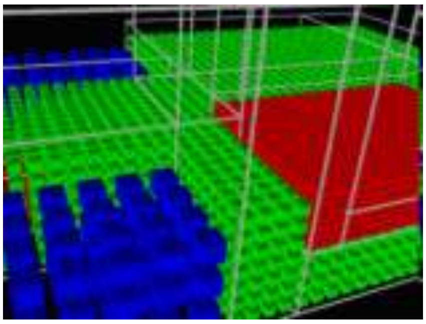
Introduction to volumetric visualization

Incluindo slides disponibilizados por
Chuck Hansen e Luciana Nedel

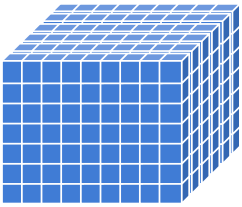
C. Freitas (2009)

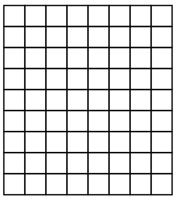
Volumetric data

- Data associated to positions or regions in a 3D space
  - Structured grids

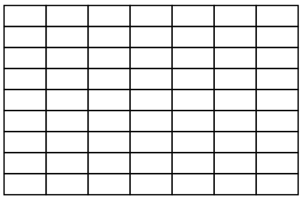


Wikipedia (Berkeley)

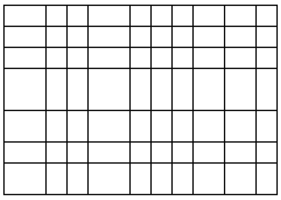




cartesian



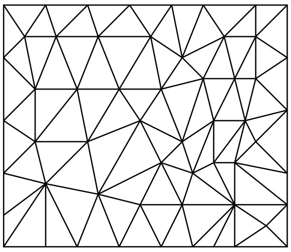
regular

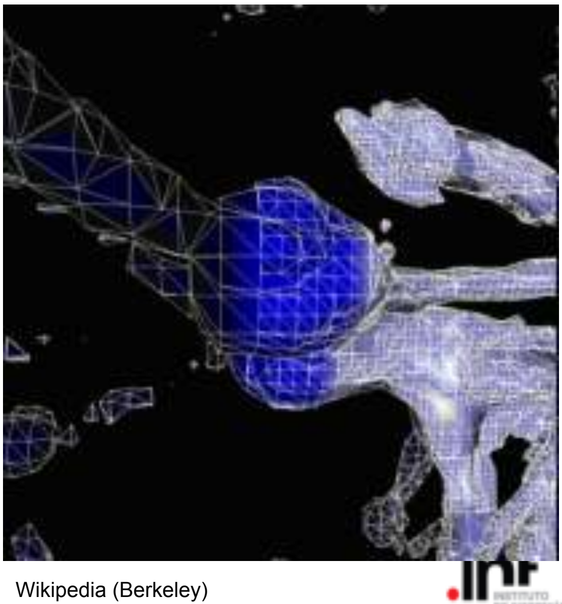


rectilinear

Volumetric data

- Data associated to positions or regions in a 3D space
  - Unstructured grids

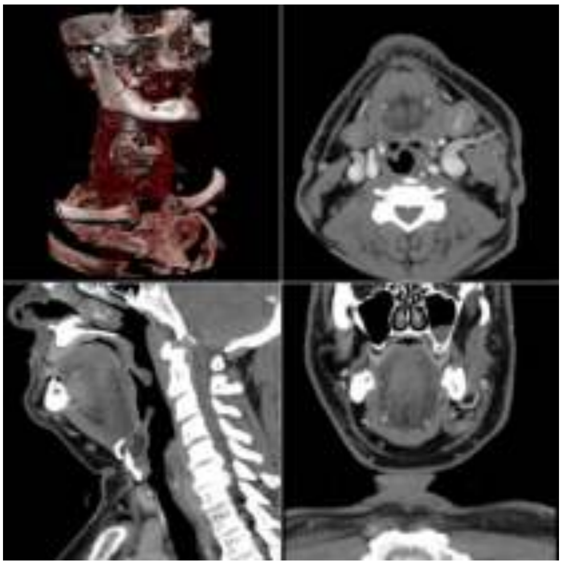


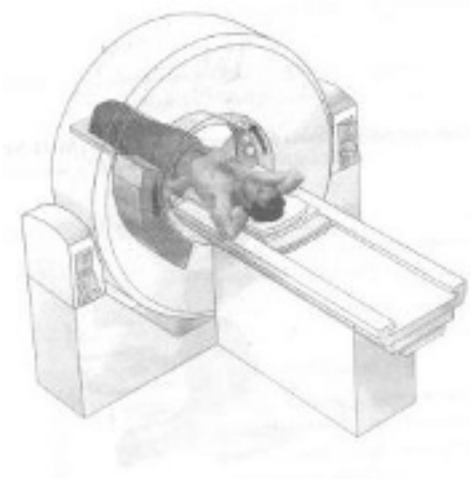


Wikipedia (Berkeley)

Acquisition: structured

- CT - Computed tomography

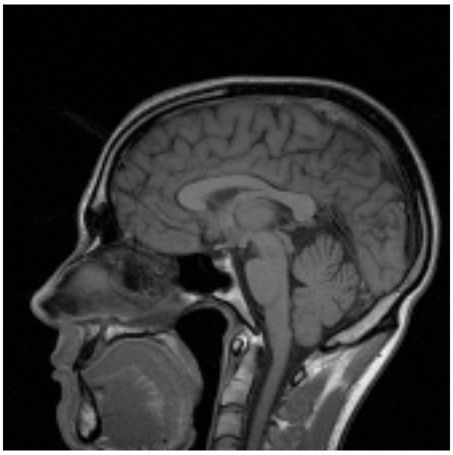




Hounsfield 1967

Acquistion: structured

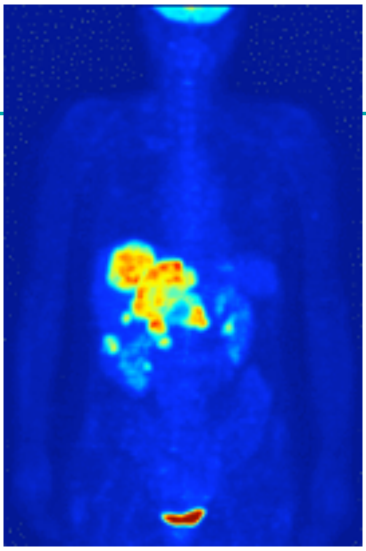
- MRI - Magnetic Resonance Imaging
  - A large magnetic field is applied to the object, which aligns hydrogen nuclei
  - Water content in the tissue is measured



Lauterbur 73

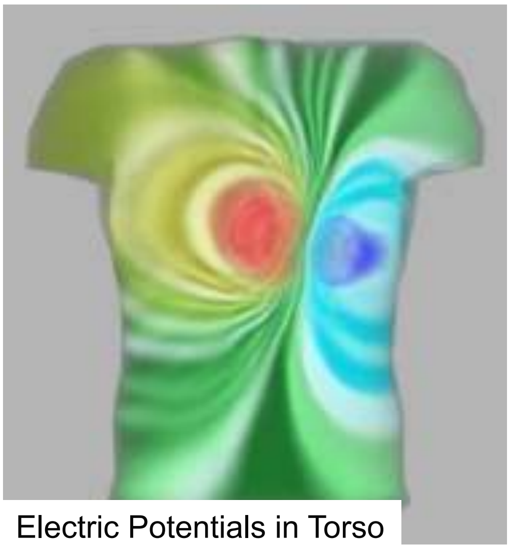
Acquisition: structured

- PET - Positron Emission Tomography
  - A radioactive isotope (into a medium) is injected into the body
  - Positron emission from decay interacts with electrons to create gamma rays
  - Gamma rays are detected to find position of isotopes in body

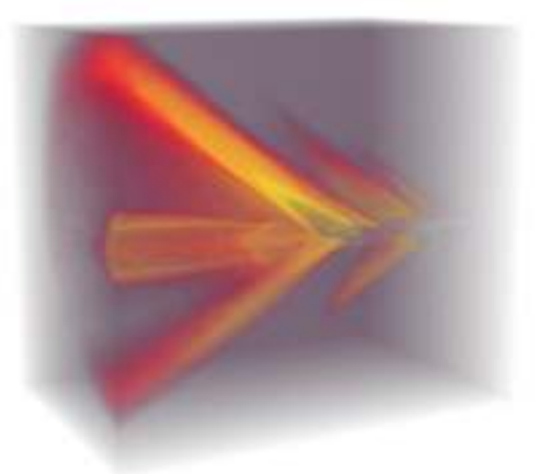


Computational simulation: (un)structured

- Computational fluid dynamics
- Structural mechanics



Electric Potentials in Torso
[MacLeod et al 94]

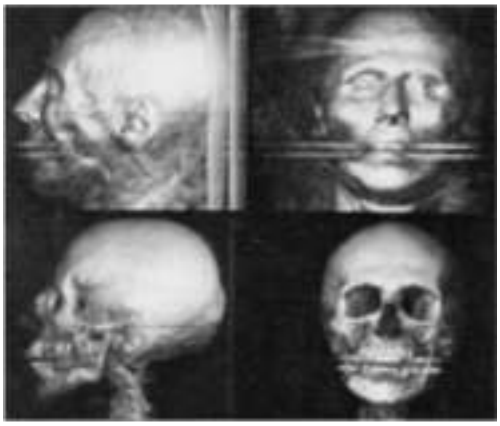


Turbulence around fighter jet
[Neely and Batina 92]

Visualization

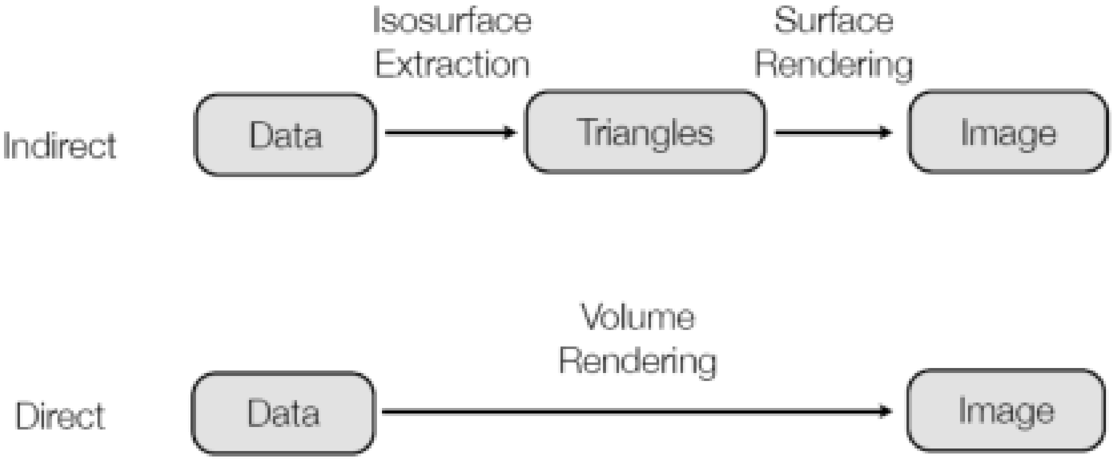
- Each voxel can contribute to the image
- Voxels can be discarded from the image depending on user interest on specific strutuctures within the volume

Ray casting [Levoy 88]

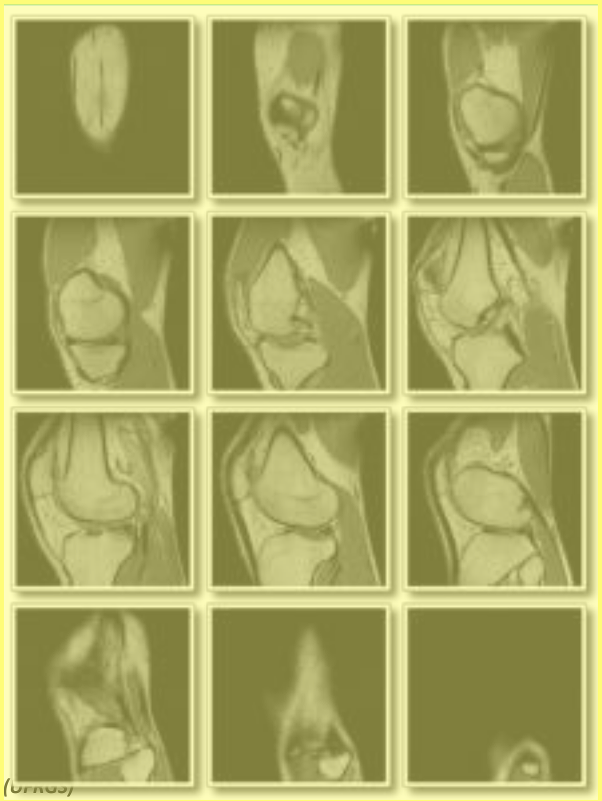


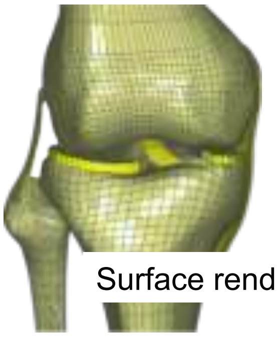
http://www.fovia.com/gallery.php

Volume visualization approaches

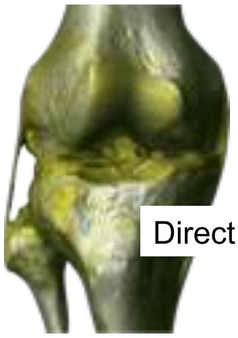


Volume visualization approaches





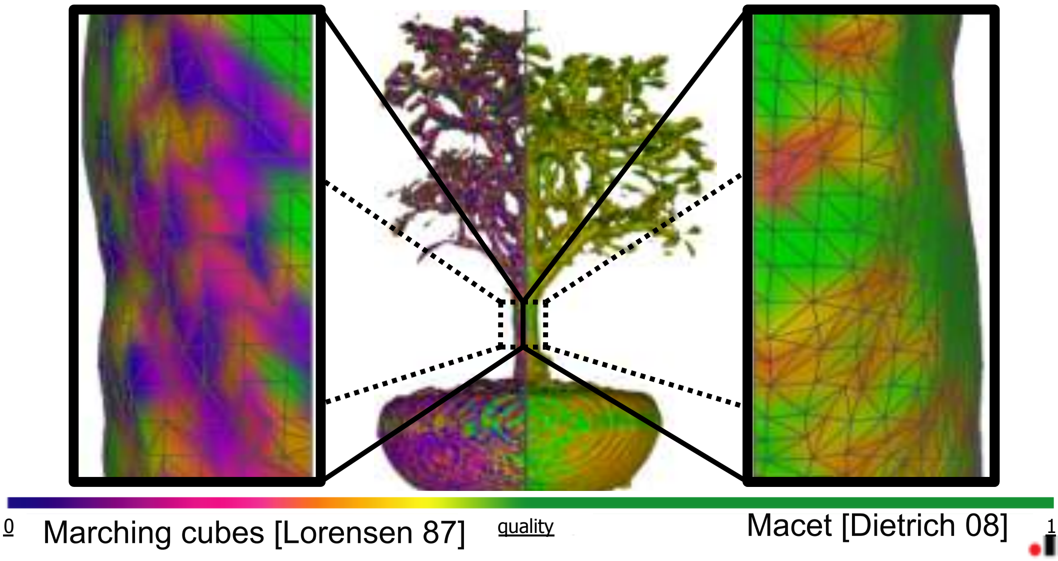
Surface rendering



Direct rendering

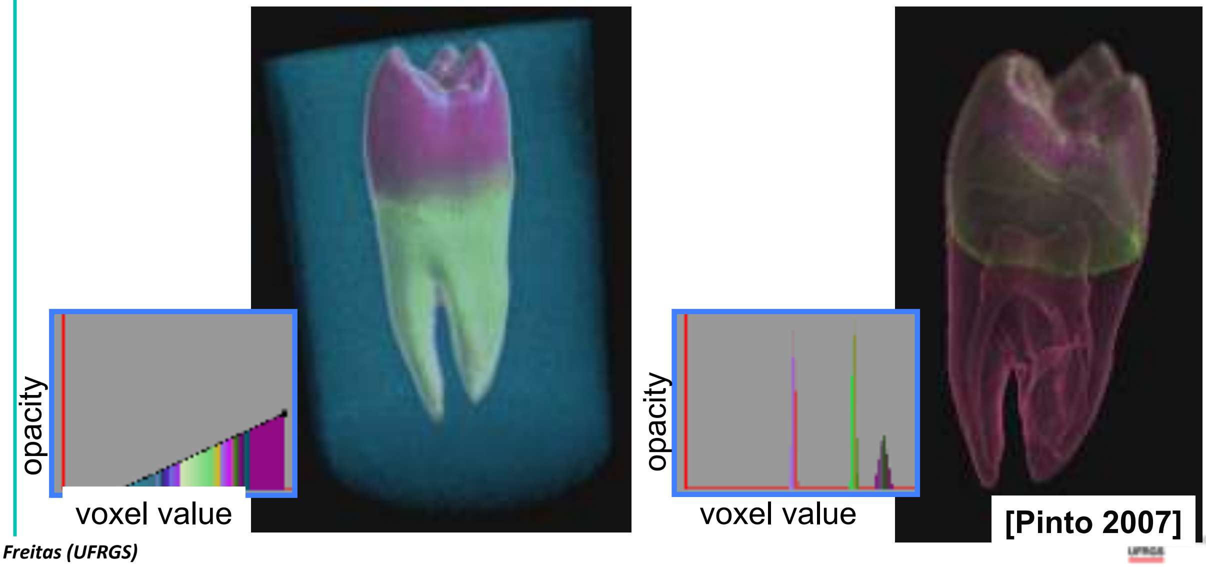
Surface rendering

- An isosurface is extracted from the data set
- Different methods extract different meshes



Direct volume rendering

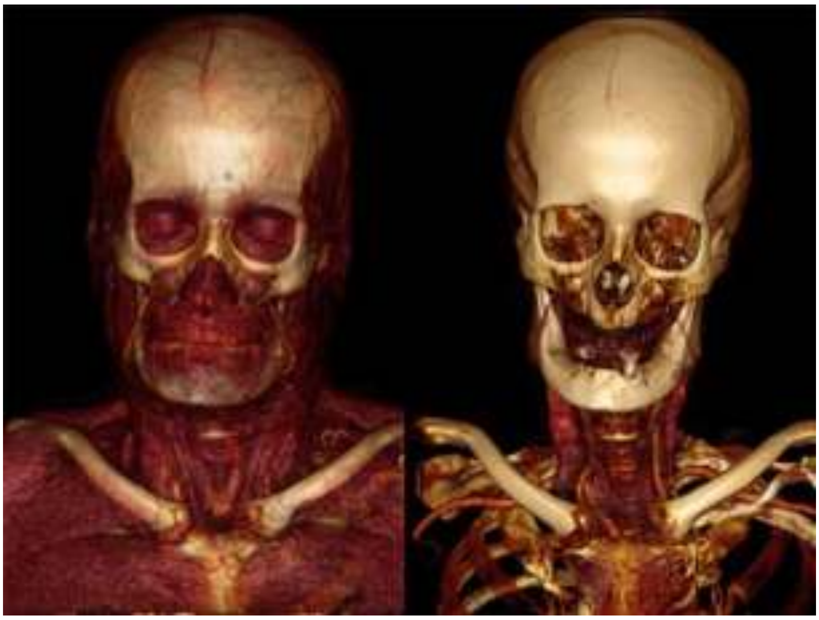
- Mapping from value to opacity and/or color
- Different mappings produce different images



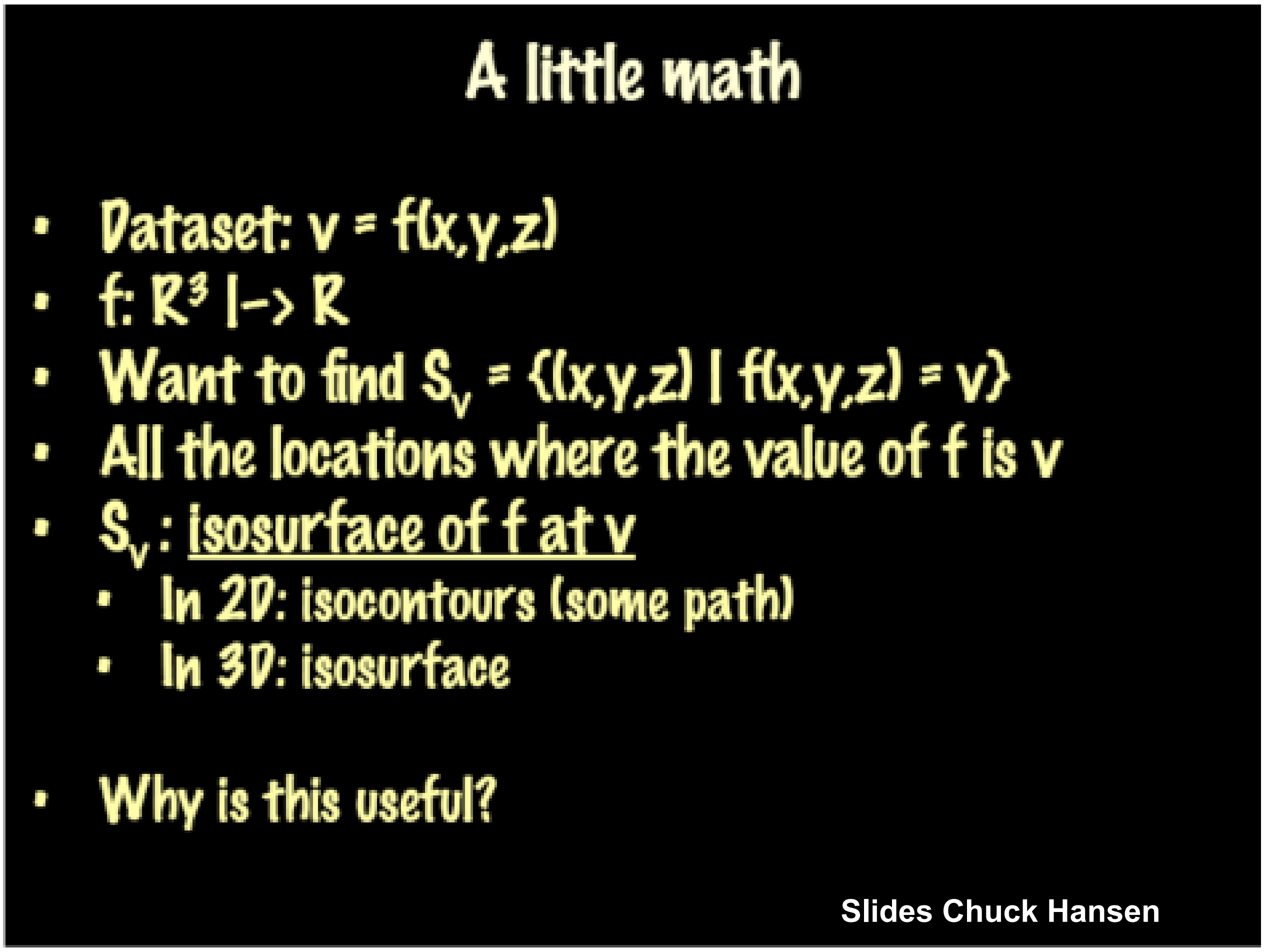
[Pinto 2007]

Isosurface extraction

- How do we specify the part of the volume corresponding to the desired inner structure?



C. Freitas (UFRGS)



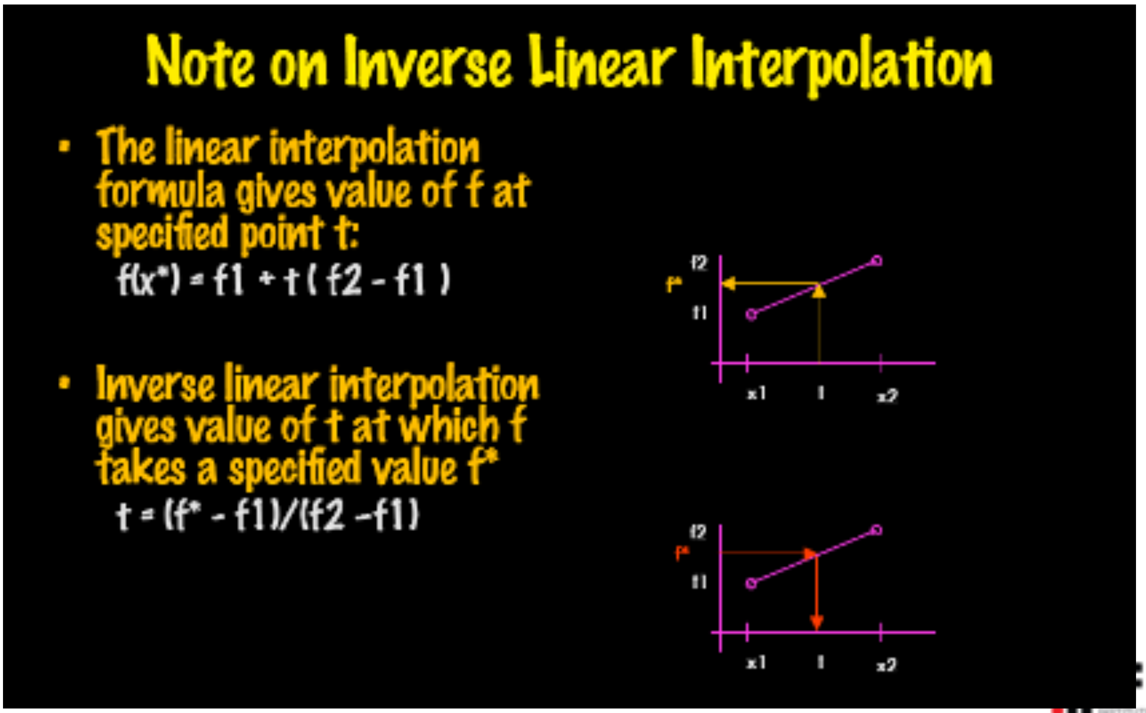
A little math

- Dataset: $v = f(x,y,z)$
- $f: R^3 \mapsto R$
- Want to find $S_v = \{(x,y,z) \mid f(x,y,z) = v\}$
- All the locations where the value of f is v
- S_v : <u>isosurface of f at v</u>
  - In 2D: isocontours (some path)
  - In 3D: isosurface

- Why is this useful?

Slides Chuck Hansen

Surface extraction



Note on Inverse Linear Interpolation

- The linear interpolation formula gives value of f at specified point t:

  $f(x^*) = f1 + t(f2 - f1)$

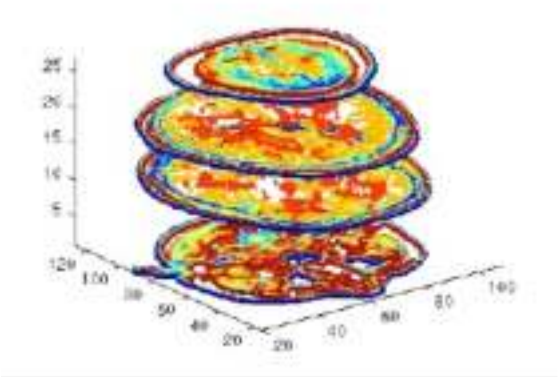
- Inverse linear interpolation gives value of t at which f takes a specified value f*

  $t = (f^* - f1)/(f2 - f1)$

Slides Chuck Hansen

Surface extraction

- Take a slice through the 3D volume, often orthogonal to one of the axes
- Obtain the 2D contour of the structures in each slice
- Connect adjacent contours forming polygons



Surface extraction

7

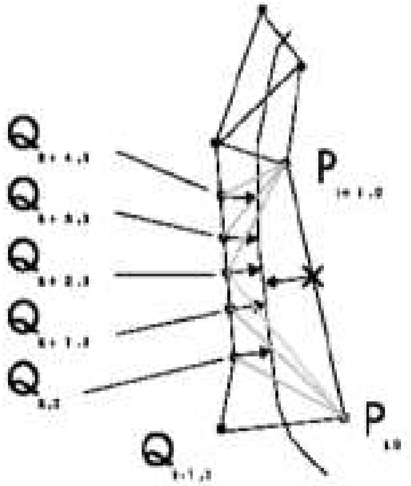
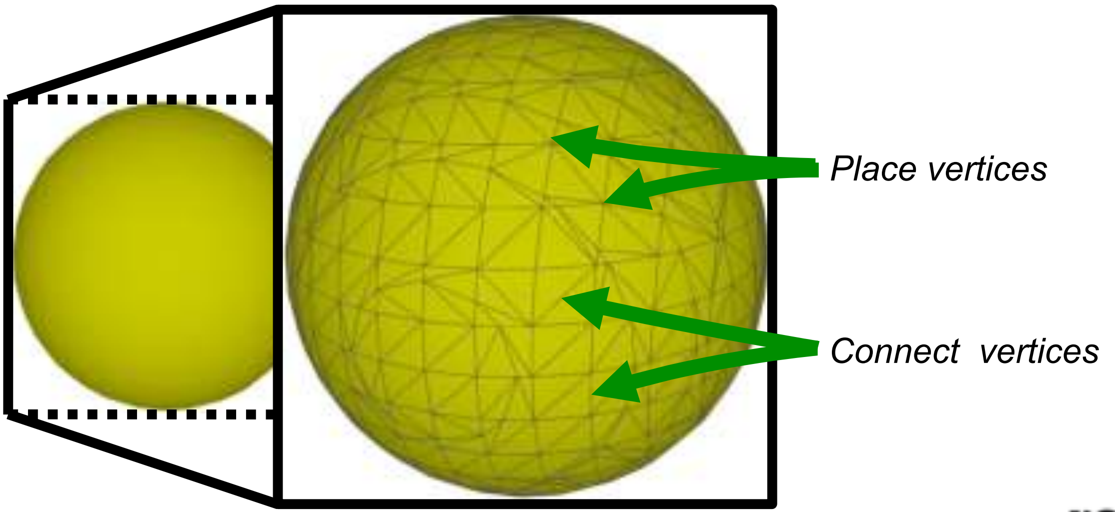
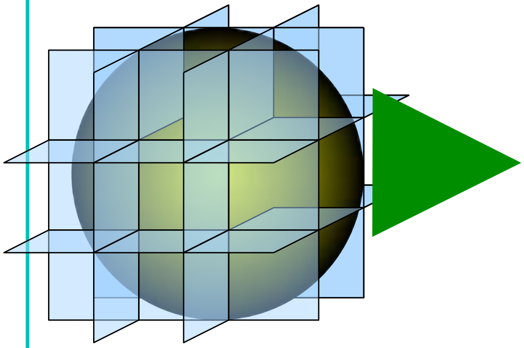


Fig. 6. Vertices $Q_{k,0}$ through $Q_{k+4,0}$ are connected to $P_{i,0}$ or $P_{i+1,0}$ resp. due to their correspondences on the medial axis.

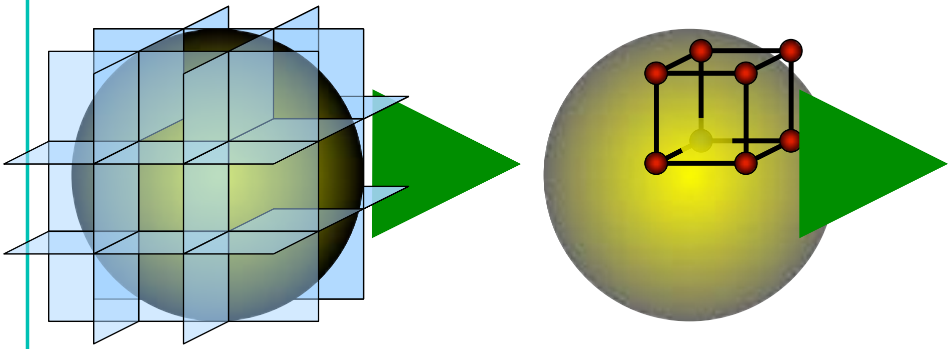
Surface extraction



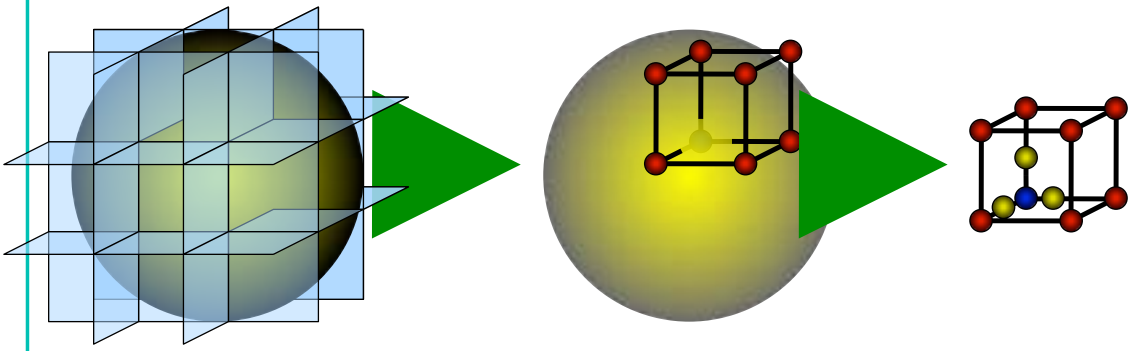
Marching Cubes (MC) (Lorensen e Cline, 1987)



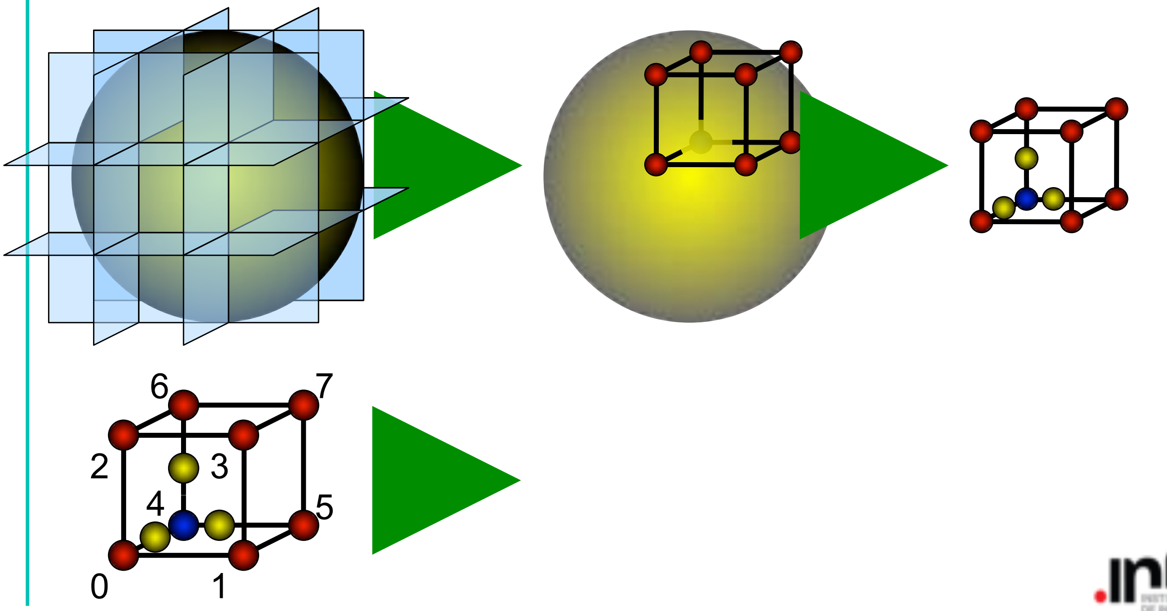
Marching Cubes (MC) (Lorensen e Cline, 1987)



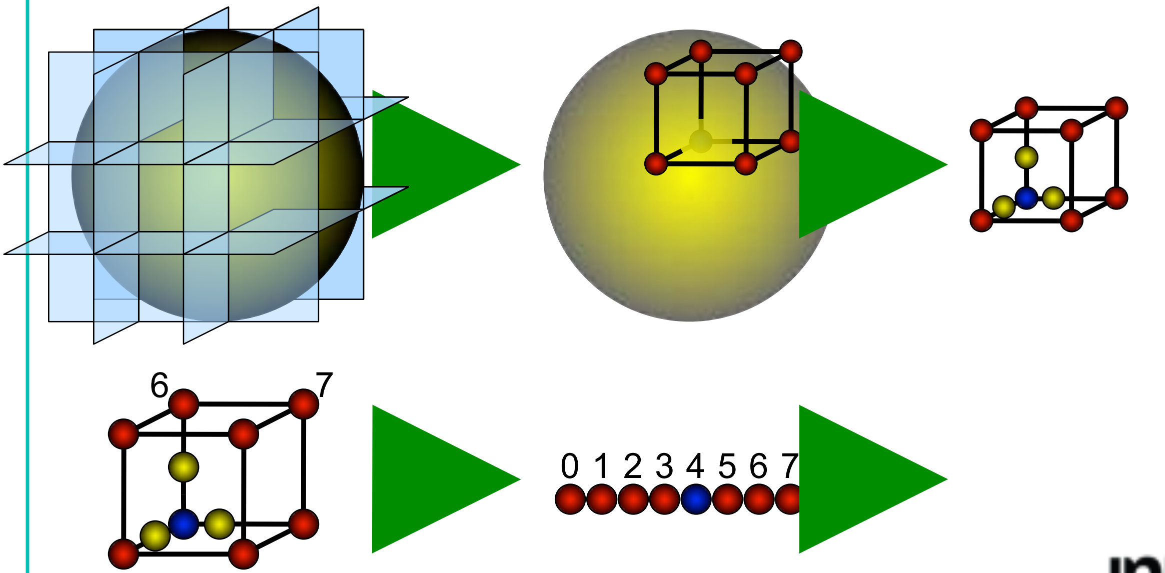
Marching Cubes (MC) (Lorensen e Cline, 1987)



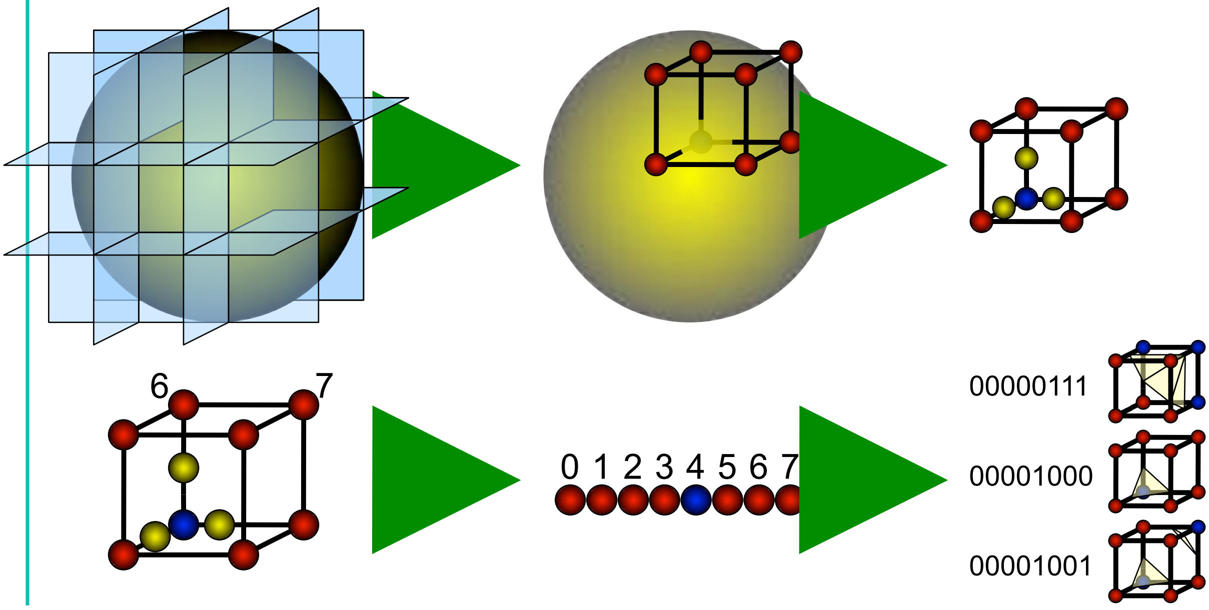
Marching Cubes (MC) (Lorensen e Cline, 1987)



Marching Cubes (MC) (Lorensen e Cline, 1987)



Marching Cubes (MC) (Lorensen e Cline, 1987)



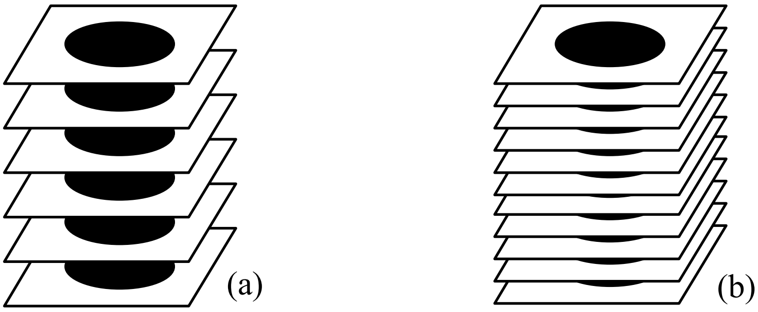
00000111

00001000

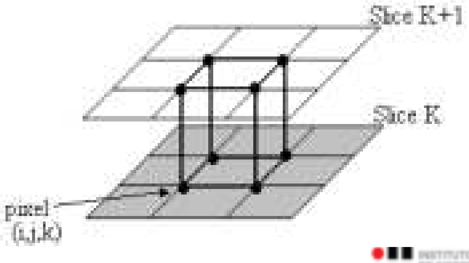
00001001

Surface extraction

- Marching cubes [Lorensen and Cline 87]

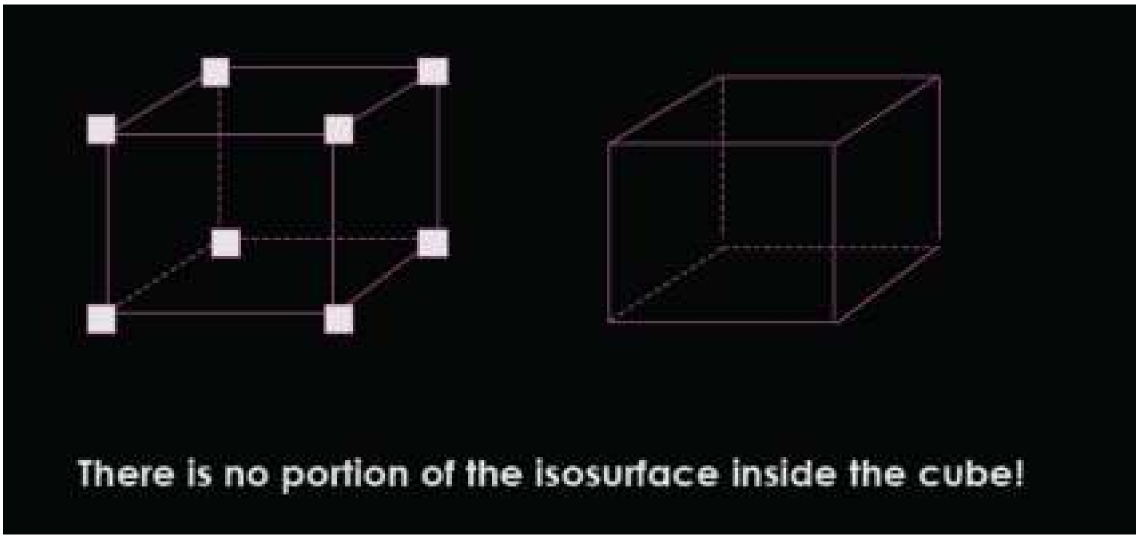


- Two adjacent slices form cubes

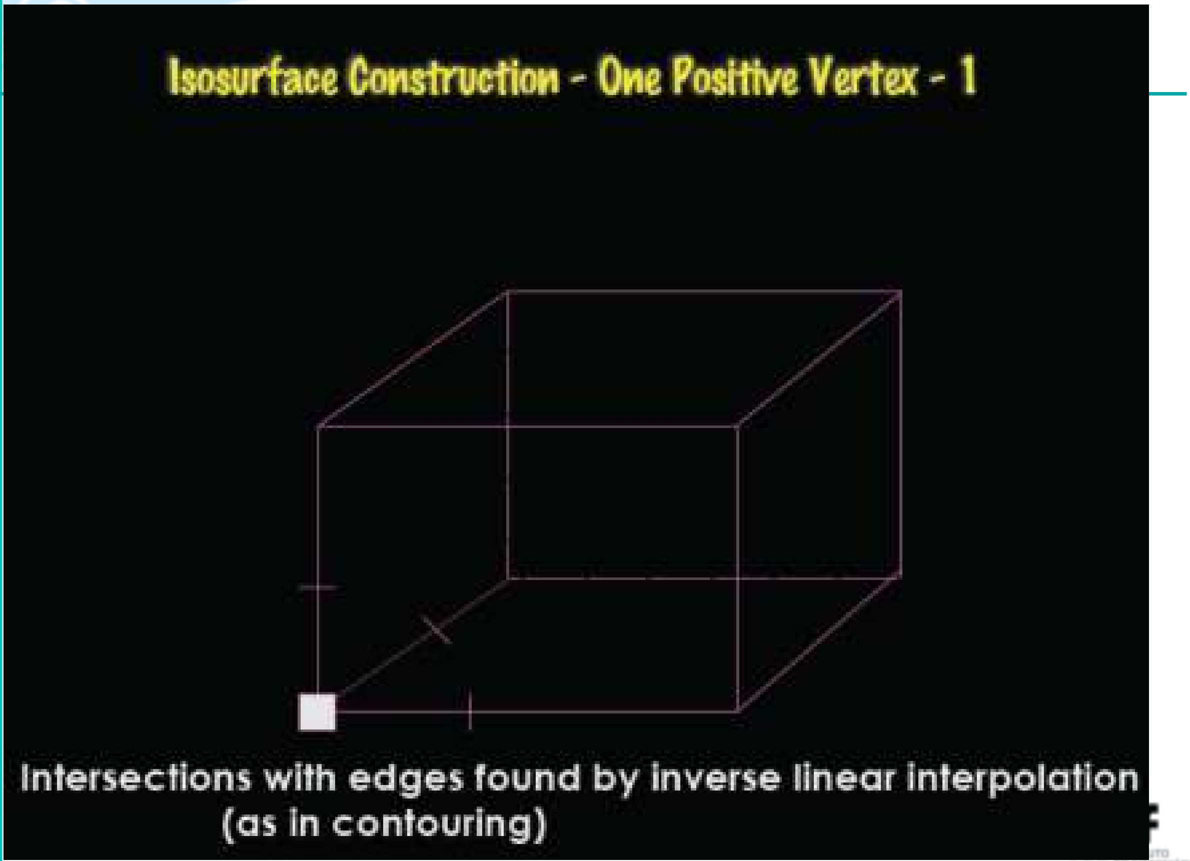


Marching cubes (Lorensen e Cline, 1987)

- Observing how a surface “passes through” a voxel

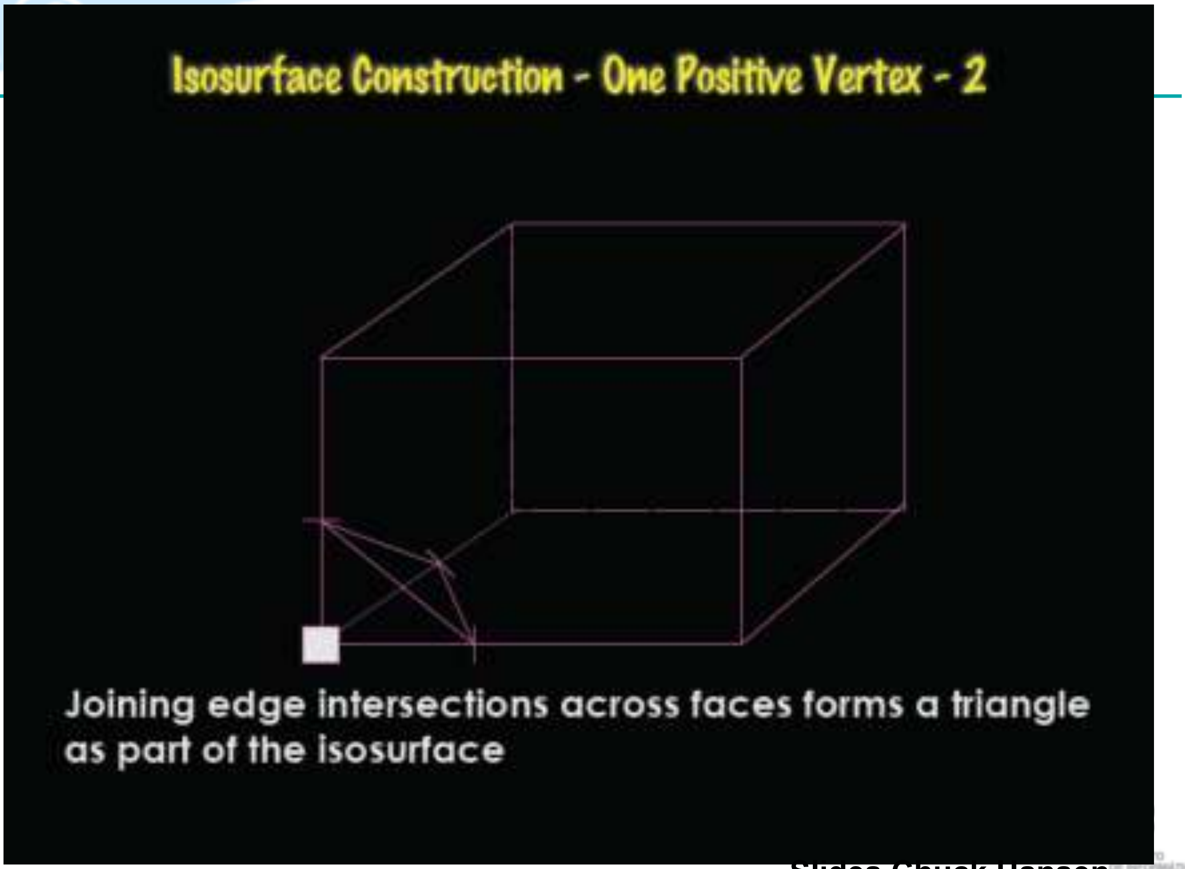


There is no portion of the isosurface inside the cube!



Isosurface Construction - One Positive Vertex - 1

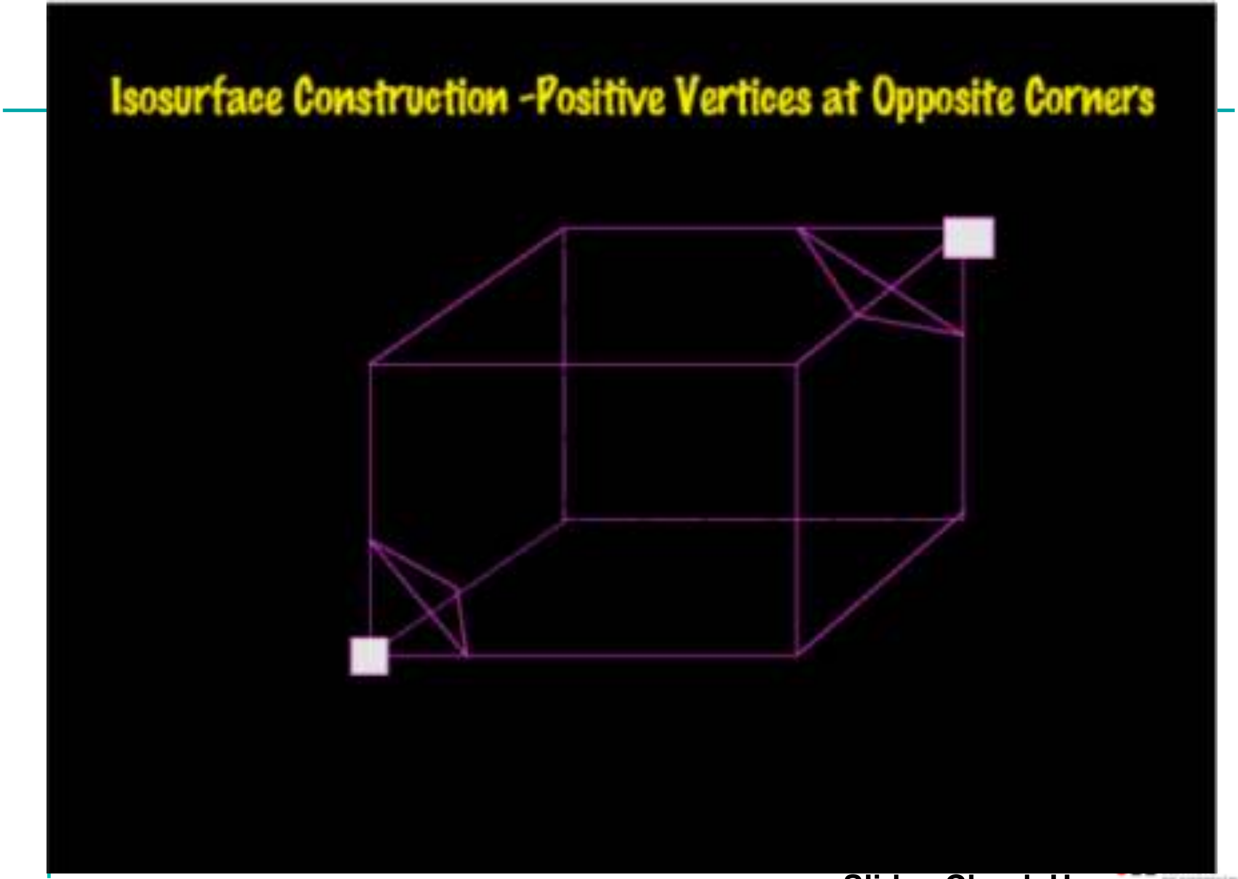
Intersections with edges found by inverse linear interpolation (as in contouring)



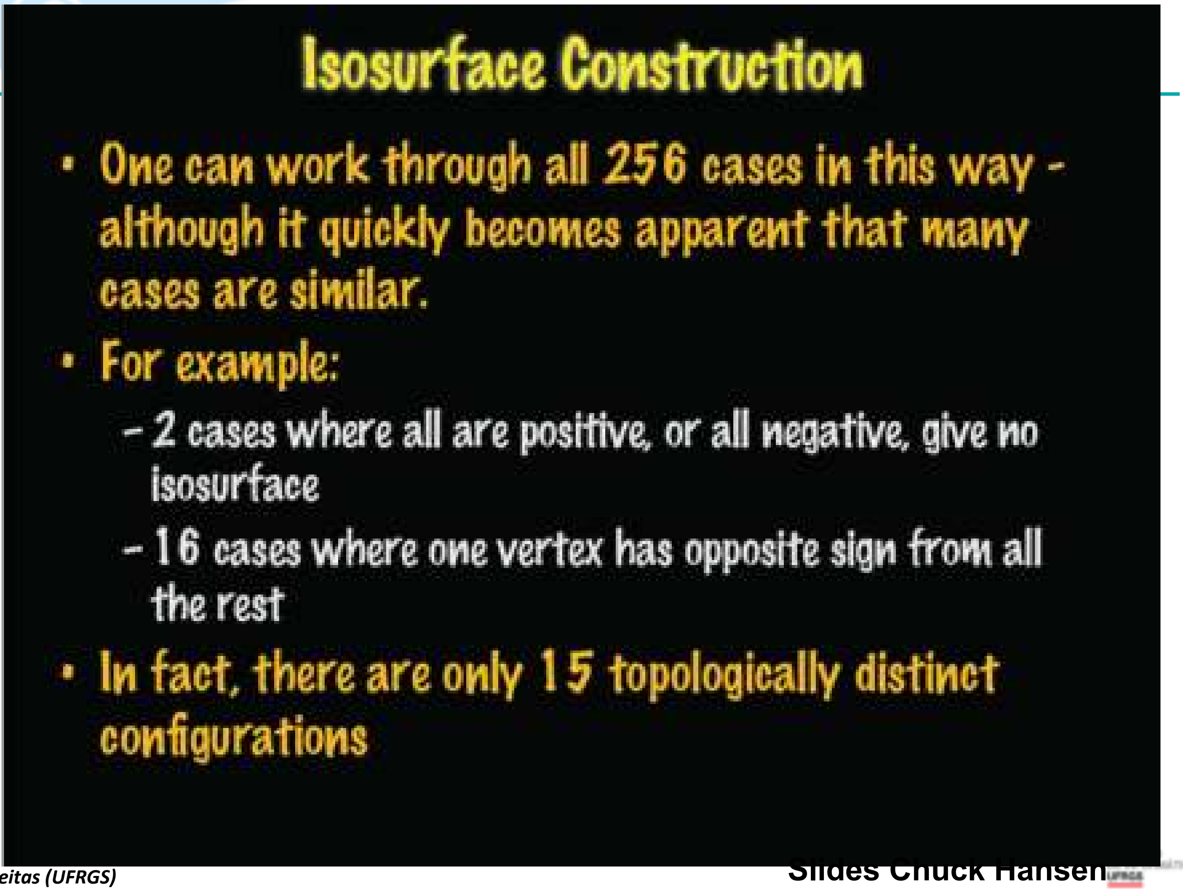
Isosurface Construction - One Positive Vertex - 2

Joining edge intersections across faces forms a triangle as part of the isosurface

Slides Chuck Hansen



Isosurface Construction -Positive Vertices at Opposite Corners

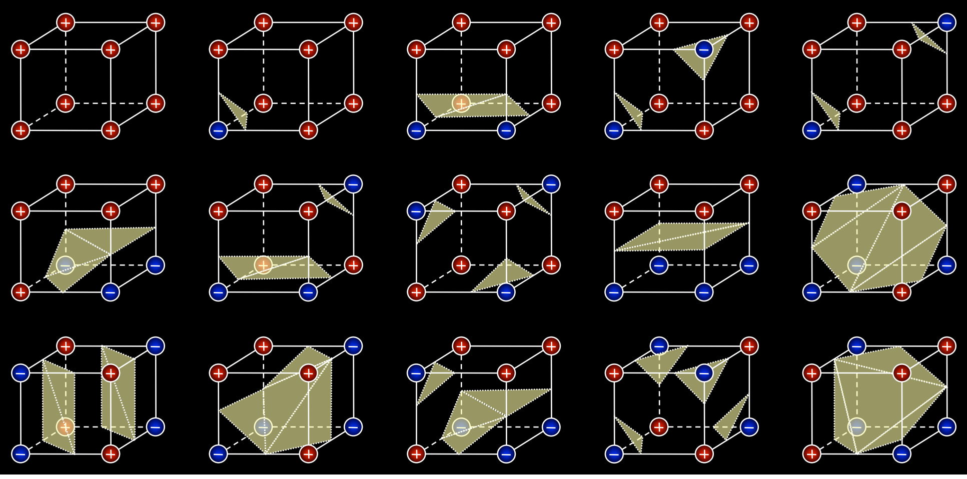


Isosurface Construction

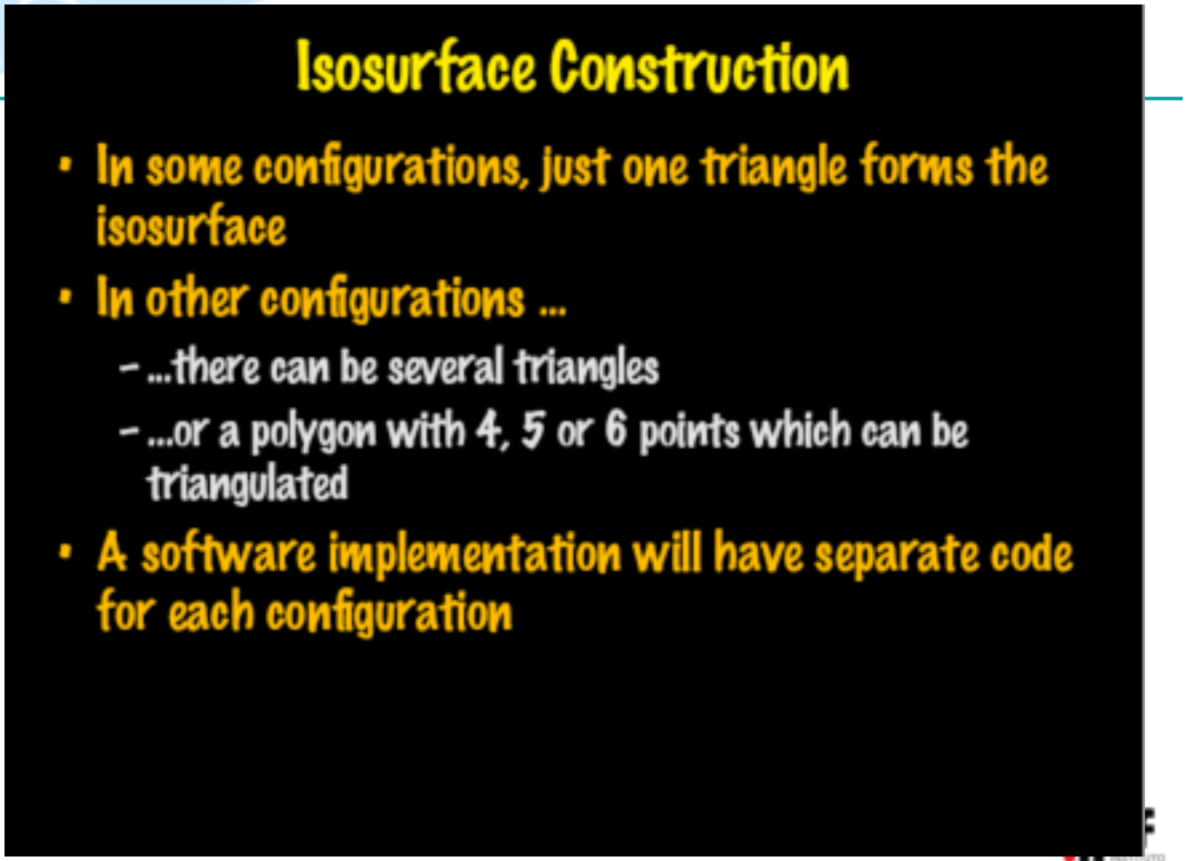
- One can work through all 256 cases in this way - although it quickly becomes apparent that many cases are similar.
- For example:
  - 2 cases where all are positive, or all negative, give no isosurface
  - 16 cases where one vertex has opposite sign from all the rest
- In fact, there are only 15 topologically distinct configurations

Marching Cubes (MC) (Lorensen e Cline, 1987)

- 15 cases to be analyzed while moving the logical cube



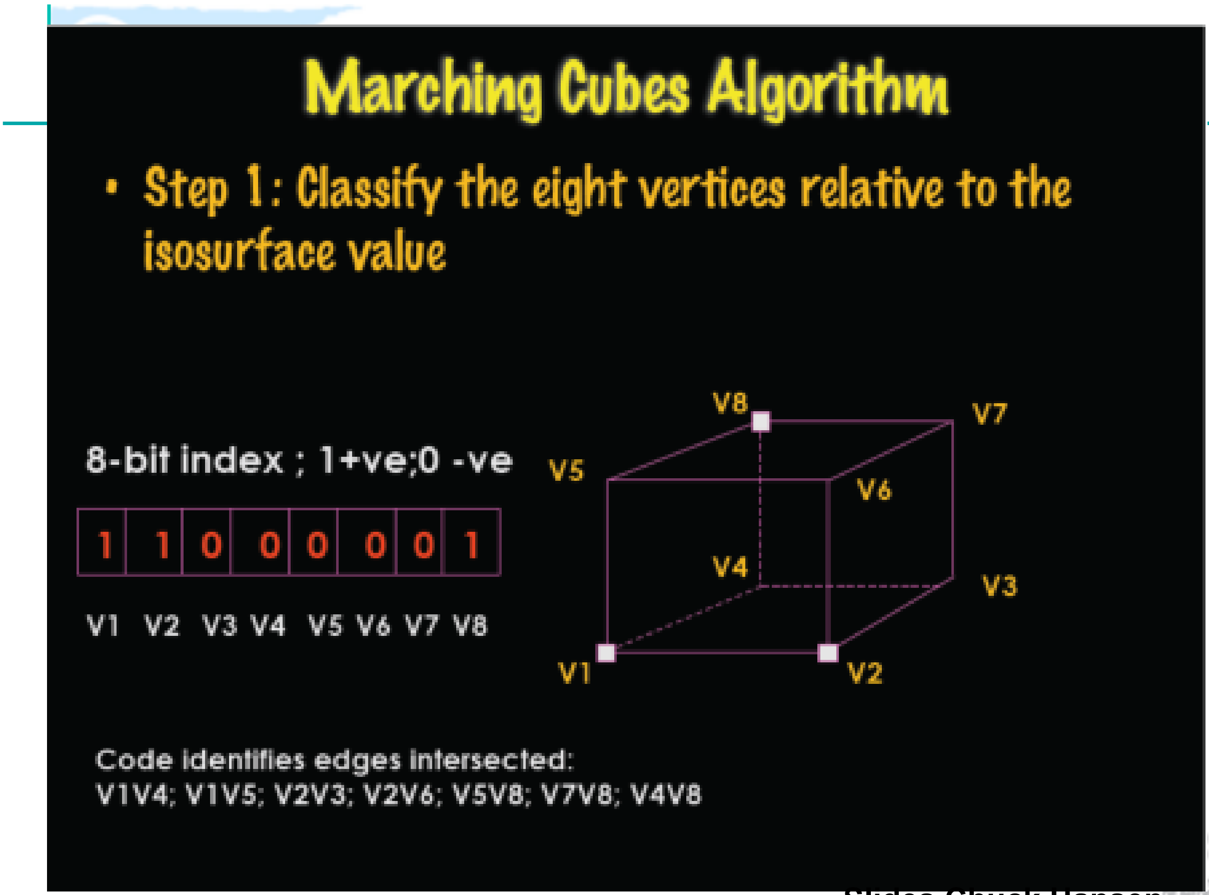
Slides Chuck Hansen



Isosurface Construction

- In some configurations, just one triangle forms the isosurface
- In other configurations ...
  - ...there can be several triangles
  - ...or a polygon with 4, 5 or 6 points which can be triangulated
- A software implementation will have separate code for each configuration

Slides Chuck Hansen



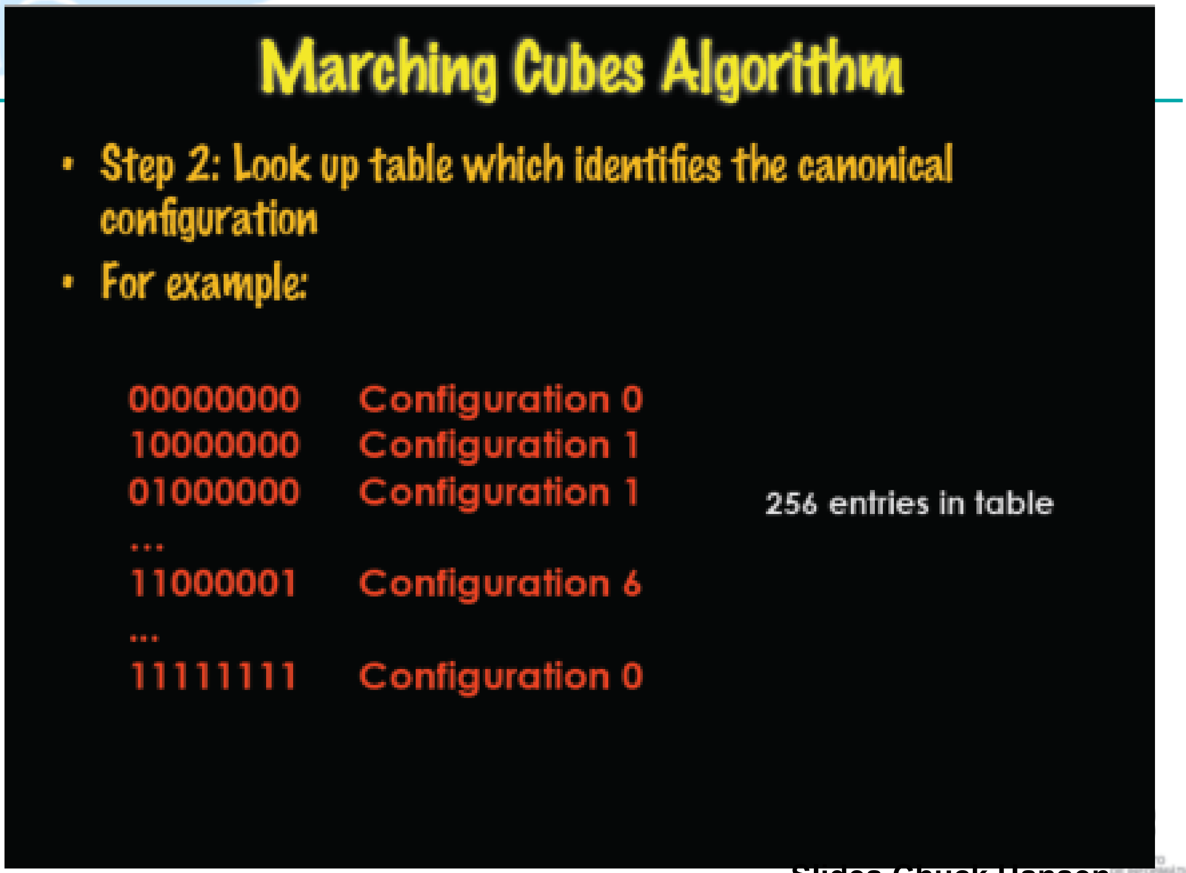
Marching Cubes Algorithm

- Step 1: Classify the eight vertices relative to the isosurface value

8-bit index ; 1+ve;0 -ve

1	1	0	0	0	0	0	1
V1	V2	V3	V4	V5	V6	V7	V8

Code identifies edges intersected:
V1V4; V1V5; V2V3; V2V6; V5V8; V7V8; V4V8

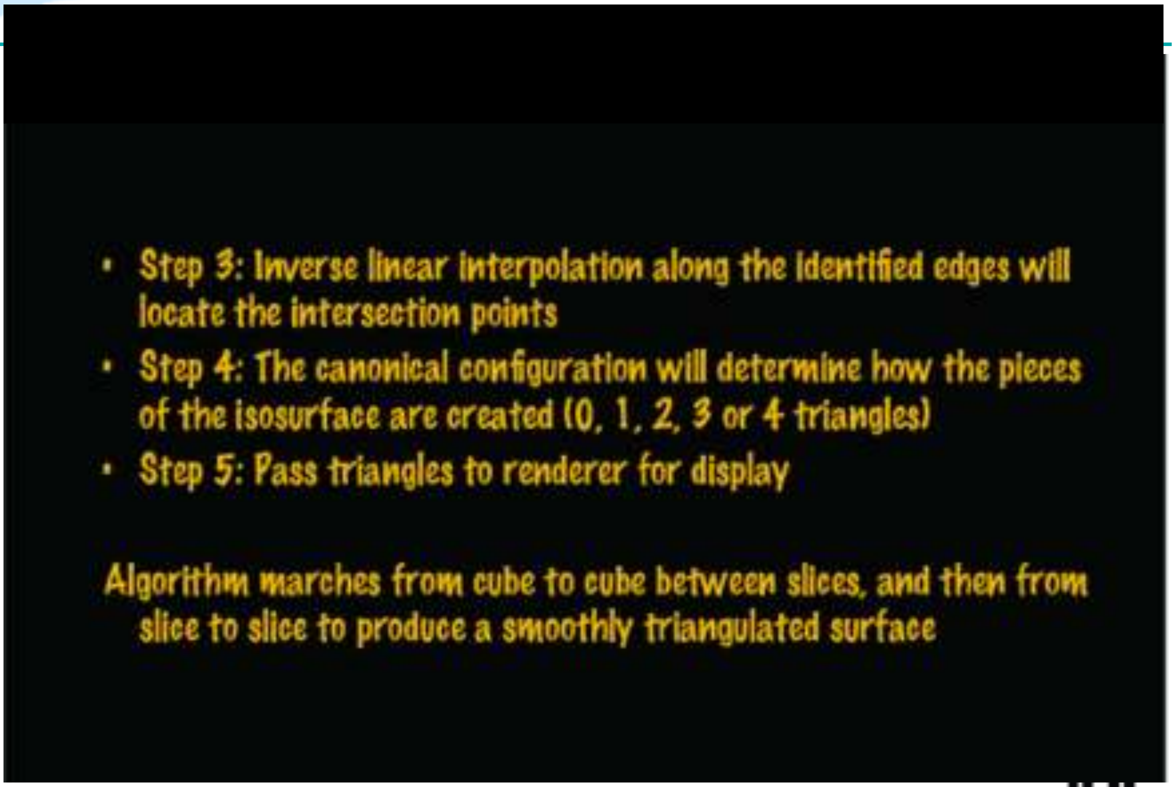


Marching Cubes Algorithm

- Step 2: Look up table which identifies the canonical configuration
- For example:

00000000	Configuration 0
10000000	Configuration 1
01000000	Configuration 1
...	
11000001	Configuration 6
...	
11111111	Configuration 0

256 entries in table

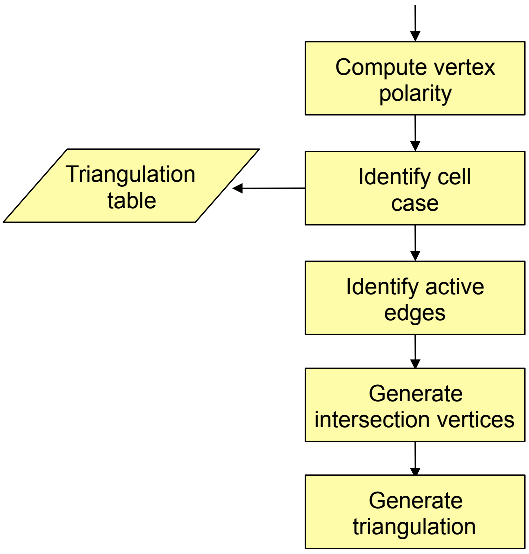


- Step 3: Inverse linear interpolation along the identified edges will locate the intersection points
- Step 4: The canonical configuration will determine how the pieces of the isosurface are created (0, 1, 2, 3 or 4 triangles)
- Step 5: Pass triangles to renderer for display

Algorithm marches from cube to cube between slices, and then from slice to slice to produce a smoothly triangulated surface

Slides Chuck Hansen

Marching Cubes (MC) (Lorensen e Cline, 1987)



Compute vertex polarity

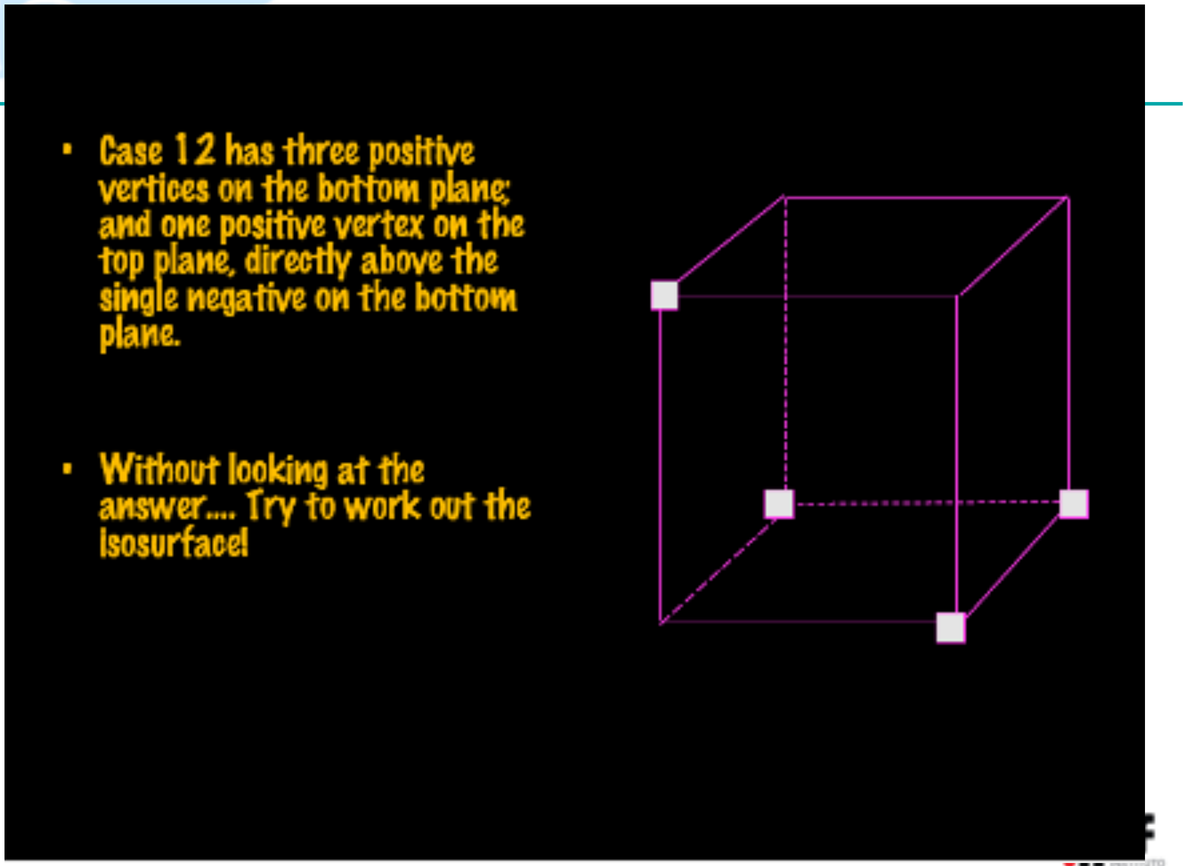
Identify cell case

Triangulation table

Identify active edges

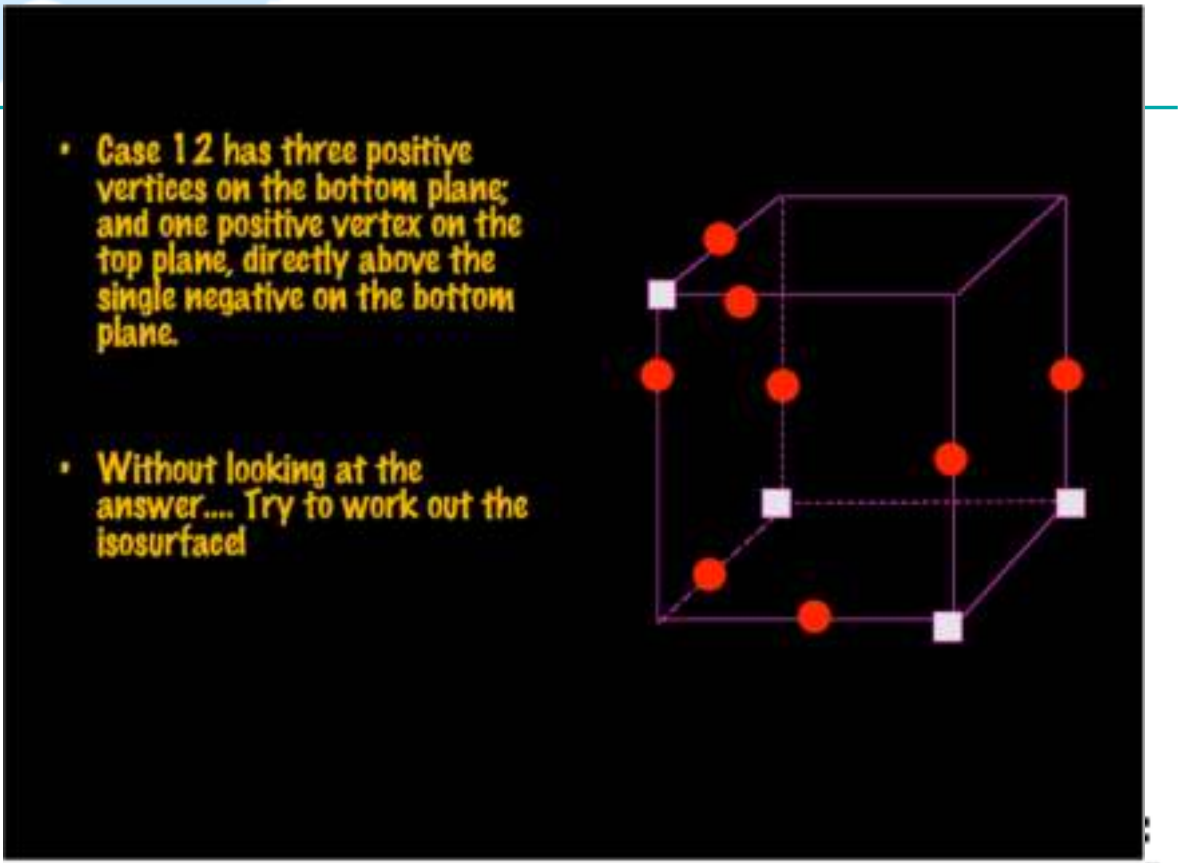
Generate intersection vertices

Generate triangulation

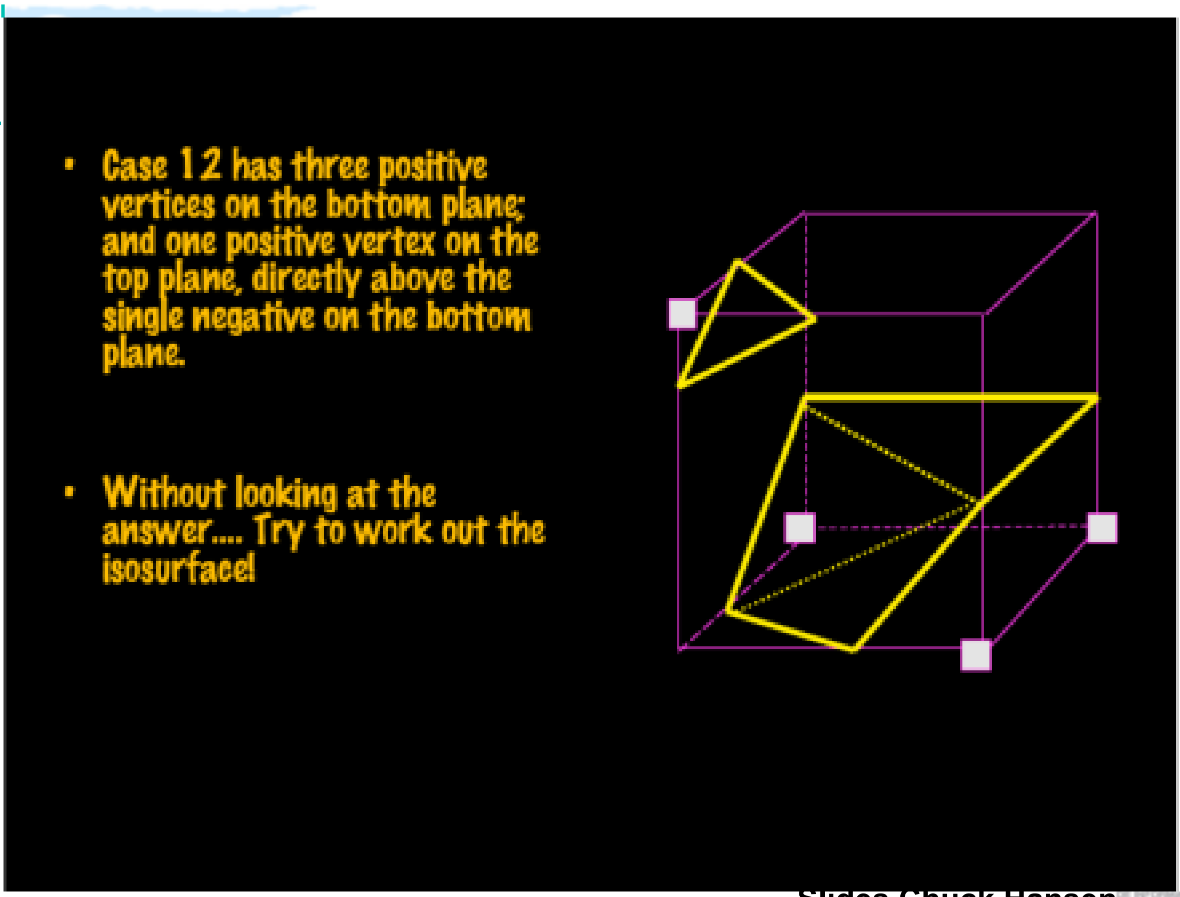


- Case 12 has three positive vertices on the bottom plane; and one positive vertex on the top plane, directly above the single negative on the bottom plane.
- Without looking at the answer.... Try to work out the Isosurface!

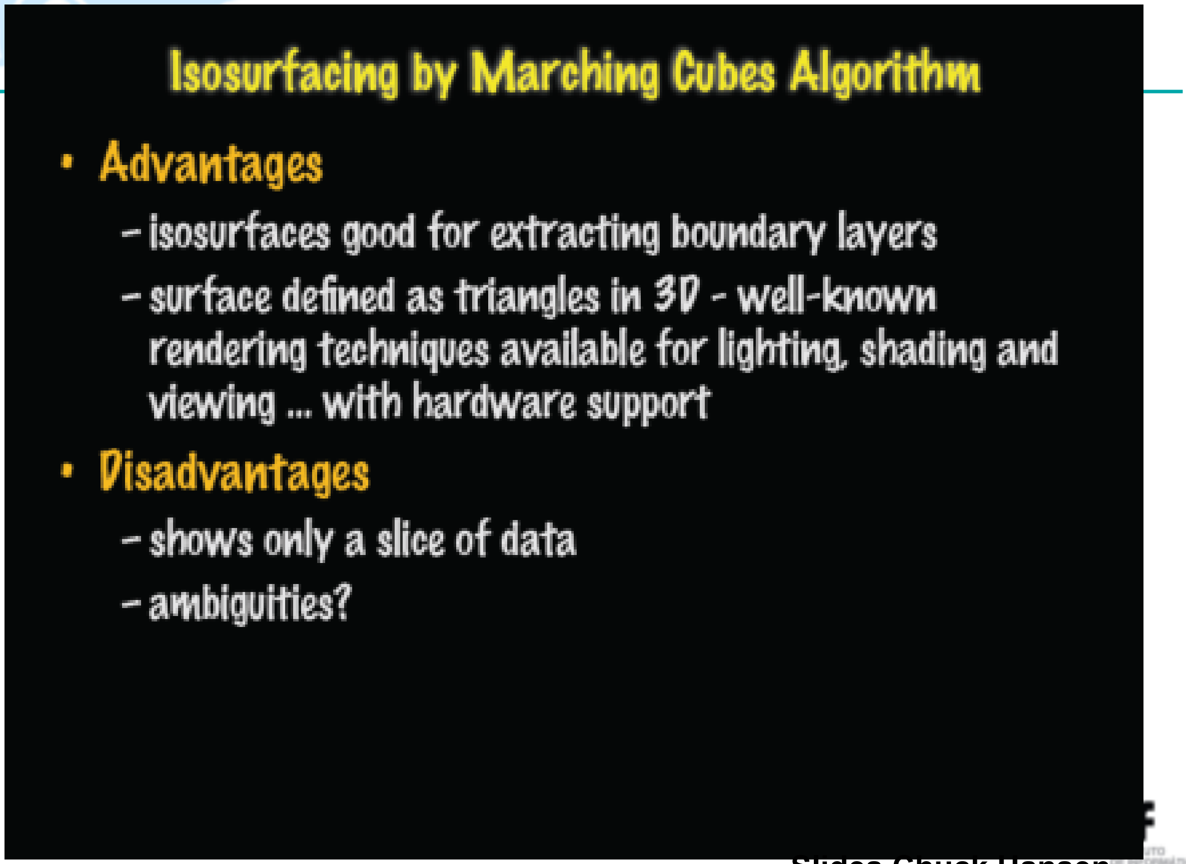
Slides Chuck Hansen



- Case 12 has three positive vertices on the bottom plane; and one positive vertex on the top plane, directly above the single negative on the bottom plane.
- Without looking at the answer.... Try to work out the isosurface!

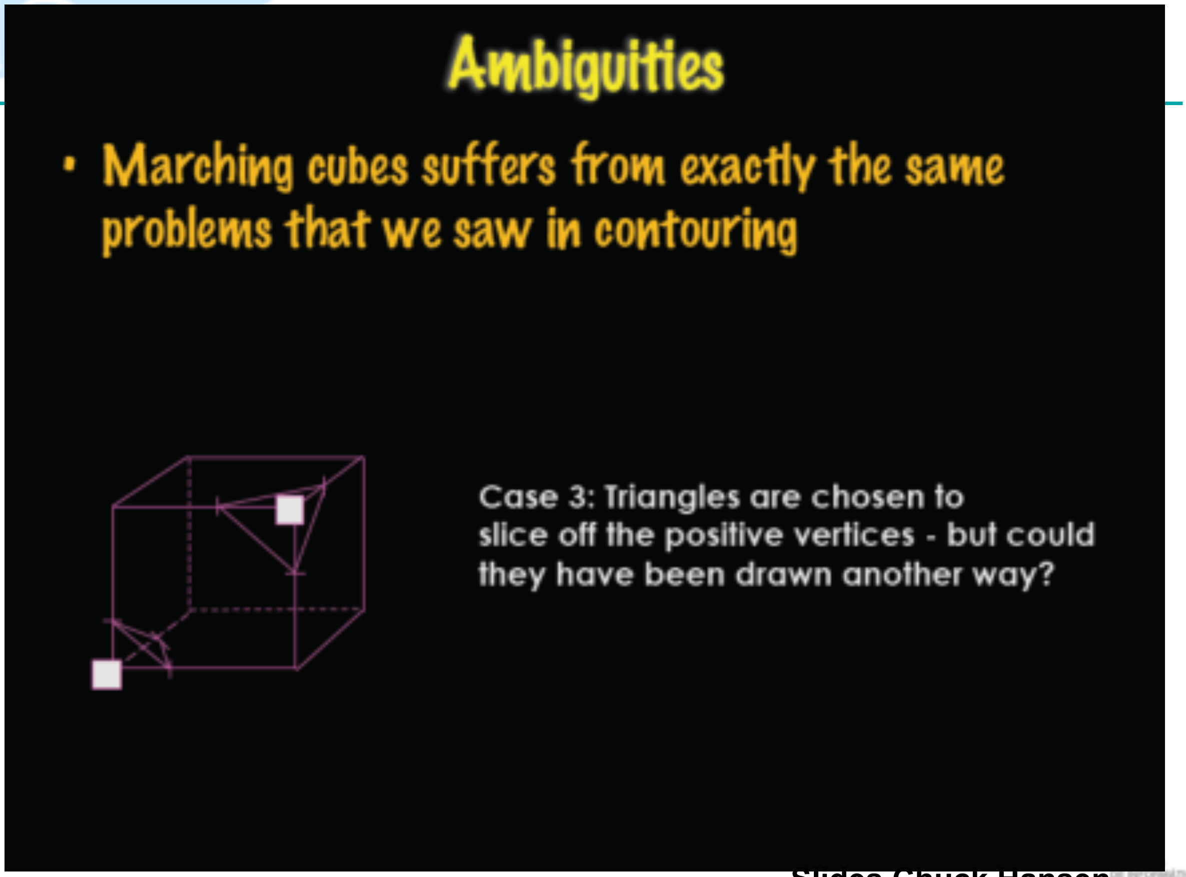


- Case 12 has three positive vertices on the bottom plane; and one positive vertex on the top plane, directly above the single negative on the bottom plane.
- Without looking at the answer.... Try to work out the isosurface!



Isosurfacing by Marching Cubes Algorithm

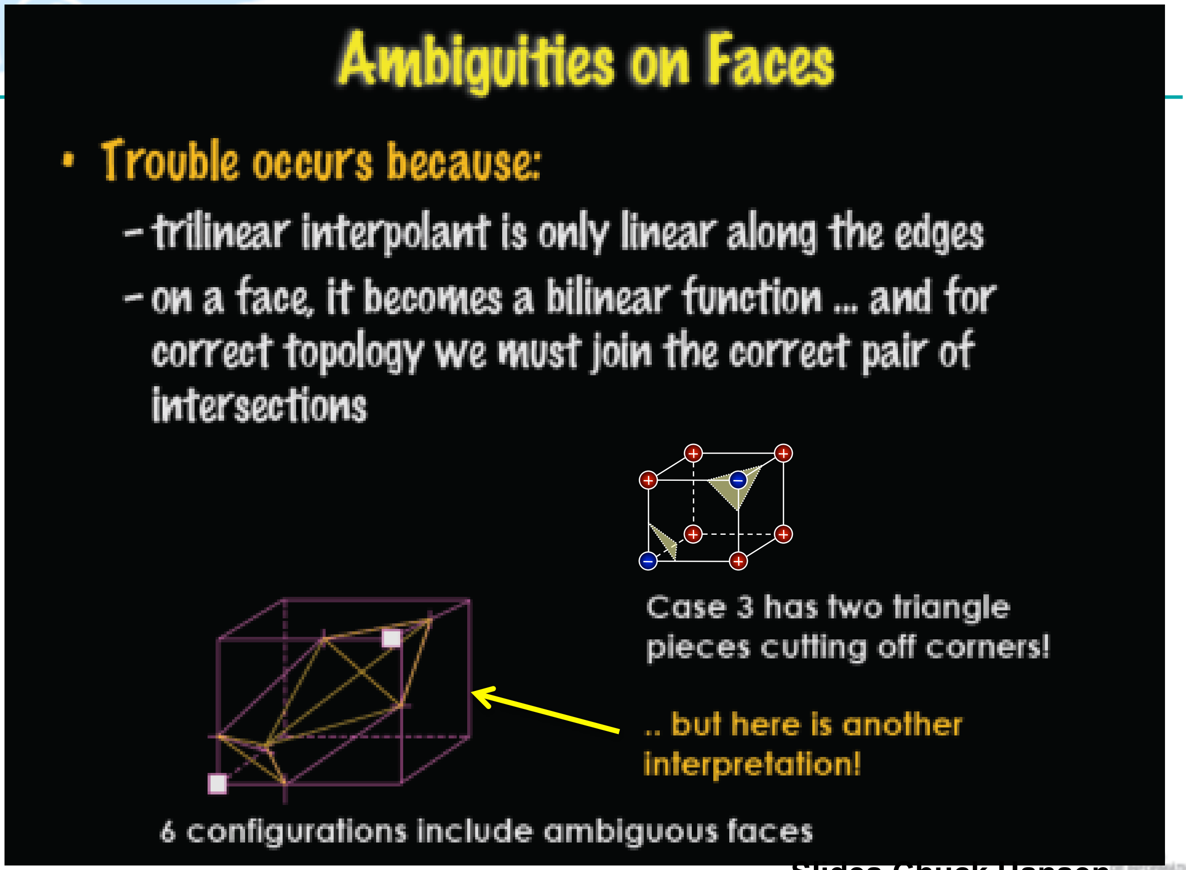
- Advantages
  - isosurfaces good for extracting boundary layers
  - surface defined as triangles in 3D - well-known rendering techniques available for lighting, shading and viewing ... with hardware support
- Disadvantages
  - shows only a slice of data
  - ambiguities?



Ambiguities

- Marching cubes suffers from exactly the same problems that we saw in contouring

Case 3: Triangles are chosen to slice off the positive vertices - but could they have been drawn another way?



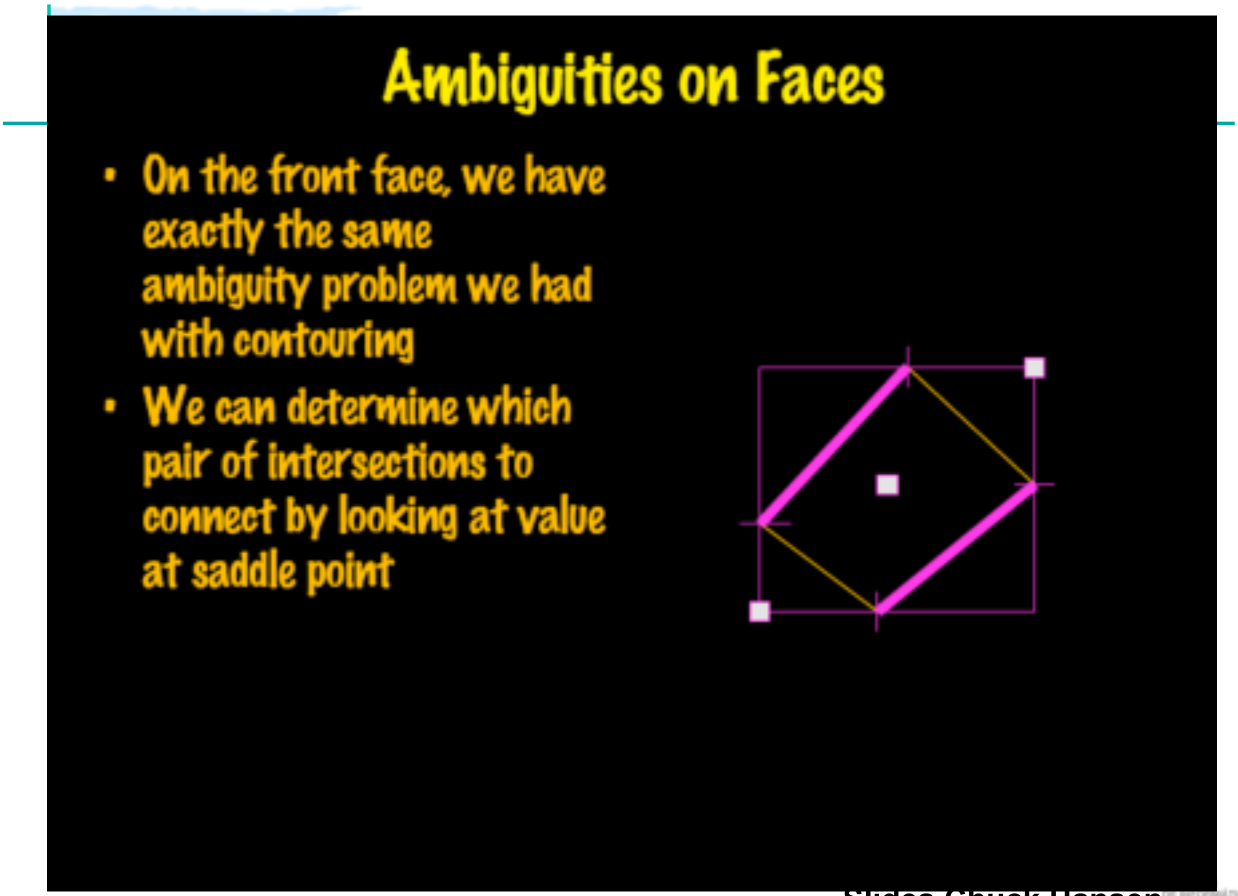
Ambiguities on Faces

- Trouble occurs because:
  - trilinear interpolant is only linear along the edges
  - on a face, it becomes a bilinear function ... and for correct topology we must join the correct pair of intersections

Case 3 has two triangle pieces cutting off corners!

.. but here is another interpretation!

6 configurations include ambiguous faces

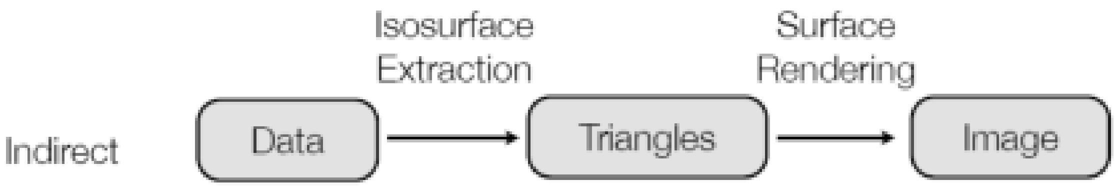


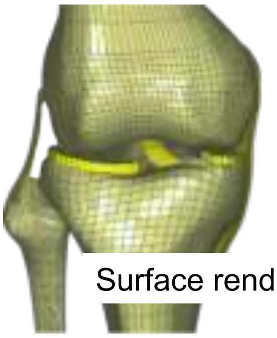
Ambiguities on Faces

- On the front face, we have exactly the same ambiguity problem we had with contouring
- We can determine which pair of intersections to connect by looking at value at saddle point

Slides Chuck Hansen

Volume visualization approaches

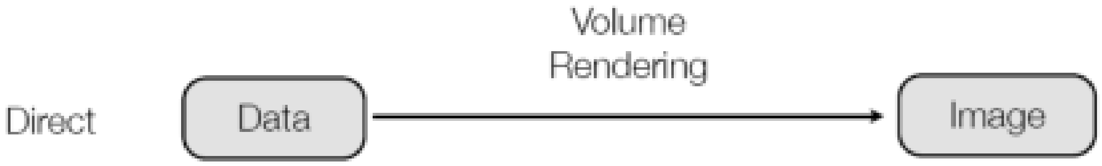




Surface rendering

Volume visualization approaches

Direct rendering



Direct volume rendering

- The data is considered to represent a semi-transparent, light-emitting medium (which absorbs light too)
  - Based on laws of physics
- Volume data is used as a whole
- Color and opacity are used to distinguish materials within the volume

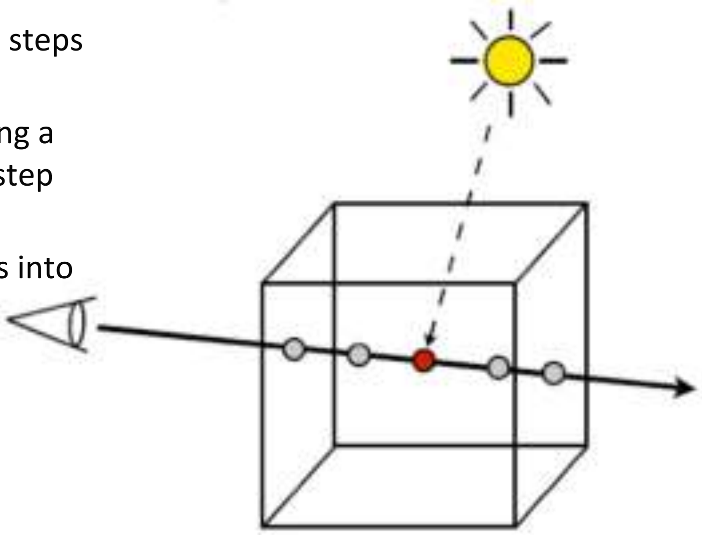
Pinto, 2007

Direct volume rendering

- Three stages of volume rendering
  - **Sampling**: Selecting the steps through the volume
  - **Classification**: Computing a color and opacity for a step
  - **Compositing:** Blending together classified steps into a final image

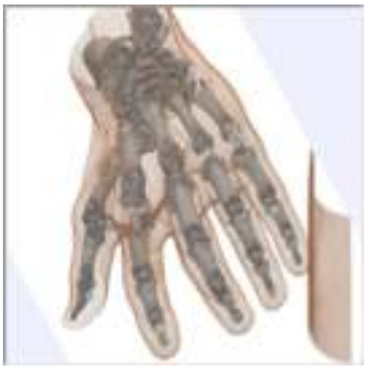
Direct volume rendering

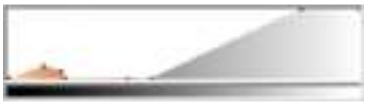
- Three stages of volume rendering
  - **Sampling**: Selecting the steps through the volume
  - **Classification**: Computing a color and opacity for a step
  - **Compositing**: Blending together classified steps into a final image

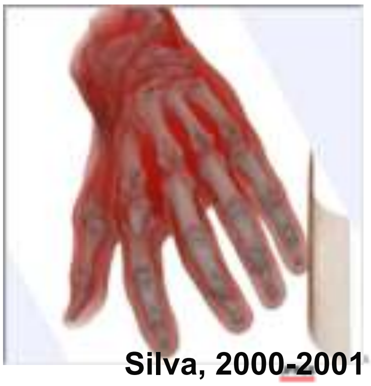


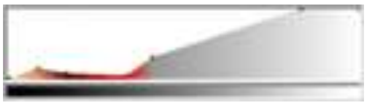
Direct volume rendering

- Three stages of volume rendering
  - **Sampling**: Selecting the steps through the volume
  - **Classification**: Computing a color and opacity for a step
  - **Compositing**: Blending together classified steps into a final image







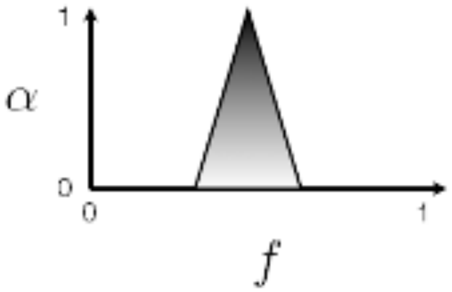


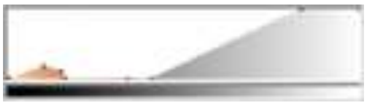
Silva, 2000-2001

Direct volume rendering

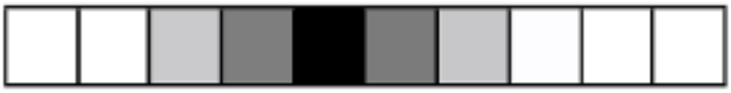
- **Classification**: Use of a transfer function

$$f(x) = \mathbf{R} \rightarrow \mathbf{R}^4, s \rightarrow (r,g,b,\alpha)$$





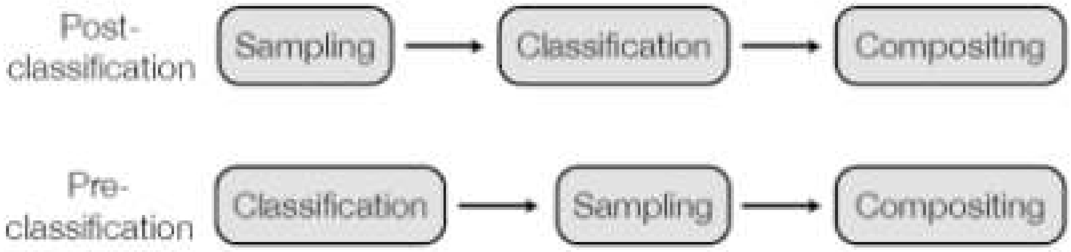
Look up table



Silva, 2000-2001

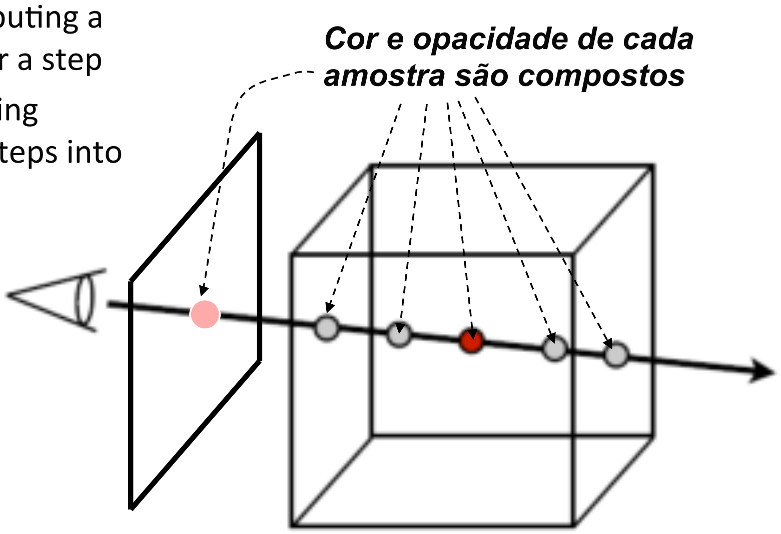
Direct volume rendering

- Three stages of volume rendering
  - Sampling: Selecting the steps through the volume
  - Classification: Computing a color and opacity for a step
  - Compositing: Blending together classified steps into a final image



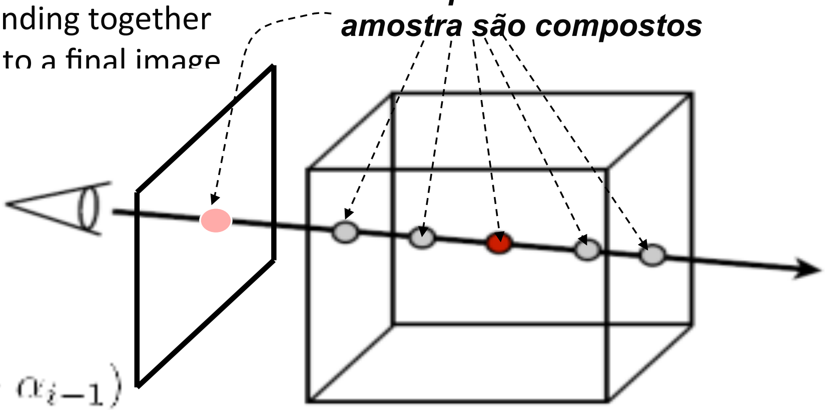
Direct volume rendering

- Three stages of volume rendering
  - **Sampling**: Selecting the steps through the volume
  - **Classification**: Computing a color and opacity for a step
  - **Compositing**: Blending together classified steps into a final image



Direct volume rendering

- Three stages of volume rendering
  - **Sampling**: Selecting the steps through the volume
  - **Classification**: Computing a color and opacity for a step
  - **Compositing**: Blending together classified steps into a final image



Cor e opacidade de cada amostra são compostos

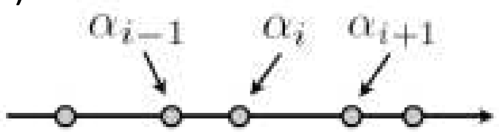
Ray casting: front to back

$$c_i = c_{i-1} + c_i\alpha_i(1 - \alpha_{i-1})$$

$$\alpha_i = \alpha_{i-1} + \alpha_i(1 - \alpha_{i-1})$$

Direct volume rendering

- Image-based (backward projection)
  - Ray-casting (front to back)



$$c_i = c_{i-1} + c_i\alpha_i(1 - \alpha_{i-1})$$

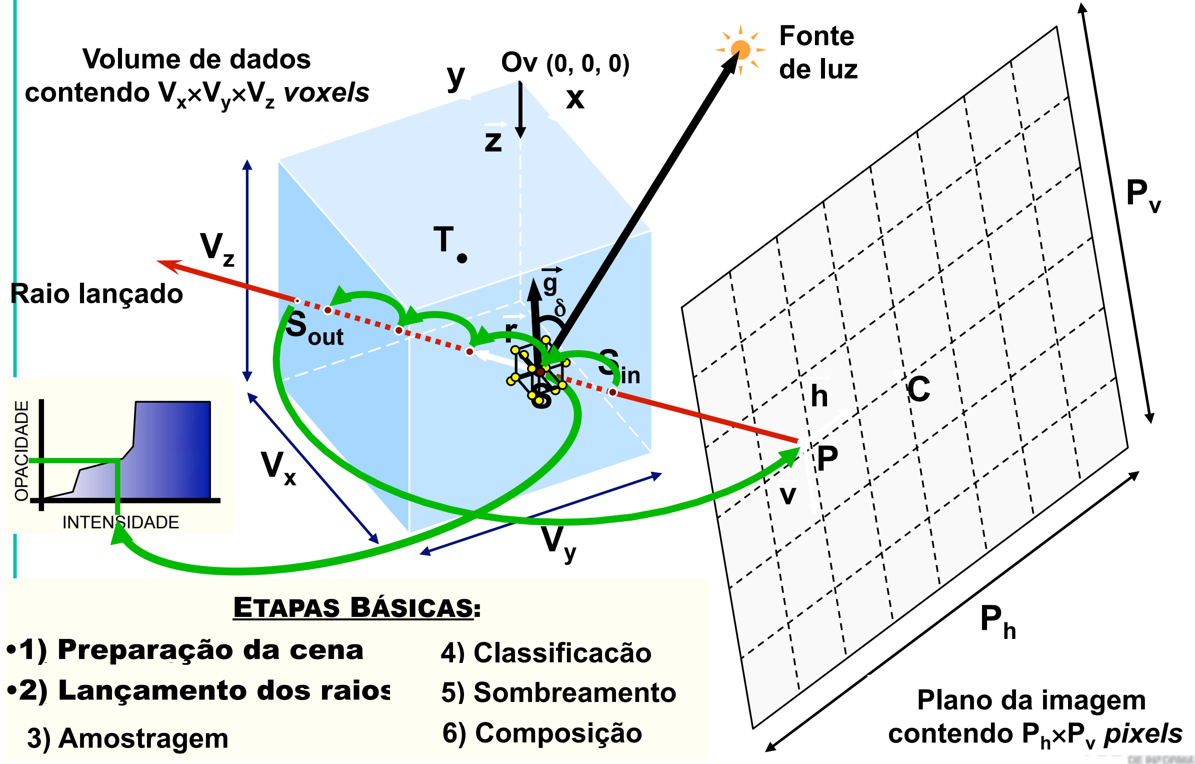
$$\alpha_i = \alpha_{i-1} + \alpha_i(1 - \alpha_{i-1})$$

- Object-based (forward projection, back to front)
  - Splatting
  - Texture mapping

$$c_i = c_i\alpha_i + c_{i+1}(1 - \alpha_i)$$

Ray casting

Silva, 2000-2001

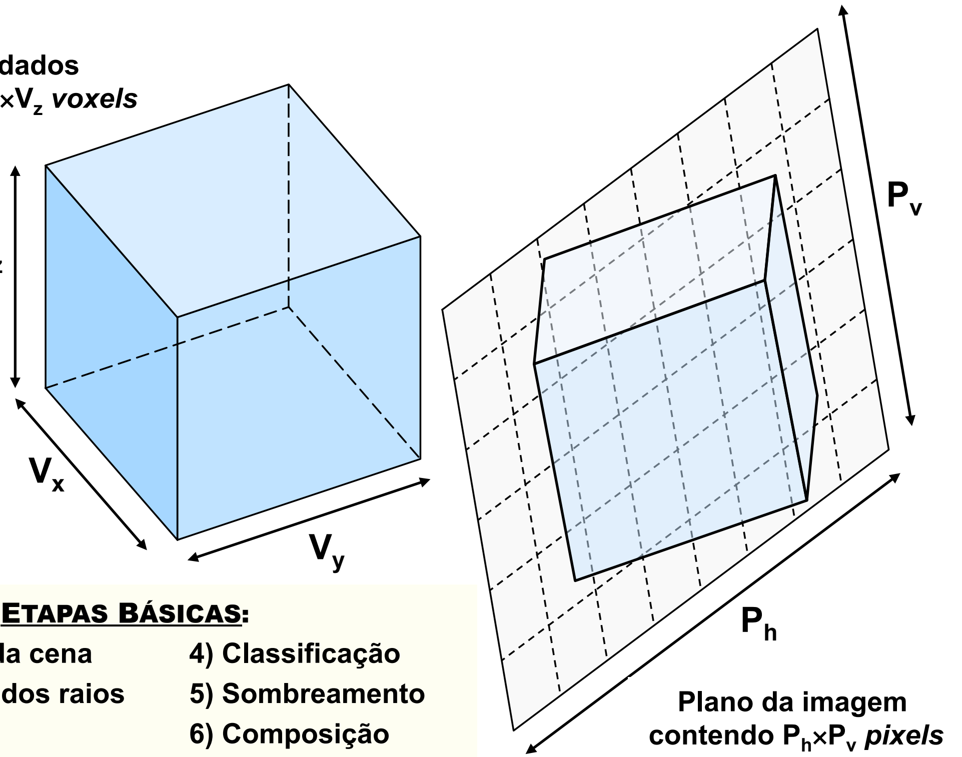


ETAPAS BÁSICAS:

- **•1) Preparação da cena**
- **•2) Lançamento dos raios**
- 3) Amostragem
- 4) Classificacão
- 5) Sombreamento
- 6) Composição

Ray casting

Silva, 2000-2001



Volume de dados
contendo $V_x \times V_y \times V_z$ *voxels*
V_z
V_x
V_y
P_v
P_h
Plano da imagem
contendo $P_h \times P_v$ *pixels*

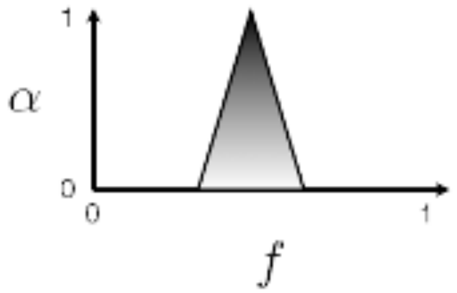
ETAPAS BÁSICAS:

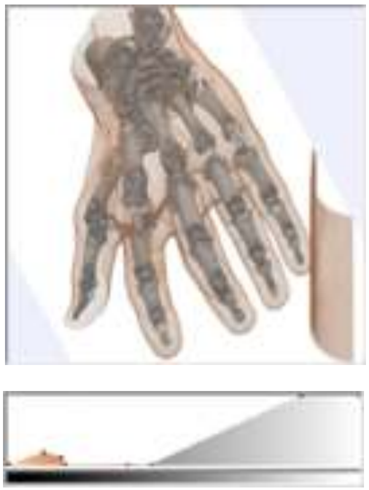
- 1) Preparação da cena
- 2) Lançamento dos raios
- 3) Amostragem
- 4) Classificação
- 5) Sombreamento
- 6) Composição

Transfer functions

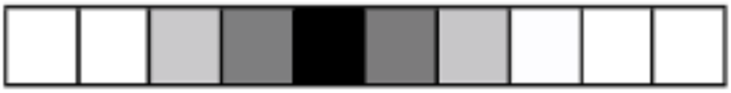
- **Classification: color and opacity**

$$f(x) = \mathbf{R} \rightarrow \mathbf{R}^4, s \rightarrow (r, g, b, \alpha)$$



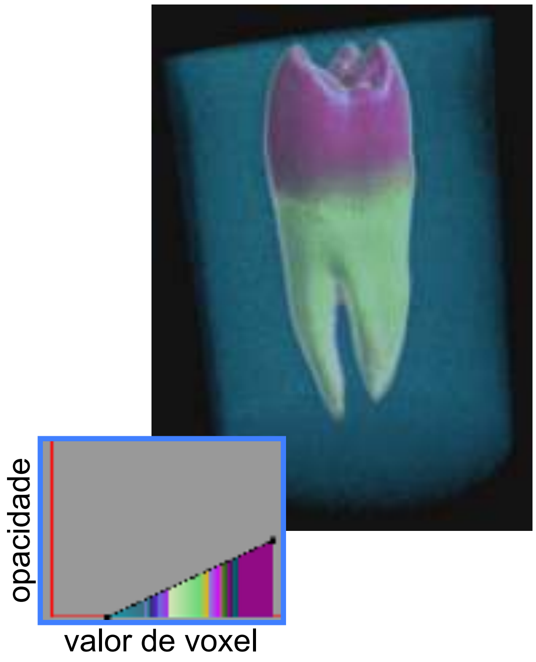


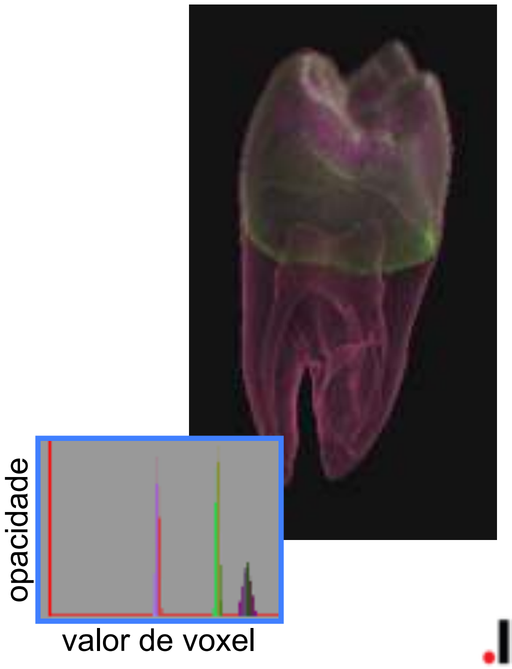
Look up table



Silva, 2000-2001

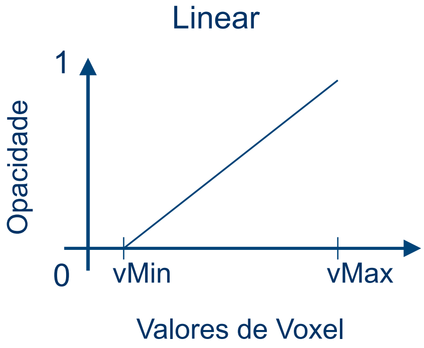
Transfer functions





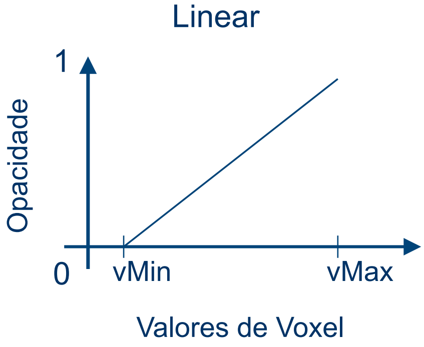
❖ FTs de opacidade 1-D

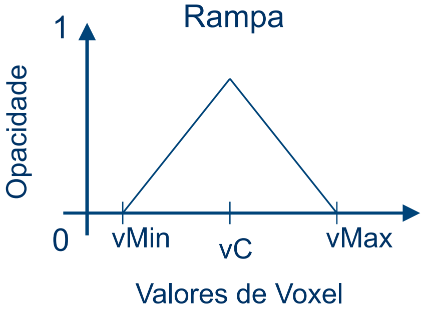
$$\alpha = f(v)$$



❖ FTs de opacidade 1-D

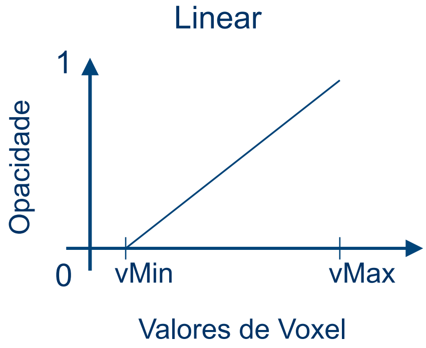
$$\alpha = f(v)$$

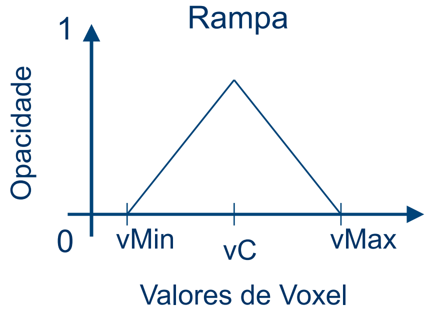


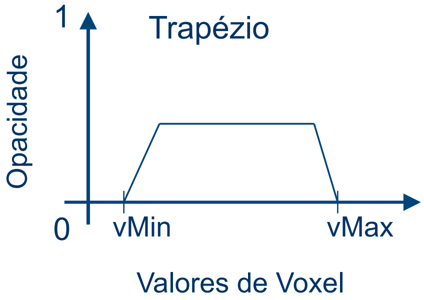


❖ FTs de opacidade 1-D

$$\alpha = f(v)$$

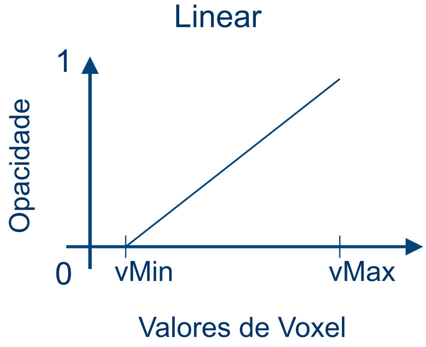


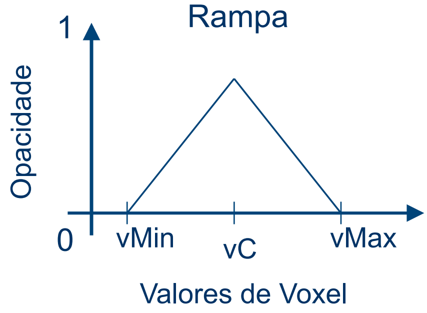


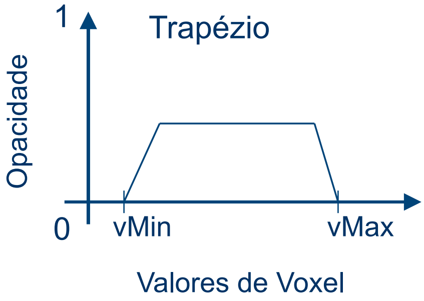


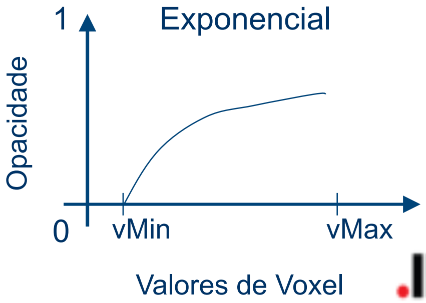
❖ FTs de opacidade 1-D

$$\alpha = f(v)$$

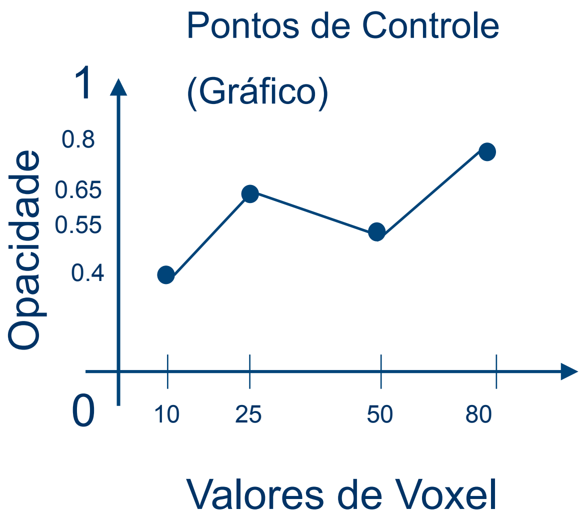




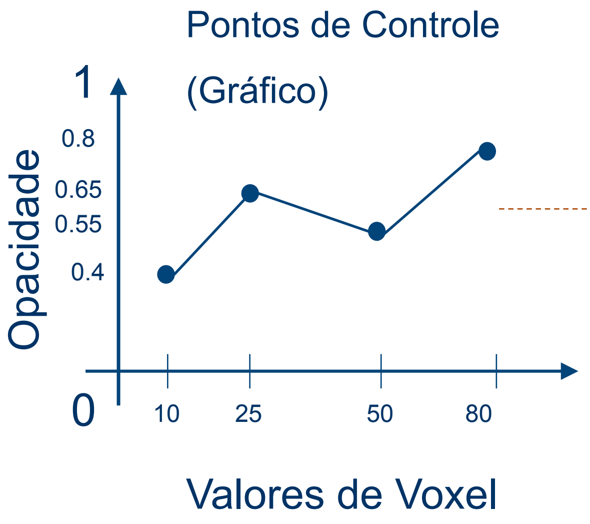




❖ Especificação Manual de FTs



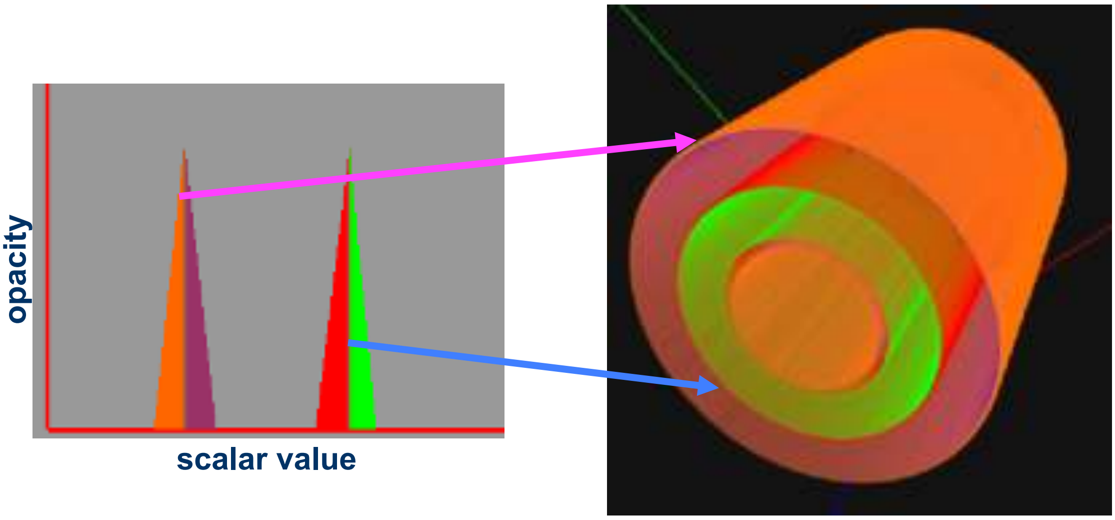
❖ Especificação Manual de FTs



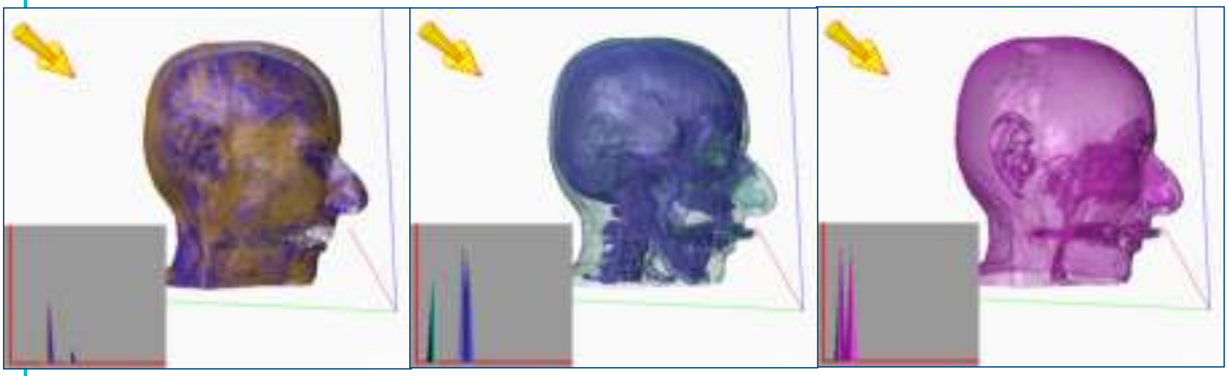
Lookup Table Resultante

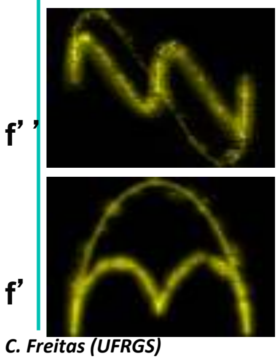
0.4	0.42	0.46	0.48	0.5	0.52	...	0.8
10	11	12	13	14	15	...	80

Enhancing transition zones



Transfer functions

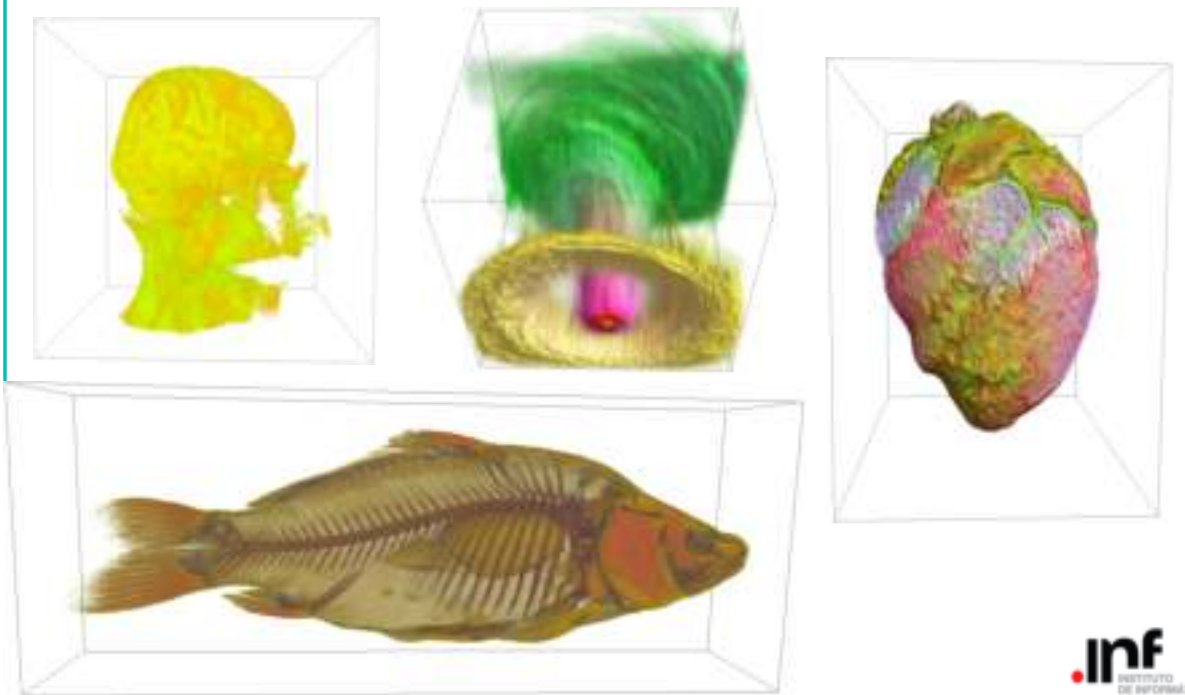




- Valores escalares de voxel são ordenados de acordo com a distância estimada até a borda mais próxima (Kindlmann e Durkin, 1998).

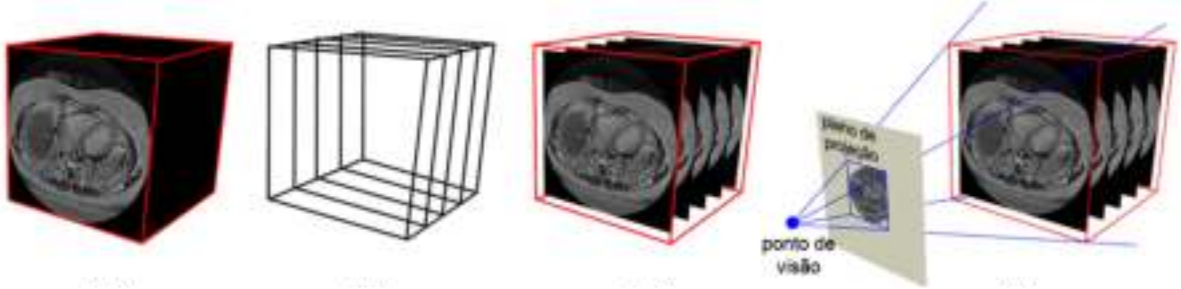
- Sorteia-se, entre os primeiros, um pequeno número de valores de voxel. Estes recebem opacidade elevada e descontinuidade de cor.

Results



Direct volume rendering

- Image-based (backward projection)
  - Ray-casting
- Object-based (forward projection)
  - Texture mapping


plano de projeção
ponto de visão