

Example-Based Expressive Animation of 2D Rigid Bodies

MAREK DVOROŽŇÁK, Czech Technical University in Prague, Faculty of Electrical Engineering

PIERRE BÉNARD, LaBRI (UMR 5800, CNRS, Univ. Bordeaux), France

PASCAL BARLA, Inria Bordeaux Sud-Ouest, France

OLIVER WANG, Adobe Research

DANIEL SÝKORA, Czech Technical University in Prague, Faculty of Electrical Engineering

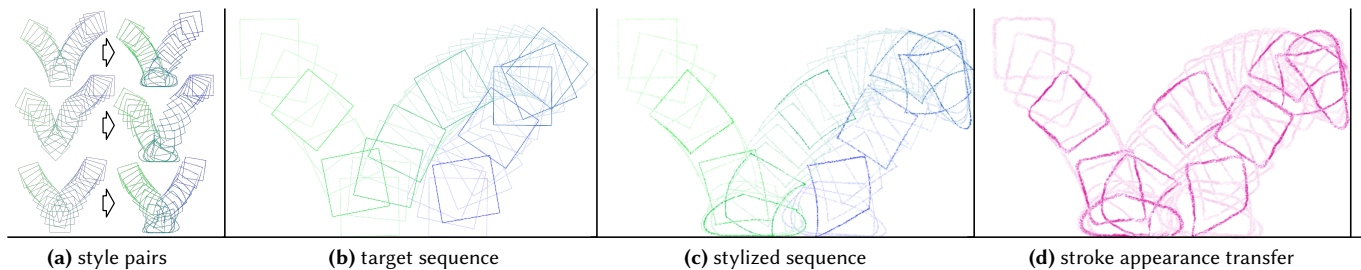


Fig. 1. Given a small set of exemplars consisting of computer-generated and hand-drawn 2D animation pairs (a), our method transfers to a new target sequence produced by physical simulation (b) both the high-level deformations and fine-scale appearance variations (c) present in the example animations. Optionally, the final appearance of the drawings can be modified by re-synthesizing different stroke textures (d).

We present a novel approach to facilitate the creation of stylized 2D rigid body animations. Our approach can handle multiple rigid objects following complex physically-simulated trajectories with collisions, while retaining a unique artistic style directly specified by the user. Starting with an existing target animation (e.g., produced by a physical simulation engine) an artist interactively draws over a sparse set of frames, and the desired appearance and motion stylization is automatically propagated to the rest of the sequence. The stylization process may also be performed in an off-line batch process from a small set of drawn sequences. To achieve these goals, we combine parametric deformation synthesis that generalizes and reuses hand-drawn exemplars, with non-parametric techniques that enhance the hand-drawn appearance of the synthesized sequence. We demonstrate the potential of our method on various complex rigid body animations which are created with an expressive hand-drawn look using notably less manual interventions as compared to traditional techniques.

CCS Concepts: • **Computing methodologies** → *Motion processing; Non-photorealistic rendering;*

Additional Key Words and Phrases: 2D animation, example-based synthesis

ACM Reference format:

Marek Dvorožňák, Pierre Bénard, Pascal Barla, Oliver Wang, and Daniel Sýkora. 2017. Example-Based Expressive Animation of 2D Rigid Bodies. *ACM Trans. Graph.* 36, 4, Article 127 (July 2017), 10 pages.

DOI: <http://dx.doi.org/10.1145/3072959.3073611>

1 INTRODUCTION

Despite the recent success of computer-generated animations, traditional hand-drawn approaches often yield more expressive and stylized looks than those produced with the currently available digital tools. However, creating hand-drawn animations is a tedious

process that requires years of training by an artist and countless hours of labor. Furthermore, style is a highly personalized concept, and two different artists never animate exactly in the same way. As a result, example-based stylization has been a long-standing goal in computer graphics.

In this work we focus on rigid bodies, which are particularly challenging to animate by hand, since multiple objects may collide and rebound in ways that are difficult to plan in advance. Conversely, using physical simulation, computer-based methods can quickly give rise to rigid body animations with realistic trajectories, but ones that lack *expressiveness*. Our main goal is therefore to combine the ease of use of computer-simulated 2D rigid body animations with the expressive qualities of hand-drawn techniques.

To accomplish this goal, we have a number of added requirements. First, editability is of paramount importance to animators, and an ideal solution should work iteratively, always providing the artist with the ability to refine the current solution. Second, producing each hand-drawn frame is time consuming, so a practical example-based 2D animation system should be able to generalize from a *very* limited set of artistic inputs, while being able to apply these edits seamlessly into the dense set of final target frames. These two requirements make example-based 2D animation out of reach of current data-driven machine learning techniques, due to the scarcity of data (tens of exemplars, rather than tens of thousands), and uniqueness of each style.

Instead, our approach is inspired by a workflow that is widespread among both traditional and digital animators. A 2D animation is successively produced and refined at three different temporal scales (see Figure 3): the full animation scale at which timing, contacts and trajectories are planned; the pose-to-pose scale, at which the overall dynamics and deformations between contacts are considered;

© 2017 ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, <https://doi.org/http://dx.doi.org/10.1145/3072959.3073611>.

and the frame-to-frame scale at which the actual drawings with secondary deformations and precise collisions are produced.

At the full animation scale, we split the input computer-generated sequences based on collision events, and we independently analyze and stylize each sub-sequence around a hit point, which we call a key pose. Next, for every sub-sequence, we estimate the spatial deformations of the hand-drawn exemplars before and after the key pose; these are then transferred using a parametric synthesis algorithm. Correspondences between sub-sequences are estimated by leveraging the physical properties of each frame, which ensures preservation of appropriate stylistic effects given the forces applied at each frame. The final frame-by-frame drawings are then synthesized from the artist drawings using a non-parametric technique, that captures residual deformations and appearance details.

We show that this organization is necessary to capture both the long- and short-range stylistic choices made by the artist, while producing results that have the desired hand-drawn look and feel. When taken as a whole, our method considerably reduces the amount of work needed to create the entire sequence by hand.

In summary, we present the following contributions:

- a careful analysis of traditional hand-drawn animations, especially focusing on the correlation between physical parameters and deformations,
- a parametric motion synthesis algorithm capable of transferring deformations from exemplars,
- an example-based non-parametric stylization technique capturing the fine-scale drawing appearance.

2 PREVIOUS WORK

We present prior work related to general stylized computer animation, followed by those driven by physical simulations, and finally example-based solutions.

Techniques inspired by the principles of animation. From its beginning, one of the goals of computer graphics has been to reproduce the expressiveness of traditional hand-drawn 2D animations, while reducing the cost and effort of producing it. The fundamental principles of animation, developed from the late 1920's to the 1930's at Walt Disney Studio [Thomas and Johnston 1981], play a crucial role in this expressiveness, and many works have tried to adapt them to digital animation tools. Lasseter [1987] describes these 2D principles – including squash and stretch, timing and spacing, anticipation and follow-through, arc trajectories and lines of action – and how they can be manually applied by an artist to produce expressive 3D keyframe animations.

Subsequent work has aimed at fully or partially automatizing those effects. Wang et al [2006] describe a simple temporal filter that produces anticipation and follow-through as well as squash-and-stretch deformations by delaying parts of an existing animation, represented by 2D polygonal shapes or motion captured (MoCap) data. Lee et al. [2012] obtain similar effects on segmented objects in a video by relating a set of representative 2D deformations to the modal analysis of the object motion. Focusing on 3D skeletal animation, Noble and Tang [2006] present a tool to automatically bend the limbs of a character following lines of actions or arcs. On the

artistically controllable side, Li et al. [2003] present a sketching interface that allows a user to guide the deformation of both the input animated skeleton and surface mesh to improve the expressiveness of MoCap data. Recently, several 2D sketching systems [Kazi et al. 2014a,b, 2016; Xing et al. 2016] have been developed to simplify the production of dynamic illustrations, reproducing most principles of character or special effects animation. However these principles are essentially encoded as scripted deformations or animated loops triggered by events, and unlike our work are not tailored to the specific style of a given artist.

Physics-driven approaches. Physical simulation is a convenient way to automatically animate a large number of 2D or 3D bodies, but expressiveness of the resulting motion is restricted by the degree of complexity modeled by the physical system. For instance, it is common to restrict the simulation to rigid bodies for computational efficiency, while traditional hand-drawn animated objects more often resemble deformable bodies governed by exaggerated physical laws. To enhance basic 3D simulations, multiple works [Chenney et al. 2002; Garcia et al. 2007; Haller et al. 2004] derive automatic procedural rules to generate squash-and-stretch and temporal effects based on motion parameters (velocity, acceleration, etc.) but such methods have limited art-directability.

To allow artistic control, the spacetime [Witkin and Kass 1988] and dynamic [Barzel and Barr 1988] constraint formulations cast physical simulation as a constrained optimization problem. Through those constraints, the artist can direct the simulation to act as a physically-plausible interpolation mechanism between key-poses. Bai et al. [2016] leverage this idea to build a 2D animation system that combines keyframing of local deformations with physical simulation for powerful inbetweening. Although this approach allows an unprecedented level of artistic control and manages to reproduce many of the principles of animation, it requires the user to specify control handles, which are constrained and unnatural to use when compared to simply drawing frames, in particular when the artist desires a precise control over shape outlines.

Example-based methods. This family of techniques provides a natural and intuitive interface, where examples are used to capture the style and intent of an artist. Such approaches have already produced impressive results for static images and videos, either using non-parametric texture synthesis [Bénard et al. 2013; Fišer et al. 2016; Hertzmann et al. 2001; Lu et al. 2012] or more recently with neural networks [Gatys et al. 2016]. Yet these methods are mostly restricted to appearance stylization, leaving motion largely untouched.

There are some exceptions, such as Bregler et al. [2002] who propose to capture and re-target motion from existing cartoon animations by combining a global affine deformation with drawings interpolation using a key-shape model. Jones et al. [2015] follows a similar approach, connecting the navigation in a simplicial complex [Ngo et al. 2000] with events of a 2D physical simulation. Pose-space interpolation can produce impressive results, but the quality of the output is highly dependent on a good choice of the key-shapes which an artist has to select and organize manually *beforehand*.

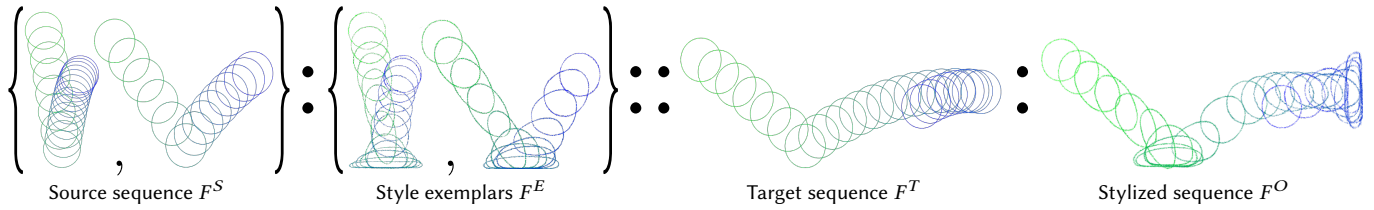


Fig. 2. Stylization analogy setup — given a set of frames F^S coming from reference 2D rigid body source animations, corresponding hand-animated exemplars F^E , and a new target animation F^T , the synthesis algorithm relates physical parameters in F^S and F^T to produce the output stylized sequence F^O that resembles F^E . (Animations are depicted with onionskins, colored from green to blue according to frame numbers.)

To guide or enrich 3D simulations, example-based approaches augment the simulated objects with examples of desirable deformations [Coros et al. 2012; Jones et al. 2016; Koyama et al. 2012; Martin et al. 2011]. In those approaches, however, exact correspondences between deformation exemplars are known beforehand and only a simple parametric deformation with limited number of degrees of freedom is used. Even though this setting may be natural for digital 3D artists, it is again limited and constraining for traditional 2D animators.

Closest to the traditional animation pipeline, Xing et al. [2015] present an interactive system for computer-assisted 2D animation. It combines discrete texture synthesis with an as-rigid-as-possible deformation model to predict and interpolate drawings based on the current and previous frames. This approach is convincing for frame-by-frame animation, but the spatio-temporal locality of the analysis makes it unsuited for pose-to-pose planning. Since the interpolations are solely based on the past drawings using local affine transformations, the predicted motion and deformations tend to be unnatural and cannot easily be edited, unless the artist draws most intermediate frames.

3 OVERVIEW

Similar in spirit to Image Analogies [Hertzmann et al. 2001], our algorithm transforms a *target* 2D rigid body animation F^T into an *output* stylized version F^O using an example-based transformation defined by a set of *source* sequences F^S and a corresponding set of hand-drawn *exemplars* F^E (Fig. 2). Sequences F^S and F^T can be computer-generated using, e.g., physical simulation. The style exemplars F^E are created by an artist digitally or physically, by redrawing a small subset of the source frames F^S . In one application, the untouched frames of F^S can be added to F^T , in which case our method can be seen as an interactive *style propagation* tool. This is shown in the accompanying video where the artist first draws over a few frames, sees the intermediate result, identifies parts which have not been successfully stylized, provides additional examples, and iterates this procedure until she is satisfied by the stylized result.

The key challenge here comes from the fact that F^T will typically not contain sub-sequences exactly like those in F^S , and thus stylized frames from F^E cannot simply replace original rigid frames in F^T . To tackle this problem, we take inspiration from guidelines in traditional 2D animation books [Thomas and Johnston 1981; Williams 2001], especially from Richard Williams’ hierarchical decomposition of the animation process (see Figure 3). We identify three main

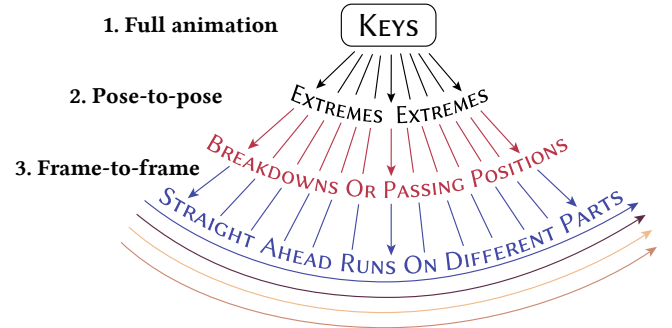


Fig. 3. Three scales hierarchical decomposition of the animation process, based on [Williams 2001, p.67].

stages or temporal scales: (1) the full animation scale, at which timing and spacing are planned by choosing the key events, (2) the pose-to-pose stage, at which the main dynamics and deformations are defined between two key poses by drawing “extremes” and “breakdowns”, and (3) the frame-to-frame scale, corresponding to final straight-ahead “runs” (or drawing passes) during which subtle variations and details are added.

Each of the three stages need to be analyzed for transferring the style of a hand-drawn animation and our method thus follows this organization. First, timing and spacing are specified by the input sequences F^S and all animations are subdivided into overlapping *sub-sequences* around key events (Section 4). The style pairs $F^S : F^E$ are then decomposed into a coarse geometric deformation \mathcal{D} and a fine-scale “residual” stylization \mathcal{R} (Section 5.1). Our aim is to transfer both \mathcal{D} and \mathcal{R} to the target sequence F^T . For each target sub-sequence independently, a new parametric deformation is synthesized by selecting and blending together multiple deformations \mathcal{D} coming from similar sub-sequences of the style pairs (Section 5.2). Finally, sub-sequences are blended together, the fine-scale details are reintroduced on a frame-by-frame basis by morphing the residual changes \mathcal{R} , and the appearance of individual strokes is changed to have a desired look of particular artistic media (Section 6). In the following, we use the classical “bouncing ball” animation to illustrate the various steps of our algorithm; results on more complex sequences (collisions between objects, bouncing square, textured objects) are shown in Section 7 and the supplemental material.

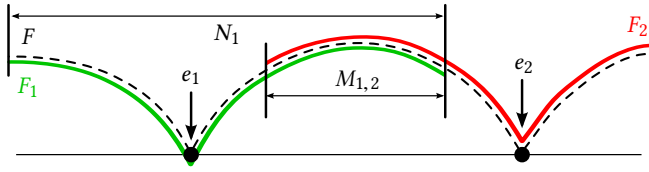


Fig. 4. Decomposition into sub-sequences — an input sequence F is subdivided into sub-sequences (F_i) of N_i frames around key events ($e_i \in \mathcal{E}$) with an overlap of $M_{i,i+1}$ frames with the following sub-sequence.

4 TIMING AND SPACING

The source F^S and target F^T input sequences consist in 2D rigid body animations produced using physical simulation. In practice, we use Box2D [Catto 2007]. Similar to prior work [Kazi et al. 2016], an arbitrary target rigid body is represented by its proxy geometry, e.g., bounding circle, square, or any other closed polygon. In addition to images, the simulation generates, at each frame and for each object, a set of physical parameters including the object velocity v and the rotation angle α around the object centroid with respect to the local trajectory frame. The timing and spacing are dictated by the simulation; the artist draws over existing frames, establishing one-to-one temporal correspondences between F^S and F^E .

The simulation also identifies frames at which semantically important events \mathcal{E} occur, such as contact points or collisions. Following the guidelines of Richard Williams [2001], these frames represent key poses. Analyzing the hand-drawn exemplars F^E , we also observed that those frames, and their immediate neighbors in time, are the ones most stylized by the artist, whereas distant frames are less modified. In addition, we noticed that the physical parameters along the simulated trajectories before and after these key events largely influence the artist’s stylization choices, e.g., the magnitude of the deformation.

These observations motivate us to subdivide F^S , F^T and F^E into a set of smaller overlapping sub-sequences F_i^S , F_i^T and F_i^E around every key event e_i of \mathcal{E} . Each sub-sequence F_i contains N_i consecutive animation frames and overlap with the next sub-sequence on $M_{i,i+1}$ frames. As shown in Figure 4, the overlapping part between two events resides at frames where there are no abrupt changes in physical parameters and moderate artistic stylization, making them most suitable for stitching.

5 POSE-CENTERED DEFORMATIONS

At this stage, we consider each sub-sequence independently, and focus on the coarse deformations used by artists when hand-animating rigid bodies to reproduce effects described in the principles of animation (squash-and-stretch, arc trajectories, lines of action). Residual deformations and appearance variation that are not captured by this coarse deformation will be reintroduced in Section 6.

5.1 Parametric deformation analysis

For each frame of a style pair $F_i^S : F_i^E$, we first estimate a coarse parametric deformation \mathcal{D} (see Figure 5(a)) using the registration algorithm of Sýkora et al. [2009] which aligns bitmap images with an as-rigid-as-possible grid deformation. However, instead of the

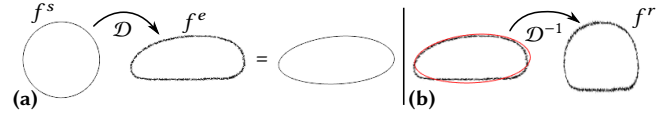


Fig. 5. Deformation analysis — (a) The parametric deformation \mathcal{D} is estimated using as-rigid-as-possible registration between the source f^s and exemplar f^e frames. (b) The residual frame f^r is then computed by applying the inverse deformation \mathcal{D}^{-1} on f^e .

deformable grid matching, we use a single quadratic transformation as in Müller et al. [2005]. The main advantage of this quadratic model is that besides shear and stretch, it also captures twist and bending modes (see Figure 6) which better represent the larger scope of deformations used in traditional animation.

The output of the image registration phase consists of 12 parameters describing the corresponding quadratic deformation that warps pixels $\mathbf{p} = (x, y)^\top$ from the source frame $f^s \in F_i^S$ to match pixels $\mathbf{p}' = (x', y')^\top$ of the stylized frame $f^e \in F_i^E$:

$$\begin{aligned} x' &= a_{11}x + a_{12}y + q_{11}x^2 + q_{12}y^2 + m_1xy + t_1 \\ y' &= a_{21}x + a_{22}y + q_{21}x^2 + q_{22}y^2 + m_2xy + t_2 \end{aligned} \quad (1)$$

Written in matrix form:

$$\mathbf{p}' = \underbrace{\begin{bmatrix} A & Q & \mathbf{m} & \mathbf{t} \end{bmatrix}}_{\mathcal{D}} \tilde{\mathbf{p}} \quad (2)$$

where $\tilde{\mathbf{p}} = (x, y, x^2, y^2, xy, 1)^\top$ is \mathbf{p} expressed in extended homogeneous coordinates, and \mathcal{D} is a quadratic transformation matrix composed of affine A , purely quadratic Q , mixed \mathbf{m} , and translation \mathbf{t} parts:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \quad \mathbf{m} = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

5.2 Parametric deformation synthesis

Based on traditional hand-drawn animation resources as well as our own observations and discussions with 2D animators, we make the key hypothesis that deformation is closely tied to motion. As a result, to perform the deformation transfer, we search for correspondences between source F_i^S and target F_j^T sub-sequences using physical parameters that describe the frame’s *motion* (velocity, trajectory orientation, and the object’s rotation), and we assume that the matching sub-sequences should undergo similar deformations \mathcal{D} as the source ones.

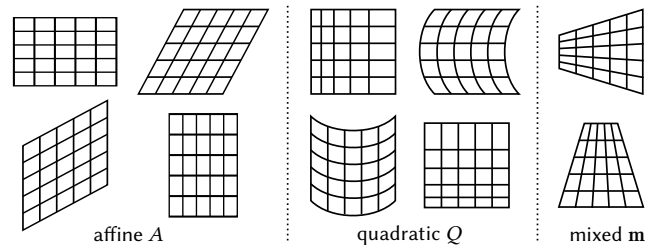


Fig. 6. Visualization of the 10 modes defined by the quadratic deformation of Müller et al. [2005].

Source-target sub-sequence matching. Practically, we define the following difference metric between a source sub-sequence F_i^S and a target sub-sequence F_j^T :

$$\begin{aligned} \text{Diff}(F_i^S, F_j^T) &= \lambda_{\text{vel}} \text{Vel}(F_i^S, F_j^T) \\ &+ \lambda_{\text{dir}} \text{Dir}(F_i^S, F_j^T) \\ &+ \lambda_{\text{rot}} \text{Rot}(F_i^S, F_j^T), \end{aligned} \quad (3)$$

where weights λ_{vel} , λ_{dir} , λ_{rot} are used to balance the influence of individual terms:

- $\text{Vel}(F_i^S, F_j^T)$ measures the difference between rigid body centroid velocities v :

$$\text{Vel}(F_i^S, F_j^T) = \sum_{n=1}^N \|v_n(F_i^S) - v_n(F_j^T)\|^2, \quad (4)$$

- $\text{Dir}(F_i^S, F_j^T)$ penalizes discrepancy of the trajectory orientation δ :

$$\text{Dir}(F_i^S, F_j^T) = \sum_{n=1}^N \|\delta_n(F_i^S) \ominus \delta_n(F_j^T)\|^2, \quad (5)$$

where \ominus computes the smallest difference between two angles,

- $\text{Rot}(F_i^S, F_j^T)$ accounts for differences in the rotation α of the rigid body around its centroid:

$$\text{Rot}(F_i^S, F_j^T) = \sum_{n=1}^N \|\alpha_n(F_i^S) \ominus \alpha_n(F_j^T)\|^2. \quad (6)$$

When computing the metric we assume that both sub-sequences are centered at a key event and have the same number of frames N . This can be done by resampling the original trajectories to have equidistant samples according to their arc length. The longest sub-sequence is trimmed to have the same length as the shortest one.

Deformation blending. Since it is unlikely that any source sub-sequence perfectly matches a given target sub-sequence F_j^T , we retrieve K nearest neighbor sub-sequences $F_1^S \dots F_K^S$ instead of a single one. For each frame in F_j^T , we then compute its stylized version as a combination of K quadratic transformations $\mathcal{D}_1 \dots \mathcal{D}_K$ from the K best corresponding frames in source sub-sequences using weights $w_1 \dots w_K$ proportional to their similarity:

$$w_k = \frac{1/\text{Diff}(F_k^S, F_j^T)}{\sum_{\kappa=1}^K 1/\text{Diff}(F_\kappa^S, F_j^T)}, \quad k \in [1 \dots K]$$

where normalization is used to obtain the partition of unity. See Figure 7 for an overview of this blending scheme (with $K = 3$) which adds robustness to the matching process and gives more stable results than simply using the single best match ($K = 1$).

To perform a meaningful interpolation of the rotational part of the transformation, the affine part A of the matrix \mathcal{D} is factorized using polar decomposition [Higham and Schreiber 1990] into a linear stretch U (not used directly) and a rotation matrix R_α , from which the rotation angle $\alpha = \arctan(r_{11}/r_{21})$ is extracted. A weighted blend is computed on $\alpha_1 \dots \alpha_K$:

$$\hat{\alpha} = w_1 \alpha_1 \oplus \dots \oplus w_K \alpha_K \quad (7)$$

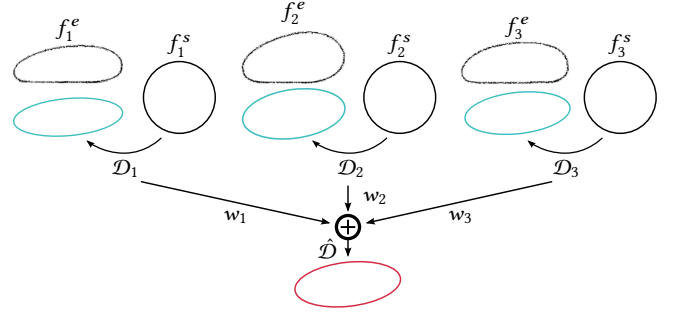


Fig. 7. Blending quadratic deformations \mathcal{D} – individual quadratic deformations $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ estimated from the source frames f_1^s, f_2^s, f_3^s and their corresponding stylized counterparts f_1^e, f_2^e, f_3^e are blended together using the weights w_1, w_2, w_3 to obtain the resulting quadratic deformation $\hat{\mathcal{D}}$.

where \oplus computes a weighted average of circular quantities. The remaining coefficients of rotation-free quadratic transformations $\mathcal{D}'_1 \dots \mathcal{D}'_K$ are similarly computed:

$$\hat{\mathcal{D}}' = w_1 \mathcal{D}'_1 + \dots + w_K \mathcal{D}'_K \quad (8)$$

where $\mathcal{D}' = R_\alpha \mathcal{D}$. Finally, the blended quadratic transformation matrix $\hat{\mathcal{D}}$ is constructed from $\hat{\alpha}$ and $\hat{\mathcal{D}}'$:

$$\hat{\mathcal{D}} = R_{\hat{\alpha}} \hat{\mathcal{D}}' \quad (9)$$

Data augmentation. To generate plausible results even when the global orientation or scale of the target trajectory departs considerably from the available exemplars, we enrich the set of source analogies by scaling, rotating, and flipping the input sequences. We can directly extract the set of required rotation angles γ by analyzing the target simulation. Based on our experiments, we also use 5 scaling factors ρ between 0.2 to 1 and allow symmetries with respect to the vertical axis only (to preserve gravity effects). For rotationally symmetric objects of order n , we modify the \ominus operator in a way that it outputs zero difference for angles $k \frac{360^\circ}{n}$ where the appearance of the rotated object is the same. For the circle the order of rotational symmetry is infinite so instead we set $\lambda_{\text{rot}} = 0$.

The drawback of the augmentation is that it may lead to incorrect stylizations when the source exemplars are far from the target motion. For example, a source exemplar corresponding to a small jump will not be equivalent to a source exemplar with a higher jump. To account for this, we dampen the resulting quadratic transformation $\hat{\mathcal{D}}$ by computing a weighted blend of $\hat{\mathcal{D}}$ with the identity matrix I using weight ξ , proportional to the ratio of the average source and target velocities:

$$\hat{\mathcal{D}} = \xi \hat{\mathcal{D}} + (1 - \xi) I \quad \text{with} \quad \xi = \frac{\sum_{n=1}^N v_n(F_i^S)}{\sum_{n=1}^N v_n(F_j^T)} \quad (10)$$

However, if rotational invariance is not (even approximately) satisfied, orientation augmentation cannot be used, and the artist will need to prepare a set of additional exemplars corresponding to the correct rotational motion.

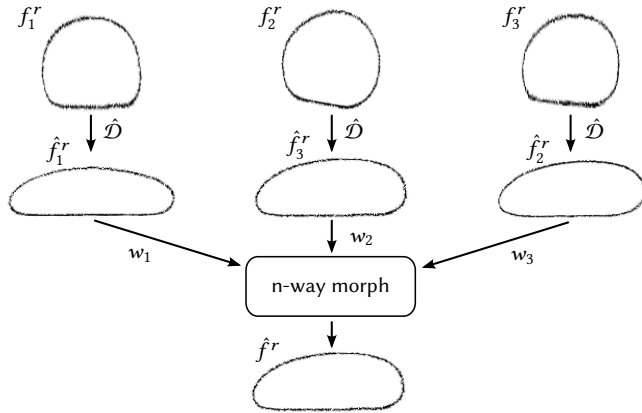


Fig. 8. Synthesis of fine-scale details \mathcal{R} — the synthesized quadratic deformation $\hat{\mathcal{D}}$ is applied to individual residual frames f_1^r, f_2^r, f_3^r producing their deformed counterparts $\hat{f}_1^r, \hat{f}_2^r, \hat{f}_3^r$. Those are then blended together using n-way morphing [Lee et al. 1998] to produce the resulting frame \hat{f}^r .

6 FRAME-BY-FRAME STYLIZATION

Although the parametric transformations \mathcal{D} capture most of the dominant global deformations, there are still small residual deformations and appearance variations \mathcal{R} (e.g., sketch lines of the drawings) which cannot be simply described by the quadratic deformation model. These residual changes represent a very important part of the stylization, as they provide much of the uniqueness of traditional hand-drawn, as opposed to computer-generated animation.

Extraction of the residual. Due to the parametric deformation model, extracting \mathcal{R} from the source exemplars is straightforward. We compute and store the residual frames in F_i^R by “rectifying” the example frames in F_i^E using the inverse transformation to the deformation \mathcal{D} estimated in Section 5.1 (see Figure 5(b)).

Synthesis of fine-scale details. For a given target frame in F_j^T , we now want to re-introduce the residual variations to the synthesized parametric transformation $\hat{\mathcal{D}}$. As illustrated in Figure 8, we deform the corresponding residual frames in f_1^r, \dots, f_K^r using $\hat{\mathcal{D}}$ to produce a set of deformed example frames $\hat{f}_1^r \dots \hat{f}_K^r$. We then compute a set of pairwise pixel-level warping fields $\phi_{\kappa, \kappa'} : \hat{f}_\kappa^r \rightarrow \hat{f}_{\kappa'}^r \forall (\kappa, \kappa') \in \{1 \dots K\}^2$ using deformable image registration [Glocker et al. 2008]. Finally we apply a weighted n-way morph [Lee et al. 1998] to produce a single output frame \hat{f}^r by displacing and blending pixels in $\hat{f}_1^r \dots \hat{f}_K^r$ according to the same weights $w_1 \dots w_K$ as in Section 5.2 and the warping fields $\phi_{\kappa, \kappa'}$.

Sub-sequence stitching. Since the style transfer is applied independently on each animation sub-sequence, stylized overlapping sub-sequences need to be stitched together to avoid abrupt changes. We apply the same approach as described in previous paragraphs, but now only with two frames and with blending weights proportional to the temporal distance τ : to stitch overlapping sub-sequences i and $i + 1$, we use $w_i = \tau/M_{i, i+1}$ and $w_{i+1} = 1 - \tau/M_{i, i+1}$, where $M_{i, i+1}$ is the number of frames in the transition.

Contact points adjustment. Our synthesis process does not guarantee that the resulting animation preserves the alignment with obstacles at contacts. This issue bears resemblance to the foot step detection/correction mechanism used when blending motions in skeletal animation. Yet our problem is simpler since we know the position of contact points; we can easily verify whether the spatial alignment with obstacles is preserved. If not, we simply shift or slightly rotate the synthesis result so that it aligns perfectly with the obstacle at collision time. To estimate the corresponding translation and rotation we use again image registration algorithm of Sýkora et al. [2009]. To avoid ambiguity in translation along the obstacle during the registration, we restrict the centroid of the synthesized drawing to move perpendicularly to (along the normal of) the nearest obstacle.

Texturing. We support two options to apply a texture inside the deformed drawings. The first one takes as input a static image whose content is constant during the full sequence. This image is first rotated according to the target sequence orientation, then it is registered with every residual frame f_i^r using [Glocker et al. 2008], and finally replaces those during the subsequent fine-scale synthesis steps (see Figure 13(c)). If the content of the texture varies in time, the artist needs to provide two versions of the style exemplar: one only showing the outline of the drawing and another with the full texture (see Figure 17). The former is used for quadratic registration whereas the latter is copied during the frame-by-frame synthesis.

Stroke appearance transfer. To offer additional artistic controls, we optionally allow the stroke appearance to be re-synthesized by exemplar using *StyLit* [Fišer et al. 2016]. This can also help suppress resampling artifacts that may occur when applying the quadratic and free-form deformations. In practice we replace the complex illumination-based guidance with a simple gray-scale guiding channel G that softly demarcates positions of strokes in the source and in the target image (see Figure 9).

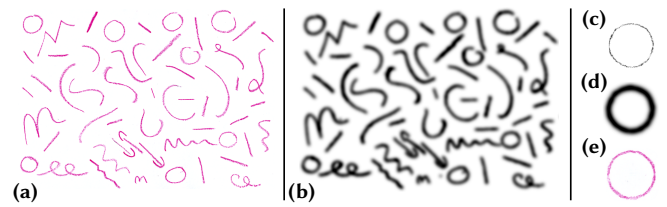


Fig. 9. Stroke appearance transfer based on *StyLit* algorithm [Fišer et al. 2016] — it takes as input an exemplar drawing containing few strokes created using the desired drawing media, e.g., color pencil (a), its corresponding guiding channel (b), the target frame synthesized using our method (c) and its guiding channel (d), and produces the resulting appearance transfer (e).

To generate G we filter the input stroke style exemplar as well as the output frame using the *FDoG* filter [Kang et al. 2007] which suppresses local discrepancies caused by physical properties of the artistic medium (in the stroke style exemplar) or by errors caused by warping and blending pixels (in the target stylized frame) and produces clean stroke outlines. Then a Gaussian blur is applied on this clean outline mask and the resulting bitmap image is normalized and used as a guiding channel for the *StyLit* algorithm.

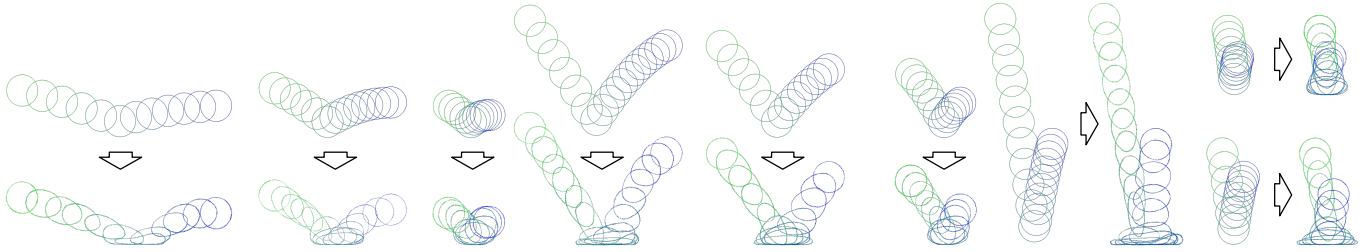


Fig. 10. Style pair database — overview of the source animations and their corresponding stylizations of a bouncing ball drawn by a professional artist.

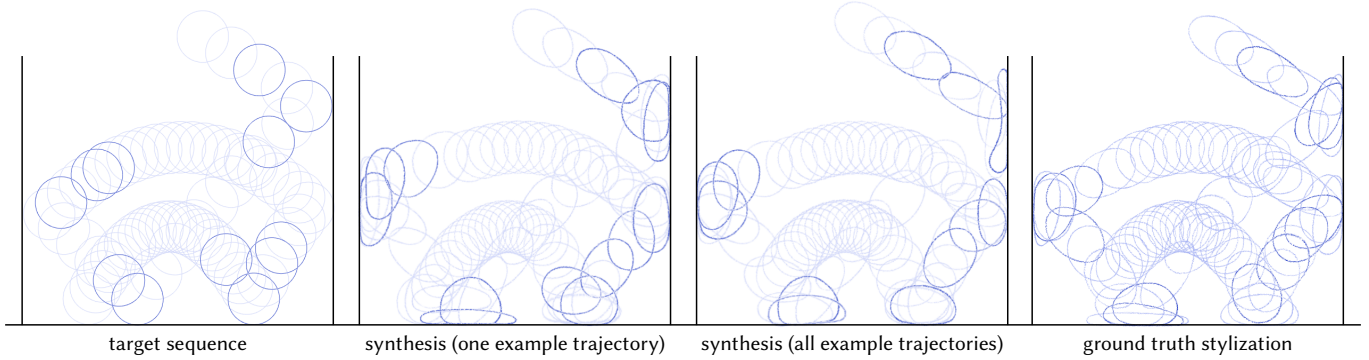


Fig. 11. Results of our synthesis algorithm — (from left to right) target animation computed using physical simulation, synthesis using only one exemplar trajectory, synthesis using all available exemplar trajectories shown in Figure 10, ground truth stylization drawn by a professional artist.

7 RESULTS

We implemented our approach using a combination of Matlab/C++ code, and integrated it as a plug-in into “TVPaint Animation”, a professional 2D animation software. In all our experiments, the input sequences were split into sub-sequences of $N_i = \frac{2}{3}\#(e_{i-1}, e_{i+1})$ frames, where $\#(e_i, e_j)$ is the number of frames between the key events e_i and e_j . The overlap $M_{i,i+1}$ was set to $\frac{2}{3}\#(e_i, e_{i+1})$. For source-target sub-sequence matching, N was set to 20 and the weights were set to $\lambda_{\text{vel}} = 1$, $\lambda_{\text{dir}} = 1$, $\lambda_{\text{rot}} = 50$ to encourage consistent rotation.

To validate our method we designed two practical usage scenarios: (1) an *off-line scenario* where an existing database of pre-made style exemplars is used to perform stylization of new more complex sequences and (2) an *interactive scenario* where an animator specifies a set of sparse style exemplars in the target sequence and those are then used as a source for the synthesis of the entire sequence.

In the case of the off-line scenario, to create the database, we asked a professional animator to stylize a few physical simulations with a single ball (see Figure 10) and square bouncing on the ground (see Figure 1). This database of exemplars was then used to stylize more complex physical simulations containing multiple interacting objects in different environment (see Figures 1, 11, 12, 16, 13, 17 and the supplementary video). For this scenario, we precompute parametric deformations for each exemplar in the database. Our implementation can precompute 10 frames in about 13 seconds on average. Using the precomputed deformations, synthesis of an animation containing 100 frames lasts approximately 56 seconds. The most time consuming parts of the pipeline are the image registration phases. Parallel processing could be used to accelerate the block

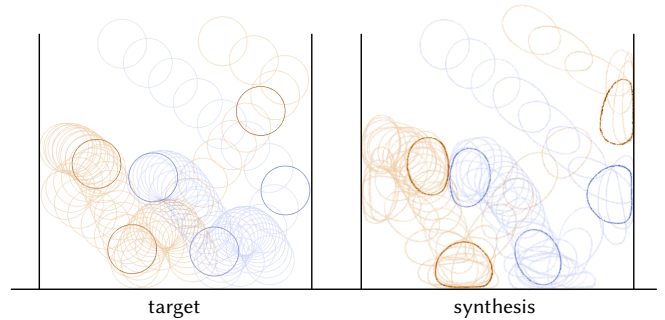


Fig. 12. Result of our synthesis algorithm for two bouncing balls — target sequence computed using physical simulation (left); resulting synthesis computed using all available exemplar trajectories shown in Figure 10 (right).

matching step of Sýkora et al. [2009] and the graph-cut computation in Glocker et al. [2008]. In Figure 14 we also demonstrate the synthesized sequences with additional stroke appearance transfer which further enhances the hand-drawn look and feel.

For the interactive scenario, we let the artist work in TVPaint Animation, a professional animation software, using our plug-in to iteratively improve the stylization of a target sequence. The artist first selects a few frames where a more pronounced stylization is required and draws over them. The target sequence is then re-synthesized taking into account those modified frames, and the artist can inspect the result, tweak already modified frames, or provide stylization for additional frames (see the supplementary video for the recording of a live editing session). In this scenario, we compute

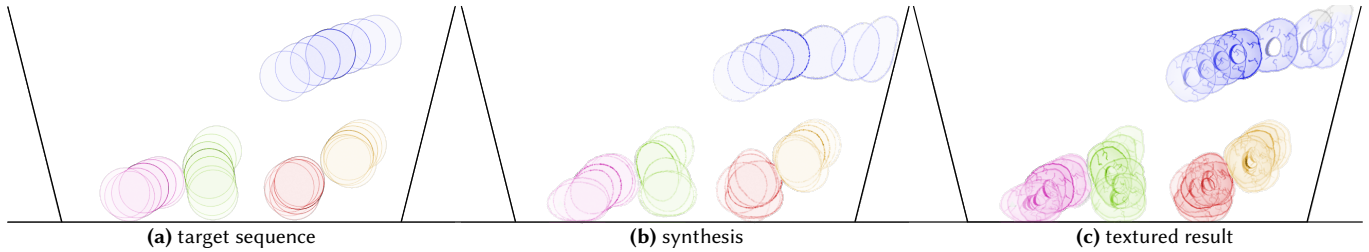


Fig. 13. Result of our synthesis algorithm for more colliding balls — (a) target sequence computed using physical simulation; (b) resulting synthesis computed using all available exemplar trajectories shown in Figure 10; (c) “stone wheel” texture applied during the frame-by-frame synthesis.

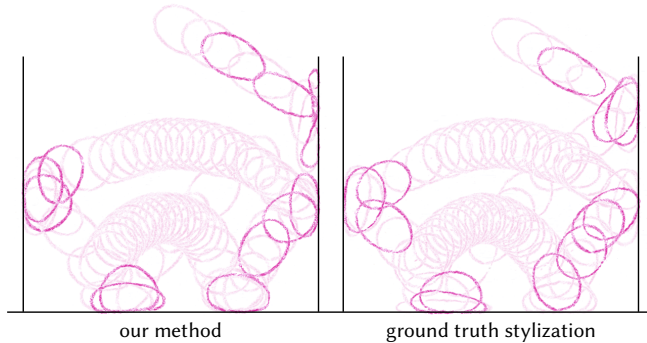


Fig. 14. Stroke appearance transfer applied to our stylized result (left), and on the same sequence drawn by a professional artist (right).

parametric deformations only for newly stylized frames and then the synthesis is executed with comparable computational overhead as in the off-line scenario. In the captured session, the artist stylized only 5 percent of the frames until she was satisfied with the result. The method could even work with a single stylized frame, but it will obviously produce visually pleasing results only when a few frames before and after a collision event are provided.

To confirm that our approach reproduces the hand-drawn style well, we asked the professional artist to stylize the entire sequence frame-by-frame (ground truth), and we also let our algorithm synthesize the same sequence using exemplars (1) from the database and (2) from a fraction of the stylized frames in the ground truth sequence (see Figure 11). Then, we asked two professional animators and three casual observers to compare the three results side-by-side and assess the overall motion stylization quality. In most cases the results were almost indistinguishable from ground truth for non-professional observers. Professional artists were able to see some discrepancies namely in frames of which assumed stylization diverted significantly from the available exemplars, however, the overall response was very positive and they highly appreciate the advantage of having a trade-off between the final visual quality and the time required to produce the exemplars.

In addition to the comparison with a ground truth sequence, we also provide an additional comparison showing the necessity of the decomposition into a quadratic deformation \mathcal{D} and residual changes \mathcal{R} . We synthesize sequences using only our parametric model and then compare with those synthesized using the full processing pipeline (see Figure 15).

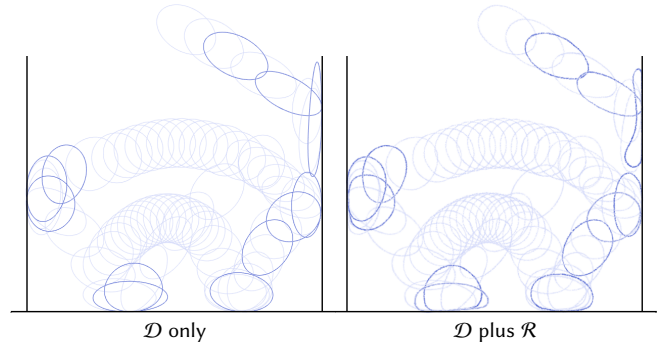


Fig. 15. Comparison — synthesis using only the quadratic deformation model \mathcal{D} (left), and with the full processing pipeline which includes both the quadratic deformation \mathcal{D} and the residual changes \mathcal{R} (right). Note how the addition of the residual changes \mathcal{R} significantly improves the expressiveness of the resulting animation.

8 LIMITATIONS AND FUTURE WORK

We have shown that our approach can produce good results using only a small subset of exemplar frames, and as a result can notably lower the amount of manual work required to create a variety of 2D animations. However, those are currently restricted to simple rigid bodies, and some practical – rather than conceptual – limitations need to be addressed to allow the stylization of more complex sequences.

First, as mentioned in Section 4, for simplicity, our method does not explicitly model the spacing variations that an artist may introduce compared to the physical simulation (e.g., ease-in / ease-out effects). This is especially noticeable in the dynamics of collisions. To a certain extent, those variations are implicitly captured by the translational part of the parametric transformation, but only as a linear frame-by-frame approximation when it should maybe be modeled as a continuous function along the rigid body trajectory. This can be seen in instances where a shape deforms along the contact plane just before or right after the collision (levitation effect).

Besides, our approach takes advantage of rigid body symmetries which are helpful namely for the synthesis of rotations. In particular, exemplars of symmetric counterparts can be reused in situations where there is no corresponding sequence available in the original orientation. However, for asymmetric objects this simplification cannot be used and thus more exemplars need to be provided in order to be able synthesize consistent orientations. To further reduce

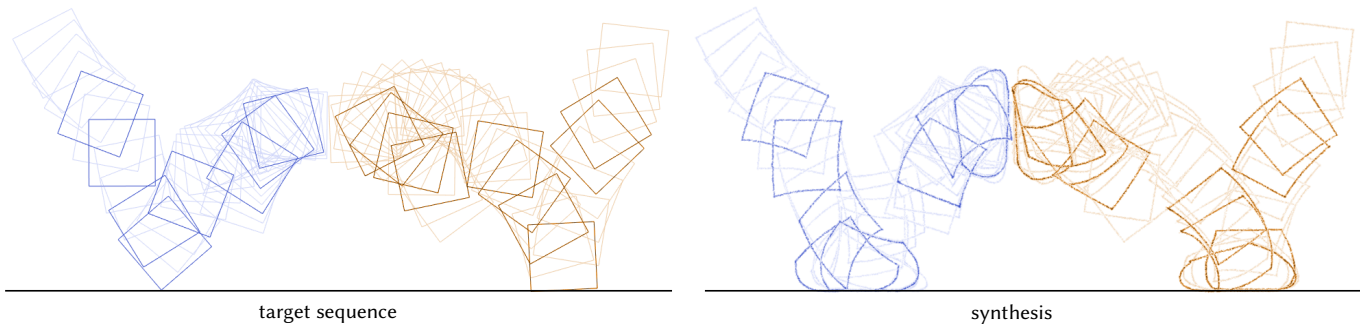


Fig. 16. Result of our synthesis algorithm for two interacting squares — the style pairs of Figure 1 are used, these only contain collisions with the ground plane.

the burden on the artist in such cases, it would be required to model the rotation of the example object separately and allow some amount of re-targeting during the transfer to the novel sequence.

The success of our method also strongly depends on the result of the automatic image registration phase, during which the quadratic deformation model is fit to the stylized exemplar. For elongated objects such as the car in Figure 17, it would be interesting to explore the use of multiple quadratic deformation clusters such that the front of the object may be stylized earlier than its rear along the motion trajectory. More complex or excessive stylization may not fit well to the available degrees of freedom and consequently the resulting residual changes may become overly large. This can cause difficulties during the deformable image registration and the subsequent n-way morphing phase. As future work, we would like to explore different parametric models which would be able to describe a larger variety of deformations occurring in hand-drawn animations. Those more complex models will probably require a better image registration algorithm with a hierarchical processing that is able to adaptively add individual degrees of freedom to avoid getting stuck in some erroneous configurations. It would also require a more advanced blending scheme.

When the exemplar animations differ considerably as compared to the target motion our method just picks a set of most similar ones which likely would not be appropriate. Moreover, when there is a large difference in velocities, the dampening phase tends to remove the stylization effect. Both effects are undesirable and serve as a signal for the artist to provide some more appropriate exemplars.

A last drawback of our technique is that it requires multiple pixel-based warping operations to produce the final shape. Although their number can be minimized by applying compound quadratic transformations, subsequent image re-sampling may still introduce artifacts that can alter the genuine appearance of the used artistic media. This limitation can be partially alleviated by additional post-processing steps such as the proposed appearance re-synthesis or vectorization.

Combining all these improvements, we plan to extend our method to articulated objects such as cartoon characters. In this case a more complex, potentially hierarchical parametric model will be needed.

9 CONCLUSION

We have presented a method allowing to perform style transfer from an existing 2D hand-drawn animation exemplar to a more complex rigid body animation. To the best of our knowledge this is the first attempt to provide a solution for such a challenging task. Despite the recent success of example-based stylization techniques focusing on appearance transfer, animation is still a mostly unexplored area and we believe that the results provided in this paper will motivate other researchers to explore and solve the large variety of challenges that emerges when considering transfer of motion style to more complex articulated objects such as cartoon characters.

ACKNOWLEDGEMENTS

We would like to thank Zuzana Studená for preparation of hand-drawn animations used in the paper, Petra Hapalová for integrating our algorithm into TVPaint, Vojtěch Votýpka for the recording of interactive sessions, and all anonymous reviewers for insightful comments and suggestions. This research 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) and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS16/237/OHK3/3T/13 (Research of Modern Computer Graphics Methods).

REFERENCES

- 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.
- Ronen Barzel and Alan H. Barr. 1988. A Modeling System Based on Dynamic Constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '88)*. ACM, 179–188.
- Pierre Bénéard, 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.
- Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishikesh Deshpande. 2002. Turning to the Masters: Motion Capturing Cartoons. *ACM Transactions on Graphics* 21, 3 (2002), 399–407.
- Erin Catto. 2007. Box2d – a 2D physics engine for games. <http://www.box2d.org>. (2007).
- Stephen Cheney, Mark Pingel, Rob Iverson, and Marcin Szymanski. 2002. Simulating Cartoon Style Animation. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*. 133–138.
- Stelian Coros, Sebastian Martin, Bernhard Thomaszewski, Christian Schumacher, Robert Sumner, and Markus Gross. 2012. Deformable Objects Alive! *ACM Transactions on Graphics* 31, 4 (2012), 69.
- 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.

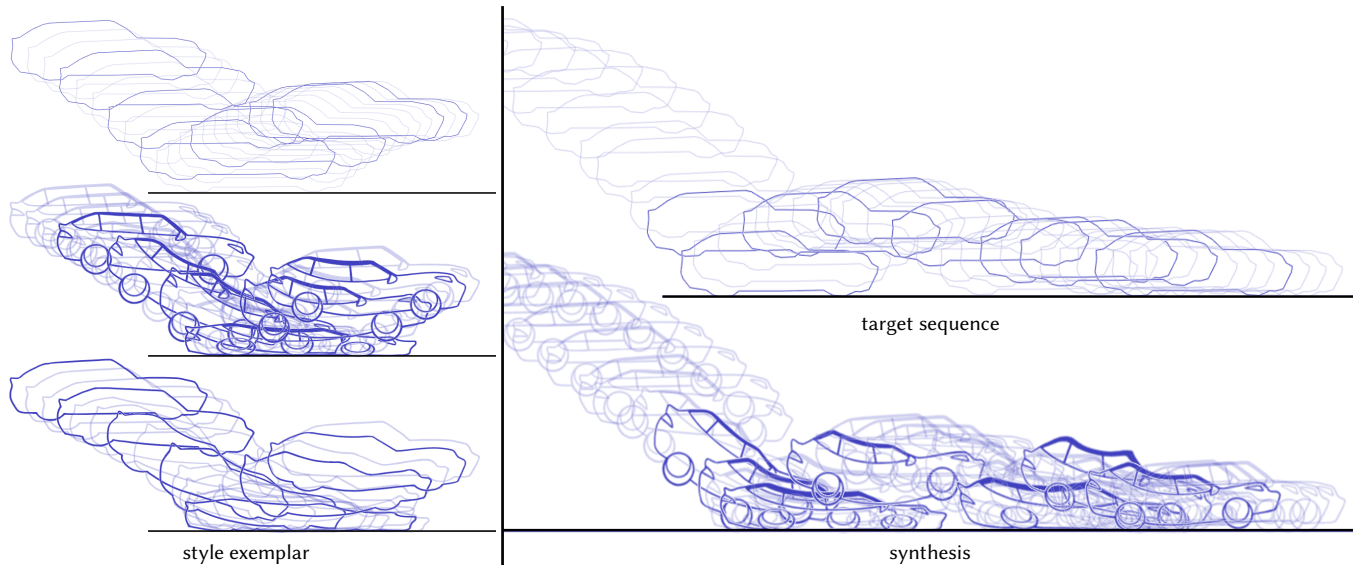


Fig. 17. Result of our synthesis algorithm for a cartoon car – a more complex polygonal proxy is used for the simulation (top left) and registered to the outline of the stylized exemplar (bottom left). During synthesis, the exemplar with texture varying in time (middle left) is used instead.

- Marcos Garcia, John Dingliana, and Carol O'Sullivan. 2007. A Physically Based Deformation Model for Interactive Cartoon Animation. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2007)*. Eurographics Association.
- Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2414–2423.
- 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.
- Michael Haller, Christian Hanl, and Jeremiah Diephuis. 2004. Non-photorealistic Rendering Techniques for Motion in Computer Games. *Computers in Entertainment* 2, 4 (2004).
- 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.
- Nicholas J. Higham and Robert S. Schreiber. 1990. Fast polar decomposition of an arbitrary matrix. *SIAM J. Sci. Statist. Comput.* 11, 4 (1990), 648–655.
- Ben Jones, Jovan Popovic, James McCann, Wilnot Li, and Adam W. Bargteil. 2015. Dynamic sprites: Artistic authoring of interactive animations. *Journal of Visualization and Computer Animation* 26, 2 (2015), 97–108.
- Ben Jones, Nils Thuerey, Tamar Shinar, and Adam W. Bargteil. 2016. Example-based Plastic Deformation of Rigid Bodies. *ACM Transactions on Graphics* 35, 4 (2016), 34:1–34:11.
- Henry Kang, Seungyong Lee, and Charles K. Chui. 2007. Coherent Line Drawing. In *Proceedings of International Symposium on Non-photorealistic Animation and Rendering*. 43–50.
- Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George W. Fitzmaurice. 2014a. Kitty: Sketching dynamic and interactive illustrations. In *Proceedings of ACM Symposium on User Interface Software and Technology*. 395–405.
- Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George W. Fitzmaurice. 2014b. Draco: Bringing life to illustrations with kinetic textures. In *Proceedings of ACM Conference on Human Factors in Computing Systems*. 351–360.
- 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.
- Yuki Koyama, Kenshi Takayama, Nobuyuki Umetani, and Takeo Igarashi. 2012. Real-time Example-based Elastic Deformation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*. Eurographics Association, 19–24.
- 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 (SIGGRAPH '87)*. ACM, 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.
- Yin Li, Michael Gleicher, Ying-Qing Xu, and Heung-Yeung Shum. 2003. Stylizing Motion with Drawings. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. Eurographics Association, 309–319.
- Jingwan Lu, Fisher Yu, Adam Finkelstein, and Stephen DiVerdi. 2012. HelpingHand: example-based stroke stylization. *ACM Transactions on Graphics* 31, 4 (2012), 46.
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based Elastic Materials. *ACM Transactions on Graphics* 30, 4 (2011), 72:1–72:8.
- Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. 2005. Meshless Deformations Based on Shape Matching. *ACM Transactions on Graphics* 24, 3 (2005), 471–478.
- Tom Ngo, Doug Cutrell, Jenny Dana, Bruce Donald, Lorie Loeb, and Shunhui Zhu. 2000. Accessible Animation and Customizable Graphics via Simplicial Configuration Modeling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM, 403–410.
- Paul Noble and Wen Tang. 2006. Automatic Expressive Deformations for Stylizing Motion. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE '06)*. ACM, 57–63.
- Daniel Šykora, 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.
- Frank Thomas and Ollie Johnston. 1981. *The illusion of life: Disney animation*. Disney Editions, New York.
- 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.
- Richard Williams. 2001. *The animator's survival kit*. Faber and Faber, London, New York.
- Andrew Witkin and Michael Kass. 1988. Spacetime Constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '88)*. ACM, 159–168.
- Jun Xing, Rubaiat Habib Kazi, Tovi Grossman, Li-Yi Wei, Jos Stam, and George Fitzmaurice. 2016. Energy-Brushes: Interactive Tools for Illustrating Stylized Elemental Dynamics. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, 755–766.
- Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete Hand-drawn Animations. *ACM Transactions on Graphics* 34, 6 (2015), 169:1–169:11.