanning large space (a), the motion sculpture is cluttered and does not convey much about the motion (b, c).

Bibliography

[1] Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. KNN matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2175–2188, Sept 2013.

[2] JL Design. CCTV Documentary (Director's cut). `https://vimeo.com/69948148`, 2013.

[3] William Freeman and Hao Zhang. Shape-time photography. In *The IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 151–157. IEEE, June 2003.

[4] Eadweard Muybridge. *Horses and other animals in motion: 45 classic photographic sequences*. Courier Corporation, 1985.

[5] Marta Braun. *Picturing Time: The Work of Etienne-Jules Marey*. Chicago: University of Chicago Press, 1992.

[6] Eyal Gever. Kick Motion Sculpture Simulation and 3D Video Capture. `https://vimeo.com/102911464`, 2014.

[7] Tobias Gremmler. Kung Fu Motion Visualization. `https://vimeo.com/163153865`, 2016.

[8] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. *ACM Transactions on Graphics*, 23(3):294–302, 2004.

[9] Kalyan Sunkavalli, Neel Joshi, Sing Bing Kang, Michael Cohen, and Hanspeter Pfister. Video snapshots: Creating high-quality images from video clips. *Visualization and Computer Graphics, IEEE Transactions on*, 18(11):1868–1879, 2012.

[10] Felix Klose, Oliver Wang, Jean-Charles Bazin, Marcus Magnor, and Alexander Sorkine-Hornung. Sampling based scene-space video processing. *ACM Transactions on Graphics*, 34(4):67, 2015.

[11] James Cutting. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception*, 31(10):1165–1193, 2002.

[12] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. Authoring illustrations of human movements by iterative physical demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 809–820. ACM, 2016.

[13] Johannes Schmid, Robert Sumner, Huw Bowles, and Markus Gross. Programmable motion effects. *ACM SIGGRPAPH*, 29(4):57–1, 2010.

[14] Simon Bouvier-Zappa, Victor Ostromoukhov, and Pierre Poulin. Motion cues for illustration of skeletal motion capture data. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*, pages 133–140. ACM, 2007.

[15] Rubaiat Habib Kazi, Tovi Grossman, Cory Mogk, Ryan Schmidt, and George Fitzmaurice. Chronofab: fabricating motion. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 908–918. ACM, 2016.

[16] Maic Masuch, Stefan Schlechtweg, and Schulz. Speedlines: depicting motion in motionless pictures. In *ACM SIGGRAPH*, 1999.

[17] Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. Stylized rendering techniques for scalable real-time 3D animation. In *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering*, pages 13–20. ACM, 2000.

[18] Yuya Kawagishi, Kazuhide Hatsuyama, and Kunio Kondo. Cartoon blur: nonphotorealistic motion blur. In *Proceedings of the Computer Graphics International Conference*, pages 276–281. IEEE, 2003.

[19] Okihide Teramoto, In Kyu Park, and Takeo Igarashi. Interactive motion photography from a single image. *The Visual Computer*, 26(11):1339–1348, 2010.

[20] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. Phosphor: explaining transitions in the user interface using afterglow effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 169–178. ACM, 2006.

[21] Eric Bennett and Leonard McMillan. Computational time-lapse video. In *ACM Transactions on Graphics*, volume 26, page 102. ACM, 2007.

[22] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the efficiency of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2593–2602. ACM, 2013.

[23] Rohit Ashok Khot, Larissa Hjorth, and Florian (Floyd) Mueller. Understanding physical activity through 3D printed material artifacts. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, pages 3835–3844. ACM, 2014.

[24] Cesar Torres, Wilmot Li, and Eric Paulos. ProxyPrint: supporting crafting practice through physical computational proxies. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 158–169. ACM, 2016.

[25] Saiganesh Swaminathan, Conglei Shi, Yvonne Jansen, Pierre Dragicevic, Lora Oehlberg, and Jean-Daniel Fekete. Supporting the design and fabrication of physical visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3845–3854. ACM, 2014.

[26] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1302–1310. IEEE, 2017.

[27] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael Black. Keep it SMPL: automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael Black. SMPL: a skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015.

[29] Matthew M Loper and Michael Black. OpenDR: an approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113. IEEE, 2016.

[31] Ce Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, May 2009.

[32] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*. IEEE, 2017.

[33] Matthew Brown and David Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[34] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. Shift-map image editing. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 151–158. IEEE, Sept 2009.

[35] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael Black. 3D menagerie: modeling the 3D shape and pose of animals. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 5524–5532. IEEE, July 2017.