Marching Cubes Without Skinny And Tiny Triangles
And How To Employ Them To Create Smooth Molecular Surfaces

Thomas Sander, Sep 2016
- What are Marching Cubes?
- Why do we care about them?
- Why do they carry small and skinny triangles?
- How to make them relinquish these triangles?
The Marching Cubes Algorithm
*Iso-Layer Triangulation from Voxel Data*

- Paper published in the 1987 SIGGRAPH
  DOI: 10.1145/37402.37422

- One of the most frequently cited papers in computer graphics field

- Patent expired in 2005

- Applied for CT, MRI, Molecular Modelling
3D-Voxel Data
3D-Voxel Data

Iso-surface at 5.0
3D-Voxel Data

Iso-surface at 5.0
3D-Voxel Data

Iso-surface at 5.0
Individual Cube’s Triangles Connect to Surface
256 Cubes with Triangles Derived From 15 Base (+8) Cube Families
Triangles Can be Classified by 8 Edge-Groups
### The Beauty: Pure Speed Through Lookup Tables

A **cube specific 8-bit key** is created from 8 corners (in or out)

```java
public static final int[] EDGE_TABLE = {
    0x000, 0x109, 0x203, 0x30a, 0x406, 0x50f, 0x605, 0x70c, 0x80c, 0x905, 0xa0f, 0xb06, 0xc0a, 0xd03, 0xe09, 0xf00, 0x190, 0x099, 0x393, 0x29a, 0x596, 0x49f, 0x795, 0x69c, 0x99c, 0x895, 0xb9f, 0xa96, 0xd9a, 0xc93, 0xf99, 0xe90, ...
};
```

```java
public static final int[][] FACE_TABLE = {
    {},
    {0, 8, 3},
    {0, 1, 9},
    {1, 8, 3, 9, 8, 1},
    {1, 2, 10},
    {0, 8, 3, 1, 2, 10},
    {9, 2, 10, 0, 2, 9},
    {2, 8, 3, 2, 10, 8, 10, 9, 8},
    {3, 11, 2}, ...
};
```
The Triangulation Process

For all cubes do

- build key from 8 corners (00000111b = 7)
- lookup affected edges: 0x70c
- calculate vertexes on affected edges
- lookup faces: \{2, 8, 3, 2, 10, 8, 10, 9, 8\}
- create triangles connecting vertexes
The Triangulation Process

For all cubes do

– build key from 8 corners (00000111b = 7)
– lookup affected edges: 0x70c
– calculate vertexes on affected edges
– lookup faces: {2, 8, 3, 2, 10, 8, 10, 9, 8}
– create triangles connecting vertexes
For all cubes do

– build key from 8 corners (00000111b = 7)

– lookup affected edges: 0x70c

– calculate vertexes on affected edges

– lookup faces: {2, 8, 3, 2, 10, 8, 10, 9, 8}

– create triangles connecting vertexes
The Triangulation Process

For all cubes do

– build key from 8 corners (00000111b = 7)
– lookup affected edges: 0x70c
– calculate vertexes on affected edges
– lookup faces: {2, 8, 3, 2, 10, 8, 10, 9, 8}
– create triangles connecting vertexes
How Does this help to create a Connolly surface? Create Voxel Grid

For every atom
set close grid points to VDW-radius - distance

probe \( r = 1.4 \)
How Does this help to create a Connolly surface?
Create Voxel Grid

For every atom
set close grid points to VDW-radius - distance
How Does this help to create a Connolly surface?

Create Voxel Grid

For every grid edge with separating positive from negative volume
determine edge position $p_1$ with value=0
for every grid point $p_2$ within probe radius
set value to $\min(\text{value}, \text{distance} - \text{probeRadius})$
How Does this help to create a Connolly surface?

Create Voxel Grid

For every grid edge with separating positive from negative volume
determine edge position $p_1$ with value=0
for every grid point $p_2$ within probe radius
set value to $\min(value, distance - probeRadius)$
How Does this help to create a Connolly surface?
Create Voxel Grid

For every grid edge with separating positive from negative volume
determine edge position \( p_1 \) with value=0
for every grid point \( p_2 \) within probe radius
set value to \( \min(\text{value}, \text{distance} - \text{probeRadius}) \)
How Does this help to create a Connolly surface?
Create Voxel Grid

For every grid edge with separating positive from negative volume
determine edge position p1 with value=0
for every grid point p2 within probe radius
set value to min(value, distance - probeRadius)

inspired by Tom Goddard,
“Molecular Surface Algorithm”,
Oct 14, 2013,
www.cgl.ucsf.edu/chimera/data/surface-oct2013/surface.html
Problem: Small and skinny triangles
Avoiding Small-Area-Triangles

New simple procedure:

For every corner:

if more than 1 close edge cuts
  - join these edge cut positions
  - remove all triangles with merged corners
Avoiding Small-Area-Triangles

New simple procedure:

For every corner:
  if more than 1 close edge cuts
    • join these edge cut positions
    • remove all triangles with merged corners
Avoiding Small-Area-Triangles

New simple procedure:

For every corner:
  if more than 1 close edge cuts
    • join these edge cut positions
    • remove all triangles with merged corners
Avoiding Small-Area-Triangles

New simple procedure:

For every corner:

if more than 1 close edge cuts
  • join these edge cut positions
  • remove all triangles with merged corners
Avoiding Small-Area-Triangles

New simple procedure:

For every corner:
   if more than 1 close edge cuts
      • join these edge cut positions
      • remove all triangles with merged corners
Second Source for Skinny Triangles: Edge Group 2
Second Source for Skinny Triangles: Edge Group 2
Second Source for Skinny Triangles: Edge Group 2
Second Source for Skinny Triangles: Edge Group 2
How To Avoid Edge Group 2

How To Avoid Edge Group 2

Most cubes can be triangulated in a different way to avoid edge group 2

How To Avoid Edge Group 2

Most cubes can be triangulated in a different way to avoid edge group 2.

Where triangulation without edge group 2 is impossible, we add a new vertex.

Original Marching Cubes: 3864 triangles, 1934 vertexes
Smooth Marching Cubes: 2956 triangles, 1480 vertexes
Original Marching Cubes
Smooth Marching Cubes With 0.1 Angstrom Resolution
Noisy MRI Dataset: Lobster from volvis.org
324*301*56 bytes, iso-surface at 40.5
Gives 360038 triangles, 179174 vertexes
Triangle Area Distribution: MC vs. SMC

Dataset = MC_molecule

Dataset = SMC_molecule

Dataset = MC_crab

Dataset = SMC_crab
Corner Angle Distribution: MC vs. SMC

Dataset = MC_molecule

Dataset = SMC_molecule

Dataset = MC_crab

Dataset = SMC_crab
New method improving the Marching Cubes algorithm
It entirely avoids small area triangles and, therefore, allows high quality rendering
There is a very small performance penalty
3D Environment independent Java source code
Robust method applicable for molecular surfaces and noisy data (CT, MRI, etc)
Thank You