# Optimizing Content-Preserving Projections for Wide-Angle Images 



Figure 1: Wide-angle photographs can appear badly distorted under existing projections, such as the perspective, Mercator and stereographic projections. Perspective projection preserves linear structures in the scene, but distorts shapes of objects. Mercator and stereographic projections preserve shapes locally, but bend linear structures. Our projection is designed to both preserve local shape and maintain straight scene lines that are marked by the user with our interactive tool.


#### Abstract

Any projection of a 3D scene into a wide-angle image unavoidably results in distortion. Current projection methods either bend straight lines in the scene, or locally distort the shapes of scene objects. We present a method that minimizes this distortion by adapting the projection to content in the scene, such as salient scene regions and lines, in order to preserve their shape. Our optimization technique computes a spatially-varying projection that respects user-specified constraints while minimizing a set of energy terms that measure wide-angle image distortion. We demonstrate the effectiveness of our approach by showing results on a variety of wide-angle photographs, as well as comparisons to standard projections.


## 1 Introduction

When we view the world with our eyes, we experience a wide field of view without any obvious distortion. In our mind's eye, objects retain their natural shape, and linear structures remain straight. Yet, creating a flat image of such a wide field of view that resembles our subjective perception is remarkably challenging.
The visual information seen from a single viewpoint is defined on a viewing sphere centered at that viewpoint. To flatten the viewing sphere we must map it onto an image plane. Yet, a sphere is a nondevelopable surface and therefore any such mapping will introduce some perceptual distortions [Zorin and Barr 1995]. When the field of view is limited it covers a small portion of the viewing sphere and these distortions are not readily apparent - for this reason photography manuals often recommend using a $40^{\circ}$ field of view for everyday snapshots. In contrast, the distortions are much more apparent in wide-angle photographs. Under any standard projection

[^0]of the viewing sphere onto the image plane objects and structures look unnaturally stretched, sheared or curved (Figure 1).
Nevertheless, wide-angle images can provide a vivid visual impression of a scene. Recent techniques for capturing and automatically stitching together multiple images of a scene [Szeliski and Shum 1997; Brown and Lowe 2003; Kopf et al. 2007] have enabled photographers at all skill levels to produce high-resolution wide field of view panoramas. At times it is necessary for photographers to use a wide-angle or fisheye lens to capture a scene; in tight spaces, such as architectural interiors, it may be impossible to move the camera far enough away from the subject of interest to use a lens with a narrower field of view. The result is that despite their problems photo sharing sites such as Flickr (www.flickr.com) contain hundreds of thousands of wide-angle and panoramic images. Yet, the vast majority of these wide-angle images deviate significantly from the way we would perceive the corresponding real-world scene.
Cartographers have long recognized that mapping a sphere to plane is difficult and over thousands of years have developed hundreds of projections that trade off different types of distortion [Snyder 1993], though none can avoid all distortions. Artists have also developed specialized techniques for handling large fields of view [Kubovy 1986; Flocon and Barre 1988]. Treatises on perspective often suggest deviating from classical perspective when drawing humans in wide-angle scenes and drawing them freehand instead [Longfellow 1901]. The key idea in this artistic approach is to locally adapt the projection based on the content of the image.
In this paper we present a new content-based approach for mapping a wide-angle photograph defined on the viewing sphere into a flat image such that the resulting image appears more natural - more like our perception of the real-world scene. Whether a depiction is "natural" is, of course, subjective and impossible to validate. Therefore, like previous work [Zorin and Barr 1995; Zelnik-Manor et al. 2005], our solution attempts to minimize the distortions that viewers most frequently complain about in wide-angle images - namely, curving of straight lines in the scene, and non-uniform deformations of objects. Like the artists, our solution locally adapts the mapping from the sphere to the image plane based on image content.
Our solution provides a graphical interface for users to mark the perceptually important lines in the image and automatically applies a low-level salience measure and a face detector [Viola and Jones 2004] to identify salient objects in the scene. Our solution then op-
timizes the mapping from the viewing sphere to the image plane such that the shapes of the salient objects are preserved and the marked lines remain straight. Although our approach requires some interaction (users must draw about 20 lines on average), in practice we have found it relatively simple to mark perceptually important lines. Moreover, by interactively specifying lines that should not be bent, users can better control the look of the final image.
Our primary contribution in this paper is an optimization approach for computing a spatially-varying mapping from the viewing sphere to a planar image that respects the user-specified constraints while minimizing a set of energy terms that measure wide-angle image distortion. We demonstrate the power of this approach with a number of results created from fisheye and panoramic inputs. We also compare the results of our technique to those of previous work.

### 1.1 Related work

Zorin and Barr [1995] proved that no global projection can both keep straight lines straight and preserve the shapes of objects. They propose a one-parameter family of projections that trades off between these two conditions by interpolating between a perspective and stereographic [Snyder 1987] projection. More recently, ZelnikManor et al. [2005] address the same problem by stitching together multiple linear perspective projections into a "multi-plane" projection; the seams are placed by the user where they are least noticeable, e.g., the edge between two walls. However, any object that passes over such a seam will have a sharp discontinuity. Their method is limited to scenes where such a seam can be well hidden and where the field of view is wide only vertically or horizontally, since their projection can only change in one direction. Our method improves upon these techniques with a locally-varying mapping designed to minimize local shape distortion and line bending where they matter in the image. That is, we do not attempt to keep all possible lines straight, only salient lines that appear in the image.
There is also commercial software for improving wide-angle images, such as DxO Optics Pro (dxo.com) and Fisheye Hemi (imagetrendsinc.com). As far as we can tell, the former allows the user to interpolate between perspective and either a cylindrical or stereographic projection, while the latter produces a cylindrical projection. In contrast, our method produces a spatially-varying mapping. Kopf et al. [2007] present a dynamic panoramic viewer that smoothly animates between a cylindrical projection when the viewer is zoomed-out, and perspective projection when the viewer is zoomed-in. We focus on creating a good static image.
Our algorithm performs a least squares optimization that, along with other conflicting constraints, tries to maintain a locally conformal mapping. Recent shape deformation techniques use a similar approach to maintain a locally conformal or rigid deformation in the face of user-specified point constraints [Igarashi et al. 2005; Schaefer et al. 2006]. Gal et al. [2006] combine this approach with user-specified annotations identifying image regions which should remain undistorted. Similarly, Wolf et al. [2007] and Wang et al. [2008] use least squares optimizations to resize images while minimizing distortion of salient content. Previously, salience metrics have also been used for automatic cropping [Suh et al. 2003] and for discrete approaches to content-aware retargeting [Avidan and Shamir 2007]. Our method differs from retargeting in that our mapping is performed from the spherical domain, and incorporates a novel set of constraints specific to wide-angle images.
Like our method, mesh parameterization techniques compute a mapping from a non-developable surface (usually a 3D triangle mesh) to a planar domain while minimizing distortion [Sheffer et al. 2006]. For example, Lévy et al. [2002] use the discrete CauchyRiemann equations to find a conformal mapping with free boundaries. Similarly, cartographers compute a mapping from a sphere to a plane that minimizes perceivable distortions [Snyder 1993].

Cartographers have also used least squares optimization to compute spatially-varying projections that adapt to a specific shape, e.g., a continent [Snyder 1985]. However, the metrics that define good mesh parameterizations and maps are different from those that produce effective wide-angle images. For example, the need to preserve straight lines is unique to wide-angle imaging.

## 2 The space of wide-angle projections

What properties are desirable in a wide-angle image? One way to approach this question is through the concept of perceptual constancy [Palmer 1999]. Constancy refers to the idea that our visual experience often more closely correlates with the intrinsic properties of an object rather than the extrinsic properties of an object's appearance on our retina. For example, size constancy refers to our ability to perceive the true size of an object despite its projected size on our eyes. We perceive the reflection of our face in a mirror as equal to the true size of our face, even though the image on the mirror is half its true size. Other perceptual constancies include shape and orientation. The best depictions of objects usually maintain such perceptual constancies and are consistent with the intrinsic properties of the objects [Zeki 2001; Durand 2002]. Based on these perceptual constancies, wide-angle projections should maintain the following properties:
Shape constancy (local). Projections should maintain the shapes of objects as they appear on the viewing sphere, which is how they would appear if one looked directly at them. For example, spheres in the world should appear as circles in the image. Conformal projections maintain this property locally.
Shape constancy (global). Projections should avoid bending global shapes. To prevent the appearance of bending, linear structures in the scene should project to straight lines in the image. In addition the scale and orientation of the projection should not vary too quickly.
Orientation constancy. Cameras are often oriented so that their upvectors are parallel to vertical linear structures in the scene. In such cases, vertical scene lines should remain vertical in the projected image. Similarly, scene lines that are perpendicular to both the optical axis of the camera and its up-vector should appear horizontal in the projected image.

It is well-known that there is no wide-angle projection that can simultaneously preserve all of these properties [Snyder 1993; Zorin and Barr 1995]. Thus, cartographers have developed hundreds of projections that preserve various subsets of these properties [Snyder 1993]. We consider the properties of perspective projection, which is the most common in photography, as well as two alternative cartographic projections, Mercator and stereographic (Figure 1).

Perspective. To form a perspective image, the viewing sphere is projected onto a tangent plane through lines emanating from the center of the sphere. The vast majority of photographs are captured with lenses that approximate linear perspective because this projection preserves several desirable properties. Lines in the scene remain straight, and when the camera is held parallel to the ground plane the orientation constancy of vertical scene lines is maintained. However, perspective projections are not conformal and objects located near the periphery of wide-angle perspective images can appear unnaturally stretched and distorted [Kubovy 1986]. As the field of view approaches $180^{\circ}$ the stretching becomes infinite. Nevertheless, perspective projection is considered robust for fields of view less than about $40^{\circ}$ as it only weakly violates conformality when the field of view is limited. At such limited fields of view the images look "correct" even when the viewer is not located at the center of projection [Vishwanath et al. 2005].
Mercator. The Mercator projection is a cylindrical projection that
is designed to maintain conformality. Thus, it locally preserves the shape of objects. As in all cylindrical projections, meridians of the viewing sphere are mapped to vertical lines in the image. Thus, when the camera is held parallel to the ground plane vertical scene lines will appear vertical in the image. The Mercator projection can handle a complete $360^{\circ}$ horizontal field of view, but stretches to infinity as the vertical field of view approaches $180^{\circ}$. The Mercator projection is often useful for panoramic images with large horizontal fields of view, such as images containing large groups of people.
Stereographic. To form a stereographic image, the viewing sphere is projected onto a tangent plane through lines emanating from the pole opposite the point of tangency. Stereographic projections are also conformal and therefore preserve the shapes of objects locally. In addition, stereographic projection preserves scene lines that pass through the center of the image. Stereographic images appear similar to those produced by fisheye lenses as well as wide field-ofview illustrations drawn using curvilinear perspective [Flocon and Barre 1988]. Like perspective projection, stereographic projection stretches objects toward the periphery; however, the objects are scaled in a conformal manner.

This analysis suggests that different projections are useful for different scenes. In fact, different projections may be useful within the same scene to emphasize different objects and features within it. Our solution is therefore designed to allow a user to explore the design space of wide-angle projections. Users can specify highlevel properties they wish to maintain in the projected image. Our system then optimizes the projection across the image, adapting it to the content in the image in order to preserve those properties and produce more effective depictions of the scene.

## 3 Our approach

We have developed a content-preserving approach for projecting wide-angle photographs from the viewing sphere onto a flat image. Our solution is comprised of two components; a simple user interface that allows users to control the appearance of the output image by specifying the properties that should be preserved (Section 3.1), and a weighted least-squares optimization procedure for generating the spatially-varying mapping from the viewing sphere to the image plane (Sections 3.2-3.7).

### 3.1 User interface

Users can load wide-angle images from any source (e.g. wide-angle perspective lens, fisheye lens, panoramic mosaic, etc.) into our interface as long as the mapping from the source data to the viewing sphere is known. We have pre-built many of these mappings into our system and in most cases the user simply specifies the type of input image (fisheye, cylindrical panorama, etc.) and its field of view. We use the standard projection models for these types of inputs and ignore the issues of lens distortion. Lens distortion has not proven to be a problem for the examples we have tried.
The main task for the user is to identify linear scene structures in the input image that should remain straight in the output. Since only perspective projection can maintain all scene lines as straight, this input allows the algorithm to focus on preserving only the lines that the user considers important. We also considered allowing the user to mark salient regions; however, we never found this input to be necessary, since the algorithm can typically find near-conformal solutions for most sets of line constraints.
Users can directly specify two types of line constraints in our interface. The general line constraint is designed to keep linear structures in the scene from bending in the output image. The fixed orientation line constraint modifies the general constraint so that the linear scene structures map to straight lines at a user-specified


Figure 2: We optimize a mapping from the viewing sphere parametrized by $(\lambda, \phi)$ to a plane parametrized by $(u, v)$. For each vertex $\lambda_{i, j}=\left(\lambda_{i, j}, \phi_{i, j}\right)$ on the sphere we calculate its projection $\mathbf{u}_{i, j}=\left(u_{i, j}, v_{i, j}\right)$ on the image plane.
orientation in the output image. Users specify both of these constraints by identifying linear scene structures in the input image. All straight lines in the scene map to great circles on the viewing sphere. Although the linear structures may appear bent in the input image, all our computations are done on the viewing sphere. Users click on the two endpoints of the linear structure to specify the constraint and our system computes and draws the corresponding arc of the great circle passing through those points.
This approach also allows users to interactively set the field of view of the input image. Given an input image with unknown field of view, our system initially assumes either a $180^{\circ}$ or $360^{\circ}$ field of view depending on the user-specified input image mapping (i.e. wide-angle perspective, fisheye, etc.). The user can click the endpoints of a line, and if the field of view is incorrect the computed line will not lie on the linear structure in the input image. The user can then adjust the field of view up or down until the drawn line matches the linear structure. This process allows the user to easily work with an image with unknown parameters. However, the procedure for inferring the FOV will not work if the input image is a perspective projection, since all lines are perfectly straight in any perspective image regardless of focal length. In this case the user must determine the focal length explicitly, possibly using EXIF tags.
To specify the fixed orientation constraint users first create a line and then type ' $h$ ' or ' $v$ ' to indicate that the line should be oriented either horizontally or vertically in the output image. Although our fixed orientation constraint can work with any user-specified orientation, we have found in practice that horizontal and vertical orientation constraints are most common. Therefore, we chose to simplify the interface rather than allow users to specify arbitrary line orientations. Given these line constraints our algorithm computes a mapping from the viewing sphere to the image plane as we describe in the next sections.

### 3.2 Mathematical setup

We define an image projection as a mapping from the viewing sphere parametrized by longitude $\lambda$ and latitude $\phi$ into a planar domain parametrized by $u$ and $v$. In vector form $\lambda=(\lambda, \phi)$ and $\mathbf{u}=(u, v)$. We can represent the mapping as the two functions $u(\lambda, \phi)$ and $v(\lambda, \phi)$, or in vector form as $\mathbf{u}(\lambda)$.
We can describe the local properties of such a mapping in terms of differential north and east vectors, $\mathbf{h}$ and $\mathbf{k}$, of the projection, where

$$
\mathbf{h}=\left[\begin{array}{c}
\frac{\partial u}{\partial \phi}  \tag{1}\\
\frac{\partial v}{\partial \phi}
\end{array}\right], \mathbf{k}=\left[\begin{array}{c}
\frac{\partial u}{\partial \lambda} \\
\frac{\partial v}{\partial \lambda}
\end{array}\right] \frac{1}{\cos (\phi)} .
$$

Without the cosines $\mathbf{h}$ and $\mathbf{k}$ correspond to the columns of the Jacobian matrix $J=\frac{\partial \mathbf{u}}{\partial \boldsymbol{\lambda}}$. However, spherical coordinates are nonEuclidean, and the cosine terms are necessary to account for the
fact that equal steps in $\lambda$ travel different distances on the sphere depending on $\phi$.
A mapping is conformal if $\mathbf{h}$ is a $90^{\circ}$ rotation of $\mathbf{k}$, such that

$$
\mathbf{h}=\left[\begin{array}{cc}
0 & -1  \tag{2}\\
1 & 0
\end{array}\right] \mathbf{k}
$$

or, equivalently

$$
\begin{equation*}
\frac{\partial u}{\partial \phi}=-\frac{\partial v}{\partial \lambda} \frac{1}{\cos (\phi)}, \quad \frac{\partial v}{\partial \phi}=\frac{\partial u}{\partial \lambda} \frac{1}{\cos (\phi)} \tag{3}
\end{equation*}
$$

These are the Cauchy-Riemann equations for mapping a sphere to a plane [Hilbert and Cohn-Vossen 1952; Snyder 1987].
With this setup we can derive and analyze the properties of various projections. For example, in a cylindrical panorama $u$ varies linearly with $\lambda$. So, if $u=\lambda$ then $\frac{\partial u}{\partial \lambda}=1$, which implies that

$$
\frac{\partial v}{\partial \phi}=\frac{1}{\cos \phi}, v=\ln (\sec \phi+\tan \phi)
$$

where the latter equation can be found in a table of integrals. These equations for $u$ and $v$ describe the Mercator projection, which is the only conformal cylindrical projection.
In our case we have locally varying constraints on the mapping function $\mathbf{u}(\boldsymbol{\lambda})$ and its derivatives, so we cannot derive a closed-form solution for the projection. Instead, we discretize the mapping by sampling a uniform grid in ( $\lambda, \phi$ ) indexed by integers ( $i, j$ ) (Figure 2). We define $V$ to be the entire set of vertices $(i, j)$ that fall in the field of view of the input image. The vertices form a quad mesh on the sphere's surface. For each spherical grid vertex $\boldsymbol{\lambda}_{i, j}$, we compute the value of its corresponding $\mathbf{u}_{i, j}$ in the output 2D domain via an optimization.

### 3.3 Conformality

We form conformality constraints on the mesh by discretizing the Cauchy-Riemann equations (3), giving

$$
\begin{align*}
u_{i, j+1}-u_{i, j} & =-\left(v_{i+1, j}-v_{i, j}\right) / \cos \phi_{i, j}  \tag{4}\\
v_{i, j+1}-v_{i, j} & =\left(u_{i+1, j}-u_{i, j}\right) / \cos \phi_{i, j} \tag{5}
\end{align*}
$$

When solving differential equations exactly, multiplying both sides of these equations by a constant has no effect on the solution. However, when solving discretized equations in the least squares sense, changing the weight of the constraints does affect the solution. All quads on the viewing sphere are not equal in size, so equally weighting the constraints biases conformality toward regions of the sphere with higher quad density. We therefore weight the constraints by the area of the quad, which at latitude $\phi$ is proportional to $\cos (\phi)$. We multiply the constraints by $\cos (\phi)$ to define the conformality energy as

$$
\begin{align*}
E_{c}= & \sum_{(i, j) \in V} w_{i, j}^{2}\left(\left(v_{i+1, j}-v_{i, j}\right)+\cos \phi_{i, j}\left(u_{i, j+1}-u_{i, j}\right)\right)^{2} \\
& +\sum_{(i, j) \in V} w_{i, j}^{2}\left(\left(u_{i+1, j}-u_{i, j}\right)-\cos \phi_{i, j}\left(v_{i, j+1}-v_{i, j}\right)\right)^{2} \tag{6}
\end{align*}
$$

where $w_{i, j}$ is a spatially varying weight that we will describe in section 3.6.

### 3.4 Straight lines

We define $L$ as the set of all (general and fixed orientation) line constraints marked by the user and $L_{f}$ as the subset of fixed orientation line constraints. On the viewing sphere straight lines in the scene project to great circles, so we refer to great circles simply as lines.
We constrain all points that lie on a line on the viewing sphere to map to points which are collinear on the image plane. However, since we are working in a discrete domain, few vertices fall directly on the line segment. Instead, if a line intersects a quad we constrain a "virtual vertex" within that quad, placed at the midpoint of the


Figure 3: (left) A line crosses the quad's sides at the yellow points on the sphere. Midway between these on sphere's surface is the midpoint of the quad-line intersection, marked in green. We project the quad vertices onto the plane tangent to the midpoint, and use the inverted bilinear interpolation equations [Heckbert 1989] to compute coefficients ( $a, b, c, d$ ) that express the midpoint as a linear combination of the projected mesh vertices, marked in red. (right) We can express the distance of the virtual vertex $\mathbf{u}_{i, j}$ to the line as the dot product of $\left(\mathbf{u}_{i, j}-\mathbf{u}_{\text {start }}\right)$ with the normal vector, or as the difference of $\left(\mathbf{u}_{i, j}-\mathbf{u}_{\text {start }}\right)$ and its component parallel to the line.
line-quad intersection. We use the midpoints because they are relatively evenly spaced along the line. We define a virtual vertex as a bilinear interpolation of the surrounding vertices. We compute the position of a virtual vertex on the sphere, and its bilinear interpolation coefficients ( $a, b, c, d$ ), as shown in Figure 3(left). For each quad intersected by a line $l$ we associate an output virtual vertex $\mathbf{u}_{i, j}^{l}=a \mathbf{u}_{i, j}+b \mathbf{u}_{i+1, j}+c \mathbf{u}_{i+1, j+1}+d \mathbf{u}_{i, j+1}$ on which we place our line constraints. We denote $V_{l}$ as the set of vertex indices $(i, j)$ corresponding to quads intersected by the line. We also denote the two line endpoints as $\mathbf{u}_{\text {start }}^{l}$ and $\mathbf{u}_{\text {end }}^{l}$. For the rest of this section we drop the superscript $l$ and assume the $\mathbf{u}$ variables correspond to virtual vertices for a particular line.
The distance of a point $\mathbf{u}$ to a line connecting two endpoints $\mathbf{u}_{\text {start }}$ and $\mathbf{u}_{\text {end }}$ is $\left(\mathbf{u}-\mathbf{u}_{\text {start }}\right)^{T} \mathbf{n}\left(\mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)$, where

$$
\begin{equation*}
\mathbf{n}\left(\mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)=R_{90} \frac{\left(\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right)}{\left\|\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right\|} \tag{7}
\end{equation*}
$$

is the normal vector of the line and $R_{90}$ is a 90 degree rotation matrix. We therefore define the line energy for a constrained line $l$ as

$$
\begin{equation*}
E_{l}=\sum_{(i, j) \in V_{l}}\left(\left(\mathbf{u}_{i, j}-\mathbf{u}_{\text {start }}\right)^{T} \mathbf{n}\left(\mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)\right)^{2} . \tag{8}
\end{equation*}
$$

This energy function is non-linear, and we could solve it using a non-linear optimization technique such as Gauss-Newton or Levenberg-Marquardt. However, we take another approach based on simplifying the line energy in two ways, each of which can be solved linearly.
We can express the distance of a point $\mathbf{u}$ to a line as the difference between $\left(\mathbf{u}-\mathbf{u}_{\text {start }}\right)$ and its projection onto the line's unit tangent vector as shown in Figure 3(right). We can thus express the line energy in an equivalent form as

$$
\begin{equation*}
E_{l}=\sum_{(i, j) \in V_{l}}\left\|\left(\mathbf{u}_{i, j}-\mathbf{u}_{\text {start }}\right)-s\left(\mathbf{u}_{i, j}, \mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)\left(\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right)\right\|^{2} \tag{9}
\end{equation*}
$$

where

$$
\begin{equation*}
s\left(\mathbf{u}, \mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)=\frac{\left(\mathbf{u}-\mathbf{u}_{\text {start }}\right)^{T}\left(\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right)}{\left\|\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right\|^{2}} \tag{10}
\end{equation*}
$$

is the normalized length of the projection of $\left(\mathbf{u}-\mathbf{u}_{\text {start }}\right)$ onto ( $\mathbf{u}_{\text {end }}$ $\left.\mathbf{u}_{\text {start }}\right)$, such that $s\left(\mathbf{u}_{\text {start }}, \mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)=0$ and $s\left(\mathbf{u}_{\text {end }}, \mathbf{u}_{\text {start }}, \mathbf{u}_{\text {end }}\right)=1$. We now have two ways to simplify the line energy: by fixing the
normal vector in equation 8 we get

$$
\begin{equation*}
E_{l o}=\sum_{(i, j) \in V_{l}}\left(\left(\mathbf{u}_{i, j}-\mathbf{u}_{s t a r t}\right)^{T} \mathbf{n}\right)^{2} \tag{11}
\end{equation*}
$$

and by fixing the normalized projections in equation 9 we get

$$
\begin{equation*}
E_{l d}=\sum_{(i, j) \in V_{l}}\left\|\left(\mathbf{u}_{i, j}-\mathbf{u}_{\text {start }}\right)-s_{i, j}\left(\mathbf{u}_{\text {end }}-\mathbf{u}_{\text {start }}\right)\right\|^{2} \tag{12}
\end{equation*}
$$

Intuitively, these linearized energy terms allow us to reduce the total line energy in two different ways: $E_{l o}$ allows points to slide freely along the line while fixing the line's orientation, and $E_{l d}$ allows the orientation of the line to change while preventing points from sliding along the line. We use the two line energies alternately in an iterative minimization scheme as follows. First, we initialize each $s_{i, j}$ using the arc length between $\boldsymbol{\lambda}_{i, j}$ and $\boldsymbol{\lambda}_{s t a r t}$ on the viewing sphere. We set $E_{l o}$ as the line energy and optimize the mapping to obtain values for each $\mathbf{u}_{i, j}$, which we then insert into equation 7 to compute a normal for each line. We use these normals and optimize the mapping using the $E_{l d}$ line energies, giving new values for each $\mathbf{u}_{i, j}$ which we use to estimate the $s_{i, j}$ variables with equation 10 . We repeat this process until convergence. For fixed orientation lines, there is no need to use $E_{l d}$, and we use the fixed orientation line energy on every iteration.
Although we cannot prove the convergence of our algorithm, we found it to converge very quickly in practice. After two or three iterations we find the projection is visually comparable to the converged result, and after 10 iterations changes are imperceptible.

### 3.5 Smoothness

The conformality and line constraints alone may lead to visual artifacts in the projection. Since conformality does not consider scale or orientation the projection may change dramatically over the space of the image, especially near line segments (inset figure). Adding a smoothness

 term limits how quickly scales and orientations are able to change. To form the smoothness constraint we again look at the differential north vector $\mathbf{h}$. If $\mathbf{h}$ changes slowly across the projection, orientation and scale must also change slowly. So, in a least squares sense we would like

$$
\frac{\partial \mathbf{h}}{\partial \lambda}=\left[\begin{array}{cc}
\frac{\partial^{2} u}{\partial \phi \partial \lambda} & \frac{\partial^{2} u}{\partial \phi^{2}}  \tag{13}\\
\frac{\partial^{2} v}{\partial \phi \partial \lambda} & \frac{\partial^{2} v}{\partial \phi^{2}}
\end{array}\right]=\left[\begin{array}{ll}
0 & 0 \\
0 & 0
\end{array}\right]
$$

Intuitively, this constraint minimizes the curvature of the mapping. We can discretize this set of equations using finite difference approximations to the second derivatives, again weighted by $\cos \phi_{i, j}$, and take the Frobenius norm of $\frac{\partial \mathbf{h}}{\partial \boldsymbol{\lambda}}$ to give the smoothness energy

$$
E_{s}=\sum_{(i, j) \in V} w_{i, j}^{2} \cos ^{2} \phi_{i, j}\| \|\left[\begin{array}{c}
u_{i, j+1}-2 u_{i, j}+u_{i, j-1} \\
v_{i, j+1}-2 v_{i, j}+v_{i, j-1} \\
u_{i+1, j+1}-u_{i+1, j}-u_{i, j+1}+u_{i, j} \\
v_{i+1, j+1}-v_{i+1, j}-v_{i, j+1}+v_{i, j}
\end{array}\right] \|^{2}
$$

where $w_{i, j}$ is a spatially varying weight that we describe in section 3.6.
Although we could place a similar constraint on $\mathbf{k}$ by setting

$$
\frac{\partial \mathbf{k}}{\partial \lambda}=\left[\begin{array}{cc}
\frac{\partial^{2} u}{\partial \lambda^{2}} & \frac{\partial^{2} u}{\partial \phi \partial \lambda}+\frac{\partial u}{\partial \phi} \frac{\tan \phi}{\cos \phi}  \tag{14}\\
\frac{\partial^{2} v}{\partial \lambda^{2}} & \frac{\partial^{2} v}{\partial \phi \partial \lambda}+\frac{\partial v}{\partial \phi} \frac{\tan \phi}{\cos \phi}
\end{array}\right]=\left[\begin{array}{ll}
0 & 0 \\
0 & 0
\end{array}\right]
$$

we found that this constraint did not improve the results, and that $\frac{\partial \mathbf{h}}{\partial \boldsymbol{\lambda}}$ combined with conformality was sufficient for a smooth mapping.

### 3.6 Spatially-varying constraint weighting

We associate a weight with each vertex, which allows us to spatially vary the strength of conformality and smoothness constraints and selectively reduce shape distortion in areas where it is likely to be most noticeable. We also increase weight to prevent excessive distortion that can occur near the endpoints of line constraints. We base the weighting function on three quantities: proximity to the endpoint of a line, a local image salience measure, and proximity to a face.
Line endpoint weights. Due to the discontinuous nature of the line constraint, highly distorted and even flipped quads can occur near line endpoints. To counteract such distortion, we increase the weighting function near line endpoints using a Gaussian function. We compute the Gaussian by defining distances between vertices to be the Euclidean distance between indices, and setting the standard deviation equal to the width of the mesh divided by 100 . To compute the total line endpoint weight $w_{i, j}^{L}$ at each vertex we sum over the Gaussians for each endpoint.
Salience weights. We weight our constraints by a local salience term to allow smooth areas of the image to be deformed more than areas with greater variation. While there are more complicated techniques for measuring image salience [Itti et al. 1998], we found that simply measuring local variance in color was sufficient for our purposes. More specifically we set the local salience weight $w_{i, j}^{S}$ at each vertex to be the standard deviation in color in a window around the vertex, and then normalize the values between 0 and 1 .
Face detection weights. People are particularly adept at noticing distortion of human faces, so we increase the weighting function near faces using the face detection algorithm of Viola and Jones [2004], as implemented in OpenCV [Bradski and Kaehler 2008]. The input image may be too distorted for the detector to reliably find faces, so we instead warp it to a Mercator projection, which has two desirable properties for face detection: it is conformal, so faces will not be stretched, and it is cylindrical, so the face will be upright assuming the camera and face were upright in the scene. The face detector returns a center and radius for each face, which we use to compute a Gaussian in the Mercator projection, with standard deviation equal to one third the face radius and unit height. We warp the weight field to the equirectangular grid to define the face weight $w_{i, j}^{F}$ at each vertex.
Total weight. We combine the line endpoint weights, the salience weights, the face weights, and a baseline weight given to all vertices to define the total vertex weights as

$$
\begin{equation*}
w_{i, j}=2 w_{i, j}^{L}+2 w_{i, j}^{S}+4 w_{i, j}^{F}+1 \tag{15}
\end{equation*}
$$

In practice, we keep the weights of this linear combination of terms, along with all other parameters associated with our system, fixed for all our results.

### 3.7 Total energy and optimization

In addition to the spatially varying weights we place on each mesh vertex, we also use global weights for each energy term to set their relative importance. Our total energy function is a weighted sum of the conformality, smoothness, and line energies:

$$
\begin{equation*}
E=w_{c}^{2} E_{c}+w_{s}^{2} E_{s}+w_{l}^{2}\left(\sum_{l \in L \backslash L_{f}} E_{l}+\sum_{l \in L_{f}} E_{l d}\right) \tag{16}
\end{equation*}
$$

For lines with fixed orientations we use the fixed-orientation energy instead of the general line energy term. To solve this minimization problem, we alternate between optimizing the total energy with fixed line orientations

$$
\begin{equation*}
E_{o}=w_{c}^{2} E_{c}+w_{s}^{2} E_{s}+w_{l}^{2} \sum_{l \in L} E_{l d} \tag{17}
\end{equation*}
$$



Figure 4: We compare the uncropped and manually cropped results of our system to three global projections. The input image was taken from a $140^{\circ}$ horizontal field of view fisheye lens. This field of view is well beyond the typical limits for perspective projection, causing severe stretching. The Mercator projection bends the lines between walls and ceiling, and the stereographic projection bends vertical lines. Our result is similar to the conformal projections, but with straight lines.
and total energy with normalized projections fixed

$$
\begin{equation*}
E_{d}=w_{c}^{2} E_{c}+w_{s}^{2} E_{s}+w_{l}^{2}\left(\sum_{l \in L \backslash L_{f}} E_{l o}+\sum_{l \in L_{f}} E_{l d}\right) . \tag{18}
\end{equation*}
$$

Since we intend the line energies to approximate hard constraints we use a very high weight on the line term. The only other important factor is the relative weighting of the conformality and smoothness terms, which we determine experimentally such that the smoothness weight is minimal, yet strong enough to correct artifacts. We use $w_{c}=1, w_{s}=12$ and $w_{l}=1000$ for all results. However, we find that using such a high weight for the first iteration can generate a highly distorted initial mapping, since errors in the initialization of the $s_{i, j}$ are large. The algorithm behaves better using a lower weight $w_{l}=10$ for the first iteration.
The quadratic energy function at each iteration of our algorithm results in a sparse linear system

$$
\begin{equation*}
A \mathbf{x}=\mathbf{0} \tag{19}
\end{equation*}
$$

where $A$ is the constraint matrix and $\mathbf{x}$ is a column vector containing the unknowns $u_{i, j}$ and $v_{i, j}$. This system is homogeneous and has trivial solutions corresponding to mapping all vertices to a single point. We can compute a non-trivial solution using an eigenvector decomposition of $A$, as described by Forsyth and Ponce [2002]. An alternative approach is to add a small regularization term for each vertex preferring some known mapping (we use stereographic), making the right-hand side of equation 19 non-zero. We find in practice that solving the regularized system is more efficient and yields visually identical results to the eigenvector solution. We solve the linear system using the PARDISO sparse direct solver [Schenk and Gärtner 2004]. In the last step of the process we render the mesh at the desired image size using bilinear texture mapping [Heckbert 1989]. The resulting image usually has irregular boundaries, as shown in Figure 4, so we let the user crop the result to a rectangular image.

## 4 Results

We demonstrate the results of our system on a number of examples. Additional results are included on our project website (http://vis.berkeley.edu/papers/capp/). The input image for Figure 1 was taken with a $180^{\circ}$ circular fisheye lens, which captures an entire hemisphere of the viewing sphere. As is typical for all our examples, the perspective projection looks undistorted only at the very center of the image, but is extremely stretched toward the pe-
ripheries. While objects and faces look undistorted in the conformal projections, bent lines in the bookcase, ceiling and table are immediately apparent. Our result shows neither type of distortion.
We show additional results in figures 4 and 5. In all of our examples we show the input image with line constraints overlaid, along with the number of line constraints. In many of our examples, our projection tends to look globally very similar to the Mercator and stereographic projections, but with lines straightened. Mercator does well with vertical scene lines, and stereographic does well with scene lines with a common vanishing point near the image center. However, where one of these projections succeeds, the other fails. Our results tend to combine the best of both projections.
In Figure 6 we compare our approach to that of Zorin et al. [1995], after adjusting their parameter to produce a result that we consider best for the input image. Although their approach does achieve a nice balance of conformality and straight line distortion, both distortions are present in the result. In contrast, our result shows neither type of distortion. Additionally, we were able to use constraints on line orientation to make vertical lines appear vertical in the image. In Figure 7, we compare our approach to the "multiplane" projection of Zelnik-Manor et al. [2005]. Where an object (the chair in this example) is distorted by a seam between projections planes, they segment out the object, apply a local perspective projection, composite the object back into the output image, and then fill any remaining holes. Even after this process discontinuity artifacts remain. Our approach finds a smooth projection which keeps all lines straight and does not distort objects. Compared to previous work, our optimization based approach has much more flexibility to adapt to specific scenes.
Since our system is interactive, the user can add or change line constraints if they are not satisfied with the results of the first try. We found that while we could often obtain the desired result on the first try, sometimes we noticed additional lines needed to be straightened or fixed in orientation, requiring a few more iterations of adding more constraints. All the results shown in this paper use between 5 and 40 constrained lines, with an average of 20.
Salience. Although we initially believed the salience-based weighting would be key to the success of our method, in the vast majority of cases its effect is subtle. The space of conformal projections is larger than we expected, and for most sets of line constraints our method can find a solution that is nearly conformal everywhere, not just in salient regions. Face detection was slightly more effective at reducing distortion since we are very perceptually sensitive to dis-


Figure 5: The input images at the top are overlaid with the constrained lines, which may be more easily viewed by zooming in. The "Dining Room" sequence is an example sourced from a $180^{\circ}$ fisheye lens. The perspective projection shows severe stretching here, as perspective stretches to infinity for a $180^{\circ}$ field of view. The Mercator and stereographic examples both bend lines; for Mercator it is most noticeable on the front edge of the carpet and the top of the back wall, and for stereographic it is most noticeable for the vertical lines toward the image peripheries. Our result looks globally very similar to the Mercator projection, but the lines are straight. The "Panorama" series used a $290^{\circ}$ horizontal by $180^{\circ}$ vertical equirectangular panorama as input. The stereographic projection is not well suited for this example; the orientations of vertical lines and objects in the scene look odd and stretching is severe. The Mercator projection does a relatively good job; however, there is bending of straight lines on the walls. Our result looks similar to Mercator in this example, but the lines are straightened.


Figure 6: The projection of Zorin et al. [1995] contains conformality distortion (the people on both sides of the image are stretched horizontally) and bends straight lines (between the ceiling and back wall, and along the door jamb on the left and pipe on the right). Our result shows neither distortion.


Figure 7: Distortions caused by discontinuities in "multi-plane" projection of Zelnik-Manor et al. [2005] are only partially corrected with object segmentation in their "multi-plane multi-view" projection. Our system handles this case well. The stitching artifacts in the image are part of the source data.
tortion in faces (Figure 8); however, in most cases, it again had little effect. Overall, the local distortion introduced by our constraints is significantly less than for an application like content-aware resizing [Avidan and Shamir 2007], so salience is less important.
Implementation details. The dimensions of our quad mesh are independent of the image resolution, and are typically much smaller. The number of vertices is dependent on two factors: the resolution of our discretization of the viewing sphere, and the portion of the viewing sphere covered by the input image. We only add vertices to the linear system if they are covered by image data. All our results are computed on some portion of a spherical mesh with 160,000 vertices. The $180^{\circ}$ fisheye examples (Figure 1 and the first two results in Figure 5) cover half the viewing sphere, requiring meshes of 80,000 vertices, and had a runtime of approximately 1 minute each on a 2.4 Ghz Core 2 Duo CPU. The full frame fisheye images


Figure 8: Our result with and without face detection weighting.


Figure 9: A failure case with long parallel lines between 2 vanishing points.
(Figure 4 and the "Street" and "Chair" examples from Figure 5) used approximately 40,000 vertices, and had a runtime of about 15 seconds. For each result we ran our optimization for 8 iterations, after which all the results converged visually.

## 5 Limitations and future work

Our system does not always compute a satisfactory result. We found three typical scenarios for which the result was often less than we desired (more examples are included on the project website).
(1) When the camera points directly at a long, planar scene, we often found ourselves trying to use the tool to simulate the effect of a multi-viewpoint panorama [Agarwala et al. 2006]. However, in such scenes there are multiple parallel lines covering nearly $180^{\circ}$ of the viewing sphere, and converging at two visible vanishing points. Straightening such lines unavoidably distorts the region between them (Figure 9).
(2) If a scene is covered with large numbers of straight lines in varying directions, it is hard to straighten them all. This challenge is not surprising, given that perspective projection is the only mapping that can preserve all straight lines. This problem often manifests in textures with many directed lines, such as a hardwood or tiled floor. This effect can be observed in Figure 4 and in the "Dining Room" example in Figure 5.
(3) Our method can yield large scale changes near the north and south poles. This limitation arises because spherical coordinates have a singularity at the poles. In most scenes this effect is not a problem; however, scenes with significant areas of interest near the poles can be problematic. This issue is not unique to our application, and others have developed geodesic grids [Sadourny et al. 1968] that discretize the sphere uniformly and without singularities. We plan to try geodesic grids as future work.
A final area of concern with our method relates to cropping: the results of our algorithm are not rectangular and need to be manually cropped to produce a final result (Figure 4). Often we found users would want to extend a crop to include some portion of the field of view, but doing so would cause the crop to extend beyond the image data. A natural extension of our method would be to include a cropping box in the set of constraints, so that the box remains filled while salient scene objects do not extend beyond its perimeter. We include uncropped versions of all our results on the project website.

One area for future work is developing a completely automatic system that identifies salient linear structures using line detection algorithms. Our initial experiments in this direction were not successful. For one, the strength of an edge in the image may not correspond to its salience. Second, we often found ourselves adding lines that do not strictly correspond to image lines (e.g., a long horizon frequently occluded by trees).
There are other potential areas for future work. Our method may be improved by using a more sophisticated salience measure [Itti et al. 1998]. Our technique could also be applied to the image resizing problem, especially for images that have many scene lines that can be distorted by current resizing techniques [Avidan and Shamir 2007; Wang et al. 2008]. An interactive viewer [Kopf et al. 2007] based on our projection could adapt to changing fields of view.

## 6 Conclusion

Producing a wide-angle image free of distortion is challenging because there is no global projection which is free of all types of perceptual distortion. Our method produces a new type of wideangle projection that adapts to content in the scene and constraints specified by the user. Human vision tends to focus on salient features rather than entire scenes. Our optimization exploits this fact by producing a locally varying projection that attempts to preserve salient features. We believe this approach can be exploited to solve a number of problems in computer graphics.
Acknowledgments. We would like to thank Sameer Agarwal for fruitful discussions. We also thank Jeff Chien and the following Flickr users for the use of their photographs: Aldo, MikeSchinkel, Niklas Hellerstedt, wili_hybrid, and Editor B. This work is partially supported by an Alfred P. Sloan Foundation fellowship and a National Science Foundation grant CCF-0643552.

## References

Agarwala, A., Agrawala, M., Cohen, M., Salesin, D., and Szeliski, R. 2006. Photographing long scenes with multi-viewpoint panoramas. ACM Trans. on Graph. 25, 3 (July), 853-861.
Avidan, S., and Shamir, A. 2007. Seam carving for content-aware image resizing. ACM Trans. on Graph. 26, 3 (July), 10:1-10:9.
Bradski, G., and Kaehler, A. 2008. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Cambridge, MA.
Brown, M., and Lowe, D. G. 2003. Recognising panoramas. In International Conference on Computer Vision (ICCV), 1218-1227.
Durand, F. 2002. An invitation to discuss computer depiction. In NPAR 2002: Second International Symposium on Non Photorealistic Rendering, 111-124.
Flocon, A., and Barre, A. 1988. Curvilinear Perspective: From Visual Space to the Constructed Image. University of California Press.
Forsyth, D. A., and Ponce, J. 2002. Computer Vision: A Modern Approach. Prentice Hall, August.
Gal, R., Sorkine, O., and Cohen-Or, D. 2006. Feature-aware texturing. In Rendering Techniques 2006: 17th Eurographics Workshop on Rendering, 297-304.
Несквеrт, P. S. 1989. Fundamentals of texture mapping and image warping. Tech. Rep. UCB/CSD-89-516, EECS Department, University of California, Berkeley, Jun.
Hilbert, D., and Cohn-Vossen, S. 1952. Geometry and the Imagination. Chelsea, New York.
Igarashi, T., Moscovich, T., and Hughes, J. F. 2005. As-rigid-as-possible shape manipulation. ACM Trans. on Graph. 24, 3 (Aug.), 1134-1141.
Itti, L., Косн, C., and Niebur, E. 1998. A model of saliency-based
visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. 20, 11, 1254-1259.
Kopf, J., Uyttendaele, M., Deussen, O., and Cohen, M. F. 2007. Capturing and viewing gigapixel images. ACM Trans. on Graph. 26, 3 (July), 93:1-93:10.
Kubovy, M. 1986. The psychology of perspective and renaissance art. Cambridge University Press.
Lévy, B., Petitiean, S., Ray, N., and Maillot, J. 2002. Least squares conformal maps for automatic texture atlas generation. ACM Trans. Graph. 21, 3, 362-371.
Longfellow, W. 1901. Applied Perspective for Architects and Painters. The Riverside Press.
Palmer, S. E. 1999. Vision Science: Photons to Phenomenology. The MIT Press.
Sadourny, R., Arakawa, A., and Mintz, Y. 1968. Integration of the non-divergent barotropic vorticity equation with an icosahedralhexagonal grid for the sphere. Monthly Weather Review 96 (June), 351-356.
Schaefer, S., McPhail, T., and Warren, J. 2006. Image deformation using moving least squares. ACM Trans. on Graph. 25, 3 (July), 533-540.
Schenk, O., and Gärtner, K. 2004. Solving unsymmetric sparse systems of linear equations with pardiso. Journal of Future Generation Computer Systems 20, 3, 475-487.
Sheffer, A., Praun, E., and Rose, K. 2006. Mesh parameterization methods and their applications. Found. Trends. Comput. Graph. Vis. 2, 2, 105-171.
Snyder, J. P. 1985. Computer-assisted map projection research: U.S. Geological Survey Bulletin 1629. U.S. Gov. Printing Office.

Snyder, J. P. 1987. Map Projections - A Working Manual: U.S. Geological Survey Prof. Paper 1395. U.S. Gov. Printing Office.
Snyder, J. P. 1993. Flattening the Earth, two thousand years of map projections. University of Chicago Press.
Suh, B., Ling, H., Bederson, B. B., and Jacobs, D. W. 2003. Automatic thumbnail cropping and its effectiveness. In Proc. UIST, 95-104.
Szeliski, R., and Shum, H.-Y. 1997. Creating full view panoramic mosaics and environment maps. In Proc. SIGGRAPH, 251-258.
Viola, P., and Jones, M. J. 2004. Robust real-time face detection. International Journal of Computer Vision 57, 2, 137-154.
Vishwanath, D., Girshick, A., and Banks, M. 2005. Why pictures look right when viewed from the wrong place. Naure Neuroscience 8, 10, 1401.
Wang, Y.-S., Tai, C.-L., Sorkine, O., and Lee, T.-Y. 2008. Optimized scale-and-stretch for image resizing. ACM Trans. on Graph. 27, 5 (Dec.), 118:1-118:8.
Wolf, L., Guttmann, M., and Cohen-Or, D. 2007. Nonhomogeneous content-driven video-retargeting. In IEEE International Conference on Computer Vision.
Zeki, S. 2001. Artistic creativity and the brain. Science 293, 5527, 51-52.
Zelnik-Manor, L., Peters, G., and Perona, P. 2005. Squaring the circles in panoramas. In IEEE International Conference on Computer Vision (ICCV), vol. 2, 1292-1299.
Zorin, D., and Barr, A. H. 1995. Correction of geometric perceptual distortion in pictures. In Proc. SIGGRAPH, 257-264.


[^0]:    http://vis.berkeley.edu/papers/capp/

