SHAKING VIDEO SYNTHESIS FOR VIDEO STABILIZATION PERFORMANCE ASSESSMENT

Hui Qu, Li Song, Gengjian Xue

Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University

ABSTRACT

The goal of video stabilization is to remove the unwanted camera motion and obtain stable versions. Theoretically, a good stabilization algorithm should remove the unwanted motion without the loss of image qualities. However, due to the lack of ground-truth video frames, the accurate performance evaluation of different algorithms is hard. Most existing evaluation techniques usually synthesize stable videos from shaking ones, but they are not effective enough. Different from previous methods, in this paper we propose a novel method which synthesize shaking videos from stable frames. Based on the synthetic shaking videos, we perform preliminary video stabilization performance assessment on three stabilization algorithms. Our shaking video synthesis method can not only give a benchmark for full-reference video stabilization performance assessment, but also provide a basis for exploring the theoretical bound of video stabilization which may help to improve existing stabilization algorithms.

Index Terms — Video stabilization, ground-truth, performance assessment

1. INTRODUCTION

Hand-held or vehicle mounted video cameras often lead to shaking videos, which usually have low visual quality and are not suitable for some post processes such as object detection and tracking. Video stabilization is a technique that aims to reduce or remove the unwanted camera motion to obtain more stable videos. Among many stabilization algorithms, Grummann’s L1 Camera Path Optimization method [1] and Liu’s Subspace method [2] are considered as state-of-the-art. However, the evaluation of these algorithms is mainly based on subjective evaluation, which is hard to analyze in quantitative way. Therefore, more effective assessment modes are needed.

In the past few years, some performance evaluation methods for video stabilization have been proposed. Morimoto et al. [3] used the peak signal-to-noise ratio (PSNR) to evaluate two kinds of fidelities of stabilized video: the interframe transformation fidelity (ITF) and the global transformation fidelity (GFT). ITF is the PSNR between two consecutive stabilized frames in the cases without ground-truth videos, while GTF is the PSNR between the ground-truth frame and the current stabilized frame. Zhang et al. [4] proposed some performance metrics for electronic stabilization, such as accuracy, capture range, tolerance, etc.

In addition to these methods based on objective metrics, subjective metrics related to Human Vision System (HVS) have also been paid much attention to, as stabilization is performed mainly to satisfy human eyes. Offiah et al. [5] proposed a subjective Mean Opinion Score (MOS) metric by extending Zhang et al.’s work [4]. They evaluated the performance of four stabilization algorithms in endoscopy by comparing the stabilized videos to the synthetic reference version based on both the objective and subjective metrics. However, one problem is that the assessment results may be not so reliable since the reference videos are hardly the real ground-truth (it is impossible to record stable videos in endoscopy). For other applications, it is better to compare the stabilized video to the ground-truth in order to obtain more accurate assessment results.

Instead of synthesizing stable version from shaking videos in [5] or modeling intentional camera motion from computed camera motion in [6], in this paper we propose to model the unwanted camera motion (known as jitter) and synthesize shaking videos from ground-truth stable ones. Then these shaking videos are stabilized by different algorithms. It is known that full-reference-based performance assessment is usually more reliable. With ground-truth videos, it is possible for us to establish an analytical model to evaluate different stabilization algorithms quantitatively.

This work also provides the basis to explore the performance bound of video stabilization algorithms. Like other image restoration problems, for example denoising, theoretical bound analysis is an active topic in recent years. Several works [7, 8, 9, 10] have discussed it because the state-of-the-art denoising algorithms have produced quite comparable results. Since video stabilization is an inverse problem like denoising, it has the similar trend in the future. Motivated by image denoising, we propose to synthesize shaking videos by adding “motion noise” to the ground-truth video just like adding noise to the original noise-free image for further study on this issue.

The rest of this paper is organised as follows. The details of proposed shaking video synthesis method are described in section II. Some preliminary performance evaluation of dif-
different algorithms are presented in section III, and conclusion comes in section IV.

2. SHAKING VIDEO SYNTHESIS

In this section, we first introduce how to model the unwanted camera motion and synthesize shaking videos from the ground-truth stable ones. Then we compare the frames of synthetic shaking videos with that of the real shaking videos to show its effectiveness.

2.1. Modeling unwanted camera motion

First, we need to extract the unwanted camera motion from the shaking motion. One simple method is to compare the shaking one with the original stable camera motion. However, this is impossible as the original motion is usually unknown. In Grundmann et al.’s paper [1], the optimal camera path $P_t$ was defined as a combination of three kinds of segments based on cinematography principles, which resulted in good performance. It is reasonable to treat this optimal camera path as the stable camera motion. The relationship between the original path and optimal path is

$$P_t = C_t B_t$$

where $C_t$ is the original camera path and $B_t$ is the transform matrix. From equation (1) we can obtain

$$C_t = P_t B_t^{-1}$$

Thus $B_t^{-1}$ can be treated as the unwanted camera motion “added” to the ideal camera motion $P_t$, which results in the true camera motion $C_t$. Therefore, we can adjust $B_t^{-1}$ to synthesize the shaking videos.

To simplify the analysis, $B_t^{-1}$ is set as a similarity transform which has four parameters:

$$B_t^{-1} = \begin{bmatrix} a_t & b_t & dx_t \\ -b_t & a_t & dy_t \\ 0 & 0 & 1 \end{bmatrix}$$

where $a_t, b_t$ are related to the scale and rotation, and $dx_t, dy_t$ denote the shift in $x$ and $y$ directions respectively. Discrete Fourier Transform (DFT) is used to analyze the frequency components of each parameter. The results of one sequence named “gleicher1” are shown in Fig.1. From the results we can see that for all parameters there are some dominated frequency components in the low frequency area, and the number of frequency components of parameter $a$ is fewer than that of the others. The similar results can be found in most of other test video sequences that are used in video stabilization algorithms in [1], [2] and [11]. Inspired by the results, we can use the following unified model to simulate different parameters:

$$p_t = \sum_{i=1}^{n} w_i \sin(2\pi f_i t + \phi_i) + \eta_t, t = 1, 2, \cdots, N$$

where $p$ can be $a, b, dx$ or $dy$, $N$ is the number of frames, $w_i$ is the weight of each sine function and set to $1/i$, $f_i$ is the different frequency components, which is a random value within a range, $\phi_i$ denotes the phase and is decided by a same random value for all $i$, and $\eta$ is the noise component. We can obtain several kinds of unwanted camera motion by adjusting the values of $n, f_i, \eta$.

![Fig. 1. Values and frequency components of each parameter of unwanted camera motion in “gleicher1”](image)

2.2. Ground-truth dataset

Ground-truth videos are the basis for full-reference-based performance assessment. Most videos in the dataset are selected
from the test sequences in video coding, which are not only representative but also have high quality. These videos are classified into three types according to the movement of the camera:

(1) static camera (type I): the camera does not move or rotate, so the background of videos is stationary. The typical videos of this kind are surveillance videos.

(2) panning camera (type II): the camera is panning when it records these videos and the location of the camera actually does not change.

(3) walking camera (type III): this type of videos are produced by a camera which moves like people walk. The location of the camera is not fixed.

In fact, ground-truth videos of the third type are hard to be found because walking cameras usually produce shaking videos. However, it is unreasonable to eliminate it since a large part of videos to be stabilized belong to this type. We extract several videos in high-definition movies and TV plays. They are shot by professional devices and therefore stable and have high quality.

The selected videos also include various scenes, e.g. static scene, scene with moving objects, scene with camera zooming, scene with distant view, making the dataset more representative.

2.3. Jitter simulation

Unwanted camera motion (jitter) is modeled in section 2.1. Here we use four parameters to characterize jitters:

- **direction**: indicates which direction has jitter, the default value is in both $x$ and $y$ directions.
- **severity**: indicates whether the jitter is severe or not. It controls the values of $n$ and $f$. When the jitter is severe, $n$ is larger and the range of $f$ is larger, as well. This parameter also affects the maximum value of $a$, $b$, $dx$, and $dy$.
- **similarity**: indicates whether there are scale and rotation changes in the jitter. We don’t consider non-rigid transformations such as affinity and homography.
- **noisetype**: indicates which kind of noise is contained in the jitter, i.e. gaussian noise or uniform random noise.

We simulate two kinds of jitter with certain value combinations of the four parameters, and parameter values are shown in Table 1. “Jitter1” is simulated as the jitter with high frequent tremble and small shift, which is often caused by wind and usual in surveillance videos. “Jitter2” is simulated as the jitter with relative low frequent shake and large shift, which is often caused by walking and usual in videos shot by hand-held cameras. Other kinds of jitters can be imitated with different combinations. It should be noted that some simulated jitters may be strange because they don’t exist in the real world.

2.4. Results and comparison

After obtaining the transform matrices of jitter, we can synthesize a shaking video from the ground-truth one by warping each frame with corresponding matrix. Then the warped frames are cropped to keep central parts only for better visual quality.

We add two kinds of jitter simulated in section 2.3 to 12 ground-truth video sequences mentioned in section 2.2. “bank” is a surveillance sequence of type I while “sidewalk” is a real shaking surveillance sequence. Part results of adding “Jitter1” to “bank” are shown in Fig.2(a), which is the stable camera path and shaking path of synthetic video in $y$ direction. And the original camera path and optimal camera path after stabilization of sequence “sidewalk” are shown in Fig.2(b) to make comparison. For “Jitter2”, “Cyclists” is a sequence of type II while “gleicher1” is a real shaking sequence shot by hand-held cameras when the person is walking. Results of sequence “Cyclists” are shown in Fig.3(a) and comparison camera paths of sequence “gleicher1” are shown in Fig.3(b).

<table>
<thead>
<tr>
<th>direction</th>
<th>severity</th>
<th>similarity</th>
<th>noisetype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jitter1</td>
<td>$x,y$</td>
<td>light</td>
<td>no</td>
</tr>
<tr>
<td>Jitter2</td>
<td>$x,y$</td>
<td>severe</td>
<td>yes</td>
</tr>
</tbody>
</table>

Fig. 2. Jitter1 results: paths of synthetic shaking video and real shaking video in $y$ direction

Fig. 3. Jitter2 results: paths of synthetic shaking video and real shaking video in $y$ direction

Table 1. Two kinds of simulated jitters
The figures show that the synthetic shaking paths (red lines in Fig.2(a) and Fig.3(a)) are very similar to the real shaking paths (blue lines in Fig.2(b) and Fig.3(b)). Besides, the visual effect of synthetic shaking video is close to that of real shaking one. Some frames of ground-truth video “Cyclists” and its synthetic shaking video are shown in Fig.4. More results and comparison are available at the website http://www.youku.com/playlist_show/id_19274439.html. These results demonstrate that our method is reliable to model unwanted camera motions and can be used in full-reference-based video stabilization performance assessment.

3. PRELIMINARY EXPERIMENTS OF REPRESENTATIVE STABILIZATION METHODS

When both ground-truth stable videos and synthetic shaking videos are available, they can be used to assess the stabilization performance of different algorithms, as well as to study the theoretical bound of video stabilization. Since our focus in this paper is shaking video synthesis, we just evaluate the performance of several representative 2D stabilization algorithms preliminarily.

Typically, PSNR and Structural Similarity (SSIM)[12] are two simple metrics of image quality. They can be extended to video quality assessment by comparing still images on frame-by-frame basis. PSNR and SSIM are similar and the values of PSNR can be predicted from SSIM and vice-versa according to Hore et al.’s work [13]. Here we choose SSIM for image quality assessment.

First, we produce shaking videos of three types of ground-truth videos introduced in section 2.2 with “Jitter1” and “Jitter2”. Then we use a commercial stabilization software Deshaker [14], Grundmann et al.’s algorithm [1] and Qu et al.’s algorithm [15] to stabilize the synthetic videos. Finally, we evaluate these algorithms in two aspects: (1) the distance per frame between the stabilized camera path and the reference video’s path, denoted by $m_{\text{path}}$; (2) the SSIM between two consecutive frames of stabilized video, denoted by $m_{\text{ssim}}$, and SSIM between adjacent frames of shaking video is also computed for comparison. Smaller $m_{\text{path}}$ indicates that the stabilized video is closer to the original ground-truth video and larger $m_{\text{ssim}}$ indicates better smoothness of the camera path since dramatic changes between adjacent frames would result in low SSIM. The results of $m_{\text{path}}$ and $m_{\text{ssim}}$ are shown in Table 2 and Table 3. We can see that Grundmann et al.’s and Qu et al.’s algorithms are better than Deshaker if only considering $m_{\text{ssim}}$. But the path distance of Grundmann’s method is larger than the other two methods. These results can also be reflected in Fig.5.

In fact, it is better to calculate SSIM between the stabilized frame and the reference frame. However, since we crop the frame in both shaking video synthesis and stabilization,
the resolution of the stabilized frame is smaller than that of the reference frame, which makes it hardly for us to compute the SSIM directly. Alternatively, we can either compare the stabilized frame with the central part of reference frame or calibrate the stabilized frame to compare it with the counterpart in reference frame. The first comparison method is simple to implement but cannot reflect the performance of stabilization algorithms exactly, because the stabilized camera path may be smooth enough with a shift compared to the reference path. In this case SSIM is very small while the performance is relatively good. The second comparison method is more reasonable if we want to assess the image quality namely blurring and deformation. However, it seems to be complex to perform frame calibration. Therefore, we choose to compute SSIM between adjacent frames rather than between stabilized frames and reference frames.

### 4. CONCLUSION AND FUTURE WORK

We have proposed a shaking video synthesis method for video stabilization performance assessment. By modeling the unwanted camera motion, we can simulate different kinds of jitter and synthesize shaking videos from ground-truth ones. The synthetic shaking videos are verisimilar to the real shaking ones on both the camera motions and the visual effects. Based on these videos, we preliminarily assess the performance of three representative 2D video stabilization algorithms on different sequences.

With the existence of ground-truth videos and its shaking ones, it is possible for us to make more robust assessment on video stabilization algorithms, especially for those with close performance judged by human eyes. Moreover, we could try to analyze the theoretical bound of video stabilization for im-

---

### Table 2. Path distance per frame between stabilized videos and reference videos for three algorithms (unit: pixel): $m_{path}$. The values in the table correspond to an average by frame.

<table>
<thead>
<tr>
<th>video type</th>
<th>video name</th>
<th>Jitter1</th>
<th>Jitter2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shaking</td>
<td>deshaker</td>
<td>Grundmann</td>
</tr>
<tr>
<td>Type I: (static camera)</td>
<td>bank</td>
<td>8.21</td>
<td>8.36</td>
</tr>
<tr>
<td></td>
<td>Night</td>
<td>12.72</td>
<td>12.94</td>
</tr>
<tr>
<td></td>
<td>Optis</td>
<td>19.84</td>
<td>14.02</td>
</tr>
<tr>
<td></td>
<td>ShuttleStart</td>
<td>19.37</td>
<td>14.87</td>
</tr>
<tr>
<td></td>
<td>station</td>
<td>198.33</td>
<td>61.63</td>
</tr>
<tr>
<td>Type II: (panning camera)</td>
<td>Cyclists</td>
<td>92.31</td>
<td>64.36</td>
</tr>
<tr>
<td></td>
<td>Raven</td>
<td>20.88</td>
<td>25.79</td>
</tr>
<tr>
<td></td>
<td>SOCCER</td>
<td>45.50</td>
<td>96.35</td>
</tr>
<tr>
<td></td>
<td>stockholm</td>
<td>33.58</td>
<td>28.16</td>
</tr>
<tr>
<td>Type III: (walking camera)</td>
<td>City</td>
<td>36.83</td>
<td>25.42</td>
</tr>
<tr>
<td></td>
<td>Django1</td>
<td>38.49</td>
<td>34.60</td>
</tr>
<tr>
<td></td>
<td>Thrones1</td>
<td>51.50</td>
<td>70.25</td>
</tr>
</tbody>
</table>

### Table 3. Inter-frame SSIM of shaking videos and stabilized videos for three algorithms: $m_{ssim}$. The values in the table correspond to an average by frame.

<table>
<thead>
<tr>
<th>video type</th>
<th>video name</th>
<th>Jitter1</th>
<th>Jitter2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shaking</td>
<td>deshaker</td>
<td>Grundmann</td>
</tr>
<tr>
<td>Type I: (static camera)</td>
<td>bank</td>
<td>0.4906</td>
<td>0.9347</td>
</tr>
<tr>
<td></td>
<td>Night</td>
<td>0.4189</td>
<td>0.8662</td>
</tr>
<tr>
<td></td>
<td>Optis</td>
<td>0.5732</td>
<td>0.9139</td>
</tr>
<tr>
<td></td>
<td>ShuttleStart</td>
<td>0.7807</td>
<td>0.9718</td>
</tr>
<tr>
<td></td>
<td>station</td>
<td>0.4842</td>
<td>0.8329</td>
</tr>
<tr>
<td>Type II: (panning camera)</td>
<td>Cyclists</td>
<td>0.6586</td>
<td>0.7910</td>
</tr>
<tr>
<td></td>
<td>Raven</td>
<td>0.5480</td>
<td>0.6878</td>
</tr>
<tr>
<td></td>
<td>SOCCER</td>
<td>0.4443</td>
<td>0.6419</td>
</tr>
<tr>
<td></td>
<td>stockholm</td>
<td>0.3979</td>
<td>0.6613</td>
</tr>
<tr>
<td>Type III: (walking camera)</td>
<td>City</td>
<td>0.3166</td>
<td>0.8295</td>
</tr>
<tr>
<td></td>
<td>Django1</td>
<td>0.3065</td>
<td>0.6640</td>
</tr>
<tr>
<td></td>
<td>Thrones1</td>
<td>0.5785</td>
<td>0.6810</td>
</tr>
</tbody>
</table>
provement on existing algorithms.

5. ACKNOWLEDGEMENT

This work was supported by 863 project (2012AA011703), NSFC (61221001, 60932006), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

6. REFERENCES


