Dual-Feature Warping-based Motion Model Estimation

[1]Shiwei Li [2]Lu Yuan [2]Jian Sun [1]Long Quan

[1]Hong Kong University of Science and Technology
{slibc, quan}@cse.ust.hk

[2]Microsoft Research
{luyuan, jiansun}@microsoft.com

Abstract

*To break down the geometry assumptions of conventional motion models (*e.g.*, homography, affine), warping-based motion model recently becomes popular and is adopted in many latest applications (*e.g.*, image stitching, video stabilization). With high degrees of freedom, the accuracy of model heavily relies on data-terms (keypoint correspondences). In some low-texture environments (*e.g.*, indoor) where keypoint feature is insufficient or unreliable, the warping model is often erroneously estimated.*

In this paper we propose a simple and effective approach by considering both keypoint and line segment correspondences as data-term. Line segment is a prominent feature in artificial environments and it can supply sufficient geometrical and structural information of scenes, which not only helps lead to a correct warp in low-texture condition, but also prevents the undesired distortion induced by warping. The combination aims to complement each other and benefit for a wider range of scenes. Our method is general and can be ported to many existing applications. Experiments demonstrate that using dual-feature yields more robust and accurate result especially for those low-texture images.

1. Introduction

The theory of conventional 2D motion models[1] is well studied [10, 22]. These models are widely used in many applications, including image registration [22], panorama [2] , and video stabilization [21], since they are parametric, computationally efficient and robust to outliers compared with general motion models (*e.g.*, optical flow [13]). Nevertheless, these conventional models are limited by ideal geometry assumptions. For example, homography only provides high accuracy modeling for single-plane scene or purely rotational camera motion.

To break down these geometry assumptions, some warping-based motion models (*a.k.a.*, mesh-based warping or multiple homographies) are proposed in recent years. Image warping is originally used for image editing (*e.g.*, shape manipulation [14], image resizing [3]). Recently, warping is also used to tackle the two-image alignment problem, such as image stitching [27, 29, 4, 15], video stabilization [16, 9, 19], and produces promising results. Such warping-based models adopt meshes transform to represent the camera motion. Comparing with single homography, it has more flexibility of dealing with scenes with multiple planes and parallax due to higher degree-of-freedom (DoF).

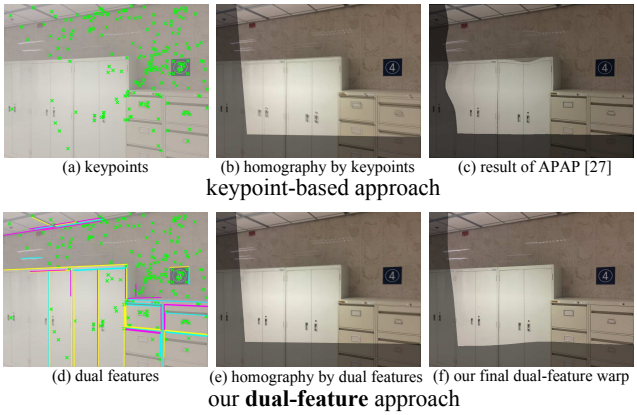


Figure 1. Top: the well-used keypoint-based approach (APAP [27] is the state-of-the-art warping-based model). Bottom: Our proposed dual-feature approach.

The estimation of warping-based models leverages matched keypoints in two images with smoothness constraints. The lack of reliable keypoints would easily cause misalignments (shown in Figure 1(b)(c)). Moreover, the distortion artifact is often induced by the flexible warps (shown in Figure 1(c)) due to insufficient corresponding information (*i.e.*, data-term). This artifact is highly noticeable and unpleasant. Although some warping models avoid distortion at salient regions [16] as possible as they can or alleviate the perspective distortion [4], the structural contents or rigid objects may not be explicitly preserved.

To deal with this problem, we resort to the line feature in image content, and consider it as another type of data term. Line structure is prominent in artificial scenarios (*e.g.*, Figure 1(d)). The line feature can not only be considered as the complement to keypoints, but also provide strong geometri-

[1]"conventional 2D motion model" refers to 3×3 homography, affine or similarity transformation.

cal and structural constraints. Recently, the progress in fast line detection [25, 1] and line matching [26, 30, 6] has made the line feature practical in many applications. The usage of *dual features* (keypoints and lines) can help achieve better estimation in single homography (shown in Figure 1(e)) and even more accurate estimation in warping-based motion model (shown in Figure 1(f)).

However, it is not a trivial task to combine dual features into an unified framework of model estimation. The most related trial is conducted by Dubrofsky [5], who combined Direct Linear Transform (DLT) formulations of points ($\mathbf{x}' = \mathbf{H}\mathbf{x}$) and lines ($\mathbf{l} = \mathbf{H}^{\mathrm{T}}\mathbf{l}'$) for single homography $\mathbf{H}$ estimation. In deed, the simple combination is hard to achieve a high-quality and robust estimation due to two reasons: 1) the parameters of line $[a, b, c]$ are not numerically stable when the detected line is short, vertical, horizontal or approaches the origin [28]; and 2) line has inconsistent distance metric [2] to keypoints, making the optimization, data normalization, outlier removal unfair or even intractable. These two fatal issues would make line feature virtually useless for most purposes.

In this paper, we present a novel dual-feature approach to warping-based motion model estimation, aiming at addressing the fragile problems of existing warping-based models, especially for low-texture images (shown in Figure 1(a)). We consider a different representation for detected *line segment*, which is parameterized by its two endpoints. The geometric distance of two line segments is thereby defined as the distance from endpoint-to-line, which has same metric to the Euclidean distance of points.

Based on the new parameterization and distance metric, we can derive a new dual-feature DLT formulation and address the normalization and RANSAC procedures for single homography estimation, known as global warp used in the first step of warping-based model estimation. Accordingly, we extend it to the second step, known as local warp, which further minimizes the registration error via mesh warping. Our quantitative evaluations indicate that the usage of dual features outperforms the well-used keypoint-based approach especially in low-texture conditions. Moreover, our dual-feature can be easily ported to existing applications (image stitching and video stabilization).

2. Related Work

With high DoF, warping-based motion model provides more flexibility of handling parallax than single homography model. These models are designed with specific priors. For example, Gao *et al.* [7] assumed most of scenes have two dominant planes (ground and distant plane), which can be described by combining two homographies. Lin *et al.* [15] modeled the scenes using smoothly varying affine field for image stitching. More recently, Zaragoza *et al.* [27] proposed an elegant as-projective-as-possible (APAP) warp for image stitching, which assumes global projectivity while allowing local non-projective deviations. Content-perserving-warps (CPW) is another warping-based model that encourages local transforms to be similarity. The technique has been successfully used in video stabilization [16, 19, 31] and image stitching [29]. Furthermore, mixture-of-gaussian model is a simplified 1D mesh motion model, which was used to rectify rolling shutter artifacts [8]. However, all of these existing models heavily rely on keypoint correspondences. Once the quality and number of keypoints become problematic, their estimations are usually not reliable.

[2]The algebraic distance $||\mathbf{l_1} - \mathbf{l_2}||$ of line and the distance $||\mathbf{p_1} - \mathbf{p_2}||$ of point have different physical measures.

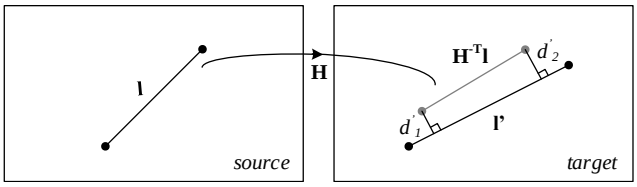


Figure 2. Illustration of line segment distance measurement with point-to-line distance, $d(\mathbf{l}, \mathbf{l}') = \sqrt{{d'_1}^2 + {d'_2}^2}$.

To consider additional correspondences, we resort to line feature. Actually in early history when people studied projective geometry, line and point were deemed equally important as they contain symmetric information (*i.e.*, *duality principle* [10]). However, the combination of both features is not popular due to the difficulties of line: in 2D, line cannot provide the exact position correspondence due to its *aperture problem*; in 3D, line requires at least three view (trifocal tensor) to construct constraints [23]. The most related work to our intended work is [5], which naively combined point and line for single homography estimation. However, their method is extremely susceptible to noise and may be not very suitable for warping-based model estimation. Unlike their method, we use a different representation (*line segment* instead of *line*) and a more reasonable distance metric for line, which help us easily consider dual features in warping-based motion model estimation.

In recent years, the progress in fast line detection [25, 1] makes the usage of line feature in image content popular. For example, some image editing algorithms [3, 12] explicitly detect the straight lines in images and preserve their properties (straightness, parallelism, etc) during warping. Note that in their methods, the line is only a constraint (or prior) for the warp. Indeed, our method considers *line correspondences* into the data term of motion estimation.

3. Dual-feature Representation and Matching

3.1. Parameterization and Distance Metric

The parametrization of 2D point in homogeneous coordinate is $\mathbf{p} : [x, y, 1]^{\mathrm{T}}$. Similarly, 2D line is parameterized

as a 3-vector, $\mathbf{l} : [a, b, c]^{\mathrm{T}}$, which corresponds to its standard equation ($\mathbf{l} : ax + by + c = 0$). Such a parametrization is convenient for matrix operation (*e.g.*, transform), but its parameters are not stable when the line is vertical, horizontal, or approaching the origin [28]. Besides, line can also be parameterized by polar coordinates $\mathbf{l} = (r, \theta)$ (*i.e.*, Hough space). It is geometrically intuitive but not convenient for matrix operation.

We adopt another parametrization – endpoint parameterization: a line segment $\mathbf{l}$ represented by its two endpoints $\mathbf{p}^0, \mathbf{p}^1$. Suppose $\mathbf{l}$ undergoes a transformation by model $\mathcal{M}$. The transformed line is $\hat{\mathbf{l}} = \mathcal{M} \circ \mathbf{l}$ and its endpoints become $\hat{\mathbf{p}}^0, \hat{\mathbf{p}}^1 (\hat{\mathbf{p}}^{0,1} = \mathcal{M} \circ \mathbf{p}^{0,1})$. Thereby, the geometrically meaningful distance can be defined as the endpoint-to-line distance. That is, the Euclidean distance between the transformed $\hat{\mathbf{l}}$ and the target $\mathbf{l}'$ is defined as the square root of sum of squared *perpendicular offset* from two transformed endpoints $\hat{\mathbf{p}}^0, \hat{\mathbf{p}}^1$ to $\mathbf{l}'$ (shown in Figure 2). *i.e.*,

$$d(\hat{\mathbf{l}}, \mathbf{l}') = \sqrt{d^2(\hat{\mathbf{p}}^0, \mathbf{l}') + d^2(\hat{\mathbf{p}}^1, \mathbf{l}')}, \tag{1}$$

where $\hat{\mathbf{p}} = [\hat{u}, \hat{v}, 1]^{\mathrm{T}}$, $\mathbf{l}' = [a', b', c']^{\mathrm{T}}$ and $d(\hat{\mathbf{p}}, \mathbf{l}') = \frac{|\mathbf{l}'^{\mathrm{T}} \cdot \hat{\mathbf{p}}|}{\sqrt{a'^2 + b'^2}}$ is the distance from transformed point $\hat{\mathbf{p}}$ to the target line $\mathbf{l}'$. Note that this distance is originally used for line-based *Structure from Motion* problem [23], and we borrow it for our 2D motion model estimation.

The motion model $\mathcal{M}$ is then estimated by jointly minimizing the well-used Euclidean distance of points and our newly defined distance of line segments:

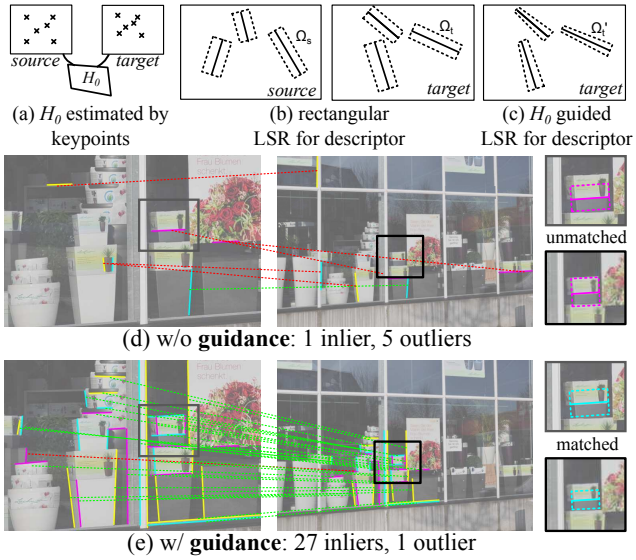
$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M}} \left(\sum\nolimits_i d^2 \left(\hat{\mathbf{p}}_i, \mathbf{p}'_i \right) + \sum\nolimits_j d^2 (\hat{\mathbf{l}}_j, \mathbf{l}'_j) \right), \tag{2}$$

where $\hat{\mathbf{p}}_i = \mathcal{M} \circ \mathbf{p}_i$ and $\hat{\mathbf{l}}_j = \mathcal{M} \circ \mathbf{l}_j$. i, j are the indices for keypoint and line segment respectively.

The endpoint parameterization for line segment has three advantages over previous representations: 1) the point-to-line distance is geometrically meaningful and has consistent metric with point-wise Euclidean distance; 2) it exempts from numerical unstable conditions [28] because the current geometric distance is invariant to rotation; 3) it provides useful locality information (length and endpoint position) for warping-based mesh model estimation.

3.2. Dual-feature Detection and Matching

To obtain keypoint matches, we simply use SIFT [20] feature implemented by VLFeat [24]. As for line segments, we adopt EDLine [1] for line detection, but line matching is more challenging due to less distinctive appearance, fragments of same line and no epipolar constraint. Although recent appearance-based line descriptors (MSLD [26], LBD [30]) produce satisfactory matching results with the presence of rotation, noise and illumination change, they cannot handle large scale or perspective change (Figure 3(d)). The major reason is that



(a) H_0 estimated by keypoints (b) rectangular LSR for descriptor (c) H_0 guided LSR for descriptor

(d) w/o **guidance**: 1 inlier, 5 outliers

(e) w/ **guidance**: 27 inliers, 1 outlier

Figure 3. Illustration of *guided line matching*.

the descriptor samples a fixed width of rectangular region (called *line supporting region*, LSR) around each line in both source and target images (Figure 3(b)), making them scale-sensitive. To get over, we leverages the global information provided by initial keypoints and propose a method called *guided line matching*.

Guided line matching. We consider two different LSRs respectively in source and target frames for extracting descriptors. The LSR in the source frame Ω_s is still a fixed-width rectangular region, while the LSR Ω'_t in the target frame may be a trapezoidal region (Figure 3(c)) transformed by an initial homography $\mathbf{H}_0$ (Figure 3(a)). Here, $\mathbf{H}_0$ is initially estimated keypoints and may provide the approximation to the scale/perspective change between two frames. In order to achieve transformed LSR Ω'_t, we first back-project the line segments from the target frame to the source frame by $\mathbf{H}_0^{-1}$, and then obtain the rectangular LSR Ω^*_t in the source frame. Next, we transform Ω^*_t to the target frame by $\mathbf{H}_0$ and yield Ω'_t, which is the final LSR for extracting descriptors in the target frame. Once the LSR for the line segment descriptor is determined, we use MSLD to encode the appearance of line into descriptor vectors, followed by the usual feature matching procedure. To reduce the biased estimation of $\mathbf{H}_0$, we adopt an iterative refinement. The pseudo codes are shown in Algorithm 1. In our experiments, it usually needs $2 \sim 3$ iterations for the convergence. Figure 3(e) shows our matching result.

4. Warping-based Motion Model Estimation

The estimation of the warping model undergoes a global step and a local step. The global transform helps greatly reduce the penalty of local warps. This strategy is proved to be effective in previous algorithms [21, 16, 19]. In the first step, a global projective warp (homography) will be estimated using line segments and keypoints (Section 4.1).

Algorithm 1 Guided line matching

estimate $\mathbf{H}_0$ from $\mathbf{p}' = \mathbf{H}_0\mathbf{p}$;
repeat
  $\mathrm{l}^* \leftarrow$ back-project l' by $\mathbf{H}_0^{-1}$;
  $\Omega_t^* \leftarrow$ the rectangular LSR for l^*;
  $\Omega_t' \leftarrow \Omega_t^*$ warped by $\mathbf{H}_0$;
  extract line descriptors in Ω_t' for l', Ω_s for l;
  find line correspondences $(\mathrm{l}, \mathrm{l}')$ between frames;
  update $\mathbf{H}_0$ from new correspondences (Section 4.1);
until corresponding pairs $(\mathbf{p}, \mathbf{p}')$, $(\mathrm{l}, \mathrm{l}')$ are not changed.

In the second step, we solve for a mesh model that further minimizes the registration error of feature correspondences via mesh warping (Section 4.2).

4.1. Dual-feature Homography Estimation

-0.5em We first consider the motion model $\mathcal{M}$ to be a 3×3 homography $\mathbf{H}$, which is estimated by minimizing the combined *geometric distances* according to Equation 1 and 2:

$$\min \left(\sum_i \|\mathbf{p}_i' - \hat{\mathbf{p}}_i\|^2 + \sum_j \frac{\left|\mathrm{l}_j'^{\mathrm{T}} \hat{\mathbf{p}}_j^0\right|^2 + \left|\mathrm{l}_j'^{\mathrm{T}} \hat{\mathbf{p}}_j^1\right|^2}{a'^2 + b'^2} \right), \quad (3)$$

where $\hat{\mathbf{p}}_i \sim \mathbf{H}\mathbf{p}_i$ and $\hat{\mathbf{p}}_j^{0,1} \sim \mathbf{H}\mathbf{p}_j^{0,1}$, and $\sim$ denotes equality up to a scalar factor.

Obtaining an optimal solution requires non-linear optimization (Levenberg-Marquardt (LM) iteration) because homography changes the value of homogeneous element. As suggested by Hartley [10], Direct Linear Transformation (DLT) algorithm is more preferable for its efficiency, linearity and simplicity in implementation. Even though it minimizes the *algebraic distance*, it can produce comparable accuracy to iterative methods (also as the initial solution of LM iteration) with a proper normalization [11, 10].

Here, we formulate Equation 3 in DLT fashion as well. Let $\mathbf{p}_i \leftrightarrow \mathbf{p}_i'$ and $\mathrm{l}_j \leftrightarrow \mathrm{l}_j'$ be pairs of keypoints and line segments, where $\mathbf{p}_i = [x_i, y_i, 1]^{\mathrm{T}}$ and $\mathrm{l}_j = [a_j, b_j, c_j]^{\mathrm{T}}$ with its two endpoints $\mathbf{p}_j^{0,1} = [u_j^{0,1}, v_j^{0,1}, 1]$. The mapping of keypoint should satisfy $\mathbf{p'}_i \times \hat{\mathbf{p}}_i = \mathbf{p'}_i \times \mathbf{H}\mathbf{p}_i = 0$, where $\times$ is cross product, and thus the *algebraic distance* $||\mathbf{p'}_i \times \mathbf{H}\mathbf{p}_i||$ is desired to be minimized.

As for the line segment, the transformed endpoints $\hat{\mathbf{p}}_j^{0,1}$ are expected to lie on the target line l_j' for line-to-line mapping, which can be denoted as $\mathrm{l}_j'^{\mathrm{T}} \hat{\mathbf{p}}_j^{0,1} = \mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1} = 0$. The *algebraic distance* $||\mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1}||$ should be minimized.

Rewriting Equation 3 in DLT fashion yields,

$$\begin{aligned} \hat{\mathbf{H}} &= \arg\min_{\mathbf{H}} \left(\sum\nolimits_i ||\mathbf{p}_i' \times \mathbf{H}\mathbf{p}_i||^2 + \sum\nolimits_j || \mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1}||^2 \right) \\ &= \arg\min_{\mathbf{H}} \left(\sum\nolimits_i ||\mathbf{A}_i\mathbf{h}||^2 + \sum\nolimits_j ||\mathbf{B}_j\mathbf{h}||^2 \right), \end{aligned}$$

where $\mathbf{h} = [h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9]^{\mathrm{T}}$ is a 9-vector

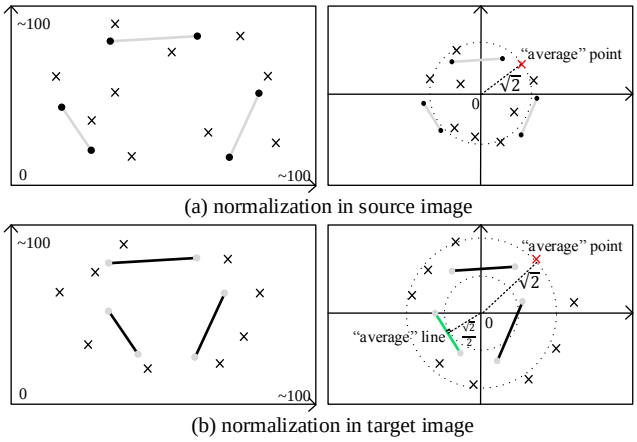


Figure 4. Illustration of normalization in source and target images.

of the entries of $\mathbf{H}$. The matrix

$$\mathbf{A}_i = \begin{bmatrix} x_i, y_i, 1, 0, 0, 0, -x_i'x_i, -x_i'y_i, -x_i' \\ 0, 0, 0, x_i, y_i, 1, -y_i'x_i, -y_i'y_i, -y_i' \end{bmatrix}$$

can be inferred from $\mathbf{A}_i\mathbf{h} = \mathbf{p'}_i \times \mathbf{H}\mathbf{p}_i = 0$. The matrix

$$\mathbf{B}_j = \lambda_j \begin{bmatrix} a_j'u_j^0, a_j'v_j^0, a_j', b_j'u_j^0, b_j'v_j^0, b_j', c_j'u_j^0, c_j'v_j^0, c_j' \\ a_j'u_j^1, a_j'v_j^1, a_j', b_j'u_j^1, b_j'v_j^1, b_j', c_j'u_j^1, c_j'v_j^1, c_j' \end{bmatrix}$$

can be inferred from $\mathbf{B}_j\mathbf{h} = \mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1} = 0$, where λ_j is a scalar factor for balancing with $\mathbf{A}_i$.

Stacking up all the formulations of points ($\mathbf{A}_i$) and line segments ($\mathbf{B}_j$) forms $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{h} = 0$, where SVD decomposition is used to obtain the coefficients $\mathbf{h}$.

Balancing is crucial since two types of feature are used in a unified minimization framework. Let $\mathbf{p}$ be any point or endpoint, $w = [h_7, h_8, h_9]\mathbf{p}$ is the scalar to achieve equality: $w\hat{\mathbf{p}} = \mathbf{H}\mathbf{p}$. The point's minimized algebraic residual $||\mathbf{A}_i\mathbf{h}||$ can be derived to $d\left(\mathbf{H}\mathbf{p}_i, \mathbf{p}_i'\right) \cdot w_i$ (*i.e.*, the geometric distance multiplying w_i).

The residual of line segment $||\mathbf{B}_j\mathbf{h}||$ can be expanded to $\lambda_j \mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1}$. To be consistent, we require the residual to equal to the geometric distance multiplying w_j:

$$\lambda_j \mathrm{l}_j'^{\mathrm{T}} \mathbf{H}\mathbf{p}_j^{0,1} = d\left(\hat{\mathbf{p}}_j^{0,1}, \mathrm{l'}_j\right) \cdot w_j^{0,1} \Rightarrow \lambda_j = \frac{1}{\sqrt{a_j'^2 + b_j'^2}}.$$

With the scalar λ_j, the weight between two residuals is balanced in terms of geometric meaning.

Normalization is an indispensable step to improve numerical precision. Its ultimate goal is to reduce the *condition number* of matrix in solving SVD decomposition. Hartley's normalization [11, 10] applies scaling and translation on the 2D point coordinate such that an "average point" is $[1, 1, 1]$, which makes all entries in matrix $\mathbf{A}$ have similar magnitude (~ 1) and thus the condition number of $\mathbf{A}$ is greatly reduced.

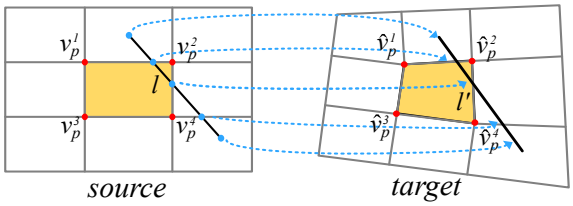


Figure 5. Line segment correspondences in mesh-based model.

In our dual-feature formulation, hopefully all entries of the stacked matrix $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$ are close to 1. Please note that in the *source* image, only points and endpoints (their parameters: x, y, u, v) are used in the formulation. Then we simply employ Hartley's normalization to all these points, namely compute a similarity transform (translate and scale) $\mathbf{T}$ applying to these points such that an "average point" is $[1, 1, 1]$ (Figure 4(a)).

The case is different in *target* image, because we actually use target points and lines (their parameters: x', y', a', b', c') in the formulation. To make these values close to 1, we first normalize a', b' by dividing c', yielding $[a'/c', b'/c', 1]$. Now the distance of this line to origin is $\frac{1}{\sqrt{(a'/c')^2+(b'/c')^2}}$, while the distance of a point to origin is $\sqrt{x'^2+y'^2}$. To ensure $a'/c', b'/c', x', y' \sim 1$, we compute the best similarity transform $\mathbf{T}'$ via least-square such that the average distance from all lines to the origin is $\frac{1}{\sqrt{1^2+1^2}} = \frac{1}{\sqrt{2}}$, while the average distance from keypoints to origin is $\sqrt{1^2+1^2} = \sqrt{2}$.

After applying $\mathbf{T}$ and $\mathbf{T}'$ to the source and target images, the homography $\mathbf{H}^\star$ is computed in normalized space. The final solution is obtained by denormalization: $\hat{\mathbf{H}} = \mathbf{T}'^{-1}\mathbf{H}^\star\mathbf{T}$. The effectiveness of our normalization and numerical stability issue are evaluated in Section 5.

Robust estimation. RANSAC is commonly used to remove outliers in robust estimation for homography. Based on consistent distance metric between both features, we can apply RANSAC on dual features simultaneously, which is intractable in previous work [5]. Specifically, in the computation of RANSAC penalties, the fitting error for point uses Euclidean distance $d(\hat{\mathbf{p}}, \mathbf{p}') = \sqrt{||\mathbf{p}' - \hat{\mathbf{p}}||^2}$, and the fitting error for line segment uses $d(\hat{\mathbf{l}}, \mathbf{l}') = \sqrt{\frac{|\mathbf{l}'^{\mathrm{T}}\cdot\hat{\mathbf{p}}^0|^2+|\mathbf{l}'^{\mathrm{T}}\cdot\hat{\mathbf{p}}^1|^2}{a'^2+b'^2}}$, where $(\hat{\mathbf{p}}^{\mathbf{0}}, \hat{\mathbf{p}}^{\mathbf{1}})$ are two endpoints of transformed line segment $\hat{\mathbf{l}}$. The inliers of dual-feature will be further used for local warp estimation in Section 4.2.

4.2. Dual-feature Local Warps

After global warping using single homography, the registration error of feature correspondences is greatly reduced. In the second step, we employ a mesh warp to further minimize the registration error. Here, we follow the framework of as-similar-as-possible warp [16] and extend it by incorporating a new data term of line segment.

The image is first divided by regular meshes (shown in Figure 5). The vertices of the mesh model that we solve for are denoted as $\mathbf{V}$ (indicated as red dots in Figure 5). Let $\mathbf{l} \leftrightarrow \mathbf{l}'$ be a pair of matched line segments and $\mathbf{p}^0, \mathbf{p}^1$ are two endpoints of line segment $\mathbf{l}$. When $\mathbf{l}$ goes across more than one meshes, we cut $\mathbf{l}$ into multiple line segments using the boundary edges of mesh. Supposing $\{\mathbf{p}_k\}$ represents all the endpoints (indicated as blue dots in Figure 5) of the cut segments in the source image, according to Equation 1, we require the distance from all these endpoints to the target line, *i.e.*, $d(\mathbf{l}', \mathbf{p}_k) = \frac{|\mathbf{l}'^{\mathrm{T}}\mathbf{p}_k|}{\sqrt{a'^2+b'^2}}$, to be minimized. To represent $\{\mathbf{p}_k\}$ by $\mathbf{V}$, we adopt bilinear interpolation, *i.e.*, $\mathbf{p}_k = \mathbf{w}_{\mathbf{p}_k}\mathbf{V}_{\mathbf{p}_k}$, where $\mathbf{V_p} = [\mathbf{v_p^1}, \mathbf{v_p^2}, \mathbf{v_p^3}, \mathbf{v_p^4}]$ is four vertices of enclosing quad of $\mathbf{p}_k$, and $\mathbf{w_p} = [w_p^1, w_p^2, w_p^3, w_p^4]^{\mathrm{T}}$ are the bilinear interpolation weights that sum to 1. Therefore, our data term for line segments is defined as

$$E_{line}(\mathbf{V}) = \sum\nolimits_{j,k} ||(\mathbf{l'_j}^{\mathrm{T}} \cdot \mathbf{V_{p_k}}\mathbf{w_{p_k}})/(\sqrt{a_j'^2 + b_j'^2})||^2.$$

The merits of $E_{line}(\mathbf{V})$ are twofold: 1) as a "data-term", it conduces to better alignment for line structures, and 2) it naturally preserves the straightness property of line during warping, as it requires all cut endpoints to be mapped onto the same straight line (like a "constraint").

The data term of keypoints $E_{point}(\mathbf{V})$ and smoothness term $E_{smoothness}$ are directly borrowed from [16]: Let $\mathbf{p_i} \leftrightarrow \mathbf{p_i'}$ be a pair of matched keypoints. Representing $\mathbf{p}_i$ by the bilinear interpolation of its locating quad, the data term for keypoints can be finally written as:

$$E_{point}(\mathbf{V}) = \sum\nolimits_i ||\mathbf{V_{p_i}}\mathbf{w_{p_i}} - \mathbf{p_i'}||^2.$$

The smoothness term $E_{smoothness}$ encourages every grid to undergo similarity transform. See [16, 19] for detailed formulation.

The final objective function combines two data terms and smoothness term together, *i.e.*,

$$E(\mathbf{V}) = E_{point}(\mathbf{V}) + E_{line}(\mathbf{V}) + \alpha E_{smoothness}(\mathbf{V}).$$

The scalar α (by default, $\alpha = 0.25$) balances the tradeoff between data terms and smoothness term. We usually use a mesh of 32×32 grids for 720p image resolution. Since the objective function is quadratic, it can be solved by a sparse linear solver. Finally, each local homography can be inferred from the transformation of each mesh vertex.

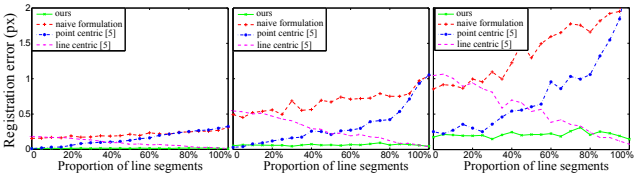
Figure 6 shows a comparison between two mesh-based estimations of only using keypoints (w/o E_{line}) [16, 19] and using dual features (w/ E_{line}). Likewise, our dual-feature method should be able to applied to other warping-based models (*e.g.*, APAP [27]), since we only need slightly modify their data terms. We will leave it as a future work.

(a) keypoint feature (b) keypoint-based homography (c) keypoint-based warp (w/o E_{line}) (d) line segment feature (e) our dual-feature warp (w/ E_{line})

Figure 6. Comparison of mesh-based homographies estimation used in image registration. Misalignments are illustrated by red arrows.

5. Quantitative Evaluation

Numerical stability. The numerical stability is an important indicator for linear methods of model fitting. Inspired by [11], the first experiment evaluates our dual-feature DLT formulation and normalization (in Section 4.1) in terms of numerical stability (*i.e.*, noise sensitivity). We synthesized a virtual image (with the resolution 1024 $\times$ 800 pixels) by randomly generating points and line segments with a total number of 300 features. We regard it as the source image. In order to synthesize the target image, we map all of original points and line segments to new positions by a predefined 3 $\times$ 3 homography matrix $\overline{\mathbf{H}}$. In addition, the mapped position is randomly perturbed by a factor ϵ[3]. In our experiment, we use all of corresponding feature pairs to estimate the homography $\hat{\mathbf{H}}$. The quality of model is measured by well-used average *registration error*, *i.e.*, $E(\overline{\mathbf{H}}, \hat{\mathbf{H}}) = \frac{1}{N} \cdot \sum_{\mathbf{x} \in \mathbf{I}} ||\overline{\mathbf{H}}\mathbf{x} - \hat{\mathbf{H}}\mathbf{x}||^2$, where $\mathbf{x}$ are all image pixels and N is the total pixel number.



(a) Perbutation factor $\epsilon = 0.5$ (b) Perbutation factor $\epsilon = 1.0$ (c) Perbutation factor $\epsilon = 2.0$

Figure 7. Comparisons of registration error between our formulation and others. Best viewed on screen.

ratio	naive formulation	point centric	line centric	ours
0%	2.12E+6	**1.04E+3**	2.12E+6	**1.04E+3**
20%	2.28E+6	2.87E+3	1.32E+6	**1.05E+3**
40%	2.60E+6	5.18E+4	6.78E+5	**1.12E+3**
60%	3.11E+6	5.30E+4	4.20E+4	**1.19E+3**
80%	3.20E+6	1.34E+5	3.26E+3	**1.30E+3**
100%	8.75E+6	8.75E+6	1.62E+3	**1.49E+3**

Table 1. The *condition number* data of matrix $\mathbf{A}$ in experiment Figure 7(c). "ratio" refers to the proportion of line segments.

We compared four linear approaches of homography estimation using dual features: *naive formulation* is to directly combine $\mathbf{p}' = \mathbf{H}\mathbf{p}$ and $\mathbf{l}_i = \mathbf{H}^{\mathrm{T}}\mathbf{l}'_i$ without normalization; *point centric* [5] is to conduct normalization only on point data; *line centric* [5] is to conduct normalization only on line segment data; *our method* (dual-feature homography, described in Section 4.1). For each approach, we test three different *perturbation factors* $\epsilon \in \{0.5, 1.0, 2.0\}$ and try a varying percentage of line segments from 0% to 100%. Figure 7 shows the comparison results. As we can see, *naive formulation* produces the worst result, and is very sensitive to slight perturbations. Either *point centric* [5] or *line centric* [5] respectively works well for either keypoints or lines segments accounting for the majority. However, our method consistently yields the smallest registration error and provides more stable solution compared with other three methods. Table 1 shows the *condition number* of the matrix for SVD decomposition in experiment Figure 7(c). The effect of a large condition number is to amplify the divergence induced by noise. The data of condition number shows consistent conclusion to the registration error.

[3]For fairness, after perturbation, the Euclidean distance to the original position is ϵ for both line segments and points.

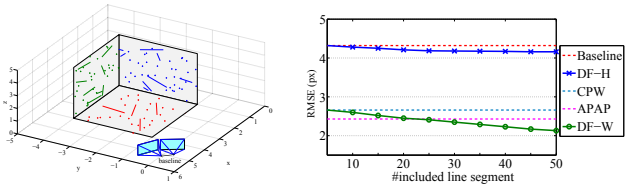


Figure 8. Illustration of translational camera motion experiment setup (left) and the measured RMSE (right) for five methods.

Translational camera motion. To further investigate the benefit of dual features in non-ideal scenes, we synthesize the scenario when the camera center is not fixed (as shown in Figure 8). The scene is a simulation of typical indoor environment with three orthogonal planes. 100 points and a varying number of line segments are randomly plotted on three planes in space, which will be projected by two virtual cameras with translational motion (0.5 units).

Five methods are compared: 1) *keypoint-based homography* (Baseline), 2) *dual-feature homography* (DF-H, Section 4.1), 3) *content-preserving-warps* [16] (CPW), 3) *as-projective-as-possible warps* [27] (APAP) , 5) *dual-feature warps* (DF-W, Section 4.1). The points are evenly splitted into two sets. The model is estimated on the first set of points plus included line segments, and the fitting error is measured on the second set of points in terms of the root mean square error (RMSE) in pixel unit. To evaluate the benefits of line feature, we vary the number of included line segments. Note that the errors for methods 1, 3, 4 are flat because they do not make use of line segments.

As can be seen in Figure 8, For single homography, the effect of extra line segments (DF-H) is trivial because the model becomes the bottleneck. For the dual-feature warp

Figure 9. The dataset of our experiments on real images.

(DF-W), including more line features helps to gradually reduce the fitting error. This is because the model is "flexible" enough to fit more features and increase the model accuracy.

Quantitative evaluation on real images. Our approach is evaluated on dataset of real images as well. Figure 9 depicts our data, which is collected from public available datasets [15, 27, 30] or captured by ourselves. Each image pair has at least 30% overlapping region but the two-view motion is not pure rotation. All images are splitted into two categories: *Category A* (low-texture images, mainly from indoor) and *Category B* (ordinary images with texture).

The same five methods are compared. Here, the accuracy is measured by the RMSE of one minus *normalized cross correlation* (NCC) over a neighborhood of 3×3 window, *i.e.*, $\text{RMSE}(\mathbf{I}_i, \mathbf{I}_j) = \sqrt{\frac{1}{N}\Sigma_\pi(1 - NCC(\mathbf{x}_i, \mathbf{x}_j))^2}$. N is the number of pixels in overlapping region π and $\mathbf{x}_i, \mathbf{x}_j$ is the pixel in image $\mathbf{I}_i, \mathbf{I}_j$ respectively.

Table 2 shows RMSE of compared methods. As we can see, our dual-feature homography consistently yields better accuracy than the Baseline (keypoint-based homography). As for warping-based model, our method performs the best in *Category A* (low-texture) as the line feature play an important role in such scenes without reliable keypoints. For *Category B* (ordinary images), though the role of line feature is reduced, it still helps to improve the accuracy of alignment. APAP performs better in *road* and *bench* because these two image pairs have wider baseline. APAP tends to be more flexible to handle larger parallax.

Time cost. The proposed method is implemented in C++ on a PC with a 3.5GHz Intel Core i7 processor. For a typical pair of images (1024×800 px), it takes around 2∼4s to find the warp. The majority of time spends on feature detection and matching, and solving the sparse matrix for local warps.

model	homography model		warping-based model		
method	Baseline	DF-H	CPW	APAP	**DF-W**
four	12.78	6.12	7.42	6.92	**2.36**
door	14.47	8.31	4.89	7.37	**3.50**
shelf	8.62	3.04	6.28	8.76	**1.54**
window	9.90	6.94	7.46	5.78	**4.94**
cabinet	6.75	3.72	3.48	4.55	**2.63**
roof	4.84	4.28	5.68	7.82	**2.25**
desk	16.94	12.71	10.67	6.17	**4.89**
corner	10.02	4.34	8.67	6.84	**1.44**
park	21.73	12.61	16.87	11.07	**8.18**
car	3.08	2.77	2.65	**2.07**	2.13
bridge	11.37	7.70	8.47	7.95	**6.60**
girl	8.76	7.82	9.17	5.20	**4.81**
villa	16.23	13.38	7.58	6.72	**5.20**
road	8.17	6.38	6.48	**2.28**	4.59
rotation	2.57	2.28	1.37	1.12	**1.06**
bench	11.5	7.18	8.97	**4.01**	7.12

Table 2. The RMSE ($[0, 255]$) for five compared methods on each image pair. Baseline: keypoint-based homography; DF-H: our dual-feature homography (Section 4.1); CPW: *content-preserving-warps* [16]; APAP: as-projective-as-possible warps [27]; DF-W: our dual-feature warps (Section 4.2)

6. Qualitative Evaluation on Applications

In this section, we apply our motion model to two applications: image stitching and video stabilization.

6.1. Image stitching

The quality of image stitching depends on the accuracy of camera motion estimation between two views. Recently, warping-based motion models [7, 15, 27, 29] are proposed to partially handle parallax issue, which is intractable for global motion model (single homography). However, all of these techniques only consider keypoints as corresponding features. Here, we want to study how well they work for challenging scenes without too many reliable keypoints.

We compared four methods on several image pairs captured in typical indoor environments (shown in Figure 10): keypoint-based homography (baseline), two representative warping-based models (content-preserving-warps [16] and as-projective-as-possible warps [27]), and our dual-feature warping-based model. We use linear blending to illustrate the misaligned regions. As we can see in Figure 10, single homography obviously cannot model the parallax well and produces misalignments. CPW allows higher DoF, but it also produces ghost artifacts (highlighted areas in Figure 10) at structural lines due to the lack of keypoints along these lines. APAP is more aggressive and easily causes local distortions on structural regions. Our method yields the best results with least ghost artifacts and preserves line structures. We show all the dual features for better understanding the role of line segments in these scenes.

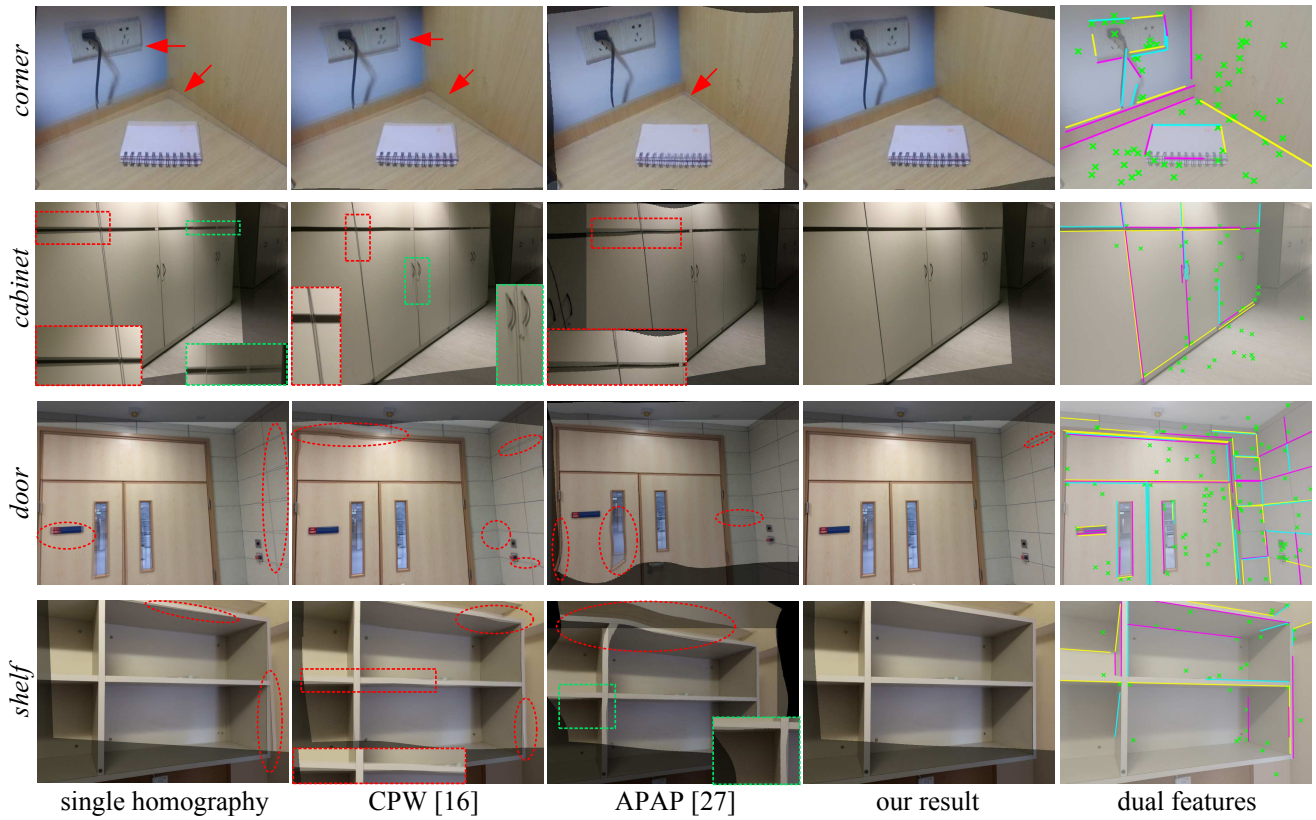
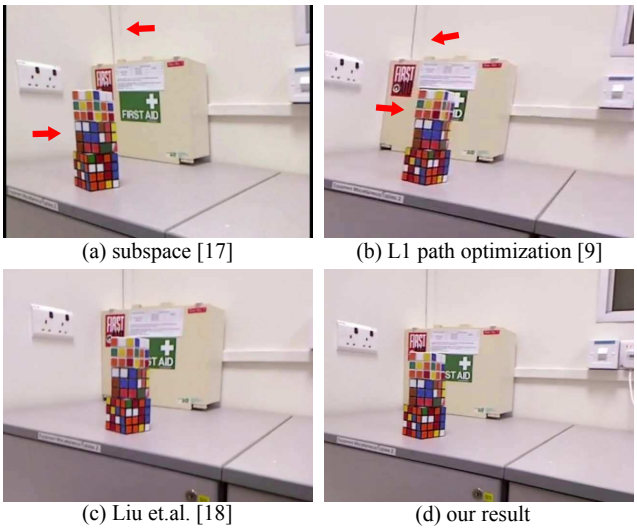


Figure 10. Comparison of image stitching on four image pairs from typical indoor scenes (best viewed on high-resolution digital display).



(a) subspace [17] (b) L1 path optimization [9]

(c) Liu et.al. [18] (d) our result

Figure 11. Comparison of different video stabilization methods on sample frames (highlights indicates shearing/skew artifacts).

6.2. Video stabilization

The first step of video stabilization is to estimate camera motion between adjacent frames. Similar to image stitching, when the video frame lacks reliable keypoints, previous methods may fail to obtain an accurate estimation, which would result in distortion or skew artifacts. However, Our dual-feature approach can better address this problem.

Figure 11 shows a challenging case (from [18]) of video stabilization, which lacks rich textures and has parallax. Two popular 2D stabilization techniques: *Subspace* [17] (with robust implementation in Adobe After Effects CS6) and L_1 *path optimization* [9] (with robust implementation in Google YouTube) fail to estimate the accurate motion. We can see shearing/skewing artifacts in sampled frames shown in Figure 11(a)(b). Our *dual-features warps* can achieve as good result as [18], which employs additional 3D depth information, but our approach is 2D motion estimation. The reason is that warping-based model, to some extent, is a good 2D solution to slight parallax, and line features further help such high DoF model achieve a robust estimation.

7. Conclusion

The paper presents a warping-based motion estimation using point and line features. We address fragile problems in existing keypoint-based model estimation methods, and suggest a practical solution for challenging scenes with less keypoint correspondences. The heart of the method is the stable combination of two types of features, making it robust for practical purposes. However, due to the intrinsic limitation of 2D model, our method cannot handle scenarios with large parallax or sudden depth variation. As the line feature gradually becomes mature, in the future we would like to study the dual-feature strategy for other problems.

Acknowledgment. This work has been partially supported by RGC-GRF 16208614 and ITC-PSKL12EG02.

References

[1] C. Akinlar and C. Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters.*, 32(13):1633–1642, 2011. 2, 3

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007. 1

[3] C.-H. Chang and Y.-Y. Chuang. A line-structure-preserving approach to image resizing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1075–1082. IEEE, 2012. 1, 2

[4] C.-H. Chang, Y. Sato, and Y.-Y. Chuang. Shape-preserving half-projective warps for image stitching. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3254–3261. IEEE, 2014. 1

[5] E. Dubrofsky. *Homography estimation*. PhD thesis, UNIVERSITY OF BRITISH COLUMBIA (Vancouver, 2009. 2, 5, 6

[6] B. Fan, F. Wu, and Z. Hu. Line matching leveraged by point correspondences. In *Proc. CVPR*, pages 390–397. IEEE, 2010. 2

[7] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 49–56. IEEE, 2011. 2, 7

[8] M. Grundmann, V. Kwatra, D. Castro, and I. Essa. Calibration-free rolling shutter removal. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012. 2

[9] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 225–232. IEEE, 2011. 1, 8

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2, 4

[11] R. I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997. 4, 6

[12] K. He, H. Chang, and J. Sun. Rectangling panoramic images via warping. *ACM Transactions on Graphics (TOG)*, 32(4):79, 2013. 2

[13] B. K. Horn and B. G. Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981. 1

[14] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141, 2005. 1

[15] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 345–352. IEEE, 2011. 1, 2, 7

[16] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. In *ACM Transactions on Graphics (TOG)*, volume 28, page 44. ACM, 2009. 1, 2, 3, 5, 6, 7

[17] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4, 2011. 8

[18] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun. Video stabilization with a depth camera. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 89–95. IEEE, 2012. 8

[19] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78, 2013. 1, 2, 3, 5

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[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.*, 28:1150–1163, 2006. 1, 3

[22] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006. 1

[23] C. J. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(11):1021–1032, 1995. 2, 3

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 3

[25] 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. 2

[26] Z. Wang, F. Wu, and Z. Hu. Msld: A robust descriptor for line matching. *Pattern Recognition Letters.*, 42(5):941–953, 2009. 2, 3

[27] J. Zaragoza, T.-J. Chin, Q.-H. Tran, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving dlt. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1285–1298, 2014. 1, 2, 5, 6, 7

[28] H. Zeng, X. Deng, and Z. Hu. A new normalized method on line-based homography estimation. *Pattern Recognition Letters.*, 29(9):1236–1244, 2008. 2, 3

[29] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3262–3269. IEEE, 2014. 1, 2, 7

[30] L. Zhang and R. Koch. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013. 2, 3, 7

[31] Z. Zhou, H. Jin, and Y. Ma. Plane-based content preserving warps for video stabilization. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2299–2306. IEEE, 2013. 2