

## Volume Visualization

Part 3 (out of 3)

---

---

---

---

---

---

## Hardware-Volume Visualization

Faster with Hardware?!

---

---

---

---

---

---

### Two Approaches



- 3D-textures:
  - ◆ Volume data stored in 3D-texture
  - ◆ Proxy geometry (slices) parallel to image plane, are interpolated tri-linearly
  - ◆ Back-to-front compositing
- 2D-textures:
  - ◆ 3 stacks of slices (x-, y- & z-axis), slices are interpolated bi-linearly
  - ◆ Select stack (most "parallel" to image plane)
  - ◆ Back-to-front compositing



---

---

---

---

---

---

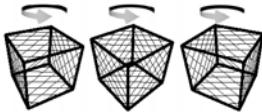
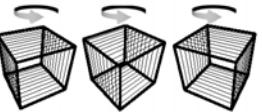
---

---

## Variation of View Point

**TU VIENNA**

- 3D-textures:
  - ◆ Number of slices arbitrary
- 2D-textures:
  - ◆ Stack change: discontinuity

Viewport-Aligned Slices      Object-Aligned Slices

Eduard Gröller, Helwig Hauser      4

---



---



---



---



---



---



---



---



---

## Special Hardware

**TU VIENNA**

- Hardware volume raycasting
  - ◆ In vertex and fragment operations of modern graphics cards
- Special Hardware
  - ◆ VolumePro board:
    - Special card for PC
    - Calculates shear-warp factorization, incl. compositing
    - Warp-step with "regular" graphics card (OpenGL)

Eduard Gröller, Helwig Hauser      5

---



---



---



---



---



---



---



---



---

## Marching Cubes (MC)

Iso-Surface-Display

**TU VIENNA**

---



---



---



---



---



---



---



---



---

## Repetition: Volume vs. Surface Rendering

**Volume rendering:**

- Direct volume visualization
- Usage of transfer functions
- Pros: look on the interior, semi-transparency

**Surface rendering:**

- Indirect volume visualization
- Intermediate representation: Iso-surface, "3D"
- Pros: shading → shape!, hardware rendering

Eduard Gröller, Helwig Hauser 7

## Concepts and Terms

**Example 1:**

- CT measurement
- Iso-stack-conversion
- Iso-surface-calculation (marching cubes)
- Surface rendering (OpenGL)

Eduard Gröller, Helwig Hauser

## Iso-Surfaces

**Intermediate representation**

**Aspects:**

- Preconditions:
  - expressive Iso-value, Iso-value separates materials
  - Interest: in transitions
- Very selective (binary selection / omission)
- Uses traditional hardware
- shading ⇒ 3D-impression!

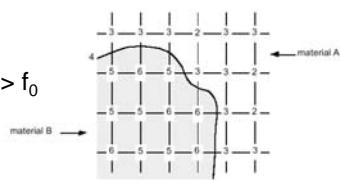
Eduard Gröller, Helwig Hauser 9

## Volume Data $\Leftrightarrow$ Iso-Surfaces



### Iso-Surface:

- ◆ Iso-value  $f_0$
- ◆ separates values  $> f_0$  from values  $\leq f_0$
- ◆ Often not known  $\rightarrow$
- ◆ Can only be approximated from samples!
- ◆ Shape / position dependent on type of reconstruction



Eduard Gröller, Helwig Hauser

10

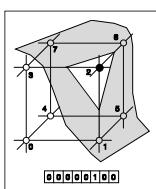


## Approximation of Iso-Surface



### Approach:

- ◆ Iso-Surface intersects data volume = set of all cells



Eduard Gröller, Helwig Hauser



## Marching Cubes

- ✓ Cell consists of 4(8) pixel (voxel) values:  
(i+[01], j+[01], k+[01])
- 1. Consider a Cell
- 2. Classify each vertex as inside or outside
- 3. Build an index
- 4. Get edge list from table[index]
- 5. Interpolate the edge location
- 6. Go to next cell



November 19, 2000

1



## MC 1: Create a Cube

✓ Consider a Cube defined by eight data values:

A 3D cube diagram with vertices labeled as follows: top-front-left (i,j,k), top-back-left (i,j,k+1), top-front-right (i+1,j,k), top-back-right (i+1,j,k+1), bottom-front-left (i,j+1,k), bottom-back-left (i,j+1,k+1), bottom-front-right (i+1,j+1,k), and bottom-back-right (i+1,j+1,k+1).

November 19, 2008 2

---

---

---

---

---

---

## MC 2: Classify Each Voxel

✓ Classify each voxel according to whether it lies outside the surface (value > iso-surface value) or inside the surface (value  $\leq$  iso-surface value)

A 3D cube diagram with dimensions 5, 8, and 10. Two smaller cubes are shown inside: one with an iso-value of 9 and another with an iso-value of 7. A legend indicates that green dots represent "inside" and blue dots represent "outside".

November 19, 2000 3

---

---

---

---

---

---

## MC 3: Build An Index

✓ Use the binary labeling of each voxel to create an index

A 3D cube diagram with vertices labeled v1 through v8. To the right, there is a table titled "Index:" with columns for v1 through v8. The table shows binary values: v1=00110000, v2=11110100, v3=00110000, v4=11110100, v5=00110000, v6=11110100, v7=00110000, and v8=11110100. A legend indicates that green dots represent "inside" and blue dots represent "outside".

November 19, 2000 4

---

---

---

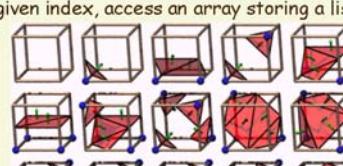
---

---

---

MC 4: Lookup Edge List

- For a given index, access an array storing a list of edges



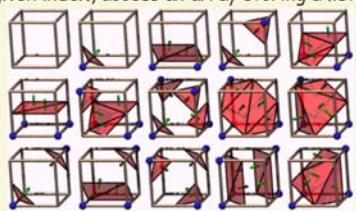
The 15 Cube Combinations

- ✓ all 256 cases can be derived from 15 base cases

November 19, 2000

- ## MC 4: Lookup Edge List

✓ For a given index, access an array storing a list of edges 



### The 15 Cube Combinations

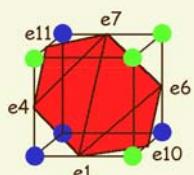
✓ all 256

- All 200 cases can be derived from 10 base cases

## MC 5: Example

## MC 5: Example

- ✓ Index = 10110001
  - ✓ triangle 1 = e4,e7,e11
  - ✓ triangle 2 = e1, e7, e4
  - ✓ triangle 3 = e1, e6, e7
  - ✓ triangle 4 = e1, e10, e6



November 19, 2000

MC 6: Interp. Triangle Vertex

- For each triangle edge, find the vertex location along the edge using linear interpolation of the voxel values

$$x = i + \left( \frac{T - v[i]}{v[i+1] - v[i]} \right)$$

$v[i] = 10$

$v[i+1] = 0$

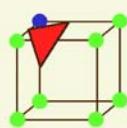
$T = 5$

$i = 8$

$x = 8 + \left( \frac{5 - 10}{10 - 8} \right) = 8 + \left( \frac{-5}{2} \right) = 8 - 2.5 = 5.5$

- ✓ For each triangle edge, find the vertex location

- ✓ For each triangle edge, find the vertex location along the edge using linear interpolation of the voxel values



丁辰

## MC 7: Compute Normals

- ✓ Calculate the normal at each cube vertex

$G_x = V_{x-1,y,z} - V_{x+1,y,z}$

$G_y = V_{x,y-1,z} - V_{x,y+1,z}$

$G_z = V_{x,y,z-1} - V_{x,y,z+1}$

$\vec{N} = \frac{\vec{G}_x \times \vec{G}_y}{|\vec{G}_x \times \vec{G}_y|}$

- ✓ Use linear interpolation to compute the polygon vertex normal

November 19, 2000

---

---

---

---

---

---

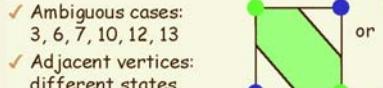
---

---

---

## MC 8: Ambiguous Cases

- ✓ Ambiguous cases:  
3, 6, 7, 10, 12, 13
- ✓ Adjacent vertices:  
different states
- ✓ Diagonal vertices:  
same state
- ✓ Resolution:  
decide for one case



---

---

---

---

---

---

Danger: Holes!

■ Wrong vs. correct classification!

Figure 4: Two internal configurations for the Marching Cubes configuration 5

---

---

---

---

---

---

## MC 9: Asymptotic Decider

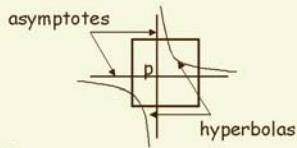
- ✓ Assume bilinear interpolation within a face
- ✓ hence iso-surface is a hyperbola
- ✓ compute the point p where the asymptotes meet
- ✓ sign of  $S(p)$  decides the connectedness

The diagram shows a square domain representing a face. Two intersecting curves, labeled "hyperbolas", are drawn within the square. The intersection point of these two curves is marked with a dot and labeled "p". Arrows point from the label "asymptotes" to the two curves, indicating that they are asymptotes of the hyperbolas.

November 19, 2000

10

- ## MC 9: Asymptotic Decider



November 19, 2000

10

- ✓ 256 Cases
- ✓ reduce to 15 cases by symmetry
- ✓ Complementary cases - (swap in- and outside)
- ✓ Ambiguity resides in cases 3, 6, 7, 10, 12, 13
- ✓ Causes holes if arbitrary choices are made.

(a) Volume data      (b) Isosurface  
 $S = f(x, y, z)$

(c) Polygonal Approximation

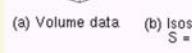
November 19, 2000

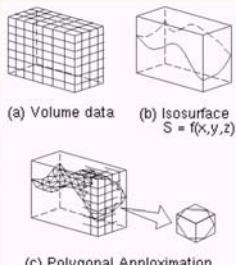
- ## Marching Cubes - Summary

  - ✓ 256 Cases
  - ✓ reduce to 15 cases by symmetry
  - ✓ Complementary cases - (swap in- and outside)
  - ✓ Ambiguity resides in cases 3, 6, 7, 10, 12, 13
  - ✓ Causes holes if arbitrary choices are made.


(a) Volume data


(b) Isosurface  
 $S = f(x,y,z)$





November 19, 2000

## Marching Cubes - Summary 2

- ✓ Up to 4 triangles per cube
- ✓ Dataset of  $512^3$  voxels can result in several million triangles (many Mbytes!!!)
- ✓ Iso-surface does not represent an object!!!
- ✓ No depth information
- ✓ Semi-transparent representation --> sorting
- ✓ Optimization:
  - Reuse intermediate results
  - Prevent vertex replication
  - Mesh simplification



November 19, 2000

12

- ✓ Up to 4 triangles per cube
  - ✓ Dataset of  $512^3$  voxels can result in several million triangles (many Mbytes!!!)
  - ✓ Iso-surface does not represent an object!!!!
  - ✓ No depth information
  - ✓ Semi-transparent representation --> sorting
  - ✓ Optimization:
    - Reuse intermediate results
    - Prevent vertex replication
    - Mesh simplification



November 19, 2000

12

**MC Examples**

**1 Iso-surface**

**3 Iso-surfaces**

**2 Iso-surfaces**

November 19, 2000



Even Further Examples

TU VIENNA

11-Jul-1990 - 14:00Z  
Surface Wind Index & Winds

18-Jul-1990 - 17:00Z  
Surface Total Precipitable Clouds & Precipitation

Precipitation Rate (mm/h)

Elevation (m)

Wind Speed (m/s)

Wind Direction (°)

27



## Literature



### ■ Paper (more details):

- ◆ W. Lorensen & H. Cline: "Marching Cubes: A High Resolution 3D Surface Construction Algorithm" in *Proceedings of ACM SIGGRAPH '87 = Computer Graphics*, Vol. 21, No. 24, July 1987

Eduard Gröller, Helwig Hauser

28



---

---

---

---

---

---

---

## Conclusion Volume Visualization

### General Remarks



---

---

---

---

---

---

---

## Surface vs. Volume Rendering



### ■ Surface Rendering:

- ◆ Indirect representation / display
- ◆ Conveys surface impression
- ◆ Hardware supported rendering (fast?!)
- ◆ Iso-value-definition



30

### ■ Volume Rendering:

- ◆ direct representation / display
- ◆ Conveys volume impression
- ◆ Often realized in software (slow?!)
- ◆ Transfer functions



---

---

---

---

---

---

---

## Conclusion VolVis



- Introduction                          ← data, simple methods  
    ← DVR vs. surf. fitting
- Direct volume visualization
  - ◆ Ray casting                          ← types of combinations
  - ◆ Splatting                              ← object-order vs. image-order
  - ◆ Shear-warp factorization            ← speed vs. quality
  - ◆ Hardware-based VolVis
- Indirect VolVis                        ← iso-value selection
  - ◆ Marching cubes (iso-surface-visualization)
- Conclusion

Eduard Gröller, Helwig Hauser

31



---

---

---

---

---

---

---

## Acknowledgments



- For material for this lecture unit:
  - ◆ Michael Meißner
  - ◆ Roger Crawfis (Ohio State Univ.)
  - ◆ Hanspeter Pfister (MERL)
  - ◆ Torsten Möller
  - ◆ Dirk Bartz
  - ◆ Markus Hadwiger

Eduard Gröller, Helwig Hauser

32



---

---

---

---

---

---

---