BiggerSelfie: Selfie Video Expansion with Hand-held Camera

Miao Wang, *Member, IEEE,* Ariel Shamir, *Member, IEEE Computer Society,* Guo-Ye Yang, *Member, IEEE,* Jin-Kun Lin, *Member, IEEE,* Guo-Wei Yang, *Member, IEEE,* Shao-Ping Lu, *Member, IEEE,* and Shi-Min Hu, *Senior Member, IEEE,*

***Abstract*—Selfie photography from hand-held camera is becoming a popular media type. Although being convenient and flexible, it suffers from low camera motion stability, small field of view and limited background content. These limitations can annoy users, especially when touring a place of interest and taking selfie videos. In this paper, we present a novel method to create what we call a *BiggerSelfie* that deals with these shortcomings. Using a video of the environment that has partial content overlap with the selfie video, we stitch plausible frames selected from the environment video to the original selfie frames, and stabilize the composed video content with a portrait-preserving constraint. Using the proposed method, one can easily obtain a stable selfie video with expanded background content by merely capturing some background shots. We show various results and several evaluations to demonstrate the applicability of our method.**

***Index Terms*—Hand-held video editing, Selfie, Video stitching**

I. Introduction

In recent years, self-portrait photographs and videos, better known as "selfies" have become extremely popular. Such photos and videos are typically created using mobile phones or portable cameras either by directly holding them in hand or fixing them on a selfie-stick. Selfie videos have advantage of detailed appearance and high expressiveness for portraits, but also bear disadvantages such as unstable camera motion and limited field-of-view.

Our work targets the problem where a much larger background is required while being limited by the shortages of selfie photography. To overcome the shortcomings, a photographer has to carefully adjust the shooting position and camera pose back-and-forth to include both the portrait and desired background content in the frame, often causing severe shakiness and jitter. Possible solutions for these issues could be to use a tripod or remote control camera tracks. However, these are toilsome and expensive for personal use.

In this paper, we present an augmentation and stabilization method to create bigger selfie video without relying on specific hardwares. The only extra effort from the user is to supplement the selfie video with some footage of the background with partial frame-overlap with original selfie video. These extra shots can be taken before or after the selfie shooting, by tilting the camera or turning around to capture background content. Given such inputs, our goal is to create an enhanced version of selfie video which is more stable and has a larger field-of-view containing more background content.

The challenges in this problem are multi-fold: (1) as demonstrated [1], [2], [3], background feature identification is crucial for reliable camera motion estimation. However, selfie videos typically contain a large portrait region and little background information. Selfies also include severe motion that can cause feature identification failures; (2) input selfie video and environment video are captured by a hand-held camera with different camera poses, camera paths and lengths, selecting plausible environment frames for seamless stitching is non-trivial; (3) hand-held video stitching is a challenging problem [4], [5], further in our case, when portrait ratio in selfie video is large, it does require additional effort to avoid portrait distortion.

To accomplish these goals, we start by robustly distinguishing foreground feature trajectories from background ones. Background trajectories are used to estimate the selfie camera motion. Given an expansion direction (either specified by the user or by the system), we present a motion-direction-aware frame selection algorithm to select stitching frames from environment video. This algorithm considers factors of feature matching, camera motion consistency, direction preference, temporal smoothness, and distortion. Finally, frame stitching and content stabilization are jointly performed by mesh-based frame warping, taking special care to preserve the portrait region.

As far as we know, this work is the first to target the specific problem of selfie video expansion. We conducted several experiments, demonstrated the robustness of our method, and compared our method with alternatives. Technical contributions of this paper can be summarized as follows:

- A multiple-span local motion analysis algorithm for robust background feature identification in selfie video.
- A novel motion-direction-aware frame selection algo-

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material is a demo video.

M. Wang, G.-Y. Yang, J.-K. Lin, G.-W. Yang and S.-M. Hu are with the Department of Computer Science and Technology, Tsinghua University, Beijing. S.-M. Hu is the corresponding author.

A. Shamir is with the Efi Arazi school of Computer Science, the Interdisciplinary Center, Israel.

S.-P. Hu is with Nankai University, Tianjin, China.

Manuscript received June 19, 2017; revised April 8, 2018; accepted July 12, 2018.

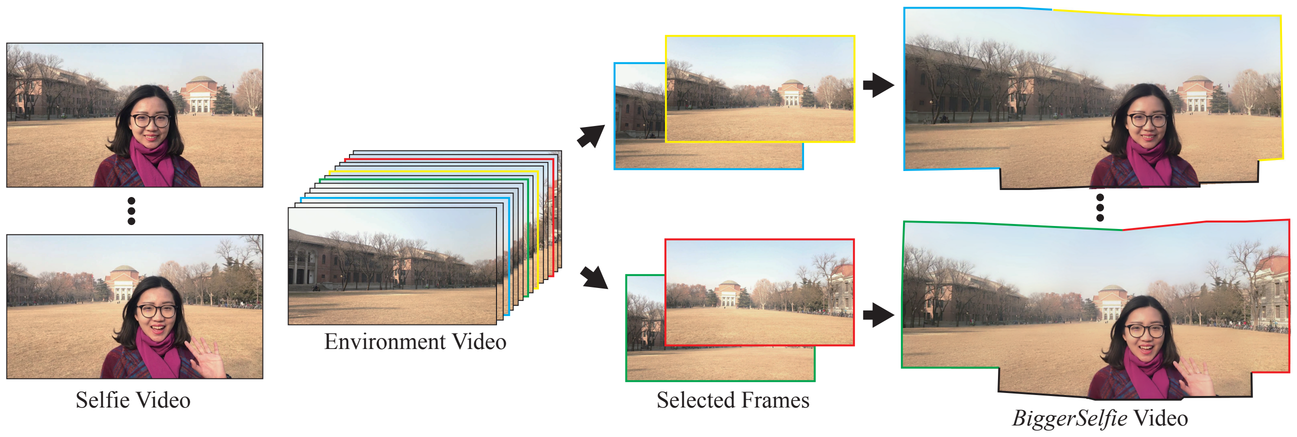


Fig. 1. We present a method which expands a selfie video using an environment video clip, to generate a *BiggerSelfie* video. *BiggerSelfie* is produced by stitching the original selfie frames and selected environment frames. The selected environment frames are temporally consistent with the selfie video in both content and camera motion, while introducing as much background content as possible. In this figure, frames are taken from the SQUARE example.

rithm for stitching asynchronous selfie and environment video frames.
- A portrait-preserving mesh-based warping method that jointly stitches video frames and stabilizes video content.

II. Related Work

Our work is closely related to video stabilization, image and video stitching and selfie photographic editing. Here we limit our discussion to state-of-the-art or representative approaches.

Video stabilization. Hand-held video commonly includes shaky or jittery content. Stabilization techniques have been proposed to stabilize video content by recovering camera paths using 2D and 3D method or a hybrid of both, producing a new stable camera path, and rendering the corresponding content. 2D video stabilization methods estimate Homography or Affine transformations between consecutive frames and smooth these transformations temporally [2], [6], [7], [8]. Camera motions are learned for video stabilization [9] or shot classification [10]. Recently, bundled 2D camera paths were explored as a powerful tool for dealing with parallax [2]. These were later adopted for real-time stabilization [11]. 3D-based methods reconstruct the scene using point clouds followed by a 3D camera path planning [12], [13]. Full 3D reconstruction can be relaxed by leveraging partial 3D information from long feature trajectories [1], [14]. A hybrid 2D-3D approach was proposed to stabilize 360° video [15]. Generally, 2D-based stabilization methods are more efficient and robust, while 3D-based methods could generate better results. In our joint stabilization and stitching approach, we adopt the 2D bundled cameras idea to estimate initial local camera poses, followed by portrait-aware spatial-tempo mesh-based warping.

Background feature identification is essential for robust camera motion estimation and stabilization [2], [3], [16], [17], [18]. Fast video segmentation [19] works well on normal videos, but appears to be fragile on selfie video. Motion segmentation approaches [16], [17], [20] can distinguish feature trajectories in short time periods using clustering algorithms. However for the foreground/background feature identification problem, they could fail to consistently segment trajectories in long periods of time. Zhang et al. [3] proposed to extract background feature trajectories, but their method can have difficulties when the foreground content is near and in the presence of non-rigid motion. In this paper, we propose a multiple span local motion analysis method, a propagation strategy as well as a refinement algorithm to identify background feature trajectories, which work well for selfie videos typically containing limited background content and non-rigid motion of portrait.

Video stitching. In early video stitching works, multiple video clips were composed as a mosaic [21]. For videos captured by multiple static cameras, Agarwala et al. [22] proposed to generate panoramic video texture from a single panning camera, using a min-cut optimization. Perazzi et al. [23] proposed a method that automatically determines the stitching order of structured videos to better hide parallax. Jiang et al. [4] proposed a content-preserving warping method to stitch multiple videos. These methods are restricted to structured camera arrays.

Recently, hand-held video stitching and stabilization attracted much attention from researchers. Lin et al. [24] used 3D-based method with scene reconstruction and virtual path optimization for stabilization and further mesh-based warping for stitching. Later, Guo et al. [5] proposed a 2D-based method which optimizes the virtual camera path and the stitching using bundled cameras [2]. Previous methods assume that the videos have already been synchronized, which in our case is not guaranteed. Moreover, the camera paths to be stitched could be different in both moving directions and angles. Before stitching, we carefully select candidate frames from the environment video with coherent camera motion while maintaining smoothness. Further, we designed an energy function for optimizing mesh layout so that we obtain temporal smoothness, spatial alignment as well as undistorted portrait content in the resulting video, without explicitly optimizing camera path in a separate step.

Selfie photographic editing. Selfie photography gained more popularity in recent years with the development of mobile photography devices. Because of this, selfie photographic editing has also become a hot topic. Such media types share particular characteristics: large portrait ratio, limited field-of-view with little (occluded) background and low temporal stability in video. Shen et al. [25] proposed to automatically segment foreground portrait from selfie photo using convolutional networks. Huang et al. [26] proposed a method to automatically replace the background of portrait video. To enlarge the field-of-view, the expansion operation was introduced to enhance images [27]. Li et al. [20] converted rotating selfie videos into panorama for extreme enlargement of the field-of-view. We adopt the expansion concept for the selfie video case. Our work focuses on enhancing selfie video by stabilizing and expanding the original content, which is a novel and challenging problem.

III. Overview

The inputs to our method are a selfie video I with n frames and an environment video clip $\widehat{I}$ with $\widehat{n}$ frames, where some of the background content has partial overlap with the selfie video. We aim to generate an enhanced selfie video I_{wide} with the same number of frames n as the selfie video, but with a wider view, by seamlessly stitching frames taken from $\widehat{I}$ to the background of I and stabilizing camera motions.

Our method consists of three main steps. **Step 1)** separates the foreground and background feature trajectories in I and estimates the selfie camera motion using only background features, with a novel multiple span local motion analysis algorithm. Portrait mask for each frame is also generated based on separated feature trajectories. In **Step 2)**, appropriate frames from the environment video $\widehat{I}$ are selected to extend the selfie background, using *motion-direction*-aware frame selection algorithm. Given desired background expansion direction(s), we search for a plausible frame subset $\widetilde{I}$, containing partial overlap and coherent motion to the selfie video I, while satisfying the expansion direction and maintaining temporal smoothness in the stitched result. In **Step 3)**, we leverage a mesh-based image warping method to warp selfie frames and selected candidate environment frames, so that the frames are stitched and stabilized, in particular, portrait region in the result is preserved from drifting. After the above three steps, stitched video I_{wide} is rendered by seamlessly blending warped images.

IV. Portrait/Background Feature Trajectory Classification

Typically in 2D-based camera motion estimation methods, the motion between adjacent frames is represented using homography or affine transformations [6], [8]. These transformations are calculated by registering a bundle of reliable matched *background* feature pairs. However, selfie video contains very large foreground portrait which occludes the background. These characteristics mean that the matched features have a high chance of belonging to the foreground, resulting in inaccurate motion estimation. Similarly, existing video foreground extraction method [19] fail to correctly classify foreground/background selfie content, while motion segmentation methods [16], [17], [3], [20] cannot separate feature trajectories accurately because the background feature motion may not be distinguished globally for a long period, or locally between neighboring frames.

The simplest cue assisting foreground/background separation in selfie videos is the existence of face [28]. We leverage an off-the-shelf face landmark detector [29] to detect face region in each frame, and force feature points inside facial region classified as foreground. However, as the foreground contains other parts such as body or hair and hats, other cues are still desired.

Having analyzed the special characteristics of selfie video, we found that feature motion is a strong cue to distinguish background features from foreground ones. For example, background features typically moves faster and sometimes more drastically than the foreground due to camera motion. Although it is difficult to distinguish features for long periods, our key insight is that they could be classified properly for some temporal spans and propagated to the whole feature trajectory. Based on this insight, we developed a multiple-span local motion analysis method. This method works as the basis of temporal propagation to classify feature trajectories into foreground and background. We also developed a spatial refinement strategy to improve the classification result.

A. Multiple Span Local Motion Analysis

The identification of foreground and background feature trajectory is based on motion analysis. Let x_t^i denotes the 2D position of a feature point i at frame t. We define the motion of feature point i at time t within a temporal span s as the vector $\Delta_t^i(s) = x_{t+s}^i - x_t^i$ (see Figure 2). Given N feature points at frame t, there is a set of N feature motion vectors for any span s, denoted by $\Delta_t(s) = \{\Delta_t^1(s), \Delta_t^2(s), \cdots, \Delta_t^N(s)\}$. For each frame t, we compute multiple motion sets $\Delta_t(S)$ with integer span set $S = \{1 \leq s \leq 30\}$, where each $\Delta_t(s)$ provides a *single-span* motion analysis.

To separate feature points at each frame into foreground and background, we cluster each $\Delta_t(s)$ set at frame t using a two-class Gaussian mixture model (GMM) to predict the *foreground* classification probability $Pr(\Delta_t^i(s))$ of each sample $\Delta_t^i(s)$ (where the background classification probability would be $1 - Pr(\Delta_t^i(s))$). For features that lie in the detected face region we initialize the Maximization step with $Pr(\Delta_t^i(s)) = 1.0$.

Next, out of all single-span motion analysis results we choose only the reliable ones, forming a reliable span set RS. This is because the existence of unpredictable camera motion and non-rigid foreground can create meaningless and low quality motions $\Delta_t(s)$. We regard the results of span s as reliable if the $L2$-distance between the average foreground motion $\overline{\Delta_t^F(s)}$ and the average background motion $\overline{\Delta_t^B(s)}$ is above

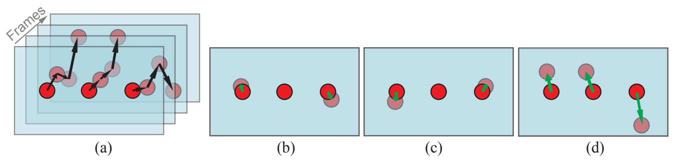


Fig. 2. An example of feature trajectories and local motion analysis. (a) shows three feature trajectories within four frames $t, t+1, t+2, t+3$. Red dots illustrate feature points and black arrows indicates feature offsets between neighboring frames. (b), (c) and (d) shows local motions $\Delta_t(s)$ with spans $s = 1, 2, 3$ respectively. Although the feature trajectories cannot be properly separated with local motion $\Delta_t(1)$, they can be distinguished with motion measurement $\Delta_t(3)$.

a threshold d. We use $d > 0.7d_{max}$, where d_{max} is the maximum distance of the cluster centers over all spans s.

Lastly, we define the *local* foreground probability for each feature i at frame t as the average probability of the feature $Pr(\Delta^i_t(s))$ for all s in the reliable span set RS: $Prob(f^i_t) = \frac{1}{|RS|}\sum_{s\in RS} Pr(\Delta^i_t(s))$. This process is our *multiple-span* local motion analysis, which is used as the basic operation in temporal propagation.

B. Temporal Propagation and Spatial Refinement

The predictions in each frame are propagated temporally along each individual feature trajectory, followed by a spatial refinement between predictions of neighboring feature trajectories.

Temporal Propagation. For each frame t, we iteratively propagate motion analysis results from the previous frames to the current frame by initializing current multiple span motion analysis. We call this process *temporal propagation*. It is performed in both forward and backward directions. In the forward propagation, we initialize the GMM Maximization step for each feature f^i_t at frame t with the average probability of all previous frames: $Prob(f^i_{\{t^i_s, t^i_s+1, \cdots, t-1\}})$, and perform multiple span analysis at frame t to get $Prob(f_t)$. In the backward propagation, we analyze backward motion vectors $\Delta^i_t(s)' = x^i_{t-s} - x^i_t$, and predict classification probability $Prob(f^{i'}_t)$ (see Figure 3(a) and (b)). Finally, we take the average value of local forward and backward motion classification probabilities from t^i_s to t^i_e as the *preliminary* foreground classification probability for each feature trajectory:

$$Prob(f^i) = \frac{1}{2(t^i_e - t^i_s + 1)}\left(\sum_{t=t^i_s}^{t^i_e} Prob(f^i_t) + \sum_{t=t^i_s}^{t^i_e} Prob(f^{i'}_t)\right) \tag{1}$$

Spatial Refinement. We refine the classification probabilities of each trajectory by performing a spatial smoothing process similar to bilateral filtering [30] among nearby trajectories. We regard trajectories f^j and f^i as neighbors only if their time intervals intersect (see Figure 3 (c)), and the distance of their corresponding feature points is smaller than $0.1 \times$ frame width in at least one frame. The smoothed foreground probability for feature trajectory f^i is defined as:

$$\overline{Prob}(f^i) = \frac{Prob(f^i) + \sum_{f^j \in N(f^i)} \omega_{i,j} \cdot Prob(f^j)}{1 + \sum_{f^j \in N(f^i)} \omega_{i,j}}, \tag{2}$$

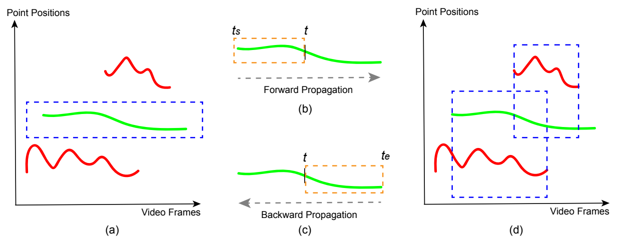


Fig. 3. Temporal propagation and spatial refinement illustration. (a) shows one foreground trajectory (green) and two background trajectories (red). For each trajectory, we firstly perform forward and backward temporal propagation. For example, we perform temporal propagation for the foreground feature (marked in dotted rectangle). (b) In forward propagation, we initialize clustering using clustered result from t_s to $t-1$ marked in dotted rectangle. (c) In backward propagation, we initialize clustering using the clustered result from t_e to $t+1$ marked in dotted rectangle. (d) shows spatial refinement for foreground feature trajectory (green), whose clustering probability is smoothed considering other feature trajectories within temporal intersections (marked in dotted rectangles).

where $N(f^i)$ is the set of neighboring trajectories of f^i, $\omega_{i,j}$ denotes the weight influence of f^j on f^i. $\omega_{i,j}$ is adaptively determined by a function that combines the average spatial distance $d^s_{i,j}$ in pixels and average appearance distance $d^a_{i,j}$ in CIELAB color space between f^i and f^j:

$$\omega_{i,j} = exp(-\alpha_s \cdot d^s_{i,j}) \cdot exp(-\alpha_a \cdot d^a_{i,j}), \tag{3}$$

where we use $\alpha_s = 10$ and $\alpha_a = 15$ to weigh the importance of spatial similarity and appearance similarity of feature trajectories. The more similar two trajectories are, the larger probability that trajectories belong to the same cluster (foreground or background).

We regard a feature trajectory as belonging to the foreground F_P only if $\overline{Prob}(f^i) > 0.6$ and belonging to the background F_B if $\overline{Prob}(f^i) < 0.4$. We discard all other feature trajectories as they are less reliable. Representative background feature identification results are shown in Figure 4.

Camera motion of selfie video is estimated only from background feature trajectories F_B. We also generate a foreground portrait mask Ψ_P for each frame by feeding the classified feature points to the grab-cut algorithm [31]. The resulting masks are used in our portrait-preserving joint stabilization and stitching as well as post-processing video rendering.

V. Motion-direction-aware Frame Selection

To expand the background of a selfie video, the user can define expansion directions as 2D vectors. If the directions are not specified by the user, our system automatically proposes the following options to choose from: UP= $\{(0,1)\}$, LR= $\{(-1,0),(1,0)\}$, LUR= $\{(-1,0),(0,1),(1,0)\}$ or CIRC= $\{(-1,0),(-width, height),(0,1),(width, height),(1,0)\}$. If input selfie has n frames, we need to select n candidate frames from the environment video $\widehat{I}$ and stitch them to the selfie video seamlessly. Candidate frames are selected along each direction separately and matched to the selfie frames. We do not preserve the ordering of the candidate frames, and do not assume that the environment video must have more

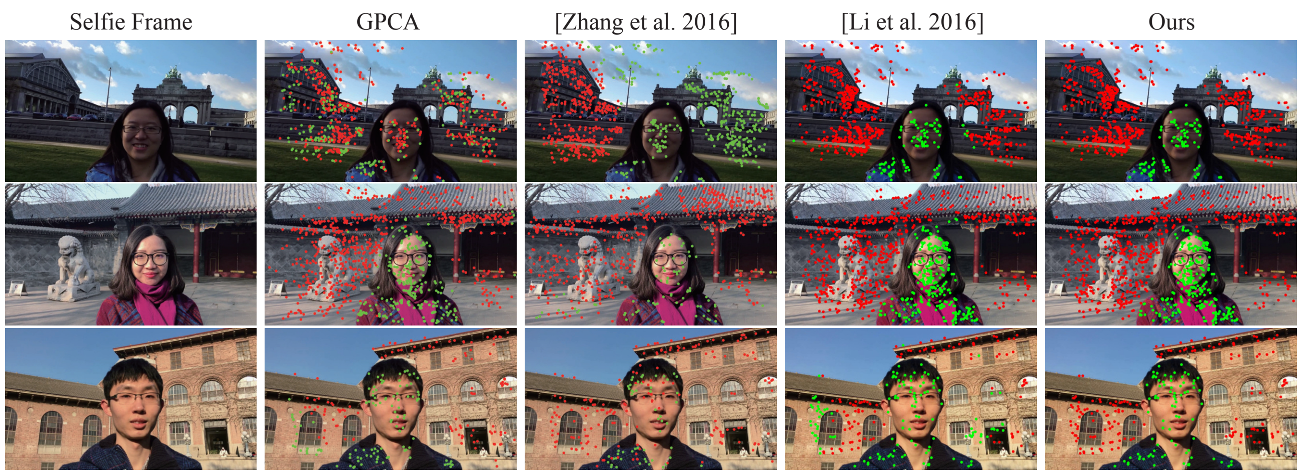


Fig. 4. Selfie foreground/background feature classification. From left to right: selfie frame, GPCA result, result from [Zhang et al. 2016], result from [Li et al. 2016], and our result. In each result, the identified foreground features are labeled as green dots, while background features are labeled as red dots.

than n frames. Therefore, we can select the same frame more than once.

A. Formulation

We formulate the frame selection as an energy minimization problem. In our formulation, five terms are considered, E_f, E_m, E_p, E_t and E_d, corresponding to frame matching, motion consistency, direction preference, temporal smoothness and distortion. Let $\Phi = \{\o_0, \o_1, \ldots, \o_n\}$ denote the sequence of indexes of the selected frames, and the 2D vector r denote one stitching direction. The cost function evaluating the retrieval result along stitching direction r is formulated as weighted sum of the penalties:

$$\begin{aligned} E(\Phi, r) =& \omega_f E_f(\Phi) + \omega_m E_m(\Phi) + \omega_p E_p(\Phi, r) \\ &+ \omega_t E_t(\Phi) + \omega_d E_d(\Phi), \end{aligned} \tag{4}$$

where $w_f = 1.0$, $w_m = 5.0$, $w_p = 1.0$, $w_t = 0.6$ and $w_d = 0.2$ are weighting factors that we tuned experimentally, for all of the examples in this paper. The novelty of this formulation is that we address motion consistency and direction guidance, which previous video matching approaches did not. Our frame selection is *motion-aware*, forcing the retrieved frames to have similar camera motion to the selfie video, which is crucial for joint stabilization and stitching. *Direction-awareness* selects frames according to the specified expansion directions. We note that the default parameters values balance the visual properties, and user can emphasize one or more visual properties by increasing the corresponding weights. Next, we introduce the different penalty terms.

Frame Matching. This is the primary term matching candidate environment frames to selfie frames. For each background region of a selfie frame I_t and a environment candidate frame $\widehat{I}_{\o_t}$, we extract $N = 1000$ SURF features [32] and compute a homography matrix H_t which aligns $\widehat{I}_{\o_t}$ to I_t using RANSAC

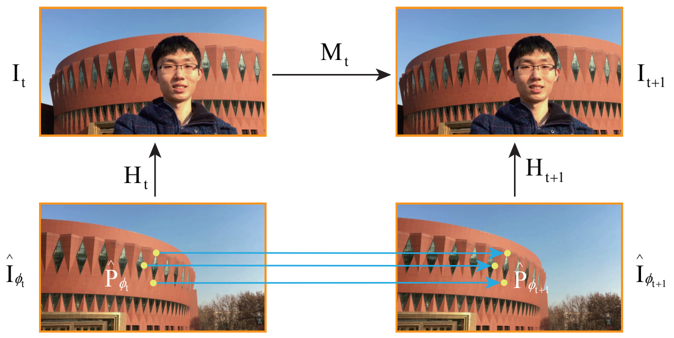


Fig. 5. The relationship of the transformations we used in motion awareness term. See text for details. The frames are from example AUDITORIUM, searching candidate frames to the RIGHT stitching direction.

[33] on sparse feature points. Following [34], given H_t, we estimate the matching error between frame I_t and $\widehat{I}_{\o_t}$ as:

$$\tau(t, \o_t) = \frac{1}{N} \sum_{i=1}^{N} ||P_t(i) - H_t \widehat{P}_{\o_t}(i)||_2, \tag{5}$$

where P_t and $\widehat{P}_{\o_t}$ are the positions of the matched feature pairs in the selfie frame and the candidate environment frame respectively, and N is the number of features. This cost measures the average 2D projection error using the transformation H_t. The frame matching term E_f for the whole candidate sequence Φ is then defined as $E_f(\Phi) = \sum_t \tau(t, \o_t)$.

Motion awareness. This term is designed to choose candidate environment frames that have similar motion to selfie video frames. Motion awareness is measured using two terms. First, the *original* motion of candidate frame sequence at $\o_t$ should follow the motion of the selfie video at frame t:

$$\pi_o(t, \o_t) = \frac{1}{R} \sum_{i=1}^{R} ||\widehat{P}_{\o_{t+1}}(i) - M_t \widehat{P}_{\o_t}(i)||_2, \tag{6}$$

where $\widehat{P}_{\o_t}$ and $\widehat{P}_{\o_{t+1}}$ are positions of the matched features in neighboring frames in the sequence Φ, R is the number of matched features, M_t is the homography transformation

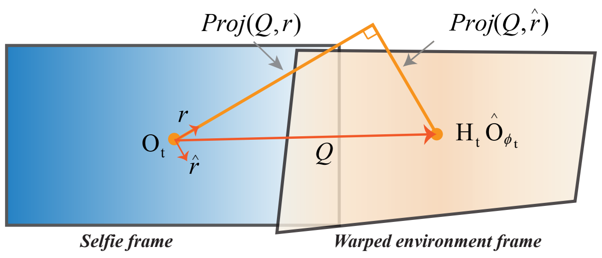


Fig. 6. The direction preference term measures how a candidate environment frame follows specified stitching direction. Vector Q connects the center of the selfie frame and the environment frame. This term encourages a large projection size of Q along given direction vector r: *Proj*(Q, r), while penalizing a large projection along the perpendicular direction: *Proj*$(Q, \widehat{r})$.

between neighboring selfie frames I_t and I_{t+1}, estimated based on the background layer features of the frames (see Figure 5).

Second, the *warped* motion, after aligning each candidate frame to its corresponding selfie frame, should also be as similar as possible to the corresponding selfie motion:

$$\pi_w(t, \o_t) = \frac{1}{R}\sum_{i=1}^{R} ||H_{t+1}\widehat{P}_{\o_{t+1}}(i) - M_t H_t \widehat{P}_{\o_t}(i)||_2. \quad (7)$$

We combine the two costs and accumulate them through the sequence Φ to get the motion awareness cost: $E_m(\Phi) = \sum_t (\pi_o(t, \o_t) + \pi_w(t, \o_t))$.

Direction preference. This term ensures that candidate frames follow specified expansion direction, and incorporate as much environment content as possible. To follow the specified expansion direction, the direction vector formed by selfie frame center and candidate environment frame center with alignment should be as close as possible to the expansion direction r. To introduce a large amount of new environment content, the distance between selfie frame center and candidate frame center with alignment should be large. Combining these two requirements, we assign a score to the matched frames I_t and $\widehat{I}_{\o_t}$ by penalizing small distances between the image centers along the direction r and large direction departure from r (see Figure 6):

$$\sigma(t, \o_t, r) = exp(-\lambda_1 \cdot Proj(Q, r)) + \lambda_2 Proj(Q, \widehat{r}) \quad (8)$$

where $Q = H_t\widehat{O}_{\o_t} - O_t$, O_t and $\widehat{O}_{\o_t}$ are the centers of selfie and candidate environment frames respectively, $\widehat{r}$ is the orthogonal direction vector of r, *Proj*(v_1, v_2) denotes the projection distance of vector v_1 along v_2, and $\lambda_1 = \lambda_2 = \frac{1}{framewidth}$ are constants. This term encourages choosing candidate frames with larger projection center distance along the direction r while having smaller direction departure from r. The direction preference term E_p for the whole candidate sequence Φ is then computed accumulatively as $E_p(\Phi, r) = \sum_t \sigma(t, \o_t, r)$.

Temporal smoothness. This term ensures smoothness of homography transformations for consecutive candidate environment frames. After aligning each candidate frame $\widehat{I}_{\o_t}$ to its corresponding selfie frame I_t via H_t, positions of the corners in consecutive candidate frames should change smoothly:

$$E_t(\Phi) = \frac{1}{4}\sum_t \sum_{i=1}^{4} ||H_{t+1}C_{\o_{t+1}}(i) - H_t C_{\o_t}(i)||_2, \quad (9)$$

where $C(i), i = 1, 2, 3, 4$ are the positions of four image corners.

Distortion. This term punishes candidate environment frames which are severely distorted after aligning to the selfie frame. For each candidate environment frame $\widehat{I}_{\o_t}$, we firstly warp it to the corresponding selfie frame I_t using H_t, then we calculate the warped frame's barycenter and align it to the barycenter of the original image frame with a transformation T_b. Like [35], the distortion term per frame is defined as the average corner distance between the original candidate frame corners and the transformed frame's corners. E_d for the whole candidate sequence Φ is computed as the sum of each frames distortion term:

$$E_d(\Phi) = \frac{1}{4}\sum_t \sum_{i=1}^{4} ||C_{\o_t}(i) - T_b H_t C_{\o_t}(i)||_2, \quad (10)$$

where $C(i), i = 1, 2, 3, 4$ are the positions of four image corners.

B. Optimization

To promote coherency and efficiency, we constrain the search of consecutive candidate environment frames by building a similarity graph from the environment video. In this similarity graph the nodes are candidate frames $\widehat{I}$ in the environment video. We connect two nodes in the graph with an edge if they are consecutive in the original video or the homography between them has a small distortion and residual error. We regard a sequence Φ as valid only if for all $1 \leq t \leq n-1$, $\widehat{I}_{\o_{t+1}} \in \mathcal{N}(\widehat{I}_{\o_t})$, where $\mathcal{N}(\widehat{I}_{\o_t})$ is the set of neighbors of $\widehat{I}_{\o_t}$ in the similarity graph. Using this constraint, our objective function for optimization is defined as:

$$\Phi' = \underset{\Phi=\{\o_1,\o_2,\ldots,\o_n\}}{\arg\min} \; E(\Phi) \quad s.t. \quad \widehat{I}_{\o_{t+1}} \in \mathcal{N}(\widehat{I}_{\o_t}), 1 \leq t \leq n-1.$$

We efficiently solve this optimization using dynamic programming. For each neighboring selfie frame pair $\{I_t, I_{t+1}\}$, we enumerate all valid candidate frames $\dot{\Phi} = \{\widehat{I}_{\o_t}, \widehat{I}_{\o_{t+1}}\}$ s.t. $\widehat{I}_{\o_{t+1}} \in \mathcal{N}(\widehat{I}_{\o_t})$, and calculate the local cost $E(\dot{\Phi})$ using Equation 4. We iteratively compute and update the accumulative minimum cost for the sub-sequence $\ddot{\Phi} = \{\o_1, \o_2, \ldots, \o_{t-1}, k\}$ of the first t frames ending with a candidate frame $\widehat{I}_k$. The accumulative minimum cost and corresponding sub-sequence $\ddot{\Phi}$ are recorded in look-up tables $A(t, k)$ and $B(t, k)$. The optimal solution Φ' is obtained by looking for the minimum value $A(n, \cdot)$ and corresponding candidate frame set $B(n, \cdot)$. The pseudocode of the dynamic programming is provided in our supplemental material.

VI. Stabilization and Stitching

After selecting the candidate environment frame sequence for each stitching direction r, the final step of our method is to seamlessly stitch them to the selfie video and stabilize the whole content while preserving the portrait region. We represent the frames using a 2D mesh model and solve the stitching and stabilization problems jointly by minimizing an objective function defined at positions of the mesh vertices. After optimization, we warp all frame pixels using bilinear interpolation driven by moving vertices.

Each video frame is divided into 10×10 regular grids. Let V denote all the vertex positions of this mesh. To obtain optimized positions V', our objective function jointly considers stabilization, stitching, as well as preventing the selfie portrait from distortion. The objective function has three components:

$$V' = \arg\min_{V} \left(D_{stab.}(V) + \beta_1 D_{stit.}(V) + \beta_2 D_{shape}(V)\right), \tag{11}$$

where $D_{stab.}(V)$ is a *portrait-preserving stabilization* term, $D_{stit.}(V)$ is a *stitching* term, and $D_{shape}(V)$ is a *shape-preserving* regularization term; $\beta_1 = 0.8$ are $\beta_2 = 1.0$ are weights, balancing the importance of portrait-preserving stabilization error, stitching error and shape preservation error. We fixed them for all examples in this paper. As will be described below, all the terms in Equation 11 are quadratic, and thus can be directly minimized by solving a sparse linear system.

Portrait-preserving Stabilization Term. As introduced in [11], video stabilization can be performed by smoothing a bundle of camera paths formed by mesh grids. Their objective function contains a data term forcing mesh vertices not to drift from their original positions and a temporal smoothness term removing jitters. Because we stabilize and stitch frames jointly, we allow vertices of environment video frames to move from their original positions for frame alignment. Therefore, we do not use data term, and replace it with a portrait-preserving constraint as follows:

$$\begin{aligned} D_{stab.}(V) = &\sum_t \sum_i \lambda_p \sum_{\dot{v}_{t,i} \in \Psi_F(t)} ||v_{t,i} - \dot{v}_{t,i}||^2 \\ &+ \sum_t \sum_i \lambda_t \sum_{j \in \Omega_t} \omega_{t,j} ||v_{t,i} + \dot{\theta}_{t,i} - (v_{j,i} + \dot{\theta}_{j,i})||^2, \end{aligned}$$

where $v_{t,i}$ denotes i-th vertex position at time t to optimize. Firstly, in the portrait preserving term, Ψ_F are foreground masks obtained for each frame after feature identification, $\dot{v}_{t,i}$ are the original mesh vertex positions of the selfie video at time t. Secondly, in the path smoothing term, $\dot{\theta}_{t,i}$ is the original camera path of the i-th vertex at time interval $[1, t]$, and Ω_t is the local temporal window centered around t, and $\omega_{t,j}$ is a temporal smoothness weight calculated using Gaussian kernel. The two terms are balanced by a weighting factor $\lambda_p = 2.0$ and $\lambda_t = 1.0$.

Stitching Term. The stitching term ensures that the content from salfie frames and selected environment frames are well

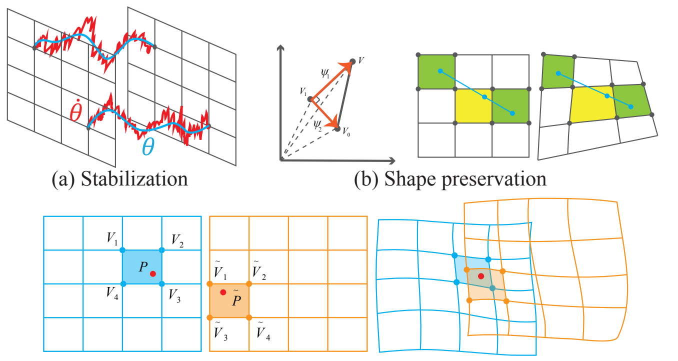


Fig. 7. Illustrations of terms used in stabilization and stitching. (a) The stabilization term temporally smooths the original vertex paths $\dot{\theta}$ to θ. (b) The similarity transformation constraint on the mesh grid, and the line preservation in shape-preserving term; (c) Stitching two frames is done by aligning the matched feature points using mesh warping.

aligned. Let P_t and $\widetilde{P}_t^k$ be positions of matched feature pairs between a selfie frame I_t, and the corresponding environment frame along stitching direction r, $\widetilde{I}_t^r$. We represent the position of each feature $P_t(i)$ using 2D bilinear interpolation of the vertex positions of its grid cell $V_t(i) = [v_t^1(i), v_t^2(i), v_t^3(i), v_t^4(i)]$, with interpolation weights $w_t(i) = [w_t^1(i), w_t^2(i), w_t^3(i), w_t^4(i)]^T$: $P_t(i) = w_t(i)V_t(i)$. Each feature $\widetilde{P}_t^r(i)$ in the environment frame $\widetilde{I}_t^r$ selected for direction r, can be represented similarly by $\widetilde{P}_t^r(i) = \widetilde{w}_t^r(i)\widetilde{V}_t^r(i)$. Like [24], we define the stitching term as follows:

$$\begin{aligned} D_{stit.}(V) = &\sum_t \sum_r \sum_i ||w_t(i)V_t(i) - \widetilde{w}_t^r(i)\widetilde{V}_t^r(i)||^2 \\ &+ \sum_t \sum_{\langle a,b \rangle} \sum_j ||\widetilde{w}_t^a(j)\widetilde{V}_t^a(j) - \widetilde{w}_t^b(j)\widetilde{V}_t^b(j)||^2. \end{aligned} \tag{12}$$

The first part ensures that feature points of each environment frame align with their matched points in selfie frame after warping, for all directions r. If there are more than one directions of expansion, the second part ensures that matched features from environment frames of any pair of directions (a and b) are also aligned after warping.

Shape-preserving Term. We consider two shape preserving terms $D_{shape}(V) = D_g(V) + 3 \cdot D_l(V)$: one for grid shape and one for line shape. We use the same grid-preserving term as in [1]:

$$D_g(V) = \sum_t \sum_i ||\psi_1 - sR_{90}\psi_2||^2, R_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{13}$$

where $\psi_1 = v_{t,i} - v_1$ and $\psi_2 = v_0 - v_1$, $v_{t,i}$, v_0 and v_1 are neighboring vertices and $s = ||v - v_1||/||v_0 - v_1||$ is the ratio of grid side lengths of the grid.

To preserve line structure, we use a line preserving term from [24] based on detected line segments [36]:

$$D_l(V) = \sum_t \sum_l \sum_{i=1}^{L-1} ||((1-\eta)V_{t,0}^l w_{t,0}^l + \eta V_{t,L}^l w_{t,L}^l) - V_{t,i}^l w_{t,i}^l||^2,$$

where $V_{t,0}^l$ and $V_{t,L}^l$ are vertices of grids containing two end points of line segment l (see green grids in Figure 7 (b)), and $\eta = \frac{i}{L}$. $V_{t,i}^l$ are the vertices of grids containing the i-th sample point of line segment l in frame t (see yellow grids in Figure 7 (b)), and $w_{t,i}^l$ are the corresponding weights. We use $L = 5$ sample points in line with [24].

VII. Implementation

A. Pre-processing

In pre-processing, we track and match features in videos and build the similarity graph for acceleration.

Selfie Feature Trajectory Extraction. In our implementation, a KLT tracker [37] is used as the basis for feature trajectory extraction. The original KLT feature points on the background could be perturbed by portrait occlusion, leading to drifting effects. Thus, we also perform a backward KLT tracking for the extracted feature points in each selfie video frame. We regard feature trajectories valid if after backward tracking the feature positions move back to the original position in the previous frame. We extract 1000 feature points in each frame, and discard very short trajectories of less than 20 frames. This leaves about 800+ features in each frame for background feature identification.

Frame Matching and Similarity Graph Construction. As described in Section V-B, we pre-construct a similarity graph [35] to determine valid transforms between candidate environment frames. For each frame $\widehat{I_t}$ in the environment video, we estimate its neighboring frames $\mathcal{N}(\widehat{I_t})$ in the similarity graph. First, we regard frames in a temporal window of $[t-20, t+20]$ in the original environment video as neighbors. Next we match $\widehat{I_t}$ to remote frames $\widehat{I_s}$ outside this temporal window using a homography computed on SURF features. We regard $\widehat{I_s} \in \mathcal{N}(\widehat{I_t})$ if the residual projection error $r_{t,s} < 0.3$ and the distortion $d_{t,s}$ computed by Equation 10 is smaller than $0.05\times$ image width.

We further pre-compute the pair-to-pair frame matching between selfie frames I_i and environment frames $\widehat{I_j}$ to compute homographies $H_{i,j}$, and compute the transformations M_t between neighboring selfie frames using identified background features F_B (see Section V-A). To accelerate the process of frame selection, all the frame matchings are calculated once and stored in tables.

B. Post-processing

The aligned frames after stitching need to be seamlessly blended together. We used the Graphcut textures algorithm [38] to find split surfaces among the 3D video voxels that minimize alignment error between the selected video content and the selfie video. We hierarchically optimize the split surface in three levels: we down sample frames with a scale factor of 2.0 three times. At the lowest resolution, we perform a full 3D graphcut texture on the coarsest level and obtain a seam in each frame. At higher resolutions, we use the seam position in the previous level as an initialization and calculate a new seam within a band whose width equals to 0.1 times the image width. In each time the portrait content is forced to appear in the result. After obtaining the seams on the highest resolution, we finally use multiple band blending [39] to render the output videos by blending the content from selfie video and environment video.

VIII. Results

We used our algorithm on 10 examples, with the same parameters settings reported previously. The input selfie videos in our examples were captured using a mobile phone fixed on a selfie-stick. Various camera paths, motions, shooting distance and background content are included in the examples. Snapshots of some results are shown in Figure 8, and the full input videos and results are provided in the supplementary video. In Table I, we report the average error of each mesh grid vertex according to Equation 11.

A. Performance

Our algorithm was implemented on an Intel Core i7-4790K CPU at 4.0GHz. The run-time for some examples are reported in Table II. Our *unoptimized* C++/Matlab code was executed on a CPU *without* GPU acceleration which could be certainly utilized for performance improvement.

B. Evaluations

Foreground/background Feature Classification. We conducted a quantitative evaluation on the proposed foreground/background feature classification method for selfie videos. The experiments were carried on a test set with 12 selfie video clips captured by mobile phone with various camera motions: standing, walking and spinning. The ground truth foreground and background features were labeled by 5 recruited labelers using paint brushes. For each feature trajectory, its label is confirmed only if the majority of labelers agree. Overall, there were 38,573 feature trajectories in the test set, in which 11,587 are labeled as background and all other are labeled as foreground.

We compared our algorithm against motion segmentation algorithm GPCA [40], and foreground/background identification methods [3], [20]. As GPCA algorithm cannot handle incomplete feature trajectories, we completed the missing parts of trajectories with the average position in previous frames before running these algorithms. Figure 4 shows some representative frames of our results and alternatives.

Table III shows the quantitative evaluation of the feature classification results. We compared the average background feature precision and recall in the test set. As can be seen our method outperforms the alternatives on these videos. Full comparisons are provided in supplemental material. With more accurate background feature trajectories, the selfie camera

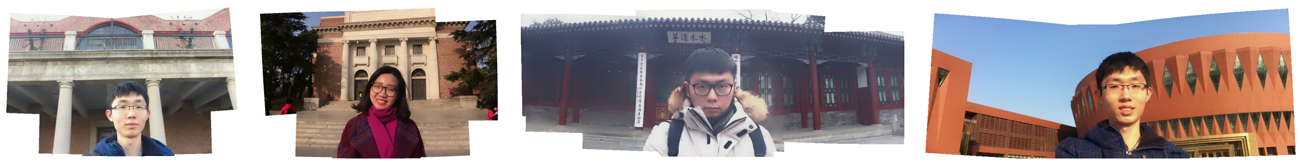

(a) GYMNASIUM (b) THEATER (c) OFFICE (d) AUDITORIUM

Fig. 8. Representative frames from BiggerSelfie videos. The supplemental material contains full videos.

TABLE I
AVERAGE ERROR OF EACH MESH VERTEX IN STABILIZATION AND STITCHING OPTIMIZATION FROM EACH EXAMPLE.

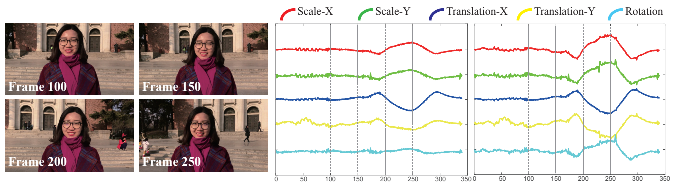
Video	Auditorium	Square	Gymnasium	Library	Theater	Office	Main Building	Museum	Pavilion	Building
Error	4.094	3.821	2.510	2.899	5.360	10.558	3.270	12.537	6.113	3.285

TABLE II
PERFORMANCE OF OUR ALGORITHM ON SELECTED EXAMPLES. EACH COLUMN SHOWS RUN-TIMES IN SECONDS FOR ALGORITHM STAGES.

Example	SQUARE	THEATER	OFFICE
Selfie Frames	176	341	259
Environment Frames	241	312	348
Stitching Directions	LR	LUR	CIRC
Timing			
Selfie Feature Class.	201s	490s	332s
Frame Selection Pre-pro.	288s	727s	649s
Frame Selection Opt.	72s	125s	235s
Joint Stitch. & Stab.	30s	77s	59s

TABLE III
QUANTITATIVE EVALUATION OF FEATURE CLASSIFICATION METHODS.

	Accuracy	Recall	F1-score
Our method	**0.903**	**0.882**	**0.892**
GPCA	0.675	0.552	0.607
Zhang 2016	0.808	0.775	0.791
Li 2016	0.826	0.819	0.822



(a) Background points (b) All points

Fig. 9. Comparing the homography parameters estimated with our background feature selection (a) and using all feature points (b). The curves show the parameters of homographies computed between each neighboring frame pair.

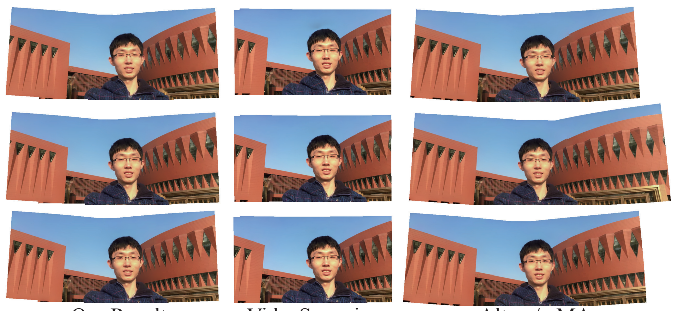

Our Result VideoSnapping Alt. w/o MA

Fig. 10. Successive output frames from frame selection algorithms. Each column shows three successive output expansion frames from top to bottom. From left to right: our result, result from VideoSnapping and result from alternative frame searching without motion-awareness ($\omega_m = 0$).

motion estimation is more robust than using all the feature trajectories. Figure 9 shows the changes of Homography parameters in THEATER example. As can be seen, using just background features provides more stable results.

Evaluation on Motion-direction-aware Frame Selection. We evaluate the proposed motion-direction-aware frame selection by comparing against the alternative searching strategy without *motion-awareness* term (short as "Alt. w/o MA"), and the frame matching method from VideoSnapping [41]. VideoSnapping matches frames between videos based on content similarity without *motion-awareness* and *direction-awareness*. Figure 10 shows successive *BiggerSelfie* frames generated from alternative frame selection methods. Generally, our frame selection achieves high motion consistency while maintaining large field-of-view. To quantitatively compare the performance of different strategies, we evaluate the content richness as well as motion coherence of the introduced content to the original selfie motion. For content richness, we measure the areal increment in the stitched result against the original selfie video, see Table IV. For motion coherence, we estimate the camera motions in the selfie expansion result, and compare the motions of the warped selfie frames against the warped environment frames from different selected frame sets. The accumulative Homography parameters of results from example AUDITORIUM are illustrated in Figure 11. The evaluation indicates that our frame selection method is able to achieve both content richness and motion coherence, which are essential for selfie video stitching problem.

Evaluation on Stabilization and Stitching. We evaluate the portrait-preserving stabilization and stitching against the approach of [5], which does not preserve the foreground portrait. As the comparative approach assumes videos are synchronized before processing, we feed it environment frames from our frame selection. As can be seen from Figure 12 and supplementary video, our method can prevent portrait distortion and drifting while the alternative cannot.

Further tests. We conducted experiments using the same input selfie video with different environment videos. The

TABLE IV
CONTENT RICHNESS OF STITCHED RESULTS. VALUES IN THE TABLE ARE THE RATIO OF AREA IN THE OUTPUT VIDEOS TO THE ORIGINAL SELFIE VIDEOS.

	SQUARE	AUDITORIUM	OFFICE
Our method	1.76	1.59	2.14
VideoSnapping	1.15	1.31	1.35
Alt. w/o MA	1.77	1.62	2.25

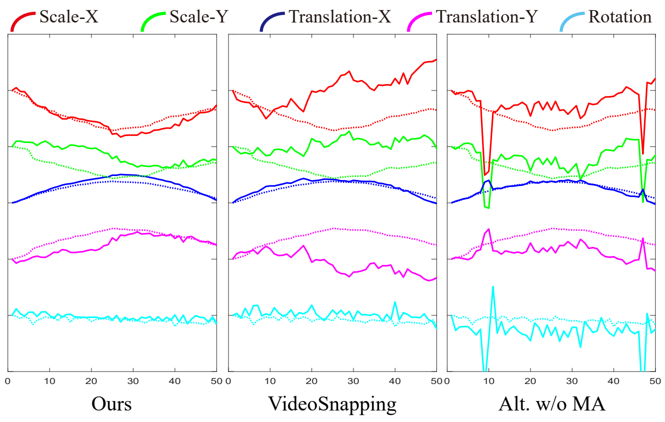


Fig. 11. Comparing homography parameters estimated from warped frames in the expansion result within a 50 frame duration. The curves show the accumulative parameter values of the homographies computed between each neighboring frame pair. From left to right: our motion-direction-aware frame selection, VideoSnapping, and frame searching without motion-awareness. In each plot, solid curves represent accumulative parameters estimated from the warped environment frames returned by the corresponding method, and dashed curves represent the accumulative parameters estimated from the warped selfie frames.

original camera motion of the environment videos are different while the contents are similar. Figure 13 (e) and (f) show two corresponding *BiggerSelfie* frames expanded along up direction using the same selfie video (see Figure 13 (a)) and different environment videos (see Figure 13 (c) and (d)). In Figure 13 (c), the camera from environment video moves generally from left to right, and in Figure 13 (d), the camera from environment video moves from right to left. The visually similar results as shown in Figure 13 (e) and (f) demonstrate the robustness of our system.

We also tested the robustness of expansion using selfie video with a very large portrait ratio. Selfie video as shown in Figure 13 (a) was cropped into Figure 13 (b) so that very limited background content left inside the frame. We ran the expansion algorithm using this cropped selfie video and environment video from Figure 13 (d). Figure 13 (g) shows the corresponding *BiggerSelfie* result, which is visually similar to Figure 13 (f).

C. Limitations

Our system has limitations. The first limitation is that structure misalignment in the stitched result can sometimes be visible. Such problem arises where frame regions lack matching features or include non-rigid object with inconsistent appearance, which is challenging for video stitching algorithms [4], [5].

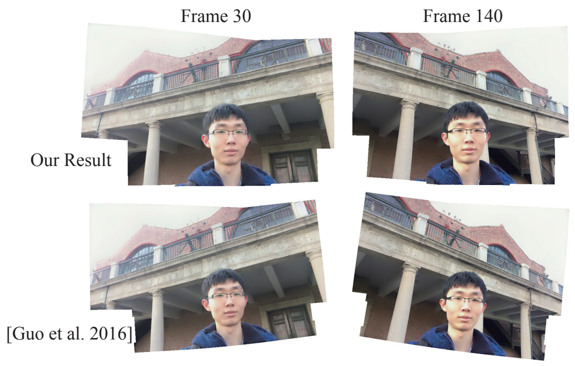


Fig. 12. Comparison against state-of-the-art hand-held video stitching algorithm [Guo et al. 2016] on selfie video. In our result the portrait is stable while in the alternative result, the portrait is distorted and drifted.

Secondly, as our method finds and selects non-consecutive candidate frames from environment video and performs graph-cut post-processing to find 3D seams, the semantic dynamics of objects in the original environment video may not be kept in the results. For example, a walking person in the background could suddenly walk backward or disappear. In some cases this problem can be solved by restricting the searching window to the following frames temporally, disabling backward and distinct frame searching. However, this comes at the cost of quality from the viewpoint of content and motion consistency.

Thirdly, if the portrait in selfie video is occluded by frame boundaries, our expansion algorithm is not able to recover the missing portrait.

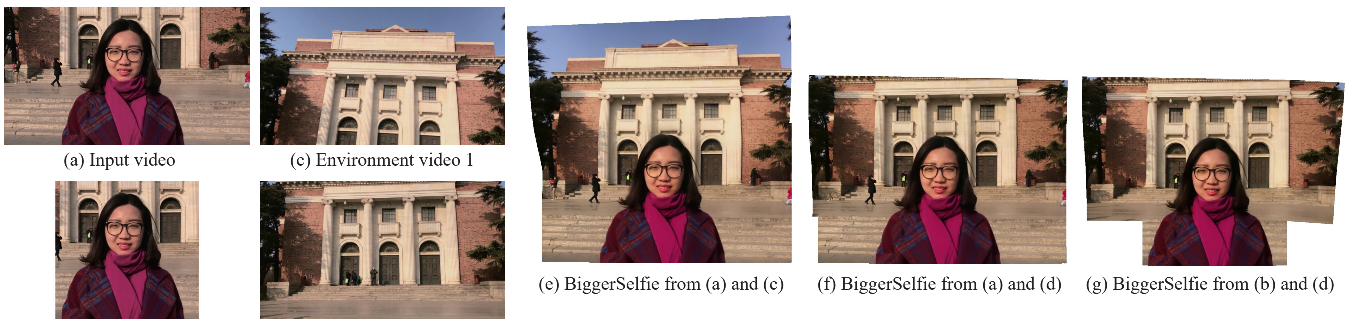
Lastly, while the proposed algorithm is able to significantly expand background content of selfie video, the expansion results are not in rectangular shape. This is common for video stitching works [24], [5], and could be improved using video completion methods, which is not the focus of this work.

IX. CONCLUSION

We have presented a method which takes a selfie video clip and an environment video as input, and outputs a *BiggerSelfie* video, with a larger field-of-view and more background content. To our knowledge, this paper presents the first approach to handle the selfie video expansion problem. Beyond this, we have also developed several key algorithms such as foreground and background feature trajectory classification in selfie video, motion-direction-aware frame selection and portrait-preserving stabilization and stitching. As selfie photography continues to attract the public attention, selfie editing applications and techniques such as the one proposed in this paper are becoming more desirable.

ACKNOWLEDGMENT

The authors would like to thank all the reviewers. This work was supported by National Natural Science Foundation of China (Project Number 61561146393 and 61521002) and China Postdoctoral Science Foundation (Project Number



(a) Input video (c) Environment video 1

(b) Cropped video from (a) (d) Environment video 2

(e) BiggerSelfie from (a) and (c) (f) BiggerSelfie from (a) and (d) (g) BiggerSelfie from (b) and (d)

Fig. 13. Further tests on large portrait ratio and different environment videos. (a) shows an input frame from the selfie video; (b) shows a video frame cropped from (a); (c) shows the selected frame from an environment video moving from left to right for selfie video (a); (d) shows the selected frame from another environment video moving from right to left for selfie video (a); (e) shows *BiggerSelfie* result from (a) and (c); (f) shows *BiggerSelfie* result from (a) and (d); (g) shows *BiggerSelfie* result from (b) and (d).

2016M601032). Ariel Shamir was supported by the Israel Science Foundation (Project Number 2216/15). Shao-Ping Lu was supported by Innoviris (3DLicornea).

References

[1] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, “Subspace video stabilization,” *ACM Trans. Graph.*, vol. 30, no. 1, pp. 4:1–4:10, 2011.

[2] S. Liu, L. Yuan, P. Tan, and J. Sun, “Bundled camera paths for video stabilization,” *ACM Trans. Graph.*, vol. 32, no. 4, pp. 78:1–78:10, 2013.

[3] F.-L. Zhang, X. Wu, H.-T. Zhang, J. Wang, and S.-M. Hu, “Robust background identification for dynamic video editing,” *ACM Trans. Graph.*, vol. 35, no. 6, pp. 197:1–197:12, 2016.

[4] W. Jiang and J. Gu, “Video stitching with spatial-temporal content-preserving warping,” in *Proc. CVPR Workshops*, 2015, pp. 42–48.

[5] H. Guo, S. Liu, T. He, S. Zhu, B. Zeng, and M. Gabbouj, “Joint video stitching and stabilization from moving cameras,” *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5491–5503, 2016.

[6] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, “Full-frame video stabilization with motion inpainting,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, 2009.

[7] 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, 2009.

[8] M. Grundmann, V. Kwatra, and I. Essa, “Auto-directed video stabilization with robust l1 optimal camera paths,” in *Proc. CVPR*. IEEE, 2011, pp. 225–232.

[9] M. Okade, G. Patel, and P. K. Biswas, “Robust learning-based camera motion characterization scheme with applications to video stabilization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 453–466, 2016.

[10] A. Kowdle and T. Chen, “Learning to segment a video to clips based on scene and camera motion,” in *Proc. ECCV*. Springer, 2012, pp. 272–286.

[11] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, “Meshflow: Minimum latency online video stabilization,” in *Proc. ECCV*. Springer, 2016, pp. 800–815.

[12] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, “Content-preserving warps for 3d video stabilization,” *ACM Trans. Graph.*, vol. 28, no. 3, pp. 44:1–44:9, 2009.

[13] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, “Video stabilization with a depth camera,” in *Proc. CVPR*, 2012, pp. 89–95.

[14] A. Goldstein and R. Fattal, “Video stabilization using epipolar geometry,” *ACM Trans. Graph.*, vol. 31, no. 5, pp. 126:1–126:10, 2012.

[15] J. Kopf, “360 video stabilization,” *ACM Trans. Graph.*, vol. 35, no. 6, pp. 195:1–195:9, 2016.

[16] Y. Ma, H. Derksen, W. Hong, and J. Wright, “Segmentation of multivariate mixed data via lossy data coding and compression,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, 2007.

[17] S. Rao, R. Tron, R. Vidal, and Y. Ma, “Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 32, no. 10, pp. 1832–1845, 2010.

[18] Y. Nakajima and H. Saito, “Robust camera pose estimation by viewpoint classification using deep learning,” *Computational Visual Media*, vol. 3, no. 2, pp. 189–198, 2017.

[19] A. Papazoglou and V. Ferrari, “Fast object segmentation in unconstrained video,” in *Proc. ICCV*. IEEE, 2013, pp. 1777–1784.

[20] K. Li, J. Wang, Y. Liu, L. Xu, and Q. Dai, “Re-compositable panoramic selfie with robust multi-frame segmentation and stitching,” *Comput. Graph. Forum*, vol. 35, no. 7, pp. 227–236, 2016.

[21] M. Irani, P. Anandan, and S. Hsu, “Mosaic based representations of video sequences and their applications,” in *Proc. ICCV*. IEEE, 1995, pp. 605–611.

[22] A. Agarwala, K. C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, “Panoramic video textures,” in *ACM Trans. Graph.*, vol. 24, no. 3. ACM, 2005, pp. 821–827.

[23] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross, “Panoramic video from unstructured camera arrays,” *Comput. Graph. Forum*, vol. 34, no. 2, pp. 57–68, 2015.

[24] K. Lin, S. Liu, L.-F. Cheong, and B. Zeng, “Seamless video stitching with hand-held camera inputs,” *Comput. Graph. Forum*, vol. 35, no. 2, pp. 479–487, 2016.

[25] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs, “Automatic portrait segmentation for image stylization,” *Comput. Graph. Forum*, vol. 35, no. 2, pp. 93–102, 2016.

[26] H. Huang, X. Fang, Y. Ye, S. Zhang, and P. L. Rosin, “Practical automatic background substitution for live video,” *Computational Visual Media*, vol. 3, no. 3, pp. 273–284, 2017.

[27] M. Wang, Y.-K. Lai, Y. Liang, R. R. Martin, and S.-M. Hu, “Biggerpicture: Data-driven image extrapolation using graph matching,” *ACM Trans. Graph.*, vol. 33, no. 6, pp. 173:1–173:13, 2014.

[28] S. Liu, Y. Zhang, X. Yang, D. Shi, and J. J. Zhang, “Robust facial landmark detection and tracking across poses and expressions for in-the-wild monocular video,” *Computational Visual Media*, vol. 3, no. 1, pp. 33–47, 2017.

[29] S. Milborrow and F. Nicolls, "Active shape models with sift descriptors and mars," in *Proc. VISAPP*, 2014, pp. 380–387.

[30] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998, pp. 839–846.

[31] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. ECCV*. Springer, 2006, pp. 404–417.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] M. Wang, J. B. Liang, S. H. Zhang, S. P. Lu, A. Shamir, and S. M. Hu, "Hyper-lapse from multiple spatially-overlapping videos," *IEEE Trans. Image Processing*, vol. 27, no. 4, pp. 1735–1747, 2018.

[35] F. Zhong, S. Yang, X. Qin, D. Lischinski, D. Cohen-Or, and B. Chen, "Slippage-free background replacement for hand-held video," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 199:1–199:11, 2014.

[36] R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.

[37] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[38] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, 2003.

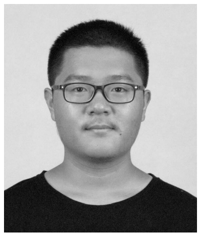
[39] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graph.*, vol. 2, no. 4, pp. 217–236, 1983.

[40] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and gpca," *Int. J. Comput. Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[41] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung, "Videosnapping: Interactive synchronization of multiple videos," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 77:1–77:10, 2014.

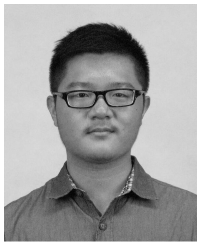
Miao Wang is currently a postdoc researcher at Tsinghua University. He received Ph.D. degree from Tsinghua University in 2016. His research interests are in visual computing, including data-driven image editing and interactive video editing.

Ariel Shamir is a professor at the Efi Arazi school of Computer Science at the Interdisciplinary Center in Israel. He received a B.Sc. and M.Sc. degrees in math and computer science Cum Laude from the Hebrew University in Jerusalem, and a Ph.D. in computer science in 2000. After that, he spent two years as a post-doctoral fellow at the computational visualization center at the University of Texas in Austin. He was a visiting scientist at Mitsubishi Electric Research Labs in Cambridge MA (2006), and at Disney Research (2014). His research interests include geometric modeling, computer graphics, fabrication, visualization, and machine learning. He is a member of the ACM SIGGRAPH, IEEE Computer and Eurographics societies.



Guo-Ye Yang is an undergraduate at Tsinghua University. His research interests include computer graphics, image analysis and computer vision.

Jin-Kun Lin is an undergraduate at Tsinghua University. His research interests include computer graphics, image analysis and computer vision.



Guo-Wei Yang is an undergraduate student at Tsinghua University. His research interests include computer graphics, image analysis and processing, video editing.

Shao-Ping Lu is currently an associate professor at Nankai University in Tianjin, China. In 2013-2017, he worked as a postdoc and senior researcher at Vrije Universiteit Brussel (VUB). In 2012 he received PhD degree at Tsinghua University in China. His primary research areas are image & video processing and editing, with particular interests in multi-view video construction, representation and compression.

Shi-Min Hu received the Ph.D. degree from Zhejiang University, in 1996. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing. He has authored over 100 papers in journals and refereed conference. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is the Editor-in-Chief of Computational Visual Media, and on the Editorial Board of several journals, including the IEEE Transactions on Visualization and Computer Graphics, and Computer Aided Design and Computer and Graphics.