21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Robust Video Stabilization Based on Bounded Path Planning

Chunhe Song[1,2], Hai Zhao[1], Wei Jing[1,2], Yuanguo Bi[1]
[1] *School of Information Science and Technology, Northeastern University, Shenyang, China*
[2] *Department of Electrical and Computer Engineering, University of Waterloo, Canada*
chunhe.song@gmail.com, zhaoh@mail.neu.edu.cn, weijing.neu@gmail.com,
biyuanguo@ise.neu.edu.cn

Abstract

This paper presents a novel video stabilization algorithm based on bounded acceleration, which consists of three steps: motion estimation, virtual path generation and motion compensation. The key insight of this paper is that, to generate a reasonable smooth virtual path, we need to 1) minimize the acceleration along the virtual path, and 2) limit the offset of translations along x and y direction to a certain range. Compared to stat-of-art video stabilization methods, the proposed algorithm can generate more reasonable virtual paths but has fewer parameters to be adjusted. Theoretical analysis and practical experimental results prove the effectiveness of the proposed method.

1. Introduction

Cameras hand-held or mounted on moving vehicles usually suffer from video instability due to unintentional shakes, which greatly degrades the quality of the video sequences. Video stabilization is, therefore, becoming an indispensable technique in advanced digital cameras and camcorders. Generally, a typical video stabilization method consists of two steps: 1) motion estimation and 2) motion compensation. The first step estimates the relative motion between adjacent frames. The second step compensates the relative motion based on special movement model. Fig.1 shows the schematic diagram of motion compensation, where the area of crop window is the visual region in the output video, w1, w2, h1, and h2 are the shifting allowance distance along directions of x-, x+, y- and y+, respectively. The motion compensation is to make the output video seem be taken by a camera moving along a smooth virtual path,

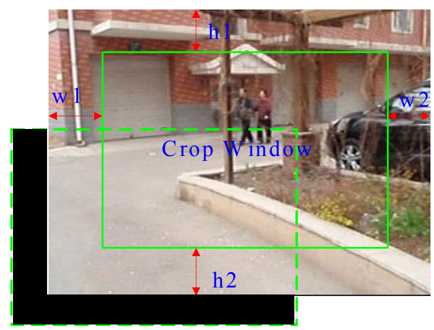

Figure 1: motion compensation of video stabilization

under the condition of motion estimation in the first step of video stabilization.

At present lots of video stabilization methods have been developed and most works focus on how to prove the accuracy of motion estimation [3] and how to generate the smooth virtual path [1,4,5,6]. Few of work notice the influence of the size of the crop window. However, from Fig.1 it can be seen that, the offset of the crop window should be in the range of [-w1, w2] and [-h1, h2] along the x direction and y direction, respectively. Although techniques such as inpaint can help to fill up the undefined region, it is inevitable decreasing the quality of the output video. Furthermore, the shifting allowance factor w1, w2, h1 and h2 depend on the size of the crop window and the full image, therefore the size of the crop window should play an important role in video stabilization.

The main contribution of this paper is a novel video stabilization algorithm which takes the size of the crop window into account. Compared to stat-of-art video stabilization methods, the proposed algorithm can generate more reasonable virtual paths but has fewer parameters to be adjusted.

2. Related work

The virtual path generation method is very important for video stabilization. Kalman filtering may be the most popular online path generation method in pervious video stabilization work [1,7]. However, it always assumes a simple motion model, such as static or with constant velocity, and in each step its result contains a random number to estimate the system noise, which makes the path not smooth. Low-pass filtering is the simplest approach to smooth camera motion which is to just run a low-pass filter over the trajectories [5]. The choice of the kernel width of low-pass filtering is a difficult issue, as small width can not ensure the smoothness of the path, while path with large width may be over smoothed and can not reflect the actual path. Polynomial path fitting can yield a smooth path in accordance with the actual path [5]. However, the virtual path is hard to adjust. When the actual path is complex, it is usually difficult to model the actual path even by a high order polynomial model.

3. Bounded acceleration based virtual path generation

3.1. The proposed method

Some recent researches attempt to plan a smooth, non-parameter path without modeling the actual path [1,4,6]. The main consideration is that a smooth path should have small acceleration. We adopt this idea and the proposed virtual path generation method can be expressed as Eq.1 and Eq.2:

$$E = \sum_{m=1}^{K}\left(\lambda_m \sum_{i=2}^{N-1} \left(2 * x_m^i - x_m^{i-1} - x_m^{i+1} \right)^2 \right) \tag{1}$$

subject to:

$$\alpha_m \le X_m^i - x_m^i \le \beta_m \tag{2}$$

where X_m^i and x_m^i are the positions of actual path and virtual path of i-th frame, α_m and β_m are constraints, m indicates the dimension of the motion model, e.g., $m = 2$ when only translation along x and y directions are considered, and $m = 3$ when translation along x and y directions and rotation are considered. λ_m is the weight parameter. $2 * x_m^i - x_m^{i-1} - x_m^{i+1}$ is the acceleration and $X_m^i - x_m^i$ is the displacement between the actual path and the virtual path, N is the number of frames, K is the dimension of the motion model. By minimizing E, a smooth path of the virtual camera may be obtained.

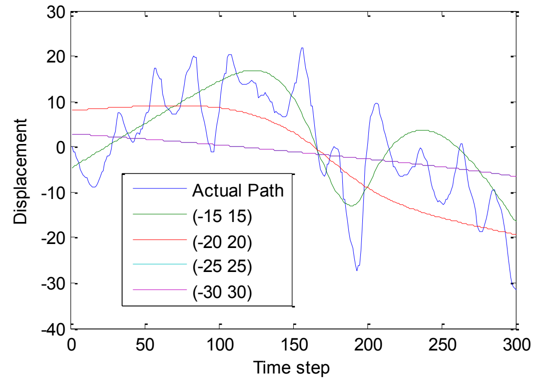

Figure 2: Comparative results of the proposed method with different size of crop windows.

In the proposed algorithm, α_m and β_m depend on the size of the crop window, in particular, in the case as shown in Fig.1, $\alpha_1 = -w_1$, $\beta_1 = w_2$, $\alpha_2 = -h_1$ and $\beta_2 = h_2$. A comparative experiment of the proposed algorithm with different size of crop window is shown in Fig.2. Without loss of generality, in Fig.2, we consider a 1-D case, and the blue curve is the actual camera path, and the green, red, cyan-blue and purple curves are the virtual paths generated by the proposed algorithm under the conditions of different sizes of crop window. For each curve, the corresponding two numbers are α and β in Eq.2. From Fig.2 it can be seen that: 1) our algorithm can generate smooth virtual path; 2) virtual paths with different size of crop windows are different, the virtual path with smaller size of the crop window is smoother, meanwhile, has larger displacement from the actual paths, and 3) when absolute value of α and β is larger than a threshold, the virtual path will be a straight line. Note that the cyan-blue path and the purple path are straight lines and overlap. We will discuss this case in section 3.3.

3.2. The relationship with previous works

The proposed algorithm is similar to same previous works [1,4,6] to a certain extent. Although having different forms, the basic ideas are same in these works, and the cost function can be expressed as the follows:

$$\begin{aligned} E = &\sum_{m=1}^{K}\left(\lambda \sum_{i=2}^{N-1} \left(\left\| x_m^{i+1} + x_m^{i-1} - 2x_m^i \right\|_L \right) \right) \\ &+ \sum_{m=1}^{K}\left(\alpha \sum_{i=1}^{N} \left(\left\| X_m^i - x_m^i \right\|_L \right) + \beta \sum_{i=1}^{N-1} \left(\left\| x_m^{i+1} - x_m^i \right\|_L \right) \right) \end{aligned} \tag{3}$$

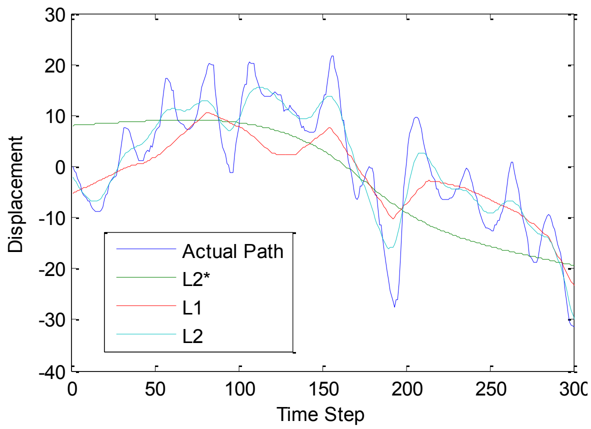

Figure 3: Comparative results of different models.

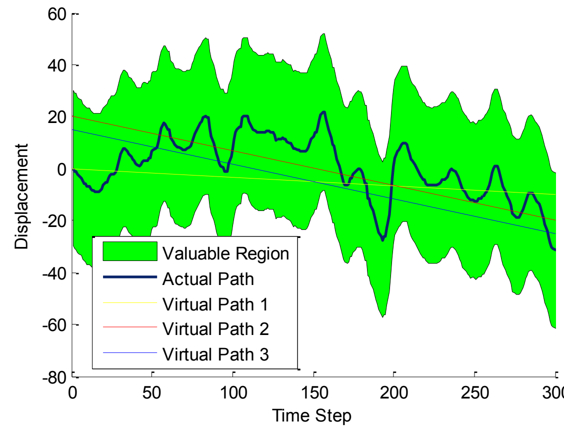

Figure 4: The uncertainty of the solution

Compared to Eq.1 and Eq.2, it is clear that the main difference between our method and algorithm using Eq.3 is that, they put constraints into the cost function. From the theoretical point of view, Eq.3 is a trade-off between trying to improve the smoothness of the path (the first item in Eq.3) and trying to reduce the undefined region (the second and the third items), however, in most of situation the two 'efforts' are conflicting, resulting in a generated virtual path which is not smooth enough meanwhile remains some undefined region in the output video. But in the proposed algorithm, Eq.2 guarantees no undefined region, and Eq.1 ensures the smooth of the virtual camera path. From a practical point of view, the output video is the region in the crop window, and we only need to guarantee no undefined region in the crop window, rather than try to minimize the undefined region in the full image. Fig.3 shows that comparative results of virtual paths using different models, where L2* is the result of the proposed algorithm, L1 is the result of [1], and L2 is the result of [4] ([6] and [4] are the same in 1-D space). In this test, in the models of [1] and [4], $\alpha = 0.001$, $\beta = 0.001$, and $\lambda = 1$. In the proposed model, $\alpha = \beta = 25$. It is clear that, although α and β is much smaller than λ , the second and third items in Eq.3 reduce the smoothness of the virtual path obviously.

3.3. The uncertainty of the solution

It is worth noting that there may be more than one satisfied solution using the proposed algorithm under a small size of the crop window. Fig.4 gives an example of such case. In Fig.4, the green region is the valuable region which means that the virtual path in this region will not bring in undefined region. In this case, virtual path 1, virtual path 2 and virtual 3 are straight lines. As all virtual paths are in valuable region, these paths are satisfactory, and can be used in a practical video stabilization system. It is also worth noting that, except the case that the actual path is a straight line, algorithms based on Eq.3 can never generate a straight virtual path, the most 'stable' path, due to the influence of inner constraints.

4. Video stabilization

Our video stabilization method consists of three steps: motion estimation, path planning and motion compensation. Our motion estimation method is based on feature points, and SURF is used to extract and match feature points in the two consecutive frames [2]. In this paper we adopted a rigid 2D model for convenience. It was expressed as:

$$\begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = S_t \begin{bmatrix} \cos\theta_t & -\sin\theta_t & T_t^x \\ \sin\theta_t & \cos\theta_t & T_t^y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{bmatrix} \qquad (4)$$

or in the form of $Y = AX$, where (x_t, y_t) is the coordinate at time t , θ_t is the rotation, T_t^x and T_t^y are the translations in the horizontal and vertical direction, respectively, and S_t is the scaling factor. Incorrect corresponding and feature points on local moving objects will reduce the accuracy of Eq.5, and we use the RANSAC algorithm to eliminate these outlier feather points.

5. Experiments

We tested the proposed approach on 20 video sequences, which were taken by a digital camera in different scenes. All experiments are performed in a laptop with a 3.0 GHz CPU and 2GB memory, and software is written in C++. As comparisons, we also test the performance of models in [1] and [4]. In the

Figure 5: The results of the proposed video stabilization algorithm. Images in the first row are original frames, and images in the second row are the video stabilization results.

proposed algorithm, $\alpha = [-40; -20; -0.1; -0.1]$, $\beta = [40; 20; 0.1; 0.1]$, and $\lambda = [1;1;100;100]$. The four dimensional vector represents parameters of the x-translation, y-translation, rotation and the scale, respectively. The parameters of models in [1] and [4] are the same in their works.

Fig.5 shows examples of the original video sequence and the results of the proposed algorithm, and we gives the corresponding comparative results of our model and models in [1] and [4] along the x-direction, y-direction, rotation and the scale factor in Fig.6.

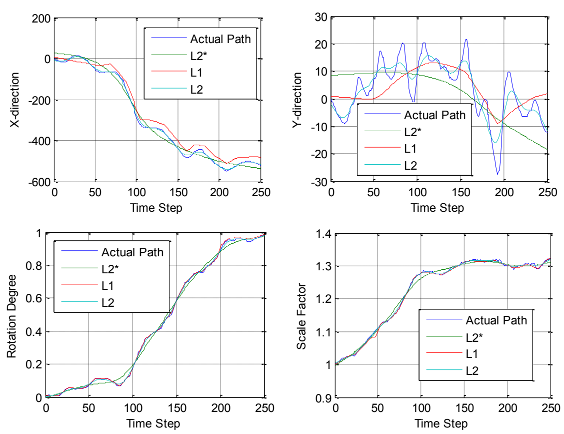

Figure 6: Comparative results of different models.

In Fig.6, L2* is the proposed model, L1 is the model in [1] and L2 is the model in [4]. We refer readers to zoom in the figure to see the differences among these curves. As shown in Fig.6, virtual paths generated by the proposed algorithm are smoother than the results from [1] and [4], which confirms the accuracy and effectiveness of the proposed algorithm.

6. Conclusion

In this paper, we propose a video stabilization algorithm based on based on bounded acceleration. The key idea of the proposed algorithm is a novel virtual path generation method based on bounded acceleration. Experiments have confirmed the accuracy and effectiveness of the proposed algorithm.

References

[1] GRUNDMANN, M., KWATRA, V., AND ESSA, I. “Auto-directed video stabilization with robust l1 optimal camera paths,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, 225-232.

[2] HERBERT B, TINNE T, LUC V. “SURF: Speeded Up Robust Features,” In Proceedings of European Conference on Computer Vision, ECCV2006: 404-417.

[3] KO, S.-J., LEE, S.-H., AND LEE, K.-H. “Digital image stabilizing algorithm based on bit-plane matching,” IEEE Transactions on Consumer Electronics, 1998: 44(3): 617-622.

[4] LEE, K.-Y., CHUANG, Y.-Y., CHEN, B.-Y., AND OUHYOUNG, M. “Video stabilization using robust feature trajectories,” In Proceedings of the IEEE International Conference on Computer Vision, 2009, 1397-1404.

[5] LIU, F., GLEICHER, M., WANG, J., AND JIN, H. “Subspace video stabilization,” ACM Transactions on Graphics 2011, 30, 1, Article No.4.

[6] SMITH, B.-M., ZHANG, L., JIN, H., AND AGARWALA, A. “Light field video stabilization,” In Proceedings of the IEEE International Conference on Computer Vision, 2009.

[7] OKUDA, H., HASHIMOTO, M., SUMI, K., KANEKO, S. “Optimum motion estimation algorithm for fast and robust digital image stabilization,” IEEE Transactions on Consumer Electronics, 2006, 52(1): 276-280.