Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Re-Compositable Panoramic Selfie with Robust Multi-Frame Segmentation and Stitching

Kai Li[1,2], Jue Wang[3], Yebin Liu[2], Li Xu[4], and Qionghai Dai[2]

[1]School of Communication and Information Engineering, Shanghai University
[2]Department of Automation, Tsinghua University [3]Adobe Research [4]SenseTime Group Limited

Figure 1: *We created a panoramic selfie that not only has a much wider background coverage, but also is re-compositable under different user-selected foreground position.*

Abstract

It is a challenging task for ordinary users to capture selfies with a good scene composition, given the limited freedom to position the camera. Creative hardware (e.g., selfie sticks) and software (e.g., panoramic selfie apps) solutions have been proposed to extend the background coverage of a selife, but to achieve a perfect composition on the spot when the selfie is captured remains to be difficult. In this paper, we propose a system that allows the user to shoot a selfie video by rotating the body first, then produce a final panoramic selfie image with user-guided scene composition as postprocessing. Our key technical contribution is a fully automatic, robust multi-frame segmentation and stitching framework that is tailored towards the special characteristics of selfie images. We analyze the sparse feature points and employ a spatial-temporal optimization for bilayer feature segmentation, which leads to more reliable background alignment than previous image stitching techniques. The sparse classification is then propagated to all pixels to create dense foreground masks for person-background composition. Finally, based on a user-selected foreground position, our system uses content-preserving warping to produce a panoramic seflie with minimal distortion to the face region. Experimental results show that our approach can reliably generate high quality panoramic selfies, while a simple combination of previous image stitching and segmentation approaches often fails.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing Algorithms

1. Introduction

Selfie, *i.e.*, self-portrait photography, has become a global phenomenon as a new form of self expression with massive user base in the past few years. A large portion of selfies are intended to capture a meaningful background, *e.g.*, tourists often take self-portraits in front of memorable landmarks. In such cases, capturing selfies with a good scene composition is challenging for the traditional

(a) (b)

Figure 2: *Results generated by the PanoSelfie app [Mob] when the user rotates (a) the camera only or (b) her body with the camera. This app can only create a panoramic selfie with the user in the center of the final image.*

photography, for a few reasons. Firstly, given the short object distance, the background is largely occluded by the person, and thus the user has to carefully adjust the camera to capture a good part of the background using very limited coverage. Secondly, because the user faces away from the background, he/she can only see a portion of it that is currently displayed on the screen, thus may not have a good idea on how to move the camera to improve the composition. Furthermore, when taking a selfie the user usually tends to adjust the camera to make him/herself look better in the final picture, which may conflicts with the goal of creating a better composition.

Creative solutions in both hardware and software have been proposed to create selfies with better background coverage. Firstly, the selfie stick is a practical and effective solution to broaden the view angle. However they are additional hardware pieces that may not always be available when needed, and large field of view is still not possible. Secondly, commercial apps such as PanoSelfie [Mob] shown in Fig. 2 allow one to create panoramic selfies, by rotating the camera to shoot a series of images, and using the built-in Gyro sensor signal to align them and stitch a panorama. However it produces a result where the person always appears at the center of the final image, which is often not desirable from composition and photo aesthetics point of view. Furthermore, it cannot handle large parallax caused by foreground occlusion. Thirdly, approaches extending the field of view of a photo like photo uncrop [SCF*14] do not work well for selfies due to the large occlusion when estimating the camera pose using structure from motion. Fourthly, another possible solution for background extension is to capture a panorama first, then capture a selfie, and use image matching and composition techniques to blend the selfie into the panorama. However, as shown in Sec. 5, this process cannot guarantee the selfie's proper embedding and blending. In addition, it still does not solve the composition problem, as the final foreground position is solely determined by the captured selfie and cannot be changed afterwords.

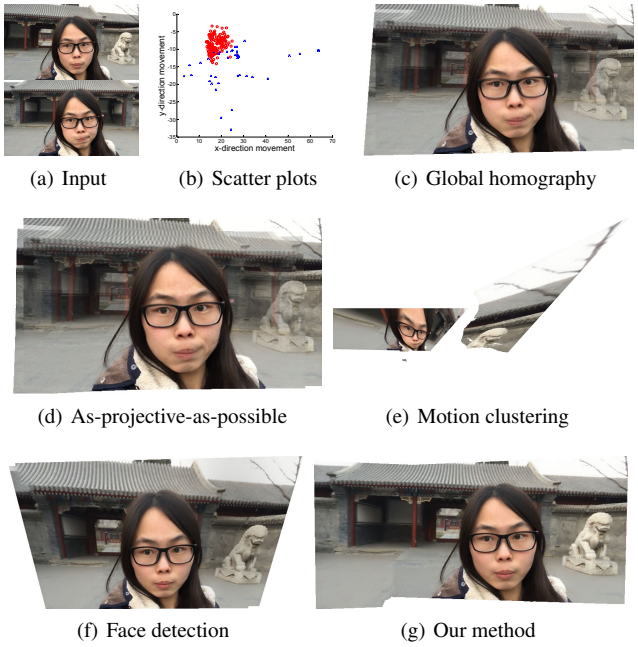
In this paper we propose a computational approach that allows the user to shoot a selfie video first, then create a panoramic selfie and determine a good composition as post-processing, for the first time, which is shown in Fig. 1. At capturing time, we ask the user to capture a short selfie video by rotating the *body with the camera*, so that the person appears with desired pose and expression on most frames, but with different background regions. Our approach then analyzes the video, separates the person from the background, align frames with the background points, and finally creates a seamless composite based on a user-specified head position. Our approach thus can produce a selfie that not only has a much wider background coverage, but also allows the user to interactively change the foreground position in the image to achieve a better person-background composition.

At first glance our system seems to be solving two standard problems: foreground segmentation and panoramic image stitching, which are well-studied in the literature. However, as we will show later, surprisingly, selfie images pose unique challenges that can fail most previous panoramic stitching approaches, and make previous motion segmentation methods produce inferior results. For instance, the large foreground occlusion makes background alignment extremely hard, even with the commonly used outlier handling routines such as RANSAC. The foreground may present dynamic expression change thus its motion may not fit a linear motion model. The local foreground and background colors may be similar in some frames, making segmentation confusing. As shown in Sec. 5, simply combining exiting approaches do not produce satisfactory results in this application.

2. Related Work

First, our work is directly related to previous research on image stitching. Image stitching has been extensively studied and is often considered as a solved problem: matured techniques can be found in many commercial products [Ado, Aut]. Please refer to the tutorial [Sze06] for a comprehensive overview on the fundamental principles. While most previous approaches assume static scenes with rotational camera motion or with no parallax [BL07, STP12], recent efforts have been put into offering compensation to the model inadequacy when handling complex scenes. Gao *et al.* [GKB11] presented a dual-homography warping strategy to handle scenes with two planes, *i.e.*, the dominant distant plane and a ground plane. The single parametric model has been extended by local ones such as smoothly-varying affine model [LLM*11] and as-perspective-as-possible warps [ZCBS13]. The combination of a projective transform and a similarity transform is advocated to ameliorate structure distortion during warping [CSC14]. To handle parallax, Zhang and Liu [ZL14] aligned input images locally by finding an optimal seam. To alleviate the projective distortion, Lin *et al.* [LPRA15] combined a linearized local homography with a global similarity transform. However, most previous approaches do not take into account large portion of dynamic objects and therefore cannot be directly applied to stitching selfie images, as we show later.

Our work is also related to previous research on motion segmentation. Motion segmentation aims to break a scene into prominent moving groups [SM98]. In this respect, it is instrumental in differentiating moving layers caused by depth variation. Most recent approaches rely on feature-based trajectories or dense motion fields to handle non-rigid motion, exemplified by the 2D optical flow [BBPW04]. Brox and Malik [BM10] segmented long-term motion trajectories using spectral clustering by finding the low-dimensional embedding. Oschs *et al.* [OB12] improved the approach with high-order constraints. Sun *et al.* [SWS*13] leveraged the layer information and occlusion to segment foreground and background motion. Narayana *et al.* [NHLM13] advocated the use of optical flow orientation instead of flow vectors so that the mo-



(a) Input (b) Scatter plots (c) Global homography

(d) As-projective-as-possible (e) Motion clustering

(f) Face detection (g) Our method

Figure 3: *Difficulties of stitching selfies images using existing methods. (a) Input images. (b) Scatter plots for feature movement, where background and foreground points are shown in blue cross and red circle, respectively. (c) Stitched result by applying a global homography with RANSAC [BL07]. (d) Stitched result by the as-projective-as-possible warp [ZCBS13]. (e) Cluster the feature motion to reject outliers before computing a global homography. (f) Compute the global homography with features that are outside the detected face region. (g) Our result.*

tion difference caused by depth variation can be canceled out. One problem of using dense motion for segmentation is its underlying assumption of a stationary camera or simple translational camera motion. Camera rotation, common in selfie video capturing, may lead to over-segmented background, which makes motion segmentation not suitable in our case with hand-held camera. Furthermore, the combination of parametric and dense motion segmentation is still an open problem.

3. Problem Analysis

We show in this section how the selfie images pose unique challenges that can fail most previous image stitching and motion segmentation approaches, which is illustrated in Fig. 3.

Traditionally, the alignment between these two images is done by computing a global homography with RANSAC [BL07]. However, the face region usually occupies a large part of the selfie image, and most of the matched feature points locate on the salient foreground. As shown in Fig. 3(b), the background points are sparse and distributed far from the principal data cluster. Therefore, RANSAC regards some sparse background points as outliers. As a result, misalignment occurs and leads to ghost artifacts when blending the overlapping regions, as shown in Fig. 3(c). Other sophisticated local warping models, such as the as-projective-as-possible approach [ZCBS13] shown in Fig. 3(d), are powerless to stitch such a scene due to the large occlusion and parallax.

Given that directly computing the correct motion model is hard using previous alignment methods, one may want to apply motion/bilayer segmentation approaches to extract the foreground first. A straightforward approach is to cluster the optical flow into two categories, and extract the one with a large variation as the background. However, this would remove a lot of feature points near the foreground boundary since the floor near the person has similar flow vectors to the foreground (optical flow calculation under large occlusion is erroneous in the flat region), leaving not enough features to compute the homography robustly, as shown in Fig. 3(e). Even if we have enough feature points for alignment, the final stitched image still contains artifacts given that the foreground regions are not correctly identified. Alternatively, if we apply face detection to remove feature points inside the face, the result is still erroneous as shown in Fig. 3(f). There are still some foreground points that cannot be rejected by RANSAC.

In conclusion, to stitch selfie images into a high quality panorama, we need (1) accurate foreground and background feature point classification for image alignment; and (2) accurate dense foreground mask to remove visual artifacts in the final composite. Previous image alignment and motion segmentation techniques can not achieve high quality results reliably on selfie images. In our work, by formulating these goals in a spatial-temporal optimization framework, and taking into account some unique characteristics of selfie images, we are able to achieve the above two goals.

4. Our Approach

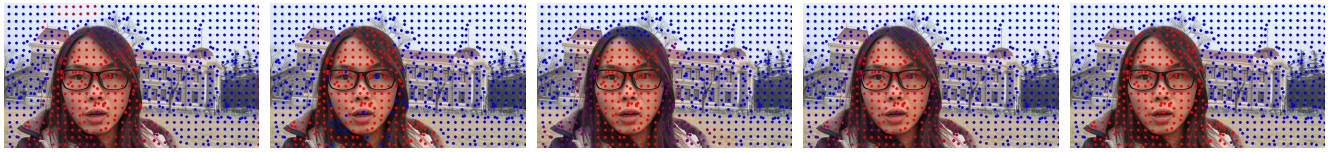
In this section, we first show how to extract the sparse feature points in selfies images and segment them into two classes: foreground (1) and background (0). The labeled feature points are used for image alignment and segmenting a dense foreground mask. Then we show how to stitch the user-specified faces with the background pieces as a final panoramic picture without distorting the target face region.

4.1. Sparse Matching and Segmentation

4.1.1. Feature Detection and Matching

Similar to previous panorama stitching systems, we first extract sparse feature points as the basis for image alignment. Traditional feature detectors, such as SIFT [Low99] and SURF [BTVG06], are designed to be robust against image transformations for matching and tracking. However, in the case of selfie images, they tend to lie right on the occlusion boundary given the large color and depth discontinuity along it, thus are not reliable for both image alignment and foreground segmentation.

In our application, we expect the sparse feature points to be evenly distributed, and do not snap to strong edges to avoid ambiguity and facilitate the sparse-to-dense propagation step later on. To achieve this, we define a uniform 2D grid mesh to partition each frame into blocks (block size is 18×18 in our system). The center point of each block on the first frame is selected as a feature point for initialization. If the center point lies on a strong edge, *i.e.*,



(a) Motion prior (b) Trajectory prior (c) Face prior (d) Labeling initialization (e) Labeling optimization

Figure 4: *Sparse feature segmentation results by (a) only the motion prior in Equ. (1); (b) only the trajectory prior in Equ. (2); (c) only the face prior; (d) the labeling initialization in Equ. (3); (e) the labeling optimization in Equ. (6). The bluer (redder) the color is, the more possible it is a background (foreground) feature.*

its gradient magnitude is larger than 0.25 (in the range of $[0,1]$), we randomly sample another one that is not on an edge in that block. The selected sparse points are tracked along the whole sequence by computing a forward dense optical flow [BBPW04]. To reduce tracking error caused by occlusions, firstly, a backward optical flow from the next frame to the current frame is computed to check whether a feature point moves back to its original position after applying both the forward and the backward flow. Secondly, if the local appearance in a 5×5 patch of the next tracked point significantly differs from that of the current point (*i.e.*, MSE difference is larger than 0.01 in the range of $[0,1]$), we also treat it as a lost feature. To detect features in newly emerged regions, we randomly re-sample a feature point from a block that has no feature points. In this way, the number of feature points in each block is guaranteed to be at least one and the feature distribution is roughly uniform over the whole image sequence.

We define some basic notations here. Let x_j^i denote the jth feature point on frame I^i. A feature trajectory $t_k^{pq}=\{x_k^p,\dots,x_k^q\}$ is defined as a feature path from frame I^p to frame I^q. A feature match $m_k^{ij}=(x_k^i,x_k^j)$, where $x_k^i,x_k^j\in t_k^{pq}$, is defined as the pair of any two feature points on the same trajectory. For all the feature matches between frame I^i and I^j, we apply the RANSAC algorithm to fit a global homography with a large fitting threshold (5% of image height). A frame match $P^{ij}=(I^i,I^j)$ is defined as a pair of matched frames whose number of matching inliers is larger than a threshold (2% of #features). Original feature matches that do not contribute to any pair of matched frames are considered unreliable and discarded. This give us robust feature matches that only exist in nearby frames.

4.1.2. Labeling Initialization

Given detected features, we aim to label each feature as either foreground or background. Only features labeled as background will participate in image alignment. The characteristics of selfie images yield some strong priors that can help us assign initial foreground probabilities to these features. The initial soft labeling results will be further refined in a graph optimization framework.

Prior 1: motion When capturing a selfie video, the user's body is much closer to the camera than the background and this distance does not change much. Furthermore, in our system since the user rotates the body rather than the camera, the foreground has relatively small movements while large motions present in the background region. Even under sudden camera shakes, the foreground motion is usually quite different from that of the background given their large depth difference. We therefore model the motion of feature points as a mixture of two Gaussian distributions, and the cluster closer to the origin is regarded as the foreground. Formally, based on motion, the foreground probability of a feature x_k^i is calculated as

$$a_1(x_k^i)=\frac{1}{N}\sum_{P^{ij}}\frac{\pi_1\psi(x_k^j-x_k^i;\theta_1^{ij})}{\sum_{c=0}^{1}\pi_c\psi(x_k^j-x_k^i;\theta_c^{ij})}, \tag{1}$$

where P^{ij} is a pair of frames (I^i,I^j) that the feature exists in, N is the total number of pairs, π_0 and π_1 are the estimated weight coefficients of the background and foreground component, respectively, and $\psi(x;\theta_c^{ij})$ is the probability density function of the corresponding Gaussian distribution with parameter mean and variance denoted as θ_c^{ij}.

Prior 2: trajectory In the captured selfie video, the user always faces the cameras while the background moves in and out fairly quickly. This results in much longer feature trajectories on the user's body, compared with trajectories that are on the background. We thus use a translated and mirrored Poisson distribution to model the length of foreground trajectories. Similarlly, we use another Poisson distribution to model the length of background trajectories. The EM algorithm is implemented to estimate the parameters of the mixture of two Poisson distributions. For a feature point $x_k^i\in t_k^{pq}$, its foreground probability based on length is computed as

$$a_2(x_k^i)=\frac{\omega_1\phi(|t_k^{pq}|;\lambda_1)}{\sum_{c=0}^{1}\omega_c\phi(|t_k^{pq}|;\lambda_c)}, \tag{2}$$

where $\phi(x;\lambda_0)=(\lambda_0)^x\exp(-\lambda_0)/x!$ and $\phi(x;\lambda_1)=(\lambda_1)^{-x+d}\exp(-\lambda_1)/(-x+d)!$ are the probability density functions of the background and foreground Poisson distribution, respectively, ω_c is the corresponding weight for each mixture, $|t_k^{pq}|$ denotes the trajectory length, and $d=$ #frames is the translation of the foreground Poisson distribution.

Prior 3: face Since the main subject in a selfie image is usually the frontal face, we run a face detector [HKS03] to localize it. As shown in Fig. 10(a), we also use a fixed body shape template to roughly identify the possible body region. Denote the combined initial face and body region as τ. We then erode τ to create a smaller interior region τ' whose size is 75% of τ, and label all feature points inside τ' with foreground probability 1. Similarly, we dilate τ to create a larger exterior region τ'' whose size is 125% of τ. We then

assign foreground probability 0 to points outside τ'', and probability 0.5 to those in $\tau'' - \tau'$.

Combining all above priors together, the initial foreground probability of a feature point is computed as

$$a(x_k^i) = \left(a_1(x_k^i) + a_2(x_k^i) + a_3(x_k^i)\right)/3. \quad (3)$$

The feature labeling results using the motion prior, trajectory prior, face prior, and all three priors are illustrated in Fig. 4(a), Fig. 4(b), Fig. 4(c), and Fig. 4(d). We can see that these three priors of selfie images all contributed to the labeling initialization.

4.1.3. Feature Graph Construction

The initial foreground probabilities of feature points tend to be noisy. To improve the spatial and temporal consistency of the labeling, we build a *non-local* graph using the feature points as nodes, and apply a graph optimization to refine the foreground probabilities. The principle for constructing such a graph is to try to connect nodes that are possibly within the same object region to encourage propagation, but disconnect nodes that are on different sides of occlusion boundaries to avoid cross-talking between the foreground and background.

Firstly, an edge exists between any pair of matched feature points according to the feature tracking results. Such a connection is reasonable since the matched feature points are highly likely to be on the same object. The arc cost for such an edge is defined as their color difference

$$\ell(x_k^i, x_k^j) = ||c(x_k^i) - c(x_k^j)||/\sigma_c, \quad (4)$$

where $c(x_k^i)$ is the color of point x_k^i (normalized to $[0,1]$), and $\sigma_c = 0.05$ is the standard deviation which quantifies the amount of variation.

Secondly, we connect a feature point to its spatial neighbors on each individual frame. For each point, we extract a feature vector consists of its spatial coordinates, color, and motion cluster label in Equ. (1): $\{x_k^i, c(x_k^i), a_1(x_k^i)\}$. A link is added between x_k^i and a nearby point x_q^i if $(a_1(x_k^i) - 0.5)(a_1(x_q^i) - 0.5) > 0$ and its cost $\ell(x_k^i, x_q^i) \leq 2$

$$\ell(x_k^i, x_m^i) = \rho||-x_k^i - x_m^i||/\sigma_x + (1-\rho)||c(x_k^i) - c(x_m^i)||/\sigma_c, \quad (5)$$

where the first and the second term are the spatial and color distance between these two points, respectively, and $\sigma_x = 0.05$ and $\sigma_c = 0.05$ control the standard deviations. We put motion clustering consistency as a strong constraint and only allow links within the same cluster. This is under the consideration that motion presents a strong cue for foreground and background separation in selfie images, as discussed in Sec. 4.1.2. Furthermore, inspired by recent image matting approaches that use global color samples [HRR*11], we set ρ to be a small value (0.1 in our system), so that far away points can still connect if they have similar colors. For feature trajectories that last for only one frame, there is no motion information, we thus just connect it with nearby points if the second condition in Equ. (5) is satisfied ($\rho = 0.5$ under this case).

4.1.4. Labeling Optimization

Given the initial foreground probabilities of feature points and the constructed feature graph, we apply a spatial-temporal optimization to compute the final labels of feature points. The optimization of final labeling probabilities α for all feature points is formulated as a minimization of the following energy function over the non-local graph

$$\min_{\alpha} \sum_i \sum_j w_j z_{ij} (\alpha_i - a_i)^2 + \eta \sum_i \sum_j z_{ij} (\alpha_i - \alpha_j)^2, \quad (6)$$

where i and j are indices over all the feature points. The first term enforces the concordance with the initialized labeling result a_i computed in Equ. (3). Since we only want to enforce this constraint for points with reliable initial estimation, we introduce a binary indicator w_j, which is set to 1 if $a_j \leq 0.2$ or $a_j \geq 0.8$, and 0 otherwise. Intuitively, $a_j \geq 0.8$ means all three priors in Equ. (3) assign high foreground probabilities to the feature point, thus it has a high confidence to be in foreground. Similarly, $a_j \leq 0.2$ means high confidence background point. $z_{ij} = \exp\{-\ell(n_i, n_j)\}$ is the affinity between node n_i and n_j according to the constructed feature graph, where $\ell(n_i, n_j)$ is the arc cost. The second term is a smoothness term weighted by η ($\eta = \sum_j w_j$/#total features in our system). By imposing the smoothness constraint on all pairs of features in the graph, above optimization model is capable to correct some prior mistakes, as shown in Fig. 4(e).

In essence, because of the efficient downsampling from the pixel space to the sparse feature space, our labeling algorithm makes global optimization over the whole sequence tractable, refraining from error propagation happens in the traditional frame-to-frame processing framework. Moreover, inspired by the idea of *all-pairs* in edit propagation [AP08] and as alluded to in the previous subsection, the similarity between any of the two feature points in the video is considered, enabling more robust and accurate labeling against color ambiguity and prior errors. As shown in Fig. 14, even if half of the features are wrongly initialized, the error rate of the optimized labels still remain at a low level.

4.2. Person-Background Composition

4.2.1. Dense Segmentation

With the sparse labels, we employ another spatial-temporal optimization to create a foreground mask on each frame. Different from the global edit propagation methods such as [AP08], the feature points are almost uniformly distributed on each frame, thus there is no need to propagate each known label as far as possible.

Firstly, thin plate interpolation [Fra82] is applied to produce a dense raw matte map b on each frame

$$b_p = \sum_j k(p, x_j)\alpha_j / \sum_j k(p, x_j), \quad (7)$$

where p is a pixel on the current frame, x_j is the jth feature point on this frame, $k(p, x_j) = \exp\{-||v(p) - v(x_j)||^2/(2\sigma_v^2) - ||c(p) - c(x_j)||^2/(2\sigma_c^2)\}$ is the affinity between pixel p and feature point x_j, $v(p)$ and $c(p)$ are the normalized optical flow and color values, respectively, standard deviation σ_v is set to 0.10, and $\sigma_c = 0.05$.

(a) All-pair (b) Sparse control (c) Our method

Figure 5: *Regarding the sparse feature points as known edits, we propagate their labels to all pixels via the (a) all-pair approach [AP08] and (b) sparse control model [XYJ13]. We can see that (c) our method achieves a much clearer foreground boundary.*

This raw matte map b serves as the input to produce the final dense mattes β through following formulation

$$\min_{\beta} \ (\beta - b)^T \delta(\beta - b) + \xi \beta^T L \beta. \tag{8}$$

The first term is a data term constrained by the raw matte map b and the second term is a Laplacian smooth regularizer guided by spatial and temporal neighbors.

The first term is a data term encouraging β to be close to b, where δ is a diagonal matrix with each diagonal element equals to a binary value δ_i. This is under the consideration that not all pixel values in b have high confidence to be used to constrain the final labeling. For example, since boundaries detected on the optical flow are not the exact foreground boundaries, pixels on salient edges of b generally have low confidences. We therefore define a binary confidence map δ with each element equals to

$$\delta_i = \ominus\{(b_i \geq 0.8 \mid b_i \leq 0.2) \ \& \ e_i = 0\}, \tag{9}$$

where e is the binary edge map of b, and $\ominus$ is the erosion operator. Namely, confident foreground/background pixels that do not snap to strong edges of raw mattes are valid for the data term.

The second term in Equ. (8) is the smoothness term, where L is the sparse Laplacian matrix for regularization, with $L_{ij} = -k_{ij}$ for $i \neq j$ and $j \in \Omega(i)$, and $L_{ii} = \sum_{j \in \Omega(i)} k_{ij}$. Here, $\Omega(i)$ is a set of neighboring pixels of i, including 4 spatial neighbors and 2 temporal neighbors. For the spatially four-connected neighboring pixels, if a neighboring pixel lies on the binary edges of its original image, it will be removed from $\Omega(i)$. This process is designed to retain the strong image edge on the alpha mattes. The temporal corresponding pixels on previous and next frame is obtained via backward and forward flow, respectively.

4.2.2. Final Composition

With the sparse labeling results for alignment and dense segmentation masks, the user then goes over the video and chooses a best frame with the desired head position and best facial expression. We call this image *foreground image*, denoted as I^f. Other frames are called *background images*, as they only contribute to the background region of the panorama.

As the first step of the stitching procedure, given that we have

(a) Original (b) Spherical (c) Person-preserving

Figure 6: *Person-preserving warping. To alleviate the shape deformation on the person, we apply a content-preserving warp [LGJA09] that warps the background spherically but keeps the foreground region in the original size.*

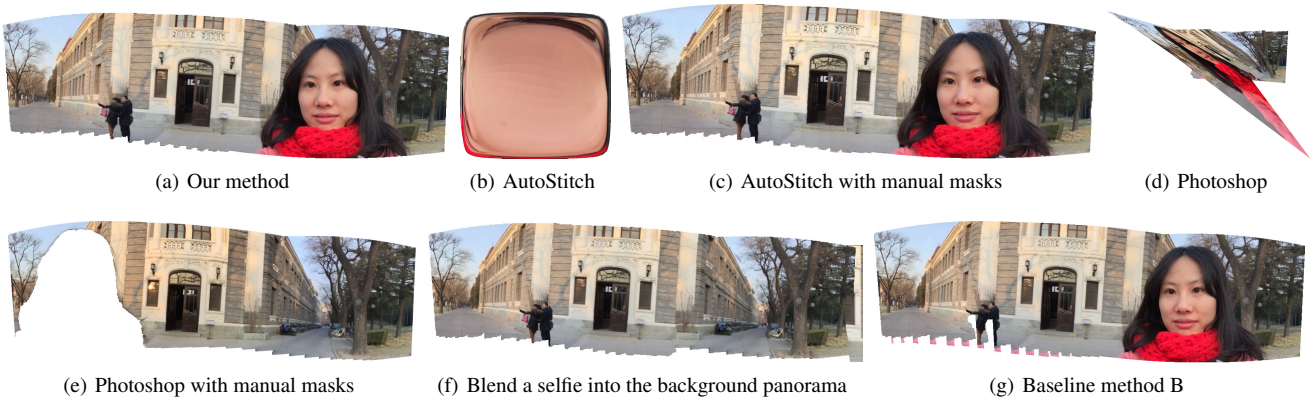
foreground masks on all background images, we remove all foreground pixels in them. All images are then aligned in a global coordinate system using the matched background feature points. To create a visually pleasing composite, all aligned background image are then projectively transformed onto a compositing surface (such as plane or spherical). However, if we apply spherical warping to I^f, the face region is often undesirably distorted, as shown in Fig. 6(b). To avoid such foreground distortion, we compute a content-preserving warp [LGJA09] on I^f, where the background region is warped according to the alignment warping, while the deformation on the foreground region is minimized, as shown in Fig. 6(c).

After all images are aligned, as in a standard panorama stitching procedure [BL07], we use GraphCut [KSE*03] to find seams in overlapping regions, and stitch images together to create a final composite. One way might be stitching a background panorama first and then pasting the selected foreground where desired using Poisson image editing method [PGB03]. However, this would lead to obvious blending artifacts near the foreground boundary since the segmentation result on I^f cannot be perfect. Therefore, with the overlapping between the foreground region in I^f and the background images due to parallax, in order to preserve the foreground with no blending artifacts, pixels in this overlapping region on all background images are also removed. With no overlapping between the foreground person and the background, GraphCut cannot produce a seam that passes through the foreground, thus seamless foreground preservation is guaranteed, even for a not perfect segmentation result on I^f.

5. Results and Discussion

In the experiments, the selfie videos are typically captured by an iPhone 6 frontal camera without a selfie stick. The photographer is roughly the rotation center. Each video with the original 1280×720 resolution at 30 fps is temporally donwnsampled to 6 fps before further processing. The field of view for each captured scene ranges from $90^\circ \sim 180^\circ$. The parameters in this paper are all fixed as stated before. The composited panoramic selfie results are shown in Fig. 1, Fig. 7(a), Fig. 8(a), Fig. 11(d), Fig. 13(a), Fig. 15, and Fig. 16. Due to the limited space, please refer to the supplementary materials for more detailed results.

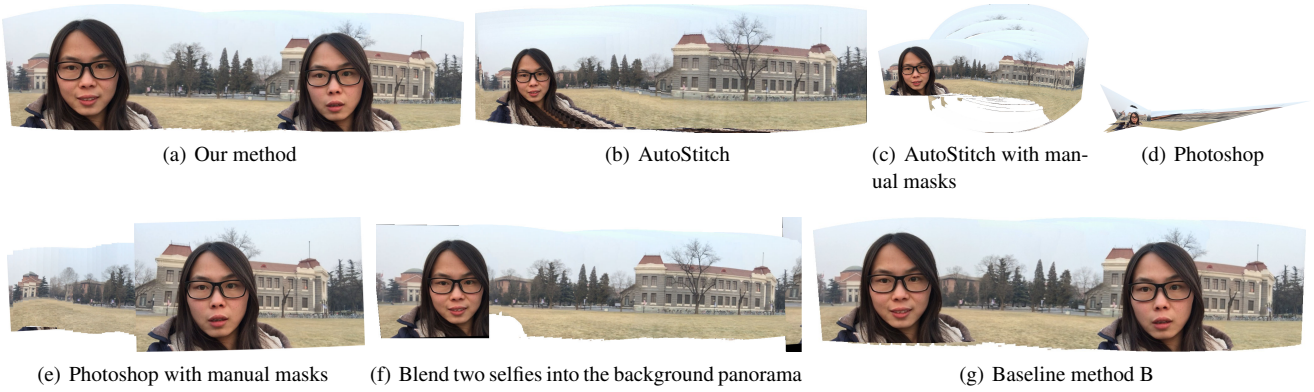
Comparison with image stitching We first compare our method with traditional image stitching approaches [Ado, Aut]. In Fig. 7, we show a result that most of the video frames are occupied by the



(a) Our method (b) AutoStitch (c) AutoStitch with manual masks (d) Photoshop

(e) Photoshop with manual masks (f) Blend a selfie into the background panorama (g) Baseline method B

Figure 7: *Composition results by different methods for a selfie video where the user occupies almost half of the image. (a) Panoramic selfie result with a user selected side face by our method. (b) Result by directly applying the AutoStitch [Aut]. (c) Result by manually masking out the unselected foreground regions before running AutoStitch. (d)-(e) Results generated by Photoshop Photomerge [Ado] similar to (b)-(c). (f) Generate a background panorama first, then blend the selected selfie image into the panorama. (g) Result by the baseline method B.*



(a) Our method (b) AutoStitch (c) AutoStitch with manual masks (d) Photoshop

(e) Photoshop with manual masks (f) Blend two selfies into the background panorama (g) Baseline method B

Figure 8: *Composition results by different methods for a selfie video where the background is textureless. (a) Result by our method (two selfie images are selected). (b) and (d): results by directly applying AutoStitch and Photoshop, respectively. (c) and (e): results by removing the unwanted foreground regions first before applying the stitching software. (f) Result by blending two chosen selfie images into the background panorama. (g) Result by the baseline method B.*

foreground, and compare our result with those produced by Adobe Photoshop Photomerge [Ado] and AutoStitch [Aut]. We can see that previous panorama methods produce entirely wrong results due to incorrect alignment. Furthermore, to demonstrate the effective of our system, we manually remove all the foreground regions except the two chosen faces. Photoshop still cannot stitch some images that have less background matches. Both approaches fail to preserve the target foreground regions. This is because that a seam is found through the target foreground region in the blending process. Similar conclusions can be made in Fig. 8. Fig. 8(c) and Fig. 8(e) show that even if the foreground person is masked out, the textureless background regions do not have enough features for reliable alignment. In Fig. 8(c), AutoStitch is only successful to preserve the last selfie image in the sequence instead of the user selected one. Note that no blending effect is applied in these experiments.

Comparison with blending the selected selfie(s) into the background panorama One possible solution for panoramic selfie composition is to capture a background panorama first, and then capture a selfie image. By using the feature matching for alignment and seam finding algorithm [KSE*03] for composition, these two images could be blended together. However, Fig. 7(f), Fig. 8(f), and Fig. 9(c) show that the this process cannot guarantee the selfie's proper embedding, especially when the background is textureless or regular. For instance, as shown in Fig. 9, the regular pattern on the background wall leads to a bad registration result. Furthermore, even if they are aligned well, the face region in the selfie image might be partially (Fig. 8(f)) or completely (Fig. 7(f)) removed when finding a seam in the overlapping area, unless a foreground mask is known beforehand. Finally, such an approach does not allow changing the position of the face region after capturing, as our system does.

Comparison with two baseline methods Two baseline method

(a) Background panorama (b) Selfie (c) Blended result

Figure 9: *Blend a selected selfie image into the background panorama. Bad registration between (a) the background panorama and (b) the selfie image due to the regular pattern on the background wall leads to (c) an improper embedding and blending.*

(a) Regions (b) Composition Result

Figure 10: *Baseline method A. (a) Detected face, body, and dilated foreground region (marked in red, blue, and green, respectively). (b) Composition result with the green region as foreground mask to our composition process in Sec. 4.2.2.*

for foreground segmentation is also compared with our approach. As stated in Sec. 4.1.2 and shown in Fig. 10(a), it is not difficult to localize a dilated foreground region via face detection. The extracted foreground masks are fed into our final composition process in Sec. 4.2.2 to produce the panoramic result, which is called the baseline method A. Furthermore, with the dilated upper body region as a prior, the foreground region could be improved by the GrabCut approach [RKB04] before final composition, which we call the baseline method B. However, such two baseline methods fail to segment the foreground correctly, resulting in artifacts in the final composition result, such as the extra foreground and missing background in Fig. 7(g) and Fig. 10(b), the floating person in Fig. 8(g), and the twisted boundary in Fig. 11(e).

Comparison with motion segmentation We compare the foreground masks and final composite image generated by our method with those by other motion segmentation approaches [ZJHB11, FZS12]. As shown in Fig. 11(c) and Fig. 11(f), the geometric constraints and the motion consistency used in the bilayer segmentation approach [ZJHB11] do not hold here. The foreground person is either partially extracted or over-segmented. Due to the rich texture in the background, the alignment between images are still correct. Similar conclusions can be drawn from the comparison example with the discontinuities tracing approach [FZS12] as shown in Fig. 13.

Comparison with edit propagation Our dense segmentation problem in Sec. 4.2.1 is in spirit similar to the edit propagation problem in video editing. As shown in Fig. 5, our method achieves better propagation results compared with the global all-pair approach [AP08], and sparse control model [XYJ13]. Our scheme introduces the raw dense matte map to take advantage of both motion and color information simultaneously. Furthermore, combined with the delicate consideration of edge information, our non-local feature graph effectively reduces the cross-talking between pixels on different layers.

Table 1: *Mean labeling error if disabling some techniques.*

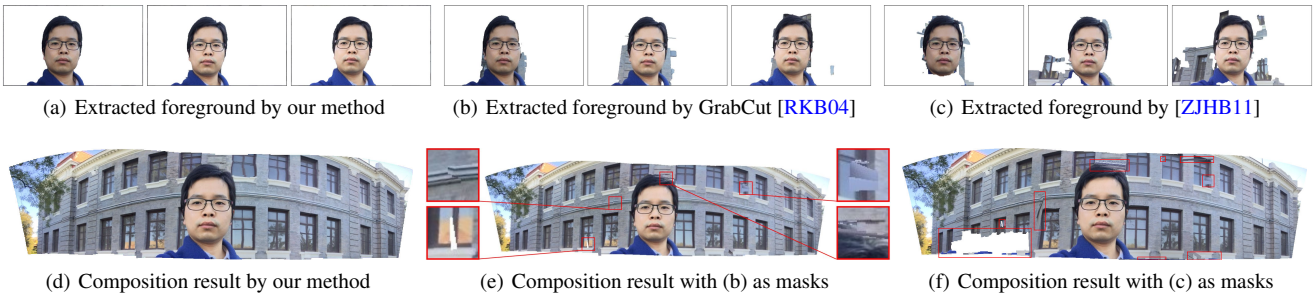
Disabled prior(s)	MSE
motion	0.1081
trajectory	0.0162
face	0.0142
motion+trajectory	0.1011
motion+face	0.0163
trajectory+face	0.0867

Robustness to initialized label errors We performed some experiments to verify the robustness of the labeling optimization framework. A comparative example for disabling some initialization priors used in Sec. 4.1.2 and Sec. 4.1.3 is shown in Tab. 1. We can see that the motion prior contributes more than the other two priors in the labeling process. However, as shown in Fig. 3(e), Fig. 11(f), and Fig. 13(b), we can not only rely on the motion prior, especially when the background is largely occluded by the person or lacks textures. Furthermore, as shown in Fig. 14, the labeling accuracy still remains at a high level when a large portion of labeling error ($\geq 50\%$) is randomly introduced into the initialization.

Importance of accurate dense segmentation We performed some experiments to elaborate that the dense segmentation is important for generating a good composition result. The groundtruth foreground mask is dilated or eroded by 5%, 10%, and 20% gradually, and fed into the final composition process to produce the panoramic result. As shown in Fig. 12, on one hand, even a slightly dilated foreground region could lead to noticeable blending artifacts near the foreground boundary. On the other hand, a slightly eroded foreground region would result in obvious foreground residuals in the final composite. The more inaccurate the segmentation is, the more artifacts the composition result will have.

Running time It takes 215.92s for our system to stitch 21 images shown in Fig. 7 on a desktop PC with 3.6GHz CPU and 8GB RAM without any code optimization, while AutoStitch and Photoshop consume 34.17s and 38.21s, respectively. The running time for each component is: 171.97s for feature detection and matching, 3.38s for labeling initialization, 2.17s for labeling optimization, 20.66s for dense segmentation, and 17.75s for final composition. Due to the time-consuming optical flow calculation, the feature detection process takes 79.64% of the total running time. Apart from the feature detection process, the rest of our system is of comparable time complexity compared with matured commercial products.

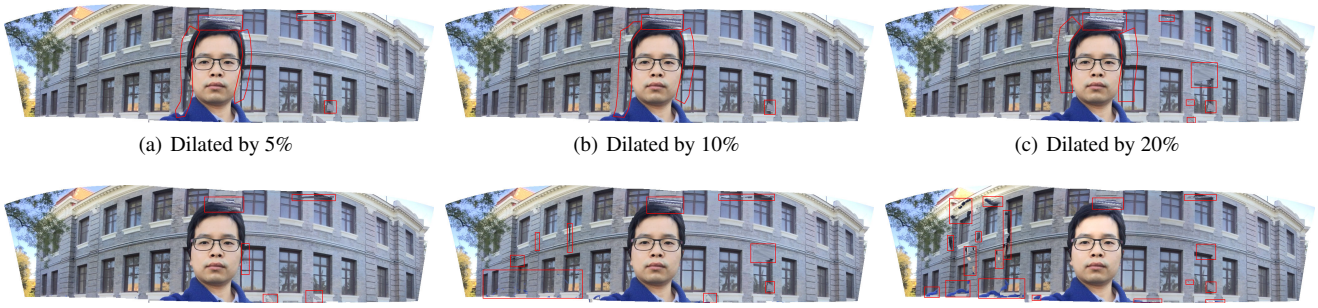
Limitations Our proposed system has some limitations. Firstly, it has difficulty in dealing with large dynamic background, such as taking selfies in a moving crowd. Fig. 16 gives one failure example. The moving people are at different depth layers and actually dominant the background, violating our bilayer, stationary background assumption. We note that it is indeed a challenging case that may fail all existing stitching approaches. Secondly, only a single picture taker is considered in our system. Generating panoramic selfies with multiple faces is more chanllenging and is our future work. Finally, background with large parallax is also problematic.



(a) Extracted foreground by our method (b) Extracted foreground by GrabCut [RKB04] (c) Extracted foreground by [ZJHB11]

(d) Composition result by our method (e) Composition result with (b) as masks (f) Composition result with (c) as masks

Figure 11: *Compared with our method, both the baseline method B and the bilayer segmentation approach [ZJHB11] fail to segment the foreground correctly, resulting in artifacts in the final composite (marked in red). (a)-(c): extracted foreground by our method, baseline method B (GrabCut [RKB04]), and bilayer segmentation approach [ZJHB11]. (d)-(f): Composition result with corresponding foreground as masks to our stitching process in Sec. 4.2.2. Note the twisted boundary near the hair, missing window, and discontinuous line on the wall.*



(a) Dilated by 5% (b) Dilated by 10% (c) Dilated by 20%

(d) Eroded by 5% (e) Eroded by 10% (f) Eroded by 20%

Figure 12: *The groundtruth foreground mask is dilated or eroded by 5%, 10%, and 20% gradually, and fed into the final composition process to produce the panoramic result. The more dilation, the more blending artifacts near the foreground boundary. The more erosion, the more foreground residuals in the final composite. Please zoom in to have a clear vision on the artifacts regions marked in red.*

(a) Our method (b) Motion segmentation [FZS12]

Figure 13: *Comparison with the motion segmentation approach [FZS12] when the background lacks textures. Over-segmented foreground regions produce missing part in the final result.*

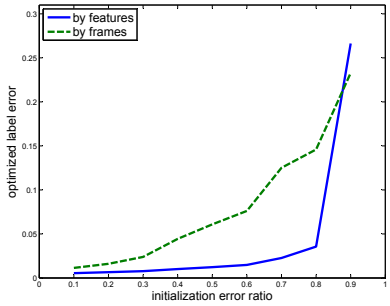


Figure 14: *Tolerance of the labeling optimization against initialization errors. x% of the labeling error is randomly introduced to the initialization to see how our optimization method reacts. Two curves are results by two different initialization method (x% of the total features, and x% of the total frames).*

6. Conclusion

We present a fully automatic approach that can generate a panoramic selfie from a selfie video captured by rotating the user's body. Our main technical contribution is a robust multi-frame segmentation and stitching framework tailored to this problem, including a spatial-temporal optimization method that can generate accurate image alignment results in the presence of large foreground occlusion; and a dense foreground segmentation approach that uses graph optimization to handle parallax in the stitching process. We also show how to stitch segmented image regions together to create a panorama without distorting the face region. Results show that our method can generate high quality panoramas that previous automatic systems cannot produce.

Figure 15: *Composition result for a vertically shot selfie sequence.*

Figure 16: *A failure case with dynamic background.*

7. Acknowledgements

This work was supported by the Project of NSFC (No. 61327902, 61522111, and 61601278), the National Key Scientific Instrument and Equipment Development Project (No. 2013YQ140517), the Program of Shanghai Academic Research Leader (No. 16XD1401200), and the Young Teacher Training Program of Shanghai Municipal Education Commission (No. ZZSD15117).

References

[Ado] ADOBE: Adobe photoshop cc. http://www.adobe.com/products/photoshop.html. 2, 6, 7

[AP08] AN X., PELLACINI F.: Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph. 27*, 3 (2008), 40. 5, 6, 8

[Aut] AUTOSTITCH: Autostitch. http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html. 2, 6, 7

[BBPW04] BROX T., BRUHN A., PAPENBERG N., WEICKERT J.: High accuracy optical flow estimation based on a theory for warping. In *ECCV* (2004), pp. 25–36. 2, 4

[BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *International Journal on Computer Vision 74*, 1 (2007), 59–73. 2, 3, 6

[BM10] BROX T., MALIK J.: Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010, pp. 282–295. 2

[BTVG06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *ECCV* (2006), pp. 404–417. 3

[CSC14] CHANG C.-H., SATO Y., CHUANG Y.-Y.: Shape-preserving half-projective warps for image stitching. In *CVPR* (2014), pp. 3254–3261. 2

[Fra82] FRANKE R.: Scattered data interpolation: Tests of some methods. *Math. Comp. 38* (1982), 181–200. 5

[FZS12] FRAGKIADAKI K., ZHANG G., SHI J.: Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR* (2012), pp. 1846–1853. 8, 9

[GKB11] GAO J., KIM S. J., BROWN M. S.: Constructing image panoramas using dual-homography warping. In *CVPR* (2011), pp. 49–56. 2

[HKS03] HANNES KRUPPA M. C.-S., SCHIELE B.: Fast and robust face finding via local context. In *Joint IEEE Int'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (2003), pp. 157–164. 4

[HRR*11] HE K., RHEMANN C., ROTHER C., TANG X., SUN J.: A global sampling method for alpha matting. In *CVPR* (2011), pp. 2049–2056. 5

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (2003), 277–286. 6, 7

[LGJA09] LIU F., GLEICHER M., JIN H., AGARWALA A.: Content-preserving warps for 3d video stabilization. *ACM Trans. Graph. 28*, 3 (2009), 44:1–44:9. 6

[LLM*11] LIN W.-Y., LIU S., MATSUSHITA Y., NG T.-T., CHEONG L.-F.: Smoothly varying affine stitching. In *CVPR* (2011), pp. 345–352. 2

[Low99] LOWE D. G.: Object recognition from local scale-invariant features. In *CVPR* (1999), pp. 1150–1157. 3

[LPRA15] LIN C. C., PANKANTI S. U., RAMAMURTHY K. N., ARAVKIN A. Y.: Adaptive as-natural-as-possible image stitching. In *CVPR* (2015), pp. 1155–1163. 2

[Mob] MOBIMAGING: Panoselfie: panorama selfie & wide angle group photo for free by front facing camera. https://itunes.apple.com/us/app/panoselfie-panorama-selfie/id852347776?mt=8. 2

[NHLM13] NARAYANA M., HANSON A., LEARNED-MILLER E.: Coherent motion segmentation in moving camera videos using optical flow orientations. In *ICCV* (2013), pp. 1577–1584. 2

[OB12] OCHS P., BROX T.: Higher order motion models and spectral clustering. In *CVPR* (2012), pp. 614–621. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Trans. Graph. 22*, 3 (2003), 313–318. 6

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (2004), 309–314. 8, 9

[SCF*14] SHAN Q., CURLESS B., FURUKAWA Y., HERNANDEZ C., SEITZ S. M.: Photo uncrop. In *ECCV* (2014), pp. 16–31. 2

[SM98] SHI J., MALIK J.: Motion segmentation and tracking using normalized cuts. In *ICCV* (1998), pp. 1154–1160. 2

[STP12] SUMMA B., TIERNY J., PASCUCCI V.: Panorama weaving: Fast and flexible seam processing. *ACM Trans. Graph. 31*, 4 (2012), 83:1–83:11. 2

[SWS*13] SUN D., WULFF J., SUDDERTH E. B., PFISTER H., BLACK M. J.: A fully-connected layered model of foreground and background flow. In *CVPR* (2013), pp. 2451–2458. 2

[Sze06] SZELISKI R.: Image alignment and stitching: A tutorial. *Found. Trends Comput. Graph. Vis. 2*, 1 (2006), 1–104. 2

[XYJ13] XU L., YAN Q., JIA J.: A sparse control model for image and video editing. *ACM Trans. Graph. 32*, 6 (2013), 197. 6, 8

[ZCBS13] ZARAGOZA J., CHIN T.-J., BROWN M. S., SUTER D.: As-projective-as-possible image stitching with moving dlt. In *CVPR* (2013), pp. 2339–2346. 2, 3

[ZJHB11] ZHANG G., JIA J., HUA W., BAO H.: Robust bilayer segmentation and motion/depth estimation with a handheld camera. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 3 (2011), 603–617. 8, 9

[ZL14] ZHANG F., LIU F.: Parallax-tolerant image stitching. In *CVPR* (2014), pp. 4321–4328. 2