Rendering of Spherical Light Fields

I. Ihm, S. Park, R. Lee
Dept. of Computer Sci.
Sogang Univ., Seoul, Korea

Image-Based Rendering

- generate a realistic image from pre-acquired imagery.
- can handle both real or virtual scene.
- rendering cost independent of scene complexity.
- generally cheaper than geometry-based rendering.
- fixed scene geometry and lighting.

- Previous work
  - View interpolation [Chen and Williams '93]
  - Quicktime VR [Chen '95]
  - Plenoptic modeling [McMillan '95]
  - Light fields [Levoy and Hanrahan '96]
  - Lumigraph [Gortler '96]
  - Hybric approach [Debevec '96]
  - Etc.

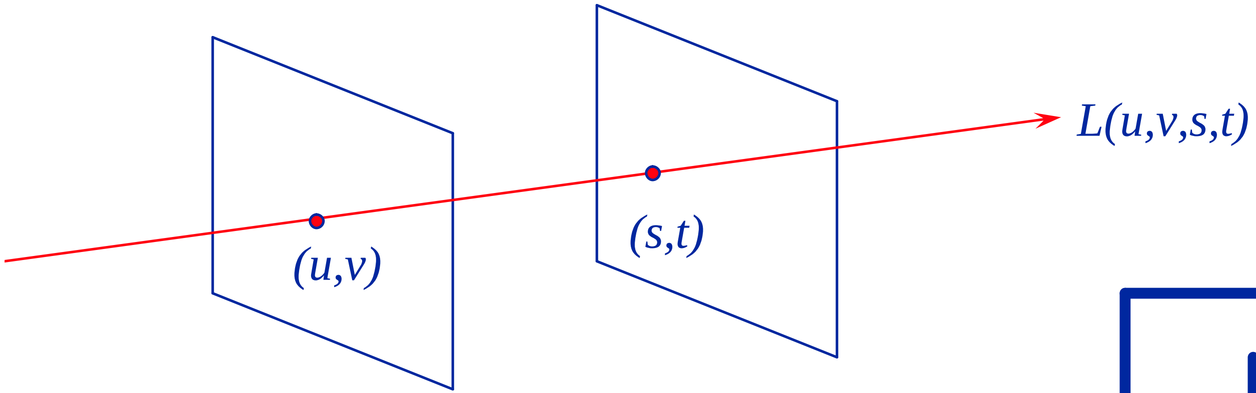
Main Contributions

- A new representation scheme for light flow: *Spherical Light Fields*
  - a sphere-based representation of plenoptic functions
  - an "object-space" rendering algo. easily embedded into a polygonal rendering system
- A new encoding scheme based on wavelets
  - provides a multi-resolution representation
  - can be adapted to other forms of light fields

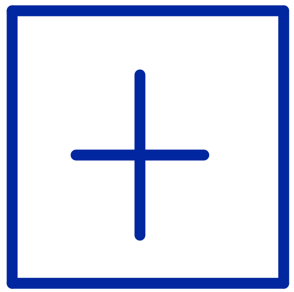
Spherical Light Fields Rendering

- Representation of Spherical Light Fields
- Discretization of Spherical Light Fields
- Polygonal Rendering of Spherical Light Fields

Light Fields [Levoy et al. '96]

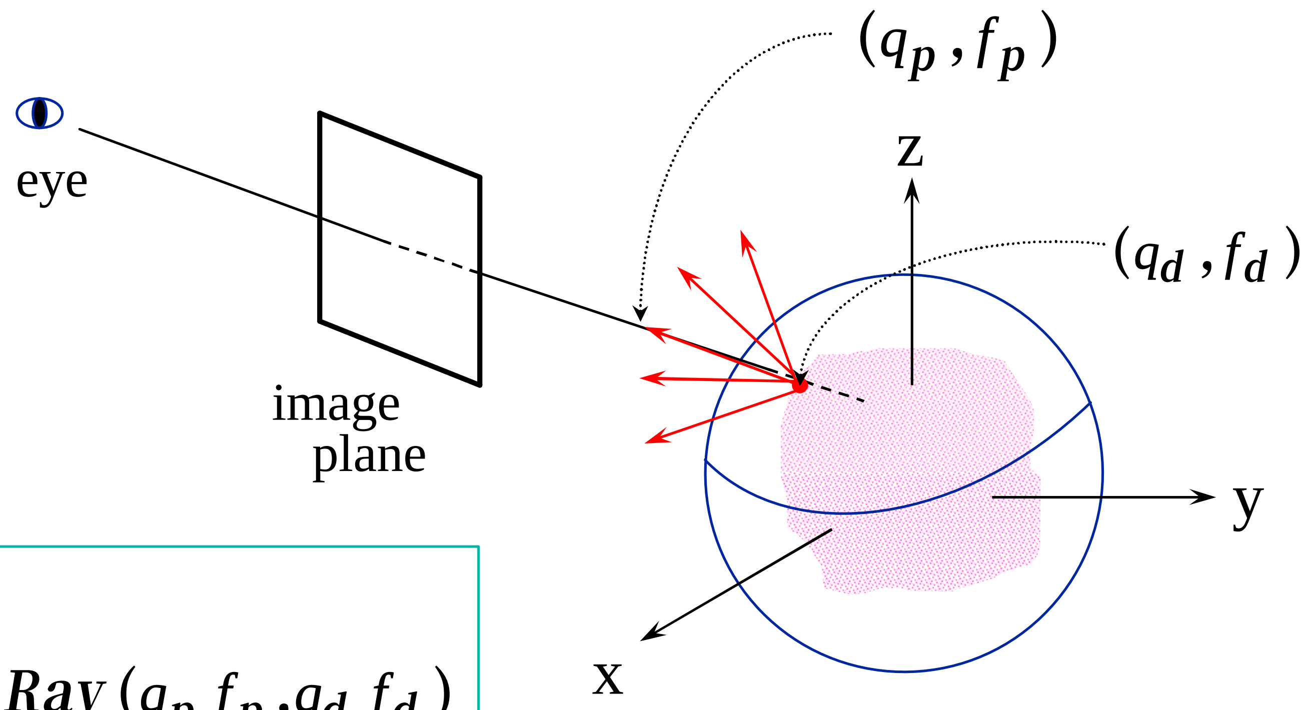


Light Slab



Example of Light Fields

4D Spherical Light Fields

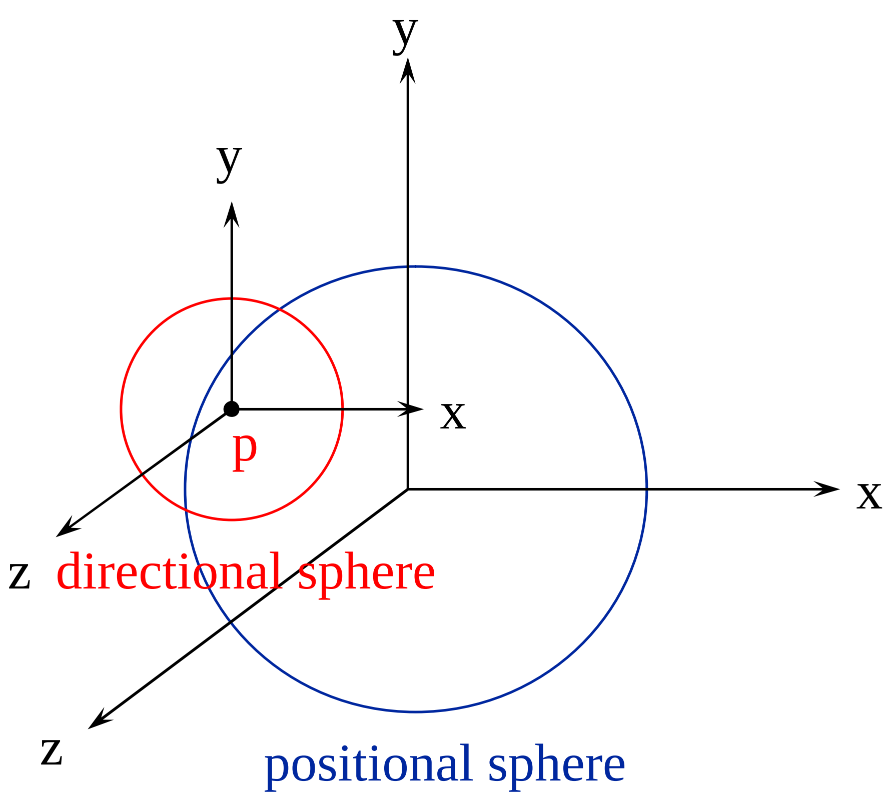


$$C \equiv \begin{pmatrix} R \\ G \\ B \end{pmatrix} = Ray\,(q_p, f_p, q_d, f_d)$$

Spherical Light Fields

Positional and Directional Spheres

- Separation of the 4D domain into two spheres
  - (q_p, f_p) : a point p on the positional sphere
  - (q_d, f_d) : a point on the directional sphere, centered at p
  - one pos. sphere and infinite number of dir. spheres



Decomposition of the function *Ray*

$$V = \{(q, f) \mid 0 \le q \le 2p, -\frac{p}{2} \le f \le \frac{p}{2}\}$$

$$f_d : V \to C$$

$$f_p : V \to (V \to C)$$

$$Ray\left(q_p, f_p, q_d, f_d\right) = (f_p(q_p, f_p)(q_d, f_d))$$

- f_p, f_d : two functions defined on the sphere
- f_p : the function value is a function defined on the sphere

Sampling of 4D Spherical Light Fields

- Sample *Ray* by sampling f_p, f_d on triangulated spheres
  - f_d (for the positional sphere) has function values at the vertices
  - f_d (for the directional sphere) has function values at the center of triangles

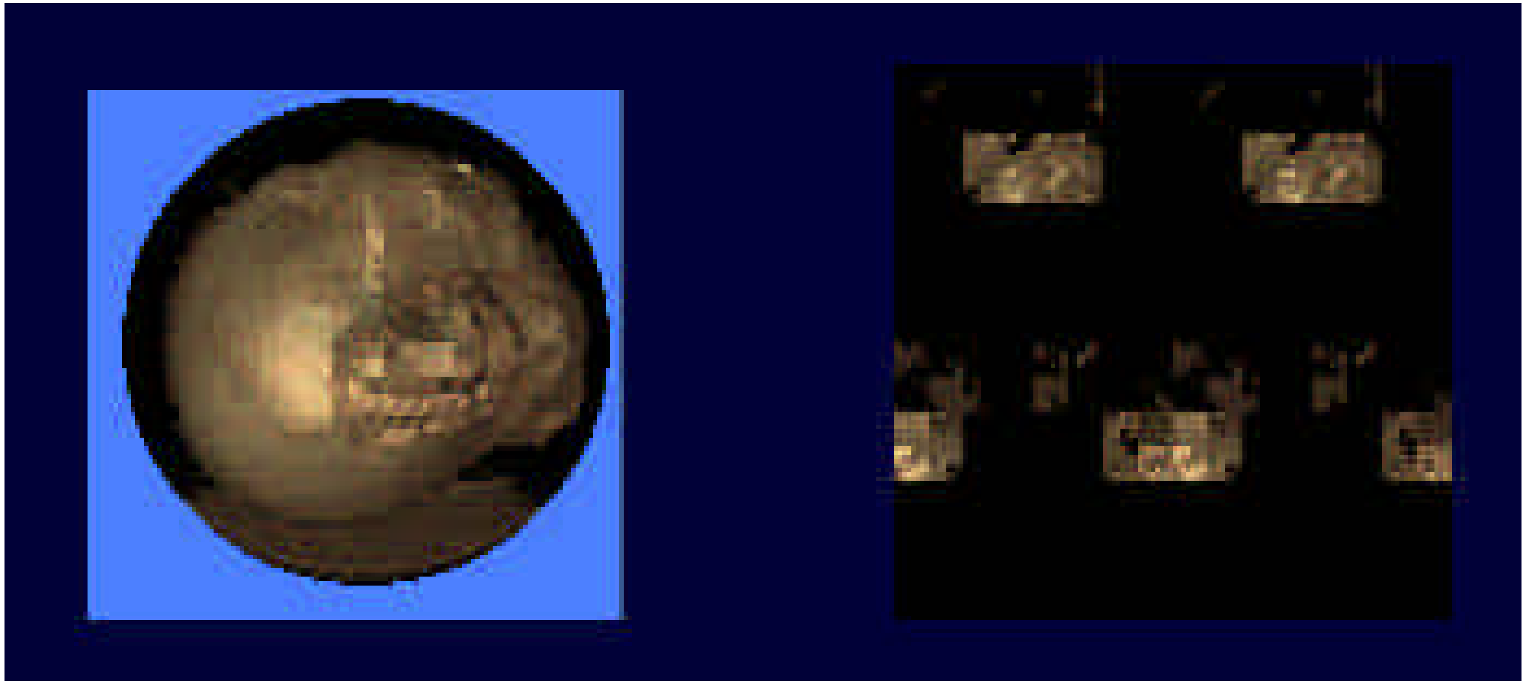
Tessellation of Directional Sphere

- Recursive subdivision from a base polyhedron

- Reordering of spherical data into 2D arrays

- Example: base polyhedron = octahedron, subdivision level = 5
  - Each base triangle: 1024 triangles, a 32X32 array
  - Each hemisphere: an 64X64 array
  - The whole sphere: an 128X64 array

  - an 128X128 array contains two directional spheres



Rendering from Spherical Light Fields

- Embed our image-based rendering process into the conventional polygon-based rendering system.

```
Determine the viewing parameters.
For each triangle of the positional
sphere,
    Cull it if it is back-faced.
    for each vertex of the triangle,
        Associate a proper color to it by evaluating
        Ray.
    Draw it using smooth shading (bilinear
    interpolation).
```

Wavelet-Based Compression

- Problems
  - A huge amount of storage is necessary for light flow ---> remove redundancy in data
    - example: 64K points for the positional sphere
      - lev. 5 sub. div. = 1.5 Gbytes
      - lev. 4 sub. div. = 384 Mbytes
  - Most of the previous 2D compression techniques are not well-suited to random access. ---> provide a low-cost random access to an individual data item
- Our approach: Design a wavelet-based compression scheme.

Previous Work

- Low-cost random access
  - Vector quantization (Levoy and Hanrahan '96)
    - a codebook and indices
    - up to 24:1 compression ratio
- Wavelet-based 2D image compression
  - zerotrees (Shapiro '93)
  - Crew (Zandi et al. '95)
  - Etc.
  - Focus only on compression ratio and image quality

Haar Wavelets

- Haar wavelet transform

$$c_L = (c_1 + c_2)/2 \quad , \quad c_H = (c_1 - c_2)/2$$

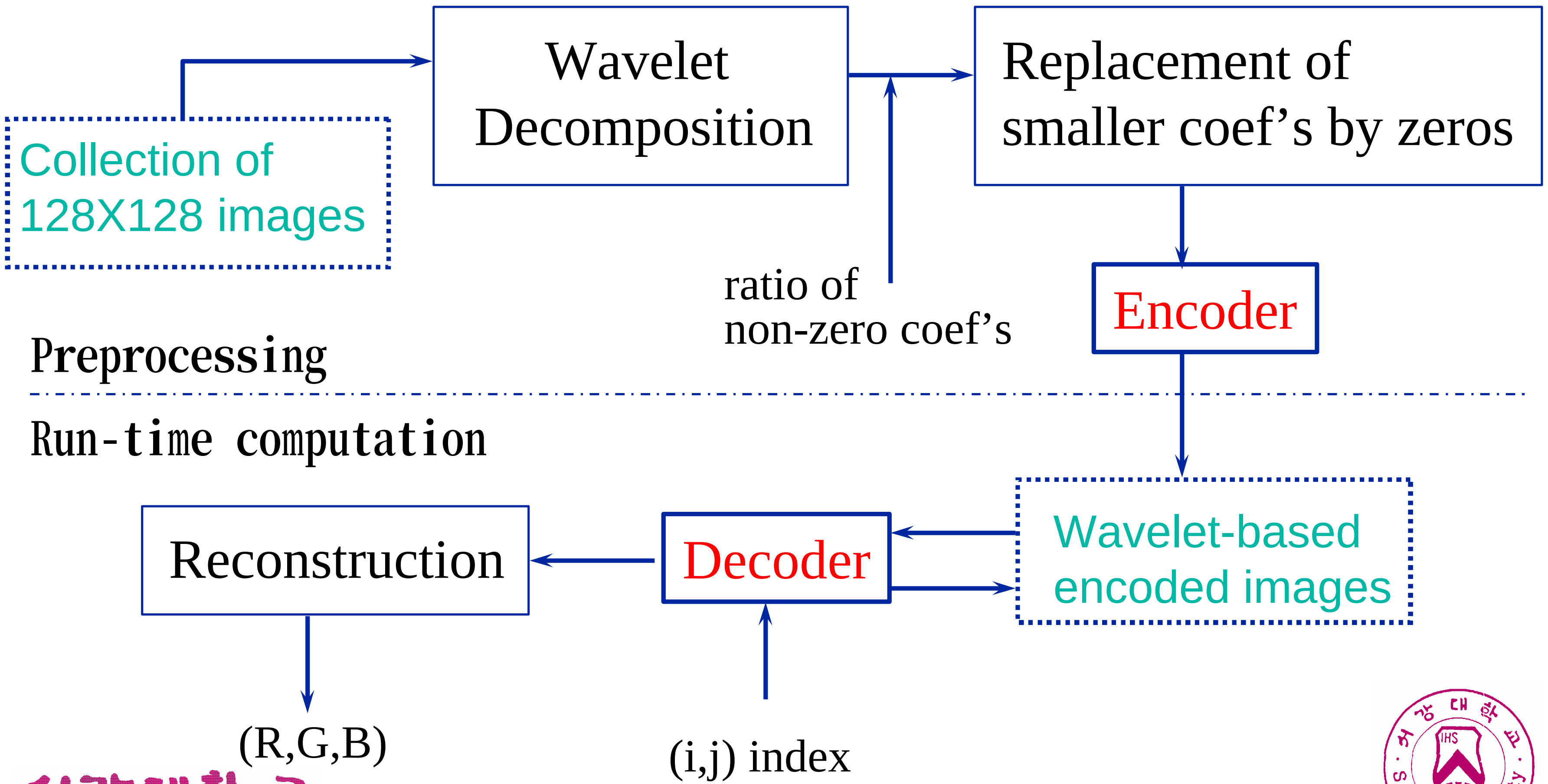
  - not the best filter, but computationally efficient

- The nonstandard decomposition for 2D image
  - separable application of vertical and horizontal filters

- Reconstruction
  - reverse the decomposition process.

- Compression
  - After wavelet transform, delete the coefficients with smaller magnitude.
  - can show this is the best choice for orthonormal bases under the L^2 norm

Our Compression Scheme

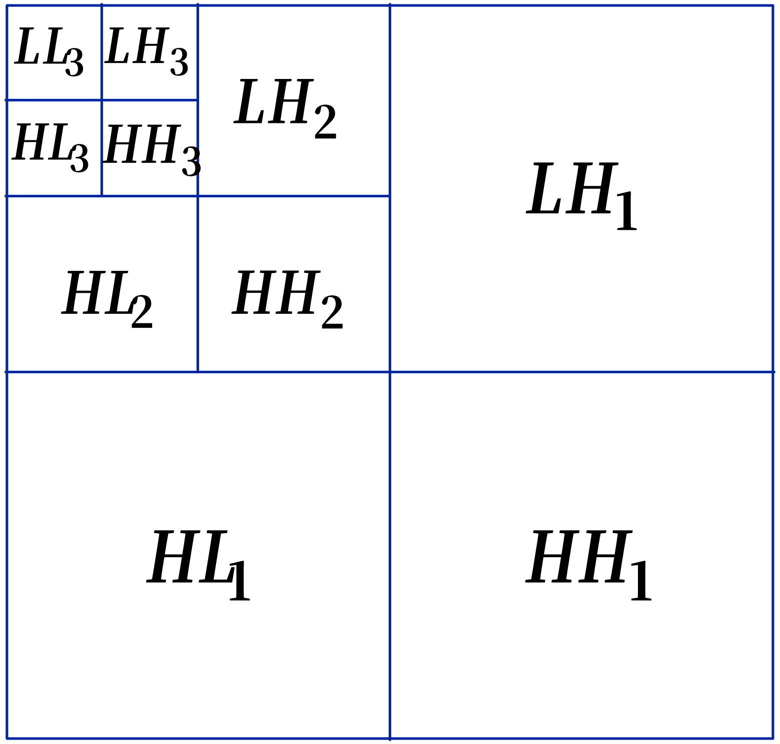


서강대학교 Sogang University

Encoding of Wavelet Coefficients

- For a 128X128 image, apply the Haar wavelet transform three times.
  - Use enough precision during decomposition, then quantize the coef's into 3 bytes.
  - Given a threshold, replace the smaller coef's by zeros.
- Now, we have an 'approximate' wavelet image whose coef's are mostly (say, 90%) zeros.

a 128X128 wavelet image



서강대학교
Sogang University

- Partition the 128X128 image into 256 16X16 subblocks.
- Tag the subblocks with integers: zero vs positive integers.
- Allocate additional memory for the significance map and offset info.
- Use a precomputed table for counting the # of 1 bits.

Decoding of Wavelet Coefficients

Input: an index (i,j) in an encoded image
Output: an RGB color

Identify the subblock S that contains (i,j).
If its tag is zero, return (0,0,0). (case A)
Compute the index (i',j') of (i,j) in S.
If the bit flag for (i',j'), return (0,0,0).
(case B)
Count the bit 1's in the sig. map before (i',j'), and compute the correct address using offset.
Access three-byte stream, and return the color. (case C)

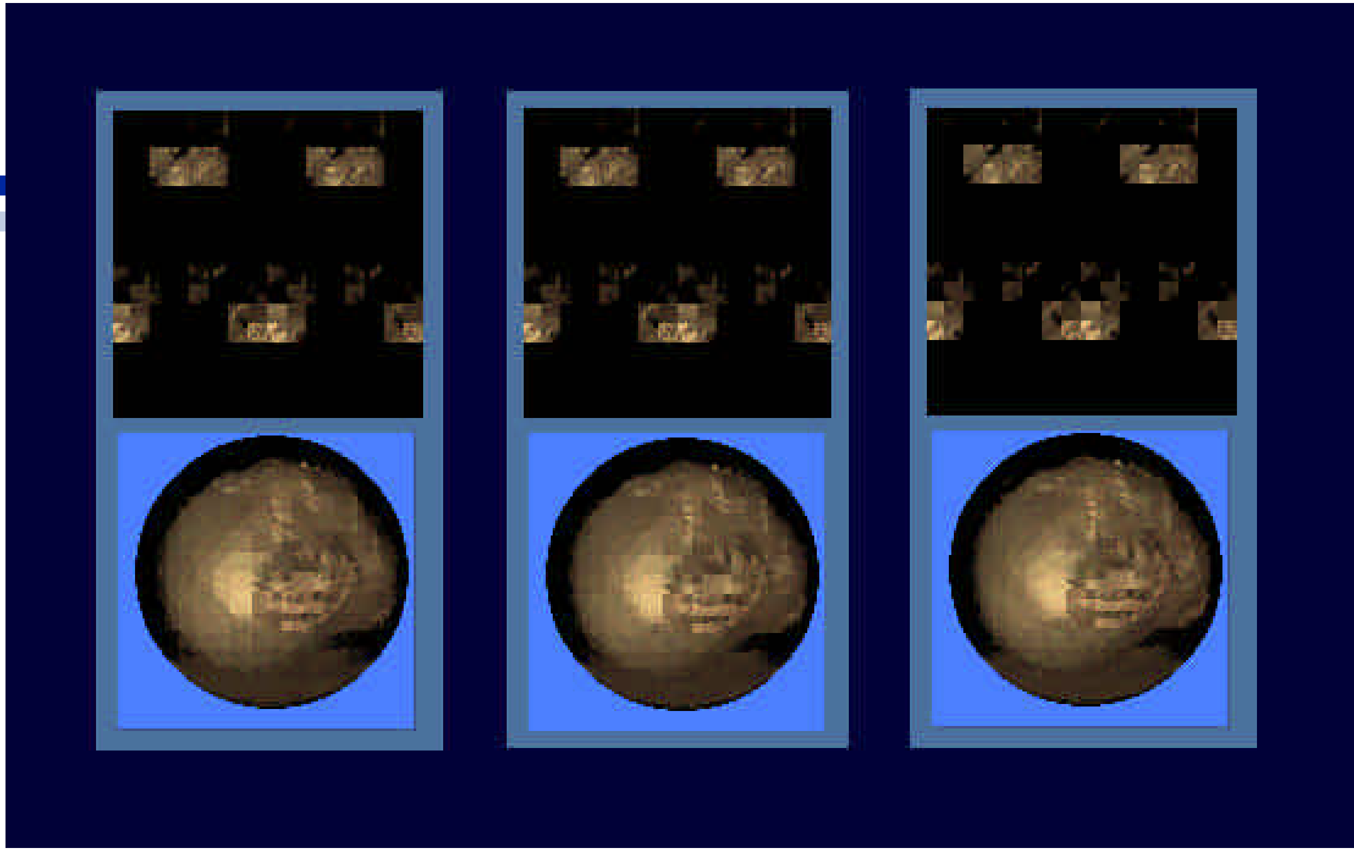
Performances

- Time
  - Most of accesses fall in the case A or B!
  - Counting bit 1's can be implemented in a few table accesses, 2.5 on the average.
- Space

$$r = \frac{a}{3} + \frac{5}{96}b + \frac{1}{192}$$

$$a: \frac{\text{\# of coef's in the three-byte stream}}{\text{\# of the whole coef's}}$$

$$b: \frac{\text{\# of non-null subblocks}}{\text{\# of the whole subblocks}}$$



Experimental Results

- Hardware
  - SGI MIPS R4400 CPU (200MHz), HighImpact graphics
  - Intel Pentium Pro CPU (200MHz), Intergraph Intense 3D graphics
- Data
  - generated three spherical light fields from the UNC head (CT) data (256X256X225).
  - classification: an opaque skull with semitransparent skin.

- Rendering time (on SGI)
  - par. proj.: 0.44, 0.29, 0.26 sec./image on the average
  - pers. proj.: took roughly twice as long
  - proportional to the number of vertices of the positional sphere.
  - almost independent of image sizes.

Conclusion and Future Work

- Presented a new parameterization of light flow.
- Presented a new compression scheme based on wavelets.

- Investigate wavelet bases other than Haar.
- Devise an adaptive rendering technique for speed enhancement.
- Extend our 2D wavelet-based encoding scheme into a three- or four-dimension.

서강대학교
Sogang University