A Geodesic-Preserving Method for Image Warping

Dongping Li[1], Kaiming He[2], Jian Sun[2], and Kun Zhou[1]

[1]Zhejiang University , [2]Microsoft Research

Abstract *The manipulation of panoramic/wide-angle images is usually achieved via image warping. Though various techniques have been developed for preserving shapes and straight lines for warping, these are not sufficient for panoramic/wide-angle images. The image projections will turn the straight lines into curved "geodesic lines", and it is fundamentally impossible to keep all these lines straight. In this work, we propose a geodesic-preserving method for content-aware image warping. An energy term is introduced to preserve the geodesic appearance of the geodesic lines, and can be used with shape-preserving terms. Our method is demonstrated in various applications, including rectangling panoramas, resizing panoramic/wide-angle images, and wide-angle image manipulation. An extension to ellipse preservation for general images is also presented.*

1. Introduction

Panoramic/wide-angle images are getting increasingly popular for common users, thanks to the inexpensive and convenient devices, such as phones that capture panoramas [2] and GoPro cameras [1] that capture wide-angle photos. Panoramic/wide-angle images are inevitably warped in the image formation procedures, so image warping is a natural way of processing and manipulating these images [14, 23, 3, 12, 6, 11, 9, 22].

Studies on image warping have paid particular attention to preserving shapes [10, 15, 24] and straight lines [6, 11, 5, 7, 9]. However, these techniques are not sufficient for warping panoramic/wide-angle images. It has been long realized in cartography [16] and computer science [18] that it is impossible to preserve all straight lines when projecting panoramic/wide-angle images[1]. As Carroll *et al*. [6] recognized, the straight-line-preserving methods can fail if the number of straight lines is large. Carroll *et al*. [6] and Kopf *et al*. [11] also observe if a straight line is long spanned, preserving its straightness can seriously distort other content. In He *et al*.'s work [9], the straight lines are detected on the projected images and then preserved. But if the lines have become curved due to the projection, this method can fail to address them (Fig. 2(b)).

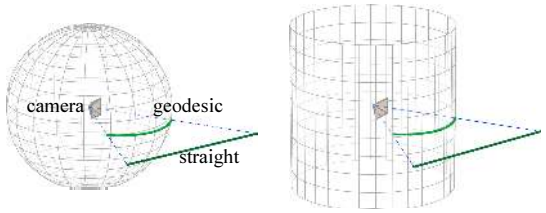


Figure 1. When a straight line is projected to a manifold, it becomes a curve. In the case of spherical projection (left), the curve is a section of a great circle; in the case of cylindrical projection (right), the curve is a section of an ellipse.

Instead of preserving straightness, in this work we propose to preserve geodesic lines. We define "geodesic lines" as projections of 3-D straight lines onto 2-D manifolds (Fig. 1). They are *great circles* on spheres and *ellipses* on cylinders[2]. Unlike the methods in [6, 11] that straighten geodesic lines, our method allows them to be curved. But an unnaturally curved geodesic line can be noticeable (Fig. 2(b)), because a geodesic line is not simply a locally smooth curve. In our solution, we constrain a geodesic line to remain "geodesic": it should be warped into another geodesic line, so can preserve its geodesic appearance. Fig. 2(c) is an example of our solution.

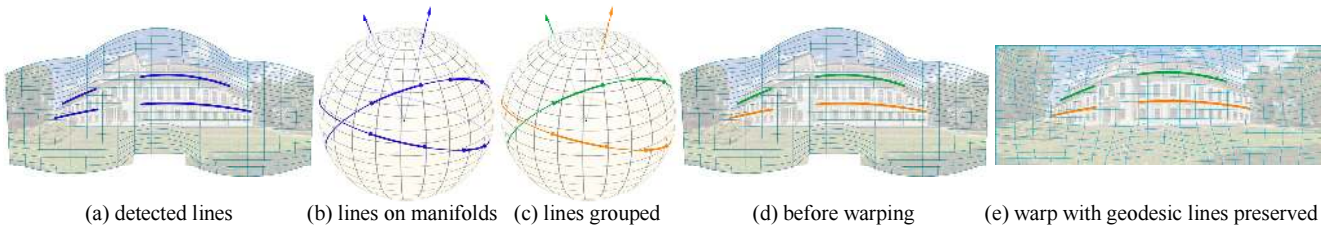
Preserving geodesic lines is a challenging task. Firstly, a geodesic line is a non-local geometric entity whose points can be far away from each other. So we need to use some non-local representations. Secondly, geodesic lines are complex nonlinear entities, and it is difficult to represent them via curve equations. To address these problems, we rely on 3-D planes that are linear and non-local. We treat a geodesic line as the intersection of a 3-D plane and the projection manifold. Then we constrain all points of a geodesic line to be on another plane after warping, so as to

[1]Perspective projection always preserves straight lines but only applies to narrow-angles $< 60^\circ$ [26, 18].

[2]A great circle is mathematically "geodesic" (shortest path), but an ellipse is not. For simplicity, in this paper we term them as "geodesic lines".

(a) input panorama (b) He *et al*.'s rectangling result (c) our rectangling result

Figure 2. An example of preserving straight lines *vs*. geodesic lines. He *et al*. [9] warp an irregularly shaped panorama to fit a rectangle, and preserve the straight lines detected on the projected image (a). This method will distort geodesic lines as in (b). Our method can preserve geodesic lines and produce a better result as in (c). On the bottom are the zoom-in regions.



(a) detected lines (b) lines on manifolds (c) lines grouped (d) before warping (e) warp with geodesic lines preserved

Figure 3. An overview of our approach. (a) Straight lines are detected on images before projection and stitching. These lines become curves (blue) on the projected image. (b) On the 3-D coordinate system, these lines are geodesic lines (great circles) on the manifold. (c) The geodesic lines are grouped according to their normal directions. We use one color to represent one group. (d) The image before warping with the grouped lines. (e) The image warped by our algorithm to fit a rectangle. (For illustrations, not all detected lines are displayed.)

preserve its geodesic appearance. We formulate this as an optimization problem.

We demonstrate our algorithm in various applications, including rectangling panoramas [9] (Fig. 2), resizing panoramic/wide-angle images, and wide-angle image manipulation [6]. Experiments show that our method is more flexible than the methods that preserve straight lines, and produces pleasing visual quality in challenging cases. We further extend our method to ellipse preservation in general (non-panoramic/wide-angle) warping tasks.

2. Related Work

Preserving Local Shapes. In the pioneer work of Igarashi *et al*. [10], an as-rigid-as-possible formulation is proposed for warping images. This method encourages each triangle in the mesh to undergo a "rigid" transform (rotation+shift). A Moving-Least-Squares method [15] generalizes it and supports affine/similarity transforms. Zhang *et al*. [24] derive a matrix form of as-similar-as-possible warping. This form produces a single linear system and is flexible to be combined with other energy terms. It has been applied in image retargeting [24, 7], video retargeting [20], video stabilization [13], rectangling panoramas [9], and content-aware rotation [8].

Preserving Straight Lines. Several methods [6, 5, 7, 9, 8] have been proposed for preserving straight lines. Though these methods have different formulations, they share some key components. Firstly, these methods cut a long straight line into small segments, such that each segment can be manipulated by a *local* mesh quad. Secondly, these methods introduce some *non-local* variables that relate these segments together. The rotational angle or its equivalence is a non-local variable used by these methods.

3. Our Approach

Spherical/cylindrical projections are commonly used projections [17]. They will map a straight line in the 3-D world into a geodesic line on the projecting manifold (Fig. 1). A 3-D straight line and the camera center form a 3-D plane. This plane intersects with the manifold (sphere/cylinder) and produces a geodesic line. In the 3-D coordinate systems, the geodesic lines are nonlinear curves - they are great circles or ellipses. In the 2-D coordinate systems on the projected manifolds, the curve equations are more complicated combinations of trigonometric functions.

Due to the nonlinearity, it is more challenging to preserve the geodesic lines than straight lines. We resort to a linear subspace as a guidance. Consider a group of points on a geodesic line. They lie on a common plane before warping. After warping, if they are on another common plane (and intersect the manifold), they can still produce another geodesic line. Because planes are linear subspaces, it is easier for us to manipulate them.

3.1. Overview

The overview of our warping algorithm is in Fig. 3. Here we use panorama rectangling [9] as an example warping application. First we detect the geodesic lines in the image (Fig. 3(a)). These lines are projected on the manifolds (Fig. 3(b)), and are clustered into groups (Fig. 3(c)). The line segments in each group are approximately in the same 3-D plane, and correspond to a consistent geodesic line. With the grouped lines, the image is warped by optimizing a grid mesh (Fig. 3(d,e)). The details are as follows.

3.2. Detecting and Grouping Geodesic Lines

We first detect the geodesic lines. We assume the projection coordinate system is known. For panoramas, the projection is used when stitching the source image sequence [17] and is known. We detect the straight lines in each source image (before projection) using the LSD method [19]. These straight lines are projected onto the stitched image and become detected geodesic lines (Fig. 3(a)). For wide-angle images, the projection can be given by camera calibration [25] and the rest is similar.

Next we group the geodesic lines, such that the lines in a group should be approximately on the same plane. We first cut all the geodesic lines into small segments, such that each segment is inside a mesh quad. For each segment, we find the 3-D coordinates of their two endpoints on the manifold, and compute their normal vector (Fig. 3(b)). Then we run k-means clustering on the set of all normal vectors. The resulting K k-means centers represent the normal vectors of the K planes (Fig. 3(c)). Each group approximately corresponds to a consistent geodesic line (Fig. 3(d)).

3.3. Energy for Preserving Geodesic Lines

The segments in the same group should have a non-local property, *i.e.*, they are expected to lie on a common plane after warping. This non-local property is given by the two rotation angles (θ, ϕ) that rotate one plane to another[3]. Next we present a warping energy function that only involves the non-local variables (θ, ϕ) and the mesh vertexes.

Consider a single segment with two endpoints $\hat{\mathbf{p}}_1$, $\hat{\mathbf{p}}_2$ before warping (Fig. 4(a)). $\hat{\mathbf{p}}_1$, $\hat{\mathbf{p}}_2$ are 3-D points and represented as 3×1 vectors. We use the camera center as the origin of the 3-D coordinates.

Assume a 3-D point $\mathbf{p}$ can be modeled by a transform $\mathcal{T}$ from $\hat{\mathbf{p}}_1$, $\hat{\mathbf{p}}_2$. The transform involves two parts. In the first part, it is shifted inside the plane spanned by the two vectors $\hat{\mathbf{p}}_1$, $\hat{\mathbf{p}}_2$ (see $\hat{\mathbf{p}}$ in Fig. 4(b)). If we use a 3×2 matrix $\hat{\mathrm{B}} = \{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2\}$ to denote the basis[4] of this plane, then $\hat{\mathbf{p}}$ can be written as $\hat{\mathrm{B}}\mathbf{s}$, where $\mathbf{s}$ is a 2×1 vector to be determined.

[3]Though a 3-D rotation should be fully represented by three Euler angles (θ, ϕ, ψ), the in-plane rotation ψ does not impact the coplanarity and thus need not be considered.

[4]These basis need not be orthogonal.

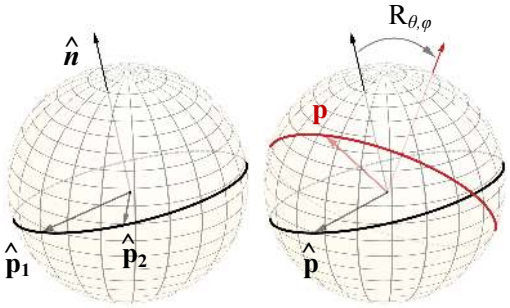


Figure 4. Left: $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$ are two end points of a geodesic line segment before warping. Right: $\hat{\mathbf{p}} = \hat{\mathrm{B}}\mathbf{s}$ is a shifted point on the same plane (black) as $\hat{\mathbf{p}}_1$ and $\hat{\mathbf{p}}_2$. This plane is rotated by $\mathrm{R}_{\theta,\phi}$ and mapped to another plane (red). Then $\hat{\mathbf{p}}$ becomes $\mathbf{p}$ after warping.

In the second part, the transform rotates this plane by some angles (θ, ϕ) (Fig. 4(b)). This rotation can be written as a 3-D rotation matrix $\mathrm{R}_{\theta,\phi}$ (see supplementary materials). Combining these two parts, the transform $\mathcal{T}$ is:

$$\mathcal{T}(\mathbf{s}, \theta, \phi) = \mathrm{R}_{\theta,\phi}\hat{\mathrm{B}}\mathbf{s}. \tag{1}$$

We define an energy to minimize the difference between a point $\mathbf{p}$ and its expected transformed position:

$$e(\mathbf{p}, \mathbf{s}, \theta, \phi) = \|\mathrm{R}_{\theta,\phi}\hat{\mathrm{B}}\mathbf{s} - \mathbf{p}\|^2. \tag{2}$$

Here $\mathbf{p}$ is the 3-D position (of $\mathbf{p}_1$ or $\mathbf{p}_2$) after warping and will be related to the mesh vertexes, and $\{\theta, \phi\}$ are *non-local* variables that are shared by all the segments in the same group.

We first minimize Eqn.(2) w.r.t. $\mathbf{s}$ and obtain:

$$\mathbf{s} = (\hat{\mathrm{B}}^{\mathrm{T}}\hat{\mathrm{B}})^{-1}\hat{\mathrm{B}}^{\mathrm{T}}\mathrm{R}_{\theta,\phi}^{\mathrm{T}}\mathbf{p} \tag{3}$$

This shows a nice property that $\mathbf{s}$ is a linear function of $\mathbf{p}$. Substituting $\mathbf{s}$ into Eqn.(2) we obtain:

$$e(\mathbf{p}, \theta, \phi) = \|\mathrm{C}_{\theta,\phi}\mathbf{p}\|^2, \tag{4}$$

with a matrix $\mathrm{C}_{\theta,\phi}$ defined as:

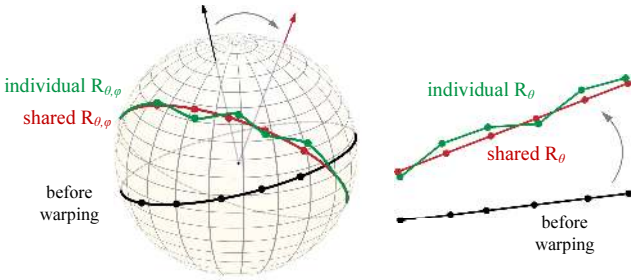
$$\mathrm{C}_{\theta,\phi} \triangleq \mathrm{R}_{\theta,\phi}\hat{\mathrm{B}}(\hat{\mathrm{B}}^{\mathrm{T}}\hat{\mathrm{B}})^{-1}\hat{\mathrm{B}}^{\mathrm{T}}\mathrm{R}_{\theta,\phi}^{\mathrm{T}} - \mathrm{I}, \tag{5}$$

where I is a unit matrix. $e(\mathbf{p}, \theta, \phi)$ is quadratic on $\mathbf{p}$.

Given all segments clustered into K groups, the energy E_G for preserving all geodesic lines is:

$$E_G(\{\mathbf{p}\}, \{\theta_k, \phi_k\}) = \frac{1}{L}\sum_{k=1}^{K}\sum_{l\in\mathcal{G}(k)}\sum_{i=1,2}\|\mathrm{C}_{\theta_k,\phi_k}\mathbf{p}_{l,i}\|^2. \tag{6}$$

Here L is the number of segments, $\mathbf{p}_{l,i\ (i=1,2)}$ are the two end points in a segment l, and l belongs to the k-th group



(a) Preserve geodesic lines (b) Preserve straight lines

Figure 5. Use *non-local* properties to preserve lines. (a) Preserve geodesic lines. The non-local properties are the 3-D rotation $\mathrm{R}_{\theta,\phi}$. If the points after warping have inconsistent θ, ϕ, the geodesic line is distorted (green). (b) Preserve straight lines [7, 9]. The non-local properties are the 2-D rotation R_θ. If the line segments have inconsistent θ, the straight line is distorted (green).

$\mathcal{G}(k)$. The notations θ_k and ϕ_k imply that the rotation angles are shared by the segments in the group k, such that these segments are expected on the same plane after warping. So $\{\theta_k, \phi_k\}$ are non-local variables of the group k.

Fig. 5(a) illustrates the effect of the non-local variables. The black points are in the same plane before warping. If they share the same rotation $\{\theta, \phi\}$, they will be on the same plane after warping. Then they still present as a geodesic line (Fig. 5(a) red). If the points have individual rotation angles, they are not on the same plane and the geodesic line is distorted (Fig. 5(a) green).

The above derivation is related to the methods for preserving 2-D straight lines as in [7, 9]. In the 2-D cases, the straight line segments are grouped into bins. The segments on the same straight line (Fig. 5(b) black) are in the same bin. The nonlocal variable is a 2-D rotation angle θ. If these segments share the same θ, then they still have the same orientations after rotation. Thus the line is still straight (Fig. 5(b) red). If the segments have individual rotation angles, the line will be distorted (Fig. 5(b) green). Our method introduces linear planes as guidance, and generalizes this behavior to 3-D.

3.4. The Energy Function

Next we incorporate the geodesic-preserving energy Eqn.(6) in a warping energy. We consider quad meshes in this paper. The vertexes are denoted as $\{\mathbf{v}_j\}$ with each $\mathbf{v}_j = (u_j, v_j)$ as 2-D coordinates. Denote all the vertexes by a vector $\mathbf{V}$. The warping energy is defined as:

$$E(\mathbf{V}, \{\theta_k, \phi_k\}) = \lambda_B E_B(\mathbf{V}) + \lambda_S E_S(\mathbf{V}) + \lambda_G E_G(\mathbf{V}, \{\theta_k, \phi_k\}). \quad (7)$$

Here E_B is a boundary-preserving term, E_S is a shape-preserving term, and E_G is a geodesic-preserving term. We

Algorithm 1 Optimization

1: Initialize $\mathbf{V}$.
2: **for** iter $= 1$ to iter_{max} **do**
3: Fix $\mathbf{V}$, update θ_k, ϕ_k in each group k.
4: Fix all $\{\theta_k, \phi_k\}$, do:
5: **for** $t = 1$ to t_{max} **do**
6: Solve a linear system on $\mathbf{V}_t$ with $\mathbf{V}_{t-1}$ fixed.
7: **end for**
8: **end for**

set $\lambda_B = 10^8$ for hard constraints, and set $\lambda_S = 1$ and $\lambda_G = 100$. The details of the terms are as below.

$E_B(\mathbf{V})$ is defined on the boundary vertexes to represent boundary constraints, as required in image resizing [7] and panorama rectangling [9]. It has a quadratic form as $E_B(\mathbf{V}) = (\mathbf{V} - \hat{\mathbf{V}})^{\mathrm{T}} \mathrm{D} (\mathbf{V} - \hat{\mathbf{V}})$. Here D is a diagonal matrix whose diagonal entry is 1 for a boundary vertex and 0 otherwise. $\hat{\mathbf{V}}$ is the pre-defined boundary constraints (usually a rectangle).

$E_S(\mathbf{V})$ is a term that preserves local shapes. We adopt the form used in [24, 7, 9] that expects the input/warped quads are "as-similar-as-possible": $E_S(\mathbf{V}) = \mathbf{V}^{\mathrm{T}} \mathrm{L} \mathbf{V}$ where L is a Laplacian matrix as derived in [24] (see also supplementary materials).

The geodesic-preserving term E_G is due to Eqn.(6). When defined on the vertexes $\mathbf{V}$, it has the following form:

$$E_G(\mathbf{V}, \{\theta_k, \phi_k\}) = \frac{1}{L} \sum_{k=1}^{K} \sum_{l \in \mathcal{G}(k)} \sum_{i=1,2} \| \mathrm{C}_{\theta_k, \phi_k} \mathbf{f}(\mathrm{M}_{l,i} \mathbf{V}) \|^2, \quad (8)$$

where $\mathrm{C}_{\theta_k, \phi_k}$ is as in (5). Here the matrix $\mathrm{M}_{l,i}$ denotes the coefficients used to bilinearly interpolate an end point of a segment from its nearby four vertexes. This matrix only depends on the input vertexes and the input position of the $(l,i)^{\text{th}}$ end point, and so can be precomputed. The reverse projection function $\mathbf{f}$ projects any point on the 2-D manifold back to 3-D. Its form depends on the projections [18]: for spherical projections $\mathbf{f}(u, v) = (\sin u \cos v; \sin v; \cos u \cos v)$ and for cylindrical projections $\mathbf{f}(u, v) = (\sin u; v; \cos u)$. The function $\mathbf{f}$ can also be other projections, *e.g.*, a lens distortion function obtained by camera calibration [25].

3.5. Optimization

We adopt an iterative algorithm to minimize Eqn.(7). We fix $\{\theta_k, \phi_k\}$ and update $\mathbf{V}$, and vice versa. The two subproblems are both non-linear.

Fix $\{\theta_k, \phi_k\}$, update $\mathbf{V}$. This subproblem is nonlinear on $\mathbf{V}$ because of the function $\mathbf{f}$ in Eqn.(8). We adopt the Gauss-

Newton method. We expand $\mathbf{f}(\mathrm{M}_{l,i}\mathbf{V})$ in (8) as:

$$\mathbf{f}(\mathrm{M}_{l,i}\mathbf{V}_t) \approx \mathbf{f}'(\mathrm{M}_{l,i}\mathbf{V}_{t-1})\mathrm{M}_{l,i}(\mathbf{V}_t - \mathbf{V}_{t-1}) + \mathbf{f}(\mathrm{M}_{l,i}\mathbf{V}_{t-1}), \tag{9}$$

where $\mathbf{f}'$ is the derivative (Jacobian matrix) of $\mathbf{f}$, $\mathbf{V}_{t-1}$ is the solution in iteration $t-1$, and $\mathbf{V}_t$ is the solution in iteration t. With $\mathbf{V}_{t-1}$ fixed, the energy is a quadratic on $\mathbf{V}_t$ and can be solved by a linear system. This procedure is iterated.

Fix V, update $\{\theta_k, \phi_k\}$. In this case, we can independently estimate θ_k, ϕ_k for each group k:

$$\min_{\theta_k,\phi_k} \sum_{l\in\mathcal{G}(k)} \sum_{i=1,2} \|\mathrm{C}_{\theta_k,\phi_k}\mathbf{f}(\mathrm{M}_{l,i}\mathbf{V})\|^2. \tag{10}$$

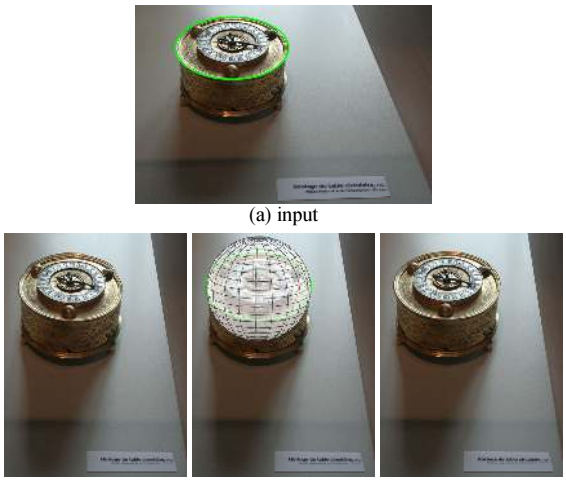
Intuitively, this is a regression problem of fitting a plane, such that all the vectors $\mathbf{p}_{i,k} \triangleq \mathbf{f}(\mathrm{M}_{l,i}\mathbf{V})$ are expected to be on this plane. We propose a simple solution based on this intuition. Suppose $\hat{\mathbf{n}}_k$ is the normal vector of the input plane (*i.e.*, before warping) corresponding to group k. We randomly select two points $\mathbf{p}_{1,k}$ and $\mathbf{p}_{2,k}$ in the group k. The cross product $\mathbf{p}_{1,k} \times \mathbf{p}_{2,k}$ will give us a normal vector orthogonal to the plane spanned by these two points. It is easy to compute the angles (θ_k, ϕ_k) that rotates $\hat{\mathbf{n}}_k$ to this new normal vector. These values of (θ_k, ϕ_k) are a candidate solution to (10). Actually, this is a solution to a single term $\sum_{i=1,2} \|\mathrm{C}_{\theta_k,\phi_k}\mathbf{f}(\mathrm{M}_{l,i}\mathbf{V})\|^2 = 0$. We randomly select 100 pairs of ($\mathbf{p}_{1,k}$, $\mathbf{p}_{2,k}$) and obtain 100 candidates solutions. Then we put each candidate solution into (10) and evaluate the energy. The candidate that gives the smallest energy will be chosen. In practice we find this simple solution effectively reduces the energy.

The optimization is described in Algorithm 1. To initialize, we solve a linear system on $\mathbf{V}$ by ignoring the term E_G in (7). The iteration numbers are fixed as $\mathrm{iter}_{\max}$=10 and $t_{\max}$=10. The size of the optimization problem is quite small. We use a mesh with around 400 vertices, so $\mathbf{V}$ is 800-dimensional. The angles $\{\theta_k, \phi_k\}$ contribute K pairs of variables, where the group number K depends on image content. In our implementation we initialize 200 clusters, and remove empty clusters during the k-means iterations. This usually leaves $K \sim 10^2$. The optimization in Algorithm 1 on such a small size is efficient, taking < 0.2 seconds in our C++ implementation.

3.6. Preserving Ellipses in General Image Warping

In above we focus on panoramic/wide-angle images. Next we extend our method to ellipse preservation in warping general (non-panoramic/wide-angle) images.

We first require the user to mark an ellipse to be preserved in the input image. Take image resizing for example. We apply an existing resizing method (*e.g.*, [7]) to preliminarily warp the image to the desired size. The sample points on the user-marked ellipse are warped to their new positions, which may not create a new ellipse if without preservation (Fig. 6(b)). Then we fit (using least squares) a new ellipse using the warped sample points, and create a virtual 3-D sphere whose cross section is the fitted ellipse (Fig. 6(c)).



(a) input

(b) line-preserving (c) virtual sphere (d) ellipse-preserving

Figure 6. Preserving ellipses in general image resizing. (a) Input and a user-marked ellipse. (b) Retargeting (50% width) with line-preserving only. (c) A virtual sphere fitted to guide ellipses. (d) Our ellipse-preserving resizing.

We expect the new ellipse to be an intersection of this sphere and a plane through the center of it. This can be easily modeled as in our above derivations. We only need to modify the projection function $\mathbf{f}$ in Eqn.(8). For an upright sphere, we use $\mathbf{f}$ in this form:

$$\mathbf{f}(u,v) = (u-u_0; v-v_0; \pm\sqrt{r^2-(u-u_0)^2-(v-v_0)^2}). \tag{11}$$

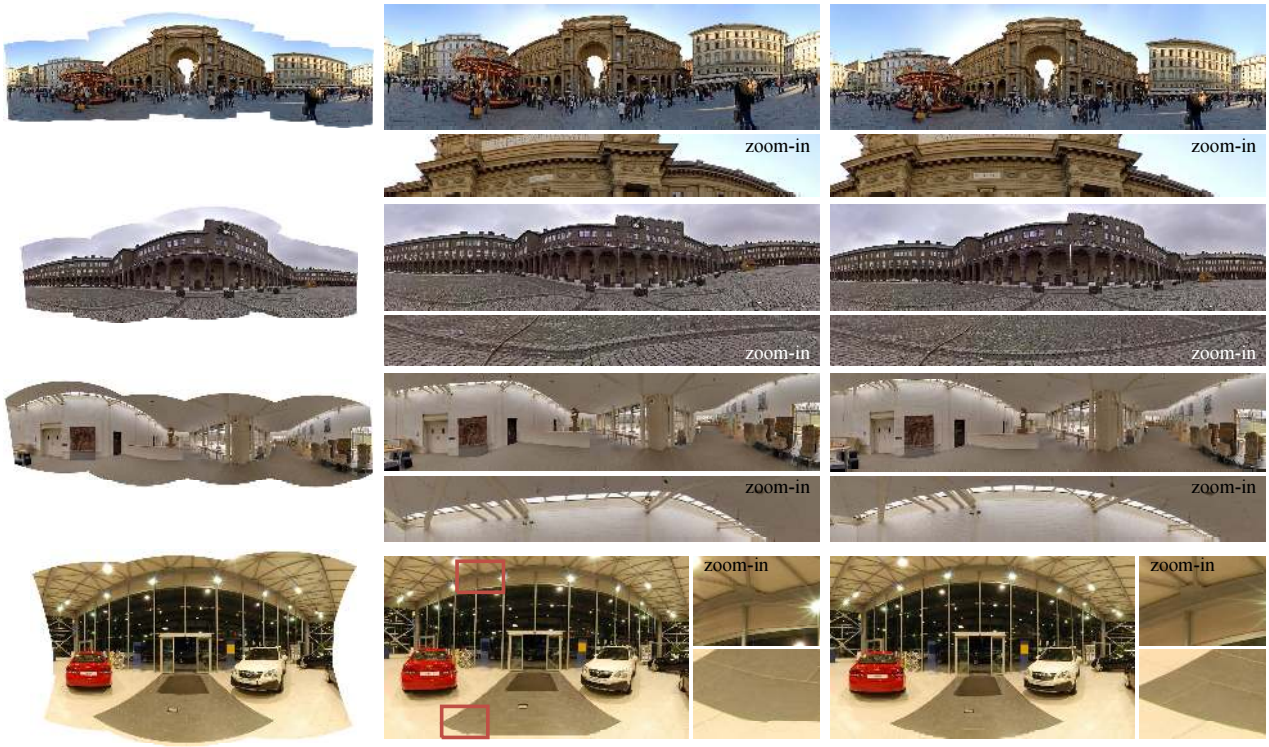
Here (u_0, v_0) is the center of the virtual 3-D sphere in the image plane, and r is its radius. The sign $\pm$ is given by the sign of $v-v_0$. Intuitively, this function $\mathbf{f}$ maps a point in the image plane back onto the surface of the sphere. With this $\mathbf{f}$, we optimize the energy in (7) and warp the image again to preserve the ellipse (Fig. 6(d)).

4. Results and Applications

4.1. Rectangling Panoramas

This is an application proposed in [9]. The purpose is to warp an irregularly shaped panorama image and fit it to a rectangle. The boundary constraint E_B in Eqn.(7) is given by the rectangle boundary. In [9] the straight lines in the projected panorama are detected and preserved.

Fig. 2 and Fig. 7 show the comparisons between He *et al.*'s and our method. The results of He *et al.* are provided by the authors. Though He *et al.*'s method is able to preserve straight lines, it distorts the geodesic lines (see the zoom-in in Fig. 7). It often treats geodesic lines as piecewise linear curves. In contrast, our algorithm manages to



(a) input panorama (b) He *et al.*'s rectangling (c) our rectangling

Figure 7. Rectangling panoramas [9]. (a) Input panoramas. (b) He *et al.*'s results [9]. (c) Our results.

(a) input panorama (b) preserving local smoothness (c) our geodesic-preserving

Figure 8. Local-smoothness-preserving *vs*. Geodesic-preserving. (a) Input panorama. (b) Rectangling result of only preserving the local smoothness of the geodesic lines. (c) Rectangling result of preserving the geodesic lines.

preserve the geodesic lines. The zoom-in windows in Fig. 2 and Fig. 7 demonstrate that the geodesic lines are non-local geometric entities: they should remain consistent on a large area of the image.

Geodesic Lines *vs*. Smooth Curves. The geodesic lines are not simply locally smooth curves but non-local entities. A key component in our method is the usage of the non-local variables (θ_k, ϕ_k). To show the importance of the non-local property, we implement a curve-preserving method that preserves the local smoothness of the curve. In this method, the adjacent segments on a curve (in our case a geodesic line) are encouraged to have similar directions, so as to preserve the local smoothness of the curve. The technical details of this alterative can be found in the supplementary materials. Fig. 8 shows the comparison between our geodesic-preserving method and the local-smoothness-preserving method. In Fig. 8(b), the long geodesic lines still appear wiggled, because they are not treated as non-local entities. In Fig. 8(c), our result does not have these artifacts.

4.2. Wide-angle/Panoramic Image Resizing

Content-aware image resizing or retargeting is a widely studied topic [4, 21, 24, 7]. However, little attention has been paid to the case of resizing wide-angle/panoramic images. We point out this is a practical scenario of resizing, because these images are often taken with special lens or photographic techniques, and thus exhibit a larger variety of aspect ratios. These images almost inevitably present geodesic lines, so our technique is desired.

In Fig. 9 we compare with Chang and Chuang's method [7], a resizing technique for preserving straight lines. For fair comparisons, we adopt the same saliency map as in [7] that weights the shape-preserving terms in Eqn.(7). As such, our method only differs from their method in the ways of addressing lines. Fig. 9 shows that preserving

Figure 9. Resizing wide-angle (top, by 2× width) and panoramic (bottom, by 50% width) images. (a) Input. (b) Uniform scaling. (c) Chang and Chuang's [7] line-preserving retargeting. (d) Our results.

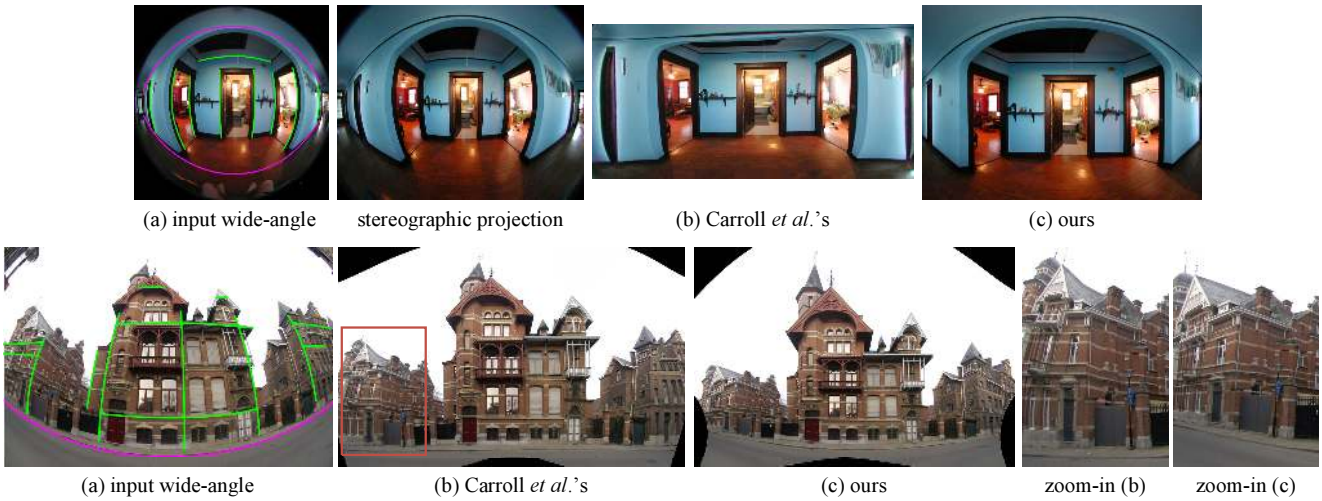


Figure 10. Wide-angle image manipulation [6]. (a) Input images with user-specified lines. (b) Carroll *et al.*'s results obtained from their paper [6]. This method attempts to keep all these lines straight. (c) Our method treats the long-spanned lines (magenta) as geodesic lines, and preserve the rest as straight lines. The results are manually cropped for display.

straight lines in wide-angle/panoramic images is not sufficient. The straight-line-preserving method treats geodesic lines as piece-wise linear curves. Our solution can better handle these images.

4.3. Wide-angle Image Manipulation

Carroll *et al*. [6] propose an interactive tool of manipulating wide-angle images - the user marks several lines in the wide-angle image, and these lines are straightened (Fig. 10(a)). But as Carroll *et al*. indicate in the limitation of their work, distortion is unavoidable if the long straight lines span very wide angles like $\sim$180°, such as the magenta lines in Fig. 10(a). In this case, the image content would be severely distorted (Fig. 10(b)).

Our method is more flexible than preserving straight lines. In this interactive case, we use user interactions to specify lines that are preserved as straight (green in Fig. 10(a)) or as geodesic (magenta in Fig. 10(a)). To preserve straight lines, we incorporate the straight-line-preserving term of [9] into our energy function. This introduces a new set of variables (2-D rotation angles $\{\theta'\}$ of the straight lines) besides $\mathbf{V}$ and $\{\theta_k, \phi_k\}$ in our energy Eqn.(7). To optimize this new energy, we simply iterate among these three sets of variables, fixing two and updating

(a) input (b) scale 50% width (c) Chang and Chuang's line-preserving (d) our ellipse-preserving

Figure 11. Images resizing with ellipse preservation. (a) Input image and user-marked ellipses. (b) Uniform scaling. (c) Chang and Chuang's [7] line-preserving retargeting. (d) Our result with ellipse preservation.

(a) input and user interactions (b) line-preserving only (c) our ellipse-preserving (d) zoom-in

Figure 12. Interactive perspective manipulation [5]. (a) Input and user interaction (more details in supplementary materials). (b) Warping for perspective manipulation with line-preserving only. (c) Our ellipse-preserving results. (d) Zoom-in. The images are manually cropped for display.

input panorama our rectangling result

Figure 13. A limitation. Our method cannot preserve an arbitrary curve that is not a geodesic line.

the rest one. Fig. 10(c) shows our result. Our method manages to preserve the long-spanned lines as geodesic lines, and straighten the other lines.

4.4. Ellipse Preservation

Our method can be extended to preserve ellipses in general image warping. Fig. 6 and 11 show examples of image resizing. Fig. 11(c) shows Chang and Chuang's [7] line-preserving result. Without ellipse preservation, this method distorts the glasses. Fig. 11(d) shows our result that preserves the ellipses.

In Fig. 12 we show examples of warping-based *perspective manipulation* proposed in [5] - the user marks several perspective lines and vanishing points, and manipulates them to change the perspective appearance. The image content is warped subject to the user-marked constraints. The warping method may lead to noticeable distortion if ellipses present (Fig. 12(b)). Our ellipse-preserving method remedies this problem (Fig. 12(c)).

5. Limitations and Conclusions

Our method is not a solution to preserving general curves. The geodesic lines and ellipses have special properties: they are curves generated by the intersections of manifolds. Our method may fail if a curve is really from a curved object. Fig. 13 shows a failure example. The geodesic line preservation term does not take effect on this curve. More failure images are in the supplementary materials.

Our algorithm requires to know the projections of the wide-angle/panoramic images. This was also required, *e.g.*, in [6]. As future work, we will study estimating the projection only from the projected image.

References

[1] GoPro cameras. http://gopro.com/.

[2] iPhone 5s tips and tricks: panorama. https://www.apple.com/iphone-5s/tips/.

[3] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. In *ACM SIGGRAPH '06*, pages 853–861, 2006.

[4] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. In *ACM SIGGRAPH '07*, 2007.

[5] R. Carroll, A. Agarwala, and M. Agrawala. Image warps for artistic perspective manipulation. In *ACM SIGGRAPH '10*, 2010.

[6] R. Carroll, M. Agrawal, and A. Agarwala. Optimizing content-preserving projections for wide-angle images. In *ACM SIGGRAPH '09*, 2009.

[7] C.-H. Chang and Y.-Y. Chuang. A line-structure-preserving approach to image resizing. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] K. He, H. Chang, and J. Sun. Content-aware rotation. In *International Conference on Computer Vision (ICCV)*, 2013.

[9] K. He, H. Chang, and J. Sun. Rectangling panoramic images via warping. In *ACM SIGGRAPH '13*, 2013.

[10] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *ACM SIGGRAPH '05*, 2005.

[11] J. Kopf, D. Lischinski, O. Deussen, D. Cohen-Or, and M. Cohen. Locally adapted projections to reduce panorama distortions. In *Computer Graphics Forum*, pages 1083–1089. Wiley Online Library, 2009.

[12] J. Kopf, M. Uyttendaele, O. Deussen, and M. F. Cohen. Capturing and viewing gigapixel images. In *ACM SIGGRAPH '07*, 2007.

[13] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. In *ACM SIGGRAPH '13*, 2013.

[14] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1144–1154, 2000.

[15] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM SIGGRAPH '06*, 2006.

[16] J. Snyder. Flattening the earth: two thousand years of map projections. *University of Chicago, Chicago*, 1993.

[17] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2006.

[18] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *ACM SIGGRAPH '97*, 1997.

[19] R. von Gioi, J. Jakubowicz, J. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 722–732, 2010.

[20] Y.-S. Wang, H.-C. Lin, O. Sorkine, and T.-Y. Lee. Motion-based video retargeting with optimized crop-and-warp. In *ACM SIGGRAPH '10*, 2010.

[21] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. In *ACM SIGGRAPH Asia '08*, 2008.

[22] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. As-projective-as-possible image stitching with moving dlt. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[23] L. Zelnik-Manor, G. Peters, and P. Perona. Squaring the circle in panoramas. In *International Conference on Computer Vision (ICCV)*, 2005.

[24] G. Zhang, M. Cheng, S. Hu, and R. Martin. A shape-preserving approach to image resizing. In *Computer Graphics Forum*, pages 1897–1906. Wiley Online Library, 2009.

[25] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2000.

[26] D. Zorin and A. H. Barr. Correction of geometric perceptual distortions in pictures. In *ACM SIGGRAPH '95*, 1995.