Robust Stroke-based Video Animation via Layered Motion and Correspondence

Tao Lin
Sun Yat-Sen University
Guangzhou, China 510006
lintao30@gmail.com

Liang Lin*
Sun Yat-Sen University &
Lotus Hill Research Institute
China 510006
linliang@ieee.org

Qing Wang
Sun Yat-Sen University
Guangzhou, China 510006
wangq79@mail.sysu.edu.cn

ABSTRACT

This paper investigates a novel approach to reduce artifacts and visual flickering in generating painterly animations from real video clips. In the traditional painterly animation methods, the brush strokes are propagated over video frames by calculating optical flows, and the visual impression of animations are severely affected by incorrect correspondences. In our method, we combine motion segmentation and occlusion handing to establish accurate dense feature correspondences, which is shown to robust propagate brush strokes against complex motions and occlusions. Moreover, a beforehand rendering strategy is presented to alleviate stroke flickering. In the experiments, we generate a number of animations in cartoon and oil painting style. The quantitative evaluations of brush stabilization is presented as well.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation; J.5 [**Computer Applications**]: Arts and Humanities

General Terms

Algorithms, Design, Experimentation

Keywords

video stylization, non-photorealistic rendering, layered motion

1. INTRODUCTION

Painterly animation is one of the major topics in Graphics and Multimedia technology, whose goal is to transform a real video into artistic styles with few user interactions. The methods of painterly animation basically include two key components: (i) stroke-based rendering for images; (ii) stroke propagating in video frames[8].

Several recent methods of stroke-based rendering (SBR) achieved impressive results, which design diverse of brush strokes that integrate various attributes related to rendering effects such as shape, color, length, opacity etc. For example, Hertzmann [4] proposed curved strokes with multiple sizes for rendering oil painting style. Zeng et al. [13] proposed a semantic-driven rendering system using example-based brushes.

However, it is still an open problem to render strokes in videos. The major challenges lie in maintaining the spatial-temporal coherences of painterly rendering in videos. To avoid visually cluttered frame-by-frame rendering, most of the systems propagate brush strokes from the key frames into other sequential frames by establishing correspondences among video frames. In literature, Hays et al. [3] arranged brush strokes into layers and calculated optical flows to propagate brush strokes. Lin et al. [7] partitioned a video sequence into multiple semantic regions and propagated strokes using sparse SIFT features. Despite acknowledged results, the artifacts and visual flickering are still unavoidable, particular for complex videos including non-rigid (3D) motions and sever occlusions in videos. The main problems are: the inaccuracy in dense correspondences and the lack of occlusion handling.

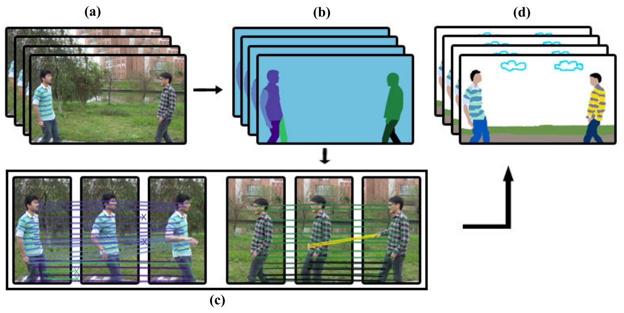

Figure 1: The framework of our approach. (a) Original frames in our experimental video walk. (b) Automatic motion Segmentation, different color for different layer. (c) Layered dense feature correspondence. Frame 1, Frame 6 and Frame 12 are shown. The yellow correspondence from frame 6 to frame 12 means correspondence in newly birth area (the moving hand). (d) Propagate brush strokes according to layered dense feature correspondence.

*Corresponding author is Liang Lin. This work was supported by the Hi-Tech Research and Development Program of China (Grant No. 2012AA011504), the National Natural Science Foundation of China (Grant No.60970156 and Grant No.61173082), and the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-12KF-06).

In this paper, we propose a general approach to generate robust painterly animation via layered motion and dense correspondences. Fig.1 gives an overview of our approach. First, we divides a frame into several non-overlapping motion layers, to eliminate motion ambiguity near the juncture of different motion areas. Then the dense feature correspondences are extracted inside each layer with an procedure of explicit occlusion handling. At last, a beforehand rendering strategy is presented to alleviate flickering in stroke rendering.

Our approach consists of four phases. (1) **Motion Segmentation.** Automatically partition a video scene into muliple motion layers. (2) **Layered Dense Feature Correspondence.** Establish dense feature correspondence in each motion layer with occlusion handling. (3) **Image Rendering.** Transform a single frame into a painterly image by an arbitrary SBR algorithm. (4) **Video Rendering.** Warp brush strokes objects according to layered dense correspondence, and with two strategies to alleviate strokes flickering. One related system was introduced in [11], which extracted layered motion in an interactive manner.

2. EXTRACTING MOTION SEGMENTATION

We adopt and modify the method described at [10] with two steps. (i) extract motion layer models in each frame; (ii) establish correspondence of motion layer models among frames.

2.1 LAYERED MOTION MODELS

To extract motion layer models, we first obtain an initial estimate of motion layer model by computing image motion between each pair of consecutive frames. Then we refine the initial estimate by the $\alpha\beta$-swap algorithm [1].

We use SIFT flow [9] to compute image motion between consecutive frames. Then we cluster points moving rigidly together to obtain rigid components. These rigid components make up an initial estimate of the motion layer model for each frame.

However, with rough layer boundaries, motion layer models computed by SIFT flow are inaccurate. Hence we have to refine these models. The refining of motion layer models can be formulated as a multi-label problem, which is going to Maximum A Posterior (MAP) estimation in a Markov Random Fields (MRF) model. The optimal labeling l on the frame t can be obtained by minimizing EQ (1):

$$E_t(l) = \sum_{p\in P} A_t(l_p) + \sum_{q\in N(p)} (\lambda_1 B_t(l_p,l_q) + \lambda_2 P_t(l_p,l_q)) \quad (1)$$

where p is a pixel in an image and N(p) is four neighbors of p. Let $I_t(p)$ be the RGB value of p in the frame t. $HC^t_{l_p}$ is the RGB histogram of the region with label l_p. $M_t(p)$ is the motion of p from the frame t to t+1 and $HM^t_{l_p}$ is the motion histogram of the region with label l_p in the frame t. The appearance term is given by

$$A_t(l_p) = -\omega_1 lnPr(I_t(p)|HC^t_{l_p}) - \omega_2 lnPr(M_t(p)|HM^t_{l_p}) \quad (2)$$

EQ (2) enforces texture consistency and motion consistency by the weights ω_1 and ω_2 respectively.

The contrast term B_t pushes the boundary of a motion layer to image edges.

$$B_t(l_p,l_q) = \begin{cases} \arctan(|I_t(p) - I_t(q)|^2 - \sigma) + \frac{\pi}{2}; & \text{if } l_p \neq l_q \\ 0; & \text{otherwise} \end{cases} \quad (3)$$

When $l_p \neq l_q$, $B_t(l_p,l_q)$ is a Heaviside function which provides a sharp change around $\sqrt{\sigma}$.

The prior term is specified by an Ising model which encourages spatial continuity.

$$P_t(l_p,l_q) = \begin{cases} T; \text{ if } l_p \neq l_q \\ 0; \text{ otherwise} \end{cases} \quad (4)$$

EQ (1) can be minimized by the $\alpha\beta$-swap algorithm, and hence we get piecewise smooth motion layer models. In the computation, our approach also encourages gradual swapping pixels along the juncture of different layers to avoid wrongly swapping pixels around the juncture with pixels far away from it.

2.2 LAYERED CORRESPONDENCE

Once the motion layer model of each frame is extracted, we need to establish the correspondences among motion layers. Suppose the frame t has n_t the motion layer denoted by $X_t = \{x^t_1, x^t_2, ..., x^t_{n_t}\}$. We define the similarity of two motion layers from the frame t to t+1 as

$$D(x^t_n, x^{t+1}_m) = \begin{cases} KL(x^t_n||x^{t+1}_m) + E_o(x^t_n, x^{t+1}_m); & \text{if } x^t_n \to x^{t+1}_m \\ \kappa; & \text{otherwise} \end{cases} \quad (5)$$

where $n \in [1, n_t]$, $m \in [1, n_{t+1}]$ and $KL(x^t_n||x^{t+1}_m)$ is the Kullback-Leibler divergence. Inspired by [6], we model the layer appearance by computing the histogram of oriented gradients in the RGB color space, and measure histogram variations by Kullback-Leibler divergence.

$E_o(x^t_n, x^{t+1}_m)$ measures the motion consistency from x^t_n to x^{t+1}_m. We warp x^t_n to $x^t_n{}'$ with the image motion from the frame t to t+1, and the $E_o(x^t_n, x^{t+1}_m)$ is computed as sum of standard deviation between $x^t_n{}'$ and x^{t+1}_m. To handle occlusions, we assign penalty κ to $D(x^t_n, x^{t+1}_m)$ if layer x_n is occluded in the frame t+1.

Determining the correspondence of layer from the frame t to t+1 now becomes finding an optimal mapping $\phi_t : X \to Y \cup \emptyset$ where the global matching cost ϕ^*_t is minimal.

$$\phi^*_t = arg \min_{\phi_t} \sum_{x^t_n \in X_t, x^{t+1}_m \in X_{t+1}} D(x^t_n, x^{t+1}_m)$$

ϕ^*_t can be interpreted as the optimal assignment problem on a bipartite graph and solved with Kuhn-Munkres algorithm.

3. DENSE FEATURE CORRESPONDENCE WITH OCCLUSION HANDLING

Once motion layers are obtained, our approach establishes dense feature correspondences inside each layer with occlusion handling.

In order to track robust and drive the strokes stably, We adopt SIFT descriptor for its invariance of transformation, illumination, scale and viewing angle along video frames.

Liu et al. [9] proposed SIFT flow for matching SIFT descriptors between two images. Let S_A, S_B be SIFT descriptors in the image A and B respectively. The matching pattern of SIFT flow is $S_A \to S_B$. SIFT flow is an injection, which means each SIFT descriptor in an image A will match to a descriptor in the image B next to it. Unfortunately, this matching pattern can't handle occlusions since not all descriptors in the image A have matched descriptors in the image B when an occlusion occurs. To cope with occlusions, the matching pattern should be altered to $S_A \rightarrow S_B \cup \emptyset$.

Inspired by the graph editing method in [5], we introduce an addition label Ø into SIFT flow to handle occlusions. We modify the objective function in SIFT flow to compute the dense feature correspondence between consecutive frames with occlusion handling.

Let f be the labeling for all the pixels. p, q are the pixels in an image, and f_p, f_q are labels at p, q. The value for f_p can be exact offset from p, or Ø, which indicates the descriptor at pixel p is occluded. The objective function of SIFT flow with occlusion handling is defined as:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{q \in N(p)} V(f_p, f_q) \tag{6}$$

The data term is defined as:

$$D_p(f_p) = \begin{cases} |S_1(p) - S_2(p + f_p)| + |f_p|; & f_p \neq \emptyset \\ \alpha_D; & f_p = \emptyset \end{cases} \tag{7}$$

where S_1 and S_2 are two SIFT descriptors to match. α_D is the cost for assigning label Ø to p. EQ (7) constrains not only SIFT descriptors to match along with offset f_p, but also f_p to be as small as possible.

The smoothness term is defined as:

$$V(f_p, f_q) = \begin{cases} min(\alpha|f_p - f_q|, d); & f_p \neq \emptyset \text{and } f_q \neq \emptyset \\ \alpha_V; & \text{otherwise} \end{cases} \tag{8}$$

α_V constrains the spatial continuity of occluded area. EQ (8) constrains offset of within a neighborhood system to be similar, and uses truncated L1 norm to handle discontinuities, with d as threshold.

Both α_D and α_V are modeled as configurable constants. It can be configured according to the different type and complexity of a motion layer. In our experiments, we configured different α_D and α_V for each motion layer to get a reasonable dense correspondence.

To reduce the computational complexity, we optimize the objective function by loopy belief propagation with optimizations described in [2].

4. STROKE-BASED VIDEO RENDERING

In the generation of video painterly animations, brush strokes must be generated and placed on a video scene, and then propagate to the rest of frames. We model brush stroke as an object with following properties: color, center, radius, length, orientation and opacity, as shown in Fig.2. Each brush stroke orients based on local gradient. To enhance the painting result near image edges, brush strokes are not allowed to cross edges of motion layers. And to emulate the coarse to fine painting process, the image is painted with multiple stroke sizes as described in [4].

Once our approach obtains a set of brush strokes, it propagates strokes and then generates video animation by applying layered dense correspondence. Instead of using key frame based rendering, our approach only renders the first frame and then propagates strokes to the rest, which avoids visual flickering among key frames.

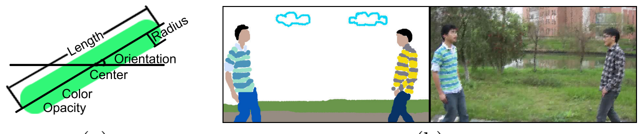

Figure 2: Brush stroke object illustration. (a) Brush stroke properties. (b) Our cartoon style created by placing strokes manually (left) and simple SBR algorithm for oil painting style (right).

Algorithm 1 Stroke-based video rendering procedure.

```
1: Extract motion layers from a video sequence.
2: Establish dense feature correspondence inside each motion layer.
3: Generate and place brush strokes on the first frame of video sequence by an arbitrary SBR algorithm.
4: Apply layered dense correspondences to brush strokes and then propagate them to the next frame.
5: Handling shrinking and expanding regions.
6: Repeat step 4 ∽ 5 until all frames are processed.
```

When propagating strokes, some regions may shrink smaller along frames while others may expand larger. Our approach handles these changes with two strategies, fading strokes in shrinking regions and beforehand rendering strokes in expanding regions.

When a region shrinks smaller, strokes in that region will be pushed together and then cover each other, which leads to a visual clutter. Our approach resolves this problem by fading these overlapped strokes, that is, decreasing their opacity gradually [3].

When a region expands larger, new strokes should be added to depict additional areas, otherwise these areas will be blank. Our approach uses beforehand rendering to cope with this problem. If a region expands from the frame t to t+k, our approach first renders the frame t+k and then uses it as the canvas for all frames from t to t+k. During the region's expansion, strokes are propagated normally. That means old strokes in expanded areas retreat, and thus expose the canvas underneath, with new strokes depicting these areas properly, as shown in Fig.3. An overview of our approach is shown in Algorithm.1.

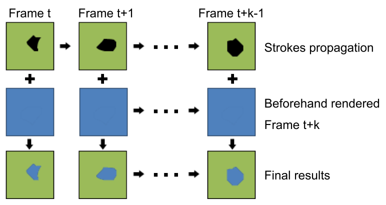

Figure 3: Illustration of beforehand rendering. (1) beforehand render the frame $t+k$ and treat it as the canvas. (2) Strokes propagation over the canvas as normal, the black areas are newly birth areas. (3) Expose the canvas through expanding areas (in blue).

5. EXPERIMENTAL RESULTS

In experiments, we tested our approach against optical flow and standard SIFT flow by comparing their temporal coherence. We tracked positions of brush strokes along

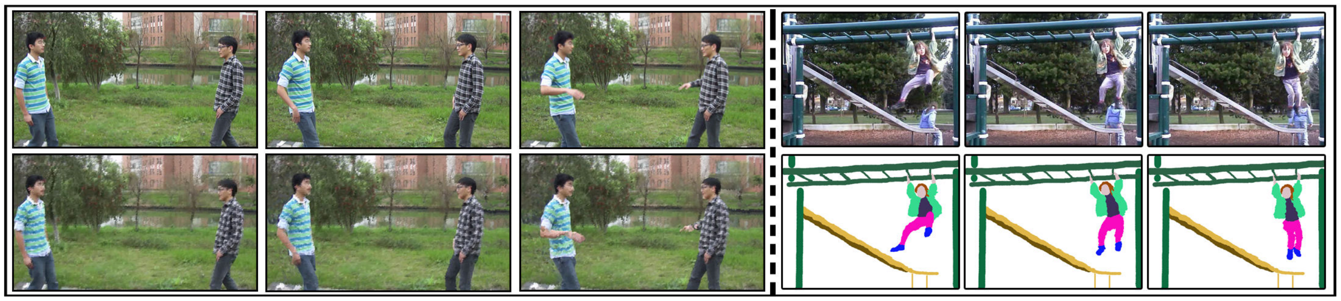

Figure 4: A few sample frames from video animations generated by our approach. The upper left row is the source frames from our walk video clip walk. The bottom left shows the results produced by our simple SBR algorithm. The upper right row is frames from video clip lena. The bottom right row is our video animation for video lena by placing and adding strokes manually.

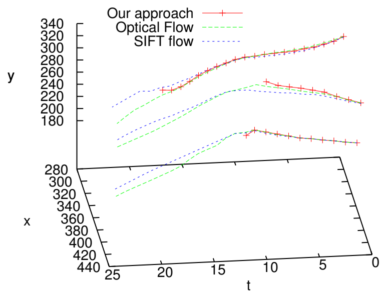

Figure 5: Trajectories of brush strokes in three areas. Without occlusion handling, brush strokes in other methods will continue even their underneath areas are occluded. Our approach effectively marks occluded areas and removes strokes.

frames in each method, and then compared their trajectories. For optical flow, we simply round the displacement to integer. The result demonstrates that our approach is more stable in complicate motion, as shown in Fig.5. It shows the advantage of our approach in maintaining temporal coherence. In our video clip walk, when three typical areas were occluded from the frame 75 to 100, our approach explicit marks them as occluded and removes them from stroke queue.

We also applied our approach to generate video animations from two video clips, walk and lena[12]. These video clips include non-rigid motions, occlusions, multiple moving objects and camera motions. In experiments, our system generated oil painting style and cartoon style. For oil painting style, we develop a simple SBR algorithm which simply generates strokes on the image grid with same length and width. We applied this simple SBR algorithm to both the first frame and beforehand rendering. Then our system automatically propagated strokes to the rest of frames. For cartoon style, we manually placed cartoon style strokes on the first frame of each clip, and then generated cartoon style animations by our approach. Fig.4 shows some frames in the results.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we present a novel approach to transform a video clip into a painterly animation via robust layered dense correspondence. We use motion segmentation and occlusion handling to enhance temporal coherence, which makes our approach robust and capable of stylizing videos with complicate motions. Furthermore, our approach is very general and can work with diverse stroke-based rendering methods.

In the future, we plan to further enhance temporal coherence by improving tracking in textureless area.

7. REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001.

[2] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, pages I–261 – I–268 Vol.1, 2004.

[3] J. Hays and I. Essa. Image and video based painterly animation. In *NPAR '04*, pages 113–120, 2004.

[4] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98*, pages 453–460, 1998.

[5] L. Lin, X. Liu, and S.-C. Zhu. Layered graph matching with composite cluster sampling. *TPAMI*, pages 1426–1442, 2010.

[6] L. Lin, P. Luo, X. Chen, and K. Zeng. Representing and recognizing objects with massive local image patches. *Pattern Recogn.*, pages 231–240, 2012.

[7] L. Lin, K. Zeng, H. Lv, Y. Wang, Y. Xu, and S.-C. Zhu. Painterly animation using video semantics and feature correspondence. In *NPAR '10*, pages 73–80, 2010.

[8] L. Lin, K. Zeng, Y. Wang, Y. Xu, and S.-C. Zhu. Painterly animation using video semantics and feature correspondence. In *TCSVT*, 2012.

[9] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, pages 978–994, 2011.

[10] M. Pawan Kumar, P. H. Torr, and A. Zisserman. Learning layered motion segmentations of video. *IJCV*, pages 301–319, 2008.

[11] L. Tao, J. Bo, and L. Liang. Painterly animation by propagating strokes via layered dense correspondence. In *ICIP*, 2012.

[12] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen. Video tooning. In *SIGGRAPH '04*, pages 574–583, 2004.

[13] K. Zeng, M. Zhao, C. Xiong, and S.-C. Zhu. From image parsing to painterly rendering. *ACM Trans. Graph.*, pages 2:1–2:11, 2009.