Eurographics Symposium on Rendering (2006)
Tomas Akenine-Möller and Wolfgang Heidrich (Editors)

Automatic Multiperspective Images

Augusto Román[†] and Hendrik P.A. Lensch[‡]

Stanford University

Figure 1: A portion of a long multiperspective image spanning 2km of 18th Ave in San Francisco, south of Golden Gate National Park. The perspective varies continuously along this image, enabling an arbitrarily long seamless panorama. The perspective is automatically computed to minimize aspect ratio distortions in regions that have large depth variation such as road intersections. This figure is continued in high resolution on the bottom of each page throughout the paper. The image is not perfectly horizontal because the city itself has hills that are captured in the panorama.

Abstract

Multiperspective images generated from a collection of photographs or a videostream can be used to effectively summarize long, roughly planar scenes such as city streets. The final image will span a larger field of view than any single input image. However, common projections used to make these images, including cross-slits and pushbroom projections, may suffer from depth-related distortions in non-planar scenes. In this paper, we use an aspect-ratio distortion metric to compare these images to standard perspective projections. By minimizing this error metric we can automatically define the picture surface and viewpoints of a multiperspective image that reduces distortion artifacts. This optimization requires only a coarse estimate of scene geometry which can be provided as a depth map or a 2D spatial importance map defining interesting parts of the scene. These maps can be automatically constructed in most cases, allowing rapid generation of images of very long scenes.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Picture/Image Generation]: Digitizing and scanning I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality

1 Introduction

Most images we are accustomed to viewing are formed via perspective projection, that is, they correspond to rays passing through a single center of projection. While perspective projection is an accurate model of image formation on the retina and on camera films/sensors, it has some limitations. For example, the visual representation of complete city blocks poses problems for perspective projections. In this case a single perspective image either can not provide the field of view to encompass the scene or is impossible to acquire due to occlusions or physical boundaries.

Multiperspective images offer a promising alternative because they are not bound by these constraints. A multiperspective image is a 2D image in which every region can have a different center of projection. An example of this can be seen in Figure 2, which shows a small portion of a continuous multiperspective image that spans approximately 860m along the street. Multiperspective images retain the simplicity of a single image, avoiding the space, time, and viewing complexities of videos or collections of individual images. They can also efficiently summarize extended scenes.

A common problem inherent to multiperspective images is the distortion introduced when the perspective in the horizontal and vertical dimensions are not the same. Depending on the depth variation in the scene, these distortions can be severe and make portions of the image unusable (see Figure 2(a)). These distortions can be avoided to some extent by carefully adjusting the perspective for each image region. Roman et. al. [RGL04] presented a framework which allows an artist to *manually* specify different perspectives for individual scene segments and to compute smooth transitions between them. Because the selection of these local perspec-

[†] aroman@stanford.edu
[‡] lensch@stanford.edu

© The Eurographics Association 2006.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

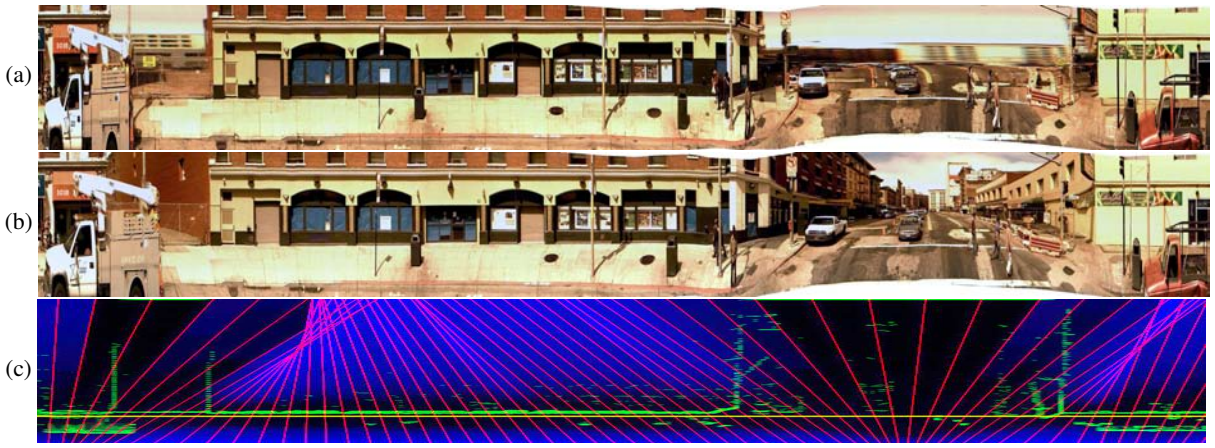


Figure 2: *Multiperspective images generated automatically from a videostream with lateral movement: (a) A pushbroom image, which uses the perspective provided by the input stream in the y-direction and an orthographic projection in x, in order to combine the information of all frames. Notice the difference in perspective in x and y leads to severe distortion at the intersection and alleyway. (b) A multiperspective image generated automatically using our technique. While the perspective in y is still the same, we optimized the perspective in x in each image segment in order to minimize distortion. Notice that this is an image with multiple perspectives – there is a vanishing point down the alleyway and a separate vanishing point down the intersection. (c) A plan view of the street showing the optimized ray directions (red). These rays nearly converge to a perspective at the intersection and again near the alleyway. The yellow line denotes the picture surface. The blue channel is a visualization of the cost function over the entire space. Notice that this set of ray directions minimizes the intersection between the scene points (green) and the error (blue).*

tives is done manually, it is tedious, error-prone, and cannot scale to larger datasets.

With virtual touring such as A9's BlockView [A905] and Microsoft's recent Live Local Preview [Mic06] becoming more common, summarizing extended geographic regions in a single image has significant practical application. When generating images for large projects encompassing an entire city, it is imperative that the process be fully automatic.

In this paper, we present an automatic technique for globally optimizing the local perspectives to minimize the distortion in a multi-perspective image (see Figure 2(b)). We make two contributions:

1. We present a framework for efficiently evaluating the depth-related distortion in multiperspective images introduced by different perspectives in the horizontal and vertical directions.
2. We describe an implementation that optimizes over a large family of multiperspective images to minimize distortion globally for a scene. The optimization requires a rough estimate of the depth distribution in the scene. The optimization is automatic and scalable.

The necessary 3D information about a scene can be derived automatically for most scenes and can be manipulated to emphasize important scene features for which the distortion in the final output is required to be minimal.

After summarizing related work in Section 2, we visualize the effects of various projections in Section 3 and derive an error metric quantifying distortion in Section 4. The optimization algorithm is explained in Section 5. In Section 6, we present results that demonstrate the effective use of our framework to construct multiperspective panoramas for indoor and outdoor scenes. For example, we generated a panorama of a scene spanning over 2km.

2 Related Work

One of the earliest studies of multiperspective images in the graphics community was by Wood et al [WFH*97] and Rademacher et al [RB98]. Their key insight was that multiperspective images can be generated by resampling images from different viewpoints. Their images were geared towards novel view generation rather than producing an image meaningful to humans.

© The Eurographics Association 2006.

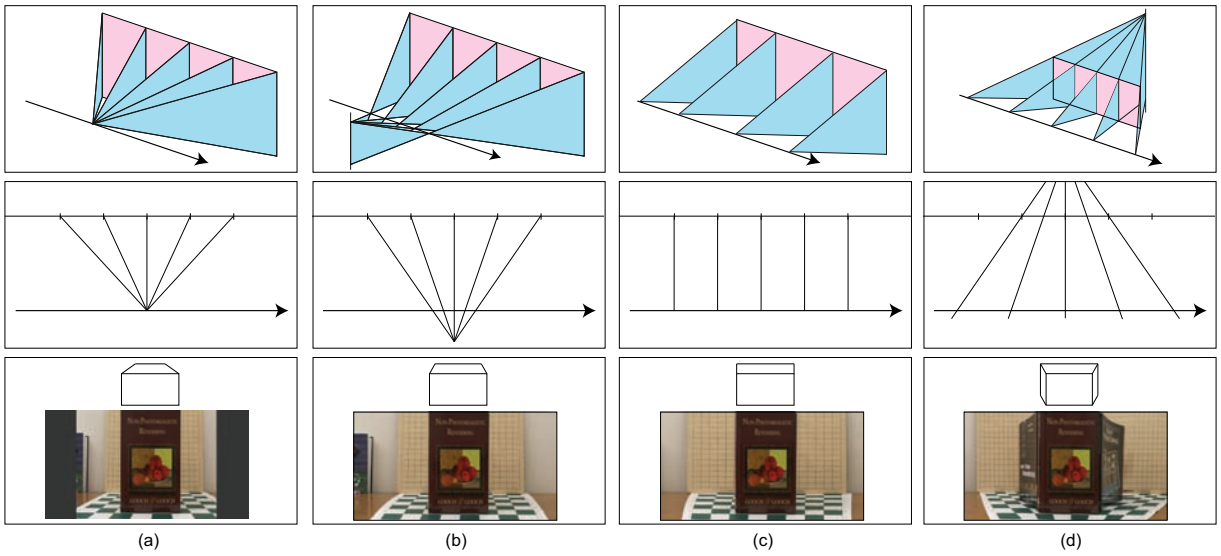


Figure 3: *Effect of varying perspective in x and y. The top row shows a 3D view of the picture surface (in pink), the camera path, and the ray manifold for different kinds of projections. The second row provides a birds-eye view and the last the resulting images when the picture surface is aligned with the front book cover. These are four types of perspectives our system can generate. For all examples the perspective in y is given by the input images. (a) A perspective projection simply corresponds to the original perspective of the camera. Notice the limited field of view. (b) A cross-slits perspective can be used to extend the field of view in x by moving the horizontal center of projection off the camera path. (c) Moving the horizontal center of projection to infinity results in a pushbroom which is horizontally orthographic (notice the checkerboard pattern). (d) In an inverse perspective the center of projection is behind the picture surface. The effect is that objects get larger with increasing distance and both sides of a cube are visible.*

Seitz and Kim [SK03] explored the generation of multiperspective images by slicing a 3D video volume (a stack of frames from a moving video camera). However, not all slices correspond to meaningful descriptions of the scene (such as epipolar images). Feldman et al [FZ03] augmented this approach to find the least distorted slice, but the distortion metric is not related to actual scene structure (unlike ours). In a related approach, Rav-Acha et al [RASP04] showed how to compensate for irregular motion of the video camera by time warping.

The idea of searching for multiperspective images to minimize a distortion metric was used by Swaminathan et al. [SGN03]. They devised a distortion metric assuming the scene consisted of simple geometric shapes such as cylinders, spheres and cones. Recently, Wexler et al [WS05] showed how to generate single-image summaries of a video sequence by choosing a nonlinear manifold through the video volume.

Slices through a video volume are a special case of general linear cameras [YM04]. For our application, we have chosen to work with cross-slits projections [ZFPW03]. In this projection, all rays of the image pass through two slits in space. If the slits are perpendicular, then the resulting image is perspective vertically, but perspective from a different location horizontally. This representation is appropriate for our application because the camera path naturally defines one of the slits of a cross-slits camera.

Feldman [FZWP03] corrects for aspect-ratio distortion in a single cross-slits image by vertically scaling the entire image. Because the distortion depends on the depth of each object in the scene, this global scale can only ensure that objects at a single depth are undistorted. Our system simultaneously optimizes both the structure of multiple cross-slits images (via the ray directions) as well as the overall scale (via the picture surface location).

The use of multiperspective images to visualize extended urban scenes was addressed by Zheng's Route Panoramas [Zhe03]. These are slit-scan images are created by fixing

© The Eurographics Association 2006.

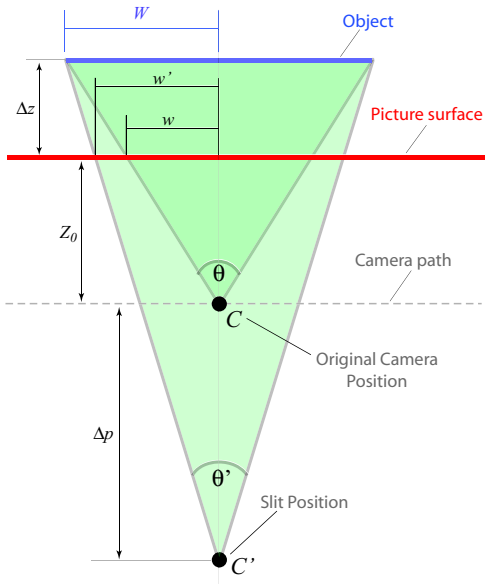


Figure 4: *Distortion due to non-uniform perspective. This is a plan view of a simple scene consisting of only a single, planar object.*

a camera to a vehicle and abutting a single column while driving down the street. Roman et al [RGL04] showed that improvements can be made by varying the column according to scene content, however this was entirely manual and not suitable for large-scale applications. This paper seeks to address that shortcoming by automatically selecting the column based on scene content.

3 Multiperspective Images

We use the multiperspective image paradigm described by Roman [RGL04]. In this case, a multiperspective image is described by a 3D picture surface defined in the camera coordinate system and a corresponding ray manifold that describes the direction that each point on the picture surface faces. A rectangular uniform sampling grid on the picture surface maps rays to the final image. The ray directions locally determine the type of projection. If the rays are parallel, the projection is locally orthographic. If all the nearby rays intersect a single point in space, the projection is locally perspective. In our case, we cannot modify the ray directions vertically – those are constrained to be perspective by our input. Our technique takes advantage of changing the horizontal ray directions and varies local perspective between perspective, cross-slits, and pushbroom to obtain minimum distortion. This ray variation should locally approximate perspective in areas of significant depth variation and blend perspectives in intermediate areas.

The picture surface defines the sampling density of rays in the scene. The scale of objects in the final images is related to the distance from this surface. For a perspective projection, objects behind the picture surface will be smaller in the image than objects in front of it. For our application, we choose to restrict ourselves to a planar picture surface approximately parallel to the camera path to prevent non-uniform object scaling across the scene.

Figure 3 shows examples of several projections and the corresponding images. Notice that the front book cover does not change regardless of the type of projection. This is because the picture surface is aligned with the book cover. As we demonstrate in Figure 7, objects that lie on the picture surface suffer no distortion, regardless of the local perspective. As we will prove in the perspective distortion formula below, the aspect distortion increases with distance from the picture surface.

4 Distortion

The most undesirable effect of non-perspective distortion is a change in the aspect ratio of an object. This *aspect ratio distortion* is caused by the cross-slits projection and is also described in Zomet [ZFPW03]. We interpret this result and show how it is consistent with intuition and real-world results.

4.1 Aspect Ratio Distortion

Figure 4 shows a linear, translating camera trajectory in plan view. The picture surface is a plane facing the camera at a fixed distance Z_0 from the camera path. A single planar object exists in the world with dimensions $2W \times 2H$, having a canonical aspect ratio of $A = \frac{W}{H}$. This object is parallel to the picture surface at a distance Δz away. These are signed distances, and all have positive values in the example in Figure 4.

Consider the projection of the object in a cross-slits image. Figure 4 shows the scene with a second slit placed a distance Δp away from the original camera path, changing the perspective structure horizontally. Notice that this will change the projected width of the object to w'. Using similar triangles, we find that

$$w' = W \frac{Z_0 + \Delta p}{Z_0 + \Delta z + \Delta p} \quad (1)$$

Remember that vertically we still have a perspective image

© The Eurographics Association 2006.

Case	Δz	Δp	D_a
Object on PS, any perspective	0		1
Object not on PS, normal perspective		0	1
Object not on PS, pushbroom		∞	$\frac{(Z_0+\Delta z)}{Z_0}$
Object at infinity, any projection	∞		$\frac{(Z_0+\Delta p)}{Z_0}$

Table 1: *Common aspect ratio distortions. This table lists several common examples of perspective distortion in multiperspective images. Notice that if an object is aligned with the picture surface (PS), then there is no distortion regardless of the perspective. Similarly, in a normal perspective there is no distortion regardless of the object placement. For pushbroom images, distortion is proportional to the distance from the picture surface.*

and so the projected height of the object will not change:

$$h' = h = H\frac{Z_0}{Z_0+\Delta z} \tag{2}$$

The aspect ratio of the object under this projection is then

$$a' = \frac{w'}{h} = A\frac{(Z_0+\Delta z)(Z_0+\Delta p)}{Z_0(Z_0+\Delta z+\Delta p)} \tag{3}$$

We define the *aspect ratio distortion*, D_a, as the change in the aspect ratio:

$$D_a = \frac{a'}{a} = \frac{(Z_0+\Delta z)(Z_0+\Delta p)}{Z_0(Z_0+\Delta z+\Delta p)} \tag{4}$$

This metric is the basis of our cost function used to evaluate the perspective distortion in a multiperspective image.

4.2 Discussion

We now verify that the distortion metric is consistent with several common cases (see summary in Table 1). Consider a perspective image where $\Delta p = 0$. The numerator and denominator are then equal and $D_a = 1$, regardless of the values of Z_0 or Δz, confirming that perspective images have no perspective distortion. Objects on the picture surface are described by $\Delta z = 0$, and again $D_a = 1$ regardless of the type or projection defined by Δp. This explains why the book cover (Figure 3) and the building front (Figure 7) suffer no distortion; in both cases they are aligned with the picture surface. For a pushbroom image, Δp approachs ∞. In this case, Equation 4 simplifies to $D_a = \frac{(Z_0+\Delta z)}{Z_0}$ and thus the distortion is linear with the object's distance from the picture surface.

The relation described in Equation 4 also applies to scenarios where the object, picture surface, and camera path are not aligned and can be evaluated by integrating the local distortion for all points across the object. Thus, it is an appropriate metric for quantifying the overall distortion in any multiperspective image.

5 Optimization

We now describe the optimization framework for minimizing the distortion error in a multiperspective image.

5.1 Input data

We make three basic assumptions about the input data. The first is that the input images are from a moving video camera. This results in a 3D video dataset instead of a 4D lightfield, restricting the set of multiperspective images that can be created. Our optimization adjusts only the horizontal perspective with the vertical perspective being fixed.

Next, the camera trajectory must be known. Structure-from-motion software can be used to extract this directly from the video. External constraints (such as placing the camera on a track) or auxiliary sensors (GPS, accelerometers, etc.) can also be used to robustly determine the camera path and improve accuracy.

Finally, our algorithm depends on having some notion of the 3D scene structure. For example, the sparse point cloud generated by most structure-from-motion algorithms is sufficient. External sensors such as laser range finders can also provide this information quickly and reliably. Alternatively, the user may manually specify important regions in the scene if the 3D structure is not appropriate.

5.2 Cost function

We define a cost function that converts the aspect ratio distortion into an error:

$$E = \begin{cases} D_a - 1 & D_a >= 1 \\ \frac{1}{D_a} - 1 & 0 <= D_a < 1 \\ \lambda - \frac{1}{D_a} & -1 < D_a < 0 \\ \lambda - D_a & D_a <= -1 \end{cases} \tag{5}$$

This relationship gives equal error to an object with half its normal aspect ratio and an object with twice its normal aspect ratio. Values of $\lambda > 1$ penalize negative aspect ratios where objects are horizontally inverted. We have experimentally determined $\lambda = 10$ to be appropriate to suppress any significant inversion in the optimization.

5.3 Optimization

The optimization is initialized with a pushbroom image placing the picture surface at an initial distance Z_0 from the camera path, as shown in Figure 5(a).

The picture surface is then discretized into N equal-length

© The Eurographics Association 2006.

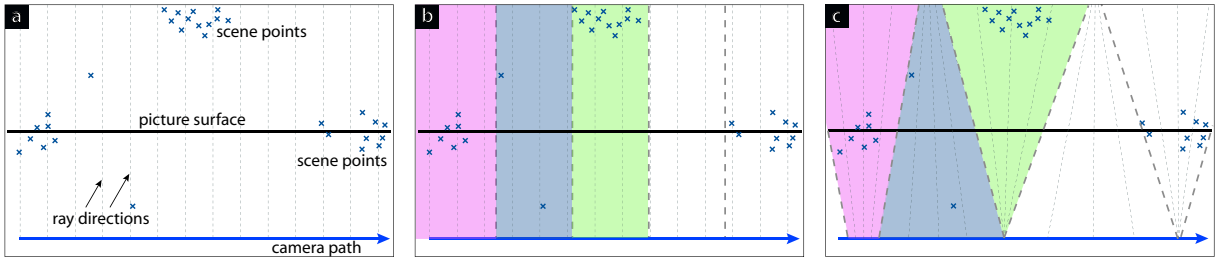


Figure 5: *The optimization starts with an initial pushbroom image defined by a picture surface and a set of parallel rays as shown in (a). (b) The picture surface is then divided into several segments. For each segment, we compute the error of all scene points that are projected into that segment, as indicated by the shaded regions. (C) The type of perspective projection in each segment is defined by the angle of that segment's boundaries. These angles are altered to minimize the total aspect ratio distortion in the final image. The position of the picture surface can be adjusted in combination with the ray directions to further minimize distortion as described in Section 5.3.*

segments. Each of these segments represents a portion of the picture surface with a single type of perspective projection. The type of perspective projection is defined by the angle of the boundaries between the two segments. To enforce that the perspective across segments varies smoothly, it is necessary for neighboring segments to use the same angle at their shared boundary. Therefore the perspective of the N segments can be parameterized by the angles at the $N+1$ boundaries.

These boundaries are defined in terms of the angle of the boundary with respect to the picture surface, θ_i. For example, the initial pushbroom image is described by $\theta_i = \frac{\pi}{2}$. The intersection of the two boundaries corresponds to the point C' in Figure 4, and thus we can directly compute the local perspective Δp for each of the segments. We can therefore compute the error of any scene point in a particular segment. The range of values for θ_i is limited by the field of view of the input imagery.

The error of a single segment is computed by summing the error from all scene points that project into that segment as indicated by the shaded regions in Figure 5(b). The error of the entire image is simply the sum of the errors of each of the segments. The optimization therefore finds the set of (Z_0, θ_i) that minimize the overall distortion. Optionally, Z_0 can be fixed and the set of boundary orientations that minimizes the error for the picture surface at that location can be found.

This can be described mathematically as follows. A given set of ray directions θ_i $(i = 0...N+1)$ defines N segments S_i and corresponding local perspectives Δp_i. Within a particular segment, Δp and Z_0 are constant, and therefore the error of a point q is described by $E(D_a(\Delta z_q, \Delta p_i, Z_0))$ where Δz_q is the orthogonal distance of q to the picture surface. The minimization therefore is:

$$\underset{\theta_i, Z_0}{\operatorname{argmin}} \left(\sum_S \sum_{q \in S_i} E(D_a(\Delta z_q, \Delta p_i, Z_0)) \right) \tag{6}$$

5.4 Implementation

The optimization was implemented in C++ and used the Opt++ [Mez94] numerical optimization library to perform a bounded Newton optimization. There are several techniques that we used to make computing the error function faster and easier.

Instead of manually computing the derivative of the cost function, we take advantage of an automatic differentiation technique described by Piponi [Pip04] which is faster and more accurate than numerical differentiation. When evaluating the aspect ratio within a given segment, Δp_i is determined by the intersection of the two ray directions of the segment:

$$Z_0 + \Delta p_i = -\frac{D}{2} \left(\frac{\cos(\theta_i + \theta_{i+1}) - \cos(\theta_{i+1} - \theta_i)}{\sin(\theta_{i+1} - \theta_i)} \right) \tag{7}$$

where D is the length of the picture surface segment. Unfortunately, this can cause a divide-by-zero error in the common situation that the rays are parallel. We avoid this intermediate computation by substituting this expression into the equation 4 and simplifying to obtain:

$$D_a = \frac{Z_0 + \Delta z}{Z_0} \frac{1}{K+1} \tag{8}$$

© The Eurographics Association 2006.

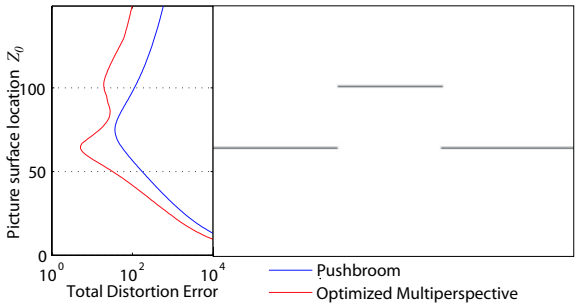


Figure 6: *This diagram shows how the total distortion error of a multiperspective image depends on the picture surface location Z_0. On the right is the importance map for a simple synthetic scene consisting of only three planes. The graph on the left shows the error of both the initial pushbroom configuration and the optimized multiperspective configuration when the picture surface is fixed at the corresponding row in the importance map.*

where

$$K = \frac{2\Delta z}{D}\left(\frac{\sin(\theta_{i+1}-\theta_i)}{\cos(\theta_i+\theta_{i+1})-\cos(\theta_{i+1}-\theta_i)}\right) \qquad (9)$$

Notice that the distortion in Eq. 4 depends only on the depth (Δz) and local perspective (Δp) of that point. It does **not** depend on the height of that point above the ground plane. We can therefore take the input 3D scene geometry and project it down into a 2D histogram parallel to the ground plane. Instead of searching for scene points that fit within each segment, we simply compute the bounds of the segment within the histogram and compute the error for that region.

This histogram is simply an image analogous to an overhead density map of the scene. By default all scene points contribute equally to the error function. The user may optionally augment the histogram with a 2D importance map that modulates the histogram. This allows the user to manually emphasize or de-emphasize regions of the scene. We have used this in Figure 9 to guide the optimization towards important regions of the scene.

Assuming the picture surface is aligned with the x axis of the histogram, the distortion is constant along x within each segment. We can efficiently integrate the contribution of each row using a summed area table [Cro84].

Smaller values of N reduce the optimization space and therefore give significant computation reduction along with fewer local minima. Larger values of N allow the possibility for fine-grained perspective adjustments at the cost of increased computation and more local minima. To avoid local minima, we perform a multiresolution optimization. Both the importance map and the number of segments along the picture surface are reduced hierarchically.

These implementation techniques make the optimization very fast. On a dual Xeon 3.2GHz PC with 1GB of ram, the entire city street example in Figure 1 takes just over 5 minutes (302 seconds) to optimize, specifying the varying perspective for a 600 megapixel image (325k pixels wide).

6 Results

We have applied our technique to indoor and outdoor scenes: a room inside a museum (Figure 9), Mission Street (presented in Figure 2), and 18$^{\text{th}}$ Ave (Figure 1) in San Francisco. The museum scene was acquired moving a sideways looking video camera along a straight line parallel to the scene and spans approximately 20m. The camera path was extracted from the video using the freely-available Voodoo Camera Tracker [Uni05] structure-from-motion software which also outputs a sparse 3D point cloud. The street scenes were captured using a sideways-looking, high-speed camera (Basler A504kc) in a car driving in normal traffic (0-20mph). The camera pose was estimated using accelerometers and GPS. The 3D scene structure was acquired using time-of-flight range finders. The Mission Street image spans about 860m while the 18$^{\text{th}}$ Ave image spans about 2088m. It is possible to use SFM to generate the required projection matrices and scene estimates for the street scenes, however many SFM algorithms do not handle extremely long, linear scenes robustly.

Table 2 summarizes the scene size, the number of input frames, number of optimization segments, output sizes, and the timings for the perspective optimization. The only user-selectable parameter is the number of segments to optimize, which should be chosen according to the scene length. Due to the hierarchical optimization and the use of summed area tables, the optimization performs well even for the larger street scenes.

6.1 Discussion

In all three scenes the artifacts due to aspect ratio distortion after optimization have been reduced to a minimum compared to the pushbroom panoramas. We will now focus on the performance of the optimization by analyzing special cases in the Mission Street panorama. Figure 7 visualizes the dependence of the error function on the placement of the picture surface z_0. If the scene is at the picture surface (Fig. 7(a) and (b)), there is no aspect ratio distortion no mat-

© The Eurographics Association 2006.

scene	size (in m)	# input frames	histogram size	# segments	output resolution	optimization (in min.)
Museum Scene	20	941	896×425	64	2522×438	0:35
Mission Street	860	21520	2482×183	512	61320×1000	2:16
18^{th} Ave	2088	61092	7312×200	512	325240×11538†	5:02

Table 2: *Facts about the different scenes. Notice that the optimization is fast even for large scenes.*
† Because 18^{th} Ave is not flat, large sections of this image are blank. The actual image area is approximately 600MP instead of 3.4GP.

ter which perspective is selected. For surface points off the picture surface, (c) and (d), the error can only be minimized by approaching the original camera perspective. If the area of the depth deviation is too large to be covered by a single input image, as in case of Figure 8(a), our optimization resorts to the closest cross-slits perspective that spans the entire gap.

The proposed error metric only accounts for aspect ratio distortion. This has the effect that a sheared perspective contributes the same error as a more symmetric setup. In Figure 8(a) all cars are shown from an oblique view. Notice that the oblique view in fact does not introduce any further distortions. Our optimization does not prefer one over the other and has the freedom to chose whatever shear fits best in order to optimize for neighboring regions.

Our approach is unaware of occlusions (Figure 9(bottom)). In this case our algorithm during optimization may consider the error even for an object that will be occluded in the final output. An optimization considering occlusions would have some impact on the resulting shear. One expects that the shear will be chosen such that foreground objects occlude as many scene points which are off the picture surface as possible.

An artifact due to an incorrect estimate of scene geometry is presented in Figure 8(b). Due to the limited range of the 3D range finders, the building in the background does not show up in the depth histogram and the algorithm allows some rays to cross in front of the building resulting in multiple copies of the building in the output image. In the pushbroom image the building is visible only once.

7 Conclusions & Future work

In this paper we discussed an error metric that expresses the aspect ratio distortion inherent to multiperspective images. Based on this error metric we developed a completely automatic, efficient and scalable optimization framework that produces panoramas with minimized distortion and effectively reduces the artifacts otherwise visible in pushbroom panoramas.

The method uses a rough estimate of the depth variation in the scene, currently in the form of a histogram of scene points in the *x*-*z* plane. One can easily modify the influence of particular scene objects on the optimization by manipulating their contribution to the histogram. By analyzing the input sequence one might be able to perform an object segmentation and determine which scene parts are important and should thereform have minimum distortion. Similarly, one could imagine detecting and emphasizing scene regions with highly regular textures for which aspect ratio distortion produces a higher visual impact than for non-textured regions.

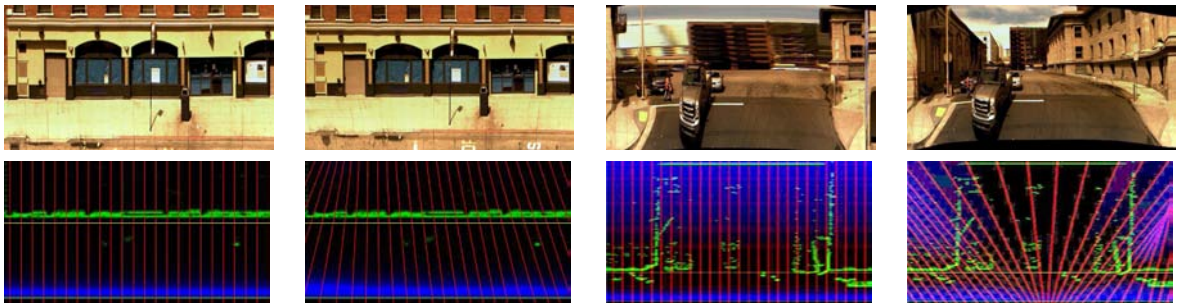
In the future we plan to extend our work to correctly handle occlusions. Instead of evaluating the error metric on a projected depth histogram, one could evaluate it for each rendered pixel in the final output image. In this way, pixels that are occluded would not contribute to the overall error. While this approach might yield even more precise results, it is inherently much costlier to compute than our presented technique.

We would also like to extend the optimization beyond cross-slits projections. This would alleviate a fundamental limitation of this approach that any variations in scene depth across the vertical axis of the image lead to distortion, but would also require extending the rendering system beyond simple cross-slits images.

Another interesting aspect to investigate is to allow for non-planar picture surfaces which could be used to emphasize or enlarge specific features in the scene. However, it is not yet clear what kinds of artifacts will be introduced by the change in sampling resolution and the resulting change in relative size of real-world objects in the final output. While our optimization algorithm is flexible enough to handle even curved picture surfaces, our current error metric does not account for this kind of distortion introduced by the variation in the output sampling.

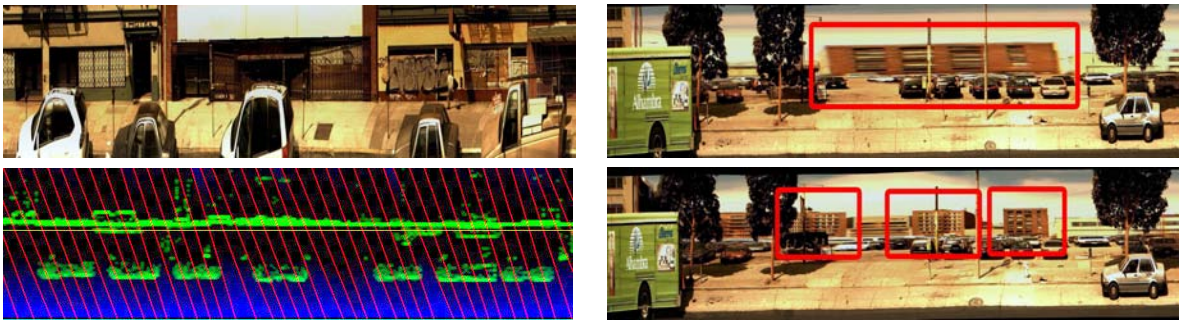
Acknowledgements We thank Marc Levoy for insightful discussion and guidance throughout this research. Thanks also to Ron Yeh for help with figures and to Vaibhav Vaish, Sarah Harriman, and Dan Morris for proofreading drafts. This work is possible due to a grant

© The Eurographics Association 2006.



(a) (b) (c) (d)

Figure 7: *These figures indicate how the image is effected by changes in Δz. The bottom row shows the ray directions (red), the picture surface (yellow), the scene histogram (green) and the distortion error (increasing shades of blue). Portions of the scene that are aligned with the picture surface, such as the store front in (a) and (b), are not affected by the ray directions. In contrast, the error of regions that have significant depth variation, such as in (c) and (d), is sensitive to the ray directions.*



(a) (b)

Figure 8: *Limitations of our algorithm. (a) The optimization is unable to eliminate the distortion of the cars because the cars form a continuous region with large depth variation. The best solution therefore is an extreme cross-slits – approaching a pushbroom perspective across the entire region. Furthermore, our error metric does not account for shear in the projection. The shear is influenced by neighboring regions not shown. (b) On top is a pushbroom image showing a single, distant, distorted building. Unfortunately the building was too far for the 3D scanner to detect and therefore was not considered in our optimization. The resulting optimized rays cross in front of the building causing a triple image to occur, as highlighted in the bottom multiperspective image.*

from the Max Planck Center for Visual Computing and Communication (BMBF-FKZ01IMC01) and funding from Google Inc.

References

[A905] A9: Blockview, 2005. http://maps.a9.com.

[Cro84] CROW F. C.: Summed-area tables for texture mapping. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1984), ACM Press, pp. 207–212.

[FZ03] FELDMAN D., ZOMET A.: *Least Distorted Mosaics*. Tech. Rep. 2003-71, Hebrew University, 2003.

[FZWP03] FELDMAN D., ZOMET A., WEINSHALL D., PELEG S.: New view synthesis with non-stationary mosaicing. In *MIRAGE* (2003).

[Mez94] MEZA J. C.: *Opt++: An Object-Oriented Class Library for Nonlinear Optimization*. Tech. Rep. SAND83-8212, Sandia National Laboratories, Livermore, CA, March 1994.

[Mic06] MICROSOFT: Live! Local Preview, 2006. http://preview.live.local.com.

© The Eurographics Association 2006.

Figure 9: *Museum scene. (top) Pushbroom. The sculptures on the left and the hallway on the right are severely distorted. (middle) Automatic perspective. The optimization reduces the distortion but is unable to distinguish artwork and other scene content. There is still some distortion in the hallway. (bottom) Importance map. Adjusting the histogram to emphasize the relevant artwork results in less distortion for these objects. However, the overall geometric distortion has been increased. Notice that our algorithm does not consider effects of occlusions as seen in the sculptures on the left.*

[Pip04] PIPONI D.: Automatic differentiation, c++ templates, and photogrammetry. In *The Journal of Graphics Tools* (2004), vol. 9.

[RASP04] RAV-ACHA A., SHOR Y., PELEG S.: Mosaicing with parallax using time warping. In *Image and Video Registration* (Washington, DC, July 2004).

[RB98] RADEMACHER P., BISHOP G.: Multiple-center-of-projection images. *Computer Graphics 32*, Annual Conference Series (1998), 199–206.

[RGL04] ROMAN A., GARG G., LEVOY M.: Interactive design of multi-perspective images for visualizing urban landscapes. In *Proceedings of the IEEE Vis 2004* (2004).

[SGN03] SWAMINATHAN R., GROSSBERG M. D., NAYAR S. K.: A perspective on distortions. In *CVPR (2)* (2003), pp. 594–601.

[SK03] SEITZ S. M., KIM J.: Multiperspective imaging. *IEEE Comput. Graph. Appl. 23*, 6 (2003), 16–19.

[Uni05] UNIVERSITY OF HANNOVER: Voodoo camera tracker, 2005. `http://www.digilab.uni-hannover.de/docs/manual.html`.

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

[WS05] WEXLER Y., SIMAKOV D.: Space-time scene manifolds. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)* (2005), vol. 1, pp. 858–863.

[YM04] YU J., MCMILLAN L.: A framework for multiperspective rendering. In *Eurographics Symposium on Rendering* (Norrkoping, Sweden, 2004).

[ZFPW03] ZOMET A., FELDMAN D., PELEG S., WEINSHALL D.: Mosaicing new views: The crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 6 (2003), 741–754.

[Zhe03] ZHENG J. Y.: Digital route panoramas. *IEEE MultiMedia 10*, 03 (2003), 57–67.

© The Eurographics Association 2006.