Image-based Modeling and Rendering

3. Panorama (A)

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

Outline

- How to represent the environment with a few images?
- With view point constraints
  - Environment mapping
  - Cylindrical panorama

Ref:

• Image-based Modeling and Rendering, SIGGRAPH'99 course notes.

• L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", Proc. SIGGRAPH'95, pp. 39-46.

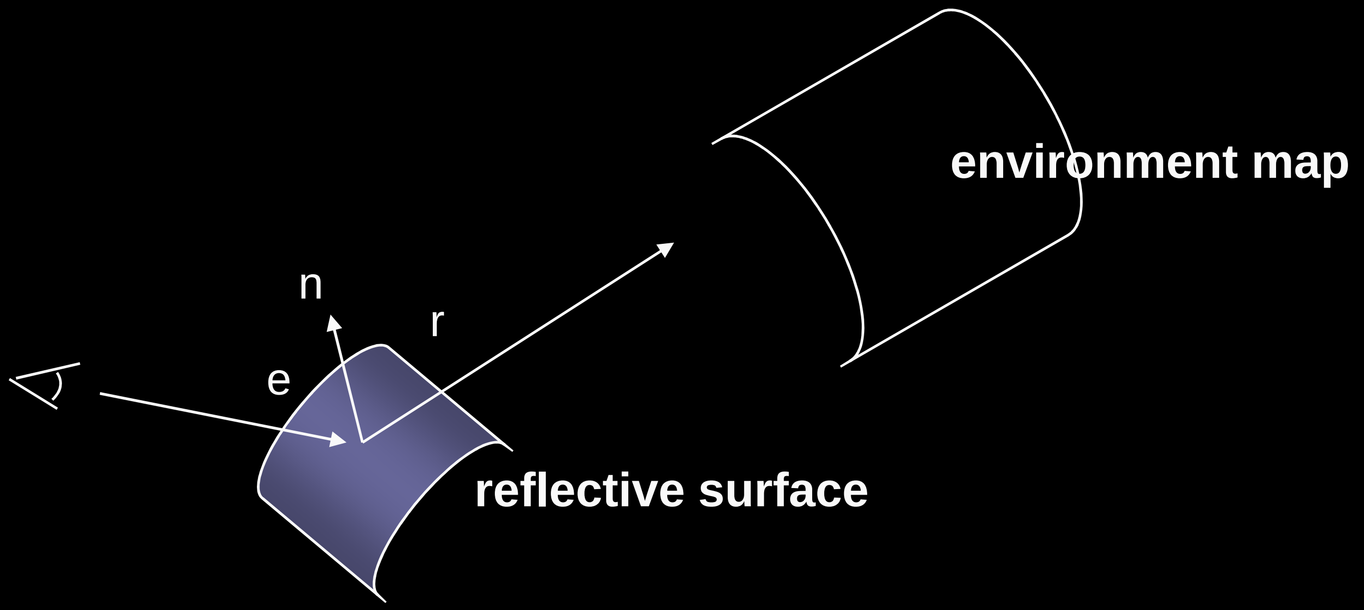
• S.E. Chen, "QuickTimeÒ VR – An Image-Based Approach to Virtual Environment Navigation", Proc. SIGGRAPH'95, pp. 29-38.

Environment Mapping

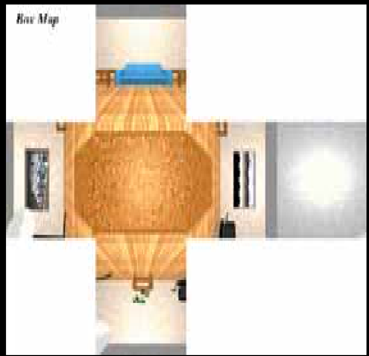
- A.k.a reflection mapping
- First proposed by Blinn and Newell.
- A efficient way to create reflections on curved surfaces
  - can be implemented using texture mapping supported by graphics hardware

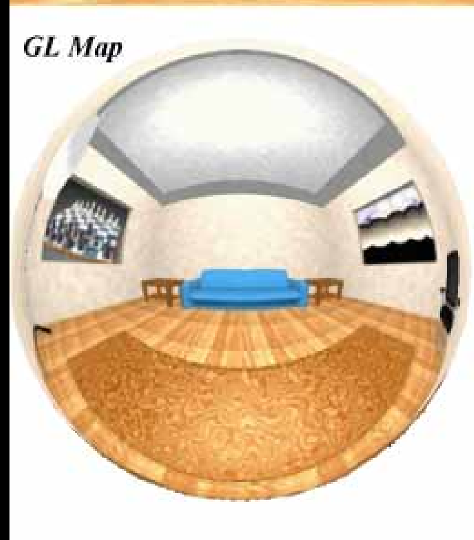
Environment Mapping

- Assume the environment is far away and there's no self-reflection
- The reflection at a point can be solely decided by the reflection vector.



Environment Mapping





Pictures from lecture notes of Computer Graphics course, UNC

Sphere Mapping

- The image texture is taken from a perfectly reflective sphere.
- Assume the size of the sphere →0. Map the rays to the environment

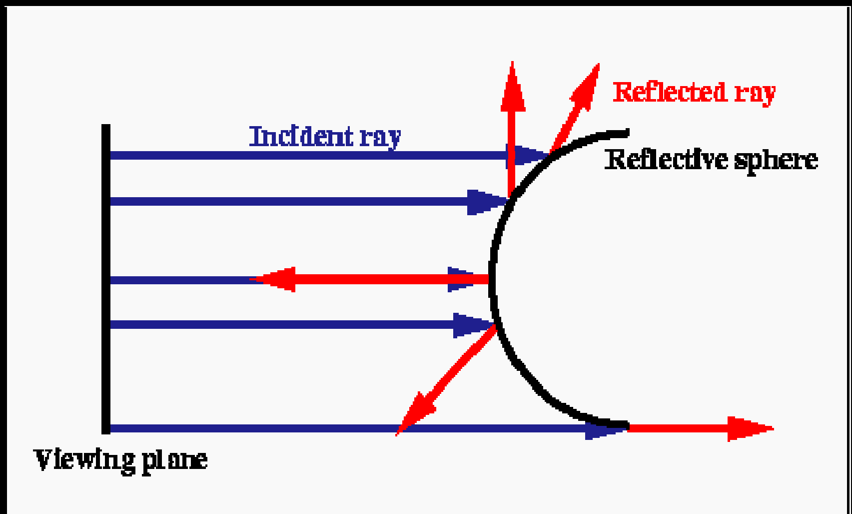


Figure 27. Creating a sphere map

Pictures from OpenGL tutorial. http://www.opengl.org

Sphere Mapping

- To access the sphere map texture
  - The surface normal (*n*) and eye (*e*) vectors need to be first transformed to the eye space
  - Then compute the reflection vector as usual

    $(r = (r_x, r_y, r_z) = e' - 2(n' \cdot e')n')$
  - Now, compute the sphere normal in the local space $n'' = (r_x, r_y, r_z) + (0,0,1)$

$$n''' = \left(\frac{r_x}{m}, \frac{r_y}{m}, \frac{r_z + 1}{m} \right)$$

$$m = \sqrt{r_x^{\ 2} + r_y^{\ 2} + (r_z + 1)^2}$$

  - Normalized the screen space from [-1,1] to [0,1]

$$s = \frac{r_x}{2m} + \frac{1}{2}$$

$$t = \frac{r_y}{2m} + \frac{1}{2}$$

Sphere Mapping (cont.)

- What're the advantages and disadvantages?
- From the aspects of :
  - Image quality (resolution)
  - Realism (accuracy)
  - Convenience
  - ……

Plenoptic Modeling: An Image-based Rendering System

Proposed by L. McMillan, G. Bishop,
Univ. North Carolina at Chapel Hill

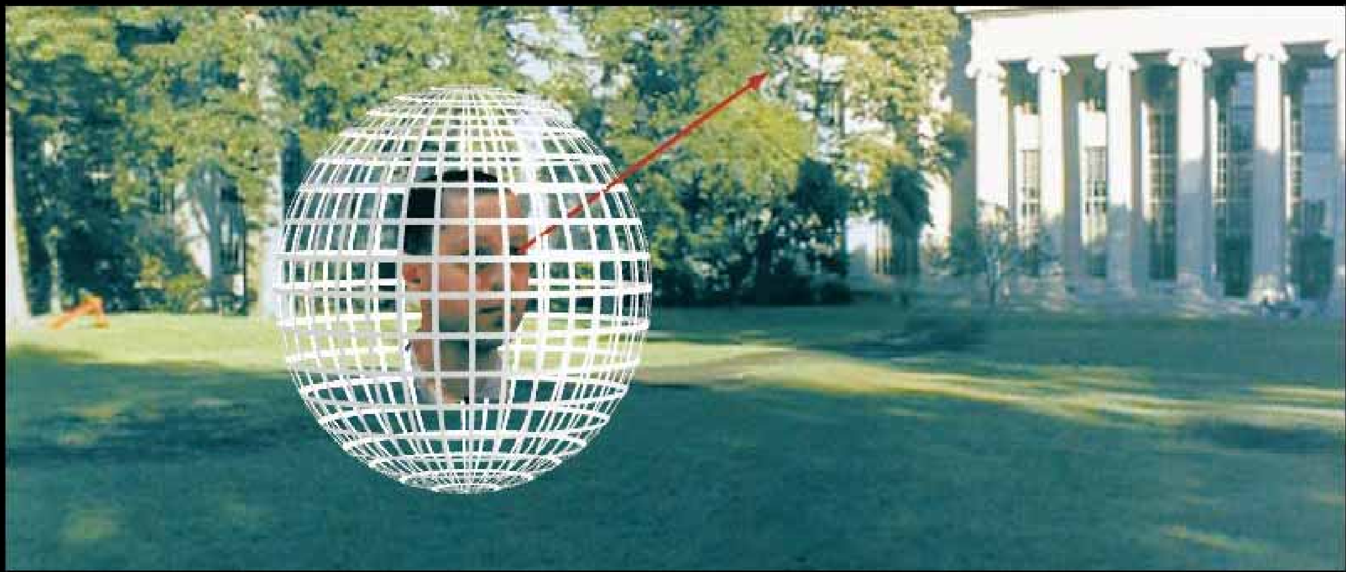
Proc. SIGGRAPH'95, pp. 39-46

Why do we use IBR?

- Very realistic scene descriptions can be acquired with a camera.
- Geometric models for real-world scenes are complicated and difficult to capture.

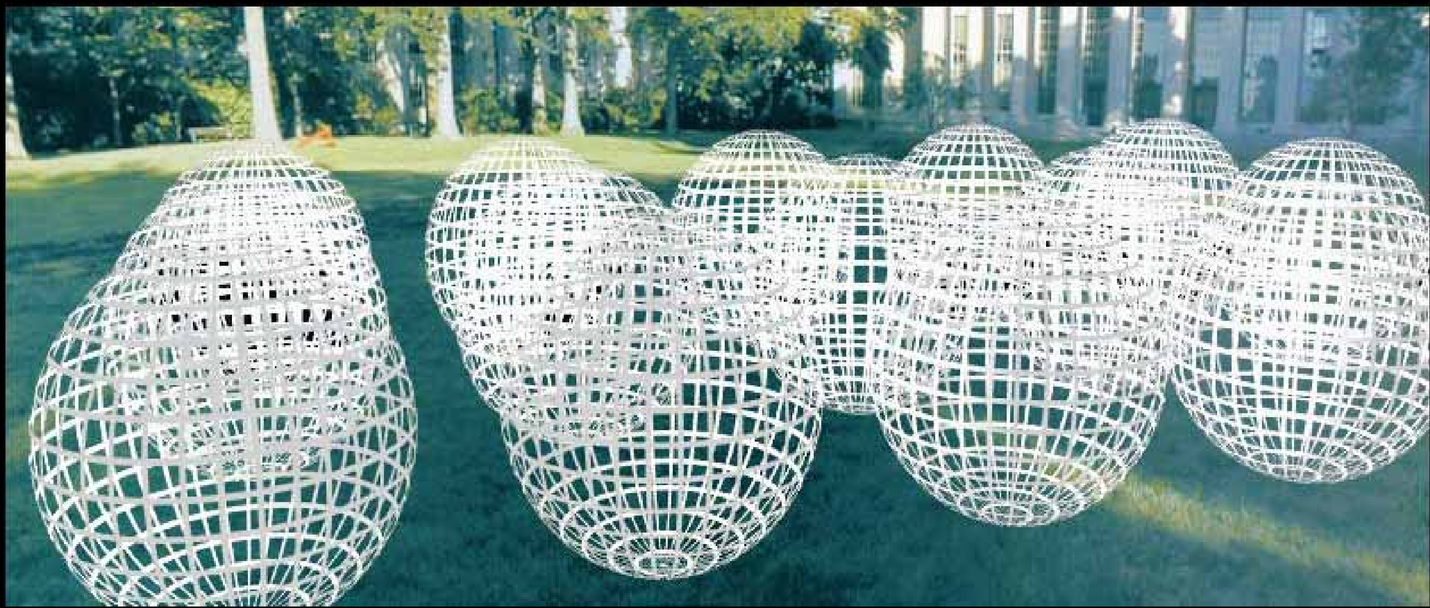
Images as a Collection of Rays

- An image is a subset of the rays seen from a given point
  - This “space” of rays occupies two dimensions



Images as a Collection of Rays (cont.)

- How to represent the whole scene with images?
- The set of rays seen from all view points.



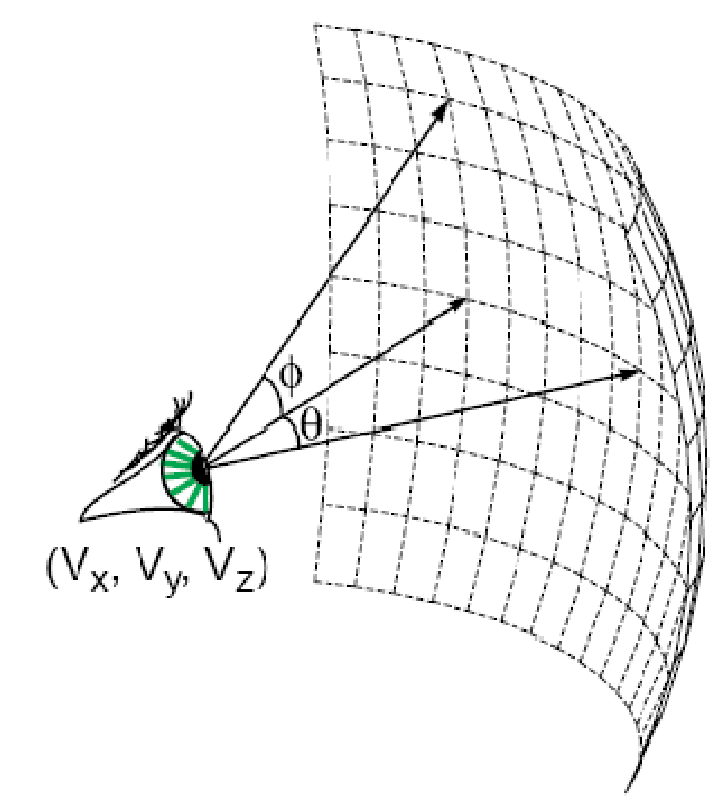
The Plenoptic Function

- A function that provides a complete description of the scene
- For every viewpoint and view direction, the plenoptic function describes the incident ray
- To render a scene, pick the view parameters and get the rays from the function.

$$\mu = P(\theta, \varphi, \lambda, V_x, V_y, V_z, t)$$

The Plenoptic Function (cont.)

- Taking into account only visible light in a static scene.
  - *t* is constant
  - *λ* is not a variable



- Plenoptic function is 5D

$$\mu = P(\theta , \varphi, Vx, Vy, Vz)$$

Panoramic images

- Choosing a warping equation that can be easily adapted.

Cylindrical Reference Images

- Good representation for real-world scenes
  - Maps to a rectangular grid easily
  - Complete coverage of scene in azimuth
  - Good coverage in elevation

- Spheres might be better, but they're difficult to map to a plane.
  - Distortion
  - Non-uniform sampling

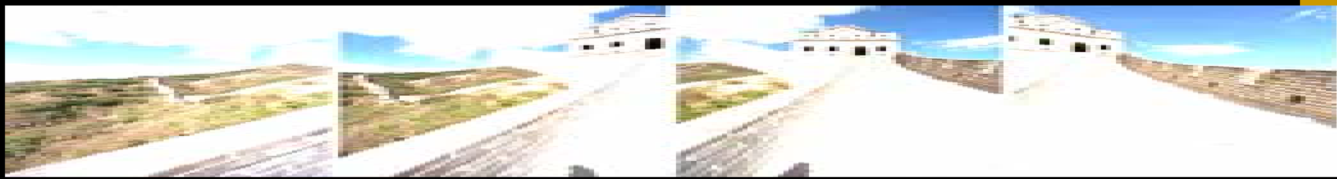
Plenoptic Modeling

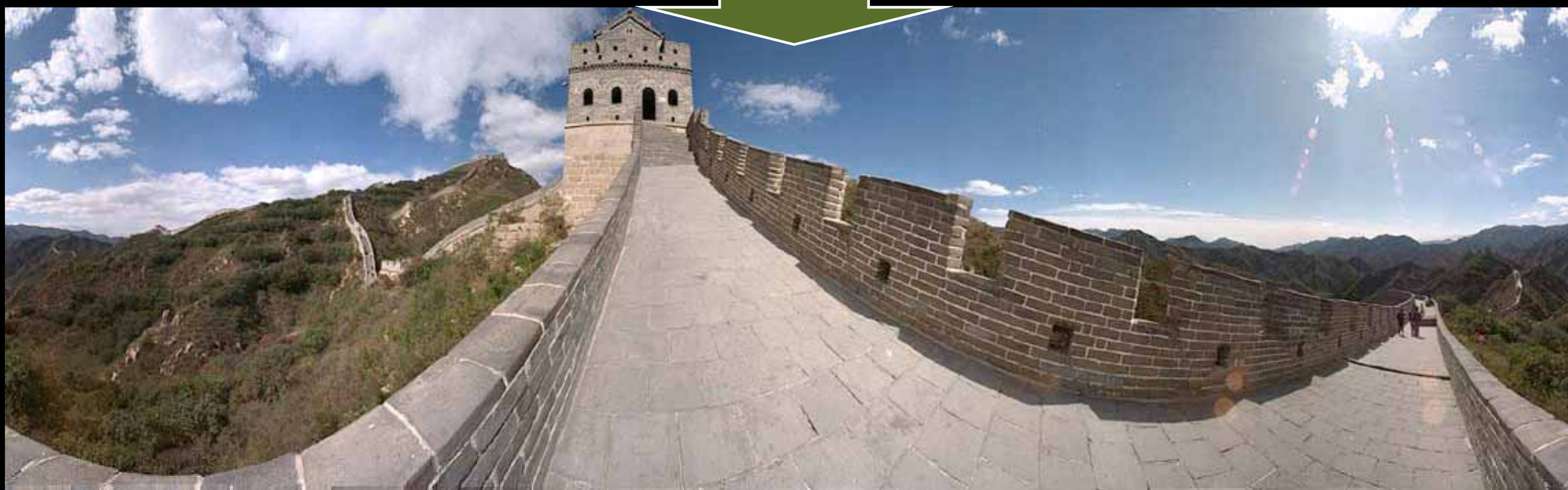
- Store reference images as cylindrical projections
- Sample PF to get reference images
- Infer flow field from reference images
- Resample PF to get the desired image

Acquiring Cylindrical Projections

- Using a regular camera with a panning tripod.
- Camera model and rotations can be inferred from planar reference images. (without camera calibration)
- Combine planar images and then compute the projection onto a cylinder.

Acquiring Cylindrical Projections





S.E. Chen, “QuickTime VR”, Proc. SIGGRAPH’95

Acquiring Cylindrical Projections

- Any two planar perspective projections of a scene which share a common viewpoint are:

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

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

- Separating intrinsic trans. **S** and extrinsic trans. ***R***:

$$\overline{u} = H_i \overline{x} = S^{-1} R_i S \overline{x}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Acquiring Cylindrical Projections

- Turn off auto-focus and other automatic adjustment function of the camera→ S is fixed.

- Estimating the parameters by optimization:

1. Initial guess.

2. Estimating $\boldsymbol{R_i}$ and $\boldsymbol{f}$.

3. Estimating $\boldsymbol{S}$.

Estimating R_i and f

- Using a linear approximation to an infinitesimal rotation.

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

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

  - f: focal length, (C_x, C_y) is the projection of the optical center. (initial guess: at the image center)

- Using the estimated translations t_i to estimate the rotation angles and focal length.

$$2\pi - \sum_{i=1}^{N} \operatorname{atan}\left(\frac{t_i}{f}\right) = 0$$

Estimating S

- S structure matrix can be decomposed into:

$$S = \Omega_x \Omega_z P$$

$$\Omega_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{bmatrix} \quad \Omega_z = \begin{bmatrix} \cos\omega_z & -\sin\omega_z & 0 \\ \sin\omega_z & \cos\omega_z & \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 & \sigma & -C_x \\ 0 & \rho & -C_y \\ 0 & 0 & f \end{bmatrix}$$

- Iteratively error minimization

$$error(C_x, C_y, \sigma, \rho, \omega_x, \omega_z) = \sum_{i=1}^{n} 1 - Correlation(I_{i-1}, S^{-1} R_y S I_i)$$

Initial: C_x = width/2, C_y = Height/2, σ=0, ρ=1, ω_x=0, ω_y=0

Panoramic Images

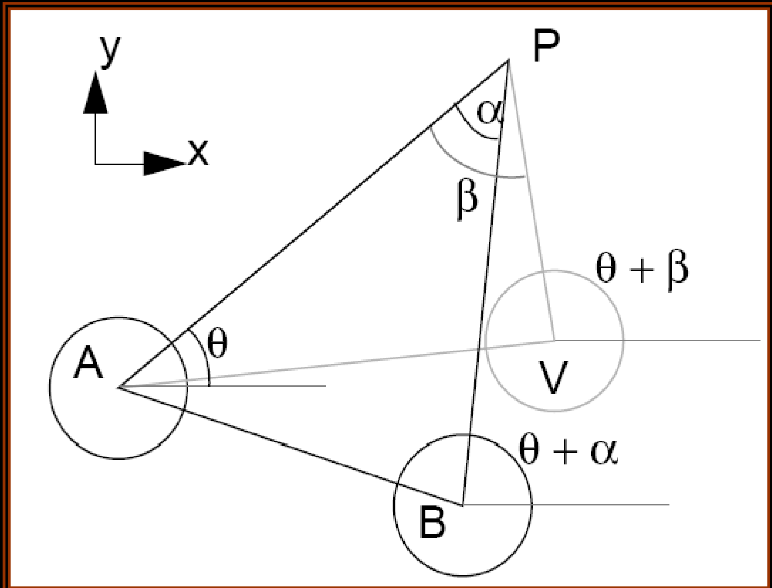
Perspective projection on view plane

Plenoptic Function Reconstruction

- How to estimate the whole plenoptic function from sparse view positions?

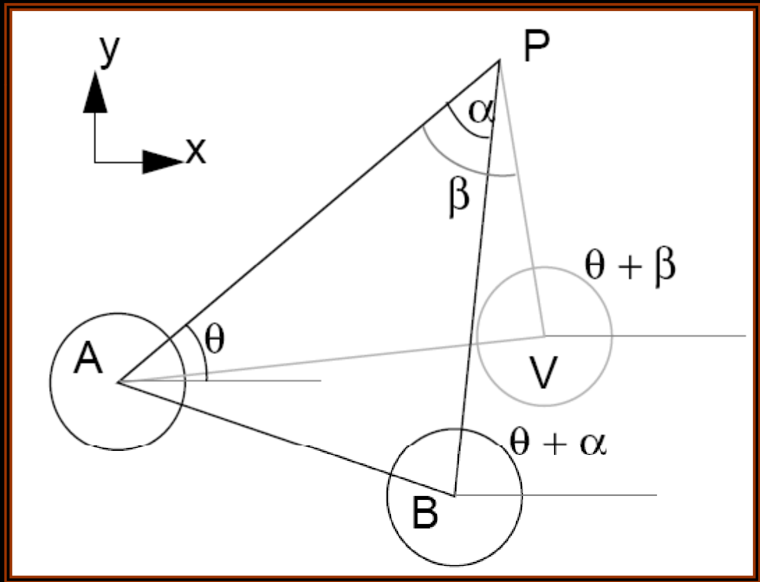
$$\bar{x}_a(\theta, v) = \bar{C}_a + t\bar{D}_a(\theta, v) \qquad \bar{D}_a(\theta, v) = \begin{bmatrix} \cos(\phi_a - \theta) \\ \sin(\phi_a - \theta) \\ k_a\left(C_{v_a} - v\right) \end{bmatrix}$$

- C_a 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
- C_{va} is the scanline where the center of projection would project onto the scene



Plenoptic Function Reconstruction

- Manually assign corresponding points for estimation of cylinder center difference and orientation alignment.
- How to reconstruct plenoptic function?
- Any problems?
  - Ambiguity
  - Efficiency of search



Visibility

- After performing the warp, we must take visibility into account.
- Scene elements may occlude other scene elements with the new center of projection.
- There is a scheme that uses Painter's algorithm along with rules for deciding which pixels get drawn first.