

Rectangling Panoramic Images via Warping

Kaiming He

Microsoft Research Asia

Huiwen Chang*

ITCS, Tsinghua University

Jian Sun

Microsoft Research Asia



(a) input panorama



(b) image completion



(c) cropping



(d) our content-aware warping

Figure 1: Rectangling a panoramic image. (a) Stitched panoramic image. (b) Image completion result of “content-aware fill” in Adobe Photoshop CS5. The arrows highlight the artifacts. (c) Cropped using the largest inner rectangle. (d) Our content-aware warping result.

Abstract

Stitched panoramic images mostly have irregular boundaries. Artists and common users generally prefer rectangular boundaries, which can be obtained through cropping or image completion techniques. In this paper, we present a content-aware warping algorithm that generates rectangular images from stitched panoramic images. Our algorithm consists of two steps. The first local step is mesh-free and preliminarily warps the image into a rectangle. With a grid mesh placed on this rectangle, the second global step optimizes the mesh to preserve shapes and straight lines. In various experiments we demonstrate that the results of our approach are often visually plausible, and the introduced distortion is often unnoticeable.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation;

Links: [DL](#) [PDF](#)

1 Introduction

With the advance of image alignment and stitching techniques [Szeliski 2006], creating panoramic images has become an increasingly popular application. Due to the projections (*e.g.*, cylindrical, spherical, or perspective) that warp the source images for alignment, and also due to the casual camera moving, it is almost unavoidable that the stitched panoramic images exhibit irregular boundaries (Fig. 1(a)). But most users favor rectangular boundaries for publishing, sharing, and printing photos. For example, over 99% images in the tag “panorama” in Flickr (*flickr.com*) have rectangular boundaries. In this paper, we study the issue of generating a rectangular panoramic image from the irregular one, termed as “rectangling” the image.

A simple solution is to crop a panoramic image with a rectangle. But cropping may lose desired content and reduce the impression

Keywords: warping, panorama editing, image retargeting

*This work is done when Huiwen Chang is an intern at Microsoft Research Asia (MSRA).

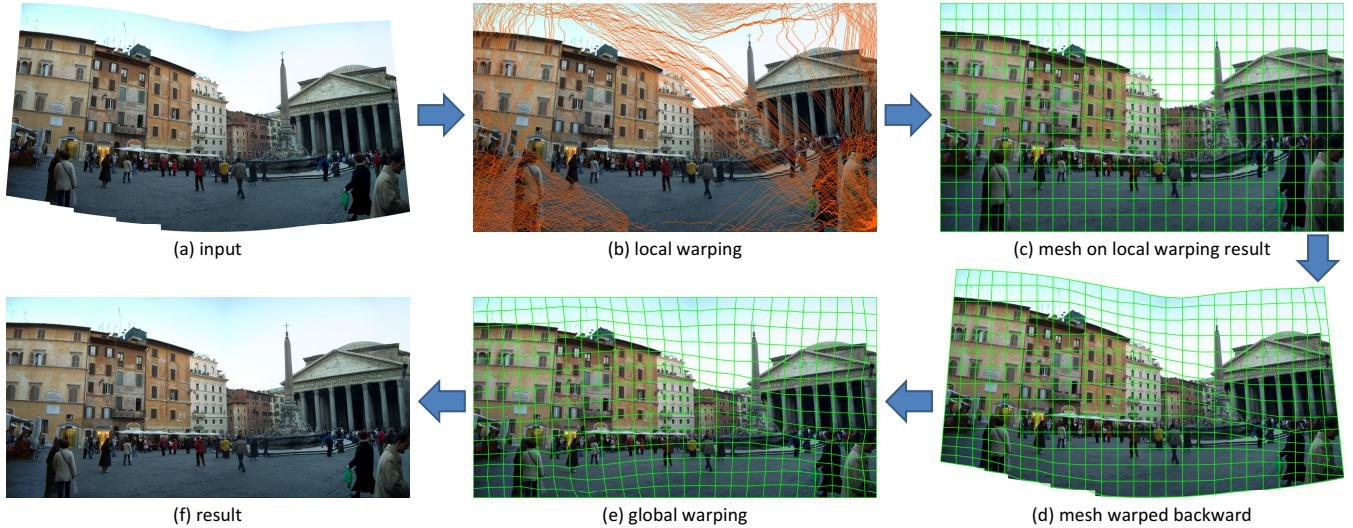


Figure 2: Algorithm pipeline. (a) Input. (b) Local warping via Seam Carving, shown with all the seams. (c) A grid mesh placed on the local warping result. (d) The grid mesh is warped backward and placed on the input. (e) The optimized mesh of global warping. (f) Result.

of a wide field of view (Fig. 1(c)). Another solution is to synthesize the missing region in a bounding box using image completion techniques (*e.g.*, [Criminisi et al. 2004; Wexler et al. 2007; Barnes et al. 2009]). Though these techniques are suitable for extending textures or simple structures like straight lines, they may fail synthesizing semantic content (Fig. 1(b)). Recently Kopf et al. [2012] combine cropping and completion to address this problem.

In this work, we propose to adopt warping for rectangling panoramic images. We argue that warping is an attractive strategy for this task, thanks to the long-realized fact that panoramic/wide-angled images inevitably exhibit distortion [Zorin and Barr 1995; Zelnik-Manor et al. 2005]. General users appear accustomed to and tolerant of this distortion - millions of panoramic images have been shared in the websites like Flickr, photosynth.net, 360cities.net, etc. Photographers and artists seem to be willing to trade warping distortion for the impressive wide fields of view. With this observation, we believe that the extra distortion introduced by a properly designed warping algorithm for rectangling can be acceptable and visually unnoticeable (Fig. 1(d)).

The challenges of developing such a warping approach come from the irregular boundaries. Improperly stretching the input boundary to a rectangle might bring in unexpected distortion, so a content-preserving solution is desired. But most existing content-preserving warping techniques, either for projection manipulations [Carroll et al. 2009; Kopf et al. 2009; Carroll et al. 2010], image retargeting [Wang et al. 2008; Zhang et al. 2009; Chang and Chuang 2012], or video stabilization [Liu et al. 2009], are based on grid meshes and assume the input images are rectangular. Shape deformation methods [Igarashi et al. 2005; Schaefer et al. 2006] require predefined meshes and user-specified control points as input/output constraints. Interpolation-based deformation methods [Ju et al. 2005; Joshi et al. 2007; Lipman et al. 2008] require user-specified polygon cages as input/output and are not content-aware.

We present a novel content-preserving warping method for rectangling panoramic images. The key idea is a two-step method that first locally warps the image to fit a rectangle and then globally optimizes a mesh placed on this rectangle. In the first step, we modify the *Seam Carving* algorithm [Avidan and Shamir 2007] to expand the irregular image to a rectangle (Fig. 2 (b)). We consider Seam Carving as a warping method that displaces all pixels on one side of

each seam. In the second step, we place a grid mesh on the rectangle image generated by Seam Carving. With the displacement field of the first step, this mesh is warped back and placed in the original irregular image (Fig. 2 (c, d)). Then we globally optimize this mesh, fitting it to a rectangle while preserving perceptual properties including shapes and straight lines. Our method is fully automatic. It is purely content-based and requires no prior knowledge about the projections.

Our warping scheme works well in a variety of cases, typically when the irregular boundaries come from the projections required for alignment and from the casual camera moving. Yet, our method may fail when the irregular boundaries are the outcome of the missing images required for stitching the full scene (*e.g.*, Fig. 13).

This paper has two main contributions. Firstly, we are the first (to the best of our knowledge) to adopt the warping idea for rectangling panoramic images. Secondly, we develop an automatic content-aware method for warping irregular panoramas, without any requirement of user-provided mesh/cage/control-point.

2 Related Work

We review the related work on panorama imaging, image warping, and image completion.

Image alignment and stitching. We briefly summarize a typical pipeline of panorama composition as follows (see [Szeliski 2006] for a comprehensive survey). The source images are registered using image features [Brown and Lowe 2003] and projected onto the same coordinate system [Szeliski and Shum 1997]. A graph-cut technique [Boykov et al. 2001] is applied to stitch the images [Agarwala et al. 2004], followed by Poisson blending [Pérez et al. 2003]. Recent improvements involve efficient blending [Agarwala 2007] and flexible seam manipulation [Summa et al. 2012].

Projections. It has been long realized [Zorin and Barr 1995; Zelnik-Manor et al. 2005] that projecting 3D scenes onto 2D images unavoidably leads to warping distortion. Commonly used projections include perspective, cylindrical, and spherical projections [Szeliski and Shum 1997]. The perspective projection preserves straight lines but may severely stretch the shapes. The cylindrical and spherical projections maintain local shapes, but they bend

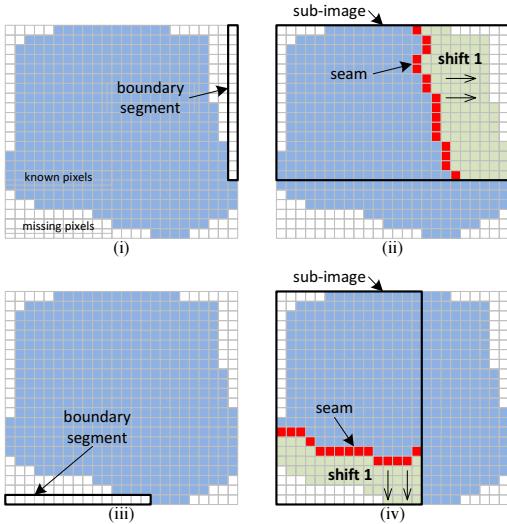


Figure 3: Local warping via Seam Carving. (i): Find the longest “boundary segment”. (ii): Compute a seam using Seam Carving in a sub-image containing this boundary segment, and shift all the pixels on one side of the seam by one pixel (marked as green). (iii) & (iv): Compute the next seam.

straight lines. These projections contribute to the irregular boundaries of the stitched images.

The method in [Peleg et al. 2000] projects and stitches the images on adaptive manifolds based on the knowledge about camera motions. It is able to create a “rectified” panorama (but still with irregular boundaries) when the camera is tilted. This method adapts to camera motions, but is not content-aware.

To maintain the perceptual properties like straight lines and/or unstretched shapes, substantial effort has been paid on locally adaptive projections [Carroll et al. 2009; Kopf et al. 2009; Carroll et al. 2010]. The idea is to re-distribute the warping distortion to the visually insignificant regions. Given some user-specified constraints, these methods optimize spatially-varying projections parameterized by meshes. Notice that despite the visual plausibility, it is still inherently impossible to remove all warping distortion.

Because warping distortion is unavoidable in global/local projections, we believe that reasonable warping distortion can be unnoticeable and tolerable when rectangling panoramic images.

Image retargeting and warping. Image retargeting is to resize the image based on the content. The Seam Carving method [Avidan and Shamir 2007] removes/inserts seams to change the image size. A variety of other methods are explicitly based on warping [Wolf et al. 2007; Wang et al. 2008; Zhang et al. 2009; Chang and Chuang 2012]. They optimize a displacement field with considerations on content-based properties like saliency, shapes, and/or straight lines.

All the above retargeting methods assume rectangular inputs; some of these methods further require grid meshes to preserve high-level properties like shapes [Wang et al. 2008; Zhang et al. 2009] and straight lines [Chang and Chuang 2012].

Image completion. Image completion has been considered as a way to creating rectangular panoramic images, e.g., in [Criminisi et al. 2004; Adobe 2009; Kopf et al. 2012]. Advancing image completion techniques are exemplar-based [Criminisi et al. 2004; Wexler et al. 2007; Komodakis and Tziritas 2007; Hays and Efros 2008; Barnes et al. 2009; Pritch et al. 2009; He and Sun 2012; Kopf



Figure 4: A zoom-in region near the fountain in Fig. 2.

et al. 2012]: they copy content from the known part to synthesize the missing part. This strategy can yield visually plausible results, typically in the case of textural and less semantic regions. But it is less suitable for synthesizing higher-level semantic content, and may fail occasionally when the missing region is large. Further, even the successfully *synthesis* is not a faithful record of the real scene. This may not be desired by all users.

3 Algorithm

Our warping algorithm contains two steps: a local warping step via Seam Carving, and a global warping step based on meshes. The local step provides a preliminary rectangular image. Its main purpose is to place a grid mesh on the input (Fig. 2(b)(c)). The global step optimizes this mesh to preserve perceptual properties including shapes and straight lines (Fig. 2(d-f)).

3.1 Mesh-free Local Warping

Our local warping step is a modification of the Seam Carving algorithm [Avidan and Shamir 2007]. The original Seam Carving algorithm inserts a horizontal/vertical seam *through* the image, so expands it by one pixel vertically/horizontally. If we allow a seam to be *not through* the image, we can change the boundary shape of the image and fit it to a rectangle. Recently it has been proposed [Qi and Ho 2012] to use Seam Carving to tailor a rectangle image into an irregular shape (like circles or ovals). This is like an opposite task to ours.

Suppose the bounding box of the irregular input image is the target rectangular boundary. We define a “boundary segment” as a connected sequence of missing pixels on one of the four sides (top/bottom/left/right) of the target rectangular boundary. Fig. 3(i) shows an example. Each time we select the longest boundary segment and insert one seam. For the situation when the boundary segment is on the right side (as in Fig. 3(i)), we insert a vertical *non-through* seam in the image. This seam shares the same starting and ending y-coordinates as the selected boundary segment (Fig. 3(ii)). Then we shift all the pixels on the right of this seam by one pixel to the right. The situation of a seam on the top/bottom/left side can be treated similarly. We repeatedly insert seams in this way (see Fig. 3(iii)(iv)), until the rectangle boundary has no missing pixel.

To find the non-through seam, we run the Seam Carving algorithm to find a through seam in a “sub-image”. For example, the sub-image in Fig. 3(ii) has the same starting and ending y-coordinates as the boundary segment. Then we apply the Seam Carving algorithm on this sub-image (in our implementation we adopt the Improved Seam Carving [Rubinstein et al. 2008]). Because the sub-image may contain missing pixels, we assign an infinite cost (10^8) to these pixels to prevent the seam from passing them.

In the viewpoint of filling the unknown regions, inserting a seam would reduce the number of missing pixels in the image (this number equals to the number of pixels on the seam). In the viewpoint

of warping, inserting a seam is equivalent to computing a displacement field $\mathbf{u}(\mathbf{x})$. We use $\mathbf{x} = (x, y)$ to denote the coordinates of an output pixel, and $\mathbf{u} = (u_x, u_y)$ to denote displacement. The output pixel value can be obtained by warping the input image:

$$\mathbf{I}_{out}(\mathbf{x}) = \mathbf{I}_{in}(\mathbf{x} + \mathbf{u}(\mathbf{x})), \quad (1)$$

where \mathbf{I}_{in} and \mathbf{I}_{out} represent the input and the current output images. For the example in Fig. 3(ii), the displacement \mathbf{u} is $(-1, 0)$ for all pixels on the right of this seam, and is zero for all other pixels. With all the seams computed, we obtain a displacement field \mathbf{u} accordingly. We term this step as local warping because the distortion is locally distributed near the seams (see Fig. 4).

3.2 Mesh-based Global Warping

To generate a rectangular image while preserving high-level properties like straight lines and shapes, we optimize a global energy based on meshes.

3.2.1 Mesh Placement

To generate a grid mesh on the input irregular image, we first place a grid mesh on the rectangular result of the local warping step (Fig. 2(c)). We use a grid mesh with around 400 vertexes throughout this paper. We warp back all the grid vertexes to the input image domain using the displacement field of local warping. Thus we obtain a mesh placed on the input image (Fig. 2(d)). The global warping step is based on this mesh. The resulting image of local warping is discarded afterward.

3.2.2 Energy Function

We design an energy function that imposes rectangular boundary constraints while preserving shapes and straight lines. Our energy definition is partially based on [Chang and Chuang 2012], but is in a simpler form and involves much fewer parameters.

We parameterize the grid mesh \mathbf{V} as $\{\mathbf{v}_i\}$, where $\mathbf{v}_i = (x_i, y_i)$ is the position of a grid vertex. We denote the input mesh as $\hat{\mathbf{V}}$. We optimize an energy function about the output mesh \mathbf{V} .

Shape Preservation. Our shape-preserving energy E_S encourages each quad to undergo a *similarity transformation* (*i.e.*, translation+rotation+scaling), as used in [Zhang et al. 2009; Chang and Chuang 2012]. It is defined as:

$$E_S(\mathbf{V}) = \frac{1}{N} \sum_q \| (A_q (\underline{A_q^T A_q})^{-1} A_q^T - I) \mathbf{V}_q \|^2. \quad (2)$$

Here N is the number of quads in the mesh, q is a quad index, I is a unit matrix, and A_q is a 8×4 matrix and \mathbf{V}_q is a 8×1 vector on the quad:

$$A_q = \begin{bmatrix} \hat{x}_0 & -\hat{y}_0 & 1 & 0 \\ \hat{y}_0 & \hat{x}_0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{x}_3 & -\hat{y}_3 & 1 & 0 \\ \hat{y}_3 & \hat{x}_3 & 0 & 1 \end{bmatrix}, \quad \mathbf{V}_q = \begin{bmatrix} x_0 \\ y_0 \\ \vdots \\ x_3 \\ y_3 \end{bmatrix}. \quad (3)$$

Here we use $(x_0, y_0), \dots, (x_3, y_3)$ to denote the four pairs of coordinates of the output quad, and $(\hat{x}_0, \hat{y}_0), \dots, (\hat{x}_3, \hat{y}_3)$ those of the input quad. The derivation of E_S can be found in [Zhang et al. 2009]. E_S is a quadratic function of \mathbf{V} .

Unlike the previous works, we do not introduce any saliency weight into the shape preserving term. This is because panorama images

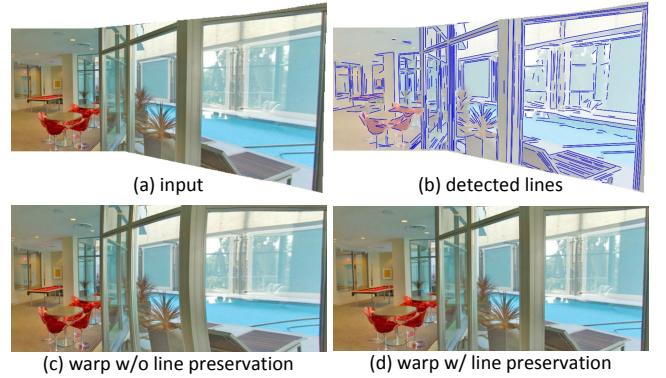


Figure 5: An example of line preservation. We remove the line preservation through setting $\lambda_L = 0$ and obtain the result in (c).

often cover a wide variety of content and contain no particularly salient object.

Line Preservation. We follow a similar way as in [Chang and Chuang 2012] to preserve lines. Our line-preserving energy E_L encourages that the straight lines are kept straight, and parallel lines are kept parallel. We use the code of [Von Gioi et al. 2010] to detect line segments in the input image¹. We cut all the detected line segments with the edges of the input mesh, so that each resulting line segment is located inside one quad. We quantize the line orientation range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ into $M=50$ bins. To preserve straightness and parallelism, we encourage that all the line segments in the same bin share a common rotation angle θ . The line preserving term E_L involves all these angles $\{\theta_m\}_{m=1}^M$.

Given a line segment, we compute its orientation vector (with a scale) \mathbf{e} using the difference vector of its two end points. If we represent the two end points of a line segment as a bilinear interpolation of its quad vertexes \mathbf{V}_q , we can write \mathbf{e} as a linear function of \mathbf{V}_q . We denote the input orientation vector of this line segment as $\hat{\mathbf{e}}$. Given a target rotation angle θ_m , we want to minimize the following distortion of a line segment:

$$\|sR\hat{\mathbf{e}} - \mathbf{e}\|^2, \quad (4)$$

where $R = \begin{pmatrix} \cos \theta_m & -\sin \theta_m \\ \sin \theta_m & \cos \theta_m \end{pmatrix}$ is a rotation matrix, and s is a scaling factor of this line segment. Minimizing with respect to s gives: $s = (\hat{\mathbf{e}}^T \hat{\mathbf{e}})^{-1} \hat{\mathbf{e}}^T R^T \mathbf{e}$. Substituting s into (4), we can show that the distortion in (4) is a quadratic function of \mathbf{e} :

$$\|C\mathbf{e}\|^2, \quad (5)$$

where the matrix C is

$$C = R\hat{\mathbf{e}}(\hat{\mathbf{e}}^T \hat{\mathbf{e}})^{-1} \hat{\mathbf{e}}^T R^T - I. \quad (6)$$

Because \mathbf{e} is a linear function of \mathbf{V}_q , the distortion in (5) can be written as a quadratic function of \mathbf{V}_q .

Our line preserving energy E_L is defined as the mean distortion for all line segments:

$$E_L(\mathbf{V}, \{\theta_m\}) = \frac{1}{N_L} \sum_j \|C_j(\theta_{m(j)}) \mathbf{e}_{q(j)}\|^2, \quad (7)$$

where N_L is the number of line segments. A line segment is indexed by j , and $q(j)$ is the quad containing it. The matrix

¹www.ipol.im/pub/art/2012/gjmr-1sd/

$C_j(\theta_{m(j)})$, computed using (6), depends on the desired rotation angle $\theta_{m(j)}$ of the bin that contains this line segment. E_L is a quadratic function of \mathbf{V} .

Unlike [Chang and Chuang 2012], our energy is decoupled from the scaling factor (s) and the translation of each line segment. Thus our energy has fewer variables and parameters.

Boundary Constraints. We want to drag the vertexes on the mesh boundary to a rectangle. The boundary constraint term E_B is simply defined as:

$$E_B(\mathbf{V}) = \sum_{\mathbf{v}_i \in L} x_i^2 + \sum_{\mathbf{v}_i \in R} (x_i - w)^2 + \sum_{\mathbf{v}_i \in T} y_i^2 + \sum_{\mathbf{v}_i \in B} (y_i - h)^2. \quad (8)$$

Here L/R/T/B denote the left/right/top/bottom boundary vertexes, and w/h denote the width/height of the target rectangle (e.g., a bounding box). Notice that we only constrain one of the two coordinates of each boundary vertex, e.g., a vertex on the top boundary is free to move horizontally.

Total Energy. Our total energy function E is:

$$E(\mathbf{V}, \{\theta_m\}) = E_S(\mathbf{V}) + \lambda_L E_L(\mathbf{V}, \{\theta_m\}) + \lambda_B E_B(\mathbf{V}), \quad (9)$$

where λ_L and λ_B are two weights. We set the boundary weight λ_B as infinity (10^8) to impose a hard boundary constraint.

The line preservation weight λ_L is the main influential parameter in our algorithm. In experiments we find it works consistently well when λ_L is sufficiently large (e.g., ≥ 10). This means the importance of line preservation (E_L) is higher than shape preservation (E_S). This implies that human eyes are more sensitive to bended straight lines than to distorted shapes. We fix $\lambda_L=100$ throughout this paper. Fig. 5 shows the effect of line preservation.

3.2.3 Efficient Optimization

We adopt an alternating algorithm to minimize $E(\mathbf{V}, \{\theta_m\})$. The initialization is given by the local warping result (i.e., simply a regular grid). We run 10 iterations of the following alternating algorithm:

Fix $\{\theta_m\}$ and update \mathbf{V} . In this case E is a quadratic function on \mathbf{V} and can be optimized via solving a linear system. Because there are only a few hundreds of vertexes in \mathbf{V} , the running time of this step is ignorable (in a C++ implementation).

Fix \mathbf{V} and update $\{\theta_m\}$. Because θ_m is independent of each other, we can separately optimize each θ_m . In this case, we minimize:

$$\min_{\theta_m} \sum_{j \in \text{bin}(m)} \|C_j(\theta_m) \mathbf{e}_{q(j)}\|^2. \quad (10)$$

This problem could be optimized by iterative solvers like the Newton's method. Rather, we adopt a simple non-iterative solution driven by the intuition of this energy function. The intuitive meaning of (10) is to find a common rotation angle θ_m for all the line segments in the m -th bin, such that θ_m approximates the relative angle between any line segment \mathbf{e}_j and its counterpart $\hat{\mathbf{e}}_j$. So we simply compute the relative angle between \mathbf{e}_j and $\hat{\mathbf{e}}_j$ for all line segments in the m -th bin, and take their average as θ_m . We find this is a sufficiently good solution to minimizing (10).

3.2.4 Stretching Reduction and Post-processing

We have assumed the bounding box is the target rectangular boundary. But if this rectangle has an unwanted aspect ratio, the output image may appear stretched. This problem is particularly obvious

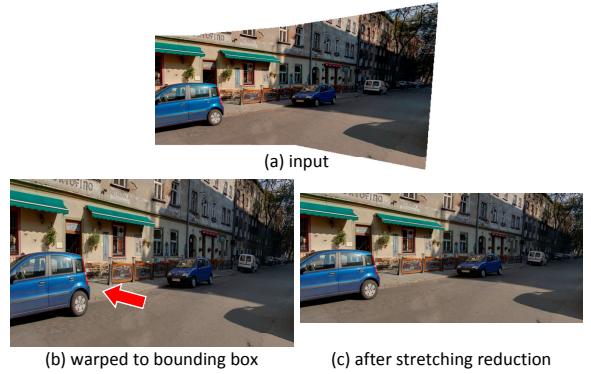


Figure 6: Stretching reduction. (a) Input. (b) The image is warped to the bounding rectangle of the input. The car is obviously stretched. (c) The target rectangle is re-scaled, and the mesh is optimized again.

in the case of perspective projection (see Fig. 6). To reduce the stretching, we update the target rectangle after the global warping step. For each mesh quad, we compute its x-scaling factor s_x simply through $s_x = (x_{\max} - x_{\min}) / (\hat{x}_{\max} - \hat{x}_{\min})$. The mean x-scaling factor \bar{s}_x is the average of s_x for all quads. The target rectangle width is then scaled by $1/\bar{s}_x$. The height is addressed similarly using the mean y-scaling factor \bar{s}_y . We run the global warping step again using this updated target rectangle. Fig. 6 shows this step reduces the stretching distortion.

With the optimized mesh, we bilinearly interpolate the displacement of any pixel from the displacement of the four quad vertexes. But this may leave a few missing pixels on the boundary occasionally, because a grid line may not well fit the irregular input boundary. We simply fill each missing pixel using the color of the nearest known pixels.

3.3 Implementation and Speed

Because the displacement map for warping is mostly smooth, we compute it at a small image scale. We first downsample the input image to a fixed size (1 mega-pixel). Both the local/global warping steps are run on the downsampled image. Then we upsample the resulting rectangular *displacement* map through bilinear interpolation. We use this map to warp the full resolution input image.

In our implementation (C++, single-core), the algorithm processes a 10-Mp panoramic image (Fig. 1) in 1.5 seconds on a PC with an Intel Core i7 2.9GHz CPU and 8GB memory. The running time is dominantly on warping (interpolating) the full resolution image with the computed displacement map. In comparison, the image completion tool “content-aware fill” in Adobe Photoshop processes this 10-Mp image (with 18% pixels missing) in 19.1s.

4 Results

We demonstrate our results on a variety of real cases in Fig. 1, 2, 7, 8, and 9. Our method successfully maintains the content without introducing noticeable artifacts. Typically, Fig. 7 (top) and Fig. 9 (top) are two challenging but practical 360°full-view panoramic images. We believe our warping method is particularly favored for creating such compelling photos.

In Fig. 1 and 7 we compare with “content-aware fill” in Adobe Photoshop CS5. This image completion tool is based on [Wexler et al. 2007; Barnes et al. 2009] (according to [Adobe 2009]). In Fig. 8

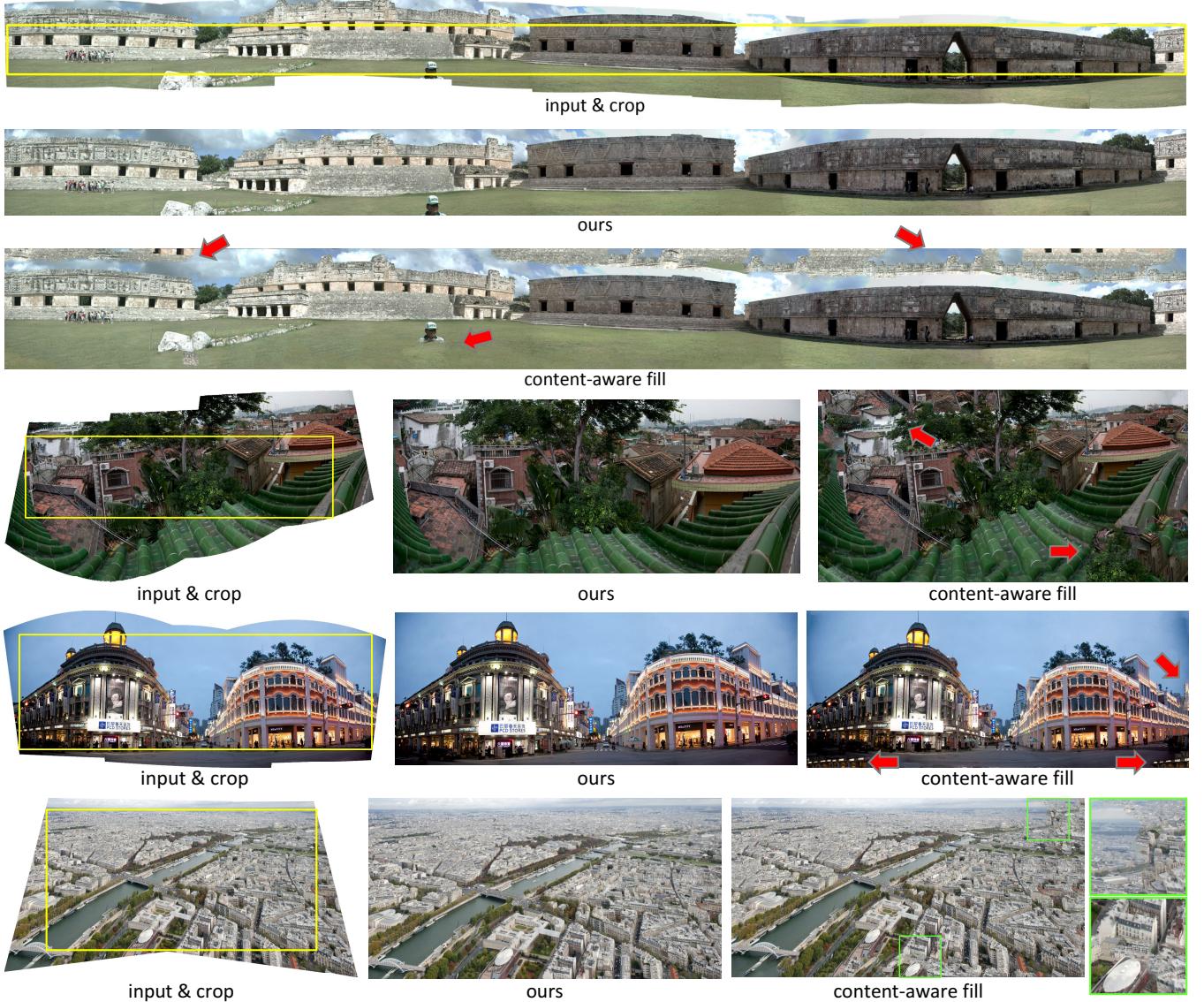


Figure 7: Comparisons with the image completion technique “content-aware fill”. The arrows and the zoom-in windows show the artifacts.

we compare with another recent completion technique [Kopf et al. 2012] on the public images/results provided by the authors. We see that these state-of-the-arts image completion techniques are limited in synthesizing semantically reasonable content, *e.g.*, for the buildings in Fig. 1(b) and Fig. 8(left). They may also treat semantic content as textures, *e.g.*, for the city-view in Fig. 7(bottom). We also show the cropping results in Fig. 1 and 7. We see that cropping may severely limit the field of view.

Fig. 10 shows an example combining image completion and cropping. In this case we manually crop the image completion result using a window that removes obvious artifacts. But we see this operation still remove much image content.

User Study. To collect a large number of panoramic images with irregular boundaries for user study, we construct a “semi-synthetic” dataset from the MIT SUN360 database [Xiao et al. 2012]. This database contains 10,405 real full-view ($360^\circ \times 180^\circ$) panoramic scenes contributed by Internet users. These scenes have been labeled as 80 categories, involving indoor/outdoor cases with various

man-made/natural scenarios (*e.g.*, bedroom, street, mountain, etc). We use the first 5 scenes (in the original order in the database) if a category contains more than 5, and obtain a subset of 367 scenes.

Xiao et al. [2012] have provided code to simulate taking a photo with a usual camera in a full-view scene². Thus we can simulate the behavior of capturing a photo sequence for stitching. We use three ways to synthesize the sequences (see Fig. 11 top). We generate a 5×1 array of images used for cylindrical projections, a 3×1 array for perspective projections, and a 3×2 array for spherical projections. The nearby images have randomly 30-50% overlapping area. The camera position of each image is slightly disturbed at random to simulate casual camera moving. We synthesize 3 sequences (stitched using the specified projection) for each full-view scene. Thus our dataset consists of 1,101 (367×3) stitched images. Some examples are in Fig. 11.

We compare our warping strategy with the image completion strat-

²mit.edu/jxiao/Public/software/pano2photo/

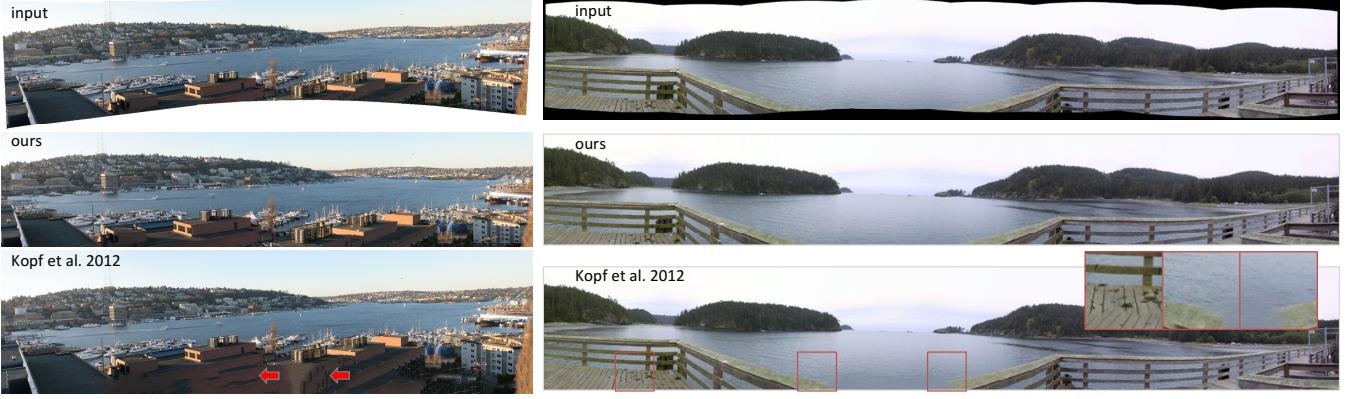


Figure 8: Comparisons with Kopf et al.’s [2012] image completion technique. The input and completion results are from Kopf et al.’s website.

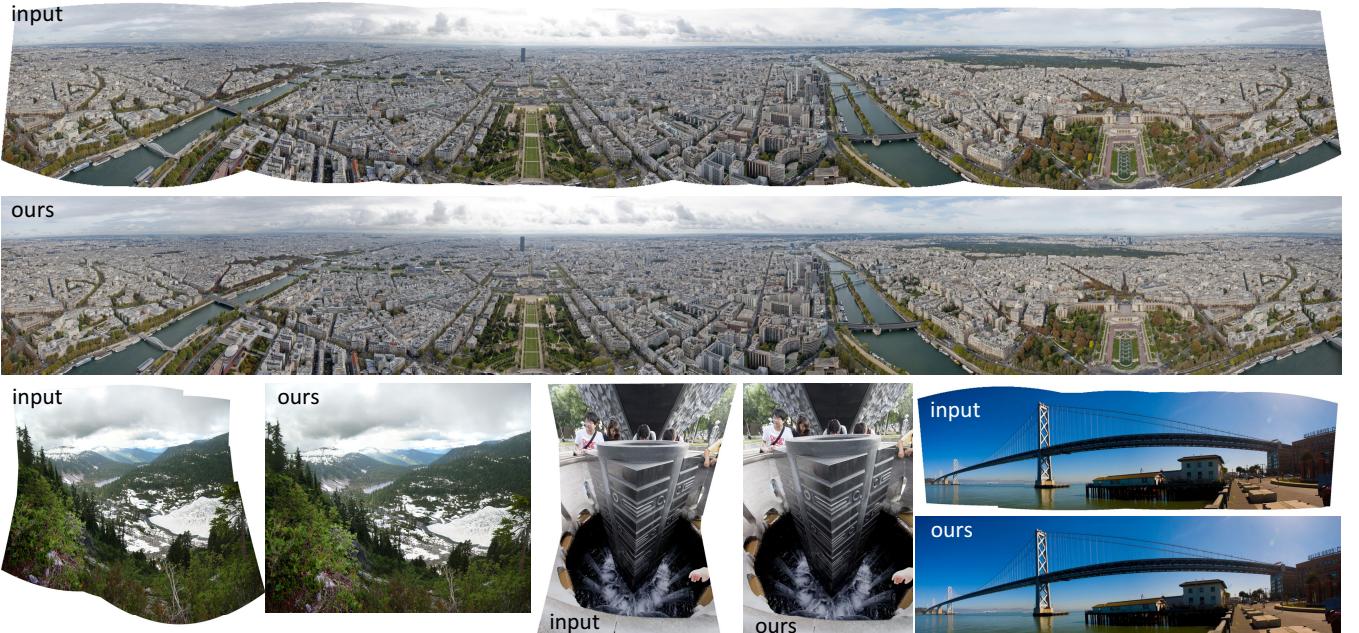


Figure 9: More results of our method. The input of the “bridge” image on the bottom right is from Kopf et al.’s paper [2012].

egy in this dataset. We choose the content-aware fill as the image completion tool. At each time three images are shown on one screen: the input, our warping result, and the completion result. The user is allowed to zoom-in the images. The user needs to answer whether ours or the completion result is preferred, or no preference. In case of “no preference” the user needs to answer whether the two results are “both good” and “both bad”. Our study has 10 participants, including 5 researchers/students with computer graphics/vision backgrounds and 5 volunteers outside this community.

Fig. 12 shows the user study results. Our method is substantially preferred. This shows the superiority of a warping strategy for rectangling panoramas. We also see our advantage is more significant in perspective projection. This is because the missing region is often larger in the perspective cases than in the cylindrical/spherical cases, and the image completion is less likely to succeed.

5 Limitations and Future Work

Our method works well when the irregular boundaries are the results of projections and casual camera movements. But we notice that our method is limited in the following aspects. (i) Our method may fail when the scene is not completely shot, e.g., when a few images are missing in an image array. In this case our approach may lead to obvious distortion (Fig. 13). (ii) Our method may distort content near very concave boundaries (Fig. 14(a)(b)). To reduce the concavity, one may manually introduce a transparent region, warp the image with this region treated as known pixels, and fill this region using image completion after warping (Fig. 14(c)). (iii) Our method may bend undetected lines. This could be addressed by a few user interactions. (iv) Our method may not preserve all lines when a local region has a large number of lines. This appears a common challenge for warping methods.

Our warping method is purely content-based, so can be widely applied in case the knowledge in the stitching step is not available. But it would be an interesting problem if one is allowed to con-



Figure 10: Comparisons with “image completion + cropping”. The completion tool is content-aware fill. The cropping window is manually chosen to avoid image completion artifacts.

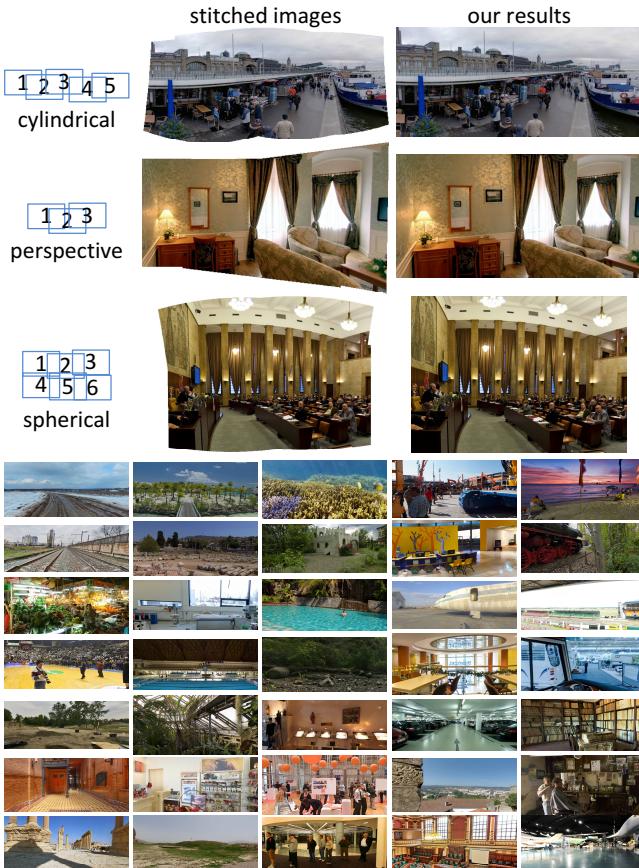


Figure 11: Example images in our dataset. Top: three projections used. Bottom: some examples of our warping results (input not shown here). These images show the variety of the dataset.

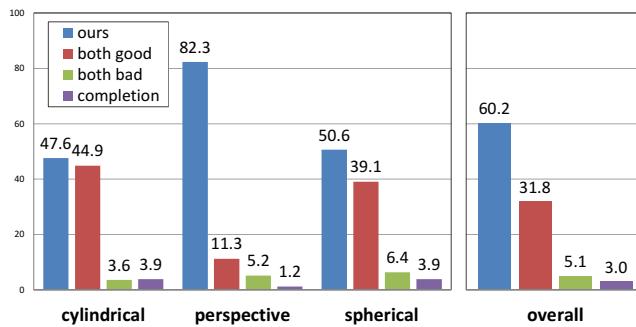


Figure 12: User study results. The numbers are shown in percentage and averaged on 10 participants. Left: the preference in each projection. Right: the preference in all 1,101 images.



Figure 13: Failure: a large portion of the scene is missing.

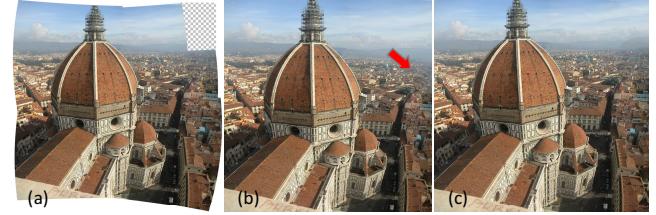


Figure 14: A very concave boundary (a) may lead to severe local distortion (b). We introduce a transparent region (marked as checkered) to reduce the concavity. Then we warp this new input, and fill the warped transparent region by image completion to obtain (c).

sider the rectangling issue during the stitching step. We leave this problem for future study.

Acknowledgements

The authors would like to thank Jiangyu Liu for making the video, Fang Wen for fruitful discussions, the volunteers for participating in the user study, and all anonymous reviewers for their insightful comments. The author Huiwen Chang was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174.

References

- ADOBE, 2009. www.adobe.com/technology/projects/content-aware-fill.html.
- AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. In *SIGGRAPH 2004*, 294–302.

- AGARWALA, A. 2007. Efficient gradient-domain compositing using quadtrees. In *SIGGRAPH 2007*.
- AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. In *SIGGRAPH 2007*.
- BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. In *SIGGRAPH 2009*.
- BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1222–1239.
- BROWN, M., AND LOWE, D. 2003. Recognising panoramas. In *International Conference on Computer Vision (ICCV)*, 1218 – 1225 vol.2.
- CARROLL, R., AGRAWAL, M., AND AGARWALA, A. 2009. Optimizing content-preserving projections for wide-angle images. In *SIGGRAPH 2009*, 43:1–43:9.
- CARROLL, R., AGARWALA, A., AND AGRAWALA, M. 2010. Image warps for artistic perspective manipulation. In *SIGGRAPH 2010*, 127:1–127:9.
- CHANG, C.-H., AND CHUANG, Y.-Y. 2012. A line-structure-preserving approach to image resizing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1075 –1082.
- CRIMINISI, A., PÉREZ, P., AND TOYAMA, K. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing (TIP)*, 1200–1212.
- HAYS, J., AND EFROS, A. A. 2008. Scene completion using millions of photographs. In *SIGGRAPH 2008*, 87–94.
- HE, K., AND SUN, J. 2012. Statistics of patch offsets for image completion. In *European Conference on Computer Vision (ECCV)*, Springer-Verlag, 16–29.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. In *SIGGRAPH 2005*.
- JOSHI, P., MEYER, M., DEROSE, T., GREEN, B., AND SANOKI, T. 2007. Harmonic coordinates for character articulation. In *SIGGRAPH 2007*.
- JU, T., SCHAEFER, S., AND WARREN, J. 2005. Mean value coordinates for closed triangular meshes. In *SIGGRAPH 2005*.
- KOMODAKIS, N., AND TZIRITAS, G. 2007. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing (TIP)*.
- KOPF, J., LISCHINSKI, D., DEUSSEN, O., COHEN-OR, D., AND COHEN, M. 2009. Locally adapted projections to reduce panorama distortions. In *Computer Graphics Forum*, Wiley Online Library, 1083–1089.
- KOPF, J., KIENZLE, W., DRUCKER, S., AND KANG, S. B. 2012. Quality prediction for image completion. In *SIGGRAPH Asia 2012*.
- LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. In *SIGGRAPH 2008*.
- LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3d video stabilization. In *SIGGRAPH 2009*.
- PELEG, S., ROUSSO, B., RAV-ACHA, A., AND ZOMET, A. 2000. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1144–1154.
- PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. In *SIGGRAPH 2003*, 313–318.
- PRITCH, Y., KAV-VENAKI, E., AND PELEG, S. 2009. Shift-map image editing. In *International Conference on Computer Vision (ICCV)*, IEEE, 151–158.
- QI, S., AND HO, J. 2012. Seam segment carving: retargeting images to irregularly-shaped image domains. In *European Conference on Computer Vision (ECCV)*, Springer-Verlag, 314–326.
- RUBINSTEIN, M., SHAMIR, A., AND AVIDAN, S. 2008. Improved seam carving for video retargeting. In *SIGGRAPH 2008*.
- SCHAEFER, S., MCPHAIL, T., AND WARREN, J. 2006. Image deformation using moving least squares. In *SIGGRAPH 2006*.
- SUMMA, B., TIERNY, J., AND PASCUCCI, V. 2012. Panorama weaving: fast and flexible seam processing. In *SIGGRAPH 2012*.
- SZELISKI, R., AND SHUM, H.-Y. 1997. Creating full view panoramic image mosaics and environment maps. In *SIGGRAPH 97*, 251–258.
- SZELISKI, R. 2006. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*.
- VON GIOI, R., JAKUBOWICZ, J., MOREL, J., AND RANDALL, G. 2010. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 722–732.
- WANG, Y.-S., TAI, C.-L., SORKINE, O., AND LEE, T.-Y. 2008. Optimized scale-and-stretch for image resizing. In *SIGGRAPH Asia 2008*.
- WEXLER, Y., SHECHTMAN, E., AND İRANI, M. 2007. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 463–476.
- WOLF, L., GUTTMANN, M., AND COHEN-OR, D. 2007. Non-homogeneous content-driven video-retargeting. In *International Conference on Computer Vision (ICCV)*, 1 –6.
- XIAO, J., EHINGER, K., OLIVA, A., AND TORRALBA, A. 2012. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2695–2702.
- ZELNIK-MANOR, L., PETERS, G., AND PERONA, P. 2005. Squaring the circle in panoramas. In *International Conference on Computer Vision (ICCV)*, IEEE, 1292–1299.
- ZHANG, G., CHENG, M., HU, S., AND MARTIN, R. 2009. A shape-preserving approach to image resizing. In *Computer Graphics Forum*, Wiley Online Library, 1897–1906.
- ZORIN, D., AND BARR, A. H. 1995. Correction of geometric perceptual distortions in pictures. In *SIGGRAPH 95*.