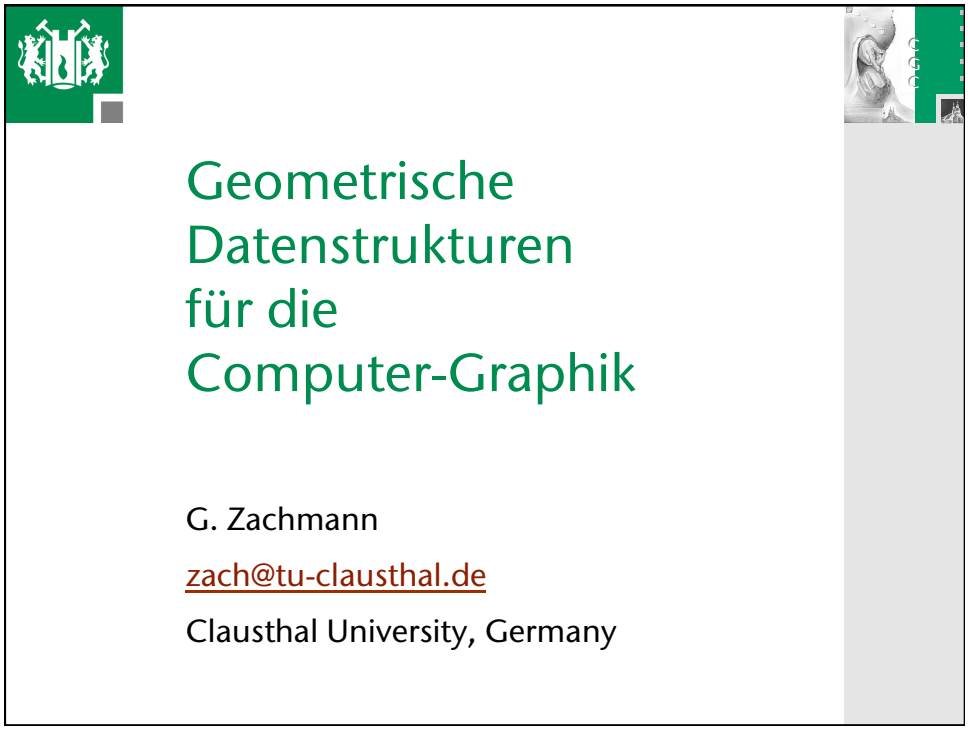
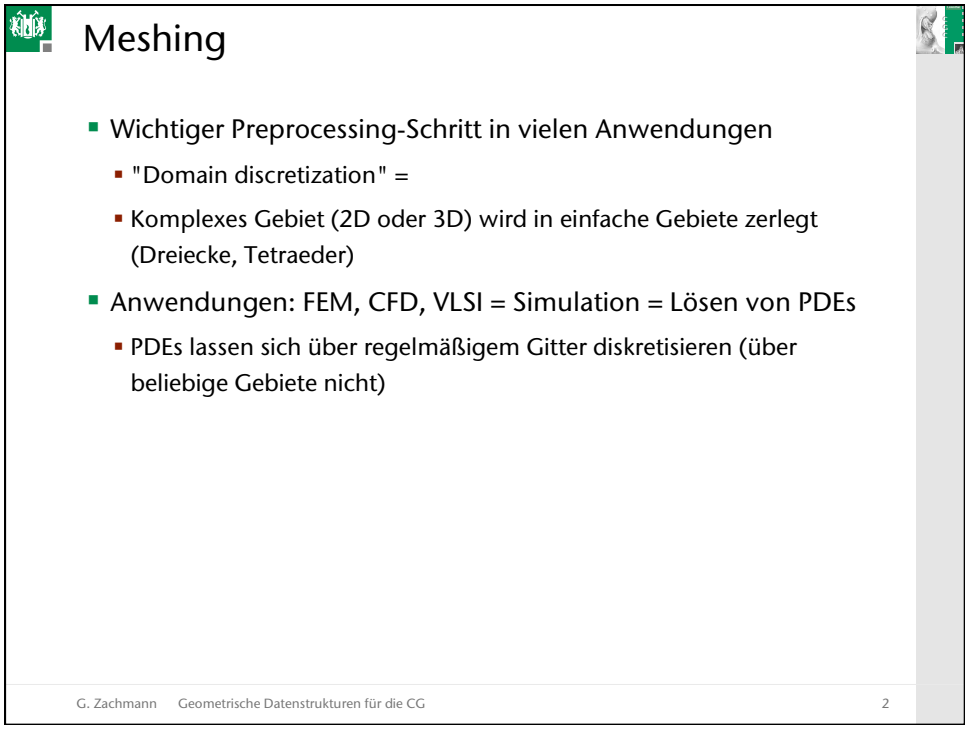


Geometrische Datenstrukturen für die Computer-Graphik

G. Zachmann

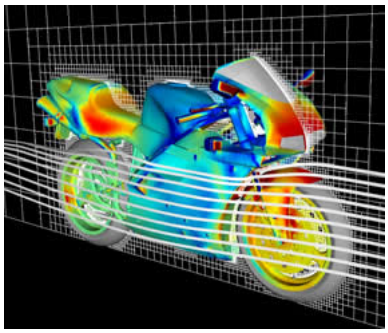
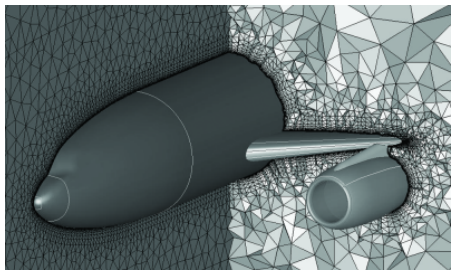
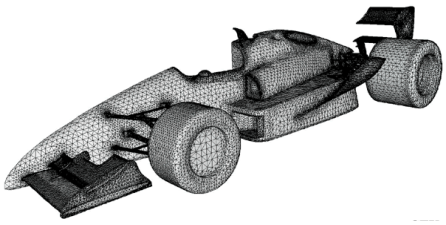
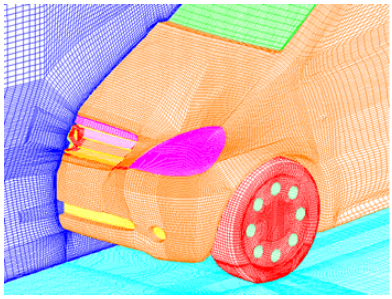
zach@tu-clausthal.de

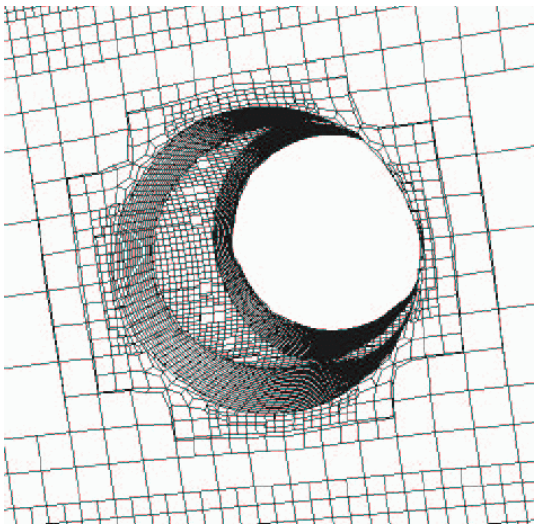
Clausthal University, Germany



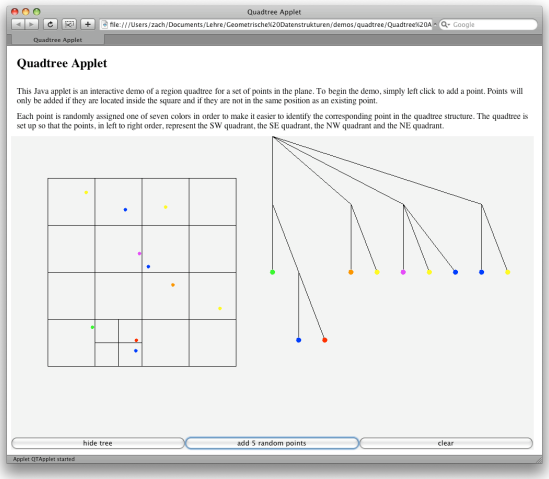
Meshing

- Wichtiger Preprocessing-Schritt in vielen Anwendungen
  - "Domain discretization" =
  - Komplexes Gebiet (2D oder 3D) wird in einfache Gebiete zerlegt (Dreiecke, Tetraeder)
- Anwendungen: FEM, CFD, VLSI = Simulation = Lösen von PDEs
  - PDEs lassen sich über regelmäßigem Gitter diskretisieren (über beliebige Gebiete nicht)

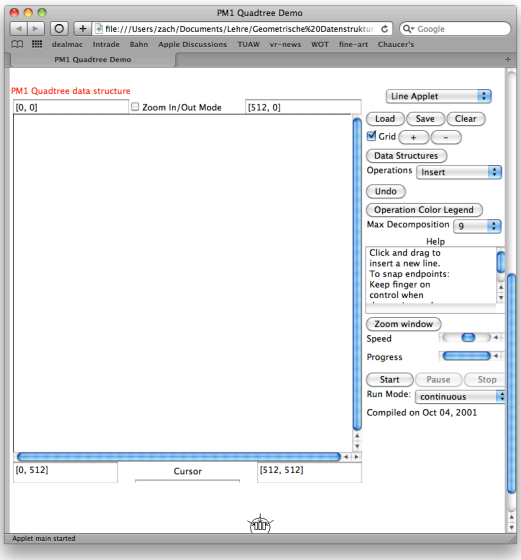







Quadtree demo



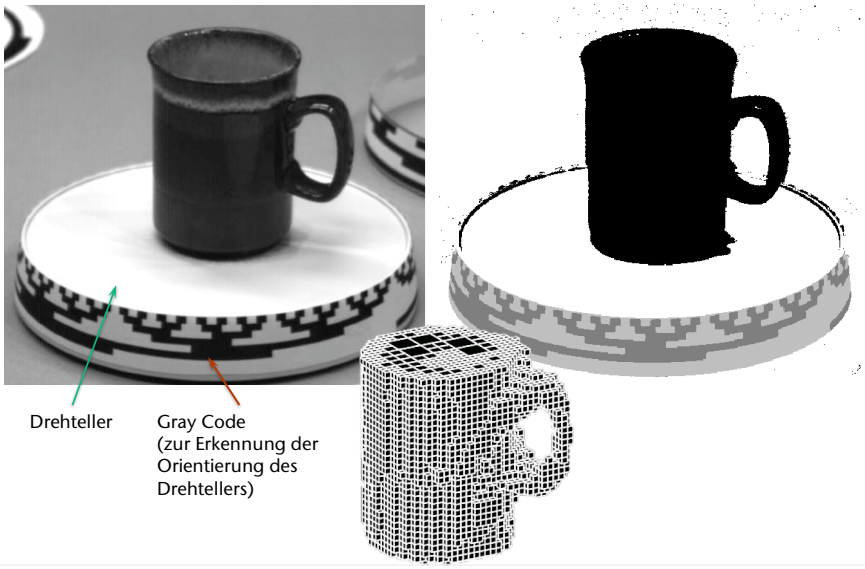
http://www.cs.utah.edu/~croberts/courses/cs7962/project/index.html



Ist IMHO buggy! (unnötige Unterteilungen)

http://donar.umiacs.umd.edu/quadtree/points/prquad.html

Octree Models from Images



Drehteller

Gray Code (zur Erkennung der Orientierung des Drehtellers)

Example Models

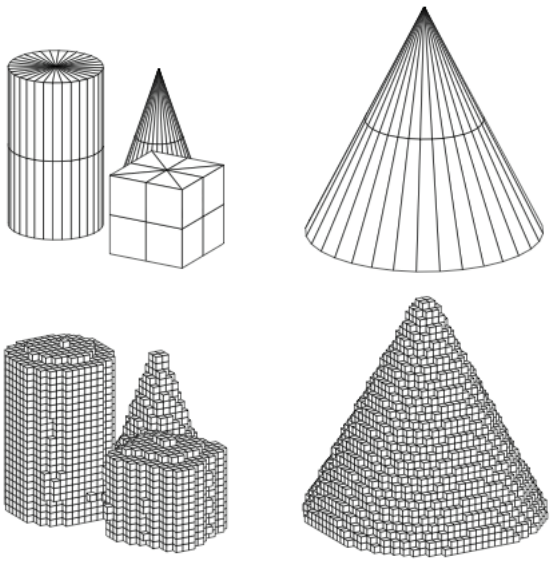
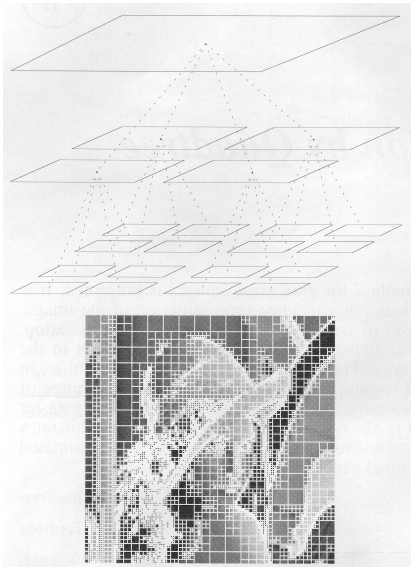
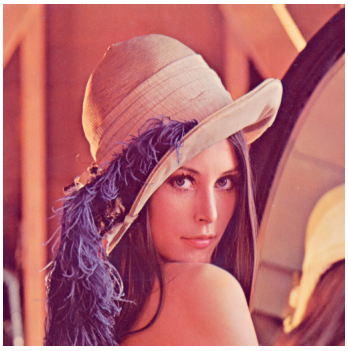


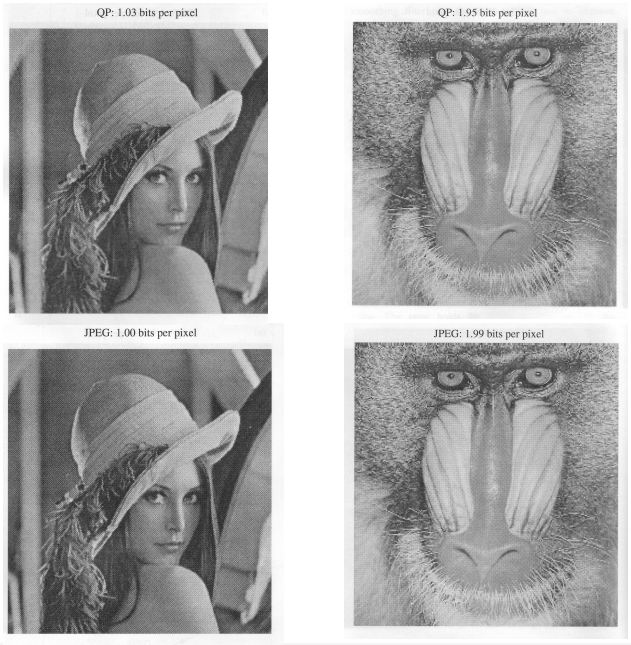
Image Compression using Quadtrees



Die beiden Test-Bilder schlechthin



Resultate

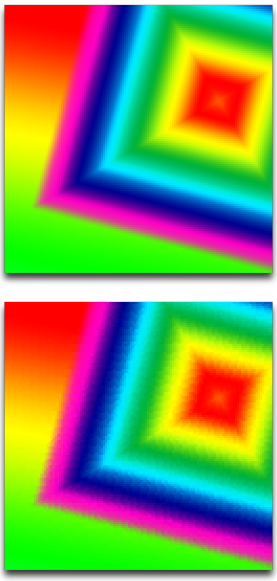


S3TC texture compression

- Vergleich:

DXT1 Uncompressed

[Philipp Klaus Krause]



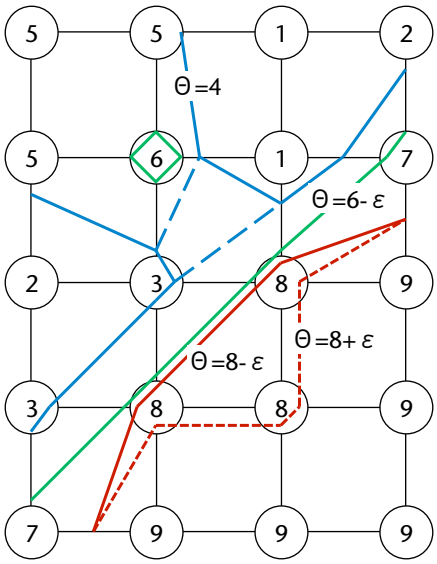
[Simon Brown]

- Vorteil: größere Texturen möglich → höhere Qualität
- Beispiel aus der Unreal Engine:

uncompressed — Unreal Retexturing Project — mit S3TC

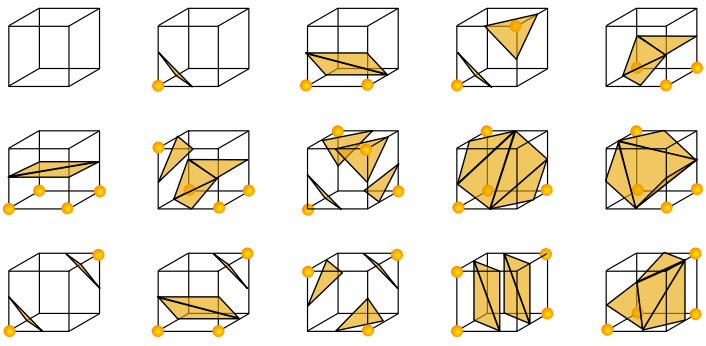
Isosurfaces

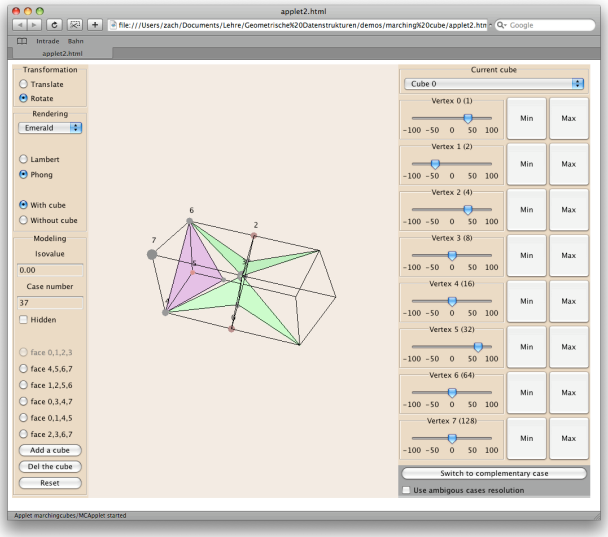
- Beispiel zur Motivation:
  - Gegeben ist ein 2D-Höhenfeld
  - Gesucht ist eine Visualisierung (in 2D!), so daß man die Form / den Verlauf des Höhenfeldes gut "erkennt"
- Eine Möglichkeit: Höhenlinien (= Konturen)
- Mögliche Probleme:
  - Plateaus
  - Singuläre "Isopunkte"
  - Uneindeutigkeiten



Beispiele für Höhenfelder und Volumendatensätze

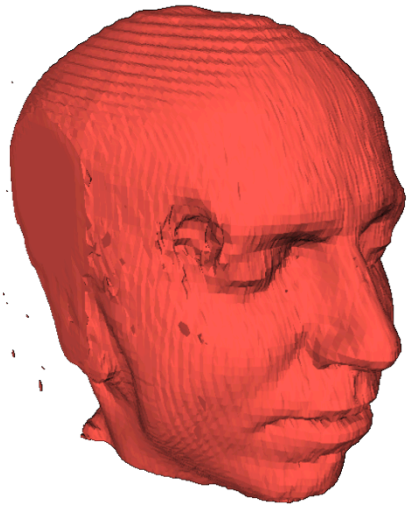
- Die 15 echt verschiedenen Fälle in 3D:



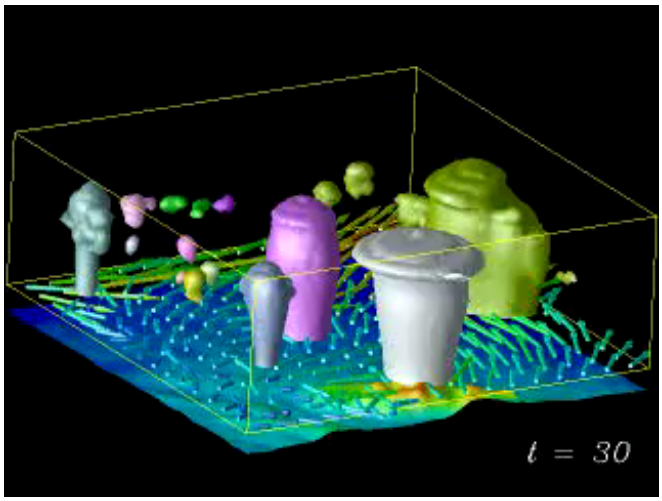


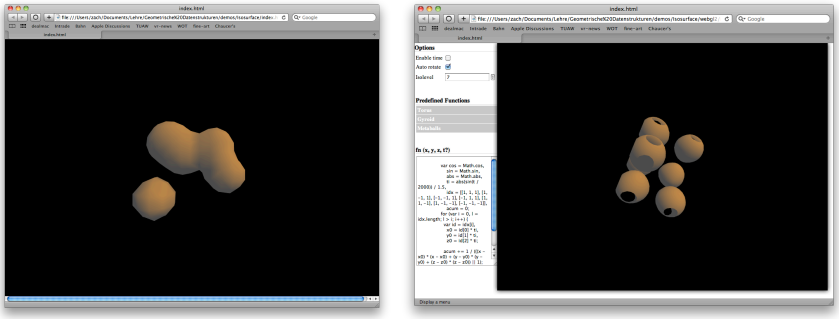
http://users.polytech.unice.fr/~lingrand/MarchingCubes/accueil.html

- Output eines einfachen Marching-Cube-Algorithmus':



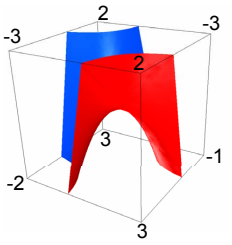
Beispiel aus einer Wetter-Simulation



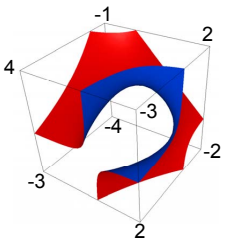


http://blog.thejit.org/2010/12/10/animating-isosurfaces-with-webgl-and-workers/

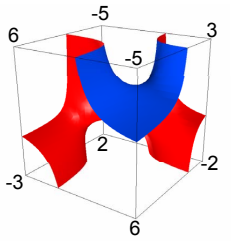
Knifflige Fälle für jeden Isosurface-Algorithmus



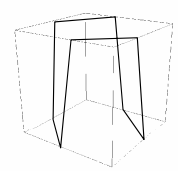
8-sided polygon



9-sided polygon



12-sided polygon

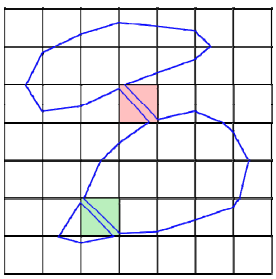


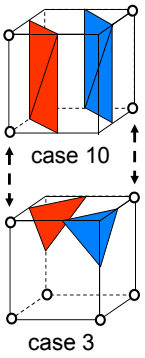
The 8-sided polygon has no valid triangulation!

- either some triangles lie on faces of the cell
- or an extra vertex has to be used

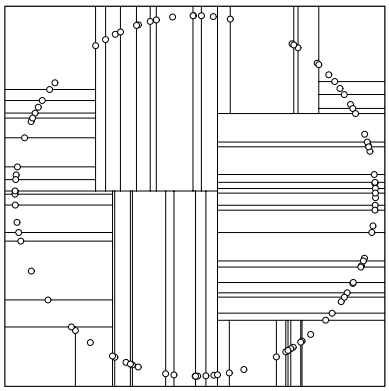
~/avs/networks/SciVis/AD*net

- Manchmal passen die Dreiecke der benachbarten Zellen nicht zusammen:
- Uneindeutiger Fall im 2D:

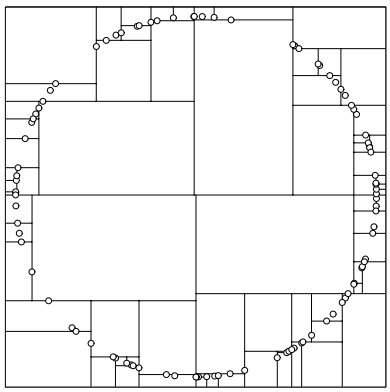




Pivot-Strategien beim Aufbau von kd-Trees

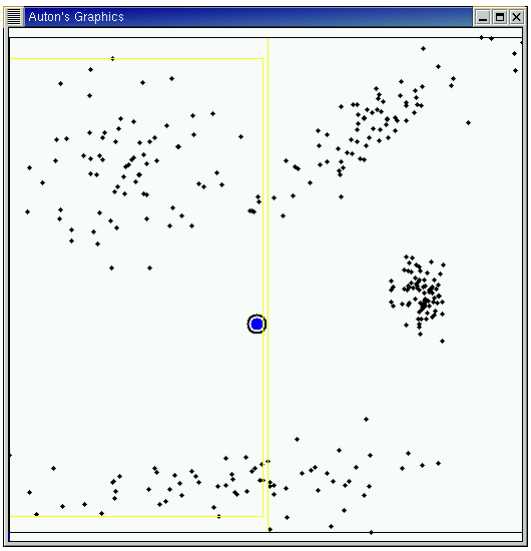


median along the dimension
with the widest spread of the points

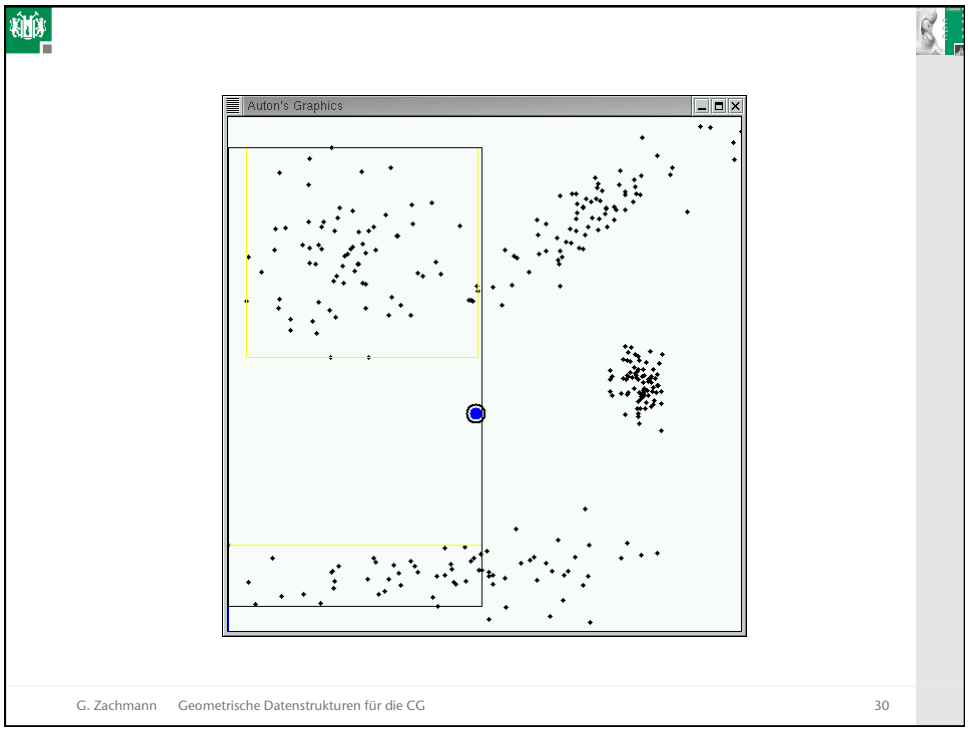


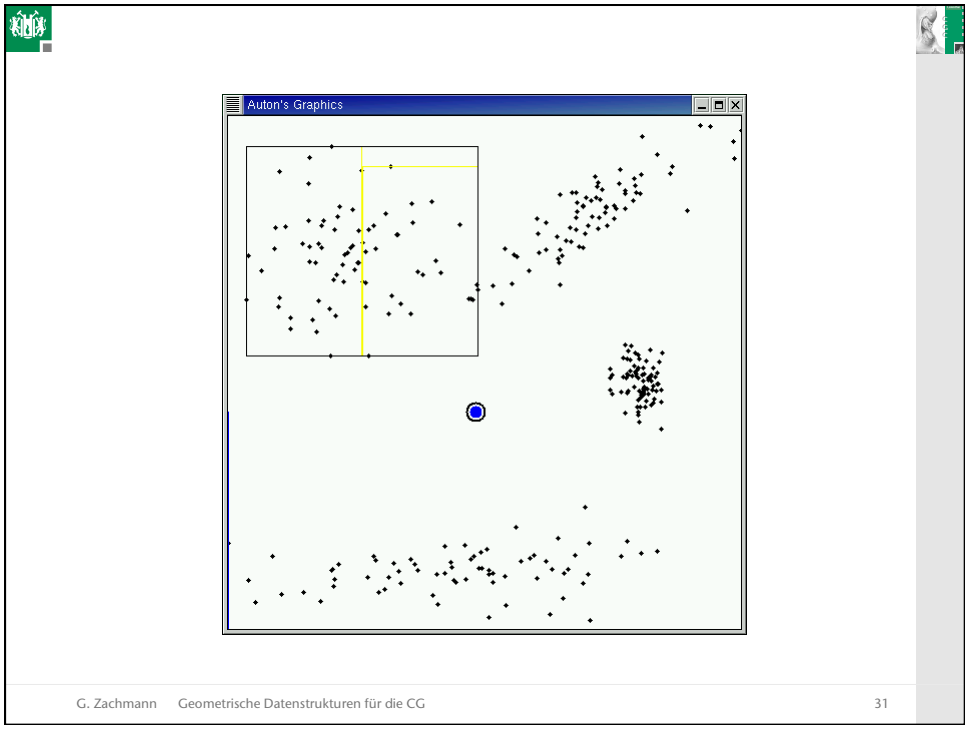
the point closest to the center along the
dimension with longest side of the region

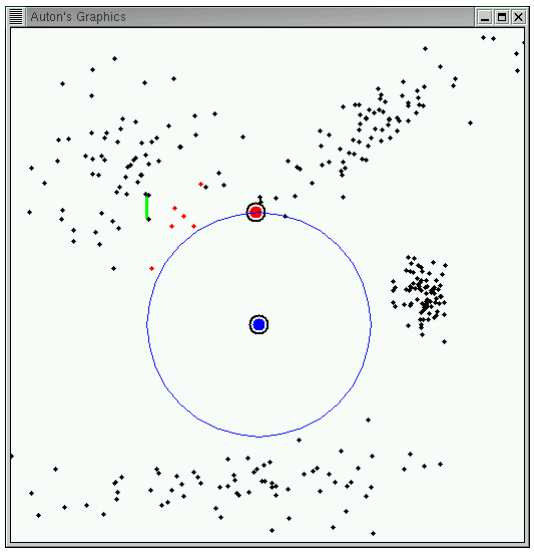
Animation of Nearest-Neighbor using kd-Trees

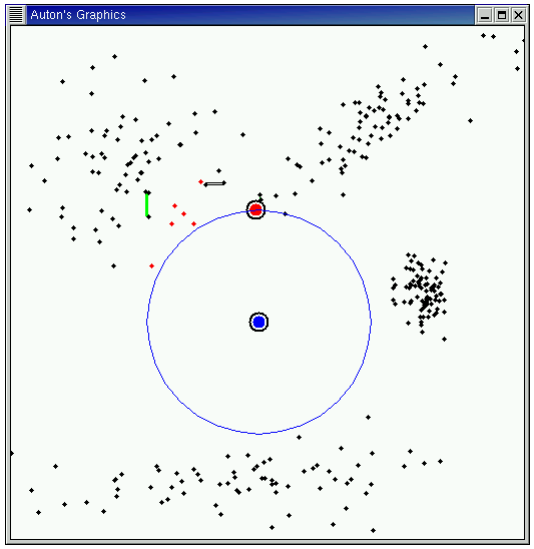


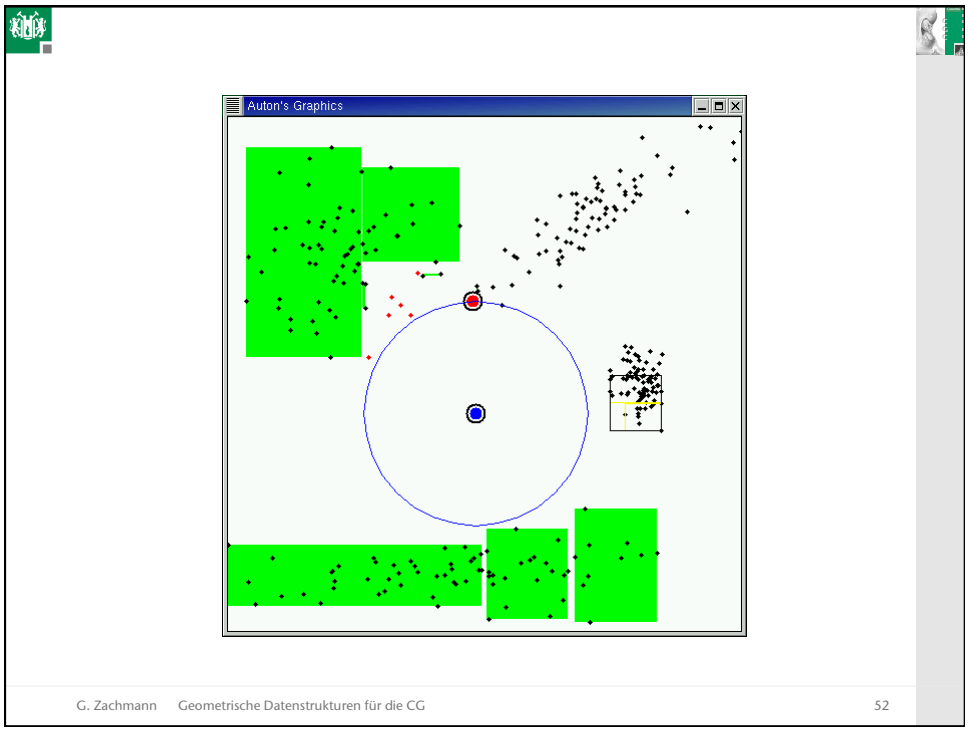
Andrew Moore, CMU

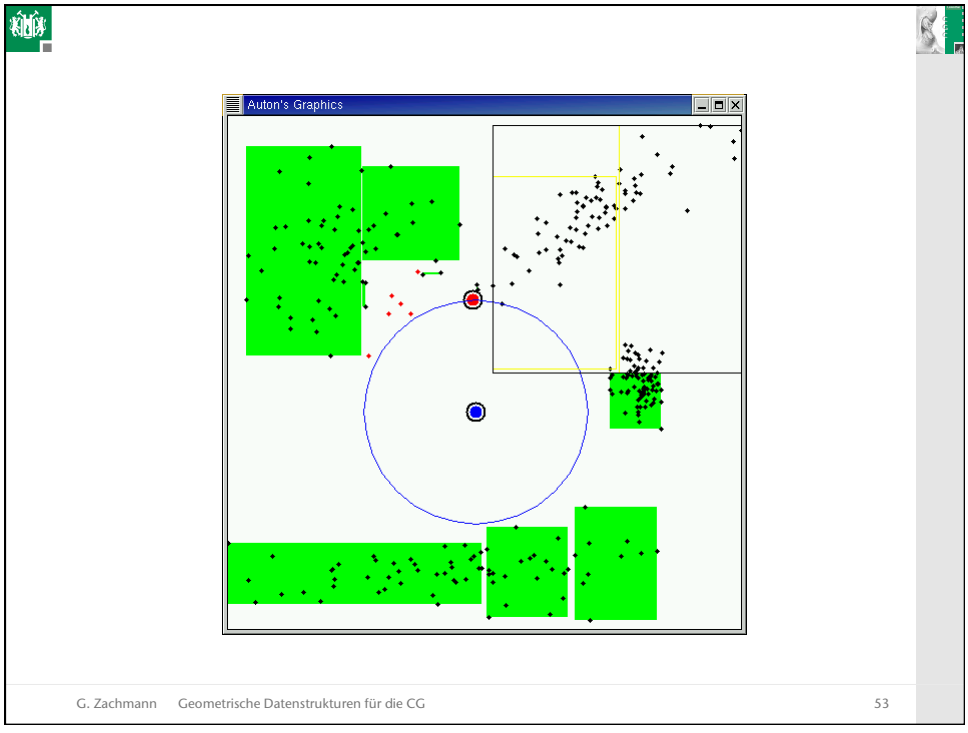

Auton's Graphics

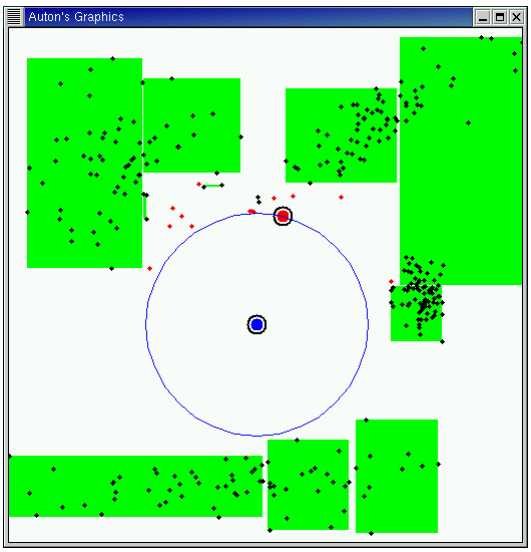

Auton's Graphics


Auton's Graphics

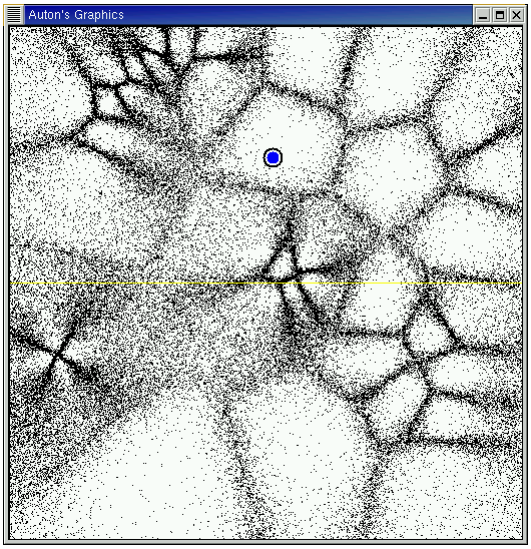

Auton's Graphics

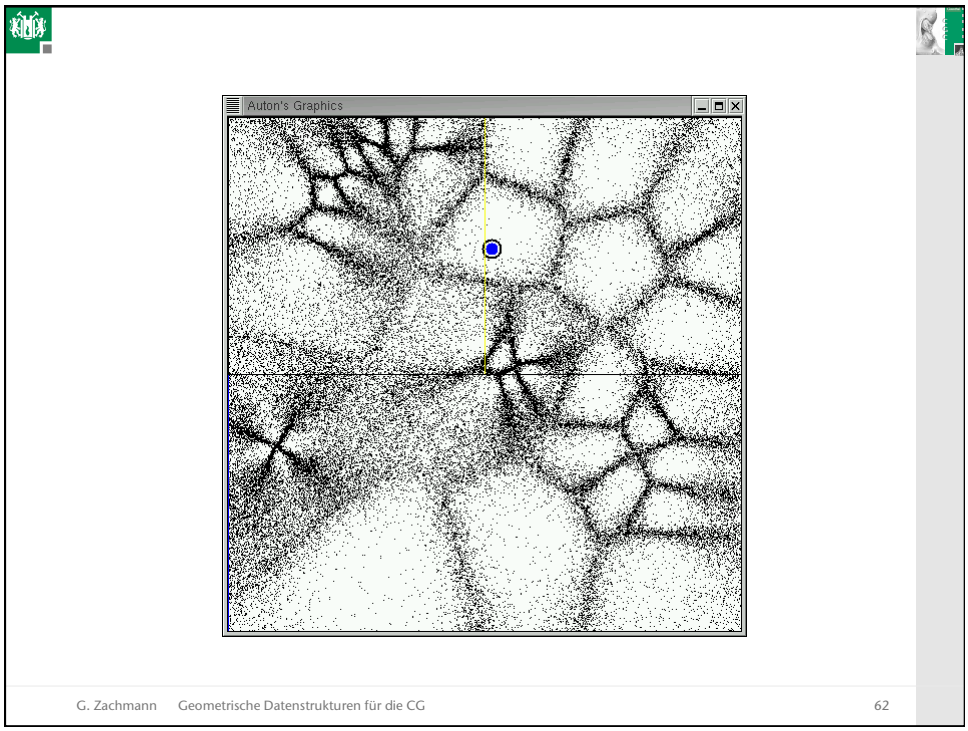

Auton's Graphics

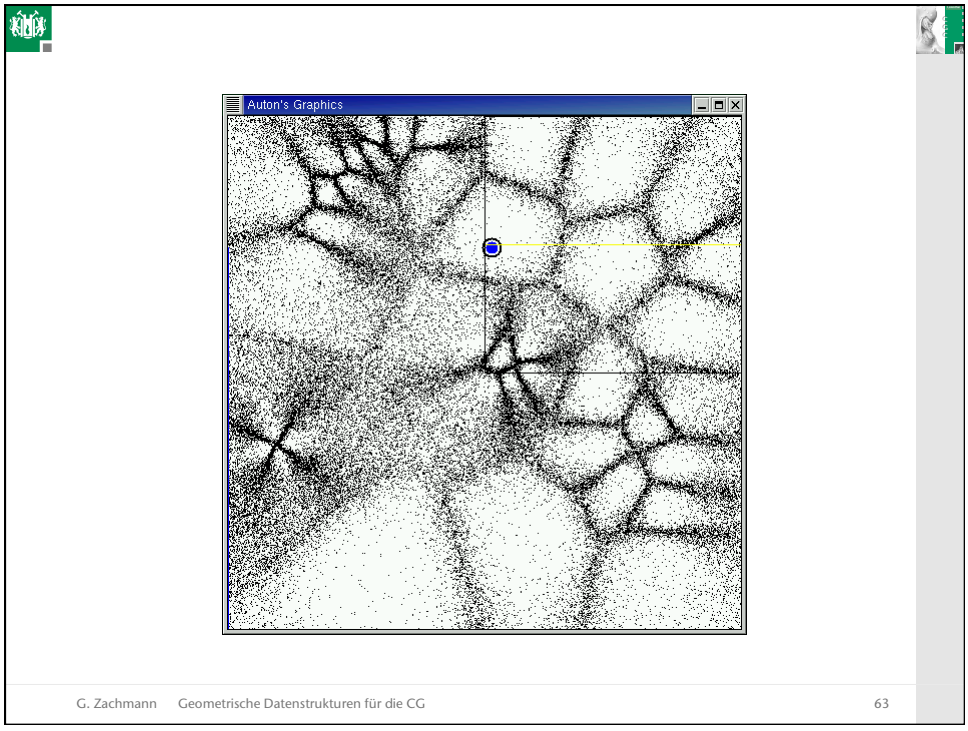

Auton's Graphics

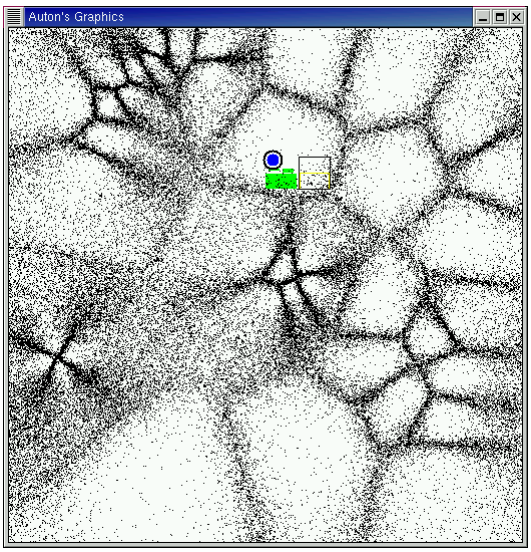


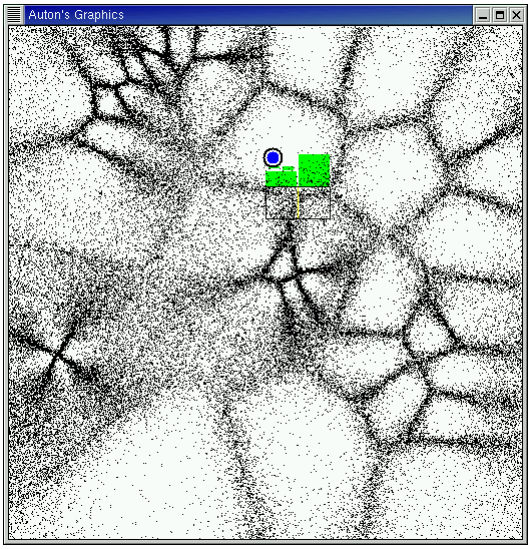
Animation of NN search with large data set

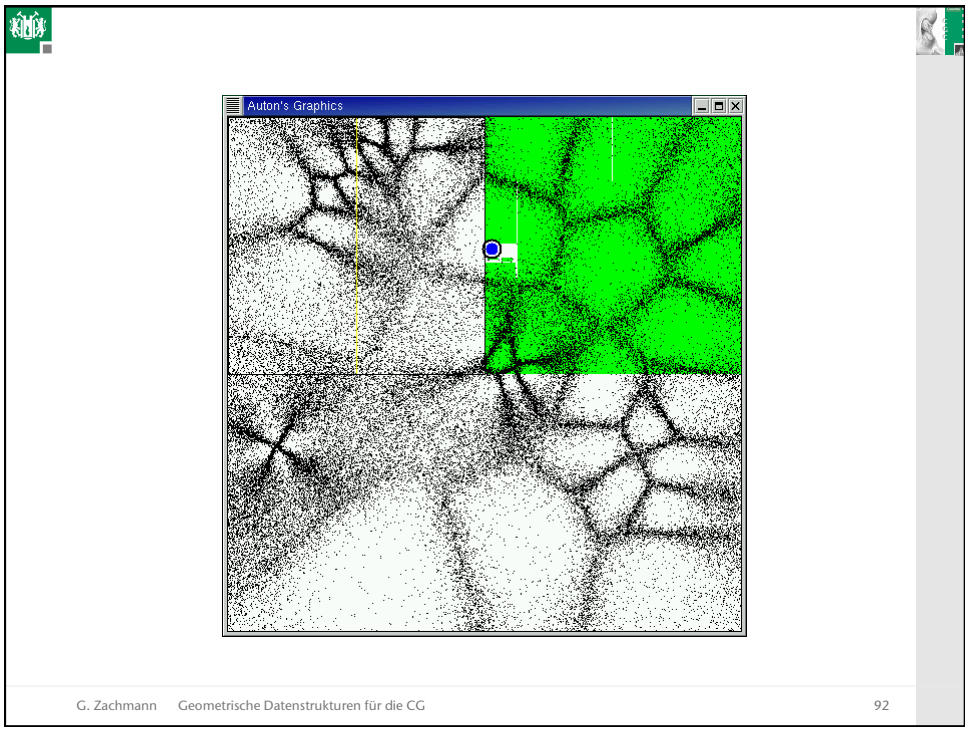


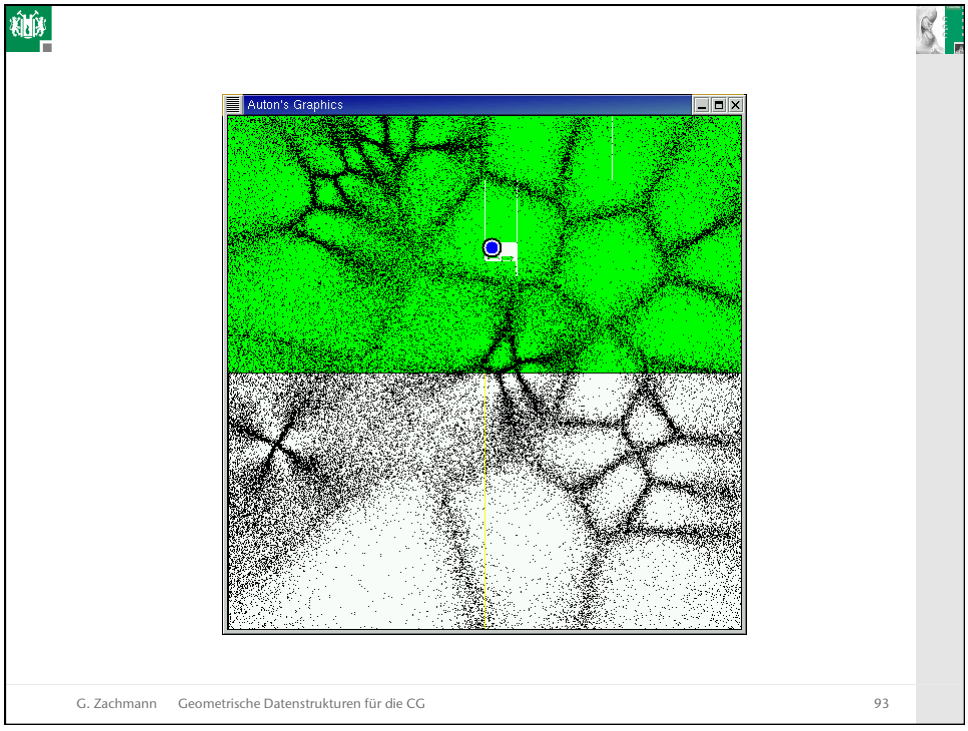

Auton's Graphics

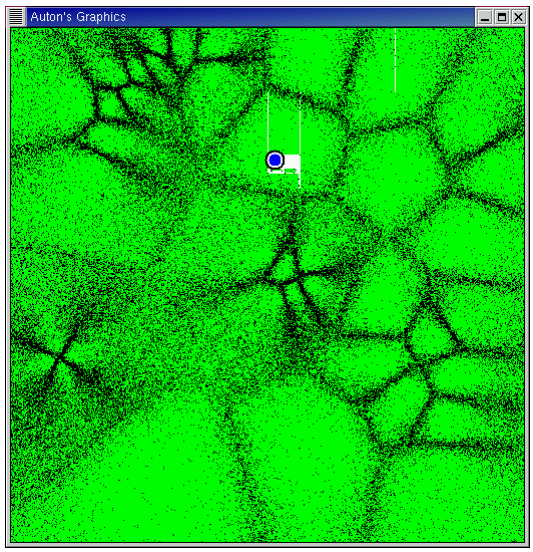

Auton's Graphics


Auton's Graphics

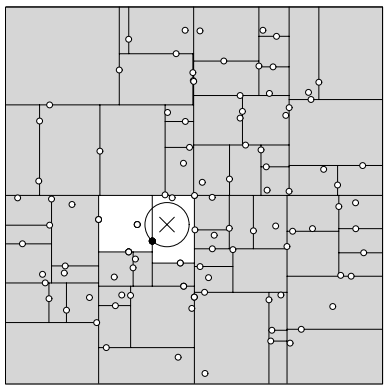

Auton's Graphics

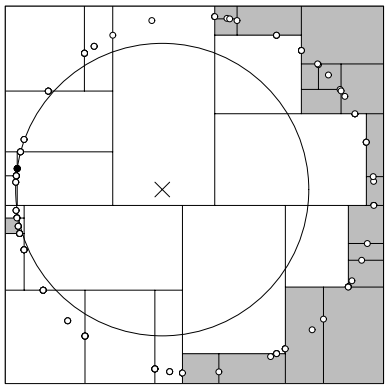

Auton's Graphics


Auton's Graphics



A worst-case for NN-Search using kd-Trees



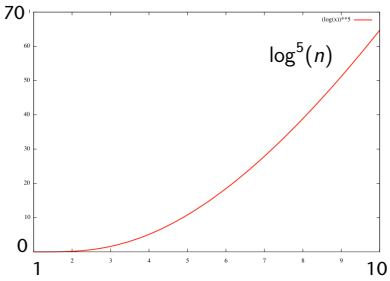


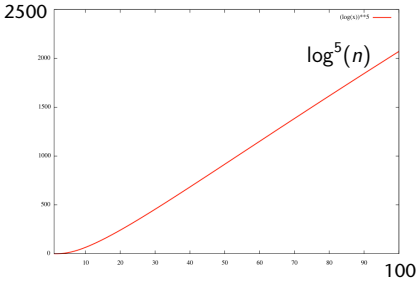
Gutartiger Fall

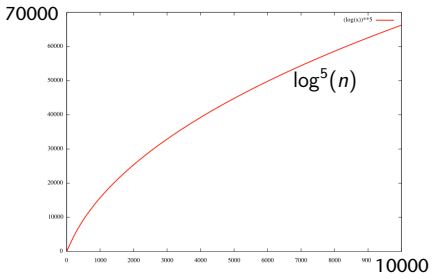
Bösartiger Fall

Alle weißen Blätter muß der NN-Algorithmus besuchen!

Zum Verhalten von $\log^d(n)$

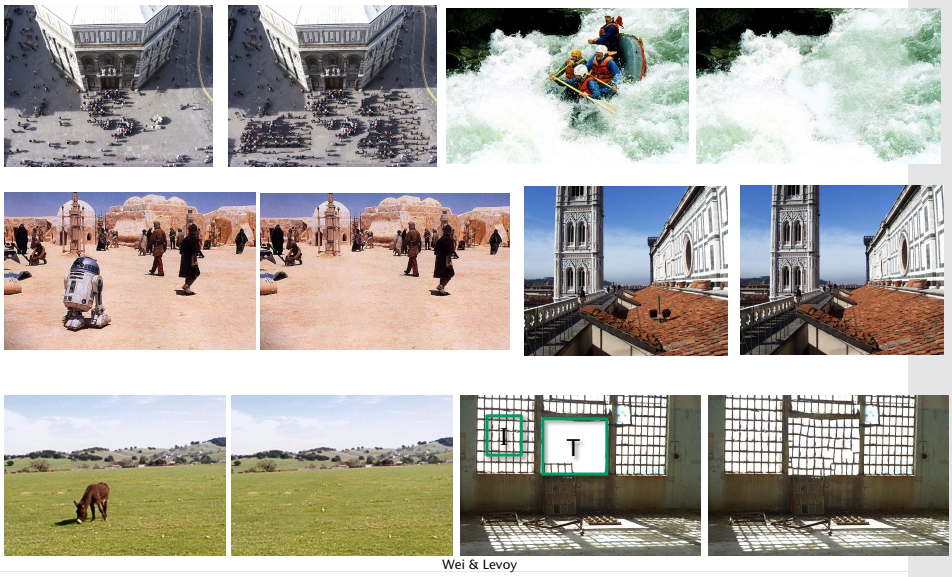




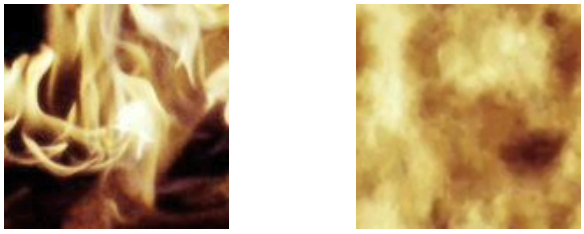


Der Algorithmus für die ANN-Suche ist also besser (asymptotisch) als brute-force-mäßig alle n Punkte zu besuchen und deren Abstand zum Query-Punkt q zu berechnen.

Texture Synthesis

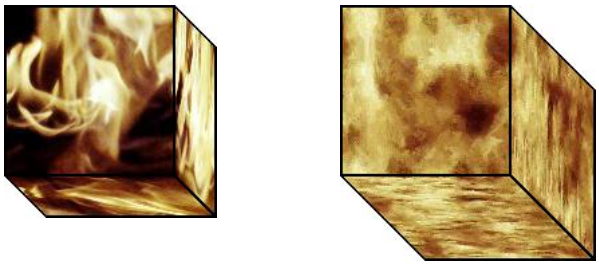


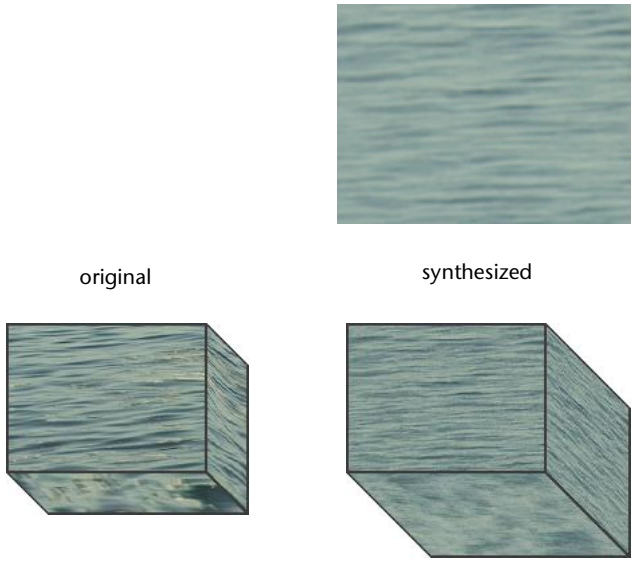
Wei & Levoy



original

synthesized





Stackless kd-tree traversal for ray-tracing

Stefan Popov, Johannes Günther, Hans-Peter Seidel, and Philipp Slusallek.
Nvidia GeForce 8800GTX, CUDA, 2007.

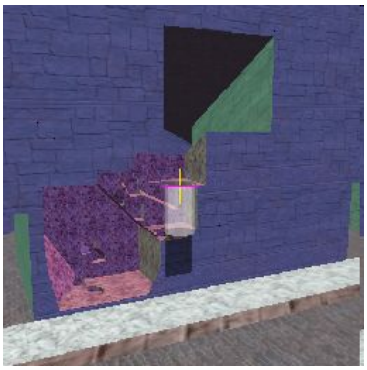
Daniel Horn, Jeremy Sugerman, Mike Houston, Pat Hanrahan
ATI X1900XTX, PixelShader 3.0, 2007

Real-Time KD-Tree Construction on Graphics Hardware

Kun Zhou, Qiming Hou, Rui Wang, Baining Guo; SIGGRAPH Asia 2008

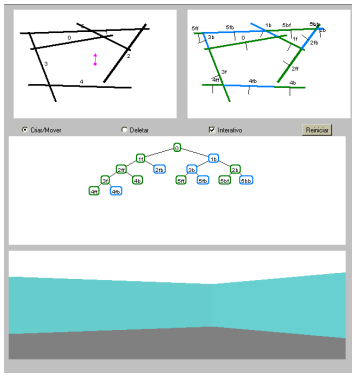
Applications of the BSP

Boolen Operations



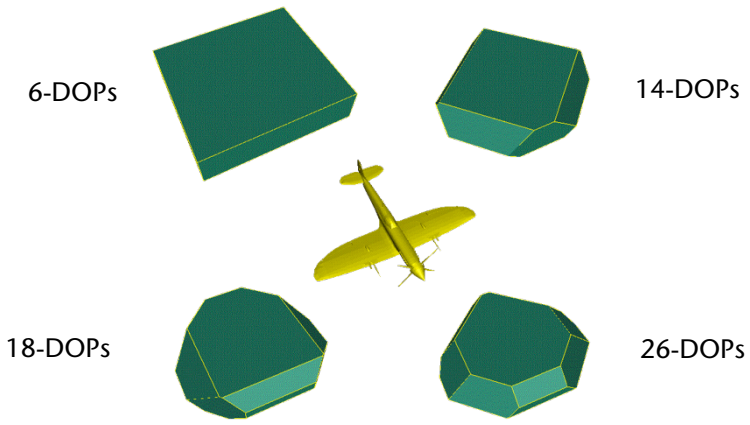
Stan Melax

Painter's Algorithm

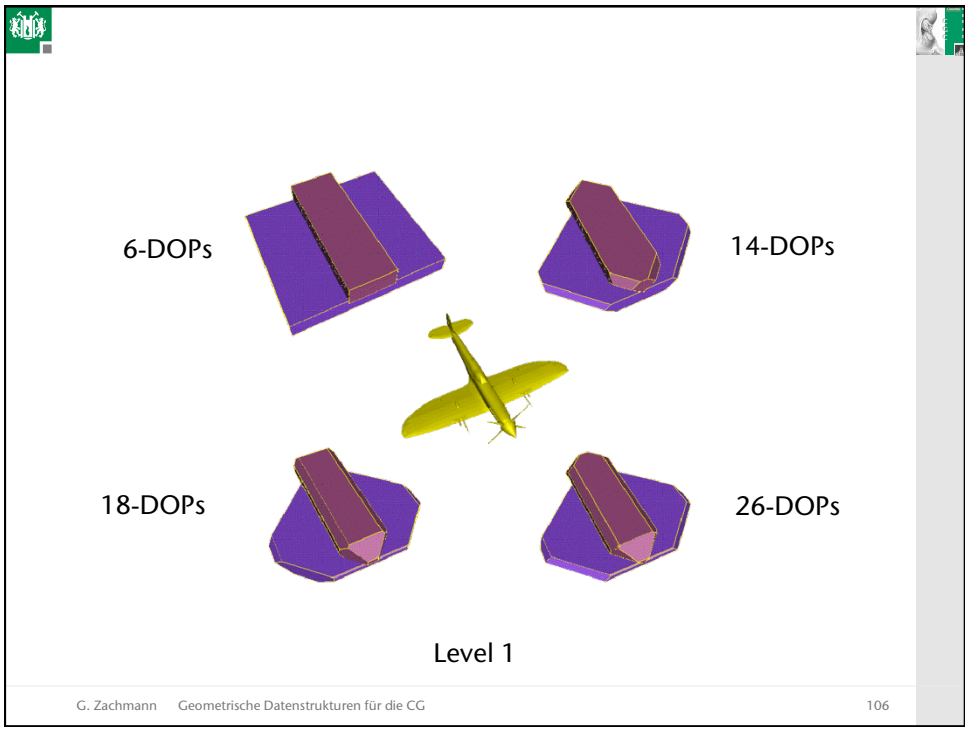


Paton J. Lewis

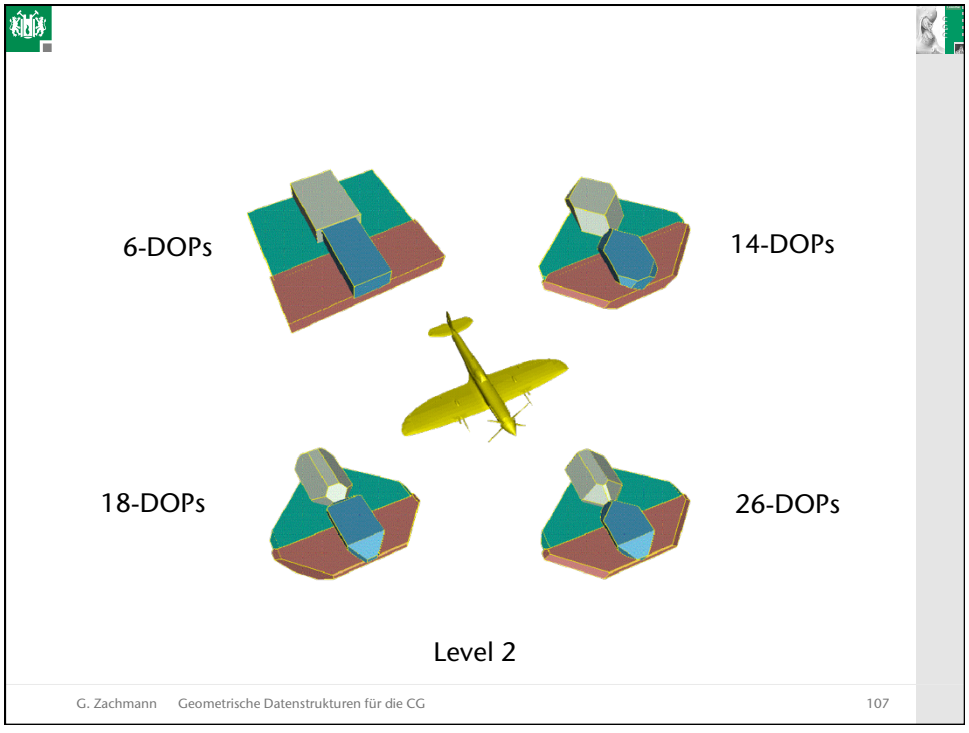
BVH mit *k*-DOPs



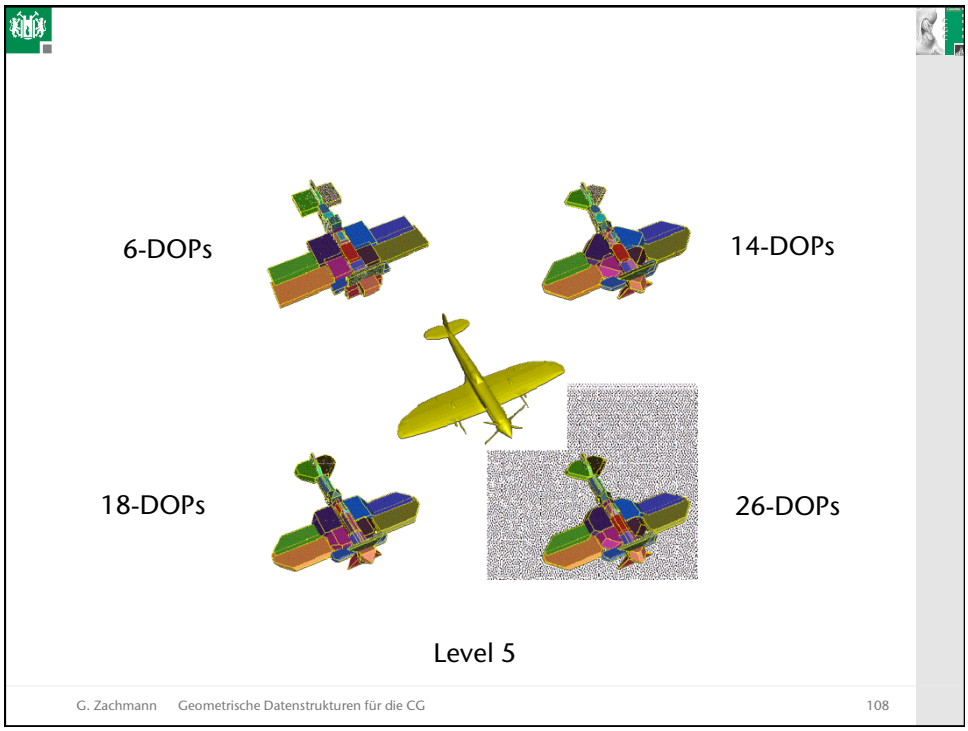
Level 0



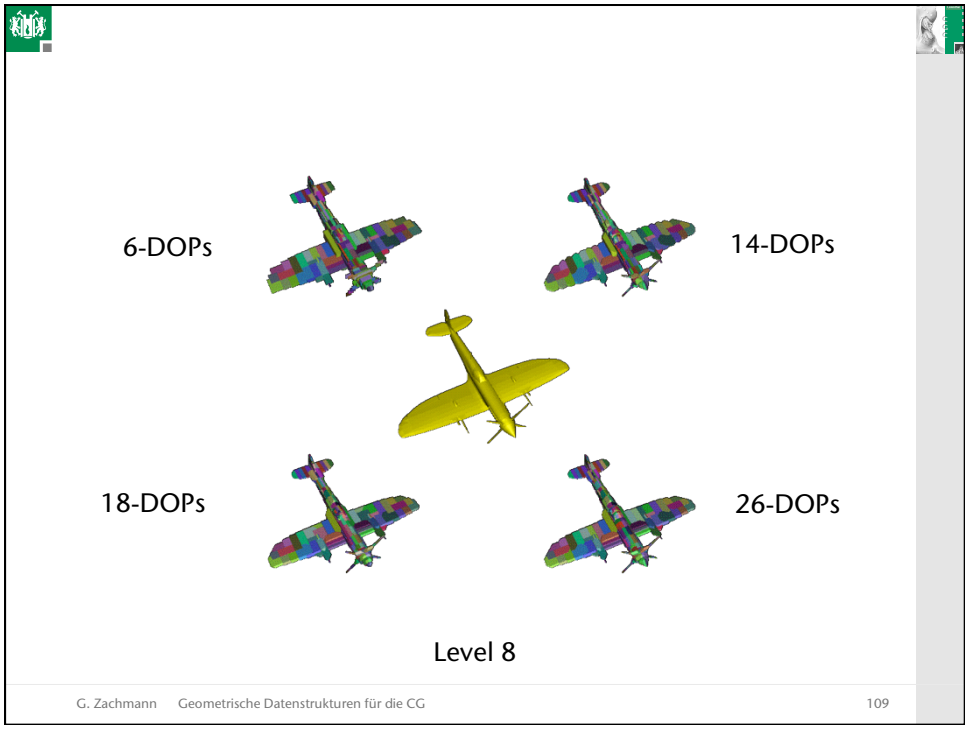
Level 1



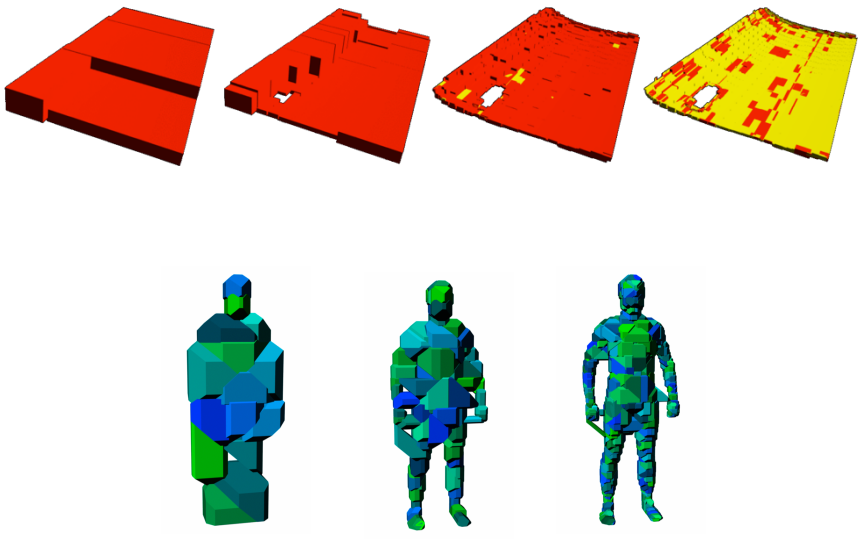
Level 2



Level 5

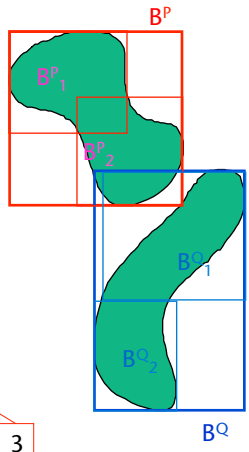


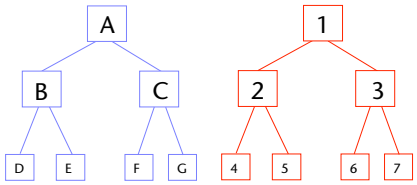
Level 8



Hierarchische Kollisionserkennung mittels BVHs

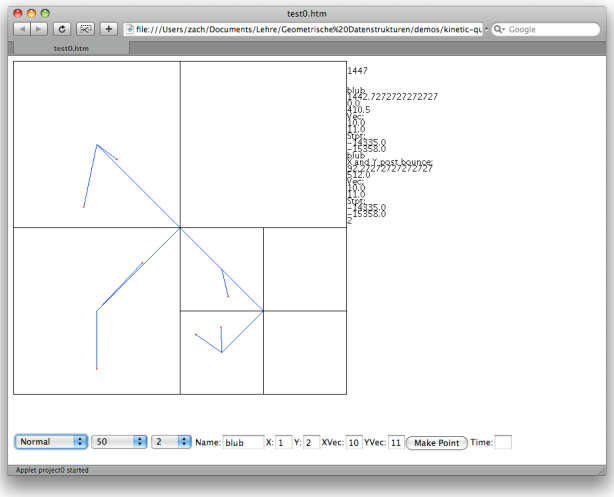
traverse(X, Y)
if X,Y do not overlap **then**
  return
if X,Y are leaves **then**
  check polygons
else
  for all children pairs **do**
  traverse(X_i, Y_j)



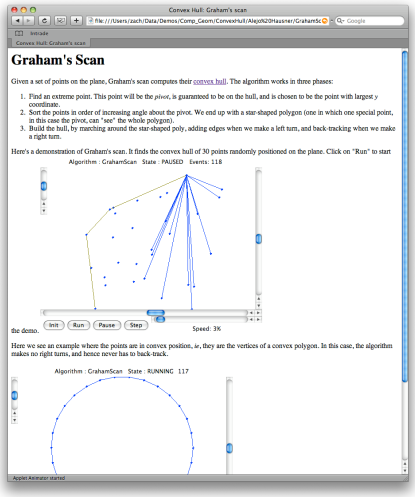


Applications using Distance Fields

Kinetic Quadtree Demo



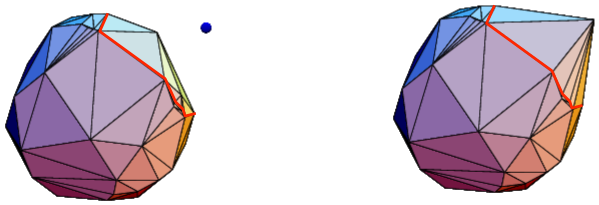
Convex Hull Demos in 2D

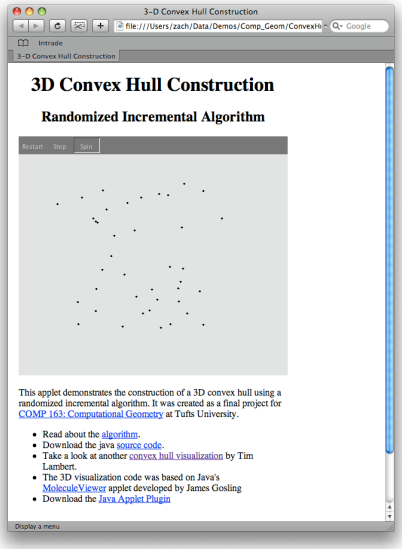


Alejo Hausner - http://www.cs.princeton.edu/~ah/alg_anim/version1/GrahamScan.html

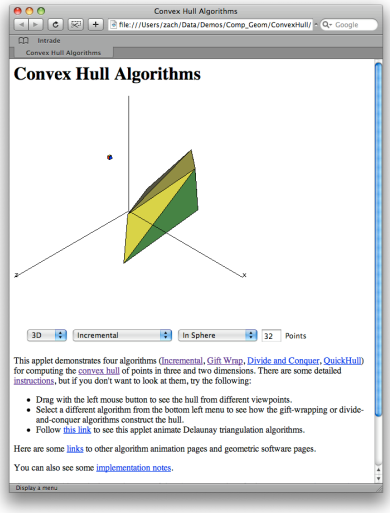
Convex Hull in 3D

- Ein Schritt des inkrementellen Algorithmus':



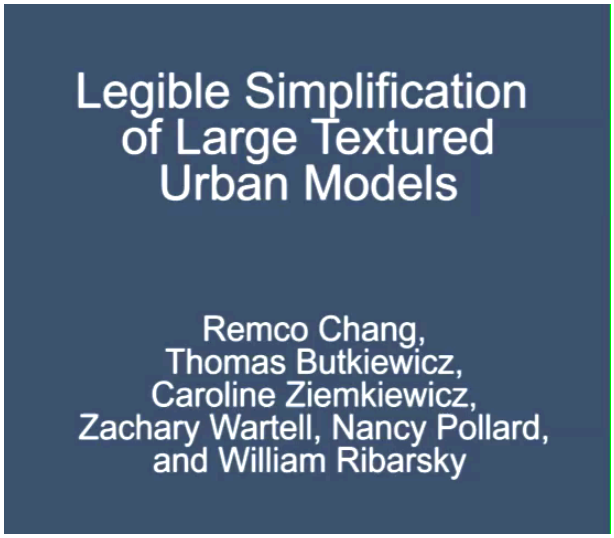


Michael Horn - http://www.eecs.tufts.edu/~mhorn01/comp163/



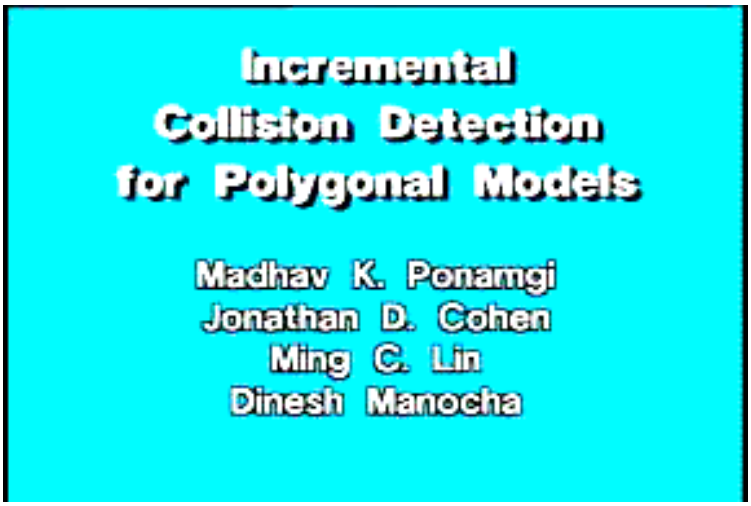
Tim Lambert - http://www.cse.unsw.edu.au/~lambert/java/3d/hull.html

Simplification of Urban Models



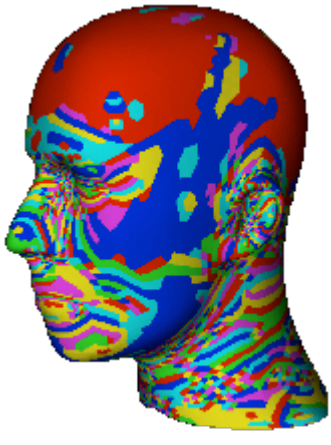
Remco Chang, Thomas Butkiewicz, Caroline Ziemkiewicz, Zachary Wartell, Nancy Pollard, William Ribarsky

Convex Collision Detection



Achtung: der hier demonstrierte Algo ist in Wahrheit
etwas komplexer als der in der Vorlesung dargestellte!
(aber möglicherweise nicht schneller ...)

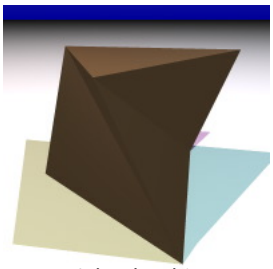
Convex Surface Decomposition



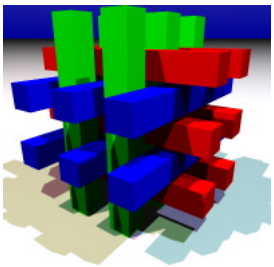
Zerlegung in
konvexe Surface-Patches

Konvexe Stücke auf einem
mittleren Level der Hierarchie
(grün = orig. Fläche, rot = freie Fläche,
gelb = "contained")

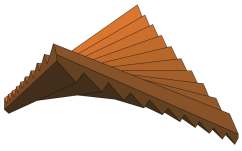
Untetrahedralizable Objects



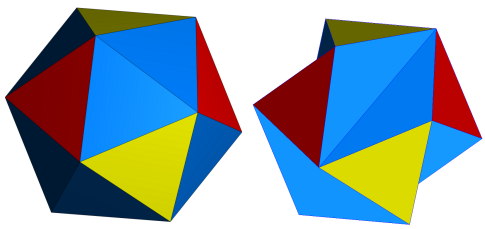
Schönhardt's
Polyeder
(1928)

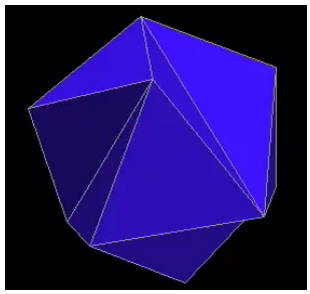


Thurston-
Polyeder
(1971)



Chazelle's
Polyeder
(1984)





Jessen's Ikosaeder