An Interactive Editing Tool for Correcting Panoramas

Junhong Gao and Michael S. Brown
School of Computing, National University of Singapore

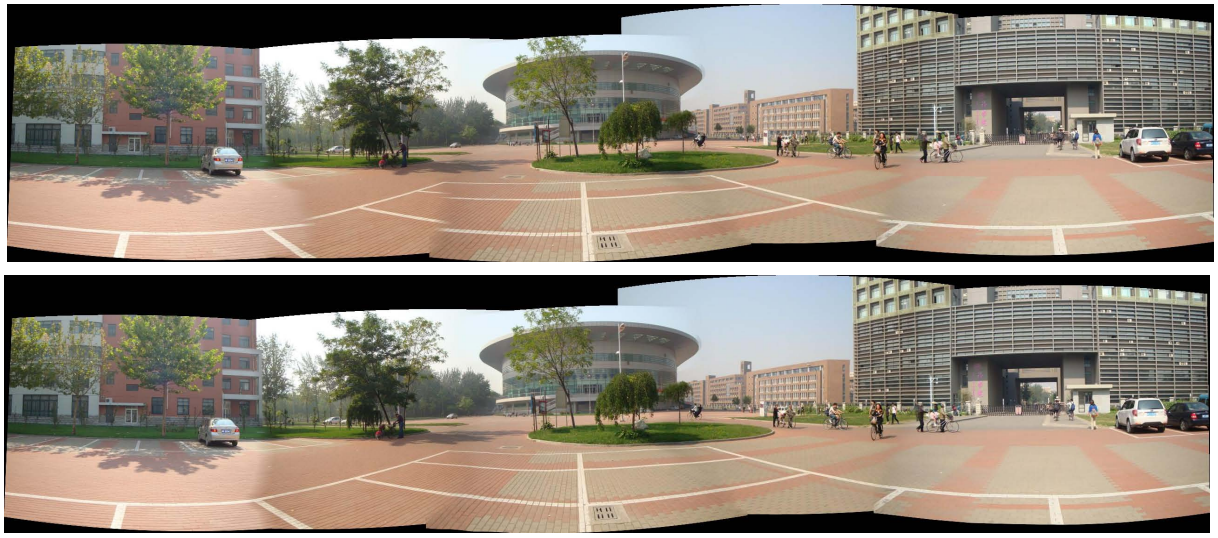

Figure 1: *Top: Example of automatically constructed image mosaic with noticeable misalignment artifacts. Bottom: Result produced using our interactive editing tool designed to post-process panoramas.*

Abstract

Constructing panoramas from an image series is a fundamental task for photo-editing. State-of-the-art photo-editing softwares generally provide automatic stitching routines that first perform image warping followed by seam-cutting or blending to hide misalignment artifacts. This latter step to hide registration errors is often crucial for achieving a perceptually seamless panorama. However, when this step fails, there are no customized photo-editing tools to allow the user to efficiently hide visual artifacts. In this paper, we introduce a simple and effective interactive tool for correcting panoramas. In particular, we describe two features: 1) a seam-editing tool that allows the user to modify blending seams in a local manner; and 2) a content-aware snapping tool to help the user better align local image content between overlapping images.

CR Categories: I.3.8 [Computer Graphics]: Application I.4.0 [Image Processing and Computer Vision]: Image processing software

Keywords: panorama correction, interactive tool

Links: DL PDF

1 Introduction and motivation

While image mosaicing has become a core photo-editing routine, it remains a challenging task. The fundamental problem lies in the conditions required for image warping to effectively align a series of images. Mosaicing techniques either require the captured images to be of a scene far enough away from the camera to be treated as planar, or require the scene to be captured by a camera that has been carefully rotated about its center of projection to avoid parallax. For consumer-level photography these two conditions are difficult to achieve. As a result, there are misalignment artifacts in the stitched panorama.

To address this issue, virtually all image stitching methods apply a post-processing step, such as image blending or seam cutting, to ameliorate misalignment artifacts. When these post-processing routines work well they can hide artifacts in a manner that produces a perceptually seamless panoramas that, on casual observation, appears visually correct. The problem addressed in this paper is what to do when this post-processing steps fails, as shown in Figure 1. Currently, for software such as Photoshop, the user is limited to standard image-processing tools to edit the seam masks or warp the individual images to achieve a better result. These routines, however, are not tailored for editing panoramic images, often making this manual correction tedious.

We propose an interactive photo-editing tool to aid panorama post-processing correction based on two features. The first is a seam-editing tool that allows the user to use markup to modify the seam in a local manner. This helps to reduce artifacts that arise due to poor initial seam estimation. Second, we provide a content-aware local image warping tool that helps the user to align overlapping image content by ”snapping” the locally warped region when the scene content matches. While our two approaches are simple, we show that these combined features make it significantly easier to post-

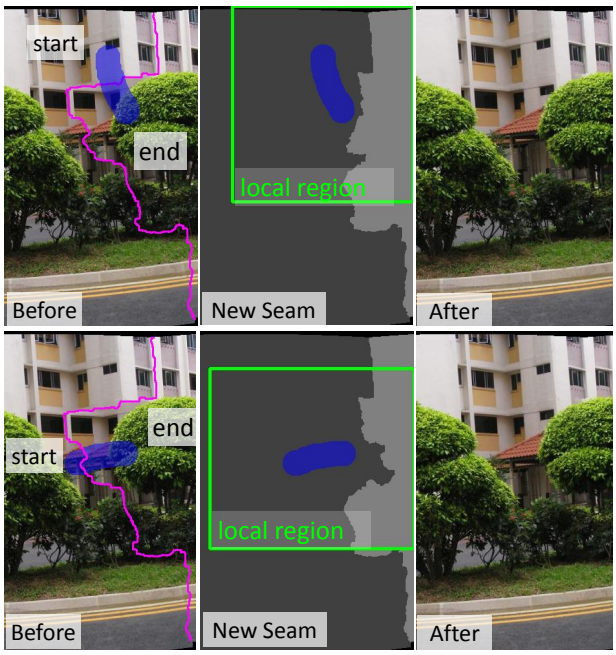

Figure 2: *Example of local seam adjustment. Figure (a) and (b) show that different markup can produce similar effects.*

process challenging panoramic than possible with current photo-editing software.

2 System Framework

Our approach follows the framework proposed by Brown et al. [Brown and Lowe 2003]. A planar transform (i.e. homography) is computed between neighboring image pairs using registered SIFT (Scale-Invariant Feature Transform) features [Lowe 2004]. Each input image is then transformed to its corresponding position using the estimated transformation and a global cylinder warping. Seams between the overlapping images are computed using the approach in [Agarwala et al. 2004]. To reduce noticeable photometric mismatches between adjacent images, we expand the seam by 16 pixels and perform a simple linear alpha blending [Szeliski 2006]. After this procedure we obtain the initial panorama.

Unique to our framework is the inclusion of an interactive editing tool that allows the user to perform local seam editing to further refine the seam-cut result; and a content-aware snapping tool to locally warp an image remove tearing artifacts. During this interactive editing process, the user can switch between these two tools until a satisfactory result is reached.

2.1 Local seam-editing

Seam computation As previously mentioned, we use seam-cutting when compositing the aligned neighboring images. The seam computation can be formulated as a Markov Random Field (MRF) segmentation problem which is described in [Boykov et al. 2001] [Li 2001]. The MRF minimizes the following global energy function:

$$E = \sum_{p} E_d + \lambda \sum_{(p,q)\in \mathbf{N}} E_s, \tag{1}$$

where E_d is the data-cost energy reflecting the saliency of a pixel, p, with label l and E_s is the smoothness energy that measure the discontinuity of adjacent pixels, p and q, with different labels.

Following the formulation in [Agarwala et al. 2004], the data-cost of each pixel is defined to be the gradient at that location:

$$E_d(p, l_p) = -\nabla I^{l_p}_{(p)}, \tag{2}$$

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

$$E_s(l_p, l_q) = (\|I^{l_p}_{(p)} - I^{l_q}_{(p)}\|^2 + \|I^{l_p}_{(q)} - I^{l_q}_{(q)}\|^2) \quad + \\ \alpha(\|\nabla I^{l_p}_{(p)} - \nabla I^{l_q}_{(p)}\|^2 + \|\nabla I^{l_p}_{(q)} - \nabla I^{l_q}_{(q)}\|^2) \quad , \tag{3}$$

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 [Boykov et al. 2001]. The weights λ and α are set to 2 in our implementation.

Local seam adjustment The operation of the local seam-editing tool is to draw strokes onto the overlapping region. All pixels that are marked by this stroke are forced to have same label with the pixel at the start point of the stroke. Therefore, a stroke operation is defined to be valid only when it overlaps with the current seam. For each valid stroke, an subregion is defined for the updates by expanding the size of the bounding box $R(w, h)$ of the stroke by $max(w, h) \times 3$. This subregion undergoes the same graph-cut segmentation process which was described in Section 2 but with pixels under the stroke's labels fixed. Figure 2 shows an example. Our interactive editing tool allows the user to toggle between adjacent overlap images to see the underlying image content to allow them to decide where to draw a markup stroke. This idea was concurrently proposed by [Summa et al. 2012] in their "panorama weaving" application which allows the user to interactively manipulate the seam via dragging control points over the seam. Our local seam-editing tool has provide an alternative scribble based interface to allow user more flexibly edit the seam. Note that the a variety of different markup can produce a similar desired result as shown in figure Figure 2.

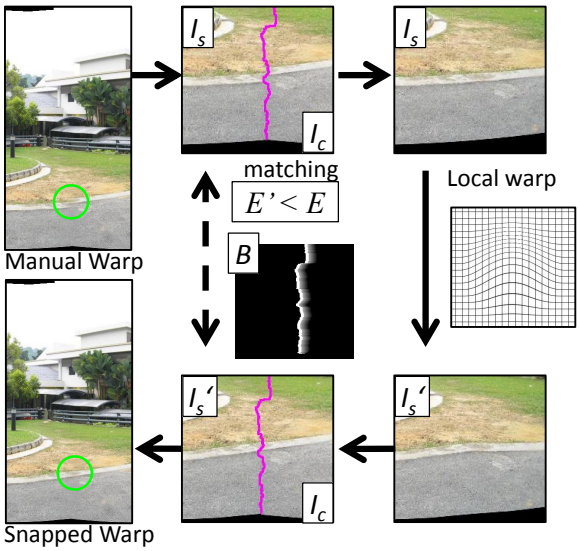

Figure 3: *This figure provides an overview of content-aware local warping. The mouse motion defines a local warp. When the matching cost computed between the warped image I_s' and overlapped image I_c reaches a local minimum, the warp cannot move until a new local minimum is found. This simulates a "snapping" effect that makes it possible to perform quick local alignment.*

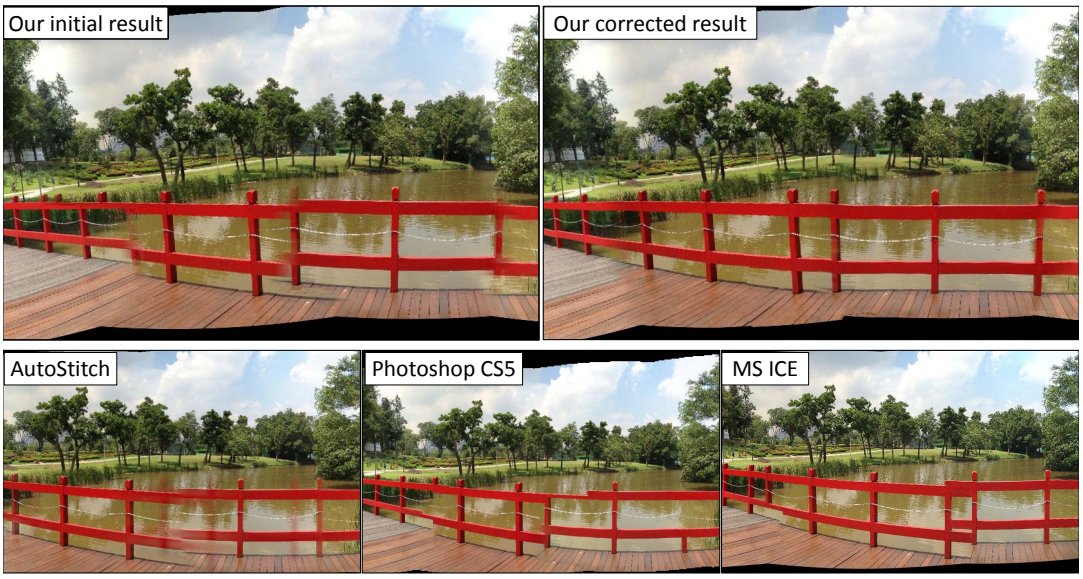

Figure 4: *An example comparing the results of Photoshop, Microsoft ICE, and Autostitch with the original and edited results generated by our editing tool.*

2.2 Content-aware snapping

Unlike the local seam-editing that only targets overlapping regions, the content-aware snapping tool allows the user to warp any region in the overall panorama. As a result, it can be used to either snap the region to the adjacent images or adjust distortions in non-overlapping areas. The interface of the warping tool is straightforward. A brush with a user-specified size is used to cover the warping region. When the left button of the mouse is pressed, the user can then drag the mouse to warp the region by moving it to a desired position.

For each mouse action, a warping function is executed to determine the displacement for each pixel $\mathbf{p}$:

$$\Delta\mathbf{p} = \beta \cdot (1 - \|\mathbf{p}_c - \mathbf{p}\|/r) \cdot (\mathbf{p}'_c - \mathbf{p}_c), \tag{4}$$

where $\mathbf{p}_c$ and $\mathbf{p}'_c$ are brush centers before and after moving respectively, r is the radius of the brush. A scalar $\beta \in [0, 1]$ controls the strength of warping. In our system, we set β as 0.7.

When the warping region overlaps with a seam, a snapping process is triggered to determine whether the warped region "snaps" to its neighboring image or not. For each motion, we crop a sub-image I_s which is the warped region of the target image under the brush and a sub-image I_c which is the complementary part in the overlapped image for the same region. The difference between I_s and I_c is then computed to represent a matching error. Since our target is to snap the content along the cutting seam to avoid tearing artifacts, a weighted map B is used to give the pixels near the seam more importance. The matching error E is defined as:

$$E = \frac{\sum(\|I_s - I_c\| + \zeta\|\nabla I_s - \nabla I_c\|) \cdot B}{\sum B}, \tag{5}$$

where $B = \{\omega \mid \omega(p) \in [0, 1],\ \omega(p) \propto^{-1}$ distant between p and the cutting seam$\}$ and ζ are set to be 2.

A minimum error E_m is set as a criterion to determine if the current warp is "snapped". Each time the current E value is updated, the display of the warp to the user only updates when $E < E_m$. This simulates a snapping effect by keeping the warp fixed a the location with matching error E_m even though the mouse is still moving. This snapping makes it easy for the user to quickly align content along the seam in the overlapping region.

3 Results and Summary

Several examples generated by our framework are shown. Figure 4 compares our result with those produced by state-of-the-art mosaicing softwares, i.e. Photoshop, Autostitch, and Microsoft ICE. For all approaches, noticeable misalignment errors are present. While the result produced by Photoshop could be further edited, software such as Autostitch and ICE provide no means to correct the results. Our initial result and edited results are demonstrated.

Figure 5 shows two additional results created by our approach. This figure shows the initial computed panorama generated by the alignment and seam-cutting steps described in Section 2. Visual artifacts are highlighted. Also shown are our "corrected" panoramas generated using our post-processing tools.

Since our results are subjective in nature, we also examine our tools performance in terms of time needed to correct mosiacing artifacts as well as the user's experience. We performed a user-study comparing our tool and Photoshop CS5. We asked ten participants who are experienced in photo editing using Photoshop to correct typical

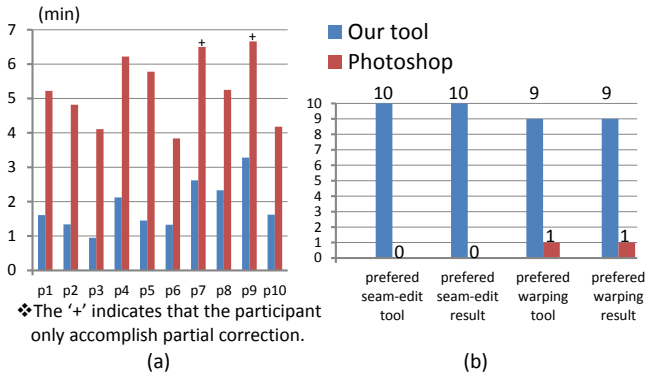

Figure 6: *User-study result of comparing our panorama correction tool and Photoshop CS5. (**a**) Timing results from 10 users. On average our tool (average around 1.5min) is significantly faster than Photoshop CS5 (average around 5min) (**b**) Preference by the user as to which application they would prefer to use, and which result they preferred. Again, we can see that our tool was most preferred among all users.*

Figure 5: *Example results generated by our approach. Shown are the original computed panoramas followed by our edited results.*

artifacts found in mosaiced images. In our experiment, all participants were first trained using sample cases to get familiar with both our tool and Photoshop. Next each participant was required to correct a test case using both our tool and Photoshop. The operating time was recorded for each tool respectively. We also asked the participants which tool provided a better experience to the user and which tool produced their preferred results. From Figure 6(a), we can see that the operating time of our tool is approximately three times faster than Photoshop. At the same time, as shown in Figure 6(b), nearly all participants felt that our tool provided a better user experience compare to Photoshop, and concluded that our tool generates preferred results. The only concern that arose in the user-study was that one user reported the snapping effect of our local warping yielded a jittering experience. However, this experience disappears when they became more familiar with the snapping tool.

To conclude, we have introduced an interactive editing tool to help hide alignment errors in panoramic images. In particular, we described methods to perform local seam-editing and local content-aware warping. By providing a post-processing tool tailored for panoramic images, we are able to relax the error tolerance for the initial warp estimation to handle cases that other software cannot process. While our focus has been solely on post-processing editing, this latter example of loosening the constraints on the initial alignment suggests that it may be beneficial to introduce an interactive step in the alignment stage together with our post-processing tool. This is an interesting topic we plan to explore in future work.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., , AND COHEN, M. 2004. Interactive digital photo-montage. *ACM Transactions on Graphics (SIGGRAPH) 23*, 3, 294–302.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *TPAMI 23*, 1222–1239.

BROWN, M., AND LOWE, D. 2003. Recognising panoramas. In *ICCV*.

LI, S. 2001. *Markov Random Field Modeling in Image Analysis (2nd Edition)*. Springer-Verlag.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60* (November), 91–110.

SUMMA, B., TIERNY, J., AND PASCUCCI, V. 2012. Panorama weaving: fast and flexible seam processing. *ACM Trans. Graph. 31*, 4, 83:1–83:11.

SZELISKI, R. 2006. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis. 2*, 1–104.