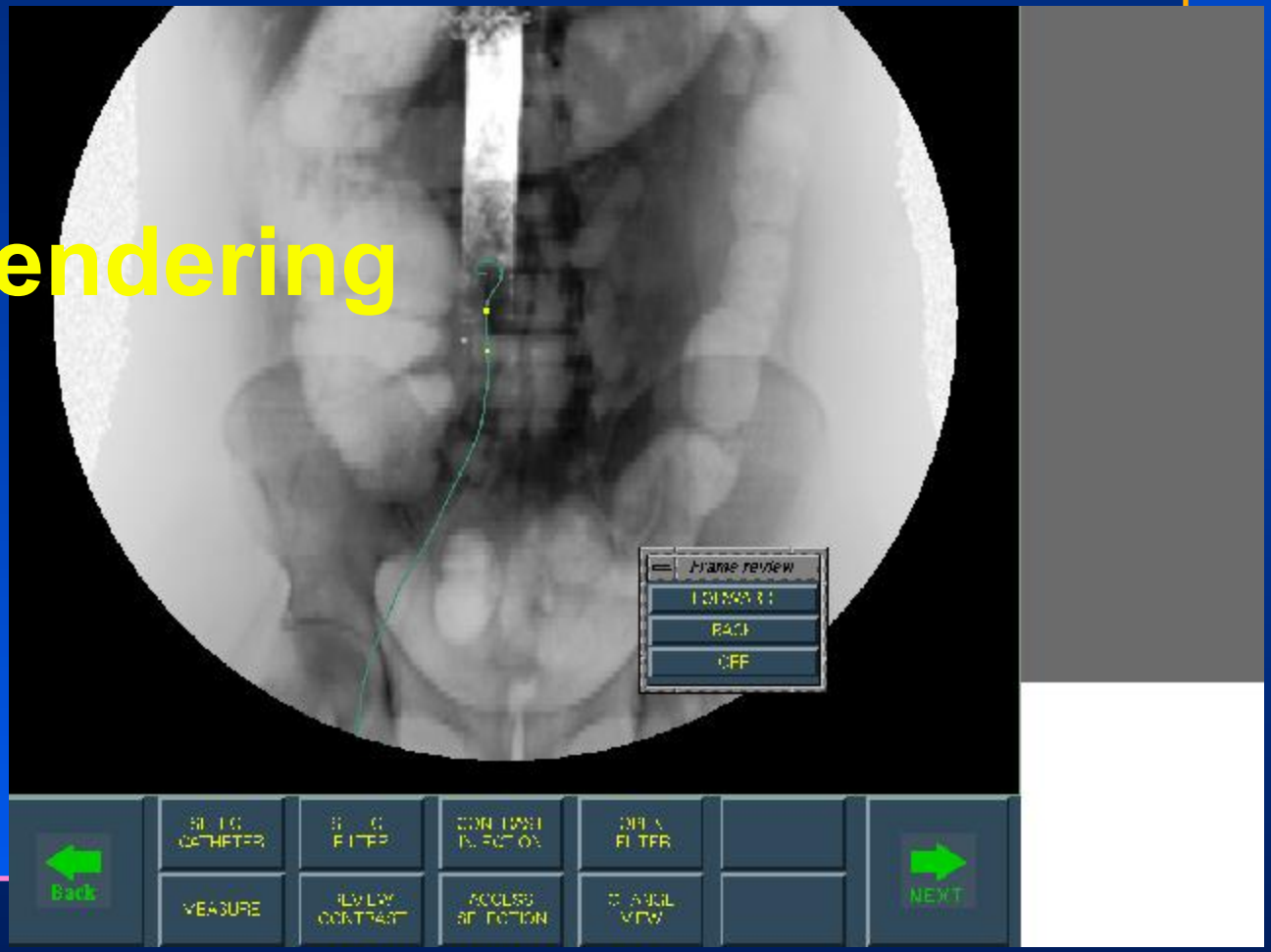
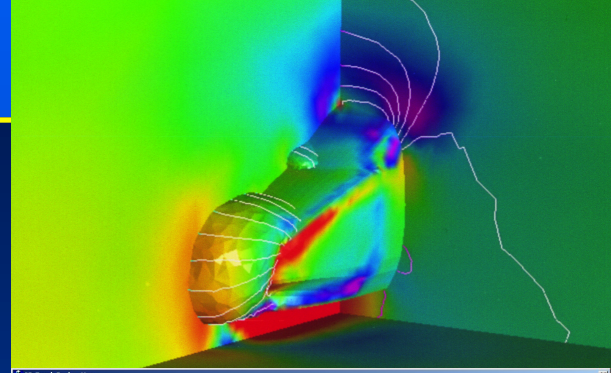
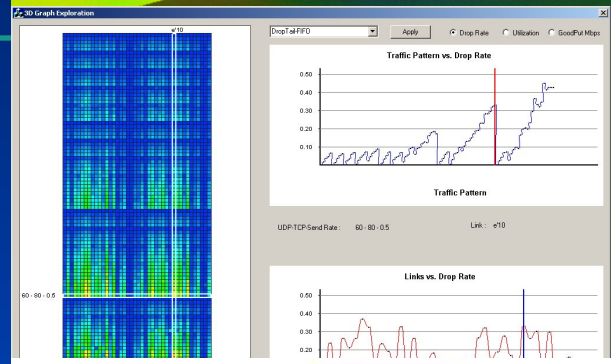
Volumetric Rendering

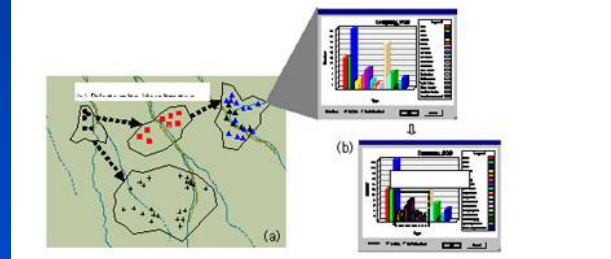


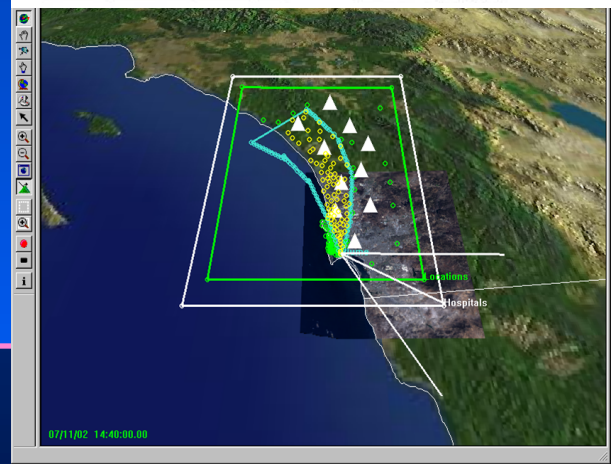
Information Visualization

- Need to handle:
  - Large amount of information
  - Multi-variable, multi-dimensional
  - Heterogeneous format
- Convert symbolic data to geometry data
- ~50% of brain's neurons associated with vision
- Insights and understanding of data
- Aid in exploring information



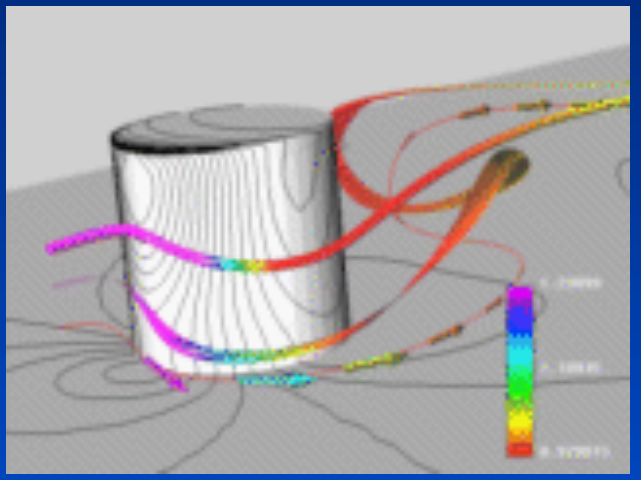






Volume Rendering Sources

- Mathematical model
  - CFD (Computational Fluid Dynamics)
  - FEM (Finite Element Models)
- Medical digitization
  - CT (Computed Tomography)
  - MRI (Magnetic Resonance Imaging)
  - Ultrasound (esp. 3-D)
- Other measured data
  - Financial analysis, sensor probes (e.g. thermocouples)



Data Volume geometries♪

- Cartesian: voxel grid, cubic, axis aligned
- Regular: rectangular cells e.g. CT
- Rectilinear: distance between cells non-uniform
- Structured (or curvilinear)
  - Non-rectangular
  - Common in CFD

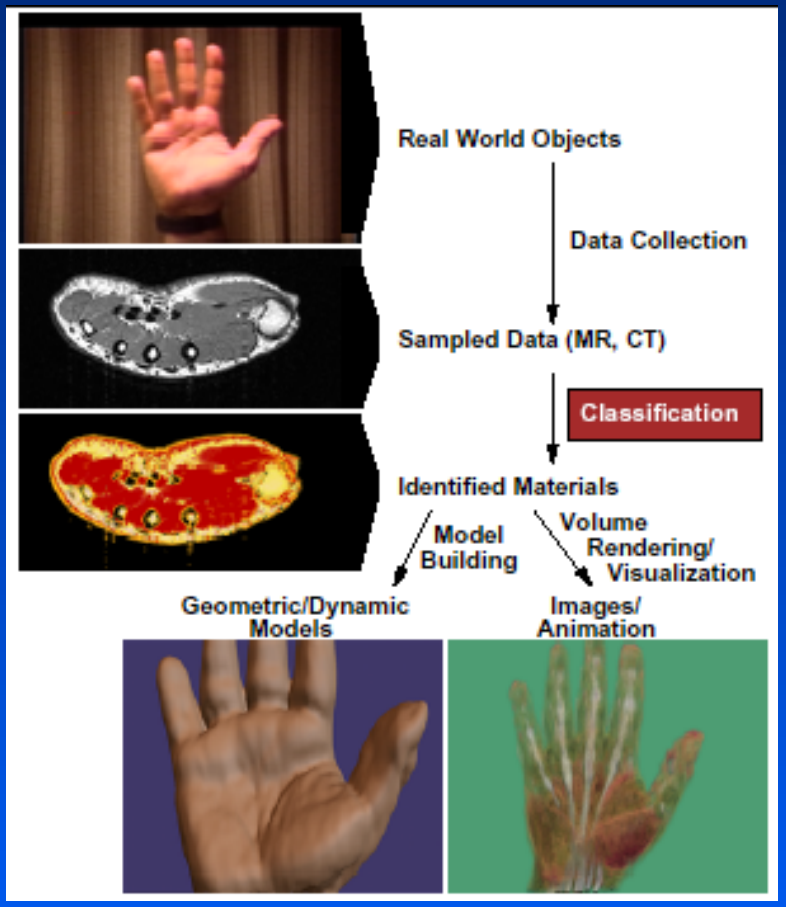
- Block structured
  - Several systems of above category in same data volume
- Unstructured
- Hybrid

Aliasing and Resampling

- Undersampling in collecting data: cannot improve using post processing
- Resampling may introduce aliasing
- Use of interpolation: must be careful not to compromise integrity of data by introducing information that is not there

Rendering

- Show one slice not-necessarily axis-aligned
  - Least ambitious, it is just 2-D
- Surface rendering
  - Segmentation
- Volume Rendering
  - Assign transparency and color to voxels

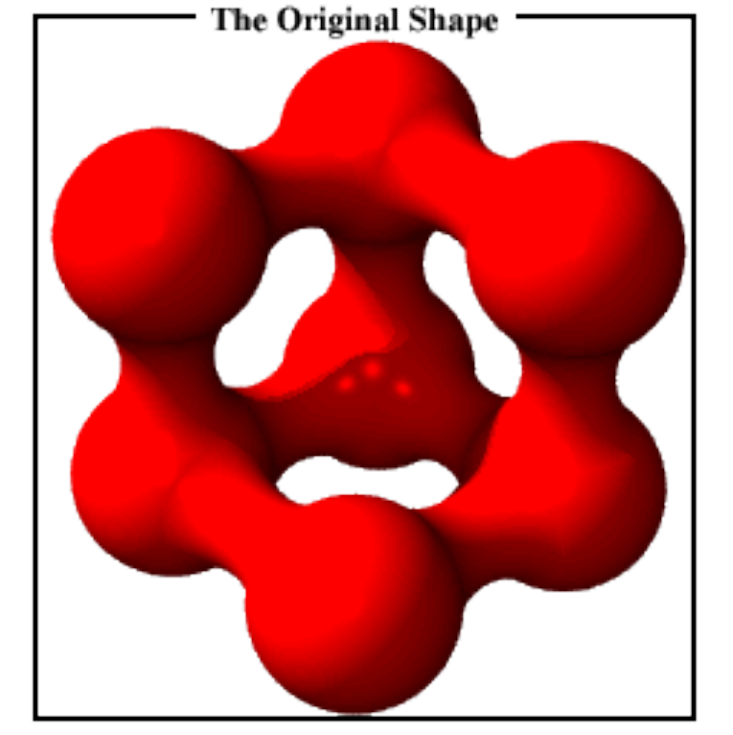


Connecting 2-D contour

- 2-D slice edge tracked to form contour
- “sew” adjacent contour to form 3-D surface
- Solving a 3-D problem using one method for 2 of the dimensions and a totally different method for the third dimension
- Problems for cases like bifurcation: how to sew two contours to one contour

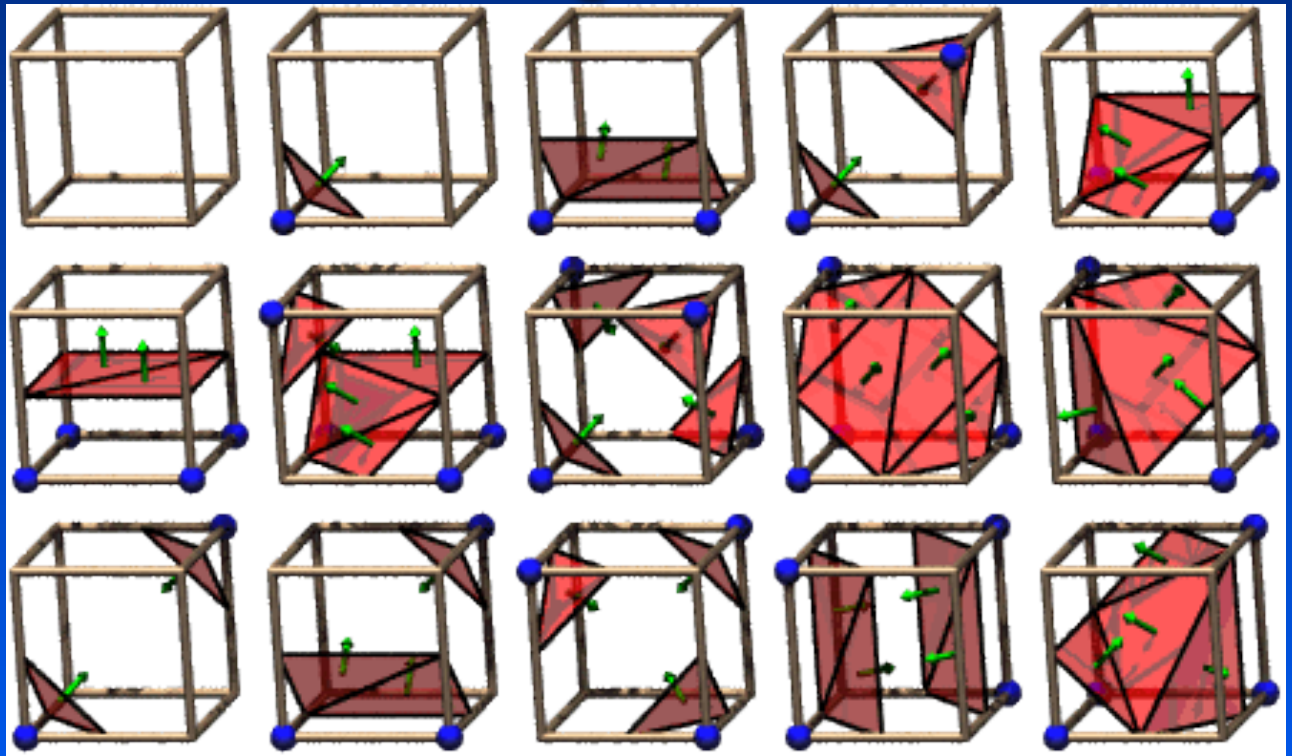
Isosurfaces by marching cubes

- Let $f(\mathbf{x})$ be the intensity value of the voxel $\mathbf{x}$
- Given an isosurface value k specified by the user, generate a polygonal mesh
- Approximate the surface where $f(\mathbf{x}) = k$

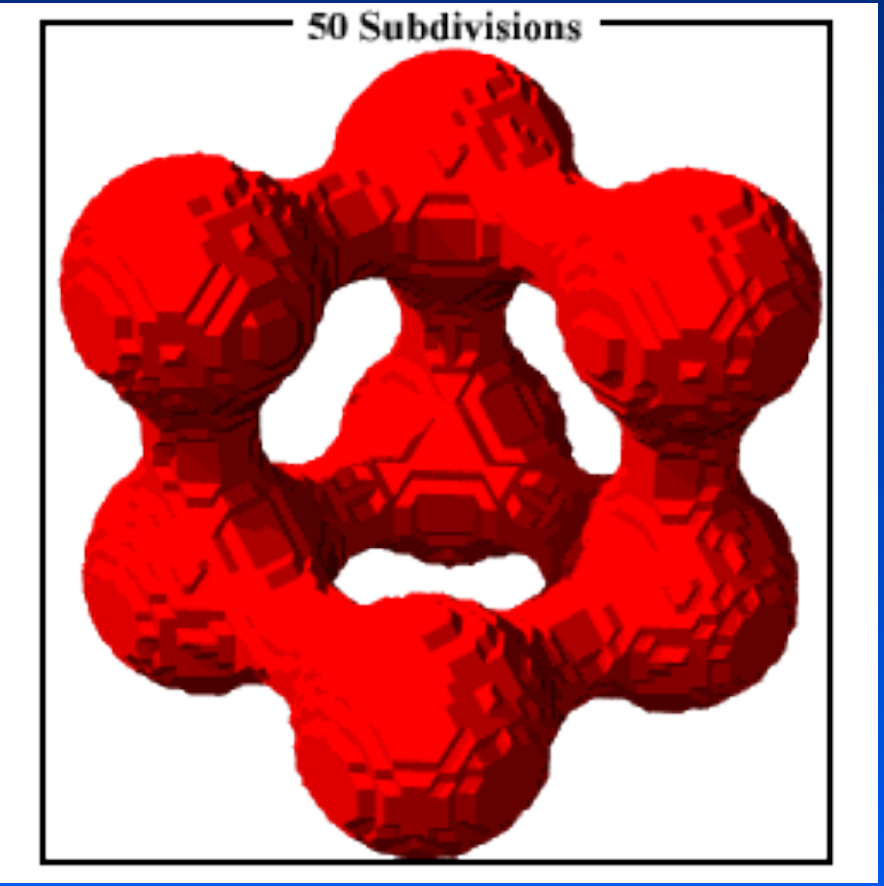


- Space is divided into small cubes (voxels)
- Evaluate $f(\mathbf{x})$ on the 8 vertices of each voxel
- Each vertex can be inside or outside:
  - If the 8 vertices are < k we are totally inside
  - If the 8 vertices are > k we are totally outside
  - If some of them are < k and some > k?

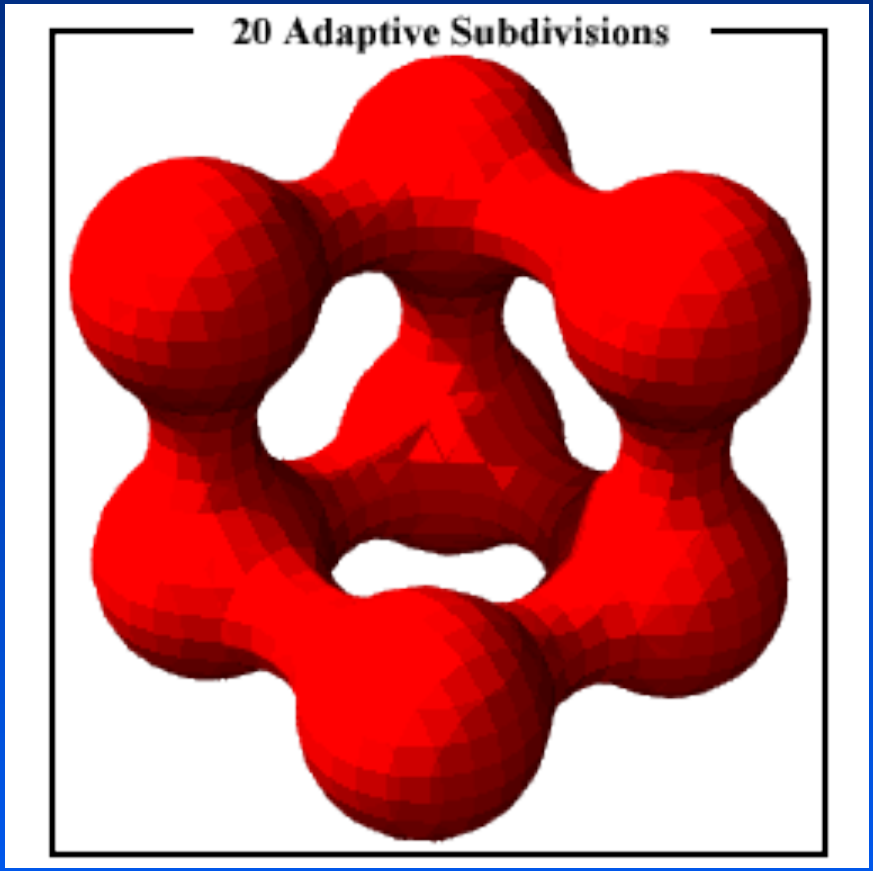
- We have 256 cases, which can be generalize to 15 cases
- Blue dots represent interior points
- Green arrows are normals pointing outside



- We can take the mid-point between the interior and exterior point
- The surface is not smooth



- Between an interior and exterior point, the point where $f(\mathbf{x}) = k$ can be found by an interpolation method



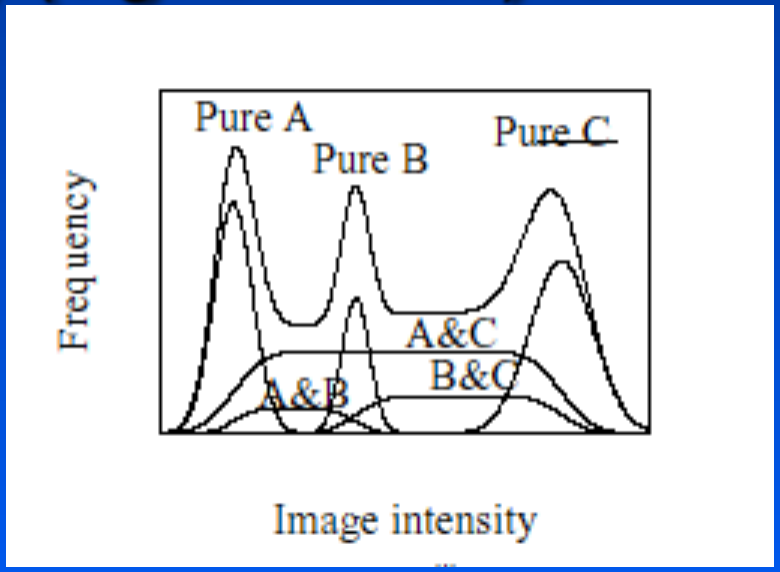
Volume rendering

- Isosurface can introduce false positives and false negatives (introduce artifacts and discard small features)
- May want to see whole volume

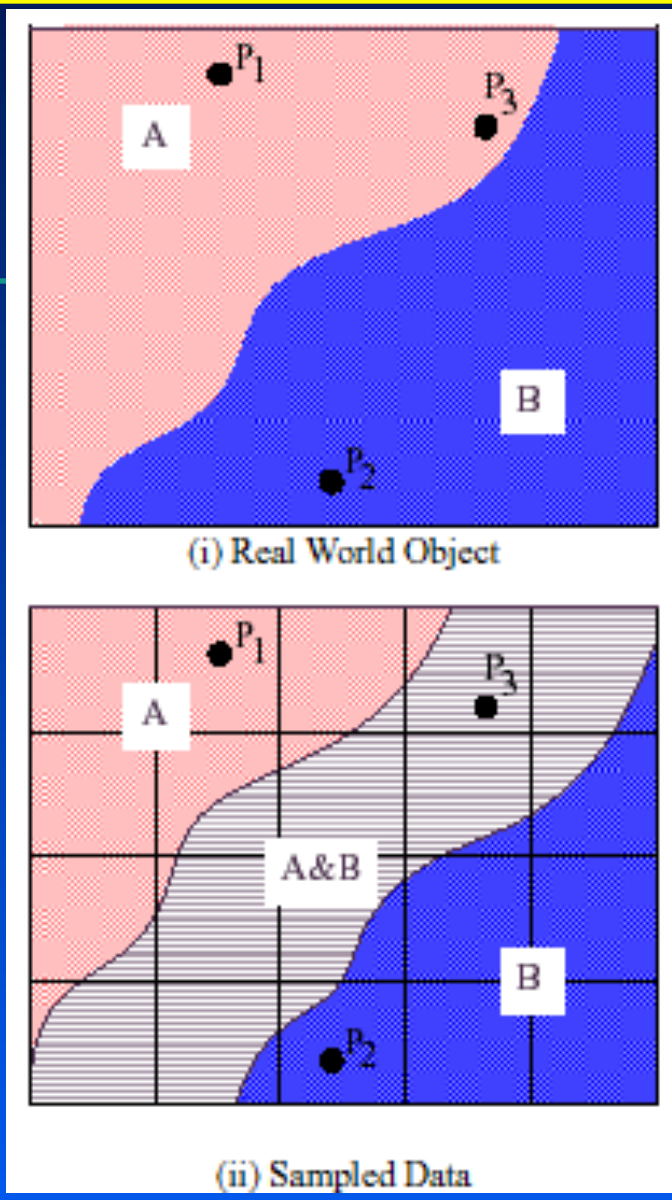
- Step 1: Classify each voxel in original data (bone, fat, etc.)
  - Assign color and transparency value (*transfer function*)
- To generate an image, for each pixel:
  - Step 2: Cast a ray by using viewing parameters
  - Step 3: Use compositing along the ray to compute the color of the pixel

Probabilistic Classification (using histogram)

- Distribution of each type (e.g. fat, bone, air…) known a priori
- Histogram of image intensity (e.g. CT value) for the data made
- Fit histogram with distribution of material

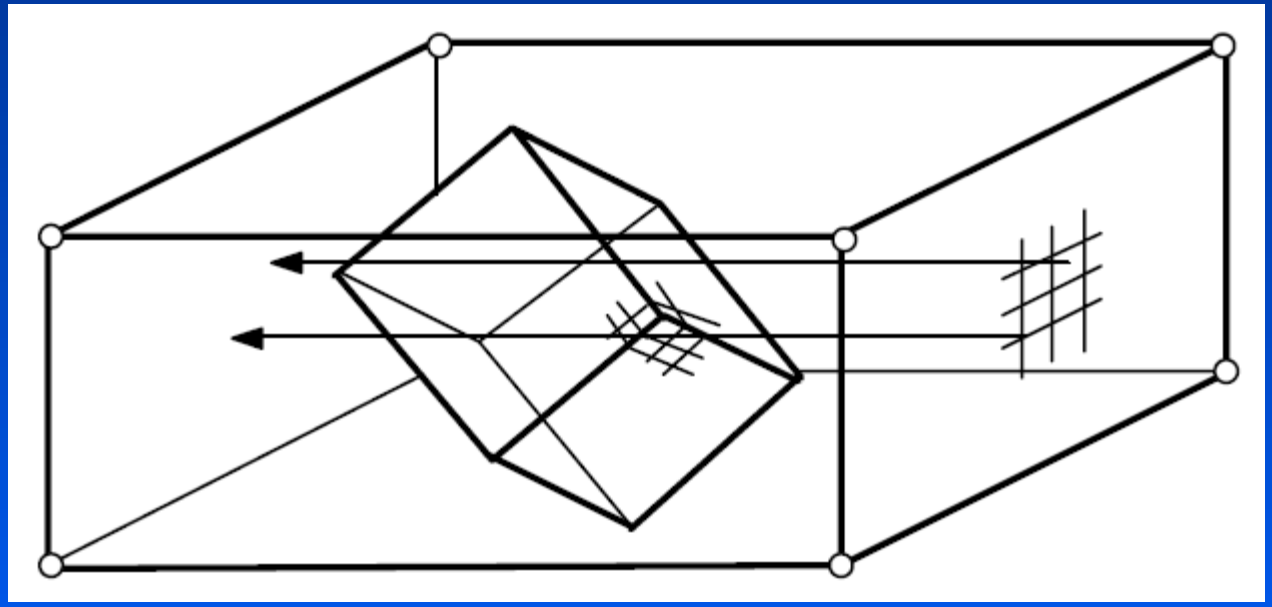


- Mapping between image intensity and type made (there may be overlapping mappings)
- Each voxel assigned a material percentage
  - Each voxel may straddle boundaries of regions of different materials

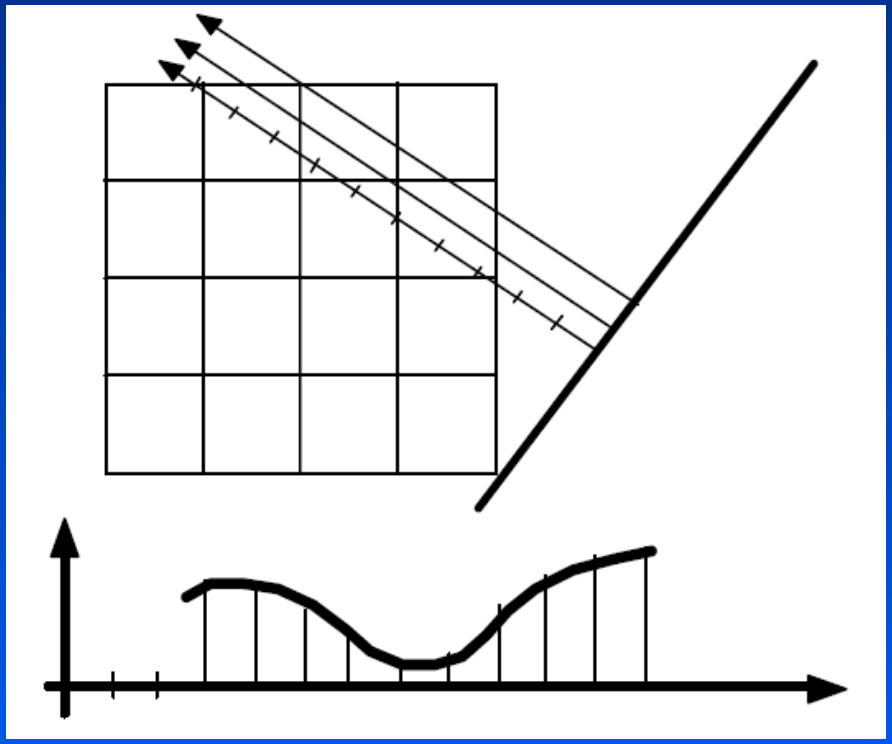


Step 2: Ray Casting

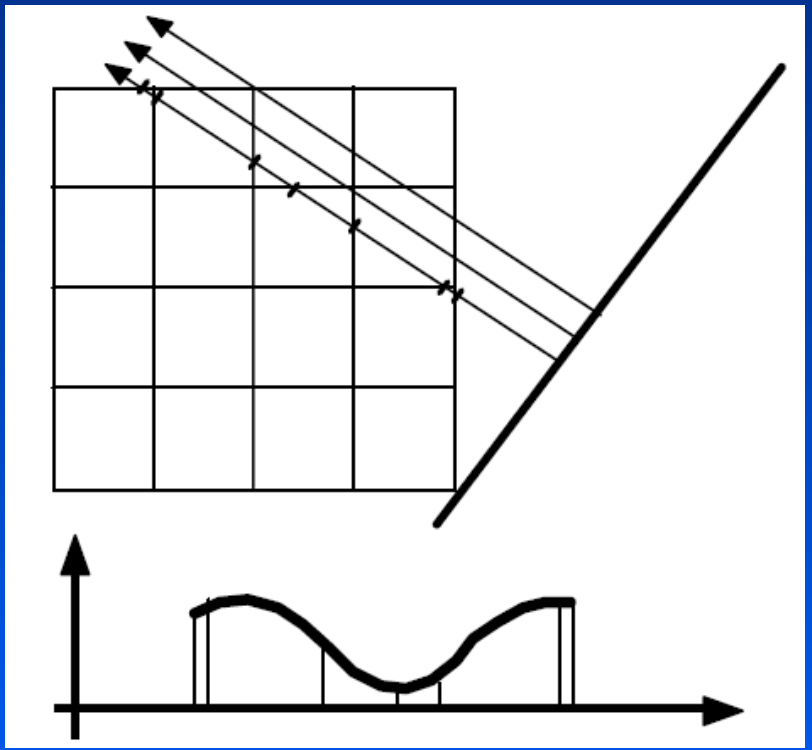
- 3D digital differential analyzer
- Traverse voxels, which values should we use?



- Uniform sampling: equal spacing in the ray

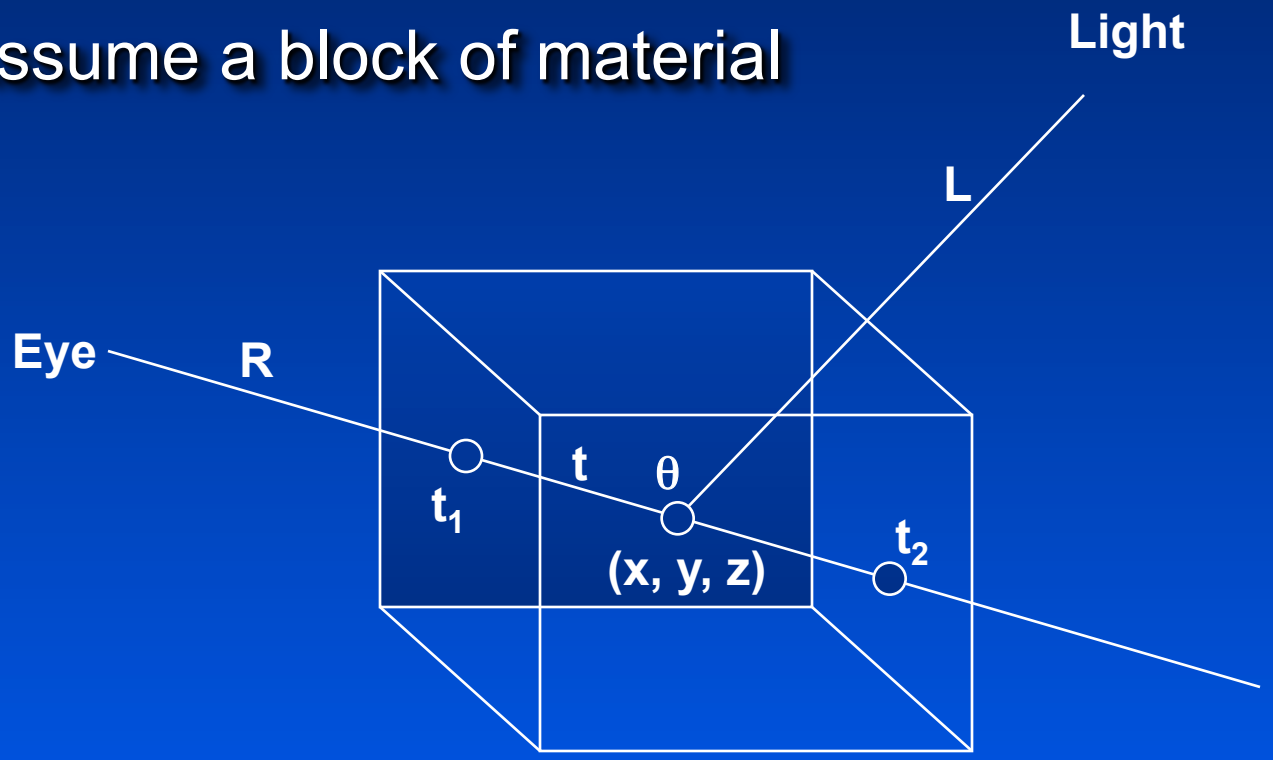


- Non-uniform sampling: at cell boundaries



Ray tracing (theoretical view)

- Assume a block of material

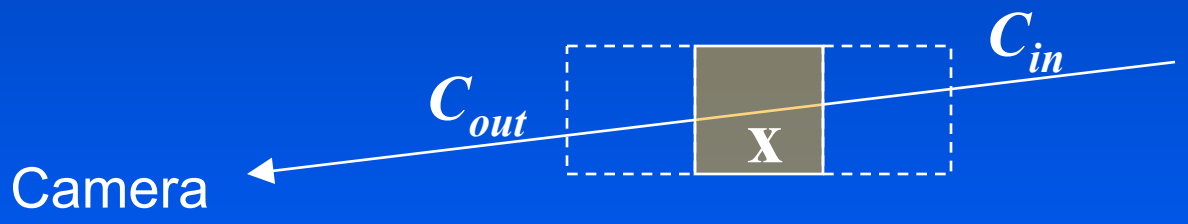


- Density at point: $D(x(t), y(t), z(t)) = D(t)$
- Illumination from light source at point: $I(x(t), y(t), z(t)) = I(t)$
  - Usually do not consider light attenuation on the way to point: same everywhere in volume
- Reflection function (phase function): P
- Illumination scattered along R from point t:
  - $I(t)D(t)P(\cos\theta)$
  - θ angle between R and L (light vector)

- Attenuation due to density along R from t_1 to t
  - $\exp(-\tau\int_{t_1}^{t} D(S)ds)$
  - τ constant, convert density to attenuation
- Intensity arriving at eye along R due to all elements along ray from t_1 to t_2
  - $\int_{t_1}^{t_2}[\exp(-\tau\int_{t_1}^{t} D(S)ds)](I(t)D(t)P(\cos\theta))dt$
- For discrete elements, summation

Volumetric compositing

- Back to front traversal
- $C_{out} = C_{in}(1-\alpha(\mathbf{x})) + C(\mathbf{x})\alpha(\mathbf{x})$
  - C_{out}: accumulated color after the voxel
  - C_{in}: accumulated color before the voxel
  - $C(\mathbf{x})$: color of the voxel
  - $\alpha(\mathbf{x})$: opacity of the voxel



- Intensity due to set of voxels that intercept the ray:
  - $$C(\mathbf{R}) = \sum_{k=0}^{K} \{C(\mathbf{x}_k)\alpha(\mathbf{x}_k) \prod_{i=k+1}^{K} (1 - \alpha(\mathbf{x}_i))\}$$
  - $\mathbf{x}_k$: k^{th} voxel along the ray **R**
  - k=0: background
  - Π term gives attenuation due to all voxels in front

Semi-transparent Gel + Surfaces

- Opaque voxels blocks every voxel behind it
- How to detect opaque voxels?
  - Compute normals as volume gradients
  - Marching cubes
- If we have normals, we can add a shading scheme by assuming light
- Both methods can add false contours

Volume gradients

- Let $f(x,y,z)$ be the intensity value of the voxel (x,y,z)
- The normal can be computed as:

$$\mathbf{N}(x,y,z) = \begin{bmatrix} f(x+1,y,z) - f(x-1,y,z) \\ f(x,y+1,z) - f(x,y-1,z) \\ f(x,y,z+1) - f(x,y,z-1) \end{bmatrix}$$

- Here we are using the 6-neighbors of (x,y,z)
- If $|\mathbf{N}(x,y,z)|$ is greater than some threshold, the voxel is opaque

Segmentation

- Used to identify parts of volume
- E.g. tissue, tumor, fat, etc.
- Easy classification impossible since same/similar CT/MR value for each part
- Hand segment using domain knowledge
- Automatic/assisted segmentation
  - Seed expansion/contraction (e.g. balloon)
  - Snakes