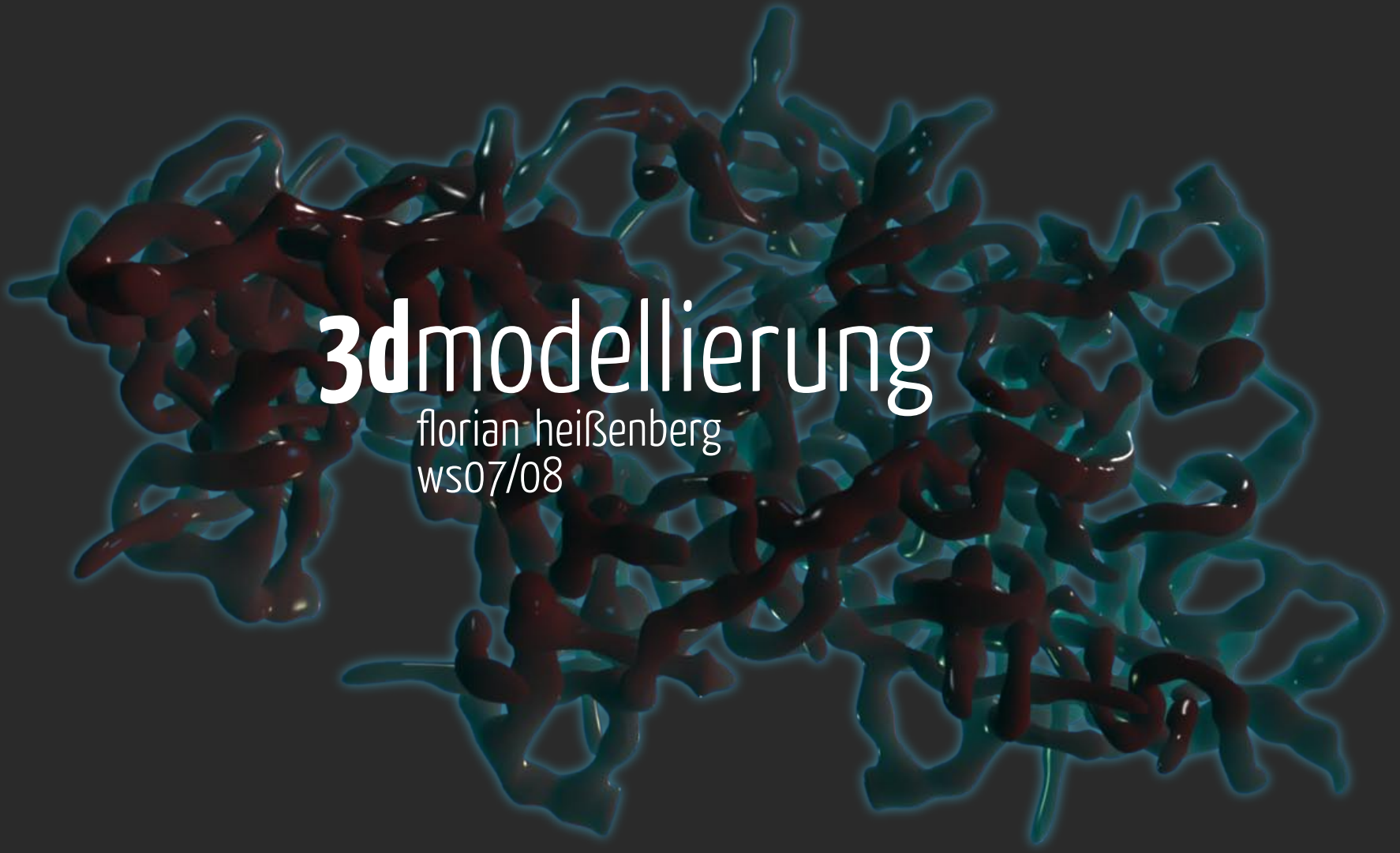
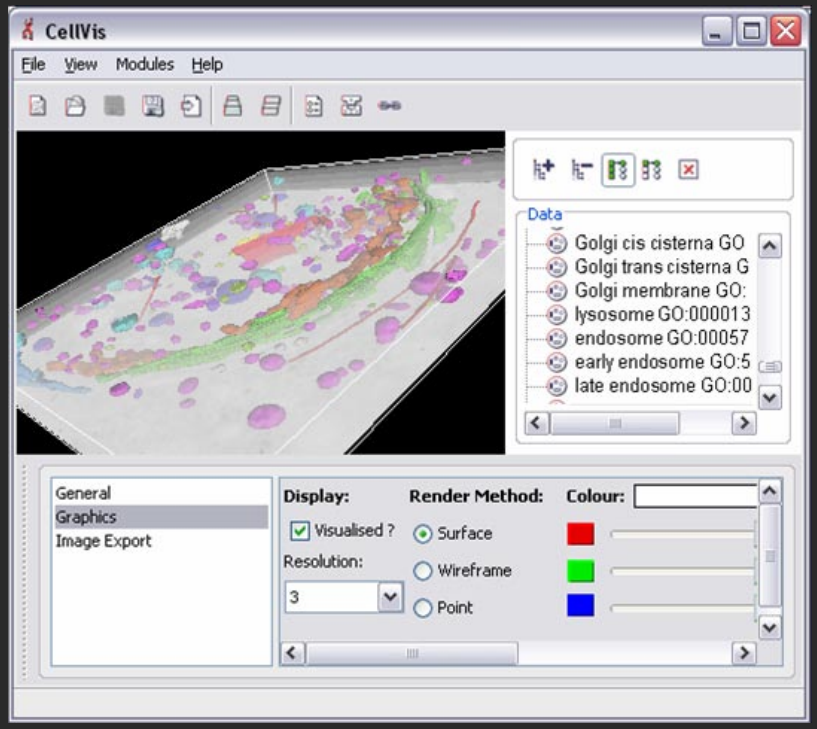


3dmodellierung

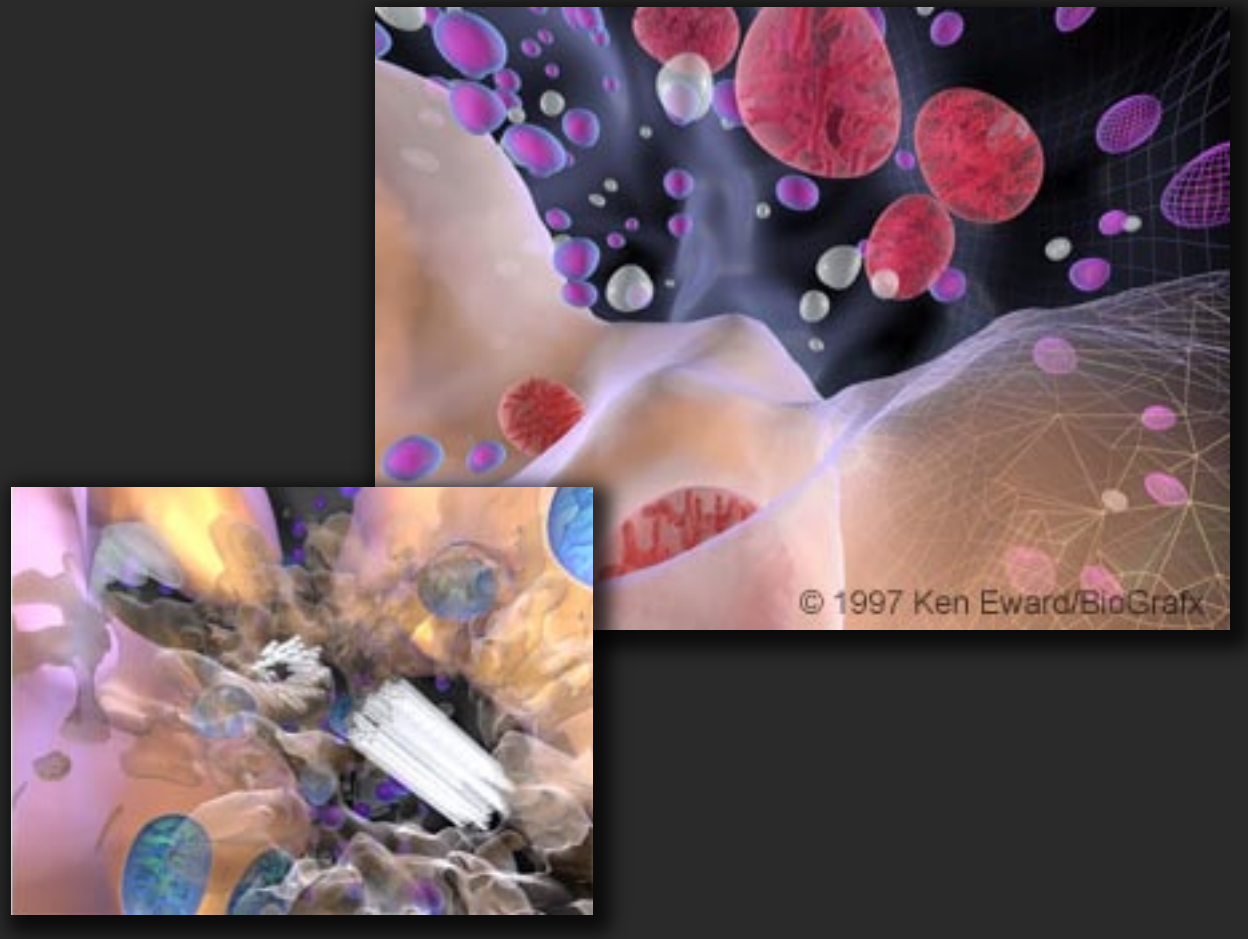
florian heißenberg
ws07/08

Visible Cell™

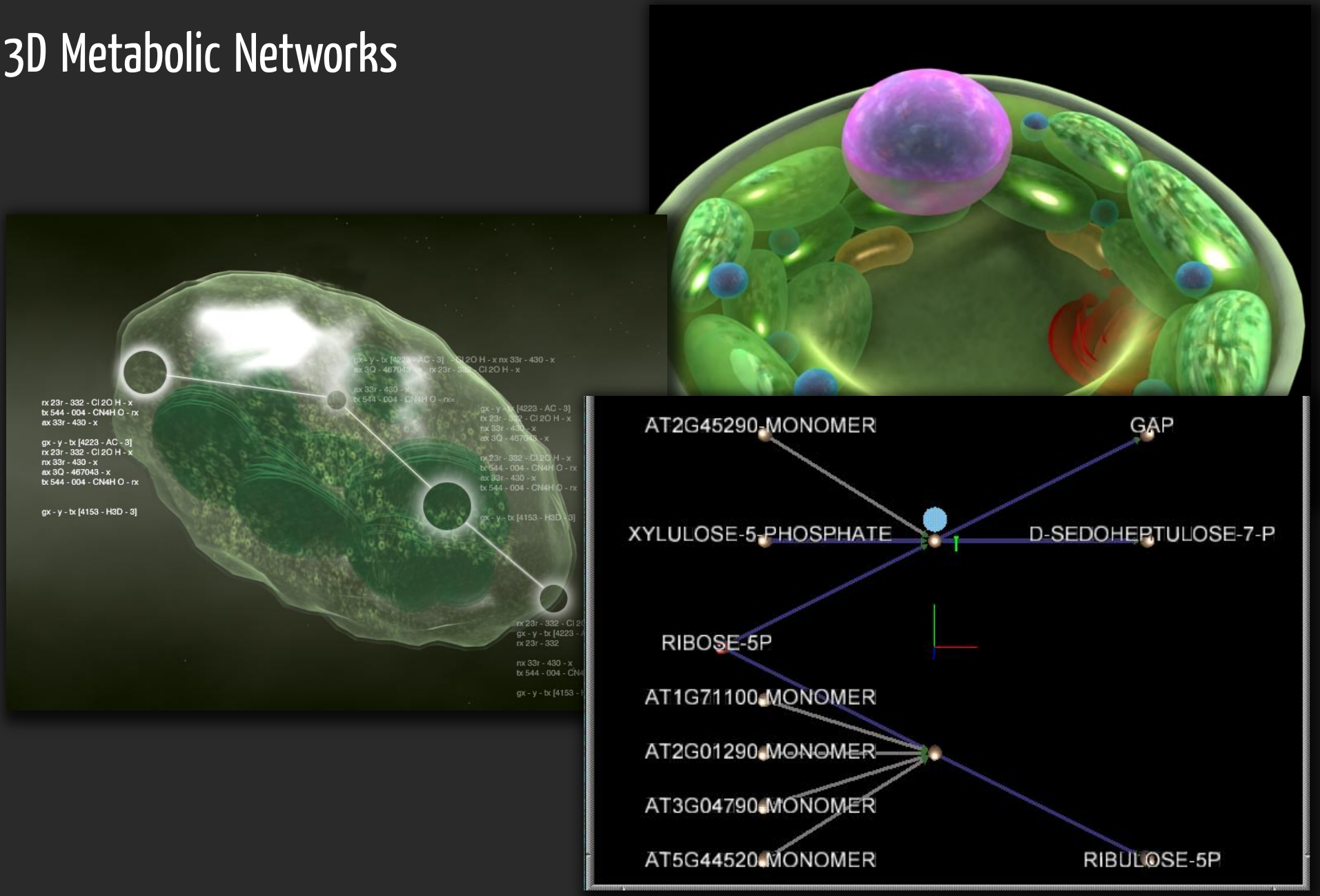
» auf 5nm präzise Darstellung einer echten Zelle



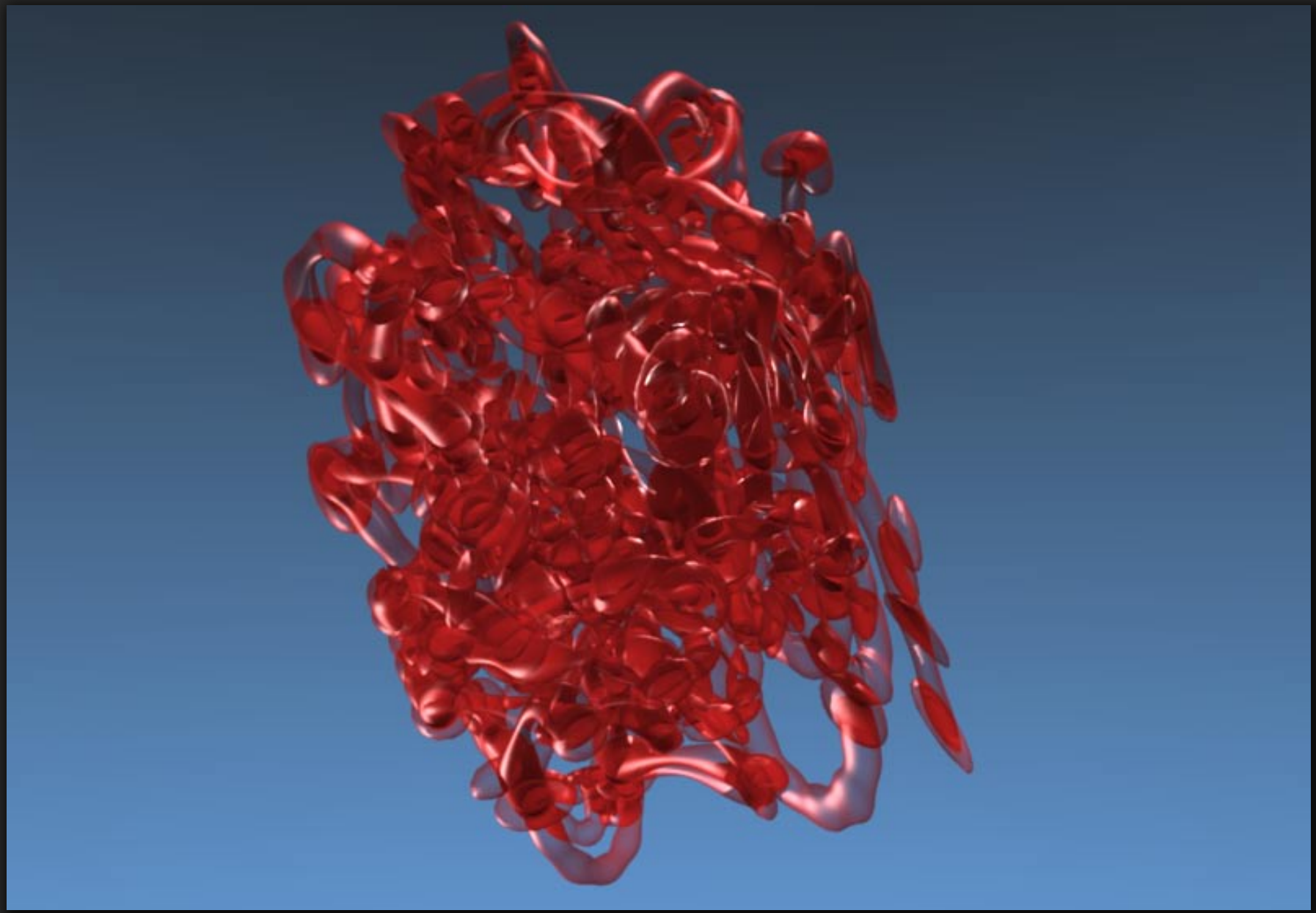
Cell Visualization Project



3D Metabolic Networks



realtime-rendering/optimierung



- Das aktuelle agranular endoplasmic reticulum ist für Echtzeitdarstellungen nicht optimal.
- über 450.000 Faces
- ausserdem semitransparent, daher kein Backface-Culling möglich
- enorme Dateigröße (18mb) durch viele Vertices
- nur sinnvoll bei starkem Zoom

Möglichkeiten zur Reduzierung:

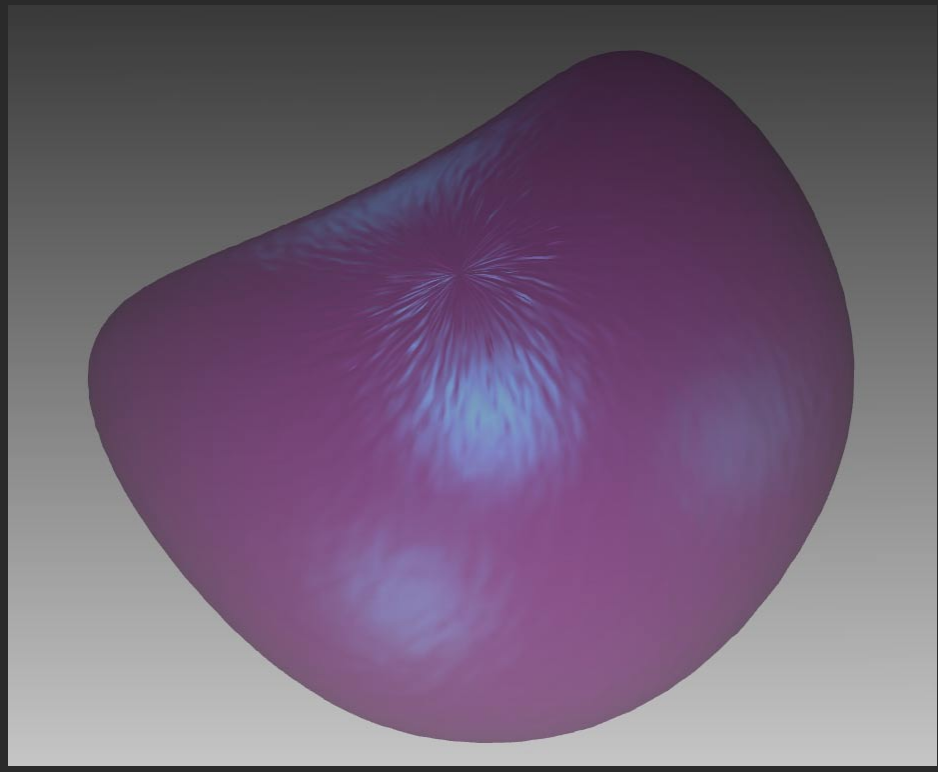
- Algorithmen zur Polygonreduzierung anwenden
- von Hand optimieren
- neu machen

Die visuelle Qualität sollte trotzdem erhalten bleiben, daher empfiehlt sich der Einsatz von LOD's (Level Of Detail), um den Detailgrad der Objekte an die Kameradistanz/den Zoomfaktor anzupassen.

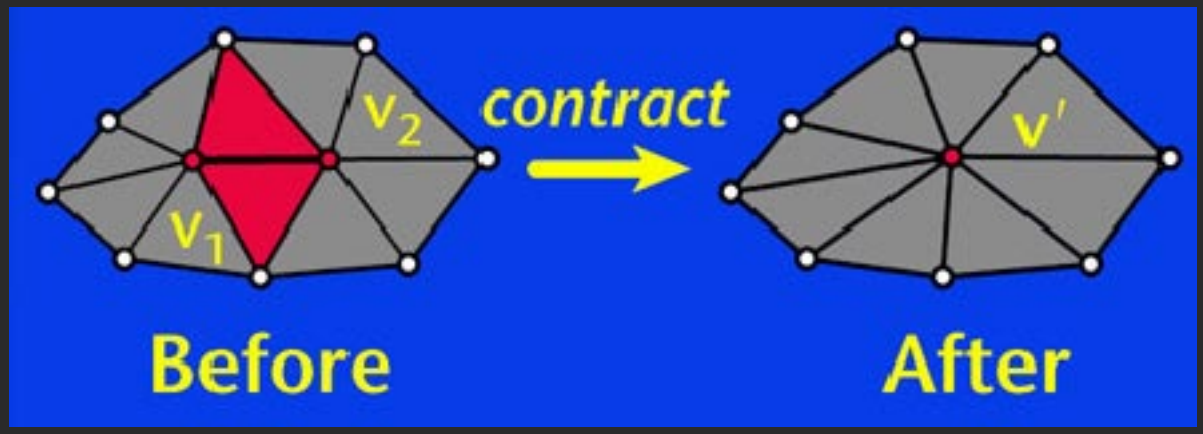
Die Detaillevel können dabei entweder mitgeliefert werden oder dynamisch erzeugt werden (aus den High-Poly-Modellen oder aus neuen Basismodellen), wobei der dynamische Ansatz zu bevorzugen wäre.

Vorschläge:

» Struktur der Oberflächen über Bump- oder Normalmaps erzeugen, statt zu modellieren
» Beispiel: Nucleus, 30.000 faces
» mit Bumpmap könnte man um den Faktor 5-10 Reduzieren.



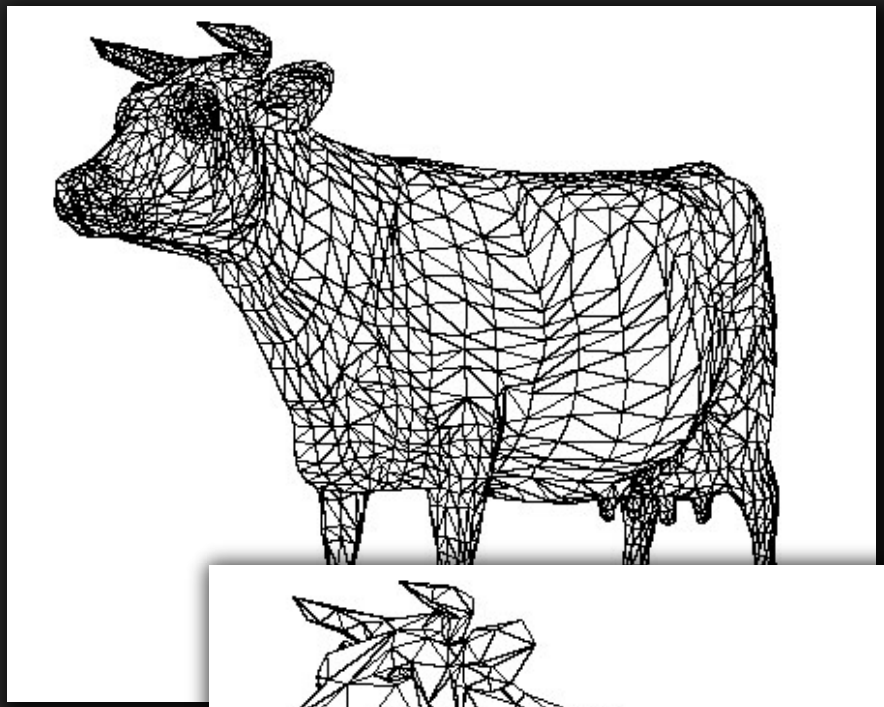
polyreducing

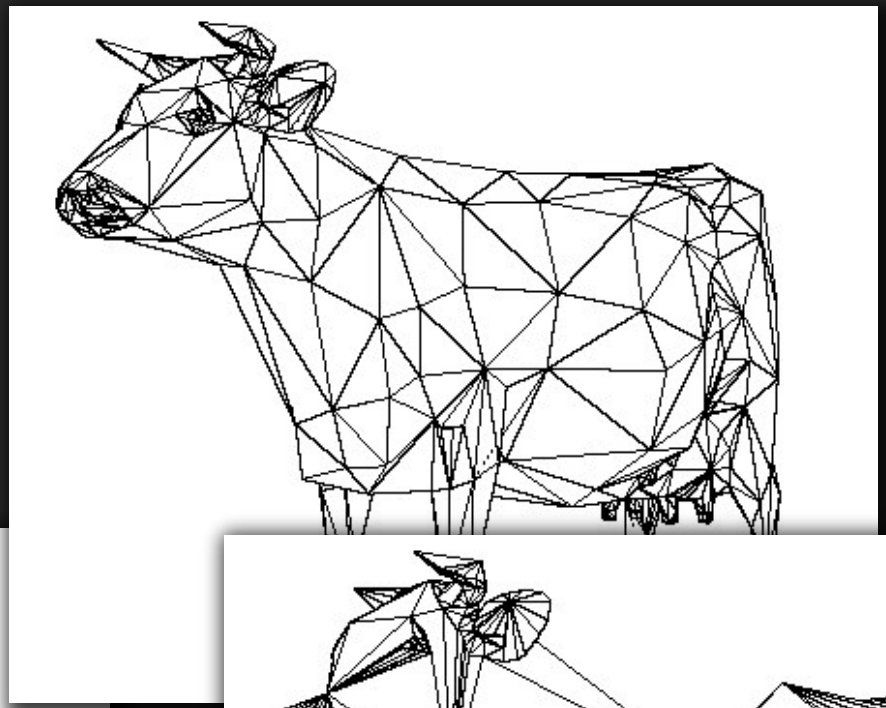


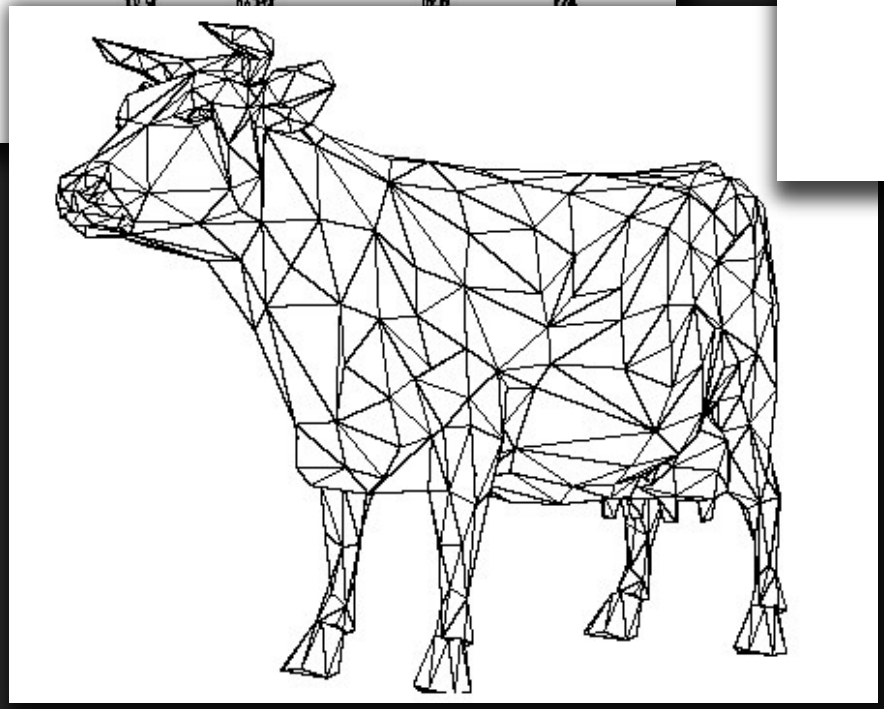
©okino

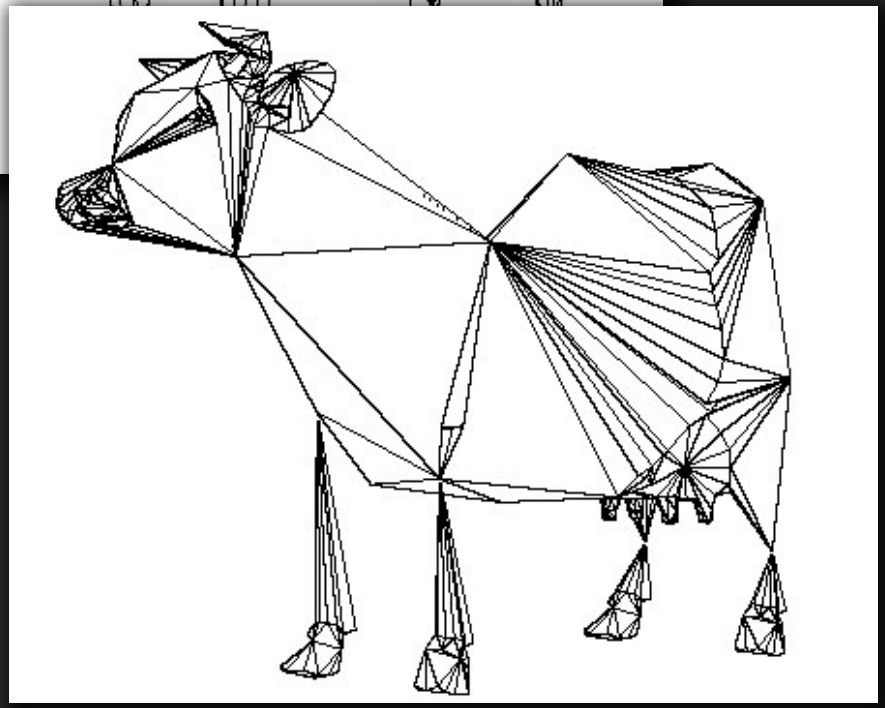
- Bsp.: Quadric-Based Polygonal Surface Simplification
  Edges mit geringem Abstand werden collapsed und so verschoben, das die geringste Abweichung zur urspünglichen Form entsteht.
- Texturkoordinaten können sich ändern
- Erfolg stark abhängig von Topologie
- Popping-Effekte können entstehen, evtl. zusätzlicher Programmieraufwand nötig

polyreducing



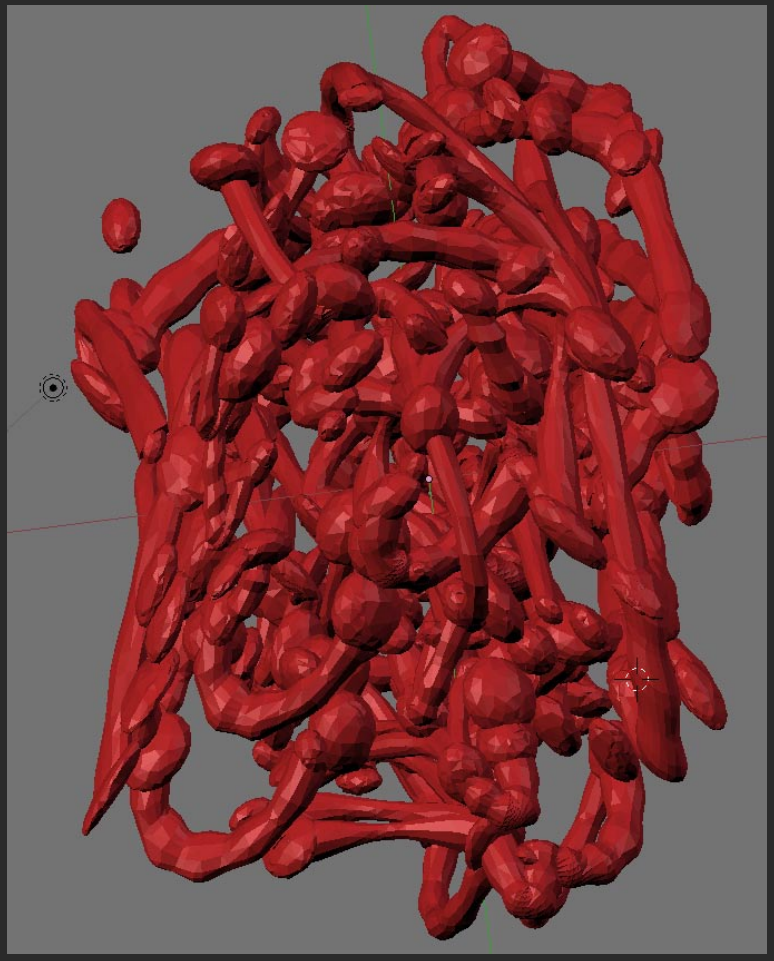




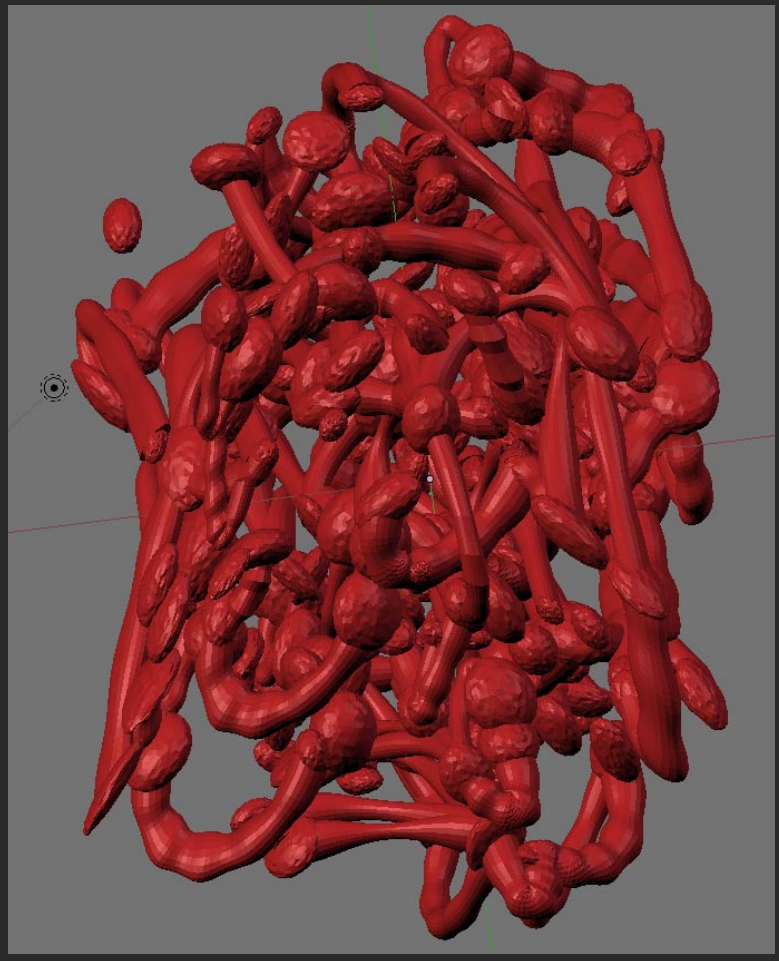


©okino

polyreducing



450.000 faces



110.000 faces

Das Blender-Skript ist dafür etwa 15 MInuten gelaufen. Da andere Programme nur bedingt schneller sind, ist dynamisches Erzeugen von LOD-Stufen mit diesen Modellen unmöglich.

- LOD-Modelle müssten also mitgeliefert werden
- Ladezeiten verschlechtern sich (alle Stufen müssen geladen werden)
- Speicherplatz wird verschwendet

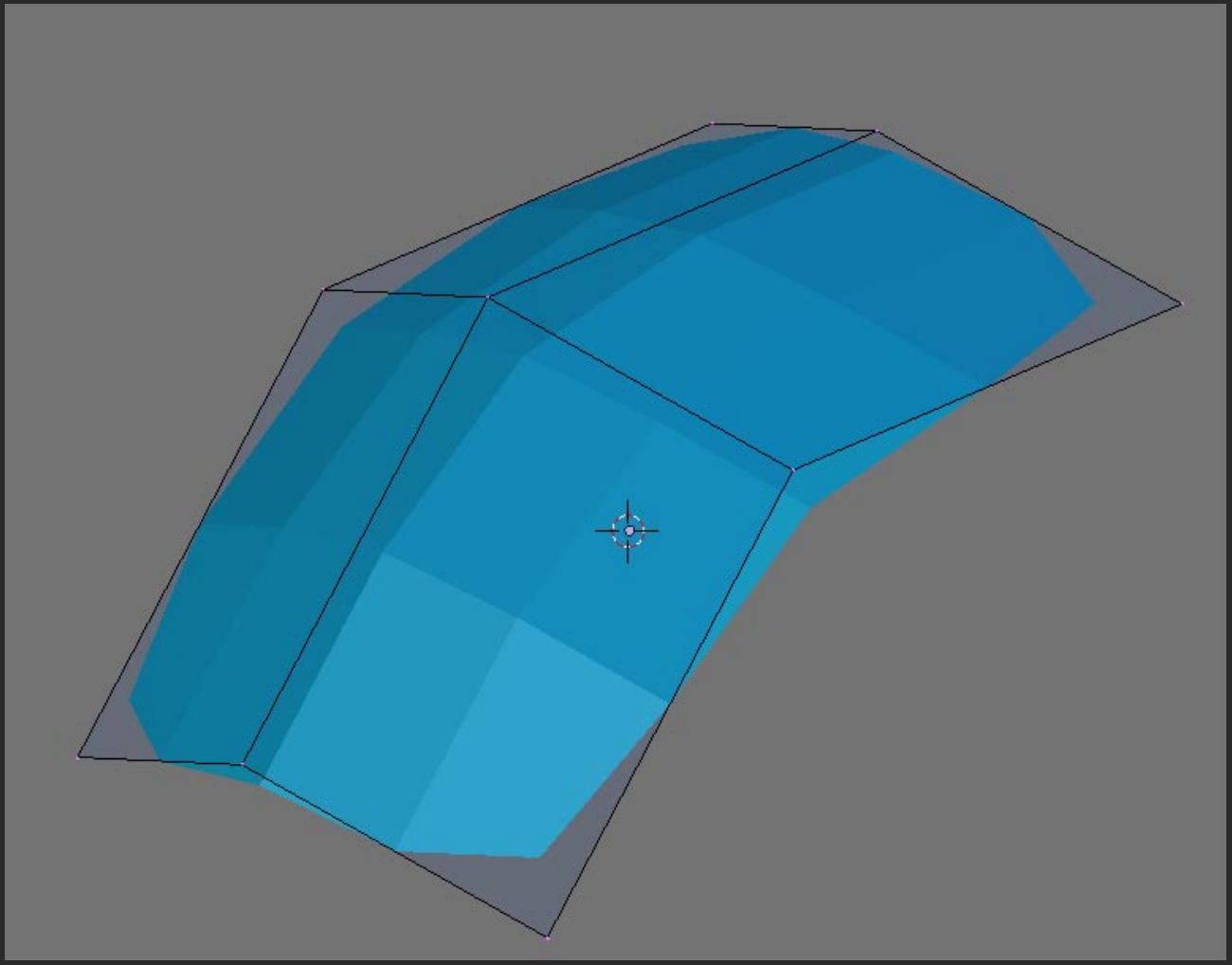
Deshalb: umgekehrter Weg, von einfachen Basismodellen bei Bedarf zu detaillierteren Modellen.

» Implementierung des Catmull-Clark-Algorithmus um Subdivision Surfaces zu benutzen.
» weit verbreitetes Verfahren, Implementierung für Java vorhanden

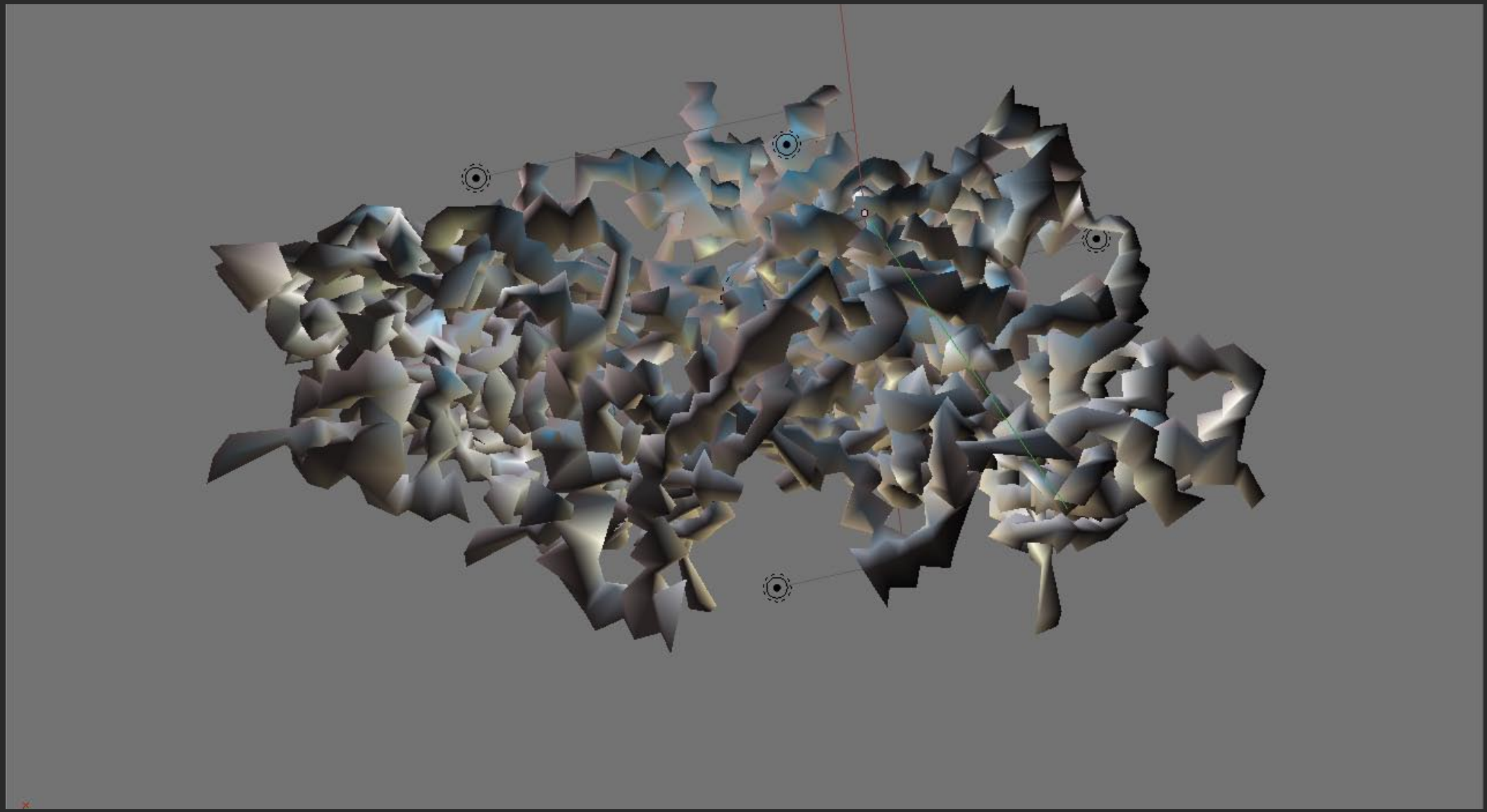
Vorteile:

» sehr einfaches Basismesh, könnte als Bounding Volume benutzt werden
» Oberflächen können je nach vorhandener Hardware und/oder Abstand zur Kamera unterschiedlich oft subdivided werden
» Texturkoordinaten bleiben erhalten
» schnell

catmull-clark

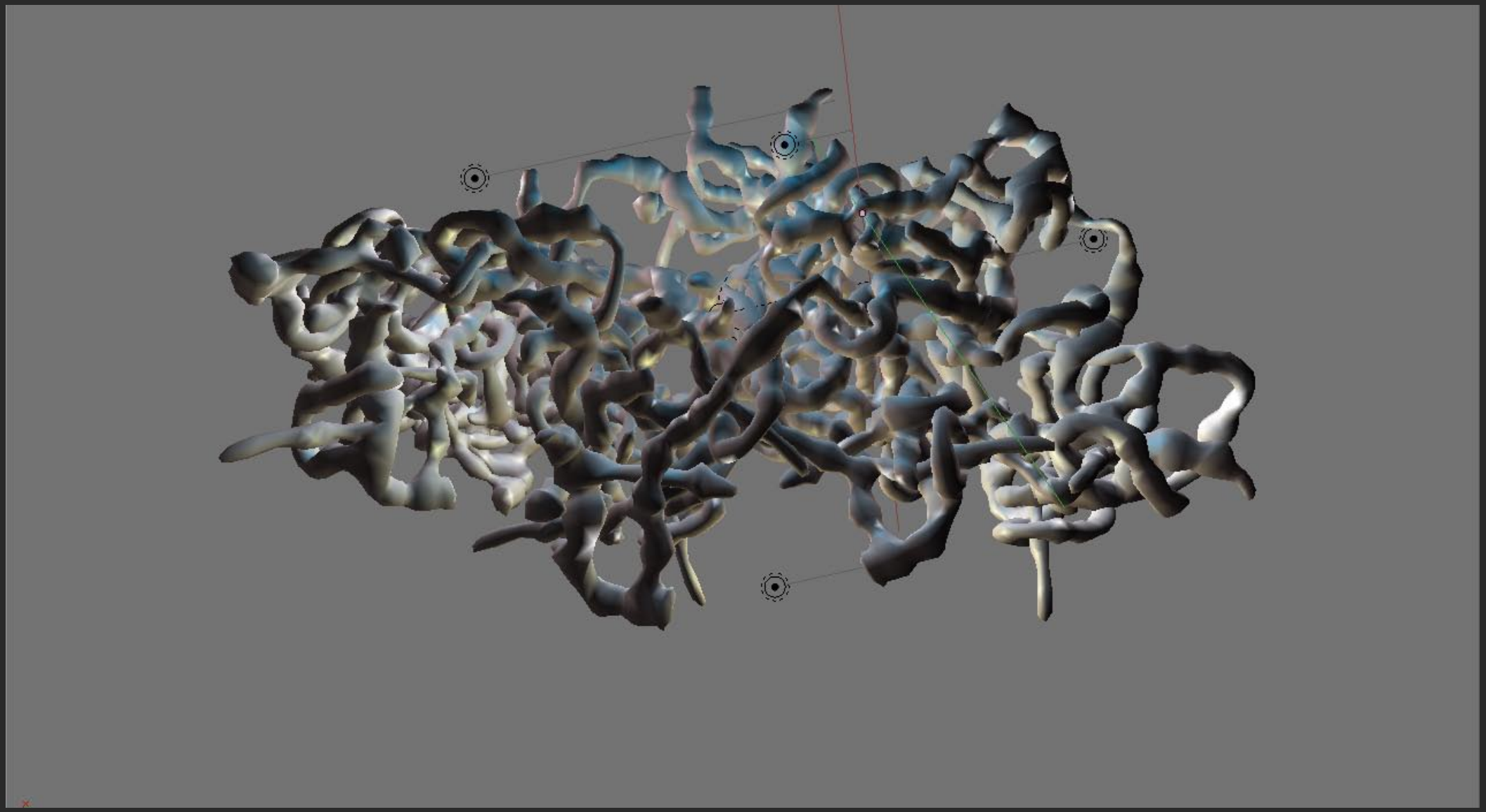


a.e.r. mit catmull-clark



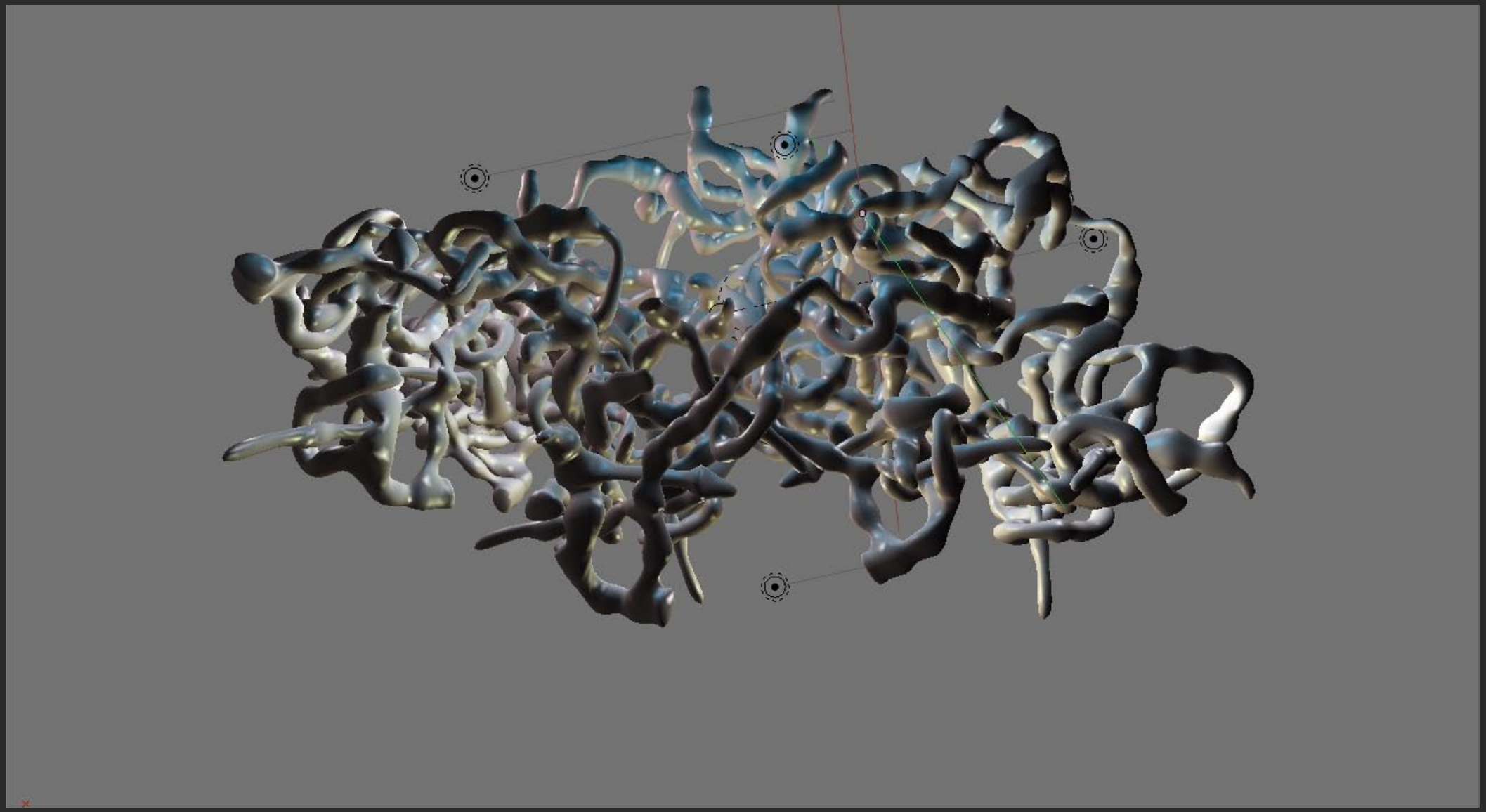
0 subdivisions, 7500 faces

a.e.r. mit catmull-clark



1 subdivision, 30.000 faces

a.e.r. mit catmull-clark



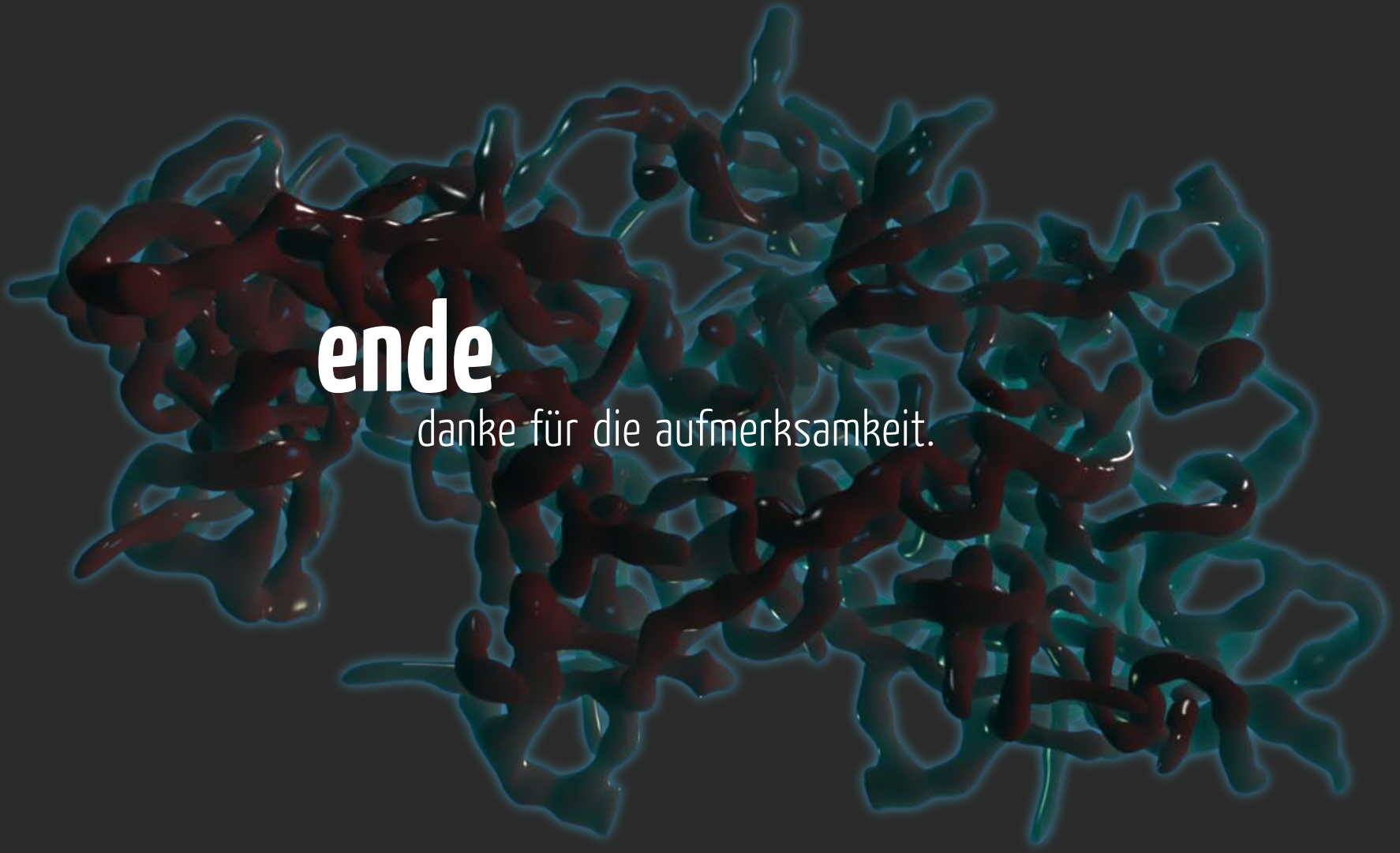
2 subdivision, 121.000 faces

Blender

- Open Source 3D-Software
- läuft auf Linux, OSX, Windows, Solaris ...
- für CAD eher ungeeignet
- hauptsächlich für Polygonmodelle, NURBS etc. eher verbesserungsfähig
- Integrierter Renderer unterstützt Raytracing, Radiosity, HDR etc.
- ausserdem Exportskripte für YAFRAY, POVRay, PRman ...
- Import VRML 1.0
- Export von X3D, VRML 1.0 und VRML97/

3dsmax (r7)

- kommerzielles 3D-Package von discreet
- für CAD geeignet
- nur für Windows
- Integration von Renderern wie Vray, mental ry, finalRender, Brazil R/S
- Unterstützung von NURBS, Ngons etc.
- (teilweise) buggy
- manche Funktionen müssen nachgerüstet werden (Meshtools, Texporter ...)



ende

danke für die aufmerksamkeit.

quellen

- Michael Garland: Quadric-Based Polygonal Surface Simplification
- Edwin Catmull & Jim Clark: Recursively generated B-spline surfaces on arbitrary topological surfaces
- Okino's Polygon Reduction System
- blender.org