# EXAMPLE-BASED MOTION MANIPULATION

*Pin-Ching Su, Hwann-Tzong Chen*

National Tsing Hua University

*Chia-Ming Cheng*

MediaTek

## ABSTRACT

This paper introduces an idea of imitating camera movements and shooting styles from an example video and reproducing the effects on another carelessly shot video. Our goal is to compute a series of transformations so that we can warp the input video and make the flow of the output video resemble the example flow. We formulate and solve an optimization problem to find the required transformation for each video frame. By enforcing the resemblance between flow fields, our method can recreate different shooting styles and camera movements, from simple effect of video stabilization, to more complicated ones like anti-blur panning, smooth zoom, tracking shot, and dolly zoom.

***Index Terms***— Motion analysis, video editing

## 1. INTRODUCTION

We present a flow-guided method for manipulating the shooting style and camera movement of an input video. Given the input video and its reference flow field, we compute a series of transformations to warp the input video, such that the flow of the warped video will be as similar to the reference flow as possible. The reference flow field may be derived from a real video or synthetically generated.

Finding the required transformation for each video frame is formulated as an optimization problem of minimizing the difference between the preferred reference flow field and the flow field of the target video frame. Fig. 1 illustrates an example of altering the zoom-in speed of a video. The input video contains discontinuous zoom-in shots. In this example we use a synthetic flow field that diverges from the center. This reference flow field is applied to each frame of the input video to create a smooth zoom-in effect. As can be seen, the original trajectories of feature points in the input video are ragged. The trajectories become much smoother after we perform the transformations estimated by our method with respect to the reference flow field.

The main contribution of this paper is introducing a new way of simulating shot types and camera motions for videos. We show that, by enforcing the flow field of target video to resemble the reference flow field, we may reproduce various types of camera movements and shooting styles, from the straightforward effect of video stabilization, to more complicated ones like smooth zoom, fast and slow panning, zooming and rotating, tracking shot, and dolly zoom.

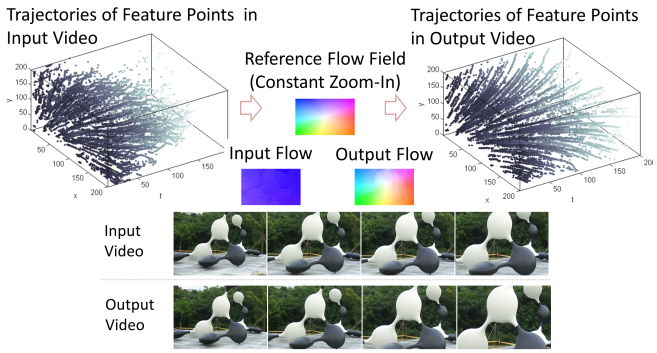


**Fig. 1**. This example illustrates the effect of changing the zoom-in speed of a video. The input video contains discontinuous zoom-in shots. We use a synthetic reference flow field that diverges outwards from the center. This reference flow field is applied to each frame of the input video to create a smooth zoom-in effect. As can be seen, the original trajectories of feature points in the input video are ragged. The trajectories become much smoother after we perform the transformations estimated by our method based on the reference flow field. A typical flow field of the input video shown here looks quite different from the reference flow field, while a typical output flow field can closely resemble the reference flow field.

### 1.1. Related Work

Flow field is useful for video editing. Interesting effects such as motion magnification can be made by modifying and interpolating the flow field locally, as shown in [1]. Dense correspondence using SIFT flow [2] can also be used to perform local warping for motion synthesis. Shiratori et al. [3] present the idea of transferring flow field for video completion. In this work, instead of editing the flow field directly, we seek to solve the problem by finding a sequence of global transformations to warp the input video, such that the flow field of the warped video is similar to the reference flow field.

Our work is related to video stabilization in the sense of adjusting the camera motion. Video stabilization aims at

removing annoying shaky motion from video. Recent research on video stabilization has shown impressive progress [4], [5] [6], [7], [8], [9]. The success of the state-of-the-art approaches is mainly due to the use of feature tracking to associate the geometry relationships across views. From an input shaky video sequence, a smoothed camera path is reconstructed by either 2D transformation or 3D structure from motion. Different methods for reconstructing smooth camera paths have been proposed, such as smoothing the space-time trajectories [5], optimizing the path with intended motion models [4], imposing subspace constraints on feature trajectories [7], and specifying a desired 3D camera path [6]. After the new camera path is estimated, content-preserving view synthesis can be performed to generate the result sequence. Our approach can be easily applied to video stabilization. We may either choose an example video that is stable or directly create a static flow field as the reference. Nevertheless, our approach is not restricted to the application of video stabilization. We point out in the experiments several interesting filming styles that cannot be achieved by video stabilization.

Our work shares a similar goal with the idea of *re-cinematography* proposed by Gleicher and Liu [10], in the aspect of making professional looking videos that involve appropriate camera movements. Gleisher and Liu introduce a local mosaic approach that creates mosaic images from a series of shorter segments of the original video. Virtual camera motions are then applied to the mosaic images so that the resulting video may look more like professional videos. The re-cinematography technique adopts several rules of transforming camera movements, including *i*) replacing small movements with a static camera, *ii*) modifying larger movements to follow directed paths and move with a constant velocity, and *iii*) using motion saliency to determine what is likely to be important to the viewer such that the subjects are properly framed. Our work differs from the work of Gleicher and Liu in that we do not include any specific rules of camera motions in our method. Camera motions are flexibly implied by the given reference flow, which can cover a wider range of shooting styles and camera movements that cannot be characterized by simple rules.

## 2. FORMULATION

Given an input video sequence $\{I_t\}$ and a reference flow $\{\mathbf{f}_t\}$, we aim to find a series of deformable transformations $\{D_t\}$ to warp the input video, so that the flow of the warped video will be similar to the reference flow. The objective function of the problem can be formulated as minimizing the difference between the intended and the reference flow fields

$$\underset{\{D_t\}}{\text{minimize}} \sum_t \|\mathbf{v}_t - \mathbf{f}_t\|^2 , \tag{1}$$

where $\mathbf{f}_t$ denotes a snapshot of the reference flow at time $t$, $\mathbf{v}_t$ indicates the flow field of the output video frame $I'_t$ after applying the deformable transformation $D_t$. The task is to find $\{D_t\}$ that warp the input video frames $\{I_t\}$ to $\{I'_t\}$, such that the difference between the flows $\{\mathbf{v}_t\}$ and $\{\mathbf{f}_t\}$ can be minimized.

One of the main challenges of solving this problem is the computational cost of estimating *i*) the deformable transformation $\{D_t\}$ and *ii*) the flow field $\{\mathbf{v}_t\}$ of the warped video. We reformulate the objective function and propose an efficient algorithm to tackle the problem. We employ two techniques to avoid the computation of flow field: One is the use of sparse feature points; the other is the constrained model of homography transformation, which imposes an implicit constraint with respect to the reduced degree of freedom on the solution space.

## 3. ALGORITHM

To begin with, we apply SIFT matching [11] to the input sequence. Let $\mathcal{U}_{t-1,t} = \{U_{t-1}, U_t\}$ be the set of the 2D homogeneous coordinates of matched feature points between $I_{t-1}$ and $I_t$. The reformulated objective function of minimizing the flow field over the sparse matched features can be written as

$$\underset{\{H_t\}}{\text{minimize}} \sum_t \| H_t U_t - H_{t-1} U_{t-1} - \mathbf{f}_{t-1}(H_{t-1} U_{t-1}) \|^2 , \tag{2}$$

where the set $\{H_t\}$ contains the series of homography matrices. By comparing the general formulation (1) and the new formulation (2), we can see that, $\mathbf{v}_{t-1}$ is replaced by $H_t U_t - H_{t-1} U_{t-1}$, defined only on feature points. The reference flows are also evaluated only on feature points. Another way to interpret the optimization is to rewrite the error term as $\|H_t U_t - (H_{t-1} U_{t-1} + \mathbf{f}_{t-1}(H_{t-1} U_{t-1}))\|^2$, and $H_{t-1} U_{t-1} + \mathbf{f}_{t-1}(H_{t-1} U_{t-1})$ can be viewed as the expected positions of feature points in the next frame according to the reference flow.

### 3.1. Sequential Approximate Optimization

In our implementation, we approximately solve the optimization of (2) in a sequential optimization manner. That is, we solve for $H_t$ one by one instead of solving for the whole set $\{H_t\}$ simultaneously. Sequential approximate optimization is fast but greedy. Moreover, the sequential optimization process might accumulate errors and thus causes the 'drifting' problem in some cases. At the initial step, we set $H_1$ as the 3-by-3 identity matrix, and assign the cropped region of $I_1$ to the output frame $I'_1$. Then, at time step $t$, we first apply SIFT matching between input frames $I_{t-1}$ and $I_t$ to obtain matches $\{U_{t-1}, U_t\}$. We then calculate the updated coordinates $\hat{U}_t$ of $U_{t-1}$ according to the previously estimated homography matrix $H_{t-1}$ plus the reference flow. As a result, we get

$$\hat{U}_t = H_{t-1} U_{t-1} + \mathbf{f}_{t-1}(H_{t-1} U_{t-1}) . \tag{3}$$

The current homography matrix $H_t$ can be solved by minimizing the distance between the corresponding points, that is,

$$H_t = \arg\min_H \|HU_t - \hat{U}_t\|^2 . \quad (4)$$

Finally, we perform view synthesis with respect to the estimated homography matrix $H_t$ using spatial domain transformation and obtain the output frame $I'_t$.

Several robustness issues should be taken into consideration. First, the false matches are inevitable in the feature matching process. Second, the updated coordinates plus the reference flow might violate the geometric constraint between $I_t$ and $I'_t$, and the abnormal points in $\hat{U}_t$ would cause a biased solution due to the nature of least squares. Such abnormal points may happen to lie at the positions that have noisy local motions in the reference flow field.

### 3.2. Robust Estimation

To deal with the robustness issues, we use robust estimators to reduce the influence of outliers and relative local motions. We compute the initial guess of the translation between $\{U_{t-1}, U_t\}$ by adopting their median values in both horizontal and vertical directions. A robust standard deviation $\sigma$ can be estimated using the *median absolute deviation* [12]. We then solve a robust error function $E(H)$ instead of the original least-squares error function in (4). The robust error function $E(H)$ is defined by

$$E(H) = \sum_{i=1}^{n} \min\{e_i(H), \lambda\sigma^2\} , \quad (5)$$

given that

$$e_i(H) = \|H\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 , \quad (6)$$

where $\{\mathbf{u}_i, \hat{\mathbf{u}}_i\}$ denotes the $i$th pair of corresponding feature points in $U_t$ and $\hat{U}_t$, for $i = 1, \ldots, n$. The homography matrix $H$ is optimized through minimizing the error function $E(H)$, given the aforementioned initial guess of translation. We use the Levenberg-Marquardt method to solve the nonlinear least-squares problem. The overall algorithm is summarized as follows.

- Set $H_1$ as the 3-by-3 identity matrix; Copy the first frame $I_1$ to the output frame $I'_1$.
- For each frame $I_t$:
  1. Perform feature matching between $I_{t-1}$ and $I_t$ to obtain the correspondences $\{U_{t-1}, U_t\}$;
  2. Calculate the updated coordinates $\hat{U}_t$ of the feature points $U_{t-1}$ according to the reference flow;
  3. Compute the initial guess of translation between $\{U_{t-1}, U_t\}$;
  4. Estimate the homograpy matrix $H_t$ by minimizing the robust error function $E(H)$ in (5);
  5. Perform view synthesis with respect to $H_t$ and produce the warped frame $I'_t$.
- Compile the results as the output sequence $\{I'_t\}$.

## 4. EXPERIMENTAL RESULTS

In the first part of the experiments we would like to show that our method is effective in resembling the reference flow field on the output video. In the second part we describe various applications of our method.

### 4.1. Evaluations

We perform the evaluations using the stabilization videos created by Liu et al. [7]. Note that, the main goal of the evaluations presented here is not to compare the performances of video stabilization. We simply would like to know how well our method can achieve the task of resembling the reference flow field on the output video. We use two standard metrics, the *absolute flow endpoint error* and the *angular error* [13], to measure the difference between flow fields. Fig 2 shows an evaluation result. The input sequence is a shaky video. We use the stabilization result generated in [7] as the reference. Our method can successfully transform the input video into a stabilized output video that mimics the stabilization result done by [7]. The proposed algorithm that solves the nonlinear least-squares problem achieves very low endpoint error ($\simeq 1$ pixel) and angular error ($\simeq 0.5$ degrees) at each frame. Fig 3 shows another evaluation result. Our method also achieves low average error rates in this dataset.

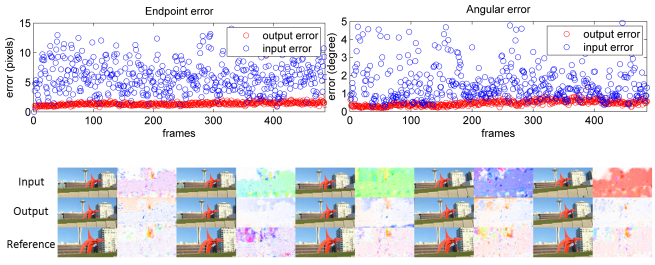


**Fig. 2**. We use *absolute flow endpoint error* (top left) and *angular error* (top right) to measure the difference between flow fields. The input sequence is a shaky video obtained from Liu et al. [7]. We use the stabilization result generated by Liu et al. as the reference. Our method can successfully transform the input video into a stabilized output video that mimics the stabilization result done by Liu et al. We also can see that, the angular error and the endpoint error are very low for the output video (red dots), in comparison with the error rates of the input video (blue dots).

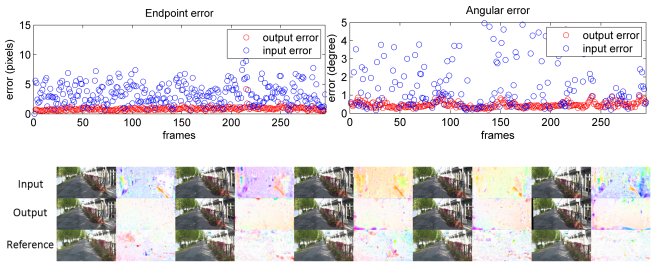


**Fig. 3**. Another evaluation result. The input sequence is also from Liu et al. [7]. The stabilization result generated by Liu et al. is the reference. The stabilized output video generated by our method successfully imitates the stabilization effect of the reference. The output error rates (red dots) are significantly and consistently lower than the input error rates (blue dots).

### 4.2. Running time

One of the advantage of our method is that it does not have to estimate optical flow on the input video. The overall computation time is $0.4$s per frame in Matlab on a $2.8$GHz quad-core PC. The frame rate is $30$fps and the frame resolution for solving the optimization is $240$p. More specifically, it takes $0.2$s to detect and compute SIFT features on each frame. To solve for the homography $H_t$ at each frame would take about $0.06$s using robust estimation and nonlinear optimization. Finally, performing the intended warping under the original $1080p$ resolution takes about $0.08$s.

### 4.3. Applications

We describe here how to use our method to generate different types of shooting styles and camera movements.

**Synthetic flow:** It is straightforward to apply a synthetic flow field to an input video if we know how to describe the intended shooting style. Given the synthetic flow fields, we can easily create the rotation effect, zoom-out effect, and static shot (zero flow) despite the discontinuous zooming-in presented in the input video. Although video stabilization is not the main goal of our work, in our experiments we also find that our method can achieve satisfactory stabilization results by simply enforcing a zero-flow field. This might be one of the easiest ways to perform video stabilization.

**Reducing fast-panning motion blur:** Fast panning of camera is likely to cause motion blur. Our method can be used to reduce such artifacts. When shooting a video, we may pan the camera slowly to take the required views. We then choose a reference flow field that presents a fast-panning effect, and apply the reference flow field to the slow-panning video. The resulting video will have a clean fast-panning effect without motion blur.

**Zooming and rotating:** This effect involves simultaneous camera rotating and zooming, which is commonly used in professional filming. Such an effect is not easy to produce if we take the video without using a tripod or camera handle. To reproduce this effect using our method, we may use an example clip taken from some professional movie that exhibits the zooming-and-rotating effect, or, we may augment our input video by adding the lacking part of zooming or rotating.

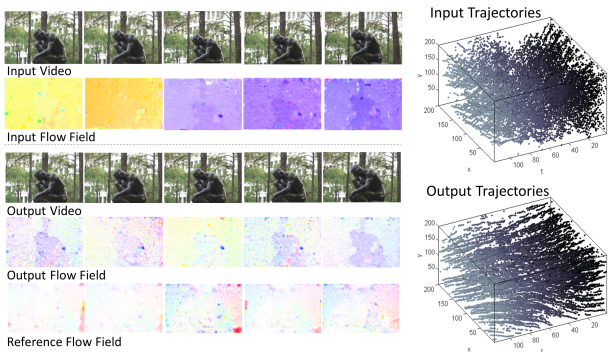


**Fig. 4**. Dolly zoom effect.

**Dolly zoom:** The dolly zoom effect is famous for its use in the movie 'Vertigo' by Alfred Hitchcock. The effect is achieved by adjusting the field of view using lens zoom while the camera moves towards or away from the subject. In such a way, the size of the subject can be kept unchanged in each frame of the video. The perspective distortion caused by dolly zoom can be used to create 'tension' between the subject and the surroundings or produce a pop-up effect. Since the size and position of the subject should be kept unchanged while we move the camera and perform zooming, it is hard to be done using a hand-held camera. Nevertheless, our method is able to produce the dolly zoom effect on the output video using a reference video that is shot with a dolly-zoom setting. The input video does not need to be taken carefully to keep the size and position of the subject fixed. Fig. 4 shows an example result, although the visual effect will be better perceived by watching the video.

## 5. CONCLUSION

The flow field is not just a function of camera motion, and that is why we do not choose to estimate the camera motion directly from the flow. Our idea is to model camera movements and shooting styles as a whole via the cues embedded in the flow. The proposed method can imitate the shooting style and the shot type that are not easy to be characterized by estimating the camera pose directly, in particular when the subject in the video is moving and the lens is zooming. We show that by maneuvering the flow field we have a relatively simpler method to achieve good results of mimicking shooting styles. The applications presented in this paper show the versatility of our method and we are seeking other possible extensions.

## 6. REFERENCES


[1] Ce Liu, Antonio Torralba, William T. Freeman, Frédo Durand, and Edward H. Adelson, "Motion magnification," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 519–526, 2005.

[2] Ce Liu, Jenny Yuen, and Antonio Torralba, "Sift flow: Dense correspondence across scenes and its applications," *PAMI*, vol. 33, no. 5, pp. 978–994, 2011.

[3] Takaaki Shiratori, Yasuyuki Matsushita, Xiaoou Tang, and Sing Bing Kang, "Video completion by motion field transfer," in *CVPR (1)*, 2006, pp. 411–418.

[4] Matthias Grundmann, Vivek Kwatra, and Irfan A. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *CVPR*, 2011, pp. 225–232.

[5] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung, "Video stabilization using robust feature trajectories," in *ICCV*, 2009, pp. 1397–1404.

[6] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala, "Content-preserving warps for 3d video stabilization," *ACM Trans. Graph.*, vol. 28, no. 3, 2009.

[7] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala, "Subspace video stabilization," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 4, 2011.

[8] Brandon M. Smith, Li Zhang, Hailin Jin, and Aseem Agarwala, "Light field video stabilization," in *ICCV*, 2009, pp. 341–348.

[9] Yu-Shuen Wang, Feng Liu, Pu-Sheng Hsu, and Tong-Yee Lee, "Spatially and temporally optimized video stabilization," in *TVCG*, 2013.

[10] Michael Gleicher and Feng Liu, "Re-cinematography: Improving the camerawork of casual video," *TOMCCAP*, vol. 5, no. 1, 2008.

[11] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[12] David C. Hoaglin, Frederick Mosteller, John W. Tukey (Editor), and John W. Tukey (Editor), "Understanding robust and exploratory data analysis," 2000.

[13] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski, "A database and evaluation methodology for optical flow," in *ICCV*, 2007, pp. 1–8.