Registration, Calibration and Blending in Creating High Quality Panoramas

Yalin Xiong
yx@robotics.jpl.nasa.gov
Jet Propulsion Laboratory
Pasadena, CA 91109

Ken Turkowski
turk@apple.com
Apple Computer
Cupertino, CA 95014

Abstract

This paper presents a system for creating a full 360-degree panorama from rectilinear images captured from a single nodal position. The solution to the problem is divided into three steps. The first step registers all overlapping images projectively. A combination of a gradient-based optimization method and a correlation-based linear search is found to be robust even in cases of drastic exposure differences and small amount of parallax. The second step takes the projective matrices and their associated hessian matrices as inputs, and calibrates the internal and external parameters of every images through a global optimization. The objective is to minimize the overall image discrepancies in all overlap regions while converting projective matrices into camera parameters such as focal length, aspect ratio, image center, 3D orientation, etc. The third step re-projects all images onto a panorama by a Laplacian-pyramid-based blending. The purpose of blending is to provide a smooth transition between images and eliminate small residues of misalignments resulting from parallax or imperfect pairwise registrations. The blending masks are generated automatically through the grassfire transform. At the end, we briefly explain the necessary human interface for initialization, feedback and manual options.

1 Introduction

A panorama is a compact representation of the environment viewed from one 3D position. While an ordinary image can capture only a small portion of the environment, a panorama can capture it all or any portion of it, depending on the geometry in which the panoramas are represented. Recently there has been an explosive popularity of panoramas on the world wide web and in multimedia as an effective tool to present a photo-realistic virtual reality. However, creating high-quality panoramas, especially those that completely enclose space, remains difficult.

This paper presents a robust general purpose system to author panoramas from rectilinear images. The images can be captured by video camera or film as long as they are captured from approximately the same nodal position. We view the authoring problem as three subproblems: the projective registrations of overlapping images, the self-calibration in which 2D image planes are positioned in 3D space, and the compositing problem in which images are reprojected to a 3D environment map with pixels in overlap regions being somehow composited from multiple images.

The pairwise projective registrations establish, in the 2D image space, the warping between two arbitrarily overlapping images. To be robust against drastic exposure differences and small amount of parallax between images, we perform a normalized correlation search in the coarsest pyramid level to initialize translations and exposure differences, and gradient-based method to register projectively.

The resulting projective matrices and their associated hessian matrices are then used to approximate the error surfaces quadratically, which are then combined together into a global objective function. We obtain the camera internal and external parameters by minimizing the objective function. The advantage of this scheme is that we avoided evaluating the real error surfaces which is prohibitively expensive.

The objective of blending is to provide a smooth transition between images and eliminate artifacts of minor misalignments resulting from parallax or imperfect pairwise registrations. The multi-resolution blending ([2]) based on Laplacian pyramids is an elegant solution for blending. Unfortunately [2] did not specify how to compute the blend masks automatically. We propose an algorithm based on the grassfire transform to compute blend masks for arbitrarily overlapping images. We further differentiate "intended" overlaps from "unintended" overlaps, and develop a labelling scheme to favor larger overlaps in order to deal with overlaps of more than two images. The overall blending algorithm presented in this paper finds the optimal transition regions, and utilizes the Laplacian-pyramid-based blending method in [2] to blend images onto panoramas.

For such a complicated system, human interface is necessary for initialization, feedback and manual options. With proper initialization, we can avoid getting trapped in local minima during the pairwise registrations or global optimization. No matter how robust we build the projective registration, it will break on some material due to excessive exposure differences, parallax, motions in the scene, etc. Thus we need a human interface to let users monitor the progress, and intervene when necessary. Though it is not the emphasis in this paper, the human interface issue will be briefly explained.

Authoring panoramas were first introduced in [6, 9], where images were mosaiced into a single large image by warping in 2D image spaces. The resulting panoramas can be interpreted as texture maps on a 2D manifold embedded in 3D ([7]). Because of the complicated 3D geometry of the manifold, it is difficult to render realistic planar images from the panoramas. The cylindrical panorama with a single nodal position was popularized by QuickTime VR ([3]) because the real-time rendering enables compelling sense of realism. In order to create such panoramas, a fisheye lens ([10]) or other specialized panorama lenses ([5]) were proposed for easy authoring. Another direction in creating panoramas is to combine the 2D image mosaicing and camera calibration of 3D orientations using techniques similar to [4]. The present paper is a significant advance in this direction. The system presented in this paper is the first in that it calibrates all camera internal and external parameters, allows images to be captured from different cameras, and automates the multi-resolution blending in the most general case.

2 Pairwise Registration

If we restrict camera motions to be rotational only, the 2D warping between images is strictly projective in absence of lens distortions, i.e.,

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix}, \qquad (1)$$

where $\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ are the homogeneous coordinates of pixel locations. In the following discussions, we will use vector $\mathbf{X}_i$ to represent the homogeneous coordinates, and matrix $\mathbf{M}_{ij}$ to represent the projective transforms between two image coordinates. Due to the scale ambiguity in the projective matrix, we set the last parameter m_8 in the projective matrices to be 1.

The objective of the pairwise registration is to estimate the projective matrix given two overlapping images. We initialize the projective matrix by the camera internal and external parameters, e.g.,

$$\mathbf{M}_{ij} = \mathbf{T}^{-1}(\mathbf{p}_i, \mathbf{q}_i)\mathbf{T}(\mathbf{p}_j, \mathbf{q}_j), \qquad (2)$$

where

$$\mathbf{T}(\mathbf{p}_i, \mathbf{q}_i) = \mathbf{R}(\mathbf{q}_i) \begin{bmatrix} 1 & 0 & -C_x^i \\ 0 & a_i & -C_y^i \\ 0 & 0 & f_i \end{bmatrix}. \qquad (3)$$

where $[C_x^i, C_y^i]$ are the image center position, a_i is the aspect ratio, f_i is the focal length, $\mathbf{p}_i = [a_i, f_i, C_x^i, C_y^i]^T$ is the internal parameters, $\mathbf{q}_i$ represents the camera orientation with respect to a common reference frame, and $\mathbf{R}()$ is the 3x3 rotation matrix computed from the orientation parameters. How we initialize the camera internal and external parameters will be briefly explained later in Section 5.

There are ten parameters in the projective registrations: eight independent parameters in the projective matrix and two parameters to compensate for brightness and contrast difference between the two images. The gradient-based optimization minimizes the following objective:

$$e_{ij} = \frac{1}{A_{ij}} \sum_{\text{overlap}} \left(s_{ij} I_j(\mathbf{X}_j) + b_{ij} - I_i(\mathbf{M}_{ij}\mathbf{X}_j)\right)^2, \quad (4)$$

where s_{ij} and b_{ij} represent the exposure difference, $I_i()$ and $I_j()$ are pixel intensity values from the two images, and A_{ij} is the overlap area. The optimizations are performed on progressively finer levels of Gaussian pyramids. However, through experimentation we found that the direct application of gradient-based optimization failed frequently due to exposure differences or large translations or both.

We use a combination of correlation-based linear search and a progressive damping (i.e. simulated annealing) of exposure parameters to alleviate the problem. On the coarsest pyramid level, we first perform a linear search over the translational parameters using normalized correlations. The idea is similar to the progressive complexity in [8]. Since the image size on the coarsest pyramid level is small, the correlations are done efficiently. Once the maximal correlations are found, the exposure parameters s_{ij} and b_{ij} are estimated through a linear regression. When the gradient-based optimization is performed on subsequent finer pyramid levels, the damping coefficients on exposure parameters are reduced exponentially, and set to zero at the finest pyramid level.

Given an arbitrary overlap of two images, we determine the number of the pyramid levels by computing eigenvalues of the 2x2 inertial tensor of the overlap polygon region. The square root l of the smaller eigenvalue is used to estimate the number of pyramid levels:

$$\log_2\left(\frac{l}{l_{\min}}\right), \qquad (5)$$

where $l_{\min}$ is the minimal size on the top pyramid level. In our system, $l_{\min}$ is set to 10 pixels.

3 Calibration and Global Optimization

The second step of authoring panoramas is to extract camera internal and external parameters from those projective matrices. In general, it is impossible to invert Eq. 2 directly to obtain the camera parameters since there are eleven camera parameters while a projective matrix provides only eight constraints. But one image usually overlaps with multiple images. Thus we can take advantage of the redundancy to obtain a consistent set of camera parameters such that they approximate all projective matrices in the same time. A global optimization is used to achieve the goal.

Since the projective matrix is a function of camera parameters as in Figure 2, we can minimize the following objective functions to extract all camera internal and external parameters,

$$E = \sum_{ij} A_{ij} e_{ij} \left(M_{ij} \left(\mathbf{p}_i, \mathbf{q}_i, \mathbf{p}_j, \mathbf{q}_j \right) \right) \qquad (6)$$

where e_{ij} is the pairwise objective function in Eq. 4.

Unfortunately, it is prohibitively expensive to evaluate the above objective functions. But we already optimized the pairwise objective function e_{ij} individually. Thus we can approximate it by a quadratic surface:

$$e_{ij}(\mathbf{M}_{ij}) \approx e_{ij}^0 + (\mathbf{M}_{ij} - \mathbf{M}_{ij}^0)^T \mathbf{C}_{ij} (\mathbf{M}_{ij} - \mathbf{M}_{ij}^0), \quad (7)$$

where e_{ij}^0 is a constant representing the minimal value achieved in the pairwise registration, $\mathbf{M}_{ij}^0$ is the 8x1 vector representing the optimal projective matrix, and $\mathbf{C}_{ij}$ is the 8x8 hessian matrix obtained when optimizing objective function e_{ij}. Note that we now represent a projective matrix as an 8x1 vector instead of a 3x3 matrix.

Once the pairwise objective functions are approximated by quadratic surfaces, the global objective function in Eq. 6 is simply a weighted sum of all those quadratic surfaces. Its gradient with respect to the camera internal and external parameters can be easily established through the chain rule,

$$\frac{\partial E}{\partial \mathbf{p}_i, \mathbf{q}_i} = \sum_j \frac{\partial e_{ij}}{\partial \mathbf{M}_{ij}} \frac{\partial \mathbf{M}_{ij}}{\partial \mathbf{p}_i, \mathbf{q}_i}, \qquad (8)$$

from Eq. 2 and Eq. 7. Since no direct evaluation on images is involved, the computation required in minimizing the global objective function is trivial.

The camera parameters for each image are four internal parameters $\mathbf{p}_i$ and three orientation parameters in $\mathbf{q}_i$. Therefore there are seven independent parameters for each image in the most general case. Every pairwise registration provides eight constraints on those camera parameters. In general, when there are plenty of overlapping image pairs, the optimization is overconstrained in that the number of independent parameters is usually less than that of constraints. But in practice, even when it appears to be overconstrained, many camera parameters are so weakly constrained that they can easily diverge the whole optimization.

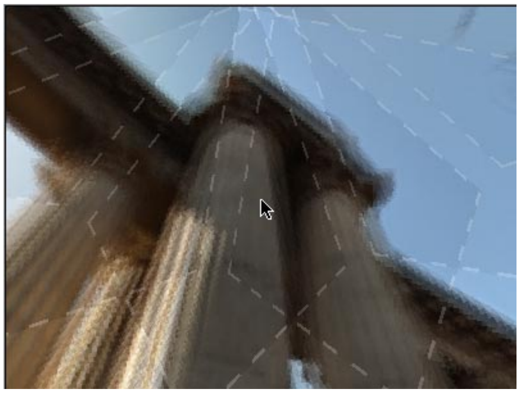

Figure 1: Images Alignment before Calibration

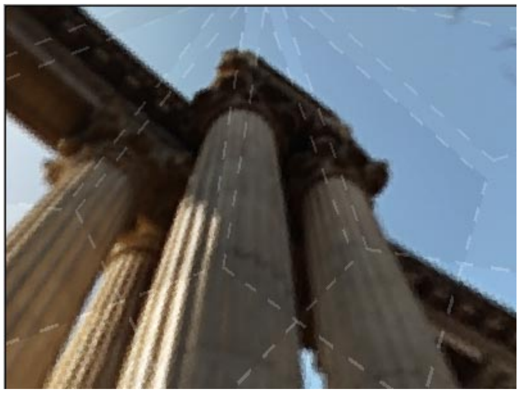

Figure 2: Images Alignment after Calibration

In order for the optimization to behave well in underconstrained or weakly constrained situations, we use simulated annealing to dampen the camera internal parameters. As the optimization progresses, we gradually reduce the damping parameters. Since we have good initial estimates of those parameters, the scheme works remarkably well in practice.

Figure 1 shows part of the panorama before the calibration through a virtual camera. The dotted lines are image boundaries, and overlapping pixels are averaged. There are obvious misalignments as shown as blurry edges due to averaging. Figure 2 shows the same view after the calibration.

The pairwise registration and the global optimization can be iterated if the alignments are still not satisfactory. The pairwise registration will use the improved

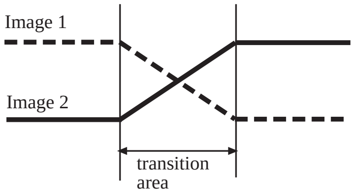

Figure 3: Blending by Weighted Average

camera parameters to initialize the projective registrations, and re-compute the optimal projective matrices and their hessians. The the improved projective matrices will, in turn, generate improved estimations of camera parameters in the global optimization.

4 Blending

In practice, the pairwise registration and the subsequent camera calibration is bound to be imperfect for various reasons. When panoramas are generated from those imperfectly aligned images, the panoramas will have "shadow" or "ghosting"([9]) effects if the images are averaged in overlap regions. Human eyes are very sensitive to those. In high quality panoramas, these shadow effects must be eliminated and drastic exposure difference between adjacent images must be smoothed. The multi-resolution blending algorithm illustrated in [2] is an elegant solution to solve these problems.

The weighted average method used in [9] is illustrated in Figure 3. In the transition region, the weights of Image 1 decrease from 1.0 to 0.0 while the weights of Image 2 increase from 0.0 to 1.0. A pixel in the transition area is a weighted sum of two pixels from two images. The multi-resolution algorithm first decomposes two images into different frequency bands by building Laplacian pyramids, and performs separate weighted averages on each pyramid level with *different* transition lengths. Figure 4 shows the transition lengths for different frequency bands. The result of these multi-resolution blending is seamless and absent of shadow effects as documented in [2] compared with the simple weighted average.

In general, however, the image overlaps are irregular in shape. If we perform the blending on the cylindrical panorama, the overlap regions are not polygons with straight edges. For an arbitrarily shaped transition region, a blend mask ([2]) is needed for the multi-resolution blending. The Gaussian pyramid of the mask image supplies the weights for every pixel at every pyramid level. Figure 5 shows an example of the mask for two overlapping images.

In order to maximize the size of the transition region, the boundary curves of the mask inside the overlap regions need to be as far as possible away from the original image boundaries. To locate the mask boundary, we perform the grassfire transform ([1]) on two images individually. The resulting distance maps represent how far away each pixel is from its nearest boundary. The pixel values of the blend mask is then set to either 0 or 1 by comparing the distance values at each pixel in the two distance maps.

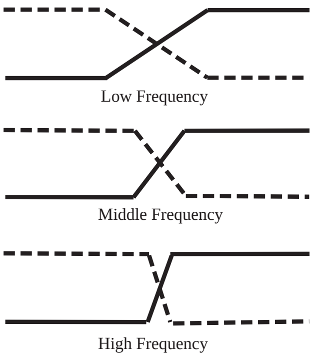

Figure 4: Transitions in Multi-Resolution Blending

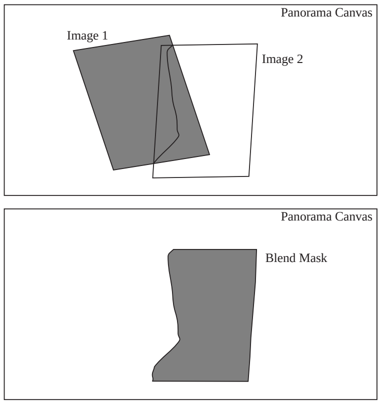

Figure 5: Blend Mask for Two Overlapping Images

We start with an empty panoramic canvas. At first we copy the first image onto it since there is no content to blend it with. Then we blend new images onto the panoramic canvas one by one. For each of those new images, we generate the blend mask from the panoramic canvas and the new image, build Laplacian and Gaussian pyramids in bounding rectangle areas of the overlap regions, blend them in pyramids, and finally copy the blended image onto the panoramic canvas.

When there are overlaps of more than two images, we may run into the problem as illustrated in Figure 6. The first (N-1) images are sequentially blended onto the panorama canvas, but when we try to blend Image N, most of the area covered by Image N is already blended by an "unintended" overlap between Image 1 and Image N-1. As a result, Image N has little effects on the panorama even though it provides much larger transition areas between Image 1 and Image N-1, and

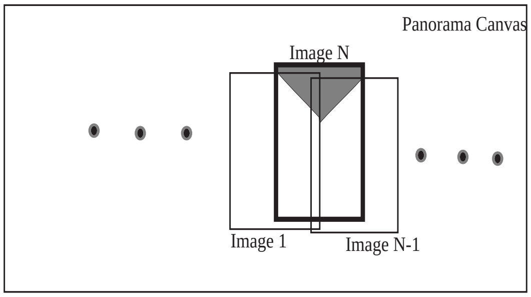

Figure 6: Unintended Overlap

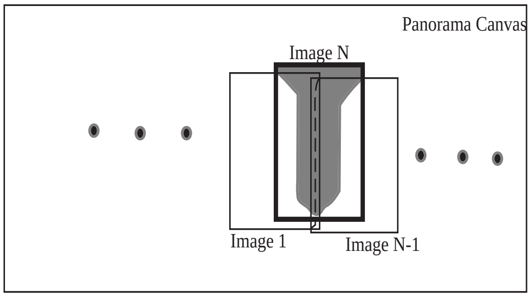

Figure 7: Blend Mask Using the Labelling Scheme

therefore, has potential of improving the quality of the panorama. The blend mask is indicated as the gray area.

We can reduce the chance of this scenario by changing the order in which images are blended onto the panoramic canvas. Images with larger overlaps should be blended onto the panoramic canvas first. But the scenario described above is unavoidable in general because of the wraparound nature of the panorama.

We develop a labelling scheme to overcome the problem of unintended overlaps. For every pixel on the panoramic canvas, we label it with a number identifying which source image contributes most. In example of Figure 5, all pixels with value 1 in the blend mask are labelled 2 since Image 2 contributes most to the pixel values in that area. In Figure 7, the dashed line is the blend mask boundary when Image 1 and Image N-1 are blended. The pixels on the left of the dashed line have label 1, while the pixels on the right side of the dashed line have label N-1. When Image N need to be blended onto the panoramic canvas, we first perform the grassfire transform on the panoramic canvas. In addition to the image boundaries, we regard the dashed line as a virtual boundary as well. We call it a “firewall” in that the grassfire cannot penetrate the invisible wall. Those virtual boundaries can be computed easily using the pixel labels and the list of all intended overlaps. The resulting blend mask is illustrated as the gray area in Figure 7. Now the blending takes advantage of both large overlaps between Image 1 and Image N, and Image N and Image N-1.

Figure 8 shows a cylindrical panorama blended from 24 images. Those images were shot in two rows using a 24mm lens. The first row is horizontal and the second row is tilted upward at about 30 degrees. The resulting panorama has a vertical field of view of more than 100 degrees. Figure 9 shows a cubic panorama (six faces) blended from 24 images shot in two rows using a 15mm lens. The tilt angle between rows is about 40 degrees. In both these two examples, despite imperfections including strong parallax and motion in the scene, the panoramas are seamless.

5 Human Interactions

Human interface is an integral part of the system. This system is trying to solve a complicated nonlinear optimization problem. No algorithm can guarantee its convergence to the global minimum. The main purposes of the human interface (other than making the system easy to use) are:

- Manual Projective Registration: Every projective registration algorithm can break down in cases such as excessive exposure difference, motions in the scene, bad initial estimate, etc. The last choice when the automatic registration fails is to use a manual registration. The human interface provides this last resort.
- Initial Calibration: The number of camera internal and external parameters is large in the general case. The global optimization needs a good initialization in order to converge to the right answer. The human interface must provide an interactive tool to initialize those parameters.
- Feedback: The system must have the ability to provide feedback in all the nonlinear optimizations to let users monitor the progress and allow them to intervene when necessary.

The core part of the human interaction is a real-time texture map engine which simulates a virtual camera looking out from the nodal point. All images are floating in a 3D space. Figures 1 and 2 show the typical user interaction window.

The camera parameters are initialized interactively. The user can directly control the camera field of view, camera 3D orientation and image center position by click and drag. The texture map engine will provide real time response to the changes. We also implemented standard GUI options such as selecting and deselecting individual image or a group of images. More details of the interface will not be addressed in this paper due to space limitation.

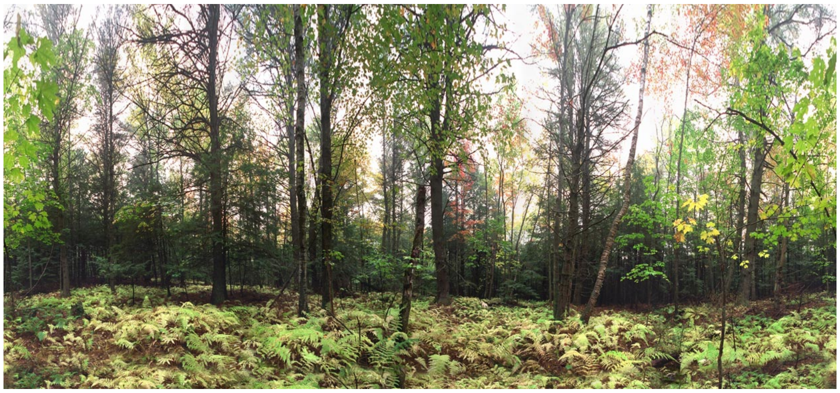

Figure 8: Cylinder Panorama Blended from 24 Images

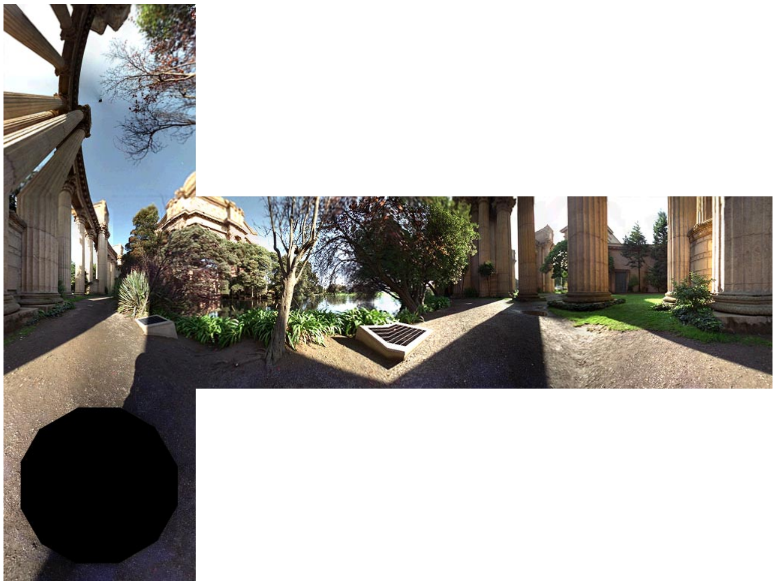

Figure 9: Cubic Panorama Blended from 24 Images

Acknowledgments

The authors would like to thank Mark Wheeler, Steve Dawson, Ed Harp and Eric Blandpied in Apple Computer for various help.

References

[1] C. Arcelli, L.P. Cordella, and S. Levialdi. A grassfire transformation for binary digital pictures. In *Proc of ICPR*, pages 152–154, 1974.

[2] P. Burt and E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[3] Shenchang E. Chen. QuickTime VR — an image-based approach to virtual environment navigation. In *Proc. SIGGRAPH Conference*, pages 29–38, August 1995.

[4] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. In *Proc. European Conference on Computer Vision*, pages 471–478, 1994.

[5] Shree Nayar. Catadioptric omnidirectional camera. In *Proc. of CVPR*, pages 482–488, 1997.

[6] S. Peleg. Elimination of seams from photomosaicking. *Computer Graphics and Image Processing*, C-26:1175–1180, 1981.

[7] S. Peleg. Panoramic mosaics by manifold projection. In *Proc. of CVPR*, pages 338–343, 1997.

[8] H. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distoriton correction. In *Proc. of CVPR*, pages 450–456, 1997.

[9] R. Szeliski. Image mosaicing for tele-reality applications. In *Proc. of Workshop on Applications of Computer Vision*, pages 450–456, 1994.

[10] Y. Xiong and K. Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *Proc. of CVPR*, pages 237–243, 1997.