Stylized Keyframe Animation of Fluid Simulations

Mark Browning
Princeton University

Connelly Barnes
University of Virginia

Samantha Ritter
Princeton University

Adam Finkelstein
Princeton University

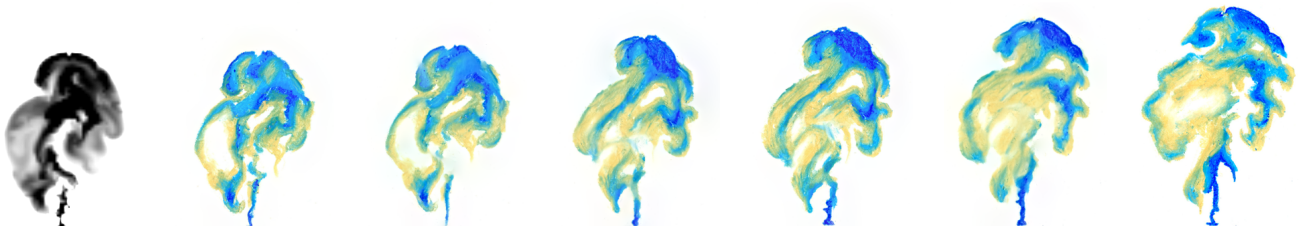

Figure 1: *An upward-flowing jet of ink rendered in oil pastel. An underlying simulation is used as input to guide the artwork (frame 45 is shown). Frames 50, 60 and 70 are hand-drawn keyframes, while the remaining in-betweens were generated automatically by our system.*

Abstract

We present a method that combines hand-drawn artwork with fluid simulations to produce animated fluids in the visual style of the artwork. Given a fluid simulation and a set of keyframes rendered by the artist in any medium, our system produces a set of in-betweens that visually matches the style of the keyframes and roughly follows the motion from the underlying simulation. Our method leverages recent advances in patch-based regenerative morphing and image melding to produce temporally coherent sequences with visual fidelity to the target medium. Because direct application of these methods results in motion that is generally not fluid-like, we adapt them to produce motion closely matching that of the underlying simulation. The resulting animation is visually and temporally coherent, stylistically consistent with the given keyframes, and approximately matches the motion from the simulation. We demonstrate the method with animations in a variety of visual styles.

CR Categories: I.3.4 [Computer Graphics]: Graphics Utilities

Keywords: non-photorealistic rendering, animation, fluids

1 Introduction

Recent advances in computer simulation have allowed animators to enhance 3D animations with realistic motion for smoke, water, plants, cloth or hair that would be painstaking to achieve by hand. Moreover, these elements are typically not the visual focus of the animation but rather provide secondary motion to augment that of the primary subject of the scene. Even if artists *could* adjust every aspect of the pose of every hair on a character, for example, their efforts would be better spent on the movement of the character itself. Thus, simulation of realistic motion for natural phenomena has come to play an important role in the 3D animation pipeline. On the other hand, its use is relatively uncommon in 2D hand-drawn animation, because it is difficult to match the rendering of simulated phenomena to artists' stylized work – realistic smoke rendered over hand-drawn fire would look wrong.

This paper introduces a method that combines hand-drawn artwork with fluid simulations to produce animated fluids with the visual style of the artist's drawings. Of course, it is possible for artists to animate fluids like smoke and water entirely by hand – water, smoke and fire appeared in hand-drawn animations before the use of computers [Thomas and Johnston 1987; Solomon 1994; Gilland 2009]. The system we introduce is therefore designed primarily to divert artists' efforts from tedious replication of faithful movement for these secondary elements so they can expend that effort on the major characters instead.

Creating fluid animation in an artist's chosen style poses several challenges. First, the workflow should be familiar to artists and provide sufficient control over a broad aesthetic range. Second, the computer-generated frames should be visually consistent with the given style. Third, the motion in the scene should be both temporally coherent and plausibly fluid-like. To address these challenges, we choose a keyframe-based approach in which the artist draws some frames "rotoscoped" to match an underlying simulation, and then the computer synthesizes in-between frames. This approach is familiar to artists and gives them control over the choice of keyframes, style, and medium. There remains a tension between the goals of faithfully reproducing the artist's chosen style and coherent motion in the scene, which we address via a recent technique called "regenerative morphing" [Shechtman et al. 2010; Darabi et al. 2012]. Regenerative morphs produce transitions between pairs of keyframes by optimizing for both visual and temporal coherence. Unfortunately, direct application of these methods produces motion that is generally not fluid-like and has no relation to the simulation from which the keyframes have been drawn. To address this problem, we adapt a method from the original regenerative morphing paper – using sparse feature correspondences between keyframes to guide the motion – by incorporating a dense set of priors derived from the underlying simulation that guide the optimization towards the desired motion.

Regenerative morphs with these motion priors are stylistically and temporally coherent and match the guiding flow between pairs of keyframes. Simply concatenating these morphs between successive pairs of keyframes produces animations that exactly interpolate the given keyframes, and for some inputs are pleasing. In other cases, however, misregistration of the keyframes with the underlying flow and the constraint of exact interpolation can produce sequences that exhibit derivative discontinuities around

Figure 2: *Same jet from Figure 1, drawn in a challenging pen and ink style. Due to many disparate features at multiple scales, including curls, sharp lines, and textured gradients, this is a difficult style both for the artist to draw keyframes and for the morph to synthesize in-betweens.*

the keyframes. To address these issues, we extend the notion of the two-keyframe regenerative morph to a multi-keyframe morph that approximates the intermediate keyframes, with parameters that control the tradeoff between interpolation and continuity. For this global morph, each resulting frame depends on many other frames, so the sequential optimization approaches used by existing methods would take days to converge even for relatively short animation sequences. By relaxing some sequential dependencies, we show that the optimization can be parallelized without adversely affecting the results, and with only moderate memory requirements for any stage. On a cluster with as many CPU cores as frames in the animation sequence, the time to compute the entire sequence is the same as that of a single frame.

The primary contribution of this paper is to demonstrate that it is possible to create coherent, stylized animations of fluids, using a workflow in which the style of automatic in-betweens is derived from hand-drawn keyframes and the motion is derived from a simulation. To our knowledge, we describe the first method that could be used in practice for this purpose. Our approach leverages recent advances in patch based morphing, but we offer three adaptations and improvements to these methods for our application: (1) use of dense motion priors based on an underlying simulation, (2) a generalization to approximating multi-keyframe morphs that avoids temporal discontinuities around keyframes, (3) a parallel approach to the multi-keyframe morph that allows it to be computed with reasonable latency on a cluster. Finally, we demonstrate our method with animations in a variety of visual styles.

2 Related work

Fluid simulation has been a highly active area of research in 3D computer animation. Our system relies on a realistic or plausible fluid simulation, both to guide the animation towards fluid-like motion and (optionally) to give the artist a starting point for drawing keyframes. Pioneering work by Stam [1999] introduced "stable fluids," an approach to provide physically plausible fluid motion at interactive frame rates, which is important for artistic control in applications like ours. Treuille et al. [2003] showed how fluid simulations could be coerced to approximate user-provided keyposes, but this required an expensive global optimization. Fattal and Lischinski [2004] described a more efficient local approach that guides fluid from one pose to another. The results shown in this paper are based on simulations produced using Stam's stable fluids, though the approach would naturally work with other simulators.

Our work builds on a thread of research for combining computer-generated elements with hand-drawn 2D animation, for example textures or shadows in cel animation [Corrêa et al. 1998; Petrović et al. 2000]. Zhu et al. [2011] presented a system that combines sketches with a fluid model to produce rich, interactive illustrations. The work of Jain et al. [2012] incorporates simulated 3D elements into hand-drawn sequences by optimizing for unknown 3D parameters like camera and depth. Our method complements this line of research by seeking to visually match the simulated elements to the look of hand-drawn art in the style of the given keyframes.

Researchers have addressed non-photorealistic rendering (NPR) of fluids, for example cartoon rendering for liquid [Eden et al. 2007; You et al. 2009; Yu et al. 2007], smoke [He and Xu 2005; McGuire and Fein 2006; Selle et al. 2004], and fire [Di Fiore et al. 2004]. Largely to address the challenge of temporal coherence in animation, this thread of research tends toward a particular cartoon-like aesthetic. In contrast, our work incorporates keyframes drawn by an animator, with two main benefits: the overall style is provided (by example) and can span a broad range of aesthetics, and the animator controls the look of chosen keyframes. Kwatra et al. [Kwatra et al. 2007] address texture synthesis with temporal coherence on the surface of fluids, though not strictly for NPR.

To construct in-betweens, our method builds morphs between successive pairs of hand-drawn keyframes. Since their introduction by Beier and Neely [1992], morphs have required substantial human intervention to establish feature correspondences and avoid ghosting artifacts [Wolberg 1998]. Mahajan et al. [2009] addressed these issues with moving gradients, a fully automatic method for interpolating between a pair of similar photographs or video frames that produces a plausible animation. Shechtman et al. [2010] recently introduced regenerative morphs, which optimize over many possible feature correspondences to provide smooth transitions that avoid ghosting, and their method was further improved by Darabi et al. [2012] by further expanding the search space. Used without modification, these methods can provide high-quality stills in a morph between neighboring keyframes, but the apparent motion is not like that of a fluid. Our work builds on these methods, but restricts the motion of the corresponding features to approximately follow the velocity field of an underlying fluid simulation.

Researchers have developed a variety of methods to advect stylized imagery to follow optical flow in video or 3D character animation. For example, Agarwala et al. [2004] and Collomosse et al. [2005] construct keyframed and temporally coherent stylized animation from video by a rotoscoping method where drawn shapes track objects in the scene via optical flow. Bousseau et al. [2007] address the problem of coherent advection of watercolor stroke texture, also following optical flow for stylizing video. Recent work by Bénard et al. [2013] generates example-based stylizations of 3D computer-generated animation by extending the image analogies method of Hertzmann et al. [2001] to provide temporal coherence. Our approach is similar to these prior methods in that drawn shapes are advected to follow the motion of an underlying fluid simulation (rather than optical flow in video or 3D animation). The initialization step for our regenerative morph (Section 4) is similar to the bidirectional advection of Bousseau et al. Both the regenerative morphs on which we build and the animation method of Bénard et al. extend non-parametric methods for example-based synthesis to achieve temporal coherence in animation via optimiza-

tion, though different goal functions are expressed. Finally, several aspects of the work of Bénard et al. rely on the source 3D geometry, and it is not obvious how their approach would adapt to fluid flows.

While we address stylized rendering of *plausible* fluid motion, our method could be complemented by other research on stylizing the motion itself, for example the mid-level control for fluids by Barnat et al. [2011], motion field texture synthesis of Ma et al. [2009] and painted motion of Lockyer and Bartram [2012].

3 Synthesis by Regenerative Morphs

Our production workflow begins with a given fixed fluid simulation created for a particular shot. The main purpose of the simulation is to provide plausible motion for the construction of the in-between sequences. A secondary benefit is that it provides animators with a reasonable target for their keyframes, since these are known to have plausible fluid-like motion for the in-between sequences. Our experiments are based on Stam's stable fluids [1999] but our method is general to any simulator that provides plausible flow. In particular, with an art-directable simulator like those of Treuille et al. [2003] or Fattal and Lischinski [2004], the animator could draw keyframes first, and the simulation would optimize to hit those frames, implying a somewhat modified workflow.

The animator chooses frames of the simulation to draw manually. These drawn frames and the simulated velocity fields at each intermediate frame are supplied as input to the interpolation module, which outputs the in-between frames. If desired, the animator can then refine the output in specific places by modifying the set of keyframes until the resulting animation is satisfactory.

A successful keyframe interpolation will appear temporally coherent and reproduce the artist's style in each intermediate frame. These goals are difficult to satisfy simultaneously: for example, simply advecting the keyframe pixels backwards and forwards and blending, as in Bousseau et al. [2007], produces intermediates that are faithful to the motion but distort the keyframe style for many kinds of artwork. The remainder of this section describes recent developments in morphing and patch-based synthesis that address these concerns. Sections 4 and 5 describe modifications to these methods that improve the motion and temporal coherence.

Regenerative Morphs with Image Melding

To achieve both temporal coherence and fidelity to the keyframes in the interpolation, we leverage two recent patch-based methods – regenerative morphing, introduced by Shechtman et al. [2010], and its generalization via image melding, due to Darabi et al. [2012]. This section presents a brief overview of these methods for background, while subsequent sections offer several improvements suitable in the context of our application. We note that these patch-methods were developed based on observations about statistics in natural imagery, and while most of their applications to date have been photorealistic in nature; however, we find them to work well in our context and expect them to find broader use in non-photorealistic rendering in the future.

The basis for these methods is the bidirectional similarity (BDS) measure described by Simakov et al. [2008]:

$$d(S,T) = \frac{1}{N_S}\sum_{P\subset S}\min_{Q\subset T} D(P,Q) + \frac{1}{N_T}\sum_{Q\subset T}\min_{P\subset S} D(Q,P)$$

where S and T are the source and target images, N_I is the number of patches in image I, P and Q are patches in images S and T, respectively, and $D(\cdot,\cdot)$ is any patch distance measure. The first term captures completeness; it is minimized when the target contains as much information from the source as possible. The second term captures coherence; it is minimized when the target contains as few artifacts that do not appear in the source as possible.

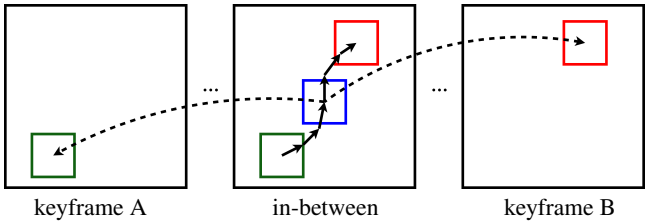

Figure 3: *Initialization: each patch (blue) of the in-between is advected forward and backward (solid arrows) to find correspondences (green and red) in the two nearest keyframes (dashed arrows). Two in-between images are constructed by the forward- and backward-advected patches, which are blended in proportion to the position of the in-between relative to the two keyframes.*

The regenerative morph is then computed as a minimization over all interpolated frames of the following objective function:

$$E(T_{1\ldots K}, S_1, S_2) = \sum_{k=1}^{K}((1-\alpha)d(T_k,S_1)+ \\ \alpha d(T_k,S_2) + \beta d(T_k,T_{k-1}) + \beta d(T_k,T_{k+1})) \quad (1)$$

where S_1 and S_2 are the keyframes, $T_{1\ldots K}$ are the interpolated frames, $T_0 = S_1$, $T_{K+1} = S_2$, $\alpha = \frac{k}{K+1}$, and β parameterizes the tradeoff between the similarity of the interpolated frames to the keyframes and the temporal coherence of the sequence.

The objective is minimized as follows. First, the optimization is initialized to a crude morph at a coarse scale, typically by cross-fading between keyframes or computing nearest neighbor offsets between keyframes and constructing each intermediate frame as an average of the patches at the linearly interpolated offset locations. Then, over multiple scales, the algorithm sweeps sequentially forwards and backwards over the intermediate frames. For each frame k, a search step computes nearest neighbor fields between the frame and its neighbors ($k-1$ and $k+1$) as well as the between frame k and the two nearest keyframes, using the Generalized PatchMatch algorithm [Barnes et al. 2010]. To ensure a coherent sense of motion, patch searches between adjacent frames are constrained to a small window around the location of the target patch. The frame is then reconstructed as a weighted combination of these nearest neighbor patches. Since each pixel in the reconstructed output frame overlaps many such nearest neighbor patches, they are combined using a *voting* scheme – the synthesized pixel is taken as a weighted combination of the associated pixels in the overlapping patches. Several iterations of the search and reconstruction steps are typically necessary for convergence.

This section has given a brief overview of the regenerative morphs of Shechtman et al. [2010] and their generalization by Darabi et al. [2012]. Our method is adapted from the implementation of Darabi et al., and except for the modifications described in the upcoming sections, the reader is referred to those papers for additional detail.

4 Motion Priors

Regenerative morphing as described in Section 3 produces compelling transitions between keyframes, but the motion can appear arbitrary because it is guided only by the nearest neighbor matching. In contrast, animated fluid should have fluid-like motion. We

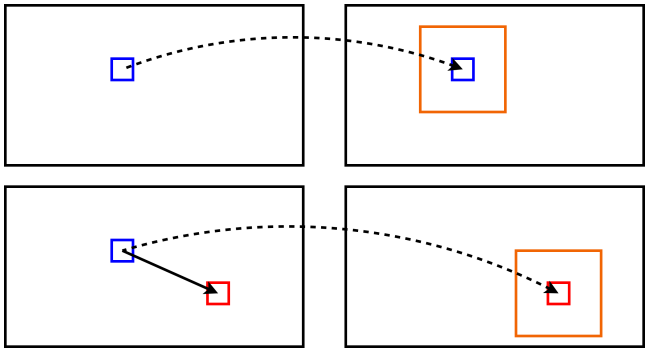

Figure 4: *Motion priors.* Top: *in a standard regenerative morph, temporal coherence is achieved by restricting patch searches between adjacent frames to a small window.* Bottom: *in our method, these constraint windows are centered around the advected location to ensure that the motion follows the flow.*

satisfy this constraint by modifying the algorithm in two ways, each of which causes the apparent motion to correspond to the underlying simulated flow.

Initialization. First, we initialize the morph such that the motion follows the simulated flow, in a fashion similiar to the method of Bousseau et al. [2007], and illustrated in Figure 3. In a given intermediate frame, each pixel is advected forward along the fluid motion to the subsequent keyframe and backward to the preceeding keyframe. Two images are constructed (one advected forward, and the other backward) by voting among overlapping patches, and they blended with weights based on the position in the sequence, as in the regenerative morph of Darabi et al. [2012].

Temporal Constraints. The second modification is to constrain the patch searches between adjacent frames according to the simulated flow. In particular, when computing $d(T_k, T_{k-1})$ and $d(T_k, T_{k+1})$, instead of constraining the search to a small fixed-size window around the location of the target patch, we constrain it to a small window around the target location after it is advected in the direction of the other image. This is illustrated in Figure 4. The blue boxes represent a patch in the current frame. In the regenerative morph of Durabi et al., the patch search is constrained to a small window (orange box) around this patch's location in the source image. In our method, the constraint window is centered on the *advected* position (red box). The degree to which the advected constraint window aligns with the standard constraint window depends on the fluid velocity at the target position. Finally, as will be described in Section 5, we modulate the width p of this (orange) constraint window based on local fluid velocity.

5 Continuous Multi-Keyframe Morphs

For some inputs, simply concatenating successive two-keyframe sequences produced by the above method produces acceptable results that interpolate the given keyframes. For complex flows and artistic styles, however, this approach can produce discontinuous "pulsing" artifacts near the keyframes. To address this problem, we generalize the regenerative morph to approximate over multi-keyframe inputs and parameterize the tradeoff between interpolation and continuity at the keyframes.

Modified Objective Function. One of the primary sources of discontinuous motion near the keyframes is misregistration of the rotoscoped input to the underlying flow. Because each keyframe is supplied independently, and because it is difficult for the artist to have a completely coherent sense of the motion between keyframes, a region of given texture and color may not match the texture and color of the region to which it is advected in an adjacent keyframe. Thus, to preserve feature correspondences and interpolate the keyframes, the morphs of Section 4 tend to advect patches in directions that do not match the desired flow.

We address this problem in two ways. First, we generate the multi-keyframe morph by optimizing a modified objective function:

$$E(T_{1\ldots M}, S_{1\ldots N}) = \sum_{m=1}^{M}\Big(\sum_{n=1}^{N}\hat{w}(m,n)d(T_m,S_n)+ \\ +\beta d(T_m,T_{m-1})+\beta d(T_m,T_{m+1})\Big) \tag{2}$$

where m ranges over the entire sequence of intermediate frames $T_1 \ldots T_M$ between the first and Nth keyframes and $\hat{w}(m,n)$ is the weight of keyframe n in the synthesis of frame m.

There are two differences between this objective function and (1). First, the optimization is performed over the entire multi-keyframe sequence. Thus, intermediate keyframes are no longer constrained to match the input exactly, but instead share the soft constraints imposed on all other synthesized frames; namely, that they are similar in the BDS sense to the input keyframes and that the motion between adjacent frames follow the given motion fields. By relaxing the interpolation constraint, we reduce the undesired motion required to transition from keyframe to keyframe while still respecting the overall style of the artist. The effect of approximating instead of interpolating is illustrated in Figure 7. The approximated keyframe is visually similar to the input keyframe, but with minor adjustments due to the temporal influence of neighboring frames.

Second, the source similarity weighting function has been generalized to include contributions from multiple keyframes. In our experiments, we found that setting $\hat{w}$ to be constant over a radius of 2-3 neighboring keyframes or using a smooth step function produced more continuous results than a simple alpha-blend of the two nearest keyframes.

Velocity Modulation. The second way in which we address discontinuities at the keyframes is based on the observation that motion that does not match the given fields is most noticeable in regions with little underlying flow. In regions where the underlying flow is large, the extraneous motion required to preserve feature correspondences between keyframes is comparatively small. We therefore modulate the size p of the prior motion constraint (the orange box in Figure 4) at each patch location (i,j) by the magnitude of the motion field at that location:

$$p_{i,j} = (1-\gamma)p + \gamma f(|\mathbf{v}_{i,j}|)p \tag{3}$$

where $f : \mathbb{R}_{\geq 0} \to [0,1]$ is some monotonically increasing function and γ is a parameter between 0 and 1 that controls the degree to which the modulation influences the prior size. For flows with very uneven distributions of velocity magnitudes, it may be desirable to modulate the prior size by histogram matching, for example, but for our inputs we found that

$$f(|\mathbf{v}_{i,j}|) = \frac{|\mathbf{v}_{i,j}|}{\max\limits_{i,j}|\mathbf{v}_{i,j}|}$$

worked well. When γ is close to 0, this modulation has little effect, which produces the discontinuous motion required to preserve

keyframe 50 in-between 53 frame 57

Figure 5: *A keyframe and two synthesized in-betweens for a simulation of opposing jets drawn in a marker style.*

keyframe correspondences. When γ is close to 1, the motion is more continuous, but at the cost of additional fading, blurring, and approximation where the keyframes are misregistered with the underlying flow.

6 Implementation

Parallelization. To produce an animation by concatenating sequences of morphs between successive pairs of keyframes, as described in Section 4, it is trivially possible to compute each morph independently and in parallel. In this case, due to the sequential sweeps across the frames at each scale, the time to completion depends on the number of frames between the most distant pair of keyframes. This sequential dependence is illustrated in Figure 6-left: during forward sweep l, frame k depends on the synthesized frame $k-1$ from the same sweep l (the direction of dependence is reversed in a backward sweep).

The continuous multi-keyframe morph developed in Section 5 results in an optimization that is global over the entire animated sequence, for which these sequential sweeps would be prohibitively slow to compute. We therefore parallelize the implementation by breaking the dependence between adjacent frames in the same sweep, as illustrated in Figure 6-right. That is, in our parallel implementation, frame k in sweep l depends only on frames from sweep $l-1$ (and the keyframes). Thus our implementation constructs all frames in a given temporal iteration independently and simultaneously, on a compute cluster.

In principle, this parallelization reduces the rate at which information can propagate between frames, which might lead to slower convergence. However, in our experiments, we found little noticeable difference between the output of the two implementations, as shown in Figure 8. Moreover, we find that the energy (2) after optimization is consistently equivalent, to within four significant digits.

Two-Pass Morph. The methods discussed so far each contribute to guiding the regenerative morph to roughly match the underlying fluid simulation and to improving continuity in multi-keyframe

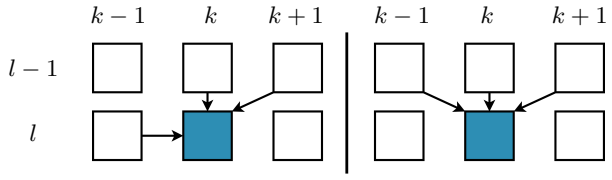

Figure 6: *Parallelization.* Left: *in the original regenerative morphing algorithm, target frames (blue) in iteration l are reconstructed by sweeping sequentially over each frame k, with arrows indicating dependences.* Right: *in our parallel implementation, frames in iteration l depend only on frames from the previous iteration $l-1$.*

keyframe 60 approximated frame 60 difference

Figure 7: *The approximating multi-keyframe morph permits the keyframe to change to better match the flow without sacrificing fidelity to the artist's input. The difference image is the L_1 distance between the two RGB images, where white is 0 and black is the maximum difference between the two images.*

parallel 63 sequential 63 difference

Figure 8: *These frames demonstrate that parallelizing the morph does not noticeably degrade the quality of the results. Small differences between the frames are noticeable only upon close inspection. Difference image as in Figure 7.*

sequences. The various combinations of these methods represent a large space of possible morphs. In our experiments, we found that the quality of results depends not only on the individual parameter settings, but also on the combination of these settings. For example, combining the approximating multi-keyframe morph with velocity modulation improves the continuity of our results, but applied independently, neither method results in comparable improvements. We found the following two-pass morph to work well across a variety of inputs.

First, we perform an initialization pass which constructs an approximating multi-keyframe morph using motion priors, as described above. The keyframe weighting function in (2) is the same alpha-blend used in (1), and the motion prior size p is not modulated by velocity. These parameters are chosen to maximize the fidelity of the reconstructed frames to the keyframes. The purpose of this pass is to produce a set of high-quality frames, which may lack continuity, for use as input to a second "smoothing" pass. In the second pass, we treat each output frame from the first pass as a keyframe and perform the approximating multi-keyframe morph. We use a constant weighting function over a radius of 2 or 3 keyframes, and modulate the prior size by velocity, as in (3), with $\gamma = 1$. The purpose of this pass is to ameliorate the discontinuity artifacts that can appear near keyframes after first pass, without sacrificing too much sharpness in the intermediate frames.

7 Results and Discussion

The video accompanying this paper presents three different simulations animated in five distinct visual styles and media, demonstrating the broad range of effects that can be reproduced. Figure 1 shows a simulation of a jet of ink, animated via eight keyframes drawn in both oil pastel and ink wash. Figures 2 and 9 show the same simulation rendered in a pen and ink style and inkwash style.

Figure 9: *Same jet from Figure 1, drawn in an inkwash style.*

The pen and ink style is both substantially more difficult for the artist to draw and more difficult to animate, due to the many small high-contrast features. Figures 6-8 show frames rotoscoped from a simulation of two opposing jets squirting liquid towards each other and producing a complex collision. This animation is based on eight keyframes drawn in a marker style exhibiting distinct lines that are well-replicated in the in-betweens. Finally, Figure 10 shows another two-jet simulation sketched in a colored pencil style.

Each of the examples described above was based on a sequence of 71 or 81 simulation frames, where the artist drew a keyframe for every tenth frame and the system produced nine in-betweens for every consecutive pair of keyframes. At this ratio the artist's workload was reduced by at least a factor of ten over drawing every frame of the animation – a clear savings in mechanical labor alone. Moreover, to achieve the same level of temporal coherence entirely by hand might impose a further burden, as these drawings were made *without* particular regard for the previous or next keyframe. It is natural to wonder whether the artist could have drawn even fewer keyframes – say every twentieth frame. We performed some preliminary experiments and found that the 20-frame spans were of visibly inferior quality compared with the 10-frame sequences shown in our paper and video. These experiments were by no means exhaustive in the large space of visual styles and optimization parameters, so it is plausible that longer sequences might be made to work. Regardless, we found the 10:1 ratio a nice compromise in terms of visual fidelity and labor savings.

For some styles, we experimented with a range of patch sizes (5,7,9,11) and motion prior sizes (1,3,5,7). We found the method to be robust across this range, producing different but acceptable results in most cases. The patch size influences the performance of the optimization (doubling patch size roughly doubles computation time) and also sets the approximate scale of features that are matched via the bidirectional similarity metric. The motion prior trades off between fidelity to the fluid motion and the fidelity to the visual style of the keyframes. All the fluid results shown in this paper and the accompanying video use patch size 5 and prior size 5 at image resolution 400×400. For this patch size, at square frames sizes of dimension 200, 400, 600 or 800 pixels, it takes roughly 10, 30, 100 or 240 minutes, respectively, on a heterogeneous cluster in which a 2.8Ghz 4-core AMD processor is typical. These timings match a cubic dependence in the linear dimension of the image. Note that since our algorithm is parallelized by frame, these times are roughly constant in the number of frames M in the full sequence, as long as M is less than the cluster size.

We found that the optimization energy in equation (1) was consistently lower using our flow-guided morphs than the regenerative morph of Darabi et al. [2012]. We believe that this is because in this huge nonlinear optimization, the motion field offers a good prior for plausible motion between the rotoscoped keyframes.

While we are generally pleased with the quality of the computer-generated morphs presented in this paper and the accompanying video, we would like to point out some visual artifacts. The in-betweens sometimes fail to reproduce fine-scale, high-contrast aspects of textures, like the fine stippling of the the pen-and-ink style in Figure 2. These differences are often hard to notice in animation, but can be more clearly seen in the stills. This problem might be addressed by running the algorithm at higher resolution with larger patch sizes (at the cost of increased computation), or by improving the reconstruction step of the morph by using the mean shift algorithm of Comaniciu and Meer [2002] instead of a weighted average, as proposed in [Wexler et al. 2007]. Also, when the morph fails to find good correspondences it often resorts to fading in (or out) some feature that a person might have been able to treat more gracefully by hand. This might be addressed by more sophisticated optimizations that could find even better patch correspondences. Finally, in some animations the regenerative morph produces a faint, diffuse halo around high-contrast areas. We believe it is a result of the gradient weighting in the image melding approach, and could probably be removed with different parameter settings (or more simply in post-production).

8 Conclusions and Future Work

This paper demonstrates that it is possible to create coherent, stylized animations of fluids, using a workflow wherein the style of automatic in-betweens is derived from hand-drawn keyframes and their motion is derived from a simulation. The animations shown in our results would have been arduous and time-consuming to produce by hand, and perhaps even impossible to do so with this degree of temporal coherence. We show a broad range of styles several of which we consider to be challenging to animate.

Our approach is artist-friendly in that it relies on a familiar keyframe-based workflow. The interpolating morph in Section 4 gives the artist complete control of the look at the keyframes. But we also find that the approximating scheme of Section 5 offers synthesized frames corresponding to the keyframes that are sufficiently close to offer substantial control of the look near the keyframes. Finally, while the simulation itself provides a helpful guide for the artist, we find that the hand-drawn frames can deviate quite noticeably from the underlying simulation and the resulting animations still look fluid-like – supporting flexibility of composition.

As mentioned in Section 1, there is a tension between coherence and fidelity to the medium and simulation. We have cast the problem as a nonlinear optimization in which we search for a local minimum via heuristics. Clearly we want as much of these desirable qualities as possible, but we do not know what the fundamental limit is (supposing a hypothetical brilliant animator or hypothetical algorithm could construct the ideal sequence that still respects this tradeoff). Therefore, we cannot know how far along

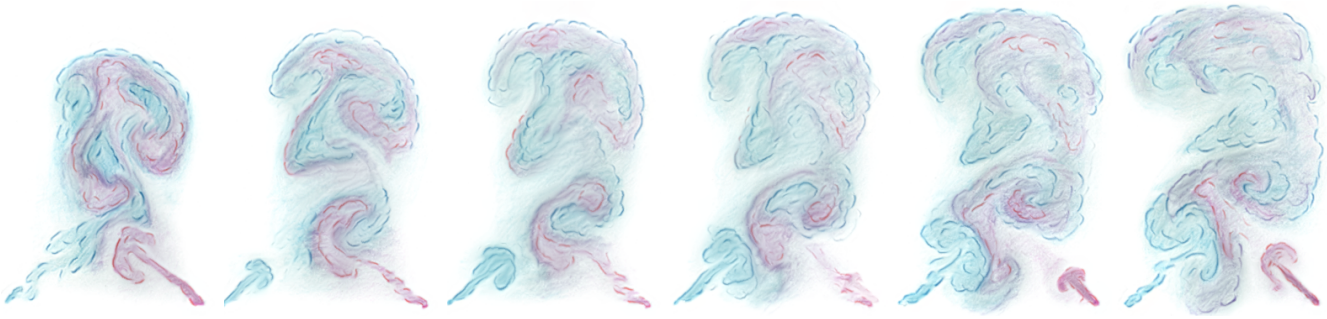

Figure 10: *A two-jet simulation sketched in colored pencil.*

this path we have gone. Also as mentioned in Section 7, it remains to be seen the maximum length of an in-between morph that can be reasonably obtained by these methods. This is likely to depend on the visual style as well as the context of the animation.

We have used simulations based on Stam's stable fluids [1999], but we believe our method would nicely complement simulations that focus on artistic control like that of Treuille et al. [2003] and Fattal and Lischinski [2004], which would necessitate a modified workflow. Moreover, while we have focused on 2D fluid simulations, we believe the broad ideas in this work could be adapted to other kinds of simulations like 3D fluid, hair and cloth. A major challenge would be to address depth and occlusion changes in this context.

Finally, we believe this to be the first use of regenerative morphs in non-photorealistic rendering, but that this general technique is likely to be broadly applicable in areas that involve stylization and artistic control. We believe the method may also extend to 3D keyframe animation.

References

AGARWALA, A., HERTZMANN, A., SALESIN, D. H., AND SEITZ, S. M. 2004. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph. 23*, 3, 584–591.

BARNAT, A., LI, Z., MCCANN, J., AND POLLARD, N. S. 2011. Mid-level smoke control for 2d animation. In *Proceedings of Graphics Interface 2011*, GI '11, 25–32.

BARNES, C., SHECHTMAN, E., GOLDMAN, D. B., AND FINKELSTEIN, A. 2010. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*.

BEIER, T., AND NEELY, S. 1992. Feature-based image metamorphosis. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, 35–42.

BÉNARD, P., COLE, F., KASS, M., MORDATCH, I., HEGARTY, J., SENN, M. S., FLEISCHER, K., PESARE, D., AND BREEDEN, K. 2013. Stylizing Animation By Example. *ACM Transactions on Graphics 32*, 4 (July).

BOUSSEAU, A., NEYRET, F., THOLLOT, J., AND SALESIN, D. 2007. Video watercolorization using bidirectional texture advection. *ACM Trans. Graph. 26*, 3 (July).

COLLOMOSSE, J. P., ROWNTREE, D., AND HALL, P. M. 2005. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Transactions on Visualization and Computer Graphics 11*, 5 (Sept.), 540–549.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (May), 603–619.

CORRÊA, W. T., JENSEN, R. J., THAYER, C. E., AND FINKELSTEIN, A. 1998. Texture mapping for cel animation. In *Proceedings of SIGGRAPH 98*, 435–446.

DARABI, S., SHECHTMAN, E., BARNES, C., GOLDMAN, D. B., AND SEN, P. 2012. Image Melding: Combining Inconsistent Images using Patch-based Synthesis. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012) 31*, 4.

DI FIORE, F., CLAES, J., AND VAN REETH, F. 2004. A framework for user control on stylized animation of gaseous phenomena.

EDEN, A., BARGTEIL, A., GOKTEKIN, T., EISINGER, S., AND O'BRIEN, J. 2007. A method for cartoon-style rendering of liquid animations. In *Proceedings of Graphics Interface 2007*, ACM, 51–55.

FATTAL, R., AND LISCHINSKI, D. 2004. Target-driven smoke animation. *ACM Trans. Graph. 23*, 3 (Aug.), 441–448.

GILLAND, J. 2009. *Elemental Magic, Volume I: The Art of Special Effects Animation*. Focal Press.

HE, H., AND XU, D. 2005. Real-time cartoon animation of smoke. *Computer Animation and Virtual Worlds 16*, 3-4, 441–449.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, 327–340.

JAIN, E., SHEIKH, Y., MAHLER, M., AND HODGINS, J. 2012. Three-dimensional proxies for hand-drawn characters. *ACM Trans. Graph. 31*, 1 (Feb.), 8:1–8:16.

KWATRA, V., ADALSTEINSSON, D., KIM, T., KWATRA, N., CARLSON, M., AND LIN, M. C. 2007. Texturing fluids. *Visualization and Computer Graphics, IEEE Transactions on 13*, 5, 939–952.

LOCKYER, M., AND BARTRAM, L. 2012. The amotion toolkit: painting with affective motion textures. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, CAe '12, 35–43.

MA, C., WEI, L.-Y., GUO, B., AND ZHOU, K. 2009. Motion field texture synthesis. *ACM Trans. Graph. 28*, 5 (Dec.), 110:1–110:8.

MAHAJAN, D., HUANG, F.-C., MATUSIK, W., RAMAMOORTHI, R., AND BELHUMEUR, P. 2009. Moving gradients: A path-

based method for plausible image interpolation. *ACM Trans. Graph. 28*, 3 (July), 42:1–42:11.

MCGUIRE, M., AND FEIN, A. 2006. Real-time rendering of cartoon smoke and clouds. In *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, ACM, 21–26.

PETROVIĆ, L., FUJITO, B., WILLIAMS, L., AND FINKELSTEIN, A. 2000. Shadows for cel animation. In *Proceedings of ACM SIGGRAPH 2000*, 511–516.

SELLE, A., MOHR, A., AND CHENNEY, S. 2004. Cartoon rendering of smoke animations. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, ACM, 57–60.

SHECHTMAN, E., RAV-ACHA, A., IRANI, M., AND SEITZ, S. 2010. Regenerative morphing. In *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*.

SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. 2008. Summarizing visual data using bidirectional similarity. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, IEEE Computer Society.

SOLOMON, C. 1994. *Enchanted Drawings: The History of Animation*. Random House.

STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 121–128.

THOMAS, F., AND JOHNSTON, O. 1987. *Disney Animation: The Illusion of Life*. Abbeville Press.

TREUILLE, A., MCNAMARA, A., POPOVIĆ, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *ACM Trans. Graph. 22*, 3 (July), 716–723.

WEXLER, Y., SHECHTMAN, E., AND IRANI, M. 2007. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 3 (Mar.), 463–476.

WOLBERG, G. 1998. Image morphing: a survey. *The visual computer 14*, 8, 360–372.

YOU, M., PARK, J., CHOI, B., AND NOH, J. 2009. Cartoon animation style rendering of water. *Advances in Visual Computing*, 67–78.

YU, J., JIANG, X., CHEN, H., AND YAO, C. 2007. Real-time cartoon water animation. *Computer Animation and Virtual Worlds 18*, 4-5, 405–414.

ZHU, B., IWATA, M., HARAGUCHI, R., ASHIHARA, T., UMETANI, N., IGARASHI, T., AND NAKAZAWA, K. 2011. Sketch-based dynamic illustration of fluid systems. *ACM Trans. Graph. 30*, 6, 134.