# Locally Adapted Projections to Reduce Panorama Distortions

Johannes Kopf
Microsoft Research

Dani Lischinski
The Hebrew University

Oliver Deussen
University of Konstanz

Daniel Cohen-Or
Tel Aviv University

Michael Cohen
Microsoft Research

## Abstract

*Displaying panoramic and wide angle views on a flat 2D display surface is necessarily prone to distortions. Perspective projections are limited to fairly narrow view angles. Cylindrical and spherical projections can show full $360°$ panoramas, but at the cost of curving straight lines, interfering with the perception of salient shapes in the scene.*

*In this paper, we introduce* locally-adapted projections. *Such projections are defined by a continuous projection surface consisting of both near-planar and curved parts. A simple and intuitive user interface allows the specification of regions of interest to be mapped to the near-planar parts, thereby reducing bending artifacts. We demonstrate the effectiveness of our approach on a variety of panoramic and wide angle images, including both indoor and outdoor scenes.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

## 1. Introduction

Wide angle panoramic views are easily captured either with a special lens or by aligning and stitching a number of narrower perspective images. While the resulting panoramas may be interactively explored using a number of widely available panoramic viewers, it is impossible to project an entire panorama, or a large portion of it, onto a flat image plane without incurring some distortion. We refer the reader to [ZB95] and [ZMPP05] and the references therein for a good review of common projections.

Linear perspective is the most common and the most natural projection to use for a limited field of view; however, when angles exceed $90°-100°$, perceivable distortions appear. This problem was addressed by Zorin and Barr [ZB95], who proposed a global projection that compromises between preserving straight lines and preserving circles. However, their approach is not designed to handle angles approaching or above $180°$, and does not take the content of the image into account.

Rather than relying on a single projection, we demonstrate *locally-adapted projections*, where the projection changes continuously across the field-of-view. Our approach makes it possible to display objects or regions of interest in a manner similar to narrow view perspective, while ensuring continuous transitions between such regions, thus preserving the overall panoramic context (see Figure 1 right).

Our result, like any lens, represents a mapping from ray directions to 2D pixel positions resulting in a new way to visualize wide angle imagery. In addition to the still image results, we also have defined a viewer that interpolates between natural perspective projections at narrow fields of view and our new locally adapted projection at wider fields of view, as in Kopf *et al.* [KUDC07].

Zelnik-Manor *et al.* [ZMPP05] recognized the problem we address and describe an approach based on simple series of vertical projection planes, each with a different perspective projection. The planes are tangent to the viewing sphere, and arranged such that it is possible to unfold them onto a plane without distortions. This works well only if vertical lines separating the planes coincide with natural existing discontinuities in the scene. However, this approach does not provide a way to limit the vertical extent of the planar regions, and may create sharp orientation discontinuities on salient features crossing the boundaries between planes as can be seen in the bottom panel of Figure 2. Dersch experimented with a similar technique, but he used splines to join the vertical planes [Der08]. This alleviates the problems with discontinuities to some degree, but regions still can't be limited in their vertical extent.

Input image / markup (cylindrical projection)          Our projection

**Figure 1:** *The left image shows a cylindrical 360 degree panorama unrolled onto a plane. Note that many straight lines in the scene are curved in this projection, which does not correspond well with the way humans perceive the world. Our system allows users to "straighten" selected regions in the image, making them appear like they would under an ordinary perspective projection, while maintaining continuity with the rest of the panorama (right image). The red curves over the left image denote the selected regions. (Image ©Bernd Dohrmann)*
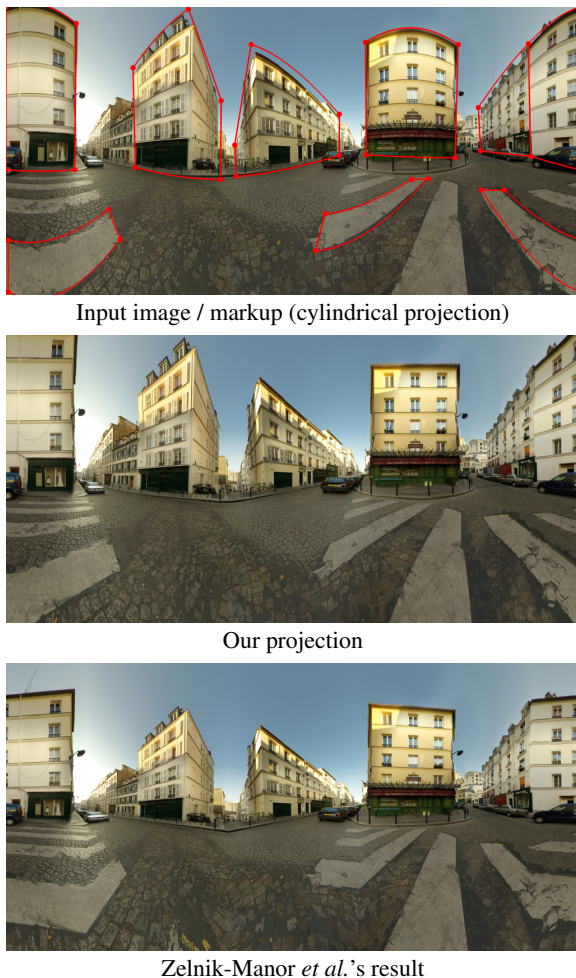


Input image / markup (cylindrical projection)



Our projection



Zelnik-Manor *et al.*'s result

**Figure 2:** *Comparison to Zelnik-Manor et al. (Image © Alexandre Duret-Lutz).*

Our work shares the same goals as Zelnik-Manor *et al.* [ZMPP05], but we remove the limitations mentioned above, enabling the user to specify more general polygonal planar regions that are limited in both their vertical and horizontal extent. These regions may be arbitrarily shaped as in the top panel of Figure 2. Thus, as demonstrated by our results, our system is applicable to a much wider range of scenes. In order to achieve this added expressiveness and flexibility we overcome some non-trivial computational challenges. Firstly, computing smooth projection surfaces involves many more degrees of freedom than piecewise planar projection surfaces. Secondly, we cannot rely on there being a suitable trivial isometric parameterization for our projection geometry.

Given a panoramic image we first remap it onto a cylindrical projection surface, and present the user with the image obtained by unfolding this cylinder onto a plane (Figure 1 left). The user is presented with a number of tools that let her specify regions where a perspective-like projection is desired. The cylindrical projection surface is then deformed to become planar in these regions. Finally, a reparameterization is computed mapping this new surface onto the image plane, while minimizing distortions.

One may think about the above process as taking a 2D slice of the plenoptic function (a set of rays through a point) and projecting these rays onto the image plane via a special, spatially varying optical system (or lens) interactively designed by the user. A number of classical image-based rendering (IBR) systems, such as the plenoptic modeling system of McMillan and Bishop [MB95], or the Lumigraph system of Gortler *et al.* [GGSC96], also operate by rendering 2D plenoptic slices, sometimes making use of a geometric proxy approximating the scene geometry. However, these IBR systems focus on rendering images from novel viewpoints, where the resulting novel views still use only conventional projections. In contrast, we keep the viewpoint fixed,

and focus on warping wide angle imagery in order to reduce the unwanted distortions that result from curvilinear projections. The auxiliary geometry used in our work serves a very different purpose: creating a general, deformed projection surface, rather than approximating the scene. The IBR works cited above could all use our results if they desire to depict wide angle fields-of-view.

Several previous works (e.g., [WFH*97, RB98]) address the creation of multiperspective projection images by assembling sets of rays going through multiple centers of projection into a single image. In contrast, in our case all rays go through a single projection center, i.e., we start and end with same single viewpoint and focus on how the set of rays through this viewpoint is mapped onto the image plane.

Finally, our approach bears a slight resemblance to content-aware image resizing of Wang *et al.* [WTSL08], where distortion of visually prominent feature is minimized when scaling and stretching images by using non-linear optimization. However, our optimization is geared at a rather different goal, and it explicitly accounts for the 3D geometry of the deformed projection surface.

## 2. Locally-adapted projections

As explained earlier, our goal in this work is to generate a projection surface custom-tailored to the contents of a specific panoramic image. Automatically identifying regions of interest, where distortions are to be avoided, is a challenging task, which we leave for future work. Instead, we entrust the creative control in the hands of the user, providing her with a simple and intuitive user interface for indicating such regions, and for prescribing their desired perspective-like appearance in the final result. Once such regions have been specified, our system automatically generates a suitable projection surface that is constrained to be flat in the user-indicated regions, and falls smoothly back to a cylindrical surface away from them. To produce an image, the projection surface must be unfolded onto the plane. Unfortunately, in general, such surfaces cannot be trivially unfolded. Thus, we seek an as distortion-free as possible parameterization: a mapping that is as isometric as possible from the 2D image plane onto the interactively designed projection surface embedded in 3D.

Below, we elaborate on the three main components of our system:

1. user interface,
2. generation of a smooth projection surface, and
3. finding a distortion-free parameterization of the surface.

### 2.1. User interface

Our system is able to handle any single-view panoramic image, such as wide angle perspective, cylindrical or spherical panoramas, fisheye, catadioptric images, etc. Regardless of

the original projection, however, we begin by remapping the image onto a cylinder (noting we may lose the polar caps), unfolding the cylinder onto a plane, and present the resulting image to the user. In this image, vertical lines in the scene (parallel to the cylinder's axis) and the horizontal horizon line remain straight in the image, while other lines become curved. The user is then able to draw polygonal frusta from the center of projection. These frusta appear as curvilinear polygons on the image (Figure 1), since general planes intersect the cylinder in curves, which in turn map to curves in the image when unrolled onto a plane. The user draws via a simple rubber-band interface that adapts to the cylindrical geometry. Mouse clicks introduce the corners vertices of the frusta, and as the mouse moves from one vertex to the next, a curved rubber-band is displayed. Each vertex has an associated 3D position on the unit cylinder, and the curved polygon boundaries correspond to the cylindrical projections of the straight lines connecting these 3D positions. This makes it easy for the user to mark regions with enclosing polygons.

In a separate view, the user is presented with a set of planar polygons (the projection planes of the frusta), with the enclosed part of the image projected onto these polygons using planar perspective projection. We perform a least-squares fit of a plane to each polygon's vertices, since polygons with more than three vertices on the unit cylinder are not guaranteed to be planar, in general. The initial projections onto these polygons serve as a preview of what these image regions would look like in the final result. At this point, the user is able to change the orientation of each polygon by clicking inside a polygon and rotating it about its centroid to control the amount of perspective foreshortening. A right-click can be used to adjust the projection plane's distance from the cylinder's main axis to control the size of each region in the final image.

### 2.2. Projection surface generation

Our next task is to generate a smooth projection surface, which is flat in the user-specified areas, continuous and smooth everywhere, and falling back to a cylinder away from the constrained regions. The flat parts of the surface, corresponding to the constrained regions, have already been determined as explained above, and now it remains to complete the rest of the surface. Since our projection surfaces look like locally deformed cylinders, we found it natural to reason about this process using cylindrical coordinates $(\theta, h, r)$, where $\theta$ is the azimuthal angle, $h$ is the vertical coordinate, and $r$ is the distance from the axis ($r \equiv 1$ for points on the unit cylinder).

Consider the 2D grid obtained by uniformly sampling the $(\theta, h)$ domain. We can construct a height field in this domain, $r(\theta, h)$, representing the radial distance to the projection surface. The planar polygons become U-shaped pieces on this height field, since their centers are closer to the cylinder axis than their edges, as shown in Figure 3b. The remainder of
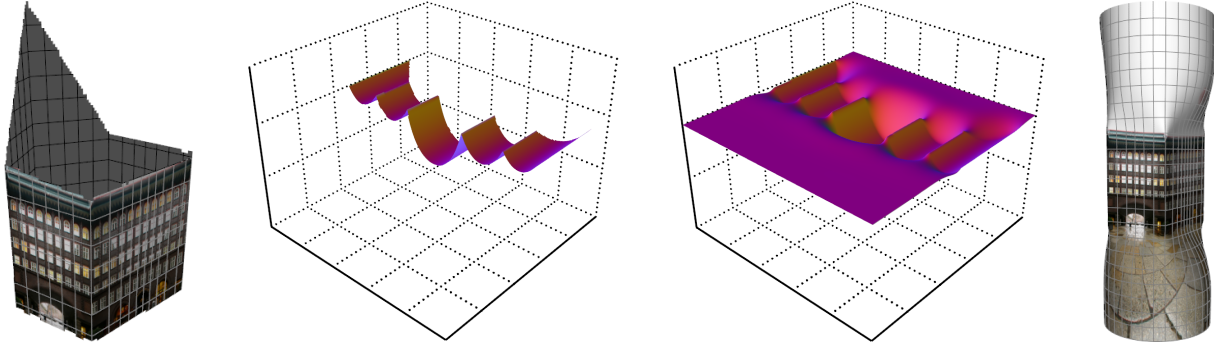
**Figure 3:** *Surface generation, from left to right: (a) constraint polygons in 3D (the constraints are from Figure 1); (b) constraint polygons on the $(\theta, r)$ height field; (c) completed height field; (d) complete projection surface in the 3D Euclidean space.*

the surface is still undefined. Our goal now is to smoothly complete the height field (Figure 3c).

We would like the resulting surface to satisfy a number of requirements: (i) it should satisfy the user-specified constraints; (ii) the transition between the constrained regions and the remainder of the surface should be $C^1$; (iii) the surface should be smooth elsewhere; and, (iv) it should gracefully fall back to a unit cylinder away from the constrained regions (Figure 3d).

The first two requirements are "hard" constraints, while the other two are "soft" ones. We encode these requirements as a set of linear equations, and compute a least-squares solution for the resulting overdetermined linear system. The equations corresponding to the "hard" constraints are included in the system with a large weighting term [Van85]. While this could lead to some small deviations from the hard constraints, this is perfectly acceptable for our purposes.

More specifically, there is a variable $x_{i,j}$ for each grid point $(\theta_i, h_j)$. For each point inside the user-constrained regions, we generate the equation,

$$x_{i,j} = r_{i,j}, \qquad (1)$$

where $r_{i,j} \equiv r(\theta_i, h_j)$ is the value of the height field inside the constrained region. For a pair of neighboring variables $x_{i-1,j}$ and $x_{i,j}$, where the former is unconstrained while the latter is constrained, we add the equation:

$$x_{i,j} - x_{i-1,j} = r_{i+1,j} - r_{i,j}, \qquad (2)$$

approximating derivative continuity. A similar constraint is added for each pair of adjacent variables across all polygon boundaries. To promote smoothness in the unconstrained regions, we introduce a Laplacian term for each unconstrained variable $x_{i,j}$:

$$4x_{i,j} - x_{i-1,j} - x_{i+1,j} - x_{i,j-1} - x_{i,j+1} = 0, \qquad (3)$$

since the solution to the Laplace equation is a smooth harmonic function. An additional equation pulls unconstrained

points towards the unit cylinder:

$$x_{i,j} = 1. \qquad (4)$$

The final overdetermined system consists of a weighted combination of the above equations. Equations (1) and (2) are given large weights (100 in our current implementation), equation (3) is weighted by $1 - \lambda$, and equation (4) by $\lambda$, where $\lambda$ is a parameter that balances between smoothness and falling back to a unit cylinder shape. In all our results we used $\lambda = 0.01$. Once the system is solved, the least-squares height field is converted back to Euclidean coordinates (Figure 3d). In our implementation we use 128x64 vertices to compute the surface. Solving the least-squares system takes about 0.2 seconds.

### 2.3. Projection surface parameterization

The completed height field defines a deformed cylinder embedded in 3D that must be unwrapped onto the plane to create the final image. Unfortunately, this surface is not *developable*, as it no longer has zero Gaussian curvature everywhere. Unfolding a general deformed surface is not possible without introducing some amount of distortion. Our goal is thus to unfold the deformed cylindrical surface onto the plane with minimal distortions.

Our problem is equivalent to one of finding a minimal-distortion parameterization of the deformed projection surface. In other words, we are seeking a mapping between the 2D image space and the projection surface that would be as close as possible to isometric. Such a mapping enables us to determine, for each pixel in the resulting image, the corresponding source location in the initial cylindrical representation.

We begin with a uniform 2D grid in $(\theta, h)$, with each 2D grid point initially corresponding to the same point on the $(\theta, h, r)$ height field defined in the previous section. If this initial grid is drawn on the corresponding 3D surface defined
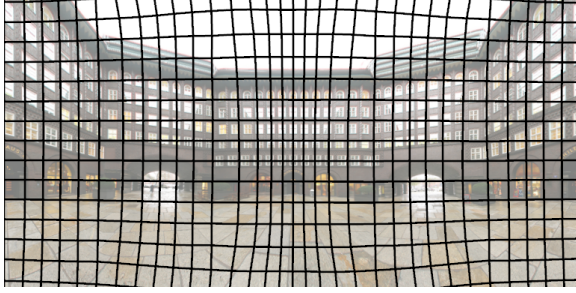
**Figure 4:** *The 2D $(\theta, r)$ warped grid after the optimization.*



**Figure 5:** *Comparison to 2D image warping.*

by the height field, we see that squares in the grid get mapped to distorted shape on the surface (Figure 3d).

Our task now is to adjust the positions of the 2D grid points such that these distortions are minimized on the surface (see Figure 4). Any remaining distortions should be evenly distributed across the grid. There are many available parameterization techniques that could potentially be harnessed to accomplish this task. We chose to use Hormann and Greiner's "Most Isometric Parameterization" (MIPS) method [HG00]. Specifically, we use the multiresolution variant of MIPS, as described in [Hor01].

The original MIPS method operates on triangle meshes. It locally optimizes the location of each vertex, so as to minimize the shape penalty for each of its incident triangles. In our case, however, we are dealing with a quadrilateral grid; we apply MIPS by optimizing each grid vertex $v_{i,j}$ with respect to the four "virtual" triangles formed by $v_{i,j}$ together with its 4-neighbors: $\langle v_{i,j}, v_{i-1,j}, v_{i,j-1}\rangle$, $\langle v_{i,j}, v_{i,j-1}, v_{i+1,j}\rangle$, $\langle v_{i,j}, v_{i+1,j}, v_{i,j+1}\rangle$, $\langle v_{i,j}, v_{i,j+1}, v_{i-1,j}\rangle$. The left and right boundary vertices are constrained to only move vertically. Grid points lying on the horizon are constrained to move only horizontally (the horizon is optionally indicated by the user). The complete optimization takes about one half second to converge.

## 3. Rendering

The resulting perturbed grid from the above optimization defines a warp field, which we can now use to warp the original panoramic image to produce the final result. Drawing the array of textured warped grid squares results in a the final static images seen in Figures 1, 2, and 6.

These results represent a new projection, in other words, a mapping from ray directions to 2D pixel positions for rendering wide angle imagery. Thus, in addition to generating static images, an interactive viewer, as in Kopf *et al.* [KUDC07], can interpolate between perspective projection at narrow fields of view and our new locally adapted projection at wider fields of view. The viewer stores two 3D coordinates for each grid point: one is the pre-computed result for our projection, the other is a standard perspective projection computed on-the-fly. A sigmoid curve provides the interpolation weights between these two projections to linearly interpolate the 3D coordinates. The curve is set such that the projection is fully perspective for fields of view between 0 and 60 degrees and fully ours beginning at 100 degrees, and smoothly interpolated in between.

## 4. Results and conclusions

Figures 1, 2, and 6 show a number of locally adapted projections generated using our system. The user time required to define each of these projections was under 2 minutes. Each of these examples manages to convey the original 3D shape of prominent structures in the scene while preserving their surrounding panoramic context. Note, in particular, the challenging interior panoramas.

Our method still has a number of limitations. While in most cases it is sufficient to loosely draw a polygon around a region without exactly tracing the boundaries, we still rely on the user to draw the polygons properly. If the perspective regions are too large (e.g. near $180°$ degrees) some of the typical perspective distortion artifacts may appear (see Figure 7). In this case the region has to be broken into several smaller regions, which might introduce a sudden bend in the image. In some cases there are several possible ways to partition a scene into perspective regions. A bit of experience helps to identify the best solution. Finally, our parameterizations are not perfectly isometric, thus, even in the marked regions some lines might still appear slightly curved. Sometimes, the orientation of a whole region changes slightly during the optimization. The most noticeable artifact due to this is that originally vertical lines sometimes appear a bit slanted in our result.

It is interesting to compare our results with those that might be obtained with the multi-plane approach [ZMPP05]. Such a result is shown in Figure 2 (bottom). While their system does enable flattening user-selected sections of the image, the flattening affects entire vertical sections, which causes sharp orientation discontinuities of some of the continuous features, such as the stairs, ceiling beams, and the straight planks of wood on the floor.

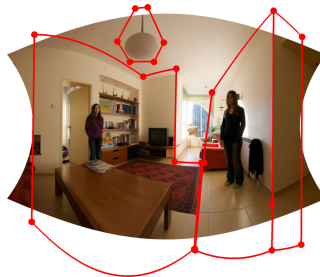Input image / markup (cylindrical projection)      Our result

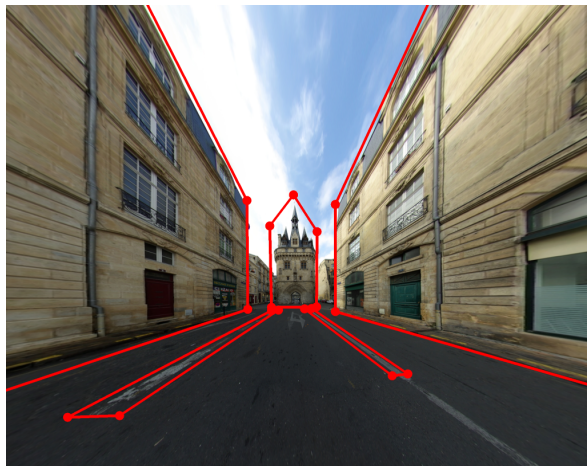Input image / markup (cylindrical projection)      Our result

Input image (fisheye projection)    Markup (cylindrical projection)    Our result

Input image / markup (wide angle perspective)      Our result

**Figure 6:** *Several results produced with our system (Images in the first row © Andrey Ilyin, images in the second row © Sébastien Pérez-Duarte, Images in the last row © Alexandre Duret-Lutz).*

Another alternative is to use a general purpose 2D image warping tool in order to achieve the desired effect. We generated the image in Figure 5 using a state-of-the-art image warping algorithm [SMW06] (which can be compared to Figure 1). When attempting to warp the same curvilinear polygons that serve as input to our method into regions with straight lines, the results are clearly unsatisfactory. This is not surprising, since such an approach does not take into account the 3D viewing geometry. Although better results may be obtained by specifying the desired 2D warp on a fine grid, that would be a much more time consuming and tedious task for the user. Additional comparisons with both of these alternatives are included in the supplementary materials.

In summary, our approach offers a new and intuitive way for displaying and exploring wide panoramic views, providing the user with significantly better control over the result than was previously possible. In future work, we would like to find ways of computing such projections in real-time to facilitate fully interactive exploration of such images.

## Acknowledgements

Markup (cylindrical projection)

Result           Surface

**Figure 7:** *A less successful case. The marked region in this example is almost* $180°$ *wide, which results in the typical perspective distortion artifacts, e.g. the stretched chairs at the bottom of the image (Image © Sébastien Pérez-Duarte).*

## References

[Der08] DERSCH H.: Personal communication. http:// webuser.hs-furtwangen.de/~dersch/mr/, 2008.

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The Lumigraph. *Proceedings of SIGGRAPH* (1996), 43–54.

[HG00] HORMANN K., GREINER G.: MIPS: An efficient global parametrization method. *Curve and Surface Design: Saint-Malo 1999* (2000), 153–162.

[Hor01] HORMANN K.: *Theory and Applications of Parameterizing Triangulations*. PhD thesis, University of Erlangen, Nov 2001.

[KUDC07] KOPF J., UYTTENDAELE M., DEUSSEN O., COHEN M.: Capturing and viewing gigapixel images. *ACM Transaction on Graphics 26*, 3 (Proceedings of SIGGRAPH) (2007).

[MB95] MCMILLAN L., BISHOP G.: Plenoptic modeling: an image-based rendering system. *Proceedings of SIGGRAPH* (1995), 39–46.

[RB98] RADEMACHER P., BISHOP G.: Multiple-center-of-projection images. *Proceedings of SIGGRAPH* (1998), 199–206.

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3 (Proceedings of SIGGRAPH) (2006), 533–540.
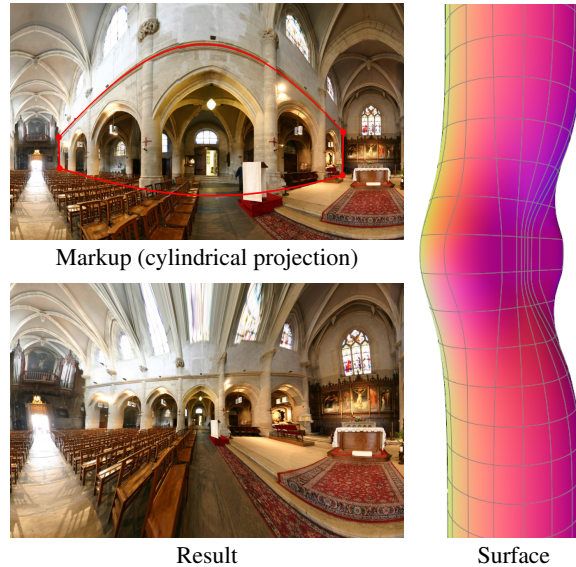
[Van85] VAN LOAN C.: On the method of weighting for equality constrained least-squares problems. *SIAM Journal on Numerical Analysis 22*, 5 (1985), 851–864.

[WFH*97] WOOD D. N., FINKELSTEIN A., HUGHES J. F., THAYER C. E., SALESIN D. H.: Multiperspective panoramas for cel animation. *Proceedings of SIGGRAPH* (1997), 243–250.

[WTSL08] WANG Y.-S., TAI C.-L., SORKINE O., LEE T.-Y.: Optimized scale-and-stretch for image resizing. *ACM Transactions on Graphics 27*, 5 (Proceedings of SIGGRAPH Asia) (2008), 118:1–118:8.

[ZB95] ZORIN D., BARR A. H.: Correction of geometric perceptual distortions in pictures. *Proceedings of SIGGRAPH* (1995), 257–264.

[ZMPP05] ZELNIK-MANOR L., PETERS G., PERONA P.: Squaring the circle in panoramas. *Proceedings of ICCV* (2005), 1292–1299.