Light Field Rendering

- by Marc Levoy and Pat Hanharan of Standford University, 1996.
- Goal: Generating new camera views at arbitrary positions by combining and resampling available images.
- Method: Interpretting input images as 2D slices of 4D function (the light field), assuming fixed illumination.

Mark Levoy

Pat Hanharan

Image-Based Rendering

- Concept: Generate views from previously acquired imagery.
- Examples: QuickTimeVR, Environment Mapping, Panoramas, etc.
- Advantage: Computationally inexpensive.
- Advantage: Rendering time independent of scene complexity.
- Advantage: Previously acquired imagery can be real or generated.
- Disadvantages: Limited by a fixed viewpoint.

- Problem: What can be done about this fixed viewpoint?
- Solution: Record radiance as a function of position and direction (Light Field).

Light-Field Rendering Goals

- Goal: Utilize 2D images (input) to generate light field.
- Goal: Utilize light field to generate different views (output).

Light-Field Rendering Challenges

- Challenge: Parameterization and representation of the Light Field.
- Challenge: Sampling pattern of the Light Field.
- Challenge: Compression of the Light Field data.
- Challenge: Fast generation of different views.

Light-Field Representation

- Question: What is a light field?
- Answer: Panoramic images at different 3D locations. (position and direction)
- Problem: Can this be reduced?
- Solution: Function on space-oriented lines (light slab).

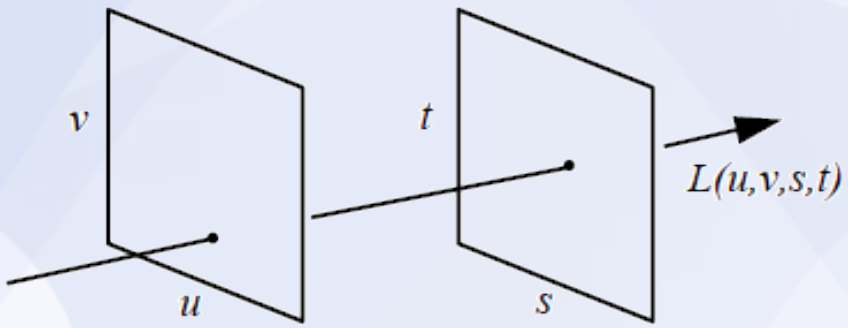


Figure 1: The light slab representation.

Light-Field Representation (cont.)

- Note: The lines going from (u,v) to (s,t) can be thought of as values in point space, where the color of the point is interpolated from (u,v) to (s,t).
- Note: Mapping from lines to points on a plane is a projective map, done with linear algebra (3x3 matrix).

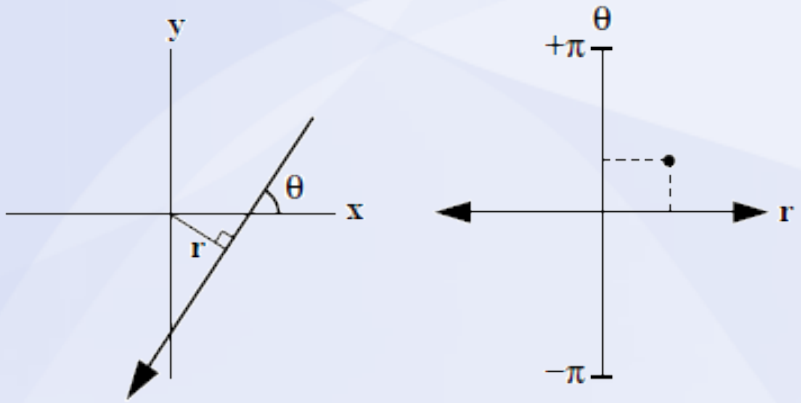
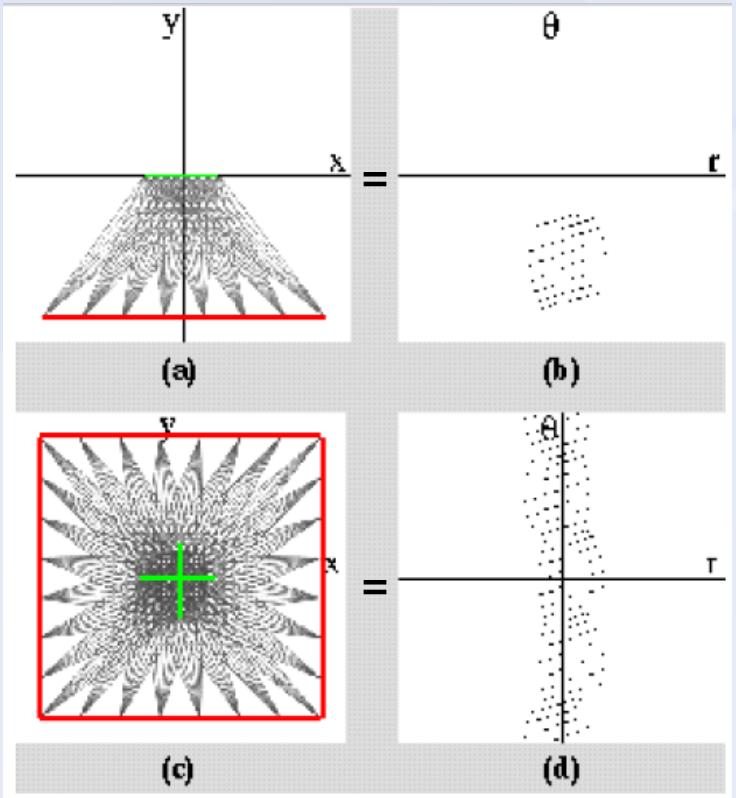
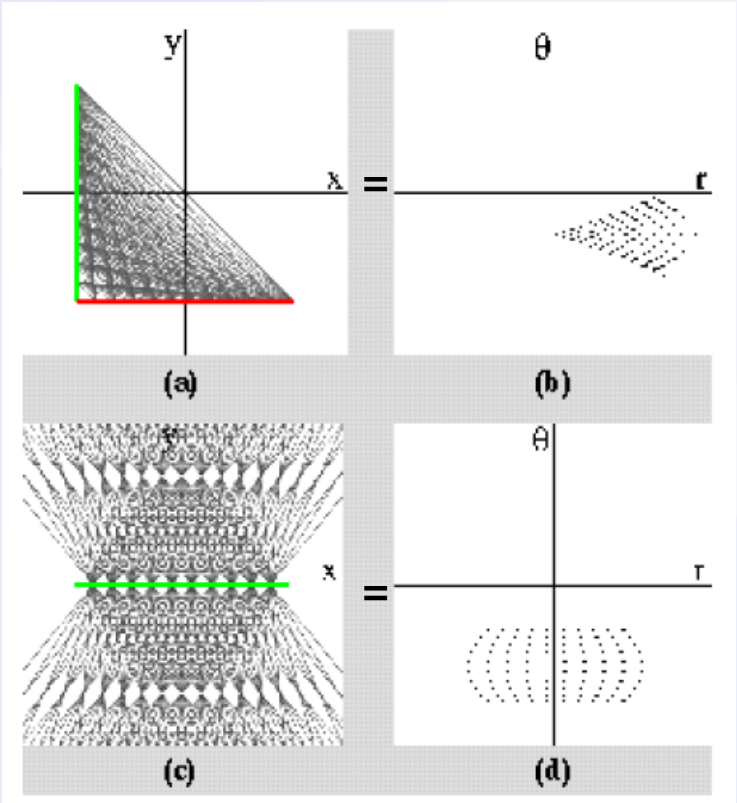


Figure 2: Definition of the line space we use to visualize sets of light rays. Each oriented line in Cartesian space (at left) is represented in line space (at right) by a point. To simplify the visualizations, we show only lines in 2D; the extension to 3D is straightforward.

Light-Field Sampling

- Goal: Evenly filling in the most points in point space.
- Question: Which of these sampling patterns is better?





Light-Field Input Data

- Method: Ray-tracing a scene (using rays as point data).
- Method: Light-slab from array of 2D images (at sheared perspectives).
  - (u, v) represents camera plane.
  - (s, t) represents focal plane.

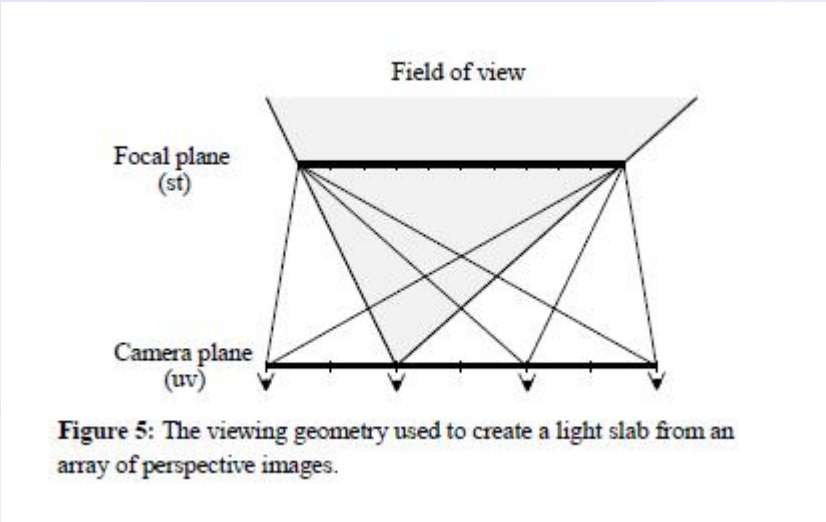
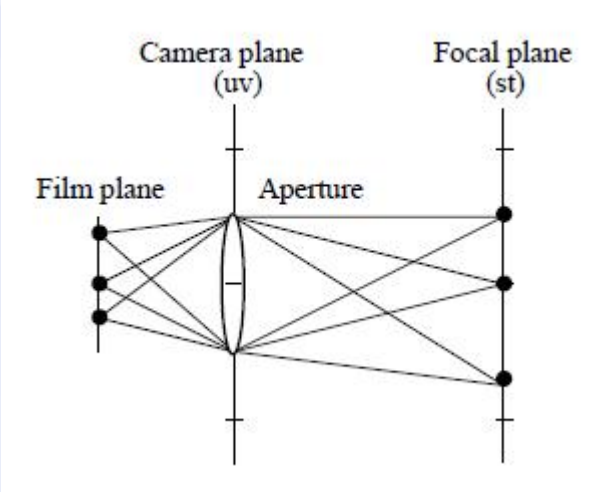
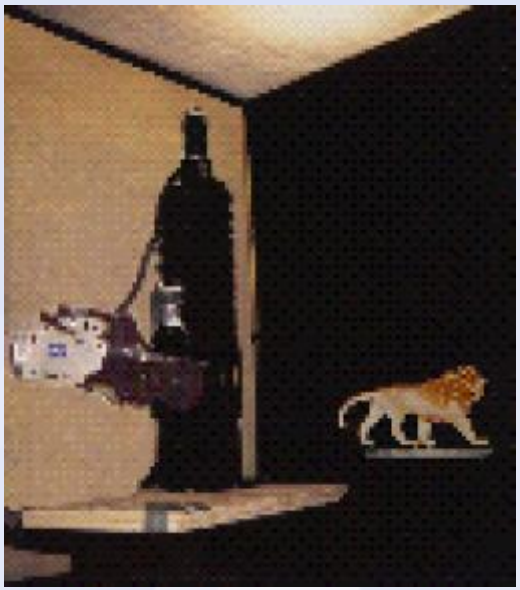


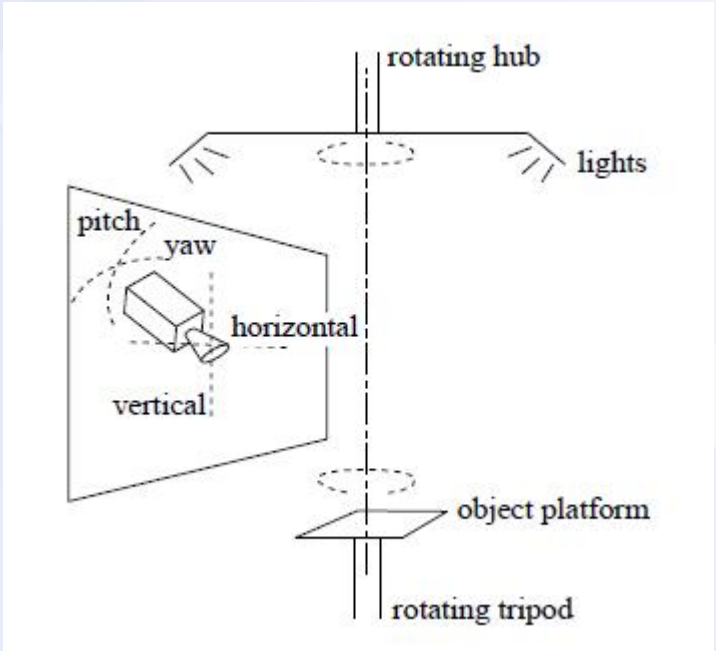
Figure 5: The viewing geometry used to create a light slab from an array of perspective images.



Light-Field Input Data (cont.)

- Method: Digitizing objects.
  - Requires knowledge of camera position.





Light-Field Data Compression

- Problem: Light-Fields are large (one example was 1.6 GB).
- Solution: Data Compression.
- Method: Vector Quantization + LZ Compression = 24:1 Compression

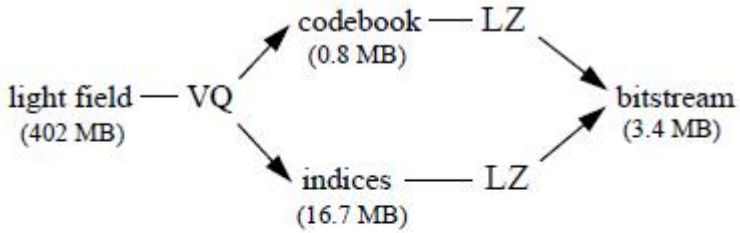


Figure 11 Two-stage compression pipeline. The light field is partitioned into tiles, which are encoded using vector quantization to form an array of codebook indices. The codebook and the array of indices are further compressed using Lempel-Ziv coding. Decompression also occurs in two stages: entropy decoding as the file is loaded into memory, and dequantization on demand during interactive viewing. Typical file sizes are shown beside each stage.

Light-Field Display

- Method: Ray Tracing from Camera position through Light-Field
  - Compute (u, v, s, t) parameters for each ray.
  - Resample the radiance at those line parameters.

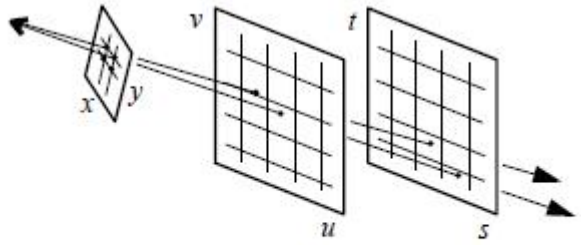


Figure 12: The process of resampling a light slab during display.

- Question: Can this be done using rasterization?
- Answer: Yes. Computing line coordinates can be done using texture mapping.

Light-Field Display (cont.)

- Method: Rasterization

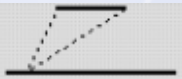
  1) The uv quadrilateral is drawn using the current viewing transformation.

  2) During scan conversion the (uw, vw, w) coordinates at the corners of the quadrilateral are interpolated.

  3) The resulting $u = uw/w$ and $v = vw/w$ coordinates at each pixel represent the ray intersection with the uv quadrilateral.

  4) Repeat 2-4 for st quadrilateral.

  5) Inverse transformation from (x, y) to (u, v, s, t) reduces to two texture coordinate calculations per ray.



(a) Buddha.

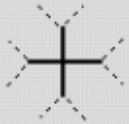
Input images: RenderMan renderings from an irregular polygon mesh model.

Light field: one slab, shown above in plan view. See also figure 3a.

(b) Kidney.

Input images: volume renderings from computed tomography data.

Light field: one slab, shown above. See also figure 4c.



(c) Hallway.

Input images: SGI RealityEngine renderings from a polygon model.

Light field: four slabs like figure 4c, arranged in a cross.

(d) Lion.

Input images: digitized video from a computer-controlled camera

Light field: four slabs arranged in a box. See also figure 3c.

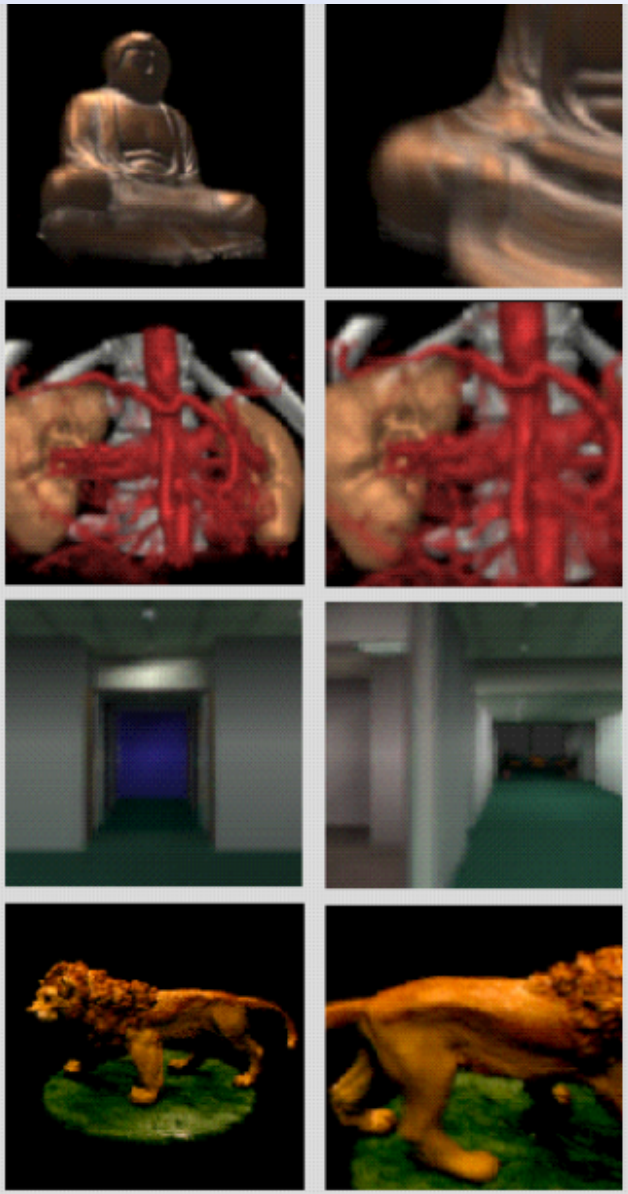


Figure 14: Example images from four light fields, extracted during a typical interactive viewing session.

Additional Information

- Wikipedia
- Stanford Light Field Archive
- LightPack

Questions?