Vis Comput (2009) 25: 997–1008
DOI 10.1007/s00371-009-0310-z

ORIGINAL ARTICLE

Video stabilization based on a 3D perspective camera model

**Guofeng Zhang · Wei Hua · Xueying Qin ·
Yuanlong Shao · Hujun Bao**

Published online: 13 February 2009

Abstract This paper presents a novel approach to stabilize video sequences based on a 3D perspective camera model. Compared to previous methods which are based on simplified models, our stabilization system can work in situations where significant depth variations exist in the scenes and the camera undergoes large translational movement. We formulate the stabilization problem as a quadratic cost function on smoothness and similarity constraints. This allows us to precisely control the smoothness by solving a sparse linear system of equations. By taking advantage of the sparseness, our optimization process is very efficient. Instead of recovering dense depths, we use approximate geometry representation and analyze the resulting warping errors. We show that by appropriately constraining warping error, visually plausible results can be achieved even using planar structures. A variety of experiments have been implemented, which demonstrates the robustness and efficiency of our approach.

G. Zhang · W. Hua · X. Qin (✉) · Y. Shao · H. Bao
State Lab of CAD&CG, Zhejiang University, Zhejiang, People's Republic of China
e-mail: xyqin@cad.zju.edu.cn

H. Bao (✉)
e-mail: bao@cad.zju.edu.cn

G. Zhang
e-mail: zhangguofeng@cad.zju.edu.cn

W. Hua
e-mail: huawei@cad.zju.edu.cn

Y. Shao
e-mail: shaoyuanlong@cad.zju.edu.cn

X. Qin
School of Computer Science & Technology, Shandong University, Jinan, People's Republic of China

Keywords Video stabilization · Structure from motion · Optimization · View warping · Warping error

1 Introduction

The goal of video stabilization is to remove annoying shaky motion from a video sequence. It plays an important role in many applications, such as video compression, video editing, background estimation, and moving objects detection.

In general, video stabilization is made up of three stages: motion estimation, filtering and compensation [15]. Motion estimation is to select what kind of motion model and how to estimate the motion, while motion filtering is to eliminate undesirable image motion caused by jittering. Compensation is carried out by image re-sampling or view warping operations, usually amounting to 2D affine or homography transformations.

Essentially, video stabilization is to eliminate the undesirable camera motion in the video caused by handheld or mechanical vibration. Traditional approaches employ simplified models (e.g. affine model or homography model [8, 10, 13]) to estimate 2D global motion of images. Homography model is appropriate if the video scene undergoes planar perspective transformation, e.g. planar scenes with arbitrary camera motion, or arbitrary scenes with fixed camera location. Conditions of affine models are even more strict. However, these methods are too strict to hold on for general video sequences because they will result in large errors if significant depth variations exist in the scenes and the camera undergoes large translational movement. Buehler et al. [4] handled this problem from image-based rendering standpoint based on projective reconstruction of video sequences and upgrading them to a quasi-affine model, and then smooth the sequences via smoothing feature points.

Some methods [5, 24] were proposed to smooth the video sequences by compensating 3D rotation, which can be compensated without the necessity of recovering depth information. Zhu et al. [27] analyzed possible motion patterns and proposed a 2.5D motion model that requires user interaction to choose the dominant character of the motion in the video sequence being analyzed. However, this method cannot deal with general movement. The estimation of general 3D perspective camera motion is the classical structure from motion (SFM) problem [7, 9, 18, 25]. As an area of endeavor, SFM problem has reached a degree of maturity with several commercial offerings [1, 19]. In our implementation, we employ the method proposed in [25].

Several filtering methods [8, 10, 13, 14, 27] have been proposed to reduce unintentional camera motion with respect to scenes while preserving the dominant, intentional camera motion. As we know, missing image areas will appear after stabilization by motion compensation. However, nearly all these previous methods did not discuss how to control the tradeoff between missing image areas and smoothness. Recently, Pilu [17] studied this problem and used the Viterbi method to solve the most stable sequence allowed by the size of the output window. However, this method is recursive and does not give a closed-form solution. Matsushita et al. [13] proposed a motion in-painting approach to fill in the missing image areas. In contrast, we formulate the stabilization problem as a quadratic cost function on smoothness and similarity constraints. This allows us to precisely control the smoothness by solving a sparse linear system of equations.

In compensation stage, original frames should be warped to obtain motion-compensated frames. However, view warping without accurate geometry information will introduce warping error, which may result in serious artifacts. With depth maps, some methods such as 3D warping [12] and layered-depth images [20] can render new views by projecting the pixels of the nearby points of view to their proper 3D locations and re-projecting them onto a new image. Recently, Bhat et al. [3] proposed to stabilize video by smoothing out the original camera path and re-rendering the scene as seen from the new camera path with recovered dense depth maps. Unfortunately, obtaining dense depth information from real images is hard even for the state-of-the-art vision algorithms. Snavely et al. [21], on the other hand, produced 3D meshes by triangulating sparse point clouds, and rendered each mesh with texture map to synthesize a new view. However, missing geometry and outlying points can sometimes cause distracting artifacts.

These limitations and practical demands motivated us to develop an effective method, which does not involve complex dense depths recovery, but can still obtain visually plausible stabilized results. In order to achieve this goal, we make an in-depth analysis of the warping error caused by geometry approximation. Our experiments show that even using planar structure to approximate scene geometry, visually plausible results can be achieved, by constraining the warping error to achieve optimal smoothness results and avoid the distortion artifacts. In order to achieve optimal smoothness, we formulate the stabilization problem as a quadratic cost function on smoothness and similarity constraints. By exploiting the sparseness of the linear system, our optimization process is very efficient and fast.

The rest of this paper is organized as follows. Section 2 gives an overview of our method, and introduces the proposed cost function for video stabilization as well as the optimization method. We elaborate our view warping method and make an analysis of warping error in Sect. 3. Experimental results are given in Sect. 4. Finally, we conclude the whole paper.

2 Our approach

2.1 Overview

We begin the description with the original video sequence $\mathbf{V_O} = \{I(o_i)|i = 1, \ldots, n\}$, where o_i is the ith original camera, $I(o_i)$ is its corresponding image, and $\mathbf{O}$ is the sequence of original cameras. We aim to obtain its motion-compensated sequence $\mathbf{V_C} = \{I(c_i)|i = 1, \ldots, n\}$, where c_i is the ith compensated camera, and $\mathbf{C}$ is its sequence. Our algorithm consists of the following three major steps:

Step 1 For each original frame $I(o_i)$, recover the extrinsic and intrinsic parameters of the corresponding original camera o_i in the set $\mathbf{O} = \{o_i|i = 1, \ldots, n\}$.

Step 2 Solve the stabilized camera set $\mathbf{C} = \{c_i|i = 1, \ldots, n\}$.

Step 3 For every $i = 1, \ldots, n$, perform view warping operations to obtain motion-compensated frames:

$$I(o_i) \to I(c_i), \quad i = 1, \ldots, n,$$

where the output frames $\mathbf{V_C} = \{I(c_i)|i = 1, \ldots, n\}$ form the resultant motion-compensated video sequence.

The first step is the SFM problem, which outputs the recovered camera motion parameters with sparse 3D feature points. In the second step, the stabilized camera motion parameters can be achieved, by solving a quadratic cost function on smoothness and similarity constraints. In the third step, we employ view warping to obtain the motion-compensated video sequence.

2.2 Camera motion estimation

Our automatic feature tracking algorithm is based upon SIFT algorithm [11] for its reliable performance even in wide-baseline matching case. We extract SIFT features from each frame of the input video sequence and match the features frame by frame. Corresponding features are constrained according to the epipolar geometry theory [26]. We use RANSAC algorithm [6] to find a set of inliers that have consistent epipolar geometry. The matched feature points constitute the feature tracks. Then we use the SFM method proposed in [25] to recover the camera motion parameters from the given video sequence. For completeness we briefly summarize the algorithm as follows.

As we know, structure and motion estimation with longer tracks is more reliable and robust than with short tracks [7, 25]. Let N be the minimum track length we require. Then we select the tracks not shorter than N as superior tracks for reconstruction. And we use the interval $(N-1)/2$ to select the key frames to ensure that all superior tracks stride over at least two key frames. Then we initialize the projective reconstruction from the reference triple key frames according to the criteria proposed in [25]. The projective reconstruction is upgraded to a metric framework at an appropriate moment through self-calibration. For each newly added frame, the new camera parameters and 3D points are initialized, and existing structure and motion are refined. Finally, the whole structure and motion are refined through bundle adjustment [23].

2.3 The cost function for video stabilization

Previous approaches usually employ classical filter algorithms (e.g. Gaussian filtering, Kalman filtering, time averaging, etc.), which is difficult to precisely control the tradeoff between missing image areas and smoothness. We adopt an optimization process to determine $\mathbf{C}$ by minimizing a cost function on smoothness and similarity constraints. Essentially, image jittering is caused by the shaky motion of a camera. Therefore, if we can smooth the camera motion, image jittering can be removed. At the same time, the motion-compensated sequence should be similar to the original sequence, in order to make missing areas small. Therefore, each term is separated into two parts, which are smoothness cost and similarity cost. We use subscripts m and s to distinguish smoothness and similarity, and superscripts c and o to indicate compensated and original parameters, respectively.

The camera motion can be decomposed into rotation, translation, and zooming components. To ensure the visual smoothness of the stabilized video, the rotational, translational and zooming acceleration should be minimized. Therefore, we respectively define the rotational smoothness term $E_{\Theta m}$, zooming smoothness term E_{fm}, and translational smoothness term $E_{\mathbf{t}m}$. The formulations are given as the following:

$$\begin{aligned} E_{\Theta m} &= \sum_{1\le i\le n-2} \left\| \Theta_i^c - 2\Theta_{i+1}^c + \Theta_{i+2}^c \right\|^2, \\ E_{fm} &= \sum_{1\le i\le n-2} \left\| f_i^c - 2f_{i+1}^c + f_{i+2}^c \right\|^2, \\ E_{\mathbf{t}m} &= \sum_{1\le i\le n-2} \left\| \mathbf{t}_i^c - 2\mathbf{t}_{i+1}^c + \mathbf{t}_{i+2}^c \right\|^2, \end{aligned} \tag{1}$$

where Θ denotes the rotational vector expressed by Euler angles, f denotes the focal length, and $\mathbf{t}$ denotes the translational vector.

Similarity constraints require that the warped views should look similar to the original ones and they share large common scene region. Therefore, the camera parameters of the original frames should be as close to those of the warped frames as possible. Our similarity terms are simply given as follows:

$$\begin{aligned} E_{\Theta s} &= \sum_{1\le i\le n} \left\| \Theta_i^c - \Theta_i^o \right\|^2, \\ E_{fs} &= \sum_{1\le i\le n} \left\| f_i^c - f_i^o \right\|^2, \\ E_{\mathbf{t}s} &= \sum_{1\le i\le n} \left\| \mathbf{t}_i^c - \mathbf{t}_i^o \right\|^2. \end{aligned} \tag{2}$$

Finally, the cost functions respectively corresponding to rotational, zooming and translational components are defined as:

$$\begin{aligned} E_{\Theta} &= \omega_{\Theta m}^2 E_{\Theta m} + \omega_{\Theta s}^2 E_{\Theta s}, \\ E_f &= \omega_{fm}^2 E_{fm} + \omega_{fs}^2 E_{fs}, \\ E_{\mathbf{t}} &= \omega_{\mathbf{t}m}^2 E_{\mathbf{t}m} + \omega_{\mathbf{t}s}^2 E_{\mathbf{t}s}, \end{aligned} \tag{3}$$

where $\omega_{\Theta m}$, ω_{fm}, $\omega_{\mathbf{t}m}$, $\omega_{\Theta s}$, ω_{fs}, $\omega_{\mathbf{t}s}$ are weights of the cost terms, and the square terms are for convenience of linear equations in (4). We should minimize $(E_{\Theta} + E_f + E_{\mathbf{t}})$ to obtain the target sequence. Since the rotation, focal length and translation are independent of each other, we can minimize E_{Θ}, E_f and $E_{\mathbf{t}}$ respectively. It can speed up computation with less memory requirement. Sequentially, without loss of generality, we can set $\omega_{\Theta s} = 1$, $\omega_{fs} = 1$, $\omega_{\mathbf{t}s} = 1$.

We mainly use $\omega_{\Theta m}$ and ω_{fm} to control smoothness since human observers are much more sensitive to rotational vibrations, and only camera rotation and zooming can be compensated without the necessity of recovering depth information. Usually $\omega_{\Theta m} = \omega_{fm} = 100$ can obtain extremely smooth results. Specifically, E_f can be ignored if the focal length is constant.

For translation, since we do not have dense depth information, any kind of warping methods for compensating the translation may result in warping error, which eventually causes image jittering. Therefore, the tradeoff between translational smoothness and warping error; i.e., the translational smoothness weight $\omega_{\mathbf{t}m}$ should be carefully set. We will discuss view warping and analyze warping error in the next section.

2.4 Optimization

The cost functions in (4) are quadratic and independent. Therefore, optimizing them is equal to solving the following linear equation arrays for rotation, focal length, and translation, respectively:

$$\begin{cases}\omega_{\Theta m}\left(\Theta_i^c - 2\Theta_{i+1}^c + \Theta_{i+2}^c\right) = \mathbf{0}, & i = 1, \dots, n-2,\\ \omega_{\Theta s}\left(\Theta_i^c - \Theta_i^o\right) = \mathbf{0}, & i = 1, \dots, n,\end{cases}$$
$$\begin{cases}\omega_{fm}\left(f_i^c - 2f_{i+1}^c + f_{i+2}^c\right) = 0, & i = 1, \dots, n-2,\\ \omega_{fs}\left(f_i^c - f_i^o\right) = 0, & i = 1, \dots, n,\end{cases} \tag{4}$$
$$\begin{cases}\omega_{\mathbf{t}m}\left(\mathbf{t}_i^c - 2\mathbf{t}_{i+1}^c + \mathbf{t}_{i+2}^c\right) = \mathbf{0}, & i = 1, \dots, n-2,\\ \omega_{\mathbf{t}s}\left(\mathbf{t}_i^c - \mathbf{t}_i^o\right) = \mathbf{0}, & i = 1, \dots, n.\end{cases}$$

For n frames, the number of equations of each array is $6n-6$, $2n-2$, and $6n-6$ for the components of rotation, focal length, and translation, respectively. Without loss of generality, here we only consider the optimization of the focal length component. If we denote

$$X = \left[f_1^c, f_2^c, \dots, f_n^c\right]^\top,$$
$$b = \left[0, \dots, 0, \omega_{fs} f_1^o, \dots, \omega_{fs} f_n^o\right],$$

then the corresponding equation array can be expressed as $AX = b$. By applying the least square method, its solution is:

$$X = \left(A^\top A\right)^{-1} A^\top b.$$

If n is large, the computation is very time-consuming. However, we find that although A is a $(2n-2) \times n$ matrix, it has only $5n-6$ non-zero elements, which means that both A and $A^\top A$ are highly sparse. Therefore, we can exploit the sparseness to speed up the computation. Here, we propose a smart method to implement $A^\top A$ by taking advantage of sparseness efficiently.

Generally, for any $m \times n$ matrix $A = (a_{ij})$, we have:

$$A^\top A = \sum_{i=1,\dots,m} A_i^\top A_i$$
$$= \sum_{i=1,\dots,m} \begin{pmatrix} a_{i1} \\ a_{i2} \\ \dots \\ a_{in} \end{pmatrix} (a_{i1}\, a_{i2}\, \dots\, a_{in}). \tag{5}$$

Here A_i denotes the ith row vector of A. If we store only the non-zero elements of A_i, then $A_i^\top A_i$ computes only the multiplication of non-zero elements to eliminate all the redundancy of the zero computation. Therefore, we have chosen to represent sparse matrices using Compressed Row Storage (CRS) format [2]. The CRS format is a general format which makes no assumptions about the sparsity structure of the matrix and does not store any unnecessary elements. The additional cost is to search index in sparse matrix $A^\top A$ due to the compressed structure. We denote N_A and N_r as the average number of non-zero elements in a row of A and $A^\top A$, respectively. We can employ a balanced binary search tree for quick searching. In our implementation, we directly use C++ STL map class, whose complexity is $O(\log_2 N_r)$. Consequently, the total cost of $A^\top A$ operation is $O(mN_A^2 \log_2 N_r)$. Then we can use a sparse linear equation solver to solve it. In our implementation, we adopt the TAUCS Library [22] to solve the sparse linear systems. Table 1 shows the running time of our optimization algorithm. The computation cost is nearly linear to the number of frames being processed. The total computation time for 521 frames is less than one second, which proves the efficiency of our optimization method.

Table 1 Efficiency examination of our optimization algorithm

Frame number	Parameter number	Calc. time (seconds)
11	77	0.015
61	427	0.093
101	707	0.141
290	2030	0.452
521	3647	0.860

3 View warping and warping error

As we know, it is still not robust to obtain dense depth maps or sparse point clouds with existing algorithms. Therefore, we try other ways to sidestep this problem. Warping without accurate geometry information will introduce warping errors, which may cause image jittering or other distracting artifacts. In this section, we make an in-depth analysis of the warping error caused by using planar approximation. We have experimented with two planar warping techniques: slant planar impostors and optimal constant depth (i.e. fronto-parallel plane), constraining warping error in two different ways.

3.1 Warping error

Warping error can be measured by the difference between the warped image and the desired accurate image. Consider a 2D point $\mathbf{x}_p$ on the original image, and its corresponding 3D location $\mathbf{X}_p$. We define warping error as the distance between the warped point $H\mathbf{x}_p$ and $\mathbf{x}_p^c \sim \mathbf{K}^c(\mathbf{R}^c\mathbf{X}_p + \mathbf{t}^c)$ on the desired image:

$$e_p^W = \left\| H\mathbf{x}_p - \mathbf{x}_p^c \right\|, \tag{6}$$

where H is a 3×3 matrix (i.e. homography). $\mathbf{K}^c$, $\mathbf{R}^c$ and $\mathbf{t}^c$ are the motion-compensated camera parameters.

For previous approaches using affine or homography models, constraining warping error is difficult, since it is hard to formulate the accurate target images due to the lack of the information of accurate camera parameters and 3D geometry.

3.2 View warping with slant planar impostors

If scenes are roughly planar, we can estimate the planar transform (or homography) H_i by minimizing the following function:

$$\min \sum_j \left\| H_i \mathbf{x}_{ij} - \mathbf{K}_i^c \left(\mathbf{R}_i^c \mathbf{X}_j + \mathbf{t}_i^c\right) \right\|^2, \tag{7}$$

where $\mathbf{K}_i^c$, $\mathbf{R}_i^c$ and $\mathbf{t}_i^c$ are the intrinsic matrix, rotation matrix and translation vector of the ith warped frame, respectively, $\mathbf{x}_{ij}$ is the 2D image position of the jth feature point on the ith original frame, and X_j is its corresponding 3D location.

The goal of this method is to minimize the total warping error of all the feature points, and this method works well in planar scenes. However, if the scenes deviate from planar structure, it may cause visible image distortions, as demonstrated in Fig. 1 and the supplementary video.

3.3 View warping with optimal constant depth

Suppose that the scene in one frame lies in a plane which is perpendicular to the viewing direction (i.e. constant depth). With the given camera parameters, we can project each pixel to its corresponding 3D location and sequently re-project it onto the target view. Similarly to the warping with slant planar impostor, this warping is also a planar transformation and can be represented by a homography. The optimal depth of this plane is $z_c = 2(z_{\min}^{-1} + z_{\max}^{-1})^{-1}$ (see Appendix for the detail), where $[z_{\min}, z_{\max}]$ is the depth range. In step 1, we can obtain a set of spare 3D feature points. According to the depth distribution of these features, we can reliably estimate $z_{\min}$ and $z_{\max}$.

Although the total warping error is a little larger than the slant planar impostors, the resulting artifacts are usually less objectionable, perhaps because we are much more sensitive to seeing unnatural image distortions and the optimal constant depth method does not cause this annoying artifact. Therefore, we prefer to use it rather than planar impostors as default for view warping.

3.4 Warping error analysis

Warping error also results in visual jittering and therefore should be contained in the measurement of video jittering. By making an analysis of the warping error, we can get the solution of optimally setting the weights $\omega_{\mathbf{t}m}$ and $\omega_{\mathbf{t}s}$.

For the ith frame, its original translation is $\mathbf{t}_i^o$, and its compensated translation is $\mathbf{t}_i^c$. Therefore, the changing translation vector $\mathbf{t} = (t_x, t_y, t_z)$ can be computed by $\mathbf{t} = \mathbf{t}_i^c - \mathbf{t}_i^o$. For point p, its 3D homogeneous coordinate is denoted as $(x, y, 1, 1/z)$, where z is its depth value. If we use z_c to estimate its depth, the warping error will be (see Appendix for the detail):

$$\mathbf{e}^W = f\left(\frac{1}{z} - \frac{1}{z_c}\right)(t_x - x t_z, t_y - y t_z)^\top. \tag{8}$$

Then we have:

$$\|\mathbf{e}^W\| = f\left|\frac{1}{z} - \frac{1}{z_c}\right| \sqrt{(t_x - x t_z)^2 + (t_y - y t_z)^2}. \tag{9}$$

Because the camera focal angle is usually small (typically less than 35 degrees) and most interesting points are close to the image center (i.e. x and y is small), so, for convenient computation, we approximate it as:

$$\|\mathbf{e}^W\| \approx f\left|\frac{1}{z} - \frac{1}{z_c}\right| \cdot \|\mathbf{t}\| = f\left|\frac{1}{z} - \frac{1}{z_c}\right| \left\|\mathbf{t}_i^c - \mathbf{t}_i^o\right\|. \tag{10}$$

Another part of video jittering results from the vibration of the compensated translation. For a point p, its image position in the compensated image i is $\mathbf{x}_i^c$, and its image position in the compensated image $i+1$ is $\mathbf{x}_{i+1}^c$. Then the displacement $\mathbf{d}^c$ can be defined as $\mathbf{d}^c = \mathbf{x}_{i+1}^c - \mathbf{x}_i^c$. If we denote $\mathbf{t'}_i = \mathbf{t}_{i+1}^c - \mathbf{t}_i^c$, we have $\mathbf{d}^c = \frac{f}{z}(\mathbf{t'}_x - x\mathbf{t'}_z, \mathbf{t'}_y - y\mathbf{t'}_z)^\top$. In order to minimize jittering, the displacement needs to be constant, i.e. $(\mathbf{d}^c)' = \frac{\partial \mathbf{d}^c}{\partial i} = 0$. Denoting $\mathbf{a} = \frac{\partial \mathbf{t'}_i^c}{\partial i} = \frac{\partial \mathbf{t}_c^{i\,2}}{\partial i^2}$, i.e. the second derivative of translation, we have:

$$\left(\mathbf{d}^c\right)' = \frac{f}{z}(a_x - x a_z, a_y - y a_z)^\top, \tag{11}$$

where a_x, a_y and a_z are the three components of the vector $\mathbf{a}$.

Similarly to (10), for $\mathbf{a} = \mathbf{t}_{i-1}^c - 2\mathbf{t}_i^c + \mathbf{t}_{i+1}^c$, we have:

$$\left\|\left(\mathbf{d}^c\right)'\right\| \approx f\frac{1}{z}\|a\| = f\frac{1}{z}\left\|\mathbf{t}_{i-1}^c - 2\mathbf{t}_i^c + \mathbf{t}_{i+1}^c\right\|. \tag{12}$$

In order to constrain the jittering, we should minimize both $\|\mathbf{e}^W\|^2$ and $\|(\mathbf{d}^c)'\|^2$. Then the cost function can be formulated as:

$$E_{jit} = \left(\frac{1}{z}\right)^2 \left\|\mathbf{t}_{i-1}^c - 2\mathbf{t}_i^c + \mathbf{t}_{i+1}^c\right\|^2 + \left(\frac{1}{z} - \frac{1}{z_c}\right)^2 \left\|\mathbf{t}_i^c - \mathbf{t}_i^o\right\|^2. \tag{13}$$

When weight ratio $\omega_{\mathbf{t}m} : \omega_{\mathbf{t}s} = z^{-1} : |z^{-1} - z_c^{-1}|$, the cost function E_{jit} is equal to the cost function $E_{\mathbf{t}}$. In this case, the solution not only minimizes $E_{\mathbf{t}}$ but also E_{jit}. That is, the best smoothness effect can be achieved.

If z is close to $z_{\min}$ (or $z_{\max}$), the corresponding optimal value is close to $z_{\min}^{-1} : |z_{\min}^{-1} - {z_c}^{-1}|$ (or $z_{\max}^{-1} : |z_{\max}^{-1} - {z_c}^{-1}|$). Especially under the optimal constant depth $z_c =$

$2(z_{\min}^{-1} + z_{\max}^{-1})^{-1}$, they become $2z_{\min}/(z_{\max} - z_{\min})$ and $2z_{\max}/(z_{\max} - z_{\min})$, respectively. Considering that z is distributed in $[z_{\min}, z_{\max}]$, the optimal $\omega_{\mathbf{t}m} : \omega_{\mathbf{t}s}$ can be computed by averaging them as follows:

$$(\omega_{\mathbf{t}m} : \omega_{\mathbf{t}s})_{\text{opt}} = \frac{2z_{\min}/(z_{\max} - z_{\min}) + 2z_{\max}/(z_{\max} - z_{\min})}{2} = (z_{\max} + z_{\min})/(z_{\max} - z_{\min}), \quad (14)$$

which becomes 1 if $z_{\max} \gg z_{\min}$. Therefore, if $z_{\min}$ and $z_{\max}$ are unknown, we usually set $\omega_{\mathbf{t}m} : \omega_{\mathbf{t}s} = 1$ as the default in (4).

4 Results

We have tested our approach with a variety of video sequences taken by a hand-held video camera. All experiments are carried out on a PC with Intel Pentium IV 2.4 GHz CPU with 1 GB memory. Appealing results are obtained in our experiments.

The current processing speed of our SFM step is relative slow, which typically takes about two minutes to process 100 frames. There is a lot of room for improvement in our unoptimized research code since we employ a standard SFM algorithm, whose running time was not our major concern in this paper. Recently, Nistér [16] proposed a fast SFM method for video with real-time performance. It would be especially beneficial to our system. Our stabilization optimization step is quite quick, which only takes about 0.14 seconds for 100 frames (Table 1). Our view warping can perform near real-time (about 0.1 second per frame with 640×480 resolution).

The evaluation and comparison of video stabilization algorithms is a difficult task, since there is no ground truth available for real sequences and the standard of evaluation is also difficult to formulate. Especially, our method employs a 3D perspective camera model, and previous approaches employ affine/homography models. Perhaps, perceptual judgment of stabilization is the best option to evaluate video stabilization algorithms aimed at the human observer. For comparison, we also implement the stabilization method proposed in [13] with both affine and homography motion models. The radius of Gaussian filtering range is 10 frames (i.e. $k = 10$ in [13]). The algorithms are shown in Table 2, where the algorithms 3 and 4 are our algorithms with different warping strategies, i.e. optimal constant depth and slant planar impostors, respectively.

Table 2 Stabilization algorithms

Algorithm	Model	Filtering	Compensation
1	affine	Gaussian	affine
2	homography	Gaussian	homography
3	3D camera	linear opt.	opt. const. depth
4	3D camera	linear opt.	planar impostors

Table 3 Smoothness evaluation on the four examples shown in this paper. A small value indicates that the stabilized result is more smooth

Smoothness Evaluation	Algorithm			
	1	2	3	4
Example A	2.37	2.39	2.02	1.97
Example B	1.01	0.99	0.59	0.57
Example C	1.21	1.20	1.23	1.16
Example D	5.10	5.15	3.39	2.58

We selected four video sequences (which we call A, B, C and D) and 7 persons to perform a user test. For each example, there are four stabilization results obtained by four different algorithms (shown in Table 2). Each user independently evaluates the stabilized results. Except the example C for which four algorithms produce comparable results, all users consider the effects with algorithms 3 and 4 significantly outperform those with other two algorithms. In order to quantitatively evaluate the stabilized results, we measured the smoothness of the tracked feature points in the stabilized sequences by computing the average of the second-order differentials of the trajectories of all feature points, i.e. $\|2\mathbf{x}_t - \mathbf{x}_{t-1} - \mathbf{x}_{t+1}\|$. Table 3 reports the detailed statistics. Again, the smoothness errors with algorithms 3 and 4 are much smaller than those with their counterparts, for all examples except the example C. This is actually in accordance with the user study. The reason is that most video sequences contain large camera translations and significant depth variations in the scenes. It should be noted that a 2D affine or homography model works well in many examples in which the scenes undergo planar perspective transformations. In these cases, our stabilized results are comparable with results by previous methods.

In order to illustrate obvious differences of view warping with optimal constant depth and slant planar impostors, we generate a simulation sequence. The reason is that real videos usually do not contain exaggerated shaky translations. In addition, exaggerated shaky motion may cause serious motion blur, which is difficult for tracking. We construct a scene which contains two planes. Figure 1 shows the different results employing slant planar impostors and optimal constant depth with different cost weights, in which the depth of the front one is 100, and that of the back one is 300. The sequence of translation which is recovered from a real video sequence is shaky, and is smoothed with two different cost weights $\omega_{\mathbf{t}m} = 2$ and $\omega_{\mathbf{t}m} = 100$ ($\omega_{\Theta m} = 100$, the focal length is constant), as shown in Fig. 1(a). As can be seen, if $w_{\mathbf{t}m}$ is set larger, the compensated translational parameters get smoother (the curves get smoother). However,

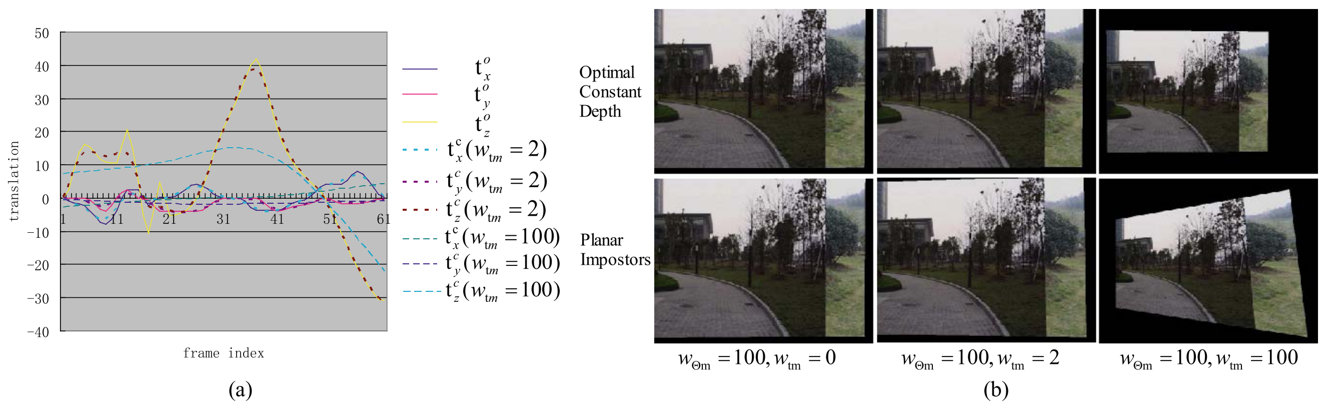


Fig. 1 The comparison of view warping in two different ways with different cost weights. (**a**) The original translational motion parameters and stabilized ones for the simulation sequence. (**b**) The comparison of the stabilized results of view warping in two different ways (optimal constant depth and slant planar impostors), with different cost weights

due to the warping error, it does not mean that the compensated video will certainly become more stable. Figure 1(b) shows the motion-compensated frames, by optimal constant depth and slant planar impostors with different cost weights. The optimal value of $\omega_{\mathbf{t}m}$ is 2 in this sequence according to (14); larger values cannot obtain smoother result. On the contrary, they can cause larger warping errors and unnatural artifacts (far objects become very shaky, etc.), as demonstrated in the supplementary video. It should be noted that view warping with slant planar impostors causes obvious image distortions in the case of large warping error, whereas the result of view warping with optimal constant depth is much more natural. However, if the scene can be dominated by a slant plane, view warping with slant planar impostors will obtain more accurate result. Figure 2 shows an example for a simulated world consisting of a single inclined plane under similar motions. We set the cost weights $\omega_{\mathbf{t}m} = 100$, $\omega_{\Theta m} = 1000$ (focal length is constant). Since the scene is a single slant plane, it can be exactly represented by plane impostors. Therefore, for this example, view warping with planar impostors can obtain the best result, without introducing warping error (Fig. 2(c)). In contrast, view warping with optimal constant depth will cause warping error (as shown in Fig. 2, (b) and (d)), so that the jittering cannot be fully eliminated.

We examine four real sequences with various scenes and camera motion taken by a hand-held camera. All these sequences contain lots of vibrations and depth variation. In these examples, the weights are all set as $\omega_{\Theta m} = \omega_{fm} = 100$ and $\omega_{\mathbf{t}m} = 1$, if without mentioning. Since human beings are much more sensitive to distracting effect in video sequences than in still images, please refer to the accompanying videos for the detail effects. In order to demonstrate the sequence effects of smoothness in still images, we mark all the superior tracks, where the tracks are showed by white lines, their image positions in the current frame are indicated by

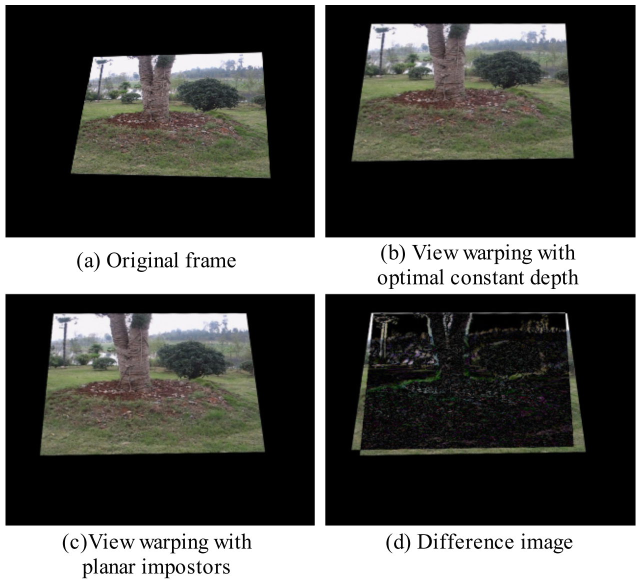

(a) Original frame
(b) View warping with optimal constant depth
(c)View warping with planar impostors
(d) Difference image

Fig. 2 A simulated example consisting of a single inclined plane. (**a**) The original translational frame. (**b**) The compensated frame by warping with optimal constant depth. (**c**) The compensated frame by warping with planar impostors. (**d**) The difference image of (**b**) and (**c**). Due to the warping error, (**b**) has a significant displacement compared with (**c**)

green crosses, and the positions in neighbor frames by red crosses.

As shown in Fig. 3, sequence A consists of complex scene with tree leaves in the front and a building in the distance. From Fig. 3(d), we notice that the building compensated by affine and homography models is slanted in different ways within 0.5 seconds, which results in obvious jittering in target video sequences. Both our methods demonstrate very stable effects in this sequence.

Sequence B was taken in our campus, and is a roughly planar structure. Only the track line and the current feature points are marked in Fig. 4. From Fig. 4(a), the routes of the

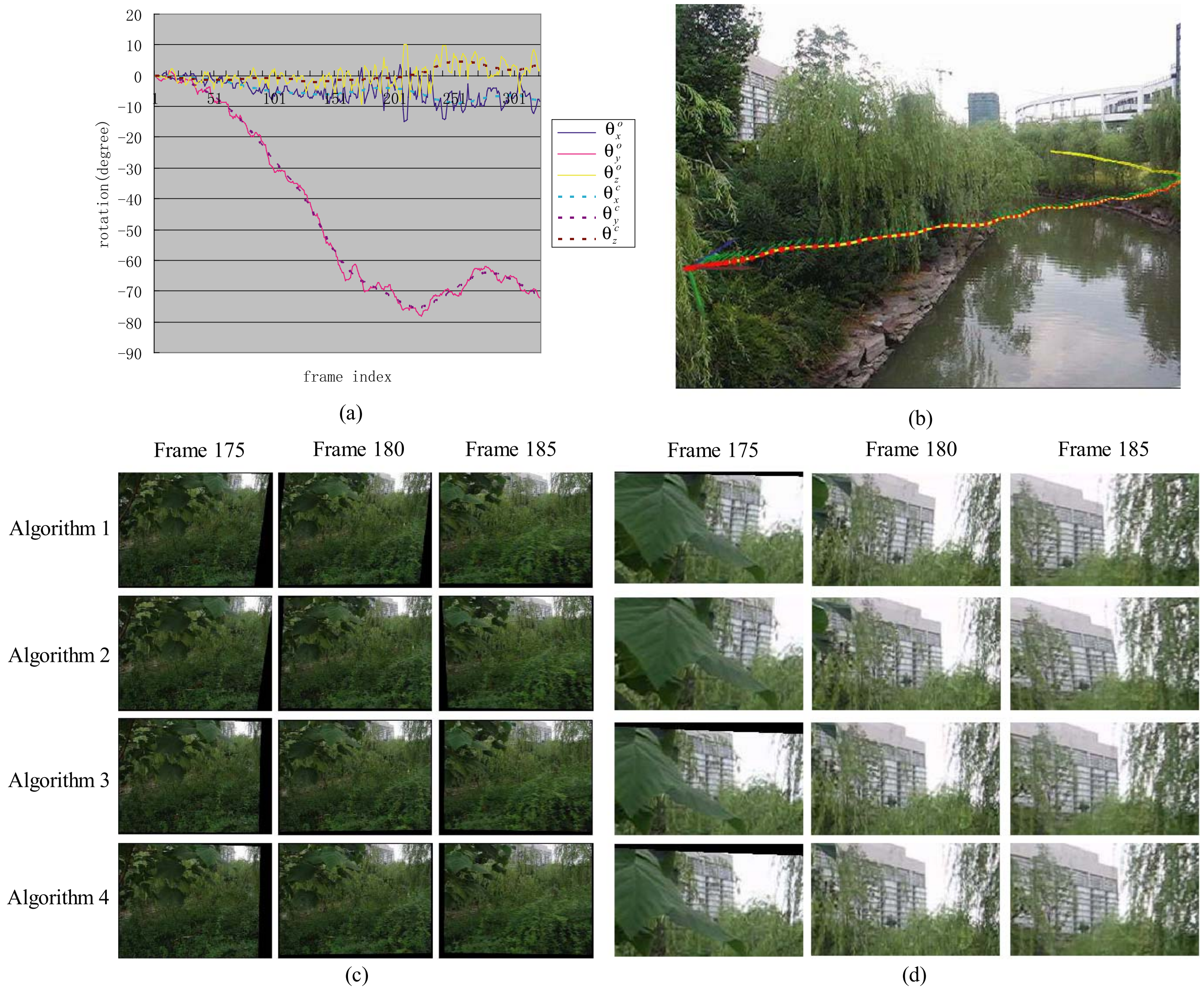


Fig. 3 Stabilization of the sequence A. (**a**) The recovered rotational parameters and the stabilized ones. (**b**) One of the original frames with the recovered camera trajectory. (**c**)Three frames of stabilized sequences with four algorithms (refer to Table 2). (**d**) The magnified snapshots of (**c**) around the building area

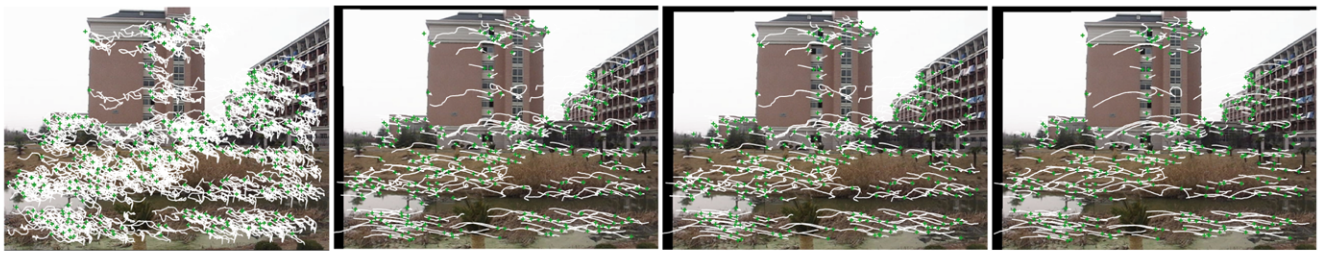


Fig. 4 The stabilization of the sequence B with different algorithms: (**a**) shows the original frame; (**b**), (**c**) and (**d**) show one stabilized frame of the algorithms 1, 2, and 4, respectively

feature points in the original sequence are very wandering, which results in very dense white line folding since the camera motion is unstable. The stabilized feature tracks based on affine model or homography model are still a little wandering, while our model method demonstrates a very smooth route. Carefully comparing the video sequences of Fig. 4, (b) and (c), we can find that homography model is prior to affine model in planar scenes. This is because that homog-

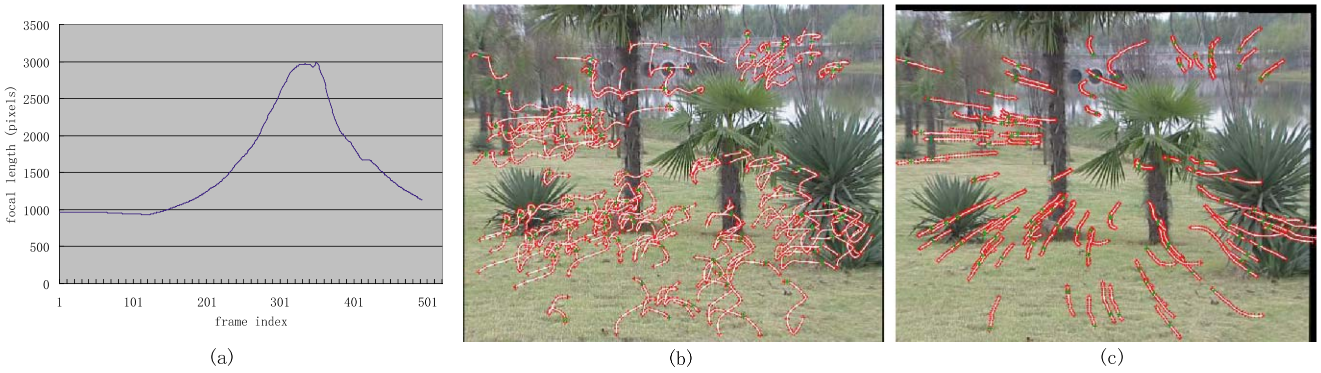


Fig. 5 The results of the sequence C: (**a**) shows the recovered focal length. (**b**) An original frame with feature tracks. (**c**) The stabilized image with feature tracks applying algorithm 3

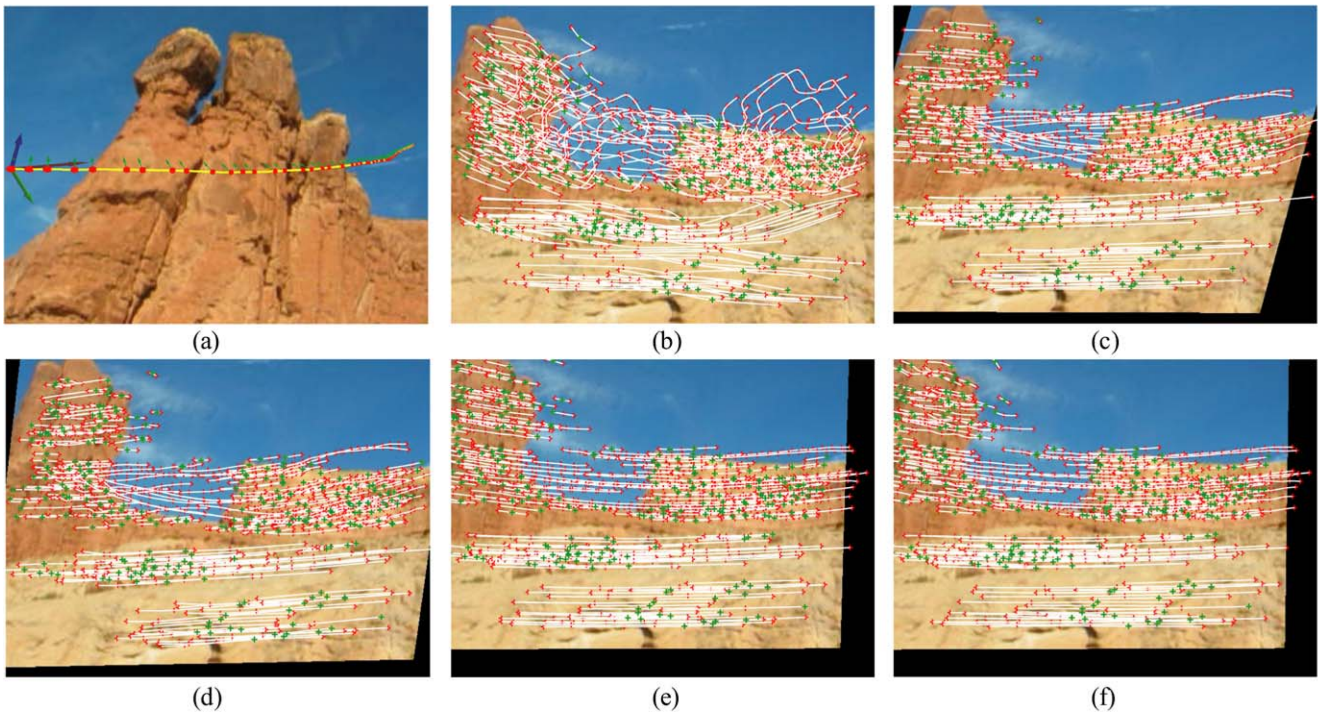


Fig. 6 The results of the sequence D: (**a**) shows the recovered camera trajectory; (**b**) shows one original frame; and (**c**), (**d**), (**e**), and (**f**) show the different stabilization results with algorithms 1, 2, 3, and 4, respectively

raphy model can accurately describe the image motion in planar scenes.

Our camera tracking method can handle variable focal lengths in a zooming sequence. Our stabilization algorithm demonstrates robust effects in video sequence C, in which the focal length varies in large range as shown in Fig. 5(a), while an original and its motion compensated frame are shown in Fig. 5, (b) and (c), respectively, with the illustrative feature tracks.

Sequence D is a mountain area (downloaded from http://iss.bu.edu/litvin/stabilization/). From the trajectory of video camera shown in Fig. 6(a), we can find that the translation distances are very uneven. Both affine and homography models are obviously distorted, especially in the end of the sequence where some objects are very close to the camera. Notice the left peak area in Fig. 6, (c) and (d), where the feature points are not proportionally spaced in the sequence, while our method is much better as shown in Fig. 6, (e) and (f). However, for our methods, the result of applying optimal constant depth is appreciably better than slant planar impostors. The reason is that the scenes deviate from planar, especially in the later part of the sequence. We compare the results of different $\omega_{\Theta m}$ in Fig. 7(b), where the larger value produces smoother results but larger missing regions.

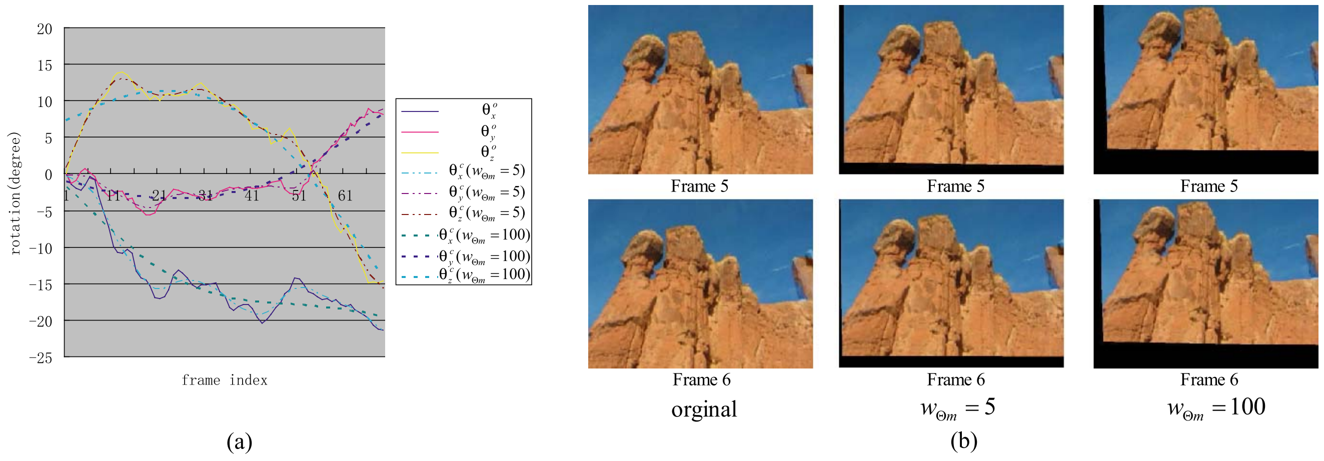


Fig. 7 Results with different cost weights. (**a**) The recovered and stabilized rotational motion parameters for sequence D with different cost weights. (**b**) The original and motion-compensated frames. The *middle column*: $\omega_{\Theta m}=5$, $\omega_{\mathbf{t}m}=1$. The *right column*: $\omega_{\Theta m}=100$, $\omega_{\mathbf{t}m}=1$

5 Conclusion and discussion

We have proposed a novel method to stabilize video sequences based on a 3D perspective camera model without recovering dense depth maps. The video stability is optimized by balancing the smoothness and similarity, which is related to the rotation, zooming, and translation components with suitable weights. Based on a 3D perspective camera model, the depth relative motion, i.e. camera translation, and depth irrelative motion, i.e. camera rotation and zooming, are separated. Consequently, the unwanted motion, especially camera rotation and zooming, can be smoothed efficiently without introducing warping error, and translation is also smoothed under control by setting optimal weight. By taking advantage of the sparseness of the linear system, our optimization process is very efficient.

We have experimented with two warping methods with planar geometry approximation, i.e. slant planar impostors and optimal constant depth method, which have obvious advantages over traditional stabilization methods, such as employing affine or homography models. Our stabilization method can produce high quality stabilized video sequences, and is very useful for some high-end applications, such as film-making and TV.

Our approach employs a 3D perspective camera model, and thus highly relies on the accuracy of the Structure-from-Motion (SFM) results. Till now obtaining accurate camera motion parameters for long video sequences is still quite challenging [7], our approach may not work well if the SFM algorithm fails to get precise results. We expect to address this problem in our future work.

Acknowledgements We would like to thank the anonymous reviewers for their constructive comments to improve the paper, and also thank Prof. Xiaofei He and Wei Chen for their enormous help in revising this paper. This work is supported by NSF of China (No.60633070), 973 program of China (No.2009CB320802), and 863 program of China (No. 2007AA01Z326).

Appendix

For pixel p in the ith frame, its 3D location is $(xz, yz, z)^\top$ in the coordinate system of the original camera, where z is its depth value. So its homogeneous coordinate is $(x, y, 1, 1/z)^\top$ in the coordinate system of the original camera. For the ith frame, its original translation is $\mathbf{t}_i^o$, and its compensated translation is $\mathbf{t}_i^c$. Therefore, the changing translation vector $\mathbf{t}=(t_x, t_y, t_z)$ can be computed: $\mathbf{t}=\mathbf{t}_i^c-\mathbf{t}_i^o$. From the original camera to the motion-compensated camera, its 3D position becomes $(xz+t_x, yz+t_y, z+t_z)^\top$, i.e. homogeneous coordinate becomes $(\frac{xz+t_x}{z+t_z}, \frac{yz+t_y}{z+t_z}, 1, 1/(z+t_z))^\top$. Then the displacement of p in the 2D image can be computed as follows:

$$d=\left(f\frac{xz+t_x}{z+t_z}, f\frac{yz+t_y}{z+t_z}\right)^\top-(fx, fy)^\top$$
$$=\left(f\frac{t_x-xt_z}{z+t_z}, f\frac{t_y-yt_z}{z+t_z}\right)^\top.$$

For convenience, we replace $z+t_z$ with z by simply offsetting the coordinate, hence

$$d=\left(f\frac{t_x-xt_z}{z}, f\frac{t_y-yt_z}{z}\right)^\top. \quad (15)$$

If we use the constant depth z_c for each pixel, instead of its true depth value, the estimated displacement becomes

$$d^W=\left(f\frac{t_x-xt_z}{z_c}, f\frac{t_y-yt_z}{z_c}\right)^\top. \quad (16)$$

Therefore, the warping error is

$$\mathbf{e}^W = d - d^W = f\left(\frac{1}{z} - \frac{1}{z_c}\right)(t_x - xt_z, t_y - yt_z)^\top.$$

We assume the depths of the scene are in the range of $[z_{\min}, z_{\max}]$, i.e. $z_{\min} \le z \le z_{\max}$. Therefore, when $z_c = 2(z_{\min}^{-1} + z_{\max}^{-1})^{-1}$, the upper bound of warping error is minimal:

$$\begin{aligned}\|\mathbf{e}^W\| &= f\left|\frac{1}{z} - \frac{1}{z_c}\right|\sqrt{(t_x - xt_z)^2 + (t_y - yt_z)^2} \\ &\le \frac{1}{2} f\left(\frac{1}{z_{\min}} - \frac{1}{z_{\max}}\right)\sqrt{(t_x - xt_z)^2 + (t_y - yt_z)^2}.\end{aligned}$$

References

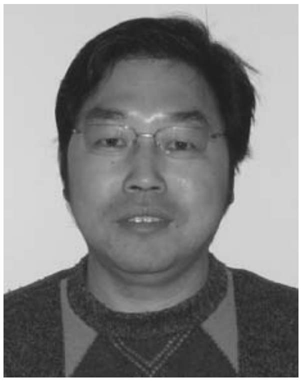
1. 2d3: http://www.2d3.com
2. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., der Vorst, H.V.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd edn. SIAM, Philadelphia (1994)
3. Bhat, P., Zitnick, C.L., Snavely, N., Agarwala, A., Agrawala, M., Curless, B., Cohen, M., Kang, S.B.: Using photographs to enhance videos of a static scene. In: Kautz, J., Pattanaik, S. (eds.) Rendering Techniques 2007, Proceedings Eurographics Symposium on Rendering, pp. 327–338. Eurographics (2007)
4. Buehler, C., Bosse, M., McMillan, L.: Non-metric image-based rendering for video stabilization. In: CVPR, pp. 609–614 (2001)
5. Davis, L.S., Bajcsy, R., Herman, M.: RSTA on the move. In: Proceedings of the ARPA Image Understanding Workshop, pp. 435–456 (1994)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: CommACM, vol. 24, p. 381–395 (1981)
7. Fitzgibbon, A., Zisserman, A.: Automatic camera tracking. In: Shah, M., Kumar, R. (eds.) Video Registration, pp. 18–35. Kluwer, Dordrecht (2003)
8. Hansen, M., Anadan, P., Dana, K., van de Wal, G., Burt, P.: Realtime scene stabilization and mosaic construction. In: Proc. IEEE Image Understanding Workshop, pp. 457–463 (1994)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
10. Litvin, A., Konrad, J., Karl, W.: Probabilistic video stabilization using Kalman filtering and mosaicking. In: IS&T/SPIE Symposium on Electronic Imaging, Image and Video Communications, pp. 663–674 (2003)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
12. Mark, W.R., McMillan, L., Bishop, G.: Post-rendering 3D warping. In: SI3D, pp. 7–16, 180 (1997)
13. Matsushita, Y., Ofek, E., Tang, X., Shum, H.Y.: Full-frame video stabilization. In: CVPR (1), pp. 50–57 (2005)
14. Morimoto, C., Chellappa, R.: Fast 3D stabilization and mosaic construction. In: CVPR, pp. 660–665 (1997)
15. Morimoto, C., Chellappa, R.: Evaluation of image stabilization algorithms. In: International Conference on Acoustics, Speech and Signal Processing, vol. 5, pp. 2789–2792 (1998)
16. Nistér, D.: Preemptive ransac for live structure and motion estimation. Mach. Vis. Appl. **16**(5), 321–329 (2005)
17. Pilu, M.: Video stabilization as a variational problem and numerical solution with the Viterbi method. In: CVPR (1), pp. 625–630 (2004)
18. Pollefeys, M., Gool, L.J.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comput. Vis. **59**(3), 207–232 (2004)
19. REALVIZ: http://www.realviz.com
20. Shade, J., Gortler, S.J., wei He, L., Szeliski, R.: Layered depth images. In: SIGGRAPH, pp. 231–242 (1998)
21. Snavely, N., Seitz, S.M., Szeliski, R.: Photo-tourism: exploring photo-collections in 3D. ACM Trans. Graph. **25**(3), 835–846 (2006)
22. TAUCS: http://www.tau.ac.il/~stoledo/taucs/
23. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: Workshop on Vision Algorithms, pp. 298–372 (1999)
24. Yao, Y., Burlina, P., Chellappa, R., Wu, T.: Electronic image stabilization using multiple visual cues. In: Proceedings of International Conference on Image Processing, pp. 191–194 (1995)
25. Zhang, G., Qin, X., Hua, W., Wong, T.T., Heng, P.A., Bao, H.: Robust metric reconstruction from challenging video sequences. In: CVPR (2007)
26. Zhang, Z.: Determining the epipolar geometry and its uncertainty: A review. Int. J. Comput. Vis. **27**(2), 161–195 (1998)
27. Zhu, Z., Xu, G., Yang, Y., Jin, J.S.: Camera stabilization based on 2.5D motion model estimation and inertial motion filtering. In: IEEE Conference on Intelligent Vehicles, pp. 329–334 (1998)

Guofeng Zhang received his BS degree in Computer Science from Zhejiang University, P.R. China, in 2003. Currently, he is a PhD candidate in computer science at State Key Laboratory of CAD&CG, Zhejiang University. His main research interests include camera tracking, 3D reconstruction, augmented reality and video enhancement.

Wei Hua received the BS degree in Biomedical Engineering from Zhejiang University in 1996, and the PhD degree in Applied Mathematics from Zhejiang University in 2002. Currently, he is an Associate Professor of the State Key Laboratory of CAD&CG of Zhejiang University. His research interests include real-time simulation and rendering, virtual reality and software engineering.

Xueying Qin received her PhD from Hiroshima University of Japan in 2001, and MS and BS from Zhejiang University and Peking University in 1991 and 1988, respectively. She was an Associate Professor of the State Key Laboratory of CAD&CG of Zhejiang University. Currently, she is a Professor of School of Computer Science & Technology, Shandong University, P.R. China. Her main research interests are augmented reality, video-based rendering, and photo-realistic rendering.

Yuanlong Shao received his BS degree in Computer Science from Zhejiang University, P.R. China, in 2007. Currently, he is a Master candidate in Computer Science at State Key Laboratory of CAD&CG, Zhejiang University. His main research interests include camera tracking, feature matching, augmented reality and video enhancement.



Hujun Bao received his Bachelor and PhD in Applied Mathematics from Zhejiang University in 1987 and 1993. He is currently the director of State Key Laboratory of CAD&CG of Zhejiang University. He is also the principal investigator of the virtual reality project sponsored by Ministry of Science and Technology of China. His research interests include realistic image synthesis, real-time rendering technique, digital geometry processing, field-based surface modeling, virtual reality and video processing.