Direct Photometric Alignment by Mesh Deformation

Kaimo Lin[1,2]   Nianjuan Jiang[2,4]   Shuaicheng Liu[3]   Loong-Fah Cheong[1]   Minh Do[2]   Jiangbo Lu[2,4]

[1]National University of Singapore   [2]Advanced Digital Sciences Center, Singapore
[3]University of Electronic Science and Technology of China
[4]Shenzhen Cloudream Technology, China

Abstract

The choice of motion models is vital in applications like image/video stitching and video stabilization. Conventional methods explored different approaches ranging from simple global parametric models to complex per-pixel optical flow. Mesh-based warping methods achieve a good balance between computational complexity and model flexibility. However, they typically require high quality feature correspondences and suffer from mismatches and low-textured image content. In this paper, we propose a mesh-based photometric alignment method that minimizes pixel intensity difference instead of Euclidean distance of known feature correspondences. The proposed method combines the superior performance of dense photometric alignment with the efficiency of mesh-based image warping. It achieves better global alignment quality than the feature-based counterpart in textured images, and more importantly, it is also robust to low-textured image content. Abundant experiments show that our method can handle a variety of images and videos, and outperforms representative state-of-the-art methods in both image stitching and video stabilization tasks.

1. Introduction

A variety of motion models have been employed in applications like image stitching [6, 30, 31, 4, 13, 14], video stitching [10, 23, 15, 11], and video stabilization [17, 19]. Global parametric models can be estimated robustly and efficiently given its simplicity and were popular among the early works (*e.g.*, [2, 21]). However, they only work well when the scene is a plane or the camera motion is a rotation. For images with parallax, a global parametric model is usually used to estimate an initial alignment for other more sophisticated methods [17, 30, 14]. Apart from the global parametric model, we can categorize most of the existing motion models into three types depending on the model complexity (Fig. 1), *i.e.*, mesh-based image warping [17, 12, 15], spatially-varying parametric motion field

[16, 30], and optical flow [23, 20].

Mesh-based image warping allows spatially varying motion models and only local rigidity is imposed. This type of methods require high quality feature matches and aim to minimize the geometric alignment error of matched features. Local rigidity is imposed by constraining the mesh cells to undergo similarity transformation [17, 31, 12] or affine transformation [32]. These methods have been proven to be sufficiently flexible for handling complex scene geometry and camera motion in most image stitching tasks (*e.g.*, [17, 19, 12]). However, their performance is highly dependent on the quality and distribution of the feature correspondences and can easily suffer from low-textured content. Spatially-varying parametric motion field models like APAP [30] and 'spatially varying affine' [16] can produce good stitching results even with non-ideal feature matches by interpolating a 2D transformation for each image pixel. Nevertheless, these methods still require a handful of quality feature matches to begin with, not to mention that they are computationally more expensive. Optical flow, instead, directly estimates 2D pixel motion with the minimum rigidity assumption. It is often used in video applications because of the good alignment quality and density it provides on both low- and rich-textured scenes. However, optical flow estimation is in general computationally expensive and an over-kill for 'synthesis quality'-driven applications, where estimating a physically accurate motion at every pixel is not necessary.

In this paper, we propose a **M**esh-based **P**hotometric **A**lignment (MPA) method that stems from the concept of optical flow but is formulated as mesh deformation. We replace geometric errors of known correspondences with photometric errors on sampled points in the current alignment, and minimize this error for near pixel-level alignment. The displacement of each sampled point is parameterized by the four nearest mesh vertices, and hence the size of the optimization problem is independent of the number of sampled points. Therefore, the proposed method not only takes advantage of the density and reliability of the variational opti-
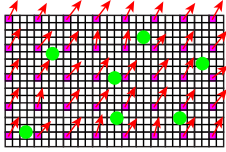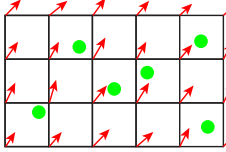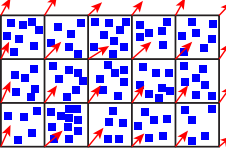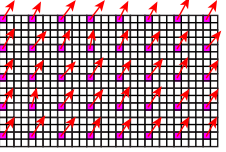
	Feature-driven alignment		Direct alignment	
	Parametric Motion Field	Mesh Warping	Our MPA	Optical Flow
Methods	■ Pixels ● Sparse matches	● Sparse matches	■ Sampled locations	■ Pixels
Variables	per-pixel 3×3 matrix $\mathbf{T}$	mesh vertices $\mathbf{V}_i$	mesh vertices $\mathbf{V}_i$	pixel motion (u, v)
Metric	appearance, geometric [30, 16]	geometric	photometric	photometric
Regularization	global smoothness	mesh grid rigidity [17]	mesh grid rigidity	piecewise smoothness
Measurements	sparse	sparse	semi-dense	dense
Speed	slow	**fast**	**fast**	slow

Figure 1: Comparison with typical motion estimation methods for image stitching and video stabilization.

cal flow for photometric alignment, but also enjoys the efficiency of mesh-based image warping for online processing. Moreover, it can readily incorporate more complex regularization priors such as a general content preserving term [17] for spatial smoothness or a curve structure preserving term [14] for minimizing unnatural scene distortion. These constraints are non-trivial for general optical flow algorithms to incorporate. In the experiments, our method consistently produces superior alignment quality on a wide variety of image and video contents and outperforms representative state-of-the-art methods in both image stitching and video stabilization tasks.

2. Related Works

Global parametric models Homography is the most widely used global parametric model in many applications. Early methods estimate a single homography model from sparse feature matches to align two images for both image stitching [25, 8, 2] and video stabilization [21]. However, the homography model only works under assumptions of pure camera rotation or planar scene. Any violation of these assumptions will introduce artifacts like ghosting in stitching and jitter in stabilization.

Mesh-based image warping For images or videos with parallax, mesh-based image warping is a popular approach. Gao *et al.* [6] employed a dual-homography model for stitching by assuming that the scene contains two dominant planes. Lin *et al.* [13] proposed a hybrid warping model that fuses two stitching fields to generate natural-looking panoramas. For low-textured images, Li *et al.* [12] proposed a dual-feature (keypoints and line segments) warping model to guide the alignment in low-textured regions. To preserve salient structures (*e.g.*, lines and curves) during the warping, Zhang *et al.* [32] and Li *et al.* [12] incorporated different line-preserving constraints into the mesh deformation process. Lin *et al.* [14] proposed a curve-preserving

term in their seam-guided local alignment method to preserve curve structures. Liu *et al.* [17] developed content-preserving warps (CPW) to warp original video frames according to a smoothed camera path obtained from sparse 3D reconstruction for video stabilization. To achieve real-time efficiency, Liu *et al.* [18] introduced MeshFlow, a non-parametric warping method for video stabilization. Recently, works [10, 15, 11] proposed warping methods specially designed by taking spatial and temporal smoothness into consideration for video stitching.

Spatially-varying parametric motion field Lin *et al.* [16] proposed a smoothly varying affine field for image stitching. The per-pixel parametric model is estimated together with feature correspondences. It allows images taken from significantly different view points and lighting conditions to be aligned globally at the cost of high computational complexity. Zaragoza *et al.* [30] introduced a more general and efficient spatially-varying projective motion model to locally align correspondences, while preserving global projective transformation.

Optical flow Optical flow is also explored in some video applications. In term of global alignment quality, optical flow usually produces better results than mesh-based warping methods. However, as a general motion estimation technique, the obtained flow field often needs post-processing for specific applications (*e.g.*, outlier filtering, occlusion detection). Perazzi *et al.* [23] used optical flow to generate panoramic videos from unstructured camera arrays. Liu *et al.* [20] proposed the concept of pixel profiles from optical flow to analyze dynamic motions and stabilize video frames. However, the computational complexity of an optical flow method limits its practical use in processing visual content in general. Our mesh-based photometric alignment, on the other hand, achieves visual alignment quality that is almost on par with optical flow based methods with low algorithm complexity.
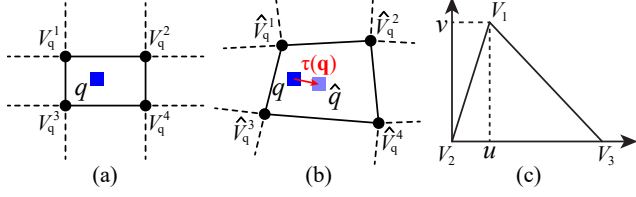
Figure 2: Photometric alignment by mesh deformation.

3. Mesh-based Photometric Alignment

Given two images (a reference image I_{ref} and a target image I_{tar}) capturing the same scene, our goal is to estimate the motion between them parameterized by mesh deformation. We assume the two images are roughly aligned (*e.g.*, consecutive frames from a video or one of the images is warped by a global homography). The proposed scheme achieves the alignment by warping I_{tar} to I_{ref} so as to minimize the photometric difference in the overlapping region.

3.1. Photometric Error

We assume the corresponding points in the two images obey the brightness constancy constraint. Given an initial alignment of the two images, we uniformly sample locations at regular intervals (three pixels) both horizontally and vertically in the overlapping region. For each sampled point location $\mathbf{q}$ in I_{tar}, we seek a 2D offset $\tau(\mathbf{q})$ that minimizes the following photometric error:

$$\|I_{tar}(\mathbf{q} + \tau(\mathbf{q})) - I_{ref}(\mathbf{q})\|^2, \qquad (1)$$

where $I_{ref}(\mathbf{q})$ is the intensity of I_{ref} at $\mathbf{q}$, and $I_{tar}(\mathbf{q} + \tau(\mathbf{q}))$ is the intensity of I_{tar} at $\mathbf{q} + \tau(\mathbf{q})$. Since the two images are roughly aligned and we can assume that $\tau(\mathbf{q})$ is small and set $\tau(\mathbf{q}) = (0,0)$ as its initial value. Using the first-order Taylor expansion of $I_{tar}(\mathbf{q} + \tau(\mathbf{q}))$, the photometric error can then be expressed as

$$E_c(\tau(\mathbf{q})) = \|I_{tar}(\mathbf{q}) + \bigtriangledown I_{tar}(\mathbf{q})\tau(\mathbf{q}) - I_{ref}(\mathbf{q})\|^2, \quad (2)$$

where $\bigtriangledown I_{tar}(\mathbf{q})$ is approximated by the intensity gradient at $\mathbf{q}$. Clearly, if we were to minimize $E_c(\tau(\mathbf{q}))$ alone, it would be the same as computing optical flow without spatial regularization.

3.2. Alignment by Mesh Deformation

To implicitly enforce spatial smoothness, we cast the photometric error minimization problem as a mesh deformation process. Specifically, we re-parameterize the offset on each sampled point using the coordinates of the four surrounding mesh vertices. Similar strategies were proposed for re-parametrizing residual flow using a parametric model within image patches [3]. For applications like image stitching or video stabilization, it usually does not require per-

pixel accuracy of the motion model. Hence, we use a simplified model and represent the offset $\tau(\mathbf{q})$ on each sampled point $\mathbf{q}$ as a 2D bilinear interpolation of the four mesh vertices $\hat{V}_q^k$ enclosing it (Fig. 2 (a) and (b)), *i.e.*,

$$\tau(\mathbf{q}) = \hat{\mathbf{q}} - \mathbf{q}, \quad \hat{\mathbf{q}} = \sum_{k=1}^{4} c_k \hat{V}_{\mathbf{q}}^k, \qquad (3)$$

where c_k are fixed coefficients computed by expressing $\mathbf{q}$ as a bilinear interpolation of the initial mesh vertices $V_{\mathbf{q}}^k$, and $\hat{V}_{\mathbf{q}}^k$ are the unknown new vertex locations to be optimized.

We use a grid mesh to represent I_{tar} and define the objective function as follows:

$$E(\hat{V}) = E_p(\hat{V}) + \lambda_1 E_s(\hat{V}) + \lambda_2 E_l(\hat{V}), \qquad (4)$$

where $\hat{V}$ are the unknown mesh vertices' coordinates. We introduce three terms in our objective function with $E_p(\hat{V})$ being the photometric term, $E_s(\hat{V})$ being the similarity transformation term and $E_l(\hat{V})$ being the line-preserving term. The associated weights of the last two terms are denoted by λ_1 and λ_2 respectively ($\lambda_1 = 0.2 \sim 0.5$ and $\lambda_2 = 1.0$ in our implementation).

Photometric term The photometric term is computed by summing up $E_c(\tau(\mathbf{q}))$ over all sampled points, except the ones with very small gradient values (less than 0.02), *i.e.*,

$$E_p(\hat{V}) = \sum_{\mathbf{q}} E_c(\tau(\mathbf{q})). \qquad (5)$$

We exclude those sampled points with too small gradient values because they do not contribute much useful information to the alignment process. Even so, the number of the remaining sampled points is significantly larger than the number of sparse feature matches that can be detected, which provide much more guidance for alignment.

Similarity transformation term To constrain image regions with insufficient or no sampled points and maintain spatial smoothness of the warping, we adopt the similarity transformation constraint in CPW [17]. The similarity transformation term measures the deviation of each warped grid cell from a similarity transformation of its initial shape. As shown in Fig. 2 (c), each grid cell can be divided into two triangles. In each triangle, we compute the local coordinates (u, v) for a vertex V_1 in a local coordinate system defined by the other two vertices, V_2, and V_3. Then, we have

$$V_1 = V_2 + u(V_3 - V_2) + v\mathbf{R}_{90}(V_3 - V_2), \mathbf{R}_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \qquad (6)$$

To encourage each grid cell to undergo a similarity transformation after warping, we only need to ensure that $\hat{V}_1$ can

still be represented by $\hat{V}_2$ and $\hat{V}_3$ using the same local coordinates (u, v) computed from its initial shape. Therefore, the similarity transformation term is defined as:

$$E_s(\hat{V}) = \sum_{i=1}^{N_t} \|\hat{V}_1^i - (\hat{V}_2^i + u(\hat{V}_3^i - \hat{V}_2^i) + v\mathbf{R}_{90}(\hat{V}_3^i - \hat{V}_2^i))\|^2,$$

(7)

where N_t is the total number of triangles in the grid mesh.

Line-preserving term The similarity transformation term alone is not sufficient to constrain structures larger than the grid cell. Lin *et al.* [14] introduced a curve preserving term to keep salient structures during warping. Here we only use constraints derived from straight lines. Specifically, we detect line segments in I_{tar} with the detector in [27]. For each line segment, we uniformly sample key points along it. For each key point on the line segment, we can compute a 1D coordinate u in the local coordinate system defined by the two endpoints of the line segment. To maintain the straightness of the line segment, we require the key point being represented by the same local coordinate u after warping. The line-preserving term is defined as:

$$E_l(\hat{V}) = \sum_{i=1}^{N_l} \sum_{j=1}^{N_k} \|L_{key}^{i,j} - (L_b^i + u(L_c^i - L_b^i))\|^2, \quad (8)$$

where N_l is the total number of line segments and N_k is the number of key points on each line segment i. The key point $L_{key}^{i,j}$ and endpoints L_b^i and L_c^i are further parameterized by the mesh vertices using bilinear interpolation. Please refer to [14] for more details.

Optimization All the cost functions are quadratic and can be easily minimized by any sparse linear solver. Each time after solving the linear equations, the mesh only deforms locally towards the final position. Therefore, we perform the optimization multiple times until the mesh becomes stable. Firstly, I_{tar} is divided into a $m \times n$ regular mesh ($m = 16$ and $n = 16$ in our implementation). Then, we uniformly sample points in I_{tar} and store them for later optimization. As the mesh gets updated in each iteration, we only use the stored sampled points inside the current overlapping region for further optimization. We consider the optimization to have converged if the average change of the vertex coordinates between iterations is smaller than a predefined threshold (one pixel in our implementation).

3.3. Coarse-to-fine Scheme

In order to handle large displacement between the input images, we adopt a coarse-to-fine scheme during the iteration. Specifically, we build a L-layer Gaussian pyramid for I_{tar} ($L = 3$) and the optimization starts from the top

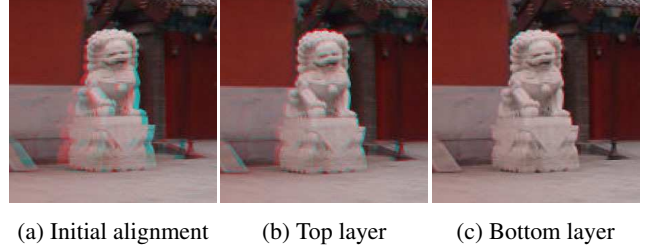(a) Initial alignment   (b) Top layer   (c) Bottom layer

Figure 3: Three-layer coarse-to-fine scheme for photometric alignment. (a) Initial alignment before optimization. (b) Optimization result on the top layer. (c) Optimization result on the bottom layer.

layer (6.25% of the original image resolution) to the bottom layer (full image resolution). For each layer, we use a fixed mesh resolution and perform our photometric alignment on it. The resulting mesh vertices are multiplied by 2 to serve as an initial alignment for the next layer. Fig. 3 shows an example of the coarse-to-fine scheme.

4. Quantitative Evaluation

To demonstrate the effectiveness of our MPA method, we conduct several experiments to quantitatively evaluate MPA on real images. The data we use include both discrete image pairs and consecutive frames from videos. For discrete image pairs with large motion or significant illumination change, we apply pre-processing operations before applying our MPA.

4.1. Pre-processing

For initial alignment of the discrete images, we first use SIFT [29] to extract sparse feature matches, followed by an outlier filtering method from [32]. Then, we apply a global homography estimated using the inliers to pre-warp I_{tar} to I_{ref}. To minimize the influence of illumination change between the input images, we first normalize the original color images according to the scheme in [9] which provides invariance under affine illumination change. After this step, we apply our alignment method to these normalized images.

4.2. Evaluation on Images Pairs

For discrete image pairs used in image stitching, we compare our MPA with two state-of-the-art image stitching methods, namely, APAP [30] and Curve-preserving warp [14]. The test image pairs include those commonly used in recent literature and those collected by us (Fig. 4, top three rows). Since no ground truth alignment is available for these images, we evaluate the alignment quality by computing the local similarity in the overlapping region. Specifically, we use the same accuracy measurement in [12] to assess the alignment quality of two aligned images. We compute the

Low-texture　Large parallax　Rotation　Zooming　Rolling shutter　Crowd

Figure 4: Our dataset of image pairs amd videos for quantitative evaluation. *Top Three Rows*: Image pairs (**01-06** are from [30], **07-09** are from [12], **10-12** are from [31], **13-15** are from [14], **16-18** are ours). *Bottom Two Rows*: Videos (**01-02** are ours. **03-12** are from [19] and [20]).

RMSE of one minus normalized cross correlation (NCC) over a neighborhood π of 5×5 window for pixels in the overlapping region, *i.e.*,

$$RMSE(I_{tar}, I_{ref}) = \sqrt{\frac{1}{N} \sum_{\pi} (1.0 - NCC(\mathbf{p}_{ref}, \mathbf{p}_{tar}))}, \tag{9}$$

where N is the number of pixels in the overlapping region π; $\mathbf{p}_{ref}$ and $\mathbf{p}_{tar}$ are the pixels in I_{ref}, I_{tar} respectively.

For curve-preserving warp and APAP, we tune the parameters of these methods to achieve the best results we can get according to the guideline suggested by the authors. Since we target global alignment in this experiment, we set equal weights to the features in the curve-preserving warp method [14]. The RMSE results from different methods are shown in Table 1. As we can see, our MPA produces better alignment than the curve-preserving warp method in most of the cases and consistently outperforms APAP [30] even on their selected datasets (**01-06**) as well as image pairs **07-09** from [12], on which the APAP method outperforms the dual-feature method [12]. APAP interpolates the pixel motion from sparsely distributed features and the alignment quality is restricted by the distribution of correctly matched feature points. In contrast, our MPA utilizes densely sampled points and image gradients for alignment guidance, and thus usually performs better globally regardless of the small grid resolution used in our MPA. Finally, Fig. 5 uses image pair **09** to show that these improvements brought about by our MPA method are often perceptually noticeable.

No.	APAP	Curve.	MPA	No.	APAP	Curve.	MPA
01	6.39	5.46	**4.65**	**10**	19.9	17.8	**16.8**
02	14.8	14.3	**11.8**	**11**	17.8	16.1	**12.0**
03	11.9	11.5	**10.4**	**12**	38.3	38.9	**32.5**
04	6.26	**5.11**	5.25	**13**	19.8	18.0	**14.5**
05	5.78	5.21	**5.19**	**14**	10.5	10.2	**7.7**
06	12.2	10.7	**9.73**	**15**	6.68	8.88	**4.94**
07	13.8	**13.38**	13.6	**16**	16.1	14.0	**13.0**
08	2.3	2.74	**1.69**	**17**	9.06	8.8	**6.06**
09	5.37	5.15	**2.80**	**18**	12.9	10.7	**2.87**

Table 1: RMSE results on image pairs for image stitching. *Curve.*: alignment errors using curve-preserving warp [14].
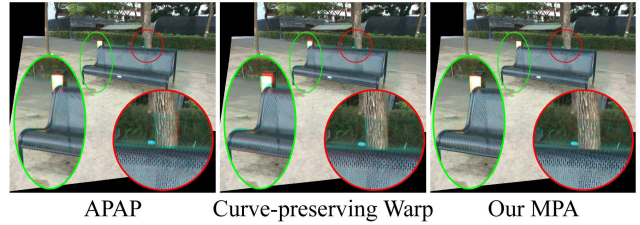

APAP　　　Curve-preserving Warp　　　Our MPA

Figure 5: Comparison with APAP [30] and curve-preserving warp [14] on image pair **09**.

4.3. Evaluation on Video Frames

The motion estimation for video stabilization requires consistently good results for satisfactory outcome. We quantitatively compare our method with two state-of-the-art motion estimation methods for video stabilization, namely, Liu *et al.*'s as-similar-as-possible warping (ASAP) [19] and the non-parametric motion estimation method in MeshFlow [18]. The former is known for its ability of handling parallax and the latter is the most recent work, which achieves real-time performance. The test videos are as shown in Fig. 4 (bottom two rows), categorized by camera motions and scene contents. For videos with dynamic foreground objects, we adopt an iterative foreground motion suppression scheme to minimize the adverse effect of foreground motion on camera motion estimation (see Sec. 6.1). For each video, we compute the alignment error between adjacent frames according to Eq. (9) and plot them in Fig. 6. For better visualization, we only show the errors of uniformly sampled frames in the video. For most cases, our MPA produces better alignment quality than ASAP and Mesh-Flow. More importantly, MPA exhibits a stable algorithm behaviour along the entire timeline regardless of the type of camera motion and scene content. In general, our method can process $2 \sim 5$ frames (640×360 resolution) per second on a PC with 2.4GHz CPU. Thus, MPA is competitive against the other two methods, considering the good balance it achieves between efficiency and high quality results.
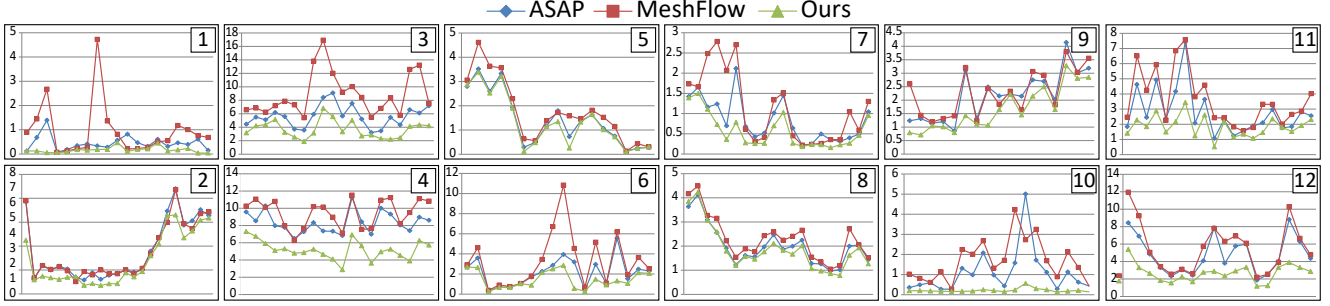
Figure 6: Quantitative alignment quality comparison on videos. *x-axis*: frame index on the timeline. *y-axis*: alignment error.

5. Application I: Stitching the Difficult Ones

Having demonstrated in Sec. 4.1 the effectiveness and advantage of MPA on typical images used for the stitching task, we now turn our attention to images that are expected to pose challenges for most state-of-the-art techniques.

5.1. Images with Large Parallax

For images with large parallax, local alignment methods [7, 31, 14] that search for a visually plausible stitching seam in local regions are usually better than global alignment methods [17, 30]. MPA can help with the local alignment in regions with few matched features around the final stitching seam. Fig. 7 shows an example of improved local alignment on the result from the state-of-the-art local alignment method SEAGULL [14]. We used the codes from [14] to generate the locally aligned meshes for this example. Then, we apply MPA only in the final stitching seam region to further improve the seam quality. As we can see, a feature-based local alignment method sometimes cannot guarantee good alignment in local regions with few feature matches. In such cases, MPA can be used as a post-processing tool to effectively remove small misalignment that is otherwise hard to get rid of for better stitching quality.

5.2. Low-Textured Images

Most stitching methods [2, 6, 16, 30, 31, 4, 13, 15] use sparse keypoint matches to estimate the motion model. For low-textured images, these methods may fail due to the paucity of matches in the low-textured regions. Li *et al.* [12] proposed to use dual-features for image alignment and their method outperforms state-of-the-art keypoint-based methods. However, this method still suffers in low-textured regions without robust line correspondences. To evaluate the effectiveness of MPA on low-textured images, we compare our method with their method [12] on their selected low-textured images, since the source code of their method is not available. Fig. 8 shows the comparison results. As we can see, both methods work well on these images and our method produces better alignment quality on image pairs
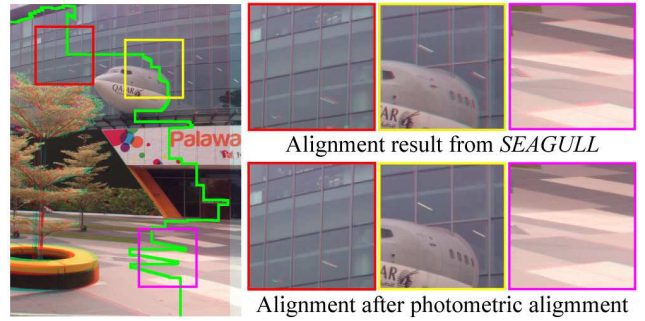

Figure 7: MPA in local seam region. *Top-Right*: Alignment around the final stitching seam from SEAGULL [14]. *Bottom-Right*: Improved alignment in local seam regions.

door and *shelf* where the extraction of line segments is difficult on small structure and over weak gradients. Our method utilizes gradient information for alignment optimization directly, thus avoiding potential problems caused by failure in line segment detection and matching.

6. Application II: Video Stabilization

In video stabilization, the presence of any dynamic foreground objects usually interferes with the estimation of the camera motion, and should thus be excluded during the motion recovery. A comprehensive solution to this problem is out of the scope of this work. However, assuming that the background motion is dominant in each video frame, we have incorporated the following simple iterative warping scheme to handle videos with dynamic foreground objects.

6.1. Dynamic Foreground Motion Suppression

Conventional sparse feature based stabilization methods use RANSAC to detect the features on dynamic objects. However, this only provides sparse partial information about the full extent of the dynamic regions. Liu *et al.*'s method [20] analyzes the behavior of the pixel profiles in a local time domain with known per-pixel motion, which is not at our disposal here. Some unsupervised and supervised object segmentation methods [22, 28, 26, 33, 1, 5]

Dual-feature results   *Our results*

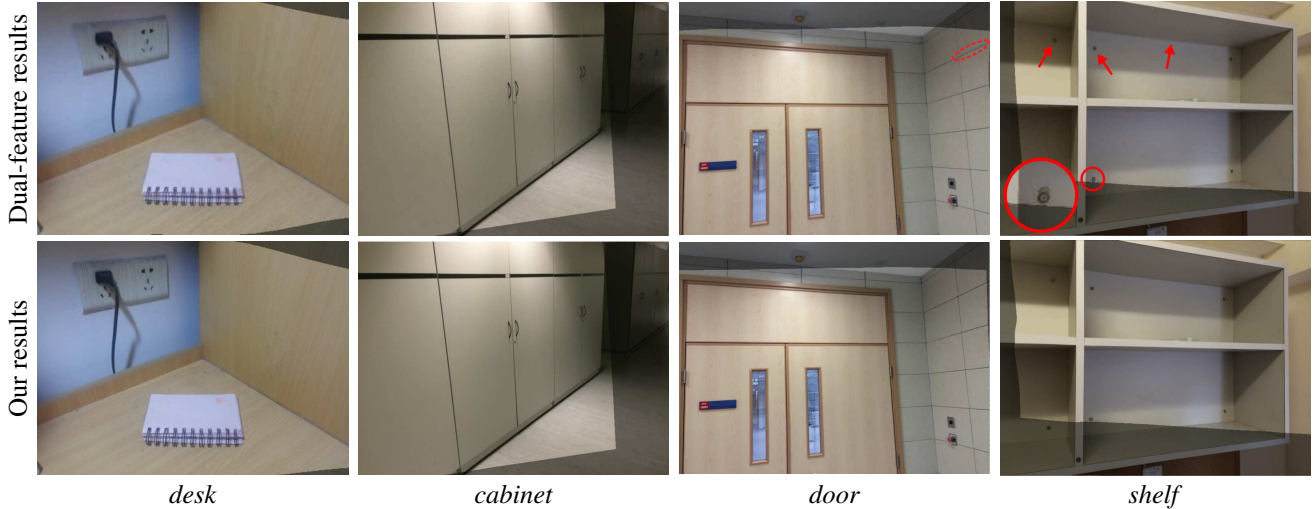*desk*   *cabinet*   *door*   *shelf*

Figure 8: Comparison with the dual-feature method [12]. *Top row*: Results from the dual-feature method. *Bottom*: Results from our photometric alignment. Here we use the same blending method as in [12] for fair comparison.

can also be used for dynamic object segmentation. However, these methods are usually either time-consuming or require manual annotation. Here, we use an online iterative dynamic motion suppression scheme that can be easily integrated into our MPA with little computational overhead.

The iterative scheme is based on the observation that dynamic foreground objects usually cause large alignment errors after the mesh-based warping, due to the regularization terms. We can use this information to roughly estimate the local region of those dynamic objects, and avoid using the sampled points in these regions in a new run. Specifically, we adopt a sampling mask for I_{tar} in our MPA and optimize the alignment by iterating over three steps. Firstly, we perform our photometric alignment using 'unmasked' sampled points. The sampling mask is initialized to blank. Then, we compute per-pixel alignment errors by calculating the intensity difference in the overlapping region. Finally, for pixels with errors larger than a predefined threshold, we compute their original locations before the warping and mark these locations as 'masked'. Then, we discard the previous warping result and re-start our MPA method with the newly updated sampling mask. We stop updating the mask when the change of the mask is small. Fig. 9 shows some examples of dynamic motion removal. As can be seen, the dynamic motions have been effectively filtered off.

6.2. Comparison on Low-Textured Videos

For rich-textured videos, sparse feature based methods [17, 19, 18] or flow-based method [20] usually generate satisfactory alignment results for video stabilization tasks. However, for low-textured videos, sufficient keypoints or robust flows usually cannot be guaranteed. The video frames may consistently have a small number of features

Figure 9: Foreground motion removal. *Left*: input images. *Middle*: Conventional optical flow. *Right*: Our flow from the warped mesh.

or the features are clustered in only a small portion of the image region. Other than low-textured videos, many videos may have frames that occasionally contain large portion of low-textured scenes. These cases pose significant challenges for camera motion estimation. It is also non-trivial to filter out the resulting wrong motions in the subsequent camera path optimization process [19, 20].

To demonstrate the effectiveness of our alignment method in such cases, we apply two state-of-the-art camera path optimization methods, namely, bundled path optimization [19] and SteadyFlow with its pixel profile optimization [20], on our alignment and compare the final stabilized results with those from the original methods. Since our MPA and ASAP [19] both represent camera motions as deformed meshes, we can directly apply the bundled path optimization method on our meshes to get a stabilized video. To apply SteadyFlow, we first compute flows from our meshes and then apply the pixel profile optimization method [20]. Fig. 10 shows the typical comparison results drawn from the video. As we can see, Liu *et al.*'s method [19] gener-
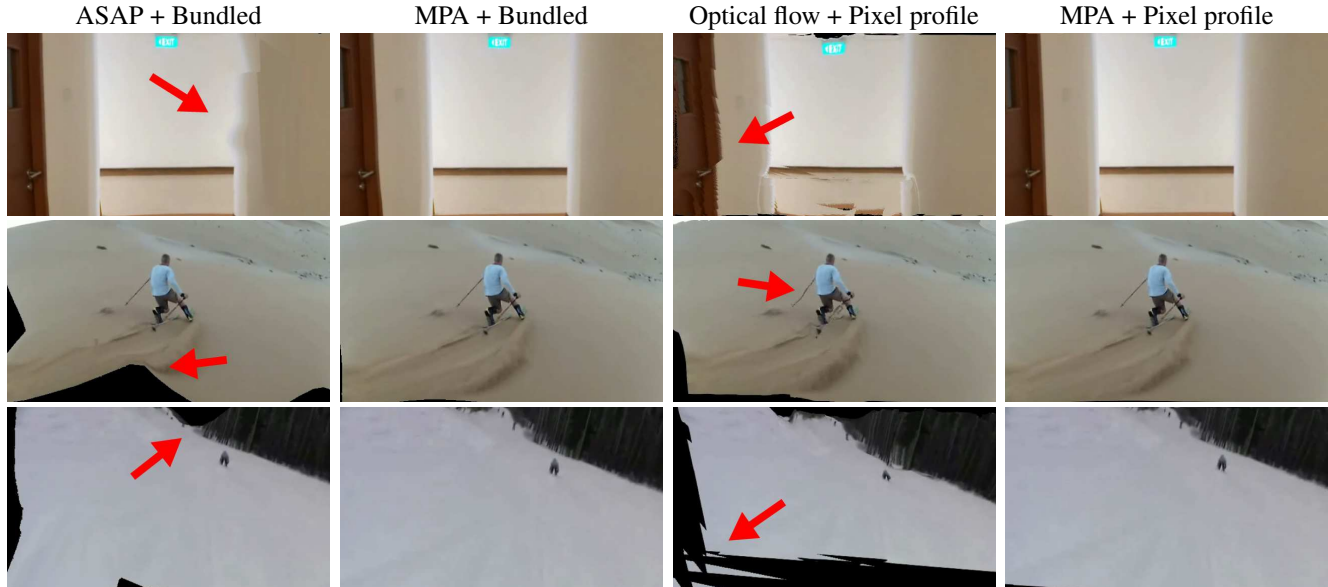
Figure 10: Comparison with Liu *et al.* [19] and SteadyFlow [20] on low-textured videos. *ASAP + Bundled*: Results from ASAP alignment and bundled path optimization. *MPA + Bundled*: Results from MPA alignment and bundled path optimization. *Optical flow + Pixel profile*: Results from optical flow alignment and pixel profile optimization. *MPA + Pixel profile*: Results from MPA alignment and pixel profile optimization.

ates noticeable distortions in low-texture regions, whereas SteadyFlow results in distorted and broken content due to unreliable optical flow estimated in homogeneous regions. Our alignment, on the other hand, produces significantly better stabilization quality on these low-textured videos.

We also implement a baseline that uses sparse matches obtained directly from optical flow for mesh warping on low-textured videos. Specifically, we first generate point trajectories using [24] and extract semi-dense matches between adjacent frames from them. Then, we apply content-preserving warp [17] using these matches to align consecutive frames. Finally, bundled path optimization is applied on the warped meshes to generate the stabilized results. Since matches estimated directly from optical flow can be very unreliable in low-textured areas due to flow errors, which often results in inaccurate mesh alignment, the stabilized results still suffer from unpleasant distortions. Our MPA, on the other hand, does the matching and alignment simultaneously with more advanced mesh regularizations to avoid gross alignment errors in these difficult areas and achieves reasonable alignment quality for the video stabilization task. The complete video stabilization results are provided in the supplementary video.

7. Conclusion & Future Work

In this paper, we propose a mesh-based photometric alignment method that generates high quality image warping for applications like image stitching and video stabiliza-

tion. Our method takes advantage of the direct photometric alignment's reliable performance in both low- and rich-textured input and formulates the semi-dense alignment optimization as an efficient mesh deformation process. The experiment results show that our method can handle a variety of images and videos, and outperforms many stat-of-the-art motion estimation methods in both image stitching and video stabilization tasks, especially for low-textured images and videos. We also observed several limitations for the current work. Firstly, MPA may not be able to estimate the camera motion correctly if the images contain large portion of homogeneous regions without any salient structures for alignment guidance, although the artifacts is usually visually unnoticeable. Secondly, we do not explicitly handle points on object boundary or in occluded regions, which may lead to misalignment in these local regions. One possible solution is to use $L1$ instead of $L2$ optimization when solving for the mesh vertices. These are all interesting directions for exploration in future work.

8. Acknowledgements

References

[1] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *Proc. CVPR*, June 2014. 6

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, 2007. 1, 2, 6

[3] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 25–36. Springer, 2004. 3

[4] C.-H. Chang, Y. Sato, and Y.-Y. Chuang. Shape-preserving half-projective warps for image stitching. In *Proc. CVPR*, 2014. 1, 6

[5] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: Non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graph.*, 34(6):195:1–195:10, Oct. 2015. 6

[6] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *Proc. CVPR*, 2011. 1, 2, 6

[7] J. Gao, Y. Li, T.-J. Chin, and M. S. Brown. Seam-driven image stitching. In *Eurographics*, pages 45–48, 2013. 6

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 2

[9] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *Proc. CVPR*, pages 2427–2434, 2013. 4

[10] W. Jiang and J. Gu. Video stitching with spatial-temporal content-preserving warping. In *CVPR Workshops*, 2015. 1, 2

[11] J. Lee, B. Kim, K. Kim, Y. Kim, and J. Noh. Rich360: Optimized spherical representation from structured panoramic camera arrays. *ACM Trans. Graph.*, 35(4):63:1–63:11, July 2016. 1, 2

[12] S. Li, L. Yuan, J. Sun, and L. Quan. Dual-feature warping-based motion model estimation. In *Proc. ICCV*, pages 4283–4291, 2015. 1, 2, 4, 5, 6, 7

[13] C.-C. Lin, S. U. Pankanti, K. N. Ramamurthy, and A. Y. Aravkin. Adaptive as-natural-as-possible image stitching. In *Proc. CVPR*, 2015. 1, 2, 6

[14] K. Lin, N. Jiang, L.-F. Cheong, M. Do, and J. Lu. Seagull: Seam-guided local alignment for parallax-tolerant image stitching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 4, 5, 6

[15] K. Lin, S. Liu, L.-F. Cheong, and B. Zeng. Seamless video stitching with hand-held camera inputs. *Comput. Graph. Forum (Proc. of Eurographics 2016)*, 35(2):479–487, May 2016. 1, 2, 6

[16] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *Proc. CVPR*, 2011. 1, 2, 6

[17] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Trans. Graph.*, 28(3):44:1–44:9, July 2009. 1, 2, 3, 6, 7, 8

[18] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision*, pages 800–815. Springer, 2016. 2, 5, 7

[19] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. *ACM Trans. Graph.*, 32(4):78:1–78:10, July 2013. 1, 5, 7, 8

[20] S. Liu, L. Yuan, P. Tan, and J. Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proc. CVPR*, pages 4209–4216. IEEE, 2014. 1, 2, 5, 6, 7, 8

[21] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1150–1163, July 2006. 1, 2

[22] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proc. ICCV*, December 2013. 6

[23] F. Perazzi, S.-H. Alexander, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross. Panoramic video from unstructured camera arrays. *Comput. Graph. Forum (Proc. of Eurographics 2015)*, 32(2), 2015. 1, 2

[24] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *Proc. ECCV*, pages 438–451, 2010. 8

[25] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 251–258, 1997. 2

[26] B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *Proc. CVPR*, June 2015. 6

[27] 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. Mach. Intell.*, 32(4):722–732, 2010. 4

[28] W. Wang, J. Shen, and F. Porikli. Saliency-aware geodesic video object segmentation. In *Proc. CVPR*, June 2015. 6

[29] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). http://cs.unc.edu/~ccwu/siftgpu, 2007. 4

[30] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving DLT. In *Proc. CVPR*, 2013. 1, 2, 4, 5, 6

[31] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *Proc. CVPR*, 2014. 1, 5, 6

[32] G. Zhang, Y. He, W. Chen, J. Jia, and H. Bao. Multi-viewpoint panorama construction with wide-baseline images. *IEEE Trans. on Image Processing*, 25:3099–3111, 2016. 1, 2, 4

[33] F. Zhong, X. Qin, Q. Peng, and X. Meng. Discontinuity-aware video object cutout. *ACM Trans. Graph.*, 31(6):175:1–175:10, Nov. 2012. 6