# ToonSynth: Example-Based Synthesis of Hand-Colored Cartoon Animations

MAREK DVOROŽŇÁK, Czech Technical University in Prague, Faculty of Electrical Engineering
WILMOT LI, Adobe Research
VLADIMIR G. KIM, Adobe Research
DANIEL SÝKORA, Czech Technical University in Prague, Faculty of Electrical Engineering

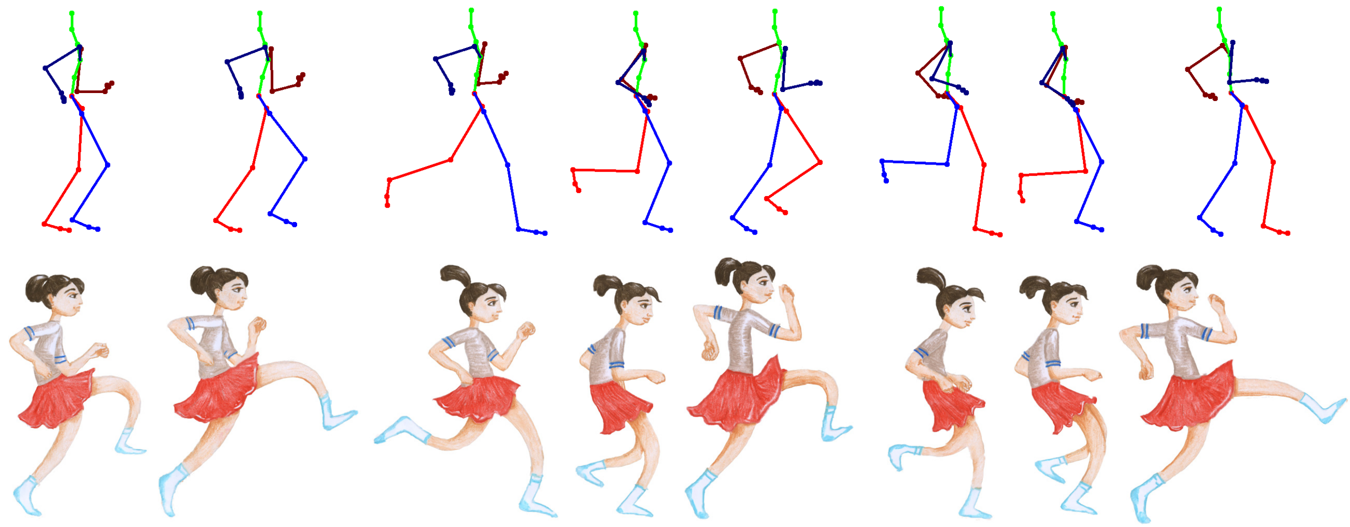

Fig. 1. An example of hand-colored animation synthesized using our approach (bottom row) following the user-specified skeletal animation (top row) and preserving the motion as well as appearance style prescribed by an artist (see a corresponding style exemplar in Fig. 10). Note how the synthesized images still resemble the hand-colored original.

We present a new example-based approach for synthesizing hand-colored cartoon animations. Our method produces results that preserve the specific visual appearance and stylized motion of manually authored animations without requiring artists to draw every frame from scratch. In our framework, the artist first stylizes a limited set of known source skeletal animations from which we extract a *style-aware puppet* that encodes the appearance and motion characteristics of the artwork. Given a new target skeletal motion, our method automatically transfers the style from the source examples to create a hand-colored target animation. Compared to previous work, our technique is the first to preserve both the detailed visual appearance and stylized motion of the original hand-drawn content. Our approach has numerous practical applications including traditional animation production and content creation for games.

CCS Concepts: • **Computing methodologies → Motion processing**; **Image processing**;

Additional Key Words and Phrases: style transfer, skeletal animation

**ACM Reference Format:**
Marek Dvorožňák, Wilmot Li, Vladimir G. Kim, and Daniel Sýkora. 2018. ToonSynth: Example-Based Synthesis of Hand-Colored Cartoon Animations. *ACM Trans. Graph.* 37, 4, Article 1 (August 2018), 11 pages. https://doi.org/10.1145/3197517.3201326

Authors' addresses: Marek Dvorožňák, Czech Technical University in Prague, Faculty of Electrical Engineering, dvoromar@fel.cvut.cz; Wilmot Li, Adobe Research, wilmotli@adobe.com; Vladimir G. Kim, Adobe Research, vokim@adobe.com; Daniel Sýkora, Czech Technical University in Prague, Faculty of Electrical Engineering, sykorad@fel.cvut.cz.


0730-0301/2018/8-ART1 $15.00
https://doi.org/10.1145/3197517.3201326

## 1 INTRODUCTION

While advances in computer graphics have contributed to the evolution of 3D animation as an expressive, mature medium, 2D animation remains an extremely popular and engaging way to tell stories. One common workflow for creating 2D animations is to decompose characters, objects and the background into separate layers that are transformed (either rigidly or non-rigidly) over time to produced the desired motion. A key advantage of this layer-based approach is that a single piece of artwork (i.e., layer) can be reused across many animated frames. As long as the appearance of the layer does not change dramatically (e.g., a character's torso turning from a front to side view), the artist does not need to redraw from scratch.

Compared to drawing and coloring every frame by hand, animating with layers greatly reduces the authoring effort, which is one reason why many modern cartoon series (e.g., Archer, BoJack Horseman, Star vs the Forces of Evil) are created in this manner.

Unfortunately, this increase in efficiency comes at a cost. While hand-created animations give artists complete freedom to specify the appearance of each frame, many styles of artwork are hard to animate using a typical layer-based workflow. Since layers are reused and transformed across several frames, painterly artwork can look awkward as textured regions are compressed and stretched. In addition, rendering styles with visible brush strokes often appear somewhat “dead” when the pattern of strokes remains fixed from frame to frame. Beyond the appearance of the artwork, the motion of layers is also constrained since commercial tools typically enable a limited set of transformations that do not directly support many secondary effects or exaggerated bending and bulging of moving parts. As a result, most layer-based animations are rendered in simple, flat-shaded styles and exhibit relatively stiff or jerky motion.

In this work, we propose an example-based layered animation workflow that allows artists to customize the appearance and motion of characters by specifying a small set of hand-colored example frames for one or more specific source motions. Our system automatically captures and applies the style of the example to new target motions. The key difference between our approach and standard layered animation is that target animation frames are generated by synthesizing each layer based on the set of example frames rather than transforming a single drawn layer. Since the synthesis procedure preserves stylistic aspects in the appearance and motion of the hand-colored source animation, our method supports a much wider range of animation styles. Compared to traditional frame-by-frame drawing, our approach allows artists to get much greater use out of their artwork, since a relatively small set of drawings can be leveraged to produce animated results for a variety of related motions (e.g., a drawn walk cycle can be used to generate a fast angry walk, slow sneaky walk, etc.).

Existing example-based techniques for 2D animation mostly focus on individual sub-problems such as 2D shape interpolation, motion, or appearance transfer. However, focusing on individual steps separately leads to noticeable discrepancies between the real hand-drawn artwork and computer generated output: either the motion characteristics or visual appearance lack quality. For example, in some cases shapes are interpolated with the proper motion characteristics, but the appearance includes artifacts due to distortion or blending of textures [Arora et al. 2017; Baxter et al. 2009; Sýkora et al. 2009]. Or, the appearance is transferred properly, but the underlying motion feels too artificial [Fišer et al. 2017, 2014]). Thus, a key remaining challenge is to combine motion and appearance stylization into a holistic framework that produces synthesis results with all the characteristics of hand-drawn animations. To our best knowledge, our approach is the first that provides such a joint solution and enables fully automatic synthesis of convincing hand-colored cartoon animations from a small number of animation exemplars.

We tailor our method to handle in-plane motions with occlusions, which are typical for cartoon animations and gaming scenarios. Focusing on such motions allows us to apply a relatively simple algorithm that still produces effective results supporting a range of practical applications. For out-of-plane motions that involve more complex depth order changes as well as topological variations, additional manual intervention would be necessary.

Our paper makes the following specific contributions. We define the concept of a layered *style-aware puppet* that is flexible enough to encode both the appearance and motion stylization properties exemplified by the artist’s hand-colored animation frames. We also present a mechanism to combine the information captured by this puppet to transfer motion and appearance style to target animations prescribed by skeletal motion. A key benefit of our technique over previous work is that we specifically designed our pipeline to preserve the visual characteristics of the original artistic media, including a user-controllable amount of temporal incoherence.

## 2 RELATED WORK

Pioneered by Catmull [1978], there has been a concerted effort over the last few decades to simulate or simplify the production of traditional hand-drawn animation using computers.

Computer-assisted inbetweening [Kort 2002] — i.e., generating smoothly interpolated animation from a set of hand-drawn keyframes — is one of the problems that has received significant attention. Various techniques have been proposed to tackle it, achieving impressive results both in the vector [Baxter and Anjyo 2006; Whited et al. 2010; Yang 2017] and raster domains [Arora et al. 2017; Baxter et al. 2009; Sýkora et al. 2009]. Some of these techniques propose N-way morphing between all available frames to widen the available pose space. Nevertheless, inbetweening is designed to deliver plausible transitions only between keyframes. To produce animation for a new target motion, artists must create additional keyframes by hand.

Another large body of research focuses on the simulation of basic motion principles seen in traditional animations, including squash-and-stretch, anticipation, and follow-through [Lasseter 1987]. Existing work proposes customized procedural techniques [Kazi et al. 2016; Lee et al. 2012; Schmid et al. 2010; Wang et al. 2006] as well as controllable physical simulation [Bai et al. 2016; Jones et al. 2015; Willett et al. 2017; Zhu et al. 2017]. Although these methods are capable of achieving the look-and-feel of traditional animation, they do not in general preserve specific motion details that often characterize a given artist’s style. These techniques also do not consider how to faithfully preserve the detailed visual appearance of hand-drawn artwork that is in motion. In most cases textures are simply stretched and deformed, which leads to visual artifacts.

To retain more of a hand-drawn appearance, some techniques directly reuse or manipulate existing hand-drawn content. They either use the animation sequences unchanged [Buck et al. 2000; van Haevre et al. 2005; de Juan and Bodenheimer 2004, 2006] and only reorder the animation frames, add more inbetweens, or directly manipulate the appearance on a pixel level [Sýkora et al. 2011, 2009; Zhang et al. 2012] to enhance the visual content or change the motion characteristics. Although these approaches better preserve the notion of hand-colored animation, their potential to make substantial changes to the motion is rather limited. Extensive manual work is typically required when a different animation needs to be produced out of existing footage.

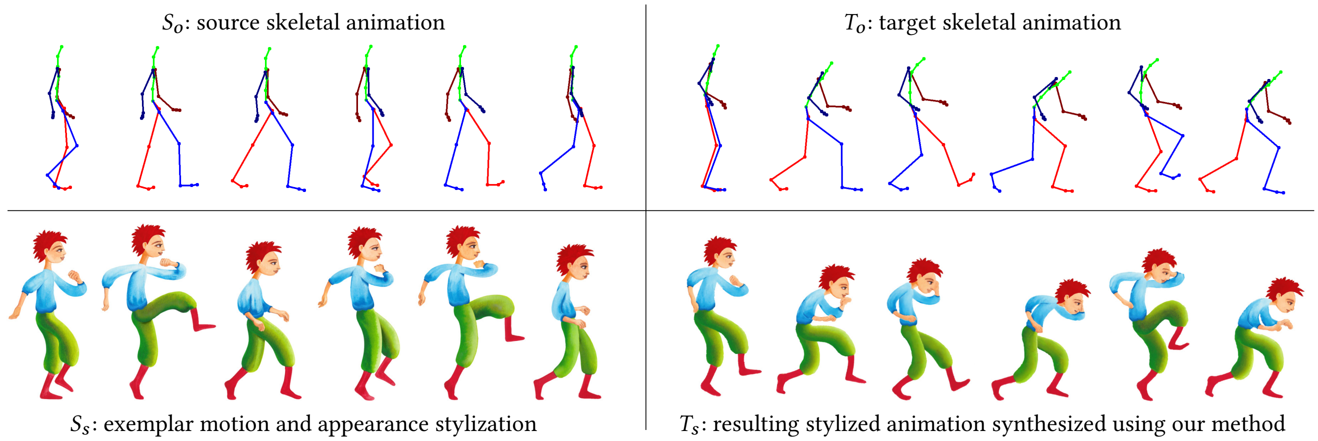


Fig. 2. The animation analogy concept: for a given source skeletal animation ($S_o$) an artist prepares a corresponding hand-colored animation which jointly expresses stylization of character's motion and appearance ($S_s$). Then for a different target skeletal animation ($T_o$) our system produces a synthetically-generated hand-colored animation ($T_s$) that respects the provided analogy $S_o : S_s :: T_o : T_s$ and transfers the motion and appearance style to ($T_o$).

Rather than directly reusing hand-drawn frames, image analogies [Hertzmann et al. 2001] provides a powerful framework for synthesizing new content based on example artwork. In this approach, a guiding image and its stylized version are provided to define the style transfer analogy. This approach has been extended to stylize animations [Bénard et al. 2013] with later work adding user control over the amount temporal flickering [Fišer et al. 2017, 2014] to better preserve the impression that every animation frame was created by hand independently. However, these analogy-based approaches only support appearance style transfer and do not consider how to represent and apply motion stylizations.

Recently, Dvorožňák et al. [2017] presented a motion style analogy framework that has similar motivations to our pipeline. In their workflow, an artist prepares a set of hand-drawn animations that stylize input rigid body motion (of circles or squares) computed using physical simulation. Then they analyze the style by registering a quadratic deformation model as well as a residual deformation. Finally, for a given target rigid body animation, they synthesize a hand-drawn animation by blending the deformation parameters from similar exemplar trajectory segments. One key difference in our work is that we focus not only on motion stylization but also appearance synthesis for fully colored drawings. While Dvorožňák et al.'s method does synthesize simple outline drawings, our approach is designed to support a wide range of hand-colored rendering styles. In addition, the previous technique only handles simple rigid body scenarios where each object in the scene can be represented by a single artwork layer and one set of deformation parameters. In contrast, we describe an analogy framework that works for complex, multi-layered, articulated characters.

Skeletal animation [Burtnyk and Wein 1976] has proven to be an efficient tool for deforming 2D shapes [Hornung et al. 2007; Vanaken et al. 2008]. It has been used to control deformation in the context of cartoon animations [Sýkora et al. 2005; Wang et al. 2013] as well as to transfer motion from a sequence of drawings [Bregler et al. 2002; Davis et al. 2003; Jain et al. 2009] or a single pose [Bessmeltsev et al. 2016] onto a 3D model. In our framework, we demonstrate that skeletal animation can be used also as an effective guide to perform style transfer between hand-drawn exemplars and target animation.

## 3 OUR APPROACH

The primary goal of our work is to help artists create hand-colored animations of characters without having to draw every frame from scratch.

Motivated by the abundance of available motion capture data thanks to recent advances in pose estimation [Mehta et al. 2017], professional MoCap systems (Vicon, OptiTrack, The Captury), and existing motion databases (CMU, HumanEva, HDM05), we assume skeletal animation is easily accessible and can serve as a basic tool to convey motion characteristics. Moreover, tools such as MotionBuilder allow users to combine and extend existing MoCap data using forward/inverse kinematics to create skeletal motions suitable for our method.

Thus, we focus on the challenge of generating colored animations that match a given target skeletal motion while at the same time follow the visual appearance and motion style of an artist-created analogy where a few hand-colored frames serve as an example of how the artist would stylize a particular skeletal animation. Inspired by previous analogy-based techniques [Dvorožňák et al. 2017; Hertzmann et al. 2001] we call our approach *animation analogies*.

In our framework, the artist first chooses a short *source skeletal animation* $S_o$ and creates a *source stylized animation* $S_s$ by authoring hand-colored frames that express the stylization of the source skeletal animation. We call this pair a *source exemplar* $S_o : S_s$. In the source exemplar, we assume that frames of $S_s$ roughly follow the motion in $S_o$, but the details of the motion can be different due to stylization effects. For example, if $S_o$ is a walk cycle, we assume the foot steps in $S_s$ are synchronized, but the legs themselves may

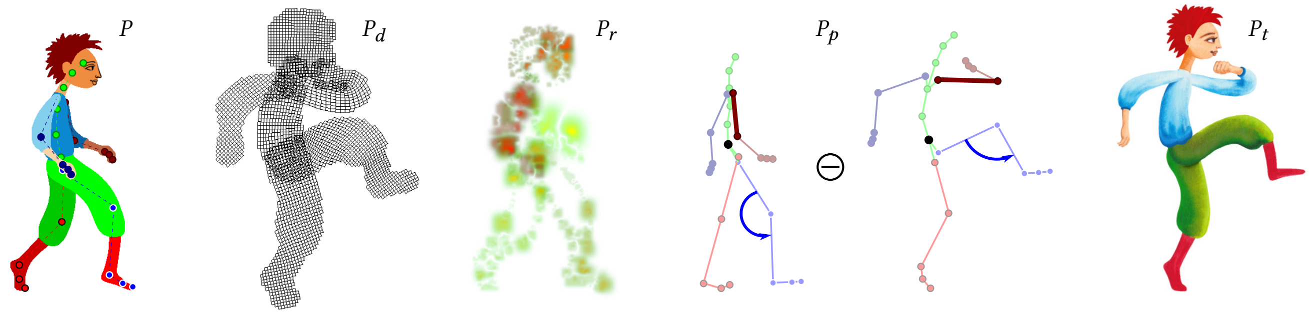


Fig. 3. A style-aware puppet $P_s$ consists of a layered template puppet $P$, coarse deformation of individual puppet shapes $P_d$, their residual elastic deformations captured by multi-layer residual motion field $P_r$ (layers and the magnitude of deformation are color-coded), the difference between the source and stylized skeletal pose $P_p$, and the stylized texture of the character $P_t$.

bend and stretch in an exaggerated way. We also assume that each stylized frame $S_s(i)$ can be separated into a consistent set of layers that are associated with the skeleton bones in $S_o(i)$ and that the depth order of the layers matches that of the corresponding bones. The shape of occluded parts in those layers can be either automatically reconstructed [Sýkora et al. 2014; Yeh et al. 2017] or manually inpainted. An artist can also specify detailed appearance stylization of those occluded parts. However, this step is optional as our appearance transfer technique can be used to fill the missing areas automatically. We then analyze the appearance and motion stylization given by the source exemplar $S_o : S_s$ and let the artist or another user provide multiple novel *target skeletal animations* $T_o$ that represent the desired motions of the character for the final animation. Finally, our method uses the analogy $S_o : S_s :: T_o : T_s$ to automatically generate the corresponding hand-colored output frames $T_s$ (see Fig. 2).

While the target skeletal motions $T_o$ can differ considerably from the source $S_o$, we expect some similarities for our analogy-based framework to work. For example, the artist might stylize a standard walk cycle and transfer the stylization to a sneaky walk, drunk walk, or running cycle. However, a jumping motion might be too dissimilar from the source to stylize successfully, in which case a different style exemplar can be created.

To enable this analogy-based workflow, we propose a guided synthesis technique that uses the style exemplar $S_o : S_s$ to generate stylized frames $T_s$ for the target skeletal motion $T_o$. Our method has two main stages. First, we analyze the source animations to determine the relationship between the skeletal animation $S_o$ and the corresponding hand-colored data $S_s$. Specifically, we construct a *style-aware puppet* $P_s$ that encodes the pose, shape, and appearance stylization $S_s$ for every frame from $S_o$. Once we have this encoding, we can automatically apply the stylization to frames in $T_o$ and generate a new hand-colored animation $T_s$. The following sections describe these two stages in detail.

## 3.1 Representing Source Stylization

Given the source skeletal $S_o$ and stylized $S_s$ animations, we construct a style-aware puppet $P_s$ that describes the pose, shape and appearance properties of the exemplars with respect to a layered template puppet $P$. The template puppet $P$ represents the character in a "neutral" pose; it has the same set of layered parts as the source artwork where each part is associated with a corresponding portion of the source skeleton (see Fig. 4). In case some parts are occluded in the original artwork, we ask the artist to complete their shapes and also specify important semantic details that needs to be preserved (e.g., facial features or cloth draping). We then encode the stylization by registering the template puppet $P$ to every hand-colored source frame $S_s(i)$. This allows us to extract the deformed skeletal pose as well as detailed shape deformation of the character with respect to the neutral pose of $P$. We also encode the appearance of the character in the form of a texture. More formally, a style-aware puppet $P_s$ consists of a layered template puppet $P$ and a tuple $[P_d, P_r, P_p, P_t]$ for each stylized frame $i$ (see Fig. 3): $P_d(i)$ captures the coarse deformation of individual puppet shapes, $P_r(i)$ their residual elastic deformation, and $P_p$ the difference between the source skeletal pose $S_o(i)$ and stylized skeletal pose $S_p(i)$. $P_t(i)$ is the stylized texture of the character. We use these tuples to stylize novel skeletal animations $T_o$.

*Layered Template Puppet Creation.* To create a layered template puppet $P$, we can either use a special unstylized frame in a rest

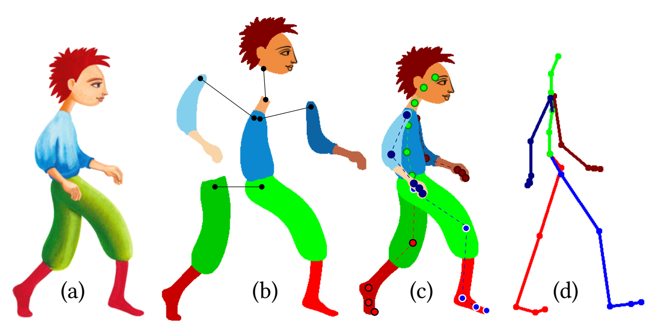


Fig. 4. An example of a layered template puppet: for a single existing hand-colored frame (a), we create a set of semantically meaningful layers which are interconnected at junctions (b) and assign joints of the source skeleton (d) to corresponding locations on each individual layer (c).

pose created by the artist or one of the frames taken from the input hand-drawn animation $S_s$. It consists of a set of semantically meaningful layers (e.g., head, body, hands, and legs) manually stitched together at locations where they naturally connect. Each layer must be attached to the underlying skeleton at one or more user-specified joints. These attachments define the correspondence between bones and layers (see Fig. 4).

*Registration.* To register the template puppet $P$ to every frame $i$ of the segmented hand-colored animation $S_s$, we use a similar approach as in Dvorožňák et al. where a coarse deformation is estimated first and then a more detailed residual motion is extracted. This coarse-to-fine strategy improves the robustness of the registration algorithm while still allowing us to encode very accurate deformations. While Dvorožňák et al. use a single as-rigid-as-possible (ARAP) mesh, a key improvement of our approach is that we use a layered ARAP model with multiple piecewise connected meshes defined by our layered template puppet $P$.

We compute the coarse deformation using the ARAP image registration algorithm [Sýkora et al. 2009], which iteratively applies two steps: the *pushing phase* shifts every point on the ARAP mesh towards a better matching location in the target image using a block-matching algorithm; and the *regularization phase* keeps the ARAP mesh consistent. To use this approach with our multi-mesh ARAP model, we adapt the pushing phase so that the block-matching only uses the content of the corresponding layer to shift each mesh (see Fig. 5, left). This concept is similar to the depth-based separation used in [Sýkora et al. 2010], which avoids clutter caused by occlusion and improves the overall accuracy of the final registration. The registration process as described is automatic. Nevertheless, there can be challenging configurations (e.g., when the deformation is large compared to the template) where manual intervention (dragging a control point to the desired location) can help to speed up the registration process or correct possible misalignments.

Once we obtain a coarse deformation of our layered template puppet $P_d(i)$, we rectify each hand-colored part by removing the computed coarse deformation and perform a more accurate elastic registration between the template and the rectified frame using the method of Glocker et al. [2008]. The result of this step is a multi-layer residual motion field $P_r(i)$ that encodes subtle shape changes of individual body-parts (Fig. 5, right).

To compute $P_p(i)$ we need to infer the stylized skeletal pose $S_p(i)$ from the configuration of the registered puppet layers. We aim to only obtain a 2D projection of the stylized pose. To do so, we use a topologically equivalent 2D representation of the skeleton that is specified by a root joint position, lengths of skeleton bones and their rotations in the ancestor bone's reference frame. Since each layer is attached to the template skeleton at specific joints, the stylized position of those joints can be directly obtained from the position of the corresponding attachment points on the deformed mesh. $P_d(i)$ is then computed as a difference between root joint positions, bone lengths and their rotations: $P_d(i) = S_p(i) - S_o(i)$.

Finally, $P_t(i)$ is obtained by storing pixels from the hand-colored artwork.

## 3.2 Style Transfer of Motion and Appearance to Target Skeletal Animation

*Synthesis of Motion.* We use the extracted style-aware puppet represented by the puppet template $P$ and the per-frame tuples $[P_d, P_r, P_p, P_t]$ to stylize new skeletal animations. We assume that the target skeleton has the same topology as the source skeleton, which is generally true for most MoCap systems.

The transfer of motion style is analogous to patch-based texture synthesis [Kwatra et al. 2005; Wexler et al. 2007] which involves two alternating steps: *search* and *vote*. In our context, instead of texture patches, these steps operate on small sub-sequences of $2N+1$ consecutive skeletal poses around each frame in the source and target animations. The search step finds the closest matching sub-sequence in the source exemplar for each frame in the target and then the voting step averages the content over all intersecting sub-sequences to obtain the final frame pose (see Fig. 6).

More formally, in the search step, we find the closest source sub-sequence $S(i) = S_o[(i-N)\dots(i+N)]$ for each target sub-sequence $T(k) = T_o[(k-N)\dots(k+N)]$ using the pose similarity metric of Kovar et al. [2002], which exploits the sum of distances between point clouds formed by the trajectories of corresponding skeleton joints in each sub-sequence after removing global translation.

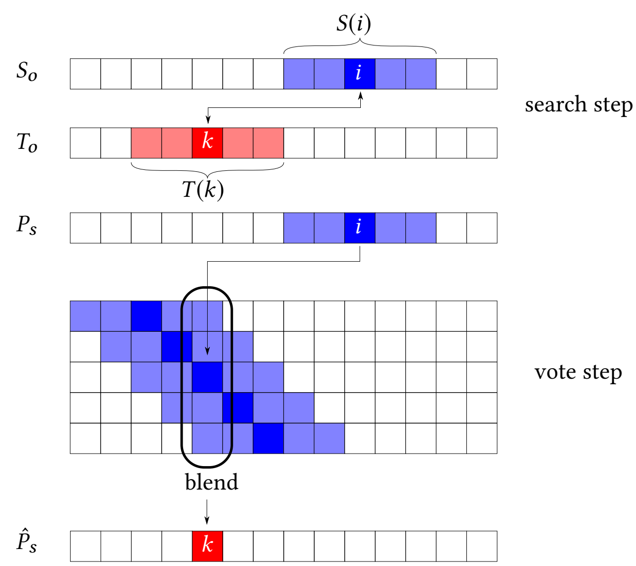


Fig. 6. Obtaining blended style-aware puppet $\hat{P}_s$ for a target frame: for a sub-sequence of the target skeletal animation $T(k)$ the closest sub-sequence of the source skeletal animation $S(i)$ is found (search step) and then a corresponding sub-sequence of style-aware puppets $P(i)$ is blended with other intersecting sub-sequences (vote step).

Once we have found the best matching source sub-sequence for each target frame, we are left with a set of overlapping source sub-sequences (see Fig. 6). At this point, we perform the voting step to blend over all the source frames (using the information encoded in the associated style-aware tuples) that correspond to each output target frame. This step results in a blended style-aware tuple $[\hat{P}_d, \hat{P}_r, \hat{P}_p, \hat{P}_t]$ for each target frame which is obtained using an

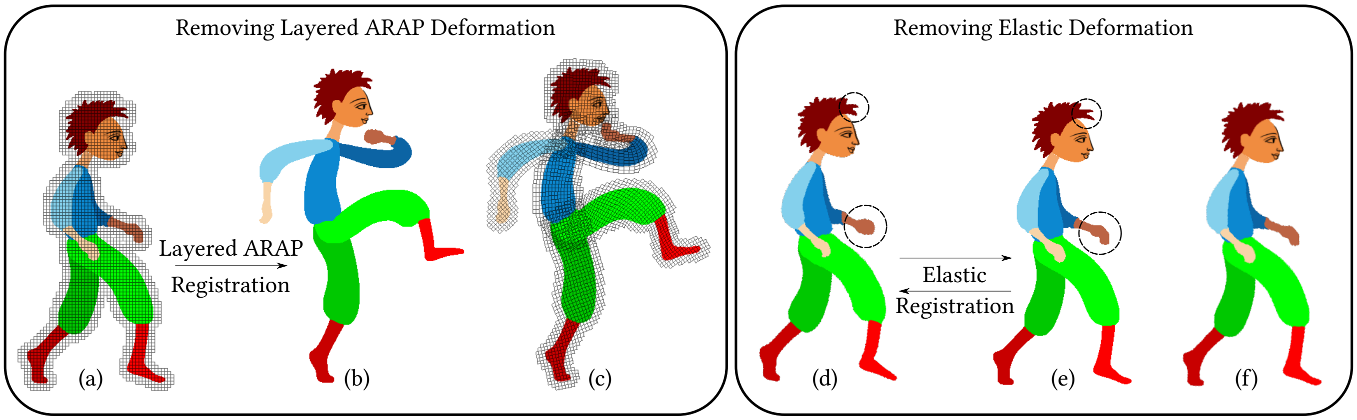


Fig. 5. An example of capturing motion stylization: a layered template puppet (a) is first registered with the segmented version of the stylized animation frame (b) with as-rigid-as-possible (ARAP) image registration [Sýkora et al. 2009] using a layered piecewise connected ARAP deformation model (c). Then, the coarse deformation is removed (d) and the rectified animation frame is registered to the template (e) using the elastic registration method of Glocker et al. [2008] resulting in a segmented stylized animation frame that has both the coarse deformation and the elastic deformation removed (f). Notice the subtle difference in the shape of the hand and hair, which the coarse deformation alone was not able to capture.

N-way ARAP interpolation [Baxter et al. 2009] of the coarse part deformations $P_d$ and a linear blend of the residual shape deformations $P_r$ [Lee et al. 1998] and skeletal pose differences $P_p$. The blended texture $\hat{P}_t$ is obtained by first rectifying the textures $P_t$ (i.e., removing $P_d$ as well as $P_r$) and then linearly blending the pixel colors. Finally, we apply the resulting blended skeletal pose difference $\hat{P}_p(i)$ to the target skeleton $T_o(k)$ to obtain its stylized pose (see Fig. 7).

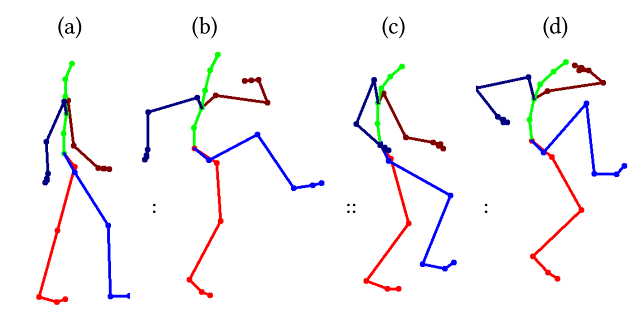


Fig. 7. Style transfer to the target skeletal animation: differences in root joint positions, bone lengths and their rotations between the source skeleton pose (a) and its stylized counterpart (b) are transferred to the target skeleton pose (c) to obtain its stylized pose (d).

*Synthesis of Appearance.* Once the stylized deformation of the target frame is known, a straightforward way to transfer the stylized appearance would be to deform the blended shapes using the new skeleton joint locations on $T_o(k)$ and warp the blended textural information accordingly. This straightforward solution, however, gives rise to numerous artifacts. Linear blending often smooths away visual details in the original hand-colored frames that are critical to the style of the artwork (see Fig. 8 and the supplementary video for comparison). This is caused mainly by the fact that high-frequency details of individual blended frames are not perfectly aligned and thus simple averaging suppresses them. Moreover, in the case where the artist specifies only the shape of the occluded layers in the style exemplar frames, the stylized target may include regions that do not contain textural information, which need to be filled as well. Finally, blending and warping typically does not produce the same type of temporal variation (i.e., “boiling”) that characterizes many hand-colored animations. Ideally, we would like to support controllable temporal flickering as in [Fišer et al. 2014].

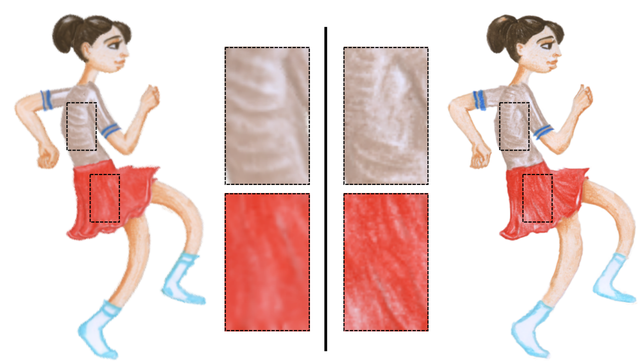

Fig. 8. When the pixel colors of textures of multiple example poses are linearly blended, the result often smooths away subtle details from the original textures (left). This is caused by the blending of slightly different textural content stored in the exemplar frames. The richness of the original textures may be preserved using guided texture synthesis (see the result on the right). See also supplementary video for an animation.

To alleviate all these issues, we replace image warping with guided texture synthesis [Fišer et al. 2017], which creates coherent, detailed texture content and has the flexibility to fill-in newly visible regions. For this technique to work properly we need to prepare a set of guiding channels that define how texture from the source stylized frames should transfer to the deformed target frames.

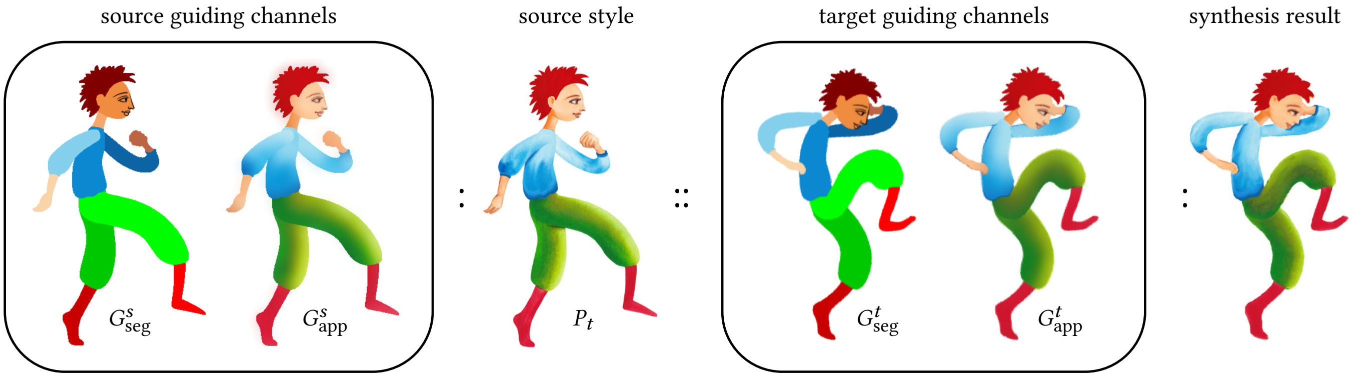


Fig. 9. An example of guiding channels produced by our method to constraint appearance style synthesis: segmentation $G_{\text{seg}}$ and temporal appearance $G_{\text{app}}$. The StyLit algorithm [Fišer et al. 2016] is used to perform the actual synthesis using both guiding channels and style exemplar $P_t$ to produce the final animation frame. The amount of blur in the $G_{\text{app}}$ controls the amount of temporal flickering in the final animation.

Since the textures for various parts of the character are usually distinct, we want to avoid texture transfer across different parts. To this end, we introduce a segmentation-based guidance channel $G_{\text{seg}}$ that represents each segmented part using a separate color label (see Fig. 9). Since the segmentation also contains important semantic details like eyes, nose, and mouth, $G_{\text{seg}}$ ensures that these details will be preserved at the appropriate locations.

In addition, we would like to preserve temporal coherence in the synthesized target textures in a controllable fashion. To do so, we introduce a temporal appearance guide $G_{\text{app}}$ that influences how consistently the texture is synthesized from one frame to the next. We define $G_{\text{app}}$ as the original texture $P_t$ for source frames, and the blended texture $\hat{P}_t$ for target frames. The details in these guiding textures encourage frame-to-frame consistency by restricting a set of matching exemplar patches. To control the amount of consistency we use a similar strategy as in [Fišer et al. 2017, 2014], we smooth $P_t$ and $\hat{P}_t$. However, contrary to Fišer et al. who uses simple Gaussian blur we employ the joint bilateral filter [Eisemann and Durand 2004] with the joint domain $G_{\text{seg}}$, i.e., we avoid blurring over part boundaries which allows to better preserve consistency of individual segments. Increasing the amount of blur in $G_{\text{app}}$ reduces restrictions on the synthesis, thereby increasses the amount of temporal flickering in the resulting synthesized target animation.

To generate the guides for the source animation, we simply render the segmentation labels and texture (with the specified amount of smoothing) for every stylized frame $S_s(i)$. For the target frames, we apply the deformations $\hat{P}_r$ and $\hat{P}_d$ to the template puppet $P$ and warp the puppet to the stylized pose using the skeleton obtained in the motion stylization step. We then render the segmentation labels for $G_{seg}$ and the smoothed texture $\hat{P}_t$ for $G_{app}$. Finally, we run the synthesis using StyLit [Fišer et al. 2016] to produce the final stylized target frames (see Fig. 9).

## 4 RESULTS

We implemented our approach using a combination of C++ and CUDA. We set $N = 4$ in all our experiments. To smoothen the texture using joint bilateral filter for the appearance guide $G_{\text{app}}$, we set $\sigma_{\text{space}} = 5$ and $\sigma_{\text{intensity}} = 1$. For the appearance transfer, the segmentation guide $G_{\text{seg}}$ has weight 2 and $G_{\text{app}}$ is set to 1. For the previously published methods utilized in our pipeline we set parameters according to recommendations in the corresponding papers.

On a quad-core CPU (Core i7, 2.7 GHz, 16 GB RAM), the analysis phase (namely the registration) takes on average 15 seconds per frame (6 seconds for ARAP registration, 9 seconds for elastic registration). Synthesizing new target animation frames takes roughly 9 seconds per frame (1 second for the motion synthesis, 8 seconds for the appearance transfer). The appearance transfer is parallelized on the GPU (GeForce GTX 750 Ti) using CUDA. Moreover, every animation frame can be synthesized independently, i.e., the synthesis process can be executed in parallel on a cluster.

To assess the effectiveness of our method, we asked an artist to prepare a set of hand-drawn exemplars for different skeletal motions selected from the CMU motion capture database[1] (walking, running, and jumping) using different artistic media (watercolor, pencil, and chalk, see Fig. 10 and 14). Then we selected a set of target sequences from the same motion capture database that have similar overall types of movement as the source animations, but different detailed characteristics. For instance, we include slower, faster and "sneaky" walking motions, and sequences that combine running and jumping motions. We also tested slow motion versions of the source skeletal animations to demonstrate that our technique can also be used for inbetweening. Figures 1, 11, 13, and 14 show static frames from some of our results, more synthesized animations can be found in the supplementary video.

Overall, the results demonstrate that our method successfully captures important aspects of the appearance and motion stylization from the different source examples. For example, the appearance synthesis preserves important characteristics of used artistic media including color variations in the water color style, the high-frequency texture in the chalk renderings, and fine shading in the pencil drawings. These characteristics persist throughout the target animations, even when the pose is significantly different from any of the example frames. The artist also added several motion

[1] http://mocap.cs.cmu.edu/

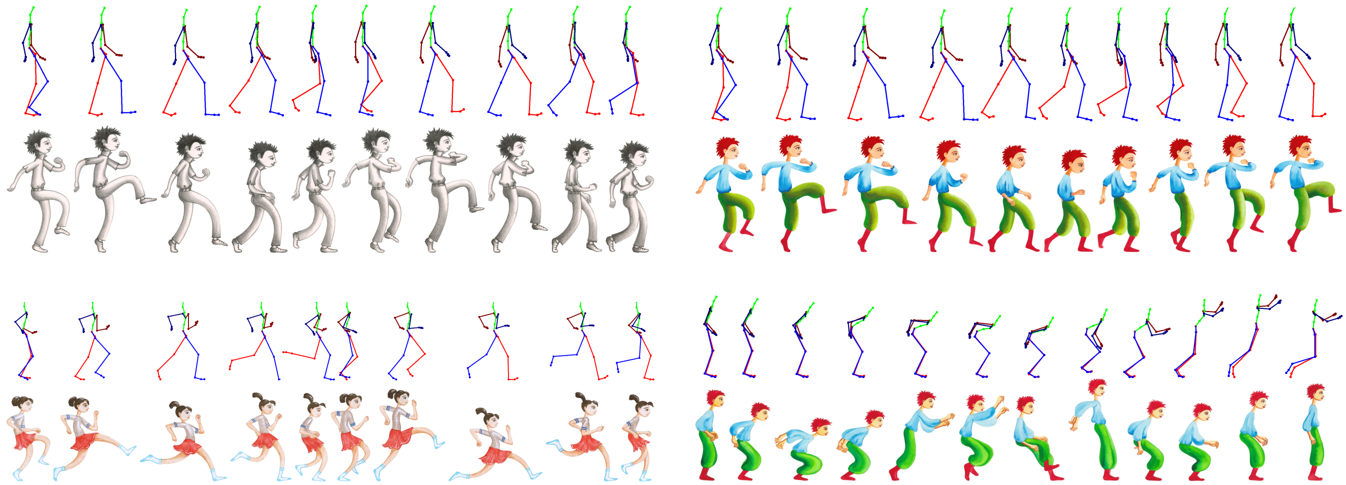

Fig. 10. An overview of exemplar animations created by an artist which we used for all results presented in this paper and in the supplementary video. In each example we show source skeletal animation (top) and its stylized hand-colored counterpart (bottom). Style exemplars: © *Zuzana Studená*

stylizations, such as the exaggerated arm swings and knee raises in the walking motions, and the secondary effects (e.g., squash and stretch) in the jumping and running animations. Our technique transfers these characteristics to the new target motions, as shown, e.g., in Fig. 1.

Our method has several components that together contribute to the quality of the final synthesized animation. To demonstrate the impact of these components, we generated comparison where we add key steps in our pipeline (ARAP deformation, residual deformation, replacing static textures with blended textures, and appearance synthesis) one-by-one, starting from a simple skeleton-based deformation of the source puppet as the baseline. We also generate results with different amounts of temporal coherence by modifying the strength of the joint bilateral blur in the guidance texture. Please refer to our supplemental videos to see these comparisons.

## 5 LIMITATIONS AND FUTURE WORK

Our results demonstrate that the proposed method can effectively transfer a range of stylizations to new target motions. However, the technique as it stands does have some limitations.

*Motion constraints.* The current version of our method does not enforce explicit constraints on the stylized target motion. As a result, artifacts like foot slip or over-exaggerated bending of joints are possible (see Fig. 12, left). It would be a relatively straightforward extension to preserve such constraints by adjusting the stylized target skeletal pose after we apply the per-frame pose deformation $\hat{P}_p(i)$.

*Sub-skeleton matching.* When finding the closest matching source sub-sequence to a given target sub-sequence, we currently incorporate all skeletal joints into the similarity metric. A possible extension for future work would be to consider only partial matches, e.g., to find separate sub-sequence matches for the upper and lower parts of the skeleton. This could provide more flexibility in adapting existing animation exemplars to a larger variety of target motions.

*Out-of-plane motions.* There are two challenges in handling out-of-plane motions with our method. First, since we project 3D skeletal poses to 2D representations, out-of-plane motions can introduce ambiguities in the search phase of the motion synthesis step (see Fig. 12, right). For example, rotating an arm towards the camera may have a similar 2D projection as rotating away from the camera, which can make it hard to automatically select the appropriate source sub-sequence to use for synthesis. To address this, we can extend our approach to use the 3D skeletal information in the source and target sequences. The second challenge involves out-of-plane motions that do not preserve a consistent depth order across the layered parts (e.g., a pirouette). Handling such motions is an interesting direction for future work.

*Reducing input requirements.* Our approach enables artists to leverage a relatively small set of hand-colored frames to synthesize many new target motions. However, there are opportunities

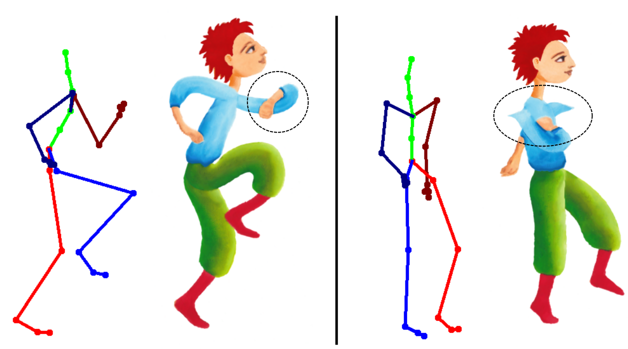

Fig. 12. Limitations: Our method does not enforce explicit constraints on the stylized target motion which may produce over-exaggerated bending of limbs (left). Combined with out-of-plane motions, the deformation may become highly inconsistent and produce visible artifacts (right).

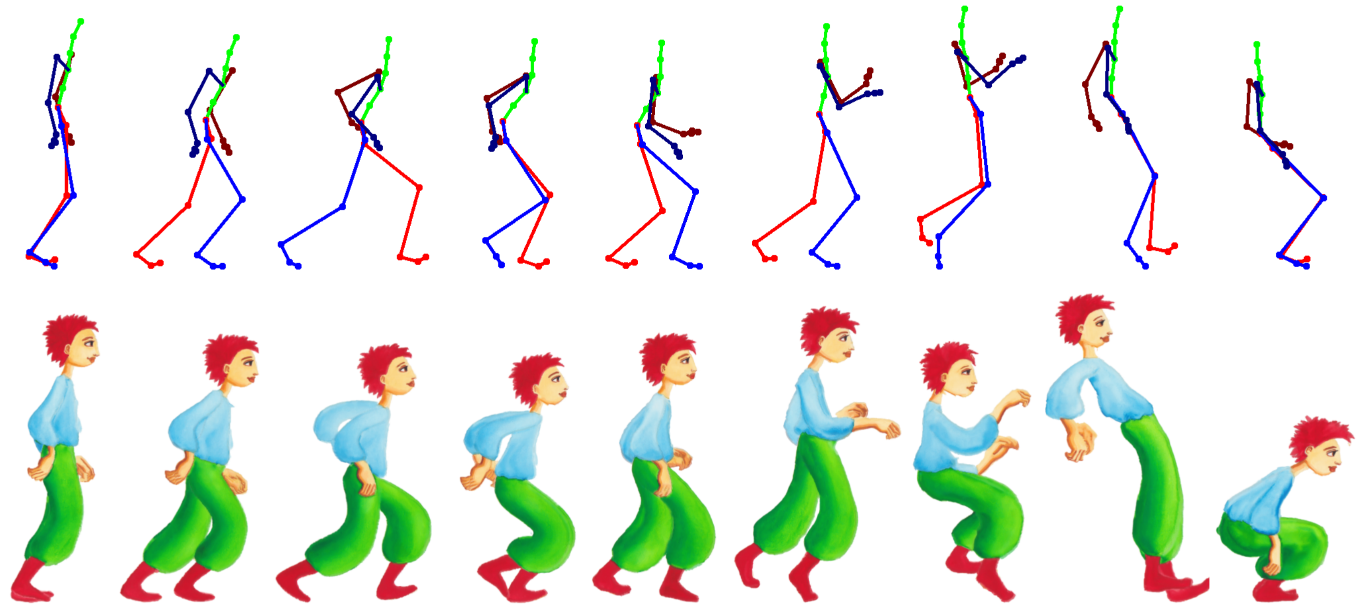

Fig. 11. An example of animation synthesized using our method: target skeletal animation (top), resulting synthesis (bottom). See the original style exemplar in Fig. 10.

to further reduce the input requirements. For example, rather than stylizing every frame of the source skeletal motion, perhaps artists could choose a few key frames to provide hand-colored examples. To support this reduced input, the analysis phase of our framework could potentially interpolate the intervening frames using a guided synthesis method similar to what we currently use to generate stylized target frames. In addition, we could try to augment our existing puppet registration method to avoid the need for a segmented version of each stylized source frame.

*Inconsistent motion or skeletal structure.* In theory, an artist can provide any pose stylization to the input sequence (e.g., mapping a jump motion to a walking sequence or using artwork that has notably different structure from the original skeleton). However, in this situation the closest match is typically very different and thus the algorithm may produce an N-way morph that is far from the expected shape prescribed by the target skeletal pose (e.g., over-exaggerated stretching). In such situations, the artist may need to provide additional stylization frames that capture the desired pose.

## 6 CONCLUSION

In this paper we presented ToonSynth a novel method for synthesizing hand-colored cartoon animations for target skeletal motions. Our approach leverages artist-created exemplars drawn in reference to source skeletal motions. We create a style-aware puppet that encodes the artist-specific stylization into a skeletal pose, coarse as-rigid-as-possible warp, fine elastic deformation, and texture. Using this represetation we can transfer stylization to many new motions by generating guiding channels that capture basic motion properties as well as provide control over the amount temporal dynamics and are used to produce the final appearance using guided patch-based synthesis. This approach enables us to provide the look and feel of hand-colored animation where each frame is drawn independently from scratch.

## ACKNOWLEDGMENTS

We would like to thank Zuzana Studená for preparation of hand-drawn exemplars, Ondřej Jamriška for help on guided texture synthesis part, and all anonymous reviewers for their fruitful comments and suggestions. This research began as an internship by Marek Dvorožňák at Adobe. It was funded by Adobe and has been supported by the Technology Agency of the Czech Republic under research program TE01020415 (V3C – Visual Computing Competence Center), by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/237/OHK3/3T/13 (Research of Modern Computer Graphics Methods), and by the Fulbright Commission in the Czech Republic.

## REFERENCES

Rahul Arora, Ishan Darolia, Vinay P. Namboodiri, Karan Singh, and Adrien Bousseau. 2017. SketchSoup: Exploratory Ideation Using Design Sketches. *Computer Graphics Forum* 36, 8 (2017), 302–312.

Yunfei Bai, Danny M Kaufman, C. Karen Liu, and Jovan Popović. 2016. Artist-directed dynamics for 2D animation. *ACM Transactions on Graphics* 35, 4 (2016), 145.

William Baxter and Ken-ichi Anjyo. 2006. Latent Doodle Space. *Computer Graphics Forum* 25, 3 (2006), 477–485.

William Baxter, Pascal Barla, and Ken Anjyo. 2009. N-way morphing for 2D animation. *Journal of Visualization and Computer Animation* 20, 2–3 (2009), 79–87.

Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. 2013. Stylizing animation by example. *ACM Transactions on Graphics* 32, 4 (2013), 119.

Mikhail Bessmeltsev, Nicholas Vining, and Alla Sheffer. 2016. Gesture3D: posing 3D characters via gesture drawings. *ACM Transactions on Graphics* 35, 6 (2016), 165.

Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. 2002. Turning to the Masters: Motion Capturing Cartoons. *ACM Transactions on Graphics* 21, 3 (2002), 399–407.

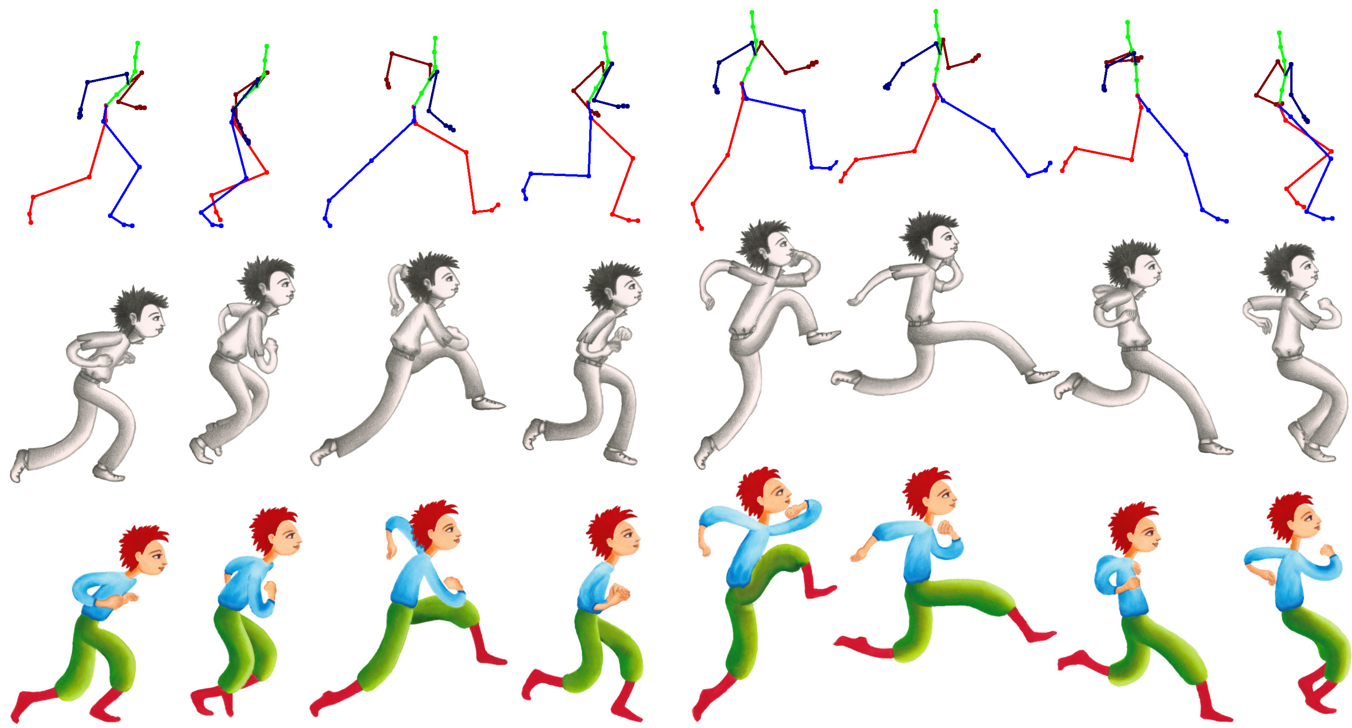

Fig. 13. An example of two hand-colored animations produced using our method (bottom) for the same target skeletal motion (top). See the original pencil and watercolor exemplars in Fig. 10.


Ian Buck, Adam Finkelstein, Charles Jacobs, Allison Klein, David Salesin, Joshua Seims, Richard Szeliski, and Kentaro Toyama. 2000. Performance-Driven Hand-Drawn Animation. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 101–108.

Nestor Burtnyk and Marceli Wein. 1976. Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation. *Commun. ACM* 19, 10 (1976), 564–569.

Edwin E. Catmull. 1978. The Problems of Computer-Assisted Animation. 12, 3 (1978), 348–353.

James Davis, Maneesh Agrawala, Erika Chuang, Zoran Popovic, and David Salesin. 2003. A sketching interface for articulated figure animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 320–328.

Marek Dvorožňák, Pierre Bénard, Pascal Barla, Oliver Wang, and Daniel Sýkora. 2017. Example-Based Expressive Animation of 2D Rigid Bodies. *ACM Transactions on Graphics* 36, 4, Article 127 (2017).

Elmar Eisemann and Frédo Durand. 2004. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics* 23, 3 (2004), 673–678.

Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. 2016. StyLit: Illumination-guided Example-based Stylization of 3D Renderings. *ACM Transactions on Graphics* 35, 4 (2016), 92.

Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. 2017. Example-Based Synthesis of Stylized Facial Animations. *ACM Transactions on Graphics* 36, 4, Article 155 (2017).

Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente, and Daniel Sýkora. 2014. Color Me Noisy: Example-Based Rendering of Hand-Colored Animations with Temporal Noise Control. *Computer Graphics Forum* 33, 4 (2014), 1–10.

Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. 2008. Dense Image Registration Through MRFs And Efficient Linear Programming. *Medical Image Analysis* 12, 6 (2008), 731–741.

William van Haevre, Fabian di Fiore, and Frank van Reeth. 2005. Uniting Cartoon Textures with Computer Assisted Animation. In *Proceedings of International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. 245–253.

Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 327–340.

Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. 2007. Character Animation from 2D Pictures and 3D Motion Data. *ACM Transactions on Graphics* 26, 1 (2007).

Eakta Jain, Yaser Sheikh, and Jessica K. Hodgins. 2009. Leveraging the talent of hand animators to create three-dimensional animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 93–102.

Ben Jones, Jovan Popovic, James McCann, Wilmot Li, and Adam W. Bargteil. 2015. Dynamic sprites: Artistic authoring of interactive animations. *Journal of Visualization and Computer Animation* 26, 2 (2015), 97–108.

Christina de Juan and Bobby Bodenheimer. 2004. Cartoon Textures. In *Proceedings of Eurographics Symposium on Computer Animation*. 267–276.

Christina N. de Juan and Bobby Bodenheimer. 2006. Re-using traditional animation: Methods for semi-automatic segmentation and inbetweening. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 223–232.

Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. Motion Amplifiers: Sketching Dynamic Illustrations Using the Principles of 2D Animation. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 4599–4609.

Alexander Kort. 2002. Computer Aided Inbetweening. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 125–132.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion Graphs. *ACM Transactions on Graphics* 21, 3 (2002), 473–482.

Vivek Kwatra, Irfan A. Essa, Aaron F. Bobick, and Nipun Kwatra. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics* 24, 3 (2005), 795–802.

John Lasseter. 1987. Principles of Traditional Animation Applied to 3D Computer Animation. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. 35–44.

Seungyong Lee, George Wolberg, and Sung Yong Shin. 1998. Polymorph: Morphing Among Multiple Images. *IEEE Computer Graphics and Applications* 18, 1 (1998), 58–71.

Sun-Young Lee, Jong-Chul Yoon, Ji-Yong Kwon, and In-Kwon Lee. 2012. CartoonModes: Cartoon Stylization of Video Objects Through Modal Analysis. *Graphical Models* 74, 2 (2012), 51–60.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM*

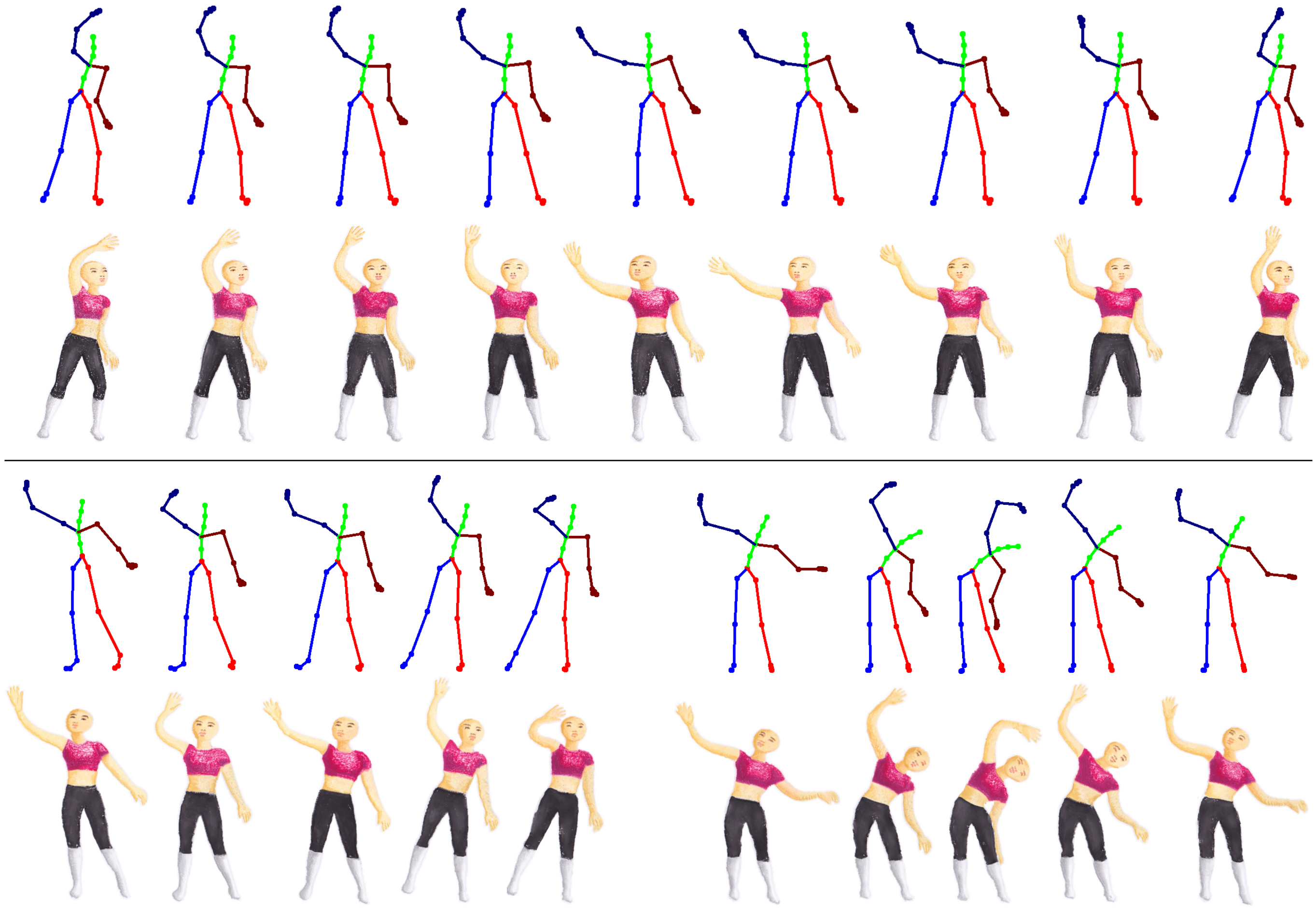

Fig. 14. A hand-colored exemplar animation created by an artist for the specific source skeletal motion (top) has been used to produce the animations at the bottom for a different target skeletal motions. Style exemplars: © *Zuzana Studená*

*Transactions on Graphics* 36, 4 (2017), 44:1–44:14.

Johannes Schmid, Robert W. Sumner, Huw Bowles, and Markus Gross. 2010. Programmable Motion Effects. *ACM Transactions on Graphics* 29, 4 (2010), 57.

Daniel Sýkora, Mirela Ben-Chen, Martin Čadík, Brian Whited, and Maryann Simmons. 2011. TexToons: Practical Texture Mapping for Hand-drawn Cartoon Animations. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 75–83.

Daniel Sýkora, Jan Buriánek, and Jiří Žára. 2005. Sketching Cartoons by Example. In *Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling*. 27–34.

Daniel Sýkora, John Dingliana, and Steven Collins. 2009. As-Rigid-As-Possible Image Registration for Hand-Drawn Cartoon Animations. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 25–33.

Daniel Sýkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. 2014. Ink-and-Ray: Bas-Relief Meshes for Adding Global Illumination Effects to Hand-Drawn Characters. *ACM Transactions on Graphics* 33, 2 (2014), 16.

Daniel Sýkora, David Sedlacek, Sun Jinchao, John Dingliana, and Steven Collins. 2010. Adding Depth to Cartoons Using Sparse Depth (In)equalities. *Computer Graphics Forum* 29, 2 (2010), 615–623.

Cedric Vanaken, Chris Hermans, Tom Mertens, Fabian Di Fiore, Philippe Bekaert, and Frank Van Reeth. 2008. Strike a Pose: Image-Based Pose Synthesis. In *Proceedings of the Conference on Vision, Modeling and Visualization*. 131–138.

Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. 2006. The Cartoon Animation Filter. *ACM Transactions on Graphics* 25, 3 (2006), 1169–1173.

Xun Wang, Wenwu Yang, Haoyu Peng, and Guozheng Wang. 2013. Shape-aware skeletal deformation for 2D characters. *The Visual Computer* 29, 6-8 (2013), 545–553.

Yonatan Wexler, Eli Shechtman, and Michal Irani. 2007. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (2007), 463–476.

Brian Whited, Gioacchino Noris, Maryann Simmons, Robert W. Sumner, Markus H. Gross, and Jarek Rossignac. 2010. BetweenIT: An Interactive Tool for Tight Inbetweening. *Computer Graphics Forum* 29, 2 (2010), 605–614.

Nora S. Willett, Wilmot Li, Jovan Popovic, Floraine Berthouzoz, and Adam Finkelstein. 2017. Secondary Motion for Performed 2D Animation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 97–108.

Wenwu Yang. 2017. Context-Aware Computer Aided Inbetweening. *IEEE Transactions on Visualization and Computer Graphics* (2017).

Chih-Kuo Yeh, Shi-Yang Huang, Pradeep Kumar Jayaraman, Chi-Wing Fu, and Tong-Yee Lee. 2017. Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo. *IEEE Transactions on Visualization and Computer Graphics* 23, 7 (2017), 1796–1808.

Lei Zhang, Hua Huang, and Hongbo Fu. 2012. EXCOL: An EXtract-and-COmplete Layering Approach to Cartoon Animation Reusing. *IEEE Transactions on Visualization and Computer Graphics* 18, 7 (2012), 1156–1169.

Yufeng Zhu, Jovan Popović, Robert Bridson, and Danny M. Kaufman. 2017. Planar Interpolation with Extreme Deformation, Topology Change and Dynamics. *ACM Transactions on Graphics* 36, 6 (2017), 213.