



Geometrische Datenstrukturen für die Computer-Graphik

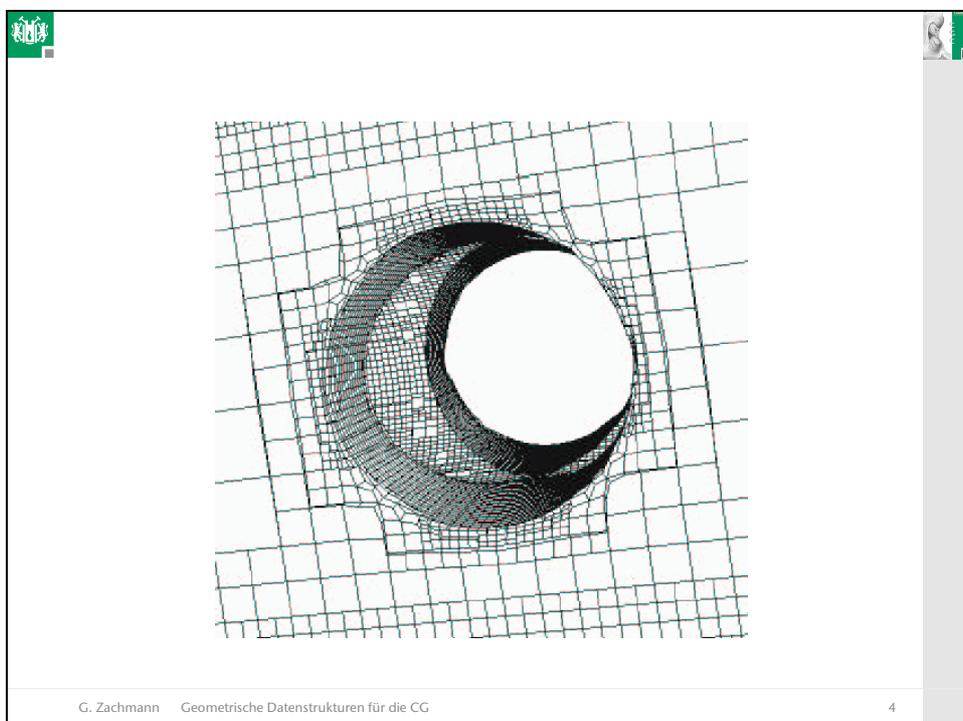
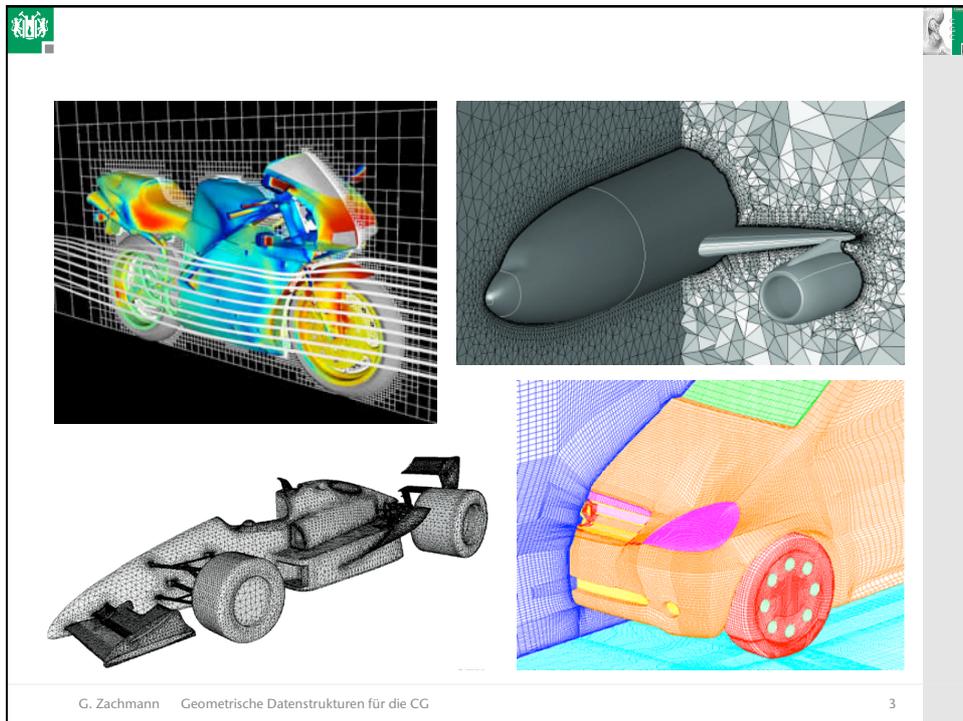
G. Zachmann
zach@tu-clausthal.de
Clausthal University, Germany



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)

G. Zachmann Geometrische Datenstrukturen für die CG 2



Quadtree demo

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

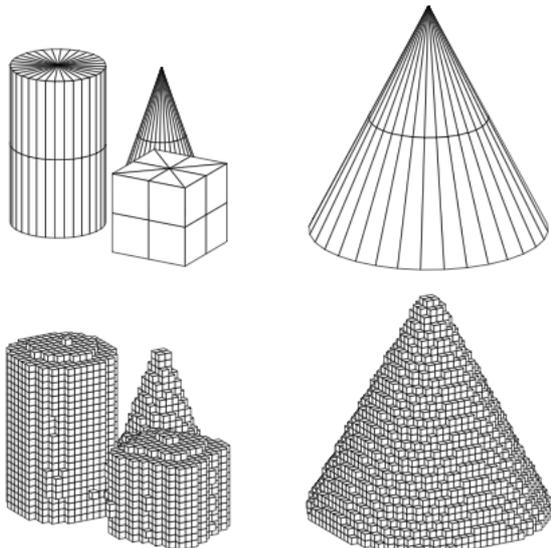
G. Zachmann Geometrische Datenstrukturen für die CG
5

PR Quadtree Demo

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

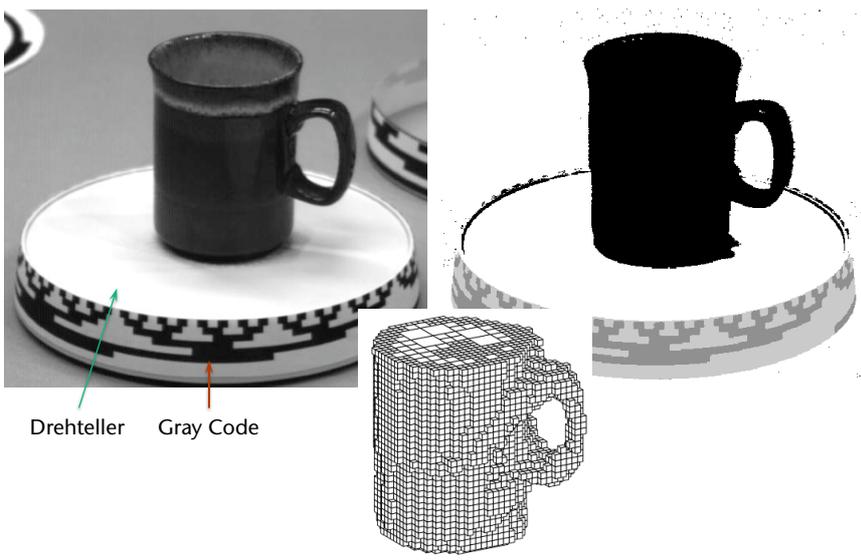
G. Zachmann Geometrische Datenstrukturen für die CG
6

Octree Models from Images



G. Zachmann Geometrische Datenstrukturen für die CG 7

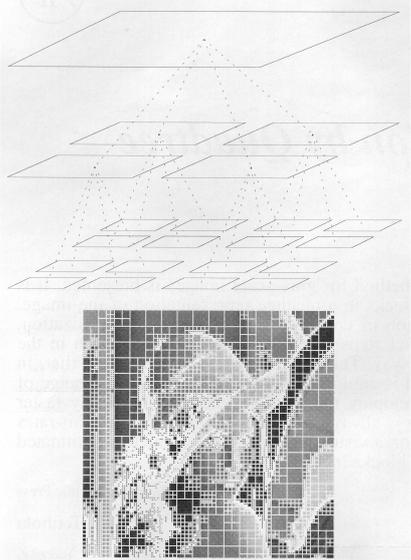
Example Image



Drehteller Gray Code

G. Zachmann Geometrische Datenstrukturen für die CG 8

Image Compression using Quadtrees



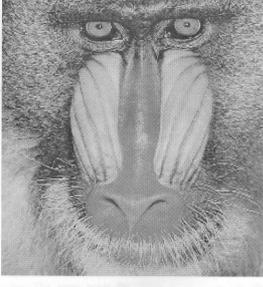
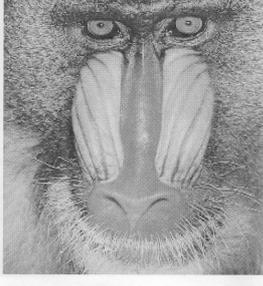
G. Zachmann Geometrische Datenstrukturen für die CG 9

Die beiden Test-Bilder schlechthin



G. Zachmann Geometrische Datenstrukturen für die CG 10

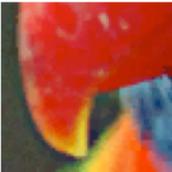
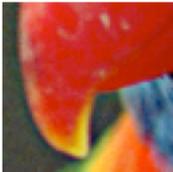
Resultate

<p>QP: 1.03 bits per pixel</p> 	<p>QP: 1.95 bits per pixel</p> 
<p>JPEG: 1.00 bits per pixel</p> 	<p>JPEG: 1.99 bits per pixel</p> 

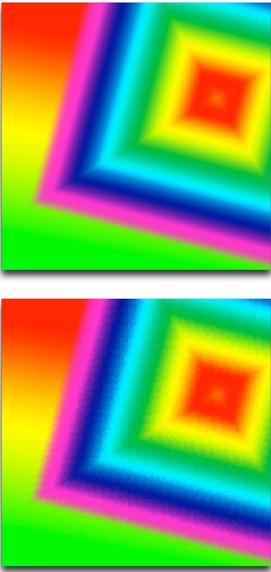
G. Zachmann Geometrische Datenstrukturen für die CG 11

S3TC texture compression

▪ Vergleich:

<p>DXT1</p>  	<p>Uncompressed</p>  
--	--

Philipp Klaus Krause



Simon Brown

G. Zachmann Geometrische Datenstrukturen für die CG 12

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



uncompressed



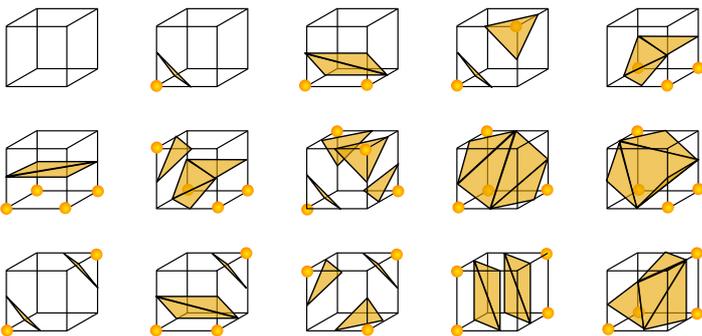
mit S3TC

Unreal Retexturing Project

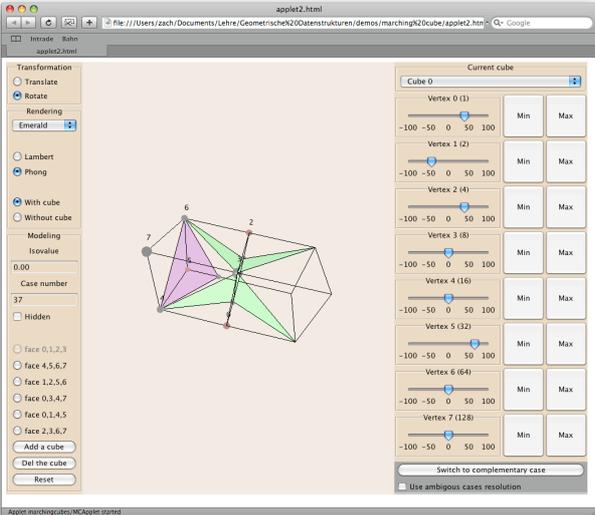
G. Zachmann Geometrische Datenstrukturen für die CG 13

Isosurfaces

- Die 15 echt verschiedenen Fälle in 3D:



G. Zachmann Geometrische Datenstrukturen für die CG 19



Transformation

- Translate
- Rotate

Rendering

Emerald

- Lambert
- Phong

Modeling

- With cube
- Without cube

0.00

Case number

37

Hidden

- face 0.1.2.3
- face 4.5.6.7
- face 1.2.5.6
- face 0.1.4.7
- face 0.1.4.5
- face 2.3.6.7

Add a cube
Del the cube
Reset

Current cube

Cube 0

Vertex 0 (1)

-100 -50 0 50 100

Min Max

Vertex 1 (2)

-100 -50 0 50 100

Min Max

Vertex 2 (4)

-100 -50 0 50 100

Min Max

Vertex 3 (8)

-100 -50 0 50 100

Min Max

Vertex 4 (16)

-100 -50 0 50 100

Min Max

Vertex 5 (32)

-100 -50 0 50 100

Min Max

Vertex 6 (64)

-100 -50 0 50 100

Min Max

Vertex 7 (128)

-100 -50 0 50 100

Min Max

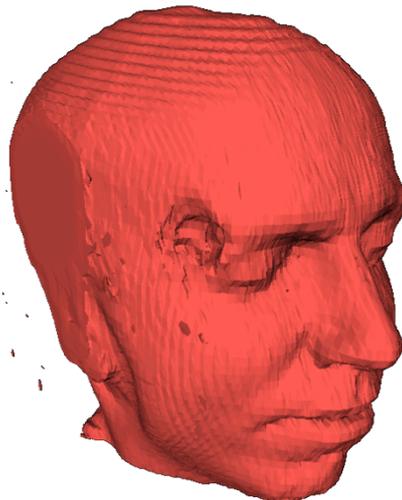
Switch to complementary case

Use ambiguous cases resolution

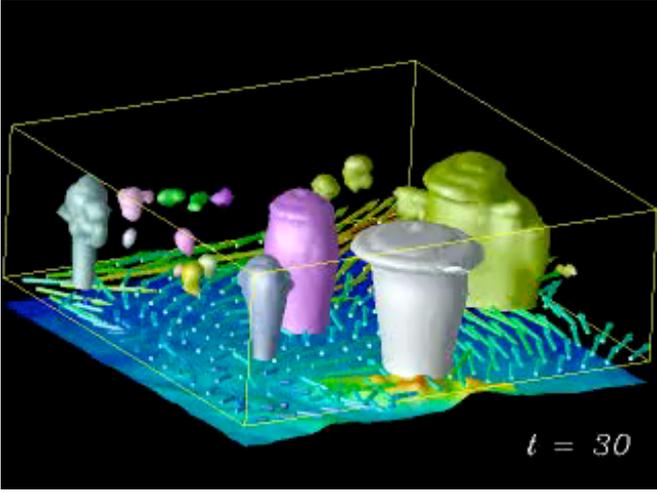
<http://users.polytech.unice.fr/~lingrand/MarchingCubes/accueil.html>

G. Zachmann Geometrische Datenstrukturen für die CG 20

- Output eines einfachen Marching-Cube-Algorithmus!:



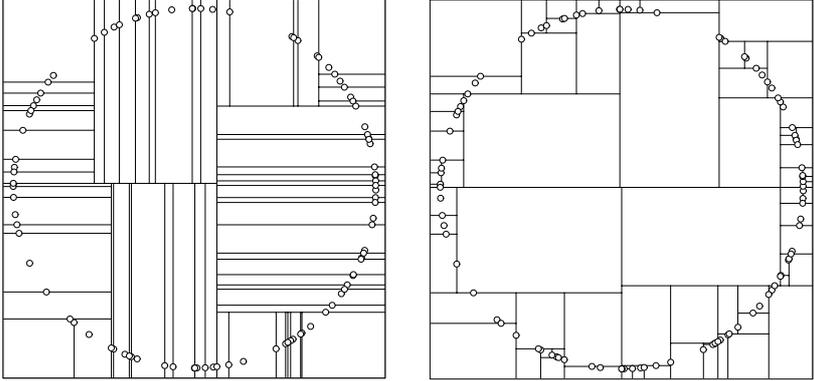
Beispiel aus einer Wetter-Simulation



$t = 30$

G. Zachmann Geometrische Datenstrukturen für die CG 22

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

G. Zachmann Geometrische Datenstrukturen für die CG 23

Nearest Neighbor Search using kd-Trees

Gutartiger Fall **Bösartiger Fall**

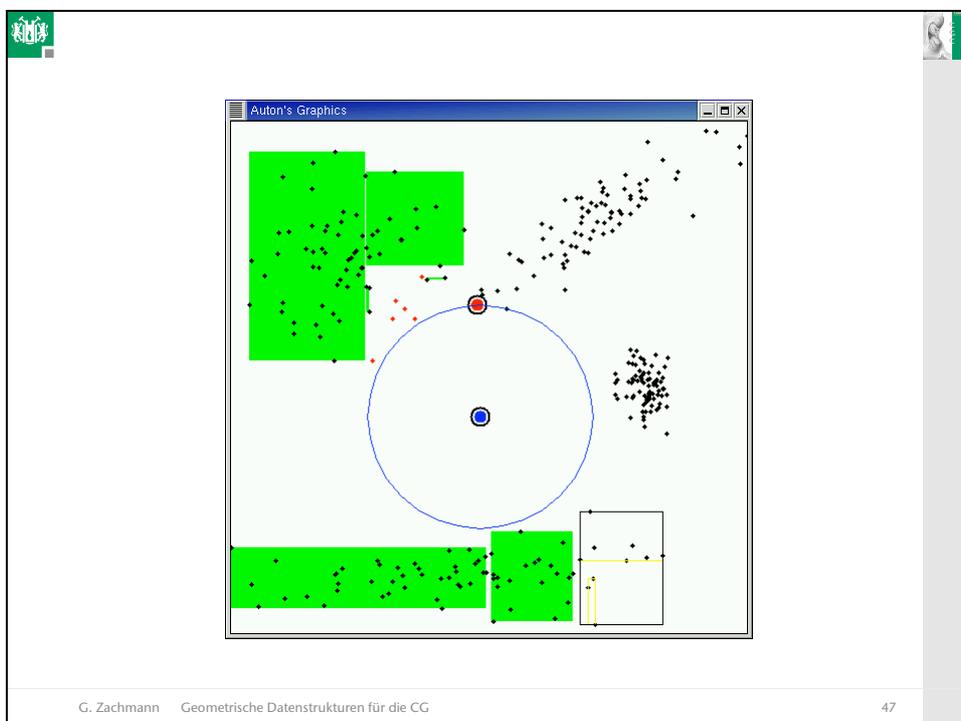
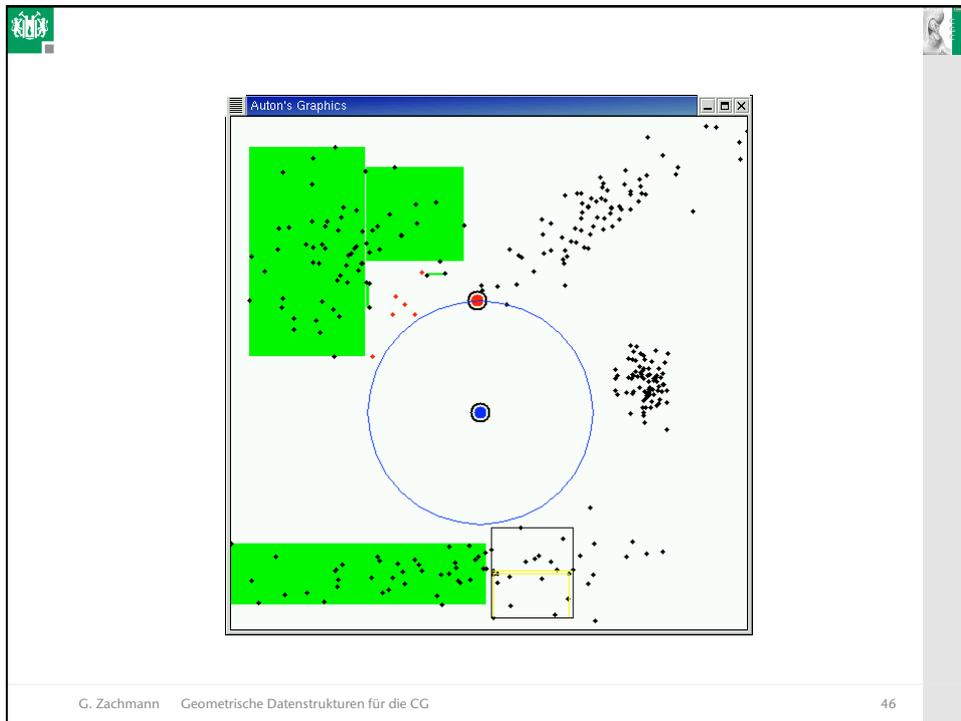
Alle weißen Blätter muß der NN-Algorithmus besuchen!

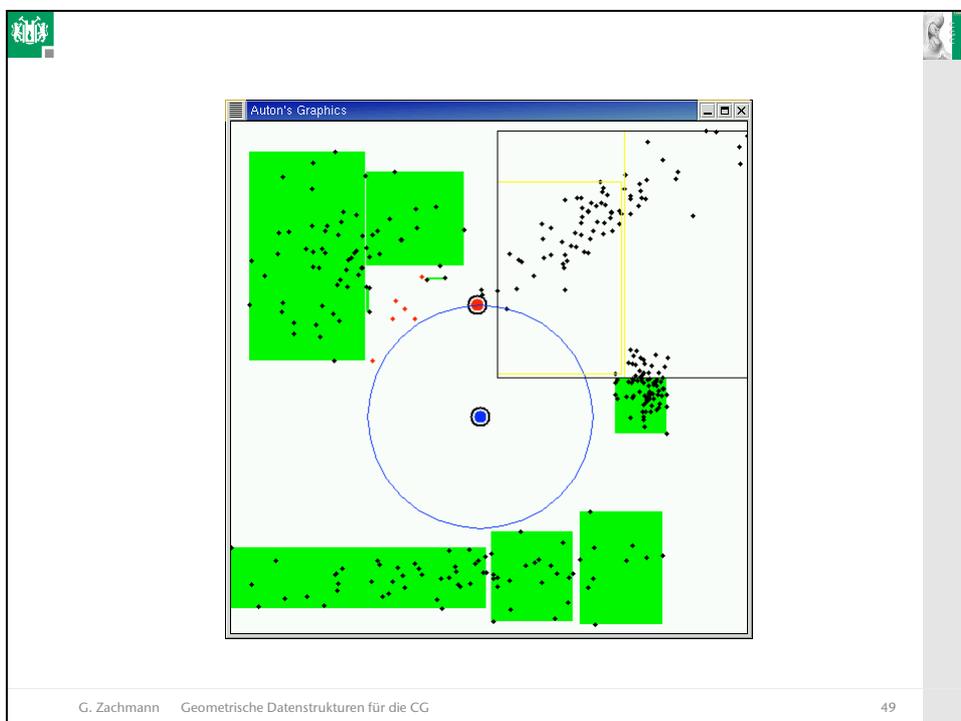
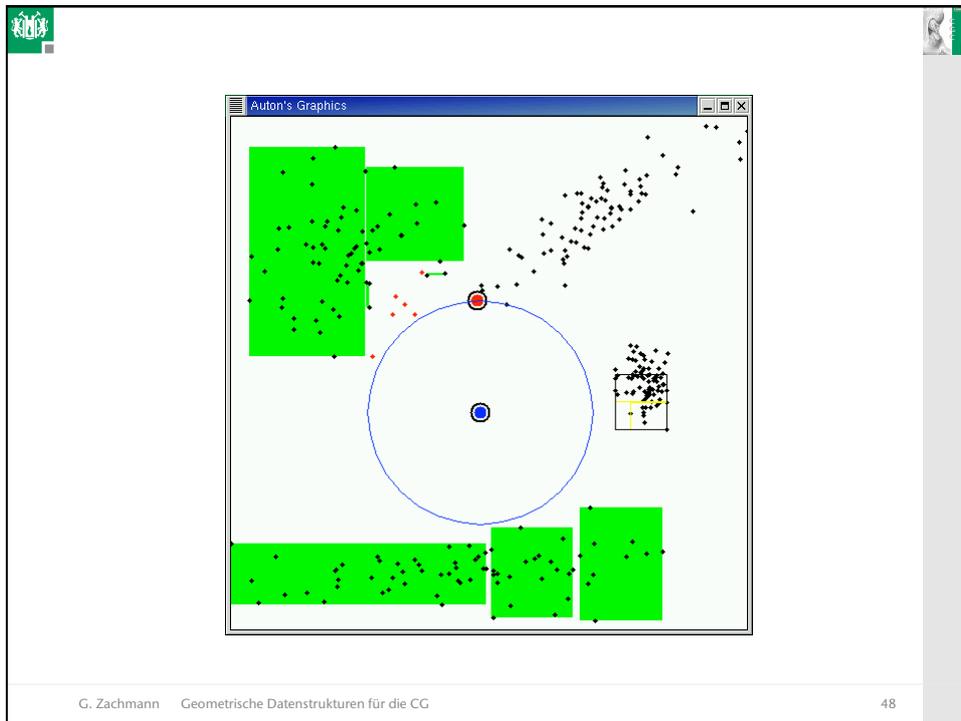
G. Zachmann Geometrische Datenstrukturen für die CG 24

Animation: NN search

Andrew Moore, CMU

G. Zachmann Geometrische Datenstrukturen für die CG 25





Wei & Levoy

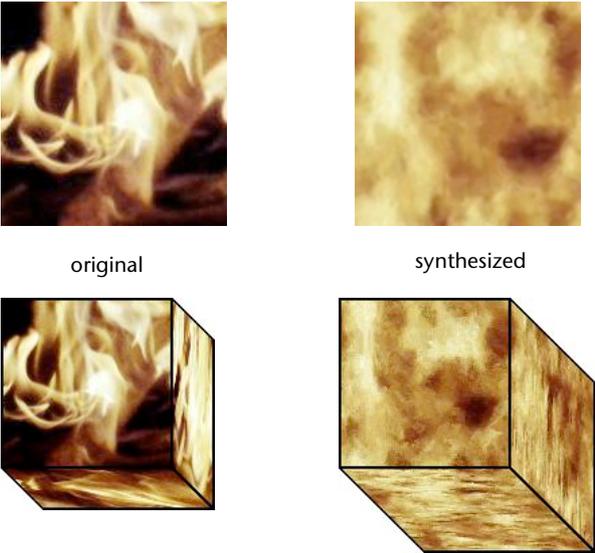


Aspen Trees Harris and Love, Inc.

Aspen Trees Harris and Love, Inc.

G. Zachmann Geometrische Datenstrukturen für die CG 92

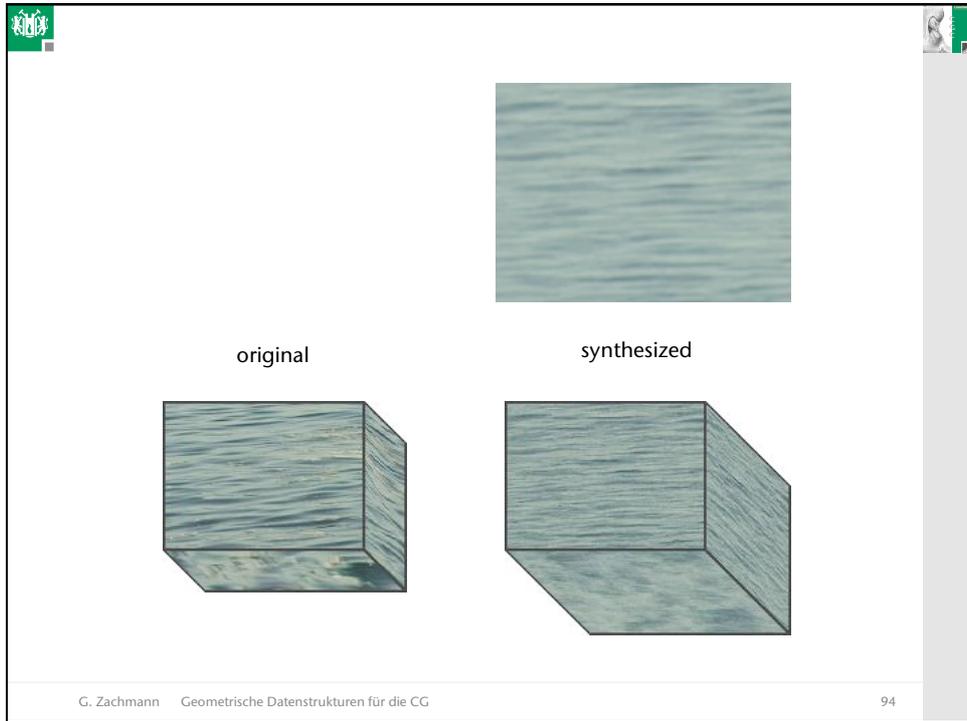
Detailed description: This slide shows a comparison between original and synthesized images. On the left, there are two identical images of a dense forest of yellow aspen trees. On the right, there are two identical images of a group of seven people standing in a snowy landscape. The text 'Wei & Levoy' is written vertically on the left side. At the bottom, the text 'G. Zachmann Geometrische Datenstrukturen für die CG' and the number '92' are present.

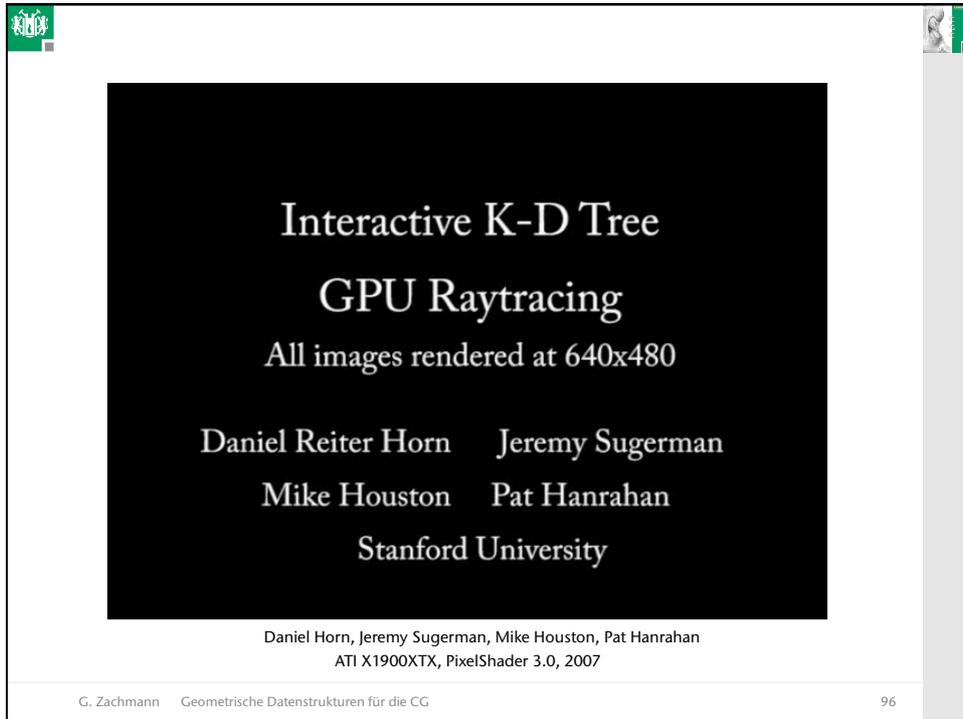


original synthesized

G. Zachmann Geometrische Datenstrukturen für die CG 93

Detailed description: This slide compares original and synthesized images of a flame. The top row shows two 2D images: 'original' on the left and 'synthesized' on the right. The 'original' image shows a clear, detailed flame with distinct structures, while the 'synthesized' image is a blurred, noisy version of the same scene. The bottom row shows two 3D renderings of a cube, each with the corresponding 2D image mapped onto its faces. The 'original' 3D rendering shows clear, detailed flame patterns on the cube's surfaces, while the 'synthesized' 3D rendering shows a noisy, less detailed pattern.



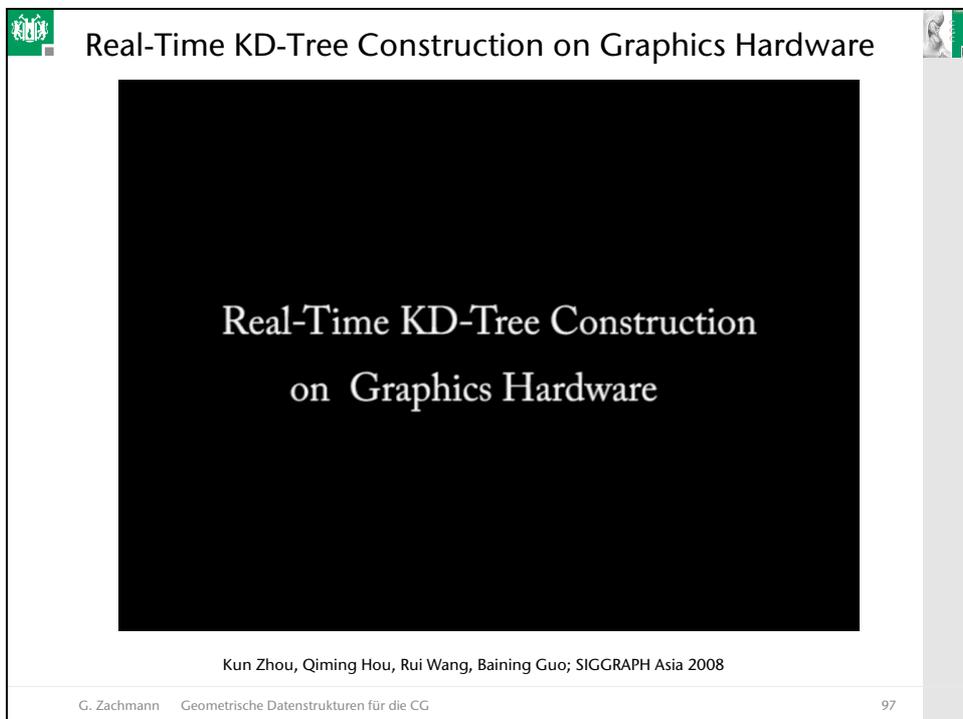


Interactive K-D Tree
GPU Raytracing
All images rendered at 640x480

Daniel Reiter Horn Jeremy Sugerman
Mike Houston Pat Hanrahan
Stanford University

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

G. Zachmann Geometrische Datenstrukturen für die CG 96



Real-Time KD-Tree Construction on Graphics Hardware

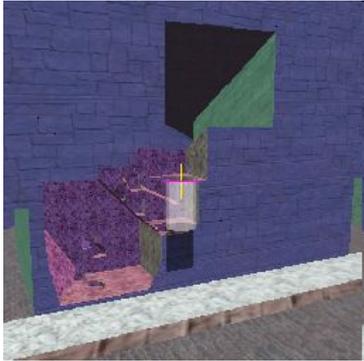
Real-Time KD-Tree Construction
on Graphics Hardware

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

G. Zachmann Geometrische Datenstrukturen für die CG 97

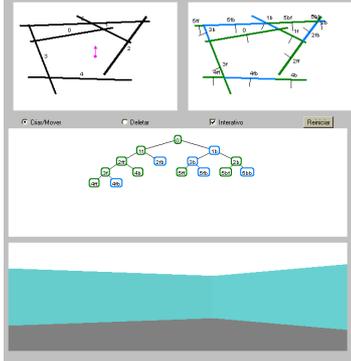
Applications of the BSP

Boolean Operations



Stan Melax

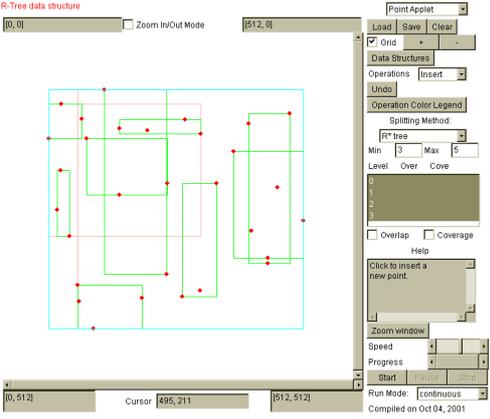
Painter's Algorithm



Paton J. Lewis

G. Zachmann Geometrische Datenstrukturen für die CG
98

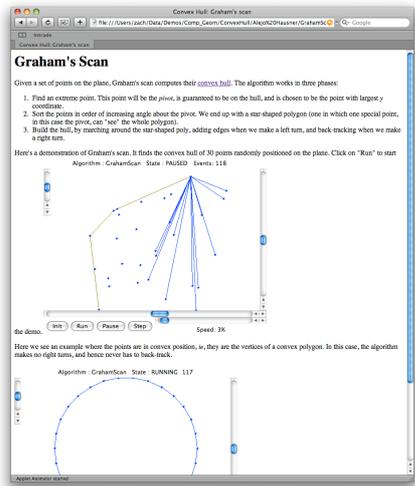
BVH demo



Interactive hierarchy construction

G. Zachmann Geometrische Datenstrukturen für die CG
99

Convex Hull Demos in 2D



Graham's Scan

Given a set of points on the plane, Graham's scan computes their [convex hull](#). The algorithm works in three phases:

1. Find an extreme point. This point will be the *pivot*, is guaranteed to be on the hull, and is chosen to be the point with largest *y* coordinate.
2. Sort the points in order of increasing angle about the pivot. We end up with a star-shaped polygon (one in which one special point, in this case the pivot, can "see" the whole polygon).
3. Build the hull, by walking around the star-shaped poly, adding edges when we make a left turn, and back-tracking when we make a right turn.

Here's a demonstration of Graham's scan. It finds the convex hull of 30 points randomly positioned on the plane. Click on "Run" to start the demo.

Algorithm: GrahamScan State: PAUSED Iterations: 118

Speed: 3K

Here we see an example where the points are in convex position, i.e., they are the vertices of a convex polygon. In this case, the algorithm makes no right turns, and hence never has to back-track.

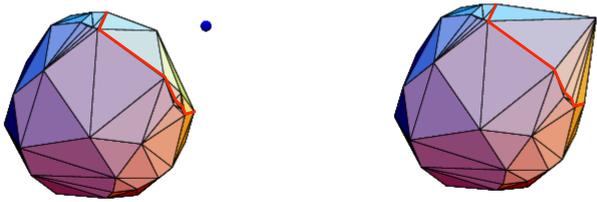
Algorithm: GrahamScan State: RUNNING Iterations: 117

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

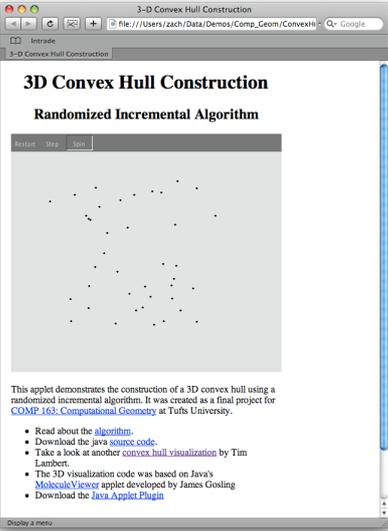
G. Zachmann Geometrische Datenstrukturen für die CG 102

Convex Hull in 3D

- Ein Schritt des inkrementellen Algorithmus':



G. Zachmann Geometrische Datenstrukturen für die CG 103



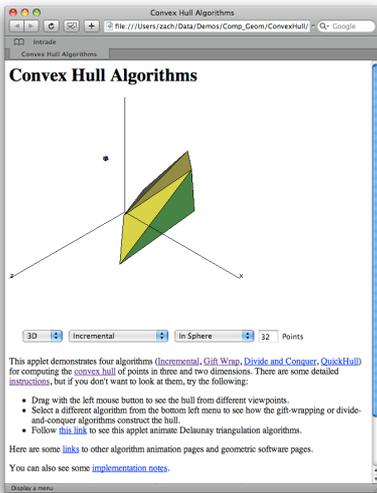
3D Convex Hull Construction
Randomized Incremental Algorithm

This applet demonstrates the construction of a 3D convex hull using a randomized incremental algorithm. It was created as a final project for [COMP 163: Computational Geometry](#) at Tufts University.

- Read about the [algorithm](#).
- Download the [java source code](#).
- Take a look at another [convex hull visualization](#) by Tim Lambert.
- The 3D visualization code was based on Java's [Molensk's Java](#) applet developed by James Gosling
- Download the [Java Applet Plugin](#)

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

G. Zachmann Geometrische Datenstrukturen für die CG 104



Convex Hull Algorithms

3D Incremental In Sphere 32 Points

This applet demonstrates four algorithms ([Incremental](#), [Gift Wrapping](#), [Divide and Conquer](#), [QuickHull](#)) for computing the [convex hull](#) of points in three and two dimensions. There are some detailed [instructions](#), but if you don't want to look at them, try the following:

- Drag with the left mouse button to see the hull from different viewpoints.
- Select a different algorithm from the bottom left menu to see how the gift-wrapping or divide-and-conquer algorithms construct the hull.
- Follow [this link](#) to see this applet animate Delaunay triangulation algorithms.

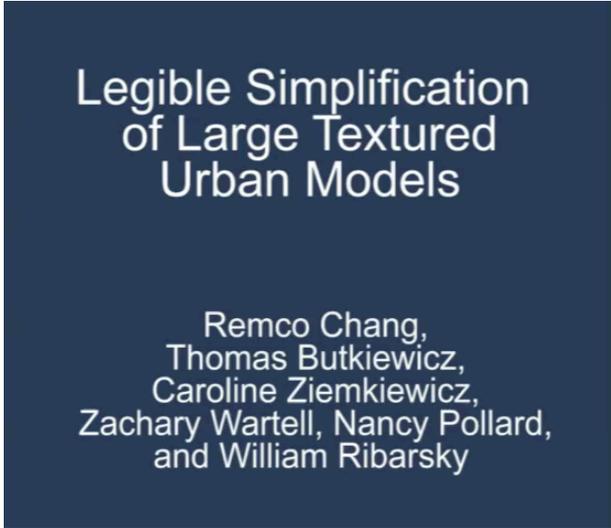
Here are some [links](#) to other algorithm animation pages and geometric software pages.

You can also see some [implementation notes](#).

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

G. Zachmann Geometrische Datenstrukturen für die CG 105

Simplification of Urban Models



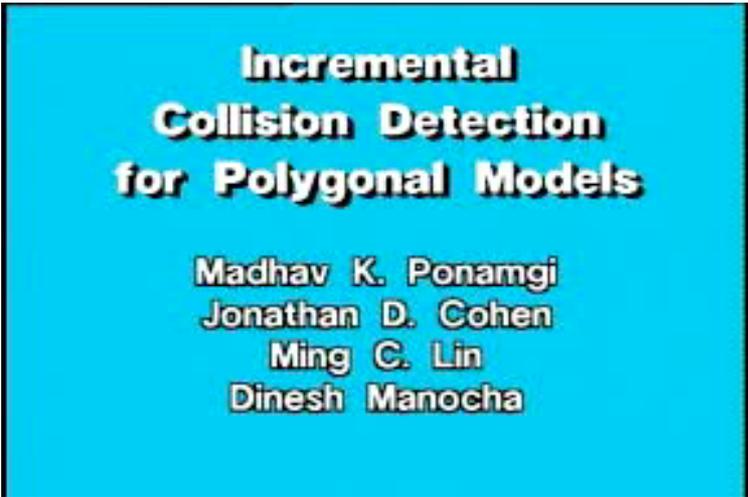
**Legible Simplification
of Large Textured
Urban Models**

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

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

G. Zachmann Geometrische Datenstrukturen für die CG 106

Convex Collision Detection



**Incremental
Collision Detection
for Polygonal Models**

Madhav K. Ponamgi
Jonathan D. Cohen
Ming C. Lin
Dinesh Manocha

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

G. Zachmann Geometrische Datenstrukturen für die CG 107

Convex Surface Decomposition



Zerlegung in
konvexe Surface-Patches



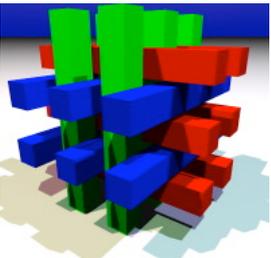
Konvexe Stücke auf einem
mittleren Level der Hierarchie
(grün = orig. Fläche, rot = freie Fläche,
gelb = "contained")

G. Zachmann Geometrische Datenstrukturen für die CG
108

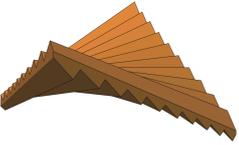
Untetrahedralizable Objects



Schönhardt's
Polyeder
(1928)

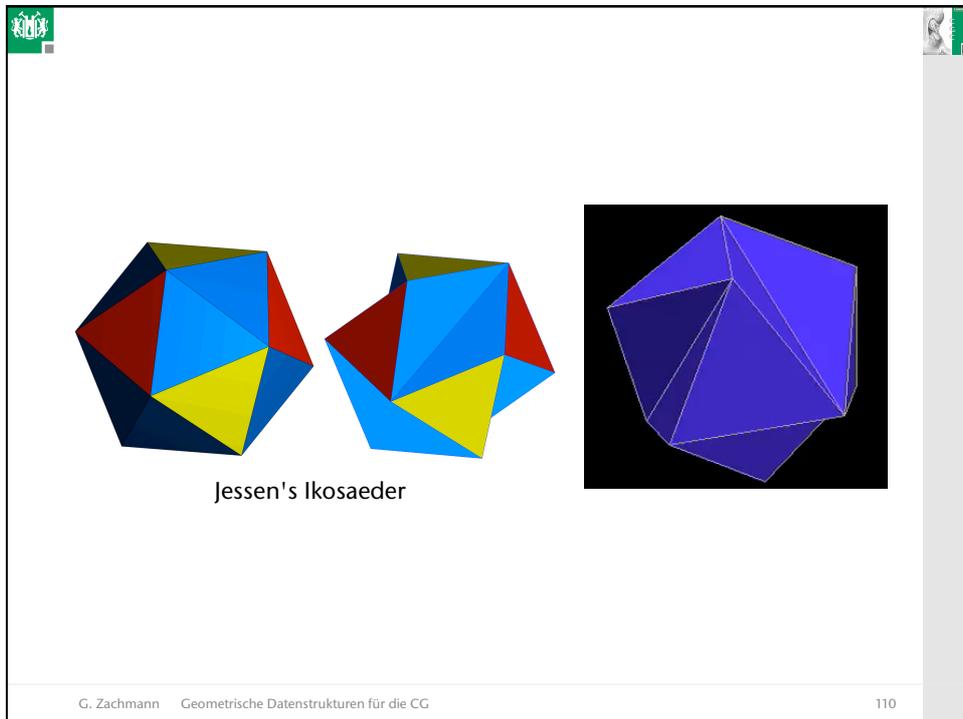


Thurston-
Polyeder
(1971)



Chazelle's
Polyeder
(1984)

G. Zachmann Geometrische Datenstrukturen für die CG
109



Voronoi-Diagramme

- Eine der ersten Erwähnungen von René Descartes (Cartesius; 1596-1650) in seiner *Principia Philosophiae*, 1644:
 - Stellte sich vor, daß das Universum mit Materie gefüllt ist, die von den Sternen angezogen wird und um diese herumwirbelt
- Georgy F. Voronoy (Георгий Ф. Вороной) 1868 – 1908
 - Geboren in Russland, heutige Ukraine
 - Professor in Warschau
 - Schüler: Delaunay




G. Zachmann Geometrische Datenstrukturen für die CG 111

Delaunay (1890 – 1980)

- Schüler von Voronoy (und Grave)
- Einer der 3 besten russischen Bergsteiger um 1930
- Russische Schreibweise: Борис Николаевич Делоне
 - Damals war Französisch (und Deutsch) die Wissenschaftssprache!



G. Zachmann Geometrische Datenstrukturen für die CG 112

- Nicht zu verwechseln mit dem Maler Robert Delaunay !
 - 1885 – 1941 ; wirklich französisch

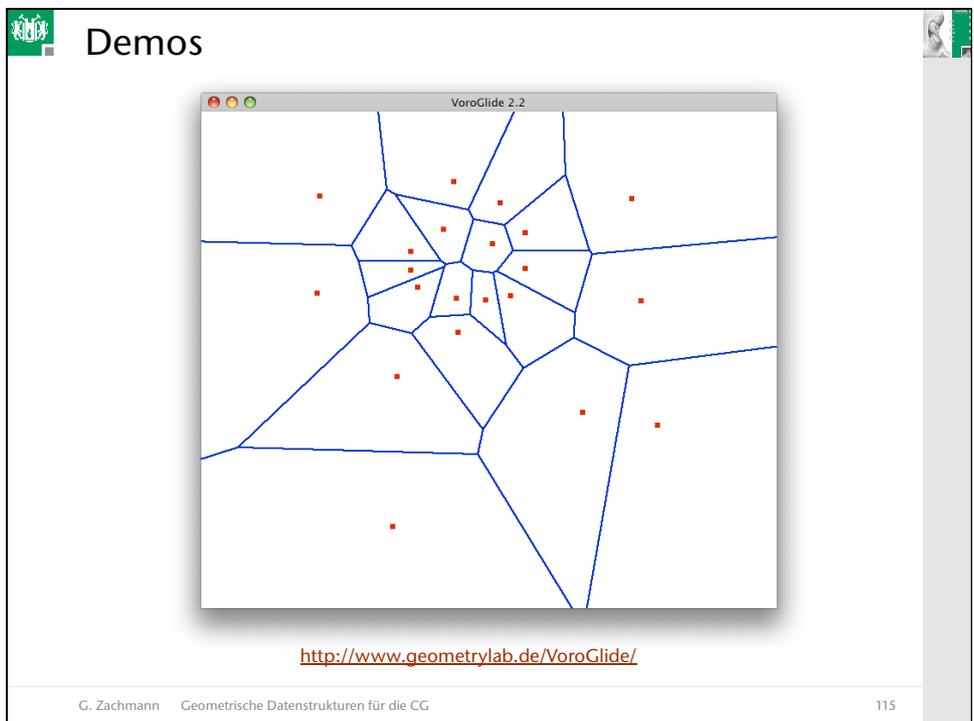
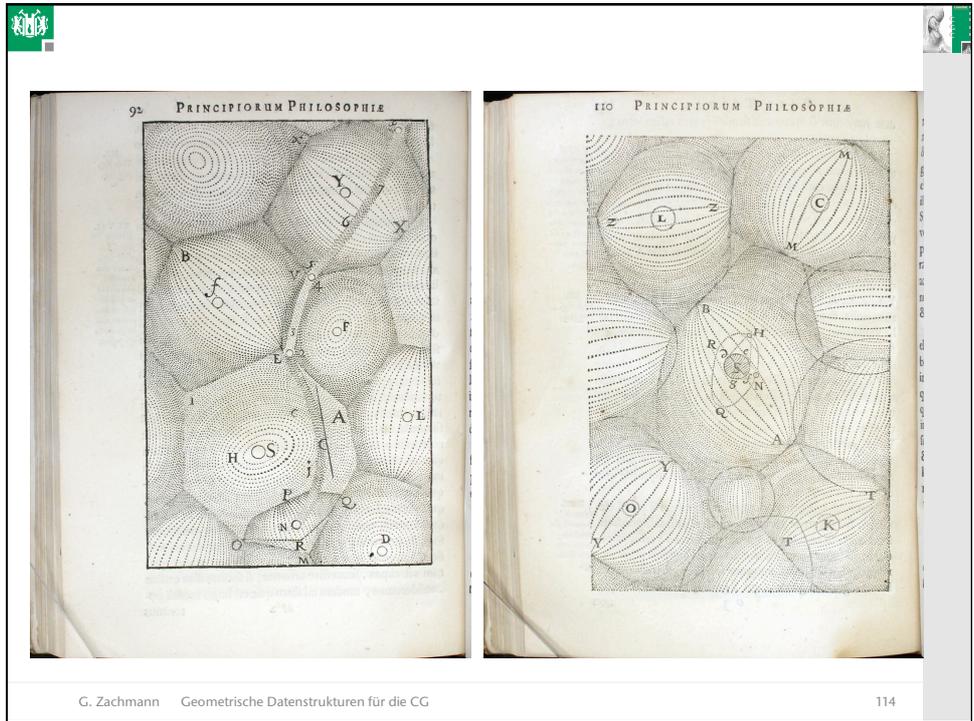


Champs de Mars. La Tour rouge. 1911

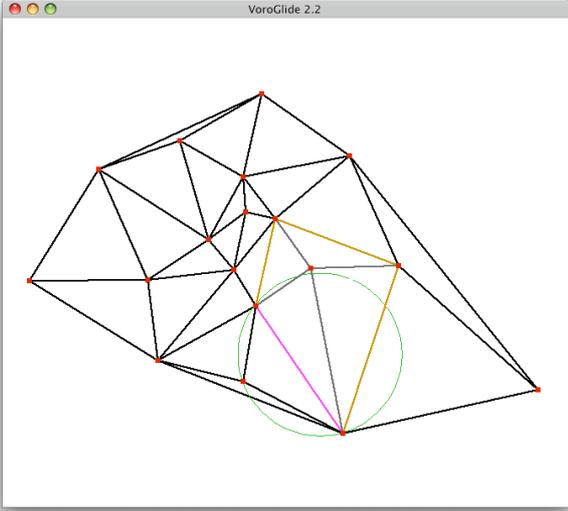


Homage à Bleriot, 1914

G. Zachmann Geometrische Datenstrukturen für die CG 113



Inkrementelle Konstruktion der Delaunay-Triangulierung



G. Zachmann Geometrische Datenstrukturen für die CG 116

Verallgemeinerung des Voronoi-Diagramms

- Andere Distanz-Funktionen
- Andere Objekte als Sites
- Höhere Dimension
- Andere Äquivalenzklassen
- ...

G. Zachmann Geometrische Datenstrukturen für die CG 117

Voronoi / Delaunay in 3D

- Delaunay-Tetraeder
- Bisektoren = Ebenen
- Edge-Flip:

G. Zachmann Geometrische Datenstrukturen für die CG 118

Slivers in 3D Delaunay Tetrahedralizations:

Delaunay Sliver Non-Delaunay

Diese beiden Ecken liegen etwas tiefer

- Fazit: die max-min-Winkel-Eigenschaft gilt nur in 2D! ☹

G. Zachmann Geometrische Datenstrukturen für die CG 119

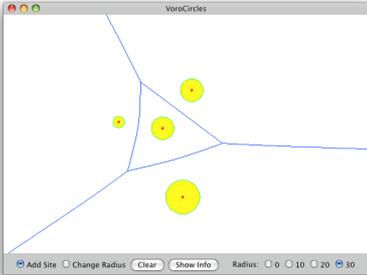
- **Komplexität:**
Ein Voronoi-Diagramm über n Punkten im d -dim. Raum enthält in jeder Dimension d , $0 \leq j \leq d-1$, eine Anzahl f_j von Facetten, wobei alle

$$f_j \in O(n^{\lceil \frac{d}{2} \rceil})$$

G. Zachmann Geometrische Datenstrukturen für die CG 120

Das Voronoi-Diagramm mit additiven Gewichten

- Distanz-Funktion zwischen Punkt x und Site $p_i =$

$$d(\mathbf{x}, \mathbf{p}_i) = \|\mathbf{x} - \mathbf{p}_i\| - r_i$$
- A.k.a. "Appolonius-Diagramm"
- Bisektoren = hyperbolische Bögen
- Beispiel:
 

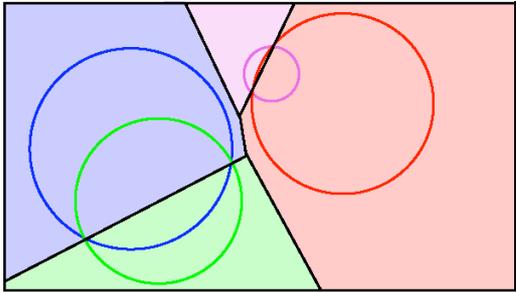
<http://www.geometrylab.de/VoroAdd/index.html>

G. Zachmann Geometrische Datenstrukturen für die CG 121

Das Power-Diagramm

- Distanzfunktion:

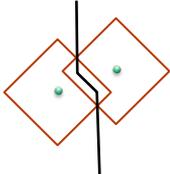
$$d(\mathbf{x}, \mathbf{p}_i) = (\mathbf{x} - \mathbf{p}_i)^2 - r_i$$
- Bisektoren = Geraden
- Beispiel:



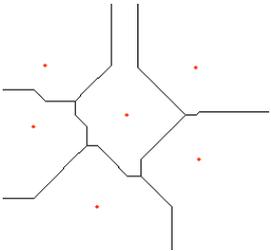
G. Zachmann Geometrische Datenstrukturen für die CG 122

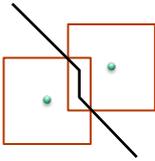
Andere Distanz-Funktionen

- Voronoi-Diagramm mit L_1 - und L_∞ -Norm:

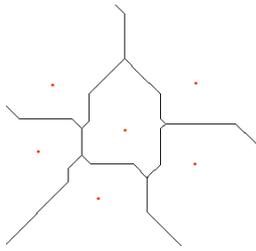


L_1 -Norm
(Manhattan norm)





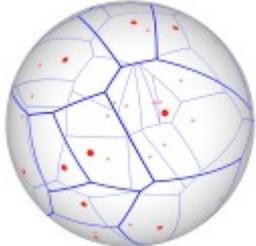
L_∞ -Norm
(supremum/max norm)



G. Zachmann Geometrische Datenstrukturen für die CG 123

Voronoi-Diagramme auf anderen Mannigfaltigkeiten

- Z.B. auf der Kugel:
 - Bisektoren = Großkreise



G. Zachmann Geometrische Datenstrukturen für die CG 124

Higher-order Voronoi Diagrams

- In einem **Voronoi-Diagramm k -ter Ordnung** $V_k(S)$ gehören alle diejenigen Punkte des Raumes zur selben Voronoi-Region, die die selben k nächsten Nachbarn aus S haben
- Unterschiede zum klassischen Voronoi-Diagramm:
 - Ein Bisektor kann zu mehreren Begrenzungskanten (-ebenen) beitragen
 - Eine Voronoi-Region muß ihre Generatoren nicht mehr enthalten

G. Zachmann Geometrische Datenstrukturen für die CG 125

Beispiel:

1-st order 2-nd order

3-rd order 4-th order

G. Zachmann Geometrische Datenstrukturen für die CG 126

Demo

Higher Order Voronoi Diagrams - Demonstration Applet

Higher Order Voronoi Diagrams - ...

Higher Order Voronoi Diagrams

Demonstration Applet

mouse=(298,170) over vertex@(298.39,168.64)

n=11; 82 edges, 64 vertices

Order k = 4 1 10

Lock: k n-k clear

Instructions

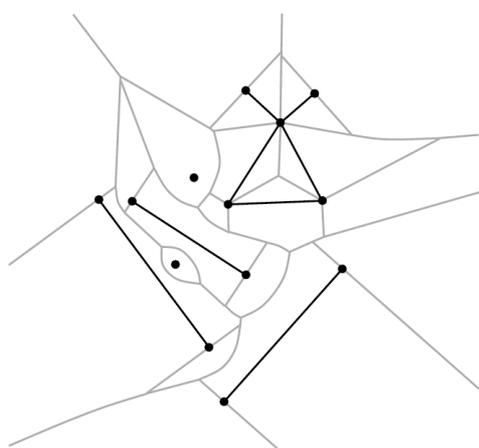
Use the mouse to add and remove points:
Applet: VoronoiOrderSource

Andreas Pollack - <http://www.pollack.org/en/otherstuff/informatics/voronoi/>

G. Zachmann Geometrische Datenstrukturen für die CG 127

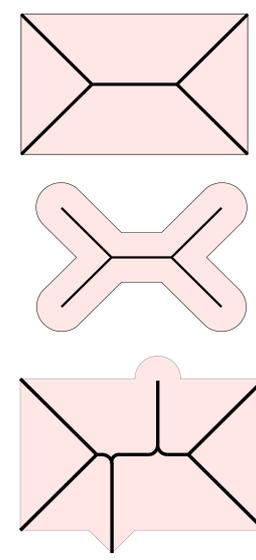
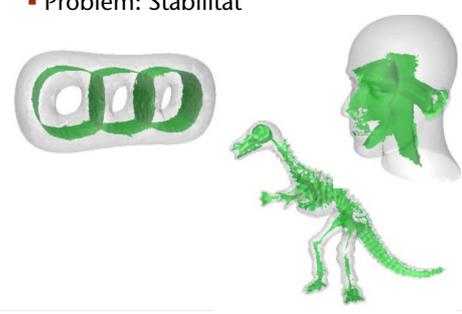
Voronoi-Diagramm von Liniensegmenten

- Sites sind jetzt Punkte + Liniensegmente
- Bisektoren = Geraden + Parabeln
- Beispiel:

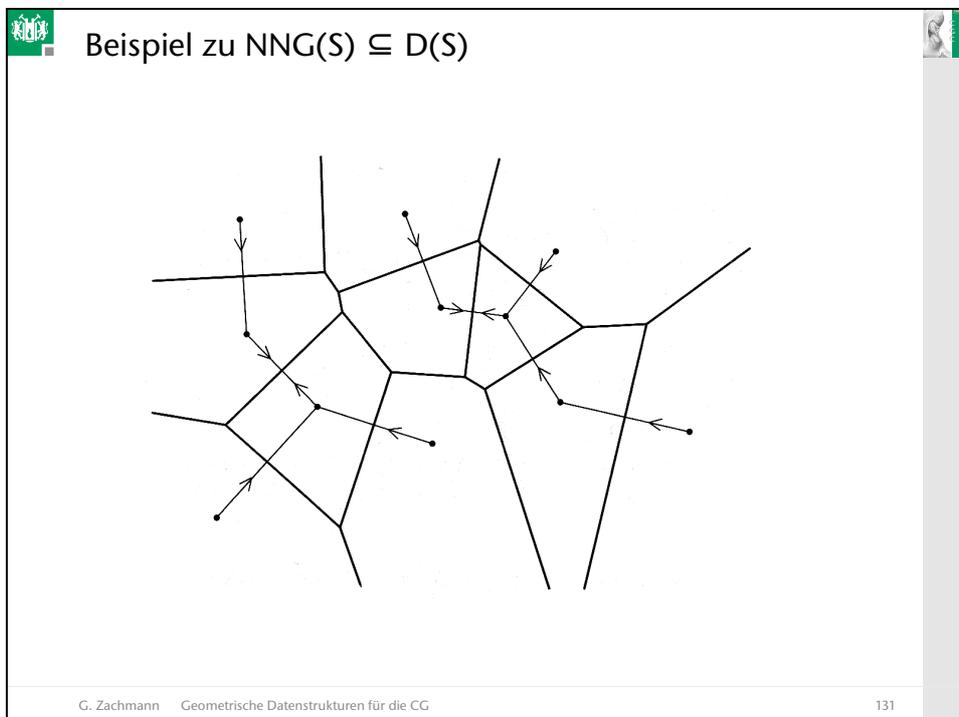
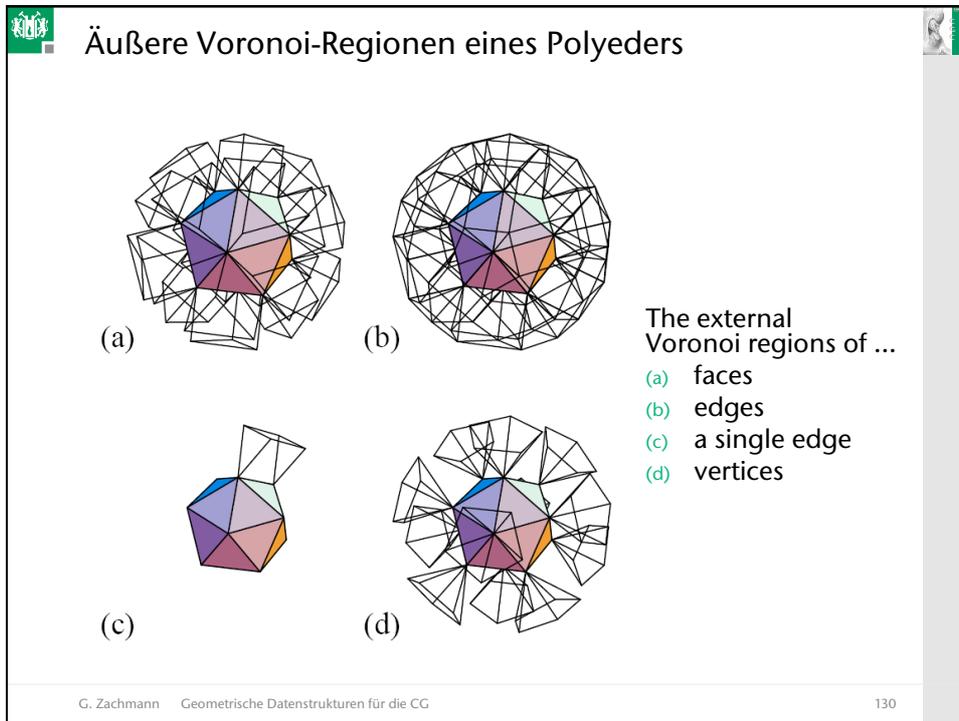


G. Zachmann Geometrische Datenstrukturen für die CG 128

- Anderer Name: **Skeleton** oder **Medial Axis**
 - Besonders im Fall von geschlossenen Objekten
 - Alle Punkte, die gleich weit von 2 Punkten des Randes eines Objektes entfernt sind
 - Problem: Stabilität

G. Zachmann Geometrische Datenstrukturen für die CG 129



Maximale, leere Kreise

(i) (ii) (iii)

G. Zachmann Geometrische Datenstrukturen für die CG 132

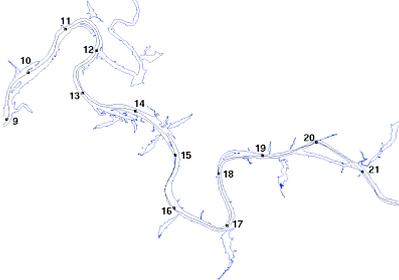
Anwendungsgebiete der Voronoi-Diagramme

- **Anthropology and Archeology** -- Identify the parts of a region under the influence of different Neolithic clans, chiefdoms, ceremonial centers, or hill forts.
- **Astronomy** -- Identify clusters of stars and clusters of galaxies (Here we saw what may be the earliest picture of a Voronoi diagram, drawn by Descartes in 1644, where the regions described the regions of gravitational influence of the sun and other stars.)
- **Biology, Ecology, Forestry** -- Model and analyze plant competition ("Area potentially available to a tree", "Plant polygons")
- **Cartography** -- Piece together satellite photographs into large "mosaic" maps
- **Crystallography and Chemistry** -- Study chemical properties of metallic sodium ("Wigner-Seitz regions"); Modelling alloy structures as sphere packings ("Domain of an atom")
- **Finite Element Analysis** -- Generating finite element meshes which avoid small angles
- **Geography** -- Analyzing patterns of urban settlements
- **Geology** -- Estimation of ore reserves in a deposit using information obtained from bore holes; modelling crack patterns in basalt due to contraction on cooling
- **Geometric Modeling** -- Finding "good" triangulations of 3D surfaces
- **Marketing** -- Model market of US metropolitan areas; market area extending down to individual retail stores
- **Mathematics** -- Study of positive definite quadratic forms ("Dirichlet tessellation", "Voronoi diagram")
- **Metallurgy** -- Modelling "grain growth" in metal films
- **Meteorology** -- Estimate regional rainfall averages, given data at discrete rain gauges ("Thiessen polygons")
- **Pattern Recognition** -- Find simple descriptors for shapes that extract 1D characterizations from 2D shapes ("Medial axis" or "skeleton" of a contour)
- **Physiology** -- Analysis of capillary distribution in cross-sections of muscle tissue to compute oxygen transport ("Capillary domains")
- **Robotics** -- Path planning in the presence of obstacles
- **Statistics and Data Analysis** -- Analyze statistical clustering ("Natural neighbors" interpolation)
- **Zoology** -- Model and analyze the territories of animals

G. Zachmann Geometrische Datenstrukturen für die CG 133

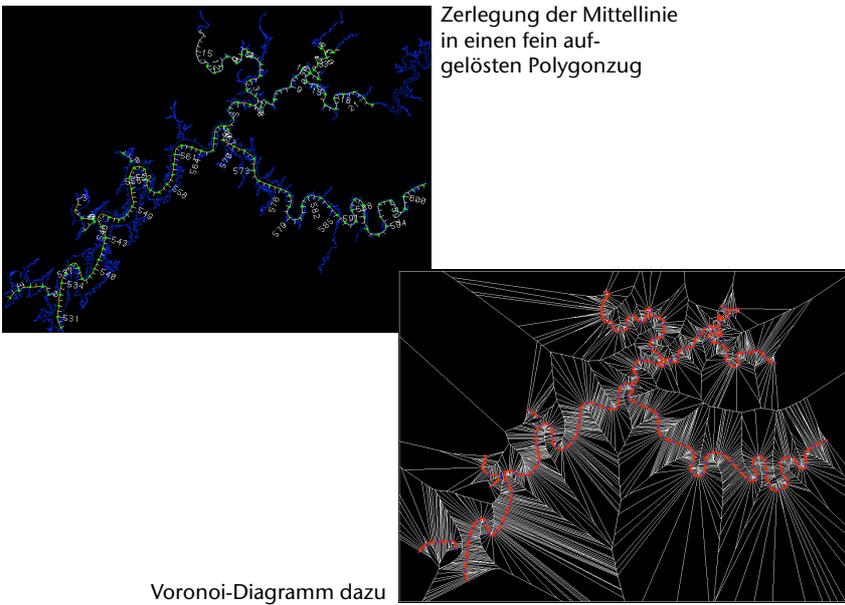
Anwendung: das River-Mile-Koordinatensystem

- Das River-Mile-Koordinatensystem:
 - Wird gerne in großen Wasserwegesystemen angewendet
 - Koordinaten eines Punktes in der Ebene = (l, q) wobei
 - l = gemessen entlang der Mittellinie des Flusses,
 - q = Entfernung von Punkt $(l, 0)$ senkrecht zur Tangente in $(l, 0)$
 - Problem: gegeben ein Punkt (x, y) – welche Koord. (l, q) hat er?



G. Zachmann Geometrische Datenstrukturen für die CG 134

Zerlegung der Mittellinie in einen fein aufgelösten Polygonzug



Voronoi-Diagramm dazu

G. Zachmann Geometrische Datenstrukturen für die CG 135

Voronoi-Clustering

- Aufgabe:
 - Gegeben: Menge von Punkten
 - Gesucht: Partitionierung der Punktmenge in "Cluster"
 - Clustering =
 - maximal intra-cluster similarity and
 - minimal inter-cluster similarity

G. Zachmann Geometrische Datenstrukturen für die CG 136

Gute Einteilung in Wahlbezirke (*redistricting*)

- Das Fairness-Prinzip: "one man, one vote"
 - Ganz einfach ... oder?
- Einfaches Beispiel:

R	R	R	R
R	R	D	D
D	D	D	D

Wählerverteilung

R	R	R	R
R	R	D	D
D	D	D	D

Gleiche Anzahl
Repräsentanten

R	R	R	R
R	R	D	D
D	D	D	D

Demokraten
gewinnen

R	R	R	R
R	R	D	D
D	D	D	D

Republikaner
gewinnen

Copyright © 2001 by Michael D. Robbins, FraudFactor.com

- Gesetzliche Kriterien für Wahlbezirke in den USA:
 - Gleiche Anzahl Wähler
 - Jeder Bezirk soll zusammenhängend sein
 - "Kompaktheit" (ist im US-Gesetz nicht klar definiert)

G. Zachmann Geometrische Datenstrukturen für die CG 137

- Böses Beispiel:**

"In gerrymandered election districts, the voters don't choose their politicians - the politicians choose their voters!"

LOUISIANA
Congressional Election District Map, including the disputed 4th district
© FraudFactor.com

1990 (?)

G. Zachmann Geometrische Datenstrukturen für die CG 138

- Ähnlicher Effekt bei Europawahlen: die Stimme eines Wählers in Malta oder Luxemburg hat 10x mehr Gewicht als die eines deutschen Wählers!**

Number of seats plotted against the population of each State

State	Population (in millions)	Number of seats
Malta	0.4	6
Luxemburg	0.5	6
Slovenia	2.1	6
Cyprus	0.8	6
Iceland	0.3	6
Lithuania	3.3	6
Ireland	4.1	6
Poland	38.0	13
Slovakia	5.4	13
Czechia	10.5	13
Belgium	10.5	13
Austria	8.5	13
Sweden	9.0	13
Hungary	10.3	13
Portugal	10.6	13
Finland	5.3	13
Denmark	5.5	13
Spain	45.9	13
Italy	60.3	13
United Kingdom	61.5	13
France	64.0	13
Germany	82.0	13

G. Zachmann Geometrische Datenstrukturen für die CG 139

- Eine mögliche Definition von Kompaktheit:
 - Sei $\mathcal{D} = \{D_1, \dots, D_k\}$
eine Menge von Wahlbezirken (*districts*).
 - Jeder Distrikt $D_i = \{p_j, \dots, p_l\} \subset P = \{p_1, \dots, p_n\}$
enthält eine Menge von Wählern p_i .
 - Die Kompaktheit eines Distrikts ist
$$c(D) = \sum_{i,j=1}^{|D|} d(p_i, p_j)$$
 - Die Gesamtkompaktheit der Einteilung in Distrikte ist
$$c(\mathcal{D}) = \sum_{i=1}^k c(D_i)$$

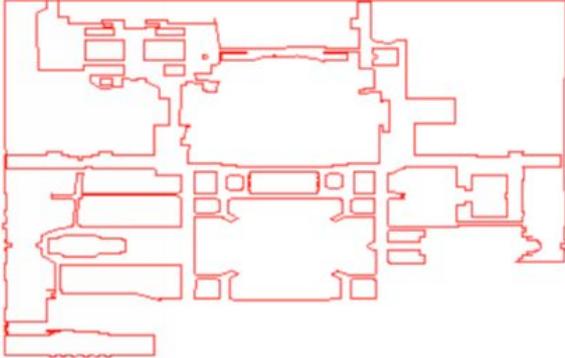
G. Zachmann Geometrische Datenstrukturen für die CG 140

- Theorem:
 - Eine optimale Aufteilung in Wahlbezirke bzgl. Kompaktheit geht aus einem Power-Diagramm hervor.
- Aufgabe :
 - Konstruiere zu gegebener Menge Wähler (p_i)
eine Menge von Voronoi-Sites mit Gewichten, so daß
 - $\forall i : |D_i| = n$
- Voronoi-Sites = "Wahllokale"
 - Gewicht = Maß für die Populationsdichte in dem zugehörigen Distrikt
(kleines Gewicht = hohe Dichte)
- Ansatz :
 - Starte mit zufälligen Sites und Gewichten
 - Verschiebe Sites und Gewichte, bis $c(\mathcal{D})$ in lokalem Minimum

G. Zachmann Geometrische Datenstrukturen für die CG 141

Pfadplanung

- Gegeben: Grundriß als Menge von Liniensegmenten
- Gesucht: Pfad (z.B. für autonomes Vehikel = Roboter) mit maximalem Abstand zu den Wänden



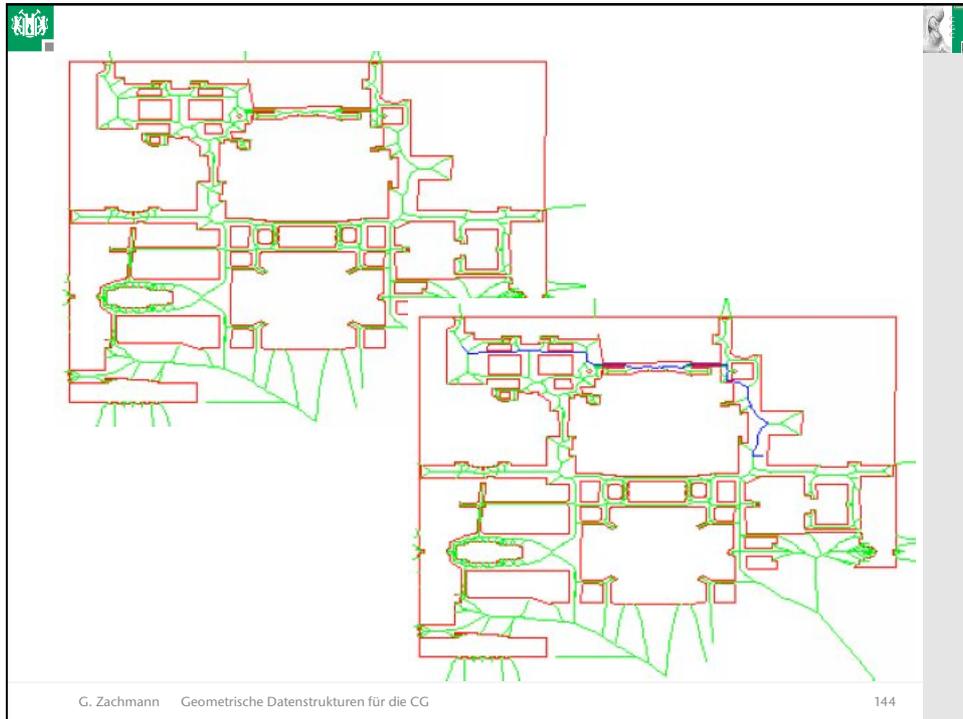
http://www.cs.columbia.edu/~pblaer/projects/path_planner/

G. Zachmann Geometrische Datenstrukturen für die CG 142

Lösung:

- (Verallgemeinertes) Voronoi-Diagramm dazu konstruieren
- Nächste Voronoi-Knoten zu Start- und Endpunkt suchen
- Mit Dijkstra-Algo kürzesten Pfad von Start- zu End-Knoten durch Voronoi-Graph suchen

G. Zachmann Geometrische Datenstrukturen für die CG 143



Bewertung von Samplings

- Beispiel: Wetterstationen
- Frage: wo ist die geringste Dichte?
- Ideales Sampling → jeder Punkt würde eine Fläche von

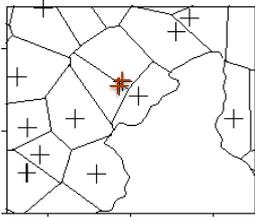
$$\bar{A} = \frac{A}{n}$$
 abdecken (A = Gesamtfläche)

G. Zachmann Geometrische Datenstrukturen für die CG 145

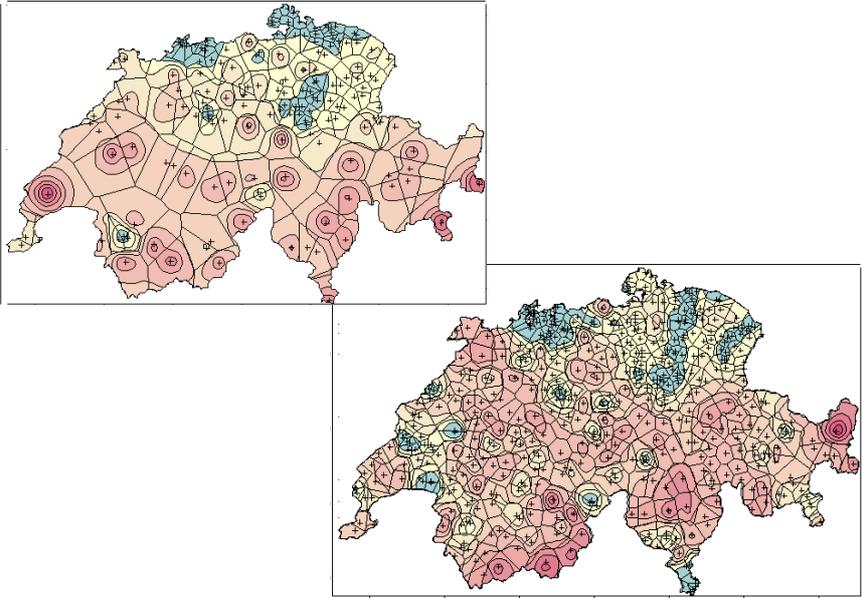
■ Lösung:

- Voronoi- und Delaunay-Diagramm berechnen
- Relative Größe pro Zelle ist

$$A_i = \frac{V_i}{A}$$
- $A_i > 1 \rightarrow$ zu geringe Dichte
- Sample-Punkte "bestrafen", falls sie dicht beieinander liegen relativ zur Größe der Zelle
 \rightarrow Distanz zum nearest neighbor



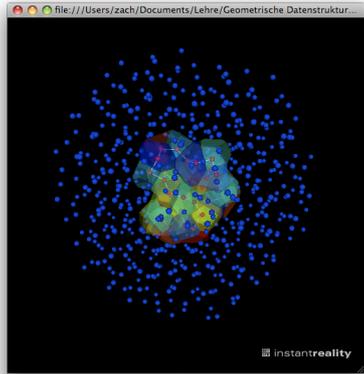
G. Zachmann Geometrische Datenstrukturen für die CG 146



G. Zachmann Geometrische Datenstrukturen für die CG 147

Protein-Struktur-Analyse

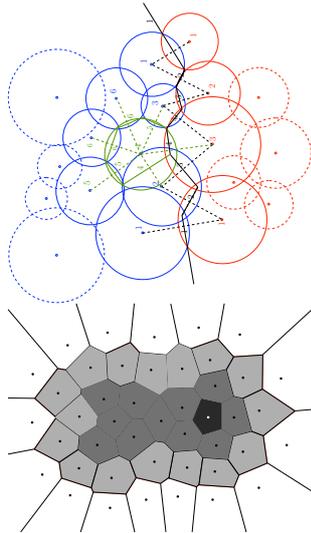
- Frage:
 - Wie sieht die aktive Oberfläche (= **Interface**) eines Moleküls aus?
 - Welche Atome interagieren mit Atomen aus der Umgebung
- Eine Lösung:
 - Platziere zufällig Atome um das geg. Molekül herum
 - Berechne das Voronoi-Diagramm alle Punkte
 - Interface = Voronoi-Facetten zwischen Molekül und Umgebungsatomen



G. Zachmann Geometrische Datenstrukturen für die CG 148

Verbesserungen

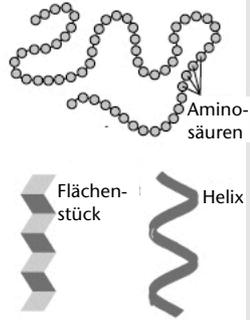
- Verwende Power-Diagramm oder Voronoi-Diagramm mit additiven Gewichten
 - Gewicht = Atomradius
- Berechne "Tiefe" pro Atom:
 - Atome mit einer Voronoi-Facette nach außen = Tiefe 1
 - Traversiere Delaunay-Graph breadth-first von außen nach innen
 - Je tiefer ein Atom, desto geringer sein Beitrag zu Wechselwirkungen



G. Zachmann Geometrische Datenstrukturen für die CG 149

Secondary Structure of Proteins

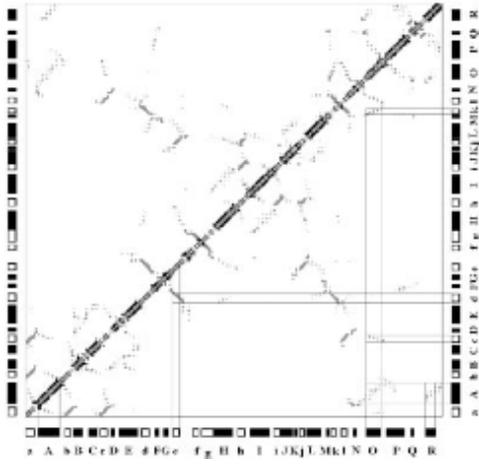
- Lange Proteine falten sich zu Helices, Knäueln, und Flächenstücken
- Ergibt Wechselwirkungen zwischen Atomen (Bindungen), die nicht in der chemischen Formel zu sehen sind
- Frage: gegeben die Positionen der Atome, wie sieht die sekundäre Struktur aus?
 - Welche Atome sind "benachbart", welche nicht
 - Wie stark sind sie benachbart?
- Lösung: Voronoi-Diagramm
 - Benachbart = gemeinsame Voronoi-Facette
 - Stärke der Nachbarschaft = Größe der Facette



The diagram illustrates the secondary structure of a protein. It shows a chain of amino acids (Aminosäuren) represented by small circles. The chain folds into a helix (Helix) and a sheet (Flächenstück). The labels 'Aminosäuren', 'Flächenstück', and 'Helix' are placed next to their respective structures.

G. Zachmann Geometrische Datenstrukturen für die CG 150

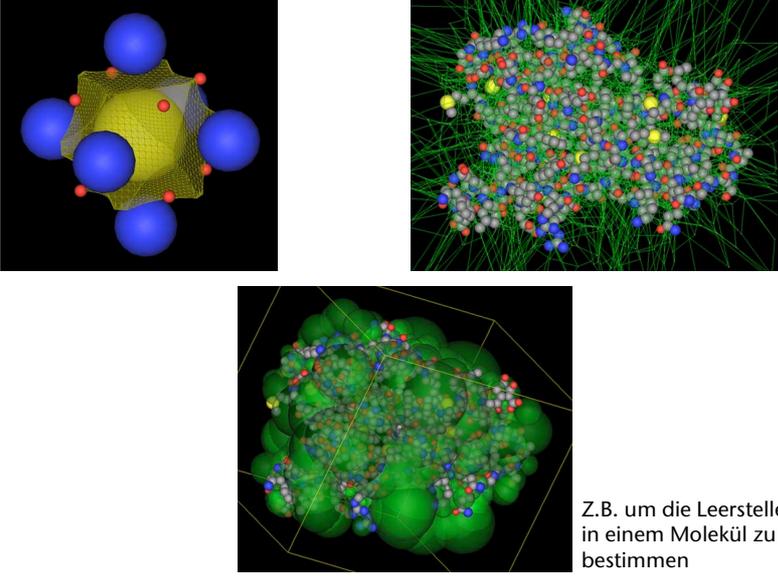
- Resultat: Adjazenz-Matrix (grau/schwarz = schwach/stark benachbart)



The adjacency matrix visualization shows a grid of squares representing the strength of adjacency between amino acids. The squares are colored gray or black, indicating the strength of the adjacency. The axes are labeled with amino acid letters: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R.

G. Zachmann Geometrische Datenstrukturen für die CG 151

Appolonius-Diagramme in 3D



Z.B. um die Leerstellen in einem Molekül zu bestimmen

G. Zachmann Geometrische Datenstrukturen für die CG 152

Visibility Sorting using Voronoi Diagrams

- Erinnerung: BSPs für Visibility-Sortierung
- Methode:
 - Definiere eine Visibility-Relation auf Voronoi-Regionen

$$R_1 \prec_v R_2$$

jeder Punkte der Voronoi-Zelle R_2 wird durch einen Punkt der Zelle R_1 bzgl. des Viewpoints v verdeckt
 - Nun gilt:

$$R_1 \prec_v R_2 \Leftrightarrow \|v - p_1\| < \|v - p_2\|$$

- Bew.: klar weil R_1 und R_2 komplett auf verschiedenen Seiten des Bisektors zwischen R_1 und R_2 liegen.

G. Zachmann Geometrische Datenstrukturen für die CG 153

- Idee:
 - Zunächst alle Pgone in Voronoi-Zellen clustern
 - Zur Laufzeit nur noch die Voronoi-Sites sortieren (inkrementell)
- Ansatz zum Voronoi-Clustering:
 - Zu Beginn eine Zelle pro Polygon mit Schwerpunkt als Site
 - Die kleinste Zelle löschen:
 - Voronoi-Diagramm lokal neu berechnen
 - Polygone der kleinsten Zelle den Nachbarzellen zuordnen
 - Abbruch falls keine Zelle mehr aufgelöst werden kann, ohne daß eine zyklische Visibility-Ordnung in einer Zelle entsteht

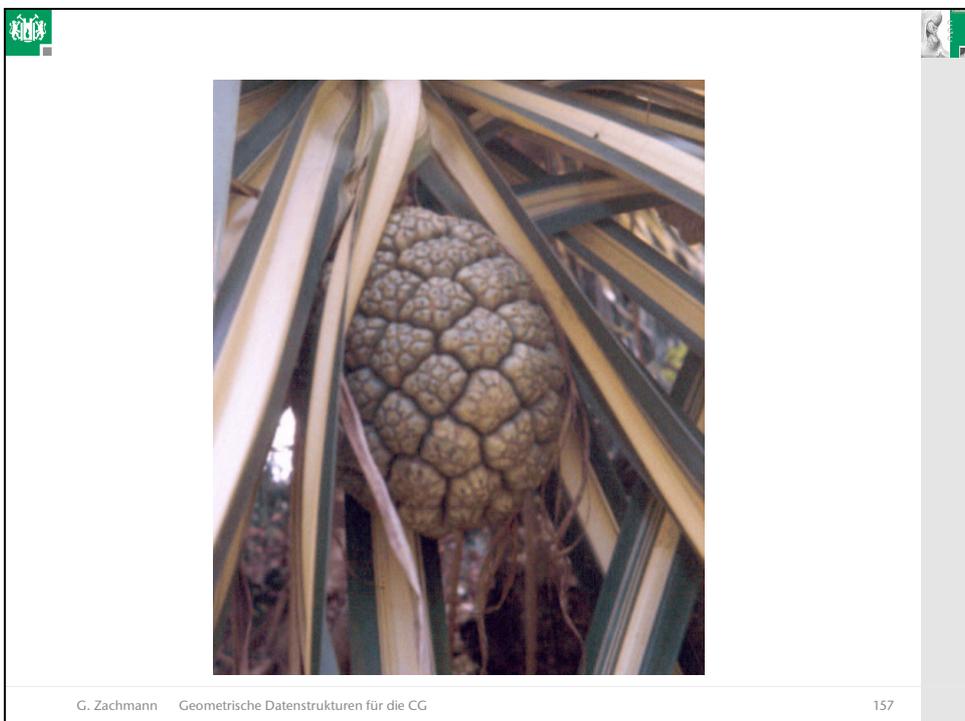
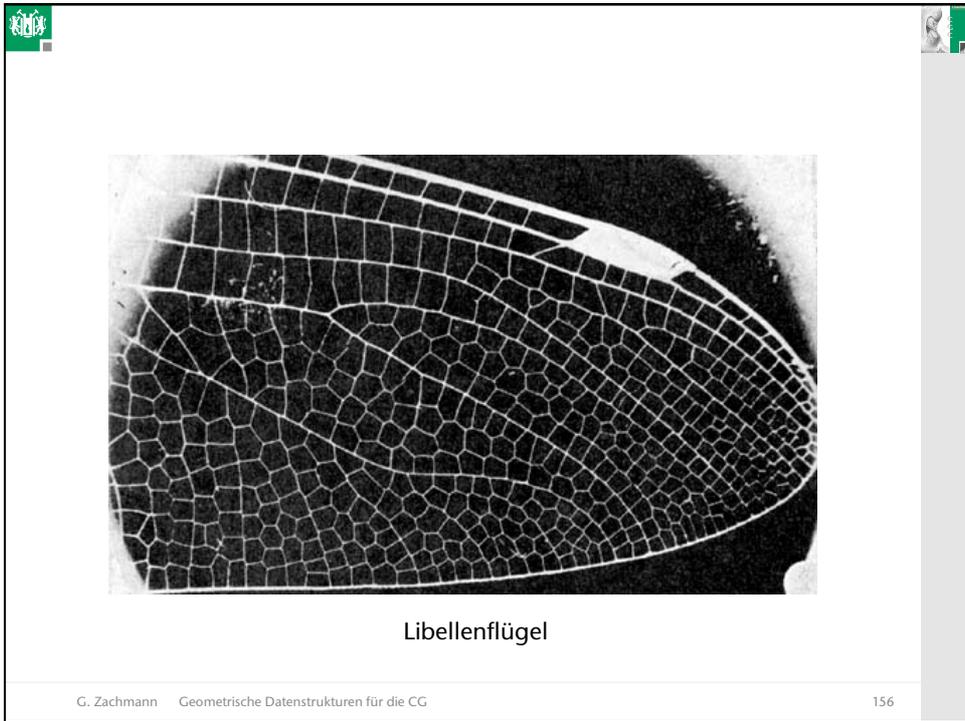
G. Zachmann Geometrische Datenstrukturen für die CG 154

Voronoi-Diagramme in der Natur

Seifenblasen in einem Glasrahmen

Bienenwaben
(centroidal Voronoi tessellation)

G. Zachmann Geometrische Datenstrukturen für die CG 155





Thanks Folks



G. Zachmann Geometrische Datenstrukturen für die CG

161

The image is a reproduction of Raphael's fresco 'The School of Athens'. It depicts a group of ancient Greek philosophers gathered in a grand, vaulted hall. The architecture features a series of arches and columns, with a prominent archway at the far end. The figures are engaged in various activities: some are seated on benches, some are standing and gesturing, and others are in the process of writing or reading. The scene is set within a large, semi-circular archway, and the floor is decorated with a geometric pattern. The overall composition is highly detailed and serves as a classic example of Renaissance art's focus on perspective and classical architecture.