A Global Approach to Fast Video Stabilization

Lei Zhang, *Member, IEEE,* Qian-Kun Xu, and Hua Huang, *Member, IEEE*

***Abstract*—This paper presents a novel formulation of video stabilization, by directly solving for optimal image warps toward stabilized sequence. With the estimated shaky motion via long or short feature trajectories, our approach encodes another two steps, motion compensation and image warping, into a single global optimization process, rather than operating as two individual steps. This process is done with only positions of embedded mesh vertices as common variables. Spatial and temporal coherence is therein re-formulated with similarity invariant representation of motion trajectories and intra-(and inter-)frame consistency of similar transformations with respect to mesh vertices. Such a one-shot formulation converts video stabilization into a quadratic energy minimization problem defined for image warps, thus can be efficiently resolved by using robust solver for sparse linear system. Experimental results demonstrate the flexibility and efficiency of our approach to produce visually plausible stabilization effects on a variety of videos.**

***Index Terms*—Video stabilization, global optimization, image warping.**

I. Introduction

THE popularization of digital devices with video cameras makes it possible to obtain videos whenever and wherever, thus achieving rapid and sustained rise of video quantity. However, visual quality of captured videos is varied and highly affected by uncertainties related to environmental setting, photographic skills, *etc*. Typically, it is hard for an amateur user to steadily hold a camera in the whole capture process, thus generating noticeable shaking effect in the output video sequence. So video stabilization becomes an important and exigent problem in video processing [1], [2], [3], [4], which strives for eliminating or reducing the visual shake for a smooth display. Some hardware-based solutions have been applied to tackle the annoying shake, like tripods, dollies, steady-cams, which are usually of heavy-load and inflexible in use for casual capture. While camera's in-built optical or electronic stabilizers only resist high-frequency jitter, they might get stuck in the occurrence of low-frequency disturbance like video shot by a walking person. On the contrary, recent contender methods treat video stabilization as a task of image post-processing, which is free of any hardware and able to deal with a wide range of types of shaky motion. Such methods do not necessarily pursue physically accurate motion plan, but seek to transform the original video frames for steady appearance as a whole sequence. In this paper, we will revisit image post-processing stabilization and present a novel approach for fast video stabilization.

Image post-processing stabilization usually follows a common three-step scheme: shaky motion estimation, smooth motion compensation and image warping. Shaky motion estimation is critical to infer the original motion trend, usually described by feature trajectories [5], [6], [7], optical flows [1], [8], [3], or parametric transformations [9], [10], [11]. Then, most of existing methods [6], [7], [12], [13], [14], [15] take the rest into two individual procedures, i.e., solving smooth motion and seeking suitable image warping afterward to make regenerated frames accommodating the smoothed motion, and build the self-governed formalism for respective optimal solutions. Such two-shot treatment has justifiable rationales, but also brings about flaws in two aspects: i) The amount of compensation to the smoothed motion might cause irreversible artifacts in image warping, because the latter has to obey the compensated motion for the purpose of stability. By irreversible we here mean warping effect can only be presented in hindsight by adjusting motion compensation ahead; ii) It usually casts stabilization as at least two separate optimization problems, which has to assume considerable computation burden to obtain the final stabilization results.

However, we argue that motion compensation and image warping are tightly coupled by a potential stabilization scheme, whereupon the two steps can be mixed as a single process toward optimally smoothed and warped appearance of video frames simultaneously. In this paper, we present a global approach for combining the two steps of motion compensation and image warping into a single one-shot optimization process. Actually, our technique is partially inspired by the recent works of [7] that uses rigid enforcement for consistent motion compensation and [11] that totally employs spatially variant homographies for the two steps, but we instead opt for similarity invariant constraints and spatially variant similar warps, both encoded by vertices of grid mesh embedded in video frames (see Fig. 1), which can achieve more flexible and efficient stabilization performance.

Our contribution to the state-of-the-art is a novel formulation on video stabilization, which is globally posed as the minimization on a quadratic functional. The sole task is to solve the optimal image warps parameterized by positions of grid mesh vertices, to be more direct and hence more economic. Experimental results show the flexibility and efficiency of our one-shot formulation in dealing with stabilization on a variety of videos.

II. Related work

Recent years have witnessed a marked progress on image post-processing based video stabilization, and different lines of

L. Zhang, Q.-K. Xu and H. Huang are with the Beijing Key Lab of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China. E-mail: huahuang@bit.edu.cn

Manuscript received XXXX XX, 20XX; revised XXXX XX, 20XX. This work was partly supported by the grants from the National Natural Science Foundation of China (No. 61133008, 61425013 and 61472035).

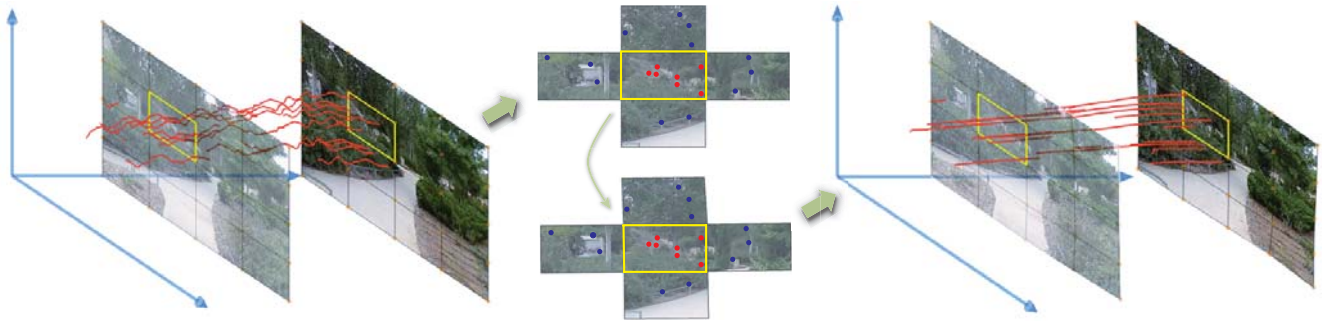


Fig. 1. Video stabilization of shaky motion (left) by using a global formulation to optimize image warps (middle) for steady motion (right), realized in a spatially and temporally coherent manner with positions of grid mesh vertices as common and sole variables in the formulation.

research come up by following the aforementioned three steps. For shaky motion estimation, an ideal solution is to recover the real three-dimensional (3D) camera path, which typically assumes the structure-from-motion (SfM) technique [13], [15], [16] or equips some depth sensors [17] for 3D reconstruction. It is known that SfM still remains costly and brittle for many challenging videos, and depth sensor like Kinect does not suit outdoor scenes. Hence, a large body of research leverages two-dimensional (2D) feature tracking to form continuous trajectories [5], [6], [14], [18] or dense optical flow [1], [8], [3], which indeed explicitly represent the shaky motion in the videos. Obviously, these methods often depend on truly obtaining long trajectories or reliable optical flow from sequential changes of video frames, and probably suffer technical failures with interruption caused by occlusion or textureless regions. Alternatively, a few methods build the shaky motion by global [9], [10], [19] or local [11], [20] parametric transformations between adjacent frames, while video stabilization amounts to optimizing the intrinsic parameters for smoothed transformations. Such methods do not confine themselves within any trajectory space and are flexible for dealing with a wide range of videos. But on the other hand, they usually incur noticeable distortion due to using accumulated transformations rather than continuous trajectories. In this paper, our approach will borrow the flexibility of local transformations but still consult reliable trajectories for more efficient video stabilization.

With the estimated shaky motion, smooth motion compensation is applied to filter the estimated motion, based on trajectories, optical flows or parametric transformations for dampening high/low-frequency jitter. It has been pointed out that such filtering favors the fashion that enables preservation on the geometric relationships among individual trajectories or transformations [6]. If camera path is firstly extrapolated by using SfM, then relationships of trajectories are engaged by the camera path as well as projection matrices. Thus, the task is changed to smooth the camera path or parameters of projection matrix of camera imaging [13], [15]. Otherwise when directly using trajectories, Liu et al. [6] employ low-rank constraints on the matrix formed by trajectories, and perform low-pass filtering on the subspace basis. This method is able to provide aggressive stabilization effects, but heavily relies on long trajectories covering the filter windows. To relax the unfavorable conditions, virtual trajectories can be added to complete short ones like using principles of epipolar geometry [14], or particle filtering is adopted to encourage competent motion for stabilization [18]. Besides, Wang et al. [7] employ Bézier curves to fit the trajectories, and preserve their offsets by enforcing spatial rigidity in the filtering. This method can deal with videos having both long and short trajectories, and also achieves fast computation, but possibly incurs much deviation from intended motion due to over-smoothness by using Bézier curves as the shake-free guidance. Besides, this method cannot well stabilize videos with zooming or fast rotation motion due to the use of rigid enforcement. Actually, smoothing inter-frame global parametric transformations, e.g., homography [9] or affine transformation [10], [19], can give a simple yet robust means to control the individual trajectory deviation for motion filtering. But there usually exists geometric distortion if inter-frame transition can not be modeled by a single global transformation, especially for videos with large parallax or non-in-plane motion. So Wang et al. [20] exploit the plane structure directly from videos, and perform motion compensation using respective homography corresponding to each plane. Obviously, this method needs correct plane detection, otherwise resulting in poor stabilization effects. Liu et al. [11] employ spatially variant homographies to locally approximate the underlying transformation, which avoids plane structure detection and is able to compensate a variety of shaky motion. But this method totally throws potential good trajectories away, and turns to iterative adjustment on homography weights to regulate the deviation from the intended motion trend, which is slow in computation. In this paper, we will utilize the trajectories to assist motion approximation by local similar transformations.

To obtain the stabilized video, frames are transformed by image warping to follow the compensated smooth motion. To preserve the original appearance, 'as-similar-as-possible' method as in [13] is widely used to keep the warping distortion with deviation of uniform scale. This method is technically simple to implement, but does not obey cinematograhic principles in reality, and usually performs poorly in large textureless regions. So Zhou et al. [15] resort to plane-based homographies based on 3D plane detection as a supplement for warping textureless regions. Considering frangibility of plane structure detection, Liu et al. [11] resort to multiple local homographies for image warping, which can generate

visually pleasing stabilization effects for many challenging videos. In this paper, our approach will follow the 'as-similar-as-possible' principle, but enforce more constraints for spatial continuity, and take up motion compensation as a part of realizing optimal image warps for the desirable stabilization effects.

III. Approach

The key idea behind our approach is to directly find a set of optimal image warps, with appropriate parametrizations, which admit visually shake-free appearance for the whole video sequence. Thus, the two steps of motion compensation and image warping are to be combined and formulated into a single global optimization process through just solving the optimal image warps. Next, we will elaborate the details of our global video stabilization formulation based on the estimated shaky motion.

A. Shaky motion estimation

Firstly, we extrapolate the shaky motion by tracing feature trajectories passing through the video sequence (see Fig. 1 left). Here, we use pyramidal Lucas-Kanade [21] to perform good feature tracking and collect the resultant trajectories in the video, denoted by $\mathcal{P} = \{P_h\}_{h=1}^{M}$. Each trajectory P_h is composed of a set of nodes as the intersection between the trajectory and the corresponding frame, i.e., $P_h = \{P_h^a, ..., P_h^b\}$, where a and b are the indices of the start and end frames. Although feature tracking is not always reliable in practice, the obtained trajectories are still believed to be competent for representing the underlying shaky motion. Similar to existing method [6], [7], [10], [11] et al., the tracked features on moving objects are discarded using the technique like RANSAC, to make our global approach more robust.

Unlike totally abandoning the trajectories in motion compensation and modeling motion using transformation matrices like [10], [11], we adopt and charge feature trajectories for the global formulation on both motion compensation and image warping in the sequel.

B. Global formulation

Our global approach is at its core a set of image warps for the corresponding frames, to achieve both shake-free and distortion-less appearance in the stabilized sequence. Formally, given a video sequence with N frames as $\mathcal{I} = \{I_t\}_{t=1}^{N}$, we want to find an image warp f_t for each frame I_t, such that the regenerated sequence $\{f_t(I_t)\}_{t=1}^{N}$ can present steady motion trend yet still adhere to its original appearance. Hence, we resort to a formulation on video stabilization as follows:

$$\mathcal{F}(f_t) = \mathcal{F}_m(f_t) + \mathcal{F}_s(f_t) \tag{1}$$

where $\mathcal{F}_m(\cdot)$ serves the steady motion of trajectories, and $\mathcal{F}_s(\cdot)$ controls visual distortion in image warping for the stabilized appearance.

With Eqn.(1), it comes with the first key ingredient for our global setup by using f_t as the common proxy in the two parts. Previous methods like [6], [7], [14], perform motion smoothing and image warping as two separate parts, thus having two optimization steps, of which the output of smooth trajectories becomes the input of the image warping. Some other methods like [9], [10], [12], directly solve for optimal warps like ours. But they appoint the proxy f_t as a single homography or affine transformation without the part of $\mathcal{F}_s$ for refraining distortion, which cannot model parallax well and always results in some notable visual artifacts in the stabilized sequence. As the second key ingredient of our global setup, we adopt vertices positions of grid mesh embedded in each frame to parameterize f_t, which is able to encode both the smooth motion and image warp simultaneously.

Concretely, a grid mesh is composed of sampled 2D vertices $\mathcal{V} = \{V_{i,j}\}$ and edges $\mathcal{E} = \{V_{i,j}, V_{i+\sigma,j+\delta}\}$, where $\sigma, \delta \in \{-1, 0, 1\}$ indicate the neighborhood relationships such that $Q_{i,j} = \{V_{i,j}, V_{i+1,j}, V_{i+1,j+1}, V_{i,j+1}\}$ forms a quad (see Fig. 2 left). For each frame, we arrange a regular grid mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}\}$, thus constructing a discrete domain for image warping. Then, image warp of each frame is determined by the positions of mesh vertices, and the interior of each quad is obtained by interpolation. For a given frame I_t, we have the mesh vertices as $\{X_{i,j}^t = (x_{i,j}^t, y_{i,j}^t) \in \mathbf{R}^2\}$, and the warped vertices after stabilization are denoted by $\{V_{i,j}^t = (u_{i,j}^t, v_{i,j}^t) \in \mathbf{R}^2\}$, where t indicates the frame index. Then, the two parts of Eqn.(1), $\mathcal{F}_m(\cdot)$ and $\mathcal{F}_s(\cdot)$, will be completely represented by the position of mesh vertices as follows:

$$\begin{aligned} \mathcal{F}_m(f_t) &= \mathcal{F}_m(V_{i,j}^t) = \lambda_1 E_{ts}(V_{i,j}^t) + \lambda_2 E_{sc}(V_{i,j}^t) \\ \mathcal{F}_s(f_t) &= \mathcal{F}_s(V_{i,j}^t) = \lambda_3 E_{sp}(V_{i,j}^t) + \lambda_4 E_{wf}(V_{i,j}^t) \end{aligned} \tag{2}$$

where the temporal smoothness term $E_{ts}(\cdot)$ and spatial consistency term $E_{sc}(\cdot)$ ensure a temporally and spatially consistent motion compensation during trajectories smoothing, and the shape-preserving term of $E_{sp}(\cdot)$ and warping fidelity term $E_{wf}(\cdot)$ impose less distortion in image warping. The coefficients $\lambda_{1,2,3,4}$ are the constant weights for these terms. Note the sole variables in these terms are positions of mesh vertices, so we can present a global formulation for video stabilization. Next, we will elaborate the details to define these terms in the formulation.

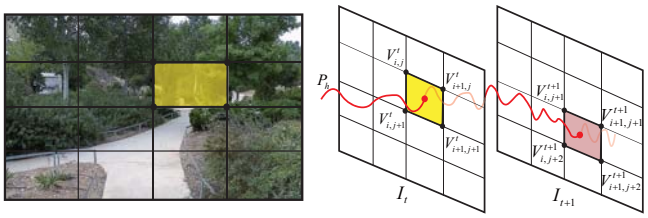


Fig. 2. **Left:** Grid mesh embedded in each frame. **Right:** Trajectory node can be represented by the four vertices of the quad with bilinear interpolation.

Temporal smoothness. The aim of stabilization is to provide the smooth display of video sequence, so both high- and low- frequency vibration should be suppressed in order to produce shake-free appearance. Inspired by the works of [10], [11], we resort to modulation of the trajectories with respective to some specified smoothness. But instead of applying full-frame affine transformations [10] or local homographies [11]

to accommodate the smoothness, we directly operate on the trajectories for steady motion. To achieve video stabilization, we can minimize the acceleration of trajectories using the following temporal smoothness term

$$E_{ts} = \sum_{h}\sum_{t} \|P_h'^t - \sum_{s\in\Omega_t} w_{sh} P_h'^s\|^2 \tag{3}$$

where $P_h'^t$ denotes the node positions of smoothed trajectory in the t-th frame, $P_h'^s$ means the node belonging to the neighborhood Ω_t of the t-th frame, and w_{sh} is a weight of $P_h'^s$. Similar to [11], we use the idea of bilateral filter with combination of two Gaussian functions into the definition of the weight w_{sh} as follows:

$$w_{sh} = G_t(\|t-s\|)G_d(\|P_h'^t - P_h'^s\|) \tag{4}$$

We set the radius of the Gaussian function $G_t(\cdot)$ equal to 6 frames, which is large enough to achieve smooth camera motion, and set the standard deviation of the Gaussian function $G_d(\cdot)$ as 10 pixels, based on an average distance between two nearby feature points gained from many experiments. Besides, using bilateral filter for temporal smoothness can also avoid a large cropping ratio when the camera motion is quite fast.

As our global formulation commits itself to variables only using the positions of mesh vertices, $P_h'^t$ needs to be encoded with $V_{i,j}^t$ in a unified form. Here, we employ the bilinear coordinates to represent the trajectory nodes as used in [6], [11], [13], which is proven to be similarity invariant. Supposing a trajectory node P_h^t that falls in a quad with four vertices as $X_h^t = [X_{i,j}^t, X_{i+1,j}^t, X_{i+1,j+1}^t, X_{i,j+1}^t]$, then it can be represented by their linear combination as $P_h^t = X_h^t \cdot C_h^t$, where $C_h^t = [c_{i,j}^t, c_{i+1,j}^t, c_{i+1,j+1}^t, c_{i,j+1}^t]^T$ are the bilinear coordinates with respect to the four vertices. These bilinear coordinates are used as interpolation weights of the trajectory node, thus representing the trajectory based on the vertices. To keep the consistency of trajectory in the smoothing and warping, we expect the warped node to have the same configuration, i.e., $P_h'^t$ has the same weights with respect to the stabilized positions of vertices $V_h^t = [V_{i,j}^t, V_{i+1,j}^t, V_{i+1,j+1}^t, V_{i,j+1}^t]$ (see Fig. 2 right). Then, we have $P_h'^t = V_h^t \cdot C_h^t$. Consequently, the temporal smoothness term can be written as

$$E_{ts}(V_{i,j}^t) = \sum_{h}\sum_{t} \|V_h^t \cdot C_h^t - \sum_{s\in\Omega_t} w_{sh} V_h^s \cdot C_h^s\|^2 \tag{5}$$

Actually, encoding with mesh vertices also has the effect that imposes the consistency of trajectories within the same quad in the smoothing.

Spatial consistency. As pointed by Liu et al. [6] that individual smoothing of every trajectory leads to broken geometric relationships among different video regions, trajectories need to be smoothed in a consistent manner across the whole frame. In our setup, the smoothed trajectories are determined by the mesh vertices, i.e., image warp function f_t of each frame. For the trajectories in a quad, the consistency has been guaranteed by the similarity invariant bilinear interpolation scheme as used in temporal smoothness term. For trajectories among different quads, they also need to go through coherent modification for the smooth motion trend, i.e., requiring the image warp to be continuous over the entire frame. To realize continuous warping among adjacent quads, we allow high-order continuity constraints on the warps $f_t(\cdot)$, requiring its derivatives $\partial^k f_t/\partial^k V$ to be smooth as well. For simplicity, we adopt the derivative continuity up to the second order, and compute them based on finite differences on the grid mesh domain. Then, we have the following spatial consistency term:

$$\begin{aligned} E_{sc}(V_{i,j}^t) = &\sum_{t}\sum_{i,j} \|V_{i,j+1}^t - 2V_{i,j}^t + V_{i,j-1}^t\|^2 \\ &+ \sum_{t}\sum_{i,j} \|V_{i+1,j}^t - 2V_{i,j}^t + V_{i-1,j}^t\|^2 \\ &+ \sum_{t}\sum_{i,j} \|V_{i+1,j+1}^t - V_{i+1,j}^t - V_{i,j+1}^t + V_{i,j}^t\|^2 \end{aligned} \tag{6}$$

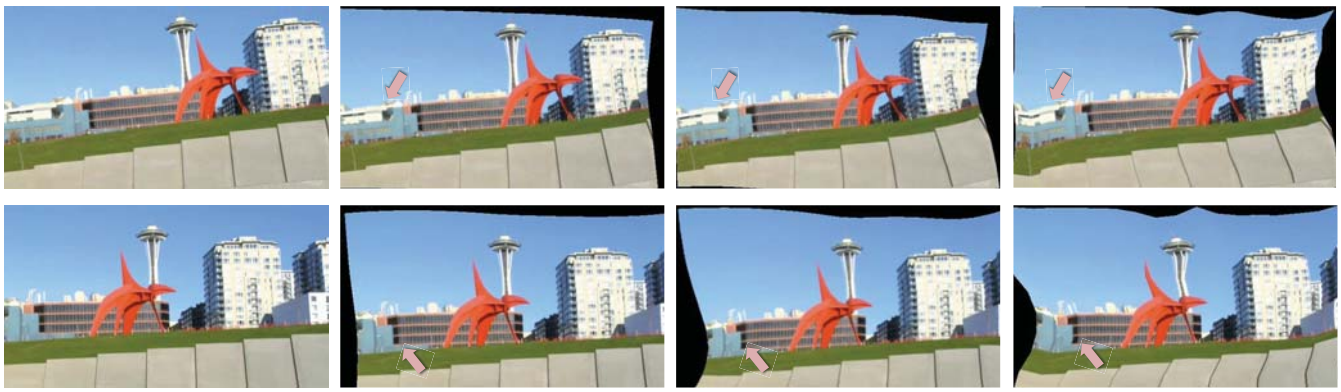
This equation is also used to solve the problem of perspective image warping [22], which enables the warps continuous across the image. Unlike the method of [7] using strict rigid transformation to enforce the spatial consistency, our method broadens the constraint to allow variance in the sense of similar transformation as defined in the following shape preservation term, which is able to deal with videos with a wide range of motions.

Actually, the continuity of image warps can enhance the consistency of smoothing trajectories within different quads, because the trend of smoothed trajectories is settled by the transformation on each quad. As a result, it can achieve uniform warping over the whole frame for less geometric distortion. Besides, remembering bilinear coordinates have similarity invariant property with high-order smoothness, so the spatial consistency does not conflict the demanding on temporal smoothness of trajectories in the same quads. Hence, with the constraints defined by Eqn.(3) and (6), we are able to obtain spatially and temporally consistent smoothing effect on trajectories under the image warping.

Shape preservation. Because we use image warps to drive motion smoothing for stabilization, visual changes of video appearance are unavoidable in the warping process. To keep the appearance close between the original and the stabilized frames, we choose image warps that are able to retain the local contents with similarity as most previous methods do in [6], [13], [23]. We therefore constrain image warps $\{f_t\}$ within the family of similar transformations, i.e., conformal mappings between the two 2D domains. Mathematically, a warping function $f_t(\cdot)$ is a conformal mapping equivalently saying it satisfies Cauchy-Riemann equation [22], whose Jacobian matrix has a skew symmetric form. So for the warping function $(u^t, v^t) = f_t(x^t, y^t)$, it has to admit two equations: $\partial u^t/\partial x^t = \partial v^t/\partial y^t$ and $\partial u^t/\partial y^t = -\partial v^t/\partial x^t$. With the regular mesh $\mathcal{M}$ as the domain, we employ finite difference on the vertices, and obtain the discrete form of Cauchy-Riemann equation as follows:

$$\begin{aligned} u_{i+1,j}^t - u_{i,j}^t &= v_{i,j+1}^t - v_{i,j}^t \\ u_{i,j+1}^t - u_{i,j}^t &= v_{i,j}^t - v_{i+1,j}^t \end{aligned} \tag{7}$$

Then, shape preservation amounts to summing up the violation of conformity over all the quads of the frames, and



(a) (b) (c) (d)

Fig. 3. Video stabilization results with the proposed energy terms. (a) Input two sampled frames (74th (top) and 242th (bottom)). (b) Results by solving the global formulation with all the four terms. (c) Results without the spatial consistency term. (d) Results without the shape preservation term.

minimizing the following energy function

$$E_{sp}(V_{i,j}^t) = \sum_t \sum_{i,j} (\|(v_{i+1,j}^t - v_{i,j}^t) + (u_{i,j+1}^t - u_{i,j}^t)\|)^2 + \sum_t \sum_{i,j} (\|(u_{i+1,j}^t - u_{i,j}^t) - (v_{i,j+1}^t - v_{i,j}^t)\|)^2 \quad (8)$$

Intuitively, Eqn.(8) encourages a sort of uniform scales of a quad in the warping, thus resisting shear and non-uniform scale on the enclosed region for shape preservation. In fact, our shape preservation term $E_{sp}(\cdot)$ bears resemblant effect to the 'as-similar-as-possible' transformation term as used in [11], [13], merely differing in the form of discretization and involving less quadratic items by using Eqn.(8), i.e., having only about one-fourth of the number of items used in [13].

Warping fidelity. Image warping based on similarity transformation can well keep the original appearance of video sequence, but possibly causes global shrinkage due to the homogeneity of Eqn.(8). Hence, we prefer the warped frame to endure a moderate shape change as a whole, i.e., constraining mesh vertices deviation. Besides, the change of trajectory nodes can be characterized by the mesh vertices based on their bilinear coordinates. Hence, it is necessary to keep the warped positions of mesh vertices within a small variance with respect to their original positions. Thus, we have the following warping fidelity term to shape the overall deviation in the warping:

$$E_{wf}(V_i^t) = \sum_t \sum_{i,j} \|V_{i,j}^t - X_{i,j}^t\|^2 \quad (9)$$

This term imposes the warped frame to occupy as much visible field as possible when applying appropriate trajectory smoothing by Eqn.(4) and (5). Besides, it also takes on an analogous role as in the method of [7] to deal with some challenging videos that have the frames containing no trajectory nodes.

Finally, per the constraints defined as above, we obtain the terms of $\mathcal{F}_m(\cdot)$ and $\mathcal{F}_s(\cdot)$ based on the positions of mesh vertices, and have the global form of the energy function as follows:

$$\mathcal{F}(V_{i,j}^t) = \lambda_1 \cdot E_{ts} + \lambda_2 \cdot E_{sc} + \lambda_3 \cdot E_{sp} + \lambda_4 \cdot E_{wf} \quad (10)$$

The weights $\lambda_{1,2,3,4}$ dominate the efficacy of the proposed four terms on the obtained image warps, thus influencing the final stabilized appearance. We will discuss the weights setting in the following section.

Solving the optimization problem of Eqn.(10) is easy, because the function $\mathcal{F}(\cdot)$ is a favorable quadratic energy function with the positions of vertices as the variables, and video stabilization is totally determined by solving their new positions. With the constraints as above, Eqn.(10) has well-posedness for a non-trivial solution. Finally, stabilization results are obtained by transformation and interpolation according to the new mesh vertices resolved by optimizing Eqn.(10) on the frames (see Fig. 3(b)).

Remark. The four terms of Eqn.(10) are all necessary for the desirable stabilization results. The temporal smoothness term E_{ts} is the key aspect to determine the temporal behavior of the smoothed sequence. Although the spatial consistency term E_{sc} and the shape preservation term E_{sp} look alike by using C^2 and C^1 continuity as constraints respectively, and C^2 implying C^1 continuity, they have different influences on the stabilization. Due to the second order continuity, E_{sc} drives the frame warping to be more uniform over the whole frame, otherwise without this term, the stabilized frame might have biased warping artifacts in different regions (see Fig. 3(c)). E_{sp} is a common form respecting shape preservation, which serves the minimization on the local geometric distortion in warping frame. The absence of this term incurs severe deviation of the stabilized video from its original appearance (see Fig. 3(d)). Actually, the use of both E_{sc} and E_{sp} enables a trade-off warping effect between the rigid transformation used in [7] and similarity transformation used in [13], towards producing better stabilization effects (see the example of *Video*#1 in the accompanying video demo *supplement.avi*, which can be viewed from our webpage[1]). The warping fidelity term E_{wf} explicitly balances the visible field change caused by using the smoothed trajectories based on E_{ts}. And more importantly, it makes Eqn.(10) having a non-trivial solution, because the other three terms are all homogeneous with respect to the variables and take all-zero as their optimal solutions.

[1]http://iitlab.bit.edu.cn/gvlab/download/stabilization/supplement.zip

C. Optimization

The sole variables of Eqn.(10) are the 2D positions of mesh vertices as $\{V_{i,j}^t\}$, which favors a one-shot global optimization on both motion compensation and image warping. As a result, the stabilized frames can be directly rendered by using the obtained vertices, rather than reassigning their positions according to the computative transformations like [10], [11]. Seemingly, our formulation of Eqn.(10) contains slightly more terms for desired stabilization, but it still presents a quadratic form with respective the variables, thus avoiding any iterative updating like [11]. To seek for the optimal solution, we compute its gradient and establish the normal equation, which is a linear system with dimensionality equal to the number of variables as $2 \times W \times H \times N$, where $W \times H$ is the number of mesh vertices in each frame and N is the number of frames. Fortunately, the coefficient matrix of normal equation appears to be sparse, due to only local neighborhood in space and time is involved in the terms. Here, we use the sophisticated sparse linear system solver like TAUCS [24], to perform fast computation, by which we immediately obtain the transformed positions of mesh vertices.

IV. Implementation

With trajectories representing shaky motion, the task of video stabilization by our approach comes down to an instance of an energy minimization problem, i.e., by solving Eqn.(10) for the optimal positions. Although variables are therein just the positions of mesh vertices and attached to a sparse linear system, it is still tough to solve the minimization problem by considering all video frames in such a global form, especially for lengthy or high-resolution videos. So we split the input video into a set of segments $\{S_k\}$ with overlap between adjacent segments. For all the examples in this paper, an input video is firstly splitted into segments with 200 frames, and assigned with 30 frames in the overlapping portion. And for every frame, we partition it into regular mesh quads with 80×80 pixels in each one. Then, we perform stabilization on the segments sequentially to further reduce the number of variables in practice.

For the overlapping frames with the start frame I_a and end frame I_b, e.g., $\{I_t\}_{t=a}^b = S_k \cap S_{k+1}$, we adapt the warped vertices $\{V_{i,j}^t\}$ obtained by optimization on S_k, and synthesize transitive vertices positions by linear blending on $\{V_{i,j}^t\}$ and $\{X_{i,j}^t\}$, i.e., $(1-\lambda)V_{i,j}^t + \lambda X_{i,j}^t$, where $\lambda = (t-a)/(b-a)$. We use the synthetic positions to initiate original mesh vertices for stabilizing the next segment S_{k+1}, which can produce smooth transition in the overlap. Consequently, we gain the overall stabilization result for the whole video. Such streaming scheme enables our approach to deal with stabilization on a variety of long or large-size videos.

Weights setting. The energy function Eqn.(10) has four weights $\lambda_{1,2,3,4}$ that influence the stabilization results. In fact, $\lambda_{1,2}$ control the spatial and temporal smoothness of both motion compensation and image warping. The setting of λ_1 directly impacts the temporal smoothness of the stabilized frames. A well-designed λ_2 can distract warping distortion uniformly across the frames. λ_3 sets the tone of similarity for image warps, thus concerning the geometric distortion in the stabilized frames. λ_4 relates to the spatial change of the stabilized frame by the deviation of mesh vertices in image warping, which partially affects the cropping areas after stabilization. Consequently, we have to adopt different strategies to set the four weights respectively for controlling the energy terms in our implementation as follows.

We set $\lambda_1 = 12$ as a constant to achieve a predominant smoothing effect for primarily serving our stabilization purpose. While for λ_2, we find a constant setting $\lambda_2 = 20$ is sufficient to work in the production of appealing results with an overall absorption on the warping distortion across the frame. We set λ_3 according to the distribution of trajectory nodes in the quads enclosed by the corresponding vertices, which is adaptively defined as

$$\lambda_3(Q_{i,j}) = \begin{cases} 5, & : \quad \mathcal{N}(Q_{i,j}) > 0 \\ 15, & : \quad \mathcal{N}(Q_{i,j}) = 0 \end{cases} \tag{11}$$

where $\mathcal{N}(Q_{i,j})$ is the number of trajectory nodes in the quad $Q_{i,j}$. Thus, the quads having trajectory nodes are prone to obey the trajectory smoothing trend, while the others devoid of any node are assumed with similar transformation. Since we want to keep as much visible field as possible after frame cropping, we stress the fixation for the four corners of each frame by the values of λ_4 on these vertices, i.e.,

$$\lambda_4(V_{i,j}) = \begin{cases} 3, & : \quad V_{i,j} \in \mathcal{C} \\ 1, & : \quad \text{otherwise} \end{cases} \tag{12}$$

where $\mathcal{C} = \{V_{0,0}, V_{0,H-1}, V_{W-1,0}, V_{W-1,H-1}\}$ are the four corners of the frame. Such weights setting is able to process various shaky videos, and we use the same weights defined as above for all the examples in the following experiments.

V. Experiments

We have implemented our algorithm and tested it on a variety of videos. The dataset comprises 100 publicly available video clips from previous works on video stabilization, which touches on different types of motions by hand-held cameras (see the examples in Fig. 4 and 7). We presented the stabilization results by our approach and evaluation for its actual performance. We also made extensive comparisons with some state-of-the-art methods and systems for video stabilization. For visual demonstration on the superiority of our approach, we provide a dynamic comparison in the companion video demo *comparison.mp4*, which can also be viewed from our webpage[2]. Besides, user study was conducted to evaluate the quality of stabilized videos. All the experiments ran on a single PC machine with Intel Core i5-2400 3.1GHz CPU and 8G RAM. Next, we will look in more details on our experiments.

A. Processing time

Provided with the feature trajectories, most efforts in our approach are devoted to optimizing the energy function of Eqn.(10). Hence, the primary time consuming is laid on constructing and solving the sparse linear system. Generally

[2]http://iitlab.bit.edu.cn/gvlab/download/stabilization/comparison.zip

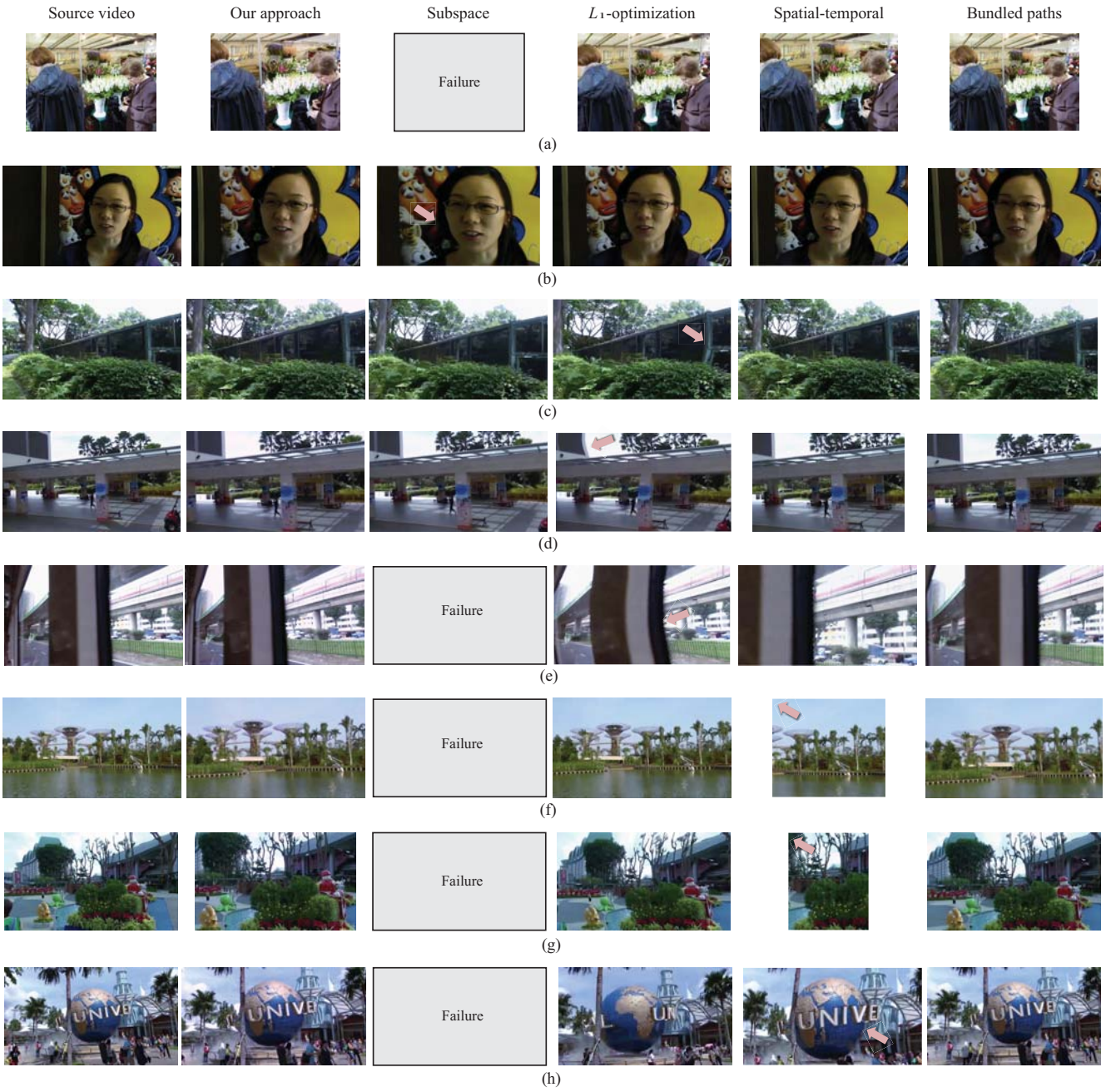


Fig. 4. Comparisons of stabilization results by our approach, subspace [6], L_1-optimization [10], spatial-temporal optimization [7] and bundled paths [11].

for 200 frames with 1280×720 resolution, feature tracking for trajectories takes about 3.3 seconds, and then constructing and solving the sparse linear system take about 4 seconds; we finally obtain the stabilized frames after cropping outranged regions and rendering them with texture mapping technique, which consumes about 3.8 seconds for 200 frames. In summary, our approach can achieve a 18fps processing toward fast video stabilization.

B. Comparison with state-of-the-art methods

We compared our stabilization results with some state-of-the-art methods, including using subspace [6], L_1-optimization [10], spatial-temporal optimization [7] and bundled paths [11]. Most results are directly obtained from the authors' publications or by using the binary code that they provided. Fig. 5 demonstrates the average performance of different stabilization methods. Since the implementation of the steps of feature tracking and frame rendering might differ from each other, we also list just the per frame average running time for the optimization steps involved in different methods (see Table I). Generally, our approach gains fast stabilization computation that is comparable with the spatial-temporal optimization method [7], but is significantly faster than the other methods. Besides, our approach can deal with

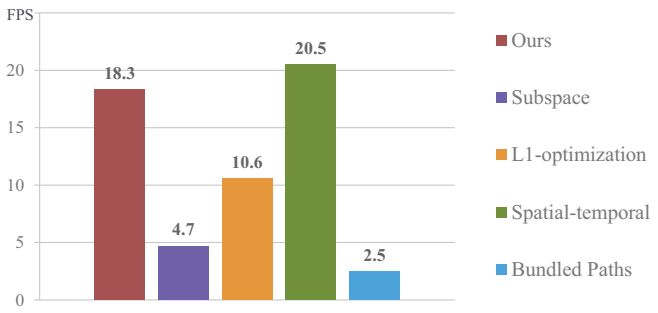


Fig. 5. Comparisons on the average time of the entire stabilization processing (measured by FPS: frames per second) by using our approach, subspace [6], L_1-optimization [10], spatial-temporal optimization [7] and bundled path-s [11].

shaky videos that cannot be processed by previous method like [6], and well preserve the original appearance with less visual distortion and frame cropping than previous methods like [7], [10] (see Fig. 4 and the dynamic comparison in the video demo *comparison.mp4*).

TABLE I
PER FRAME AVERAGE RUNNING TIME (IN SECONDS) FOR THE OPTIMIZATION STEPS OF DIFFERENT METHODS.

Our approach	Subspace	L_1-optimization	Spatial-temporal	Bundled paths
0.02	0.1	0.05	0.015	0.06

The subspace [6] method is based on smoothing long feature trajectories with low-rank constraints to perform motion compensation, and the smoothed trajectories are necessary to render stabilized video. So when there is no long trajectory, for example, with occlusion, fast rotation or zooming motion, this method fails in the motion smoothing step (see Fig. 4(a), (f) and (h) and the corresponding videos in *comparison.mp4*). On the contrary, the setup of our formulation can accommodate both long and short trajectories, and the quadratic energy function can always be optimized toward a non-trivial solution. Hence, our approach are less sensitive to trajectory length and can produce better stabilization results. Fig. 6 shows the distribution of trajectories before and after stabilization by applying our approach on the videos in Fig. 4(a), (f) and (h), which involves motions with occlusion, fast rotation and zooming respectively. For visual clarity, we just sample a subset of trajectories as shown in the figure. Actually, even for the video that has trajectory passing through only two frames, e.g., the video in Fig. 4(a) (showing the corresponding trajectories in Fig. 6(a)), our approach can work well without any processing failure, just possibly incurs minor stability artifacts. Hence, our approach is more robust to deal with both long and short trajectories, like the spatial-temporal optimization method [7].

The method based on L_1-optimization [10] is usually robust enough to process many shaky videos. But it uses only one homography matrix that cannot well represent scenes with large parallax correctly, and usually causes visible distortion due to its 2D motion model (see Fig. 4(c-e) and the corresponding videos in *comparison.mp4*). Because our method employs conformal mapping for shape preservation, it is more

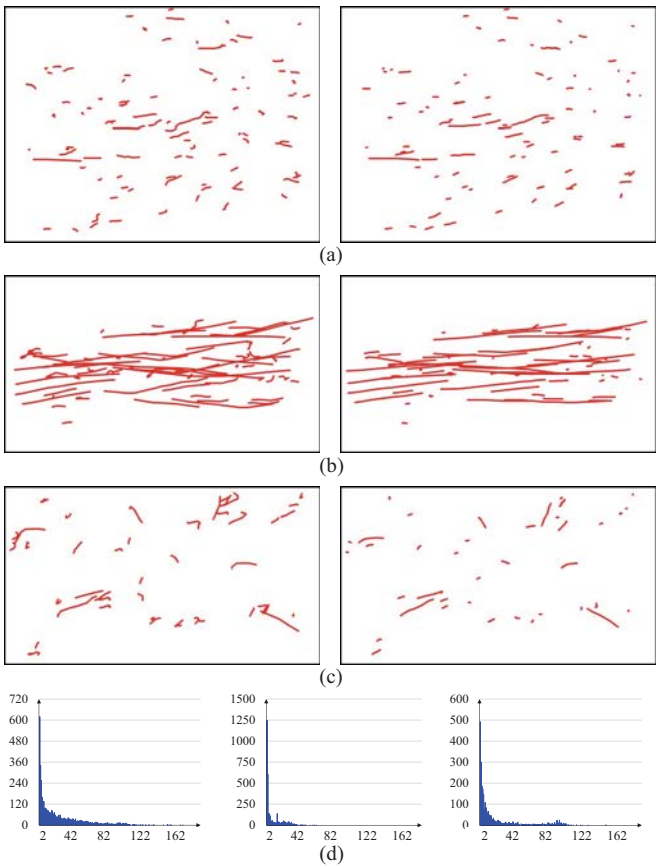


Fig. 6. Trajectories of the videos with (a) occlusion (in Fig. 4(a)), (b) fast rotation (in Fig. 4(f)) and (c) zooming (in Fig. 4(h)). **Left:** Trajectories before stabilization. **Right:** Trajectories after stabilization by our approach. (d) shows the statistics on the number of trajectories (vertical axis) of different lengths (horizontal axis) within the range $[2, 200]$ for the corresponding three videos.

flexible to handle parallax well by smoothing motion with spatial-temporal coherence. Besides, our approach gains faster performance due to optimizing a single quadratic energy function in the sense of L_2-optimization.

By employing Bézier curves to fit the trajectories, spatial-temporal optimization method [7] can usually achieve preferable stabilization results from shaky videos rapidly, which has a commendably fast computation in the motion smoothing step. But when camera motion is fast and shake is acute, such as fast rotation and zooming, Bézier curves are quite difficult to fit a smoothed camera path with a reasonable balance between video stability and image cropping ratio. With the constraint of Bézier curves and rigid transformations that they use for consistent smoothing, this method always results in large cropping ratio for some videos (see Fig. 4(f-g) and the corresponding videos in *comparison.mp4*). Besides, by using rigid transformation as constraints, this method usually incurs noticeable distortion for videos with zooming motions, which essentially have nearly uniform changes between adjacent frames (see Fig. 4(h)). On the contrary, our method gives up fitting method and chooses motion smoothing and image warping with shape preservation based on the 'as-similar-as-possible' principle, which is able to process and produce better

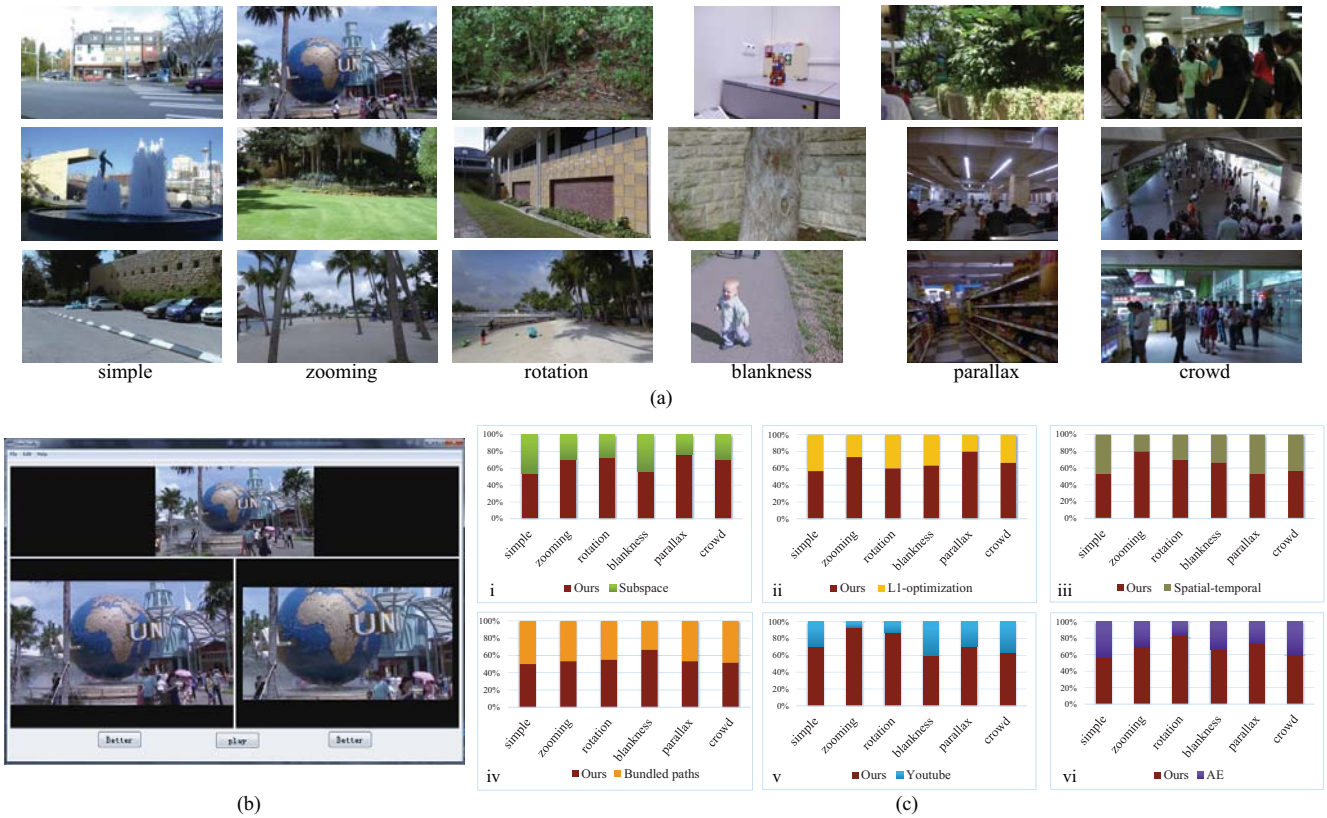


Fig. 7. User study. (a) Testing videos with different shaky motions. (b) User interface. (c) Statistics of user study on stabilization results by our approach, subspace [6], L_1-optimization [10], spatial-temporal optimization [7], bundled paths [11], Youtube and AfterEffects.

stabilization effects for shaky videos with zooming motion.

The bundled paths method [11] uses local homographies to represent camera motion and optimizes bundled paths, which offers much better stabilization effects over previous methods (see Fig. 4 and the corresponding videos in *comparison.mp4*). Actually, trajectories and homographies are two types of motion representation. The use of trajectories in our approach can enrolls just the positions of mesh vertices as the sole variables of the global formulation, which can significantly reduces the computational complexity. Concretely, the bundled path method needs to estimate local homographies between neighboring frames, which results in a linear system with dimensionality of the number of variables as $8 \times (W-1) \times (H-1) \times N$, where $(W-1) \times (H-1)$ is the number of quads in each frame, N is the number of frames. Such linear system typically has the non-zero ratio approximating to $61/((W-1) \times (H-1) \times N)$, which makes its computation slow. While dimensionality of our linear system is $2 \times W \times H \times N$ and the nonzero ratio is about $13/(W \times H \times N)$, our approach therefore has much less cost and a faster computation than bundled paths for producing the comparable stabilization results.

User study. We also conducted a user study that collect and analyze feedbacks on viewing the stabilized videos, which are produced by using the above methods and our approach. In our study, we have 30 participants coming from diverse backgrounds and ages. We selected 18 videos from previous publicly testing videos that have various motion styles as shown in Fig. 7(a). We designed a side-by-side view interface for video display (see Fig. 7(b)), where the source shaky video at the top, and the stabilization results by our approach and one of the other four methods at the below. The order of the two stabilized videos are random, and the corresponding stabilization methods or our approach are also anonymous to participants. Every participant is asked to select a better one between the two stabilized videos that they view in the interface, and we collect their feedbacks on choosing a better stabilization result. Fig. 7(c:i-iv) shows the statistics on the users' preferences. It can be seen that for the videos with simple motions, our approach achieves competitive evaluation, while for more complex motions or scenes, e.g., videos with fast rotation, zooming or large parallax, our approach generally gains more favors in the comparisons.

C. Comparison with state-of-the-art systems

Currently, YouTube Stabilizer[3] and Warp Stabilizer in Adobe AfterEffects CS6 (AE) [4] are the two most popular video stabilization systems. As stated in [7], [11], [25], these two systems usually incur noticeable shearing or skewing artifacts, and AE Warp Stabilizer sometimes generates severe cropping in the stabilized frames (see the examples in Fig. 8

[3]http://www.youtube.com
[4]http://www.adobe.com/en/products/aftereffects/warp-stabilizer.html

and the corresponding videos in *comparison.mp4*). On the contrary, the stabilization results by our approach gain overall advantage of balancing visual distortion and cropping. We also enrolled these two systems in the user study as described above to obtain more objective evaluation of the effectiveness of our approach. From the statistics in Fig. 7(c:v-vi), it can be seen that our approach has an overall superiority over the other two stabilized systems, i.e., more participants prefer the stabilization results by our approach, especially for videos with rotation, zooming and parallax.

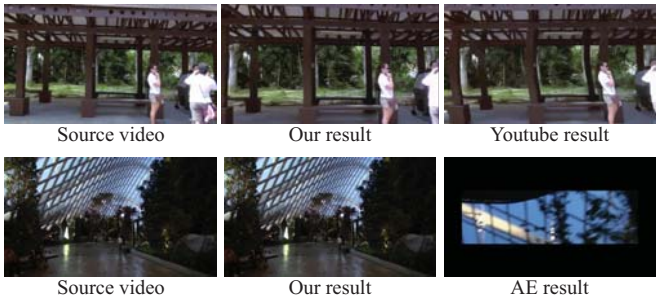


Fig. 8. Comparison of stabilization results by our approach, Youtube and AfterEffects.

D. Limitation and discussion

Our approach is a new attempt to globally address the problem of video stabilization by rephrasing it into a compact one-shot formulation, but we must leave several aspects unexplored as its downside. Firstly, our approach still relies on robust trajectories detection, thus possibly cracks itself in case feature tracking fails. A potential solution to this problem is to adopt local parametric similar transformations between adjacent frames like [11]. Secondly, our approach uses local similar transformations without guarantee on collinearity in image warping. Hence, our approach might generate severe artifacts for videos with apparent line structure (see Fig. 9). This issue can be resolved by adding line-preserving constraints like the one used in [22] in our formulation. Thirdly, there are cases that our approach cannot correctly handle videos with severe rolling shutter effects, because the spatial configuration in each frame has to be preserved to realize consistent motion compensation. Nevertheless, we can borrow the technique of calibration-free rolling shutter removal [25] on the stabilized videos to suppress the remaining skew as a post-processing. Additionally, the computational cost of our approach is directly related to the setting of mesh quad size. It takes about 9 times effort by using the half-size quads. Actually, 80×80 mesh quad is sufficient to produce the desirable stabilization results for 720p videos, and the use of denser mesh has little influence on the visual quality in our experiments (see the example of *Video*#2 in the accompanying demo *supplement.avi*).

VI. Conclusion

We have presented a novel formulation for video stabilization in a global optimization manner. The core is to encode motion compensation and image warping with positions of mesh vertices as the common embodiment, thus resulting in a fast one-shot solution on image warps toward stabilization. Besides, our approach can significantly reduce geometric distortion caused by image warping due to use of similarity-invariant representation for spatially and temporally consistent smoothing on trajectories. Experiments illustrate the potential of our approach in terms of flexibility and efficiency, to produce compelling results over the state-of-the-art methods.

Fig. 9. Limitation. Our approach might generate severe distortion in the case of apparent line structure in the video.

There are multiple areas of future work. Although our approach can generate pleasing stabilization by using local similar transformations to approximate inter-frame motion, it still incurs distortion for videos with quite large depth variations. So we plan to incorporate homographies in our approach to adaptively set local transformations for quads. Such hybrid image warps are also rewarding to deal with rolling shutter effect caused by fast camera motion. Besides, GPU acceleration is necessary, especially for solving the sparse linear system to enable a real-time stabilization.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] G. Puglisi and S. Battiato, "A robust image alignment algorithm for video stabilization purposes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1390–1400, Oct. 2011.
[2] S.-P. Lu, S.-H. Zhang, W. Jin, S.-M. Hu, and R. R. Martin, "Timeline editing of objects in video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1218–1227, Jul. 2013.
[3] S.-C. Liu, L. Yuan, P. Tan, and J. Sun, "Steadyflow: spatially smooth optical flow for video stabilization," in *Proc. CVPR*, 2014.
[4] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin, "Internet visual media processing: a survey with graphics and vision applications," *The Visual Computer*, vol. 29, no. 5, pp. 393–405, Mar. 2013.
[5] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, "Video stabilization using robust feature trajectories," in *Proc. ICCV*, 2009, pp. 1397–1404.
[6] F. Liu, M. Gleicher, J. Wang, H.-L. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Transactions on Graphics*, vol. 30, no. 1, pp. 4:1–4:10, Feb. 2011.
[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*, vol. 19, no. 8, pp. 1354–1361, Aug. 2013.
[8] J. Zhou, H. Hu, and D.-R. Wan, "Video stabilization and completion using two cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 12, pp. 1879–1889, Dec. 2011.
[9] Y. Matsushita, E. Ofek, X.-O. Tang, and H.-Y. Shum, "Full-frame video stabilization," in *Proc. CVPR*, 2005, pp. 50–57.
[10] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust L_1 optimal camera paths," in *Proc. CVPR*, 2011, pp. 225–232.
[11] S.-C. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 78:1–78:10, Jul. 2013.

[12] M. L. Gleicher and F. Liu, "Re-cinematography: Improving the camera dynamics of casual video," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 5, no. 1, pp. 2:1–2:28, Oct. 2008.
[13] F. Liu, M. Gleicher, H.-L. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 44:1–44:9, Aug. 2009.
[14] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," *ACM Transactions on Graphics*, vol. 31, no. 5, pp. 126:1–126:10, Sep. 2012.
[15] Z.-H. Zhou, H.-L. Jin, and Y. Ma, "Plane-based content-preserving warps for video stabilization," in *Proc. CVPR*, 2013, pp. 2299–2306.
[16] C. Buehler, M. Bosse, and L. McMillan, "Non-metric image-based rendering for video stabilization," in *Proc. CVPR*, 2001, pp. 609–614.
[17] S.-C. Liu, Y.-T. Wang, L. Yuan, J.-J. Bu, P. Tan, and J. Sun, "Video stabilization with a depth camera," in *Proc. CVPR*, 2012, pp. 89–95.
[18] J. Yang, D. Schonfeld, and M. Mohamed, "Robust video stabilization based on particle filter tracking of projected camera motion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 945–954, Jul. 2009.
[19] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin, "Capturing intention-based full-frame video stabilization," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1805–1814, Sep. 2008.
[20] Z.-Q. Wang, L. Zhang, and H. Huang, "Mulitplane video stabilization," *Computer Graphics Forum*, vol. 32, no. 7, pp. 265–273, Sep. 2013.
[21] J. B. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR*, 1994, pp. 593–600.
[22] R. Carroll, A. Agarwala, and M. Agrawala, "Image warps for artistic perspective manipulation," *ACM Transactions on Graphics*, vol. 29, no. 4, pp. 127:1–127:9, Jul. 2010.
[23] J. Wei, L. Chen-Feng, S.-M. Hu, R. R. Martin, and C.-L. Tai, "Fisheye video correction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 10, pp. 1771–1783, Jul. 2012.
[24] "TAUCS: a library of sparse linear system," http://www.tau.ac.il/ stoledo/taucs/.
[25] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Effective calibration free rolling shutter removal," in *Proc. ICCP*, 2012.

Lei Zhang received the B.S. and Ph.D. degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively. He is currently an Associate Professor at the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include image and video processing, computer graphics. He is a member of IEEE.

Qian-Kun Xu received the B.S. degree in computer science from Shandong Normal University, Jinan, China, in 2012. He is currently a graduate student at the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interest is image and video processing.

Hua Huang received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1996 and 2006, respectively. He is currently a Professor at the School of Computer Science, Beijing Institute of Technology, Beijing, China. He is also an Adjunct Professor at the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China. His research interest is image and video processing. He is a member of the IEEE.