Constructing Image Panoramas using Dual-Homography Warping

Junhong Gao   Seon Joo Kim   Michael S. Brown

School of Computing; National University of Singapore

junhong | kimsj | brown@comp.nus.edu.sg

Abstract

This paper describes a method to construct seamless image mosaics of a panoramic scene containing two predominate planes: a distant back plane and a ground plane that sweeps out from the camera's location. While this type of panorama can be stitched when the camera is carefully rotated about its optical center, such ideal scene capture is hard to perform correctly. Existing techniques use a single homography per image to perform alignment followed by seam cutting or image blending to hide inevitable alignments artifacts. In this paper, we demonstrate how to use two homographies per image to produce a more seamless image. Specifically, our approach blends the homographies in the alignment procedure to perform a nonlinear warping. Once the images are geometrically stitched, they are further processed to blend seams and to reduce curvilinear visual artifacts due to the nonlinear warping. As demonstrated in our paper, our procedure is able to produce results for this type of scene where current state-of-the-art techniques fail.

1. Introduction

Although image mosaicing has received a great deal of attention over the years, it remains a challenging problem. The most troublesome issue is that the assumptions about the input, e.g. images are of a distant scene or images are captured by rotating the camera about its center of projection, are rarely met. This results in misalignments in the stitched images. These misalignments are hidden via post-processing techniques, such as image blending or seam cutting, to varying degrees of success, often producing mosaics that appear seamless at a quick glance, only to reveal breaks and tears on closer inspection.

We focus on one common type of scene that is particularly troublesome for existing methods: a scene containing both a distant plane and a ground plane that sweeps out from the camera's location. This type of scene, as illustrated in Figure 1, is common at tourist locations where panoramic imaging is desired. Existing approaches rely on estimat-

Figure 1. A scene containing two dominant planes targeted by our mosaicing approach.

ing a *single* planar perspective transform (homography) per image to align the scene. However, a single homography cannot align the image content when the input images violate the imaging assumptions. The only option now is to attempt to hide the misalignments via post-processing.

In this paper, we describe how to address this type of panoramic scene *before* the post-processing stage by focusing on how to improve the image alignment. This is achieved by estimating two homographies per image-pair and blending these homographies to align the images. As demonstrated by our results, this approach can produce more seamless results than those obtained using current state-of-the-art methods that rely on a single homography per image pair. While a seemingly straight-forward idea, given that our scene has two dominate planes, there are several issues addressed in our paper that must be considered to make this approach work. These issues include estimating the dual homographies from matched points, weights for blending the homographies, extending the nonlinear warping to concatenate adjacent images, and post-processing to blend the images and to reduce undesired curvilinear artifacts introduced by the nonlinear warping.

The remainder of this paper is organized as follows: Section 2 discusses related work; Section 3 describes our dual-homography estimation and blending for an input image pair and its extension for multiple image concatenation; Section 4 describes post-processing procedures to re-
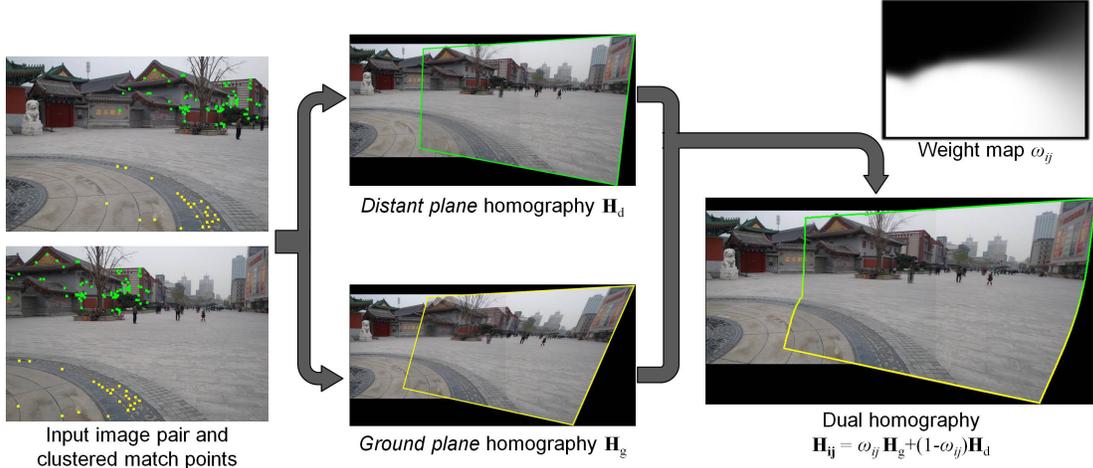
Figure 2. Work flow of our dual-homography computation. Candidate points are first clustered into groups from which two homographies are estimated using RANSAC. A per-pixel weight map ω_{ij} is computed to control the blending of the two homographies.

duce visual seams and rectify curvilinear artifacts; Section 5 demonstrates results obtained using our approach followed by a discussion and summary of our work in Section 6.

2. Related Work

Image mosaicing is a well-studied area in computer vision (representative works include [18, 19, 6, 7, 9, 23]); for an excellent survey see [22]. In addition, several freeware and commercial softwares are available for performing image stitching, notably: AutoStitch [1], Microsoft's Image Compositing Editor [2], and Adobe's Photoshop CS5 [3] mosaicing feature.

These approaches all work under the assumption that the input images contains little or no parallax, which implies that the scene is either sufficiently far away from the camera to be considered planar, or that the images have been taken from a camera carefully rotated about its center of projection. Under this assumption the images can be aligned via perspective planar (i.e. homography) transformations [11]. These existing techniques work exceedingly well under these conditions, but quickly begin to exhibit misalignment artifacts when the imaging assumptions are violated. When this occurs the algorithms attempt to hide the misalignments using image blending (e.g. [7, 12]) or seam-cutting [4].

The idea of fixing misalignments (sometimes called deghosting [21]) due to camera model violations, propagation errors, or parallax, is not new. Various approaches use local alignment matching [21] together with scatter-point-interpolation [8], or other nonlinear warping methods [13] to correct problems in the overlapped regions. These approaches, however, assume that the input images can be reasonably aligned using an initial global alignment and that the misalignments are relatively small. Methods that have

more flexible imaging models include ReliefMosaic [15] that uses dense matches to perform view-morphing to produce a 2.5D light-field that can be rendered to a mosaic. This approach requires an image scene with sufficient texture for estimating the quasi-dense image disparity. Work using manifold mosaicing [20] also allows a general imaging framework but requires dense input to select image strips in the fashion of a strip camera. Our work, with its more constrained scene type, lies somewhere between traditional mosaicing with strong imaging assumptions, and these general approaches that require either dense matching or dense video input. As far as we are aware, the idea of the dual-homography has not been used before.

3. Dual-Homography Alignment

We first describe how to compute the two homographies for a pair of overlapping images, and how to use the dual-homographies to perform nonlinear alignment. We then describe how to extend this warping to adjacent images in the mosaic. For the sake of simplicity, we assume that each image in our panorama has at most two overlapping regions with images on either of its borders.

3.1. Dual-Homography Estimation

Our approach models the relationship between two overlapping images in our panorama using a blending of two homographes, $\mathbf{H}_g$ and $\mathbf{H}_d$, expressed as:

$$\mathbf{H}_{ij} = \omega_{ij}\mathbf{H}_g + (1 - \omega_{ij})\mathbf{H}_d, \qquad (1)$$

where $\mathbf{H}_g$ and $\mathbf{H}_d$ represents the ground plane and distant plane homographies respectively, and ω_{ij} is a per pixel weight that controls the contribution of each homography at the pixel location (i, j).

As with other mosaicing techniques, our approach begins with a set of corresponding points matched between the input images. We use SIFT features [17] to establish correspondences as done in [7]. Since we are computing two homographies per image pair, our first step is to cluster the correspondences into two groups $\mathbf{G}_g$ and $\mathbf{G}_d$ based on the spatial location in the image. This is done using K-means clustering, with 2D seed points taken to be:

$$\mathbf{c}_g = (\frac{\sum_i^n x_i}{n}, 0)^T, \mathbf{c}_d = (\frac{\sum_i^n x_i}{n}, h)^T, \qquad (2)$$

where h is the height of input image. These seed points are selected to ensure that the final clusters are oriented about the top and bottom of the image, since we expect the distant plane and ground plane will most likely reside at the top and bottom of the image respectively. We note that this initial step does not need to be too accurate as outliers will be discarded in the homography estimation procedure.

After the feature points are grouped, Random Sample Consensus (RANSAC) [10] is used to robustly estimate the homography for each group. In each trial, we select four matched feature points to compute the homography. If the consensus reaches 95%, the estimated homography $\mathbf{H}_g$ and $\mathbf{H}_d$ is considered as the transform of its corresponding group of features. After RANSAC is performed, outlier SIFT features are removed to form new groups $\mathbf{G}_g'$ and $\mathbf{G}_d'$.

The two resulting homographies can partially align the image as shown in Figure 2. The goal now is to assign a weight to each pixel location $\mathbf{p}_{ij}$ to determine how much of each homography should be used. A natural choice is to weight the warping based on spatial proximity, i.e.:

$$\omega_{ij} = d_g/(d_g + d_d), \qquad (3)$$

where d_g and d_d are the $|| \cdot ||_2^{-1}$ distances to the closest feature points in the sets $\mathbf{G}_g'$ and $\mathbf{G}_d'$ respectively. Figure 2 shows an example of this weighting map and results from this weighted dual-homography warping.

3.2. Extending to Multiple Images

Unlike single homography approaches that can easily concatenate multiple homographies into a single matrix, our nonlinear warping requires more care. Figure 3 is used to help illustrate the concatenation procedure using three overlapping images I_0, I_1, and I_2 that we wish to map to the virtual mosaic image plane $I_\mathcal{V}$. I_0 is used as the root image of the mosaic and undergoes no warping.

The first dual-homography used to map I_1 to I_0 is denoted as $\mathbf{H}^{1\to0}$, dropping the subscripts, ij, for clarity. This dual-homography is computed as previously described and is the only mapping required to place I_1 in the virtual image: i.e. $\mathbf{H}^{1\to\mathcal{V}} = \mathbf{H}^{1\to0}$.

The task now is to add another dual-homography pair, $\mathbf{H}^{2\to\mathcal{V}}$, that maps I_2 to I_1 in the virtual image plane. This is
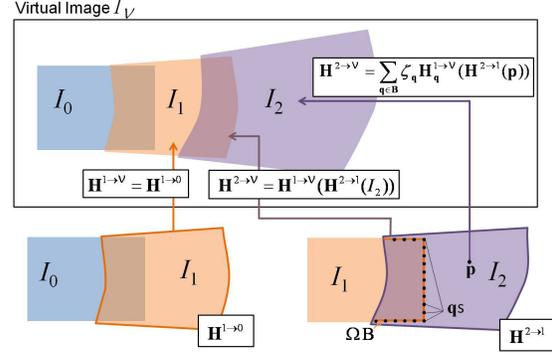

Figure 3. This figure illustrates how to concatenate multiple images into the virtual image plane, $I_\mathcal{V}$ using dual-homographies. Refer to the text in Section 3.2 for details.

achieved by first computing the dual-homography, $\mathbf{H}^{2\to1}$, to aligned I_2 to I_1. This is computed as described in the prior section. Note that this estimation is done agnostic of the relationship between I_1 and I_0. However, in the virtual image, $I_\mathcal{V}$, image I_1 is expressed as $\mathbf{H}^{1\to0}(I_1)$. Thus, in the *overlapped region* between I_2 and I_1, we can relate I_2 to the virtual image by $\mathbf{H}^{1\to0}(\mathbf{H}^{2\to1}(I_2))$.

The question now is what do we do about the non-overlapped region of I_2 with I_1. Unlike single homography approaches where the mapping between homography can be applied to points outside the overlapping region, our warping function $\mathbf{H}^{1\to0}$ is not defined for points outside I_1. To 'virtually extend' the reach of $\mathbf{H}^{1\to0}$, we use a weighted blending of the boundary points of $\mathbf{H}^{1\to\mathcal{V}}$, resulting in:

$$\mathbf{H}^{2\to\mathcal{V}}(\mathbf{p}) = \begin{cases} \mathbf{H}^{1\to\mathcal{V}}(\mathbf{H}^{2\to1}(\mathbf{p})) & , if \ \mathbf{p} \in I_1 \cap \mathbf{H}^{2\to1}(I_2) \\ \sum_{\mathbf{q}\in\Omega\mathbf{B}} \zeta_\mathbf{q}\mathbf{H}_\mathbf{q}^{1\to\mathcal{V}}(\mathbf{H}^{2\to1}(\mathbf{p})) & , otherwise, \end{cases}$$
$$(4)$$

where $\Omega\mathbf{B}$ is the boundary of I_1 that overlaps with I_2. The scalar $\zeta_\mathbf{q} = 1/\|\mathbf{H}^{2\to1}(\mathbf{p}) - \mathbf{q}\|^2$ is computed using the Euclidean distance from point $\mathbf{p}$ to all the boundary points $\mathbf{q}$ on $\Omega\mathbf{B}$. Note that these weights are normalized such that $\sum \zeta_\mathbf{q} = 1$.

While this procedure has been illustrated using only three images, it is straight-forward to derive a recursive expression from Eq. 4 to concatenate further images.

4. Post-Processing the Mosaic

This section describes two post-processing techniques, seam blending and straightening, that are used to enhance the aesthetic appearance of our final results. Seam-blending aims to reduce the visible seams between overlapping images. Content-aware straightening is unique to our mosaicing procedure and is applied to reduce curvilinear effects introduced by the dual-homography warping.
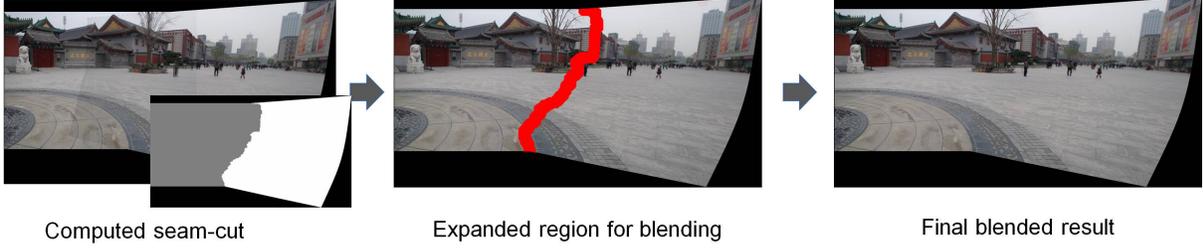
| Computed seam-cut | Expanded region for blending | Final blended result |

Figure 4. This example shows our blending procedure where a seam-cut is first computed, followed by blending about the seam.

4.1. Seam-Blending

There are two main approaches for reducing visible seams in overlapped regions. The first is to blend the entire overlapped region, such techniques include feathering techniques [22], multi-band blending [7] and gradient domain stitching [14]. The second is to perform image cutting between the images [4]. We found that a combination of both produced the best results.

Seam Cutting To compute an optimal seam between two images, for each pixel p in the final panorama result, its intensity should be mapped from one of the warped source images. This segmentation problem is formulated as a binary labeling Markov Random Field (MRF) where each pixel p is assigned a label $l \in \{0, 1\}$ (see [16] for details on MRFs). To solve the MRF, the following energy terms are minimized in order to find the optimal pixel labels:

$$E = E_d + \lambda E_s, \quad (5)$$

where E_d is the data-cost energy reflecting the likelihood of assigning a l to each pixel and E_s is the smoothness energy representing the cost of assigning different labels to adjacent pixels. We fixed the weight λ as 2.

Following the formulation in [4] the data-cost is defined to be the gradient of a pixel at that location:

$$E_d(p, l_p) = -\nabla I_{(p)}^{l_p}, \quad (6)$$

where the binary label l_p decides which gradient between the two overlapped images to use. The smoothness cost between two pixels p and q is defined as:

$$E_s(l_p, l_q) = (\|I_{(p)}^{l_p} - I_{(p)}^{l_q}\|^2 + \|I_{(q)}^{l_p} - I_{(q)}^{l_q}\|^2) + \beta(\|\nabla I_{(p)}^{l_p} - \nabla I_{(p)}^{l_q}\|^2 + \|\nabla I_{(q)}^{l_p} - \nabla I_{(q)}^{l_q}\|^2) \quad (7)$$

which represents discontinuities between each pair of neighboring pixels. We see that if $l_p = l_q$, the smoothness cost is 0, while if $l_p \neq l_q$, the smoothness cost is the intensity and gradient difference of the corresponding point in image l_p and l_q. Graph-cut optimization is used to assign the labels to our MRF [5].

Blending While seam cutting produces an image with no overlaps, color discontinuities may still be noticeable. To reduce this, we expand the seam by 16 pixels and perform a simple linear alpha blending [21] to the pixels in this expanded seam as shown in Figure 4. We found this combined approach of seam-cutting and local seam blending produced better result than either seam-cutting or blending the entire overlapped region alone.

4.2. Global Straightening

The linear interpolation of the homographies makes the images warp in a quadratic fashion. This can introduce a curved effect that may be aesthetically unappealing for the final result.

To ameliorate this effect, we employ a content-aware warping method similar to that used in [24, 25]. In particular, our final virtual image, I_V is tesselated into a polygon mesh, where each polygon, $\mathbf{Q}$, is either a quad or a triangle. Our mesh is computed by tessellating the input images into quadrilaterals and then determining how these warp into the virtual image, I_V. Overlapped regions between input images are zippered together using triangles. An example of this polygonal mesh is shown in Figure 5.

Each polygon is composed of vertices, $\mathbf{v}_i$, that can be deformed to a new set of polygons, $\mathbf{Q}'$, with the following two distortion energies:

$$D_s(\mathbf{Q}, \mathbf{Q}') = \min_s \sum_n \|s(\mathbf{v}_i) - \mathbf{v}'_i\|^2, \quad (8)$$

where $\mathbf{v}'_i$ is the target warping position of $\mathbf{v}_i$ and $s(\cdot)$ is a similar transformation function which has a general form:

$$s(\mathbf{v}) = \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix}. \quad (9)$$

This energy attempts to constrain the deformation on all the points $\mathbf{v}_i$ such that the polygons, $\mathbf{Q}$, only undergo a similarity transformation.

The other distortion energy is according to the observation in [7] that people rarely twist the camera relative to the horizon when shooting the panorama images. Hence, we give each point pair $\mathbf{v}_1$ and $\mathbf{v}_2$ in the vertical edge $\mathbf{E}$ a bending cost energy defined as follows:

$$D_l(\mathbf{E}, \mathbf{E}') = \min_l \|(\mathbf{v}'_1 - \mathbf{v}'_2) - l \begin{bmatrix} 0 \\ 1 \end{bmatrix}\|^2, \quad (10)$$
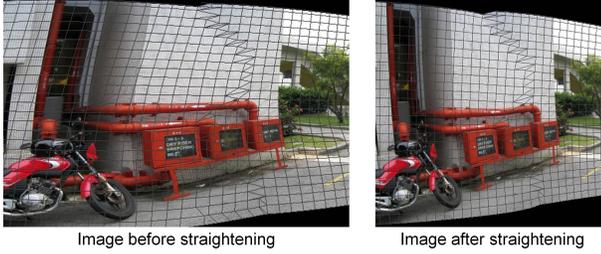
Image before straightening   Image after straightening

Figure 5. Before and after the application of content straightening.

where l is an arbitrary scaling factor. This energy constrains all the vertical edges in the source image more to maintain a vertical position in the final result. Combining these two energies we can obtain the straightened result as the points set $\mathbf{V}' = \{\mathbf{v}'_i\}$ which minimizes the following energy:

$$D = \sum_{\mathbf{Q}} \mu_s^{(\mathbf{Q})} D_s + \mu_l \sum_{\mathbf{E}} D_l. \qquad (11)$$

Here, the weight $\mu_s^{(\mathbf{Q})}$ is what allows this deformation to be flexible. This weight is determined by the underlying image content contained within each polygon, $\mathbf{Q}$, defined as:

$$\mu_s^{(\mathbf{Q})} = \frac{\sum_n \nabla I_{\mathcal{V}}(x,y)}{n}, \forall I_{\mathcal{V}}(x,y) \in \mathbf{Q}, \qquad (12)$$

where $\nabla I_{\mathcal{V}}$ is the image gradient inside a polygon. This weights polygons with higher image energy to resist deformation, while polygons with lower energy can be deformed more. To enforce the vertical edge constraint, the weight μ_l we give a comparatively large number which equals to 20 times of $\max\{\mu_s^{(\mathbf{Q})}\}$ to force the lines straight.

This approach is similar to that proposed by [24, 25] for content-aware image resizing, however, we have utilized it in a completely different manner, exploiting its ability to linearize the mesh using Equation 11. As discussed in [25], this system is directly computed using an over-determined linear solver. Figure 5 shows examples of our result before and after this straightening procedure.

While the nonlinear straightening is intended to fix artifacts introduced by the dual-homography warp, it can sometimes introduce noticeable bending on image content. As such, we have a simple UI to allow the user to specify the image region whose shape cannot be changed as shown in Figure 6. This can be achieved by increasing the $\mu_s^{(\mathbf{Q})}$ for the polygons falling within the user specified regions. Figure 6 shows an example of how user-markup can be used to protect regions in this straightening process.

5. Results

Our results are compared against three state-of-the-art softwares, AutoStitch [1], Photoshop CS5's mosaicing feature [3] and Microsoft's ICE [2]. AutoStitch [1] is based on

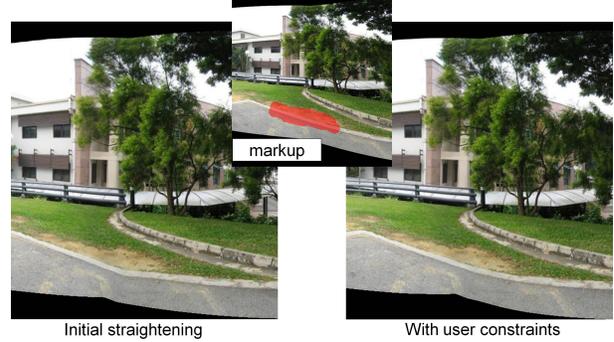Initial straightening   With user constraints

Figure 6. An example where the user can specify regions that should remain intact in the straightening process.

the work in [6], while Photoshop and ICE appear to combine elements of [6] and seam cutting [4]. All of these techniques utilize single homography alignment. Images were taken at locations that fit our targeted scene description. When imaging these scenes, we purposely moved the camera in an arc fashion to violate the common center of projection assumption for the input images.

Figure 8 and Figure 9 shows results of our algorithm on a panorama sequence containing five input images. These results are compared with AutoStitch and Photoshop CS5. While all the results appear good at first glance, we highlight a few notable regions where results from AutoStitch and Photoshop exhibit tears and misalignments. Our results, however, remain seamless in these regions. Misalignments in the results produced by Photoshop are often the hardest to detect since they use seam-cutting to stitch together images. While not shown here, ICE produced results similar to those by Photoshop on these examples, with slightly more noticeable seams. For the example in Figure 8, the user markup shown in section 4 was used.

Figure 10 shows a result comparing our approach with all three softwares. Again, at first glance each approach appears to have performed well, on close inspection we can see artifacts due to alignment issues not exhibited in our results. Note that most of the errors happen in the bottom part of the panoramic image. This implies that the single-homography based approaches are using the points in the distant plane region to compute their homographies, resulting in misalignments in the ground plane region. We also note that our lack of color balancing of the input images is sometimes noticeable when compared with the results from Photoshop CS5.

Our approach fails when the scene contains a relatively large structure that does not belong to either the ground plane nor distant plane. In such case, the parallax effect will occur since there are more than two homographies involved. Figure 7 shows an example of this failure case.

The run time of our approach is around 5~10 seconds for the homography estimation stage, 5~10 seconds for the

Figure 7. A failure case that is caused by a large structure that violates our two plane assumption.

seam cut stage and 5∼10 seconds for straightening stage (for a typical five images input). Compare to other softwares, which is around 10∼15 seconds for both AutoStich and Photoshop, our approach does need a comparatively longer time to process, mainly due to the complex warping procedure.

6. Discussion and Summary

We have demonstrated how to use dual-homography warping to align images of panoramic scenes containing a dominant distant and ground plane. While this approach requires more work when handling multiple images than single-homography approaches, the additional processing is offset by the ability to produce seamless images. Combined with standard post-processing approaches (seam cutting and blending), together with a straightening procedure, our method can produce results void of breaks and tears found in current state-of-the-art approaches.

Unlike prior work, our approach is not based on any particular camera model and therefore has no physical meaning regarding light transport into the final panorama. However, prior work is forced to hide misalignment artifacts when imaging assumptions are violated, while we instead hide errors by relaxing the imaging model. In essence, our dual-homography approach represents an effective way to align panoramic images captured under non-ideal imaging conditions.

A natural extension of our work is to consider more than two planes. This extension can quickly drive itself towards full 3D scene understanding and ultimately proxy geometry estimation in a shape from motion setting. Determining the minimum number of planes to estimate to maintain a seamless panorama in a given scene is an interesting avenue for future work.

Acknowledgement

References

[1] Autostitch, http://cvlab.epfl.ch/ brown/autostitch/autostitch.html. 2, 5

[2] Microsoft Image Composite Editor, http://research.microsoft.com/en-us/um/redmond/groups/ivm/ICE/. 2, 5

[3] Adobe Photoshop CS5, http://www.adobe.com/products/photoshop. 2, 5

[4] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, , and M. Cohen. Interactive digital photo-montage. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):294–302, 2004. 2, 4, 5

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:1222–1239, November 2001. 4

[6] M. Brown and D. Lowe. Recognising panoramas. In *ICCV*, 2003. 2, 5

[7] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007. 2, 3, 4

[8] P. Burt and E. Adelson. A multiresolution spline with applications to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983. 2

[9] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *CVPR*, 1998. 2

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981. 3

[11] R. Hartley and A. Zisserman. *Multiview Geometry in Computer Vision*. Cambridge University Press, 2004. 2

[12] J. Jia and C.-K. Tang. Image registration with global and local luminance alignment. In *ICCV*, 2003. 2

[13] J. Jia and C.-K. Tang. Image stitching using structure deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):617–631, 2008. 2

[14] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *In Proceedings of the European Conference on Computer Vision*, 2006. 4

[15] M. Lhuillier, L. Quan, H. Shum, , and H. Tsui. Relief mosaics by joint view triangulation. In *CVPR*, 2001. 2

[16] S. Li. *Markov Random Field Modeling in Image Analysis (2nd Edition)*. Springer-Verlag, 2001. 4

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004. 3

[18] D. Milgram. Computer methods for creating photomosaics. *IEEE Transactions on Computers*, C-24(11):1113 – 1119, 1975. 2

[19] D. Milgram. Adaptive techniques for photomosaicking. *IEEE Transactions on Computers*, 26(11):1175–1180, 1977. 2

[20] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *CVPR*, 1997. 2

[21] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *Internal Journal of Computer Vision*, 36(2):101–130, 2000. 2, 4
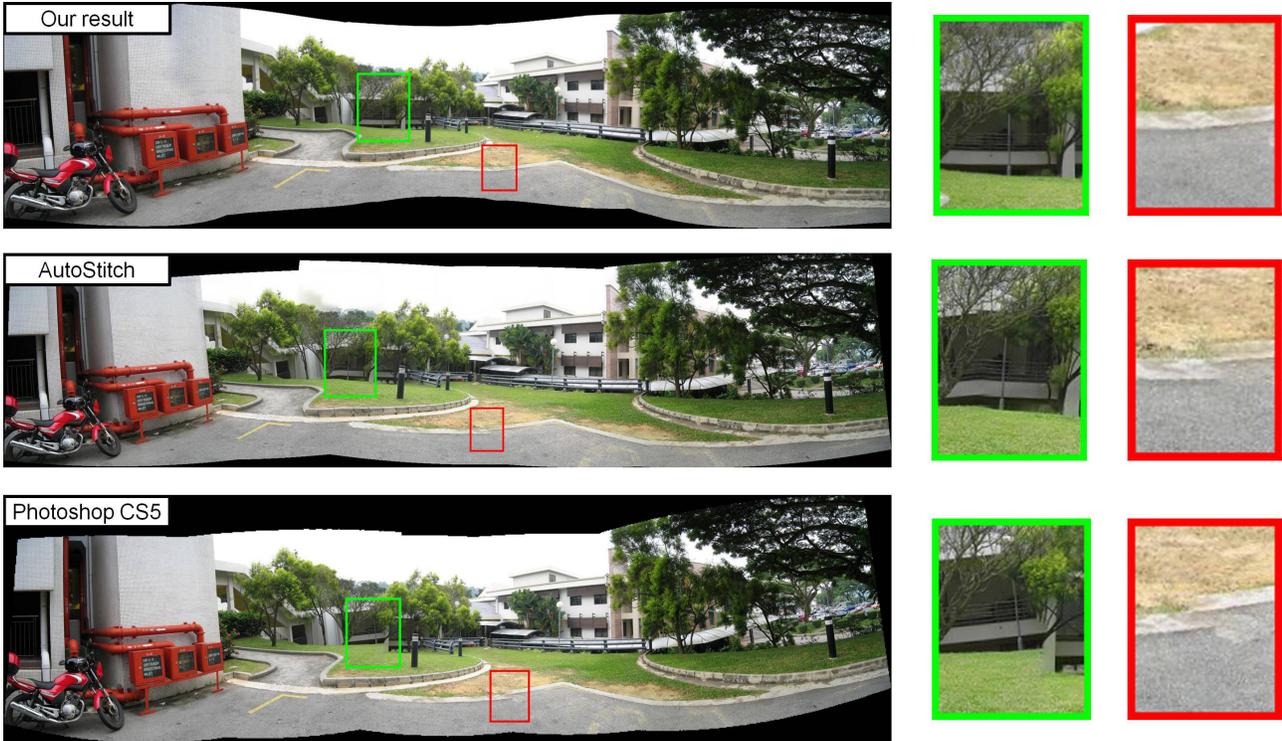
Figure 8. Example of our approach compared with AutoStitch and Photoshop CS5. Example regions with noticeable tears in the single homography-based techniques are shown in the zoomed insets.
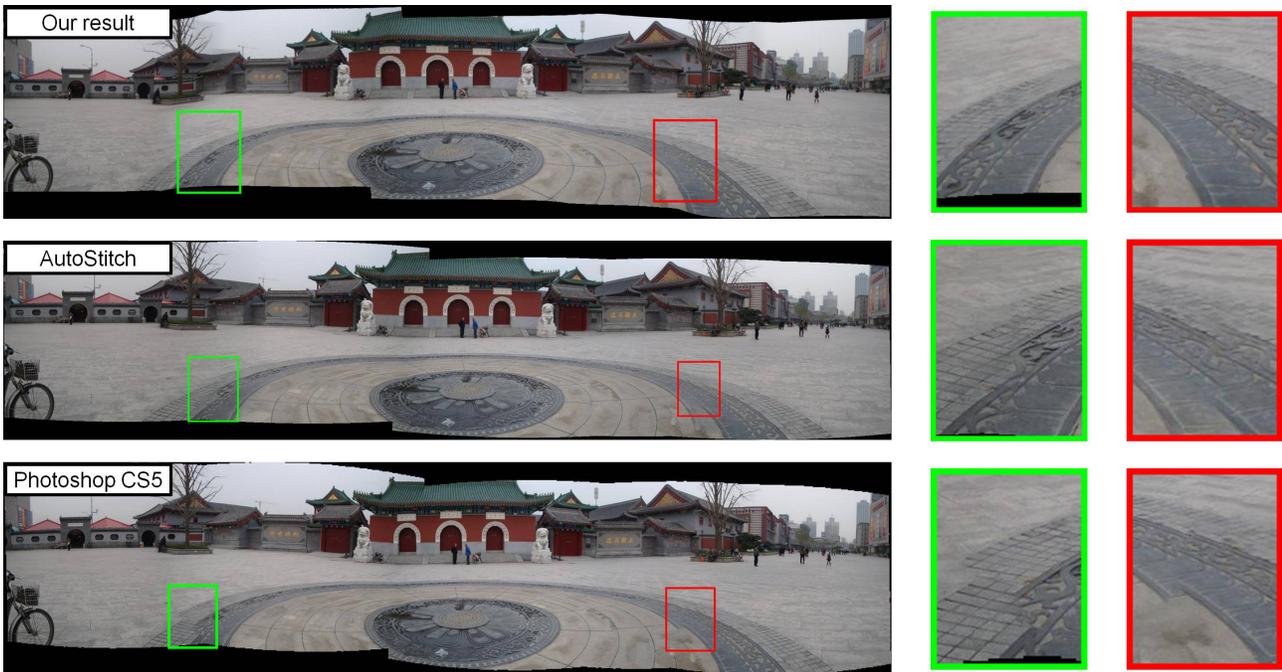

Figure 9. Another example comparing our approach with AutoStitch and Photoshop CS5. Tears in the ground plane are highlighted.
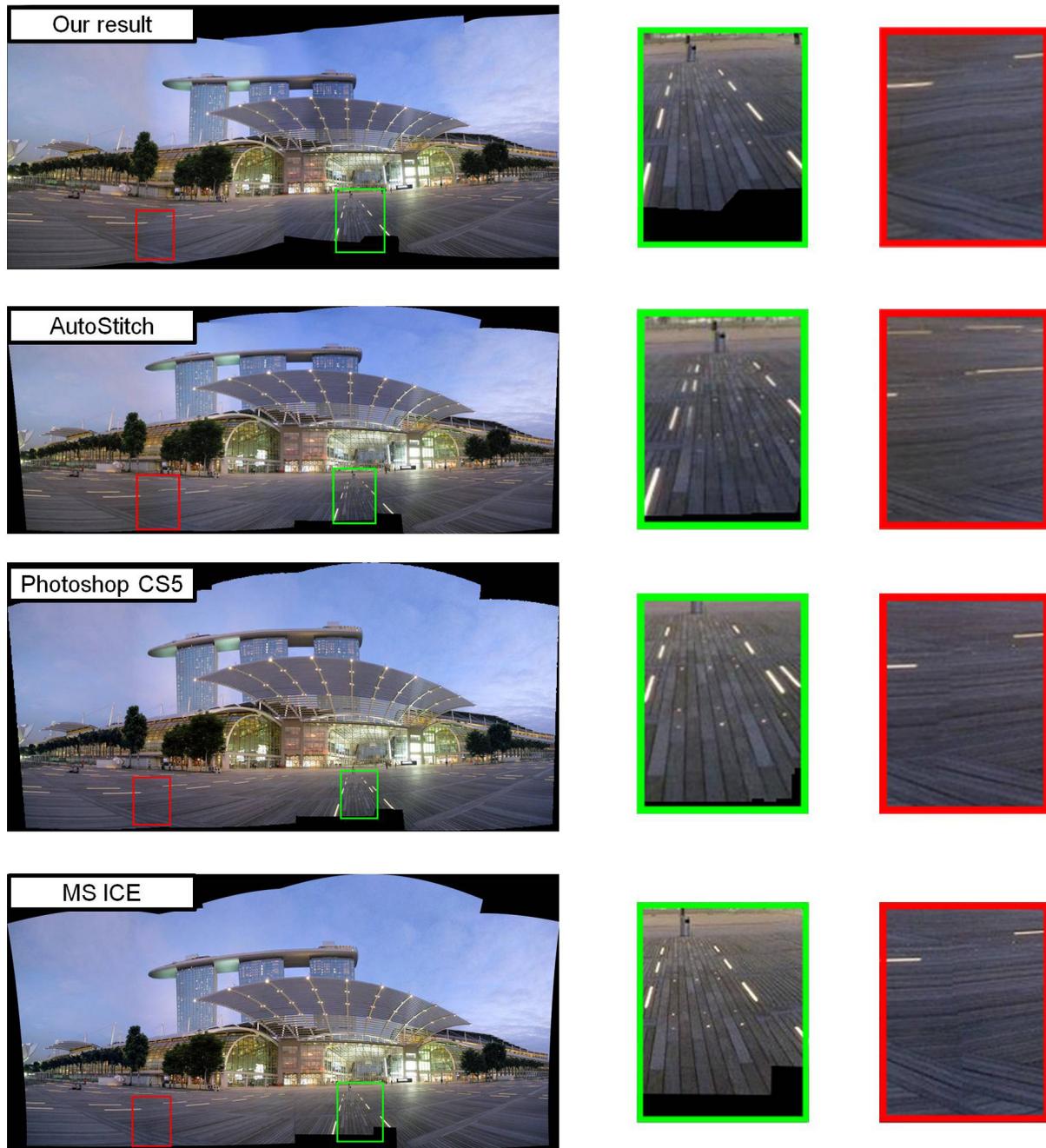
Figure 10. Example of our approach compared with AutoStitch, Photoshop, and Microsoft ICE. Regions with noticeable alignment artifacts are shown in the zoom insets. Our results, however, are able to maintain a seamless alignment over the panorama.

[22] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2:1–104, 2006. 2, 4

[23] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. In *SIGGRAPH*, 1997. 2

[24] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH ASIA*, 27(5), 2008. 4, 5

[25] G.-X. Zhang, M.-M. Cheng, S. min Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Computer Graphics Forum*, 28:1897–1906, October 2009. 4, 5