

Image Warps for Artistic Perspective Manipulation

Robert Carroll
University of California, Berkeley

Aseem Agarwala
Adobe Systems, Inc.

Maneesh Agrawala
University of California, Berkeley



Figure 1: We present a tool for controlling the perspective in photographs by manipulating vanishing points and lines. We can constrain the lines to converge sharply as in a wide-angle image or more weakly as in a telephoto image (Figure 2). Our perception of the room changes for each result; the former result feels more tightly enclosed and the later feels more spacious.

Abstract

Painters and illustrators commonly sketch vanishing points and lines to guide the construction of perspective images. We present a tool that gives users the ability to manipulate perspective in photographs using image space controls similar to those used by artists. Our approach computes a 2D warp guided by constraints based on projective geometry. A user annotates an image by marking a number of image space constraints including planar regions of the scene, straight lines, and associated vanishing points. The user can then use the lines, vanishing points, and other point constraints as handles to control the warp. Our system optimizes the warp such that straight lines remain straight, planar regions transform according to a homography, and the entire mapping is as shape-preserving as possible. While the result of this warp is not necessarily an accurate perspective projection of the scene, it is often visually plausible. We demonstrate how this approach can be used to produce a variety of effects, such as changing the perspective composition of a scene, exploring artistic perspectives not realizable with a camera, and matching perspectives of objects from different images so that they appear consistent for compositing.

1 Introduction

Since the Renaissance, artists have used perspective as a means of depicting a three-dimensional scene by projecting it onto a two-dimensional surface. The rules of perspective follow from the projection model of a pinhole camera; points in the scene are projected onto a picture plane through straight lines converging on the center of projection. However, this three-dimensional model is often of little practical use as a drawing tool for illustrators and painters. Such artists more commonly work with a two-dimensional image space representation of the scene [Willats 1997]. When drawing in perspective most artists do not trace rays from the scene into the

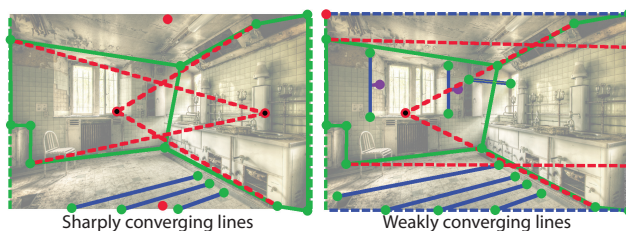


Figure 2: The vanishing point, plane, and lines constraints used to produce the results shown in Figure 1.

image plane. Instead they sketch vanishing points and lines to guide the construction of their images [D’Amelio 2004].

Artists also realize that strict adherence to the rules of perspective does not always achieve the most effective composition. Careful analysis of perspective paintings reveals a variety of inconsistencies in vanishing points, sizes of objects, and foreshortened textures [Criminisi et al. 2002; Kubovy 1986; Loran and Cézanne 1985]. Paintings deviate from the rules of perspective to achieve various effects, such as depicting objects from inconsistent viewpoints, or locally adapting the projection of a particular object to avoid perspective distortion. Agrawala et al. [2000] show how artists often combine multiple projections in a single image to express a mood, to improve the comprehensibility of a scene, or to better visualize the spatial relationships between objects in the scene.

With the invention of photography, the perspective camera became an optical reality rather than just an abstract model to aid painters. Yet, cameras strongly restrict the possibilities for controlling perspective. Photographers control the projection by adjusting camera placement and field of view. Using more advanced hardware such as a view camera or perspective control lens they can also adjust the orientation of the image plane. Post-processing image manipulation software is even more restrictive, offering control over just a few of these projection parameters. At best such software allows photographers to reproject the image via a homography. None of these tools provide direct image space controls over the vanishing points, construction lines and object shapes that illustrators commonly adjust to create perspective images. Moreover, these tools limit photographers to using a single, consistent projection across the entire image. Unlike painters, photographers cannot create images from physically inaccessible viewpoints, or locally combine multiple perspectives for different objects.

We present a tool that gives photographers the ability to manipulate

http://vis.berkeley.edu/perspective_manipulation

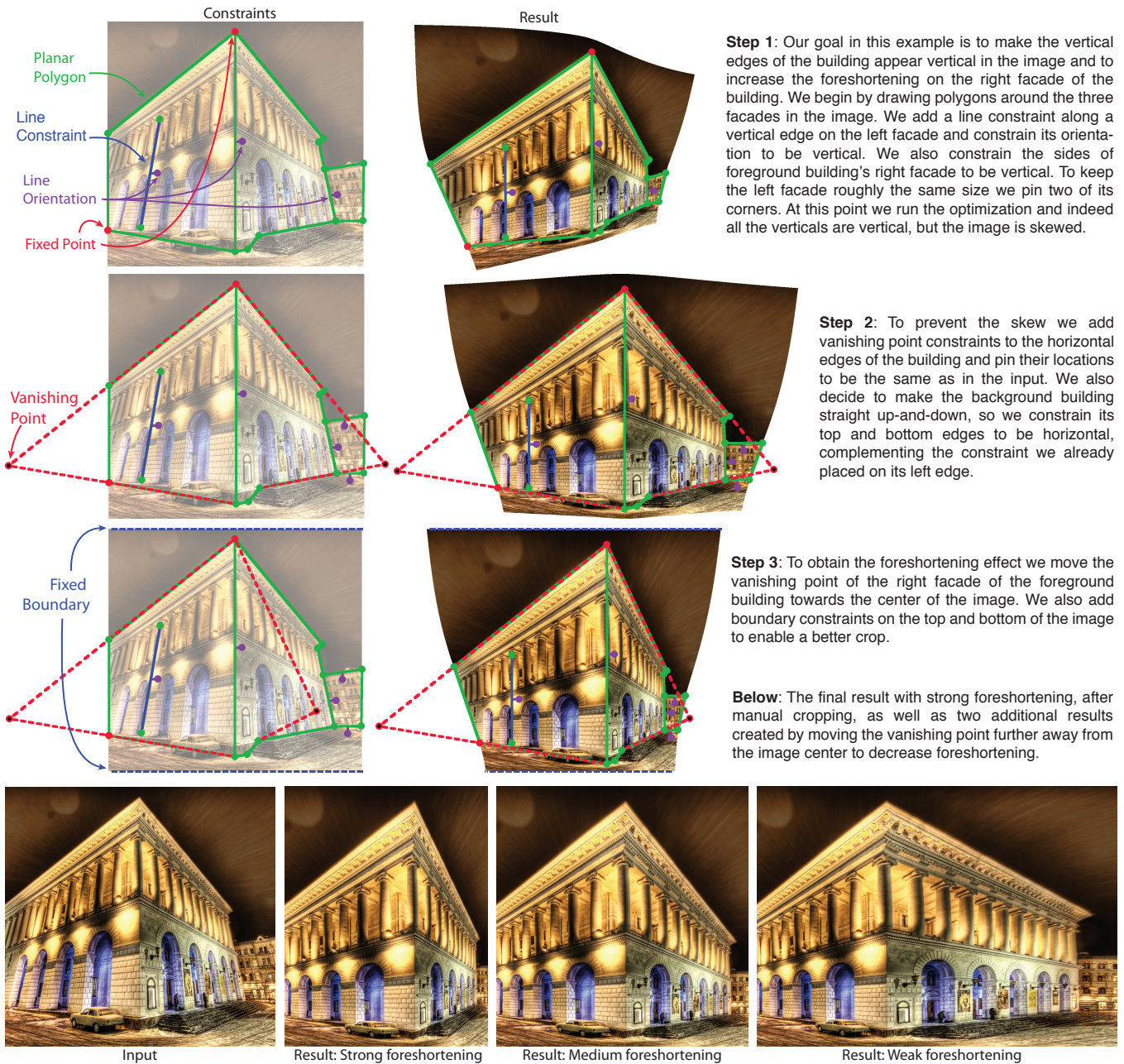


Figure 3: A typical interaction with our tool. We show the intermediate steps used to foreshorten the right wall of a building.

perspective in images after they are captured using image space controls similar to those used by artists to construct perspective images (Figure 1). Our approach uses 2D image warps guided by constraints based on projective geometry. A user annotates an image by marking a number of image space constraints including planar regions of the scene, straight lines, and associated vanishing points (Figure 2). The user can then use these primitives as handles to control the warp. Our system optimizes the warp such that straight lines remain straight, planar regions are transformed as they would under a perspective projection (i.e. according to a homography) and the entire mapping is as shape-preserving as possible.

The warped image produced by our tool is not an accurate perspective projection of the scene, i.e., it may not be realizable by placing a camera in the scene. In fact, users may specify constraints that correspond to changing the position of the camera relative to

the scene. Producing an accurate perspective projection would require knowing 3D geometry, and even if the geometry visible in the original image was known, changing the viewpoint would cause disocclusions due to parallax. Instead, our approach finds a smooth mapping from input image to output image, allowing us to sidestep the challenging issue of hole-filling. Nevertheless, we are often able to produce perceptually plausible images that look as if the position of the camera moved within the scene. Furthermore, our tool allows users to manipulate images according to constraints that are based on perspective rules, but are not necessarily consistent with a single perspective projection.

Our work makes two main contributions. First, we propose an interface that allows users to manipulate perspective in photographs using controls similar to those used by illustrators in the construction of perspective images, namely vanishing points and lines. Second,

we propose an optimization-based warping model which provides a means to plausibly simulate moderate changes in perspective induced by these controls.

2 Related Work

The oldest photographic perspective manipulation tool is the view camera or perspective control lens [Stroebel 1999], which allows the photographer to change the orientation of the film plane relative to the scene. The ability to apply a similar effect digitally through a projective transform is a common feature in photography software. Our method can be seen as an enhancement to this approach. We add the ability to limit the extent of a projective transform to a specific image region, and the ability to apply different transforms to different regions. Finally, we also propose a different control metaphor through the manipulation of vanishing points and lines.

There are a number of techniques that support the manipulation of both standard and artistic perspective; only a few, however, take photographs as input without the benefit of known 3D geometry. Agarwala et al. [2006] described how to create multi-viewpoint panoramas of scenes too long to depict with standard perspective, starting with a series of photographs. Zelnik-Manor and Perona [2007] automated the process of compositing a series of photographs from nearby viewpoints into Hockney-like joiners. Carroll et al. [2009] and Kopf et al. [2009] addressed the issue of projecting wide fields of view, such as spherical and cylindrical panoramas, into flat images that avoid perspective distortion or curved lines. We follow the approach of Carroll et al. and optimize a spatially-varying warp from a source to a target domain that avoids distortion while respecting user constraints. However, our application is for changing perspective rather than wide-angle imaging.

The warping technique at the heart of our method resembles several recent methods for 2D shape manipulation [Igarashi et al. 2005; Schaefer et al. 2006] and image retargeting [Wolf et al. 2007; Gal et al. 2006; Wang et al. 2008]. Shape manipulation techniques find warps that are locally as free of distortion as possible under user-specified point constraints, where the energy and constraints are defined on a mesh. In image retargeting, the energy function is usually weighted spatially according to image content and the borders are constrained to the output frame. Our setup is similar in that we optimize a mapping over a meshed image, but we also use constraints designed to give high level control over perspective.

A more fully explored topic is the artistic manipulation of perspective given known 3D geometry. A number of researchers have addressed the problem of creating projections that contain multiple viewpoints, either interactively [Agrawala et al. 2000; Coleman and Singh 2004] or by combining multiple views along a camera path [Wood et al. 1997; Rademacher and Bishop 1998; Popescu et al. 2009]. Several researchers have proposed alternative projection models that go beyond the standard linear projection model, and they have used these models to produce artistic renderings [Yu and McMillan 2004; Hall et al. 2007]. Unlike these methods our approach does not rely on access to 3D scene geometry.

Another approach to manipulating the perspective in a photograph is to first reconstruct its 3D geometry and camera parameters, and then manipulate the camera or use the above techniques. Horry et al. [1997] introduced a simple interface that takes a central perspective image and fits a scene model consisting of a box for the background and axis-aligned planes for the foreground. Criminisi et al. [2000] showed how 3D models could be created from perspective images by propagating the positional information from a few manually-specified 3D seed points with user guidance. This technique can be used to infer the 3D locations of geometric primitives, notably the corners of planar polygons. The first stage of our approach is quite similar, with the user’s effort dominated by

drawing lines and polygons in the input image. However, our interaction metaphor is entirely 2D, and we do not require the user to specify any 3D coordinates. Also, in many cases we can get away with specifying constraints on only a small portion of the image, whereas with 3D reconstruction only portions of the image with 3D geometry can be re-projected. Automatic single-view modeling is also an active topic of research [Hoiem et al. 2005; Saxena et al. 2009]. These techniques are designed for outdoor scenes and report success rates of 30% and 67%, respectively; our goal is to create a more robust, user controllable tool that works on indoor and outdoor scenes. Another issue with single view modeling techniques is that they do not produce a complete representation of the 3D scene. Manipulating perspective with such an incomplete representation can lead to disocclusions or holes that need to be filled.

3 Approach

Our perspective manipulation system is comprised of a user interface for interactively specifying constraints and a nonlinear least-squares optimization that computes the best warp satisfying these constraints. We first describe our user interface and demonstrate how it can be used for perspective manipulations. We then describe the details of our mesh-based warp in Section 3.2.

3.1 User interface

We provide the following controls for manipulating perspective.

Planar Regions. Users can constrain a polygonal image region so that it is transformed as a projection of a rigid 3D plane, i.e., it is a homography.

Line Segments. Users can constrain line segments to remain straight. This constraint can also be used in conjunction with the orientation and vanishing point constraints.

Line Orientation. Users can constrain a line segment to be oriented vertically or horizontally.

Vanishing Points. Users can connect multiple line segments to a vanishing point constraint to ensure that they all intersect at the vanishing point. Moving vanishing points around the image plane allows users to control perspective changes.

Fixed Points. Users can constrain a point in the input image so that it does not move.

Borders. Users can constrain each side of the image boundary to stay fixed or optionally to just remain straight.

We demonstrate how these constraints can be used to control perspective in a typical user interaction with our tool (Figure 3).

3.2 Warping Function

As shown in Figure 4, our image warp is defined in terms of a mapping from the input image parameterized by $\mathbf{x} = (x, y)$ into a planar domain parameterized by $\mathbf{u} = (u, v)$. We can represent the mapping as the two functions $u(x, y)$ and $v(x, y)$, or in vector form as $\mathbf{u}(\mathbf{x})$. We discretize the mapping by sampling a uniform grid in (x, y) indexed by integers (i, j) , forming a quad mesh upon which we define our constraints. For each grid vertex $\mathbf{x}_{i,j}$, we compute the value of its corresponding $\mathbf{u}_{i,j}$ in the output domain via an optimization.

Planar Regions. Under a perspective projection planar regions in the scene are transformed according to a homography. Therefore, we constrain the set of mesh vertices enclosed by a user marked polygon to be mapped according to some (unknown) homography. In homogeneous coordinates this constraint can be expressed as

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

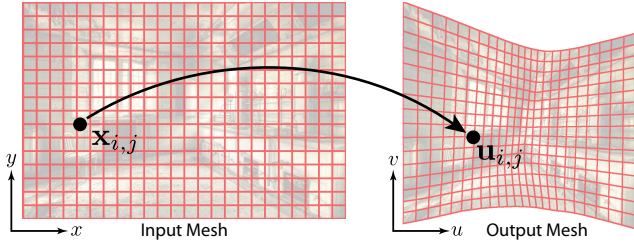


Figure 4: The input and warped mesh computed for Figure 1, shown at half resolution. The mesh defines a discrete mapping from the xy -plane to the uv -plane.

and $(u, v) = (U, V)/W$. We use this constraint to define an energy term for a planar polygon as

$$E_h = \sum_{i,j} \left\| \begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix} - \begin{bmatrix} h_1 x_{i,j} + h_2 y_{i,j} + h_3 \\ h_7 x_{i,j} + h_8 y_{i,j} + 1 \\ h_4 x_{i,j} + h_5 y_{i,j} + h_6 \\ h_7 x_{i,j} + h_8 y_{i,j} + 1 \end{bmatrix} \right\|^2 = \sum_{i,j} \|\mathbf{u}_{i,j} - \mathbf{H}(\mathbf{x}_{i,j})\|^2 \quad (2)$$

In general, both the homography variables $h_{1..8}$ and the output coordinates $(u_{i,j}, v_{i,j})$ are unknowns that we optimize.

Homography Compatibility. If two planar polygons share a common edge it is necessary to constrain their homographies to be compatible along that edge to prevent discontinuities in the mesh. That is, if \mathbf{x}_1 and \mathbf{x}_2 are points that define a shared edge between two polygons in the input image domain with associated homographies H_a and H_b , we must ensure $\mathbf{H}_a(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) = \mathbf{H}_b(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2)$ for all $\alpha \in [0, 1]$. It is sufficient to constrain the homographies for a few values of α evenly sampled from 0 to 1, giving the energy term

$$E_{hc} = \sum_{i=0}^n \|\mathbf{H}_a(\alpha_i \mathbf{x}_1 + (1 - \alpha_i) \mathbf{x}_2) - \mathbf{H}_b(\alpha_i \mathbf{x}_1 + (1 - \alpha_i) \mathbf{x}_2)\|^2, \quad (3)$$

where $\alpha_i = \frac{i}{n}$. We found $n = 10$ to be sufficient. The homography parameters are the only unknowns in this energy function.

Straight Lines. We constrain a user-specified line segment to remain straight by discretizing it into a set of points $\mathbf{x} \in V_l$, one for each quad the line segment crosses. The mapping $\mathbf{u}(\mathbf{x})$ of all $\mathbf{x} \in V_l$ should be collinear. If we parameterize lines by the mapping of one of these points $\mathbf{u}(\mathbf{x}_0)$ and an orientation θ , then the distance of a point \mathbf{u} to the line is $[\sin(\theta) \cos(\theta)](\mathbf{u} - \mathbf{u}(\mathbf{x}_0))$. In our implementation \mathbf{x}_0 is one of the endpoints of the input line segment. We can constrain points to be collinear in the output image with the energy term

$$E_l = \sum_{\mathbf{x} \in V_l} ([\sin(\theta) \cos(\theta)](\mathbf{u}(\mathbf{x}) - \mathbf{u}(\mathbf{x}_0)))^2. \quad (4)$$

where θ may be known or unknown in our optimization depending on whether the user fixed the orientation of the line. Since a line segment is unlikely to pass through any of the mesh vertices exactly, each $\mathbf{u}(\mathbf{x})$ is a bilinear combination of the form $\mathbf{u}(\mathbf{x}) = a\mathbf{u}_{i,j} + b\mathbf{u}_{i+1,j} + c\mathbf{u}_{i+1,j+1} + d\mathbf{u}_{i,j+1}$, where we compute the interpolation coefficients (a, b, c, d) by inverting the bilinear interpolation equations [Carroll et al. 2009].

Vanishing Points. This constraint restricts a set of lines to intersect a common vanishing point. Given a user specified vanishing point \mathbf{u}_{van} and a set of lines L_{van} that pass through it, we define the vanishing point energy term as

$$E_v = \sum_{l \in L_{van}} ([\sin(\theta_l) \cos(\theta_l)](\mathbf{u}_{van} - \mathbf{u}(\mathbf{x}_{0,l})))^2. \quad (5)$$

The coordinates of the vanishing point may be knowns if the user fixes them, or they can be unknowns, which could be useful if the user only wants the set of lines be consistent with each other. However, in all our results we fixed the location of the vanishing points.

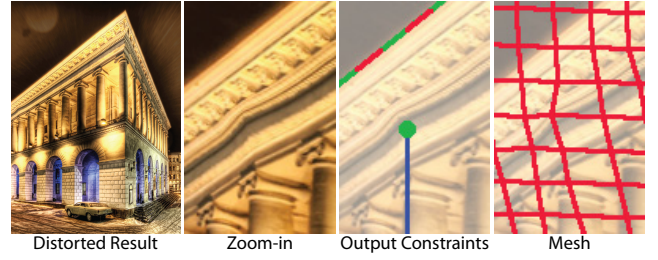


Figure 5: Insufficiently strong weighting of the homography constraint can lead to distortions. This example contains a vertical line constraint inside a homography constraint (see Figure 3). A low weight on the homography constraint produces a wavy distortion on the building facade.

Point Constraints. This energy term restricts the location of a vertex to some fixed target location, and is given by

$$E_p = \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_{target}\|^2. \quad (6)$$

where $\mathbf{u}(\mathbf{x})$ is bilinearly interpolated from neighboring vertices.

Borders. In some cases it is useful to allow the user to fix all or part of the image boundary in order to produce rectangular edges. Without the constraint the resulting image will usually have an irregularly shaped boundary that may be difficult to crop into a rectangle. For a particular side of the boundary we constrain all vertices to map to the same u coordinate (left and right sides) or v coordinate (top and bottom), but we allow vertices to slide along the boundary. The energy function is given by

$$E_b = \sum_{j=1}^h (u_{1,j} - u_{left})^2 I_{left} + (u_{w,j} - u_{right})^2 I_{right} + \sum_{i=1}^w (v_{i,1} - v_{top})^2 I_{top} + (v_{i,h} - v_{bottom})^2 I_{bottom} \quad (7)$$

where I_{side} is a binary indicator variable specifying whether a side is fixed. The variables $(u_{left}, u_{right}, v_{top}, v_{bottom})$ may be knowns or unknowns in the optimization depending on whether the user chooses to fix the locations of the borders or only constrain the boundaries to remain vertical/horizontal. The former approach is useful for ensuring that the aspect ratio of the image is fixed, while the latter approach can be useful for maintaining a rectangular image with arbitrary aspect ratio. However, in many cases constraining the entire border is too limiting, causing distortions in the result, and we often leave some or all of the borders free.

Shape Preservation. The constraints described so far generally only involve a subset of the mesh vertices, and may have multiple valid solutions. We prefer mappings that change smoothly and are locally similarity transformations, because such mappings appear less distorted. Each point in a mapping is locally a similarity transformation, or conformal, if the Jacobian matrix is of the form $J = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$. Equivalently, the mapping must satisfy the Cauchy-Riemann equations $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}$ and $\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$. We can turn these equations into an energy function based on finite differences

$$E_c = \sum_{(i,j) \in V} \left((v_{i+1,j} - v_{i,j}) + (u_{i,j+1} - u_{i,j}) \right)^2 + \sum_{(i,j) \in V} \left((u_{i+1,j} - u_{i,j}) - (v_{i,j+1} - v_{i,j}) \right)^2. \quad (8)$$

We measure the smoothness of the mapping using the Hessian of $\mathbf{u}(\mathbf{x})$, an order 3 tensor containing all the second partial derivatives, which is the rate of change of the Jacobian matrix: $H = \frac{\partial^2 \mathbf{u}}{\partial \mathbf{u}^2}$. If the mapping is perfectly smooth, the Jacobian will be constant and $H \equiv 0$. We again compute second derivatives using finite difference approximations and use the Frobenius norm to give the smoothness

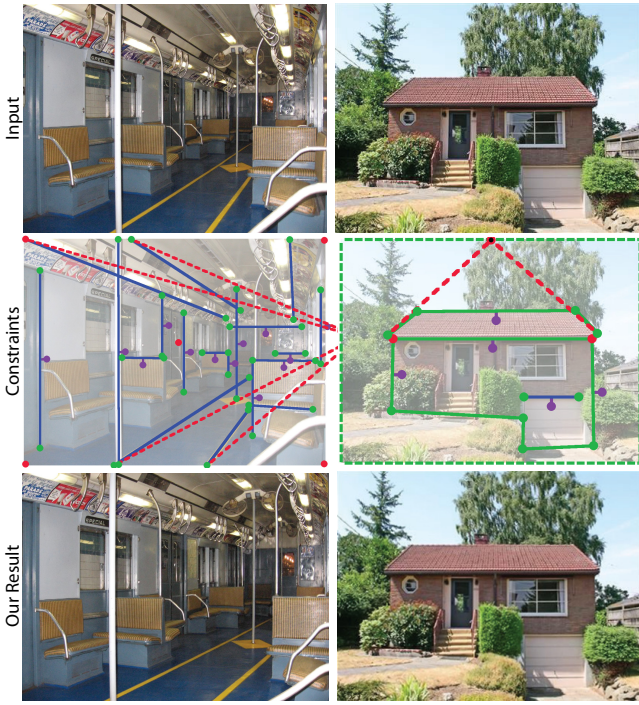


Figure 6: *Left:* This train interior was shot with a wide angle lens. We generate a more telephoto appearance from a viewpoint that is unreachable without exiting the train. We move the vanishing point for the receding parallel lines further from the center of the image. We also add vertical and horizontal constraints to preserve straight lines within the scene. Our result appears to flatten the space, as if it were shot with a longer focal-length lens. *Right:* The input image is a frontal perspective of a house, but the viewpoint is slightly off-center and the vanishing point of the roof is not aligned with the center of the house. We center the vanishing point and fix the misalignment.

energy as

$$E_s = \sum_{(i,j) \in V} \left\| \begin{array}{c} u_{i,j+1} - 2u_{i,j} + u_{i,j-1} \\ v_{i,j+1} - 2v_{i,j} + v_{i,j-1} \\ u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \\ v_{i+1,j} - 2v_{i,j} + v_{i-1,j} \\ u_{i+1,j+1} - u_{i+1,j} - u_{i,j+1} + u_{i,j} \\ v_{i+1,j+1} - v_{i+1,j} - v_{i,j+1} + v_{i,j} \end{array} \right\|^2, \quad (9)$$

where the partial derivatives have been rearranged here into a column vector.

3.3 Optimization

The total energy for the optimization is a weighted sum of the energy terms defined in the previous section.

$$E = w_h^2 \sum_{planes} E_h + w_{hc}^2 \sum_{pairs} E_{hc} + w_l^2 \sum_{lines} E_l + w_v^2 \sum_{vanpts} E_v + w_p^2 \sum_{fixedpts} E_p + w_b^2 E_b + w_c^2 E_c + w_s^2 E_s. \quad (10)$$

We weight the homography, homography compatibility, straight line, vanishing point, fixed point, and border energy terms very strongly compared to the shape preservation energy terms, because they are meant to mimic hard constraints. We determined the weights by experimenting with different values and visually inspecting the results. Although we found a large range of relative weights to work well, for all the results in this paper we use weights

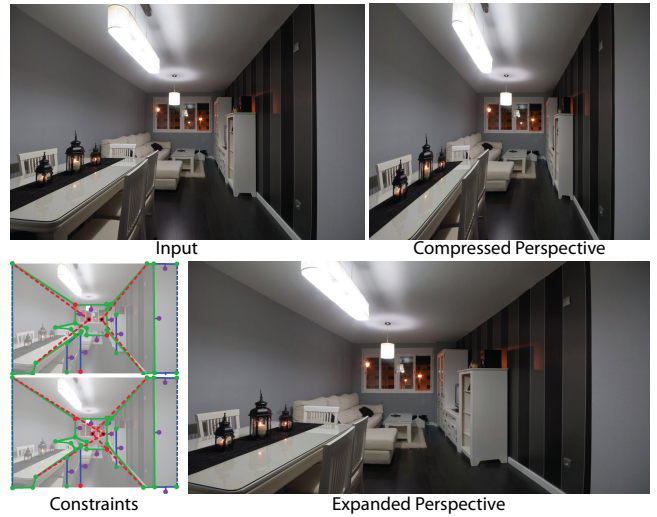


Figure 7: Two perspective manipulations of a room. The left and right walls are parallel and therefore share a vanishing point at the center of the input image. We create two non-photorealistic perspectives by separating the vanishing points of the left and right walls. Pulling each vanishing point in towards its corresponding wall creates the perception of a thinner, longer room, while crossing the vanishing points makes the room appear wider.

of $w_h = w_{hc} = 200$ and $w_l = w_v = w_p = w_b = 100$ for the energy terms that are meant to approximate hard constraints, and $w_s = 12$ and $w_c = 1$ for the weak constraints. The homography and homography compatibility constraints are weighted especially strongly compared to the other strongly weighted constraints. With lower weights on these two constraints we noticed visible deviations from true homographies (Figure 5), and tearing between neighboring planar regions.

In some cases the constraints will directly oppose each other. For example, constraining a region with a homography constraint can make that region less conformal overall, so it is important that the homography constraint be weighted much more strongly than conformality. The combination of these two constraints will result in the homography that is most conformal, while still satisfying other constraints.

None of our constraints are necessarily satisfied absolutely, because even the strongly weighted constraints are defined by least-squares energy terms. Global interactions between the constraints can lead to cases where, for example, vertically constrained lines are not quite vertical, or lines do not intersect an associated vanishing point exactly. Usually in cases where such deviations are noticeable the user has specified a set of constraints that is impossible or difficult to satisfy exactly.

The total energy we minimize is a least-squares function and is non-linear because of the line and homography constraints. We optimize the warp using the Gauss-Newton method with a backtracking line-search procedure to ensure the residual error is reduced at each iteration [Nocedal and Wright 2006]. At each iteration we update the mapping parameters as $\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha \Delta$, where α scales the Gauss-Newton step Δ . We initialize α to 1, and if the error increases in any iteration we divide α by two and recompute \mathbf{x}_{t+1} . The optimization time is dominated by solving the linear system at each iteration; we use the PARDISO sparse direct solver [Schenk and Gärtner 2004]. For a single linear system a significant portion of the solver’s time is spent analyzing the sparsity pattern of the matrix; however, the sparsity pattern is constant for all iterations and we perform this analysis only once as a preprocess. Finally, we render the mesh using bilinear texture-mapping.



Figure 8: *Top:* We transform the building so that it appears as it would under an oblique projection. Parallel lines in the scene remain parallel in our result and the front of the building is parallel to the image plane. *Bottom:* We simulate an orthographic view of downtown Los Angeles by constraining the faces of buildings with homographies and constraining all horizontal lines to be horizontal in the image.

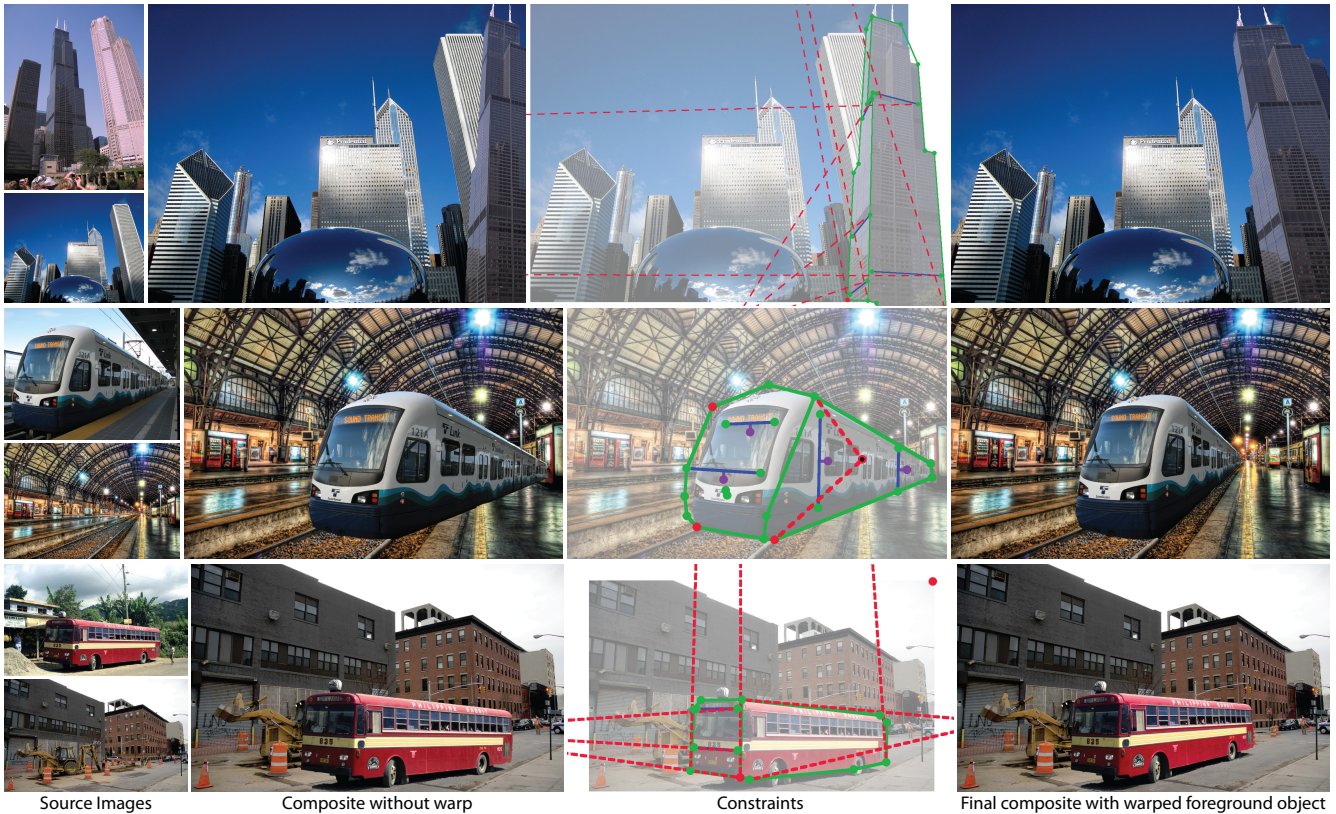


Figure 9: Three compositing examples where we match the perspective of an object to a target scene. *Top:* The black building from the source image looks inconsistent when pasted into the target image, until we warp it so all the vanishing points match. *Middle:* The vanishing point of the lines along the length of the train are shifted to match the vanishing point of the train tracks, and the horizontal lines along the front face of the train are constrained to be level. *Bottom:* All three vanishing points of the mutually orthogonal bus faces are aligned with the three vanishing points of the buildings.

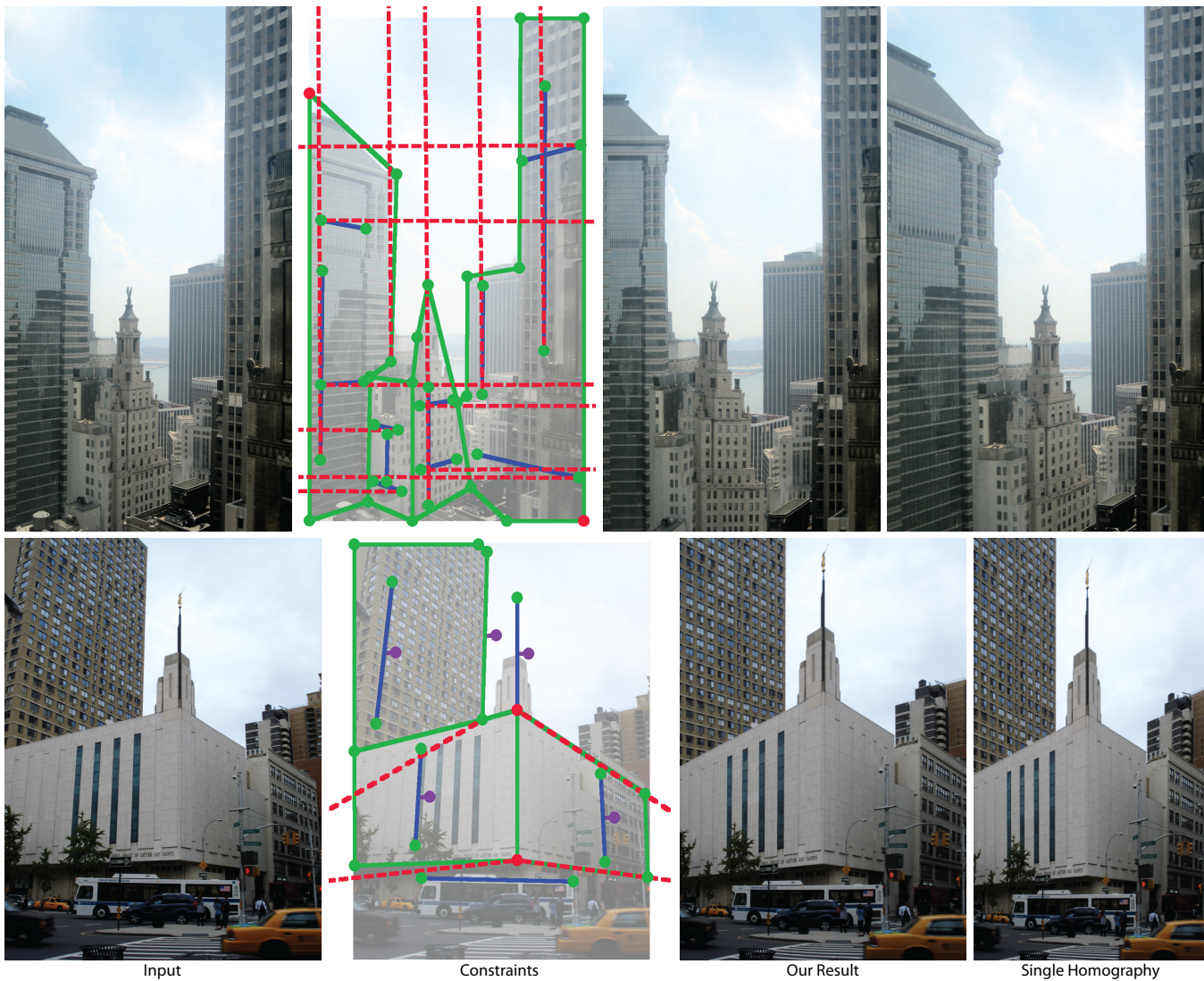


Figure 10: *Top:* We simulate an orthographic projection of a cityscape. We represent the complex geometry roughly with four polygons and move the vanishing points off to infinity. We also create a homography that best matches the constraints, which we compute by drawing a polygon around the entire image. Unlike our result the homography does not achieve the desired orthographic look. *Bottom:* We warp an image so that both buildings are upright and the church is shown from a straight-on viewpoint, where both left and right faces recede at the same angle. To create our result we constrain two vanishing points and several vertical lines. A single homography can satisfy these constraints; however, this homography horizontally compresses the two buildings compared to the input and our result.

4 Results

We have tested our tool on a variety of images and with a number of different goals; our results are shown in Figures 1,3,6-11. We group these results into several different categories.

Viewpoint Shift. In some examples we simulate perspectives that are physically realizable, but were not captured in the input image, either by mistake or because the viewpoints were not reachable. In Figure 1 we show two manipulations of an image to make it look more like a wide-angle photo shot from up-close, or more like a telephoto image taken from far away. In Figure 3 we show various edits which correspond to several different perspectives of a building. Figure 6-left simulates a viewpoint of a subway car that is further away than the confined space would allow. This effect is often called a *dolly-zoom* or *Vertigo* effect when applied to video. Figure 6-right shows a shift to a centered viewpoint on a house which could not be achieved with a simple transformation like a single homography. Figure 10-bottom shows an image of buildings

on a street corner that could have been created by a view camera from the appropriate viewpoint (but not from the viewpoint captured).

Non-photorealistic Perspectives. In other examples, we show perspectives that are not physically realizable by normal cameras, similar to those sometimes used by artists. In Figure 7 we take two walls in a room that should share a single vanishing point and separate their converging lines into two vanishing points. Figure 8-top shows an oblique projection, which is commonly used in architectural renderings, but impossible to capture with a standard camera. In Figure 8-bottom and Figure 10-top we show orthographic projections of cityscapes. The results are not truly orthographic in that occlusions do not change and the sizes of objects still diminish with distance, but the lack of converging lines gives a qualitatively more orthographic effect, as if the camera were zooming in from a distance. These results also exhibit a mix of perspectives, as the larger buildings appear orthographic, but some of the smaller buildings do

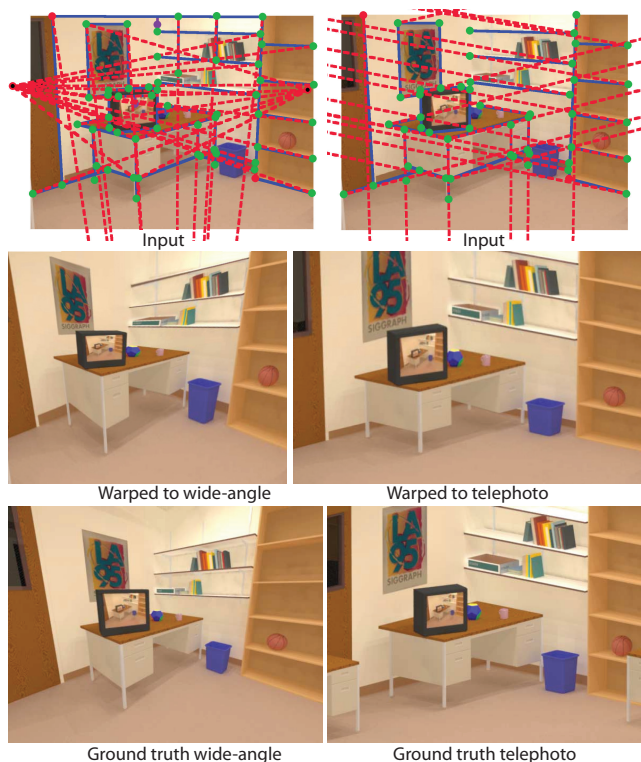


Figure 11: In this example taken from the work of Zorin and Barr [1995], we compare our approach to ground-truth for a rendered computer graphics scene. We begin with an image rendered with a standard focal length and create warps that resemble wide-angle and telephoto renderings. Our warps look plausible, but compared to ground-truth differences can be seen in visibility and the shape of individual objects.

not.

Compositing. Another usage scenario is matching perspective for compositing, which is a frequent problem when combining images [Eismann 2004]. In the three examples in Figure 9, the perspectives of the composited objects do not match the target scenes. We reconcile the disparate perspectives by aligning their vanishing points using our warping tool.

Finally, we compare our results to both standard techniques and ground-truth. For most of our examples the constraints are not satisfiable by a standard transform such as a homography, and attempting to find a homography that best matches our results demonstrates the advantage of our approach (Figure 10-top). One case where the constraints can be met by a single homography is Figure 10-bottom, but our result is still more faithful to the shapes in the input. We can compute the best-fit single-homography version of any result by adding a single polygon around the entire image. In cases where a single homography will suffice, our tool still provides an attractive alternative to the standard interface of dragging four corners of an image or manipulating a virtual camera. Instead, the user directly specifies the desired image-space properties. In Figure 11, we use computer graphics renderings to compare to ground-truth. Our results are visually plausible and look quite similar to the ground-truth renderings, but there are differences in occlusions and object shapes.

Performance. Our tool does not currently respond in real-time to user-specified constraints, but it does respond quickly enough to allow the user to quickly explore many iterations of constraints. For all our results we use a mesh 50 quads wide with height dependent

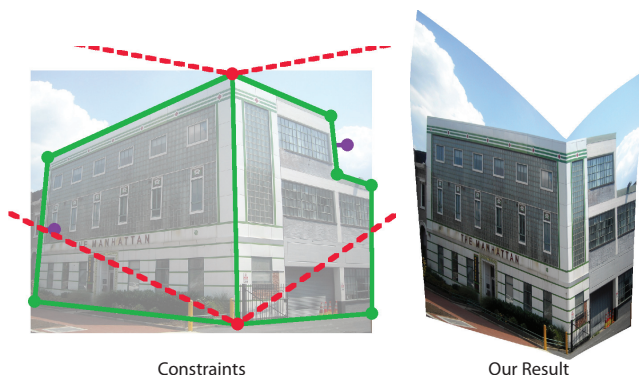


Figure 12: We simulate a view looking down on the building from above. Moving the vanishing points upward gives the desired perspective for the facades. However, the roof is not visible in the input image and the building looks hollow in the result.

on aspect ratio, i.e., 50x50 for a square image. On average we used 2155 vertices and the optimization averaged 12 iterations and 3.37 seconds (run on a 2.4GHz Intel Core 2 Duo). Most of the user’s time is spent specifying constraints, mostly polygon edges and line segments. On average we specified a total of 22 line segments and polygon edges per image.

5 Discussion and Limitations

Our results demonstrate various ways our system can be used to manipulate perspective by interacting with the lines, planes, and vanishing points in an image; however, its applicability is limited. From the outset we designed our tool around an image warping paradigm which ignores three dimensional effects like parallax and thus it can not be used to reproduce these effects, as demonstrated in Figure 12. This example also demonstrates another limitation of our warping technique; it will not generally produce a rectangular image. Most results need to be manually cropped, unless border constraints are used on all four sides. Furthermore, as with all warping techniques, rendering may introduce sampling artifacts, especially in areas undergoing a scale increase. For example, decreasing the foreshortening of a plane will require upsampling of the distant end.

Our tool is designed primarily around manipulating planes and straight lines and is thus well suited to man-made scenes, a property that is also true for the perspective constructions used by painters. Natural scenes do demonstrate perspective effects like foreshortening and diminution, but they do not typically contain strong parallel lines converging to vanishing points. While it is acceptable for more organic textures to be present in the images we manipulate, and such textures can even be useful for hiding distortion, they should not be the focus of the manipulation.

The interface of our tool does have a learning curve, and a user needs to understand the basic principles of perspective construction, such as vanishing points and lines, in order to use it. However, we expect that users interested in manipulating photographic perspective are likely to be familiar with these concepts. Also, it is quite possible for a user to input a set of constraints that are so far from the original image that they cannot be satisfied without extreme distortion. As future work, we would like to explore constraining the interface to only allow feasible constraint configurations. Finally, the shape preservation terms of the energy function are rotation invariant, but are not scale invariant. Mappings that globally reduce image size can produce lower energies for these terms. If the user does not enter at least two fixed point constraints or several boundary constraints, the output will tend to rotate and shrink, possibly to

a single point. As future work we would like to explore additional default constraints that prevent such rotations or shrinkage.

6 Conclusion

Perspective is a powerful tool for depicting a three-dimensional world in a two-dimensional image. However, photographers have not traditionally had the ability to explore its full expressive range as a post-process, at least without significant manual effort. We enable photographic perspective manipulation by combining recent image warping techniques and constraints from projective geometry with an interaction metaphor based on the primitives used by illustrators to construct perspective. Although the results of our image warps are not necessarily physically accurate, by aiming for visual plausibility rather than physical realism we gain several advantages; we sidestep challenging issues of 3D reconstruction and disocclusion handling, and we provide an interface that avoids 3D perspective control in favor of the image-space control metaphors familiar to artists and photographers who may not be experienced in 3D modeling.

Acknowledgments. We would like to thank David Herreman, Trey Ratcliff, Oliver Mallich, David Acevedo Godoy, Chris-Håvard Berge, Chor Ip, Reto Fetz, Oran Viriyincy, John Ward, Tracy Collins, Wally Gobetz, Josh Howell, Flickr user ArtJoy4Ever, and the SLAC National Accelerator Laboratory for use of their photographs. This work is partially supported by 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.
- AGRAWALA, M., ZORIN, D., AND MUNZNER, T. 2000. Artistic multi-projection rendering. In *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, 125–136.
- CARROLL, R., AGRAWALA, M., AND AGARWALA, A. 2009. Optimizing content-preserving projections for wide-angle images. *ACM Trans. on Graph.* 28, 3 (Aug.), 1–9.
- COLEMAN, P., AND SINGH, K. 2004. Ryan: rendering your animation nonlinearly projected. In *NPAA 2004*, 129–138.
- CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *Int. J. Comput. Vision* 40, 2, 123–148.
- CRIMINISI, A., KEMP, M., AND ZISSERMAN, A. 2002. Bringing pictorial space to life: computer techniques for the analysis of paintings. In *Proc. Computers and the History of Art (CHART)*, 77–99.
- D’AMELIO, J. 2004. *Perspective Drawing Handbook*. Dover Publications.
- EISMANN, K. 2004. *Photoshop Masking & Compositing*. New Riders Press.
- GAL, R., SORKINE, O., AND COHEN-OR, D. 2006. Feature-aware texturing. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, 297–304.
- HALL, P. M., COLLOMOSSE, J. P., SONG, Y.-Z., SHEN, P., AND LI, C. 2007. Rtcams: A new perspective on nonphotorealistic rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics* 13, 966–979.
- HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. on Graph.* 24, 3 (Aug.), 577–584.
- HORRY, Y., ANJYO, K.-I., AND ARAI, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proc. SIGGRAPH*, 225–232.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. on Graph.* 24, 3 (Aug.), 1134–1141.
- KOPF, J., LISCHINSKI, D., DEUSSEN, O., COHEN-OR, D., AND COHEN, M. F. 2009. Locally adapted projections to reduce panorama distortions. *Comput. Graph. Forum* 28, 4, 1083–1089.
- KUBOVY, M. 1986. *The psychology of perspective and renaissance art*. Cambridge University Press.
- LORAN, E., AND CÉZANNE, P. 1985. *Cezanne’s composition*. University of California Press.
- NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*, 2nd ed. Springer Verlag, New York.
- POPESCU, V., ROSEN, P., AND ADAMO-VILLANI, N. 2009. The graph camera. *ACM Trans. on Graph.* 28, 5 (Dec.), 1–8.
- RADEMACHER, P., AND BISHOP, G. 1998. Multiple-center-of-projection images. In *Proc. SIGGRAPH*, 199–206.
- SAXENA, A., SUN, M., AND NG, A. Y. 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 5, 824–840.
- 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.
- STROEBEL, L. 1999. *View Camera Technique*, 7th ed. Focal Press.
- 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.
- WILLATS, J. 1997. *Art and Representation: New Principles in the Analysis of Pictures*. Princeton, N.J.: Princeton University Press.
- WOLF, L., GUTTMANN, M., AND COHEN-OR, D. 2007. Non-homogeneous content-driven video-retargeting. In *IEEE International Conference on Computer Vision*.
- WOOD, D. N., FINKELSTEIN, A., HUGHES, J. F., THAYER, C. E., AND SALESIN, D. H. 1997. Multiperspective panoramas for cel animation. In *Proc. SIGGRAPH*, 243–250.
- YU, J., AND McMILLAN, L. 2004. General linear cameras. In *European Conference on Computer Vision*.
- ZELNIK-MANOR, L., AND PERONA, P. 2007. Automating joiners. In *NPAA ’07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, 121–131.
- ZORIN, D., AND BARR, A. H. 1995. Correction of geometric perceptual distortion in pictures. In *Proc. SIGGRAPH*, 257–264.