EIE4512 Final Project 2019–Video Stabilization

The Chinese University of Hong Kong, Shenzhen

Abstract

Handling both the foreground object and the background object is challenging for video stabilization, because the foreground object and the background object have relative displacement, which makes it hard to decide the lens' shaking direction. Existing video stabilization method utilized sparse trajectories of the features to conduct trajectory smoothing followed by image deformation. However, the displacement between the original trajectory and the smooth trajectory for the foreground and background object often have different direction and magnitude. Therefore, global rigid image transformation, such as Affine transformation, fails to tackle this issue. Instead of global transformation, local transformation would be more suitable for complicated videos. My method utilizes non-rigid mesh based transformation, and spatial optimization to conduct video stabilization for each subregion of the image piecewise. The experiments show that the proposed method is able to accomplish video stabilization for complicated videos with prominent foreground object.

1 Introduction

Currently, it is popular for people to capture videos using cell phones to record their memorial moments. However, video captured using cell phones often suffered from flucturation because of shaking lens caused by the shaking human hands. Shaking lens would significantly deteriorate the quality of the video. Current solutions to stabilize a video is to use Optical Image Stabilization(OIS) or pan-tilt. However, these two methods required additional hardware, and are expensive to implement. Therefore, a zero-cost, efficient, and robust video stabilization is demanding for the users.

To conduct video stabilization, generally there are three steps–identify the videos' motion, smooth the motion, frame deformation [4–7,9]. Video motion can be roughly defined by an Affine transformation between adjacent frames. There are two types of video motion in a video, which are meaningful motion and noisy motion. The meaningful motion is attributive to the conscious-driven movement of the lens, while the noisy motion is attributive to the shaking environment in which the lens locate, which is not conscious-driven. Only the conscious-driven movement of the lens is expected for the user, therefore, it is essential to eliminate the noisy motion of the video. Smoothing methods can be utilized to eliminate the noisy motion of the video. The noisy motion of the video can be regarded as a movement generated from Gaussian distribution, while the conscious-driven motion of the video can be regarded as a low-frequency motion with significant autocorrelation. Therefore, lowpass filters can be employed to eliminate the noisy motion. After the smooth motion is obtained, the frame should be deformed according to the displacement between the original motion and the smooth motion, so that the deformed frame locates in the expected location that the noisy motion being eliminated.

The major challenge for video stabilization is that there usually both moving foreground and background in the video. For moving foreground and background, the displacement between the original motion and the smooth motion of them may have different direction and magnitude. Therefore, applying global transformation such as Affine transformation [9] would lead to under-stabilized

Paper Report of Final Project for EIE4512 - Digital Image Processing.

results, because the foreground object and the background object corresponds to two different sets of transformation's parameters, in which global transformation fails to compensate the shaking for both foreground and background. In this work, I proposed the weighted feature displacement energy and blending energy for spatial optimization to tackle this challenge. Experiments show the effectiveness of my proposed method.

1.1 Related Work

Video motion's estimation Conventional methods use registration matching to capture the frames' motion [9]. However, videos' motion estimated using registration matching yields inaccurate motion's estimation. Other studies utilize feature points' extraction and Kanade-Lucas-Tomasi Tracking Method(KLT) to track the feature points' motion. [4,6,7,9], which yield accturate motion's estimation. Moreover, Hierarchical Model-Based Motion Estimation was utilized in some studies to estimate video's motion [1,5], which firstly build Laplacian pyramid for input images, and conduct motion's estimation in a coarse-to-fine manner utilizing Brightness Constancy Equation. In this work, Harris Detector [3] and KLT algorithm are adopted to extracted features points and conduct tracking.

Trajectory smoothing To yield a smooth video, it is essential to smooth the feature points' trajectory. Most commonly used smoothing algorithm for video smoothing is bézier curve fitting [7, 9] because bézier curve fitting is able to produce curves with smooth curvature. I proposed a smoothing method combining the median filter and bézier curve fitting. I firstly smooth the trajectories using the median filter, then applied bézier curve fitting on the result of the median filter. By scaling the kernel size of the median filter, it is now able to scale the intensity of the smoothing procedure. Which means, the users can determine the intensity of the video stabilization.

Mesh-based image warping and spatial optimization In this work, mesh-based image warping [8] is utilized to deform the image, which is also the case in [7]. An image is firstly divided into quads composing a mesh. Then, optimization is conducted on the spatial mesh. A sophisticated energy term would help to yield a smooth video. In this work, I proposed two energy terms–weighted feature displacement energy and blending energy, which yields qualified image warping result for video stabilizatoin.

1.2 Contribution

In conclusion, the main contribution of this work is the proposed weighted feature displacement energy and the blending energy, which solve the problem of video stabilization for complicated videos with prominent foreground objects.

2 The Proposed Algorithm

2.1 Algorithm Overview

There are three steps in the proposed algorithm. Firstly, the Harris detector is used to extract the feature points' location, and KLT algorithm was used to track the feature points for the whole span of the video. Secondly, after the trajectories are obtained, the next step is to apply the median filter and bézier curve fitting to smooth the trajectories. Thirdly, optimization was utilized to calculate the optimal deformation mesh guiding by the weighted feature displacement energy and blending energy, followed by mesh-based image warping on every frame of the video one by one.

2.2 Tracking Trajectory

Firstly, the Harris detector was used to detect all the qualified corner points of the first frame in the video, which is denoted by $H_t = \{p_{t,1}, p_{t,2}, ..., p_{t,i}, ...\}, p_{t,i} \in \mathbb{R}^{1\times3}$, where t denotes the $t-th$ frame and the 2-D point is represented using homogeneous coordiantes, therefore it has three dimensions. For every 5 frames, the Harris detector was employed again to detect new corner point. Then, the KLT algorithm was utilized to track the all the valid corner points, and yielding a set of trajectories $T = \{\{p_1\}_{t=s_1}^{t=e_1} \{p_2\}_{t=s_2}^{t=e_2}, ..., \{p_i\}_{t=s_i}^{t=e_i}, ...\}, i = 1, 2, ..., N, T \in \mathbb{R}^{M\times N\times3}$ where M is the total numebr of frames, N is the total number of trajectory; s_i denotes the frame where the $i-th$

trajectory starts; e_i denotes the frame where the $i-th$ trajectory ends. The tracking for each corner point starts from the frame that it first appears and ends at the frame that the tracking was failed. Trajectories last less than 5 frames would be thrown away. After all the trajectories are obtained, the next step is to smooth the them.

2.3 Trajectory Smoothing

Firstly, a median filter with kernel size= 13 was used to smooth the trajectory. Then, a bézier curve fitting was employed on the result of the median filter using the following equation:

$$B(x) = \sum_{t=s_i}^{e_i} \binom{n}{t} p_{t,i} (1-x)^{n-t}, x \in [0,1] \tag{1}$$

However, the above equation cannot be used to calculate the bézier curve for a long trajectories (last longer than 500 frames), because it requires the computer to compute an extreme large number $\binom{n}{t}$, which is not plausible. Therefore, the result of the median filter is divided into segments which are short trajectories lasting only 12 frames. For smoothing propose, the first two elements of a segment is the last two elements of the prior segment. Then, the bézier curve fitting is applied on all these segments, and the results are cliped and concatenated together to form an intact trajectory having the original length. Figure 1 shows the smoothing result of one trajectory. As shown in Figure 1, the median filter significantly smoothes the trajectory. However, the result of the median filter still suffers from slight fluctuation, with non-continuous curvature. The result of the bézier curve fitting is smoother than the median filter, with continuous curvature. The smooth trajectory is denoted by $\widetilde{T} \in \mathbb{R}^{M \times N \times 3}$.

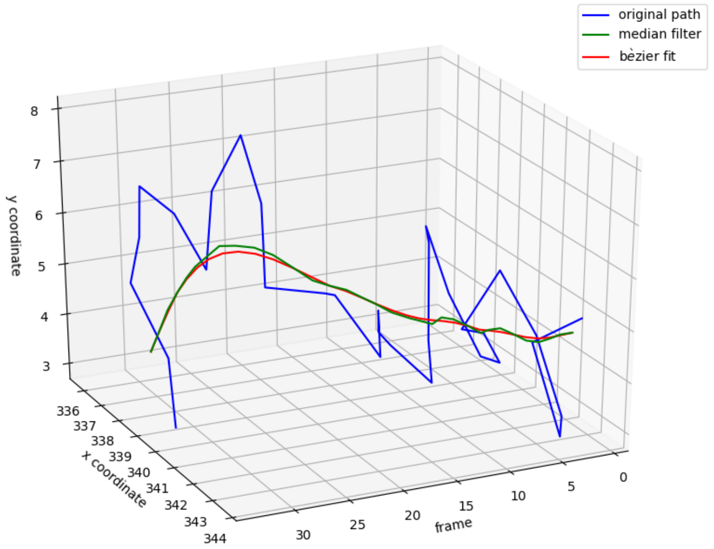


Figure 1: Smoothing result of one trajectory.

2.4 Mesh-Based Image Warping and Spatial Optimization

Mesh-based image warping is based on dividing the image uniformly into squares, and conduct rigid transformation on each square quad seperately [8]. The standard mesh dividing the image can be denoted by $\boldsymbol{M}^{W \times H \times 3}$, $\boldsymbol{M}_{i,j}$ denoted the upper left vertex of the quad in $i-th$ row and $j-th$ column, the vertexs are represented in homogeneous coordinates. For a mesh, except the vertexs, the mesh can be considered composed by quad faces, which is denoted by $\boldsymbol{Q}$ and $\boldsymbol{Q}_{i,j}$ indicates the quad in $i-th$ row and in $j-th$ column. And there are four edges in a quad, the edges of a quad is denoted by $\boldsymbol{E}(\boldsymbol{Q}_{i,j}) \in \mathbb{R}^{4\times3}$, the vertexs of a quad is denoted by $\boldsymbol{V}(\boldsymbol{Q}_{i,j}) \in \mathbb{R}^{4\times3}$.Mesh-based image warping is useful for videos with prominent foreground object because during the stabilization procedure, foreground object and background object usually should have different transformation parameters. Therefore, global rigid transformation fails, however, mesh-based image warping is suitable because each quad has its independent transformation parameters.

Spatial optimization is utilized to compute the optimal deformed mesh, which is then used in image warping. In this task, Sequential Least Square Programming [2] is utilized to find the optimal solution. There are several energy terms for the optimization, which are affine transformation energy $\mathcal{L}_{aff}$, weighted feature displacement energy $\mathcal{L}_{fea}$, and blending energy $\mathcal{L}_{blend}$.

$\mathcal{L}_{aff}$ specifies how far the deformed mesh is from the mesh generated by global affine transformation. $\mathcal{L}_{aff}$ greatly stabilize the optimization because most of the quads share similar transformation parameters. The global affine transformation is calculated using weighted least square approximation according to the original trajectories and the smooth trajectories. $\mathcal{L}_{aff}$ can be formulated in the following equations:

$$\begin{aligned}
\mathcal{L}_{aff} &= \|\hat{\boldsymbol{M}}_t - \boldsymbol{A}_t \odot \boldsymbol{M}\|^2 \\
\boldsymbol{A}_t &= \{(W_t \odot T_t)^T \odot (W_t \odot T_t)\}^{-1} \odot \{(W_t \odot T_t)^T \odot (W_t \odot \widetilde{T}_t)\} \\
W_{t,i,i} &= \begin{cases} 0, \quad if \quad t < s_i \, or \, t > e_i \\ \frac{t-s_i}{10}, \quad else\, if \quad s_i < t < e_i \, and \, t < s_i + 10 \\ \frac{e_i - t}{10}, \quad else\, if \quad s_i < t < e_i \, and \, t > e_i - 10 \\ 1, \quad else \end{cases}
\end{aligned} \tag{2}$$

In the above equations, $\odot$ indicates matrix dot product; $\boldsymbol{M}$ is the standard mesh composed of square quads while $\hat{\boldsymbol{M}}$ is the estimation of the deformed mesh composed of deformed quads. $W \in \mathbb{R}^{M \times N \times N}$ is a diagonal weight matrix in which $W_{t,i,i}$ indicates the significant weight for the point in $i - th$ trajectory at $t - th$ frame. It is essential to use the weight matrix because when new corner points are detected and being tracked, the weight of this trajectory should be slowly increased, so that the global Affine matrix will not change abruptly because of the new trajectories. Vice verse, the weight of the trajectory should be gradually decreased before the trajectory is lost. In words, the weight matrix is trying to smooth the influence brought by the new trajectories so that the global transformation matrix change smoothly.

2.5 Weighted Feature Displacement Energy and Blending Energy

The weighted feature dispacement energy $\mathcal{L}_{fea}$ specifies how far away the feature points in the deformed image from the smooth trajectory. It encourage the image warping to make the feature points close to the smooth trajectory as much as possible, so that this feature point's trajectory in the resulting video would be smooth. $\mathcal{L}_{fea}$ can be formulated in the following equation:

$$\begin{aligned}
\mathcal{L}_{fea} &= \sum_{i=1}^{N} W_{t,i,i} \|p_{t,i} - \hat{p}_{t,i}\|^2 \\
\hat{p}_{t,i} &= \sum_{k=1}^{4} \alpha_k \hat{V}_{t,k}, \; \hat{V}_t \in \boldsymbol{V}(\hat{\boldsymbol{Q}}_{i,j}), \; p_{t,i} \;\; is \; in \; quad \; \boldsymbol{Q}_{i,j}
\end{aligned} \tag{3}$$

In the above equation, $\hat{V}_t$ are four vertexs of the a specific quad in which $p_{t,i}$ locates; α_k are coefficients considering the surrounding vertexs of quad $\boldsymbol{Q}_{i,j}$ in standard mesh as orthogonal basis that compose $p_{t,i}$. Using the same α_k, but the vertexs in the quad $\hat{\boldsymbol{Q}}_{i,j}$ of the deformed mesh, the bottom equation in equation 3 can be utilized to compute the location of the corresponding feature point's location in the resulting deformed image. The weight matrix W still serves the propose to smooth the influence brought by the appeareance of the new corner points so that the mesh will not deform abruptly when a new corner point first appears.

However, $\mathcal{L}_{fea}$ can only affect the quads that contain corner points, and in practice, a large propotion of the quads do not have any corner points. $\mathcal{L}_{fea}$ itself would make some of the quads deform significantly while the other quads have no deformations, which will significantly distort the image. Therefore, $\mathcal{L}_{blend}$ is proposed to spread the deformation of the quads to the neighboring quads, so that the quads have corner points will not deform significantly, and the neighboring quads' will be deformed accordingly. The blending energy can be formulated using the following equation:

$$\mathcal{L}_{blend} = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \{ \sum_{k=1}^{4} \alpha_1 (\|\boldsymbol{E}(\boldsymbol{Q}_{i,j})_k\|^2 - \|\boldsymbol{E}(\hat{\boldsymbol{Q}}_{i,j})_k\|^2) + \alpha_2 cos(\theta_k) \} \tag{4}$$

In the above equation, θ_k indicates the four corner's angle of the quad $\hat{\boldsymbol{Q}_{i,j}}$; α_1 and α_2 are coefficients balancing the two terms.$(\|\boldsymbol{E}(\boldsymbol{Q}_{i,j})_k\|^2 - \|\boldsymbol{E}(\hat{\boldsymbol{Q}}_{i,j})_k\|^2)$ encourge the deformation do not change the length of the edges while $cos(\theta_k)$ encourges the quad to be a rectangle and eliminates shear transformation because shaking lens should not result in any shear transformation that need to be compensated.

The total energy was described in the following equation:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{aff} + \lambda_2 \mathcal{L}_{fea} + \lambda_3 \mathcal{L}_{blend} \tag{5}$$

where $\lambda_1, \lambda_2, \lambda_3$ are coefficients balancing the optimization energy.

3 Implementation Details

In this experiement, the values for the several parameters mentioned above are $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 4, \alpha_1 = 1, \alpha_2 = 1.5$. It takes around 1 sec to calculate the deformed mesh and deform the image per frame on Intel Xeon CPU E5-2620, speedup by multi-threading. The optimizer is Sequential Least Square implemented using Scipy,the objective tolerance for the optimizer is 0.01. The meshes contains 150 quads for a video with size 600×400, 15 quads per row and 10 quads per column. More quads are required for a video with higher resolution.

4 Experiments

In order to evaluate the effectiveness of the proposed algorithm, experiements are conducted. An evaluation dataset of shaking input videos was built to conduct evaluation.

4.1 Evaluation Dataset

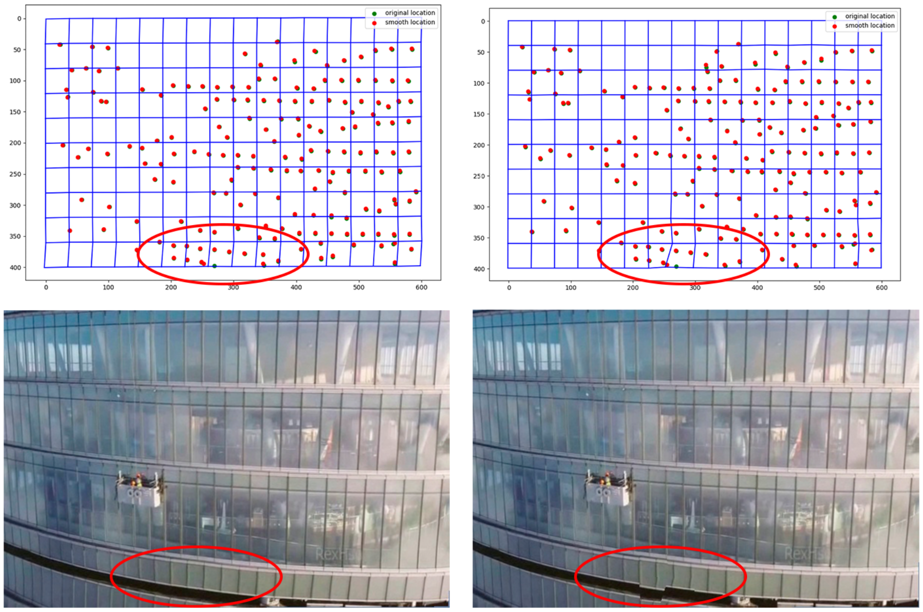
The evaluation dataset contains totally 79 videos, and the average duration of the videos are 8 seconds. The videos are collected on the internet, and most of them are aerial photography captured by drones, some videos contain people with various motion. All the videos have constant exposure. The original version of the videos are not shaking significantly, therefore, I use Affine transformation to randomly translate and rotate each frame in the video to simulate unstable videos. All of the videos are resized and cropped to 600×400 for simplicity. In order to challenge the proposed algorithm, part of the videos have complicated lens movement or prominent foreground object with complex motion.

4.2 With or Withour Blending Energy

In this section, the comparison between using blending energy and without using blending energy is conducted. The qualitative comparison is shown in Figure 2. As shown in Figure 2, the quads when using blending energy have relatively smaller deformation, and they have similar surface areas. However, when not using blending energy, some quads are deformed significantly while the others are not; and some quads have significant larger surface area than the others. The resulting deformed image without using blending energy has significant distortions while the distortion in that using blending energy has trivival distortions.

4.3 Comparison with Global Rigid Transformation

[9] conducts video stabilization using global rigid transformation, more specifically, Affine transformation. I reproduced this work by replacing the image warping by global Affine transformation, and the global Affine transoformation matrix is computed using equation 2. Comparison is conducted between my proposed method and [9]. Quantitative comparison is conducted using average trajectory acceleration (ATA) of a video as a metric, computed by averaging the magnitutude of the second



With blending loss

Without blending loss

Figure 2: comparison between with and without blending energy.

Comparison pairs			
video set	video set	ATA difference	standard deviation
Input videos	Wu, et al. [9]	1.7205	0.329
Input videos	ours	1.7723	0.332
Wu, et al. [9]	ours	0.0518	0.028

Table 1: Comparison among input videos, videos genernated by [9] and ours. The ACE difference is computed by the ACE score of the videos set in the first column subtracting the ACE score of the videos set in the second column.

derivate of the trajectories' motion. If the ATA is high for a video, the video is shaky, otherwise it is stable. For two set of videos derived from the evaluation dataset, the ATA difference is computed by subtracting the ATA scores for each pair of videos originated from the same source video. If the ATA difference between two set of videos is significant $(p < .05)$, then one set of video is smoother than the other set significantly. The quantitative comparison of ATA scores differences among the input shaking videos, videos stabilized by [9] and ours method are shown in Table 1.

From Table 1, conclusion can be drawn that both [9] and my method produce smooth videos that are significantly smoother than the input shaking videos $(p < .01)$. And my method produce a smoother video than [9] with 95% confidence level $(p < .05)$. The experiment verifies the effectiveness of my proposed method.

5 conclusion

I proposed a video stabilization method that successfully stabilize the videos in the evaluation dataset which contains complicated videos. The weighted feature displacement energy and the blending energy successfully make the resulting videos to have smooth trajectory motion without image distortion. Experiments further shows that my method outperforms [9] significantly.

References

[1] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *Proc Eccv*, 11(12):237–252, 1992.

[2] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.

[3] C. G. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[4] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung. Video stabilization using robust feature trajectories. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1397–1404. IEEE, 2009.

[5] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):1150–1163, 2006.

[6] J. Shi and C. Tomasi. Good features to track. Technical report, Cornell University, 1993.

[7] Y.-S. Wang, F. Liu, P.-S. Hsu, and T.-Y. Lee. Spatially and temporally optimized video stabilization. *IEEE transactions on visualization and computer graphics*, 19(8):1354–1361, 2013.

[8] G. Wolberg. *Digital image warping*, volume 10662. IEEE computer society press Los Alamitos, CA, 1990.

[9] S. Wu and Z. Ren. Video stabilization by multi-trajectory mapping and smoothing. In *2005 5th International Conference on Information Communications & Signal Processing*, pages 542–545. IEEE, 2005.