Hochschule
Ravensburg-Weingarten
Elektrotechnik und Informatik

Eine konstruktive Methode zum Rendern von Marching Cubes Voxeln mit unterschiedlichen Füllmengen und Materialien

Wissenschaftliches Projekt von

Matthias Mettenleiter

Betreut durch
Prof. Dr. Daniel Scherzer

Gliederung

- Motivation
- Grundlagen
- Zielsetzung
- Umsetzung der Oberflächengenerierung
- Umsetzung der Texturierung
- Fazit
- Ausblick

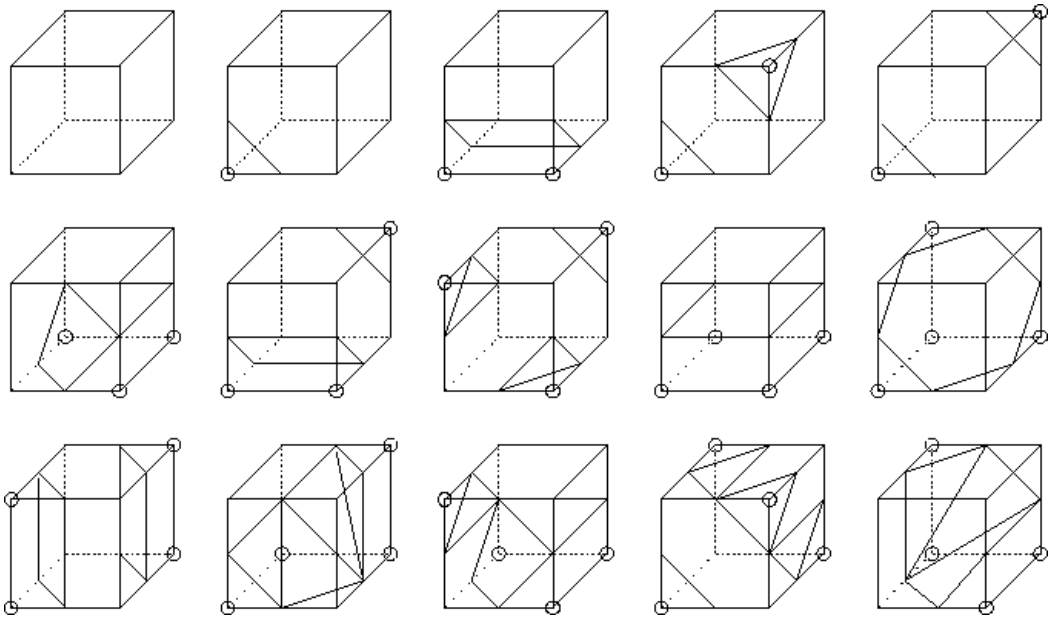
Motivation

Existierende Marching Cubes Größen-Anpassungen in medizinischen Bereichen anhand gemessener Eintrittspunkte in Objekte.

Konzept: Größenanpassung mit konstruierten Werten

Grundlagen (Marching Cubes)

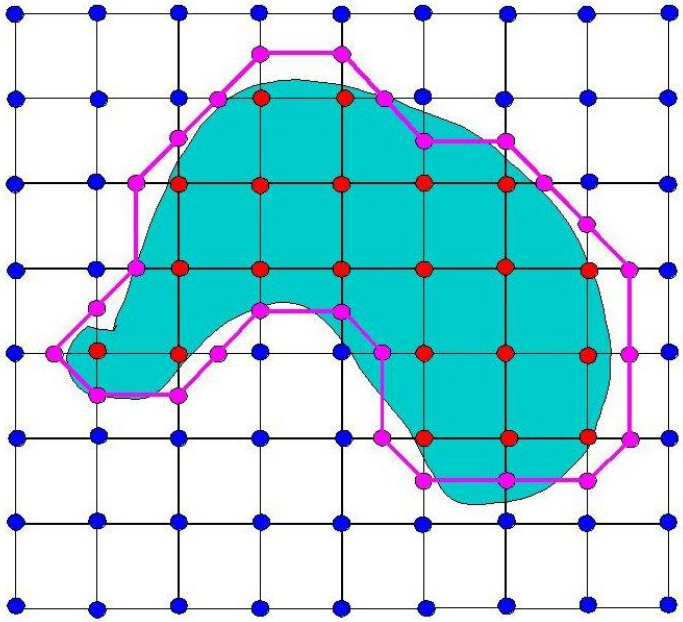
Verschiedene Faces anhand einer Lookup Table zwischen 8 kubisch angeordneten Voxeln generieren:



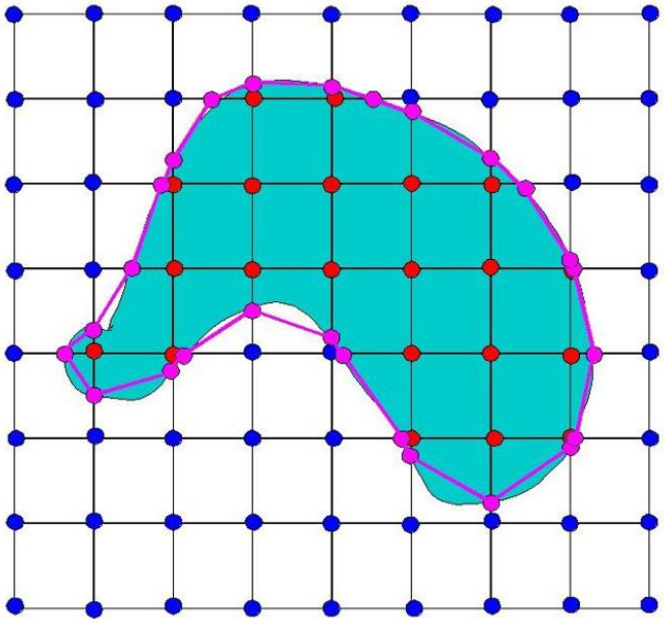
[1]

Grundlagen (Marching Cubes)

Im rekonstruktiven Bereich Verschiebung der Vertices zu den gemessenen Eintrittspunkten in das Objekt:



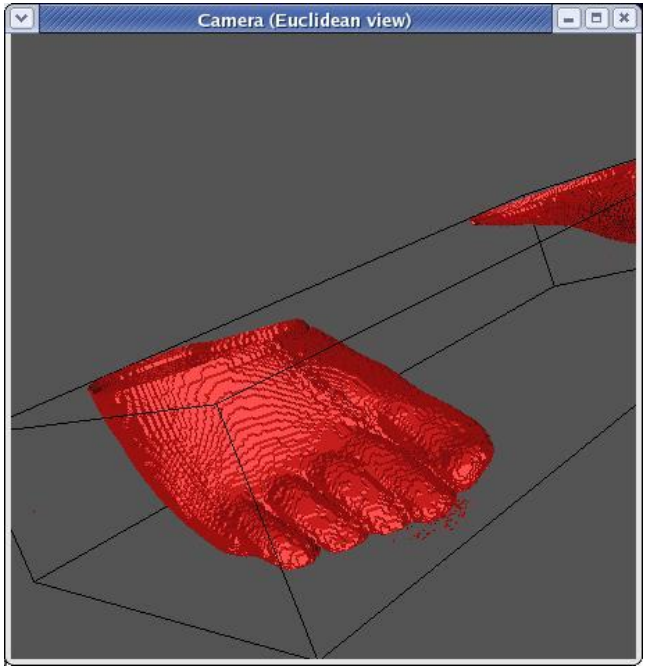
[2]



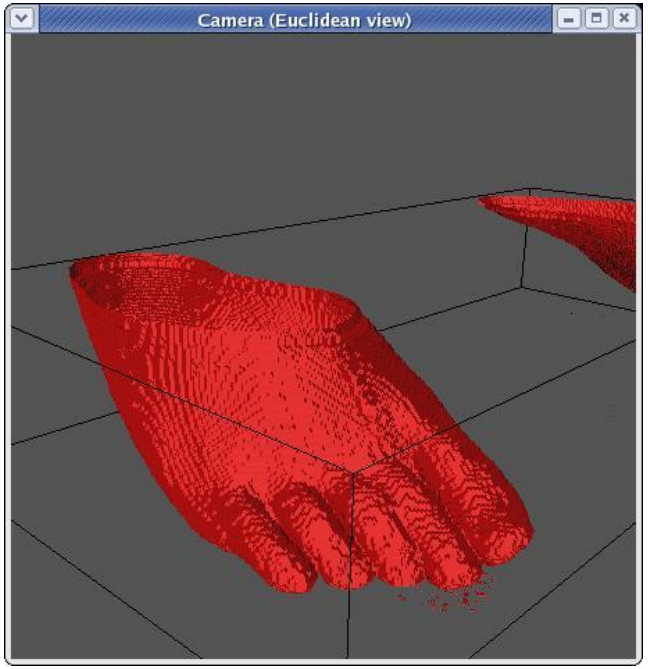
[3]

Grundlagen (Marching Cubes)

Beispiel aus dem Medizinischen Bereich:



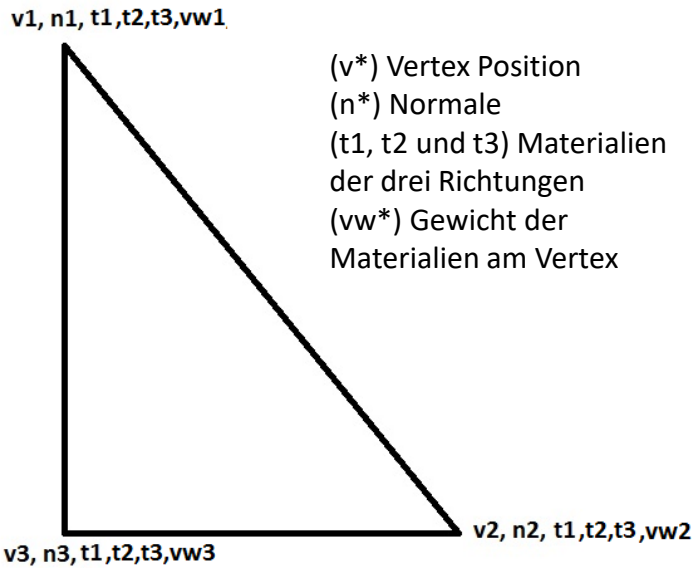
[4]



[5]

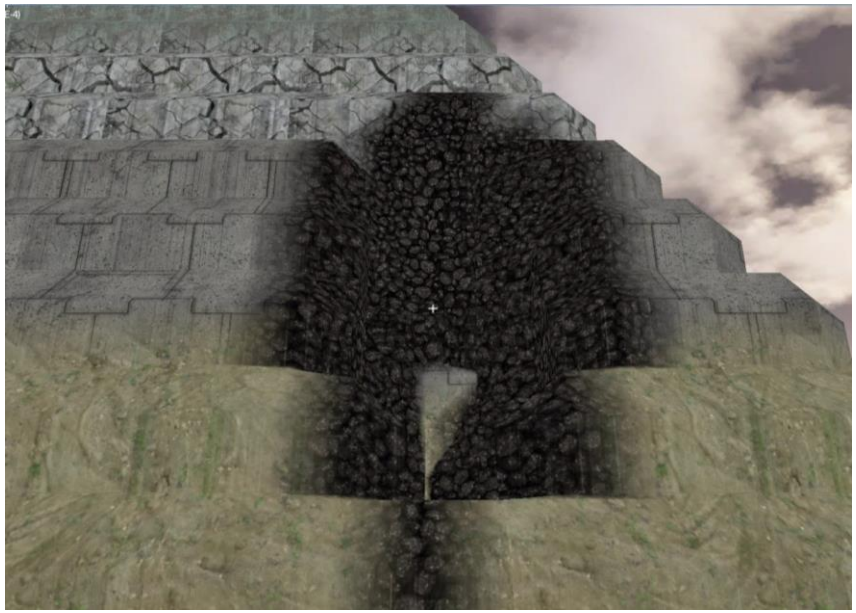
Grundlagen (Texturierung)

Eine Möglichkeit zur Interpolation von Texturen/Farben in Marching Cubes:



(v*) Vertex Position
(n*) Normale
(t1, t2 und t3) Materialien der drei Richtungen
(vw*) Gewicht der Materialien am Vertex

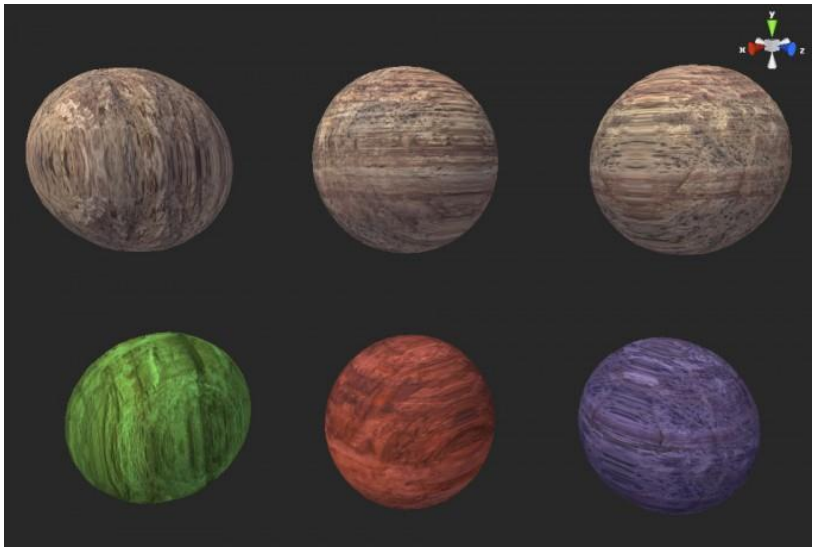
[6]



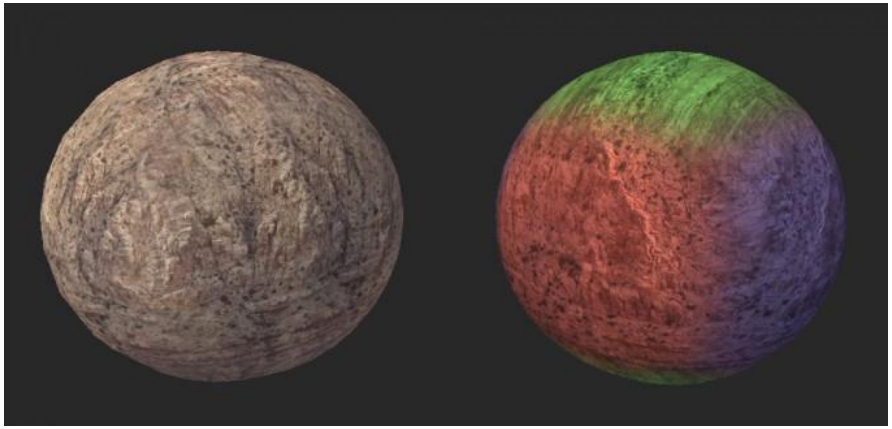
[7]

Grundlagen (Texturierung)

Triplanares Mapping zur Darstellung von Texturen:



[8]



[9]

Zielsetzung

- Jedem Voxel werden ein Material und eine kontinuierliche Füllmenge $\in [1; 0]$ zugewiesen.
- Es wird pro Achter-Tupel ein Mesh erzeugt. Hierbei haben die einzelnen Voxel nur Zugriff auf die ihnen zugewiesenen Informationen und keine Referenzen oder Zugriffe auf andere Voxel, wie beispielsweise ihre sechs direkten Nachbarn.
- Die Gröβe der Meshes wird sich anhand der Füllmenge innerhalb eines Voxels verändern und nachvollziehbar sein. Nachvollziehbar bedeutet in diesem Zusammenhang, dass bei mehr Füllmenge ein größerer Mesh entsteht und umgekehrt. Es ist in diesem Fall keine Anforderung, dass die Ausdehnung volumetrisch korrekt ist.
- Es ist möglich ein Objekt zu erzeugen, das nur aus waagerechten und senkrechten Flächen besteht. Daraus geht hervor, dass es möglich ist 90° Außenwinkel aus senkrechten und waagerechten Flächen zu erzeugen.

Zielsetzung

- Da durch die Skalierung der Marching Cubes unterschiedlich große Flächen in den Meshes entstehen, wird Solid Texturing verwendet, um unterschiedliche Detailgrade zu vermeiden.
- Es werden fließende Übergänge zwischen den Texturen der verschiedenen Materialien gerendert. Dabei werden ebenfalls nur die Daten der acht Voxel verwendet, die zum Finden der Marching Cubes Variante in der Lookup-Table benötigt werden.
- Die Füllmenge eines Voxels beeinflusst auch, wie das Material des Voxels im Verhältnis zu den Materialien der Nachbarn in den Texturübergängen gemischt wird.

Umsetzung der Oberflächengenerierung

Schwellwertformel:

$$s = \frac{1}{l^2}$$

s =Schwellwert, $l = Anzahl\ der\ Sechs\ direkten\ Nachbarn, die\ nicht\ gefüllt\ sind$

Grundausdehnung:

$$a_g = \frac{f_g - s_g}{1 - s_g} * \text{scale}$$

$a_g = Grundausdehnung$, $s_g = Füllmenge\ des\ Gefüllten\ Voxels$, $s_g = Schwellwert\ des\ gefüllten\ Voxels$,

scale = z. B. : $\frac{\sqrt[3]{\frac{3}{\sqrt{2}}}}{\sqrt{2}}$ (Hälfte der Diagonale im Octaeder mit Volumen 1)

Gesamtausdehnung:

$$a = a_g + (1 - a_g) * \frac{f_n}{s_n}$$

a = Gesamtausdehnung, $a_g = Grundausdehnung$,
$s_n = Füllmenge\ des\ nicht\ Gefüllten\ Voxels$, $s_n = Schwellwert\ des\ nicht\ gefüllten\ Voxels$

Umsetzung der Texturierung

Abstand zum Voxel im Achtertupel:

$$a_i = dif(x_i, \mathrm{x}) * \mathrm{dif}(\mathrm{yi}, \mathrm{y}) * \mathrm{dif}(\mathrm{zi}, \mathrm{z})$$

a_i =Abstand vom i-ten Voxel, $(x, y, z) = Position\ innnerhalb\ des\ Tupels$, $(x_i, y_i, z_i) = Position\ des\ i-ten\ Voxels\ innnerhalb\ des\ Tupels$, $\mathrm{dif}(a, b) = Differenz\ zwischen\ a\ und\ b$

Materialmischung zwischen den 8 Voxeln im Tupel (von 0-7 nummeriert):

$$m = \frac{\sum_{i=0}^{7} m_i * a_i * f_i}{\sum_{i=0}^{7} a_i * f_i}$$

$m = Materialmischung$, $m_i = Materialeigenschaft\ des\ i-ten\ Voxels$, $a_i = Abstand\ vom\ i-ten\ Voxel$, $f_i = Füllmenge\ des\ i-ten\ Voxels$

Fazit

- Die entwickelte Methode ermöglicht unterschiedlich große und komplexe Oberflächen über den Marching Cubes Algorithmus aus konstruierten Voxeldaten zu rendern.
- Optimierungen und Änderungen wären denkbar

Ausblick

- Die Lookup-Table kann angepasst werden, um die Trennung der gefüllten von den nicht gefüllten Voxeln über andere Oberflächen vorzunehmen. Dabei ist es denkbar, zu berücksichtigen, wie sich unterschiedliche Materialien zueinander verhalten. So kann eine Methode entwickelt werden, gleichartige Materialien zu verbinden und unterschiedliche Materialien zu trennen. Dazu ist auch zu beachten, wie sich die Übergänge zwischen den Materialien an solchen Stellen verhalten.
- Die Normalen an den Vertices können so angepasst werden, dass ein eher fließender Übergang an den Kanten stattfindet, der die Beleuchtung beeinflusst. Dieses Verhalten könnte vom Winkel zwischen den Flächen an den Kanten abhängig gemacht werden. Es ist auch hier möglich dieses Verhalten von den Materialien abhängig zu machen, so dass es Materialien mit härteren Kanten und weicheren Kanten gibt.

Ausblick

- Es ist denkbar, nicht ein Material pro Voxel zu verwenden, sondern verschiedene Mengen an verschiedenen Materialien innerhalb eines Voxels zu definieren, deren Gesamtmenge dann der Füllmenge entsprechen würde. Dazu muss die Methode zum Darstellen der Texturen auf den Voxeln angepasst werden.
- Den Materialien könnten noch renderspezifische Verhaltensweisen wie unterschiedliches Beleuchtungsverhalten oder Reflektionen zugewiesen werden.
- Es gibt Methoden dazu, Normal Mapping mit Triplanarem Mapping zu verbinden. Diese Ansätze ließen sich mit den Ergebnissen dieser Arbeit vereinen. Verschiedene Methoden Schatten zu berechnen, können im Zusammenhang mit dieser Methode getestet werden.
- Es ist denkbar zu versuchen die Methode so anzupassen, dass die Ausdehnung der Objekte im Verhältnis zu den Füllmengen der Voxel volumetrisch korrekt gerendert wird.

Vielen Dank für die Aufmerksamkeit!

Noch Fragen?

Bildquellen

1. http://www.cs.unc.edu/~marc/tutorial/img801.png
2. http://www.cs.carleton.edu/cs_comps/0405/shape/images/connectedobj.jpg
3. http://www.cs.carleton.edu/cs_comps/0405/shape/images/2Dintersected.jpg
4. http://www.cs.carleton.edu/cs_comps/0405/shape/images/blockyfoot1.jpg
5. http://www.cs.carleton.edu/cs_comps/0405/shape/images/intersectedfoot.jpg
6. http://3.bp.blogspot.com/-aNniGiuNEZM/U3PH6S4fkUI/AAAAAAAAAJg/xv-LmW2Gnds/s1600/triangleCorrected.png
7. http://1.bp.blogspot.com/-MPQU-iZaAWE/U3PV7DgIYuI/AAAAAAAAAJs/3nCm8f9KE4w/s1600/doesItBlend.jpg
8. http://www.martinpalko.com/wp-content/uploads/2014/03/3PlanarMaps-680x456.jpg
9. http://www.martinpalko.com/wp-content/uploads/2014/03/TriplanarMapped-680x325.jpg