# Artist-Directed Dynamics for 2D Animation

Yunfei Bai[1,2] Danny M. Kaufman[1] C. Karen Liu[2] Jovan Popović[1]
[1]Adobe Research [2]Georgia Institute of Technology

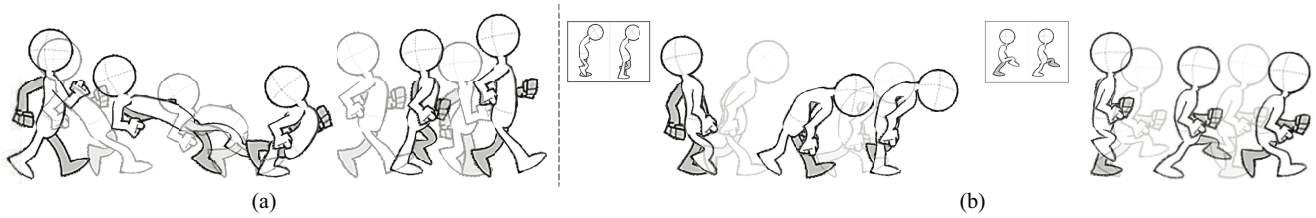


**Figure 1:** *Artist-Directed Dynamics provides an interactive workflow with sparse keyframing, simulation, and example artwork. After creating an expressive walk with keyframes for hand and feet we add a single keyframe to the neck to introduce overlap (a); add two art examples to transform the motion to a sad walk (b, left); or two alternate poses for a sneak walk (b, right).*

## Abstract

Animation artists enjoy the benefits of simulation but do not want to be held back by its constraints. Artist-directed dynamics seeks to resolve this need with a unified method that combines simulation with classical keyframing techniques. The combination of these approaches improves upon both extremes: simulation becomes more customizable and keyframing becomes more automatic. Examining our system in the context of the twelve fundamental animation principles reveals that it stands out for its treatment of exaggeration and appeal. Our system accommodates abrupt jumps, large plastic deformations, and makes it easy to reuse carefully crafted animations.

**Keywords:** simulation, animation, deformation

**Concepts:** •**Computing methodologies** → **Physical simulation;**

## 1 Introduction

Animation is the art of timing and spacing. Timing determines the beat, rhythm, and tempo of motion: a large, heavy, hard ball falls with a thud, its timing between each bounce decreasing rapidly; a small, light, soft ball springs across the screen, its timing almost constant in comparison. Given the timing of key events and extremes, the spacing determines the movement and deformation in frames between the keys. At the top of its arc, the ball slows down so its spacing in nearby frames reduces (overlaps) until it falls down again, gaining speed and traversing more space on every frame. Both elements have profound effect on the final appearance. Slower timing transforms a nervous finger tap into a sluggish pensive motion. Different spacing can remake a simple up-down elevator motion into either a bouncing ball–slowing down (easing in) motion at the top–or into a yo-yo–slowing down at both the top and the bottom. Great animation requires both great timing and great spacing.

Timing is a global property and spacing is a local feature. This dichotomy complicates all systems that must control the two. Consider two classical workflows in hand-drawn animation: straight-ahead and pose-to-pose animation. Straight-ahead animation proceeds locally from one frame to another. Pose-to-pose animation maps out the most important key poses first, before breaking them down into extremes and other salient poses (e.g., passing positions). Each system has advantages and disadvantages. Straight-ahead induces creativity with fluid and natural action but timing can begin to drift and wander. Pose-to-pose crystallizes the timing but the action can become choppy and unnatural. This same tension reappears in modern workflows with simulation and keyframing. Simulation, like straight-ahead animation, sequentially propagates initial values to subsequent frames. Keyframing, like pose-to-pose, breaks down extremes into corresponding inbetweens. Simulation will deliver frame-to-frame realism but the end result may not be at the right place at the right time. Keyframing will keep key events and extremes at the right place and time but the inbetweens may not flow together.

Naturally, a combined approach can be more effective than either extreme. In hand-drawn animation, the recommended hybrid workflow begins with pose-to-pose planning. But then, instead of breaking down these extremes with passing poses, it uses straight-ahead animation to fill in and improvise between global, pose-to-pose placeholders. After one pass, the process repeats, each refinement adding more detail to the previous structure. In modern animation systems, the combined simulation and keyframing workflow is less settled. Some solutions, like pose-to-pose workflow, focus on timing by seeking simulations that place objects with the right timing [Witkin and Kass 1988]. Others, like straight-ahead workflows focus on spacing by driving simulations towards planned keys and poses [Popović et al. 2000].

Our animation system formulates a fluid workflow allowing artists to use both keyframes and simulation. When simulation needs guidance, the artist adds keyframes to nudge it to the desired path. When keyframes stifle lively action, artist lets the simulation act instead. At the core, each frame is simulated, but each simulation step incorporates the artist's examples and trajectories. Examples guide the spacing by exaggerating or suppressing deformation, or both, on different parts of the object. Trajectories command the timing by prescribing it for the entire object, some of its parts, or none at all. Simulation fills in each unspecified part with naturalistic animation.


SIGGRAPH '16 Technical Paper,, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925884

The system employs *handles* as the primary interface for the artist. A handle defines new degrees of freedom (affine transformation) for the object. The artist translates, rotates, and scales handles to create examples with elastic deformation [Jacobson et al. 2011; Jacobson et al. 2012]. Alternatively, they constrain the motion of that handle by providing a trajectory for all or some subset of its degrees of freedom. Handles also identify different parts of the object. When two examples are not related by elastic deformation, the artist indicates their semantic similarity by marking handles in correspondence.

We validate and examine our simulation-based animation workflow in the context of Twelve Principles of Animation [Thomas et al. 1995]. Our work investigates three previously unaddressed challenges required by the Twelve Principles of Animation: (1) what should happen when elastic (or other) simulation deviates from artistic intent; (2) how to create and simulate physically infeasible exaggerations; and (3) how to reuse simulations on drastically different artworks. We found it necessary to develop new techniques and to unify previously separate methods.

## 2 Related Work

Physically-based animation methods employ *simulation* to automatically displace and deform objects in a lifelike way. Rigid bodies move according to their masses, velocities and imposed forces; elastic objects further deform according to internal energy and can be modeled with reduced coordinates, e.g., frame-based degrees of freedom (DOF) [Gilles et al. 2011; Faure et al. 2011]. In contrast to keyframing, physical modeling frees the animator from manually constructing naturalistic motions, e.g., easing in and out of each pose, but since these are fundamentally initial value problems whose outcome is determined at start of time, creative control of resulting animation is challenging at best: one must somehow optimize or infer simulation parameters that yield the desired outcome [Popović et al. 2000; Chenney and Forsyth 2000; Twigg and James 2007; Ha et al. 2013].

Example-based simulation addresses this critical limitation by allowing simulated elastic objects to be augmented with examples of desirable deformations [Martin et al. 2011; Schumacher et al. 2012; Koyama et al. 2012; Coros et al. 2012; Jones et al. 2013]. These methods then attract the motion towards examples with dynamic but less controllable minimization of deformation energy. Animators think and act with visual examples and thus example-based dynamics is a better animation tool than a physically accurate simulation. However, animators often create keyframes not achievable by elastic deformations of a single shape [Lodigiani 2013]. These frames cannot be used in existing example-based techniques.

Animators enjoy the benefits of simulation but do not wish to be blocked by its constraints. Consider, for example, Chuck Jones's classic Road Runner characters that pause in midair before pouncing, or stretch into a dramatically different shapes as they move [Lodigiani 2013]. For the seamless creation of the full range from plausible [Barzel et al. 1996] to cartoon physics scenarios, we need to enable beyond-physical exaggeration and allow for the artist's natural desire to impose additional control simultaneously on both timing *and* spacing.

The variational approach of spacetime constraints provides an optimization framework for controlling physically based simulations [Witkin and Kass 1988]. At their best, this family of methods delivers a high-powered pose-to-pose solution that animates automatically, given the timing of key events and the spacing of physically based simulation. The long-term challenge has been to realize robust and fast space-time solutions. Towards this goal, works have explored the benefits of approximate dynamics [Liu and Popović 2002; Kass and Anderson 2008; Barbič et al. 2012; Hildebrandt et al. 2012], and approximate optimization [Popović et al. 2000; Chenney and Forsyth 2000; Twigg and James 2007].

In seeking interactive control over timing we turn to the greedier formulation of *constraint-based dynamics*. Barzel and Barr [1988] introduced this approach for driving rigid-body dynamics, coining the term *dynamic constraints* in contrast to the variational method of spacetime constraints. Recent works have adapted dynamic constraints for thin-shell simulations [Bergou et al. 2007] and further integrated it with traditional rig-based systems [Hahn et al. 2012]. All three papers have pioneered new workflows for combining keyframing and simulation. Our work goes further by allowing stylized shape transitions between different topologies and parameterizations. We also introduce an approach for reusing simulation (or their parts) on another shape. Instead of tweaking parameters, animators can now copy-paste simulations from one shape to another. Deformation Transfer [Sumner and Popović 2004] delivers this operation for *static* deformation, but not *dynamic* simulation. We show how to integrate example-based simulation, dynamic constraints, and deformation transfer in an animation system that delivers automation and enables stylization.

## 3 Methods

We construct a novel, interactive, physics-based tool to create animations with an intuitive example-based workflow. Here we present the pieces that together form a simple-to-use, handle-focused interaction tool where timing and shape are directly controlled by the artist and simulation fills in remaining degrees of freedom interactively. In addition to keyframes, artists can use examples to attract simulation, or to guide it through extreme artistic transitions. When desired, final results can be reused on another shape, simultaneously preserving the motion and adapting to new geometry.

**Workflow** A typical session begins with a set of artworks that are then equipped with handles. Handle locations determine the triangulation and define the rigging of artwork meshes. Artwork layers are then additionally attached, if desired, by setting positional constraint correspondences between handles. For timing control, per-handle keyframing begins the roughing out of an animation. Interactive preview of the current animation is then available at all times. To further embellish the animation, artists can add example shapes by editing existing artworks and so support desired deformation via example-based simulation. If a desired animation is not realizable by a smooth deformation of an artwork mesh, new artworks are imported, and handle and/or vertex correspondences are set by direct selection on the meshes. With correspondences in place, artist-directed shape transitions are generated to map between artworks and enable exaggerated, physically based animation. Finally, artists can copy and paste to retarget animation from one artwork to another by selecting desired regional correspondences between the two.

### 3.1 Background

We begin with a set of artworks each given as an arbitrary subregion $\Omega \in \mathbb{R}^2$ discretized as a triangle mesh with $n$ rest pose vertices $\bar{\mathbf{x}}_i \in \mathbb{R}^2$ and corresponding deformed vertices $\mathbf{x}_i$.

**Linear Blend Skinning** We equip each artwork mesh with control handles and corresponding affine transforms $\mathbf{T}_j$. The deformation map from handles to mesh vertices is completed by skinning weights $\alpha_{ij} \in \mathbb{R}$ specifying the influence of each handle $\mathbf{T}_j$ on vertex $\mathbf{x}_i$. Linear blend skinning (LBS) deformation of mesh vertices

$\mathbf{x}_i$ for handles $\mathbf{T}_j$ is then

$$\mathbf{x}_i = \sum_{j=1}^{m} \alpha_{ij} \mathbf{T}_j \begin{pmatrix} \bar{\mathbf{x}}_i \\ 1 \end{pmatrix}. \tag{1}$$

Vectorizing the affine transforms $\mathbf{t}_j = \mathrm{vec}(\mathbf{T}_j)$ and concatenating give us handle state $\mathbf{t} = (\mathbf{t}_1^\mathsf{T}, .., \mathbf{t}_m^\mathsf{T})^\mathsf{T}$ per artwork. Concatenating vertices as $\mathbf{x} = (\mathbf{x}_1^\mathsf{T}, .., \mathbf{x}_n^\mathsf{T})^\mathsf{T}$ we then define the LBS map equivalently in matrix-vector form as

$$\mathbf{x} = \mathbf{U}\mathbf{t}. \tag{2}$$

**Variational Time Stepping** Time stepping to state at time $t$ with implicit Euler can be cast in variational form (see e.g., Martin et al. [2011]) by minimizing the incremental potential

$$\min_{\mathbf{x}^t} \tfrac{1}{2h^2}(\mathbf{x}^t - \mathbf{x}^\mathrm{p})^\mathsf{T}\mathbf{M}(\mathbf{x}^t - \mathbf{x}^\mathrm{p}) + \sum_k E_k(\mathbf{x}^t) + \sum_l \mathbf{f}_l^\mathsf{T}\mathbf{x}^t, \tag{3}$$

where $\mathbf{x}^\mathrm{p} = 2\mathbf{x}^{t-1} - \mathbf{x}^{t-2}$ is a predicted state, $h$ is timestep size, $\mathbf{M}$ is the discretized mass matrix, $E_k$ are internal energies, and $\mathbf{f}_l$ are external forces, such as gravity. Stationarity of (3) with standard elastic energies in $E_k$ is equivalent to standard implicit Euler equations of motion. However, as we can add arbitrary energies to the incremental potential, the variational form enables a simple, flexible and easily generalizable framework for artist-directed dynamics.

### 3.2 Timing Control

Artists should be able to easily control the timing of any spatial region, in any subset of frames, with an arbitrary trajectory. We then want simulation to promptly fill in any unspecified portions of trajectory with physically plausible motion. In setting keyframes, artists should not need to concern themselves with physical limitations, mesh connectivity or unnecessary restrictions on their ability to set partial keyframes. Control of timing in animation should thus be intuitive, interactive and sparse as desired.

For usability we project our dynamics directly onto a reduced LBS handle space. Forming a reduced dimensional time stepper in the handle subspace enables artists to employ a small number of handles to rig, control, simulate and, as we will see later in Section 3.4, retarget animations with a consistent low dimensional representation. To project dynamics onto the handle subspace we construct the reduced variational time stepper utilizing (2) to get

$$\begin{aligned}\min_{\mathbf{t}^t} \tfrac{1}{2h^2}(\mathbf{t}^t - \mathbf{t}^\mathrm{p})^\mathsf{T}\tilde{\mathbf{M}}(\mathbf{t}^t - \mathbf{t}^\mathrm{p}) + \sum_k E_k(\mathbf{U}\mathbf{t}^t) \\ + \sum_l \tilde{\mathbf{f}}_l{}^\mathsf{T}\mathbf{t}^t + \sum_p E_p(\mathbf{t}^t),\end{aligned} \tag{4}$$

where $\mathbf{t}^\mathrm{p} = 2\mathbf{t}^{t-1} - \mathbf{t}^{t-2}$ and reduced mass and external forces are given by $\tilde{\mathbf{M}} = \mathbf{U}^\mathsf{T}\mathbf{M}\mathbf{U}$ and $\tilde{\mathbf{f}}_l = \mathbf{U}^\mathsf{T}\mathbf{f}_l$ respectively.

With reduced dynamics in hand, the variational stepper (4) now includes energies $E_p$ directly on handles. We will use these handle energies as our means of directing dynamics. For example, to enable arbitrarily sparse timing control of keyframes in space and time, we simply add, per timestep, soft constraint potentials

$$E(\mathbf{t}^t) = \frac{1}{2}\beta_j \|\mathbf{t}_j^t - \mathbf{k}_j^t\|^2, \tag{5}$$

to the stepper energy (4) where $\mathbf{k}_j^t$ is a desired handle value at time $t$ for handle $j$ specified by the artist with weighting strength $\beta_j$. We will include other potential energies in $E_p$ later in this section.

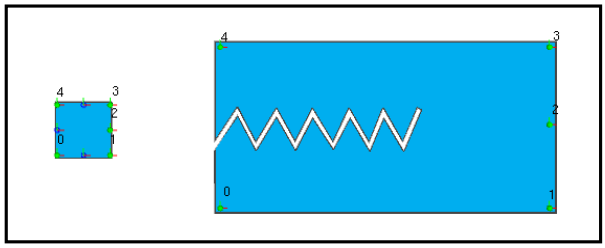


**Figure 2:** ***Transition mapping between two example-based manifolds.*** *The handles with the same number are used to specify the correspondence between the rest shape of a square and the rest shape of a "crocodile".*

### 3.3 Artist-Directed Shape Transitions

As with timing, artists should likewise be able to directly control the deformation shapes generated by simulation in (4). Example-based simulation [Martin et al. 2011; Koyama et al. 2012; Coros et al. 2012; Bouaziz et al. 2014] constructs elastic materials drawn to rest *subspaces* formed from artist provided poses. Dynamics are then attracted to the nearest shape in the subspace at each timestep rather than a single rest shape. Example-based materials are a natural and intuitive starting point for our system as they allow artists to drive deformation dynamics by simply creating artwork examples.

Starting with a base rest-pose mesh $\bar{\mathbf{x}}^1$ and a set of additional rest shape artworks $\bar{\mathbf{x}}^k,\ k \in [2, e]$ deformed from $\bar{\mathbf{x}}^1$, an example manifold $\mathcal{E}$ of realizable shapes is constructed. We start with a standard elastic potential $E_\mathrm{I}(\bar{\mathbf{x}}, \mathbf{x})$ that is attracted to a single rest shape $\bar{\mathbf{x}}$. The corresponding example-based elastic energy that attracts instead to the closest rest shape in the example manifold is then

$$E_\mathrm{E}(\mathbf{x}) = E_\mathrm{I}(\bar{\mathbf{x}}_w, \mathbf{x}) \quad s.t. \quad \bar{\mathbf{x}}_w = \operatorname*{argmin}_{\bar{\mathbf{x}} \in \mathcal{E}} \|\bar{\mathbf{x}} - \mathbf{x}\|^2. \tag{6}$$

**Example Manifold in Handle Space** We construct our example manifold directly in handle subspace to continue to enable animation manipulation and creation directly on handle DOFs.

For each example pose $\mathbf{x}^k$ of an LBS rigged artwork, we have the corresponding handle state $\mathbf{t}^k$ and interpolatory weight $w^k \in \mathbb{R}$. Identifying $\mathbf{x}^0 = \bar{\mathbf{x}}$ and recalling the LBS subspace construction, the blend shape $\mathbf{x}^\mathrm{E} \in \mathbb{R}^{2n}$ in the example manifold at $\mathbf{w} = (w^0, ...w^e)^\mathsf{T}$ is simply

$$\mathbf{x}^\mathrm{E}(\mathbf{w}) = \mathbf{U}(\sum_{k=0}^{e} w^k \mathbf{t}^k), \tag{7}$$

so that the blend-shape handle rig is correspondingly

$$\mathbf{t}^\mathrm{E}(\mathbf{w}) = \sum_{k=0}^{e} w^k \mathbf{t}^k. \tag{8}$$

The handle-space, example-based energy is then simply minimization on weights $\mathbf{w}$

$$\begin{aligned}E_\mathrm{E}(\mathbf{t}^t) =& E_\mathrm{I}(\mathbf{x}^\mathrm{E}(\alpha), \mathbf{U}\mathbf{t}^t) \\ & s.t. \quad \alpha = \operatorname*{argmin}_{\mathbf{w}} \left\|\mathbf{U}\left(\mathbf{t}^t - \mathbf{t}^\mathrm{E}(\mathbf{w})\right)\right\|^2, \quad \mathbf{x}^\mathrm{E}(\mathbf{w}) \in \mathcal{E}.\end{aligned} \tag{9}$$

**Example-Based Materials and Transition Maps** While an example-manifold enriches the range of animations we can create with simulation, we observe that artists quickly run up against the barriers imposed by a single example-based elastic-material. Simulated example-based materials capture a wide range of deformations but can not realize shapes that are not realizable by a smooth deformation of the rest-pose mesh. Artists, however, naturally explore and would like to enrich simulations with arbitrarily shaped artworks to support jumps over topology changes and often large exaggerations. On the other hand, artists also have a clear sense of how correspondences should be maintained between shapes across such large jumps.

To support the artist's inclination to create keyframe artworks with *arbitrary* exaggerations, we extend example-based simulation with a transition map. Artists can now construct a set of example-based manifolds $\mathcal{K} = \{\mathcal{A}, \mathcal{B}, \mathcal{C}, ...\}$ each from an independent artwork with a possibly varying topology and/or handle rigging. We treat each example-based manifold as a single deformation family of shapes that can be derived from one another using provided handle rigging. We then define transition mapping between these disconnected manifolds that are otherwise unreachable from one another due to rips, differing handles, or extreme deformations.

Each deformation family $\mathcal{A}$ can have its own unique mesh topology, vertices $\mathbf{x}^{\mathcal{A}}$ and handle rigging $\mathbf{t}^{\mathcal{A}}$. Our system then requires the artist to specify a correspondence across the rest shapes of the deformation families by matching a subset of handle-to-handle parameters (e.g., position, rotation) with $\Omega_{(\mathcal{A},\mathcal{B})}\mathbf{t}^{\mathcal{A}} \leftrightarrow \Omega_{(\mathcal{B},\mathcal{A})}\mathbf{t}^{\mathcal{B}}$ and/or vertex-to-vertex correspondences with $\Psi_{(\mathcal{A},\mathcal{B})}\mathbf{x}^{\mathcal{A}} \leftrightarrow \Psi_{(\mathcal{B},\mathcal{A})}\mathbf{x}^{\mathcal{B}}$. For example in Figure 2, the correspondence between the rest shape of a square and the rest shape of a "crocodile" is specified by pairs of handles (illustrated with the same number).

Given current state $\mathbf{t}^{\mathcal{A}} = \mathbf{t}^t$ in the current deformation family $\mathcal{A}$, we maintain a time varying closest shape correspondence to each family $\mathcal{B}$ by weighting the least squares correspondences with the elastic energy of the example-based material $\mathcal{B}$ given by $E_{\mathrm{E}}^{\mathcal{B}}$. The handle configuration $\mathbf{t}^{\mathcal{B}}$ that realizes this closest point shape in $\mathcal{B}$ is then a transition *candidate* and is found at each timestep by

$$\begin{aligned}\mathbf{t}^{\mathcal{B}} = \underset{\mathbf{t}}{\operatorname{argmin}} \; & \|\Omega_{(\mathcal{A},\mathcal{B})}\mathbf{t}^{\mathcal{A}} - \Omega_{(\mathcal{B},\mathcal{A})}\mathbf{t}\|^2 \\ & + \|\Psi_{(\mathcal{A},\mathcal{B})}\mathbf{U}^{\mathcal{A}}\mathbf{t}^{\mathcal{A}} - \Psi_{(\mathcal{B},\mathcal{A})}\mathbf{U}^{\mathcal{B}}\mathbf{t}\|^2 \\ & + \lambda\, E_{\mathrm{E}}^{\mathcal{B}}(\mathbf{U}^{\mathcal{B}}\mathbf{t}).\end{aligned} \tag{10}$$

where we set $\lambda = 10^{-2}$ to weight shape correspondence against internal deformation energies.

An artistic transition from current example material $\mathcal{A}$ to $\mathcal{B}$ is then applied if the transition candidate minimizes internal energy over all possible deformation families so that

$$\begin{aligned} E_{\mathrm{E}}^{\mathcal{B}}(\mathbf{U}^{\mathcal{B}}\mathbf{t}^{\mathcal{B}}) &< E_{\mathrm{E}}^{\mathcal{A}}(\mathbf{U}^{\mathcal{A}}\mathbf{t}^t) \\ &\text{and} \\ E_{\mathrm{E}}^{\mathcal{B}}(\mathbf{U}^{\mathcal{B}}\mathbf{t}^{\mathcal{B}}) &\leq E_{\mathrm{E}}^{\mathcal{K}}(\mathbf{U}^{\mathcal{K}}\mathbf{t}^{\mathcal{K}}),\ \forall \mathcal{K}. \end{aligned} \tag{11}$$

Upon transition we update state with $\mathbf{t}^t \leftarrow \mathbf{t}^{\mathcal{B}}$ while the velocity-like term $\mathbf{t}^{\mathrm{p}} = 2\mathbf{t}^{t-1} - \mathbf{t}^{t-2}$, required for time stepping with the variational solve of (4), is updated to the new deformation family using the previously computed candidate transitions to $\mathcal{B}$ from the last two time steps. Observe that we have constructed our transition to a new family if the deformation energy *strictly* decreases and so effectively have found an improved representation for deformation in the new family. This formulation is reminiscent of strategies employed in variational adaptive remeshing [Mosler and Ortiz 2007] where a similar metric is applied to argue when a jump to a new basis is warranted by better minimization of deformation energies.

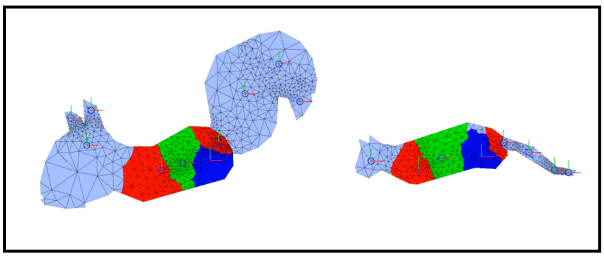

**Figure 3:** ***Corresponding regions between two meshes.*** *The regions with the same color are the corresponding regions between two meshes. The region is automatically determined by the user-specified handles and the threshold of the bounded biharmonic weights.*

### 3.4 Simulation Copy and Paste

Iterating to a complete, satisfying animation will always take time and effort. While our system so far enables the rapid design, edit, and refinement of animations, we also wish to be able to copy and paste carefully crafted motions to new artworks.

We propose "copy-and-paste physics" to enable this easy transfer and reuse of artistic assets. The first step is to create the desired correspondence between source animation artwork and a target asset artwork. As in artistic transitions there are many possible and often equally desirable correspondences available between artworks. We currently support an interactive tool that allows artists to guide and explore various copy and paste correspondences.

Here again handle level DOFs and LBS subspace provide a quick and intuitive interface for creating and working with correspondences between artwork assets. Artists select corresponding handles between artworks and then set thresholds for the blend weights to define region-to-region correspondences. For example in Figure 3, to copy and paste from a cat to a squirrel, we select one handle at the middle of the cat body and set the weight threshold to 0.4. All vertices with weight greater than 0.4 in the cat body are automatically selected (illustrated as the green region).

With regional correspondences completed we can copy and paste with a TRACKS [Bergou et al. 2007] constraint to enforce region-to-region centroid matching between source $\mathcal{S}$ and target $\mathcal{T}$ artworks. The handle-space TRACKS constraint to match centroid of target artwork to the centroid of a source animation per user specified regions $K$ is then

$$g_K(\mathbf{t}) = \|\mathbf{S}_K^{\mathcal{S}}\mathcal{M}^{\mathcal{S}}\mathbf{U}^{\mathcal{S}}\mathbf{t}^{\mathcal{S}} - \mathbf{S}_K^{\mathcal{T}}\mathcal{M}^{\mathcal{T}}\mathbf{U}^{\mathcal{T}}\mathbf{t}\|, \tag{12}$$

where $\mathbf{S}_K$ is the selection matrix that extracts the $r$ vertices in region $K$ and $\mathcal{M}$ is the Petrov-Galerkin mass matrix [Bergou et al. 2007] computing the weighted average.

However, when the scales of source and the target objects vary, target objects will be undesirably stretched and squeezed by centroid translation constraints alone. Moreover, the translation of a centroid does not fully describe nor prescribe the deformation of corresponding regions. We demonstrate these issues in Figure 4 (center panel) and observe that constraining centroid translation alone can not and will not induce a similar deformation to the source.

We require additional, readily available constraints on the local deformation to make the constraint mapping robust between disparate

**Figure 4:** ***Comparison of different constraints in simulation retargeting.*** *Left: The corresponding regions between two rest shapes. Middle: With only centroid translation constraints the retargeted lion tail is overly stretched and does not deform with the source motion (illustrated as the red mesh). Right: With linear transformation constraints the retargeted lion tail closely resembles the deformation on the source object.*

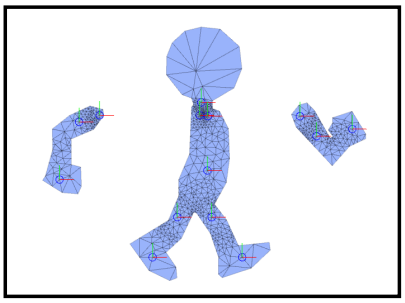

**Figure 5:** ***The meshes and handles used in the articulated character example.*** *The articulated character consists of three artworks. The blue circles represent the handle positions. The coordinate frames shown as red and green line segments represent linear transformation of the handles.*

shapes. For each correspondence region $K$, we find a close-as-possible affine map $\mathbf{A}_K$ that deforms the rest shape $\bar{\mathbf{x}}^{\mathcal{S}}$ of the region to its current shape:

$$\min_{\mathbf{A}_K} \|\mathbf{A}_K \Delta(\bar{\mathbf{x}}^{\mathcal{S}}) - \Delta(\mathbf{x}^{\mathcal{S}})\|^2, \tag{13}$$

where the matrix operator $\Delta$ composes column-wise per vertex in region $K$ with the vertex to centroid difference $\Delta(\mathbf{x}) = [\mathbf{x}_1 - \mathbf{c}_K(\mathbf{x}), ..., \mathbf{x}_r - \mathbf{c}_K(\mathbf{x})]$, where $\mathbf{c}_K(\mathbf{x})$ gives the centroid of region $K$.

The corresponding constraint to match the transform in the target is then

$$h_K(\mathbf{t}) = \|\mathbf{A}_K \Delta(\mathbf{U}\bar{\mathbf{t}}^{\mathcal{T}}) - \Delta(\mathbf{U}\mathbf{t}^{\mathcal{T}})\|. \tag{14}$$

For direct copy and paste while simulating, we construct and add the corresponding constraint penalty energy

$$E_{\mathrm{T}}(\mathbf{t}) = \frac{1}{2}\sum_K \left(\gamma_K g_K(\mathbf{t})^2 + \eta_K h_K(\mathbf{t})^2\right) \tag{15}$$

to the variational solve of (4) where $\gamma_K, \eta_K$ define constraint penalty weights per region (Figure 4, right).

### 3.5 Reduced Projective Dynamics and Implementation

Interactive feedback for fast iteration and refinement is key in designing our physics-based workflow. If we restrict ourselves to internal energies $E_{\mathrm{I}}$ on the mesh that can be formulated as a local-global energy, i.e., a globally quadratic term and locally nonlinear constraint, such as As-Rigid-As-Possible (ARAP) or mass-spring, our reduction to the handle subspace maintains the same local-global structure on the handle DOF. This enables us to construct an efficient projective dynamics solver [Liu et al. 2013; Bouaziz et al. 2014] to minimize the full incremental potential (4) with example-based materials, artist-directed shape transitions, and copy-paste physics at each time step.

In our implementation we employ the simple mass-spring energy for $E_{\mathrm{I}}$ [Liu et al. 2013] (ARAP follows similarly with minimal additional overhead [Alexa et al. 2000; Sorkine and Alexa 2007; Sýkora et al. 2009]) and compute LBS weights for reduced handle coordinates with the bounded biharmonic weights [Jacobson et al. 2011].

Applying a sum of squared edge differences as our similarity metric, the energy for the reduced example-based material energy in (9) simplifies to a quadratic program (QP) in the projection

$$E_{\mathrm{E}}(\mathbf{t}^t) = E_{\mathrm{I}}(\mathbf{x}^{\mathrm{E}}(\alpha), \mathbf{U}\mathbf{t}^t)$$
$$s.t.\ \ \alpha = \operatorname*{argmin}_{\mathbf{w}\in[0,1]^{e+1}} \sum_{(i,j)\in\text{edges}} \|\mathbf{B}_{i,j}(\mathbf{t}^t - \sum_{k=0}^{e} w^k \mathbf{t}^k)\|^2, \tag{16}$$

where $\mathbf{B}_{i,j} = [\mathbf{S}_j - \mathbf{S}_i]\mathbf{U}$, the matrix $\mathbf{S}_i$ gives the $\mathbb{R}^{2\times n}$ selection that extracts vertex $\mathbf{x}_i \in \mathbb{R}^2$, and the bound constraints in the QP ensure that example blend shapes lie within the convex hull of the example domain. We further approximate the QP with Koyama et al.'s [2012] boundary projected solution in which smallest negative weights are iteratively eliminated by zeroing and an even redistribution among remaining weights. While a fully optimal bounded QP solution would be a simple extension we found the boundary projected solution sufficient in all our tests.

## 4 Results

Artist-directed dynamics delivers the benefits of both keyframing and simulation. We first demonstrate the features and workflows of our system for artistic creation. We then use the Twelve Principles of Animation [Thomas et al. 1995] interpreted by a professional animator [Lodigiani 2013] to evaluate the system from each perspective: (1) how keyframing enhances simulation and (2) how simulation enhances keyframing.

### 4.1 Artistic Creation

We use the following five different scenarios to demonstrate the process of artistic creation using our system and showcase the main features of our method: handle control, example-based dynamics, artist-directed shape transition, and simulation copy and paste. Please see the accompanying video for the resultant motion.

**Articulated Figure.** First, we demonstrate that our method applies to articulated characters (Figure 5). We create a walking cycle by keyframing the handles at the hands and the feet of the articulated character, as shown in Figure 6 (nine keyframes per walking cycle). We demonstrate a variety of ways to enrich the simple walking cycle, such as changing the physical properties of the character, modifying the spacing and timing of the handles, or adding examples. For example, we delay the motion of the head by keyframing the handles on the head differently (Figure 1 (a)), or create a "sad" walk and a "sneaky" walk by using example poses (Figure 1 (b)).

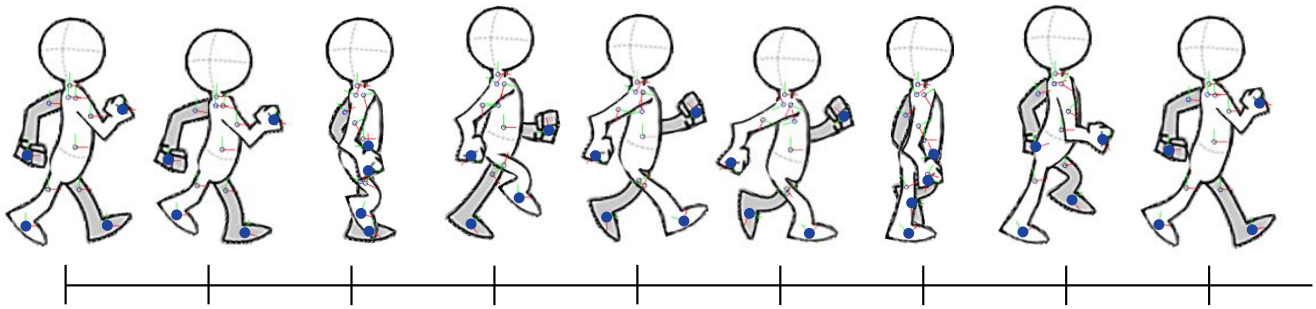

**Figure 6:** ***Keyframes used in the articulated character walk example.*** *The artist only specifies keyframes for a subset of handles (handles at hands and feet) which are shown as blue dots. Nine keyframes are used to create a walking cycle. Their timing is visualized by the black lines at the bottom. The artworks are adapted from Angryanimator.com (http://www.angryanimator.com/)*

**Crocodile.** Elastic deformation helps with the transition, but is not sufficient to create smooth transitions between drastically different examples. In this illustration, the artist intends to transform a square to a crocodile with different mesh topology and handle rigging. This jump cannot be realized by a smooth deformation of a single example-based elastic-material, but can be achieved by artist-directed shape transition. Our system complies by first deforming the square and then, when appropriate, switching to the new shape before continuing simulation.

**Cat.** For the cat example a user first creates a cat-like motion by rigging handles, attaching layers by setting positional constraint correspondences between handles on legs and body, and keyframing the handles on the legs. Once a satisfying cat motion is simulated, the user wishes to reuse this asset by transferring parts of the cat motion to a squirrel, a rabbit, and a mouse. Our system allows the user to pick and choose the desired parts of the motion to transfer and lets the physical simulation determine the remainder of the motion. In this example, the user retargets the trunk and limbs motion from the cat while the physics determines the motion of the tails and ears based on each animal's physical properties (Figure 7). The user can then iteratively refine the motion by adjusting physical properties or repeatedly applying "copy-and-paste". For example, the user can tweak the stiffness of the mouse tail or copy the tail motion of the squirrel to the mouse.

**Bouncing Ball.** We start from a light and rigid ball bouncing forward. The position of the middle handle is constrained to the trajectory of a simulated bouncing particle. All the other handles are governed by the physical simulation and contact constraints. By simply adjusting the simulation parameters, such as initial height, velocity, and coefficient of restitution, we achieve different motions of the bouncing ball accordingly. For example, we increase the height of the floor in the particle simulation. This change makes the ball squashed more when it is in contact with the floor. To accentuate the anticipation behavior, one effective strategy is to let the ball stretch out before contacting the ground. To do so, we add stretch and squash example poses and set the keyframes on top and bottom handles to deform the ball and control the timing. To illustrate our simulation copy and paste, we reuse the simulation results on two new characters: Lola and Sam. We copy and paste the motion of the bouncing ball to the ball on which Lola sits. The reactive motion of the other parts of Lola such as legs and head are governed by physics. We also copy and paste the motion of the ball to the legs of Sam so that the legs stretch and squash in a similar way as the bottom part of the ball. The motion of the other part of Sam is mainly due to physics simulation, such as the responsive motion on Sam's belly.

**Walking Arc.** The bouncing ball examples showcase many features of our system but does not require artistic transition. In this example, we enable smooth transitions between two shapes that are difficult to transition between using pure elastic deformation. We first demonstrate that a seemingly simple task of transitioning a square shape to an arc shape cannot be done by simply controlling a few handles on the square because the deformed square is highly distorted and nowhere near the shape of the arc. One can add more handles and more keyframes to improve the shape of deformation, at the cost of losing appealing responsive motion from physics simulation. Our solution is to apply artistic transition to produce smooth deformation from a square to an arc without suppressing natural physics effects. In the video, we create a slinky-like walking motion by alternating the shape of arc and the shape of square using our artistic transition method. Other features can also be applied simultaneously. For example, we add another "mountain" shape example to create a different style of slinky.

### 4.2 Evaluation

Our system is designed for limitless customization. We try to explore this range comprehensively in context of the Twelve Principles of Animation. To do that objectively, we attempt to recreate animations of the ten relevant animation principles, which were independently created by another animator [Lodigiani 2013]. The results are shown in Figure 8 and the accompanying video.

**Timing and Spacing.** The dramatic impact of careful speed customizations can be seen in the animation that illustrates the timing and spacing principle. As shown in the accompanying video, the large box moves in different timing and spacing to give the animation a different feel, such as "punch" and "push". In our approach, both the timing and spacing can be controlled precisely and directly by specifying trajectories for some subset of handles.

**Slow In Slow Out.** Slow-in and slow-out, another physically inspired principle, emerges automatically as simulated objects move in response to external forces and collisions. The slow-in slow-out spacing on one of the boxes creates the illusion of faster arrival in addition to making that box appear heavier. When that movement needs to be refined precisely, our technique allows an artist to control speed with keyframes; or, when more convenient, by adjusting the mass, coefficients of friction, and other simulation parameters.

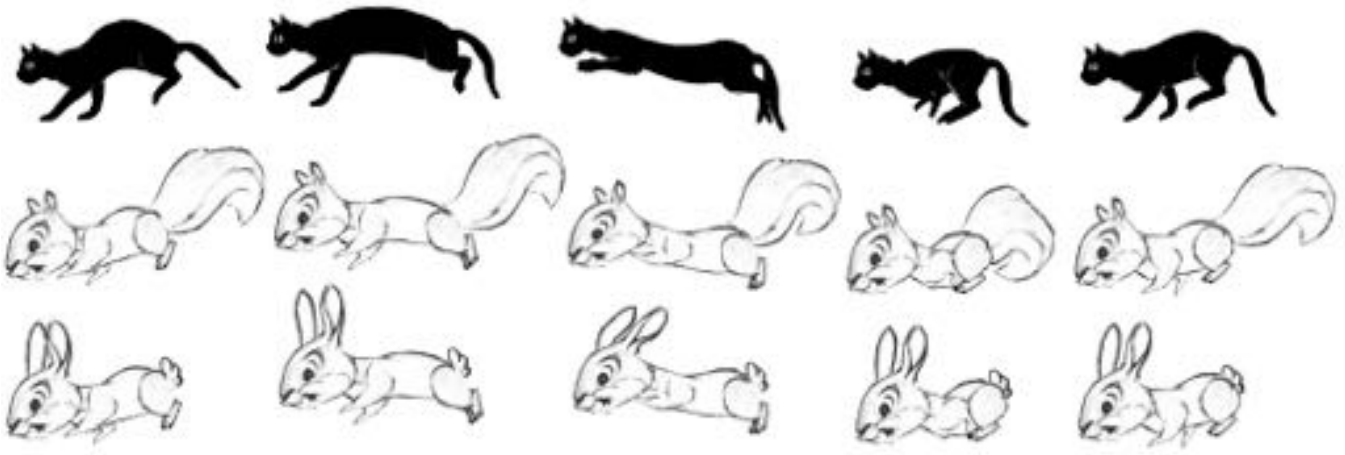

**Figure 7:** ***Illustration of retargeting a cat's trunk and limb motion to a squirrel and a rabbit.*** *The artworks are adapted from The Catterwall (http://thecatterwall.tumblr.com/) and Cartoon Animation [Blair 1994].*

**Squash and Stretch.** Elastic simulation produces squash and stretch automatically. One can modify simulation properties to adjust these deformations, but as already observed that indirect approach can fall short of typically desired customization [Martin et al. 2011]. Furthermore, as seen in animated cartoons, even the elastic model may not be appropriate for some squash-and-stretch animation. This can be easily seen in the corresponding interpretation of squash-and-stretch, where the amount of deformation necessitates *plastic* instead of elastic deformation. In this example, we recreate the extreme squash and stretch effect, by repeatedly expanding and compressing handles at the top and bottom.

**Anticipation.** Preparing for action requires planning instead of simulation. Proper anticipation arises only when simulation is paired with control or corresponding (spacetime or dynamic) constraints. Our system adapts dynamic constraints, allowing the simulation to track trajectories that induce desired anticipation. We illustrate this principle using a square rotating about one corner. We set the orientation constraints on the bottom handle to turn the square back and forth at the beginning in preparation for the final tumbling.

**Follow Through and Overlapping Action.** Analogous to the earlier squash-and-stretch example, simulation yields movement with realistic follow through but without the control needed for further customization and exaggeration. Overlapping action then synchronizes deliberate and often keyframed actions, which initiate the movement, with the follow-through and other reactions that appear in response. We show how our system supports this synergy by emulating sample animation for the corresponding principle. First, we exaggerate the drag with a pin constraint, delaying the action at the top of the rectangle. As the animation progresses, we release the constraint allowing the elastic simulation to take over and recreate corresponding follow through. The initial movement, drag, and follow through can all be further customized either with existing or new constraints.

**Secondary Action.** Simulation is routinely used to add secondary animation automatically. Our system improves on this process with simpler synchronization of primary and secondary actions. In this example, the larger box follows the motion similar to the anticipation example, and we embellish it with the secondary action where the smaller box bounces on its top. The contact force between the big and small boxes is scripted. After the small box bounces twice, we constrain the two bottom corners of the small box so it stays on the big box. The animation of each box can be further refined and adjusted leaving it to constraints to maintain proper synchronization.

**Arc.** Most natural actions tend to follow arc trajectories. The handle trajectories in our system make it possible to express arc paths directly, as shown in the corresponding example. The arc motion for the top and bottom handles initiates the square-to-arc transformation, but because this transition is not a purely elastic deformation, our system also incorporates the arc as another example and uses our artistic transition method to create smooth transitions between these two examples. As one handle drags behind the other, the simulation transitions from square to arc automatically, and when they come back together, the simulation reverts back to the square.

**Exaggeration.** The need to jump to any example is most apparent in the illustration of the exaggeration principle. We refer the reader to the crocodile example described in Section 4.1.

**Straight Ahead and Pose To Pose.** Straight ahead and pose to pose are the two classical animation workflows with complementary benefits and drawbacks. Pose to pose makes it easier to plan out timing, but the result may end up too choppy. Straight-ahead workflow leads to fluid animation, but the timing can wander and drift. We observe similar pros and cons with simulation and keyframing. A straight-ahead simulation of a box fails to catch the small box at the right time and the right place, while pose-to-pose keyframed animation looks too stiff without natural reactive motion. As with the two classical workflows, the hybrid approach leads to a better workflow in our system: one begins with keyframed trajectories with desired timing, and then refines animation either by removing the trajectories, so that simulation fills in fluid motion, or by adding trajectories to constrain simulation as needed.

**Appeal.** The goal of this principle is to establish a connection between the viewer and the animation. The simulation copy and paste feature of our system makes it easier to experiment with different appeals. An artist can draft first animation on a simple geometry,

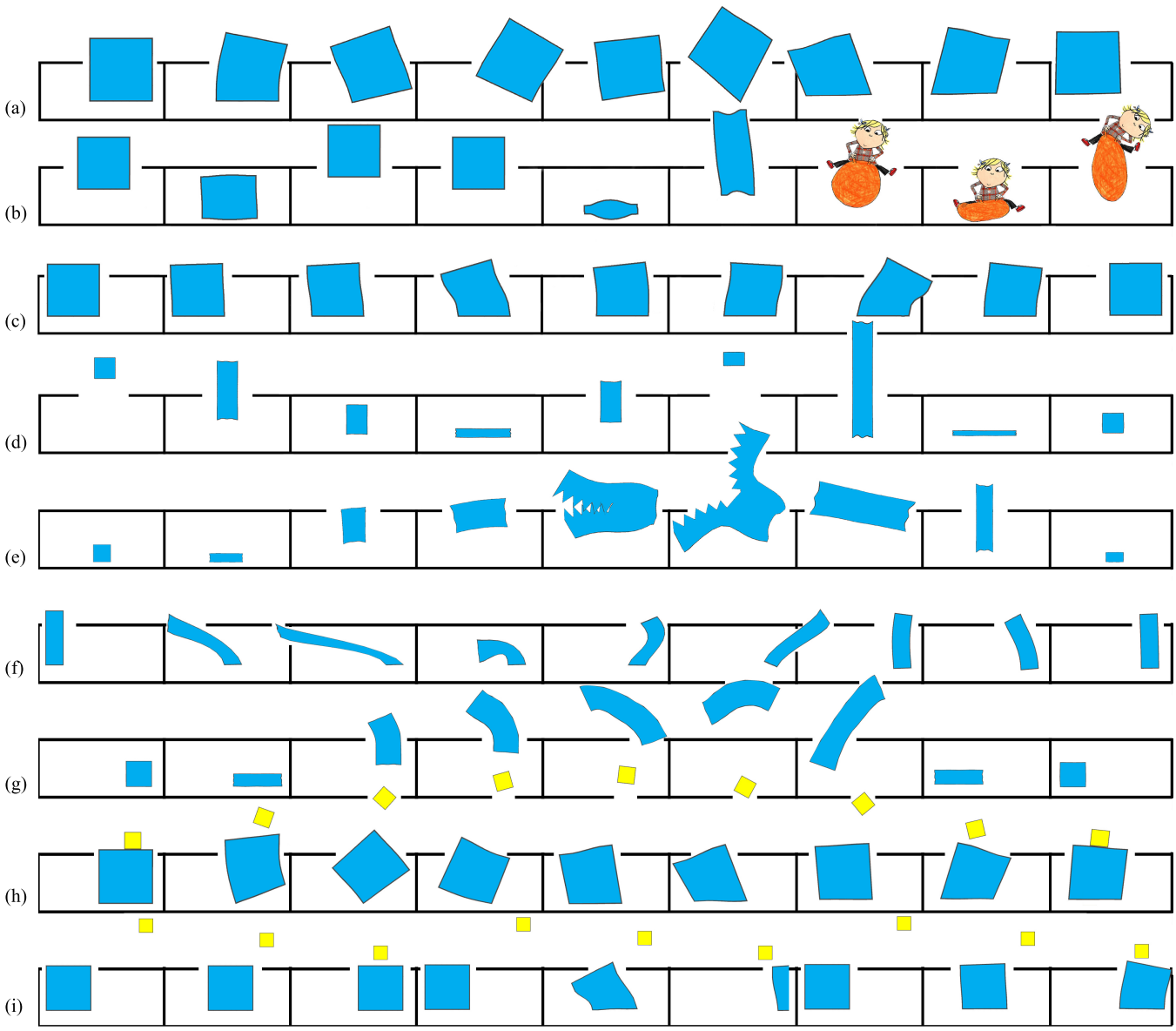


**Figure 8:** ***Illustration of animation principles using our animation system.*** *(a) Anticipation. (b) Appeal. Left three: Bouncing square. Middle three: Bouncing square with example poses. Right three: Retargeted motion on Lola. (c) Slow in slow out. (d) Squash and stretch. (e) Exaggeration. (f) Follow through and overlapping. (g) Arc. (h) Secondary Action. (i) Straight ahead and pose to pose. Left three: Pose-to-pose only. Middle three: Straight-ahead only. Right three: Combined workflows.*

then transfer it to another when seeking to improve appeal. As an example, we retarget the motion of a bouncing square onto Lola, a girl character from a popular story book. The correspondence is established by relating the regions around the handles on each object. Although Lola is sitting on a ball, the copy-paste approach works well for the square. The rest of the body is freely simulated, and could be further tuned with trajectories, examples, wind, and other simulation properties.

### 4.3 Implementation

We implemented our entire framework in C++ with the Eigen library (eigen.tuxfamily.org) for dense and sparse linear algebra. We apply ten iterations of local-global steps to solve Equation 4 in all presented examples and applied Cholesky decomposition to solve the linear system. We employed a time step 0.02s which obtained realtime simulation performance across all examples.

## 5 Conclusion

We have introduced a unified framework for tightly integrating simulation and keyframing—two typically disparate animation workflows. Artists can now fluidly *create*, *customize*, and *retarget* an animation's timing and spacing in a seamless and interactive fashion. The key to making this work was integrating recent developments in *example-based* dynamics for spacing with *constraint-based* dynamics for timing. We further introduced reduced dynamics on handle-based degrees of freedom to provide intuitive artistic exploration of 2D animation and extended the expressive range of

example-based dynamics with *artistic transitions*. The result is a system that satisfies the full range of classical animation goals, as codified by the Twelve Principles of Animation.

Our unified framework with the handle interface introduce new interaction possibilities. Instead of depending on kinematically keyframed trajectories alone, artists can also benefit from other simulations and procedural methods. They could design the center-of-mass trajectory with a simple particle system, before moving on to further refinement. They could also employ more elaborate parameterizations [Kass and Anderson 2008] or other optimization systems [Barbič et al. 2012; Hildebrandt et al. 2012] to define such trajectories. Indeed, even animations of our own animation system can be further nested and composed by using any handle to key the motion of another. Overlapping actions and colliding interactions are most obvious applications, but we also go further to demonstrate simulation retargeting operations on our animations. With these features, artists have both the ability to customize any animation, and then compose and reuse the best of them by reapplying them onto other objects.

**Future Work and Limitations** While artist-directed shape transitions enable maps between disparate shapes they do not guarantee smooth transitions. Artist-directed transitions require suitable example shapes to satisfy animation goals. If artist-provided example shapes are too disparate or too far from desired deformations, results may not be pleasing or may stray from the envisioned animation sequence. As such, the system relies on artists to provide reasonable example shapes to achieve desired deformations. However, we observe that providing reasonable intermediate shapes for pleasing transitions generally follows from artistic intent and fits well with the workflow provided. As an example consider the process of creating the "Walking Arc" example (illustrated in the accompanying video), where a single simulated square is insufficient to capture the desired smooth transitions between a square and arc. If an artist selects a rectangle as a secondary shape for artistic transition, the simulation remains unable to smoothly reach the desired inelastic deformation and the result is unsightly. If instead, the artist adds the desired end-shape arc as an example, artistic-directed transitions are then able to simulate the desired transitions smoothly.

Handle rigging also plays an important role in the success of an animation. As shown in the accompanying video, a square cannot be deformed into an arc with a too small number of handles. We expect that automated handle-rigging, targeted to satisfy artist intent, will be an important direction of future research. Alternatively, as discussed above, we also observe that artist-directed transitions can be applied in these cases to deform via intermediate shapes when a desired deformation cannot otherwise be achieved. Likewise, recent work on generalizing LBS with point handles [Wang et al. 2015] could also potentially be applied in our system to provide suitable DOF for desired deformations.

Simulation copy-and-paste builds on deformation transfer and TRACKS-based dynamic constraints. However, as with all dynamic constraint approaches, retargeting with constraints that violate a physical system's symmetries will not and can not preserve momentum.

For evaluation we have developed a prototype system that is not comprehensive. In presented examples we currently apply pin constraints to mimic inelastic collisions. Frictional contact modeling can and should be added to the simulation dynamics in (4) for automatic contact handling. Similarly, rendered silhouettes are currently jaggy (see e.g., Arc examples) when low resolution meshes are employed. Boundary smoothing can and should be applied in such cases.

Finally, while the tools developed in this paper have been evaluated for the creation of 2D animation, we note that the core techniques we utilize, including projective dynamics and example-based materials, have been demonstrated on thin shells and volumetric shapes. This points to promising avenues for future work in 3D both to examine the system proposed here for its suitability for thin shells and to enable practical application on volumetric shapes.

## Acknowledgements

We thank the anonymous reviewers for their helpful comments. We also thank Wenhao Yu for image editing.

## References

ALEXA, M., COHEN-OR, D., AND LEVIN, D. 2000. As-rigid-as-possible shape interpolation. In *Proceedings of ACM SIGGRAPH 2000*, Annual Conference Series, 157–164.

BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Transactions on Graphics (TOG) 31*, 4, 70.

BARZEL, R., AND BARR, A. H. 1988. A modeling system based on dynamic constraints. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, ACM SIGGRAPH, Annual Conference Series, 179–188.

BARZEL, R., HUGHES, J. F., AND WOOD, D. N. 1996. Plausible motion simulation for computer graphics animation. In *Computer Animation and Simulation '96*, Proceedings of the Eurographics Workshop, 184–197.

BERGOU, M., MATHUR, S., WARDETZKY, M., AND GRINSPUN, E. 2007. TRACKS: Toward directable thin shells. *ACM Transactions on Graphics 26*, 3 (July), 50:1–50:10.

BLAIR, P. 1994. *Cartoon Animation*. Walter Foster Publishing.

BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG) 33*, 4, 154.

CHENNEY, S., AND FORSYTH, D. A. 2000. Sampling plausible solutions to multi-body constraint problems. In *Proceedings of ACM SIGGRAPH 2000*, Annual Conference Series, 219–228.

COROS, S., MARTIN, S., THOMASZEWSKI, B., SCHUMACHER, C., SUMNER, R., AND GROSS, M. 2012. Deformable objects alive! *ACM Transactions on Graphics 31*, 4, 69:1–69:9.

FAURE, F., GILLES, B., BOUSQUET, G., AND PAI, D. K. 2011. Sparse meshless models of complex deformable solids. *ACM transactions on graphics (TOG) 30*, 4, 73.

GILLES, B., BOUSQUET, G., FAURE, F., AND PAI, D. K. 2011. Frame-based elastic models. *ACM transactions on graphics (TOG) 30*, 2, 15.

HA, S., MCCANN, J., LIU, C. K., AND POPOVIC, J. 2013. Physics storyboard. *Computer Graphics Forum (Eurographics) 32*.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Transactions on Graphics 31*, 4, 72:1–72:8.

HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2012. Interactive spacetime control of deformable objects. *ACM Transactions on Graphics (TOG) 31*, 4, 71.

JACOBSON, A., BARAN, I., POPOVIC, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph. 30*, 4, 78.

JACOBSON, A., BARAN, I., KAVAN, L., POPOVIĆ, J., AND SORKINE, O. 2012. Fast automatic skinning transformations. *ACM Trans. Graph. 31*, 4 (July), 77:1–77:10.

JONES, B., POPOVIC, J., MCCANN, J., LI, W., AND BARGTEIL, A. 2013. Dynamic sprites. In *Proceedings of Motion on Games*, ACM, New York, NY, USA, MIG '13, 17:39–17:46.

KASS, M., AND ANDERSON, J. 2008. Animating oscillatory motion with overlap: wiggly splines. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 28.

KOYAMA, Y., TAKAYAMA, K., UMETANI, N., AND IGARASHI, T. 2012. Real-time example-based elastic deformation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, SCA '12, 19–24.

LIU, C. K., AND POPOVIĆ, Z. 2002. Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics 21*, 3 (July), 408–416.

LIU, T., BARGTEIL, A. W., O'BRIEN, J. F., AND KAVAN, L. 2013. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG) 32*, 6, 214.

LODIGIANI, V., 2013. The illusion of life. http://the12principles.tumblr.com.

MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. *ACM Trans. Graph. 30*, 4 (July), 72:1–72:8.

MOSLER, J., AND ORTIZ, M. 2007. Variational h-adaption in finite deformation elasticity and plasticity. *International Journal for Numerical Methods in Engineering 72*, 5, 505–523.

POPOVIĆ, J., SEITZ, S. M., ERDMANN, M., POPOVIĆ, Z., AND WITKIN, A. 2000. Interactive manipulation of rigid body simulations. In *Computer Graphics (Proceedings of SIGGRAPH 2000)*, ACM SIGGRAPH, Annual Conference Series, 209–218.

SCHUMACHER, C., THOMASZEWSKI, B., COROS, S., MARTIN, S., SUMNER, R., AND GROSS, M. 2012. Efficient simulation of example-based materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, SCA '12, 1–8.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, vol. 4.

SUMNER, R. W., AND POPOVIĆ, J. 2004. Deformation transfer for triangle meshes. *ACM Transactions on Graphics 23*, 3 (Aug.), 399–405.

SÝKORA, D., DINGLIANA, J., AND COLLINS, S. 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.

THOMAS, F., JOHNSTON, O., AND THOMAS, F. 1995. *The illusion of life: Disney animation*. Hyperion New York.

TWIGG, C. D., AND JAMES, D. L. 2007. Many-worlds browsing for control of multibody dynamics. *ACM Transactions on Graphics 26*, 3 (July), 14:1–14:8.

WANG, Y., JACOBSON, A., BARBIČ, J., AND KAVAN, L. 2015. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG) 34*, 4, 57.

WITKIN, A., AND KASS, M. 1988. Spacetime constraints. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, vol. 22, 159–168.