Robust and Fast Motion Estimation for Video Completion

Shaodi You[†] Robby T. Tan[‡] Rei Kawakami[†] Katsushi Ikeuchi[†]

[†]The University of Tokyo
{yousd, rei, ki}@cvl.iis.u-tokyo.ac.jp

[‡]Utrecht University
R.T.Tan@uu.nl

Abstract

A motion estimation method for completing a video with large and consecutive damage is introduced. It is principally based on sparse matching and interpolation. First, SIFT, which is robust to arbitrary motion, is used to efficiently obtain sparse correspondences in neighboring frames. To ensure these correspondences are uniformly distributed across the image, a fast dense point sampling method is applied. Then, a dense motion field is generated by interpolating the correspondences. An efficient weighted explicit polynomial fitting method is proposed to achieve spatially and temporally coherent interpolation. In the experiment, quantitative measurements were conducted to show the robustness and effectiveness of the proposed method.

1 Introduction

Video completion repairs damaged or undesired regions by filling them with the most suitable data, and thus makes the whole video visually as realistic as possible. The damaged regions can be caused by watermarks, logos, mud, undesired objects, raindrops adhered to the lens, etc, which possibly occupy large space and appear in a few consecutive frames. Completing these regions is challenging, since properly interpolating large damaged regions spatially and temporally is rather problematic.

Methods based on the motion field is usually used to solve the completion problem. They assume the target objects or regions to be removed are either static or moving smoothly in consecutive frames. If the motion trajectory can be correctly modeled, they can fill in the damaged regions by copying the pixels along its trajectory. However, in real videos, the whole environment motion can be arbitrary and complex. It forces them to focus on modeling the specific motion of the target regions in specific perspective and to have strong constraints to simplify the environment motion. Zhang *et al.*[12] and Jia *et al.*[3] limit the background motion to be translation only. Jia *et al.*[4] and Patwardhan *et al.*[8] assume the background to be static. Shiratori *et al.*[9] and Liu *et al.*[6] uses an existing optical flow method [1] to calculate the motion. Moreover, the accuracy of optical flow calculated from damaged videos poses another problem, since the existing methods of optical flow assume the input video does not contain any damaged regions.

Instead of modeling specific object motion, in this paper, we focus on modeling more general environment motion. We propose a method that utilizes sparse matching and interpolation to estimate the environment motion. First, we employ SIFT [7], which is robust to arbitrary motion, to find sparse correspondences in neighboring frames. We remove the pixel correspondences in the damaged regions, and thus avoid their influences. We adopt a fast dense point sampling method to ensure the correspondence is uniformly distributed. Then, we generate a dense motion field by interpolating the sparse correspondences. To achieve spatially and temporally coherent interpolation, we propose a weighted explicit 2D polynomial fitting method. Unlike 3D polynomial fitting, the proposed 2D fitting has significantly efficient computational time. Finally, we finish the video completion by copying the correspondences indicated by the motion trajectory.

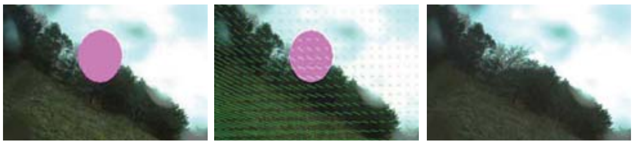

(a) Damaged video (b) Motion estimation (c) Completed video

Figure 1. Video completion using the proposed method. (a) Input video with large and consecutive damage. (b) Motion estimation using the proposed method. (c) Video completion using the motion.

The proposed method is generally applicable to spatially and temporally smooth motion, and is robust to handle a severely damaged video. In our experiment, it also achieved high computational efficiency which was 7 times faster than the optical flow based methods. Fig. 1 shows the result of the proposed method.

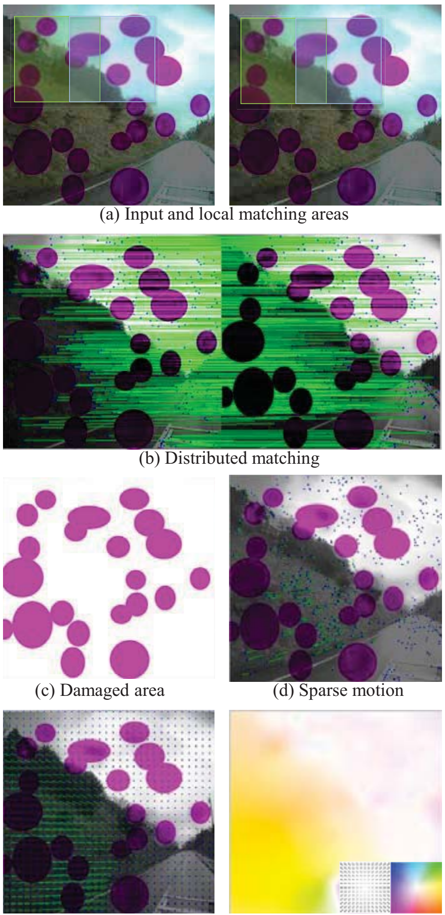
The rest of the paper is organized as follows. Section 2 describes the sparse matching method. Section 3 explains the interpolation and completion method. Section 4 shows quantitative experiments and applications in motion estimation and video completion. Section 5 concludes the paper.

2 Robust Sparse Matching

Sparse Matching In video, the appearance of an object can continuously change in terms of scale, position, direction and perspective. [12, 4, 3, 8] make constraints to simplify the motion estimation. Although [1, 5] are generally applicable to arbitrary motion, like the methods by [9, 6], they suffer from the presence of damaged regions (or the regions of undesired objects).

In the proposed method, first, the SIFT-based sparse matching is used to overcome the changes of appearance. To some extent, SIFT keypoints are invariant to scale, position, rotation, and perspective transformation [7]. Second, in sparse pixel correspondences, one pixel correspondence can be assumed to be independent from the other correspondences. Therefore, deleting the correspondences that represent the damaged regions does not influence the correspondences of non-damaged regions.

Well Distributed Correspondences The proposed method uses sparse correspondence as anchor points for motion interpolation. It requires that the sparse correspondences are distributed uniformly



(a) Input and local matching areas

(b) Distributed matching

(c) Damaged area

(d) Sparse motion

(e) Dense motion

Figure 2. The proposed motion estimation method. (a) For 2 consecutive frames, sparse matching is performed in each corresponding squared windows. (b) The sparse matching results, which are well distributed across the image. (c) The damaged area. (d) The motion (green needles), which is calculated by the matching. Motion correspondences in the damaged regions are removed. (e) The interpolated dense motion using the proposed weighted polynomial fitting. Left image: represented by needles. Right image: represented by color.

across the images, in such a way that in any area, there exist sufficient anchor points for interpolation. However, the original SIFT algorithm tends to find correspondences in highly textured regions and to ignore others. To address this problem, we modify the matching strategy of SIFT. As illustrated in Fig. 2 (a), we do not apply the SIFT matching for the whole images, but for small windows. As default, the size of each window is 80 × 80 pixels and at least 3 matching pixels are found in each pair of windows. These small windows across the whole image ensure the correspondences are well distributed. For the two neighboring windows, there are 30 pixels overlapping so that the matching pixels in the window's boundary is not neglected. This matching strategy does not influence the computational time significantly, since only half of a window is matched twice. This strategy is inspired by Tuytelaars [11].

Fig. 2 (b) shows an example of the sparse matching. Mathematically, we denote all the N matching pixels found between frame t_1 and frame t_2 as:

$$\{(x_k, y_k, x'_k, y'_k)\}_{t_1,t_2}, k = 1, 2, \cdots, N, \qquad (1)$$

where (x_k, y_k) is a pixel in frame t_1 and (x'_k, y'_k) is its correspondence in frame t_2. We apply the matching between consecutive frames. Specifically, for a given frame, the matching is found in both the previous and the subsequent 5 frames.

Correspondence to Motion Estimating the motion of corresponding pairs is straightforward. Referring to the notation in Eq. (1), for a corresponding pair (x, y) and (x', y'), the motion at (x, y) is denoted as $(\delta x, \delta y)$, which is equal to $(x' - x, y' - y)$. Specifically, we can denote all the corresponding pairs of the sparse motion between frame t_1 and t_2 as:

$$\{(x_k, y_k, \delta x_k, \delta y_k)\}_{t_1,t_2}, k = 1, 2, \cdots, N. \qquad (2)$$

Figs. 2 (c)and(d) shows an example, where the sparse motion is represented by short arrows. Erroneous matching are directly removed.

3 Fast Space-time Motion Interpolation

Explicit Polynomial Fitting Having found the sparse motion, we estimate the dense motion by doing interpolation based on 2D explicit polynomial fitting. First, we introduce the un-weighted 2D explicit polynomial fitting, where an m degree 2D explicit polynomial P_m can be expressed as:

$$P_m(x, y) = \sum_{i+j=0,1,\cdots,m} a_{ij} x^i y^j, \qquad (3)$$

with $\{a_{ij}\}$ the polynomial coefficients. We interpolate the sparse motion in the x direction and the y direction separately. The interpolation, in the x direction for example, implies finding the polynomial coefficients $\{a_{ij}\}$ that minimizes the squared sum fitting error:

$$\sum_{k=1,2,\cdots,N} |\delta x_k - P_m(x_k, y_k)|^2, \qquad (4)$$

where $\{(x_k, y_k, \delta x_k)\}_{t_1,t_2}$ are found by the sparse matching (Eq. (2)). We use the eigen-based method, which is significantly fast, to solve Eq. (4). More details about the method can be found in [10].

Temporal Coherent Weighted Fitting The fitting introduced in Eqs. (3) and (4) is temporally incoherent, since each frame is fitted independently. To make it temporally coherent, we propose an efficient weighted 2D polynomial fitting method to fit multiple frames simultaneously. Referring to the notation in Eqs. (2) and (4), the weighted fitting means to find the 2D polynomial P_m that minimizes the following error function:

$$\sum_{j=-J}^{J} \left(W(t_j - T) \sum_{k=1}^{N} |\delta x_{k,t_j} - P_m(x_{k,t_j}, y_{k,t_j})|^2 \right), \qquad (5)$$

where T is the center frame and $\{t_j\}$ are its previous and subsequent J frames. $W(\cdot)$ is a weight function which only depends on the temporal distance. $W(\cdot)$ is expressed as:

$$W(\Delta t) = \frac{1}{(\Delta t)^2} \frac{|J - \Delta t + 1|}{J}, \qquad (6)$$

where $\Delta t = t_j - T$ is the temporal distance between the corresponding pairs, $\frac{1}{(\Delta t)^2}$ is called the speed term

Table 1. Average completion error

	Fast moving area	Slowly moving area	Static area
Before completion	120.1	79.5	171.7
Criminisi 2003	23.5	34.4	**3.5**
Shiratori 2006	80.9	77.1	190.7
The proposed method	**13.2**	**20.7**	**3.5**

and $\frac{|J-\Delta t+1|}{J}$ is called the coherent term. The speed term converts the distance of the corresponding pairs to average speed. The coherent term is a pyramid function, such that high weight is given to temporally close matching pairs. Having $\boldsymbol{P}$ found from the fitting, the motion at any place (x, y) to any temporal distance Δt is calculated as:

$$(\delta x, \delta y)_{\Delta t} = \boldsymbol{P}(x, y)\Delta t = (P^x(x, y)\Delta t, P^y(x, y)\Delta t). \tag{7}$$

Considering the balance between accuracy and efficiency, as default, we set $J = 5$ and $m = 10$. Fig. 2(e) shows an example of the interpolated motion field.

Considerable efficiency can be achieved by the proposed 2D fitting. Referring to Eq. (3), the number of coefficients $\{a_{ij}\}$ to be solved is in $O(m^2)$ complexity. If we use 3D polynomial, $P_m(x, y, t) = \sum_{i+j+k=0,1,\cdots,m} a_{ijk}x^iy^jt^k$, the number of coefficients $\{a_{ijk}\}$ is $O(m^3)$, which is considerably time consuming.

Video Completion We fill in the damaged regions in the input video by utilizing the estimated motion function. For a given frame with its motion function $\boldsymbol{P}$, the damaged regions are completed in a pixel-by-pixel basis. For a given damaged pixel (x, y), its correspondence (x', y') in other frames is found by:

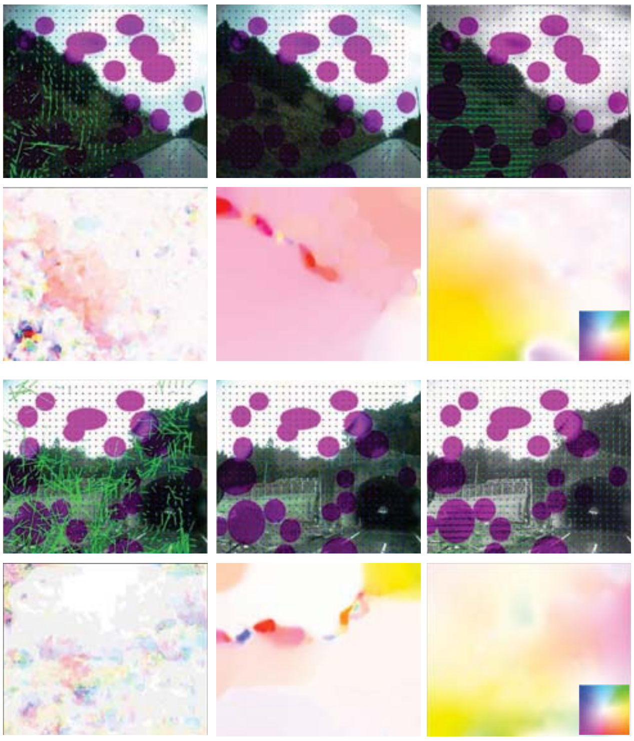
$$(x', y') = (x + P^x(x, y)\Delta t, y + P^y(x, y)\Delta t). \tag{8}$$

According to Eq. (8) we can find one correspondence in each of the neighboring frame. The spatially and temporally closest undamaged correspondence is considered to be the most coherent and thus chosen to be the best. Then, (x, y) is completed by copying the best correspondence. In the final stage, for those pixels whose correspondence is in the damaged regions, we adopt an image inpainting method [2] to complete them. Fig. 1(c) shows the result of a completed video.

4 Experiments

Robustness Real videos captured by a car-mounted camera were used to test the robustness of the proposed motion estimation method. As shown in Fig. 3, we randomly deleted one third of the frames which makes the video seriously damaged. Since the car was moving along the road, the motion of the foreground should point to the end of the road, and the nearer object should have larger motion. For comparison, two typical optical flow methods were also tested: L-K-flow [1] which is used by Shiratori *et al.*[9] and SIFT-flow [5] which is the state-of-art. As one can see, only the proposed method estimated the motion more robustly.

Accuracy To quantitatively demonstrate the accuracy of the proposed repairing method, we used a video without damage as the ground truth. For the input, three regions are deliberately deleted: (1) regions where the motion is fast, (2) regions where the motion is slow, and (3) static regions. Shiratori *et al.*[9], method of Criminisi *et al.*[2] and the proposed method were used to complete the damaged area separately. Other methods [12, 4, 3, 8] have some motion constraints, which do not hold in this video and thus were not compared. A selection of the results is shown in Fig. 4. We quantitatively compared the 8 bit $(R; G; B)$ value differences between the ground truth and the repaired video. The average errors are listed in Table. 1, where we can see the accuracy of the proposed method outperformed all the other methods.



(a) L-K flow (b) SIFT flow (c) Proposed method

Figure 3. Two experiments on robust motion estimation. Row 1 and 3: input video and motion needles. Row 2 and 4: motion visualized by color.

Table 2. Average completion time per frame

Criminisi 2003	Shiratori 2006	The proposed method
80s	145s	19s

Efficiency Under the same hardware and environment, the average time used to repair one frame (640×480) using the proposed method and the two other methods is listed in Table 2. As shown in the table, the proposed method is significantly faster. The proposed method is also generally applicable to any large and consecutive video damage, as shown in Fig. 5.

5 Conclusion

We have proposed a sparse matching and interpolation based motion estimation method for completing video with large and consecutive damage. The SIFT based matching is used to estimate the initial sparse correspondences, followed by a dense motion interpolation using a weighted 2D polynomial is applied. Lim-

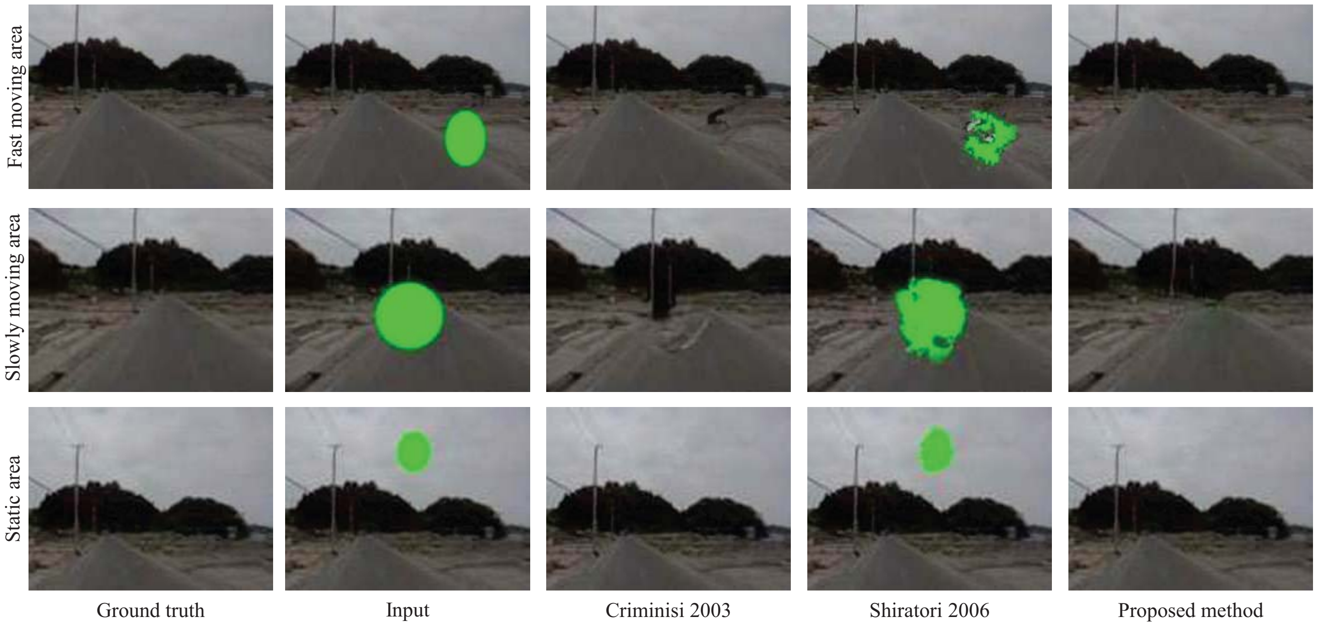


Figure 4. Accuracy comparison on video completion.

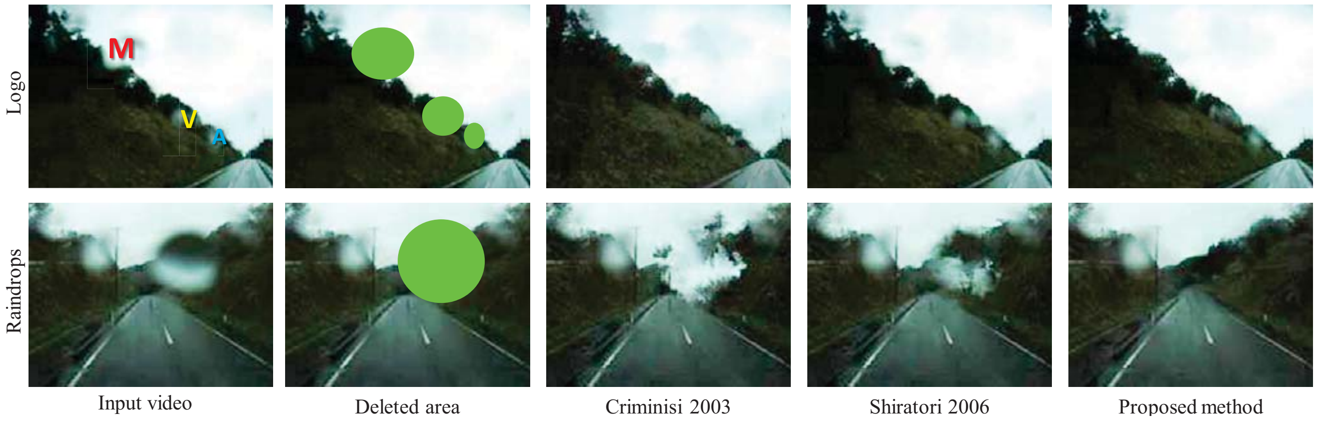


Figure 5. Applications of video completion.

itations of this method include the inaccuracy in capturing sharp and small motion, which we consider to be our future work.

Acknowledgment

This research is granted by the Japan Society for the Promotion of Science (JSPS) through the "Funding Program for Next Generation World-Leading Researchers (NEXT Program)," initiated by the Council for Science and Technology Policy (CSTP).

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision(IJCV)*, 56(3):221–255, 2004.

[2] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing(TIP)*, 13(9):1200–1212, 2004.

[3] J. Jia, Y. Tai, T. Wu, and C. Tang. Video repairing under variable illumination using cyclic motions. *IEEE Transaction on Pattern Analysis and Machine Intelligence(TPAMI)*, 28(5):832–839, 2006.

[4] J. Jia, T. Wu, Y. Tai, and C. Tang. Video repairing: Inference of foreground and background under severe occlusion. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[5] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transaction on Pattern Analysis and Machine Intelligence(TPAMI)*, 33(5):978–994, 2006.

[6] M. Liu, S. Chen, J. Liu, and X. Tang. Video completion via motion guided spatial-temporal global optimization. *ACM Multimedia(ACMM)*, 2009.

[7] D. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision(ICCV)*, 1999.

[8] K. Patwardhan, G. Sapiro, and M. Bertalmio. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Image Processing(TIP)*, 16(2):545–553, 2007.

[9] T. Shiratori, Y. Matsushita, S. B. Kang, and X. Tang. Video completion by motion field transfer. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[10] N. Trefethen. *Spectral methods in MATLAB*, volume 10. Society for Industrial Mathematics, 2000.

[11] T. Tuytelaars. Dense interest points. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. *IEEE Workshops on Application of Computer Vision*, 2005.