Creating Image-Based VR Using a Self-Calibrating Fisheye Lens

Yalin Xiong Ken Turkowski
QuickTime VR Group
Apple Computer
Cupertino, CA 95014

Abstract

Image-based virtual reality is emerging as a major alternative to the more traditional 3D-based VR. The main advantages of the image-based VR are its photo-quality realism and 3D illusion without any 3D information. Unfortunately, creating content for image-based VR is usually a very tedious process. This paper proposes to use a non-perspective fisheye lens to capture the spherical panorama with very few images. Unlike most of camera calibration in computer vision, self-calibration of the fisheye lens poses new questions regarding the parameterization of the distortion and wrap-around effects. Because of its unique projection model and large field of view (near 180 degrees), most of the ambiguity problems in self-calibrating a traditional lens can be solved trivially. We demonstrate that with four fisheye lens images, we can seamlessly register them to create the spherical panorama, while self-calibrating its distortion and field of view.

1 Introduction

Image-based virtual reality is emerging as a major alternative to the more traditional 3D-based VR. Unlike virtual environments generated by 3D graphics, in which the information to represent the environment is kept internally as geometry and texture maps, image-based VR represents the environment by one or more images, which can be either captured by camera or synthesized from 3D computer graphics. There are two types of image-based VR representations: the single-node 2D representation [2], which represents the virtual world around one nodal point by a panorama, and the light-field 4D representation [5], which represents the virtual world contained in a pre-defined 3D volume. The main advantages of image-based VR are its simplicity for rendering, photographic quality realism, and the 3D illusion experienced by users.

This paper is concerned with creating content for single-node 2D panoramas. The conventional way to create a surrounding panorama is by rotating a camera around its nodal point. Using a 15mm lens with 35mm film, it takes about 12 pictures to capture a panorama with 90-degree vertical field of view. Capturing a full spherical panorama requires at least 30 pictures and involves rotating the camera along two different axes. In addition, the image registration process becomes complicated. Fortunately, some commercially available fisheye lenses enable us to capture spherical panoramas using far less number of pictures because of their near 180-degree field of view.

Surprisingly, there is little literature on the calibration of fisheye lenses. Most of the published and patented works on using fisheye lens assume either an ideal projection model [1, 8] or use the distortion model of rectilinear lenses by adding more nonlinear terms [7]. We found in experiments that none of the above two schemes is accurate enough for the purpose of registering multiple fisheye images into panoramas. Furthermore, we also need to minimize the requirements for elaborate calibration equipment so that it is easy to use. Therefore, self calibration of the fisheye lens is also desirable.

The fundamental difference between a fisheye lens and an ordinary rectilinear lens is that the projection from a 3D ray to a 2D image position in the fisheye lens is intrinsically non-perspective. There are many projection models for fisheye lenses proposed in literature [6]. We found that the equi-distance model is a reasonable first-order approximation. On top of the equi-distance model, we model the additional radial lens distortion by a third order polynomial. Experimental results demonstrate that fisheye images can be registered seamlessly when the distortions are corrected.

By establishing the correspondence between two or more images, it is shown in [3] that many camera parameters can be recovered without *a priori* knowledge of the camera motion or scene geometry. Unfortunately, self calibration in general is unstable if the image center and the field of view are unknown. The self-calibration of a fisheye lens is even more difficult because of its unknown lens distortion. But for a fisheye lens, its image center can be determined trivially

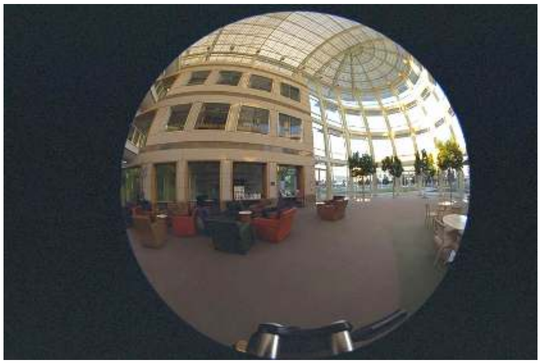


Figure 1: An Image from a Fisheye Lens

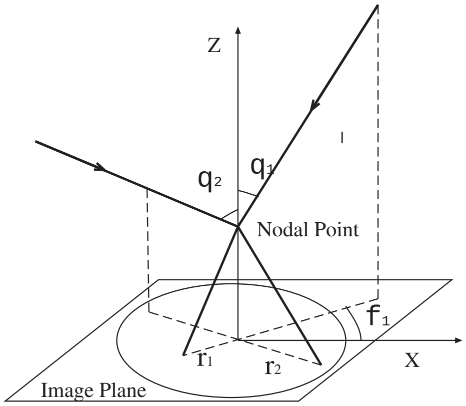


Figure 2: Equi-Distance Projection Model

as the center of the ellipse which envelopes the image (Figure 1). When we rotate the camera around its nodal point to capture the spherical panorama, the wrap-around effect, i.e., the overlap between the first and last images, provides enough constraints for its field of view. Once we know those intrinsic parameters, the self calibration becomes very stable. Hartley in [4] proposed a similar self-calibration approach for a rectilinear lens by rotating the camera, though it is difficult to assess his results for image registration purpose.

Another major difference between the work presented in this paper and other published works on self-calibration is that we register images *while* self-calibrating the camera. The benefit is that the quality of the calibration is iteratively improved because of the improved image registration, and the quality of the image registration is iteratively improved because of the improved calibration. We adopt a multi-level gradient based registration to register the fisheye images while self-calibrating its distortion parameters and field of view. Using the Levenberg-Marquardt minimization, we show that the registration process with the radial distortion modelled as a cubic polynomial results in excellent spherical panoramas.

2 Fisheye Projection Model and Distortion

The projection from 3D rays to 2D image positions in a fisheye lens can be approximated by the so-called "equi-distance" model. Suppose a 3D ray from the nodal point of the lens is specified by two angles θ and ϕ as in Figure 2. Then the equi-distance projection model projects the 3D ray into an image position (x, y), in which

$$x = c\theta\cos\phi, \tag{1}$$

$$y = c\theta\sin\phi, \tag{2}$$

where c is a scale factor determined by the focal length and the scale of the film scanning. In other words, the equi-distance model maps the latitude angle θ to the polar distance r in the image, i.e., $r = \sqrt{x^2 + y^2} = c\theta$, as well as the longitude angle to the polar direction in the image.

The advantage of this projection model over the traditional planar projection model is that it allows an arbitrarily large field of view, at least mathematically. Current commercial fisheye lenses include Nikon 8mm (180-degree FOV) and 6mm (220-degree FOV). We tested the equi-distance projection model in the 8mm fisheye lens, and found that it is a good first-order approximation as we will show later.

The radial distortion model models the higher order effects in the mapping between the latitude angle θ and the polar distance r:

$$r = c_1\theta + c_2\theta^2 + c_3\theta^3 + \cdots, \tag{3}$$

where the order of the polynomial can be determined experimentally.

3 Image Registration and Self Calibration

3.1 Camera Setup

Figure 3 shows the setup for capturing spherical panoramas and self-calibrating. The Nikon N900 camera is mounted on a platform, which can slide in two orthogonal directions. The pointing direction of the camera is slightly tilted upward for reasons we will explain later.

The nodal point of the fisheye lens needs to be adjusted so that it lies on the rotation axis of the tripod. Once the camera is set up properly, we can take either four pictures by rotating the camera 90 degrees after

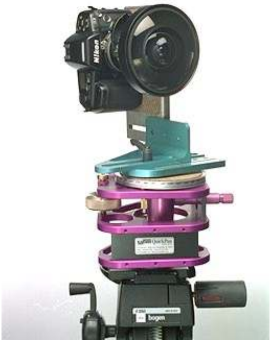


Figure 3: Camera Setup

every shot, or three pictures by rotating it 120 degrees. We prefer the four-picture method simply because it provides larger overlap regions.

3.2 Objective Function and Minimization

Given the four images I_0, I_1, I_2, and I_3, we formulate the registration and self-calibration problems as a single nonlinear minimization problem. The 3D reference frame is the camera coordinate of image I_0. The following 34 parameters are adjusted in the minimization process:

- Camera rotations: We fully parameterize the relative orientations of the camera coordinates of I_1, I_2, and I_3 with respect to the reference frame I_0 in order to accommodate arbitrary, unconstrained rotations. Three angles (roll, pitch, yaw) for each image yield nine rotation parameters $\mathbf{q}_i$, $(i = 1, 2, 3)$.
- Image Centers and Radii: As shown in Figure 1, the envelope of the image is an ellipse with two slightly different principal radii. The parameters are image center positions $\mathbf{o}_i$ and radii $\mathbf{R}_i$ $(i = 0, 1, 2, 3)$. The total number of parameters is sixteen.
- Radial Lens Distortion: We use one cubic polynomial to represent the mapping between the latitude angle and the polar distance for all images. The parameters are c_1, c_2, and c_3. The reason to choose a cubic polynomial is purely experimental, and specific to the Nikon 8mm fisheye lens we have. For other fisheye lenses, the order of the polynomial may need to be higher or lower.
- Image Brightness Difference: The brightness scaling factor s (contrast) and offset a (brightness). The six illumination parameters are s_i and a_i, $(i = 1, 2, 3)$.

Let us first consider the registration of two fisheye images I_i and I_j. The objective function is:

$$\begin{aligned} S_{ij} &= \frac{1}{A_{ij}} \sum_{x_k \in A_{ij}} e_k^2, & (4) \\ e_k &= (s_i I_i(\mathbf{x}_k) + a_i) - (s_j I_j(T(\mathbf{x}_k; \mathbf{p})) + a_j). \end{aligned}$$

where A_{ij} is the overlap region, $T(\cdot)$ is a function which transforms the image position $\mathbf{x}_k$ in I_i to its corresponding position in I_j, and $\mathbf{p}$ is the vector of all parameters listed above except the brightness compensation parameters. The overlap region A_{ij} is determined by the current estimate of the camera parameters and rotations.

The transformation function can be decomposed into three concatenated functions:

$$T(\mathbf{x}_k) = T_3\left(T_2\left(T_1(\mathbf{x}_k)\right)\right). \tag{5}$$

The first function $T_1(\mathbf{x}_k)$ transforms the image position $\mathbf{x}_k$ into a 3D ray direction (θ, ϕ). In the following discussion , we will drop the subscript k to simplify the notation. Let

$$\begin{aligned} \mathbf{x} &= \left[\begin{array}{cc} x & y \end{array}\right]^T, & (6) \\ \mathbf{o}_i &= \left[\begin{array}{cc} o_x^i & o_y^i \end{array}\right]^T, & (7) \\ \mathbf{R}_i &= \left[\begin{array}{cc} R_x^i & R_y^i \end{array}\right]^T, & (8) \end{aligned}$$

we can represent the image position $\mathbf{x}$ in the polar coordinate of image I_i as

$$\begin{aligned} r_i &= \sqrt{\left(\frac{x - o_x^i}{R_x^i}\right)^2 + \left(\frac{y - o_y^i}{R_y^i}\right)^2} & (9) \\ \phi_i &= \text{atan2}\left(\frac{y - o_y^i}{R_y^i}, \frac{x - o_x^i}{R_x^i}\right), & (10) \end{aligned}$$

where atan2 is the arc tangent function with quadrant information. Therefore, the 3D ray direction[1] of $\mathbf{x}$ represented in the camera coordinate of I_i is:

$$\begin{aligned} \theta_i &= \Theta(r_i; c_1, c_2, c_3), & (11) \\ \phi_i &= \phi_i, & (12) \end{aligned}$$

where $\Theta(\cdot)$ is the inverse function of the distortion polynomial in Eq. 3. In practice, the inverse can be solved using the Newton-Raphson root-finding method.

The second function $T_2(\cdot)$ converts the 3D ray direction into the camera coordinate of I_j. Let $\mathbf{M}_i$ and

[1] Note that we use the same notation for the 2D polar direction and the 3D longitude angle because they are the same as long as the tangential distortion is zero, which is assumed in this paper.

$\mathbf{M}_j$ be 3×3 rotation matrices computed from the roll/pitch/yaw angles $\mathbf{q}_i$ and $\mathbf{q}_j$, we then have

$$\left[\begin{array}{ccc} u_x^j & u_y^j & u_z^j \end{array} \right]^T = \mathbf{M}_j \mathbf{M}_i^{-1} \left[\begin{array}{ccc} u_x^i & u_y^i & u_z^i \end{array} \right]^T, \quad (13)$$

in which

$$\left[\begin{array}{ccc} u_x^i & u_y^i & u_z^i \end{array} \right]^T = \left[\begin{array}{c} \sin\theta_i \cos\phi_i \\ \sin\theta_i \sin\phi_i \\ \cos\theta_i \end{array} \right]. \quad (14)$$

Therefore, the 3D ray direction in the camera coordinate of I_j can be represented as

$$\begin{aligned} \theta_j &= \mathrm{acos}(u_z^j), & (15) \\ \phi_j &= \mathrm{atan2}(u_y^j, u_x^j). & (16) \end{aligned}$$

The third function $T_3(\cdot)$ maps the 3D ray (θ_j, ϕ_j) onto the image position in $I_j(x', y')$. The image position in polar coordinate is

$$\begin{aligned} r_j &= c_1\theta_j + c_2\theta_j^2 + c_3\theta_j^3, & (17) \\ \phi_j &= \phi_j. & (18) \end{aligned}$$

In Cartesian image coordinate, the position is

$$\begin{aligned} x' &= o_x^j + R_x^j r_j \cos\phi_j, & (19) \\ y' &= o_y^j + R_y^j r_j \sin\phi_j. & (20) \end{aligned}$$

The minimum of the objective function S_{ij} in Eq. 4 is reached when its derivative is zero. When four images are considered together, the overall objective function is the sum of the all image pairs with overlap:

$$S = \sum_{\forall\{i,j\}:A_{ij}\neq\emptyset} S_{ij}. \quad (21)$$

The Levenberg-Marquardt method is then used to minimize the objective function with proper initial estimates of parameters.

3.3 Initial Estimates and Damping

The initial estimate problem is important for any nonlinear optimization in order to avoid local minima and divergence. Among the parameters we need to optimize, we can set the initial radial distortion model to the ideal equi-distance projection ($c_1 = 2.0/\pi, c_2 = c_3 = 0.0$), and brightness difference parameters to either $s = 1.0$ and $a = 0.0$ or values computed from camera exposure/aperture settings. We need to be especially careful about the rotation angles, image centers and radii because they are the main sources of the nonlinearity in the objective function. The optimization can rarely recover from grossly erroneous rotation angles, image centers or radii.

Between two arbitrary fisheye images taken by rotating the camera around its nodal point, we need to have an initial estimate of the rotation represented by either the roll/pitch/yaw angles $\mathbf{q}$ or a rotation matrix $\mathbf{M}$. If we have, for example, three points in two images matched manually, we can minimize the following function to get an initial estimate of the rotation matrix:

$$E = \sum_{i=0}^{2} (\mathbf{u}_i' - \mathbf{M}\mathbf{u}_i)^2 + \lambda_0 C(\mathbf{M}), \quad (22)$$

where $\mathbf{u}_i'$ and $\mathbf{u}_i$ are the two 3D rays computed as in Eq. 14 from the image positions using the current camera parameters, and the term $C(\mathbf{M})$ constrains the matrix $\mathbf{M}$ to be a rotation matrix.

It is well known that the self calibration is difficult when the image center position is unknown. Fortunately we have an independent way to compute a good initial estimate of the image center due to the unique projection model in the fisheye lens. According to the equi-distance and the radial distortion model, the image center position coincides with the center position of the ellipse. The initial estimates of radii and image centers are obtained by fitting the ellipse. In order for the nonlinear optimization to be stable and more likely to converge to the global minimum, we find from experiments that we need to dampen the image center position and radii.

4 Experiments

4.1 Minimization Feedback

There is no foolproof way to guarantee that the Levenberg-Marquardt method or any other non-linear minimization method converges to the global minimum. Therefore, we need to provide the users with necessary feedback in order for them to tune parameters. In addition, providing feedback while performing the nonlinear minimization will increase the user-friendliness as well.

In the nonlinear minimization process, after every iteration, we show users the current status of the registration. The issue is how to display the current spherical panorama to users in an efficient and intuitive way. In the experiments shown below, we use the ideal equi-distance projection to project the whole spherical panorama into an image (Figure 4) as if it were imaged by an ideal fisheye lens with FOV of 360 degrees. The north and south poles are indicated as in the figure, and the outmost circle of the image corresponds to a single ray $\theta = \pi$. This 360-degree spherical mapping can be physically approximated by the reflection on a shiny ball such as a Christmas ornament when viewed from far away.

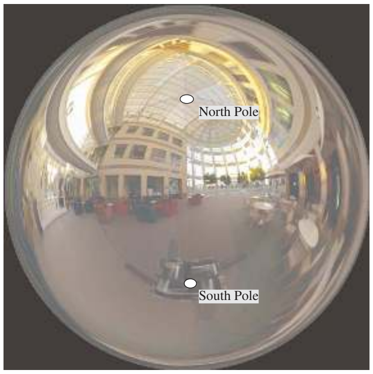


Figure 4: Feedback from the minimization

4.2 Experimental Results

We tested our algorithm using four fisheye images (Figure 5) taken by rotating the camera roughly 90 degrees for every shot. We used Kodak ASA 400 film, and the images were scanned at resolution of 768×512 with 24-bit color. In the bottom portion of each image the tripod is visible. Since the field of view of the fisheye lens is near 180 degrees, and its nodal point has to be on the rotation axis, there appears to be no easy way to get around the problem. In our minimization, we do not take the bottom portion of the fisheye images into account. The reason we usually tilt the camera upward is that since the bottom portion contains the tripod anyway, we are better off tilting it upward so that the top portion (near north pole) is covered redundantly.

In the minimization, the inital rotation angles are 90-degree apart, and the initial rotation axis is pointing north. The image registration is gradient-based. We currently use the derivative of Gaussian as the gradient filter and the Gaussian as the smoothing filter. The size of the smoothing and gradient filters are adjustable to achieve registration at different scales. Figure 6 shows the feedback information during the minimization. The lens distortion model is the cubic polynomial as in Eq. 3. The seams in the feedback images are intentionally left so that the users know where each fisheye image is mapped. Those seams will not be visible in the final stitched panoramas. We can see that the optimization converges quickly to the global minimum. Figure 7 shows the final results of minimizations when the ideal equi-distance projection model and our cubic distortion model are used. We also tested the same optimization on three other sets of fisheye images taken indoor and outdoor using the same fisheye lens. In all cases, we were able to converge to the global minimum in our first try.

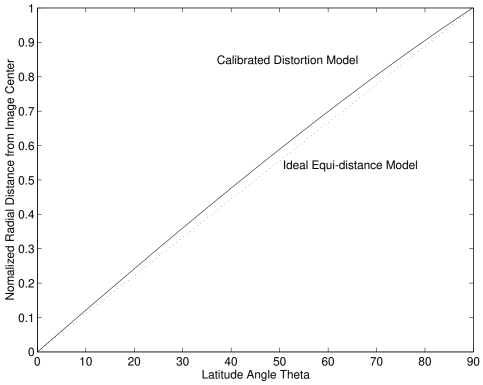


Figure 8: Calibrated Lens Distortion

The result of the self calibration of the fisheye lens is the cubic polynomial of the projection model specified in Eq. 3. Figure 8 shows the calibrated projection model and the ideal equi-distance model.

Once the four fisheye images are registered and the fisheye lens is calibrated, we can represent the spherical panorama using any projection. The equi-distance projection we used in the minimization feedback is one choice. We can also, for example, project the spherical panorama onto a cube. Figure 9 shows the texture maps as projected on the six faces of the cube.

References

[1] Zuoliang Cao, Sung J. Oh, and Ernest Hall. Omnidirectional dynamic vision positioning for a mobile robot. *Optical Engineering*, 25(12):1278–1283, December 1986.

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

[3] O.D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *Proc. European Conference on Computer Vision*, pages 321–334, 1992.

[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] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. SIGGRAPH Conference*, pages 31–42, August 1996.

[6] Kenro Miyamoto. Fish eye lens. *Journal of Optical Society of America*, 54:1060–1061, 1964.

[7] S. Shah and J. K. Aggarwal. A simple calibration procedure for fisheye (high distortion) lens camera. In *Proc. Int'l Conference on Robotics and Automation*, pages 3422–3427, 1994.

[8] Steven D. Zimmermann. Omniview motionless camera orientation system. *U.S. Patent No. 5185667*, 1993.

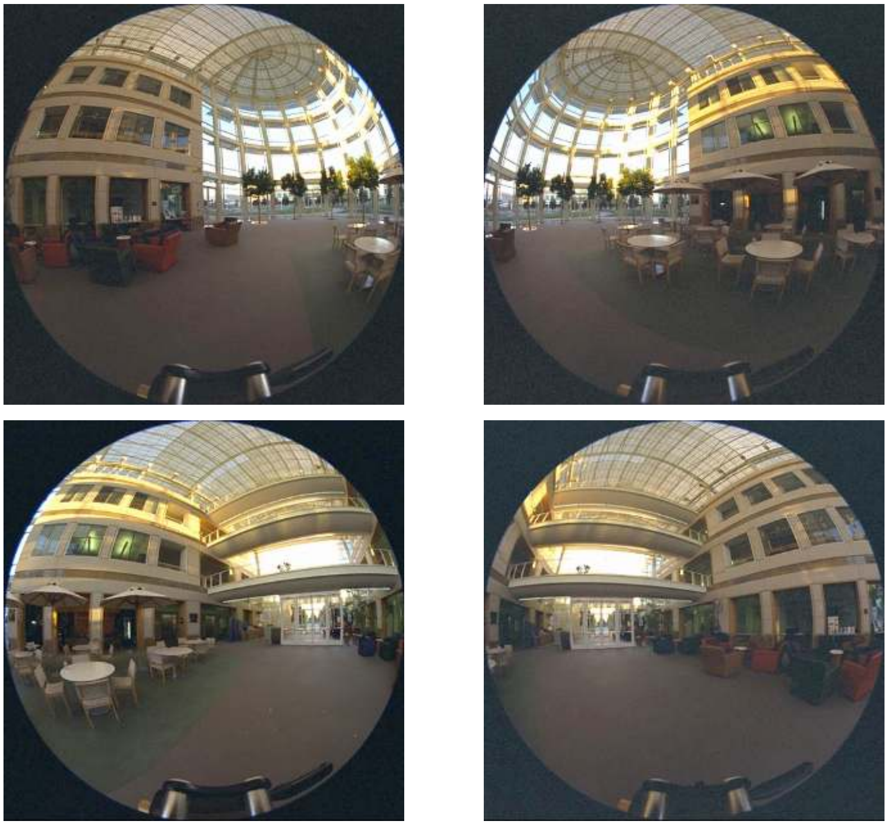
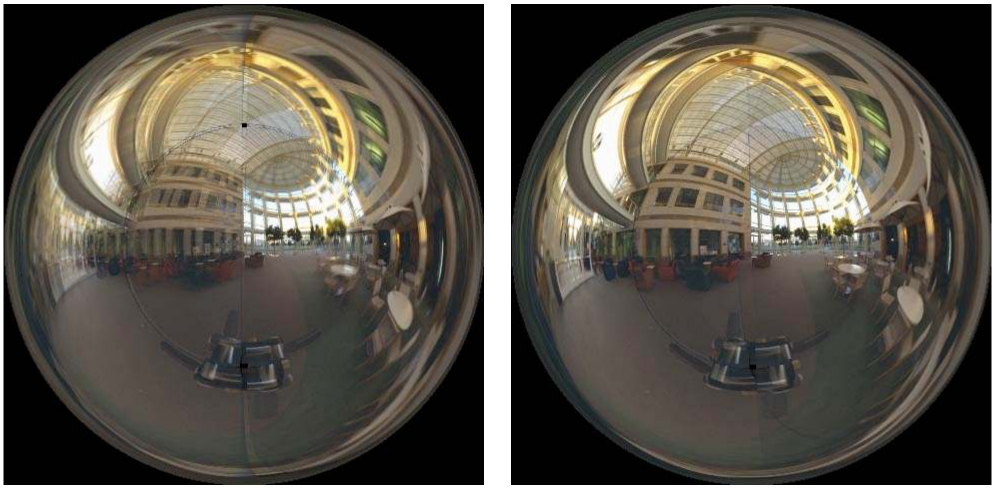


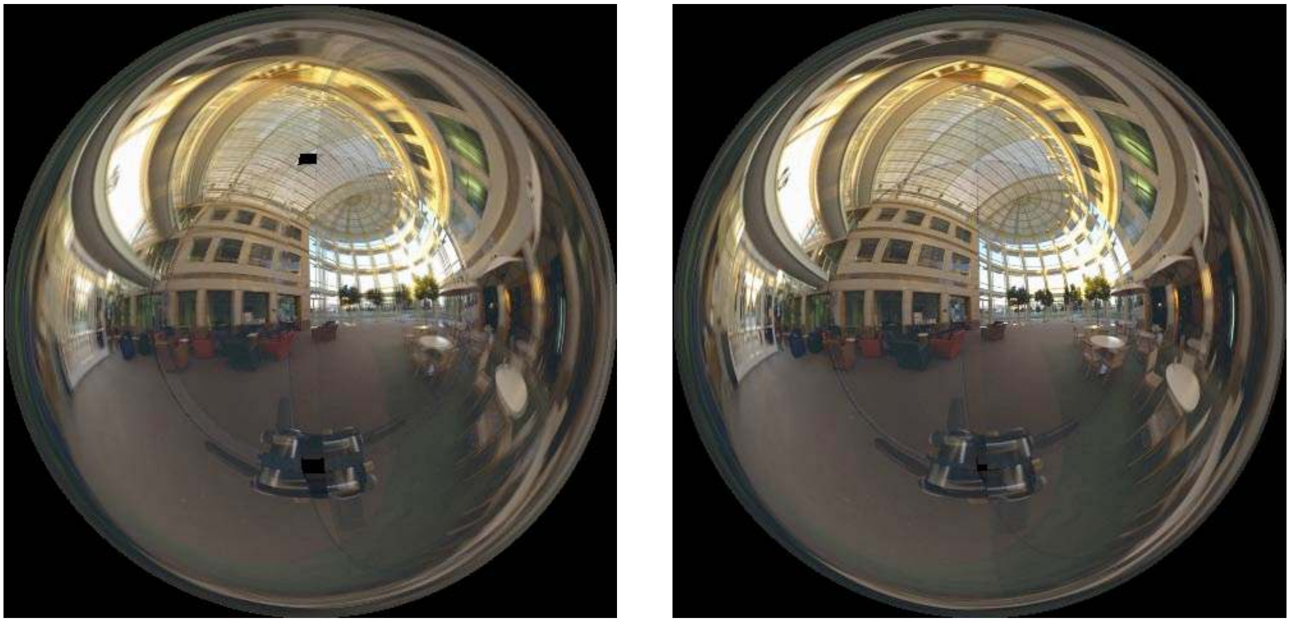
Figure 5: Four Fisheye Images



1 Iteration

10 Iterations

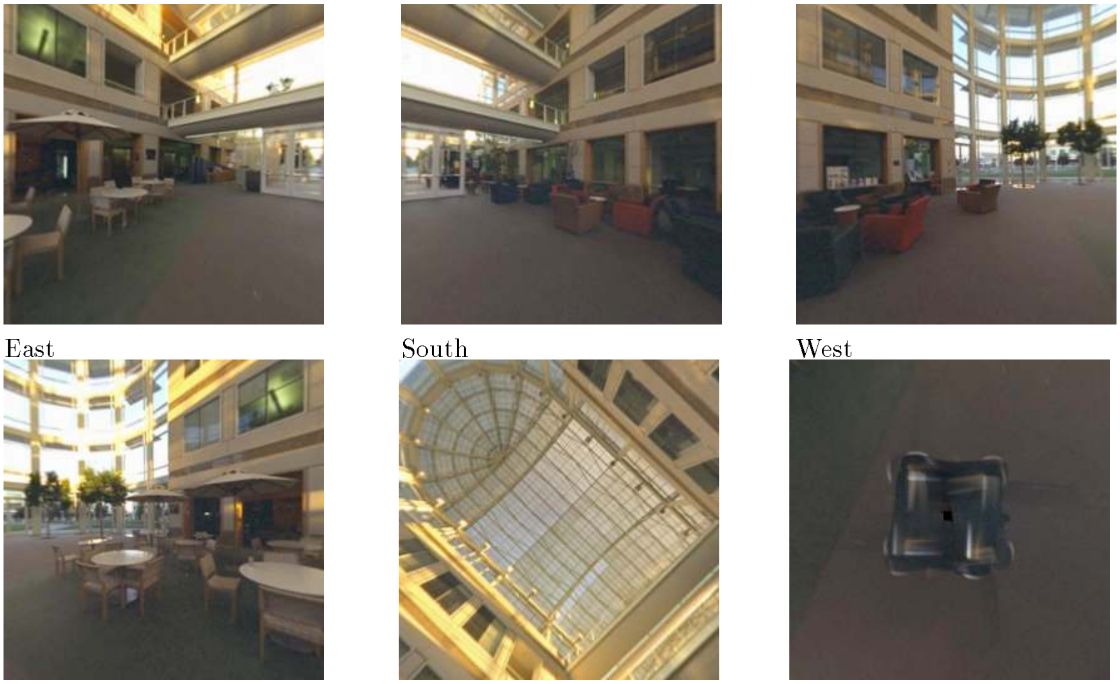
Figure 6: Iteratively Registering Fisheye Images



Ideal Equi-distance Model

Cubic Distortion Model

Figure 7: Final Results from Registration



East

South

West

North

Top

Bottom

Figure 9: Cubic Representation of the Panorama