Small-Scale Video-Motion Editing using Confidence-Aware Flow-Based Deformation

Liang-Kang Huang
National Taiwan University

Ken-Yi Lee
National Taiwan University

Yung-Yu Chuang
National Taiwan University

Abstract

Manipulating motion characteristics in videos is one of the most interesting operations in video editing. By changing the motion presented in the video, users can either emphasize different contents or tell a different story with the synthesized video. While being a powerful editing operation, the task is undeniably challenging. To make the synthesized video seamless in appearance both spatially and temporally, tedious and time-consuming manual editing and tweaking are often required with existing editing tools.

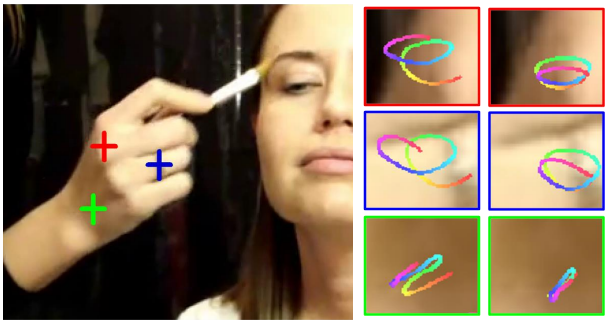
This paper proposes a framework to assist users on editing small-scale video motions with ease. By limiting the editing subject to small-scale video-motions, carefully utilizing the results from motion estimation and solving a space-time optimization to warp the video frames, the proposed framework is capable of producing seamless videos with certain motion characteristics with only simple user inputs. Extensive experiments are conducted to show that our method can produce visually pleasing results on a wide range of videos with different motion characteristics.

1. Introduction

Motion is the major content in videos. Besides the pixel values in each frame which depict the appearance of the scene, correlation between pixels across frames —namely video-motion—allows video to tell more of a story than static images. Manipulating video-motion is thus considered to be one of the critical task of video editing.

However, editing video-motion is a difficult task that often poses the following challenges. First, naïve ways to modify video-motion can introduce serious artifacts in the synthesized video, mainly spatial and temporal discontinuities. The root cause is that every pixel has certain relations with pixels in its spatial and temporal neighborhood. Directly modifying the motion without considering these relations thus results in holes and seams within and across frames. This problem becomes more severe when the extent of modification is larger. Second, perfect motion information is hard to extract in several scenes. Before the motion in a video can be modified, it is critical that the original motion in the video being accurately analyzed. Unfortunately, despite great advancement in the field of motion estimation and optical flow, the task remains challenging in several cases. Thus, methods for editing video-motions that rely on perfect motion estimation results are likely to fail in practice. Finally, user input for editing video-motion can be inaccurate. The task of editing video-motion requires

(a) Motion localization and the spatial discontinuities



(b) Motion concatenation and the temporal discontinuities

Figure 1. An example of using our framework to perform (a) Motion Localization: localize the motion to occur only on the hand. We show the spatial discontinuity at the mask boundary produced by the naïve masking method and the seamless results produced by our framework. (b) Motion Concatenation: concatenating two sets of frames (in this case, the ending and starting frames of this video sequence). We plot the motion trajectory of the three selected points in the original sequence and the synthesized sequence produced by our framework. Note the three seamless trajectory that formed three closed loops.

several low level information to be processed and aggregated. Without professional knowledge and training, it is hard for users to provide inputs that are ideal for those low level tasks. Thus, methods that aims to minimize the user efforts should consider the imperfections in the user inputs and deal with them accordingly. With the presence of these challenges, existing video-motion editing tools are fairly limited in the type of video they can handle and the type of editing operations they could provide.

In this paper, we draw insights from the above considerations, and design a framework that aims at assisting users to modify video-motion with ease. By limiting its usage to small-scale video-motion editing (*i.e*. the extent of modification on the motion will not be too large), the framework is capable of offering two motion-editing operations: motion localization and motion concatenation. Motion localization is an editing operation that allows users to make the motion only happen within part of the subject while the rest keeps still. More specifically, users can edit the video-motion to happen in only part of the scene by simply drawing a mask on a particular frame. Motion concatenation is another editing operation that allows users to make seamless transition between frames that are similar in their appearance and motion. Through this operation, users can either truncate redundant content of the video while keeping it visually pleasing or even create a seamless-looping video that presents an endless repeating motion.

Our framework relies on the result extracted by existing optical flow implementations, but in the mean while accounting for the possible errors in their results. Specifically, instead of assuming the motion estimation result is perfect, our framework computes and references the confidence value of the estimated motion on each pixel. Finally, to account for the inaccuracy of the user input and further reduce the users' burden, we proposed methods to refine the user inputs when performing the two editing operations.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed framework in detail. Section 4 evaluates our framework on videos with different types of motion. Finally, Section 5 concludes the paper.

2. Related work

Previous work related to video-motion editing differentiate with each other by targeting on different types of videos or different types of video-motion editing operations. Among those literatures, the closest to ours are the works related to cinemagraph creation. A cinemagraph is a still photograph that contains within itself a repeating motion in part of the subject, in contrast with the stillness of the rest of the image(for example, a girl keeps still in expressions and postures with her hair blowing in the breeze). Though lacking a formal definition, cinemagraph is often considered to be a video sequence with its motion being localized and repeating. From this point of view, creating cinemagraph is a task of localizing video motion and concatenating the ending and the starting frames of the video, which is similar to the targeted task in this paper.

Several software and tools aims at creating cinemagraphs from video are available on the web, such as [2] and [1]. To localize motion, they adopt the basic approach we referred to as the naıve masking method. The naıve masking method apply the user drawn mask on every frame in the video, so that only the content in the mask will change over-time, while regions outside the mask being a static image. However, this simple strategy often results in severe artifacts in the output video in many cases. Besides these commercial software, [13] and [6] has extended the naıve masking method with the aid of computer vision technology such as video stabilization and loop selection. However, the problem of spatial and temporal visual discontinuities are not fully addressed and handled in their paper.

Another work closely related to ours is [3]. The method proposed in their paper facilitates not only the creation of cinemagraphs but also other applications such as visualizing or emphasizing motion in video. Though they also addressed the spatial and temporal discontinuities likely to appear during video-motion editing, they design different approaches for these two problems. Additionally, the strategy they use for resolving temporal seams is essentially selecting transitions in video frames. In contrast, in this paper we design a unified framework for resolving the spatial and temporal discontinuity. Instead of selecting good transitions in the video frame, we develop ways to warp each frame to resolve the temporal seams. This makes the cases we can handle different from their work.

Despite these works, there are other works target on synthesizing new motion sequence such as [10], [16] and [14], which aim at modifying the scale of video-motion. And a series of work for creating video textures from a video [12] [8], a set of images [9], a single image [5], and images on the web [17]. Although these works target on video-motion editing, they focus on other aspects of this task and look into problems different from this paper.

3. Video-motion editing framework

In this section, we describe the proposed framework in detail. The section is organized as follows. Section 3.1 explains how to estimate motions and evaluate the confidences of the estimated motions. Section 3.2 proposes a framework for general warping-based video processing. We use it in Section 3.3 and 3.4 to perform two video-motion editing operation: motion localization and motion concatenation. Finally, Section 3.5 presents the method we use to refine the user input.

3.1. Motion estimation and confidence

Before modifying the motion characteristics of the video, the original motion of the video should be extracted. Given two images I_{t_1} and I_{t_2}, the dense motion field $\mathbf{u}^{t_1 \to t_2}$ can be estimated by optical flow algorithms [4] such that:

$$\mathbf{x}' = \mathbf{x} + \mathbf{u}_{\mathbf{x}}^{t_1 \to t_2}, \tag{1}$$

where $\mathbf{x}$ is the position of a pixel in I_{t_1} and $\mathbf{x}'$ is the position of its corresponding pixel in I_{t_2}. To represent the correspondences between different frames more concisely, we define a mapping function: $M_{t_1}^{t_2}(i) = j$ if $\mathbf{x}_j^{t_2} = \mathbf{x}_i^{t_1} + u_{\mathbf{x}_1}^{t_1 \to t_2}$. That is, $M_{t_1}^{t_2}(i) = j$ if the pixel $\mathbf{x}_j$ in I_{t_2} corresponds to the pixel $\mathbf{x}_i$ in I_{t_1}.

In addition, instead of assuming the extracted motion to be perfect, our framework account for the possible error by computing a confidence value for each pixel. We use the bidirectional consistency measure with the intuition that if the motion is correctly estimated, the forward and backward flows should compensate each other as much as possible. That is, moving a pixel $\mathbf{x}$ with its forward motion $u_{\mathbf{x}}^{t_1 \to t_2}$ to its corresponding pixel $\mathbf{x}'$ in the next frame, followed by the corresponding pixel's backward motion $u_{\mathbf{x}'}^{t_2 \to t_1}$, should take us back to pixel $\mathbf{x}$. Specifically, the motion confidence $C^{t_1 \to t_2}$ for the pixel $\mathbf{x}$ is measured as

$$C^{t_1 \to t_2}(\mathbf{x}) = \max(1 - |(u_{\mathbf{x}}^{t_1 \to t_2} + u_{\mathbf{x}'}^{t_2 \to t_1})/K|^2, 0) \tag{2}$$

where K is a fixed threshold, and $0 \le C^{t_1 \to t_2}(\mathbf{x}) \le 1$.

Note that both $\mathbf{u}_{\mathbf{x}}^{t_1 \to t_2}$ and $\mathbf{x}'$ could have sub-pixel accuracy. Whenever we need to extract the sub-pixel value of a function f defined on integral coordinates, the bilinear interpolation strategy as below is used throughout the paper

$$\begin{aligned} f(\mathbf{x}') \equiv (1-\alpha)(1-\beta) f(\lfloor \mathbf{x}' \rfloor) + \alpha(1-\beta) f(\lfloor \mathbf{x}' \rfloor + (1,0)) \\ + (1-\alpha)\beta f(\lfloor \mathbf{x}' \rfloor + (0,1)) + \alpha\beta f(\lfloor \mathbf{x}' \rfloor + (1,1)), \end{aligned} \tag{3}$$

where $\mathbf{x}' - \lfloor \mathbf{x}' \rfloor = (\alpha, \beta)$

3.2. Warping-based video processing

Recently, warping-based approaches have been used in many different image/video processing problems such as video stabilization [11], retargetting [15] and disparity mapping [7]. Most warping-based video processing approaches represent each frame as a quad mesh and find the new vertex positions of all quad meshes to satisfy sparse position constraints while avoiding spatial and temporal distortion. The output video is then synthesized according to the deformed quad meshes.

Here, we address a framework for general warping-based video processing and utilize it for video-motion editing. In the framework, video frames are represented by a set of quad meshes, $\mathcal{V} = \{\mathcal{V}_t | 1 \le t \le T, t \in \mathbb{N}\}$, where $\mathcal{V}_t$ is the quad mesh for $\mathbf{I}_t$. Although it is possible to use larger quad sizes for efficiency, the quad size is set to 1×1 for all the results in this paper, *i.e.* every pixel is taken as a vertex in the quad mesh. The deformed quad meshes $\hat{\mathcal{V}}$ is created by adjusting the vertex positions of $\mathcal{V}$ through minimizing the following energy function $\mathbf{E}$ subject to some additional constraints:

$$\mathbf{E}(\hat{\mathcal{V}}) = \lambda_S \mathbf{E}_S(\hat{\mathcal{V}}) + \lambda_T \mathbf{E}_T(\hat{\mathcal{V}}), \tag{4}$$

where $\mathbf{E}_S$ and $\mathbf{E}_T$ measures the spatial and temporal distortion respectively; λ_S and λ_T are weights balancing the spatial and temporal distortion, whose value depend on the problem to be solved.

In later sections, we will see the two motion-editing operation can be performed with this framework, by imposing different constraints and setting different values for λ_S and λ_T. Before that, we introduce how the spatial and temporal distortion are measured in our framework.

3.2.1 Spatial distortion

The term $\mathbf{E}_S$ measures the extent that the spatial relation, which in our case being the position and the shape of the quads, are distorted by the deformation.

$$\mathbf{E}_S(\hat{\mathcal{V}}) = \sum_{t=1}^{T} \sum_{\mathbf{x}_i^t \in \mathcal{V}_t} \sum_{\mathbf{x}_j^t \in \mathcal{N}(\mathbf{x}_i^t)} w_S(\mathbf{x}_i^t, \mathbf{x}_j^t) |(\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_j^t) - (\mathbf{x}_i^t - \mathbf{x}_j^t)|^2 \tag{5}$$

where $\mathcal{N}(\mathbf{x}_i^t)$ is the set of four neighboring vertices of $\mathbf{x}_i^t$ in $\mathcal{V}_t$, $\hat{\mathbf{x}}_i^t \in \hat{\mathcal{V}}_t$ is the deformed vertex position of $\mathbf{x}_i^t$, and w_S is a weight to allow neighboring vertices to have more spatial distortion if they belong to different objects. In Equation 5, for each edge in the original quad mesh formed by two neighboring vertices, $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$, we prefer that the corresponding edge $(\hat{\mathbf{x}}_i^t, \hat{\mathbf{x}}_j^t)$ in the deformed mesh maintains the same orientation and size by measuring the deviations. As for the weight w_S, by assuming the vertices on the same object undergo similar motions, we define w_S based on the motion similarity of the vertices:

$$w_S(\mathbf{x}_i^t, \mathbf{x}_j^t) = \exp^{-(\mathbf{u}_{\mathbf{x}_i^t}^{t \to t+1} - \mathbf{u}_{\mathbf{x}_j^t}^{t \to t+1})^2 / \sigma_S^2} \tag{6}$$

where σ_S is a given constant.

3.2.2 Temporal distortion

Similar to $\mathbf{E}_S$, the term $\mathbf{E}_T$ measures the extent that the temporal relation, which in our case being the motion between frames, are distorted by the deformation. Specifically, we calculate the difference between the motion in the original sequence and the deformed sequence.

$$\begin{aligned} \mathbf{E}_T(\hat{\mathcal{V}}) = \sum_{t=1}^{T-1} \Big(\sum_{\mathbf{x}_i^t \in \mathcal{V}_t} w_T(\mathbf{x}_i^t, \mathbf{x}_{i'}^{t+1}) |(\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_{i'}^{t+1}) - (\mathbf{x}_i^t - \mathbf{x}_{i'}^{t+1})|^2 + \\ \sum_{\mathbf{x}_j^{t+1} \in \mathcal{V}_{t+1}} w_T(\mathbf{x}_j^{t+1}, \mathbf{x}_{j'}^t) |(\hat{\mathbf{x}}_j^{t+1} - \hat{\mathbf{x}}_{j'}^t) - (\mathbf{x}_j^{t+1} - \mathbf{x}_{j'}^t)|^2 \Big) \end{aligned}$$

where $i' = M_t^{t+1}(i)$ is the vertex corresponding to $\mathbf{x}_i^t$ in $\mathbf{I}_{t+1}$; $j' = M_{t+1}^t(j)$ is the vertex corresponding to $\mathbf{x}_j^{t+1}$ in $\mathbf{I}_t$; and w_T is the weight that accounts for the reliability of the estimated motion. Equation 7 accumulates the difference between the original motion $\mathbf{x}_i^t - \mathbf{x}_{i'}^{t+1}$ and the modified motion $\hat{\mathbf{x}}_i^t - \hat{\mathbf{x}}_{i'}^{t+1}$ for each

vertex. We do so for both forward (the $\mathbf{x}_i^t \in \mathcal{V}_t$ part) and backward ($\mathbf{x}_j^{t+1} \in \mathcal{V}_{t+1}$) motions. w_T is defined by the confidence of motion estimation as:

$$w_T(\mathbf{x}_p^{t_1}, \mathbf{x}_{p'}^{t_2}) = C^{t_1 \to t_2}(\mathbf{x}_p^{t_1}). \tag{7}$$

3.3. Motion localization

Motion localization is the editing operation that allows users to make the motion only happen within part of the subject while the rest keeps still. To perform motion localization, user is required to draw a mask Ω on a single frame, specifying the part of the subject in the scene where motion should be preserved.

The major challenge of motion localization is that, if the pixel on the boundaries of the masks are not still in the original video(*e.g.* the mask is drawn on part of a moving subject, or the original video contains camera motion), applying the same mask Ω to every frame could result in serious spatial discontinuities through out the video. To resolve this problem, we perform motion localization by utilizing the framework proposed in Section 3.2 with additional spatial constraints that pixels on the boundary should stay still throughout the video:

$$\underset{\hat{\mathcal{V}}}{\text{minimize}} \quad \mathbf{E}(\hat{\mathcal{V}}) \tag{8}$$

$$\text{subject to} \quad \hat{\mathbf{x}}_{i'}^t = \mathbf{x}_i^1, \forall i \in \{k | \mathbf{x}_k^1 \in \partial\Omega\}, 1 \le t \le T \tag{9}$$

where $\partial\Omega$ is the boundary of Ω and $i' = M_1^t(i)$ is the vertex index of the correspondence of $\mathbf{x}_i^1$ at frame $\mathbf{I}_t$. The boundary alignment constraint (Equation 9) makes all correspondences of the vertex $\mathbf{x}_i^1 \in \partial\Omega$ in any other frame to have the same position as $\mathbf{x}_i^1$, thus aligning contours.

Note that in motion localization, we should allow temporal deformation because localizing motion indeed changes the original motions of points along the boundary of the mask. In this case, λ_T is set to be 0.1 and λ_S is set to be 0.9. Larger λ_T preserves the original motions more but makes the output video contains spatial seams.

3.4. Motion concatenation

The motion concatenation operation aims at letting users to make seamless transitions between two frames I_m and I_n. Though we require the two frames to be similar in appearance and motion, noticeable transitions, namely the temporal discontinuities, could still be perceived if we do not adjust each frame of the video. If we simply warp the two frames to be identical with each other, the original temporal discontinuities would appear in other frames. To smoothly eliminate the temporal discontinuity, we utilize the general warping-based video processing framework to adjust every frame in the video, by solving the optimization with an additional temporal constraint to force I_m and I_n resembling each other:

$$\underset{\hat{\mathcal{V}}}{\text{minimize}} \quad \mathbf{E}(\hat{\mathcal{V}}) \tag{10}$$

$$\hat{\mathbf{x}}_{j'}^m = \hat{\mathbf{x}}_j^n, \forall j \in \{k | \mathbf{x}_k^m \in \Omega\} \tag{11}$$

where $j' = M_n^m(j)$ is the vertex index of the correspondence of $\mathbf{x}_j^n$ in I_m. By solving the above optimization, the visual difference of the I_m and the I_n will be distributed across different frames. Thus, the induced content distortion is rather small, but overall they compensate the rather large difference between I_m and I_n.

In motion concatenation, we want to better preserve the object motions while eliminating the temporal discontinuity. In this case, λ_T is set to be 0.5 and λ_S is set to be 0.5. $\mathbf{E}_S$ plays as a regularization term here to prevent the situation that w_T is zero when the estimated motion is unreliable.

3.5. User input refinement

3.5.1 Optimal mask Ω

In motion localization, the user will draw a region-of-interest $\mathcal{R}$ to specify where motion should be preserved on frame f. However, the boundary of $\mathcal{R}$ may be hard to be align across frames if the adopted algorithm fails the estimate the motion of pixels along the boundary. Thus, we would like to find an optimal mask Ω to prevent unreliable boundary alignment. First, we dilate $\mathcal{R}$ by several pixels to be $\mathbf{O}$, and then find an optimal mask Ω where $\mathcal{R} \subseteq \Omega \subseteq \mathbf{O}$ by minimizing the following energy function:

$$\mathbf{E}_{SB}(\partial\Omega | T) = \sum_{\mathbf{x}_i^f \in \partial\Omega} (1 - \min\{C^{f \to t}(\mathbf{x}_i^f) | 1 \le t \le T\}). \tag{12}$$

$\mathbf{E}_{SB}$ measures the unreliability of the estimated motions between frame f and every other frames along the boundary of Ω, where the reliability of a pixel is defined by the minimum confidence between the pixel in frame f and its correspondence in any other frame. We use the minimum since any unreliable estimated motion could result in artifacts on the boundary. The best $\partial\Omega$ minimizing $\mathbf{E}_{SB}$ can be found by searching for the shortest path on the graph constructed by the below strategy.

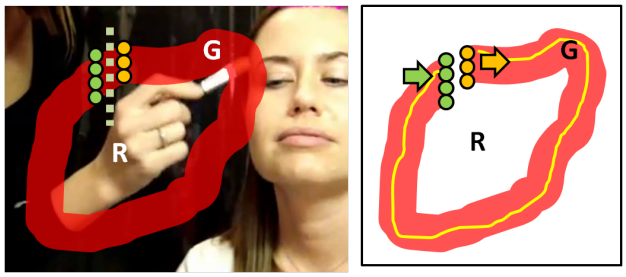


Figure 2. Refining the user specified mask boudary $\partial\Omega$ by finding the shortest path in $\mathbf{G}$.

As shown in Figure 2, we break the region $\mathbf{G} = \mathbf{O} \backslash \mathcal{R}$ to obtain a graph with each pixel in $\mathbf{G}$ being the vertices, and assign directed edges between neighboring pixels. The weight of an edge that goes from $\mathbf{x}_i^1$ to $\mathbf{x}_j^1$ is computed as $(1 - \min\{C^{1 \to t}(\mathbf{x}_j^1)$. The optimal boundary $\partial\Omega$ thus corresponds to the shortest path that goes from one of the starting vertices(colored as yellow) to one of the ending vertices(colored as green) on $\mathbf{G}$, which could be efficiently computed by most of the shortest path algorithms.

3.5.2 Optimal transition frame I_n

In motion concatenation, user creates a transition in video by specifying two frames I_m and I_n that should be concatenated. Given

I_m, our system refines I_n by selecting the best frame among frames near I_n according to the following criteria: the motion estimated from I_n to I_m should be reliable. Specifically, we pick the frame that has the lowest energy among all frames near I_n: function $\mathbf{E}_{TB}$:

$$\mathbf{E}_{TB} = \sum_{\mathbf{x}_i^m \in \Omega} (1 - C^{m \to n}(\mathbf{x}_i^m)) \tag{13}$$

$\mathbf{E}_{TB}$ measures the unreliability of the estimated motions between I_m and I_n;

As shown in Figure 3, it is interesting to point out that though we do not estimate the energy according to the appearance similarity between frames explicitly, the two local minimum in the curve actually corresponds to the frame that the make-up motion is about to repeat again. This is because that the estimated flow will be more robust when the two frame is more similar in their appearance and motion, which is a desired property for our case.

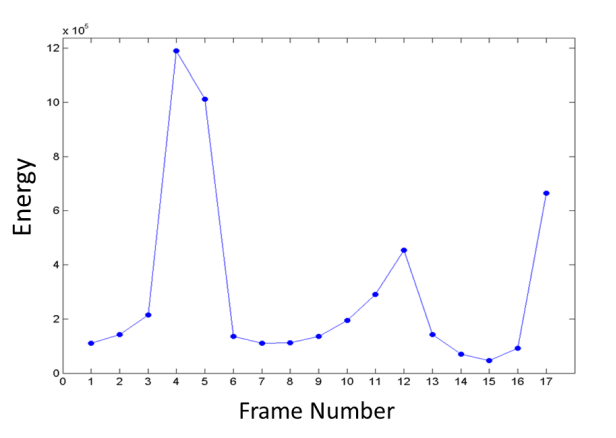


Figure 3. Visualizing $\mathbf{E}'_{TB}$ for the *Make up* sequence (Figure 1).

4. Results

In the result section, we evaluate both the motion localization and the motion concatenation operation on video sequences with different motion characteristics. We choose to demonstrate the effectiveness of our technique by using these two operations for creating high quality cinemagraphs. Specifically, we first use the motion localization operation to modify the video sequence to have motion occur only in part of the scene, and then we use the the motion concatenation operation to require the last few frames concatenate with the first frame. The final synthesized video is a video with both localized and repeating motion, which are the two major elements in a cinemagraph.

Before viewing the results, we give an overall explanation about the figures shown in this section. Figures in this section mainly illustrates how and how good our method is able to solve the spatial and temporal discontinuities and produces high quality cinemagraphs. For spatial discontinuities, we demonstrate our effectiveness by comparing individual frames generated by our method to those generated by the naïve masking method. For temporal discontinuities, which actually exists across frames, we developed the following way to better visualize them using static images. We pick certain points and visualize their two-dimensional motion trajectories in the original sequence, the sequence after motion localization, and the sequence after motion concatenation. The motion trajectories are drawn by tracking the points according to the estimated flow between every two neighboring frames. The trajectories start from frame 0 and end in frame T. To better visualize the trajectories, we fit a cubic spline to the sparse positions of a certain point in each frame, and zoom in by a factor of 4. We also color the trajectories from its start to its end, following the order of the HSV color wheel from 0 degree to 360 degree, which is identical to the color order of a rainbow(red, orange, yellow, green, blue, indigo and purple). By comparing the trajectories in three different sequences, we show how our method intended to closeout the gaps between the starting point and the ending point of the trajectories, which corresponds to the temporal discontinuities in the original sequence. For cases whose motion is complex, we also demonstrate the effectiveness of incorporating the confidence measure to our method.

In the following, we divide the test cases into three groups according to the motion characteristics contained in each sequence: rigid motions, non-rigid motions and complex motions, and discuss them separately. The final results are best viewed in video form. Please see the supplementary materials for the synthesized video sequences and more other results.

4.1. Rigid motions

Rigid motions refers to the motion class with objects moving rigidly. Such motions can often be faithfully captured and depicted by an optical flow field. Thus, the spatial and temporal boundary constraints imposed by flows can provide very good cues on correcting appearance discontinuity both spatially and temporally.

Make up. Figure 1 shows the result of this video sequence. The original sequence contains one's hand performing a repetitive making up motion. The major challenge rises as the contour specified by the user go through the wrist, which moves up and down along time. Without taking motion into account, there will be noticeable visual seems. Another challenge resides in the misalignment of the first frame and the last frame. Although the content of these two frames are visually similar, it is unlikely that the positions of the moving hand perfectly align to each other. These challenges together lead to severe spatial and temporal discontinuities when applying the naï"ve masking method.

In this example, to reduce spatial discontinuities along the boundary, our method tends to shift the position of the hands at each frame to meet the static wrist outside the mask. For temporal discontinuities, we pick three points and visualize their trajectories in Figure 1. The starting and ending points of the trajectories are originally separate by a certain distance and thus results in temporal discontinuities. After motion localization, the gap still exists, though the gap of the green point is greatly reduced already, as the green point lies near the aligned boundary. Finally after motion concatenation, all the gaps are fully resolved with the starting points and ending points of the trajectories stick to the same position. Notice how our method fills the gap while in the mean time preserving the original shape the motions between frames. Our resulting animated sequence looks natural in its motion and shows great improvements against the naïve masking method.

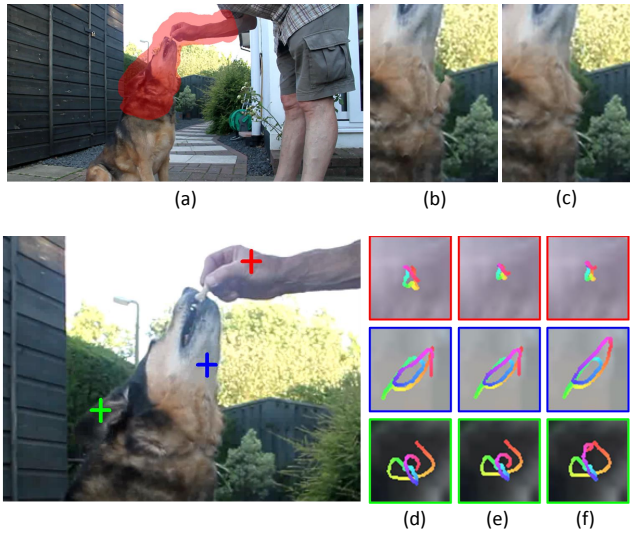


Figure 4. Feeding dog. (a)The user drawn mask on the first frame. (b)Part of a frame produced by the naïve masking method. (c)Part of a frame produced by our motion localization operation. We also visualize the motion trajectories of the selected points in the (d)original sequence (e)the sequence after motion localization, and (f)the sequence after motion concatenation.

Feeding dog. This is a testing sequence used in Tompkin *et al.*'s paper [13]. In this example, the user tries to generate a sequence that the dog bites the bone endlessly, while the body of the dog and the man remain still. The result provided by Tompkin *et al.* suffers from severe artifacts of obvious discontinuities around the dog's neck and the man's shoulder. In addition, there is also a noticeable jump between the first frame and the last frame, despite the quick motion in the sequence. Our method is able to overcome the discontinuities and generate a seamless looping sequence.

Mice. Here we evaluate our method on a sequence captured by a hand-held camera. Due to the drastic handshake, instead of directly computing the optical flow between the first frame and other frames, we accumulate the flow between neighboring frames from frame 0 to frame T. Figure 5 compares our result to the one generated by the naïve masking method. As the global camera motion causes the originally static background objects to move drastically, directly masking and pasting without adjusting the contents results in severe spatial discontinuities. As for the motion trajectories, due to the hand shake, all three points have drastic motions in the original sequence. After our motion localization operation, the green point on the background is nearly still, while the other two points still remains the motion of the mouse. After motion concatenation, the motions are modified to be loopable. Overall, our method is able to overcome the drastic camera motion, int the mean time preserving the subtle motion of the mouse.

4.2. Non-rigid motions

Non-rigid motions refer to the motion class with the objects performing elastic motion, such as pouring liquids, changing of facial expressions, blowing clothes and so on. For these types of motions, usually the estimated flow does not necessarily correspond to the true motion in the physical world. However, the motion field is still capable of capturing time-varying appearances of the objects well. Thus, warping with the flow field still gives reasonable results.

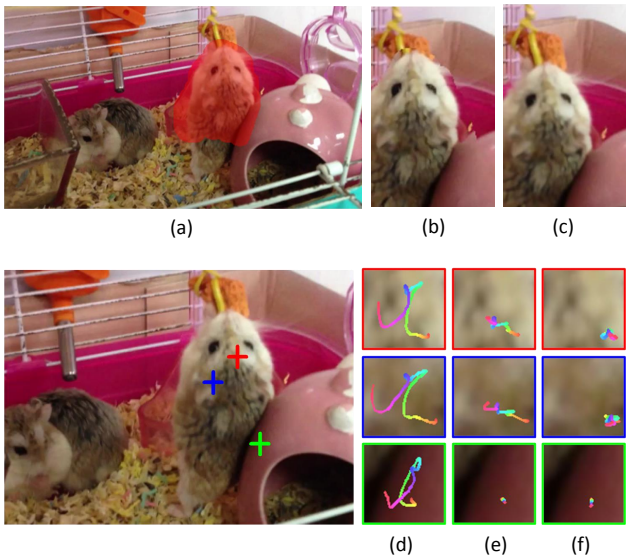


Figure 5. Mice. (a)The user drawn mask on the first frame. (b)Part of a frame produced by the naïve masking method. (c)Part of a frame produced by our motion localization operation. We also visualize the motion trajectories of the selected points in the (d)original sequence (e)the sequence after motion localization, and (f)the sequence after motion concatenation.

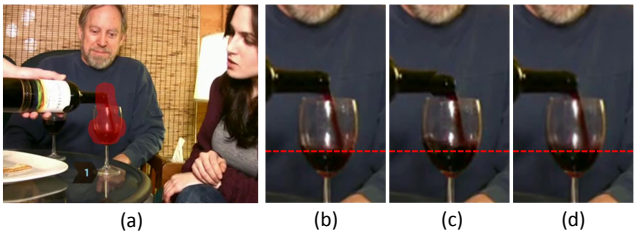


Figure 6. Pouring wine. (a)The mask specified by the user on the first frame. We compare the region of interest in (b)the first frame, (c)the T-1 frame generated by naïve masking and (d)the T-1 frame generated by our method. Notice the discontinuity around the bottle in (c) and different amount of wine in (c) and (d). Illustrating how our method resolves the temporal discontinuity.

Pouring wine. Pouring liquid is one of the popular cinemagraph topics users like to explore. Although simple in concept, it could be quite challenging for the naïve masking techniques to reach high quality results, as we demonstrated in this example. With the naïve masking methods, first, in order to avoid spatial discontinuities, the performer must try hard to keep the pouring bottle as still as possible. Second, as the amount of wine in the glass keep increasing throughout the sequence, there is probably no intuitive way to create a seamless looping sequence directly.

Our method is able to overcome these difficulties and greatly

reduces the effort users need to pay on setting up the scene. As the boundary of the user-specified mask goes across part of the bottle, the bottle is stabilized after motion localization. The seamless looping effect is produced by manipulating the shape of the wine in the last few frames, as illustrated in Figure 6. With the red dotted line, we can see our method "reduces" the amount of wine in the last few frames, in order to make the resulting sequence loop back to the first frame seamlessly.

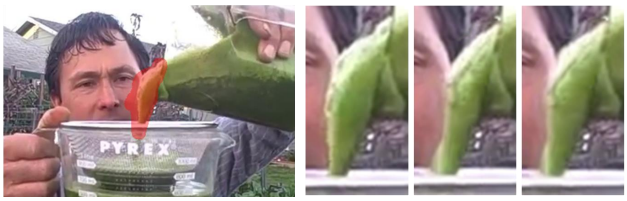


Figure 7. Pouring juice. (a)The mask specified by the user on the first frame. We compare the region of interest in (b)the first frame, (c)the T-1 frame generated by naïve masking and (d)the T-1 frame generated by our method. Notice that the synthesized frame (d) has a much similar appearance with (b) than (c). Illustrating how our method resolves the temporal discontinuity.

Pouring juice. This is another interesting example on liquid pouring. This example gives a clearer look on how the seamless effect is produced by our method. In the original sequence, the liquid in the first frame has a different appearance (both in shape and texture) with the one in the last frame, causing noticeably discontinuous transition every time advancing to the next loop.

As the estimated flow actually interprets the sequence as a non-rigid object changing its shape at a fixed position, our method can create a seamless transition by making the liquid in the last few frames grow thicker to resemble the liquid in the first frame.

Eating. This is another type of non-rigid objects. The original sequence presents a boy chewing the sandwich. Apparently the performer is unlikely to stay still during chewing, not even mentioned how hard it would be for one to perform a chewing motion that could loop seamlessly. Applying the naïve masking method gives poor results that comes with severe spatial and temporal discontinuities. Our approach works out by manipulating the shape of the face, and is able to achieve high quality results despite the change of lighting conditions across frames.

4.3. Complex motions

Complex motions refers to the class of motions that cannot be well represented by two-dimensional optical flow field. Examples include burning flames and changing numbers on electronic clocks. These cases point out the necessity of incorporating the confidence measure for the estimated flow into our system. Flows with lower confidence values would be discarded, preventing them from messing up the results. It is interesting to mention that, in the case where almost every flow on the complex objects is considered unreliable, the objects will actually remain the same as in the original sequence. In this case, since the motion of complex objects are originally full of spatial and temporal discontinuities, leaving the frames untouched often gives acceptable results.

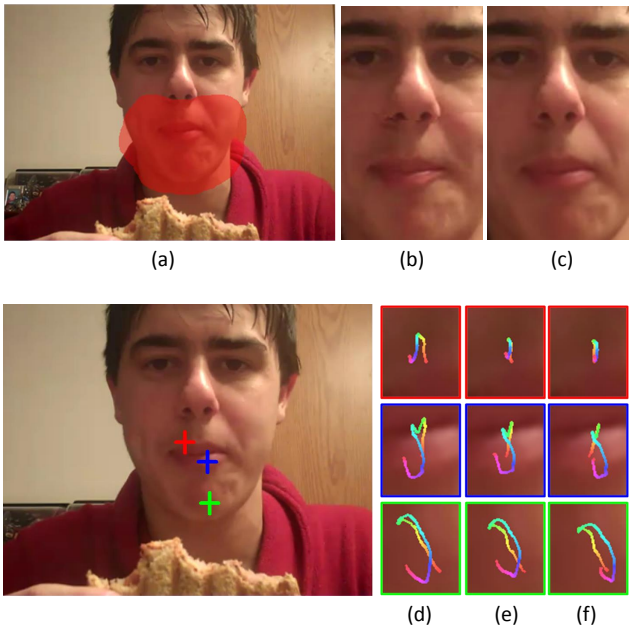


Figure 8. Eating. (a)The user drawn mask on the first frame. (b)Part of a frame produced by the naïve masking method. (c)Part of a frame produced by our motion localization operation. We also visualize the motion trajectories of the selected points in the (d)original sequence (e)the sequence after motion localization, and (f)the sequence after motion concatenation.

Pedestrain light. In this sequence, the user tries to generate a sequence with the pedestrian light counting down from 5 to 1 repeatedly. The original sequence contains mainly two parts of motions. One is the camera motion due to the drastic handshake and the other is the motion of the time counter (changing numbers on the light patterns). The main challenge is thus to remove the global camera motion while preserving the local counting down motion.

Figure 9 shows the results obtained by our method with and without the confidence map. As the counting down motion can't be properly represented with a 2-D vector field, the flow estimation can fail dramatically. Warping without taking the confidence measures into account may cause the result to corrupt since pixels are forced to adjust their positions according to wrong flows. In this sequence, the confidence measure improves the result by greatly reducing the impact of the unreliable constraints and keeping the pixels at their original positions.

Stove. In this sequence, the user tries to generate a cinemagraph with the fire burning endlessly. Similar to the previous example, we show the results obtained with and without the confidence measure. As the motion of the flame is hard to capture with optical flow estimation, optimizing according to the wrong flow produces noticeable artifacts. In contrast, by evaluating the confidence of the flow, our method leave the fire to its original appearance but the final sequence is still able to loop naturally.

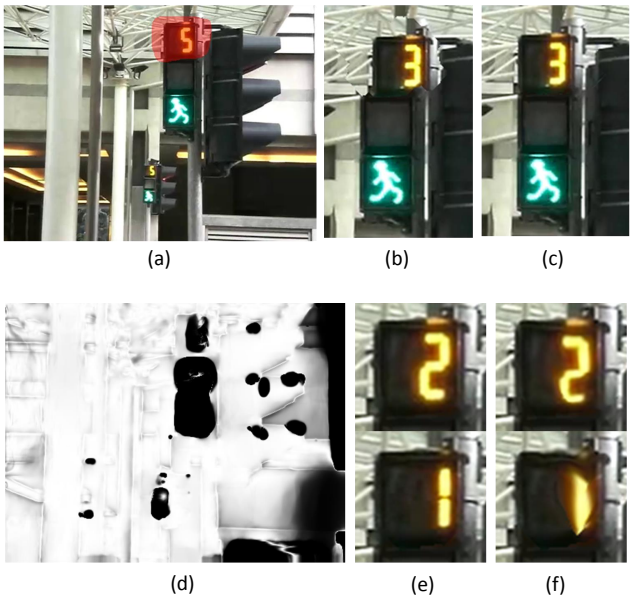


Figure 9. Pedestrain light. (a)The user drawn mask on the first frame. (b)Part of a frame produced by the naïve masking method. (c)Part of a frame produced by our motion localization operation. (d)the confidence map of the flow between the first frame and the last frame. We also compare the motion concatenation result (e)with and (f)without the confidence map.

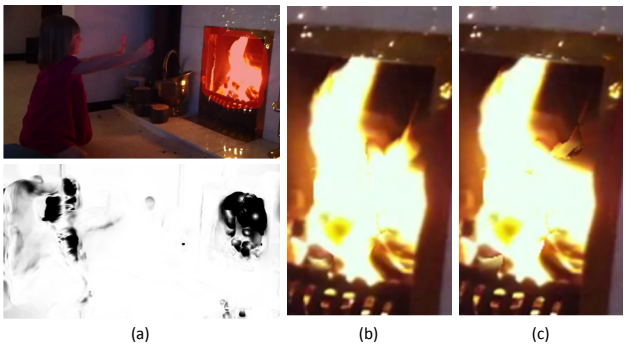


Figure 10. Stove. (a)The user drawn mask on the first frame and the confidence map of the flow between the first frame and the last frame. We also compare the motion concatenation result (b)with and (c)without the confidence map.

5. Conclusions

In this paper, a warping based video processing framework is proposed. We demonstrate how this framework can facilitate interesting operations for video-motion editing, and effectively resolve the spatial and temporal discontinuities. We also evaluate the proposed method on various type of videos and provide detail analysis on how our method work in practice. We believe our work has further investigated and derived insights to the problem of video-motion editing.

References

[1] iCinegraph. http://www.icinegraph.com/. 2

[2] kinotopic. http://kinotopic.com/. 2

[3] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Selectively de-animating video. *ACM Transactions on Graphics*, 2012. 2

[4] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:500–513, 2011. 3

[5] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. Salesin, and R. Szeliski. Animating pictures with stochastic motion textures. *ACM Trans. Graph.*, 24(3):853–860, 2005. 2

[6] N. Joshi, S. Mehta, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen. Cliplets: Juxtaposing still and dynamic imagery. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pages 251–260, 2012. 2

[7] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross. Nonlinear disparity mapping for stereoscopic 3D. *ACM Trans. Graph.*, 29(4, article 75), 2010. 3

[8] Z. Liao, N. Joshi, and H. Hoppe. Automated video looping with progressive dynamism. *ACM Trans. Graph.*, 32(4):77:1–77:10, July 2013. 2

[9] Z. Lin, L. Wang, Y. Wang, S. B. Kang, and T. Fang. High resolution animated scenes from stills. *IEEE Transactions on Visualization and Computer Graphics*, 13:562–568, 2007. 2

[10] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. *ACM Trans. Graph.*, 24(3):519–526, July 2005. 2

[11] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3D video stabilization. *ACM Trans. Graph.*, 28(3, article 44), 2009. 3

[12] A. Schödl, R. Szeliski, D. H. Salesin, R. S. D. H. Salesin, and I. Essa. Video textures. In *Proceedings of ACM SIGGRAPH*. 2

[13] J. Tompkin, F. Pece, K. Subr, and J. Kautz. Towards moment images: Automatic cinemagraphs. In *Visual Media Production (CVMP), 2011 Conference for*, pages 87–93, November 2011. 2, 6

[14] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4):80:1–80:10, July 2013. 2

[15] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5, article 118), 2008. 3

[16] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012. 2

[17] X. Xu, L. Wan, X. Liu, T.-T. Wong, L. Wang, and C.-S. Leung. Animating animal motion from still. *ACM Trans. Graph.*, 27(5):117:1–117:8, 2008. 2