Acquiring Immersive Virtual Environments with an Uncalibrated Camera

Leonard McMillan

Abstract

Immersion is arguably the most significant distinction of virtual environments from traditional three-dimensional computer graphics. When using standard geometric scene representations we need only provide rich object descriptions of appropriate scales to achieve such an immersive environment. The situation is more complicated in the case of image-based rendering systems.

In this report a method for generating a 360 degree field-of-view image from a sequence of planar projections is described. This method determines a camera model based on the analysis of an image sequence that results from the constant panning of a camera about its nodal point. Once this camera model is determined the images may be reprojected onto any arbitrary surface such as a cylinder. I also describe an important geometric constraint that is useful for relating the geometry between any pair of cylindrical projections. This constraint describes a curve in space which completely describes the apparent trajectory of points resulting from motion of the camera along the direction connecting the two projections. This constraint plays the same similar role for cylindrical projections that the epipolar constraint plays for planar projections.

Introduction

A major feature of virtual environments is their ability to immerse users within a synthetic computer-generated environment which responds naturally to the users' movements and other interactions. The specification, or modeling, of such environments is a nontrivial task, particularly when the goal is to simulate a visually realistic setting. Ironically, we are surrounded everyday by a world from which we derive our notions of realism. We have access mechanisms for recording this realism, in the form of photographs and video. And recently, the apparatus for transferring these recorded images into computer readable form has become widely available. Nonetheless, it is still very difficult to use real-world images within virtual environments in all but the most trivial of ways, such as paintings or wallpaper.

There are many fundamental difficulties in placing accurate geometric interpretations on image data. This results from the fact that an image embodies a projection of a complex interaction of the light with diverse surfaces, both seen and unseen, which compose a given scene. Since projection alone is a many-to-one mapping, it introduces the possibility of ambiguous solutions. Plus, precise surface properties and characterization of the environment's lighting are seldom available, further complicating matters. A wide variety of approaches have been attempted including shape-from-shading, shape-from-texture, shape-from-motion, shape-from-stereo, shape-from-focus, and structured light approaches. Large segments of the computer-vision and photogrammetry fields have devoted themselves to overcoming these problems, and most are still active areas of research.

In this paper, I will suggest methods for acquiring a series of images which can be registered, combined, and related geometrically to create panoramic (full 360 degree field-of-view) models which are suitable for immersive virtual environments. A significant feature of this approach is that the projective relationships between the images, and their positions in the real world is inferred from the image data alone, with a minimum of human intervention. This allows one to construct realistic models without the need for tracking devices, special cameras, motion platforms, or calibration procedures. In fact, little more than a camera, tripod, and an optional tape measure is needed. The tape measure is needed only for a single measurement between two points within the environment to establish a scale factor. Often times this factor can be accurately estimated without the need for an actual measurement. In the first section, I will discuss a robust technique for acquiring and registering images into a panoramic cylindrical view. In the second section, I will discuss some of geometric invariants of cylindrical projections and their ramifications for both locating and extracting geometry from multiple cylindrical views.

Creating Cylindrical Projections

The process of projection involves the reduction or collapsing of a spatial dimension. In the case of three-dimensions, a projection is a systematic mapping of all points in space onto a two-dimensional manifold. Perspective projections, an important subset of such mappings, correspond to views of the world as seen from a particular point. In this work we will consider cylindrical perspective projections (i.e. panoramic views) which encompass all possible gaze directions.

There are an endless number of possible surfaces upon which perspective projections can be mapped. The most natural surface for projecting a complete panoramic view is a sphere centered about the viewpoint, or so-called center of projection. One difficulty of a spherical projection, though, is the representation of such a mapping in a form which is suitable for storage on a computer. This is particularly difficult if a uniform (i.e. equal area) discrete sampling is required. This difficulty is reflected in the various distortions which arise in planar projections of world maps in cartography. Those uniform mappings which do exist are generally ill-suited for systematic access as a data structure. And those which do map to a plane with consistent neighborhood relationships are generally quite distorted.

A set of six planar projections in the form of a cube has also been suggested by [Green86] as an efficient representation for environment maps. While this representation can be easily stored and accessed by a computer, it provides significant problems relating to acquisition, alignment, and registration when used for real, non-computer-generated images. The orthogonal orientation of the cube faces requires precise camera positioning. The wide, 90 degree field-of-view of each face requires expensive lens systems to avoid optical distortion. Also, the planar mapping does not represent a uniform sampling, but instead, is considerably oversampled in the edges and corners. However, the greatest difficulty of a cube oriented planar projection set is computing the correspondence information required to extract geometry at and across the boundaries between faces and at corners. This is not an issue when the environment is used solely as a reflection map, but it adds considerable nonlinearities and boundary conditions when used in image analysis.

I have chosen to use a cylindrical surface upon which to project the environments for this work. One advantage of a cylinder is that it can be easily unrolled into a simple planar map. The surface is without boundaries in the azimuth direction, which simplifies correspondence searches. One shortcoming of a projection on a finite cylindrical surface is the boundary conditions introduced at the top and bottoms. I have chosen not to employ end caps on the tops and bottoms of the projection. This has the problem of limiting the vertical field of view within the environment. However, with a multiple cylinder model this liability can be overcome by dynamically switching to a more back-facing projection as the user looks up or back.

The most significant advantage of a cylindrical projection is the simplicity of acquisition. The only equipment required is a video camera and a tripod capable of continuous panning. Ideally the camera's panning motion would be the exact optical center of the camera. In practice, in a scene where all objects are relatively far from the tripod's rotational center, a slight misalignment of the rotational axes can be tolerated, as shown in the following analysis.

Any pair of planar perspective projections of a static scene with a common center of projection are related by a two-dimensional homogenous transform:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = \frac{u}{w} \qquad y' = \frac{v}{w}$$

where x and y represent the pixel coordinates of an image I_0 and x' and y' are their corresponding coordinates in a second image related by the transform.

This well known result has been reported by several authors [Heckbert89], [Wolberg90], [Mann94]. This relationship can be seen intuitively by considering that each point on the image plane corresponds to a ray originating from the center of projection and passing through a point in the scene. As long as the center of projection remains fixed, the set of rays is identical regardless of the viewplane's position and orientation. The relative positions of a ray on the image plane is completely determined by the projection process which can be described by a homogenous transform. The images resulting from typical camera motions, such as pan, tilt, roll, and zoom can all be related in this fashion. For creating a cylindrical projection we will only need to consider panning motions.

For convenience we will define the camera's local coordinate system such that the panning takes place entirely in the x-z plane. One additional requirement of the algorithm is that some part of each image of the sequence also be visible in the successive image. For the final image, some part must be visible in the first image of the sequence.

In order to reproject an individual image into a cylindrical projection, we must determine several camera parameters or, equivalently, the appropriate homogenous transforms. The registration process determines the transformation for all of the images of the sequence to any other image.

Many different techniques have been developed for inferring the homogenous transformation between two images sharing the same center of projection. The most common technique used by [Heckbert89], [Mann94], and [Szeliski94], involves establishing four corresponding points across the image pair. The resulting transforms provide a mapping of pixels from the planar projection of one image to the planar projection of the second. The lower right element, a_{33}, of the homogenous transform matrix serves as a global scale factor which is projected out (divided out) when the homogenous coordinate is converted to an image space coordinate. Several images could be composited in this fashion by first determining the transform which maps the Nth image to the N-1st. These transforms can be concatenated to form a mapping of each image to the plane of the first. This approach in affect avoids direct determination of all of the relevant camera parameters by performing all mapping between different instances of the same camera. However, we cannot avoid determining the camera's position when composing an arbitrary reprojection from a set of planar projections.

We can relate a set of homogenous transforms, $\boldsymbol{H_i}$, in a way that will allow arbitrary reprojections if we decompose each matrix into two parts: an intrinsic transform, $\boldsymbol{S}$, which is determined entirely by properties of the camera, and an extrinsic transform, $\boldsymbol{R_i}$, which is determined by the motion of the projection plane about the camera's center of projection.

$$\vec{u} = \boldsymbol{H}_i \vec{x} = \boldsymbol{S}^{-1} \boldsymbol{R}_i \boldsymbol{S} \vec{x}$$

This decomposition allows us to arbitrarily reproject the images since we have, in effect, decoupled the projection and rotational components of the homogeneous transform. By an appropriate choice of coordinate systems and by limiting the camera's motion to panning, the extrinsic transform component can be neatly constrained to a function of a single parameter, which describes a rotation about the panning axis and has the following form.

$$\boldsymbol{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Since the intrinsic component's properties are invariant over all of the images, the problem can be neatly broken into two part: the determination of the extrinsic component describing the motion, and then, the determination of an extrinsic component describing the projection. The first step in my method determines estimates for the extrinsic panning angle between each pair of the panning sequence. This is accomplished by using a linear approximation to an infinitesimal rotation, Ω. This linear approximation results from substituting $1 + O(\Omega^2)$ for the cosine terms and $\Omega + O(\Omega^2)$ for the sine terms of the rotation matrix. This infinitesimal perturbation has been shown by [Kanatani88] to reduce to the following approximate equations:

$$x' = x - f\Omega - \frac{\Omega (x - C_x)^2}{f} + O(\Omega^2)$$

$$y' = y - \frac{\Omega (x - C_x)(y - C_y)}{f} + O(\Omega^2)$$

where f is the apparent focal length of the camera measured in pixels (this value will be derived in the registration process), and (C_x, C_y) is the pixel coordinate of where the center-of-projection projects onto the image plane along the direction of the optical axis (sometimes called the look-at or gaze direction). (C_x, C_y) is initially estimated to be at the center pixel of the viewplane. A more refined estimate is found by the intrinsic matrix solution.

From these equations it can be easily seen that a small panning rotation can be approximated by a translation for pixels near the center of the screen. Given the requirement that successive camera images must already contain overlapping regions in order to achieve any sort of correspondence and that a typical wide-angle video lens of approximately 9 mm lens on a 1/3 inch CCD provides only a 40 degree field-of-view, this suggests a reasonable limit of 20 degrees, assuming that at least half of the image plane is visible in successive frames. Kanatani's approximation can be shown to exhibit less than one-half of a pixel deviation from an exact solution for pixels in the center third of the screen for fields-of-view in this 20-degree range.

Therefore, the first stage of the cylindrical registration process attempts to register the set of images by computing the optimal translation in x which maximizes the normalized correlation within a constrained region about the center third of the screen. This is first computed at a pixel resolution then repeated at a 0.1 subpixel grid, using a Catmull-Rom interpolation spline to compute subpixel intensities. Once these translations are computed we use Newton's method to convert these translations to estimates of rotation angles and the focal length, according to the following equation:

$$2\pi - \sum_{i=0}^{N} \operatorname{atan}\left(\frac{t_i}{f}\right) = 0$$

This usually converges in as few as 5 iterations, depending on the original estimate for f. This first phase determines an estimate for the relative rotational angles between each of the images (our extrinsic parameters) and the focal length measured in pixels (one of the intrinsic parameters).

The second stage of the registration process determines the S matrix mentioned previously. This so-called structural matrix describes various camera properties as well as tilt and roll angles which are assumed to remain constant over the group of images. The following model is used to describe the elements of the projection matrix.

$$S = \Omega_x \Omega_y \Omega_z \boldsymbol{R}_x \boldsymbol{P}$$

where **P** is the projection matrix:

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & f \end{bmatrix}$$

and (C_x, C_y) is the estimated center of the viewplane (i.e. the projection of the center-of-projection onto the viewplane along the optical axis) and f is the focal length in pixels determined from the first stage.

The $\mathbf{R}_x$ term is a single parameter matrix which describes the relative tilt of the camera's optical axis out of the panning plane.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

If ϕ is zero the planar projections are all tangent to a cylinder. For a nonzero ϕ the locus of projections is tangent to a cone.

$$\Omega_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{bmatrix} \qquad \Omega_y = \begin{bmatrix} \cos\omega_y & 0 & -\sin\omega_y \\ 0 & 1 & 0 \\ \sin\omega_y & 0 & \cos\omega_y \end{bmatrix} \qquad \Omega_z = \begin{bmatrix} \cos\omega_z & -\sin\omega_z & 0 \\ \sin\omega_z & \cos\omega_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The remaining three terms, Ω_x, Ω_y, and Ω_z, describe the deviations of the viewplane's orientation from perpendicular to the optical axis. Ideally, the viewplane would be normal to the optical axis, but manufacturing tolerances allow these numbers to vary by as much as two degrees. In addition, the ω_z term is indistinguishable from the camera's roll angle and, thus, represents both the image sensor's and the camera's rotation. Each of the viewplane orientation matrices are defined by a single parameter. This gives six unknown parameters, (C_x, C_y, ϕ, ω_x, ω_y, ω_z), to be determined in the second stage. Notice that, when combined with θ_i and f determined in the first stage, this gives a total of eight parameters for each image, which is consistent with the number of free parameters in a general homogeneous matrix. This is also consistent with our claim that $\mathbf{S}^{-1}\mathbf{R}_y\mathbf{S}$ is a homogeneous transform.

We can easily cast the solution of **S** as a nonlinear minimization problem, which attempts to minimize the following error function:

$$E(C_x, C_y, \phi, \omega_x, \omega_y, \omega_z) = \sum_{i=1}^{N} 1.0 - Correlation\left(I_{i-1}, \boldsymbol{S}^{-1}\boldsymbol{R}_{y_i}\boldsymbol{S}I_i\right)$$

where I_{i-1} and I_i represent the center third of the pixels from image i-1 and i respectively. Using Powell's multivariable minimization method and the following initial values for our six parameters,

$$C_x = \frac{\text{screen width}}{2} \qquad C_y = \frac{\text{screen height}}{2}$$

$$\phi = 0 \qquad \omega_x = 0 \qquad \omega_y = 0 \qquad \omega_z = 0$$

the solution typically converges in as few as six iterations. At this point we will have a new estimate for (C_x, C_y) which can be fed back into stage one and the entire process can be repeated.

The registration process results in a single camera model, $S(C_x, C_y, \phi, \omega_x, \omega_y, \omega_z, f)$, and a set of the relative rotations, θ_i, between each of the sampled images. Using these parameters we can compose mapping functions from any image in sequence to any other image as follows.

$$I'_i = S^{-1} R_{y_{i+1}} R_{y_{i+2}} R_{y_{i+3}} \dots R_{y_j} S I_j$$

We can also reproject images onto arbitrary surfaces by modifying **S**.

Relating Cylindrical Projections

Given two or more cylindrical projections from different positions within the same static scene, computed using the methods described previously, the question immediately arises - how might we relate these images? This amounts to determining the relative positions of centers of projection and establishing geometric constraints on the cylindrical projective frame. In general, a solution can be computed which is accurate to a scale factor. The intuitive argument is that from a set of images alone, one cannot determine if the observer is looking at a model or a full-sized scene. This implies that at least one measurement is required to establish a scale factor. The measurement may be between objects that are visible in both scenes or between the cameras where the images were generated. Both techniques have been used with little difference in results.

To establish the relative relationships between any pair of cylindrical projections the user specifies a set of corresponding points that are visible from both views. These points can be treated as rays in space with the following form:

$$\vec{x}_a(\theta, v) = \vec{C}_a + t \begin{bmatrix} \cos(\phi_a - \theta) \\ \sin(\phi_a - \theta) \\ k_a(C_{v_a} - v) \end{bmatrix}$$

where $C_a = (A_x, A_y, A_z)$ is the unknown position of the cylinder's center of projection, ϕ_a is the rotational offset which aligns the angular orientation of the cylinders to a common frame, k_a is a scale factor which determines the vertical field-of-view, and C_{va} is the scanline where the center of projection would project onto the scene (i.e. the line of zero elevation, like the equator of a spherical map).

A pair of tiepoints, one from each image, establish a pair of rays which would ideally intersect at the point in space identified by the tiepoint. In general, however, these rays will be skewed. Thus, we use the point which is simultaneously closest to both rays as an estimate of the point's position. This can be determined by the following derivation.

$$\vec{p} = \frac{\vec{x}_a + \vec{x}_b}{2} = P((\theta_a, v_a), (\theta_b, v_b))$$

$$\vec{x}_a = \vec{C}_a + t\vec{D}_a(\theta_a, v_a) \qquad \text{where } (\theta_a, v_a) \text{ is the coordinate of the tiepoint on cylinder A}$$

$$\vec{x}_b = \vec{C}_b + s\vec{D}_b(\theta_b, v_b) \qquad \text{where } (\theta_b, v_b) \text{ is the coordinate of the tiepoint on cylinder B}$$

$$t = \frac{\mathrm{Det}\left\{\left(\vec{C}_b - \vec{C}_a\right), \vec{D}_b(\theta_b, v_b), \vec{D}_a(\theta_a, v_a) \times \vec{D}_b(\theta_b, v_b)\right\}}{\left\|\vec{D}_a(\theta_a, v_a) \times \vec{D}_b(\theta_b, v_b)\right\|^2}$$

$$s = \frac{\mathrm{Det}\left\{\left(\vec{C}_b - \vec{C}_a\right), \vec{D}_a(\theta_a, v_a), \vec{D}_a(\theta_a, v_a) \times \vec{D}_b(\theta_b, v_b)\right\}}{\left\|\vec{D}_a(\theta_a, v_a) \times \vec{D}_b(\theta_b, v_b)\right\|^2}$$

This formulation allows us to pose the problem of finding a cylinder's position as a minimization problem. For each pair of cylinders we have two sets of 6 unknowns, [(A_x, A_y, A_z, ϕ_a, k_a, C_{va}), (B_x, B_y, B_z, ϕ_b, k_b, C_{vb})]. In general, we have good estimates for the k and C_v terms, if these values are returned by our reprojection phase. We can determine the position of the cylinder by minimizing the following error function:

$$\mathrm{Error}\left((A_x, A_y, A_z, \phi_a, k_a, C_{v_a}), (B_x, B_y, B_z, \phi_b, k_b, C_{v_b})\right) =$$

$$\sum_{i=0}^{N} \frac{\left\|\vec{x}_a(\theta_a, v_a)_i - P((\theta_a, v_a)_i, (\theta_b, v_b)_i)\right\|}{\left\|t\vec{D}_a(\theta_a, v_a)_i\right\|} +$$

$$\sum_{i=0}^{N} \frac{\left\|\vec{x}_b(\theta_b, v_b)_i - P((\theta_a, v_a)_i, (\theta_b, v_b)_i)\right\|}{\left\|s\vec{D}_b(\theta_b, v_b)_i\right\|} + \lambda\left((k_a - k_{a_0})^2 + (k_b - k_{b_0})^2\right)$$

The value of λ associates a penalty for shrinking the vertical height of the cylinder in order to bring points closer together. I have found a value of $\lambda = 1.0$ to be adequate. The remaining error terms are independent of a scale factor since they represent a ratio of two lengths whose ideal values would be zero. Powell's method was again used to solve for the local minimum. We can initialize the search with $C_a = (0, 0, 0)$ and $C_b = (1, 0, 0)$ if estimates for the centers are unknown. Next we can determine a scale factor for the system by computing $d = \left\|\vec{x}_a(\theta_a, v_a)_i - \vec{x}_a(\theta_a, v_a)_j\right\|$, for two points of a known separation and scaling the entire system such that d equals this known distance.

I have tested this approach using from 12 to 500 tiepoints, and have found that it converges to a solution in as few as ten iterations. Since no correlation step is required, this process is considerably faster than the minimization step described in the previous section.

The use of a cylindrical projection introduces significant geometric constraints on where a point viewed in one projection might appear in a second. We can capitalize on these restrictions when we wish to automatically identify corresponding points on both cylinders. While an initial set of 100 to 500 tiepoints might be established by hand, this process is far to tedious to establish a mapping for the entire cylinder. I will present a specific geometric invariant for cylindrical projections which constrains the positions where a point can appear on a cylindrical projection once given its position in some other cylinder. This constraint plays the same role that the epipolar geometry constraint, commonly used in the computer vision community for depth-from-stereo computations, plays for planar projections.

Before deriving the invariant, I will present an intuitive argument for its existence. Consider yourself at the center of a cylindrical projection. Every point on the cylinder around you corresponds to a ray in space whose equation was given previously. When one of the rays is observed from a second cylinder, its path projects to a curve which appears to begin at the point corresponding to the origin of the first cylinder, and it is constrained to pass through the point's image on the second cylinder.

This same argument could obviously have been made for a planar projection. And, since two points are identified (the virtual image of the camera in the second projection along with the corresponding point) and, because a planar projections preserve lines, a unique, so called epipolar line is defined. This is the basis for an epipolar geometry, which identifies pairs of lines in two planar projections such that if a point falls upon one line in the first image, it is constrained to fall on the corresponding line in the second image. The existence of this invariant reduces the search for corresponding points from an $O(N^2)$ problem to $O(N)$.

Cylindrical projections, however, do not preserve lines. In general, lines map to quadratic parametric curves on the surface of a cylinder. One would expect, since a parametric quadratic has three free variables, that at least three points would be required to establish any sort of invariant. Surprisingly, however, we can specify the form of the curve with exactly the same information as the planar case. In addition, the paths of these curves can be uniquely determined using a phasor diagram, similar to those used in engineering disciplines to relate two orthogonal variables.

This *Cylindrical Epipolar Geometry* is established as follows:

From before, the equation of a ray with an origin corresponding to the center of a cylindrical projection is specified as:

$$\vec{x}_a(\theta, v) = \overrightarrow{C_a} + t\begin{bmatrix} \cos(\phi_a - \theta) \\ \sin(\phi_a - \theta) \\ k_a(C_{v_a} - v) \end{bmatrix} = \vec{C}_a + t\vec{D}_a$$

$$\vec{x}_b(\theta, v) = \overrightarrow{C_b} + t\begin{bmatrix} \cos(\phi_b - \theta) \\ \sin(\phi_b - \theta) \\ k_b(C_{v_b} - v) \end{bmatrix} = \vec{C}_b + t\vec{D}_b$$

We can determine the unique plane connecting the points C_a, C_b, and a specific vector direction on the cylinder D.

$$\vec{N} \cdot \vec{x} = d$$
$$\vec{N} = (\vec{C}_b - \vec{C}_a) \times \vec{D} = \vec{C}_b \times \vec{D} + \vec{D} \times \vec{C}_a$$
$$d = \vec{C}_a \cdot \vec{N} = \vec{C}_a \cdot (\vec{C}_b \times \vec{D} + \vec{D} \times \vec{C}_a) = \vec{C}_a \cdot (\vec{C}_b \times \vec{D}) = Det\{\vec{C}_a, \vec{C}_b, \vec{D}\}$$

Substituting the parametric line $\vec{x}_a(\theta, v)$ into the plane equation yields:

$$(\vec{C}_a + t\vec{D}_a) \cdot (\vec{C}_b \times \vec{D} + \vec{D} \times \vec{C}_a) = \vec{C}_a \cdot (\vec{C}_b \times \vec{D})$$
$$\vec{C}_a \cdot (\vec{C}_b \times \vec{D}) + \vec{C}_a \cdot (\vec{D} \times \vec{C}_a) + t\vec{D}_a \cdot (\vec{C}_b \times \vec{D} + \vec{D} \times \vec{C}_a) = \vec{C}_a \cdot (\vec{C}_b \times \vec{D})$$
$$t\vec{D}_a \cdot (\vec{C}_b \times \vec{D} + \vec{D} \times \vec{C}_a) = t\vec{D}_a \cdot \vec{N} = 0$$

This equation has two solutions. The first, where t is equal to zero, is uninteresting. The second solution corresponds to

$$\vec{D}_a \cdot \vec{N} = 0$$
$$N_x \cos(\phi_a - \theta) + N_y \sin(\phi_a - \theta) + N_z k_a(C_{v_a} - v) = 0$$

This result can be rearranged to give a function of v in terms of θ.

$$v = \frac{N_x \cos(\phi_a - \theta) + N_y \sin(\phi_a - \theta)}{N_z k_a} + C_{v_a}$$

This formula gives a concise expression for the curve formed by the projection of a ray across the surface of a cylinder, where the ray is specified by its position on some other cylinder. A similar expression results when calculating the curve traversed on cylinder b.

$$v = \frac{N_x \cos(\phi_b - \theta) + N_y \sin(\phi_b - \theta)}{N_z k_b} + C_{v_b}$$

Recognizing that the sum of two sinusoids of the same frequency always yields a third sinusoid, we can think of these equations as a sum of two orthogonal functions. This amounts to a phasor diagram relating the amplitudes of the cosinusoidal and sinusoidal parts. Thus, every point on a cylindrical projection can be associated with a phasor which completely defines its potential positions on all other cylinders. Other points in the projection which share this same phasor are constrained to lie in the same plane through the two cylinders' centers-of-projection.

Results

In my initial experiment I collected a series of images using a video camcorder set up on a leveled tripod in the front yard of my home. The leveling process was not extremely accurate, and luckily it is not necessary for the method to work. When the data was collected, no attempt was made to pan the camera at a uniform angular velocity. The autofocus and autoiris features of the camera were disabled, in order to maintain a constant focal length during the collection process. The frames were the digitized at a rate of approximately 5 frames per second to a resolution of 320 by 240 pixels. An example of three sequential frames are shown below.

Immediately after the collection of the first data set, the experiment was repeated at a second point approximately 60 inches from the first. The two image sequences were then separately registered using the methods described in the section on “Acquiring Cylindrical Projections”. The images were reprojected onto the surface of a cylinder with a resolution of 3600 by 300 pixels. The results are shown below.

A second experiment was devised to determine the accuracy of Kanatani’s linearization of the panning rotation. A still-video camera was mounted to a theodolite capable of measuring azimuth angles to an accuracy greater than 1 minute of arc. Then 40 images were computed at a constant angular separation of 9 degrees. During the digitization process, one of the 40 images was lost. In addition, it is believed that a measurement error occurred during the collection of the first image.

The following angles and camera matrix were computed using the registration procedure described.

$$S = \begin{bmatrix} 0.999967 & -0.00700886 & -109.298 \\ 0.00714953 & 0.999311 & -146.123 \\ -0.0037297 & 0.0364548 & 460.355 \end{bmatrix}$$

$\theta_0 = 6.46681$ $\theta_1 = 9.00737$
$\theta_2 = 9.01939$ $\theta_3 = 9.00737$
$\theta_4 = 8.98332$ $\theta_5 = 8.99534$
$\theta_6 = 9.04343$ $\theta_7 = 8.9713$
$\theta_8 = 8.9713$ $\theta_9 = 9.04343$
$\theta_{10} = 8.99534$ $\theta_{11} = 9.00737$
$\theta_{12} = 8.99534$ $\theta_{13} = 9.04343$
$\theta_{14} = 8.99534$ $\theta_{15} = 9.00737$
$\theta_{16} = 9.03141$ $\theta_{17} = 8.98332$
$\theta_{18} = 8.99534$ $\theta_{19} = 9.04343$
$\theta_{20} = 9.00737$ $\theta_{21} = 9.0915$
$\theta_{22} = 9.04343$ $\theta_{23} = 9.0915$
$\theta_{24} = 9.04343$ $\theta_{25} = 9.04343$
$\theta_{26} = 9.04343$ $\theta_{27} = 9.04343$
$\theta_{28} = 9.04343$ $\theta_{29} = 9.04343$

$\theta_{30} = 17.5455$ $\theta_{31} = 9.03141$
$\theta_{32} = 9.06746$ $\theta_{33} = 9.04343$
$\theta_{34} = 9.04343$ $\theta_{35} = 9.00737$
$\theta_{36} = 9.01939$ $\theta_{37} = 9.00737$
$\theta_{38} = 11.1336$

Aside from the errors mentioned, the estimated angles are all better than one tenth of a degree or 6 minutes of arc. The resulting image of the UNC Sitterson Hall lobby is shown below.

Acknowledgments

This research was done as a course project requirement for COMP 238 in the fall semester of 1994. The course was taught by J. Turner Whitted.

References

[Bolles87] Bolles, R. C., H. H. Baker, and D. H. Marimont, *"Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion,"* **International Journal of Computer Vision**, Vol. 1, 1987.

[Greene86] Greene, N., *"Environment Mapping and Other Applications of World Projections,"* **IEEE Computer Graphics and Applications**, November 1986.

[Hartley94] Hartley, R.I., *"Self-Calibration from Multiple Views with a Rotating Camera,"* **Proceedings of the European Conference on Computer Vision**, May 1994.

[Heckbert89] Heckbert, P. S., *"Fundamentals of Texture Mapping and Image Warping,"* Masters Thesis, Dept. of EECS, UCB, Technical Report No. UCB/CSD 89/516, June 1989.

[Horn81] Horn, B., and B.G. Schunck, *"Determining Optical Flow,"* **Artificial Intelligence**, Vol. 17, 1981.

[Kanatani88] Kanatani, K., *"Transformation of Optical Flow by Camera Rotation,"* **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 10, No. 2, March 1988.

[Lenz87] Lenz, R. K. and R. Y. Tsai, *"Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology,"* **Pro-**

ceedings of IEEE International Conference on Robotics and Automation, March 31 - April 3, 1987.

[Lippman80] Lippman, A., *"Movie-Maps: An Application of the Optical Videodisc to Computer Graphics,"* **SIGGRAPH '80 Proceedings**, 1980.

[Longuet81] Longuet-Higgins, H. C., *"A Computer Algorithm for Reconstructing a Scene from Two Projections,"* **Nature**, Vol. 293, September 1981.

[Longuet84] Longuet-Higgins, H. C., *"The Reconstruction of a Scene From Two Projections - Configurations That Defeat the 8-Point Algorithm,"* **Proceedings of the First IEEE Conference on Artificial Intelligence Applications**, Dec 1984.

[Lucas81] Lucas, B., and T. Kanade, *"An Iterative Image Registration Technique with an Application to Stereo Vision,"* **Proceedings of the Seventh International Joint Conference on Artificial Intelligence**, Vancouver, 1981.

[Mann94] Mann, S. and R. W. Picard, *"Virtual Bellows: Constructing High Quality Stills from Video,"* **Proceedings of the First IEEE International Conference on Image Processing**, November 1994.

[Szeliski94] Szeliski, R., *"Image Mosaicing for Tele-Reality Applications,"* **DEC and Cambridge Research Lab Technical Report**, CRL 94/2, May 1994.

[Tomasi92] Tomasi, C., and T. Kanade, *"Shape and Motion from Image Streams: a Factorization Method; Full Report on the Orthographic Case,"* Technical Report, CMU-CS-92-104, Carnegie Mellon University, March 1992.

[Tsai87] Tsai, R. Y., *"A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,"* **IEEE Journal of Robotics and Automation**, Vol. RA-3, No. 4, August 1987.

[Wolberg90] Wolberg, G., **Digital Image Warping,** IEEE Computer Society Press, Los Alamitos, CA, 1990.