Visualization-Specific Compression of Large Volume Data

C. Bajaj, I. Ihm** and S. Park**

**The Univ. of Texas at Austin, U.S.A.*
***Sogang University, Korea*

Motivations

- Observations
  - Volume data are often very large.
  - Lossy compression is often inevitable.
  - Sometimes, we wish to visualize some fixed set of coherent features that are concentrated in a few regions.
  - For most lossy compression schemes, information on those features is *uniformly* lost regardless of the visualization task to be performed.
  - *Can we interactively visualize a 2GB volume data set on my computer with 128MB of main memory?*

- Question
  - Is it possible to design a compression scheme that enables to focus more on important voxel, used more frequently during visualization?

Research Goals

- Develop a technique
  ① that classifies voxels according to their importance in the visualization task,
  ② that assigns appropriate weights to voxels, and
  ③ that uses those weights during lossy compression so that *important* features remain as correct as possible after reconstruction.

- Propose a new framework for lossy compression!

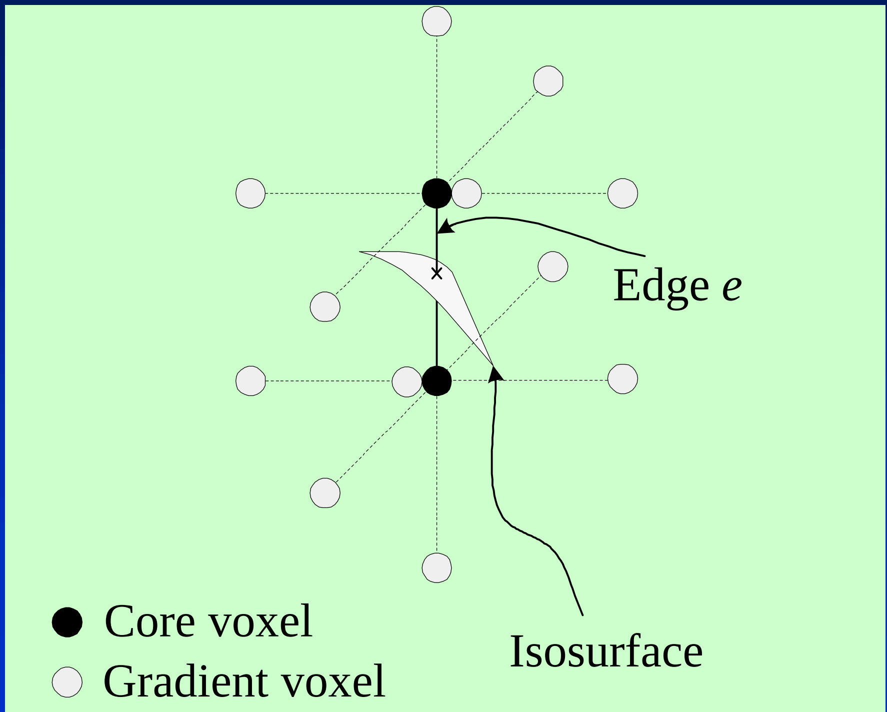
Related Work

- There are a lot of results on volume compression.
  - Compression is dependent on the spatial properties of images.
  - We want to compress volume data in accordance with the purpose of a visualization task.
- It was difficult to find a feature-based compression method in volume visualization.

Visualization Model

- Two popular visualization techniques
  - Isosurface extraction: marching cube
  - Direct volume rendering: ray casting
- Features to be visualized are known beforehand.
- A 3D volume data is viewed as a 3D graph (V, E)
  - V: a set of all voxels v
  - E: a set of all unordered pairs (u, v), where two voxels u and v are 6-neighbors to each other
  - $d(v)$: a density function from V to R

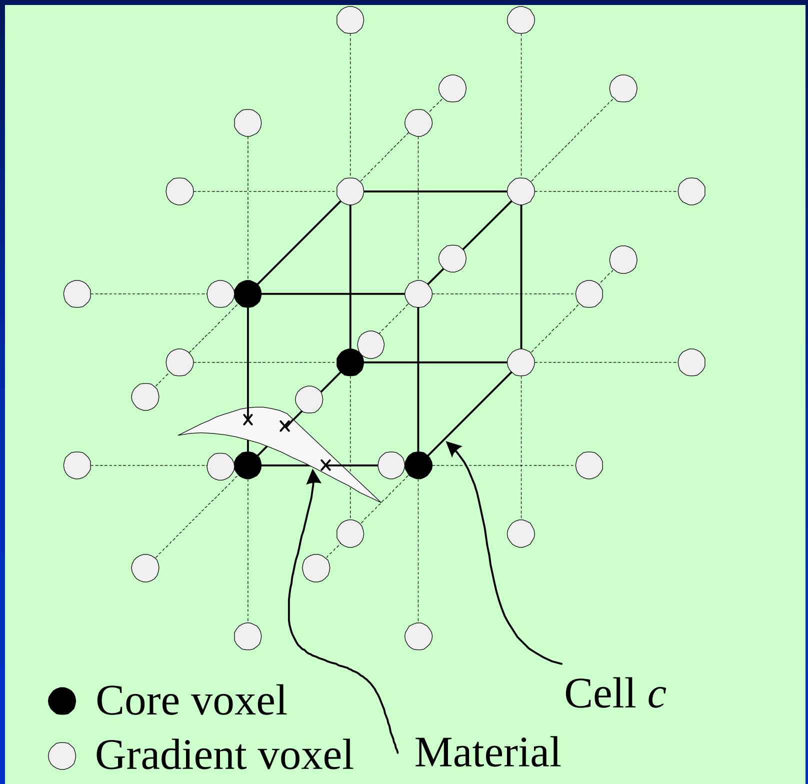
Marching Cube [9]



- The edge intersection and normal vectors are linearly interpolated along the edge.

- The correctness of isosurfaces with normals are dependent on the densities of incident voxels of all intersecting edges, and their 6-neighbors.

Ray Casting [7]



❖When a point inside a cell is resampled, the shaded colors and normal vectors at the eight voxels are tri-linearly interpolated.

❖The correctness of the computed (C_i, a_i) are dependent on the densities of the eight voxels of all resampled cells, and their 6-neighbors.

Voxel Classification: Core Voxels

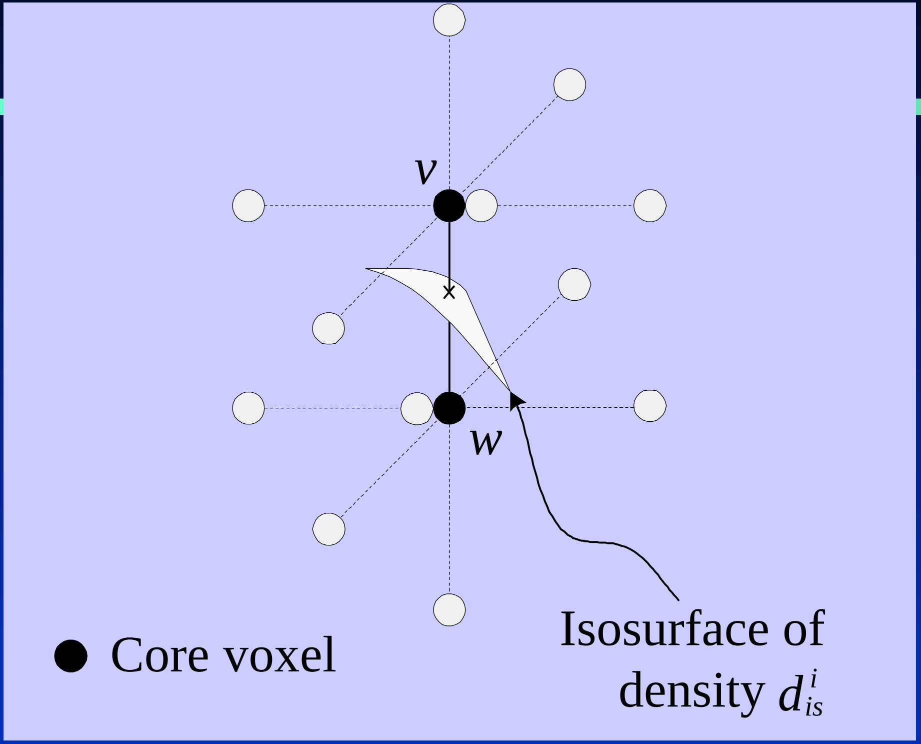
- Density values and density interval of interest

$$D_{is} = \left\{ d_{is}^{i} \mid i = 0,1,\cdots, p-1 \right\}: p \text{ density values}$$

$$D_{rc} = \left\{ [d_{rc}^{i0}, d_{rc}^{i3}] \mid i = 0,1,\cdots, q-1 \right\}: q \text{ density intervals}$$

- *Core voxels*
  - Voxels that directly affect isosurface extraction and/or ray-casting.

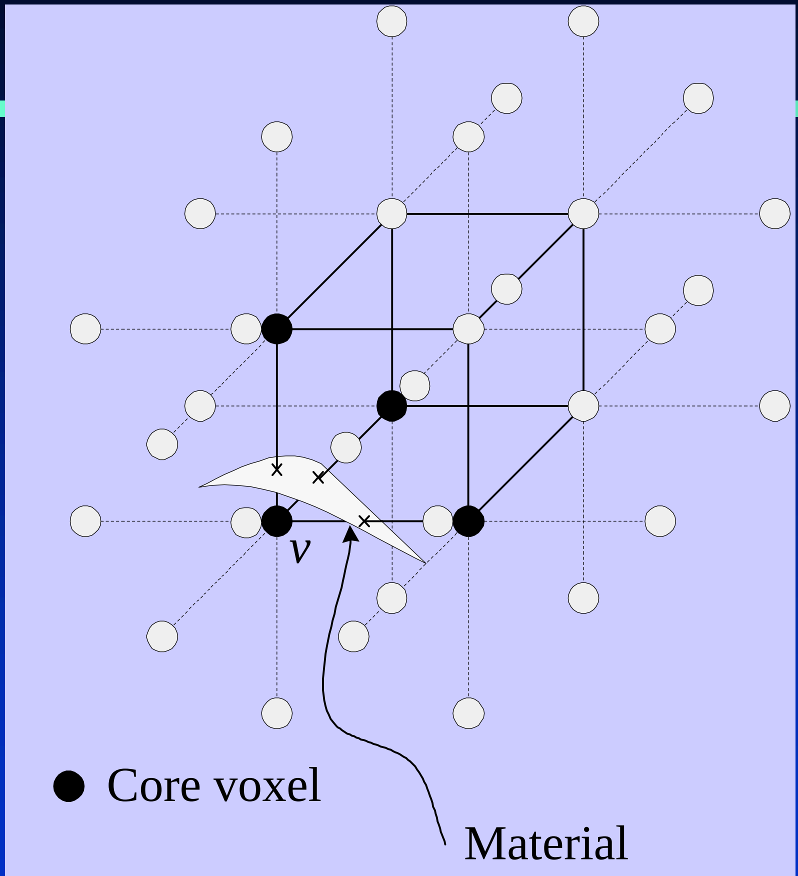
$$V^{c} \equiv V^{c}(D_{is}, D_{rc}) = V_{is}^{c}(D_{is}) \cup V_{rc}^{c}(D_{rc})$$

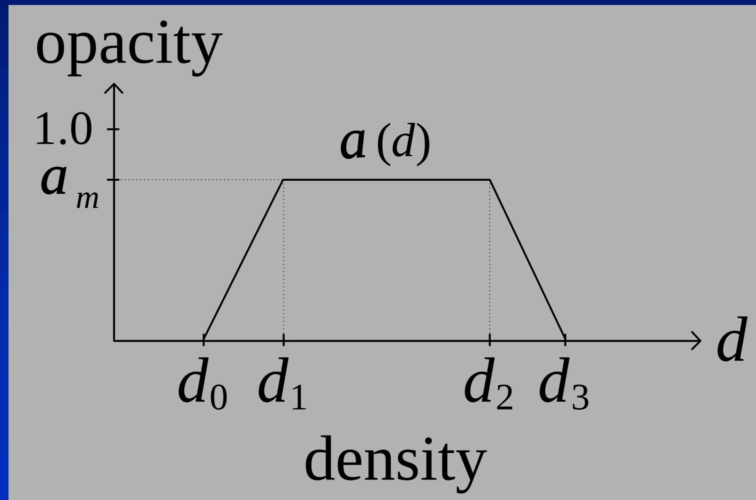


$$V_{is}^{c}(D_{is}) = \left\{ v \in V \mid d_{is}^{i} \in I_d(v,w) \text{ for some } d_{is}^{i} \in D_{is} \text{ and } w \in N_6(v) \right\},$$

$$\text{where } I_d(v,w) = \left\{ [\min(d(v), d(w)), \max(d(v), d(w))] \right\}, \text{ and}$$

$$N_6(v) \text{ consists of all 6-neighbors of } v.$$



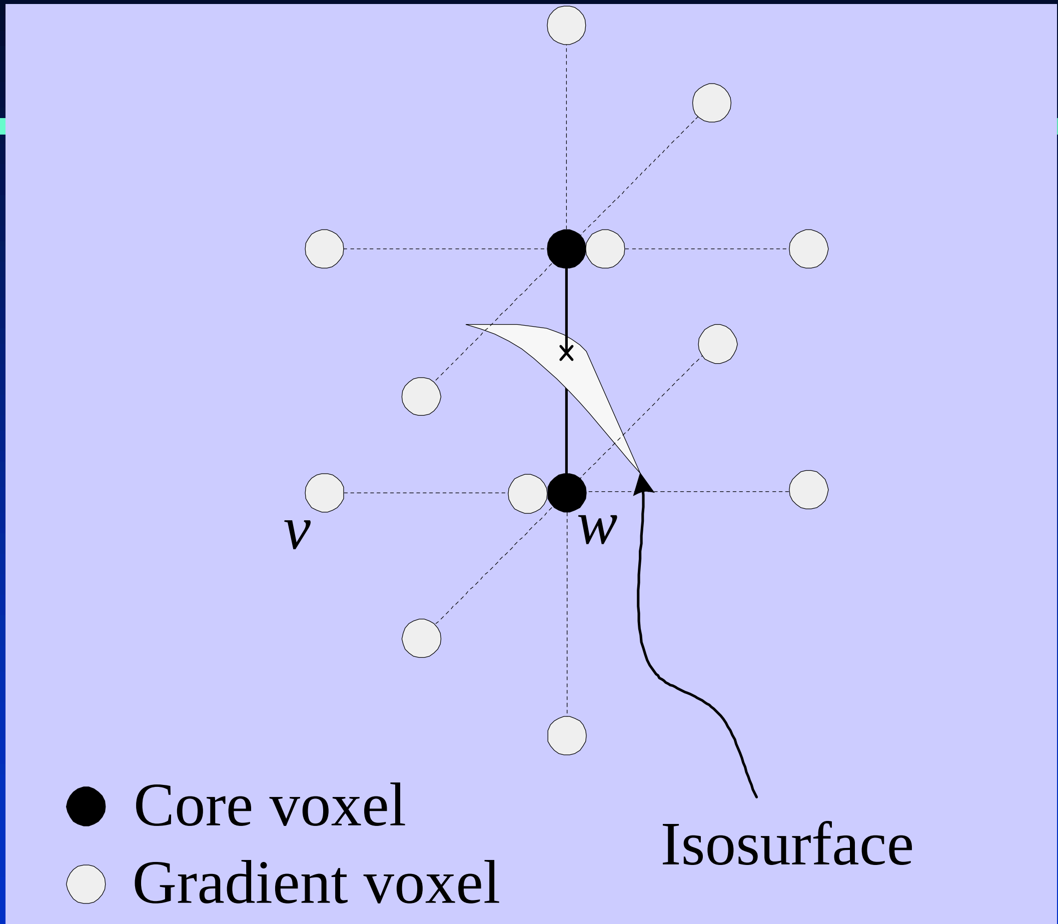


$$V_{rc}^{c}(D_{rc}) = \left\{ v \in V \mid \sum_{i=0}^{q-1} a_i(v) \geq a_e \text{ for some } a_e > 0 \right\}$$

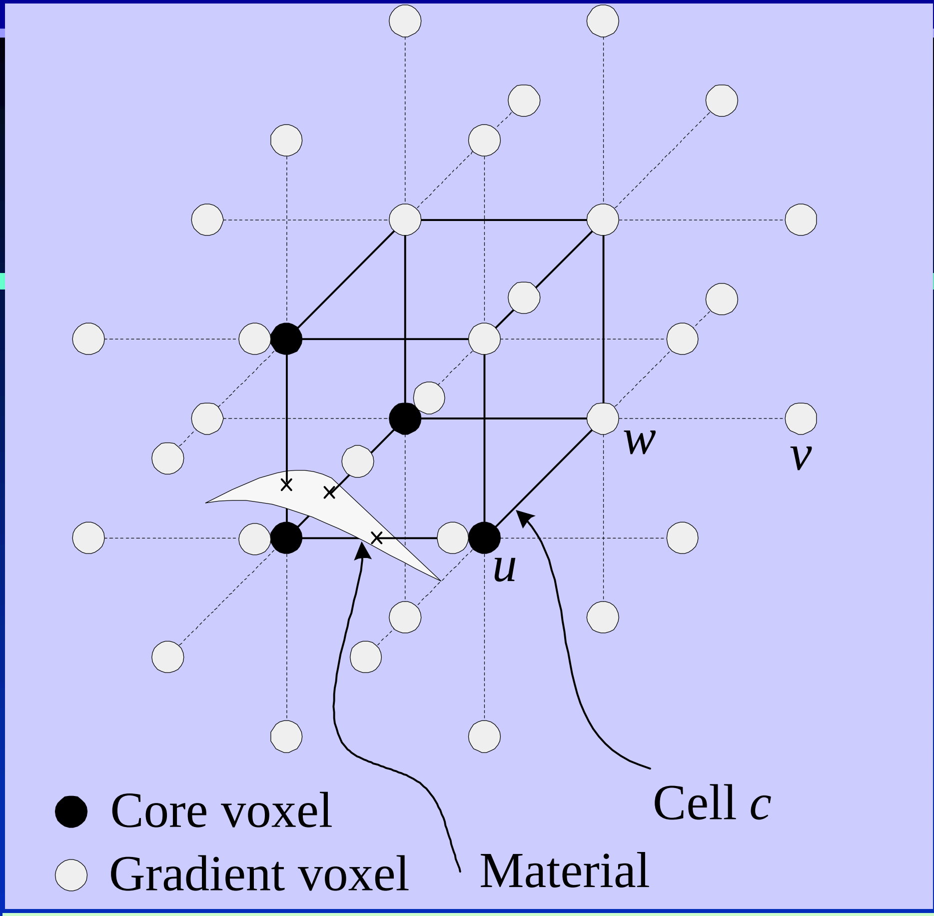
Voxel Classification: Gradient Voxels

- *Gradient voxels*
  - Voxels that affect the gradient computations in isosurface extraction and/or ray-casting.

$$V^g \equiv V^g(D_{is}, D_{rc}) = V_{is}^g(D_{is}) \cup V_{rc}^g(D_{rc})$$



$$V_{is}^{g}(D_{is}) = \left\{ v \in V \mid w \in V_{is}^{c}(D_{is}) \text{ for some } (v, w) \in E \right\}$$



$$V_{rc}^{g} = \{ v \in V \mid v \in N_6(w) \text{ for some } w \in V(c) \text{ such that } u \in V_{rc}^{c} \text{ for some } u \in V(c) \},$$

where $V(c)$ is a set of eight voxels of cell c.

Voxel Classification: Unimportant Voxels

- Unimportant voxels
  - Voxels that are neither core nor gradient

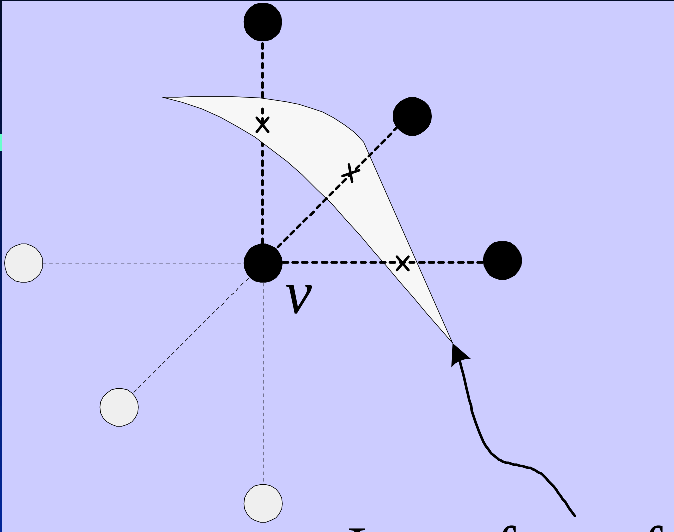
$$V^u \equiv V^u(D_{is}, D_{rc}) = V - (V^c \cup V^g)$$

Definition of weight function

- Goal
  - Weight voxels according to their possible usage, or importance in the visualization task.

- For v in V_{is}^{c}
  - Give larger weights to voxels whose incident edges are cut through by more isosurfaces.

$(V_{is}^{c})^{i}, i = 0,1,\cdots, p-1$: the subset of V_{is}^{c}, made of core voxels relative to the ith material only



$$d_{is}^{i}(v)=\begin{cases} |\{w \mid w\in N_6(v)\cap (V_{is}^{c})^{i}\}|, & \text{if } v\in (V_{is}^{c})^{i}, \\ 0, & \text{otherwise} \end{cases}$$

$$f_{is}^{c}(v)=\max_{0\le i\le p-1} d_{is}^{i}(v)$$

Maximum possible value: 6

- For v in V_{rc}^{c}
  - Preserve well the voxels' densities that actually contribute to color-opacity accumulation during rendering.

$$f_{rc}^{c}(v) = \max_{0 \le i \le q-1} a_i(v)$$

Maximum possible value: $\max_{0 \le i \le q-1} a_{mi}$

- For v in V_{is}^{g}
  - V's density becomes more important as more isosurfaces pass through the edges incident to the neighbors.

$$f_{is}^{g}(v) = \max_{0 \le i \le p-1} \sum_{w \in N_6(v)} d_{is}^{i}(w)$$

Maximum possible value: 36

- For v in V_{rc}^{g}
  - Count the number of incident cells that have at least one voxel with nontrivial compute opacity.

$$d_{rc}^{i}(v) = |\{ c \in C \mid c \in C(w) \text{ for some } w \in N_6(v) \text{ such that } u \in V_{rc}^{c} \text{ for some } u \in V(c) \}|, \text{ where } C(w) \text{ is all eight cells incident to } v$$

$$f_{rc}^{g}(v) = \max_{0 \le i \le q-1} d_{rc}^{i}(v)$$

Maximum possible value: 32

- Finally, the weight function is

$$f_v(v) = f\left(f_{is}^{c}(v), f_{rc}^{c}(v), f_{is}^{g}(v), f_{rc}^{g}(v) \right)$$

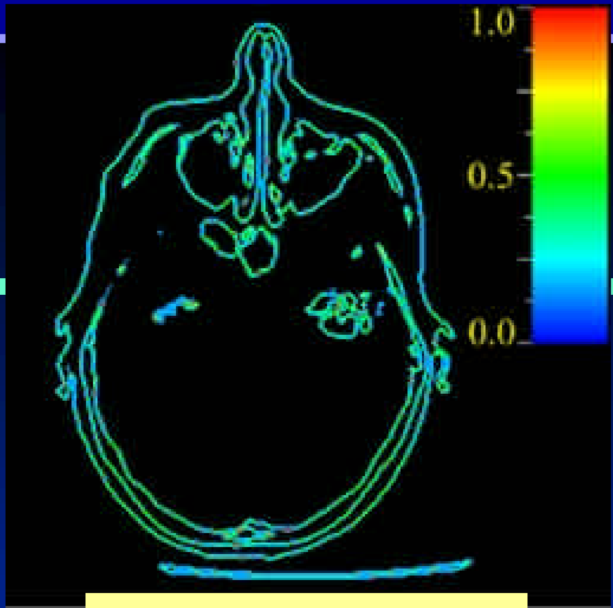
  - The function values in the argument list are defined in the different domains.
  - They must be normalized according to volume data.

Example: CT Data

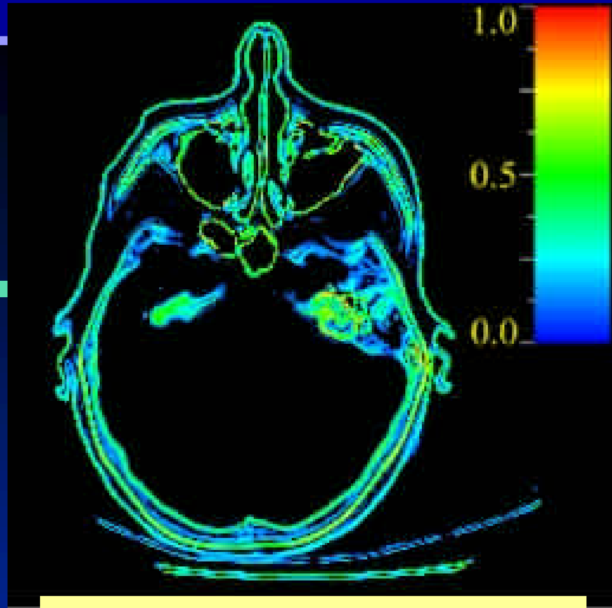
- The UNC Bighead data (256 X 256 X 225)

$$D_{is} = \{ 66, 160 \}$$
$$D_{rc} = \{ [40, 90], [95, 255] \}$$

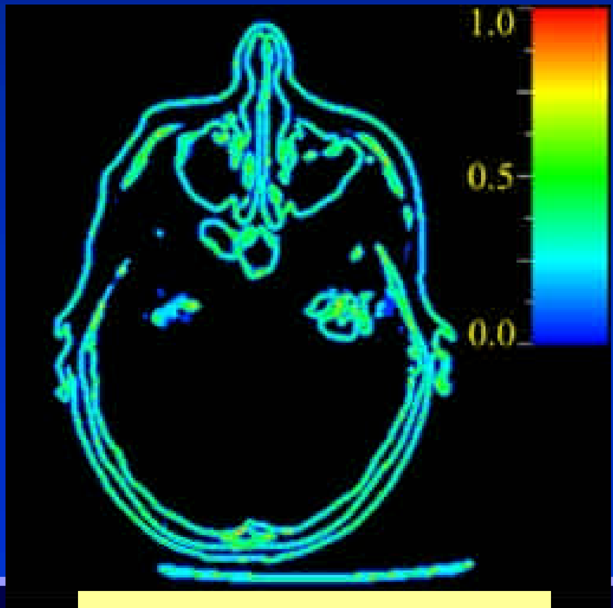
```
Opacity tr. ftn. : [40, 45, 88, 90],
                   [95, 135, 253, 255]
Gradient tr. Ftn. : [0, 95]
```



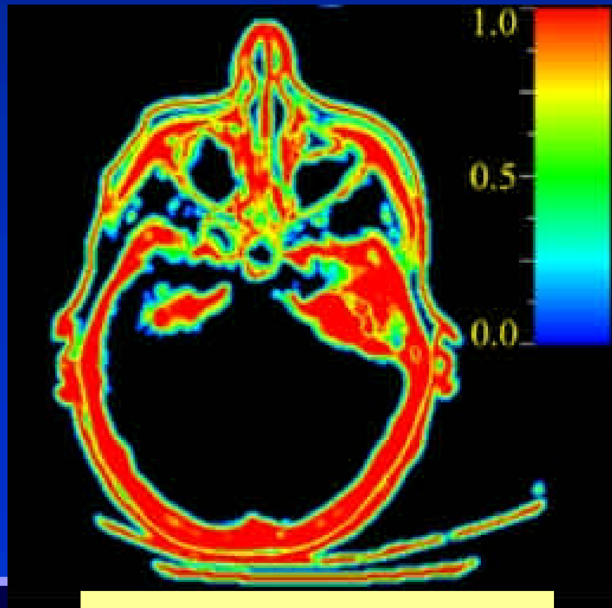
$f_{is}^{c}(v)$: slice max = 6



$f_{rc}^{c}(v)$: slice max = 0.642

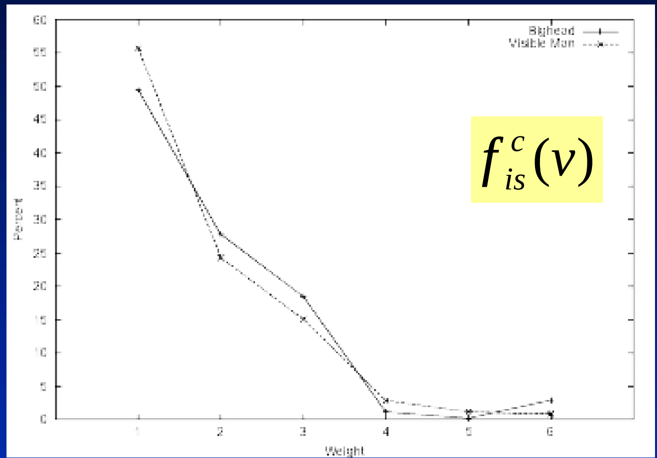


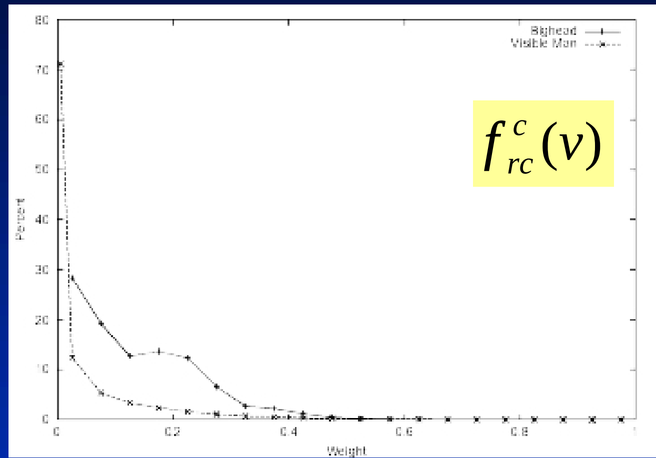
$f_{is}^{g}(v)$: slice max = 23

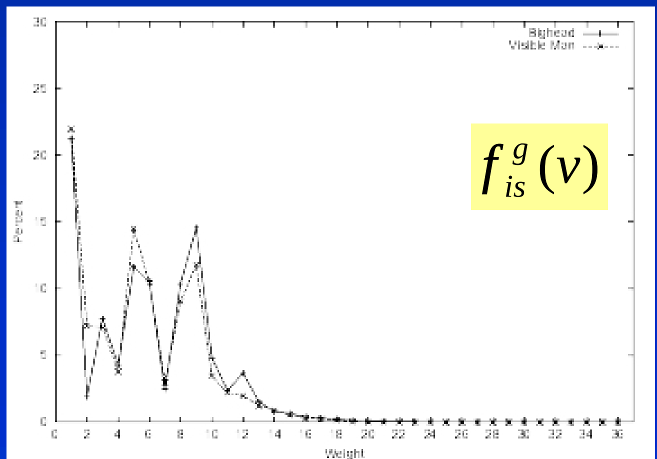


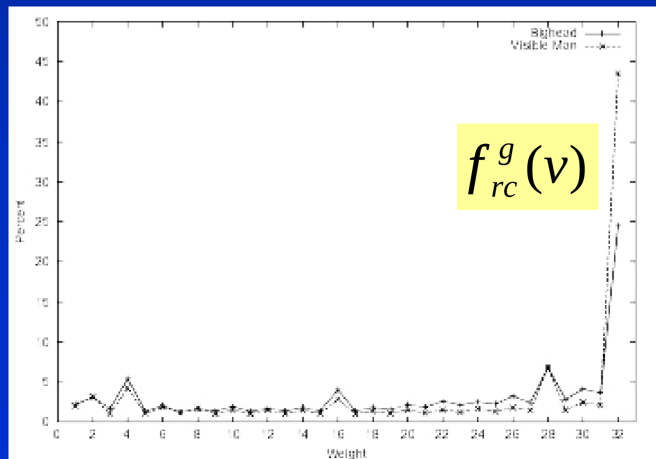
$f_{rc}^{g}(v)$: slice max = 32

Distribution of Weights



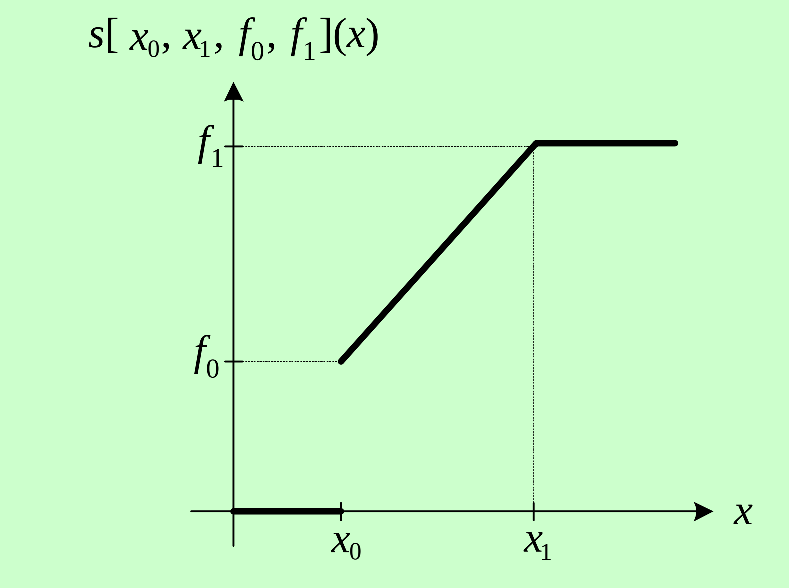






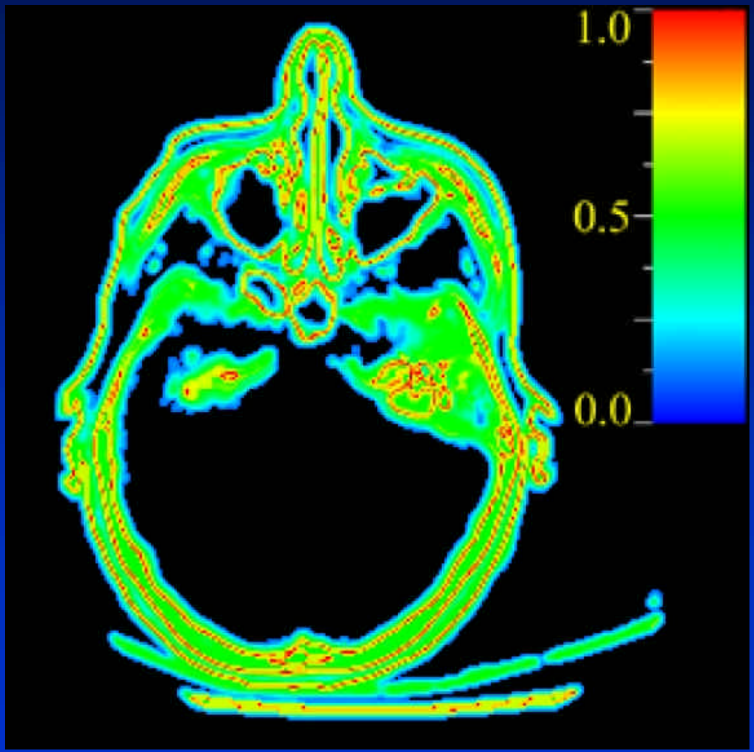
Combining the Weight Functions

- Use a heuristic method.
  - ① Rescale the four weights.
  - ② Control the relative strength of two visualization methods.

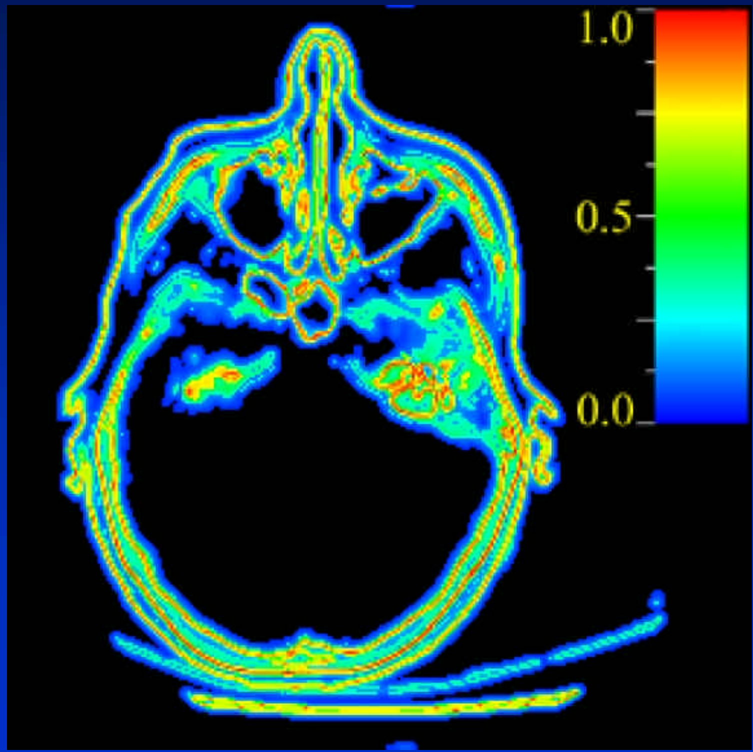


$$f_v(v) = \max\{ k_{is} \cdot (s[1,4,0.7,1.0](f_{is}^{c}(v)) + s[1,20,0.2,0.5](f_{is}^{g}(v))),$$
$$k_{rc} \cdot (s[0.05,0.6,0.4,1.0](f_{rc}^{c}(v)) + s[1,30,0.1,0.5](f_{rc}^{g}(v))) \}$$

Examples: Combined Weights



Maximum weight



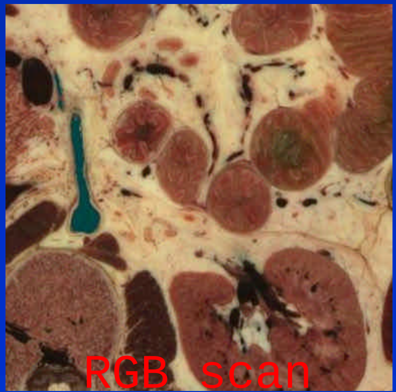
Summed weight

Application to Compression Scheme

- So far, we’ve proposed how to weigh voxels with respect to a visualization task.
- How do we use those weights in compressing volume data?
- Consider lossy compression schemes such as
  - Zerobit encoding
  - Vector quantization
  - Etc.

Zerobit encoding [Bajaj, Ihm, Park '01]

- A new compression scheme that supports the fast random decoding and multi-resolution representation.
- Appropriate for developing real-time/interactive applications that must handle 3D images.
- Based on the simple 3D Haar filter.



2D Haar Wavelet Transforms

Decomposition

$$w_{00} = (d_{00} + d_{01} + d_{10} + d_{11}) / 4$$
$$w_{01} = (d_{00} + d_{01} - d_{10} - d_{11}) / 4$$
$$w_{10} = (d_{00} - d_{01} + d_{10} - d_{11}) / 4$$
$$w_{11} = (d_{00} - d_{01} - d_{10} + d_{11}) / 4$$

d_{00}	d_{01}
d_{10}	d_{11}

$\longrightarrow$

w_{00}	w_{01}
w_{10}	w_{11}

Two level applications

A		B	
d_{00}	d_{01}	d_{02}	d_{03}
d_{10}	d_{11}	d_{12}	d_{13}
d_{20}	d_{21}	d_{22}	d_{23}
d_{30}	d_{31}	d_{32}	d_{33}
C		D	

$\longrightarrow$

w_{00}	w_{11}	w_{01}	w_{21}
w_{12}	w_{13}	w_{22}	w_{23}
w_{02}	w_{31}	w_{03}	w_{41}
w_{32}	w_{33}	w_{42}	w_{43}

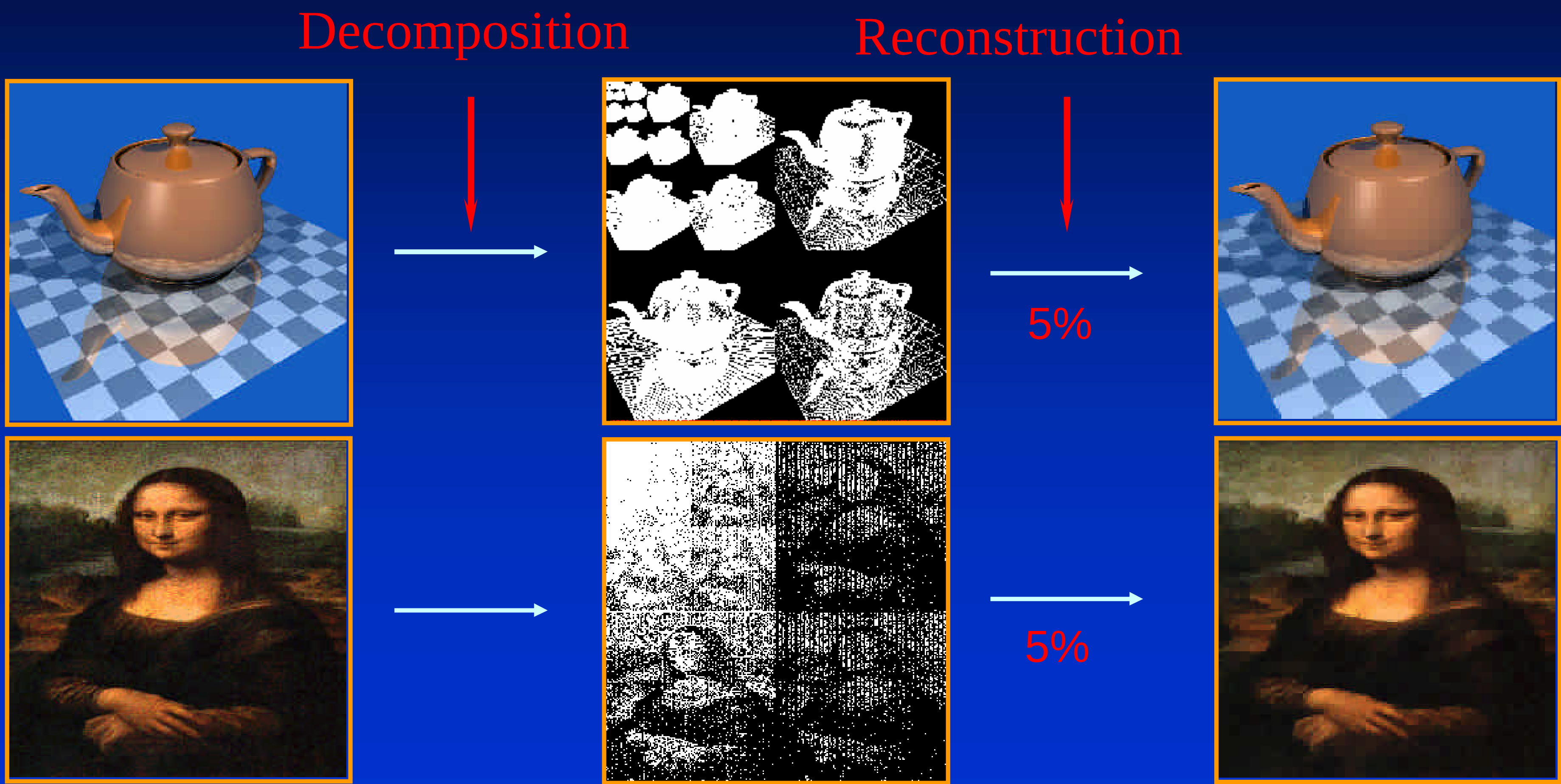
Wavelet-Based Compression

- Idea
  - Decompose an input data set through wavelet transforms.
  - Sort the wavelet coefficients in order of decreasing magnitude.
  - Given an error measure, delete as many coefficients with smaller magnitude as possible.

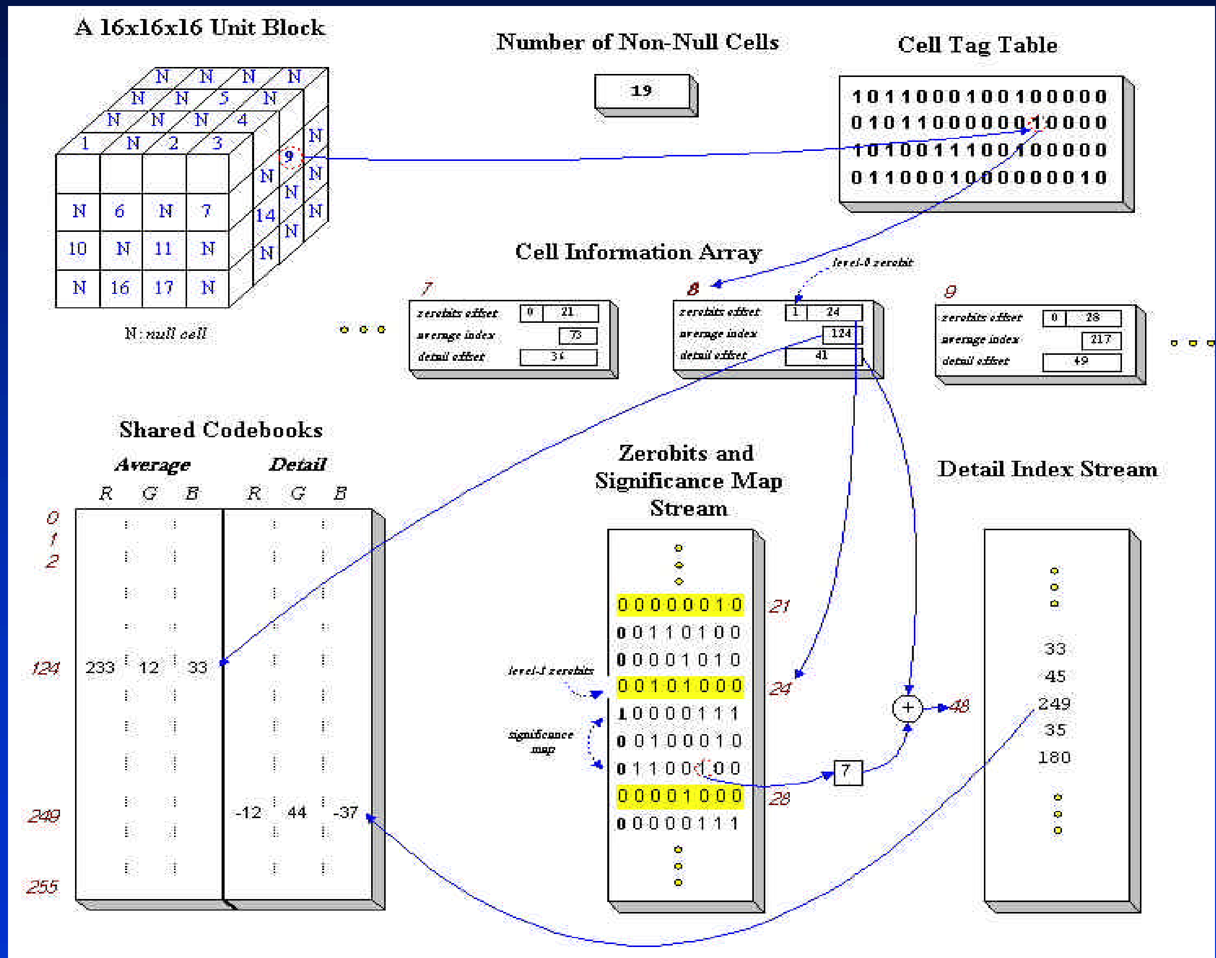
$$f(x) = \sum_{i=1}^{m} c_i \cdot u_i(x) \quad \Rightarrow \quad \overline{f}(x) = \sum_{i=1}^{\overline{m}} \overline{c}_i \cdot \overline{u}_i(x) \quad (\overline{m} \le m)$$

☞ This is the best choice for orthonormal bases under the L2 norm.

2D Image Compression Using Haar Wavelets

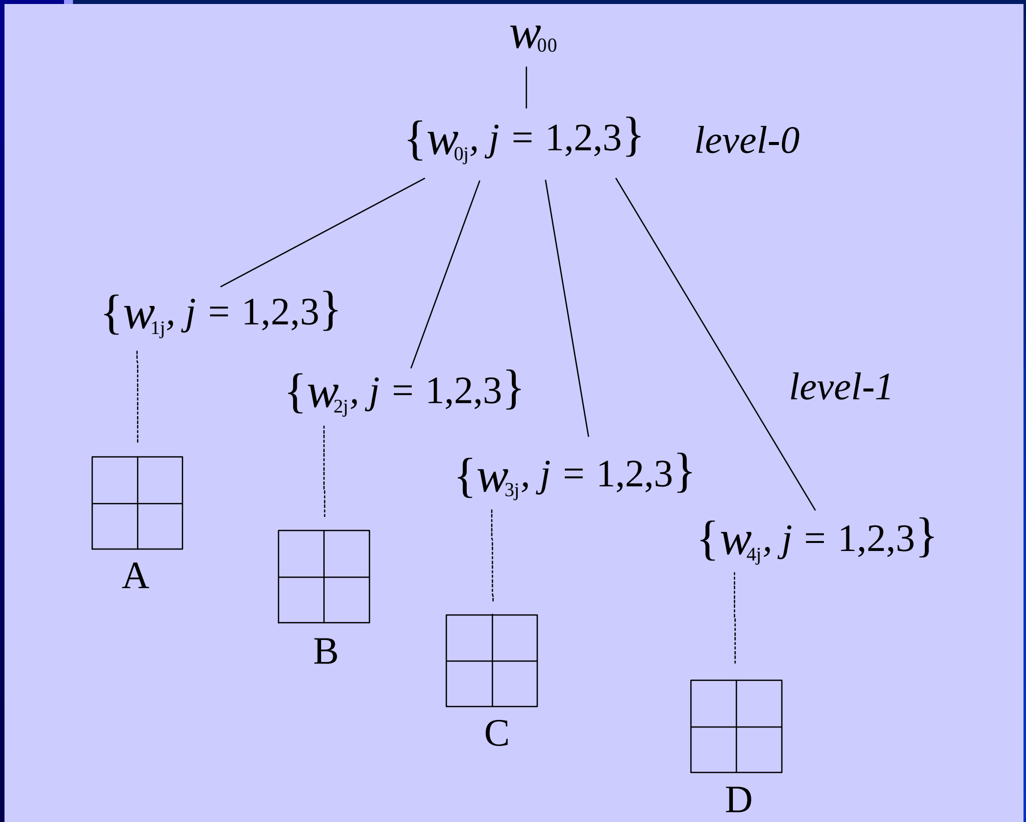


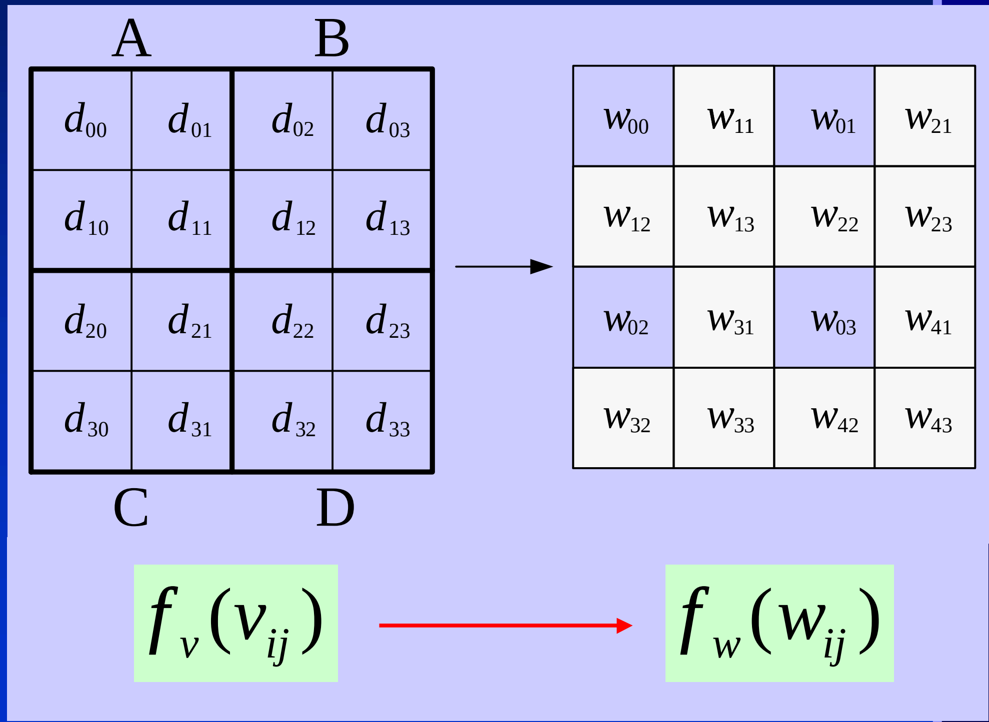
Zerobit Encoding (3D RGB)



Weights in the Wavelet Space

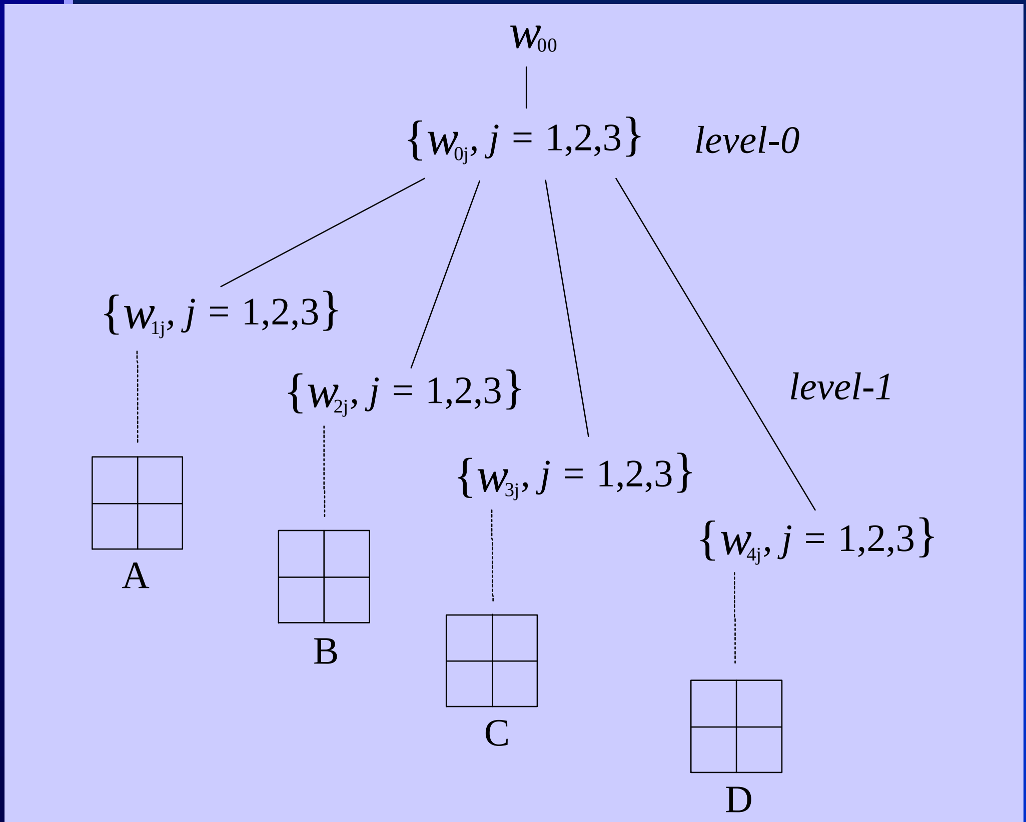
Decomposition tree





Computation of Weight

- Idea
  - The wavelet coefficients are important if the corresponding voxels are important.
  - All the 4 wavelet coefficients are used simultaneously when any voxel in a 2x2 area.
  - ⇨ Assign the largest of the 4 weights to the 4 wavelets.



- Traverse the decomposition tree in bottom-up fashion.
  - Level 1 detail node

$$f_w(w_{ij}) = \max_{k=1,2,\cdots,8} f_v(v_{ik}) \text{ for } j = 1,2,\cdots,7$$

  - Level 0 detail node

$$f_w(w_{0j}) = \max_{k=1,2,\cdots,8} f_w(w_{k1}) \text{ for } j = 1,2,\cdots,7$$

  - Level 0 average node

$$f_w(w_{00}) = f_w(w_{01})$$

Truncation of Wavelet Coefficients

- In truncating wavelets,
  - use the measure $|n(f_w(w_{ijk})) \cdot w_{ijk}|$ instead of $|w_{ijk}|$.

- Need to rescale the wavelets' weights properly.
  - for instance, $n(f_w(w)) = g_0 + g_1 \cdot f_w(w)$

Experimental Results - Bighead

- The UNC Bighead data
- Resolution: 256X256X225
- Specification
  - $D_{is} = \{66, 160\}$
    $D_{rc} = \{[40, 90], [95, 255]\}$
  - Important voxels: 19.89%

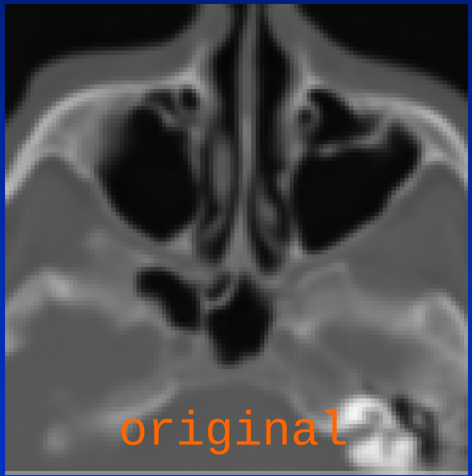
```
Opacity tr. ftn. : [40, 45, 88, 90],
                   [95, 135, 253, 255]
Gradient tr. ftn.: [0, 95]
```

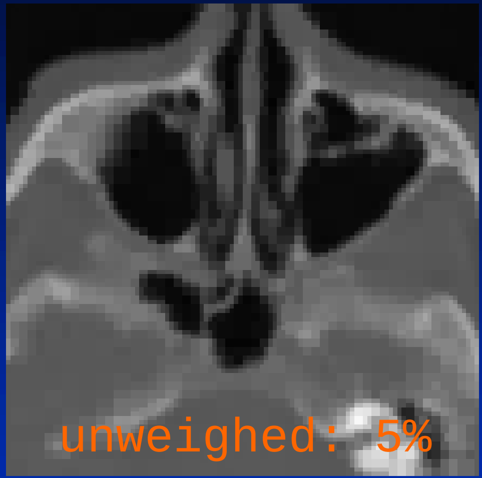
Quantitative Analysis of Reconstruction Errors

		Ratio of the used wavelet coefficients			
		2%	3%	5%	7%
Unweighed Haar	RMSE*	10.78	8.76	6.38	4.91
	PSNR	27.48	29.28	32.04	34.31
Weighted Haar	RMSE*	6.78	4.86	2.82	1.79
	PSNR	31.50	34.40	39.12	43.06

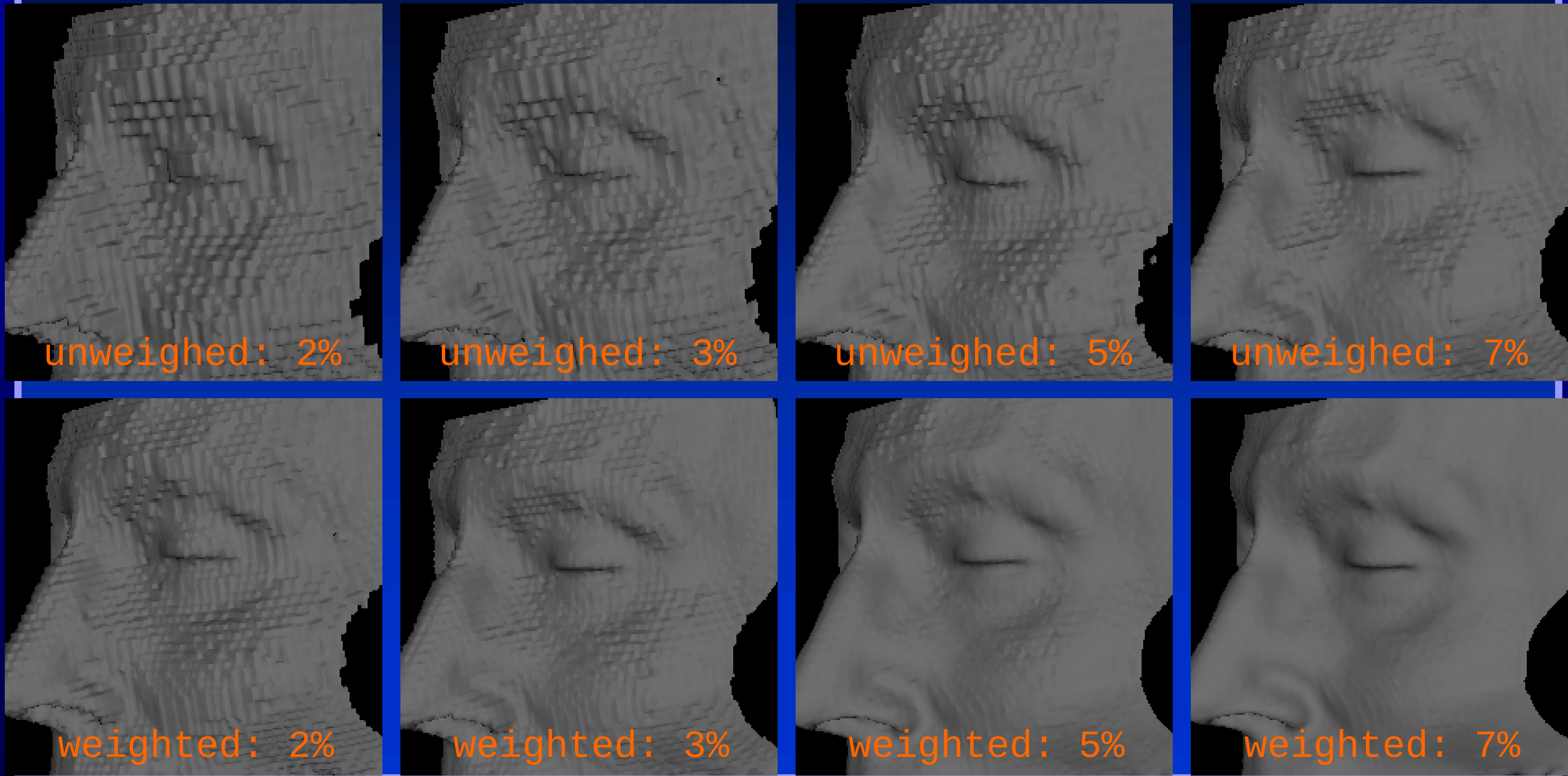
(*: density range 0 ~ 255)

Reconstructed Images (114th slice)

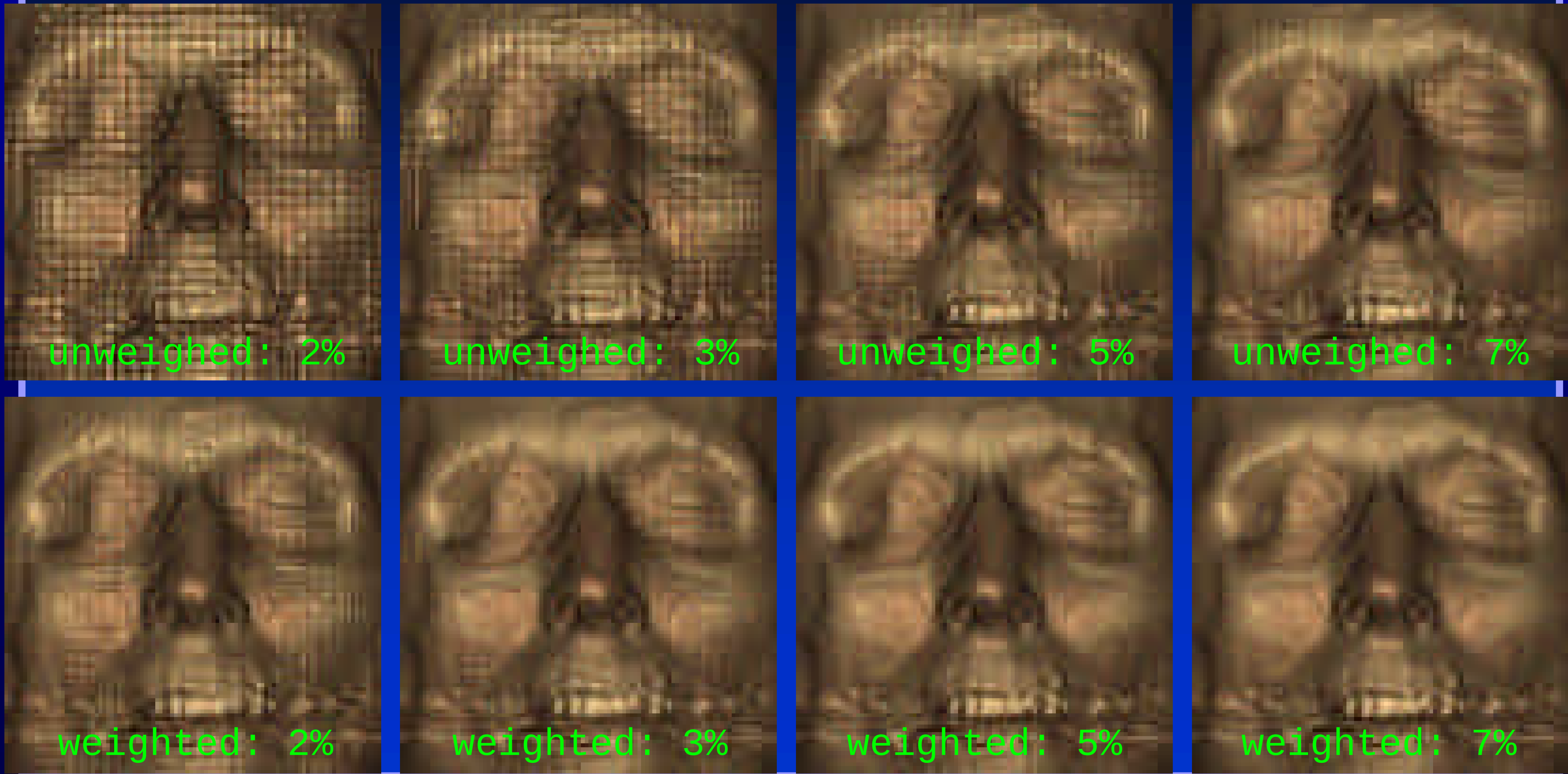




Isosurface Rendering (d = 66)



Ray Casting ([40,45,88,90], [95,135,253,255])

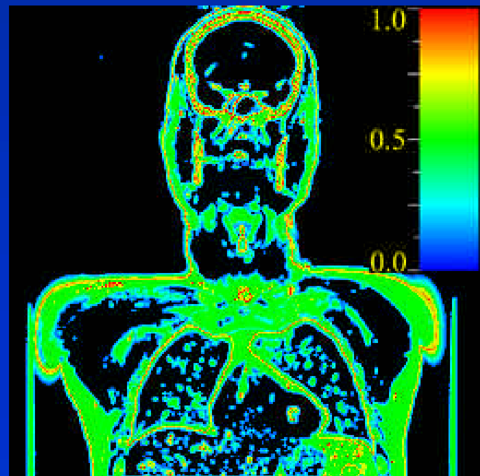


Experimental Results – Visible Man (NLM)

- A preprocessed FRESH CT data
- Resolution: 512X512X512 (512MB)
- Specification
  - $D_{is} = \{880, 1800\}$

    $D_{rc} = \{[320, 992], [1120, 2400]\}$
  - Important voxels: 23%

```
Opacity tr. ftn. : [320, 800, 960, 992],
                   [1120, 1300, 2300, 2400]
Gradient tr. ftn.: [0, 1120]
```



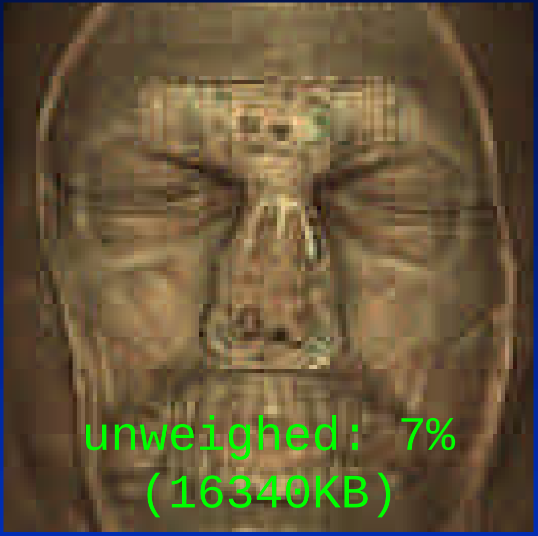
Quantitative Analysis of Reconstruction Errors

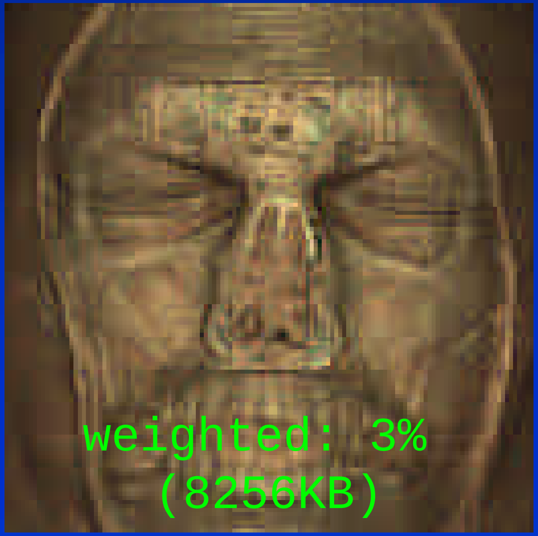
		Ratio of the used wavelet coefficients			
		2%	3%	5%	7%
Unweighed Haar	RMSE*	66.61	51.55	36.02	27.10
	PSNR	35.77	38.00	41.11	43.58
Weighted Haar	RMSE*	46.92	34.21	20.88	13.92
	PSNR	38.82	41.56	45.85	49.37

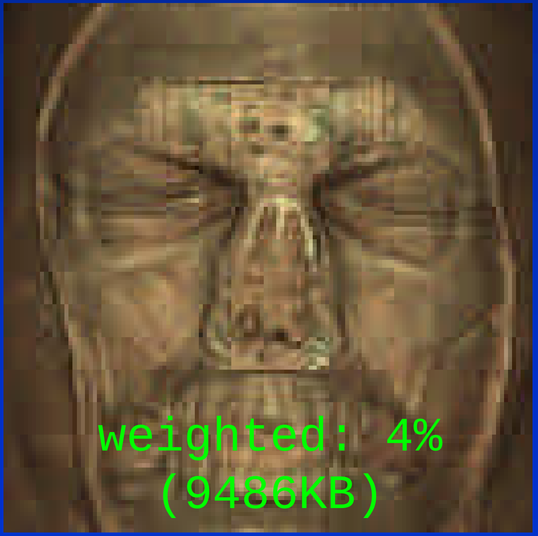
(*: density range 0 ~ 4095)

Ray Casting

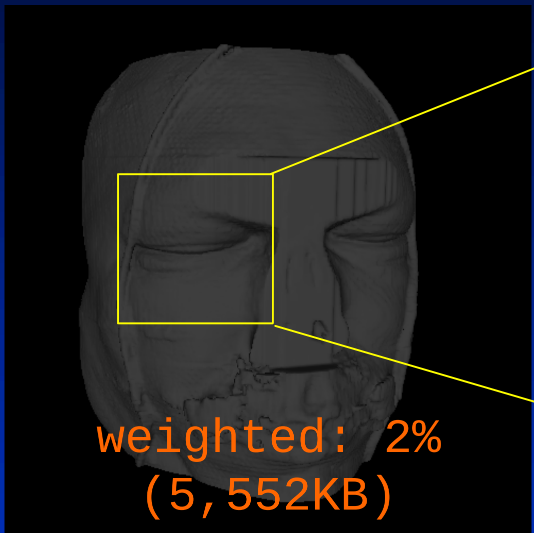
Visual quality:
Unweighed 7% ~ weighted 3-4%

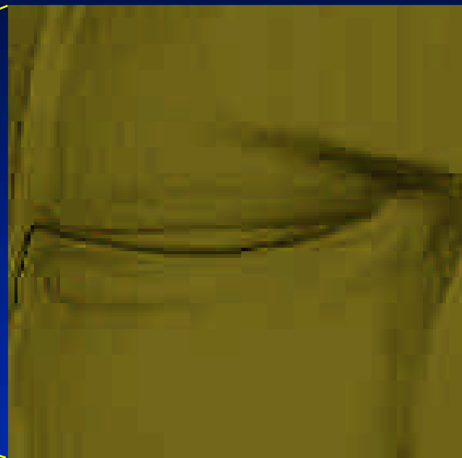


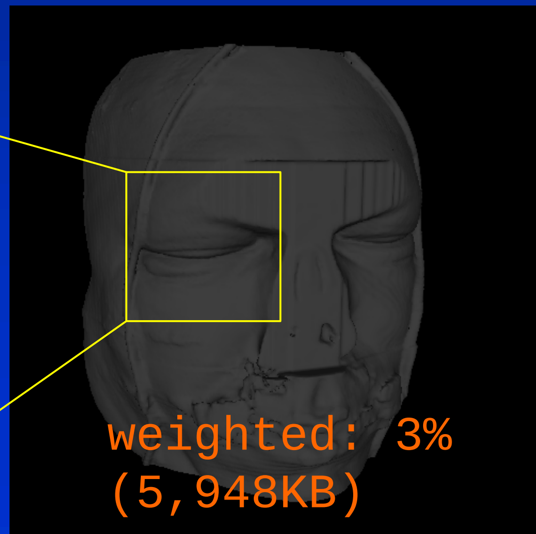




Isosurface Rendering (d = 600)







$$D_{is} = \{\, 600 \,\}$$

$$D_{rc} = \varnothing$$

Important voxels:5.27%

Discussion

- Presented a new volume compression method.
  - Proposed a method for classifying voxels according to their importance in visualization, not their spatial properties.
  - Applied the weight information to a compression scheme, called Zerobit encoding.

⇨ When the possible visualization features can be pre-determined, our new scheme will be used effectively.

Future Work

- Need a *more scientific* method to determine the coefficients in the various function definitions.
  - Currently, they are determined experimentally.
- Apply our scheme to other compression schemes.
  - For instance, an enhanced codebook could be built.

Related Work

- Ning et al. [Vis. 93]
  - Vector quantization
- Muraki [Vis. 92, CG&A 93]
  - 3D Wavelet compression
- Ghavamnia et al. [Vis. 95]
  - Laplacian pyramid
- Yeo et al. [IEEE TVCG 95]
  - DCT-based 3D compression
- Thoma et al. [IEEE Mult. 97]
  - Experimental results on 2D compression of VH