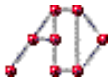
Algorithmische Geometrie

Prof. Dr. Thomas Ottmann

Mitarbeit: PD Dr. Sven Schuierer
Dr. Stefan Edelkamp

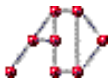
Literatur:

M. de Berg, M. van Krefeld, M. Overmars
O. Schwarzkopf: Computational Geometry
(Algorithms and Animations)



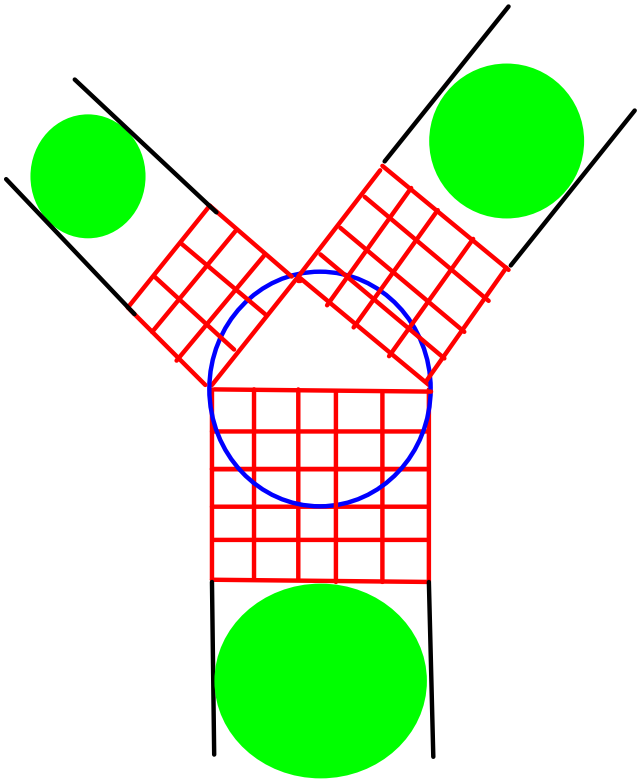
Algorithmische Geometrie

- Einführung
  - Historie
  - Problemfelder
  - Ein Beispiel: Konvexe Hülle
    - "Naives Verfahren"
    - Graham’s Scan
    - Untere Schranke
  - Entwurf, Analyse und Implementation



Historie: Beispiel Beweis-basierter Geometrie

Satz des Pythagoras (582-497 v.Chr.)



Schon Babyloniern und Ägyptern als experimentelle Tatsache bekannt

Pythagoräische Innovation: Ein von numerischer Verifikation unabhängiger Beweis.

Quadrat ist ein Flächenmaß

Satz gilt für jede Klasse ähnlicher Figuren (also auch für Kreise, Dreiecke,...)

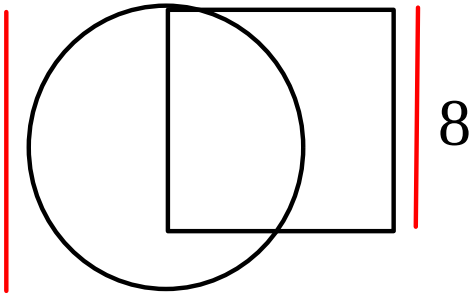
Beispiel aus der Antike für algorithmische Geometrie

Auszug aus dem Rhid Papyrus (ca. 1700 v.Chr.)
Kopie eines älteren Papyrus von ca. 1900 v.Chr.

Problem 50: Berechnung der Fläche eines kreisförmigen Flächenstücks mit Durchmesser 9 Stäbe.

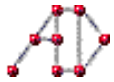
Ziehe vom Durchmesser den neunten Teil ab
was 8 Teile ergibt. Multipliziere 8 mit 8, was 64 ergibt.
Dann ist die Fläche 6 kha und 4 setat.

A=(8/9 2r)^2
= 256/81 r^2
= ca. 3.16 r^2



approximiert π
bis auf 2 Prozent

"Experimentelle Quadratur des Zirkels"



Historisches Beispiel axiomatischer Geometrie

Einige Axiome aus den "Elementen" des Euklid (~325- v.Chr)

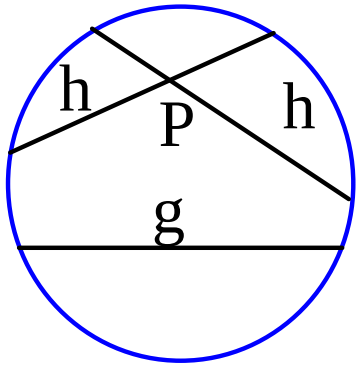
Grundbegriffe: Punkt, Gerade, Ebene
Inzidenzrelation ("liegt auf", "geht durch")

A1: Zu je 2 verschiedenen Punkten P,Q gibt es genau eine Gerade g auf der P und Q liegen

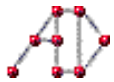
A2: Zu jeder Geraden g gibt es einen Punkt, der nicht auf g liegt

A3: Zu jederGeraden g und jedem Punkt P, der nicht auf g liegt, gibt es genau eine Gerade h, auf der P liegt und die mit g keinen Punkt gemeinsam hat.

Frage: Ist A3 unabhängig von A1 und A2?



Klein'sches Modell

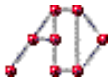


Algorithmische Geometrie heute

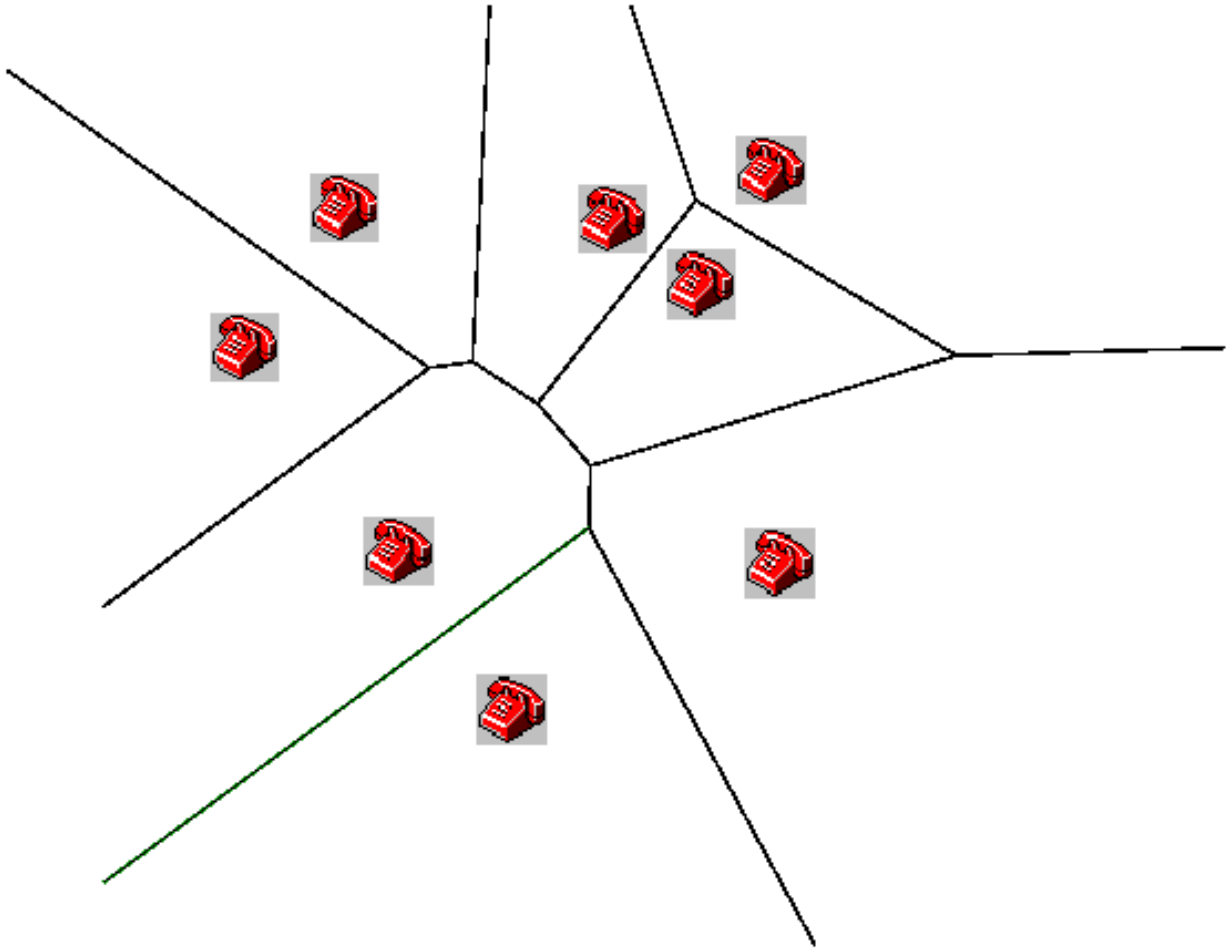
Rückbesinnung auf historische Wurzeln

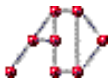
Suche nach einfachen, robusten, effizienten Algorithmen
Aufspaltung in
- eher theoretische Untersuchungen
- Entwicklung praktisch nutzbater Werkzeuge

Hunderte von Arbeiten/Jahr
Anwendung algorithmischer Techniken und Datenstrukturen
Effiziente Lösung einfacher, "trivialer" Probleme
Entwicklung neuer Techniken und Strukturen
- Randomisierung und inkrementelle Konstruktion
- kompetitive Verfahren

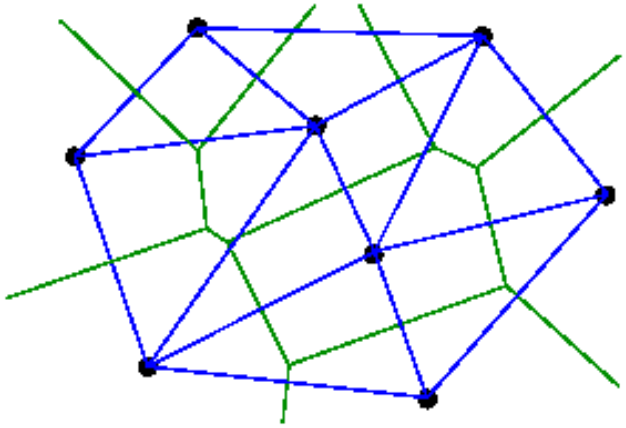


Die Suche nach einem öffentlichen Telefon

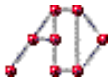




Verschiedene Algorithmen für Punkte

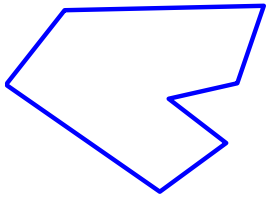


Minimaler Spannbaum
Delaunay Triangulation
Konvexe Hülle
Voronoi-Diagramm



Konvexität

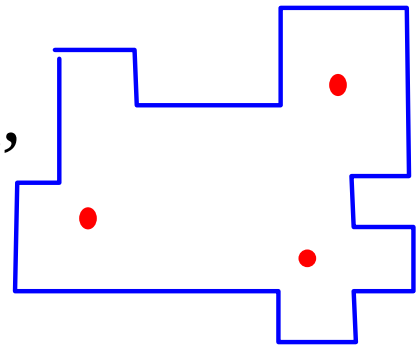
Konvexe Hülle (folgt)



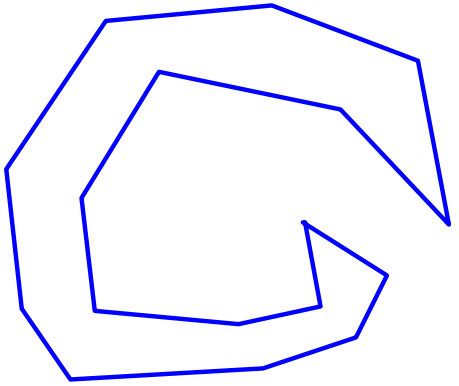
Kern eines Polygons

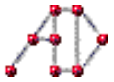
Verallgemeinerungen (z.B. orthogonale Konv.)

Art-Gallery Probleme:
Wieviele Wächter sind nötig, um eine (rechteckige) Art-Gallery zu überwachen



Berechnung optimaler Wächter-Routen (für spiralförmige Polygone)

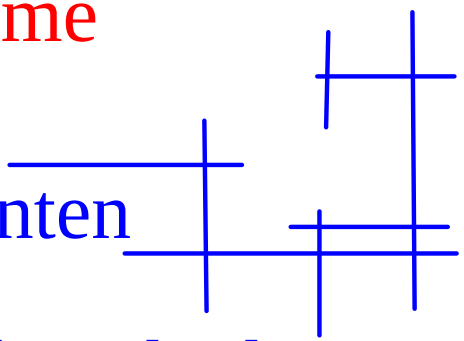


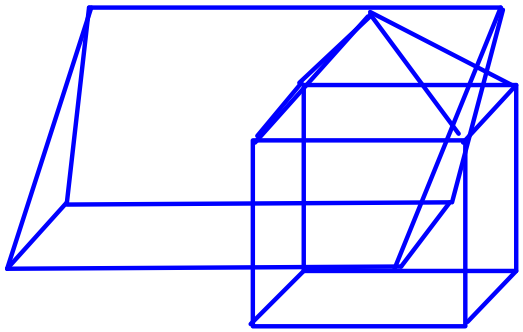


Schnitt- und Sichtbarkeitsprobleme

Geg.: Menge von
- (iso-orientierten) Liniensegmenten
- Rechtecken, Polygonen, etc.

Ges.: alle Paare sich schneidender Objekte

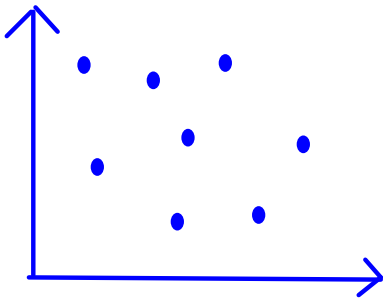




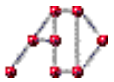
Hidden-line-Elimination

Visible-surface-Berechnung

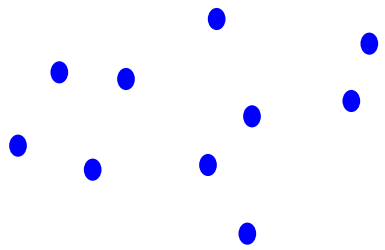
(direkte) Inklusionen



(direkte) Dominanzen

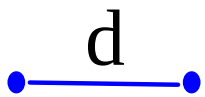


Geometrische Suche



- Bereichssuche
- Closest Pair
  (nächster Nachbar)

Bsp. Closest Pair



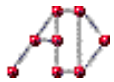
n(n-1)/2 Elementaroperationen ->
trivialer O(n^2) Algorithmus

Gibt es einen effizienteren Algorithmus?

Komplexitätstheorie: $\Omega(n \log n)$

Kann Lücke zwichen oberer und unterer Schranke geschlossen werden.

Asymptotik ist praktisch relevant

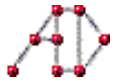


Unterschied zwischen n log n und n^2

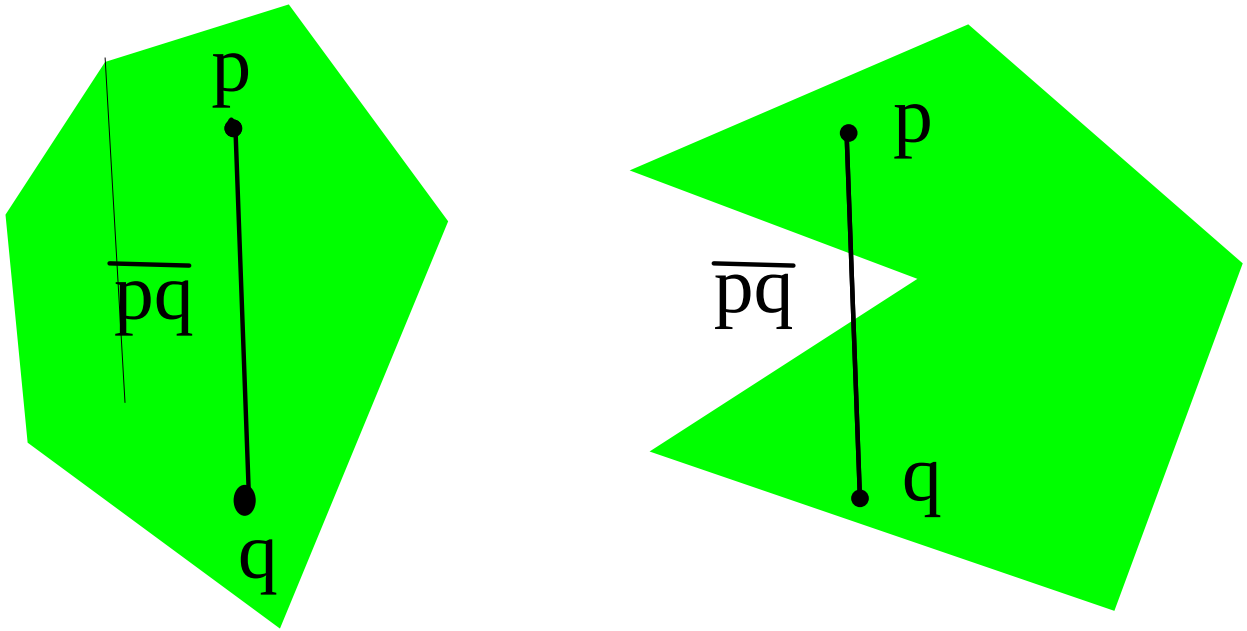
n	n log n	n^2
2^10~10^3	10 2^10~10^4	2^20~10^6
2^20~10^6	20 2^20~2 10^7	2^40~10^12

interaktive Verarbeitung	n log n Algo	n^2 Algo
n=1000	ja	nein
n=1000000	ja	nein

Algorithmische Geometrie hat neue Typen von Algorithmen entwickelt, die viele 2-d-Aufgaben einfach und effizient lösen.

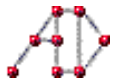


Konvexe Hülle



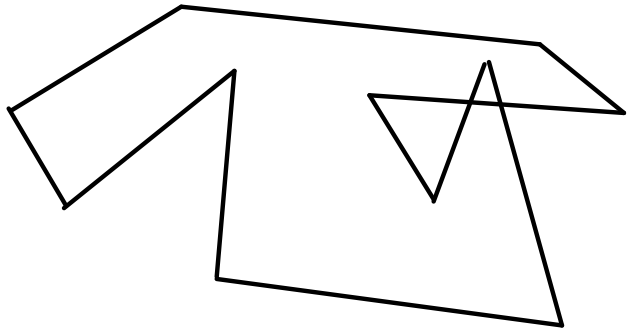
Teilmenge S der Ebene konvex, wenn für alle Paare p,q die Strecke $\overline{pq}$ in S liegt.

Die konvexe Hülle CH(S) ist die kleinste konvexe Menge, die S enthält.



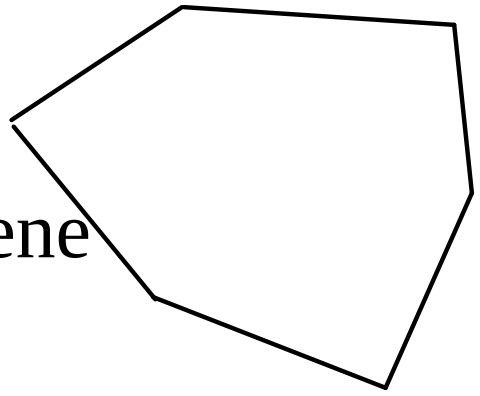
Polygone

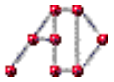
Ein Polygon P ist ein geschlossener Kantenzug in der Ebene.



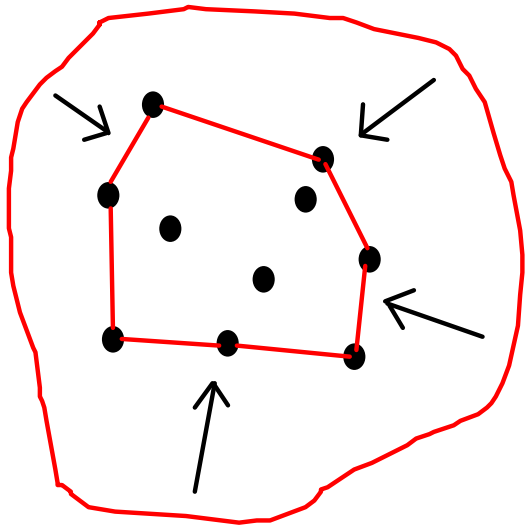
Ein Polygon heißt einfach, wenn es sich nicht selber schneidet.

Ein einfaches Polygon heißt konvex, wenn die eingeschlossene Fläche konvex ist.



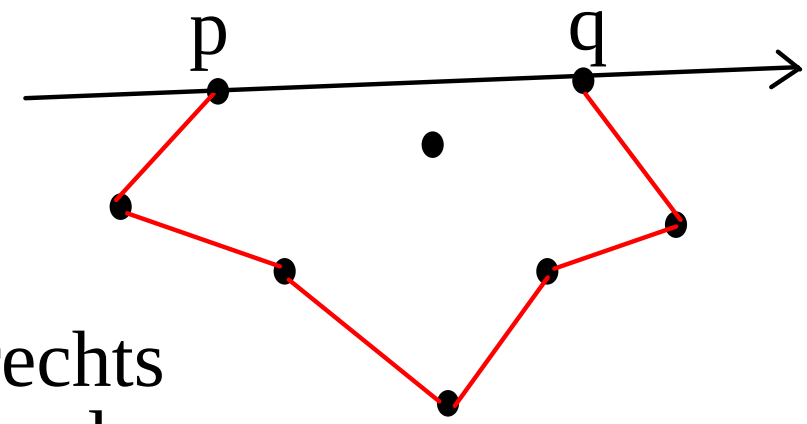


Konvexe Hülle für Punktmengen



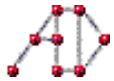
Das Gummiband

zur Bestimmung einer Konvexen Hülle einer Punktmenge P.



Die Rechtsregel:

Alle Punkte liegen rechts von $\overline{pq}$, dann und nur dann, wenn $\overline{pq}$ eine Kante von KH(P) ist.



"Naives Verfahren"

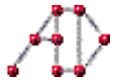
Eingabe: Menge von Punkten P
Ausgabe: Konvexe Hülle CH von P

E={}
for all (p,q) aus P x P mit p<>q
    valid = true
     for all r in P mit r <> q and r <> q
        if r liegt links von $\overline{pq}$
            valid = false
     if valid then E = E $\cup$ {$\overline{pq}$}

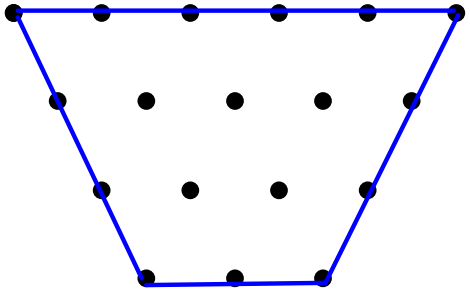
Konstruiere CH(P) als eine als eine Liste von Knoten aus E

Laufzeit: O(n^3)

Langsam.run

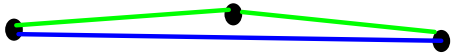


Problemfälle (Degenerationen)



1. Lineare Abhängigkeit: mehrere Punkte liegen auf einer Linie

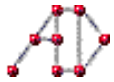
Lösung: Erweiterte Fallunterscheidung



2. Rechenungenauigkeiten

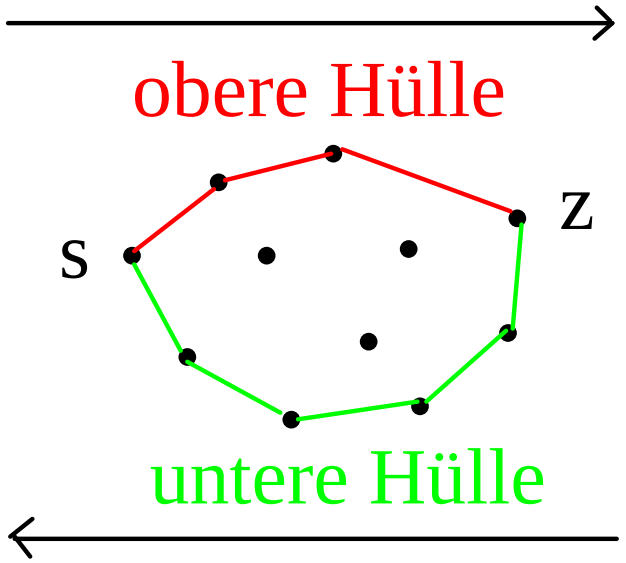
aufgrund der Rechnerarithmetik(floats)

Lösungen: Exakte Arithmetik, Intevallarithmetik



Dynamische, zweigeteilte Berechnung

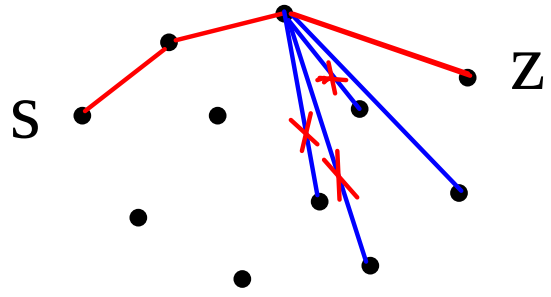
Aufteilung des Problems:

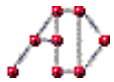


Inkrementalität

Gegeben: Hülle für $p_1,\ldots,p_{i-1}$

Gesucht: Hülle für $p_1,\ldots,p_i$





Schnelle Berechnung der Konvexen Hülle

Eingabe/Ausgabe: Siehe "naives Verfahren"

Sortiere P nach x Koordinaten

LU = {p1,p2}

for i=3..n

  LU = LU $\cup$ {p_i}

  while |LU|>2 und letzten 3 keine Rechtskurve

    Lösche den mittleren der 3

LL = {p_n,p_{n-1}}

for i=n-2..1

  LL = LL $\cup$ {p_i}

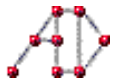
  while |LL|>2 und letzten 3 keine Rechtskurve

    Lösche den mittleren der 3

Lösche ersten und letzten Punkt in LL

CH(P) = LU $\cup$ LL

upper.tun, convex.run



Laufzeit

Satz: CH(P) kann in O(n log n) berechnet werden

Beweis:

Obere Hülle

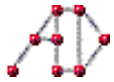
Sortieren der Punkte in O(n log n) möglich.

Die for-Schleife wird linear oft aufgerufen.

While-Schleife insgesamt O(n).

Analog für untere Hülle

.



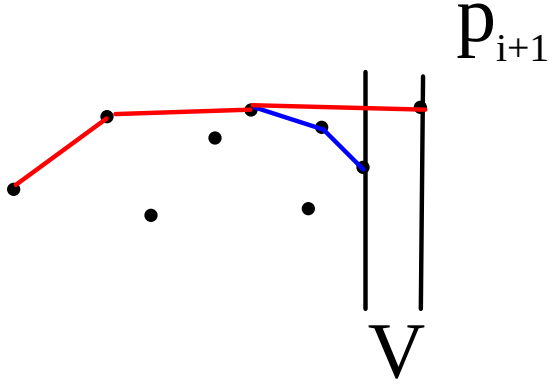
Korrektheit

Satz: Die Berechnung der konvexen Hülle istkorrekt.

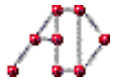
Beweis: Obere Hülle (Induktion)

$\{p_1,p_2\}$ ist eine UH.

Sei $\{p_1,\ldots,p_i\}$ eine UH und betrachte p_{i+1}



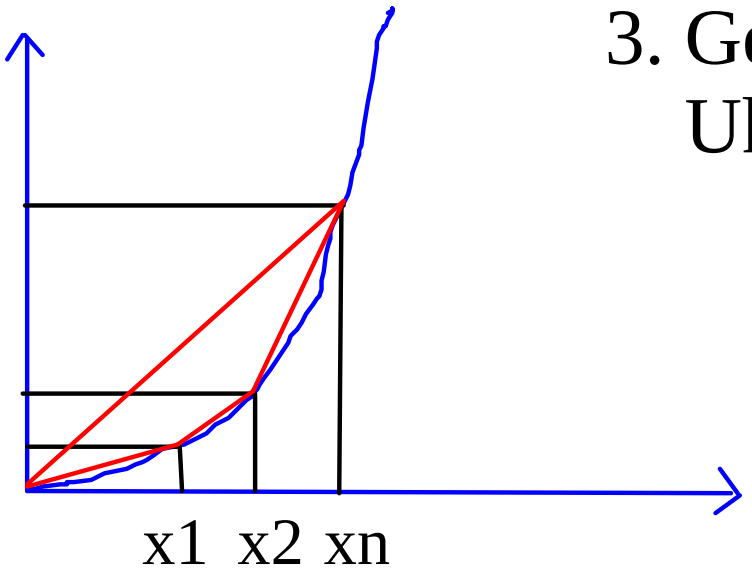
Per Induktion kann ein zu hoher Punkt nur in der Spalte V liegen. Dies widerspricht jedoch der lexikographischen Ordnung.



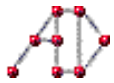
Komplexität

Reduktion des Sortierproblems auf die Berechnung der konvexen Hülle

1. x1,...,xn -> (x1,x1^2),...,(xn,xn^2) O(n)
2. Konstruiere konvexe Hülle für diesen Punkt



3. Gebe Punkte im Uhrzeigersinn aus



Entwurf, Analyse und Implementation

1. Entwerfe Algorithmus, ignoriere Sonderfälle

2. Behandle aller Sonderfälle

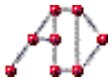
3. Implementation

   Berechnung geometrischer Objekte
   -> so gut wie möglich
   Entscheidungen (z.B. Vergleichsoperationen)
   -> müssen exakt sein

Hilfen

Bibliotheken: LEDA, CGAL

Visualisierungen: VEGA

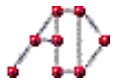


Anwendungsgebiete

Computer Graphik

2-dimensionale Graphik:
Anfragen zur Punktlokalisation oder Schnittberechnungen

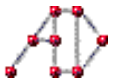
3-dimensionale Graphik:
Hidden Surface Removal (Löschen verdeckter Linien)



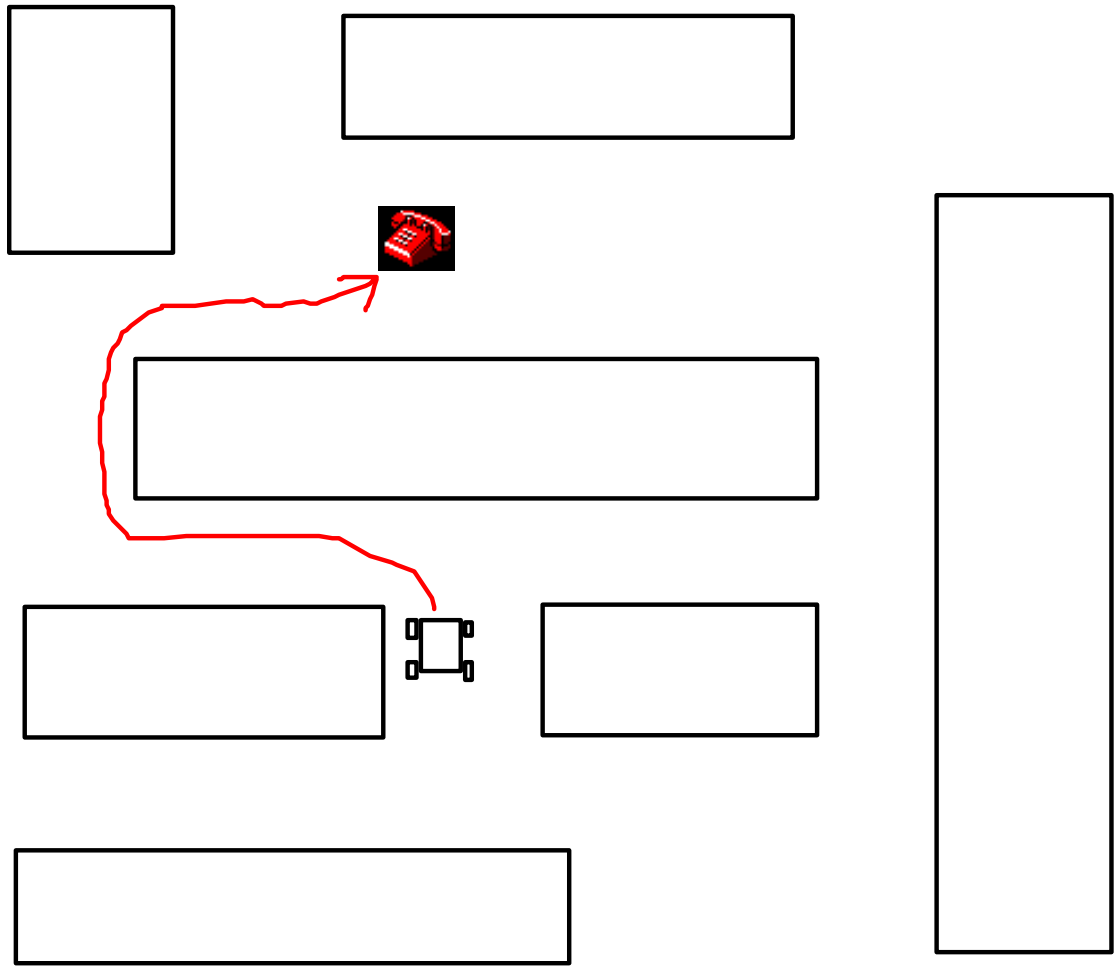
Robotik

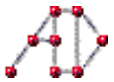
Laserscan-Roboter finden sich besser zurecht.

Doch auch ihnen mangelt es schnell an Orientierungen. In unbekannten Umgebungen bieten sie das beste Beispiel eines On-line Szenarios für geometrische Algorithmen.



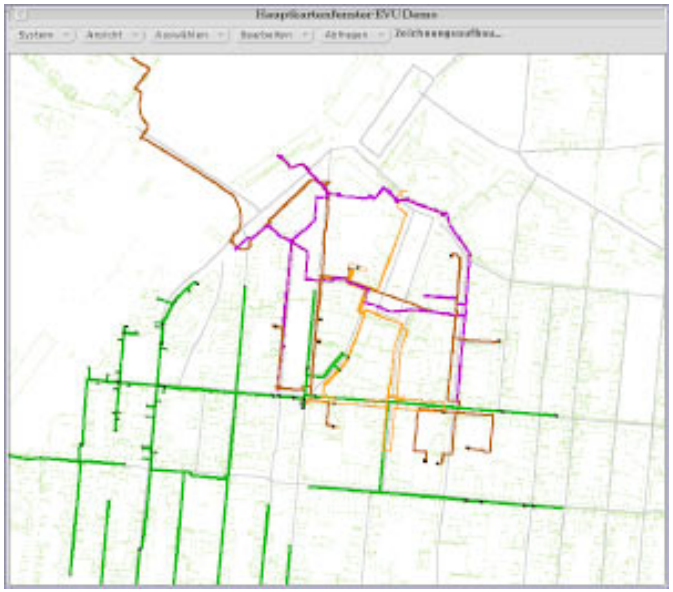
Bewegungsplanung eines Roboters

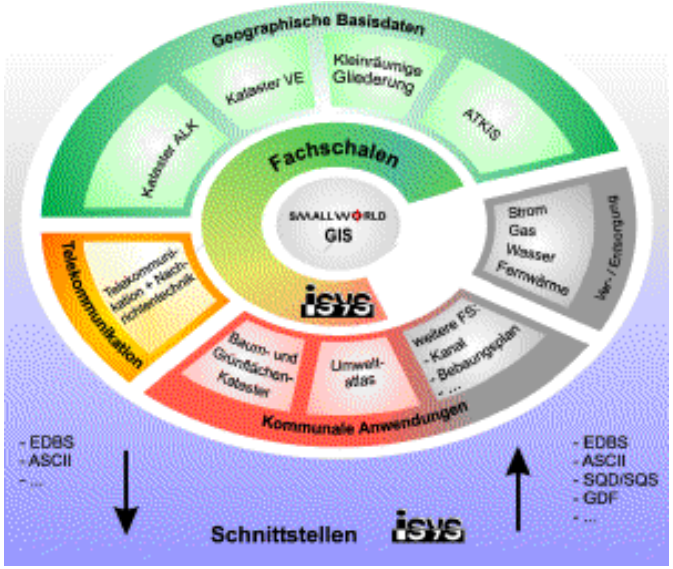




Geographische Informationssysteme

Uni-Offspring
ISYS





Dokumentation, Analyse und Fortschreibung von

Gas-, Kanal- und Telekommunikationsleitungen