Lecture 11:

The Light Field and Image-Based Rendering

Visual Computing Systems
CMU 15-869, Fall 2013

Demo (movie)

Royal Palace: Madrid, Spain

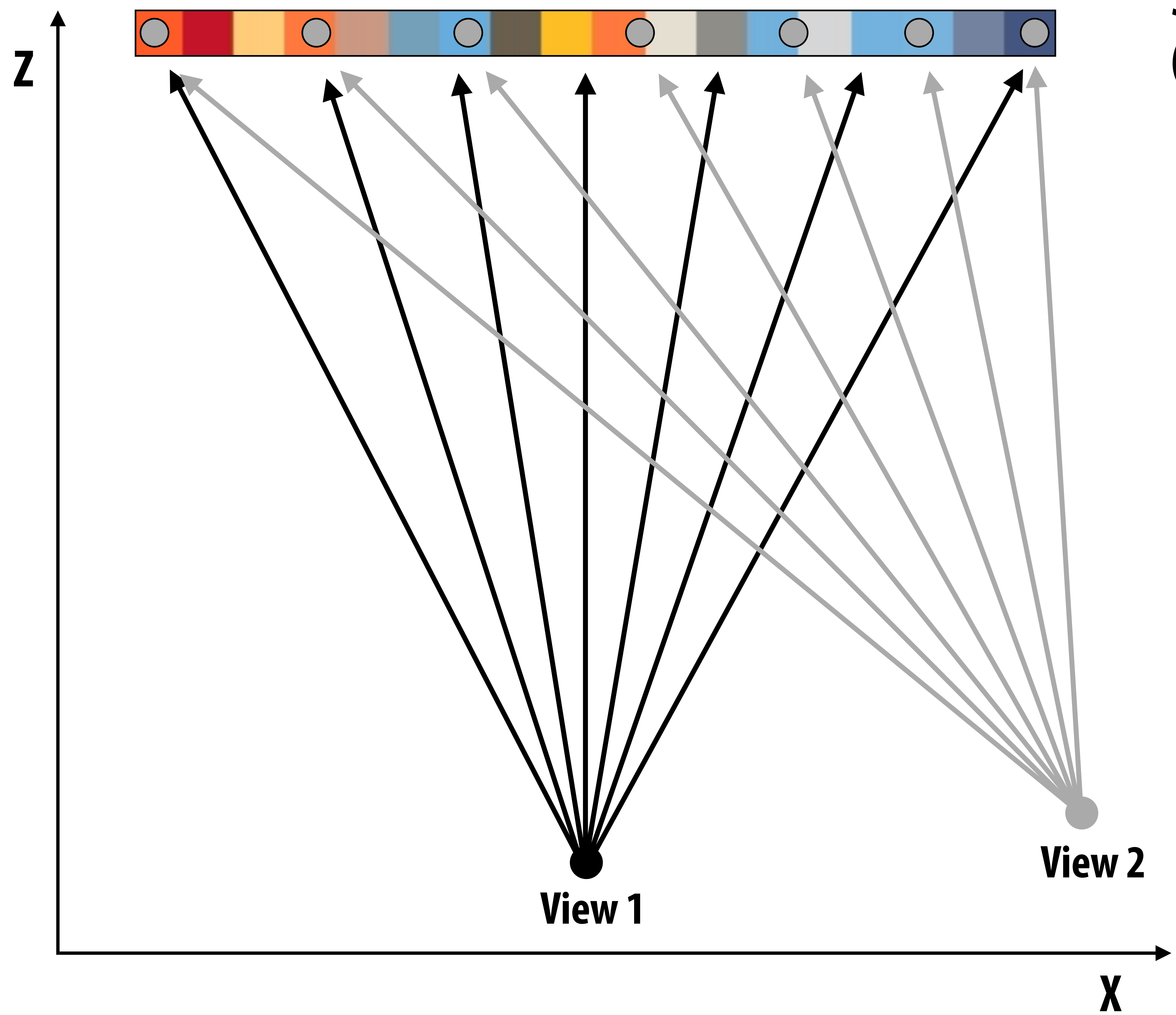
Image-based rendering (IBR)

- **So far in course: rendering = synthesizing an image from a 3D model of the scene**
  - **Model: cameras, geometry, lights, materials**

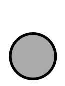
- **Today: synthesizing novel views of a scene from <u>existing images</u>**
  - **Where do the input images come from?**
    - **Previously synthesized by a renderer**
    - **Acquired from the real world (photographs)**

Why does this view look wrong?

What's going on? consider image resampling



Scene:
(represented by image rendered from view 1)

 = Image sample for view 2

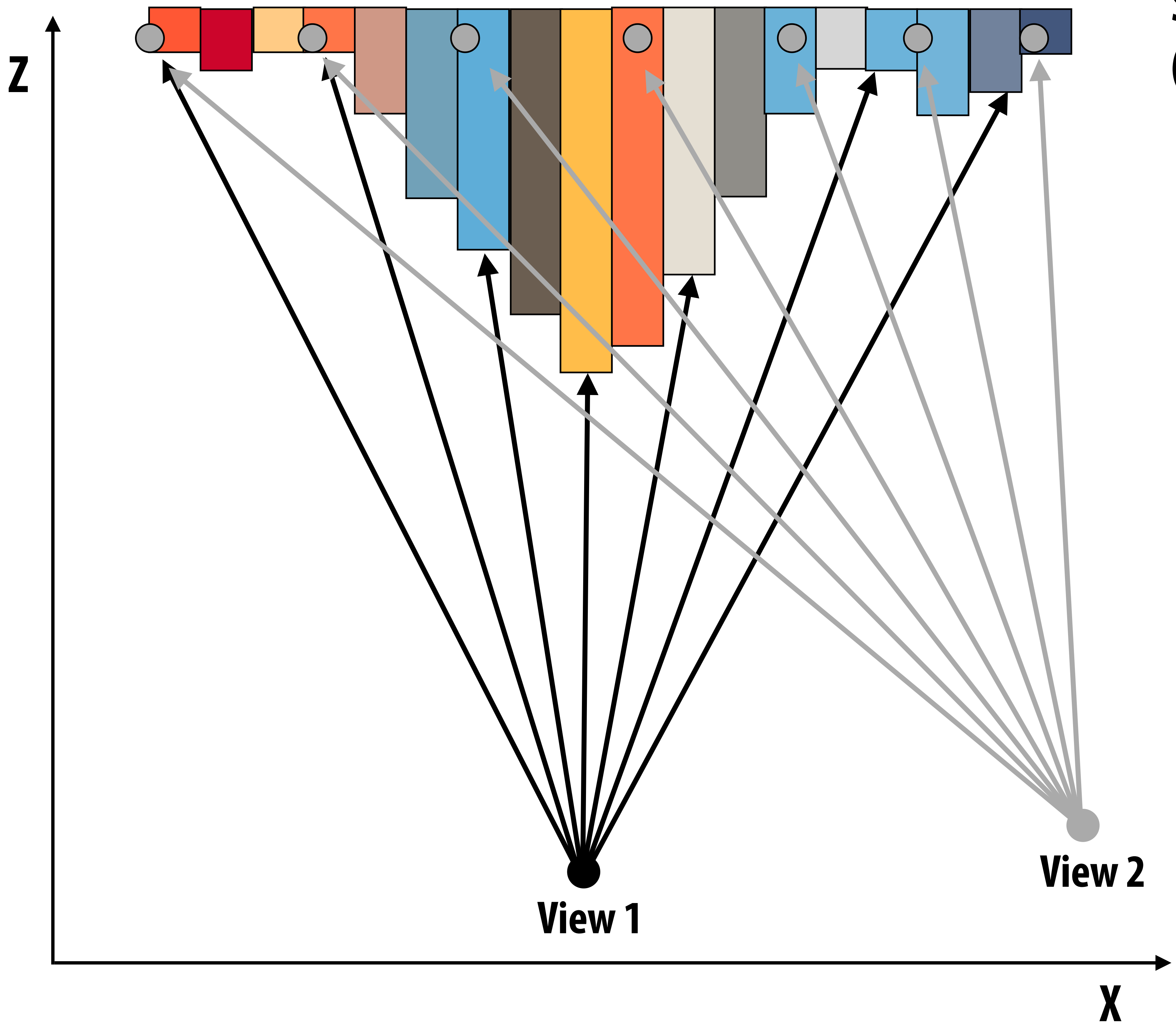
Synthesizing novel scene views from an existing image

- If scene lies approximately on a plane, affine transform of the image from view 1 yields accurate image of scene from view 2

- Recall: this is texture mapping

- Have I made assumptions in addition to planar scene geometry?
  - Hint: think about reflectance

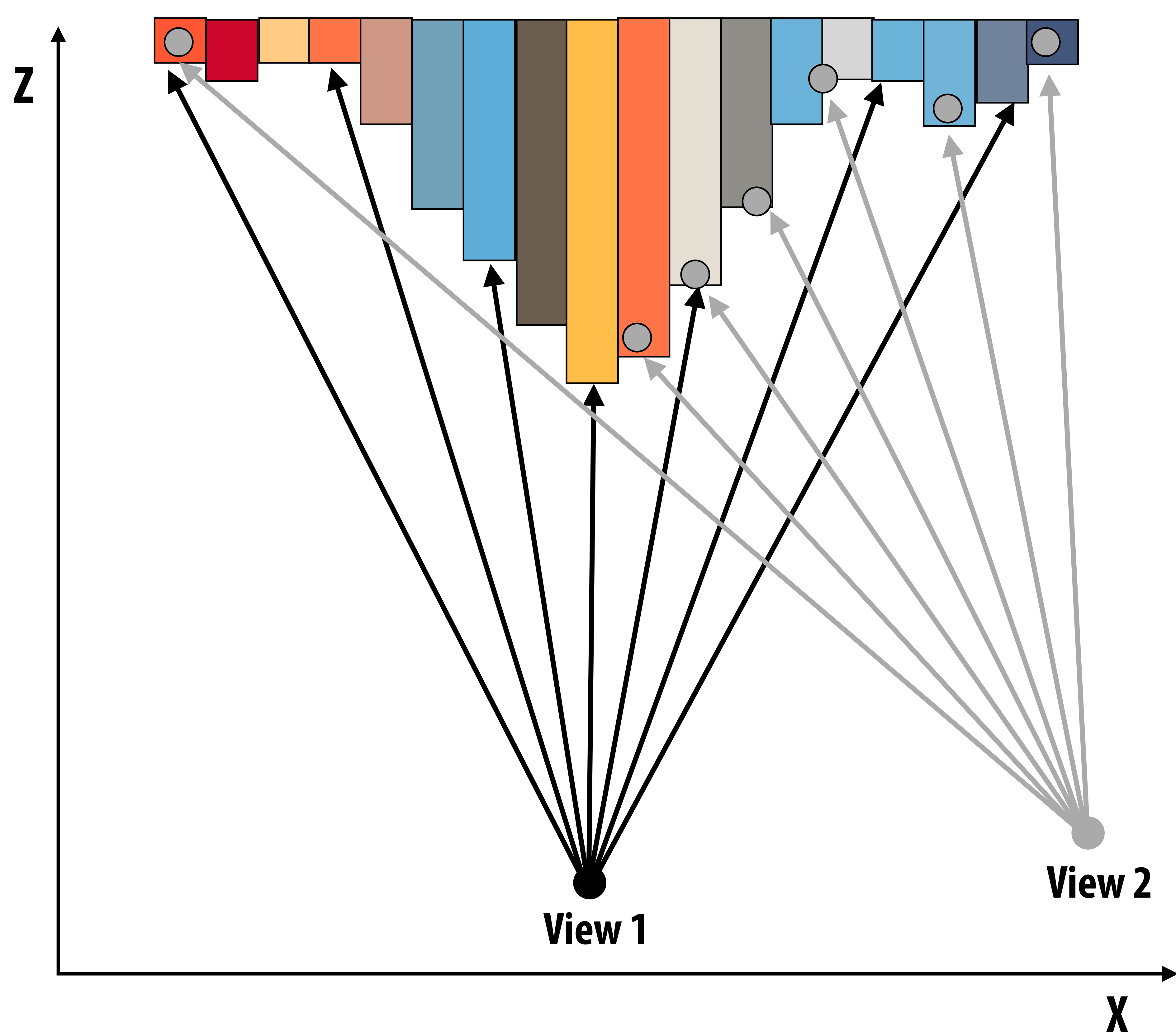
Non-planar scene



Scene:
(represented by image rendered from view 1)

= Image sample for view 2

Non-planar scene

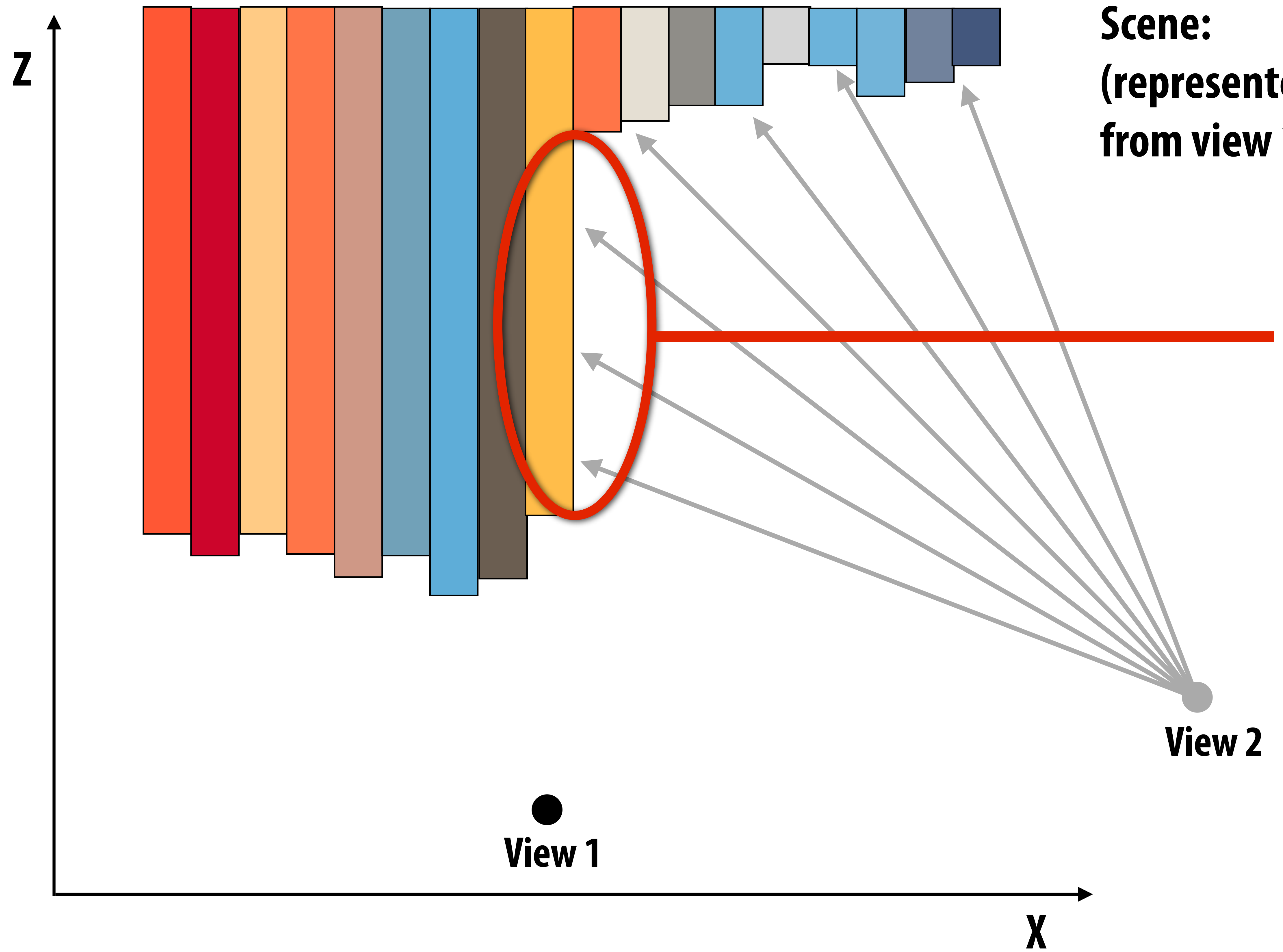


Scene:
(represented by image + depth rendered from view 1)

● = Image sample for view 2

Synthesis of novel scene views with correct perspective requires non-affine transformation of original image

Artifact: undersampled source image

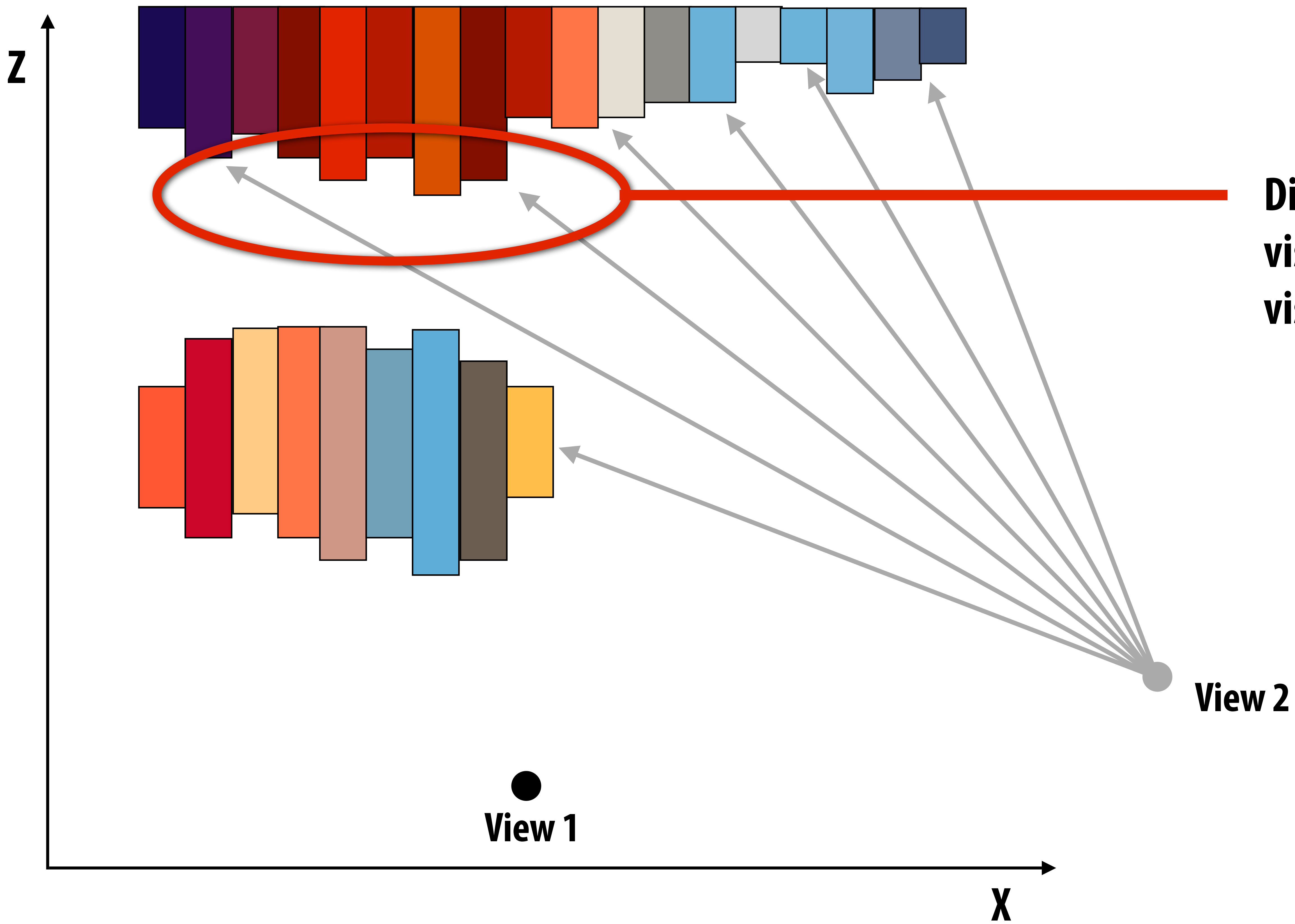


Scene:
(represented by image + depth rendered from view 1)

Undersampling: Image generated from view 1 has sparse sampling of surface regions oriented at grazing angle to view 1.

(surface region projected to one pixel in view 1, but several pixels in view 2)

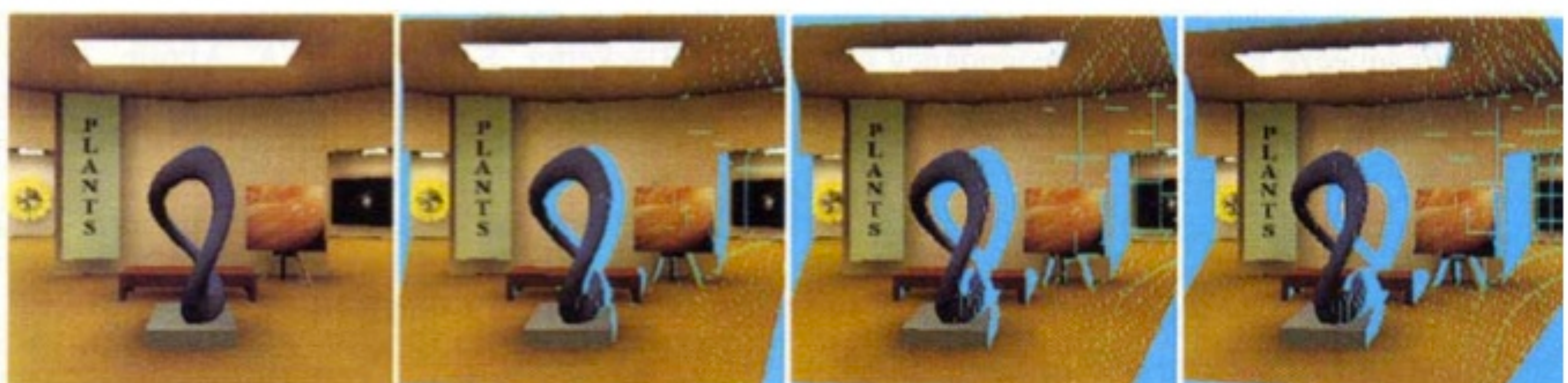
Artifact: disocclusion



Disocclusion: surface region visible from view 2 was not visible from view 1

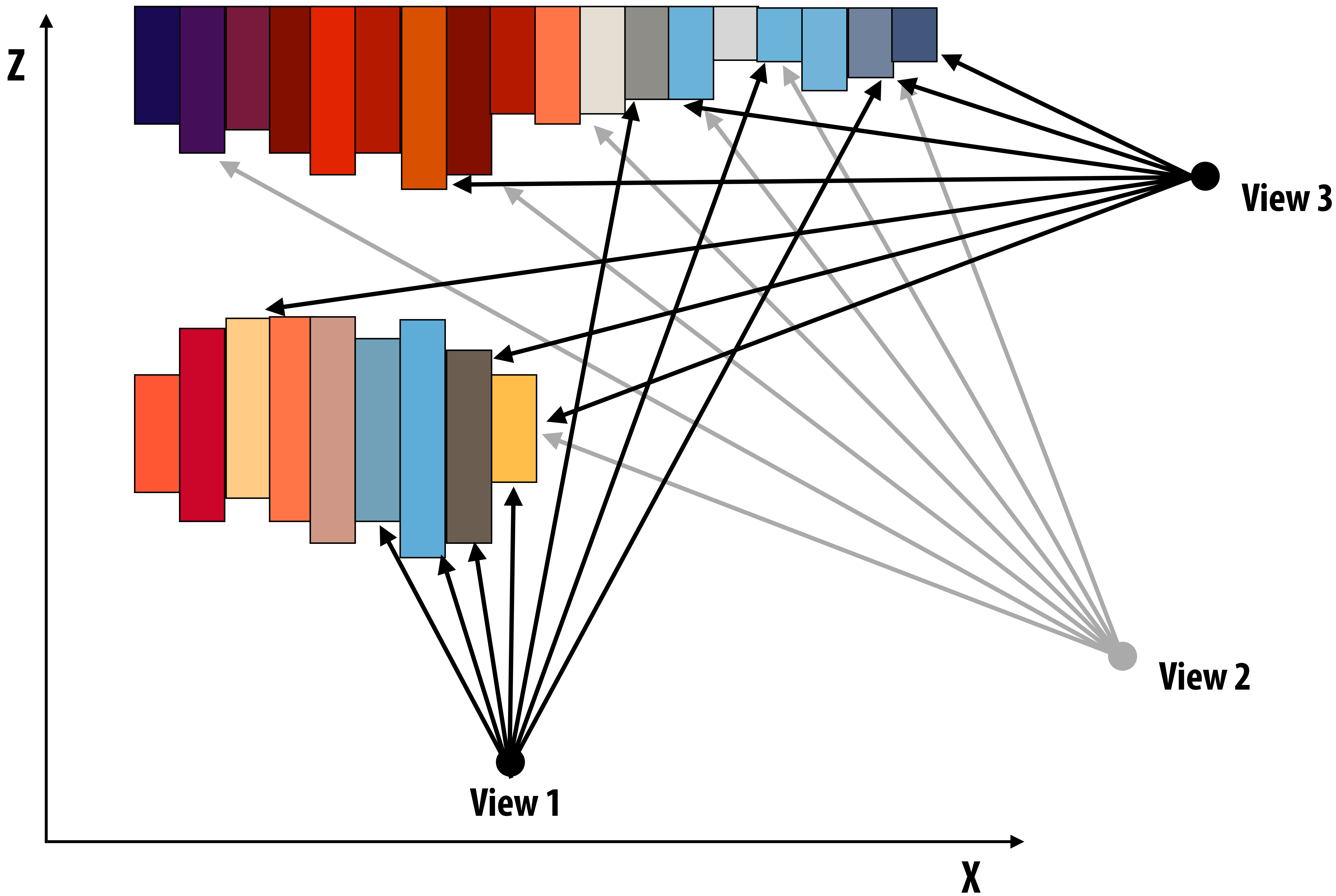
Disocclusion examples

[Credit: Chaurasia et al. 2011]



[Credit: Chen and Williams 93]

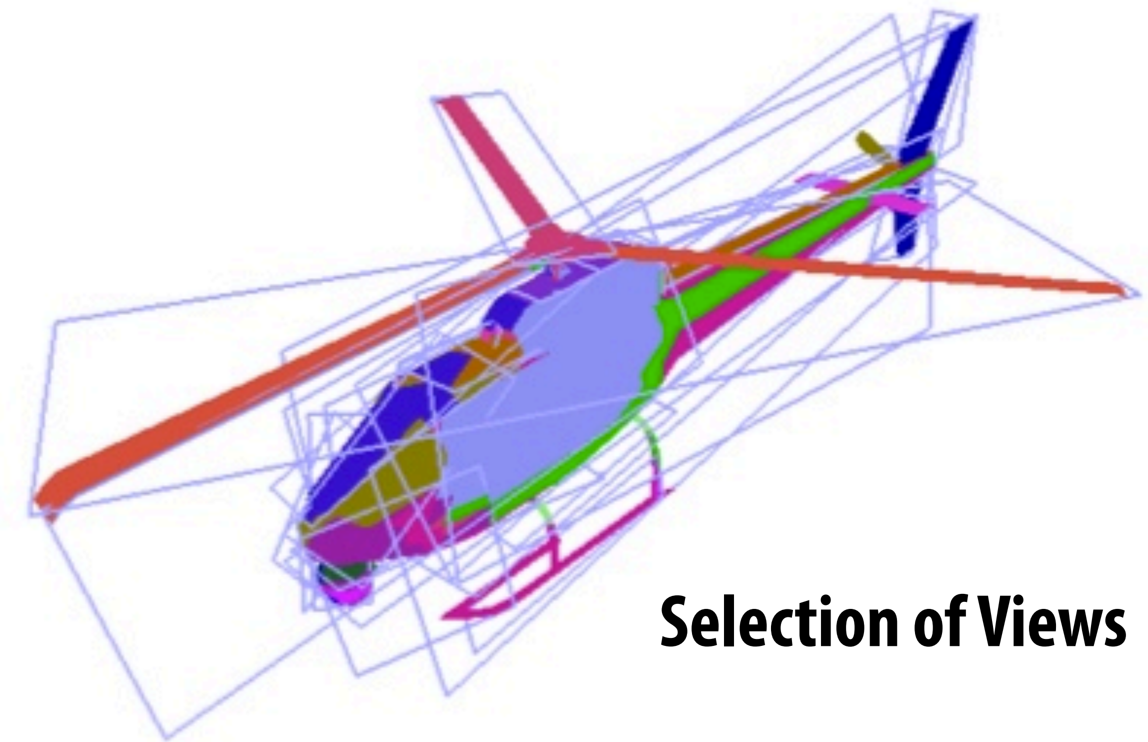
View interpolation



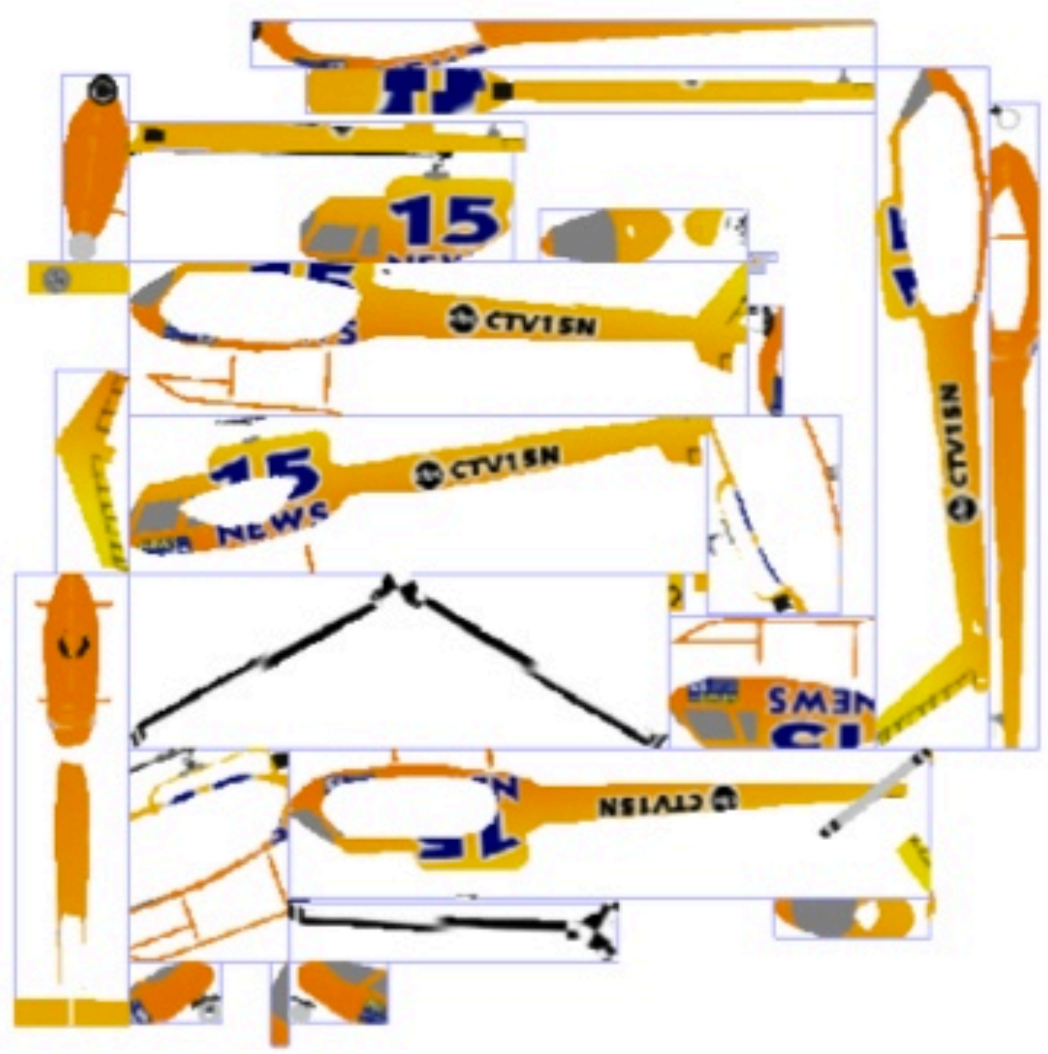
Combine results from closest pre-existing views
Question: How to combine?

Sprites

Original (complex) 3D Model
(expensive to render)



Selection of Views



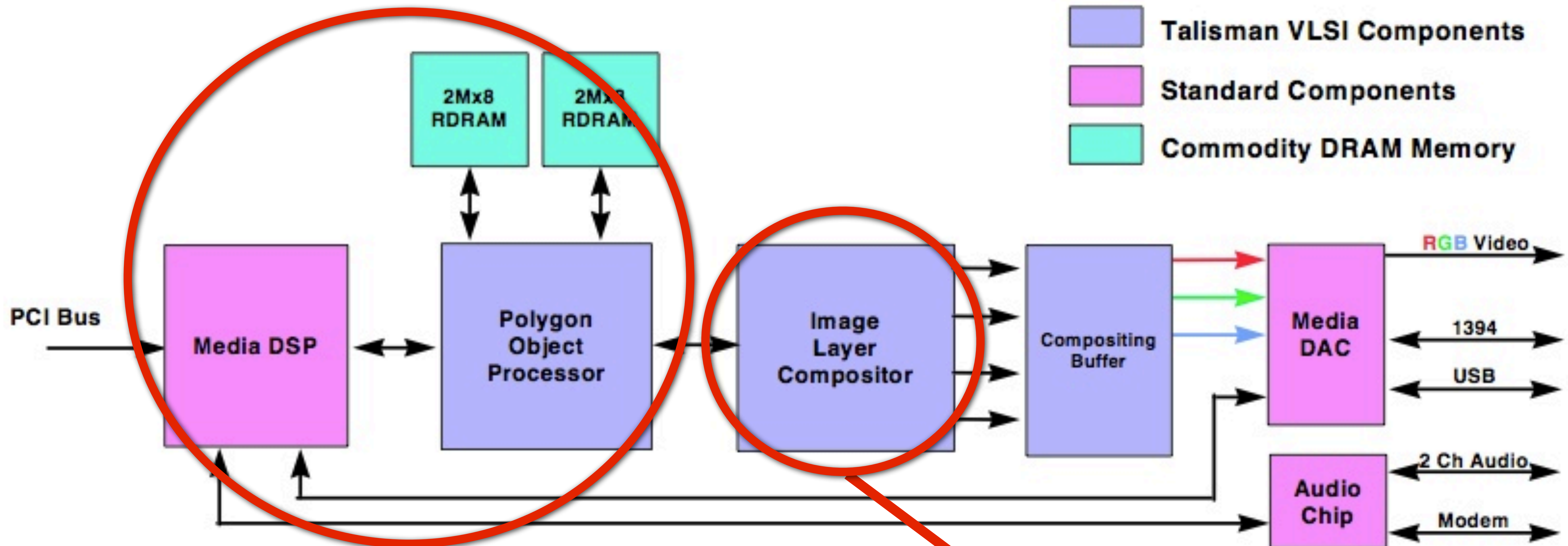
Prerendered Textures

Novel view of object synthesized
from rendering sprites

Microsoft Talisman

[Torborg and Kajiya 96]

Proposed GPU designed to accelerate image-based rendering for interactive graphics (motivating idea: exploit frame-to-frame temporal locality by not rendering entire scene each frame)

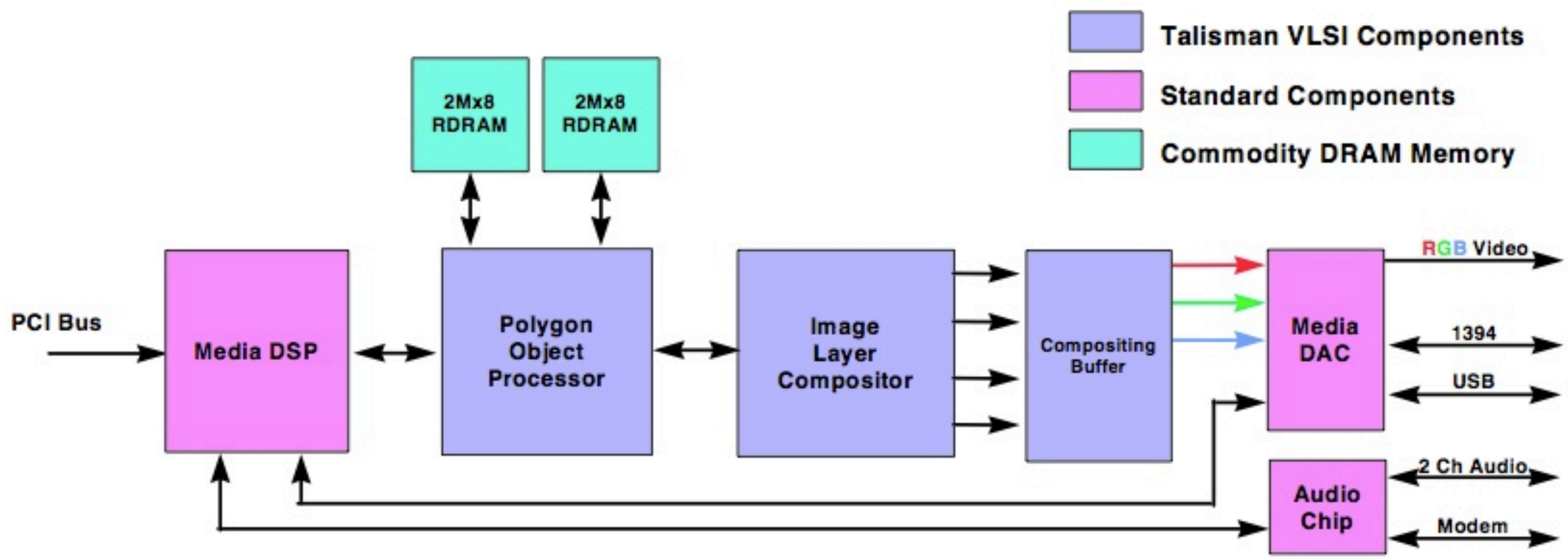


Implements image transform and image compositing operations

Implements traditional rendering operations (renders geometric models to images)

Microsoft Talisman

[Torborg and Kajiya 96]



Each object is rendered separately into its own image layer
(intent: high-quality renderings from 3D model, but not produced at real-time rates)

Image layer compositor runs at real-time rates
As scene changes (camera/object motion, etc.), image layer compositor transforms each layer accordingly, then composites image layers to produce complete frame

System detects when image warp likely to be inaccurate, makes request to re-render layer

Image-based rendering in interactive graphics systems

- Promise: render complex scenes efficiently by manipulating images

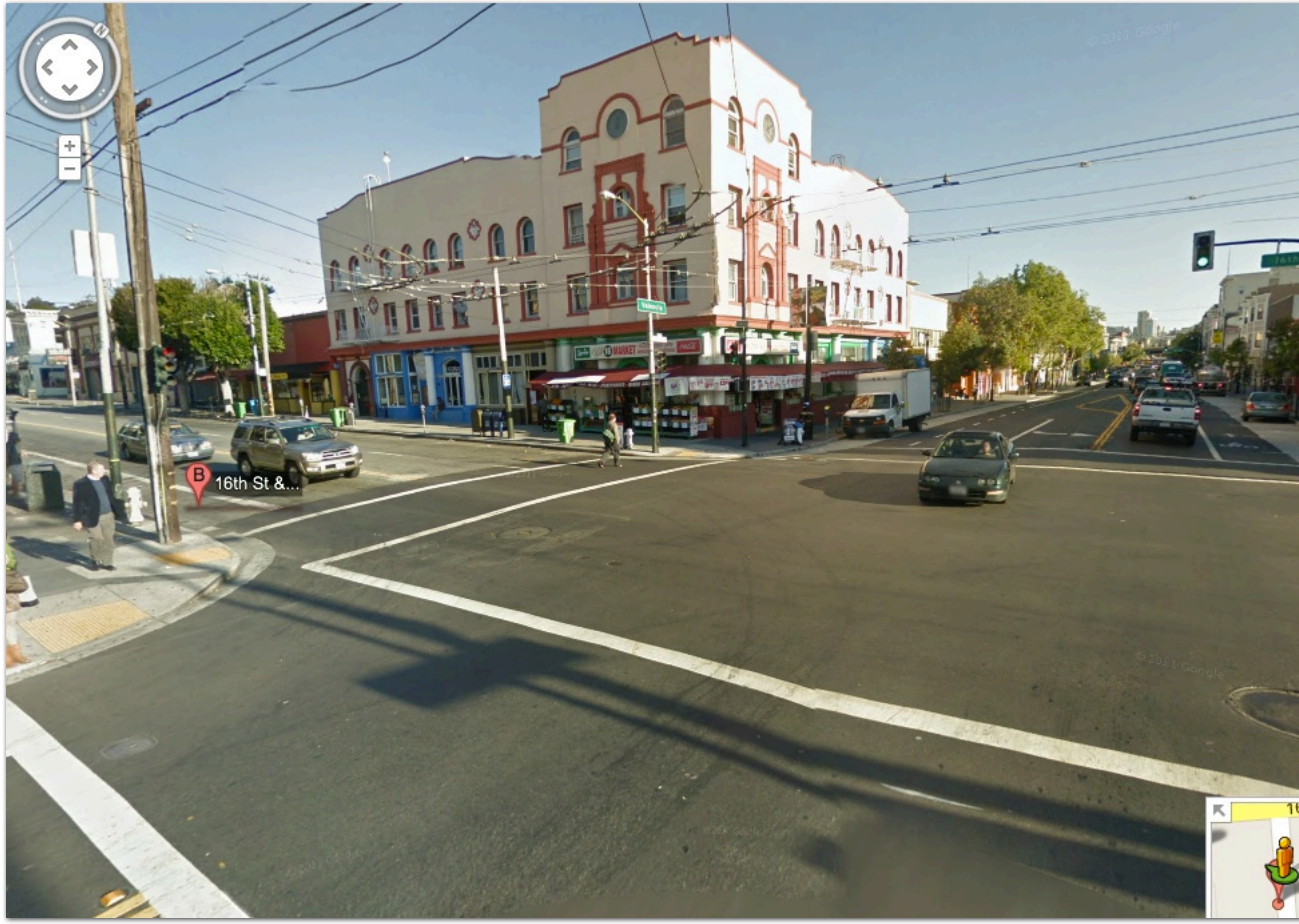
- Reality: never has been able to sufficiently overcome artifacts to be a viable replacement for rendering from a detailed, 3D scene description
  - Not feasible to prerender images for all possible scene configurations and views
  - Decades of research on how to minimize artifacts from missing information (intersection of graphics and vision: understanding what's in an image helps fill in missing information... and vision is unsolved)

Good system design:
efficiently meeting goals, subject to constraints

- **New application goals:**
  - Map the world
  - Navigate popular tourist destinations
  - Non-goal: virtual reality experience (artifact-free, real-time frame rate, viewer can navigate anywhere in the scene)

- **Changing constraints:**
  - Can't pre-render all scene configurations?
    - Ubiquity of cameras
    - Cloud-based graphics applications: enormous storage capacity
    - Bandwidth now available to access server-side capacity from clients

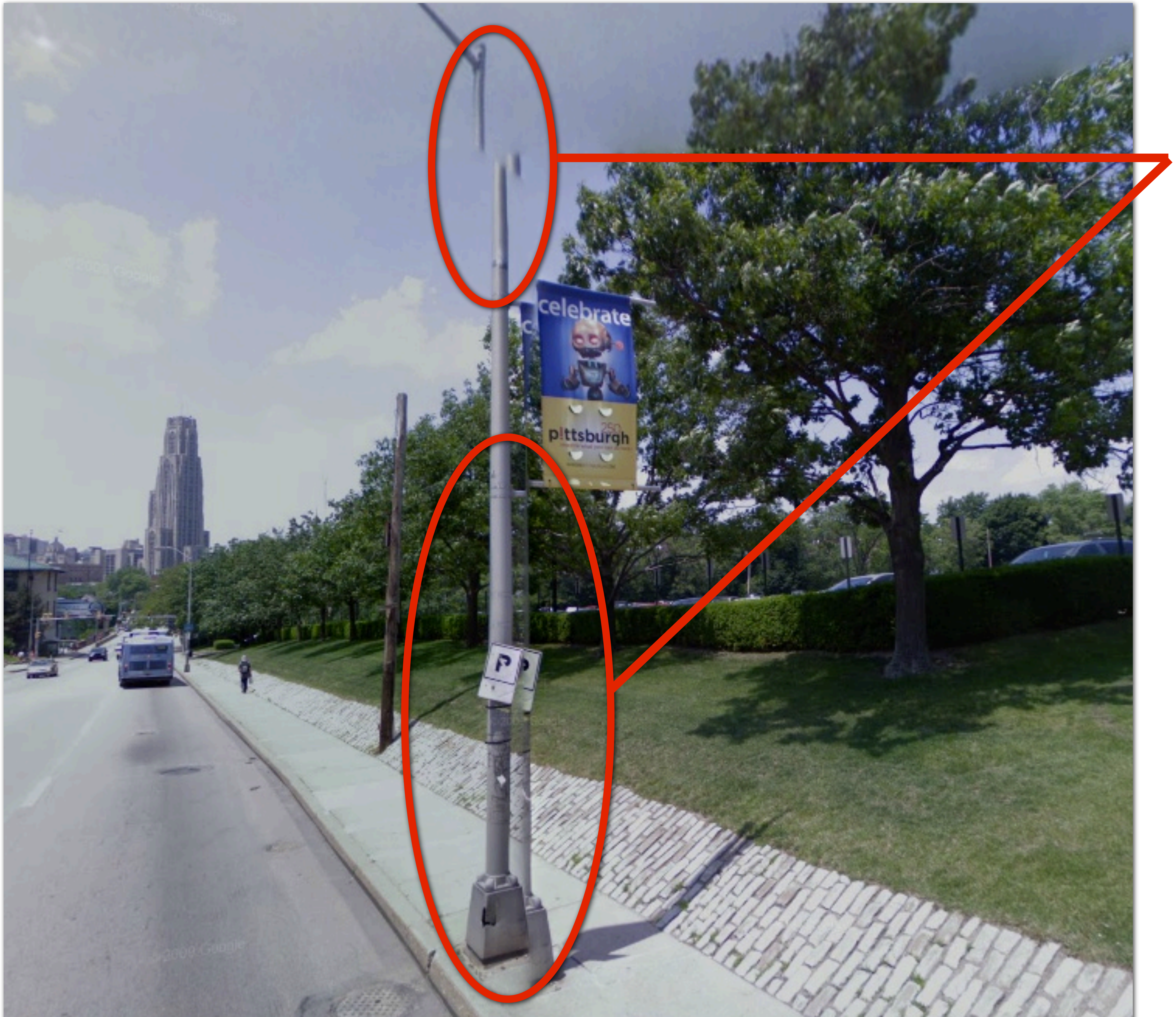
Google Street View



Goal: orient/familiarize myself with 16th and Valencia, San Francisco, CA

Imagine complexity of modeling and rendering this scene (and then doing it for all of the Mission, for all of San Francisco, of California, of the world...)

Google Street View



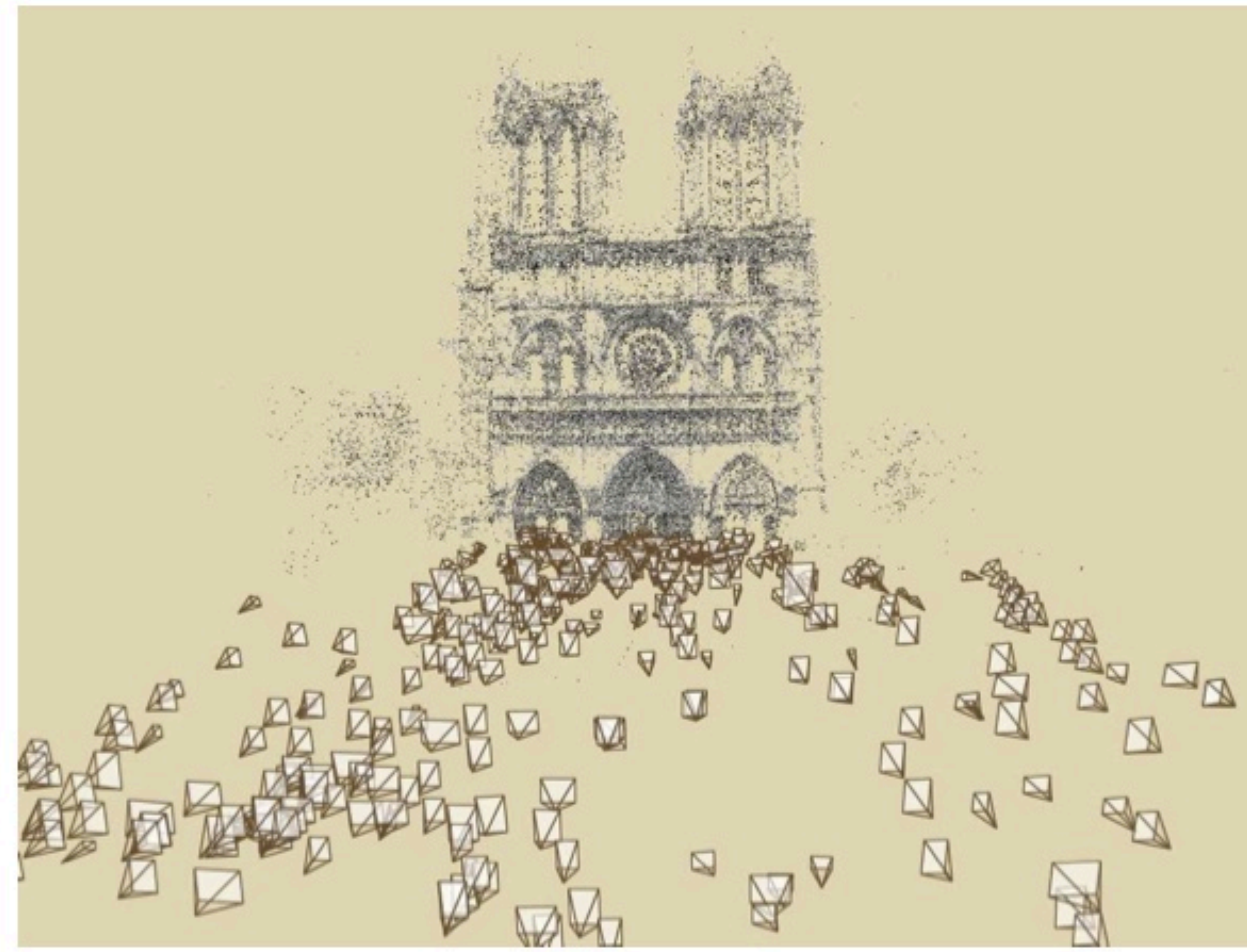
Imagine if your GPU produced images that had geometric artifacts like this!

Imagine if moving through a 3D rendered environment in a game had transitions like Google Maps

Photo-tourism (now Microsoft Photosynth)

[Snavely et al. 2006]

Input: collection of photos of the same scene



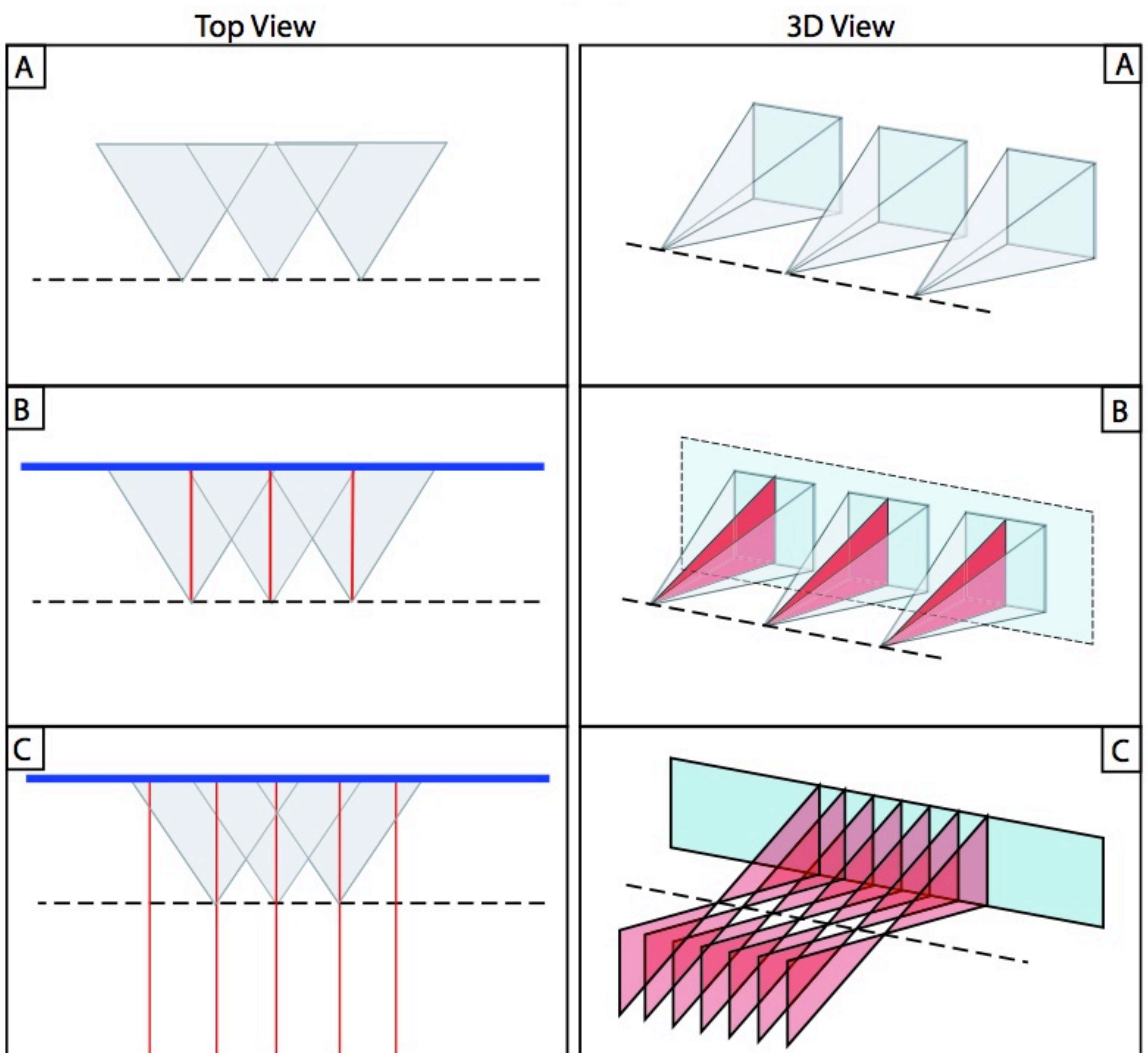
Output: sparse 3D representation of scene, 3D position of cameras for all photos

Goal: get a sense of what it's like to be at Notre Dame Cathedral in Paris

Alternative projections

Pushbroom projection

[Image credit: Roman 2006]



Each pixel column in image above is column of pixels from a different photograph

Result is orthographic projection in X, perspective projection in Y

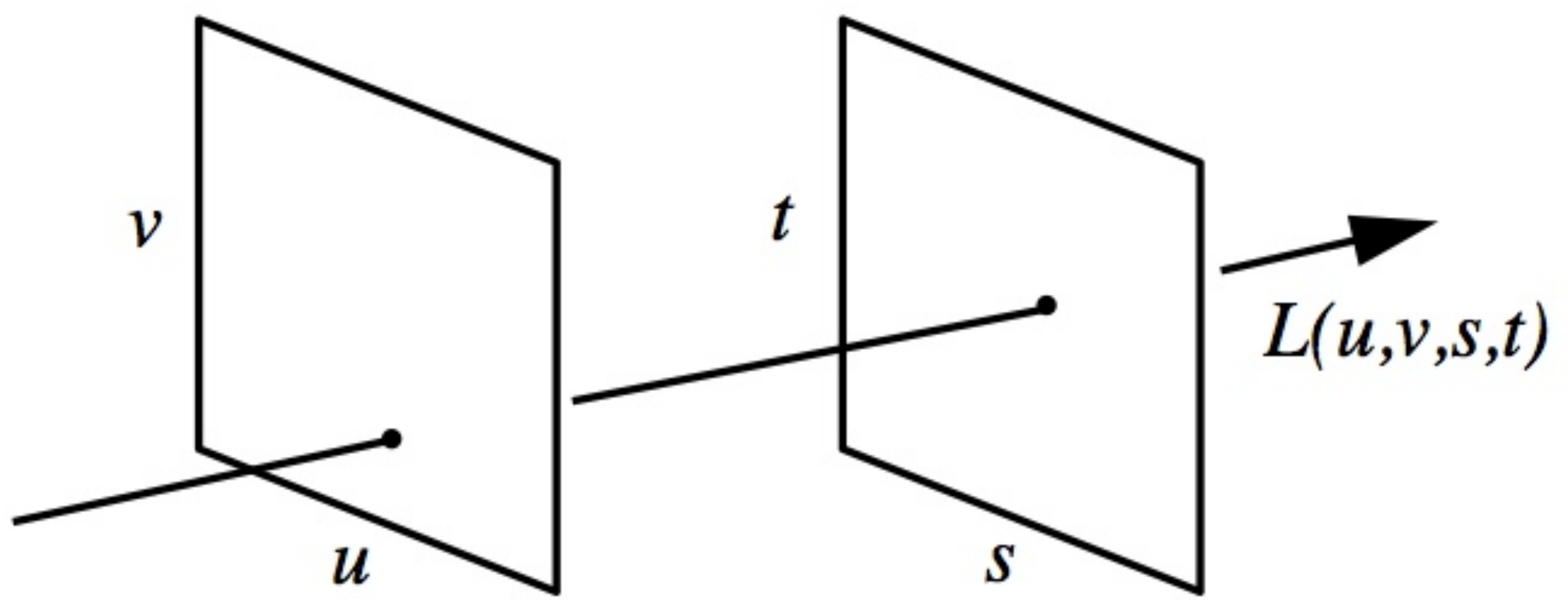
The Light Field

[Levoy and Hanrahan 96]

[Gortler et al., 96]

Light-field parameterization

Light field is a 4D function (represents light in free space: no occlusion)

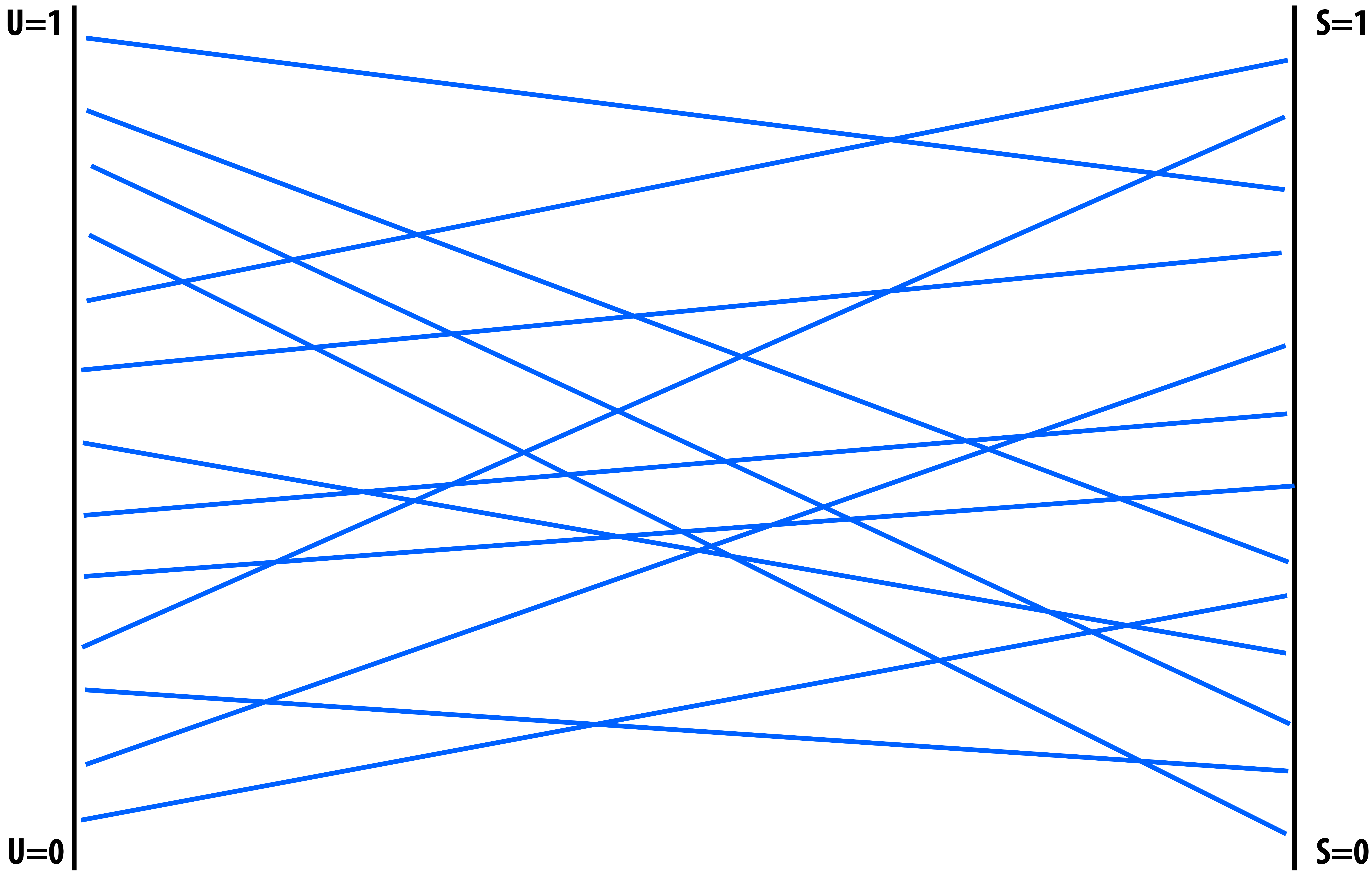


[Image credit: Levoy and Hanrahan 96]

- Efficient two-plane parameterization
- Line described by connecting point on (u,v) plane with point on (s,t) plane
- If one of the planes placed at infinity: point + direction representation

- Levoy/Hanrahan refer to representation as a “light slab”: beam of light entering one quadrilateral and exiting another

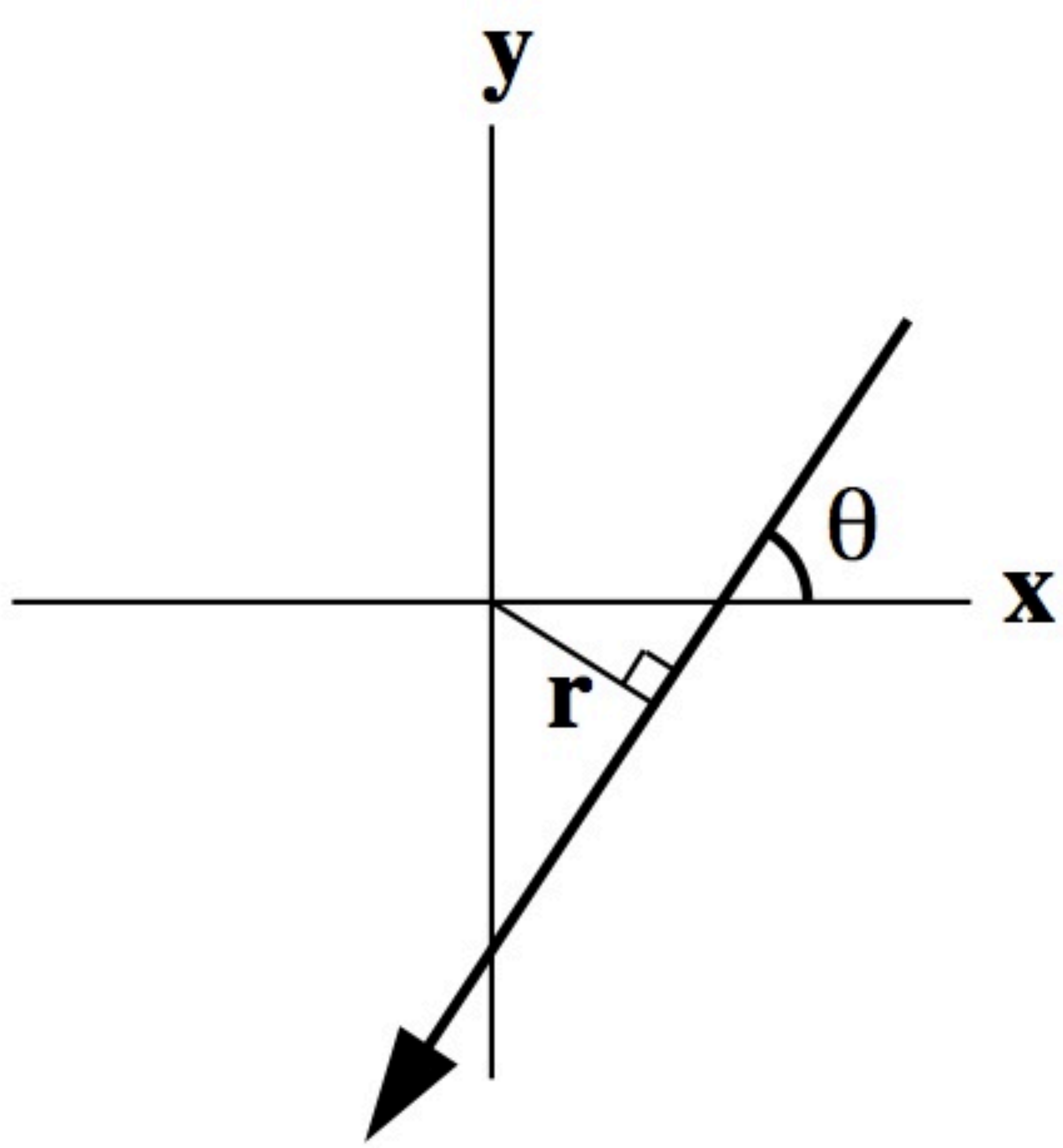
Sampling of the light field



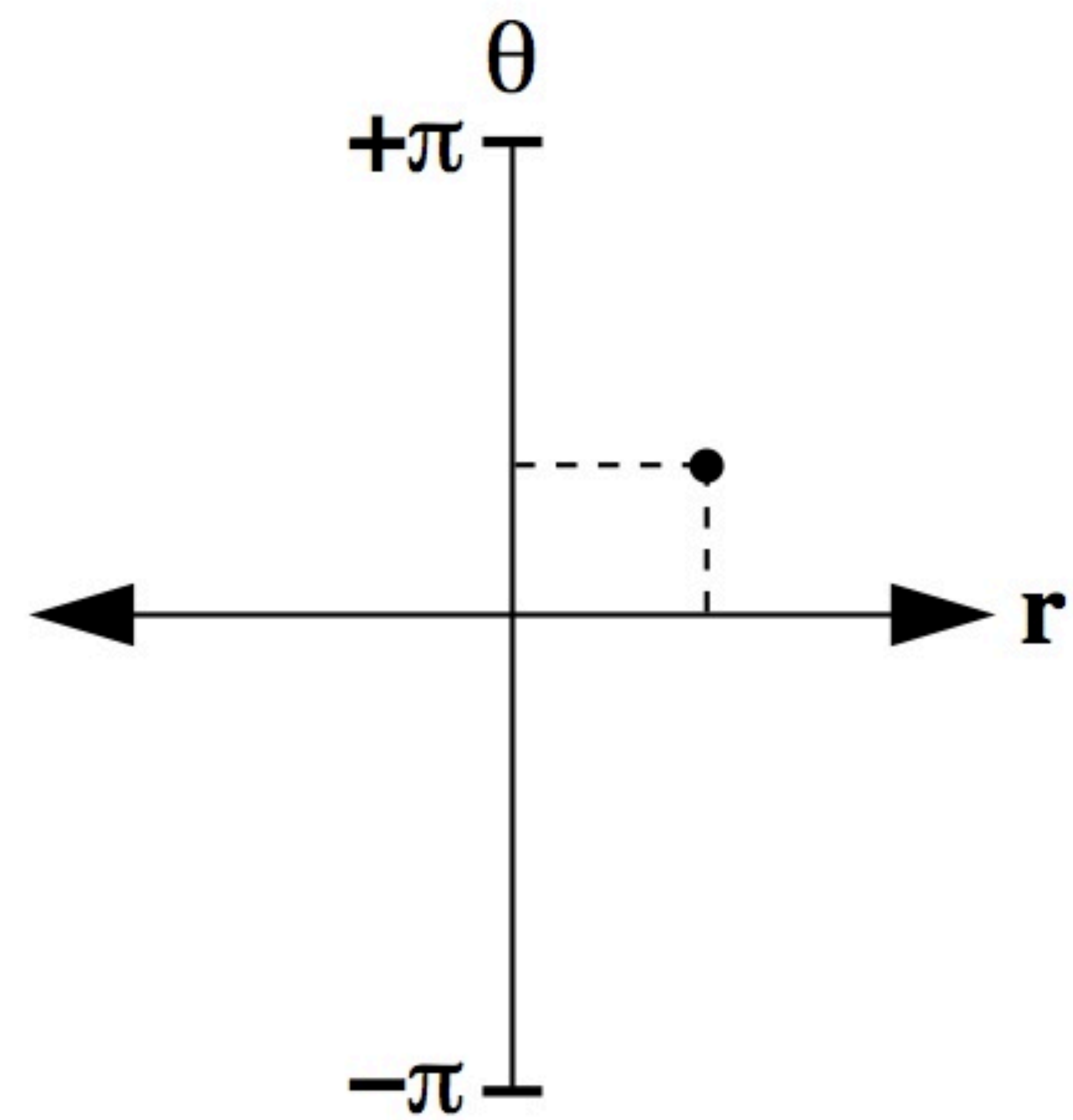
Simplification: only showing lines in 2D

Line-space representation

Each line in Cartesian space represented by a point in line space**



Cartesian space

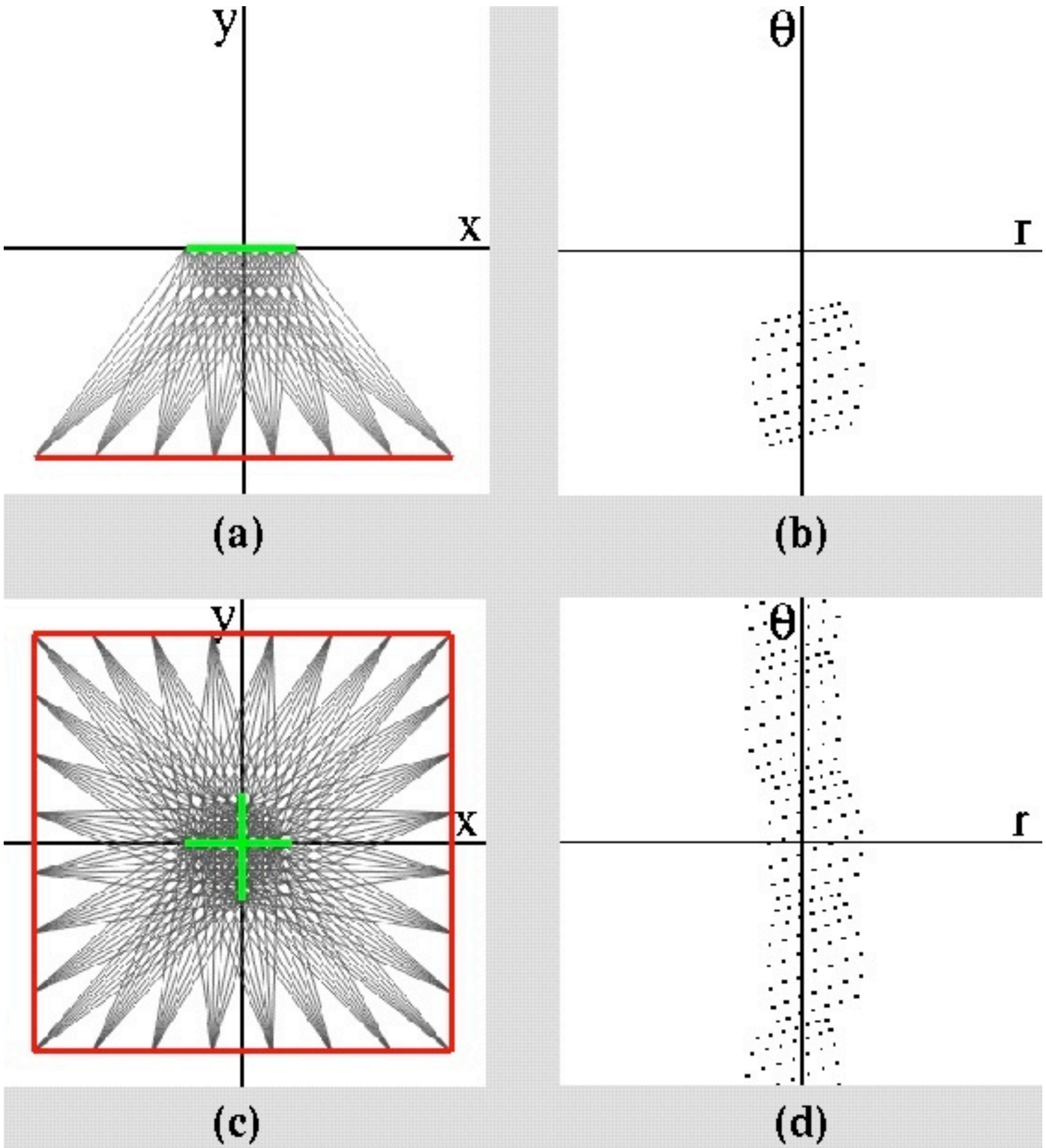


Line space

** Shown here in 2D, generalizes to 3D Cartesian lines

[Image credit: Levoy and Hanrahan 96]

Sampling lines



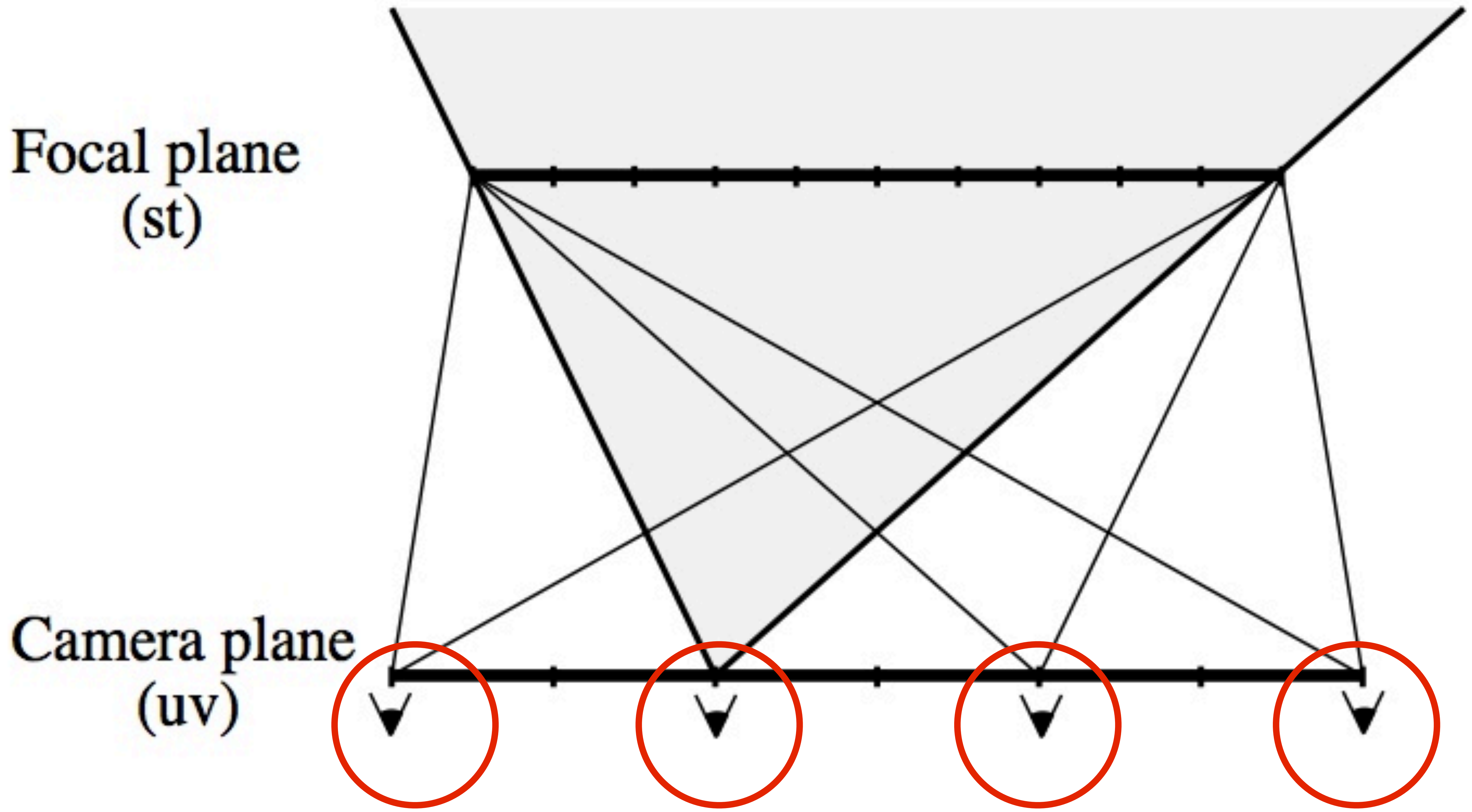
[Image credit: Levoy and Hanrahan 96]

To be able to reproduce all possible views, light field should uniformly sample all possible lines

Lines sampled by one slab

Four slabs sample lines in all directions

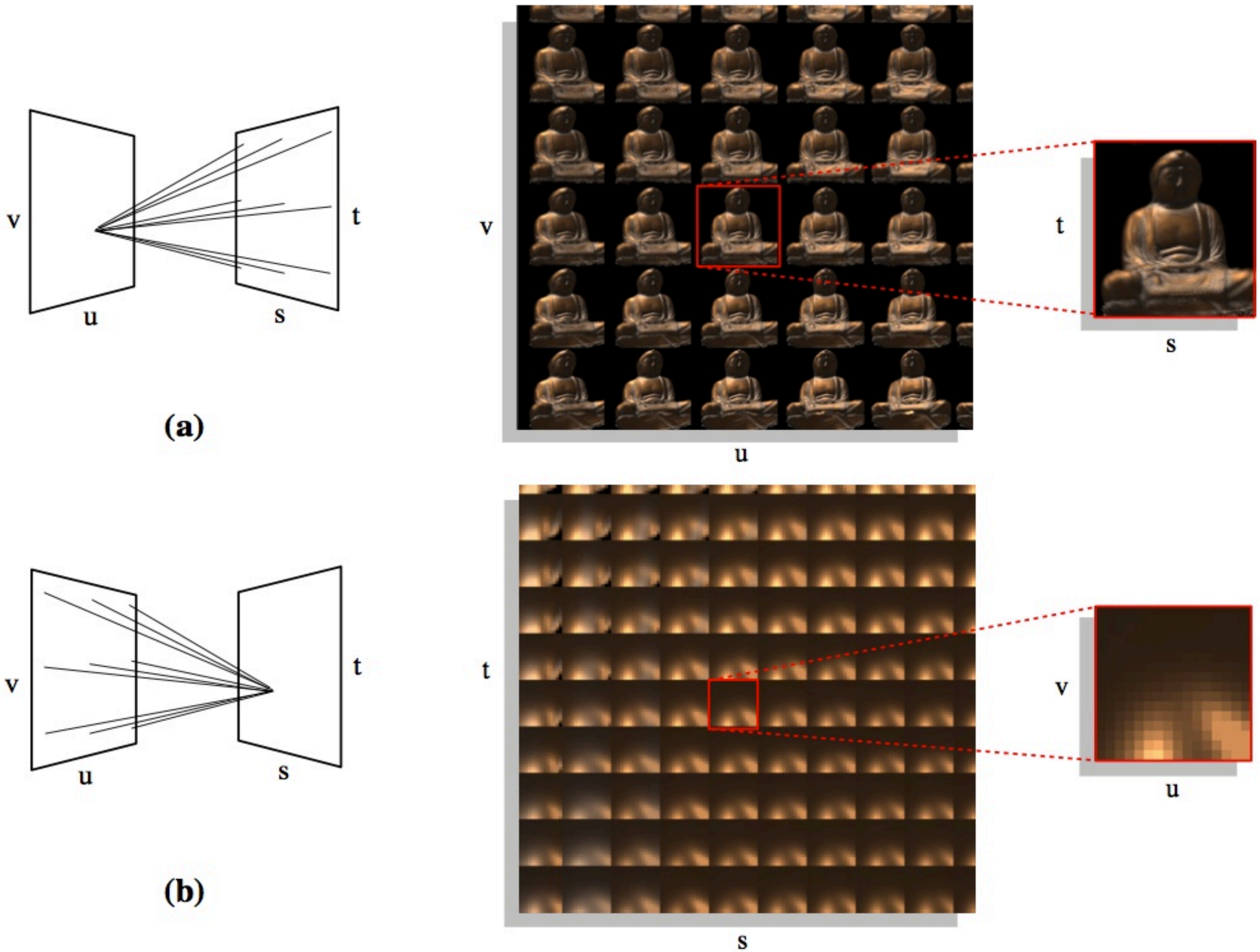
Acquiring a light field



Measuring light field by taking multiple photographs
(In this example: each photograph: constant UV)

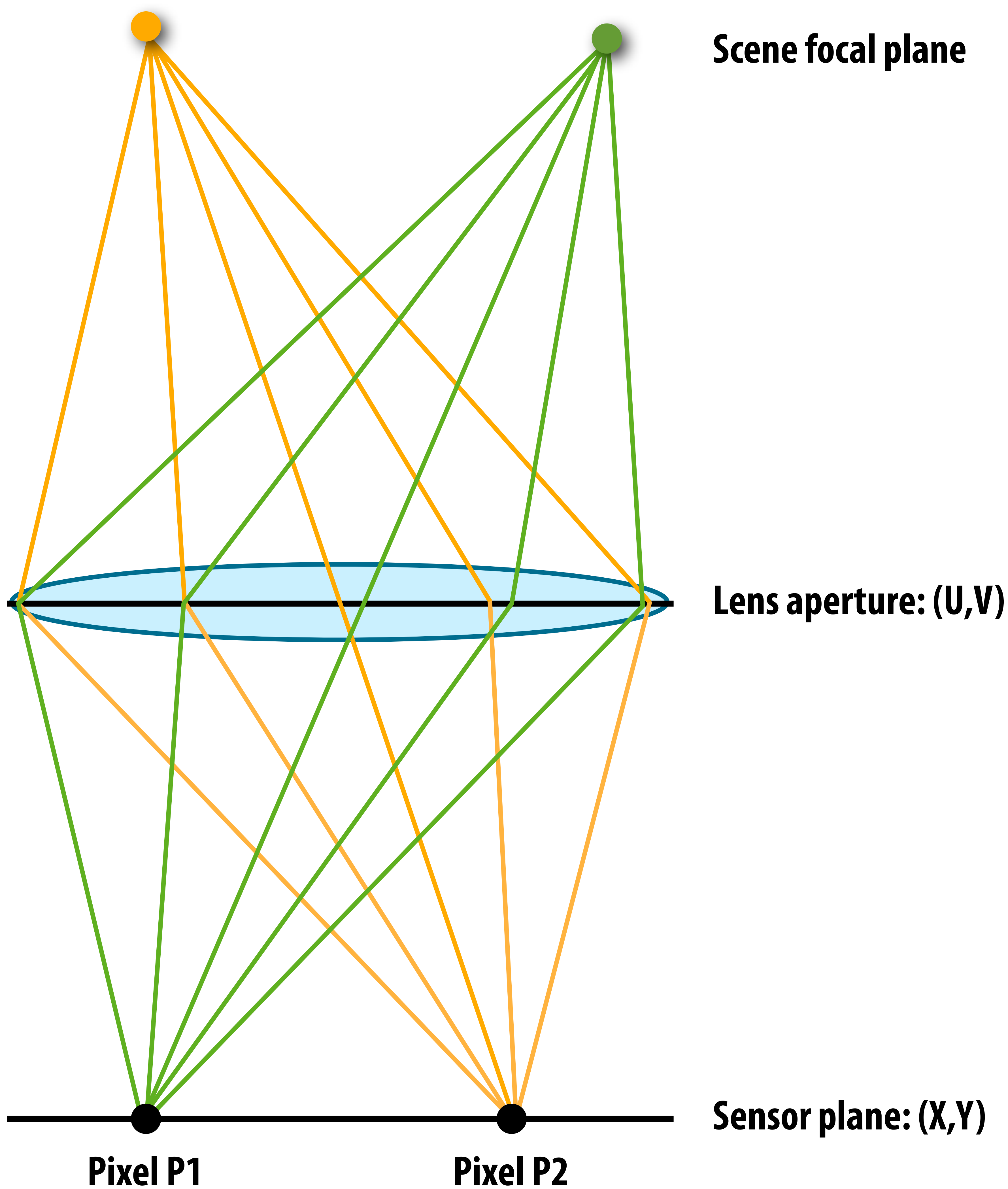
[Image credit: Levoy and Hanrahan 96]

Light field storage layouts

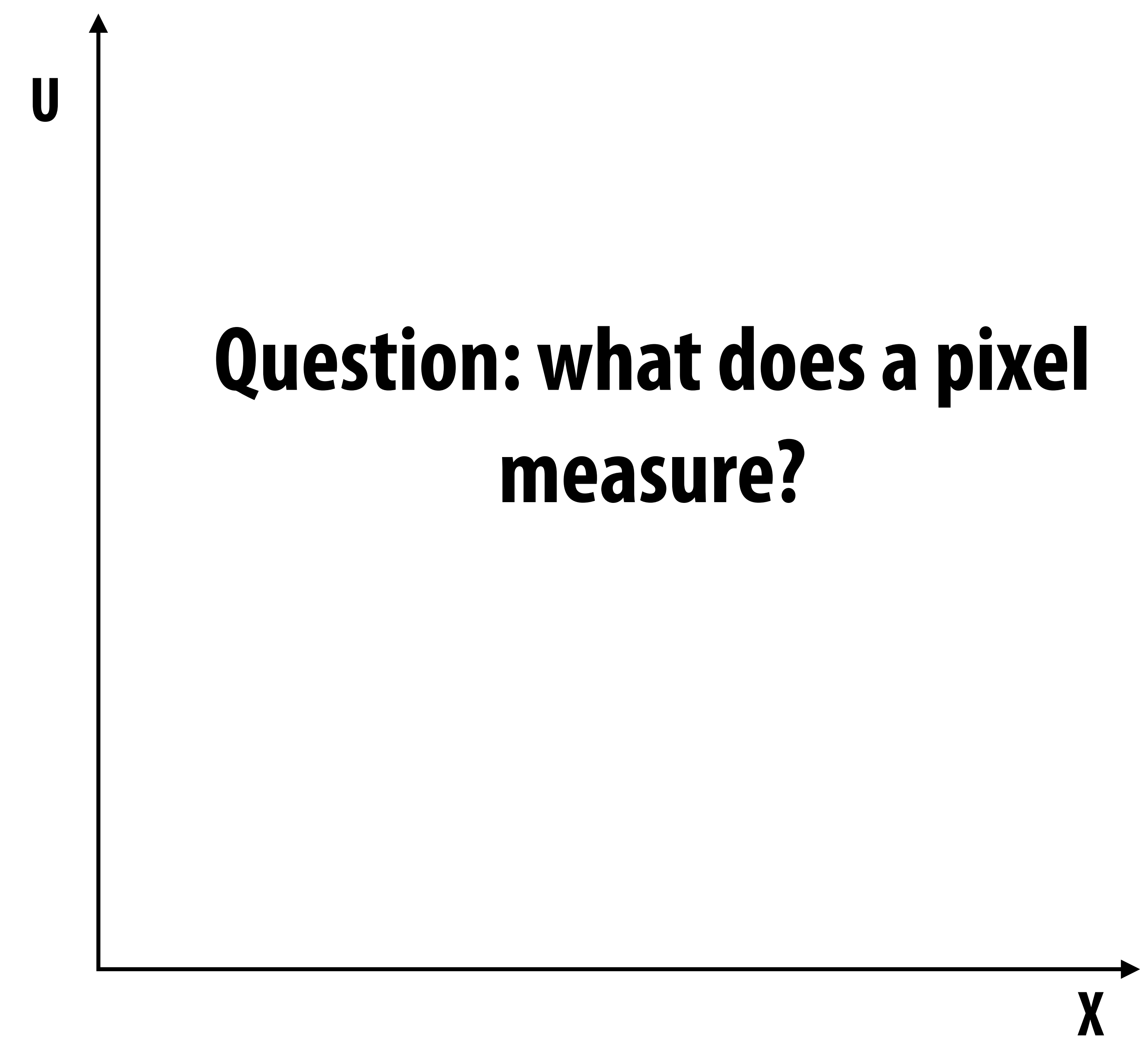


[Image credit: Levoy and Hanrahan 96]

Light field inside a camera

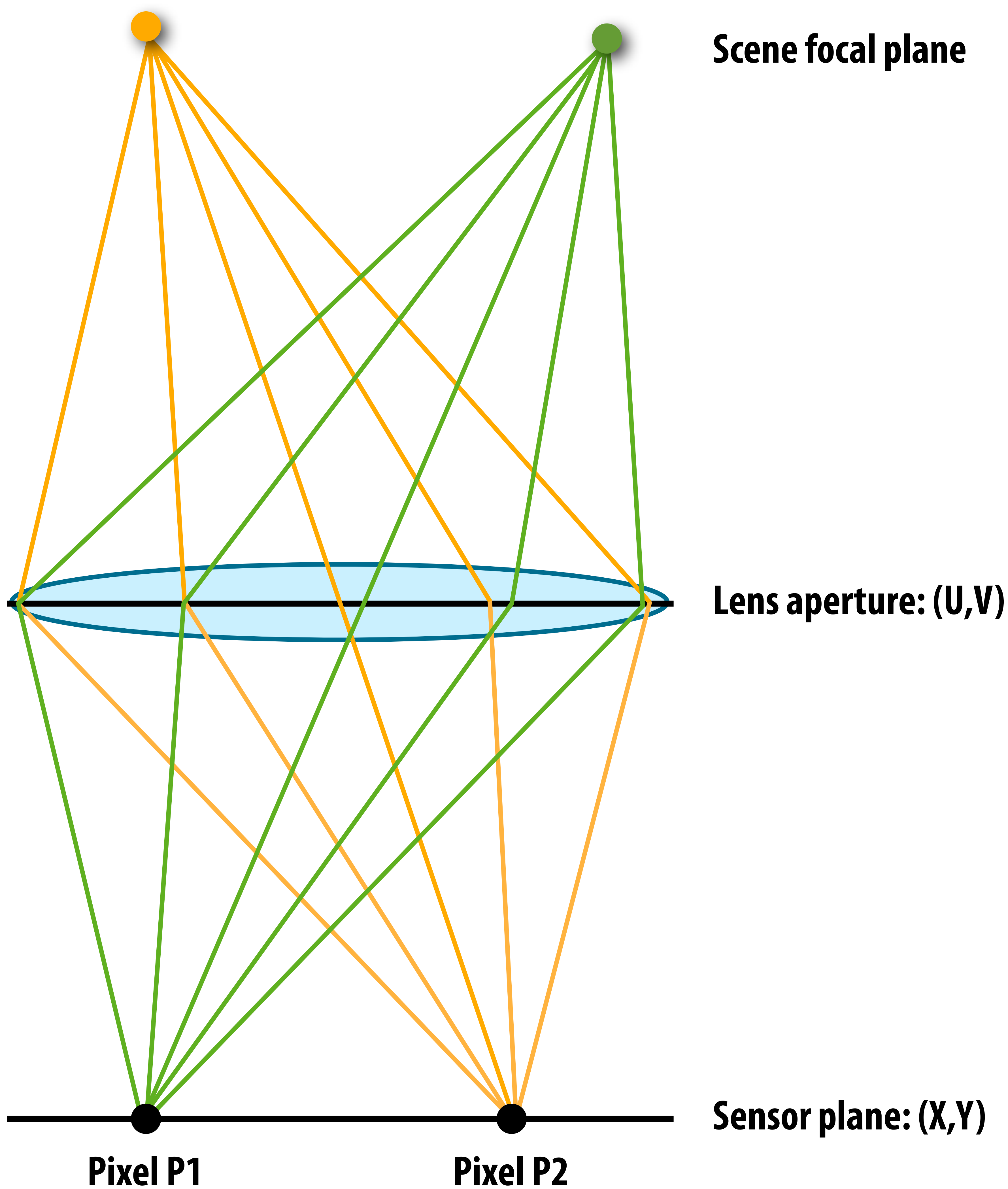


Ray space plot

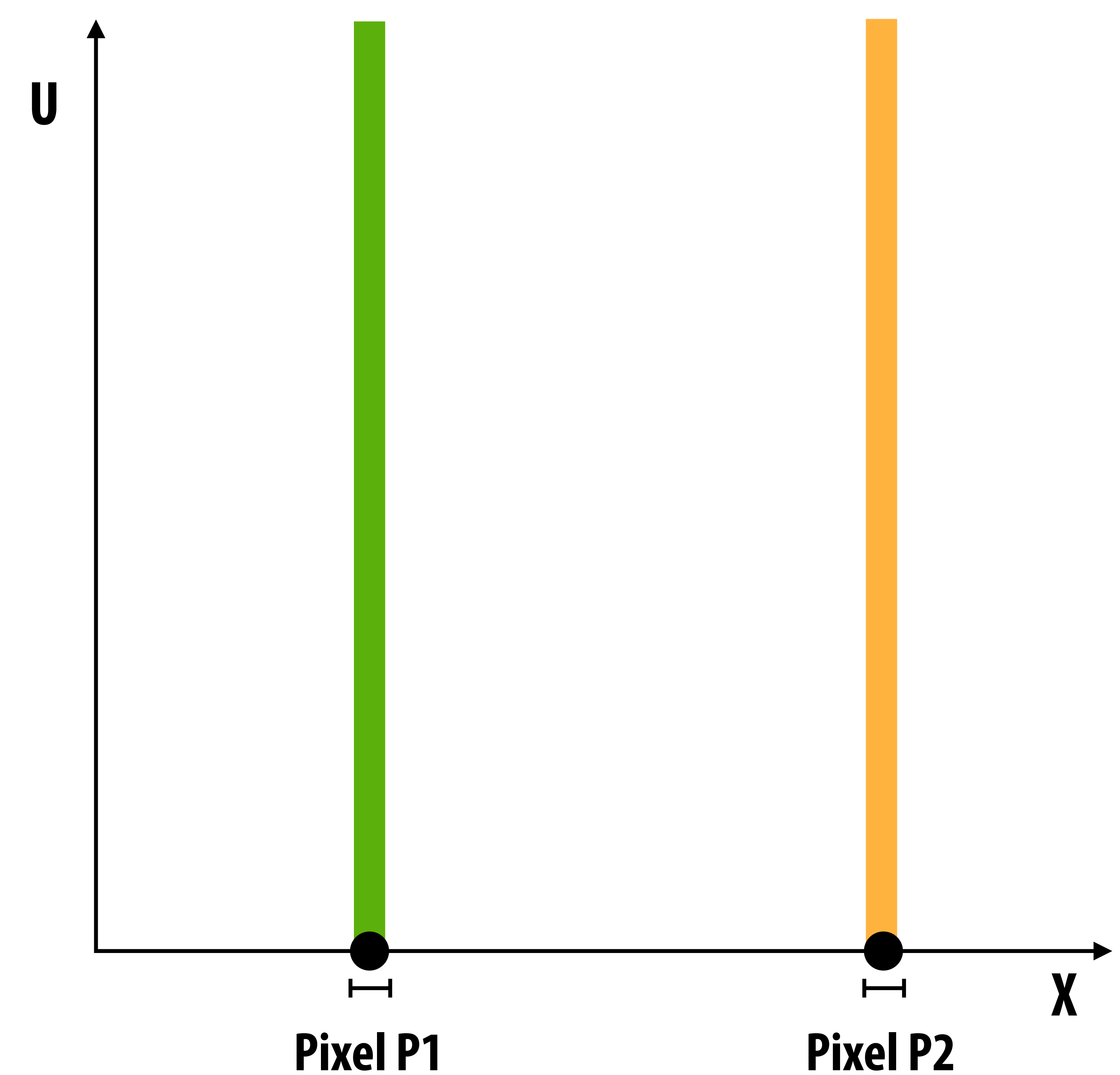


Question: what does a pixel measure?

Light field inside a camera



Ray space plot



Readings

- M. Levoy and P. Hanrahan. *Light Field Rendering*. SIGGRAPH 1996