Seamless Stitching using Multi-Perspective Plane Sweep

Sing Bing Kang, Richard Szeliski, and Matthew Uyttendaele

June 2004

Technical Report

MSR-TR-2004-48

Microsoft Research

Microsoft Corporation

One Microsoft Way

Redmond, WA 98052

http://www.research.microsoft.com

1 Introduction

With digital cameras becoming increasingly cheaper and more accessible to consumers, there exists a strong need for better automated digital image processing. Red-eye removal and color correction are two of the more popular image processing capabilities that have been incorporated into commercial software packages aimed at photographers. Another popular application is *image mosaicking* or *stitching*, which can generate a panoramic image from multiple overlapping images of the same scene, possibly taken at (slightly) displaced locations.

The biggest problem in image mosaicking is ghosting due to the presence of parallax. While others have used dense sampling to overcome this problem (e.g., MCOP [Rademacher and Bishop, 1998] and manifold projection [Peleg and Herman, 1997]), there has been no satisfactory solution to the case of sparse sampling, where the overlap between images is 50% or less and parallax is significant. Shum and Szeliski [Shum and Szeliski, 2000] use a local patch-based deghosting technique, but it does not address the issue of significant parallax. This is because their technique is based on local 2D matching (with arbitrary 2D motion estimation), and their corrective warping may produce strange-looking artifacts.

Three other approaches related to ours are [Kumar *et al.*, 1994, Rousso *et al.*, 1998, Zhu *et al.*, 2001]. Kumar *et al.* [Kumar *et al.*, 1994] compute a dense parallax field from a collection of images with highly overlapping fields of view. They call this representation a "surface plus parallax" or "mosaic plus parallax". A novel view can be generated from this representation, but it is not clear how to handle long image sequences or images with small amounts of overlap.

Using a technique described in [Peleg and Herman, 1997], Rousso *et al.* [Rousso *et al.*, 1998] construct appropriately sampled strips from images based on camera motion. One of the major differences is that if the images are not sampled densely enough, additional intermediate images are generated using optic flow. The strips are then sampled from both original and synthetically generated intermediate images. However, only a small fraction of the intermediate images is ultimately used, resulting in a significant amount of wasted processing.

Zhu *et al.* [Zhu *et al.*, 2001], on the other hand, produce a mosaic by finding matches within the overlap region and triangulate the overlap region using the matched points and boundary points. The depth information provided by the matches is used for warping and image mosaicking. Each facet in the triangulation is assumed planar. Moreover, as in other cases, a relatively dense image sampling is assumed.

In our work, we assume we may be given as few as two images, possibly with only a small overlap region. We handle this problem by considering it from a geometric point of view. Given

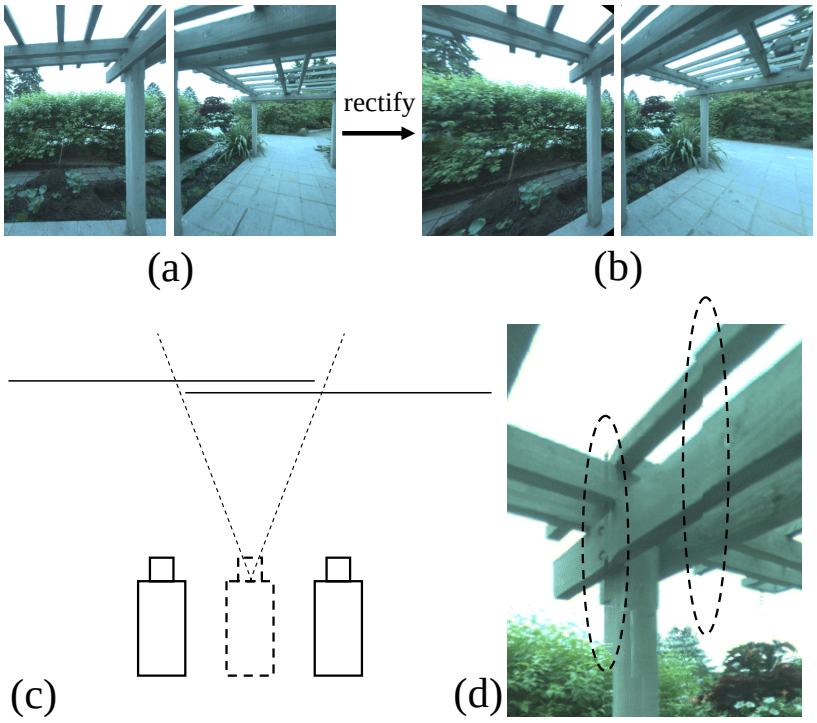


Figure 1: *Problem with using only one intermediate viewpoint.* (a) Input image pair, (b) same input image pair after rectification, (c) using an intermediate virtual view to remove seam, and (d) undesirable result of using only one intermediate viewpoint (notice the discontinuities circled).

a pair of images to stitch, we compute their relative orientation and rectify these images such that the epipolar lines are horizontal. We then find their overlap region which is to be de-ghosted.

One direct method for generating a seamless overlap would be to use the view morphing technique of Seitz and Dyer [Seitz and Dyer, 1996]. Unfortunately, their technique requires manual correspondence. Another direct method would be to apply stereo on the overlap region and generate an intermediate virtual viewpoint, as shown in Figure 1. However, this method creates two discontinuities. While the overlap region (virtual region) is seamless, it is discontinuous with respect to the left non-overlapped image due to the change in camera viewpoints. The same overlap region is also discontinuous with respect to the right non-overlapped image for the same reason.

To avoid the discontinuity problem, we use multiple intermediate cameras instead, as shown in Figure 2. As before, the non-overlapped regions are associated with their respective original camera locations. However, now the *columns in the overlapped area are associated with virtual camera locations between the two original camera locations*. This minimizes object distortion (which is unavoidable unless full 3D is known or recovered everywhere) while producing a practically seamless

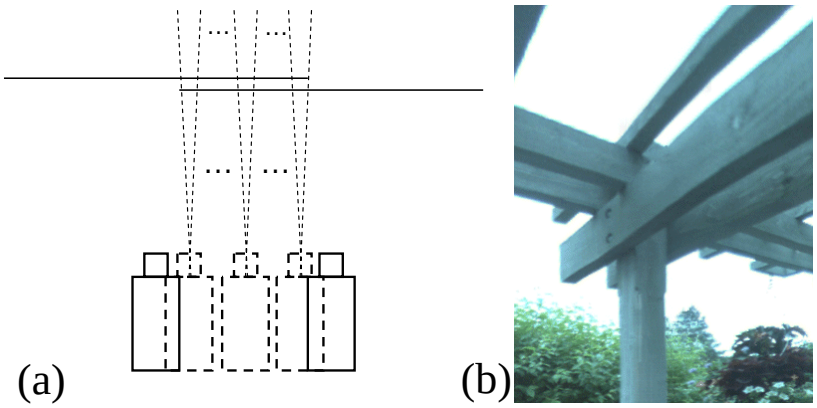


Figure 2: *Our approach.* (a) Using multiple intermediate virtual viewpoints, and (b) result of using multiple intermediate viewpoints (notice the discontinuities are gone).

composite, as shown in Figure 2(b). Computing the appearance of each column within the overlapped region is accomplished using a plane sweep [Collins, 1996, Kang *et al.*, 2001]. We call this technique *Multi-Perspective Plane Sweep* (MPPS), because the resulting image is a Multiple Center of Projection (MCOP) image [Rademacher and Bishop, 1998]. The MPPS can also be thought of as a manifold projection [Peleg and Herman, 1997] of a view morph [Seitz and Dyer, 1996].

We also propose another technique, *Stretching Stereo* (SS), that is significantly faster and usually works just as well. However, it has two disadvantages: The first is more conceptual—it does not have a strict geometrical interpretation as MPPS. Second, the disparities being swept along a scanline (or epipolar line) are graduated, which will cause problems if objects with large disparities occur very close to the edge of the overlap region. Section 6 has a more detailed description of this particular problem.

2 Plane Sweep Stereo

The idea of plane sweep was originally proposed by Collins [Collins, 1996], in the context of computing depth of edge features. It has also been used by Kang *et al.* [Kang *et al.*, 2001] for dense stereo. Figure 3 illustrates the idea of the plane sweep. The matching errors are computed over all the pixel locations of the reference cameras at all the hypothesized depths, resulting in the Disparity Space Image (DSI). The depths can then be obtained by applying winner-take-all or using a global optimization technique such as the graph cut [Boykov *et al.*, 1999]. One simple interpretation of the plane sweep is that we are stepping through disparity values per pixel and voting based on the

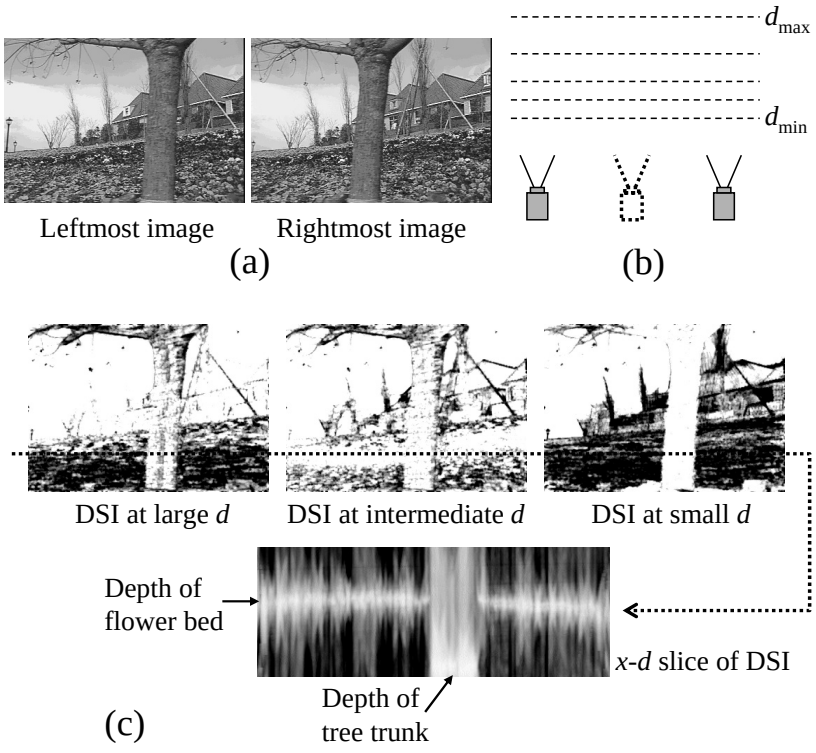


Figure 3: *Plane sweep stereo.* (a) Two of the input images, (b) reference view (dotted camera) and hypothesized set of depths (ranging from d_{min} to d_{max}), and (c) the matching errors at different depths. The darker the pixel, the larger the matching error. The volume of matching error data as the function of (x,y,d) is called the Disparity Space Image (DSI).

minimum image color difference.

3 Multi-Perspective Plane Sweep

The MPPS approach is illustrated in Figure 4. For a pair of images to stitch, the steps are: rectify, find the area of overlap, subdivide the area of overlap and assign each sub-area to a different (intermediate) virtual camera position, and finally, plane sweep to assign appearance to each sub-area. The block diagram depicting these steps is shown in Figure 5. In more detail:

1. *Rectify the images.* Image rectification is a common step in stereo matching used to warp the input images such that the epipolar lines correspond to the scanlines. This is done to speed up the plane sweep step. In our examples, we precalibrate the cameras to compute their poses. The rectified image plane is picked to be exactly between those of the input camera pair. If

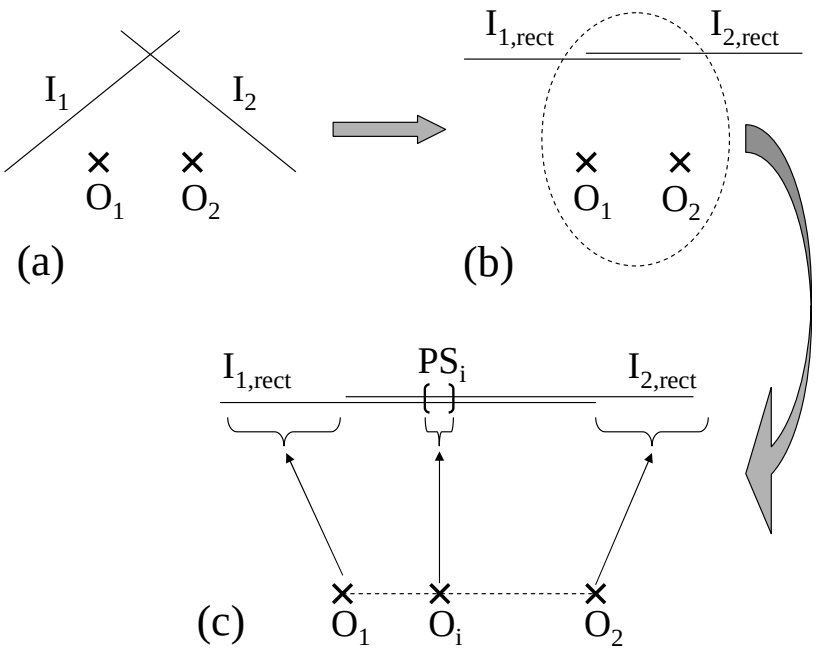


Figure 4: *Multi-perspective plane sweep idea:* (a) Original two images I_1 and I_2 with centers O_1 and O_2, respectively, (b) after rectification, yielding images $I_{1,rect}$ and $I_{2,rect}$, and (c) a close-up of (b), showing contributions at different image centers. The non-overlapped areas are unaffected, while plane sweeps at different overlapped areas PS_i are done at different virtual viewpoints O_i.

the camera parameters are unknown, any of the various techniques to extract the epipolar geometry can be used, followed by image rectification [Loop and Zhang, 1999].

2. *Compute the overlap region.* This is done by simply taking the intersection between the rectified images.

3. *Plane sweep per column.* The process in this step is as described in Section 2, except that the plane sweep is carried out on a per column (or group of columns) basis, and each column (or group of columns) correspond to a *different* virtual camera center. The cost metric is just sum-of-squared difference (SSD) between the shifted left and right images, and a 3×3 window is used to aggregate the matching error. We used 12 disparity levels in all our examples.

4. *Blend colors.* In this step, the computed depths from the plane sweep are used to retrieve corresponding colors from the input images by inverse sampling. Given a pixel depth at a virtual camera position, we can compute its mapped location at the left and right images. The color values are then bilinearly interpolated. These colors are then blended to produce

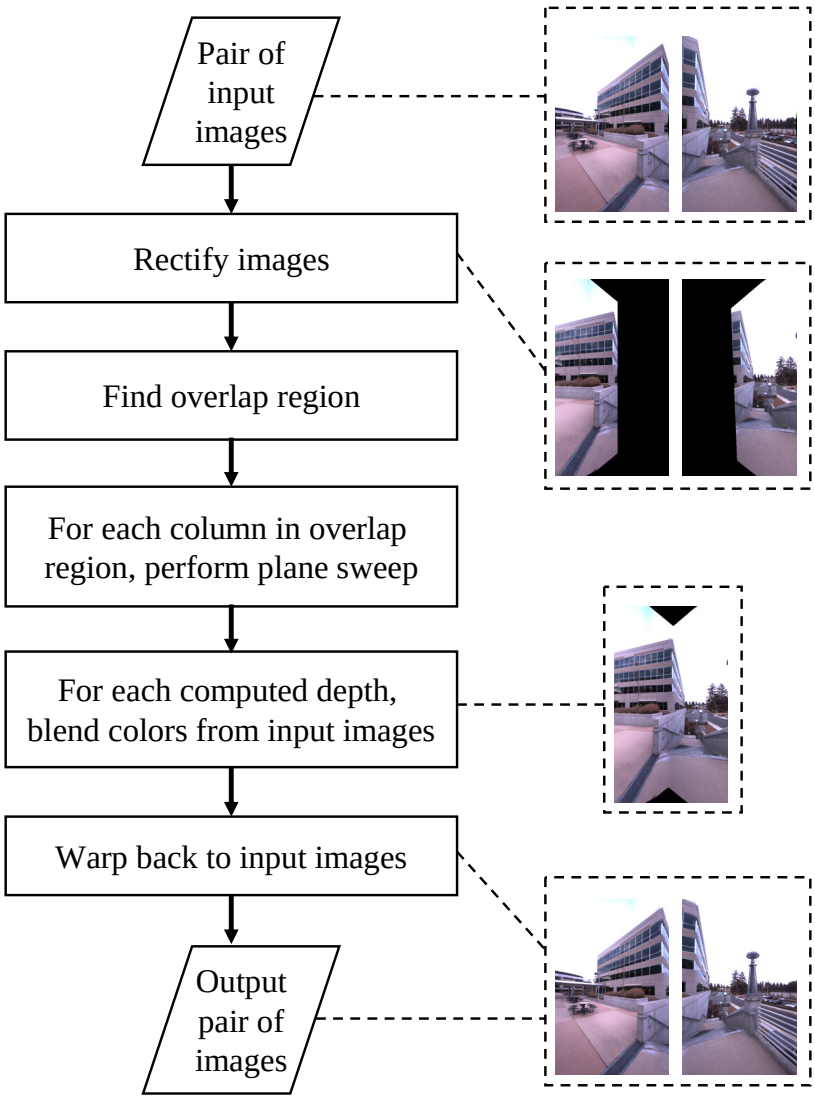
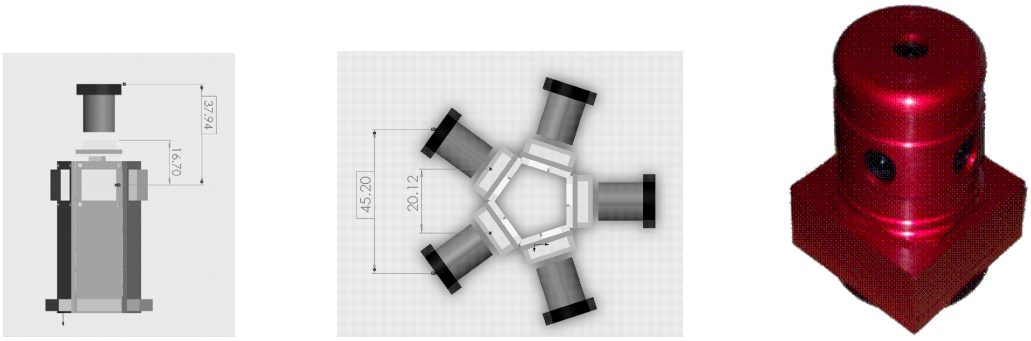


Figure 5: *Block diagram for MPPS.* The images to the right are examples at the respective stages.

the final composite. The blending weights used are simply a linear function based on the proximity to the edge of the overlap region.

Say a pixel is located λ_1 pixels away from the left boundary of the overlap region (i.e., the boundary closer to camera C_1) and λ_2 pixels away from the right boundary (closer to camera C_2). Say also the mapped color from image $I_{1,rect}$ is $\mathbf{c}_1$ and that from image $I_{2,rect}$ is $\mathbf{c}_2$. The blended color of the pixel is then $\left(\frac{\lambda_2}{\lambda_1+\lambda_2}\mathbf{c}_1 + \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbf{c}_2\right)$.

5. *Warp back.* Once the overlap image region has been computed, it is warped back from its rectified state to the original two images, which now align seamlessly in the overlap region



(a) (b) (c)

Figure 6: *The 6-camera omnicam system.* (a) Side view (schematic), (b) top view (schematic), and (c) actual camera system.

Figure 7: *Set of 6 images (corrected for radial distortion) from the omnicam.*

(when stitched onto the plane at infinity). These corrected images can then be used to form seamless texture maps for an environment map [Greene, 1986] or to create a panoramic image.

4 An image mosaicking result using MPPS

For our experiments, we use the six-camera *Ladybug* system (shown in Figure 6) developed by PointGrey, with five of the cameras looking horizontally outwards and one looking up. Each adjacent camera pair is designed to visually overlap by about 10%. The resolution of each image is 1024×768. We use this system to capture omnidirectional video for virtual walkthroughs. To produce a realistic-looking walkthrough, seamless stitching of each frame from the omnidirectional video is critical.

An example set of images acquired using this capture system (after approximate correction for radial distortion) is shown in Figure 7. These images have also been corrected for vignetting. The vignetting parameters are found using calibration images taken inside a diffuse integrating sphere.

Figure 8: *Effect of using MPPS.* The two leftmost images are composites using MPPS, with the rightmost two images the result of using only feathering.

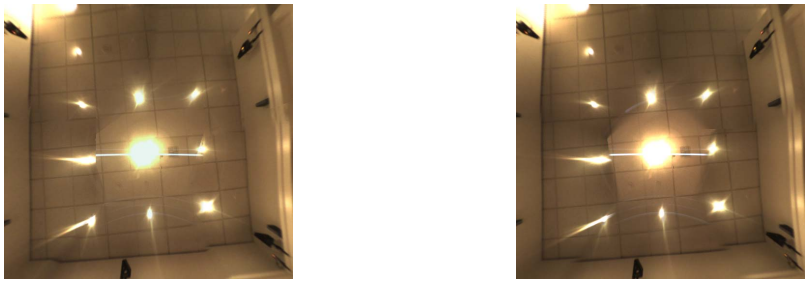


Figure 9: *Effect of using MPPS and color correction (left), without (right).*

The original set of images is shown in Figure 7. Figure 8 shows the difference between using our technique compared to simple feathering. (By feathering, we mean simple blending based on proximity to the overlap edges.) The composite generated using the MPPS technique is generally much sharper, and it looks credible.

The upward-looking (top) image is a special case, since it overlaps with the other five images. We first perform color correction on the top image by finding the best color mapping (on R, G, and B bands separately) between the overlapped areas. In our current version, we compute the appearance of the overlap between each pair (top and each of the other five images) independently. The proper procedure would be to plane sweep with all the cameras simultaneously, and to consider each pixel in overlap areas having a different virtual camera position. However, this would take a long time to process, and would be more complicated to implement. The results of computing the composite for the top image is shown in Figure 9.

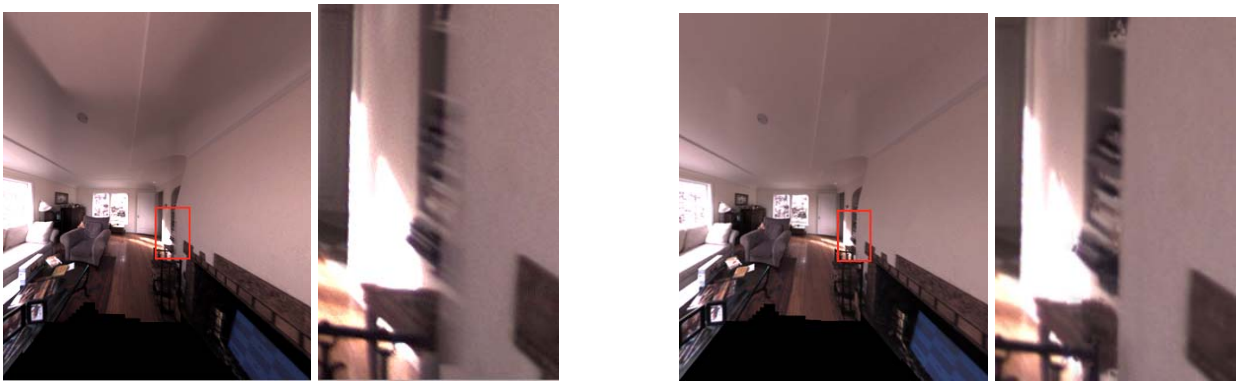


Figure 10: *An image mosaicking example (home scene, 2 images):* Direct feathering results are shown on the left (notice the blurring in the middle–see inset), The result of using MPPS is shown on the right. The squares show the highlighted inset areas that are magnified.

Figure 10 shows the results of using our MPPS technique on a pair of images taken from horizontally adjacent cameras. Note how blurry the edge of the wall is if simple feathering is used. (An overlap has a left edge and a right edge. For a contributing image, one edge is closer to its image center than the other edge. The weight is linearly inversely proportional to the distance to this edge.)

5 The Stretching Stereo Algorithm

To create a virtual walkthrough, thousands of frames may have to be stitched. As a result, timing considerations are very important. We have developed an alternative image mosaicking approach, which we call *Stretching Stereo* (SS). While this approach is much faster, unlike the MPPS, it is not based on a physical geometric model. Rather, it is an image-based solution. The idea is that the closer the overlap pixel is to the non-overlap edge, the smaller its range of motion. In practice, it seems to work almost as well as the MPPS approach.

The concept of SS is shown in Figure 11. In this example, images I_1 and I_2 are the rectified images to be stitched. For purposes of illustration, we concentrate on one row of pixels. On the left edge of the overlap region, the pixel of I_1 is anchored in place (with disparity being 0, i.e., with depth at infinity). The disparity of the pixel of I_1 is increased linearly ("stretched") as a function of its distance from the anchor point, with the maximum being d, the given maximum disparity. The same argument applies for the pixels of I_2. The plane sweep then proceeds in the same manner as MPPS, with $d = d_{min}, ..., d_{max}$. The winning depth distribution is the one associated with the

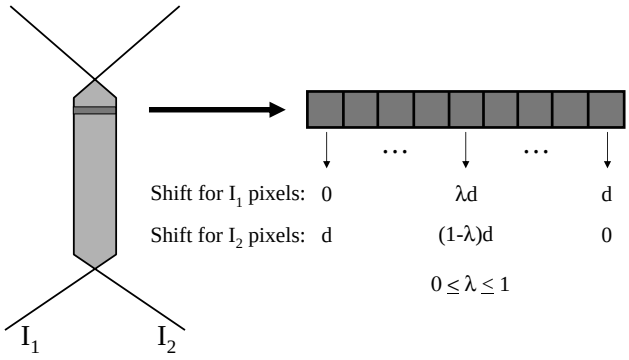


Figure 11: *Concept of Stretching Stereo.* I_1 and I_2 are the two (rectified) images to be stitched. Here we are considering only one row of pixels within the area of overlap as an illustration.

(a) (b) (c)

Figure 12: *Stitching example (outdoor garden, 2 images):* (a) Result of using direct feathering. Notice the blurring in the middle of the image (within the rectangle), (b) result of using MPPS, and (c) result of using SS. The rectangles show the highlighted inset (closeup) areas.

minimum warped image difference. In this case, because the shifts are horizontal (since the images were rectified), they can be computed very quickly.

The stitching time per image pair on a 2 GHz PC is about 6 secs for SS, compared to about 12 secs for MPPS (for 12 disparity search levels). The resolution of each image is 1024×768, and each overlap is about 80 pixels wide.

6 More image mosaicking examples

In this section, we show two examples of image mosaicking using simple feathering, MPPS, and SS. In all cases, the stitched (overlapped) areas are about 80 pixels wide. An image mosaicking example of an outdoor garden scene is shown in Figure 12, while Figure 13 shows a second example of an indoor scene. The linear perspective images in the left column of Figure 12 are the result of

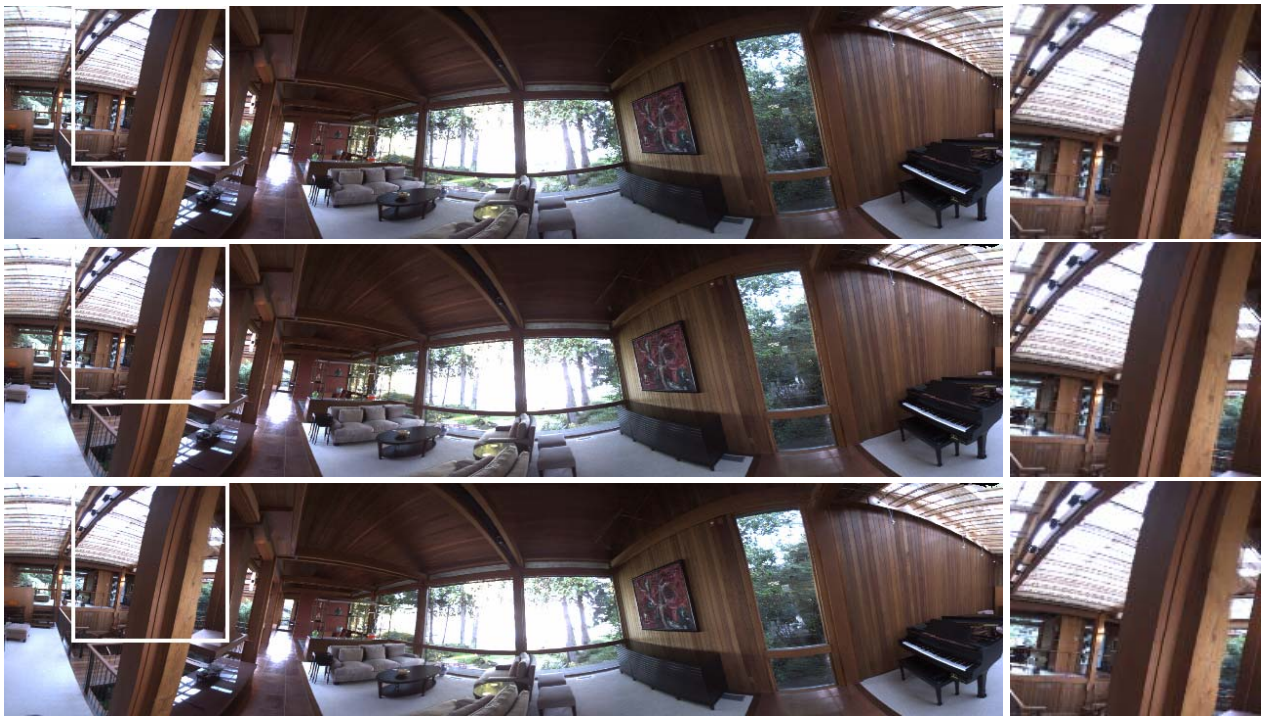


Figure 13: *Stitching example (interior of a house, 5 images):* From top to bottom: results of using direct feathering, MPPS, and SS. The vertical edges appear slanted because the camera setup was slightly tilted. The squares show the highlighted inset areas that are magnified on the right. Notice the blurring in the top left region of the cylindrical panorama for direct feathering (more obvious in inset).

stitching two images, while the cylindrical panoramas on the left of Figure 13 were generated from five images.

For the garden example (Figure 12), the stitched area is in middle. The ghosting artifact is evident for the feathering case. For the indoor example shown in Figure 13, there are 5 stitched areas (5 input images covering 360° horizontally). The ghosting artifact can be clearly seen in one stitched area for the simple feathering case (top left portion of the cylindrical image in Figure 13). The parallax is significant at the support beam due to its proximity to the cameras (about only 2 feet away). The inter-camera distance is about 1 inch. The beam looks slanted because the camera setup was slightly tilted when the images were taken.

In both these examples, the results for SS are a little worse than MPSS. This can be attributed to the graduated disparity sweep in SS. To see this, let us assume there exists an object with significant parallax very close to the edge (with $\lambda \ll 1$ as defined in Figure 11). In order for correct matching to occur between the two images, the edge disparity d (see Figure 11) must be very large, since $(\lambda$

d) must be large. MPPS does not suffer from this problem, because the disparity across the scanline (or epipolar line) is computed using the same disparity range.

7 Discussion

In Stretching Stereo (SS), a set of "disparities" is computed only once. By comparison, the MPPS algorithm uses multiple reference viewpoints (one viewpoint per column). As a result, MPPS uses multiple sets of disparities with respect to the original rectified images. This accounts for SS being significantly faster than MPPS.

MPPS is designed primarily for pairwise image mosaicking. However, it could be adapted to stitch multiple overlapping images if their centers of projection lie approximately on a plane. This assumption is required to enable warping of the input images onto a single, common image plane. Rather than associating each column with a unique virtual camera (which does not make sense for multiple cameras in general), we can instead subdivide the overlap area into small regions, and associate each region with a unique virtual camera. In the limit, each pixel can be associated with a different virtual camera, but this is very computationally expensive for large images. The process of plane sweeping and color assignment in the overlap region can then proceed as with the current MPPS technique. Unfortunately, it is not possible to extend Stretching Stereo to multiple overlapping images in the same manner, since it requires rectified images as inputs.

There is some similarity between MPPS and Concentric Mosaics (CMs) [Shum and He, 1999]. Multiperspective images are also constructed for CMs, and depth is used to generate the synthetic view. However, the depth correction used is manually obtained, while ours is automatic. In addition, the depths per column in CMs are all the same. For MPPS, these depths vary from pixel to pixel.

One disadvantage of these two techniques is that straight lines that appear in adjacent images may become curved in the overlap region. In order for this not to happen, the *non-overlapping* areas would have to be properly warped (by their true parallax) as well, and a single center of projection for the whole mosaic would have to be used. This is not possible if additional information such as depth or 3D shape is not known. Straightening just the straight structures might also produce strange-looking results if the rest of the scene were not corrected at the same time. This problem appears to be endemic for sequences with limited amounts of overlap.

8 Conclusions

In this paper, we have described an effective technique for seamless image mosaicking using a geometric-based approach. We call this technique the Multi-Perspective Plane Sweep (MPPS). Our approach divides the overlap regions between pairs of images into subregions, each of which is assigned a separate virtual camera view. The plane sweep algorithm is then applied to each subregion in turn to produce a photoconsistent (ghost-free) appearance. Our approach is designed for image sequences with small to moderate amounts of overlap between adjacent images, where previous techniques (based on either strip extraction from dense video or full correspondence for large overlaps) are not applicable.

We have also described a faster technique called the Stretching Stereo (SS). While MPPS uses a constant disparity at each step inside the sweep, SS uses a linearly varying disparity distribution. For each image, the disparity distribution is always zero (acting as anchor points) at border pixels within the overlap region.

Results show that the MPPS and SS produces significantly better results than conventional feathering that is currently used in commercially available stitching software.

References

[Boykov *et al.*, 1999] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Seventh International Conference on Computer Vision (ICCV'99)*, pages 377–384, Kerkyra, Greece, September 1999.

[Collins, 1996] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR'96*, pages 358–363, San Francisco, CA, June 1996.

[Greene, 1986] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, November 1986.

[Kang *et al.*, 2001] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR'2001*, pages 103–110, Kauai, HI, Dec. 2001.

[Kumar *et al.*, 1994] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR'94*, pages 685–688, Jerusalem, Israel, October 1994.

[Loop and Zhang, 1999] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *CVPR'99*, pages 125–131, Fort Collins, CO, June 1999.

[Peleg and Herman, 1997] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *CVPR'97*, pages 338–343, San Juan, Puerto Rico, June 1997.

[Rademacher and Bishop, 1998] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *SIGGRAPH'98*, pages 199–206, July 1998.

[Rousso *et al.*, 1998] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha. Universal mosaicing using pipe projection. In *ICCV'98*, pages 945–952, Bombay, India, 1998.

[Seitz and Dyer, 1996] S. M. Seitz and C. M. Dyer. View morphing. In *Computer Graphics Proceedings, Annual Conference Series*, pages 21–30, ACM SIGGRAPH, Proc. SIGGRAPH'96 (New Orleans), August 1996.

[Shum and He, 1999] H.-Y. Shum and L.-W. He. Rendering with concentric mosaics. In *SIGGRAPH'99*, pages 299–306, ACM SIGGRAPH, Los Angeles, August 1999.

[Shum and Szeliski, 2000] H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *IJCV*, 36(2):101–130, Feb. 2000.

[Zhu *et al.*, 2001] Z. Zhu, E.M. Riseman, and A.R. Hanson. Parallel-perspective stereo mosaics. In *ICCV'2001*, pages 345–352, Vancouver, BC, Canada, 2001.