

Parallax-tolerant Image Stitching

Fan Zhang and Feng Liu
Department of Computer Science
Portland State University
{zhangfan, fliu}@cs.pdx.edu

Abstract

Parallax handling is a challenging task for image stitching. This paper presents a local stitching method to handle parallax based on the observation that input images do not need to be perfectly aligned over the whole overlapping region for stitching. Instead, they only need to be aligned in a way that there exists a local region where they can be seamlessly blended together. We adopt a hybrid alignment model that combines homography and content-preserving warping to provide flexibility for handling parallax and avoiding objectionable local distortion. We then develop an efficient randomized algorithm to search for a homography, which, combined with content-preserving warping, allows for optimal stitching. We predict how well a homography enables plausible stitching by finding a plausible seam and using the seam cost as the quality metric. We develop a seam finding method that estimates a plausible seam from only roughly aligned images by considering both geometric alignment and image content. We then pre-align input images using the optimal homography and further use content-preserving warping to locally refine the alignment. We finally compose aligned images together using a standard seam-cutting algorithm and a multi-band blending algorithm. Our experiments show that our method can effectively stitch images with large parallax that are difficult for existing methods.

1. Introduction

Image stitching is a well-studied topic [22]. Its first step is to align input images. Early methods estimate a 2D transformation, typically a homography, between two images and use it to align them [23, 3]. Since a homography cannot account for parallax, these methods require that the input images should be taken from the same viewpoint or the scene should be roughly planar. Otherwise, no homography exists that can be used to align these images, resulting in artifacts like ghosting or broken image structures. While advanced image composition techniques, such as seam cutting [2, 12] and blending [4, 17], can relieve these artifacts, they cannot address significant misalignment.

Recent image stitching methods use spatially-varying warping algorithms to align input images [13, 27]. While spatially-varying warping can better handle parallax than homography, it still cannot work well on images with large parallax. Figure 1 shows a challenging example with a significant amount of parallax in input images. Notice the horizontal spatial order of the car, the tree, and the chimney in the input images shown in Figure 1(a). In the left input image, the chimney is in the middle of the car and the tree while in the right image, the tree is in the middle of the car and the chimney. For this example, one image actually needs to be folded over in order to align with the other. This is a fundamentally difficult task for the warping methods as they either cannot fold over an image or will bring in objectionable distortion, as shown in Figure 1(c).

In this paper, we present a parallax-tolerant image stitching method. Our method is built upon an observation that aligning images perfectly over the whole overlapping area is not necessary for image stitching. Instead, we only need to align them in such a way that there exists a local region in the overlapping area where these images can be stitched together. We call this *local stitching* and develop an efficient method to find such a local alignment that allows for optimal stitching. Our local stitching method adopts a hybrid alignment model that uses both homography and content-preserving warping. Homography can preserve global image structures but cannot handle parallax. In contrast, content-preserving warping can better handle parallax than homography, but cannot preserve global image structures as well as homography. Moreover, local stitching still prefers a well aligned, large local common region. However, when homography is used to align images with large parallax, the local region size and alignment quality are often two conflicting goals. We address this problem by using homography to only roughly align images and employing content-preserving warping to refine the alignment.

We develop an efficient randomized algorithm to search for a homography for inexact local alignment first. Therein, we predict how well a homography enables local stitching by finding a plausible seam from the roughly aligned im-



(a) Input images



(b) AutoStitch



(c) APAP [27]



(d) Our result

Figure 1. Parallax problem in image stitching. All the results are partially cropped for the sake of layout. For input images with large parallax, homography-based methods, such as AutoStitch, cannot align input images and suffer from ghosting artifacts (b). Spatially-varying warping methods, such as [27], can align images but introduce apparent visual distortion (c). Our method can produce an artifacts-free stitching result (d).

ages and using the seam cost to score the homography. We develop a graph-cut based seam finding method that can estimate a plausible seam from only roughly aligned images by considering both geometric alignment and image content. Once we find the optimal homography, we use it to pre-align the input images and then use content-preserving warping to refine the alignment.

The main contribution of this paper is an efficient and robust stitching method that handles images with large par-

allax well. The power of our method comes from local stitching, which, enhanced by content-preserving warping and seam cutting, explores both image content and geometric alignment and finds an optimal local region to stitch images together. As shown in our experiments, our method can stitch images with a significant amount of parallax.

2. Related Work

Image stitching has been well studied in the fields of computer vision and graphics. A good survey can be found in [22]. This section only gives a brief overview and focuses on parallax handling.

Most existing image stitching methods estimate a 2D transformation, typically a homography, between two input images and use it to align them [23, 3]. These homography-based methods can work well only when the input images have little parallax as homography cannot account for parallax. When input images have large parallax, artifacts like ghosting occur. Local warping guided by motion estimation can be used to reduce the ghosting artifacts [21]. Image composition techniques, such as seam cutting [12, 2, 7] and blending [4, 17], have also been employed to reduce the artifacts. However, these methods alone still cannot handle significant parallax. We also use seam cutting and blending as the final steps in our method. The recent dual-homography warping method can stitch images with parallax, but it requires the scene content can be modeled by two planes [8].

Multi-perspective panorama techniques can handle parallax well [20, 28, 16, 6, 19, 1, 26, 18]. These techniques require 3D reconstruction and/or dense sampling of a scene. They are either time-consuming or cannot work well with only a sparse set of input images, as typically provided by users to make a panorama. The idea behind some of these multi-perspective panorama techniques inspired our work. That is, input images do not need to be perfectly aligned over the whole overlapping image region. As long as we can piece them together in a visually plausible way, a visually pleasing panoramic image can be created.

A relevant observation has also been made in a recent work that the best-fitting homography does not necessarily enable optimal image stitching [9]. They estimate a set of homographies, each representing a planar structure, create multiple stitching results using these homographies, and find the one with the best stitching quality. This method can successfully handle parallax for some images and also inspired our work; however, it is slow as it needs to create and score multiple stitching results. More importantly, sometimes none of the homographies that represent some planar structures can enable visually plausible stitching. In contrast, our method evaluates the alignment quality without creating the stitching results and is more efficient. Also, our method integrates content-preserving warping and loosens the alignment requirement for homography estima-

tion, thus providing more alignment candidates. Moreover, our method, considering both image content and geometric alignment in searching for a local alignment, can obtain an alignment that suits image stitching better.

Recently, spatially-varying warping methods have been extended to image stitching. Lin *et al.* developed a smoothly varying affine stitching method to handle parallax [13]. Zaragoza *et al.* developed a technique to compute an as-projective-as-possible warping that aims to be globally projective while allowing local non-projective deviations to account for parallax [27]. These methods have been shown to work well on images with parallax that are difficult for homography-based methods. However, they still cannot handle images with large parallax, as shown in Figure 1. Our method also employs a variation of spatially-varying warping method, but only uses it to align input images over a local overlapping region.

3. Parallax-tolerant Image Stitching

Our method uses a common image stitching pipeline. Specifically, we first align input images, then use a seam cutting algorithm to find a seam to piece aligned images together [12], and finally employ a multi-band blending algorithm to create the final stitching result [4]. Our contribution is a novel image alignment method which can align images in such a way that allows for optimal image stitching.

Our observation is that we do not need to perfectly align images over their whole overlapping area. In fact, for images with large parallax, it is very difficult, if not impossible, to align them perfectly. Our goal is to align images in a local region where we can find a seam to piece them together. We employ a randomized algorithm to search for a good alignment. Specifically, we first detect SIFT feature points and match them between two images [15]. We then randomly select a seed feature point and group its neighboring feature points to estimate an alignment as our goal is to estimate an alignment that aligns images over a local region with a compact feature distribution. We evaluate the stitching quality of this alignment. If this alignment is determined good enough for stitching, we stop; otherwise we repeat the alignment estimation and quality evaluation. Below we first discuss some key components of this algorithm and then provide a detailed algorithm description.

3.1. Alignment Model Selection

The first question is what alignment model to use. There are two popular options: global 2D transformation, typically homography, and spatially-varying warping, such as content-preserving warping [14, 24]. Most existing methods use a global 2D transformation to align two images. A global 2D transformation has an important advantage in that it warps an image globally and avoids some objectionable local distortions. For example, homography can preserve

lines and similarity transformation can preserve the object shape. But they are too rigid to handle parallax. For image stitching, while we argue that it is not necessary to align images exactly in their whole overlapping area, it is still preferable to align images well over an as large as possible common region. However, for images with large parallax, a 2D transformation, even a homography, can often only align images over a small local region. In contrast, content-preserving warping is more flexible and can better align images, but it often introduces objectionable local distortion.

Our solution is to combine these two alignment models to align images well over a large common region with minimal distortion. Given a seed feature point, our method incrementally groups its neighboring feature points to fit a 2D transformation (a homography by default). Here we use a slightly large fitness threshold in order to group as many feature points as possible although this makes the homography unable to fit these feature correspondences exactly. Loosing the fitness of the homography can be compensated by applying content-preserving warping later on, as content-preserving warping is well suited to local warping refinement without introducing noticeable distortion.

3.2. Alignment Quality Assessment

A straightforward way to evaluate the stitching quality of the above mentioned hybrid alignment is to first warp an image using the homography and apply content-preserving warping. We can then compare the warped image and the reference image to examine how well these two images are aligned. This approach, however, cannot reliably predict whether a good seam can be found in the overlapping region. Furthermore, this approach does not consider the effect of image content on stitching. For stitching, salient image features, such as edges, should be well aligned while image regions like the sky do not necessarily need to be perfectly aligned. Finally, this approach is slow as it needs to run content-preserving warping whenever we evaluate the alignment quality inside the randomized algorithm.

We address the above problems as follows. First, we examine the alignment quality based on the image edges instead of the raw image directly. Second, we only evaluate how the homography supports stitching. This simplification can be justified by the fact that content-aware warping is very effective if only minor adjustment to the global warping is required. But it also brings in a challenge: the homography in our method is designed to be loose and does not align two images exactly. Then we need to predict how well the alignment enables seamlessly stitching from only roughly aligned images. We address this challenge by finding a plausible seam from the roughly aligned images and using the seam cost to score the alignment.

We first down-sample the input images to both improve speed and tolerate the small misalignment. We then com-

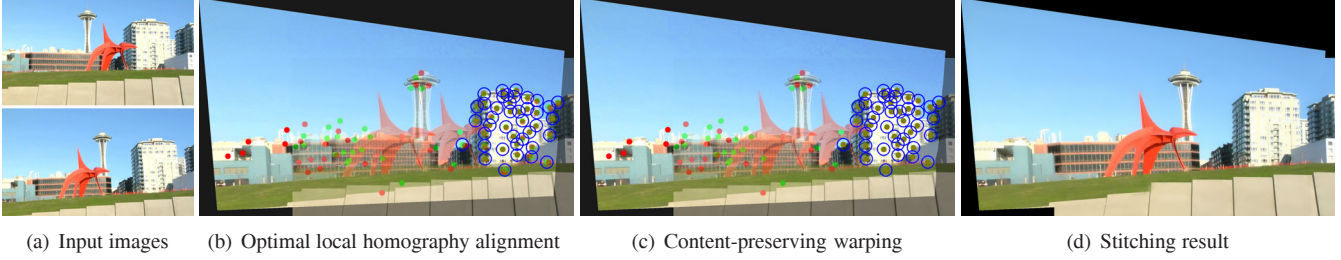


Figure 2. Stitching pipeline. Please zoom in this figure to better examine the alignment results at (b) and (c). Given input images with large parallax (a), our method first estimates an optimal homography that roughly aligns images locally (b) and is predicted to allow for optimal stitching as described in Section 3.2. In (b) and (c), we only blend aligned images by intensity averaging to illustrate alignment. The red and green points are the SIFT feature points in the warped image and the reference image, respectively. When two feature points are aligned, they appear **olive green**. Only a subset of feature points, indicated by blue circles, are selected to fit a homography loosely. Our method then locally refines alignment using content-preserving warping (c), and finally employs seam-cutting and multi-band blending to create the final stitching result (d).

pute the edge maps for the input images using the Canny edge detection method [5]. The edge maps are low-pass filtered to tolerate the small misalignment. We compute the difference between the warped edge map and the reference image’s edge map and obtain the difference map E_d . A plausible seam should avoid passing pixels with large values in the difference map in order to obtain a seamless stitching result. We extend the graph-cut seam finding method [12] to find a plausible seam. Briefly, we consider each pixel in the overlapping region as a graph node. We define the edge cost between two neighboring nodes s and t as follows,

$$e(s, t) = f_c(s)|E_d(s)| + f_c(t)|E_d(t)| \quad (1)$$

where we use an alignment confidence function $f_c(s)$ to weight the edge cost. $f_c(s)$ is computed to further account for the fact that the homography can only align two images roughly and content-preserving warping will be used to refine the alignment. Specifically, if a local region has a SIFT feature point, the alignment there can very likely be improved by content-preserving warping and thus the misalignment from only using the homography should be deemphasized. We compute $f_c(s)$ to deemphasize the misalignment according to the SIFT distribution as follows,

$$f_c(s) = \frac{1}{\sum_{P_i} g(\|P_s - P_i\|) + \delta} \quad (2)$$

where P_i is the position of a SIFT feature point and P_s is the position of pixel s . g is a Gaussian function and is used to propagate the effect of a SIFT feature to its neighborhood. δ is a small constant with a default value 0.01.

Based on the edge cost defined in Equation 1, the seam finding problem can be formulated and solved as a graph-cut problem [12]. Once we obtain this seam, we use the cost associated with this seam to score the alignment quality.

3.2.1 Homography Screening

While some homographies can allow for seamless stitching, they sometimes severely distort the images and lead to

visually unpleasant stitching results. We detect such homographies and discard them before evaluating their alignment quality. We measure the perspective distortion from applying a homography H to an image I by computing how H deviates from its best-fitting similarity transformation. Denote C_i as one of the four corner points of the input image I and \bar{C}_i is the corresponding point transformed by H . We find the best-fitting similarity transformation \hat{H}_s as follows,

$$\hat{H}_s = \arg \min_{H_s} \sum_{C_i} \|H_s C_i - \bar{C}_i\|^2, \text{ where } H_s = \begin{bmatrix} a & -b & c \\ b & a & d \end{bmatrix} \quad (3)$$

Once we obtain \hat{H}_s , we sum up the distances between the corner points transformed by H and \hat{H}_s to measure the perspective distortion. If the sum of the distances normalized by the image size is larger than a threshold (with default value 0.01), we discard that homography.

3.3. Alignment Algorithm Summary

We now describe our randomized algorithm to estimate a good alignment for stitching.

1. Detect and match SIFT features between input images [15] and estimate edge maps for input images [5].
2. Randomly select a seed feature point and group its spatially nearest neighbors one by one until the selected feature set cannot be fitted by a homography with a pre-defined threshold. We maintain a penalty value for each feature point to identify the times that it has been selected during the iteration process. When a feature point is selected, we increase its penalty value by one. In each iteration, to be selected as a valid seed, a feature point should not have been selected as a seed before and its penalty score is below the average penalty value of all the feature points.
3. Evaluate the alignment quality of the best-fitting homography from Step 2 using the algorithm described in Section 3.2. If the homography meets the pre-defined

quality threshold, go to Step 4. Otherwise, if the average penalty value is low, go to Step 2; otherwise select the best homography estimated during the iteration process and go to Step 4.

4. Employ the optimal homography to pre-align images and use content-preserving warping guided by the set of selected feature points to refine the alignment, as described in Section 3.3.1.

Figure 2 shows the pipeline of our method. Given input images (a), our method first finds an optimal local homography and a subset of feature points that are loosely fit by this homography as shown in (b). We illustrate the selected feature pairs using blue circles. Notice that the homography does not align these features exactly. We then use content preserving warping to refine the alignment. As shown in (c), the selected feature pairs are now well aligned. Our method finally composes the aligned images together (d).

3.3.1 Content-preserving warping

Various content-preserving warping methods have been used in applications, such as video stabilization [14] and image and video retargeting [25, 24]. While content-preserving warping alone cannot always be used to align images over their whole overlapping area, it is well suited for small local adjustment. Therefore, we use it to further align the pre-warping result from the optimal homography to the reference image as shown in Figure 2 (b) and (c).

We use I , \bar{I} , and \hat{I} to denote the input image, the pre-warping result, and the final warping result, respectively. We divide the input image I into an $m \times n$ uniform grid mesh. The vertices in I , \bar{I} , and \hat{I} are denoted using V_i , \bar{V}_i , and \hat{V}_i . We then formulate the image warping problem as a mesh warping problem, where the unknowns are \hat{V}_i . \bar{V}_i is known from pre-warping. This mesh warping problem is defined as an optimization problem that aims to align \bar{I} to the reference image while avoiding noticeable distortions. We now describe the energy terms in detail below.

Local alignment term. The feature points in image I and \bar{I} should be moved to match their corresponding positions in the reference image so that they can be well aligned. Since a feature point P_j is not usually coincident with any mesh vertex, we find the mesh cell that contains P_j . We then represent \bar{P}_j , the corresponding point of P_j in \bar{I} , using a linear combination of the four cell vertices of the corresponding cell in image \bar{I} . The linear combination coefficients are computed using the inverse bilinear interpolation method [10]. These coefficients are used to combine the vertices in the output image \hat{I} to compute \hat{P}_j . We can then define the alignment term as follows.

$$E_p = \sum_{j=1}^n \left\| \sum_{k=1}^4 \alpha_{j,k} \hat{V}_{j,k} - \bar{P}_j \right\|^2, \quad (4)$$

where n is the size of the selected feature set from the alignment optimization step (Section 3.3), $\alpha_{j,k}$ is the bilinear combination coefficient, and $\hat{V}_{j,k}$ is a vertex of the mesh cell that contains \hat{P}_j , and \bar{P}_j is the corresponding feature point in the reference image.

Global alignment term. The alignment term above only directly constrains warping of the overlapping image region with selected feature points. For other regions, content-preserving warping often distorts them. As the pre-warping result \bar{I} has already provided a good approximation, our method encourages the regions without feature points to be close to the pre-warping result as much as possible. We therefore define the following global alignment term,

$$E_g = \sum_i \tau_i \|\hat{V}_i - \bar{V}_i\|^2, \quad (5)$$

where \hat{V}_i and \bar{V}_i are the corresponding vertex in the content-preserving warping result and in the pre-warping result. τ_i is a binary value. We set it 1 if there is no feature point in the neighborhood of V_i ; otherwise it is 0. This use of τ_i provides flexibility for local alignment.

Smoothness term. To further minimize the local distortion during warping, we encourage each mesh cell in the pre-warping result to undergo a similarity transformation. We use the quadratic energy term from [11] to encode the similarity transformation constraint. Specifically, consider a triangle $\Delta \bar{V}_1 \bar{V}_2 \bar{V}_3$. Its vertex \bar{V}_1 can be represented by the other two vertices as follows,

$$\bar{V}_1 = \bar{V}_2 + u(\bar{V}_3 - \bar{V}_2) + vR(\bar{V}_3 - \bar{V}_2), R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad (6)$$

where u and v are the coordinates of \bar{V}_1 in the local coordinate system defined by \bar{V}_2 and \bar{V}_3 . If this triangle undergoes a similarity transformation, its coordinates in the local coordinate system will not be changed. Therefore, the similarity transformation term can be defined as follows,

$$E_s(\hat{V}_i) = w_s \|\hat{V}_1 - (\hat{V}_2 + u(\hat{V}_3 - \hat{V}_2) + vR(\hat{V}_3 - \hat{V}_2))\|^2, \quad (7)$$

where u and v are computed from Equation 6. We sum $E_s(\hat{V}_i)$ over all the vertices and obtain the full smoothness energy term E_s . Here w_s measures the saliency value of the triangle $\Delta \bar{V}_1 \bar{V}_2 \bar{V}_3$ using the same method as [14]. We use this saliency weight to distribute more distortion to less salient regions than those salient ones.

Optimization. We combine the above three energy terms into the following energy minimization problem,

$$E = E_p + \alpha E_g + \beta E_s, \quad (8)$$

where α and β are the weight of each term with default values 0.01 and 0.001, respectively. The above minimization problem is quadratic and is solved using a standard sparse linear solver. Once we obtain the output mesh, we use texture mapping to render the final result.

4. Experiments

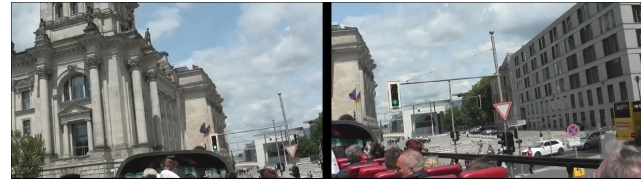
We experimented with our method on a range of challenging images with large parallax. We also compared our method to the state-of-the-art methods, including Photoshop, AutoStitch, as-projective-as-possible stitching (APAP) [27], and our implementation of seam-driven stitching (SEAM) [9]. For APAP, we used the code shared by the authors. Since that code only aligns images, we applied the same seam-cutting and multi-band blending algorithm used in our method to the APAP alignment results to produce the final stitching results. This paper only shows some representative stitching results that are partially cropped for the sake of layout. Please refer to the project website for more results that are not cropped and more intermediate results¹.

Figure 3(a) shows two input images with a significant amount of parallax. Photoshop failed to produce any result. AutoStitch could not align two images well using a global 2D transformation, therefore the stitching result suffers from ghosting, as indicated by the red circle in Figure 3(b). The traffic light is duplicated in the final result. The SEAM method did not find a local plane represented by a homography that allows for seamless stitching, and duplicated the traffic light too as shown in Figure 3(c). The APAP method creates a reasonable stitching result as shown in Figure 3(d); however, as APAP tries to align two images over the whole overlapping region, it distorts the salient image structure, such as the pillar indicated by the red rectangle. Our method can handle this challenging example by aligning the input images locally in a way that allows for optimal stitching, as shown in Figure 3(e). We also show the stitching seam in red.

Figure 4(a) shows another challenging example. The two input images have a large amount of parallax, and there is no global transformation that can align them well over the whole overlapping region. As shown in Figure 4(b), the AutoStitch result suffers from significant ghosting artifacts. While blending can relieve misalignment, it causes severe blurring artifacts as indicated by the red circle. Both Photoshop and SEAM duplicated the red structure, as shown in Figure 4(c) and (d). APAP bends the straight line as shown in Figure 4(e). Our result in (f) is free from these artifacts.

4.1. Discussion

Our method only needs to align input images locally and fit a homography loosely, as described in Section 3.2. Therefore our method can sometimes use a more restrictive global transformation than a homography to remove the perspective distortion from homography. Figure 5(a) shows a stitching result from our method using homography for initial alignment, which suffers from noticeable perspective distortion. Once we replace homography with similarity



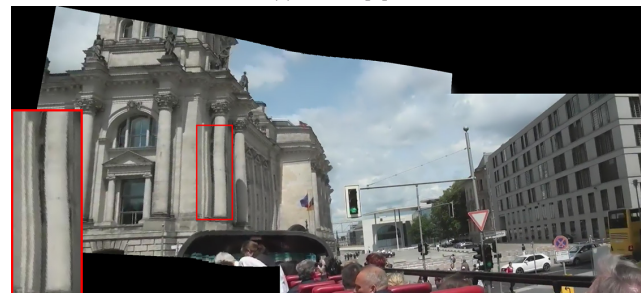
(a) Input images



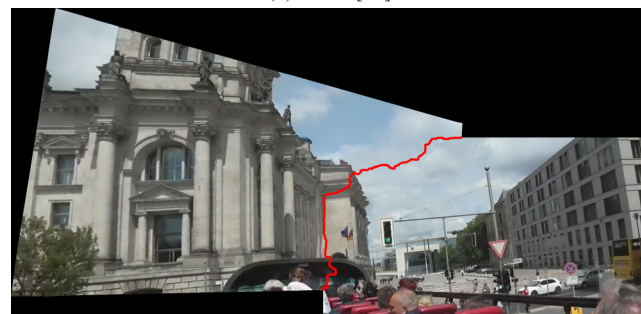
(b) AutoStitch



(c) SEAM [9]



(d) APAP [27]



(e) Our result with seam

Figure 3. Comparisons among various stitching methods. transformation for initial alignment, the stitching result suffers from less distortion, as shown in Figure 5(b).

We also tested how the homographies selected by our method differ from the best-fitting ones by computing the distances between the transformed image corner positions with our homographies and the best-fitting ones. Over 75% of the examples shared in our project website has the av-

¹<http://graphics.cs.pdx.edu/project/stitch>



Figure 4. Comparisons among various stitching methods.

erage corner position distance larger than 36 pixels (given an image with width 1000 pixels). The median distance is around 60 pixels. This confirms that our method uses different homographies than the best-fitting ones.

Our method works well on a range of examples with large parallax as well as all the examples reported in the recent APAP paper [27]. Meanwhile, we also found some failure cases as shown in the project website. One was that input images have very large parallax and are full of salient structures. For stitching, images must be aligned so that there at least exists a local common region where a good seam can be found. In images with large parallax, there is often no such a local region that can be aligned. Our method explores the fact that non-salient areas often need not be well aligned and considers this in searching for a good local region alignment. But if an image has large parallax and is full of salient structures, our method sometimes cannot work as no non-salient region exists.

Our method adopts a common image stitching pipeline. Its major novelty is in its step to align images such that op-

timal stitching can be achieved. This step, including optimal local homography estimation and content-preserving warping, typically takes from 20 to 40 seconds on a desktop machine with Intel i7 CPU and 8 GB memory to align two images with width 1000 pixels. All the other steps are shared by off-the-shelf image stitching methods.

5. Conclusion

This paper presented a parallax-tolerant image stitching method. We observed that images with significant parallax often cannot be aligned well over the whole overlapping region without suffering artifacts like folding-over and these images actually do not need to be aligned perfectly over the whole overlapping region for image stitching. We then developed a method that aligns input images locally in such a way that allows for optimal stitching. We designed an efficient algorithm to estimate how an alignment result enables seamless stitching without actual stitching. Our experiments on challenging stitching tasks showed the effectiveness of our method.



(a) Homography

(b) Similarity transformation

Figure 5. Homography vs. Similarity transformation. Our method is flexible in choosing a 2D global transformation for initial alignment. Sometimes we can replace the commonly used homography with a similarity transformation to reduce distortion.

Acknowledgements. The images in Figure 3 are used under a Creative Commons Attribution License from Youtube user *Lars Troen*. This work was supported by NSF grants IIS-1321119, CNS-1205746, and CNS-1218589.

References

- [1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.
- [2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.
- [3] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, 2007.
- [4] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.
- [5] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.
- [6] F. Dornaika and R. Chung. Mosaicking images with parallax. *Signal Processing: Image Communication*, 19(8):771–786, 2004.
- [7] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *IEEE CVPR*, pages 2498–2505, 2006.
- [8] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *IEEE CVPR*, pages 49–56, 2011.
- [9] J. Gao, Y. Li, T.-J. Chin, and M. S. Brown. Seam-driven image stitching. In *Eurographics 2013*, pages 45–48, 2013.
- [10] P. S. Heckbert. Fundamentals of texture mapping and image warping. Master’s thesis, Citeseer, 1989.
- [11] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics*, 24(3):1134–1141, 2005.
- [12] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.
- [13] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *IEEE CVPR*, pages 345–352, 2011.
- [14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3):44:1–44:9, 2009.
- [15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [16] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1144–1154, 2000.
- [17] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.
- [18] A. Rav-Acha, G. Engel, and S. Peleg. Minimal aspect distortion (mad) mosaicing of long scenes. *Int. J. Comput. Vision*, 78(2-3):187–206, 2008.
- [19] A. Roman and H. P. Lensch. Automatic multiperspective images. In *Eurographics Symposium on Rendering*, pages 83–92, 2006.
- [20] S. M. Seitz and J. Kim. Multiperspective imaging. *IEEE Computer Graphics and Applications*, 23(6):16–19, 2003.
- [21] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *IEEE ICCV*, pages 953–956, 1998.
- [22] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.
- [23] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *ACM SIGGRAPH*, pages 251–258, 1997.
- [24] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5):118:1–118:8, 2008.
- [25] L. Wolf, M. Guttman, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *IEEE ICCV*, 2007.
- [26] J. Yu and L. McMillan. A framework for multiperspective rendering. In *EGSR*, pages 61–68, 2004.
- [27] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving DLT. In *IEEE CVPR*, 2013.
- [28] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):741–754, 2003.