Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Preserving global features of fluid animation from a single image using video examples*

Yan GUI†[1], Li-zhuang MA[1], Chao YIN[1], Zhi-hua CHEN[2]

([1]*School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*)

([2]*Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China*)

†E-mail: guiyan122@sjtu.edu.cn

Received Nov. 18, 2011; Revision accepted Mar. 29, 2012; Crosschecked May 31, 2012

Abstract: We synthesize animations from a single image by transferring fluid motion of a video example globally. Given a target image of a fluid scene, an alpha matte is required to extract the fluid region. Our method needs to adjust a user-specified video example for producing the fluid motion suitable for the extracted fluid region. Employing the fluid video database, the flow field of the target image is obtained by warping the optical flow of a video frame that has a visually similar scene to the target image according to their scene correspondences, which assigns fluid orientation and speed automatically. Results show that our method is successful in preserving large fluid features in the synthesized animations. In comparison to existing approaches, it is both possible and useful to utilize our method to create flow animations with higher quality.

Key words: Single image, Video example, Fluid feature, Fluid motion, Flow animation
doi:10.1631/jzus.C1100342 **Document code:** A **CLC number:** TP391.4

1 Introduction

When we contemplate an image with a natural scene, a large number of distinguishable elements contained in the image, such as water, fire, and smoke, often cause us to imagine that the scene is in motion. In the past decade, many approaches proposed for creating a flow animation have achieved significant progress. Nevertheless, how to design a continuous flow animation from still images remains a challenging problem, even though some attempts have been made to address this issue (Chuang *et al.*, 2005; Okabe *et al.*, 2009; 2011).

The aim of animating fluid images is to capture dynamic quality while preserving fluid features of video examples in the synthesized flow animations. In this paper, we make a modification to the proposed animation system that has been highlighted in Okabe *et al.* (2009). Our system is equipped with the following distinguishing features: high accuracy optical flow estimation, automatic flow field design, and spatio-temporal coherency. To extract high quality residuals, a large displacement optical flow algorithm (Brox and Malik, 2011) is used to estimate fluid motions even if video examples exhibit drastically changing dynamics. To obtain the flow field from the target image, our method replaces the manual design process with automatic transfer of motion using the scale invariant feature transform (SIFT) flow algorithm (Liu *et al.*, 2008), based on a large fluid video database. Finally, to provide equivalent residuals of the target image with spatio-temporal coherency, the video example needs to be resized using the graph-cuts based seam carving algorithm (Rubinstein *et al.*, 2008), depending on the fluid region of interest in the target image.

* Project supported by the National Basic Research Program (973) of China (No. 2011CB302203) and the Innovation Program of the Science and Technology Commission of Shanghai Municipality, China (No. 10511501200)

Following this process, the equivalent residuals are then obtained by warping the residuals decomposed from the video example.

2 Related work

The many methods proposed for the creation of animations from a single picture can be roughly categorized into three classes. The first class employs physically based fluid techniques to simulate and control a wide variety of fluids. Shinya *et al.* (1999) synthesized the stochastic motion of plants under the influence of wind. Treuille *et al.* (2003) applied a continuous quasi-Newton optimization for controlling smoke simulations through user-specified keyframes. Chuang *et al.* (2005) focused on animation of a natural phenomenon through synthesizing a video with motion textures from a still picture. However, these methods are computationally expensive when setting appropriate parameters for a type of fluid.

The second class of techniques has been developed to animate a single image by efficiently manipulating the image objects via user control. Freeman *et al.* (1991) used oriented filters to create the illusion of motion in a single image that appears to move continuously without movement. Litwinowicz and Williams (1994) used line drawings to create 2D animations by deforming an image as keyframes. Barrett and Cheney (2002) proposed object-based image editing for animation and manipulation of static pictures. Igarashi *et al.* (2005) presented an interactive system that performs shape deformation by controlling user-specified handles to create 2D animations. However, these methods that are usually proposed for creating 2D character animations are outside the focus of our research.

The third, most recent class of techniques creates continuous flow animations relying on dynamic textures. Schodl *et al.* (2000) presented a technique for analyzing a video, where essentially, the transitions between similar frames take place to synthesize a seamless video of arbitrary length. Doretto *et al.* (2003) applied an auto-regressive moving average (ARMA) model to infer the stochastic motion to generate infinitely long videos. Wang and Zhu (2003) proposed a generative model to represent an image with a number of bases and analyzed their motions in order to synthesize video sequences. Kwatra *et al.* (2003) synthesized video textures by merging the 3D space-time blocks of video with graph-cuts. However, these methods did not address the problem of animating a fluid picture. Sun *et al.* (2003) proposed a video-input driven animation (VIDA) system to drive the physical simulation of synthetic objects. Bhat *et al.* (2004) proposed an algorithm to synthesize a fluid animation by analyzing the motion of texture particles of video examples along user-specified flow lines. Lin *et al.* (2007) synthesized dynamic motion from a small collection of high resolution stills. More recently, a complete system (Okabe *et al.*, 2009) has been proposed to design a continuous flow animation by adding motion to a fluid of a video example in an image. Okabe *et al.* (2011) further presented a method for synthesizing fluid animations from a single image using a video database, which is an extension of the original synthesis algorithm.

3 Video examples based animation system

3.1 Analysis of the existing animation system

The system proposed by Okabe *et al.* (2009) is divided into two main phases, video analysis and video synthesis. Given a video example, the analysis procedure consists of decomposing the video example into the flow field and the residuals. However, computing the flow field using the Lucas-Kanade method (Lucas and Kanade, 1981) may fail to obtain smooth velocity when the video example exhibits drastically changing fluid motions, in that it computes optical flow only at feature points.

The video synthesis procedure consists of flow field design and transfer of residuals. The manual design process for obtaining flow field works well for various dynamic scenes in given target images, but it is inappropriate for scenes characterized by a lot of small or ambiguous fluid features, since it might require a certain amount of user interactions. The system automatically transfers residuals of the video example to generate the equivalent residuals over the target image by patch-based sampling. However, we observe that the designed flow field causes visual discontinuities between patches because there are independent fluid features and undesirable intensity or illumination discrepancies in patches. In addition, the method is time-consuming due to exhaustive patch

searching.

3.2 Overview of our system

To overcome the aforementioned shortcomings, we present an improved system which could be decomposed into four components, including video adjustment, video analysis, motion estimation, and video synthesis (Fig. 1). Given a target image of fluid (Fig. 1a), the first step is to detect the fluid region of interest in the target image. To accomplish this, an alpha matte (Fig. 1b) is required to extract fluid parts from the scene. Then, the adjustment of a given video example (Fig. 1c) is performed in order that the resized video example (Fig. 1d) has the same size as the fluid region. Such effective resizing of the video example seeks to intactly preserve the inherent important fluid features.

Given the resized video example, the analysis of the video example consists of obtaining the average image (Fig. 1e) and the differences (Fig. 1f), where the optical flow is computed by employing large displacement based motion estimation for flow field (Fig. 1g) extraction. The residuals (Fig. 1h) are obtained as differential images between the differences and corresponding warped differences according to the flow field, and also capture the non-translational high-frequency information of the video example.

To obtain the flow field of the target image, our system automatically estimates motion field from a single image using a fluid video database. A matched image (Fig. 1i) with a visually similar scene is found from the video database by applying the SIFT flow algorithm. The transferred flow field (Fig. 1k) over the target image is then obtained by warping the temporal optical flow field (Fig. 1j) of the matched image based on the computed SIFT flow. With this method, the manual design process for obtaining the flow field of the target image is no longer required.

Given the extracted residuals and the transferred flow field, our system first generates new residuals of arbitrary length using the video textures technique, and then warps the synthesized residuals according to the transferred flow field to produce the equivalent residuals over the target image (Fig. 1l). Finally, the system combines the approximated average image (Fig. 1m) and the synthesized differences (Fig. 1n) to create a continuous flow animation (Fig. 1o). The final appearance is synthesized by adjusting the appearance between the synthesized

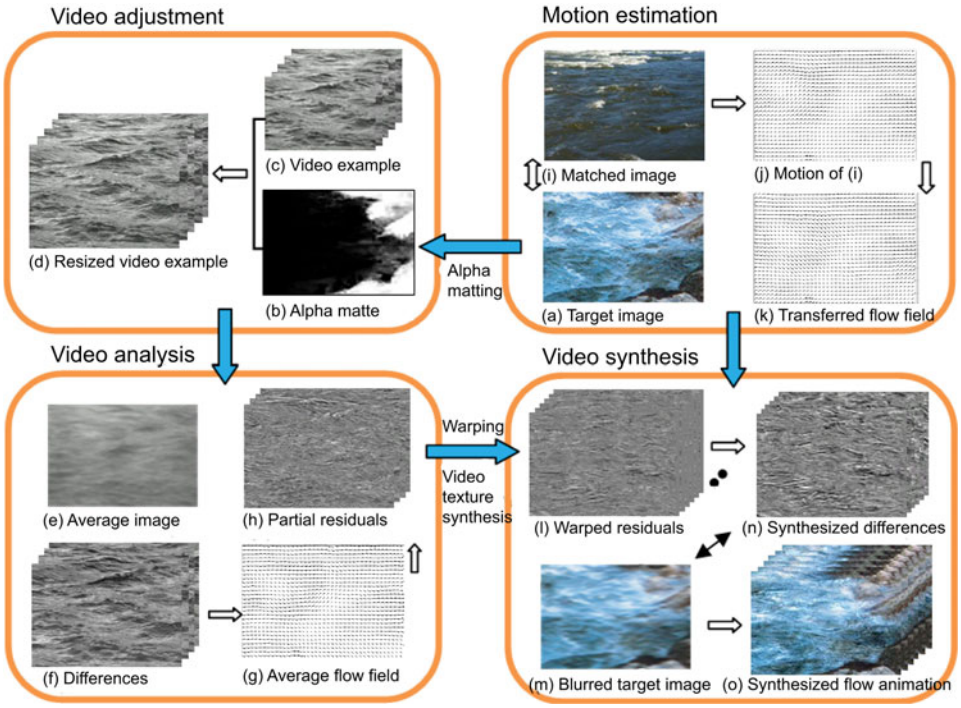

Fig. 1 Overview of our system which consists of four parts, including video adjustment, motion estimation, video analysis, and video synthesis

flow animation and the target image using histogram matching.

The differences between our proposed system and the existing animation system include: (1) Our system automatically designs the flow field of the given target image using a fluid video database; (2) Our current methodology involves resizing a video example in terms of the fluid region in the target image and synthesizing residuals globally. The implementation details for creating a flow animation from a single image will be discussed in the next section.

4 Feature preserving animations of fluid pictures

4.1 Video examples adjustment

The scene in a given target image includes not only fluid regions but also still objects such as rocks or trees. Our objective is to synthesize fluid regions into animated sequences, and to keep still objects invariable in the synthesized animation. We apply a closed-form formula to natural image matting (Levin *et al.*, 2008) based on limited user inputs. With this method, two open parameters, the weight of the regularization term ε and the window size, have an impact on matte reconstruction. In our implementation, we set $\varepsilon=10^{-7}$ to avoid oversmoothed alpha mattes. Since we process images based on a multi-resolution scheme, the window size will be set to 3×3 for reducing computation time. Fig. 2a shows the target images, and Figs. 2b–2e show the intermediate results by using natural image matting. Finally, we use the resulting alpha matte as the fluid region of interest in the target image.

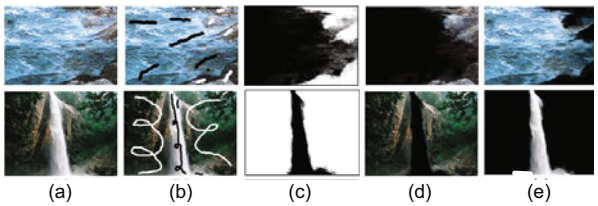

Fig. 2 Results of natural image matting. (a) Target images; (b) User inputs; (c) Extracted alpha mattes; (d) Extracted foregrounds; (e) Extracted backgrounds

Obviously, the sizes of the fluid regions between a user-selected video example and the target image are not equal to each other (The size of the fluid region is the size of the bounding box that surrounds the whole dark region in the extracted alpha matte). That is why the existing animation system (Okabe *et al.*, 2009) performs automatic transfer of residuals, in order that the transferred residuals with respect to the target image are satisfactory. However, one significant problem in the transfer of residuals is the presence of broken fluid features and the color discrepancy in each animated frame, in that global features of the fluid cannot be maintained after local transfer. By contrast, we apply a graph-cuts based seam carving method (Rubinstein *et al.*, 2008) to adjust the video example according to the size of the fluid region in the target image. The final resized video example is obtained by removing or inserting all the computed vertical and horizontal seams.

As shown in Fig. 3, the top row (Figs. 3a–3e) illustrates five original frames randomly selected from a video example. The significant fluid features of the video example can be well preserved in the resized frames (Figs. 3f–3j). As the last row of each resized result shows, excessive expansion of the video example by using seam carving would most probably create blurred fluid features because of pixel duplication (Figs. 3m and 3n), even though the expansion performed in the vertical direction can produce a perfect result (Fig. 3l). Traditional image resizing is sufficient, since it is good at handling textured regions through uniform scaling, and achieves impressive results (Fig. 3o). In addition, a cropping operation is useful when the fluid region in the target image is composed of multiple independently connected regions, which can be used to cut the most important regions from the video example.

4.2 Flow field and residuals extraction

The video analysis phase deals mainly with the extraction of the flow field and residuals from the resized video example, depending on optical flow computation. Corpetti *et al.* (2002), Brox *et al.* (2004), and Courty and Corpetti (2007) have developed several approaches for estimating optical flow fields. We use the method proposed by Brox and Malik (2011) to handle video examples that have drastically changing fluid motions. This method constructs a variational model with a coarse-to-fine warping scheme. In this manner, our system can estimate an optical flow with the same high

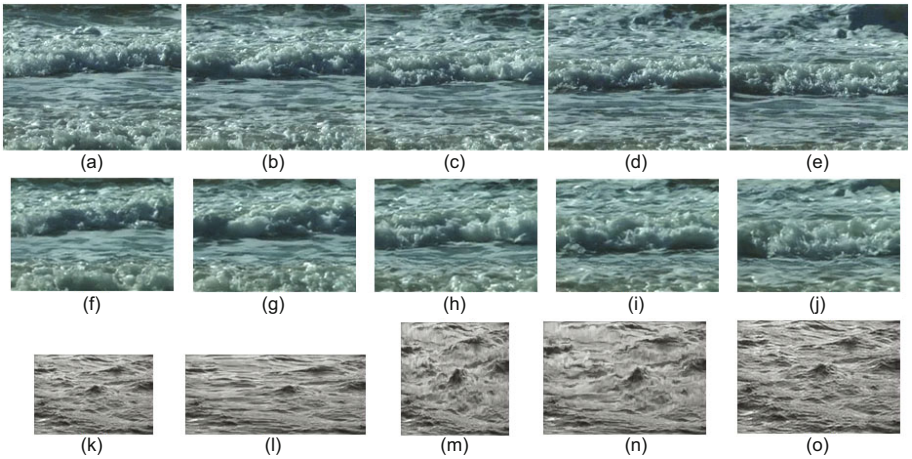

Fig. 3 Adjustment of video examples. (a)–(e) are original frames of a video example with size 288×352; (f)–(j) are resized frames corresponding to the original frames with size 176×254; (k) is the original frame of the other video example with size 112×186; (l) is vertical expansion with size 112×254; (m) is horizontal expansion with size 176×186; (n) is the resized frame with size 176×254 using the seam carving algorithm; (o) is the resized frame with size 176×254 using traditional scaling

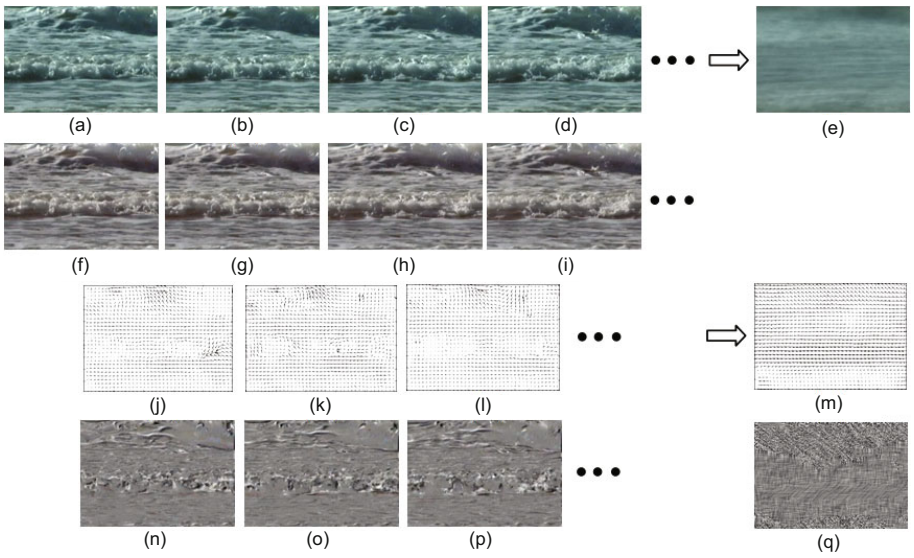

Fig. 4 Video analysis procedure. (a)–(d) are successive frame sequences of a given video example; (e) is the average image; (f)–(i) are the differences; (j)–(l) are the optical flow field; (m) is the average flow field of the video example, which is visualized with a noise image using line integral convolution (q); (n)–(p) are the residuals

accuracy from the variational optical flow setting, handling the video example with drastically changing fluid motions. In Fig. 4, as the third row of each result shows, dense optical flow fields are estimated correctly from a given video example, which exhibits a dynamic scene like ocean surf.

We use the variational optical flow algorithm for capturing inherent fluid motion from resized video examples. To extract the flow field and the residuals, the video analysis procedure is summarized as follows.

Step 1: Compute the average image and differences. We first average all the frames of the video example as $A = (\sum_{i=1}^{N} F_i)/N$, where N is the total number of frames and F_i respects a frame of the video example (Fig. 4e). We then compute the differences as $D_i = F_i - A$ (Figs. 4f–4i).

Step 2: Construct the flow field. Two neighboring differences D_i and D_{i+1} are taken as inputs and the optical flow field is generated (Figs. 4j–4l). Then

we average the optical flows through all the differences to obtain the flow field $\boldsymbol{\nu}(\boldsymbol{x})$ (Fig. 4m), where $\boldsymbol{x} = (x, y)$ is a pixel position.

Step 3: Compute the residuals. The residuals (Figs. 4n–4p) are obtained as the differential image R_i between the $(i+1)$th frame of differences and the warped ith frame of differences, which is defined as follows:

$$R_i = D_{i+1} - W(D_i), \tag{1}$$

$$W(D_i) = D_{i-1}(\boldsymbol{x} - \boldsymbol{\nu}(\boldsymbol{x})), \tag{2}$$

where $W(\cdot)$ is a warping function based on backward mapping that warps differences according to the constructed flow field $\boldsymbol{\nu}(\boldsymbol{x})$.

4.3 Flow field estimation of target images

In this subsection, we describe the algorithm that has been used for estimating the flow field from the target image. The basic idea is to find an image (video frame) that has a visually similar scene to the target image, from a fluid video database. To ensure efficient best-match searching, we construct a large enough fluid video database consisting of 479 videos among various scenes, including waterfalls, rivers, fires, and smoke. An information rich video database is important to achieve high accuracy data-driven scene matching. We collected these videos from the Internet and commercial material databases or shot them by ourselves. The videos satisfying our requirements must meet the conditions that the fluid is the primary features of the video and that other objects in the video are still. In addition, all videos are of high quality, with a resolution range from 320×240 to 640×480 at 25 frames/s, and are captured within 10–30 s. However, we found that in most cases, the videos differed in size from the target images. To make scene possible, during each search, we simply scale or crop each video to match with the size of the corresponding target images.

To perform a fast search, we extract SIFT features of each frame (Figs. 5b and 5e) by applying the SIFT keypoint detector method (Lowe, 2004), and compute a 128-dimensional feature vector as the SIFT descriptor (Figs. 5c and 5f) at each point. We then apply the SIFT flow algorithm (Liu *et al.*, 2008) to perform scene alignment between the target image and each nearest neighbor. The video frame with the maximum alignment score is the best-match image. We compute the optical flow field of the selected frame from its attached video. Note that the nearest neighbors may choose either from different videos or from one video.

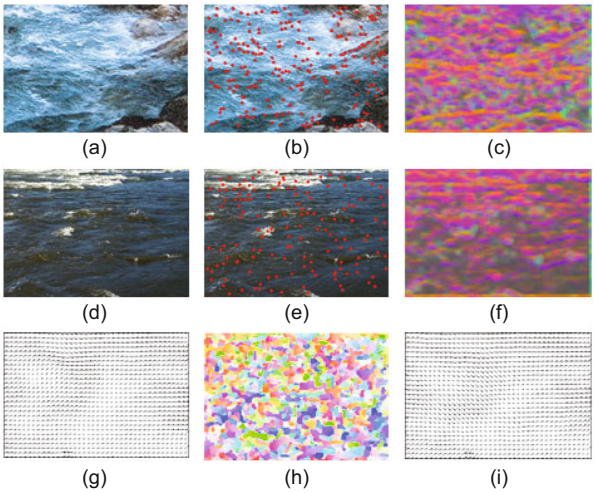

Fig. 5 Automatic transfer of fluid motion. (a) is the target image and (d) is the best-match image; (b) and (e) show the extracted SIFT features; (c) and (f) are the quantized SIFT descriptors, visualized using the color-coding scheme; (g) is the optical flow field over (d); (h) shows the SIFT flow field; (i) is the transferred flow field over (a)

Although the best-match image has a similar scene to the target image, there are notable differences in both color values and gradients. The SIFT flow algorithm allows for the construction of meaningful correspondences among objects located at different parts of the scene. Given the best-match image and the target image, the SIFT flow field is obtained when solving the constructed correspondence. The flow field of the target image is obtained by warping the temporal optical flow field of the best-match image over the image space using the computed SIFT flow field. Readers may argue that there exist flow vectors with non-zero values with respect to the still objects in the target image. The fact is that, however, we will ultimately replace the fluid animation created from such flow vectors with the still objects. Fig. 5 illustrates the designed flow field of the target image while automatically transferring the optical flow field of the best-match image.

4.4 Animation synthesis

Once we obtain the residuals of the video example and the designed flow field of the target image, a key task is to specify equivalent residuals over the target image. We first take a short residuals

clip as a sample to generate similar looking residuals of arbitrary length. We do this by using a video texture synthesis technique (Schodl *et al.*, 2000) to make an infinitely flowing animation. On the other hand, since the motion of fluid features in the synthesized residuals must conform to the underlying motion pattern of the target image, we then warp the synthesized residuals according to the designed flow field to obtain the equivalent residuals of the target image. Given the synthesized residuals R, the equivalent residuals R' over the target image are obtained using the following equations:

$$R'_i = W'(R_i) = R_i(\boldsymbol{x} - \boldsymbol{\nu}'(\boldsymbol{x})), \tag{3}$$

where $W'(\cdot)$ is the warping function based on backward mapping according to the designed flow field $\boldsymbol{\nu}'(\boldsymbol{x})$. Other equivalent information is the average image approximated from the target image by using a motion blur method (Brostow and Essa, 2001). We proceed to interpolate paths along those pixels of the target image that will be blurred according to the designed flow field. We then apply Gaussian filtering to obtain the smoother average image A'. Fig. 6 shows the approximated average image of the target image.

Fig. 6 Approximation of the average image. (a) Target image; (b) Motion blurred image; (c) Average image approximated using the Gaussian smoothing filter

Given the warped residuals R' and the warping function W', we can reconstruct the differences D' by accumulating the residuals within the life time parameter τ using a composite function W'^n:

$$D'_i = \sum_{k=i-\tau}^{i-1} W'^{(i-k-1)}(R'_k), \tag{4}$$

$$W'^n = W' \circ W'^{(n-1)}(R'_k),\ n = i-k-1. \tag{5}$$

We then combine the approximated image A' and the reconstructed differences D' to create the final flow animation defined as $F' = A' + D'$, which is computed from F'_τ to F'_{N-1} where N is the total number of warped residuals R'. There exist distinguished contrast and brightness between the target image and the reconstructed differences D'. We then apply pyramid-based texture synthesis with histogram matching (Heeger and Bergen, 1995) to reduce the contrast and recover the original appearance of each animation frame. With this method, the original appearance of the target image is well preserved in the synthesized flow animation. Fig. 7 demonstrates that the contrast and brightness of the reconstructed frame are recovered using histogram matching.

Fig. 7 Contrast and brightness adjustment using histogram matching. (a) Target image; (b) A reconstructed frame; (c) Final synthesized frame preserving the original appearance of the target image

5 Experiment results and discussions

Fig. 8 illustrates a comparative study between our method and the existing animation system proposed by Okabe *et al.* (2009). The main difference is the generation of residuals for target images. Given the target image (Fig. 8a) and the video example (Fig. 8g), the existing animation system usually copies fragments of residuals of the video example to produce the residuals of the target image. This is performed by initiating the best-match search between the flow fields of the video example and the target image. As shown in Figs. 8b–8f, high levels of verbatim copying are found in these partially transferred residuals. When these residuals are used to synthesize the final flow animation, noticeable visual discontinuities appear in each animated frame (Figs. 8m–8r). In contrast, our method attempts to adjust the video example according to the fluid region in the target image. The residuals over the target image (Figs. 8h–8i) are obtained by warping the residuals of the resized video example. Our methodology, which simply focuses on using such residuals, successfully preserves the large fluid features in the final synthesized frames (Figs. 8s–8x).

In Fig. 9, we applied our method to various types of fluid images. As demonstrated by these examples,

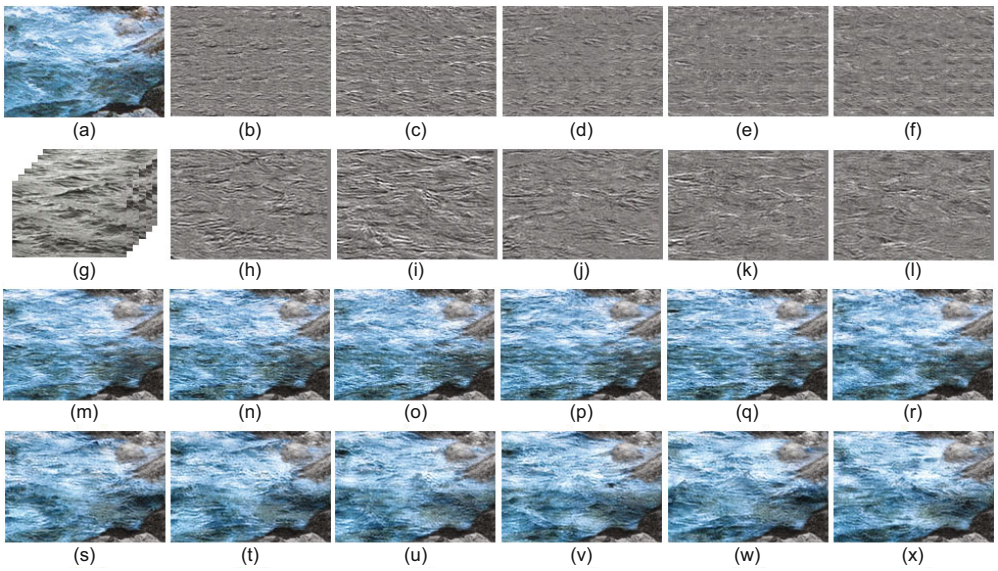

Fig. 8 A comparison between our proposed system and the existing flow animation system proposed by Okabe *et al.* (2009). (a) Target image; (b–f) Partially synthesized residuals by Okabe *et al.* (2009); (g) Video example; (h–l) Partially synthesized residuals by our method; (m–r) Partially synthesized flow animation by *et al.* (2009); (s–x) Partially synthesized flow animation by our method

these frames show both the fluid region and the still objects. The alpha matte extracted from the target image is useful for recovering the still objects of the target image in the synthesized animation. On the other hand, specifically for the flame and smoke animations, even though the approximated average image has a static overall shape, the shapes of the fluid region in the resulting flow animations are variable when the alpha matte is used to perform composition with the dynamic fluid features of the synthesized animation.

As shown in Fig. 10, the target images contain unconnected fluid regions, surrounded by rectangles of two different colors, which can be animated using different video examples. Taking the first target image as a sample, we use a graph-cut based seam carving technique to resize the given video examples. Compared to the second target image, since undesired objects contained in the given video examples, such as stone wall in the first video example and two ducks in the second video example, have much impact on flow animation synthesis, we usually use the video pieces by applying the cropping operation to cut the video examples, which contain only dynamic scenes. The last four columns show the partially animated frames in a synthesized flow animation.

Table 1 summarizes the information related to the results in Figs. 8–10. In our implementation, we use a workstation with an Intel® Xeon® 2.80 GHz CPU and 6 GB RAM for synthesizing the flow animations. As shown in Table 1, the second column gives the resolution of the synthesized flow animations. The third and fourth columns show that the fluid regions usually have different sizes from the given video samples. The algorithm spends 3–5 min on the adjustment of video examples mainly because the 3D graph-cut based seam carving technique is time-consuming. It is also computationally cheap to use traditional scaling or cropping operations. The time on scene matching relies on the number of videos. The algorithm spends more than 10 min on matching river scenes and 3–7 min on selecting flame and smoke scenes. The fluid video database collects 242, 110, 55, and 72 videos for water, fire, smoke, and cloud scenes, respectively. The transfer of fluid motion is fast because the SIFT descriptor and the optical flow field of the best-match scene are pre-computed. τ is the life time for the video example, which controls the quality of the reconstructed differences. We set τ to a large value when the video example has drastically changing dynamics, and a small value when the video example exhibits stationary dynamics. The last column shows the number of frames for each synthesized flow animation when no video synthesis technique is used.

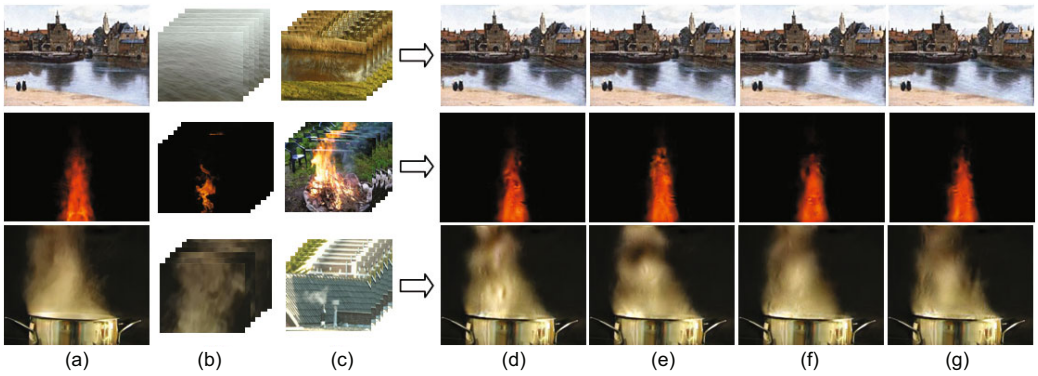

Fig. 9 Synthesized flow animations from various types of fluid images using our method. (a) Target images that contain river, flame, and smoke scenes from top to bottom; (b) Video examples; (c) Retrieved videos from the fluid video database; (d–g) Partial animation frames

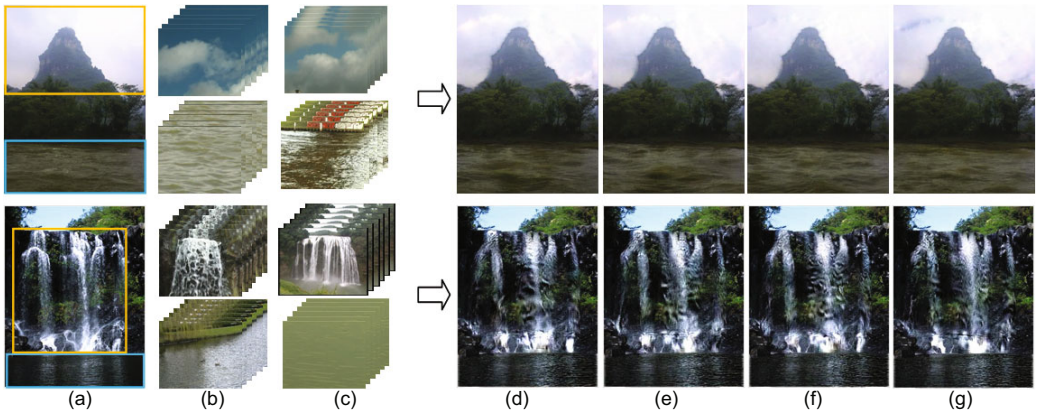

Fig. 10 Flow animation design by applying the fluid motions extracted from more than one video example. (a) Target images; (b) Video examples; (c) Retrieved videos from the fluid video database; (d–g) Partial animation frames

Table 1 Information for the synthesized flow animations

Scene	Resolution	Fluid region	Video example	Resizing time (min)	Scene matching time (min)	Motion transfer time (s)	τ (frame)	Number of frames
I(river)	176 × 254	176 × 254	112 × 168	5	15	40	5	59
II(river)	208 × 298	109 × 298	288 × 325	2	16	10	5	150
III(flame)	240 × 320	170 × 90	240 × 320	1	7	10	12	88
IV(smoke)	202 × 247	156 × 160	288 × 352	3	5	12	7	96
V(cloud)	384 × 288	187 × 288	288 × 352	3	3	15	5	146
V(river)	384 × 288	112 × 288	288 × 352	5	12	10	5	146
VI(waterfall)	400 × 310	275 × 245	288 × 352	1	5	20	12	100
VI(river)	400 × 310	75 × 310	288 × 352	1	10	10	5	100

τ is the life time for the video example, which controls the quality of the reconstructed differences

There are some limitations to our animation system. (1) The selection of video examples: we need to carefully specify the video examples suitable for the target images; (2) The flow field estimation from a single image: even though the automatic flow field design for the target image reduces a large amount of user interactions, we cannot control the orientation or the speed in the designed flow field because of the fully automatic transfer of fluid motion; (3) The quality of the synthesized flow animation: our system does not handle the target images or video examples with high resolution. On the other hand, the original detailed features in the target image are covered up in the synthesized flow animation when

transferring the fluid features of the video example.

6 Conclusions

We have proposed a modified method for synthesizing a continuous flow animation from a single image using a video example. We resize the video example using video retargeting techniques based on an alpha matte that extracts the fluid region. Thus, our system involves transferring the fluid features of the video example globally, which successfully preserves the integrity of the fluid features. Our system searches for the best-match scene based on a large fluid video database and obtains the flow field of the fluid image, which reduces a certain amount of user interactions. A key challenge that we have faced is in maintaining the original features of the target image. Experimental results demonstrate that our method works well for creating a continuous flow animation of various types of fluid images. In our future work, we plan to build an information rich video database that contains almost all possible videos with fluid scenes.

References

Barrett, W.A., Cheney, A.S., 2002. Object-Based Image Editing. SIGGRAPH, p.777-784. [doi:10.1145/566570.566651]

Bhat, K.S., Seitz, S.M., Hodgins, J.K., Khosla, P.K., 2004. Flow-based video synthesis and editing. *ACM Trans. Graph.*, **23**(3):360-363. [doi:10.1145/1015706.1015729]

Brostow, G.J., Essa, I., 2001. Image-Based Motion Blur for Stop Motion Animation. SIGGRAPH, p.561-566. [doi:10.1145/383259.383325]

Brox, T., Malik, J., 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(3):500-513. [doi:10.1109/TPAMI.2010.143]

Brox, T., Bruhn, A., Papenberg, N., Weickert, J., 2004. High Accuracy Optical Flow Estimation Based on a Theory for Warping. 8th European Conf. on Computer Vision, p.25-36.

Chuang, Y.Y., Goldman, D.B., Zheng, K.C., Curless, B., Salesin, D.H., Szeliski, R., 2005. Animating pictures with stochastic motion textures. *ACM Trans. Graph.*, **24**(3):853-860. [doi:10.1145/1073204.1073273]

Corpetti, T., Memin, E., Perez, P., 2002. Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(3):365-380. [doi:10.1109/34.990137]

Courty, N., Corpetti, T., 2007. Crowd motion capture. *Comput. Anim. Virt. Worlds*, **18**(4-5):361-370. [doi:10.1002/cav.199]

Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S., 2003. Dynamic textures. *Int. J. Comput. Vis.*, **51**(2):91-109. [doi:10.1023/A:1021669406132]

Freeman, W.T., Adelson, E.H., Heeger, D.J., 1991. Motion without Movement. SIGGRAPH, p.27-30. [doi:10.1145/122718.122721]

Heeger, D.J., Bergen, J.R., 1995. Pyramid-Based Texture Analysis/Synthesis. SIGGRAPH, p.229-238. [doi:10.1145/218380.218446]

Igarashi, T., Moscovich, T., Hughes, J.F., 2005. As-Rigid-as-Possible Shape Manipulation. SIGGRAPH, p.1134-1141. [doi:10.1145/1186822.1073323]

Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A., 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, **22**(3):277-286. [doi:10.1145/882262.882264]

Levin, A., Lischinski, D., Weiss, Y., 2008. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**(2):228-242. [doi:10.1109/TPAMI.2007.1177]

Lin, Z., Wang, L., Wang, Y., Kang, S.B., Fang, T., 2007. High resolution animated scenes from stills. *IEEE Trans. Visual. Comput. Graph.*, **13**(3):562-568. [doi:10.1109/TVCG.2007.1005]

Litwinowicz, P., Williams, L., 1994. Animating Images with Drawings. SIGGRAPH, p.409-412. [doi:10.1145/192161.192270]

Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T., 2008. SIFT flow: dense correspondence across different scenes. *LNCS*, **5304**:28-42. [doi:10.1007/978-3-540-88690-7_3]

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, **60**(2):91-110. [doi:10.1023/B:VISI.0000029664.99615.94]

Lucas, B.D., Kanade, T., 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. Int. Joint Conf. on Artificial Intelligence, p.674-679.

Okabe, M., Anjyo, K., Igarashi, T., Seidel, H.P., 2009. Animating pictures of fluid using video examples. *Comput. Graph. Forum*, **28**(2):677-686. [doi:10.1111/j.1467-8659.2009.01408.x]

Okabe, M., Anjyo, K., Onai, R., 2011. Creating fluid animation from a single image using video database. *Comput. Graph. Forum*, **30**(7):1973-1982. [doi:10.1111/j.1467-8659.2011.02062.x]

Rubinstein, M., Shamir, A., Avidan, S., 2008. Improved seam carving for video retargeting. *ACM Trans. Graph.*, **27**(3), Article No. 16, p.1-9. [doi:10.1145/1360612.1360615]

Schodl, A., Szeliski, R., Salesin, D.H., Essa, I., 2000. Video Textures. SIGGRAPH, p.489-498. [doi:10.1145/344779.345012]

Shinya, M., Aoki, M., Tsutsuguchi, K., Kotani, N., 1999. Dynamic Texture: Physically Based 2D Animation. SIGGRAPH, p.239. [doi:10.1145/311625.312130]

Sun, M., Jepson, A.D., Fiume, E., 2003. Video Input Driven Animation (VIDA). Proc. 9th IEEE Int. Conf. on Computer Vision, p.96-103. [doi:10.1109/ICCV.2003.1238319]

Treuille, A., McNamara, A., Popovc, C., Stam, J., 2003. Keyframe control of smoke simulations. *ACM Trans. Graph.*, **22**(3):716-723. [doi:10.1145/882262.882337]

Wang, Y., Zhu, S.C., 2003. Modeling Textured Motion: Particle, Wave and Sketch. Proc. 9th IEEE Int. Conf. on Computer Vision, p.213-220. [doi:10.1109/ICCV.2003.1238343]