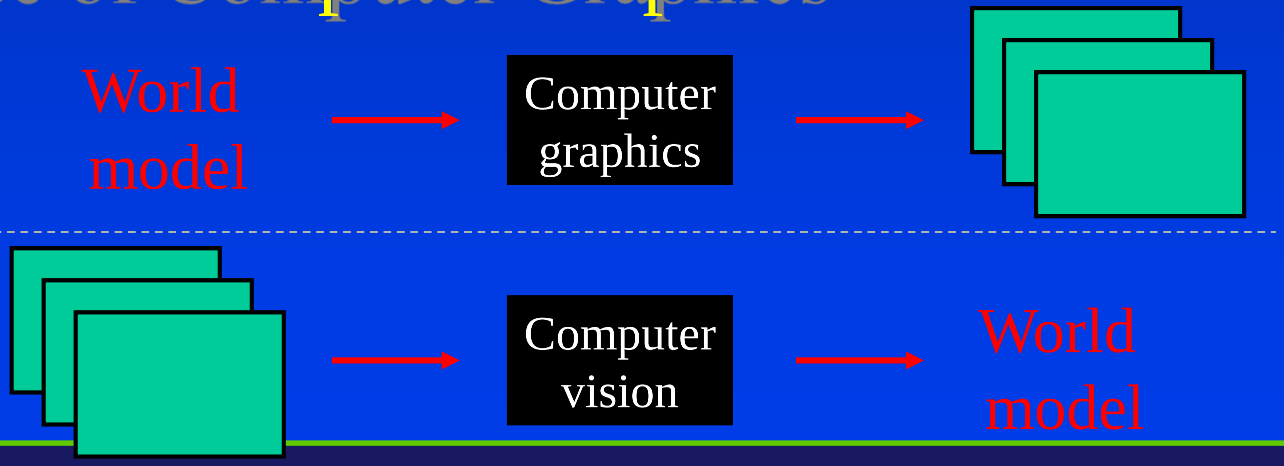
CSE528 Computer Graphics: Theory, Algorithms, and Applications

Hong Qin
Rm.151, NEW CS Building
Department of Computer Science
Stony Brook University (State University of New York)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu; or qin@cs.sunysb.edu;
http://www.cs.sunysb.edu/~qin

Image Based Modeling and Rendering

Computer Vision

- Also known as: Image Understanding (AI, behavior)
- Computer emulation of human vision
- A sensor modality for robotics
- Inverse of Computer Graphics



Graphics and Vision

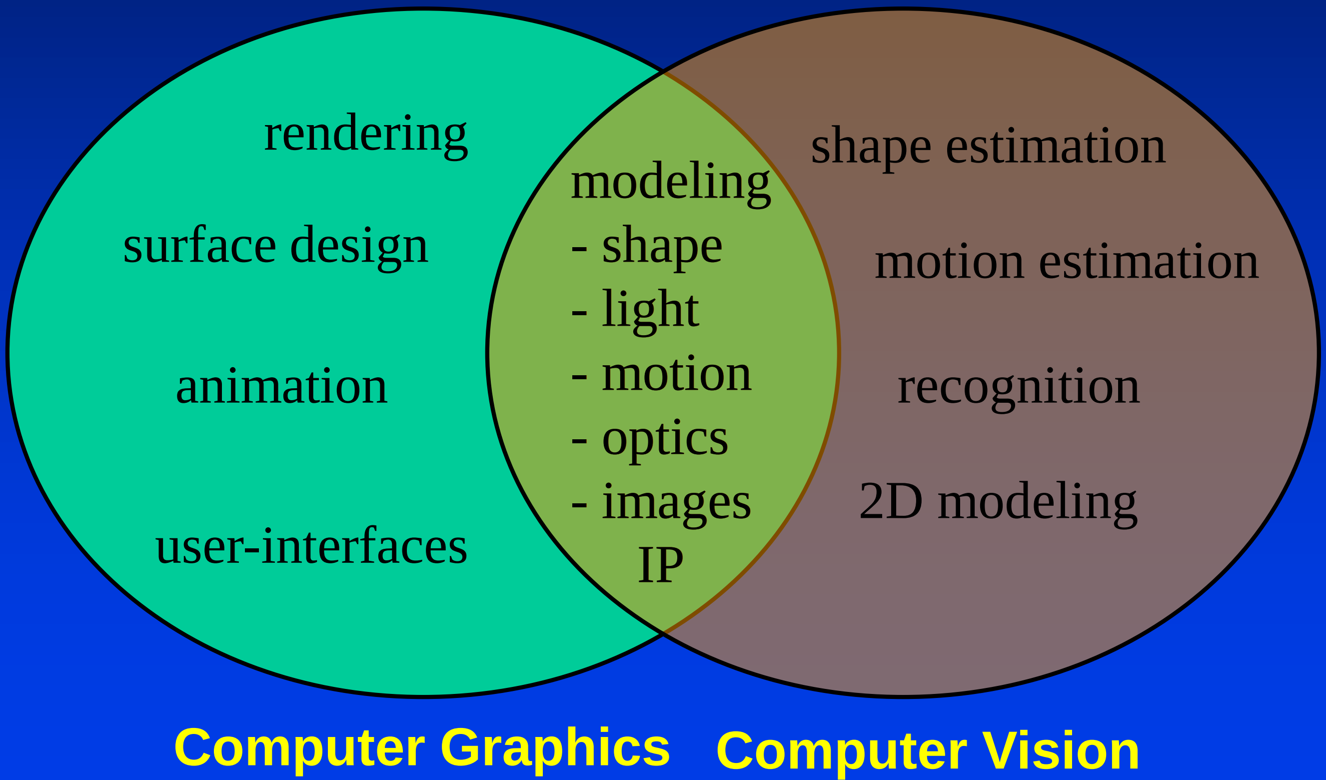
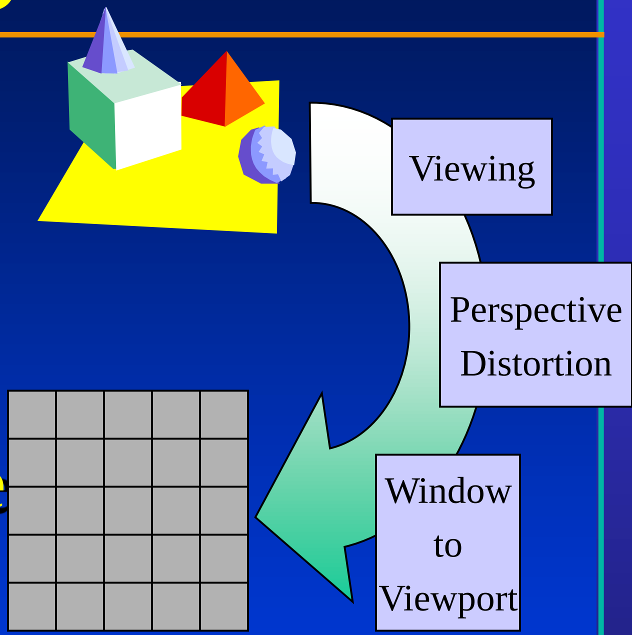


Image Based Rendering

- Traditional Graphics
  - Geometric modeling hard
  - Global illumination slow
- If we have a picture of an object from one point of view, can we simulate the object from another point of view
- Geometry and shading replaced by image(s)
- Need to generalize images from specific points of view to support an arbitrary point of view



$$\begin{bmatrix} x'w \\ y'w \\ 0 \\ w \end{bmatrix} = WPV \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$

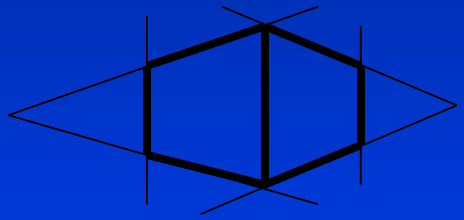
Projective Maps

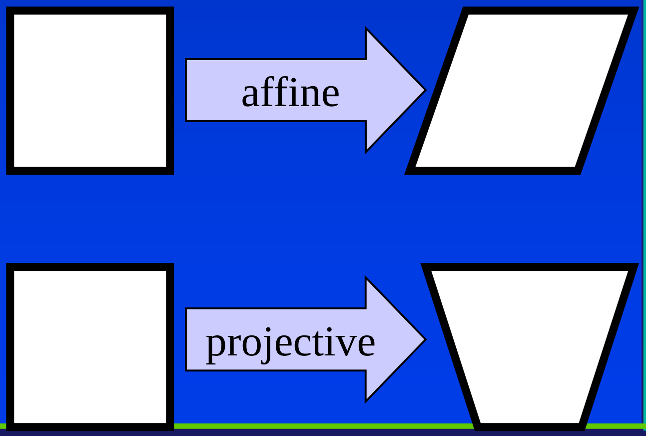
$$\begin{bmatrix} x'w \\ y'w \\ w \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 2-D homogeneous transformation
- Includes affine maps
  - scale (stretch, squash)
  - rotate
  - translate
  - shear
- Non-affine projections
  - different lines converge to point
  - different perspectives

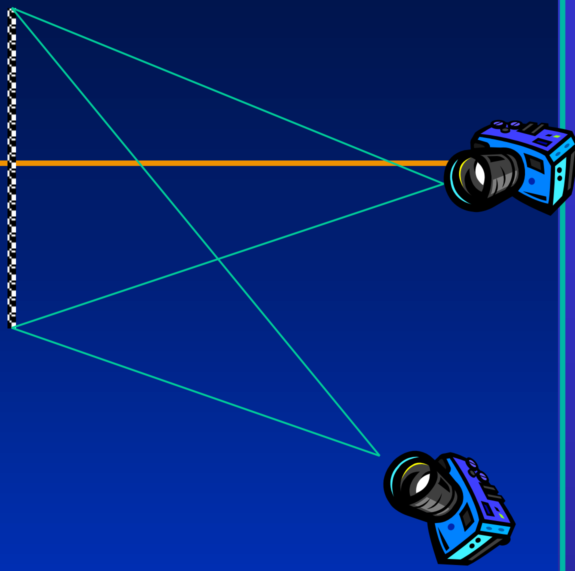
$$x' = \frac{m_1 x + m_2 y + m_3}{m_7 x + m_8 y + m_9}$$

$$y' = \frac{m_4 x + m_5 y + m_6}{m_7 x + m_8 y + m_9}$$



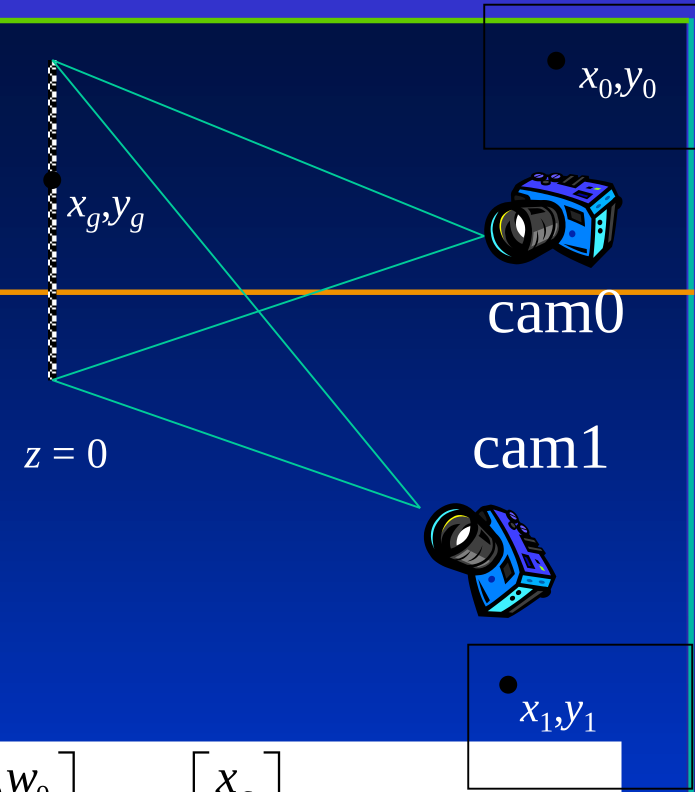


2-D Warping



- Warp is a generalization of a transformation
  - non-affine
  - includes perspective
- Difference between two views of a textured plane from a pinhole camera
  - Use a previous image to generate a new view
  - Avoid re-rendering
- Difference between two views from a stationary camera rotating and zooming
  - Create panoramic images
  - Environment maps

Two Views of a Plane



- Question: where is pixel (x,y) from image 0 located in image 1?
- Assume image 0 a picture of the x-y coordinate plane at $z = 0$
  - Plane can be placed arbitrarily, but setting $z = 0$ allows us to ignore z coordinate
  - Can find affine transformation that takes arbitrary plane to $z = 0$
- $M_0 = WPV$, takes point on plane (x_g,y_g) to pixel (x_0,y_0)
  - Invertible
  - distorts, doesn't project

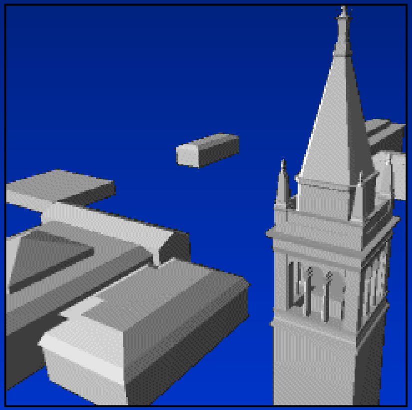
$$\begin{bmatrix} x_0 w_0 \\ y_0 w_0 \\ w_0 \end{bmatrix} = M_0 \begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 w_1 \\ y_1 w_1 \\ w_1 \end{bmatrix} = M_1 \begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix} = M_1 M_0^{-1} \begin{bmatrix} x_0 w_0 \\ y_0 w_0 \\ w_0 \end{bmatrix}$$

$$\begin{bmatrix} x_1 w \\ y_1 w \\ w \end{bmatrix} = M_1 M_0^{-1} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

Image-Based Texturing

- Take several pictures of the same object
  – Set a large variety of viewpoints
  – Pictures should cover the entire object
- Create a rough polygonal model of object
  – Planar sections
  – No need to model details
- Render final textured model
  – Warp photographs onto sections
  – Texture appears as model detail
  – Composite multiple textures
- Fails at silhouettes, parallax



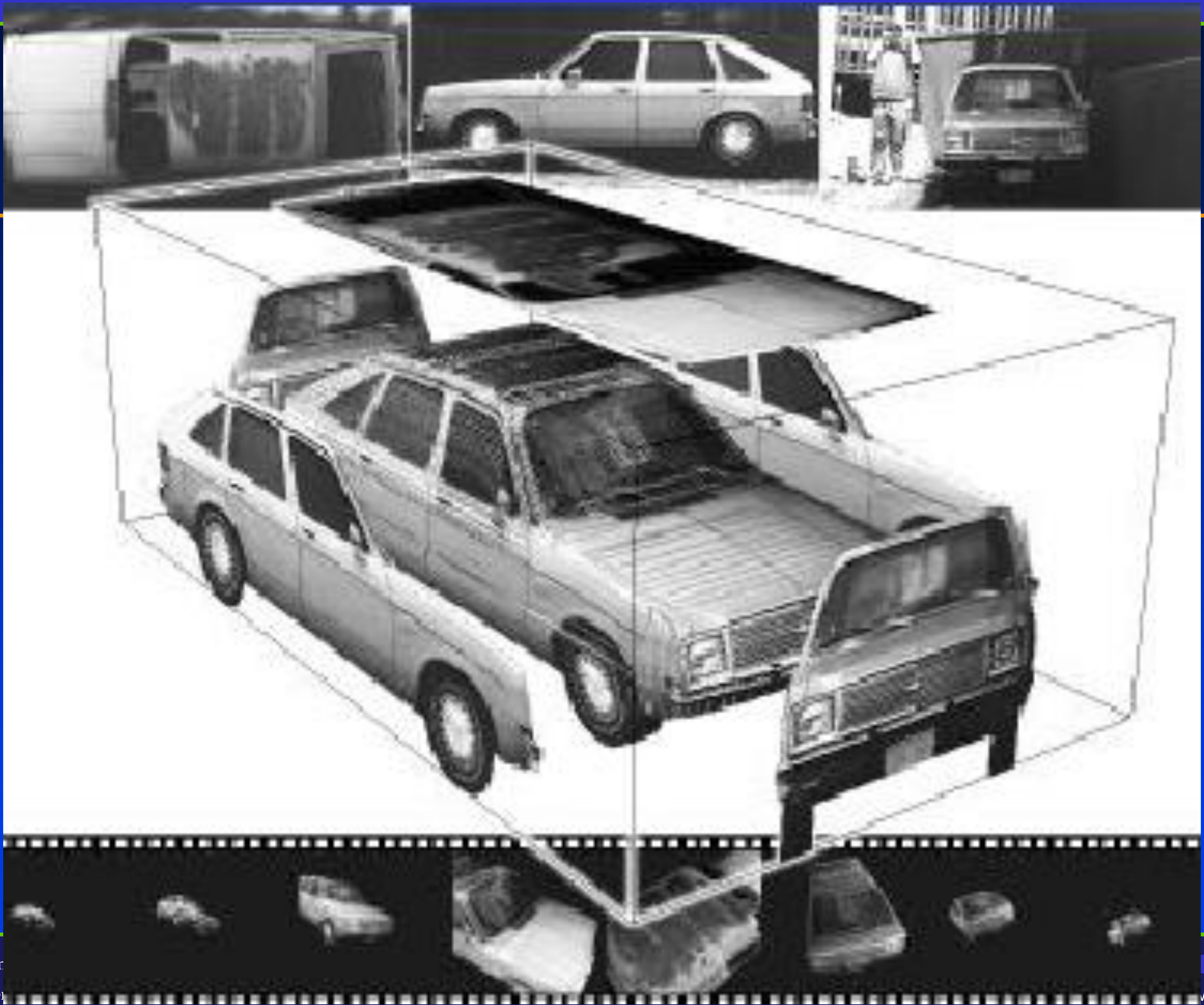
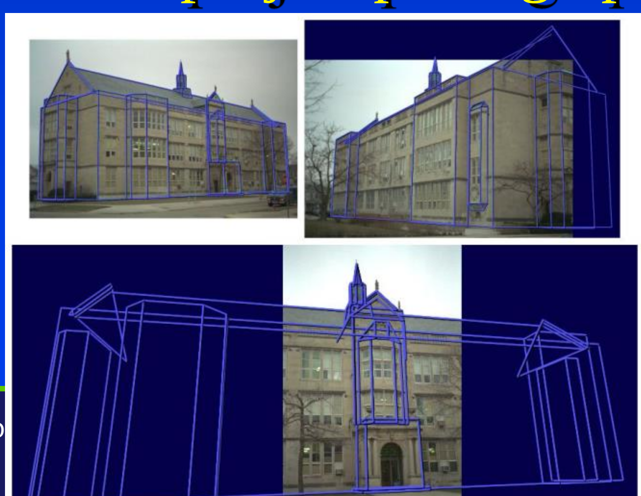
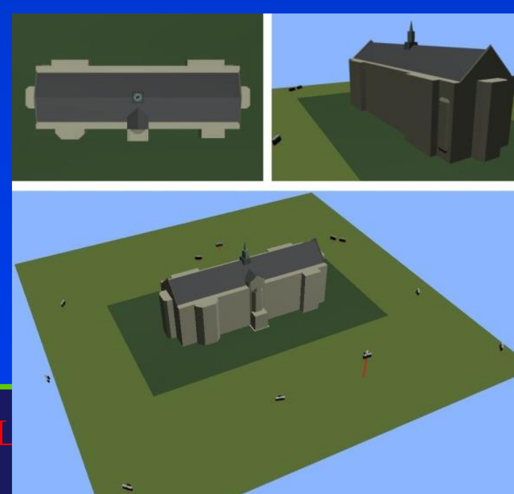


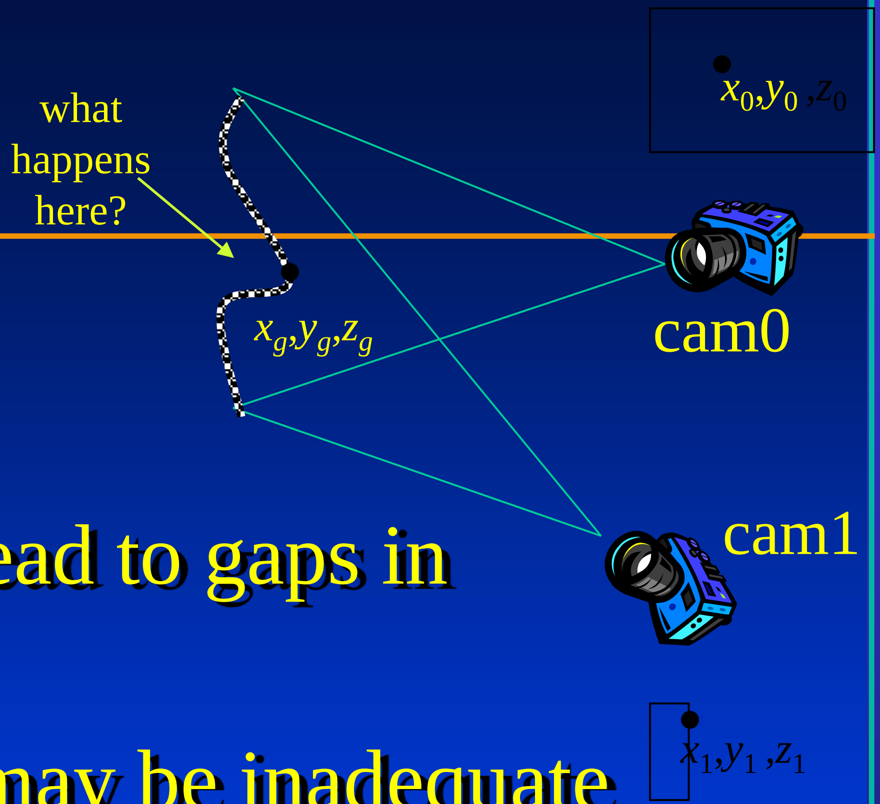
Image-Based Modeling

- Take several pictures of a shape
- Find features in each picture
- Correlate features between pictures
- Solve for world-coordinate camera position
- Use features to create a crude model
- Re-project photographs on crude model





Adding Depth



- Math. derivation is easy
- Implementation is hard
- Depth interpolation can lead to gaps in shape
- Sampling for one image may be inadequate for warped images

$$\begin{bmatrix} x_0 w_0 \\ y_0 w_0 \\ z_0 w_0 \\ w_0 \end{bmatrix} = M_0 \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 w \\ y_1 w \\ z_1 w \end{bmatrix} = M_1 M_0^{-1} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

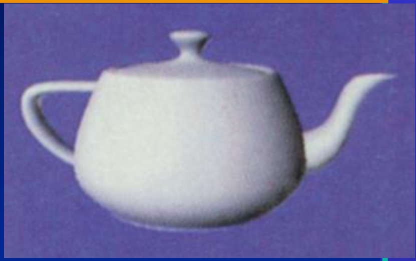
View Interpolation

- Williams & Chen, S93
- Computes images between two images
- Used to animate walk-through
- Linear interpolation of pixel positions and depth values
- Depth values determine occlusion
- Holes filled with "local color"
- Linear interpolation correct for plane if view direction perpendicular to plane
- Correct if the camera trucks (and zooms) parallel to the plane
- Higher-order interpolation more accurate

$$\mathbf{p}_i = V_i \mathbf{p}_g$$

$$V(s) = (1-s)V_0 + sV_1$$

$$\mathbf{p}(s) = (1-s)\mathbf{p}_0 + s\mathbf{p}_1 = V(s)\mathbf{p}_g$$



View Morphing

- Seitz & Dyer, S96
- Handles case of moving camera as well as moving scene
- View interpolation works when “scene planes” are parallel for cam0 and cam1
- View interpolation incorrect if the scene plane has rotated for cam0 and cam1
- Solution: prewarp cam0 and cam1 to be parallel, then interpolate

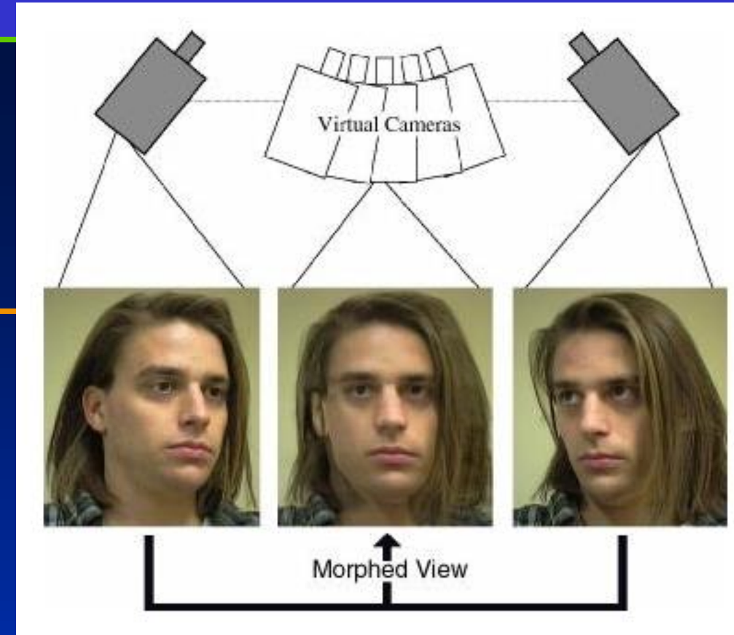


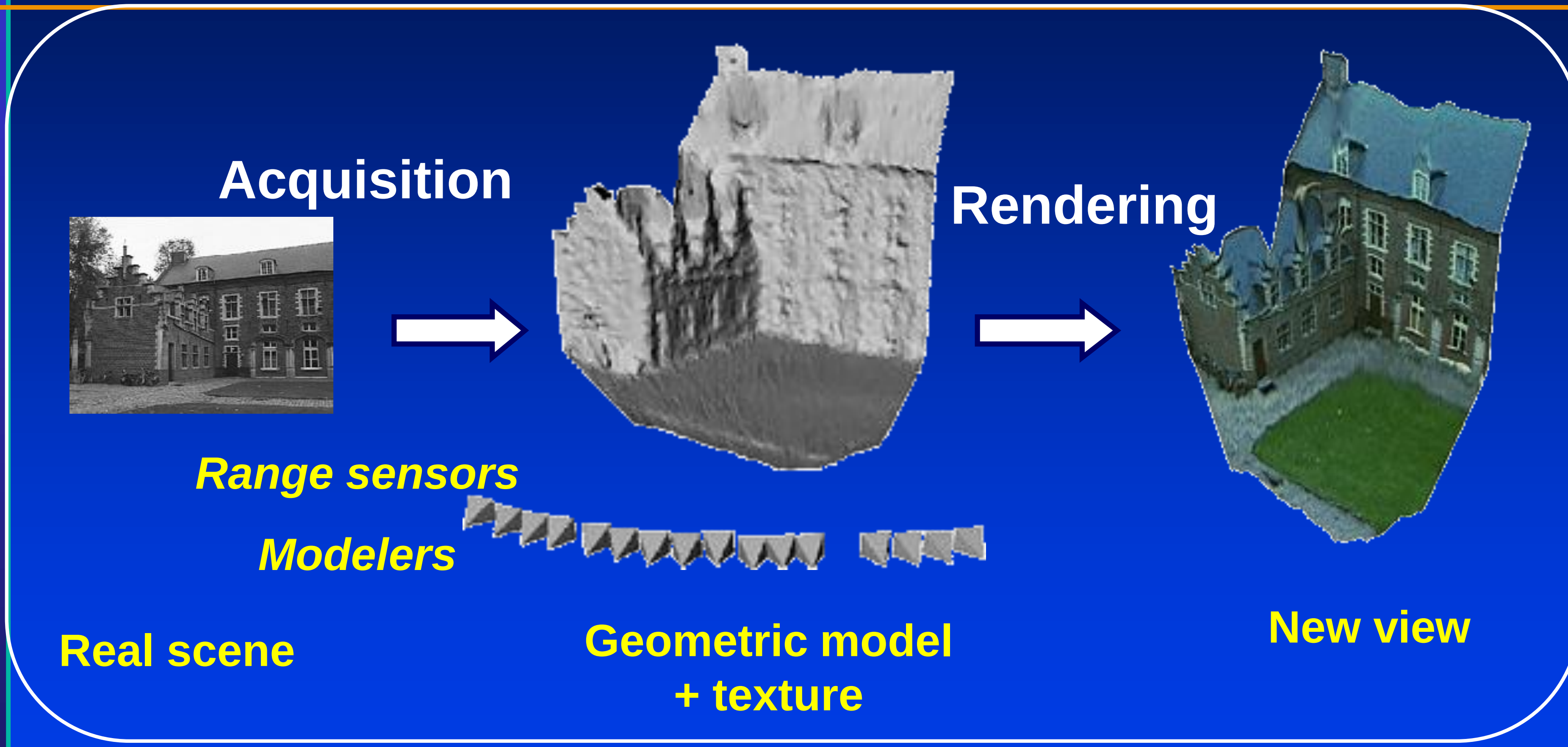
Image-Based Rendering

- What is image-based rendering?
  - *The synthesis of new views of a scene from pre-recorded pictures*
- Why image-based rendering?
  - Many applications

Example: Panoramic Mosaics

-  +  + … +  =

3D Modeling in Computer Graphics



- Graphics model: 3D detailed geometric model of a scene
  Goal: rendering new views

Image-based Rendering

- How? General pipeline:

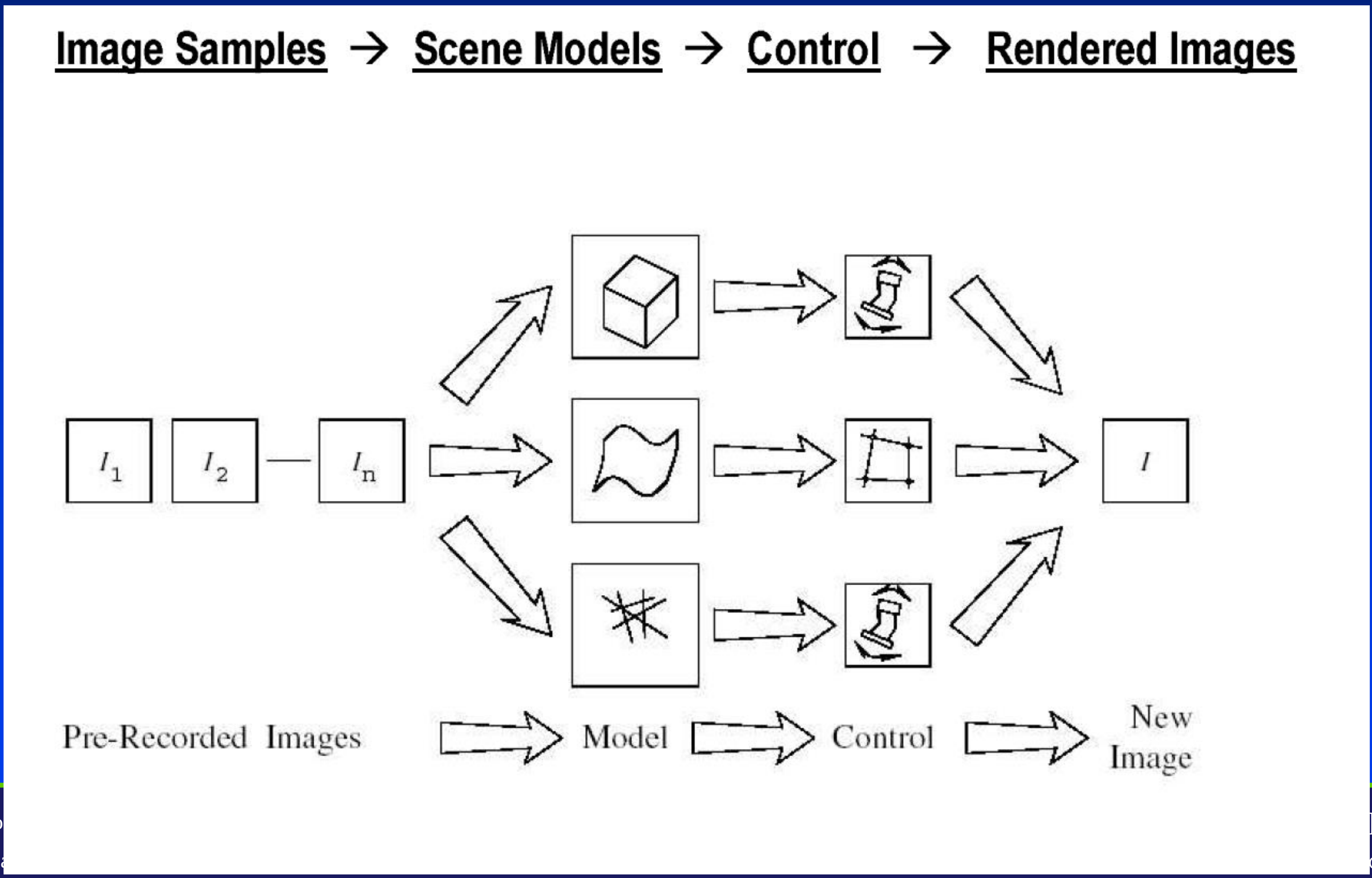


Image-based Rendering

Three approaches:

1. 3D model construction from image sequences
2. Transfer-based image synthesis
3. Light field

3D Model Construction from Image Sequences

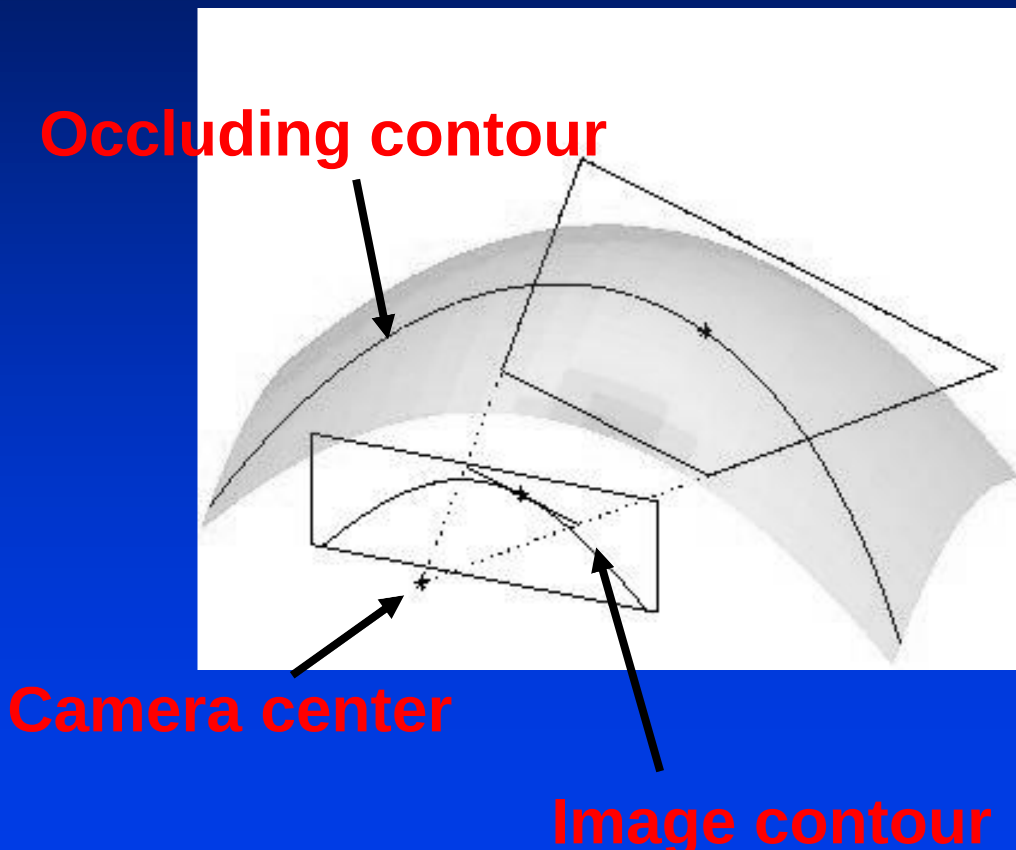
- Techniques that first recover a three dimensional scene model from a sequence of pictures, then render it with traditional computer graphics tools

- Scene modeling from:
  1. Registered images
  2. Unregistered images

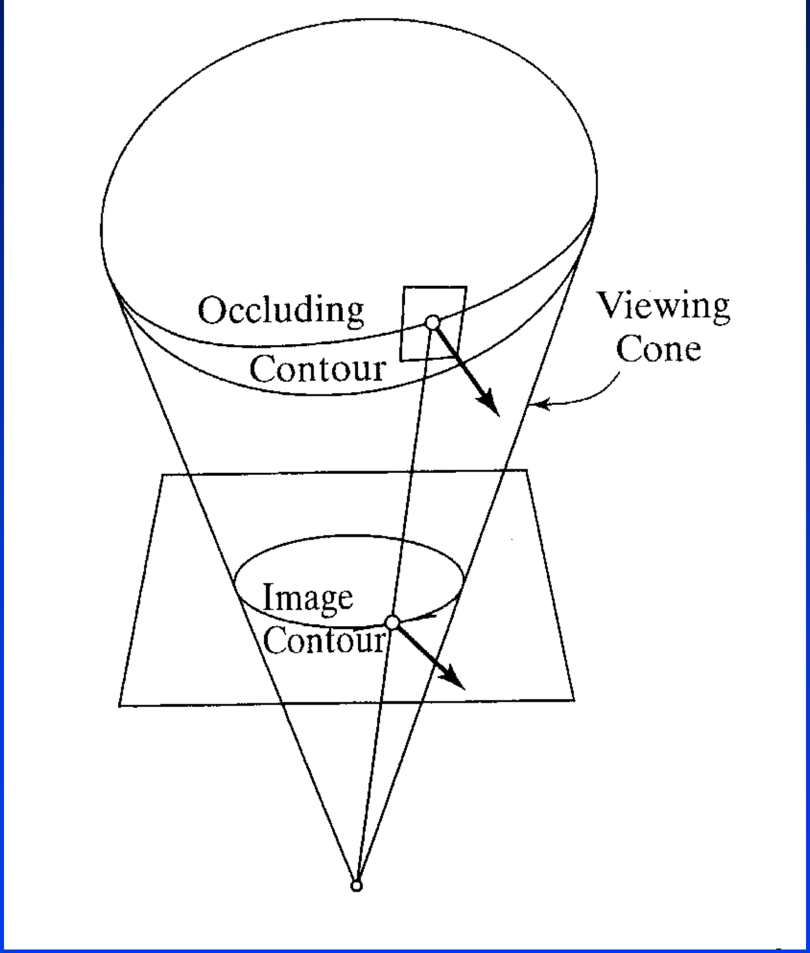
Scene Modeling from Registered Images

- All images are registered in the same global coordinate system
- What kinds of reconstruction?
  1. Volumetric reconstruction
  2. Surface reconstruction
  3. Depth maps
  4. ….

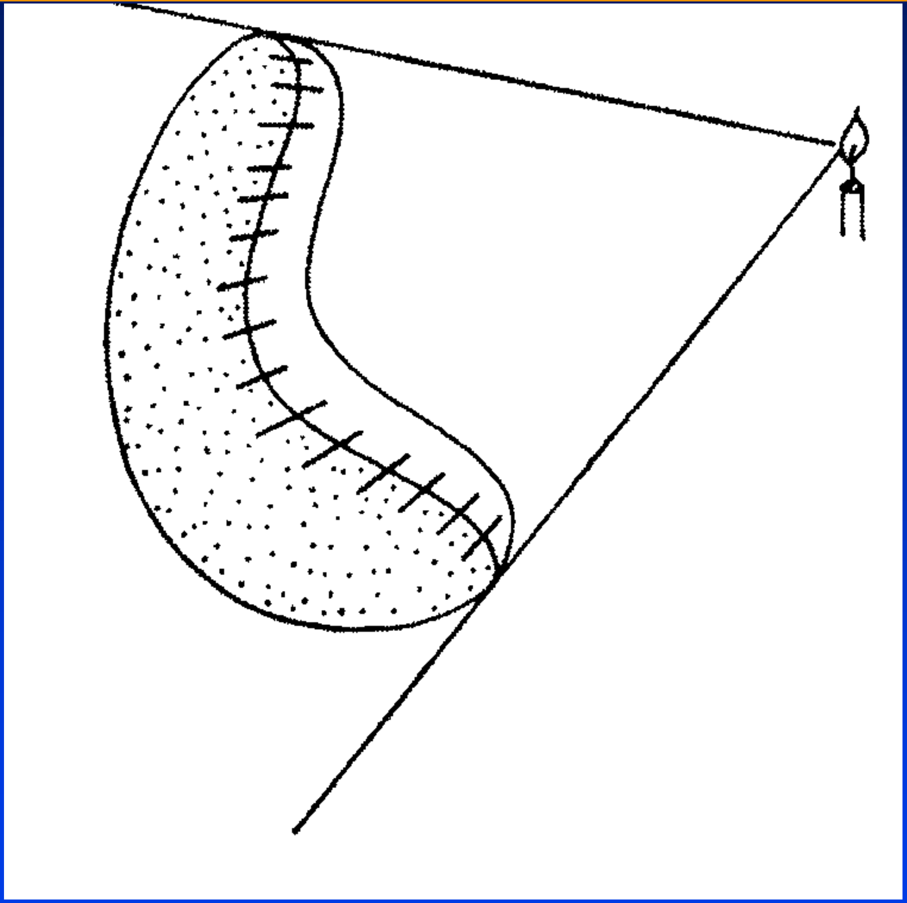
Surfaces and Their Outlines



Surfaces and Their Outlines



The viewing cone



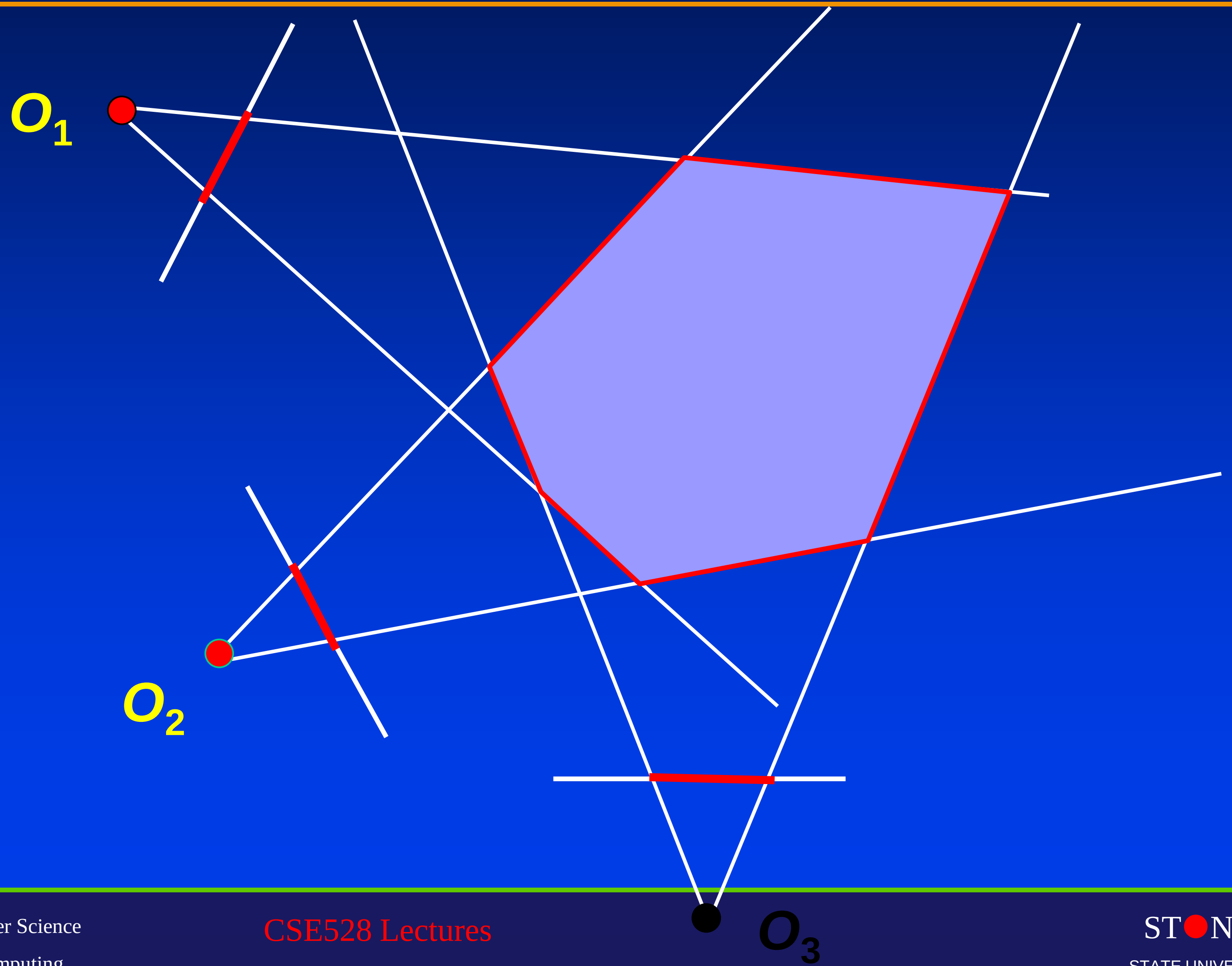
Shadow boundary

Visual Hull

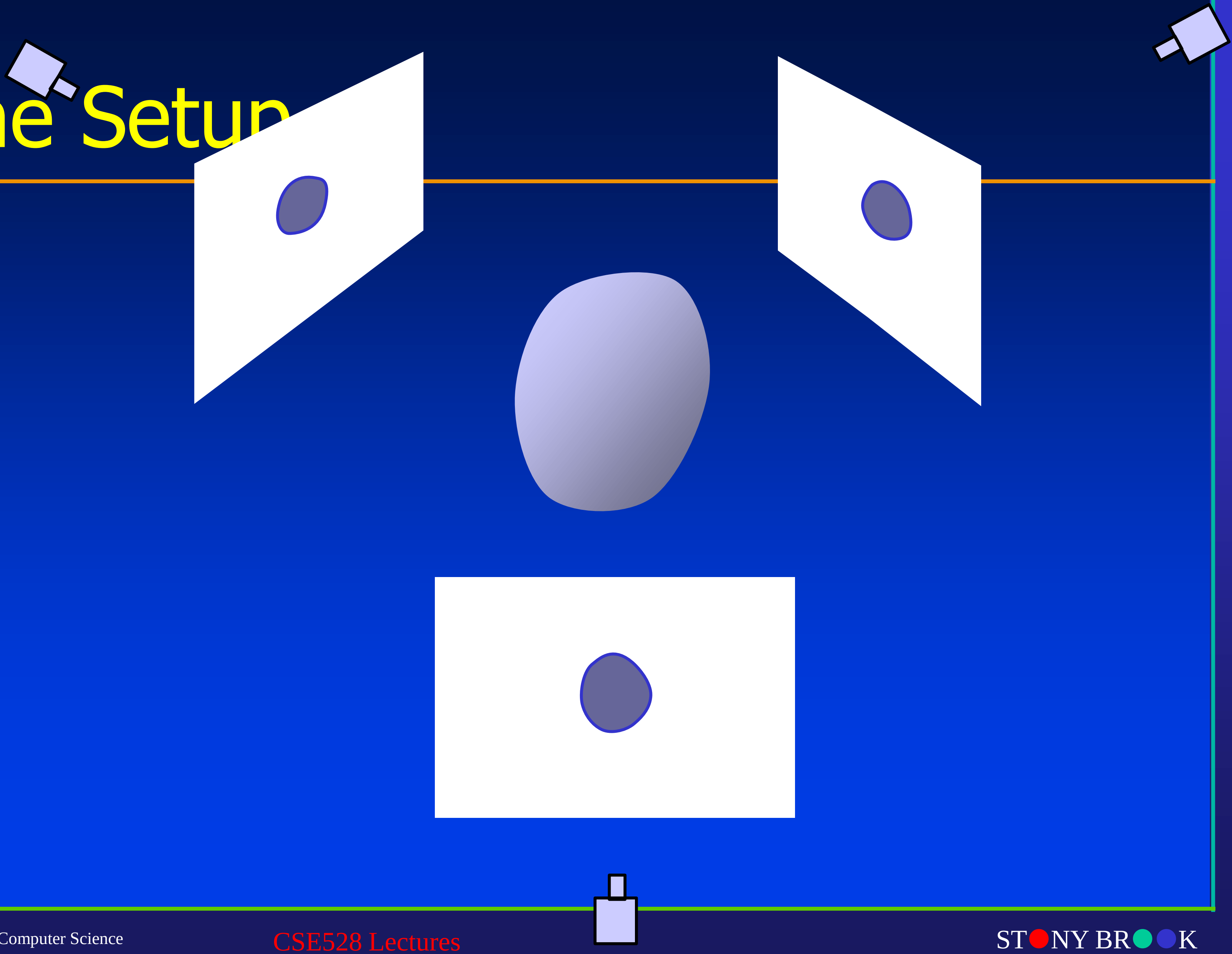
Visual Hull?

- The ***visual hull*** of an object with respect to a set of input views is…
  - The maximal shape that yields the same silhouettes as the original object in all the input views
  - The intersection of the solid visual cones formed by back-projecting the silhouettes in all the input views

Visual Hull: A 2D Example

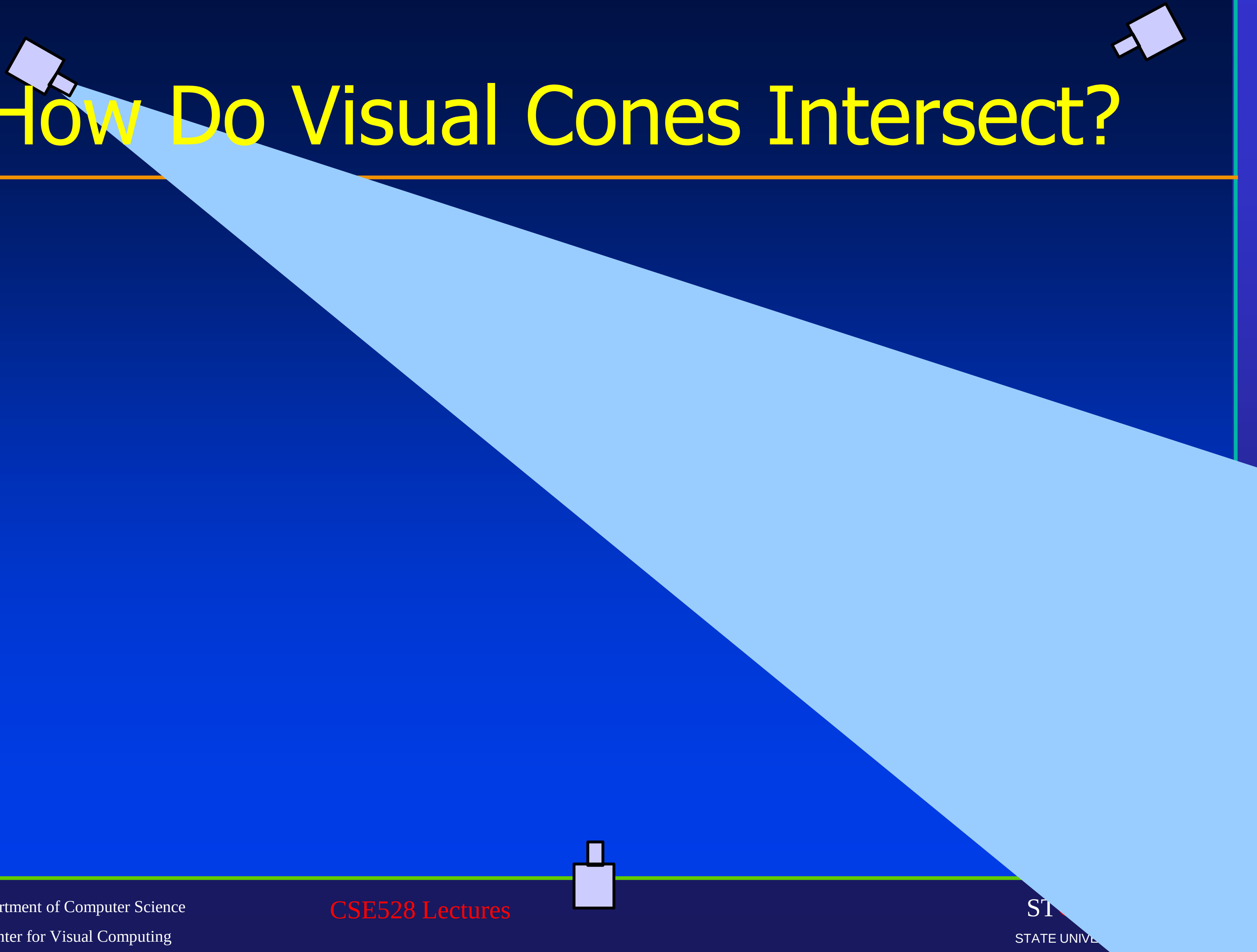


The Setup

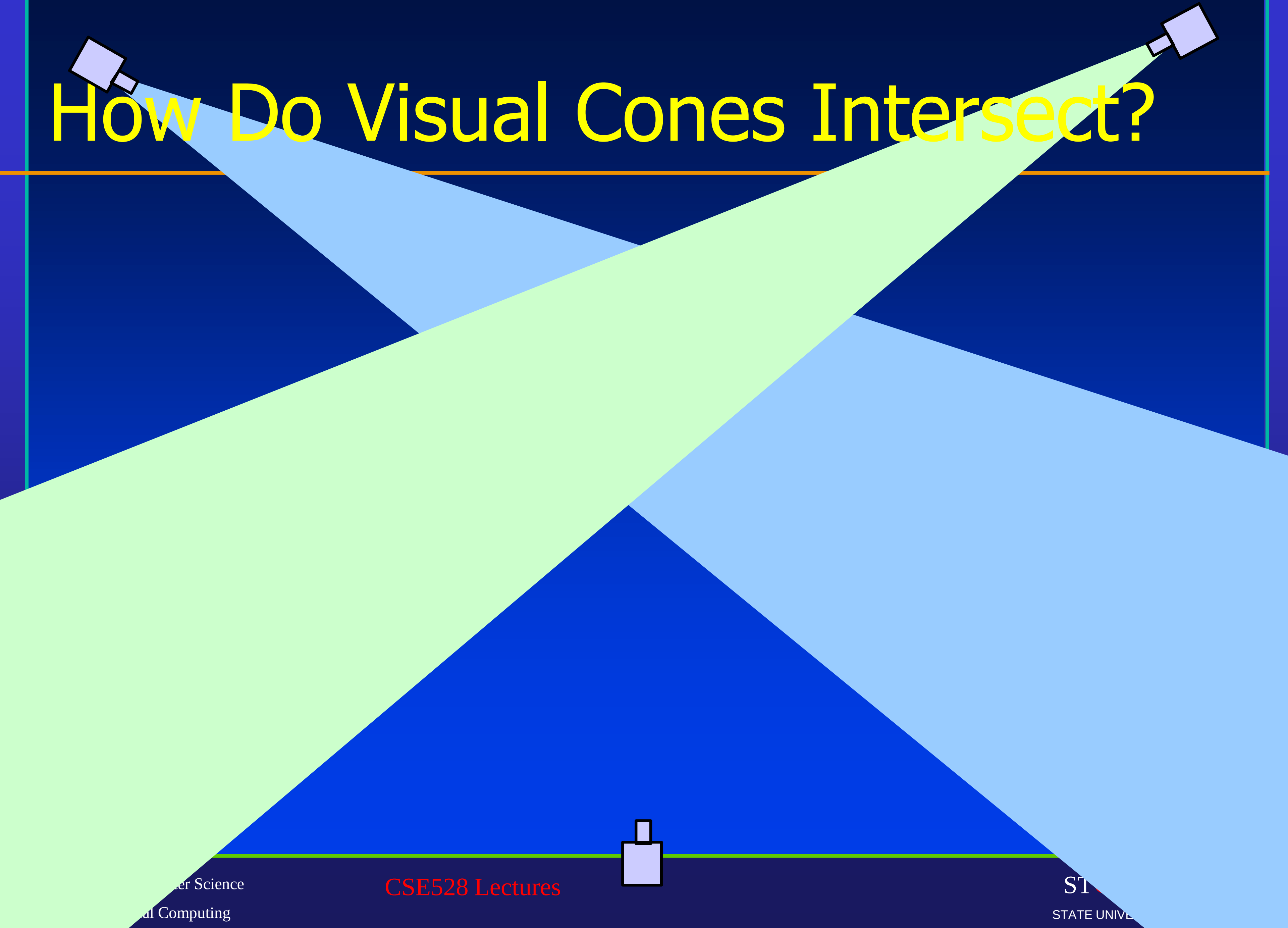


How Do Visual Cones Intersect?

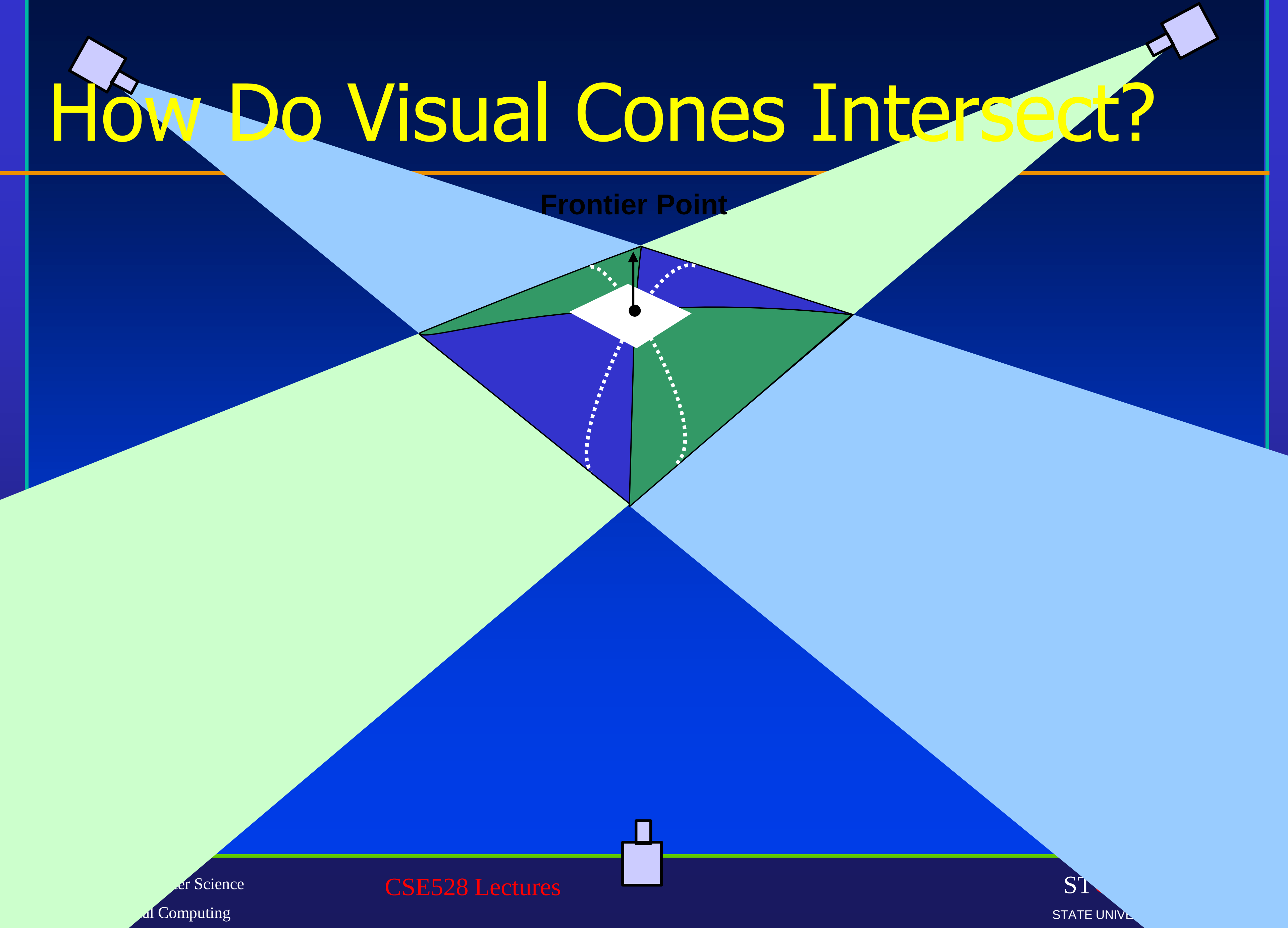
How Do Visual Cones Intersect?



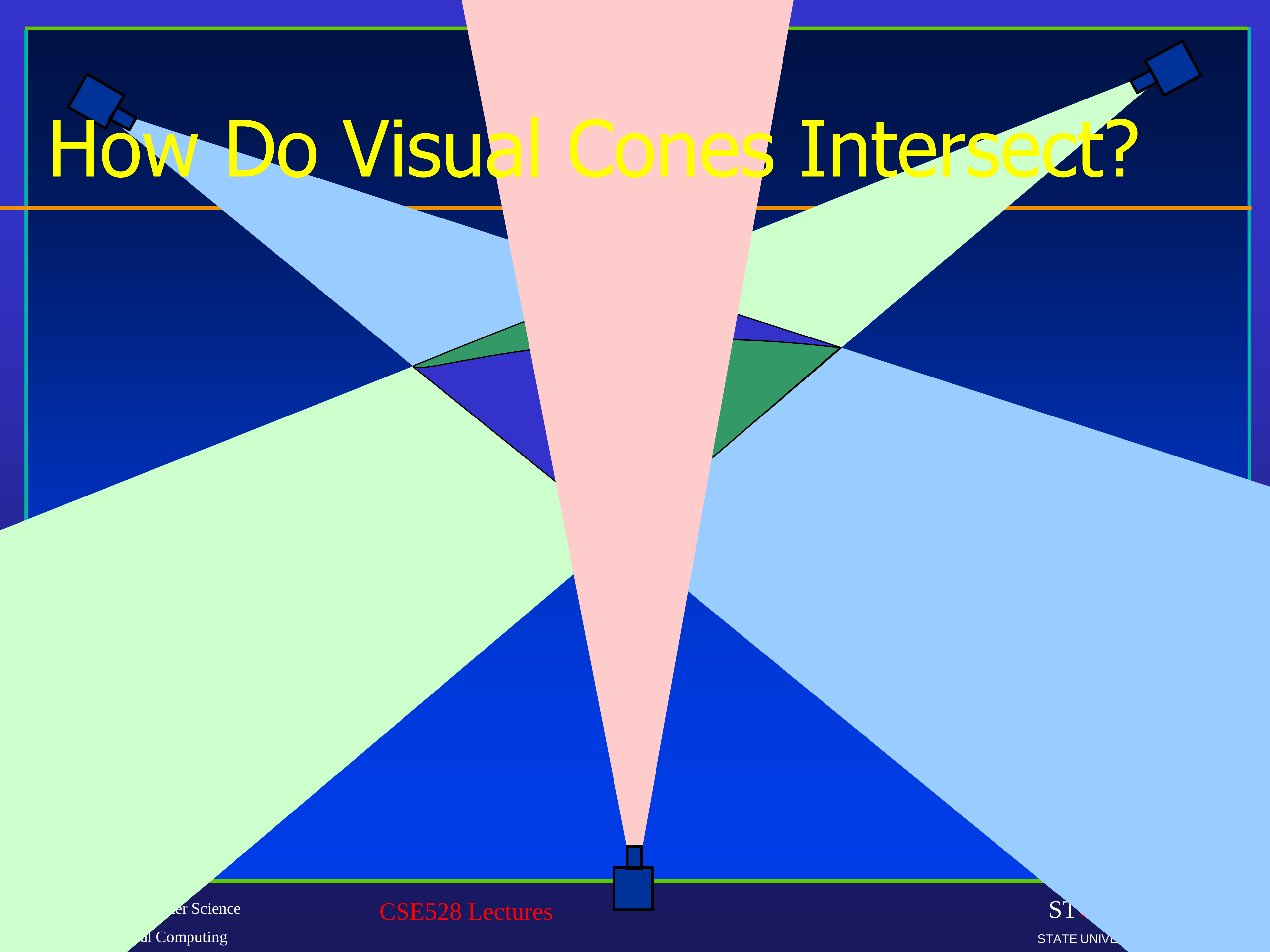
How Do Visual Cones Intersect?



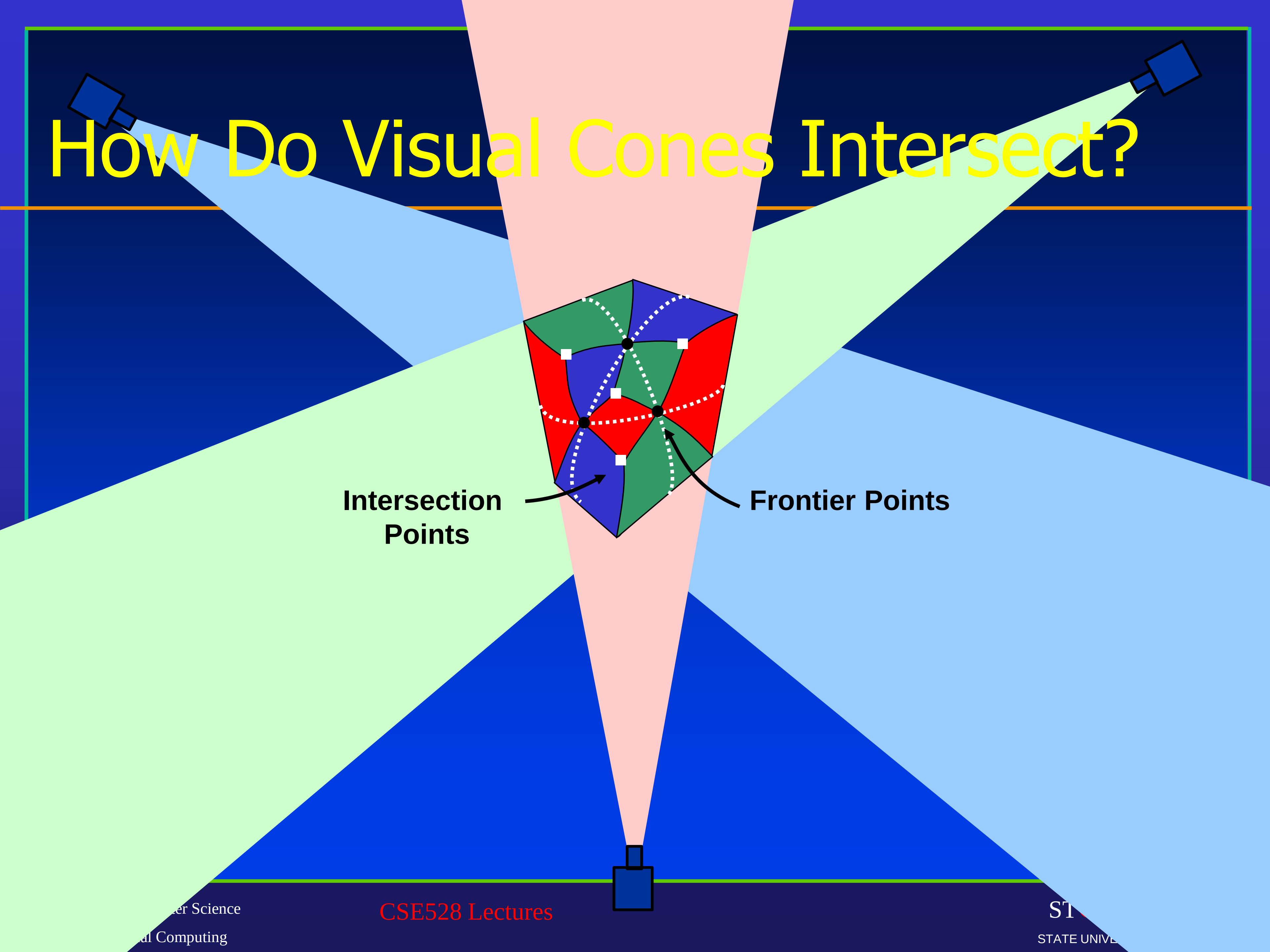
How Do Visual Cones Intersect?



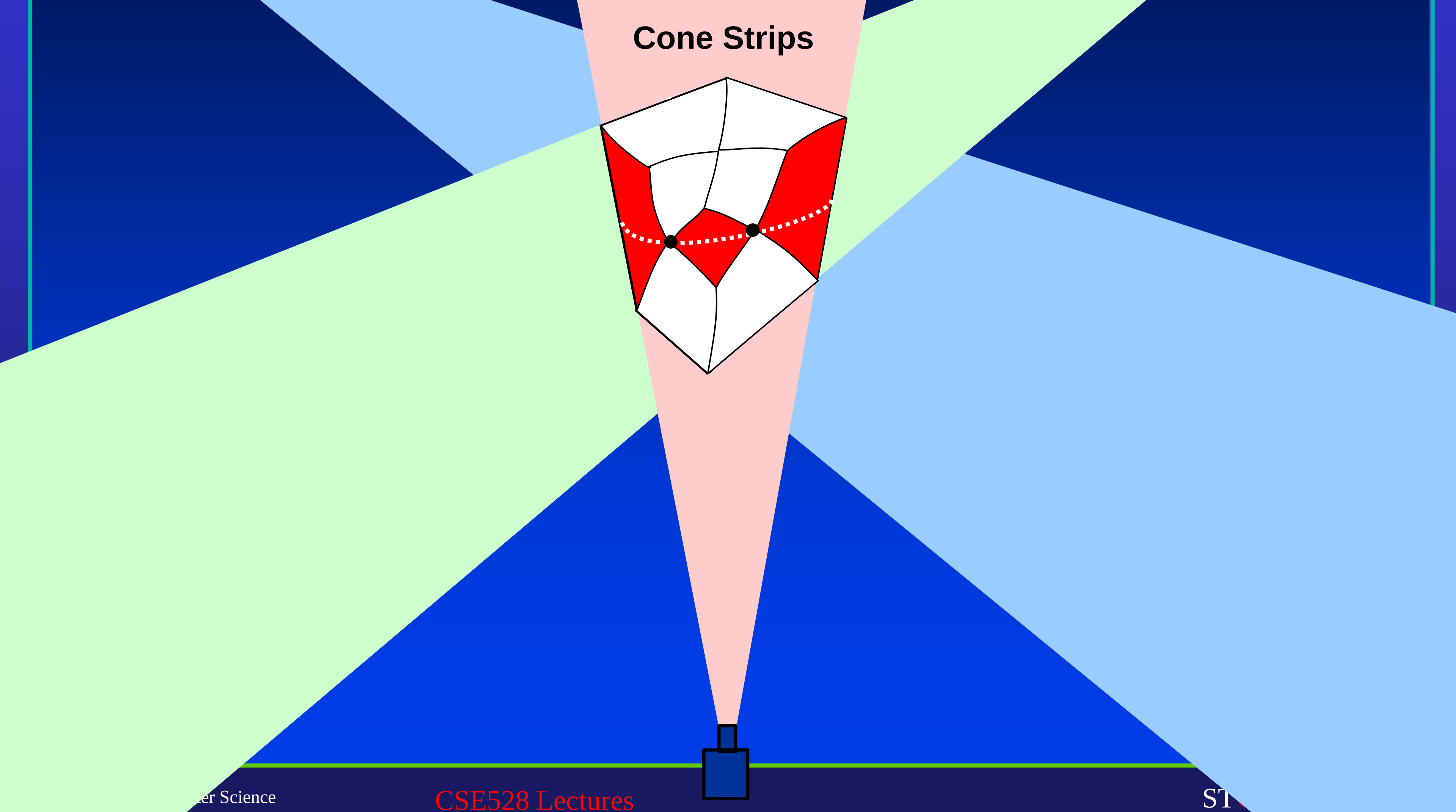
How Do Visual Cones Intersect?



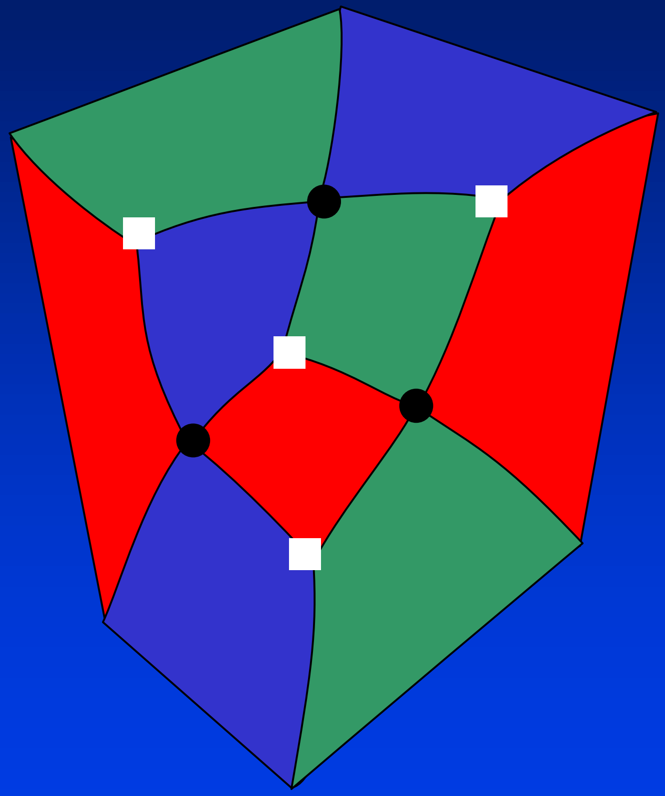
How Do Visual Cones Intersect?



How Do Visual Cones Intersect?



Visual Hull As Topological Polyhedron

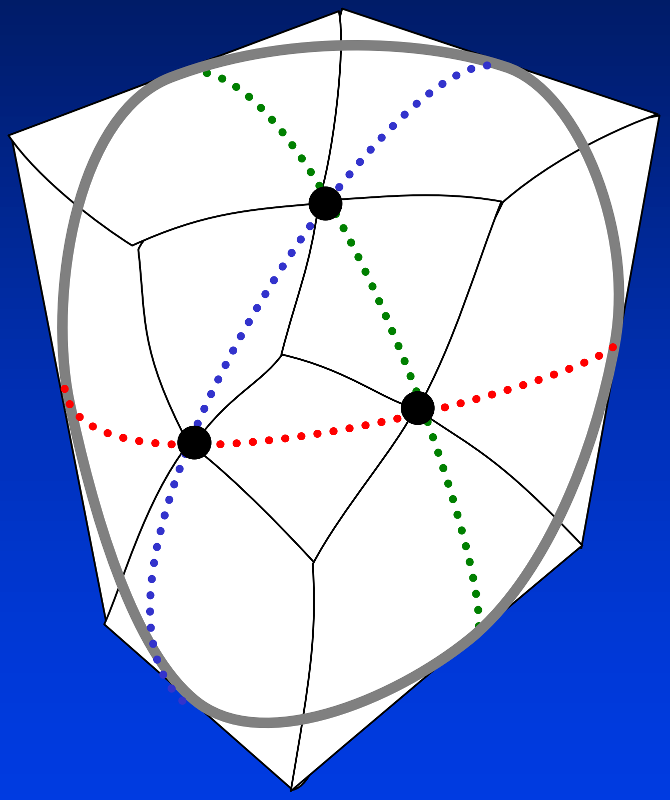


- **Vertices:** frontier points + intersection points
- **Edges:** intersection curve segments
- **Faces:** visual cone patches

Visual Hull Mesh

The Arrangement of Rims on the Surface of the Object

- **Vertices:** frontier points
- **Edges:** rim segments
- **Faces:** regions on the surface of the object

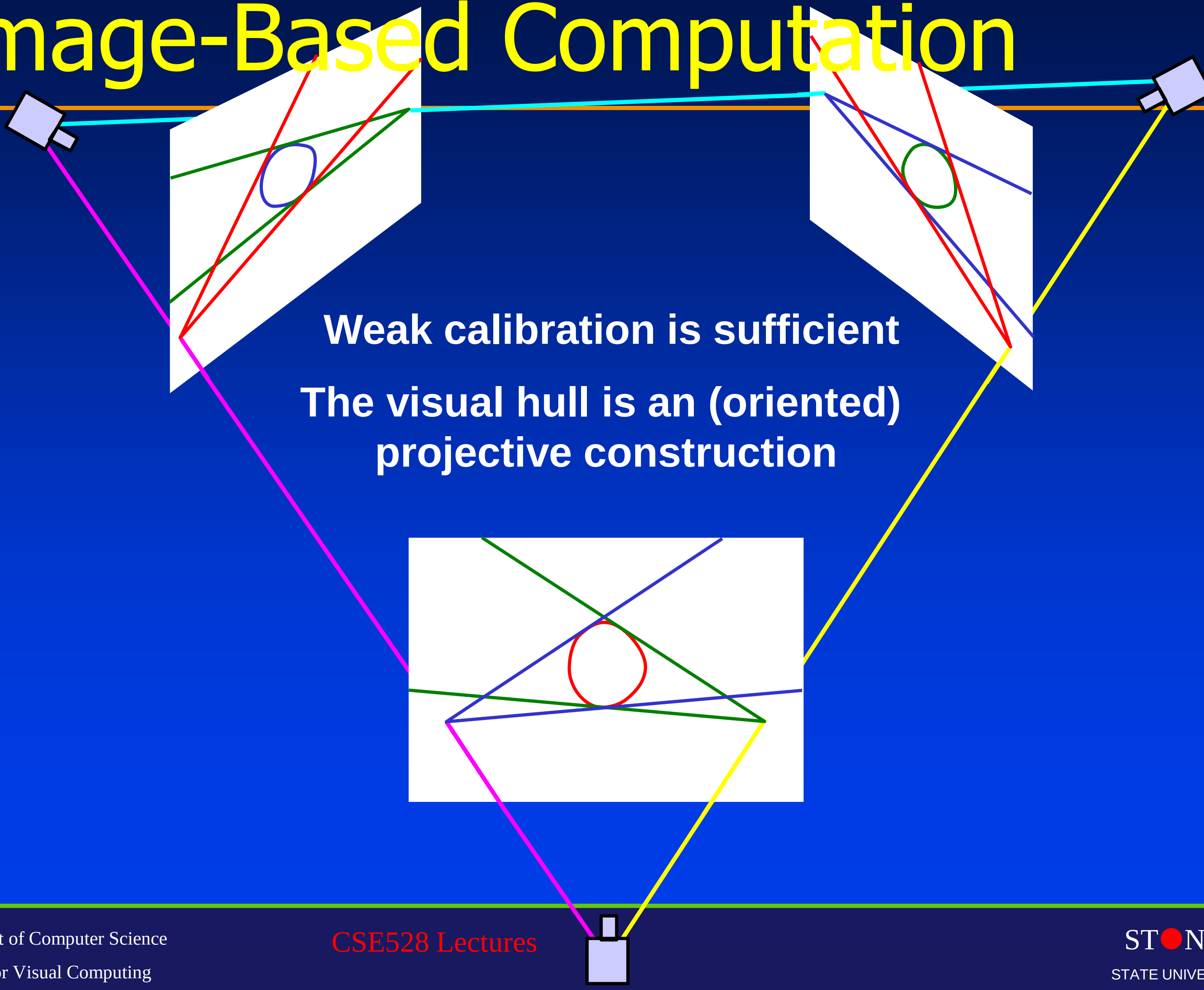


Rim Mesh

Department of Computer Science
Center for Visual Computing

STONY BROOK
STATE UNIVERSITY OF NEW YORK

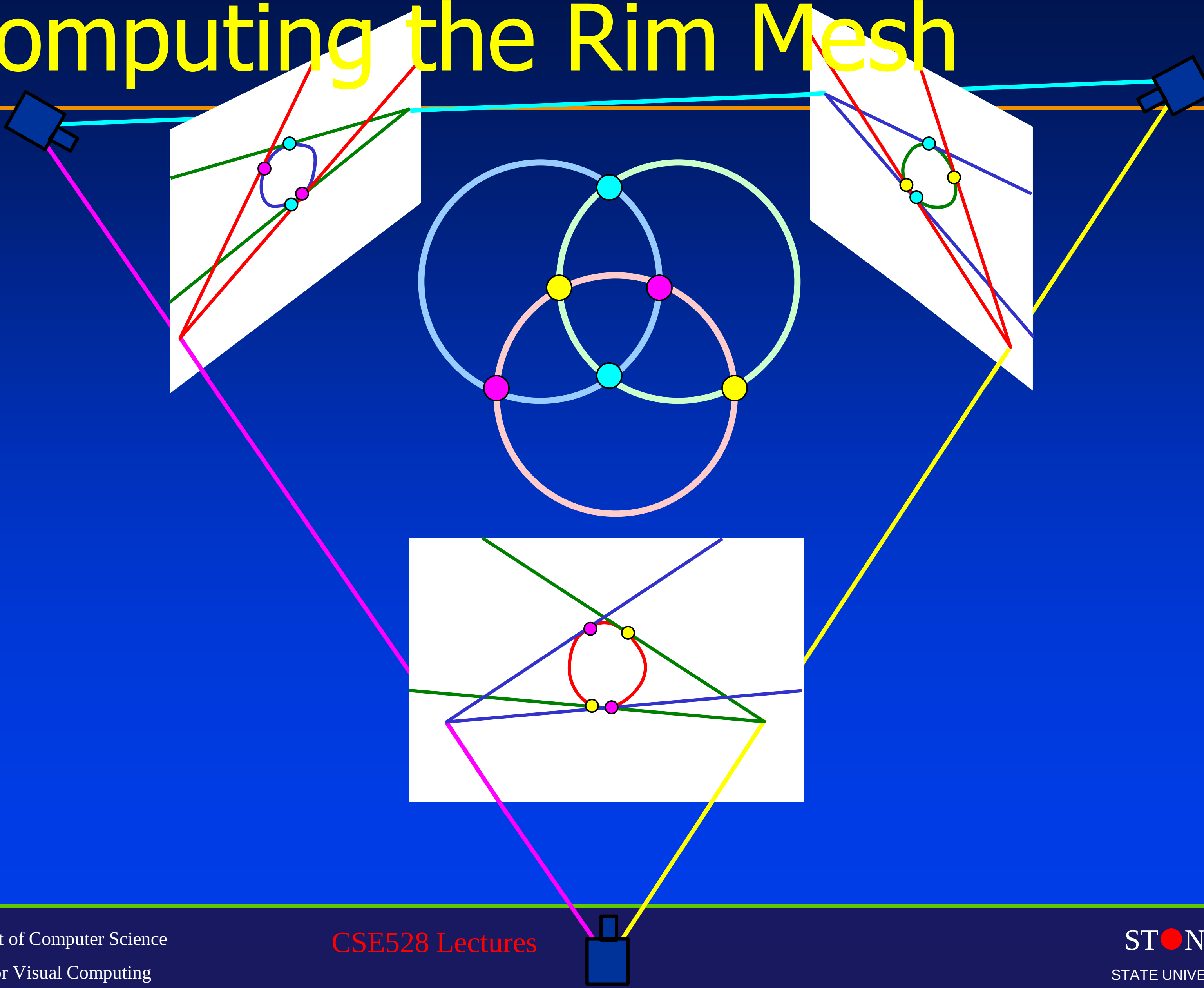
Image-Based Computation



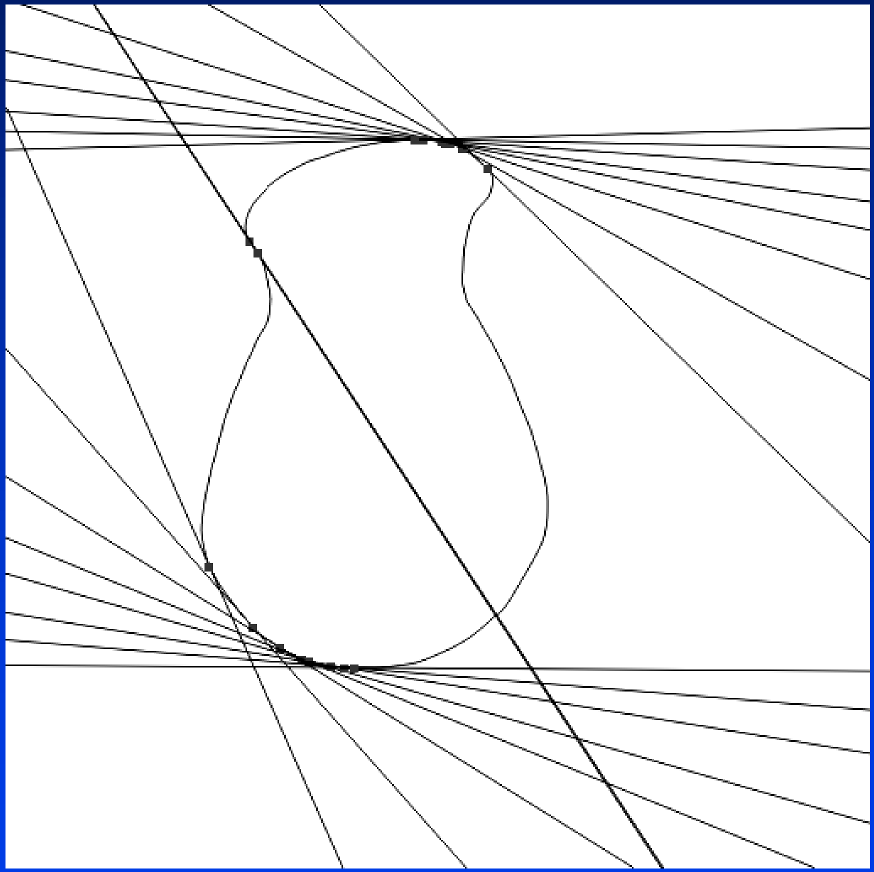
Weak calibration is sufficient

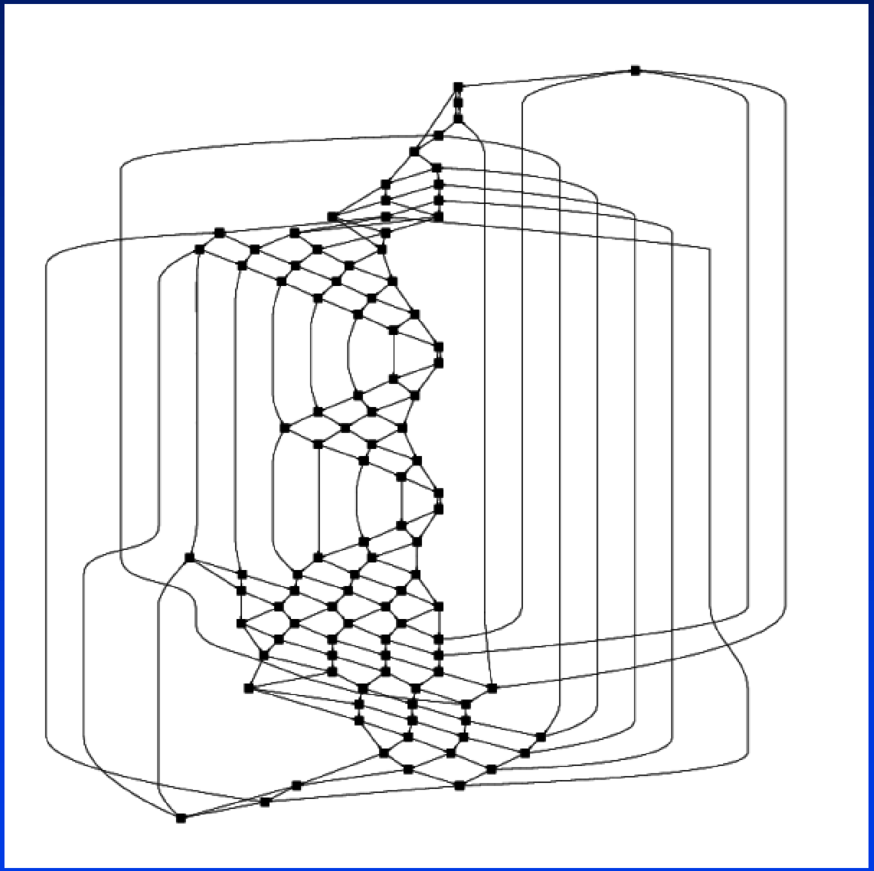
The visual hull is an (oriented) projective construction

Computing the Rim Mesh



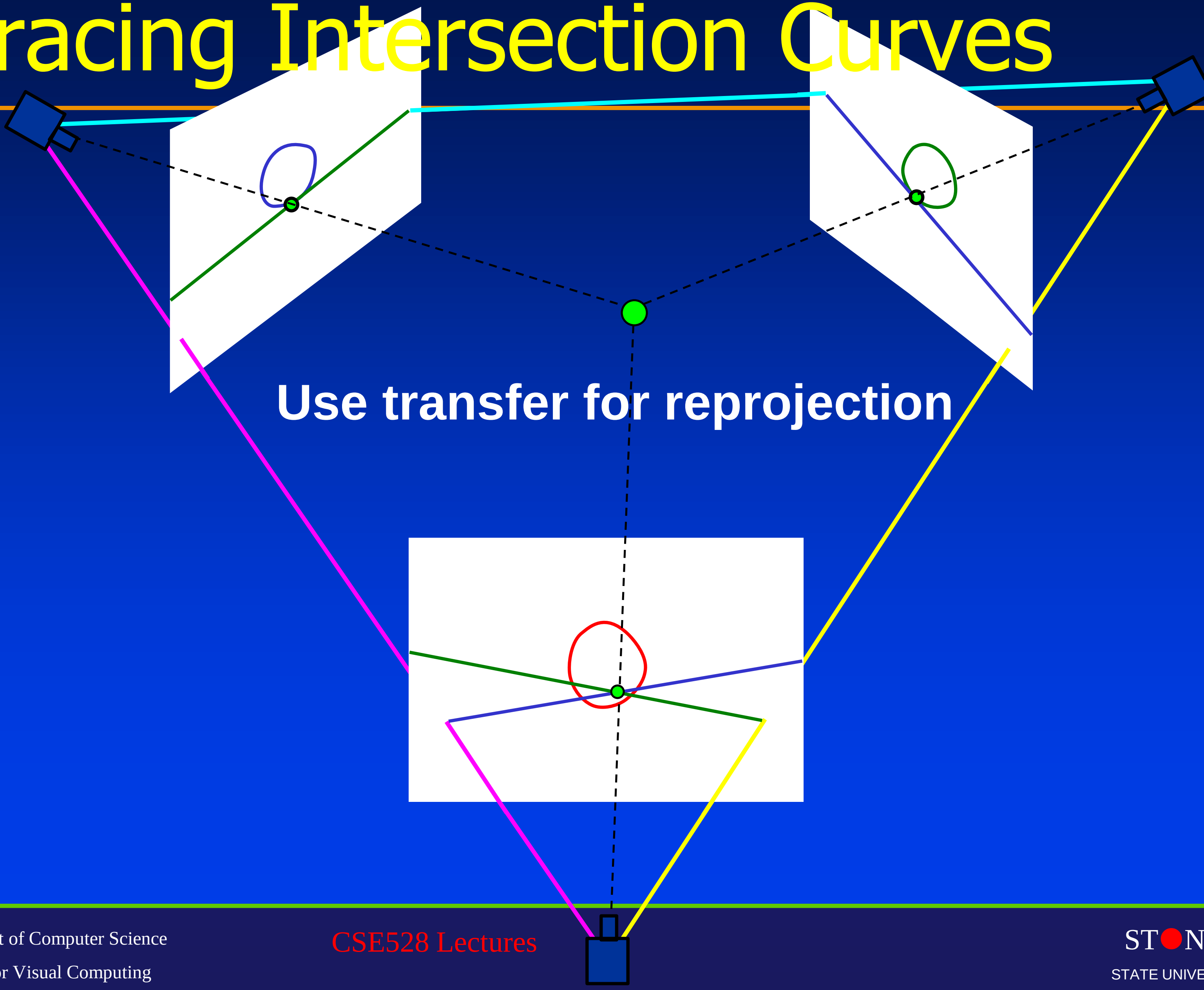
Rim Mesh Example





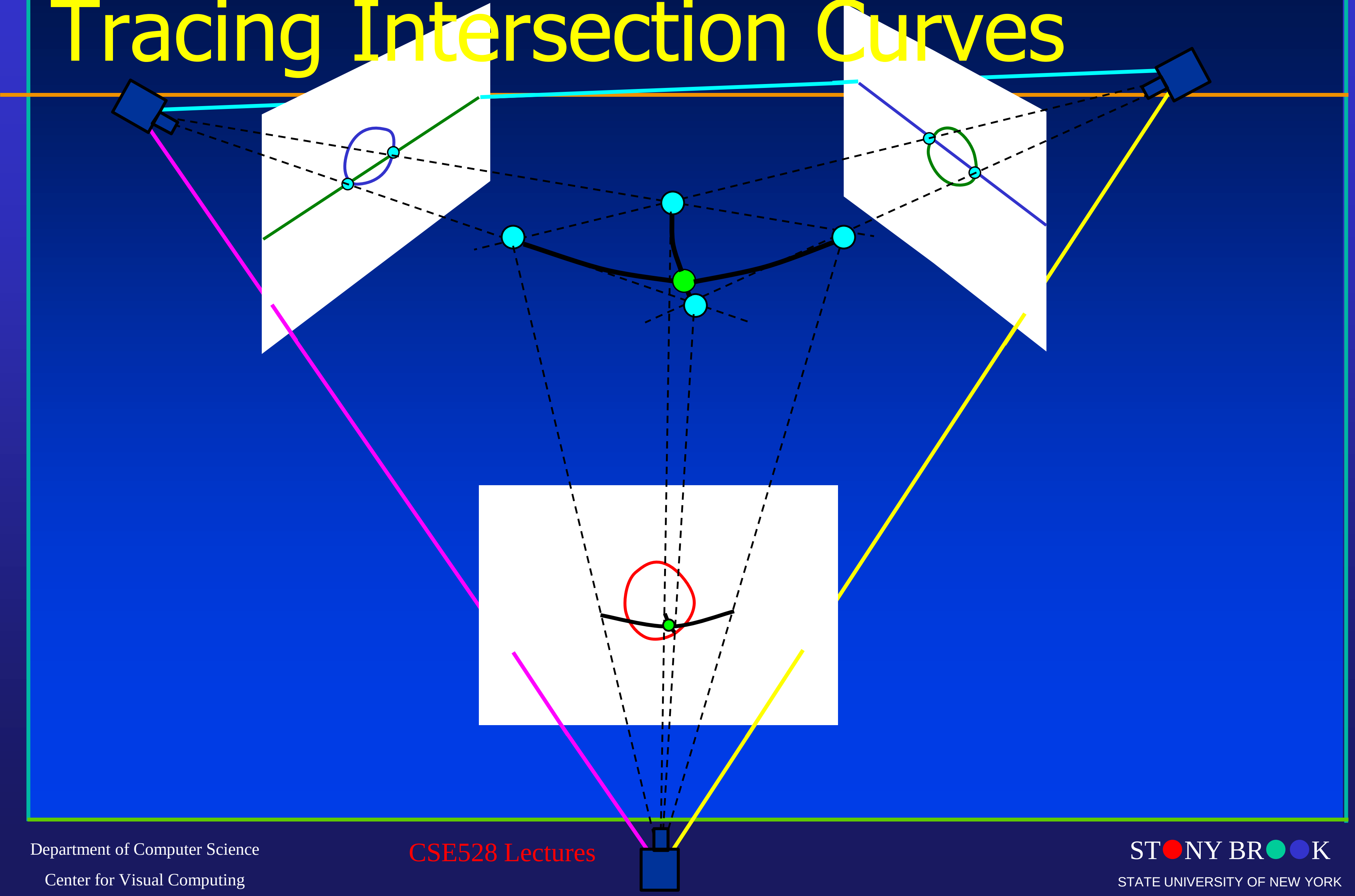
104 frontier points

Tracing Intersection Curves

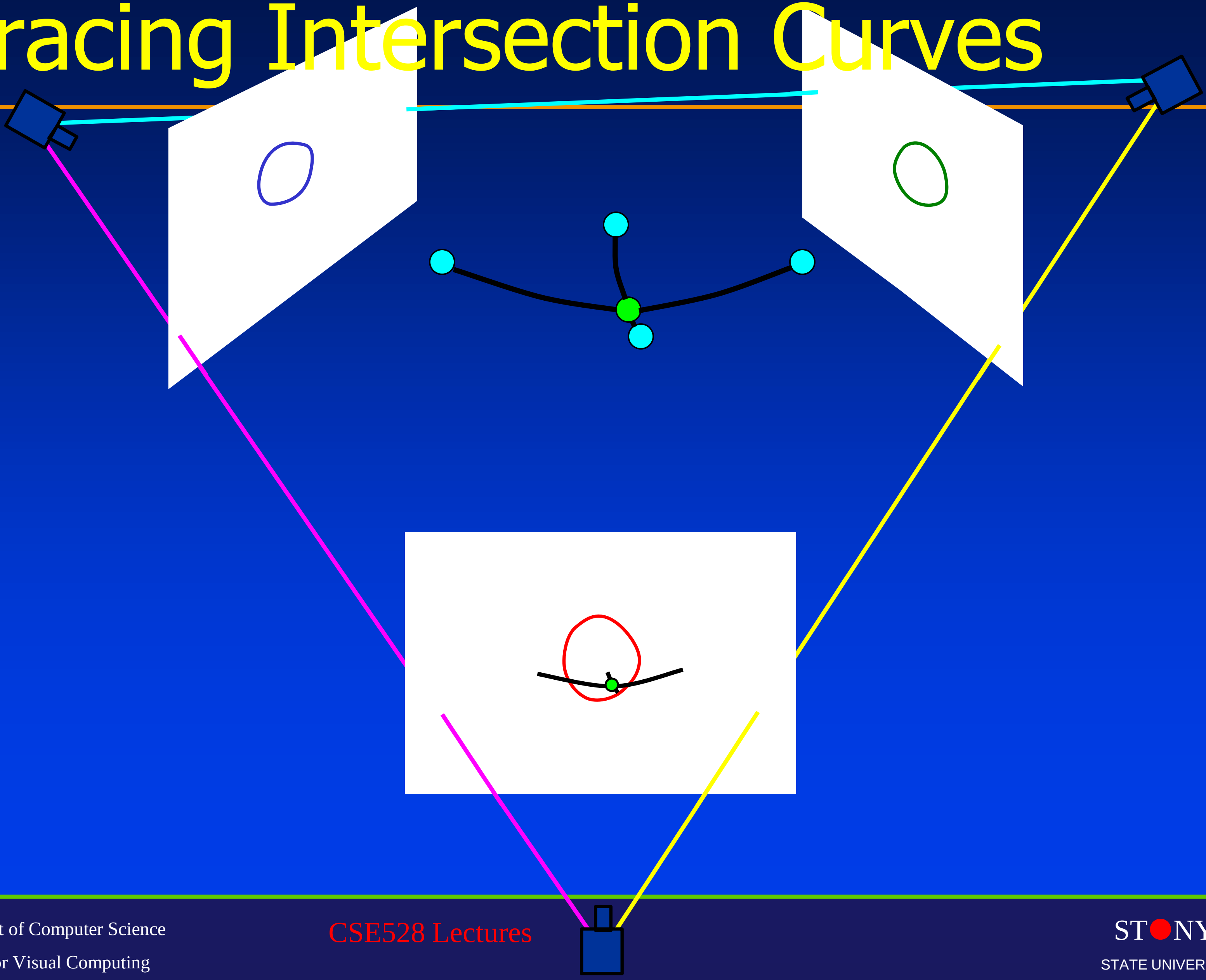


Use transfer for reprojection

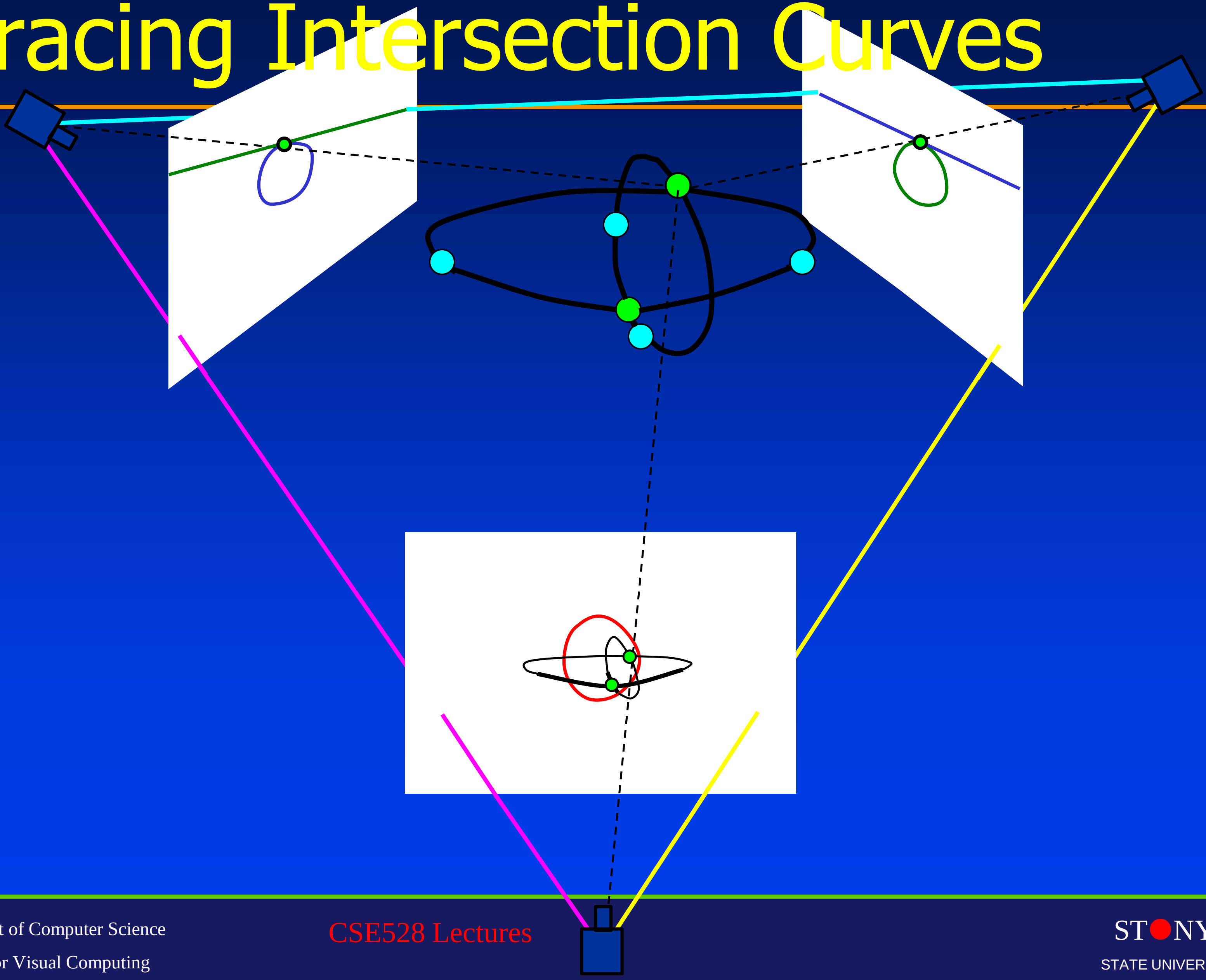
Tracing Intersection Curves



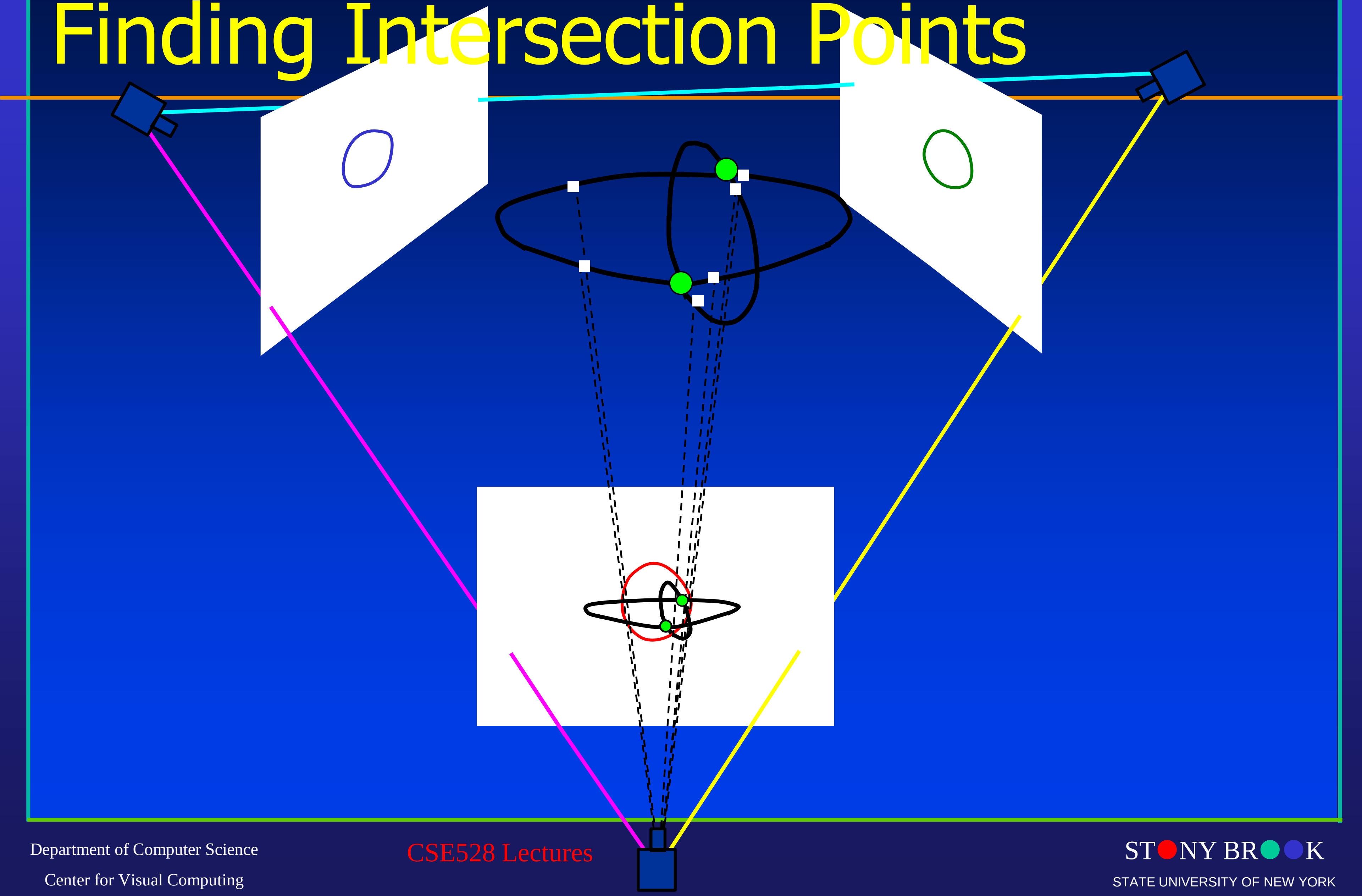
Tracing Intersection Curves



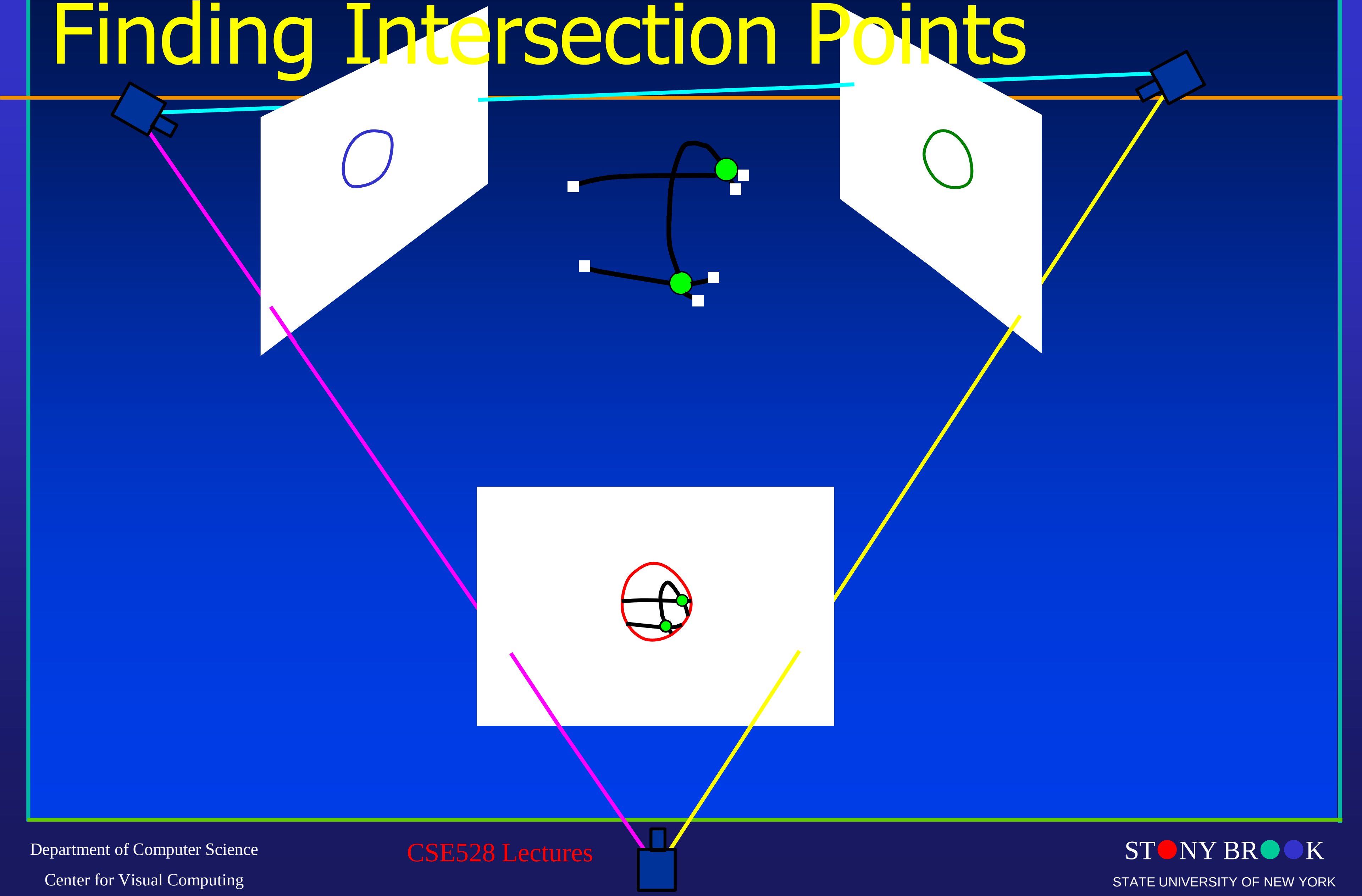
Tracing Intersection Curves



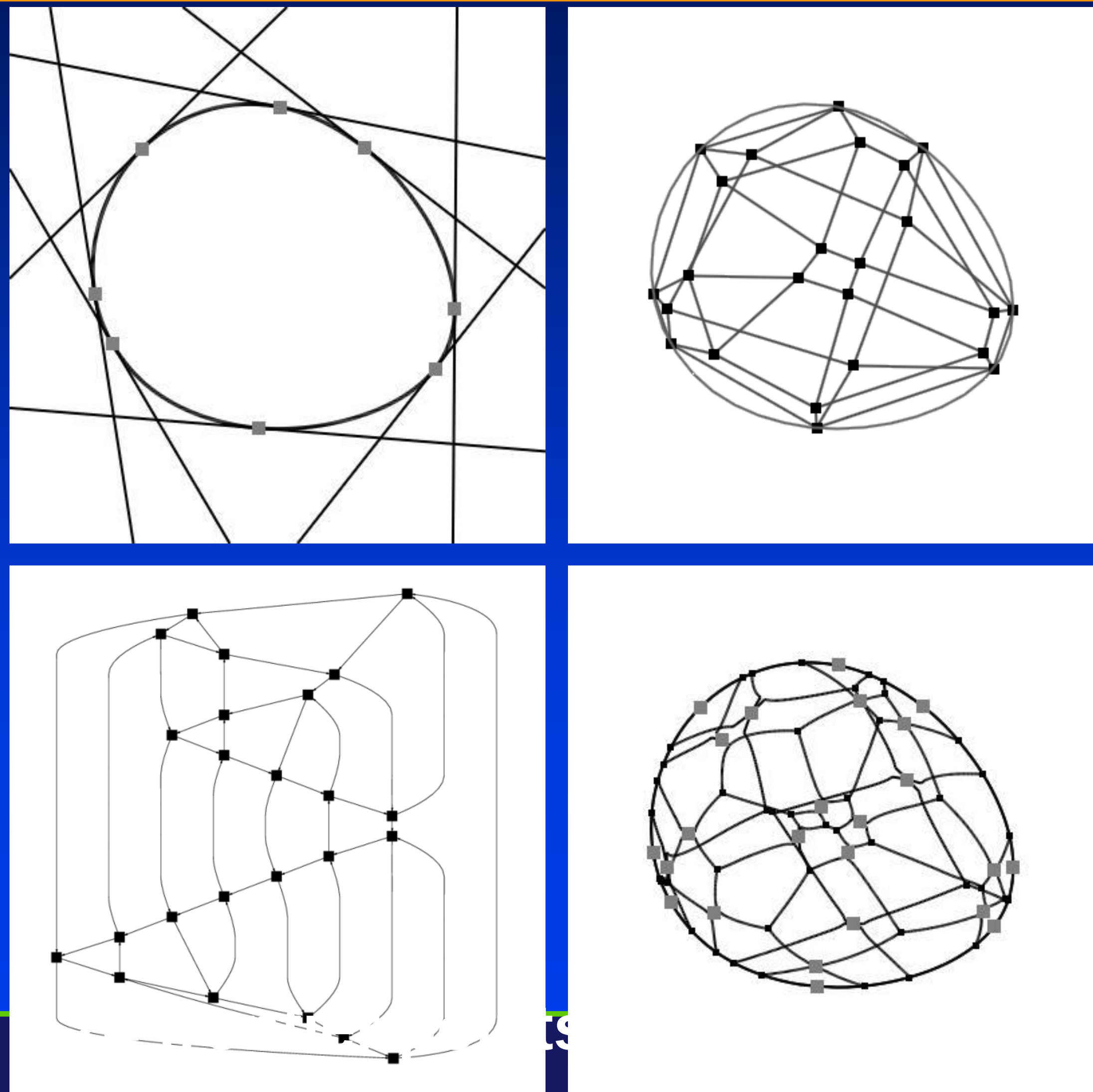
Finding Intersection Points



Finding Intersection Points



The Egg (Synthetic, 6 Views)



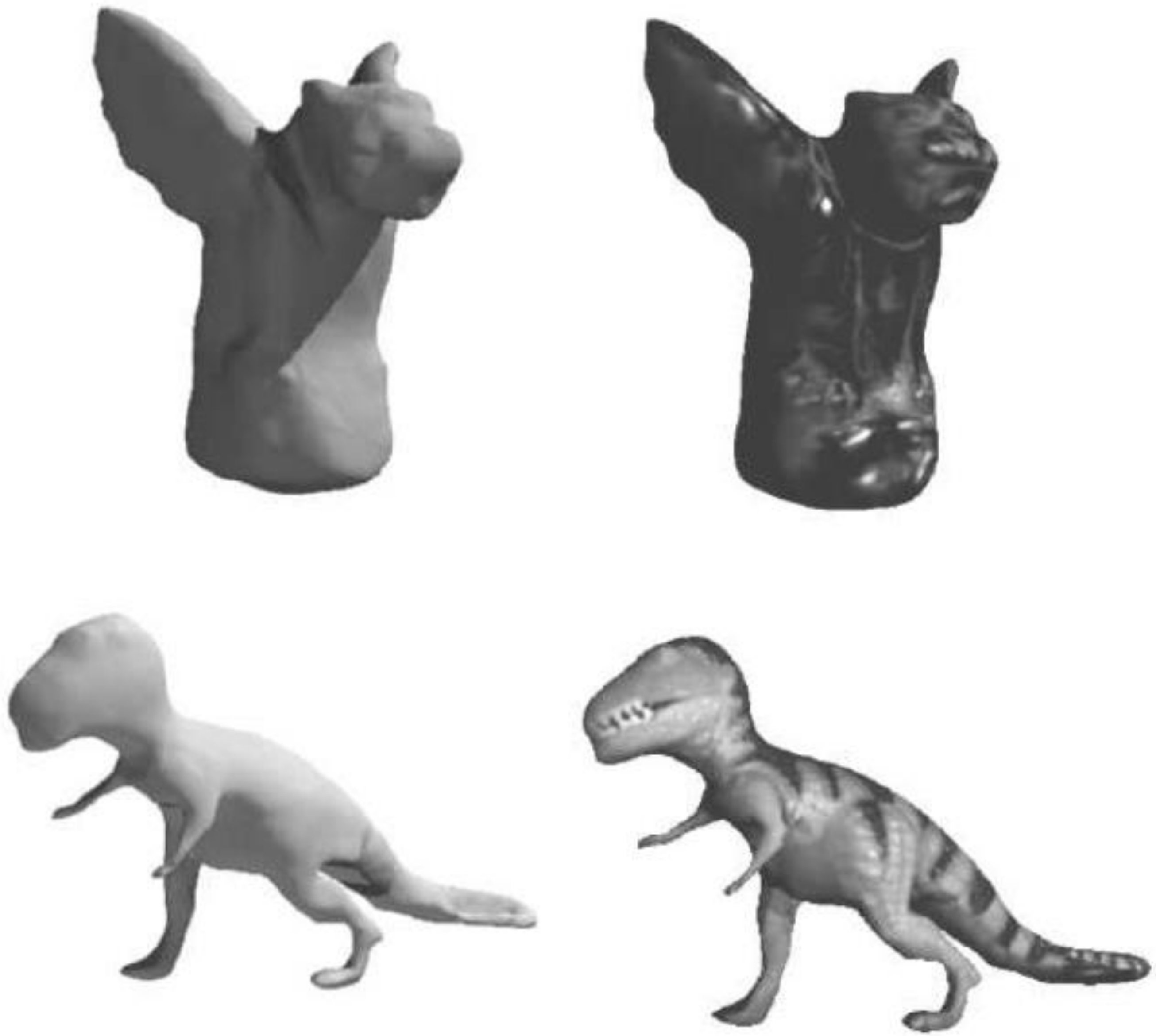
Volumetric Reconstruction

- It is impossible to uniquely reconstruct an object from its image contours! Why?
- Two main constraints imposed on a solid shape by its image contours:
  1. The shape should lie in the intersection of all viewing cones
  2. The cones should be tangent to its surface
- Techniques:
  1. Voxel carving
  2. Polyhedral approximation
  3. Smooth surface fitting

Smooth Surfaces from Image Contours

- Example by Ponce: which minimizes the Spline parameterization energy:

$$\frac{1}{q}\sum_{i=1}^{q} d^2(R_i, S) + \lambda \sum_{i=1}^{r} \iint \left[|P_{uu}|^2 + 2|P_{uv}|^2 + |P_{vv}|^2\right] du dv$$



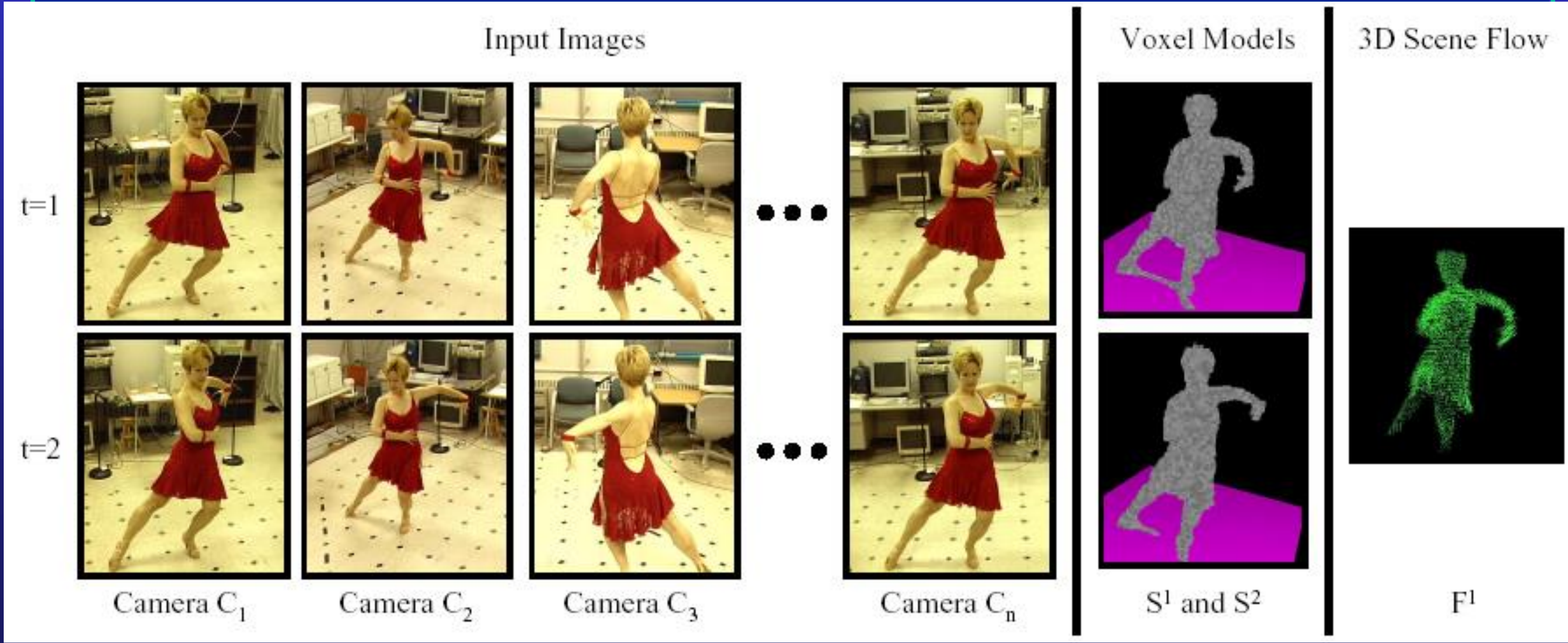
Virtualized Reality

- Capture synchronized video from a full hemisphere of views

- Perform new view generation

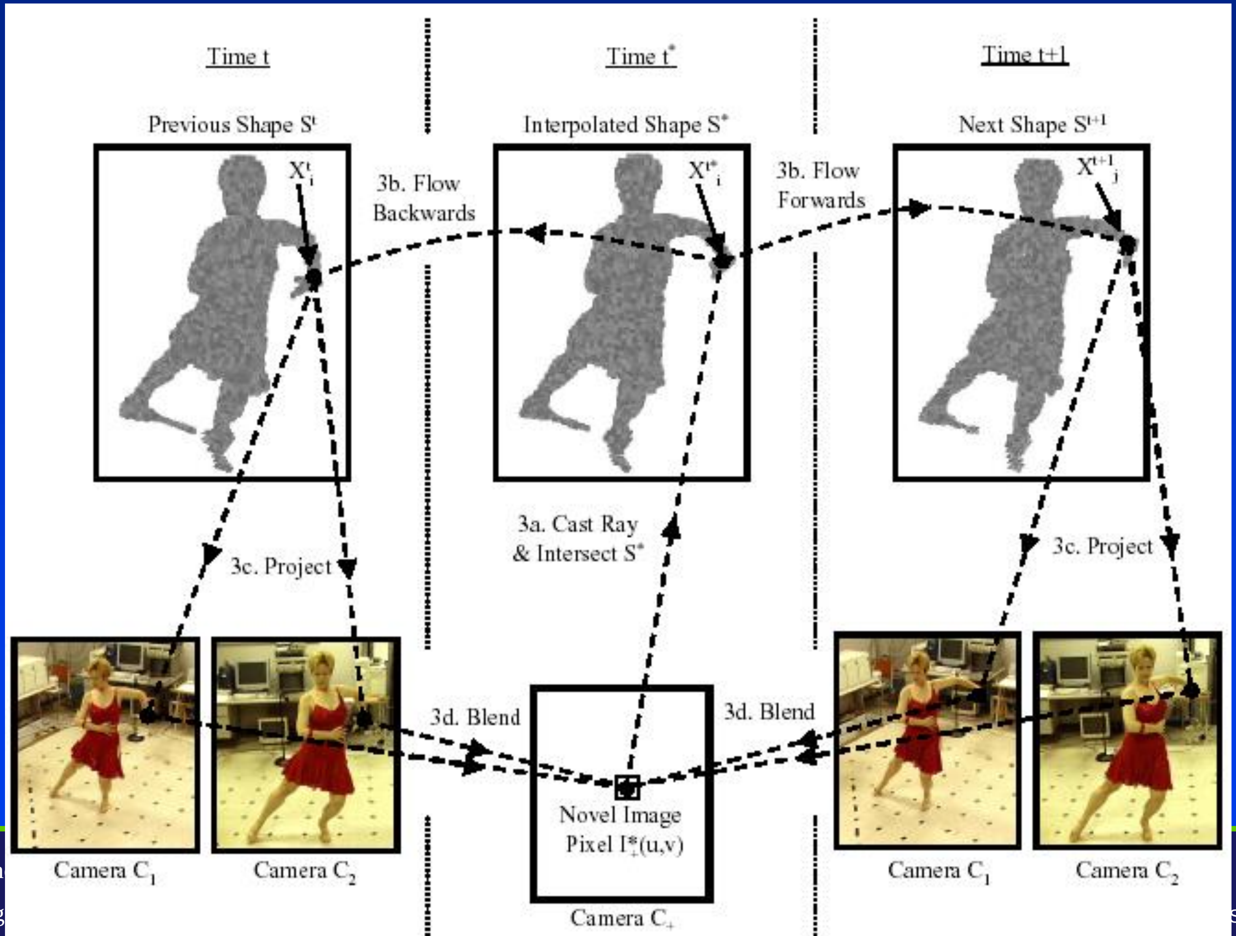
Virtualized Reality

- Spatio-Temporal View Interpolation
  S. Vedula, S. Baker, and T. Kanade
  Eurographics Workshop on Rendering, June, 2002.



Virtualized Reality

- Build 3D model and compute 3D scene flow, interpolate view and time.



Scene Modeling from Unregistered Images

- Not necessary to reconstruct all images into one global coordinate system
- A priori model of the scene

Image-based Modeling

Modeling and Rendering Architecture from Photographs

Debevec, Taylor, and Malik 1996

Original photograph with marked edges | Recovered model | Model edges projected onto photograph | Synthetic rendering

Facade

- Select building blocks
- Align them in each image
- Solve for camera pose and block parameters (please make sure also using constraints)

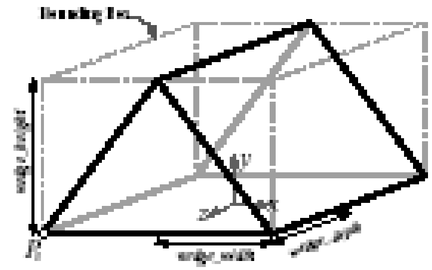
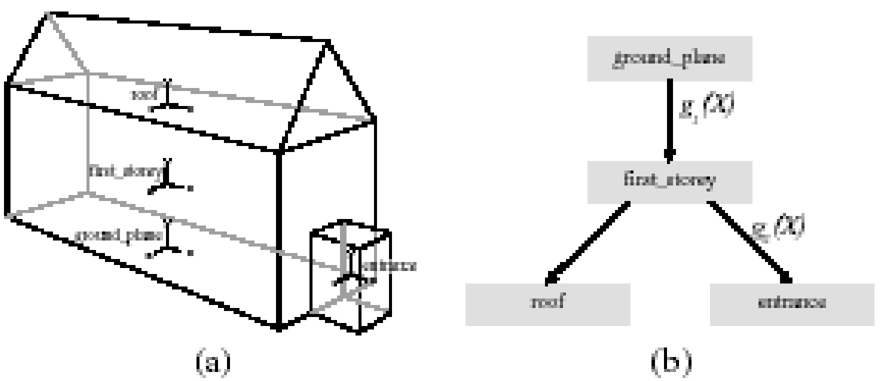


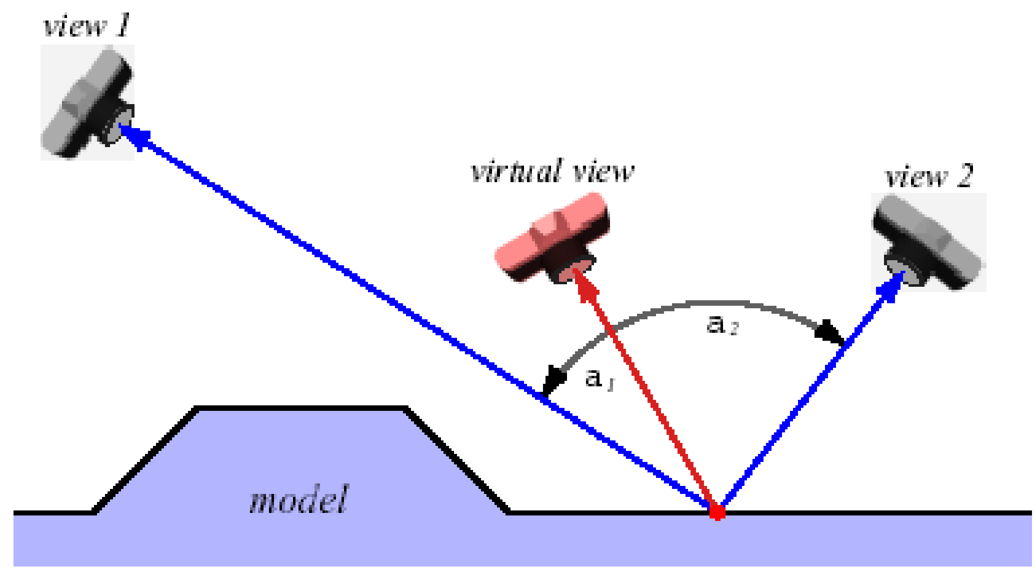
Figure 3: *A wedge block with its parameters and bounding box.*

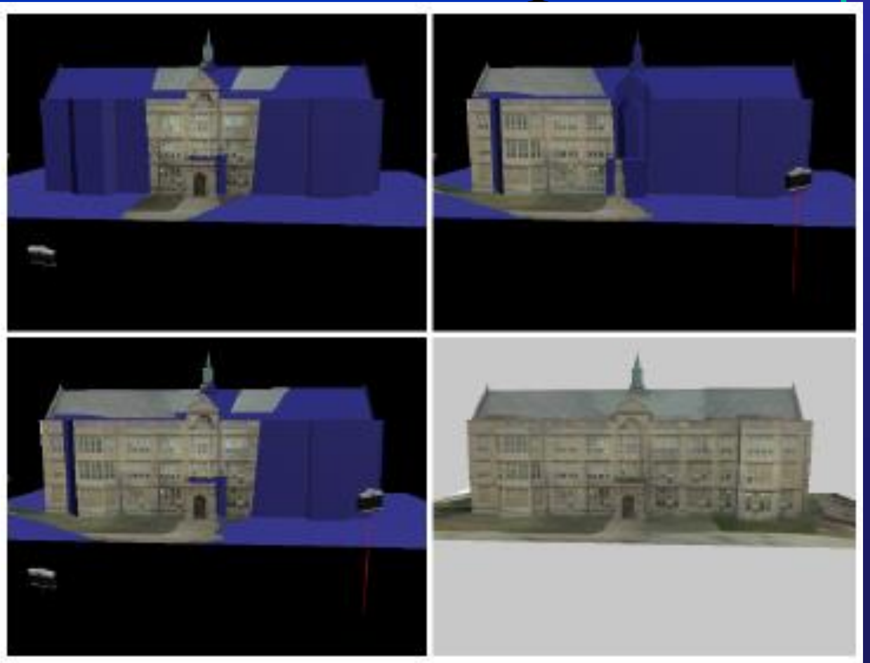


(a) *A geometric model of a simple building.* **(b)** *The ierarchical representation. The nodes in the tree repre- metric primitives (called* blocks*) while the links contain l relationships between the blocks.*

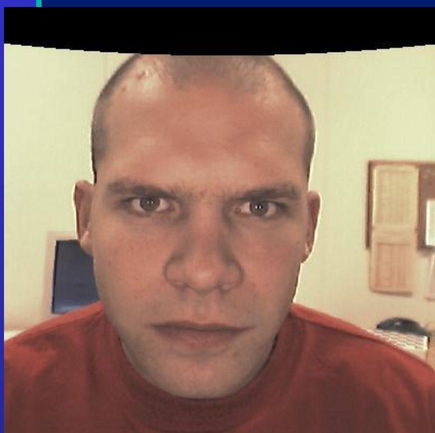
View-dependent Texture Mapping

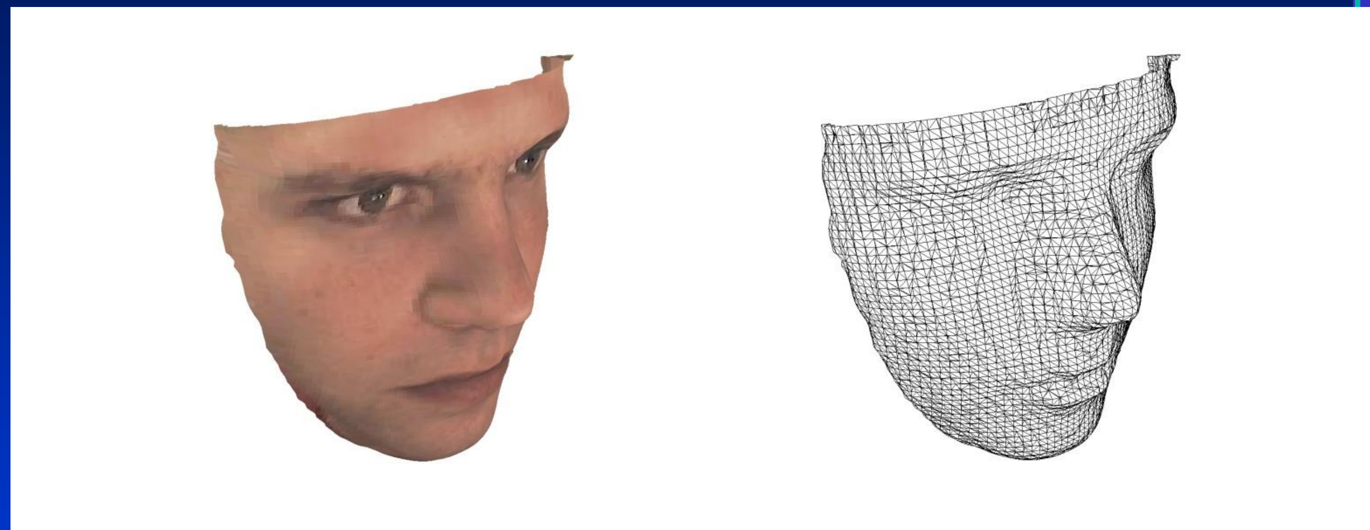
- Determine visible cameras for each surface element
- Blend textures (images) depending on distance between original camera and novel viewpoint

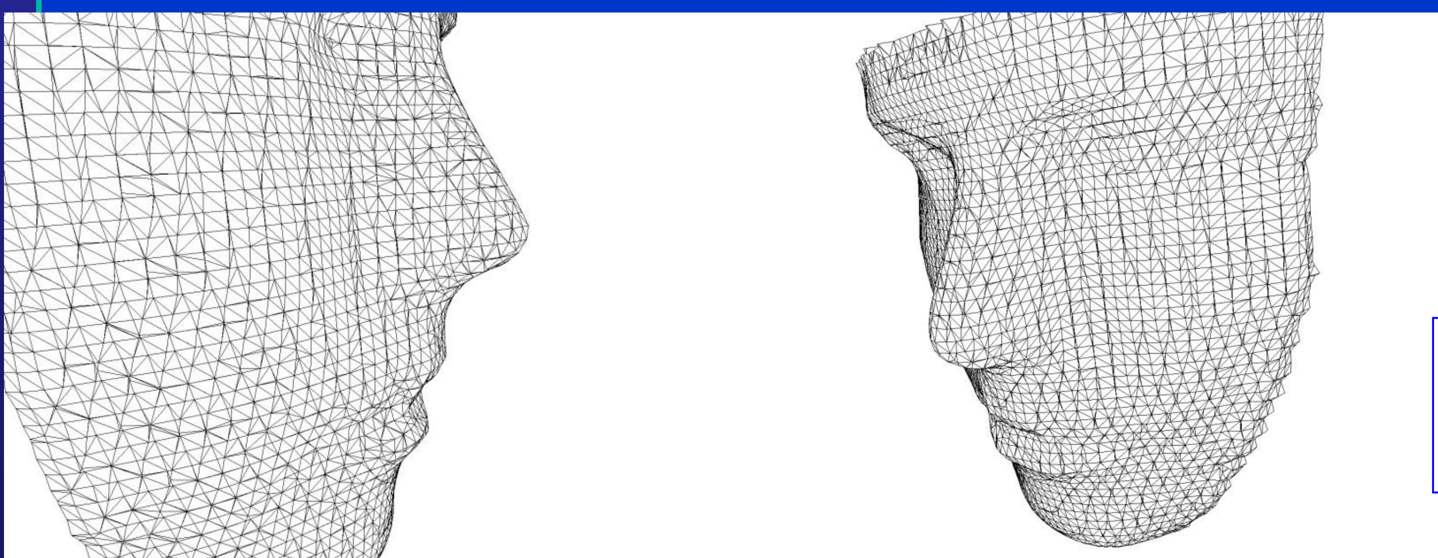




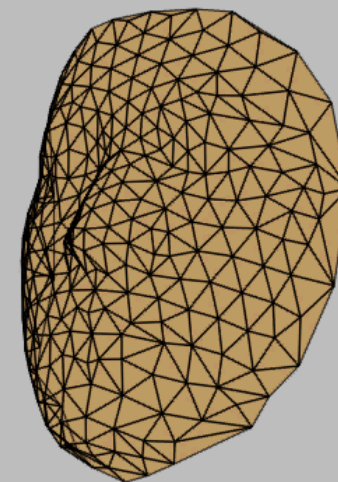
Model-based Reconstruction from One Image

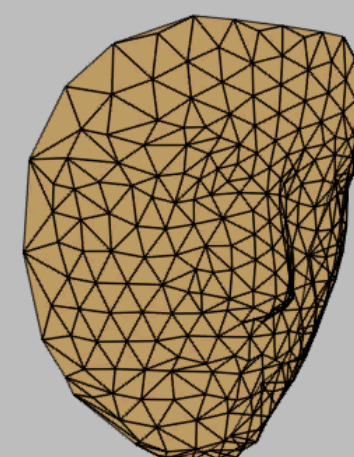




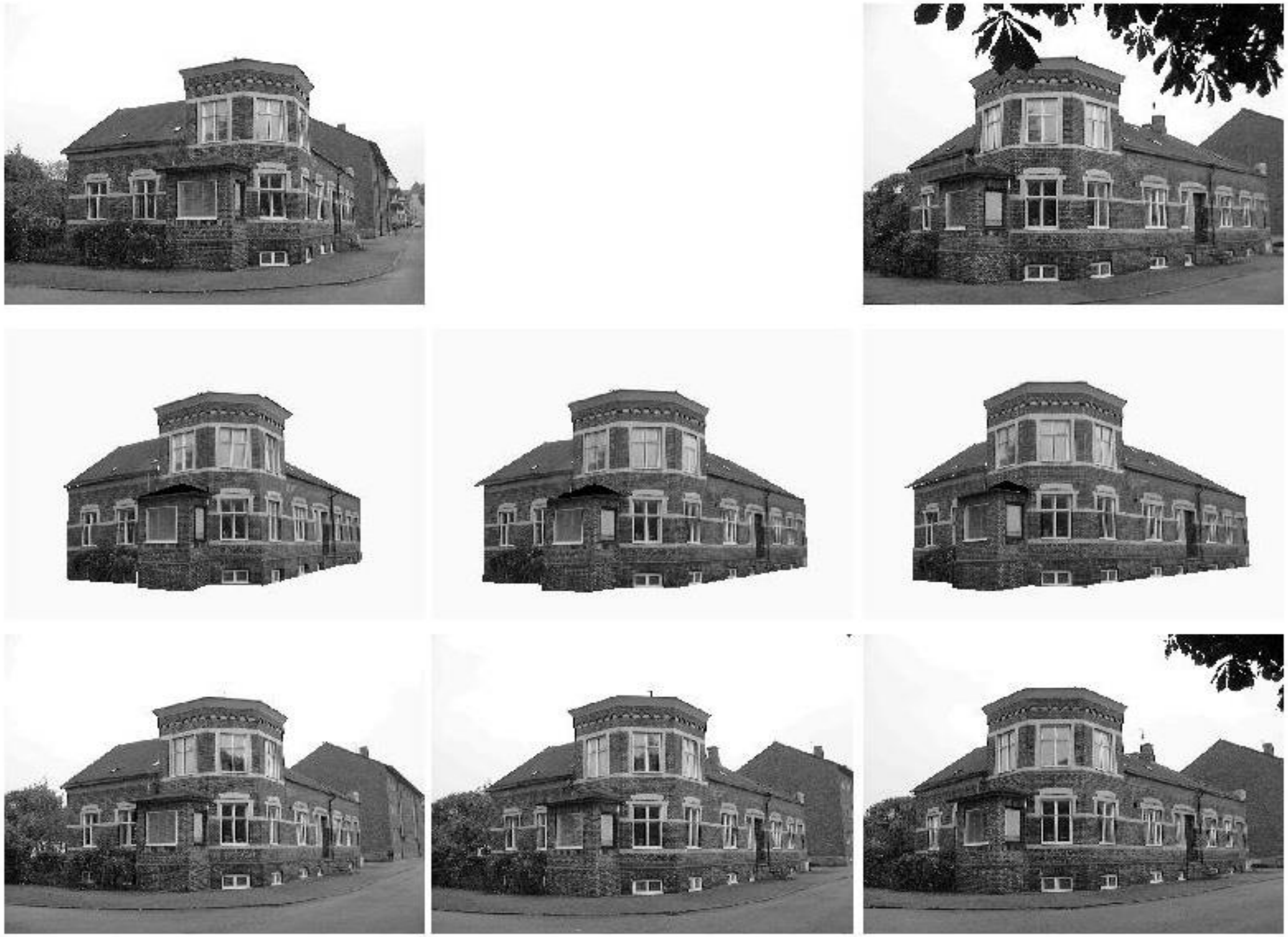


J-E Solem, F. Kahl, 2005





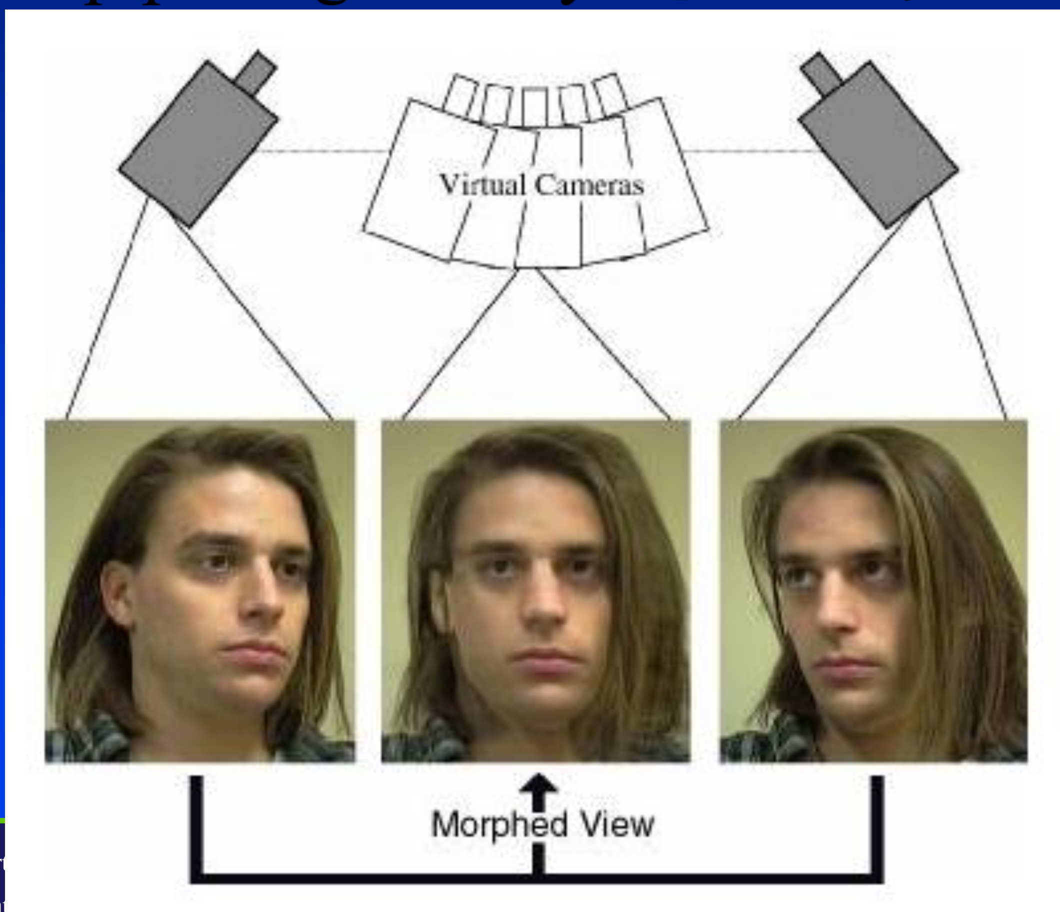
Approach 2: Transfer-based Image Synthesis



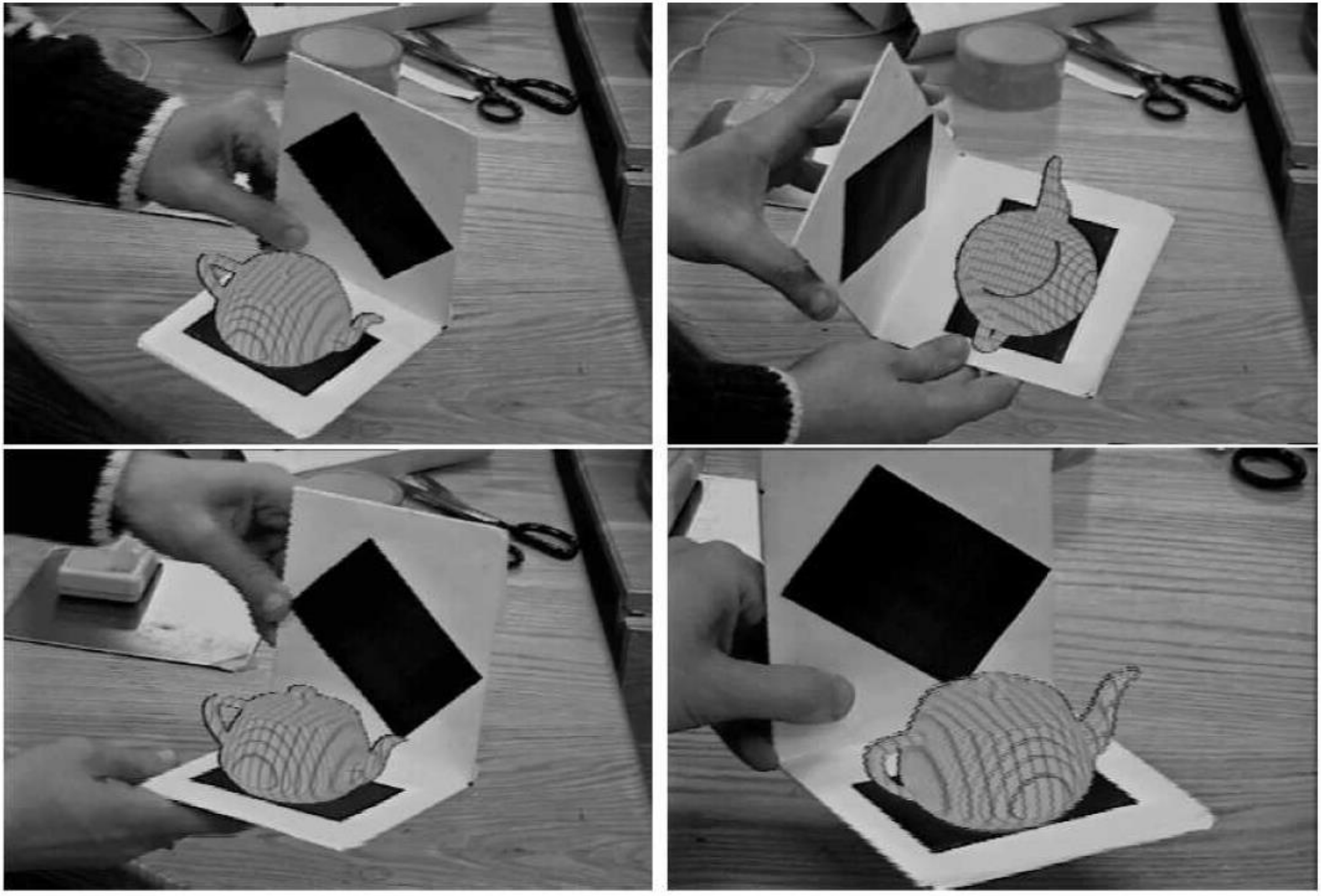
This example is based on computing consistent homographies between all planes (*B. Johansson, 2003*)

View Morphing

- Morph between pair of images using epipolar geometry [Seitz & Dyer, SIGGRAPH'96]



Affine View Synthesis

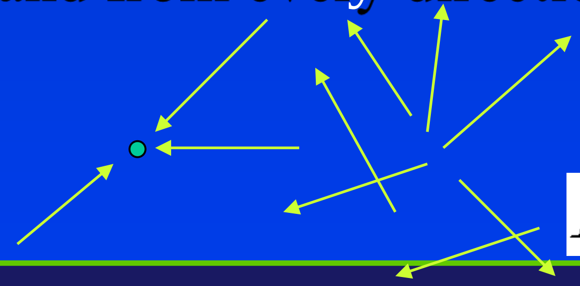


The Light Field

What is Light?

- Electromagnetic radiation (EMR) moving along rays in space
  - R(λ) is EMR, measured in units of power (watts)
    - λ is wavelength

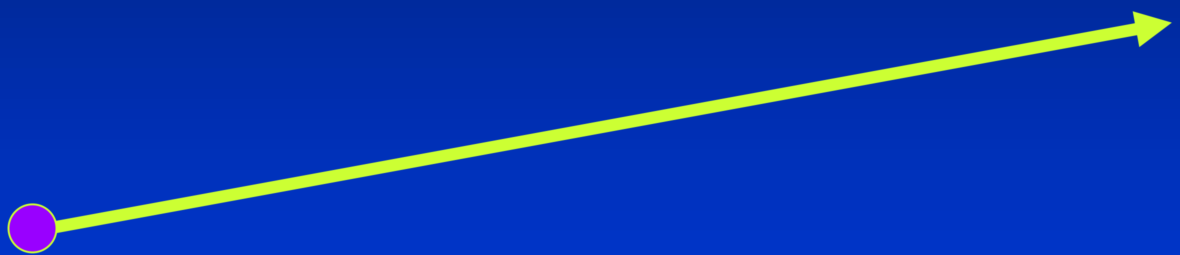
- Light field
  - We can describe all of the light in the scene by specifying the radiation (or **“radiance”** along all light rays) arriving at every point in space and from every direction



$$R(X, Y, Z, \theta, \phi, \lambda, t)$$

Ray

- Constant radiance
  - time is fixed



- 5D
  - 3D position
  - 2D direction

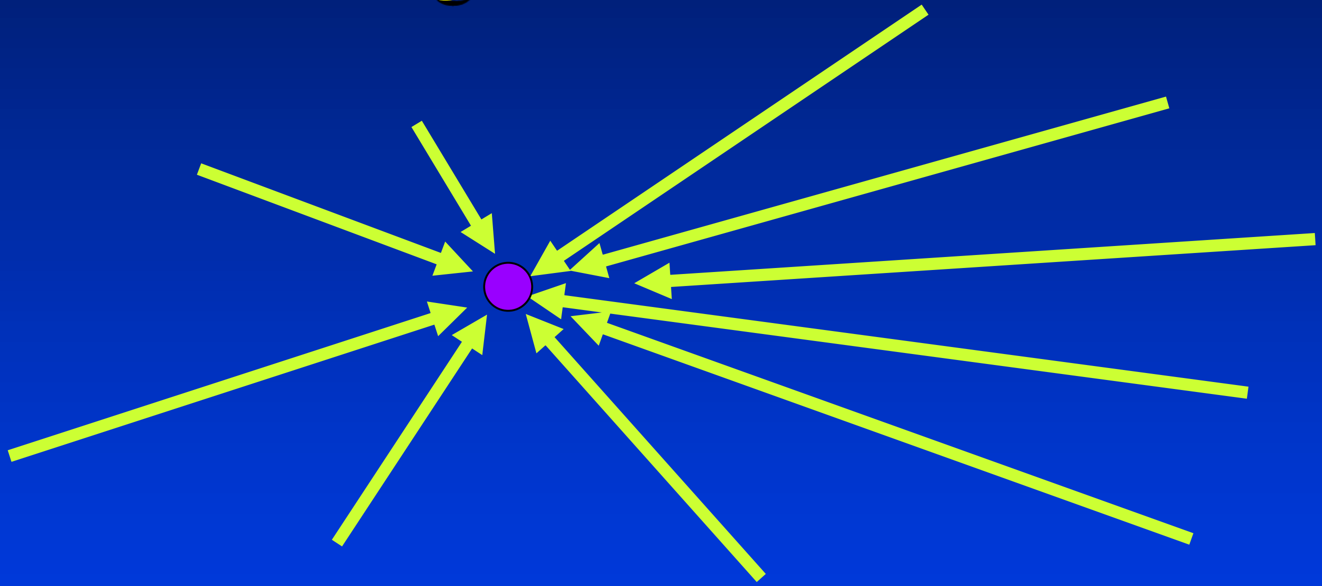
Line

- Infinite line

- 4D
  - 2D direction
  - 2D position
  - non-dispersive medium

Image

- What is an image?



- All rays pass through a point
  – Panorama

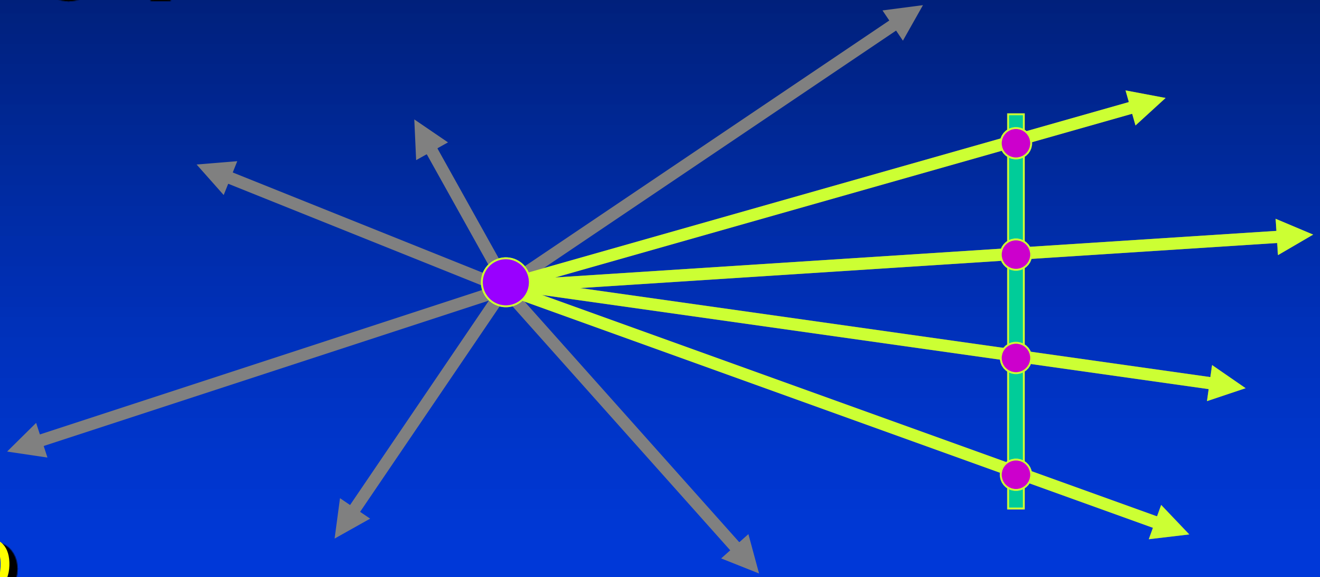
Panoramic Mosaics

- Convert panoramic image sequence into a cylindrical image

-   + . 

Image

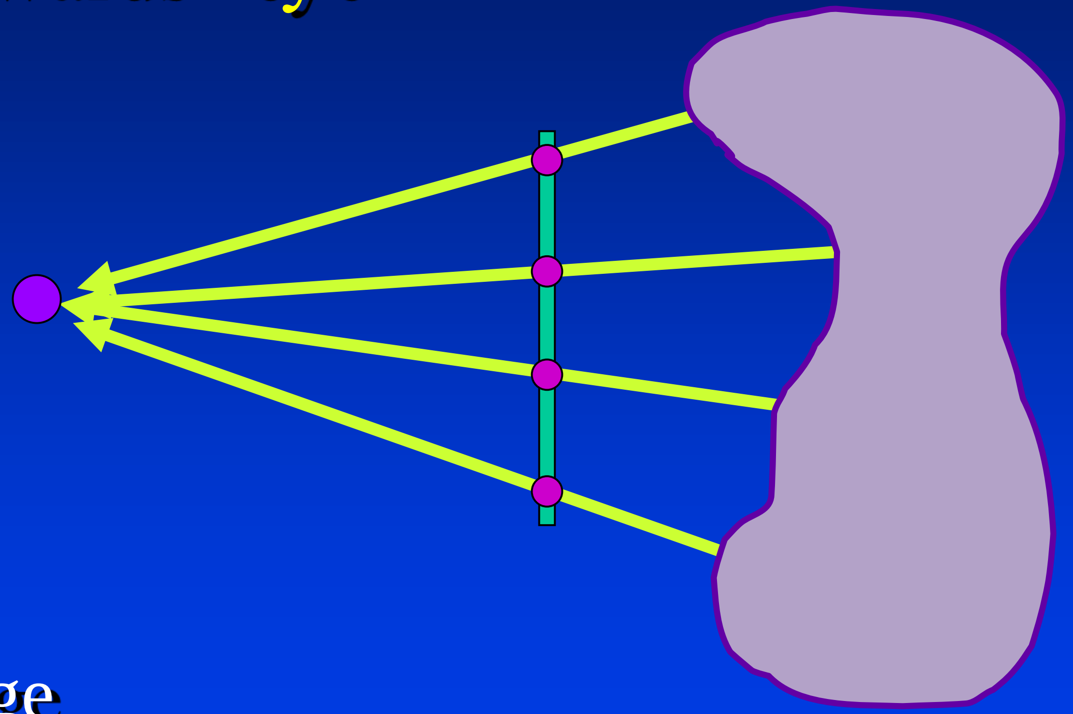
- Image plane



- 2D
  - position in plane

Object

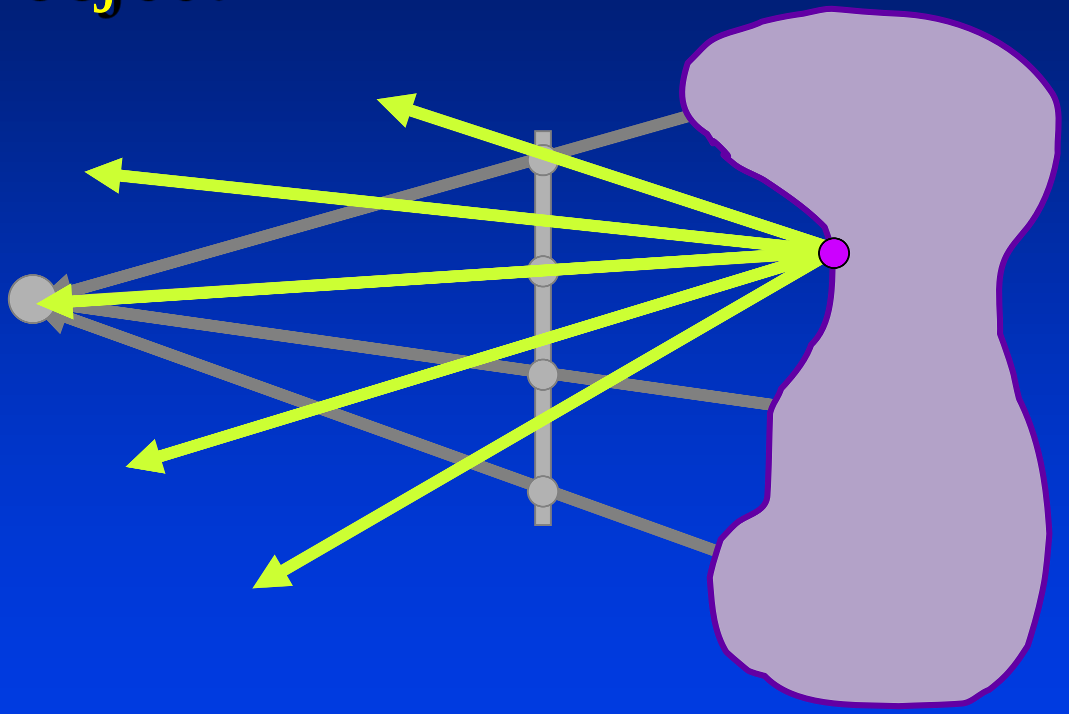
- Light leaving towards "eye"



- 2D
  - just dual of image

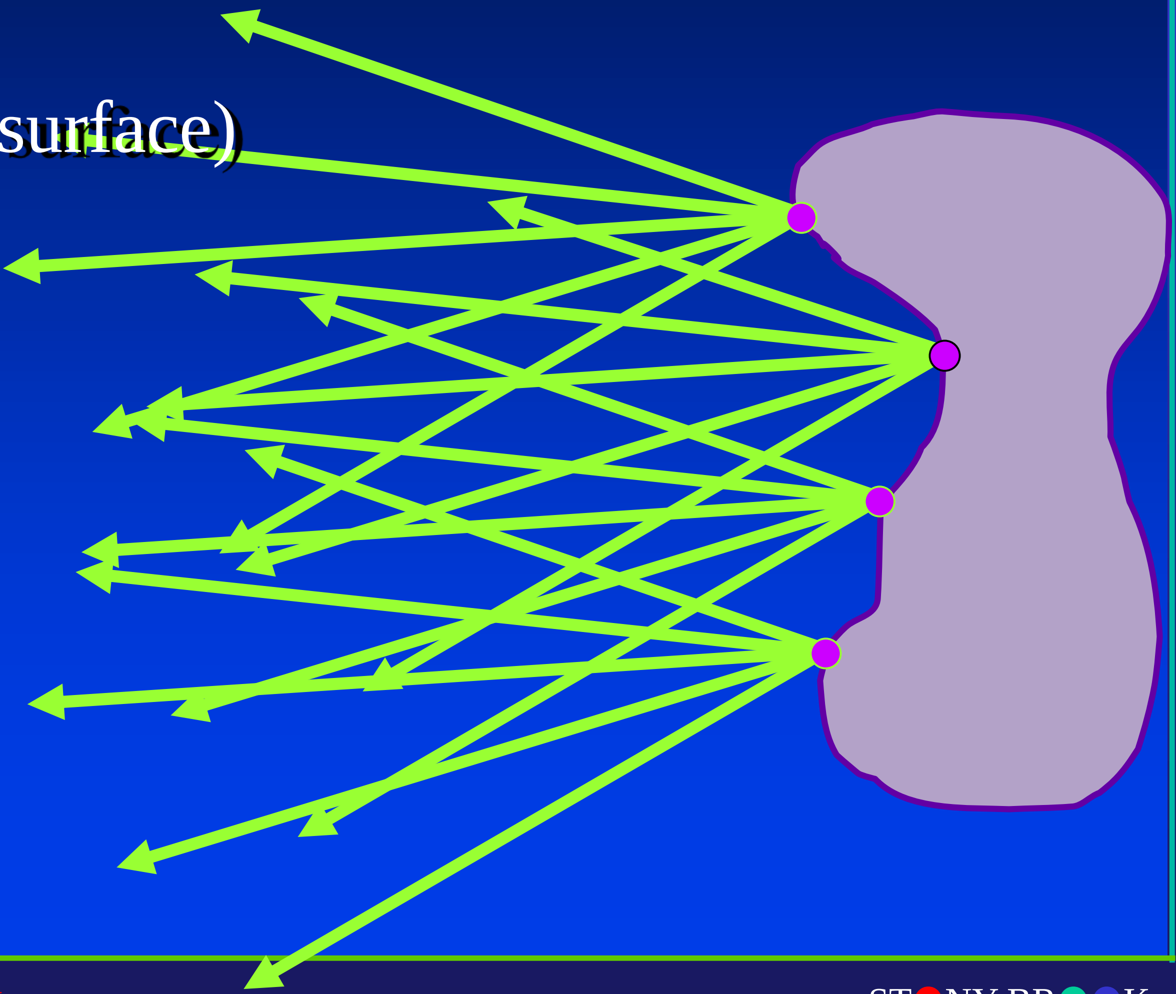
Object

- All light leaving object



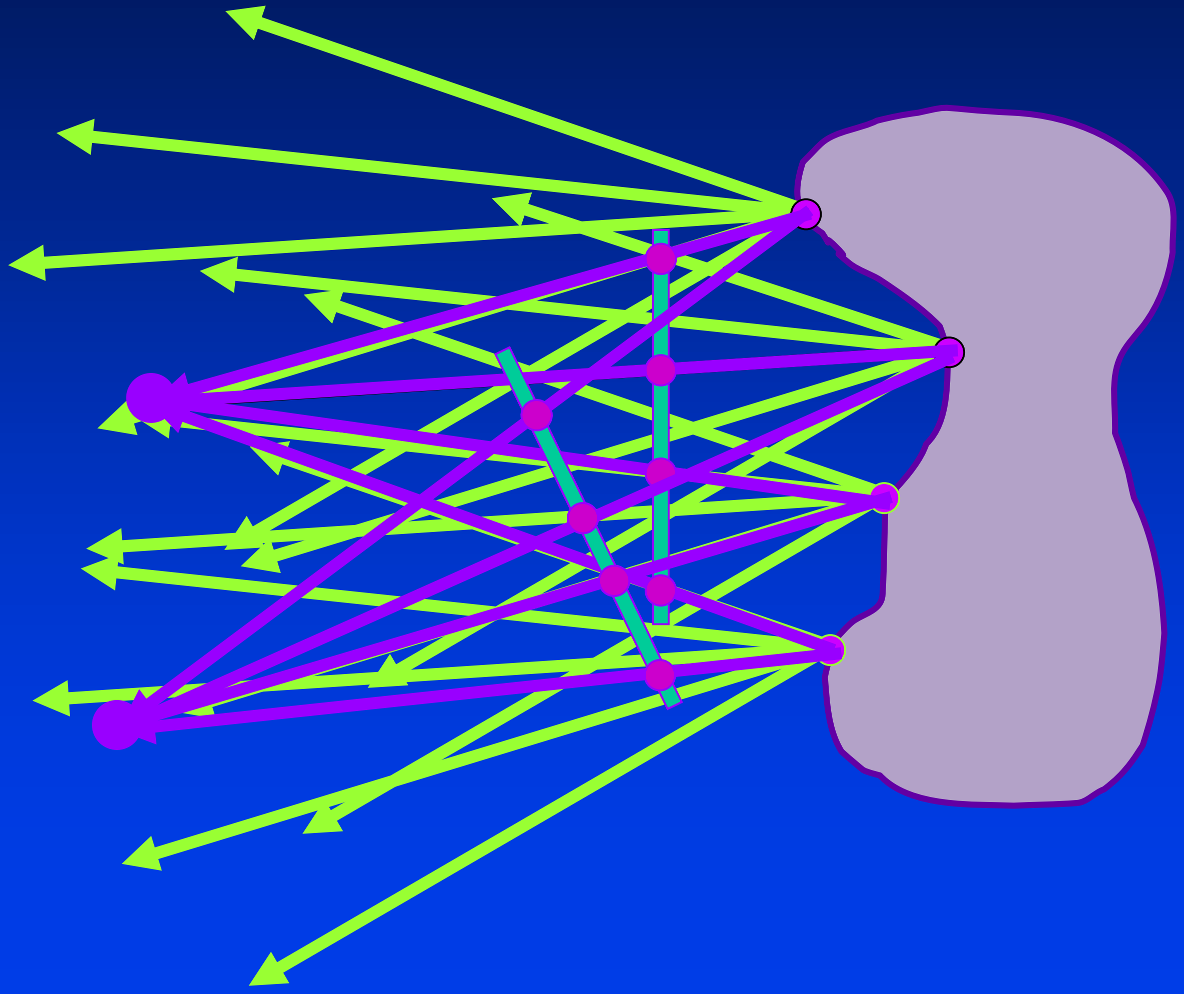
Object

- 4D
  - 2D position (on surface)
  - 2D direction



Object

- All images



The Light Field

Summary:

- Capture as many images as possible
- Store them in a smart way
- Discretize rays to synthesize new images

Complex Light Field Acquisition

- Digital Michelangelo Project
  - Marc Levoy, Stanford University
  - Lightfield ("night") assembled by Jon Shade

Surface Light Fields

- [Wood et al, SIGGRAPH 2000]