Berechnung konvexer Hüllen in 2D

Michael Hoffmann

1. Oktober 2007

Inhaltsverzeichnis

1 Organisation

Language?

Dozenten. Bernd Gaertner und Michael Hoffmann. Übungen Marek. $\rightarrow$ Webseite (siehe Übungsblatt).

Testatbedingungen. In every lecture we provide you with an exercise sheet. You should solve it in written form and return the solutions at the beginning of the subsequent lecture. Solving the exercises in teams is not allowed.

Your solutions will be graded. If you reach at least 80% of all possible points you will get the grade 6.0, 40% of all possible points correspond to the grade 4.0.

Prüfung. There will be an oral exam of 15 minutes during the examination period. Your final grade consists to 50% of the grade for the exam and to 50% of the grade for the exercises.

Literatur.

- Mark de Berg, Marc van Kreveld, Mark Overmars, Otfried Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer, 2000 (2nd edition).
- Franco P. Preparata, Michael I. Shamos, Computational Geometry: An Introduction, Springer, 1985.
- Bernd's Skript aus Berlin von 1996.

Hinweis auf Handapparat in der Informatikbibliothek.

Prerequisites.

- Entwurf und Analyse von Algorithmen (big-Oh, fundamentale Algorithmen und Datenstrukturen wie Sortieren, Suchbäume, etc.)
- Graphen, Lineare Algebra, Elementare Wahrscheinlichkeitstheorie

Mehr Geometrie. (siehe Folie)

2 Berechnungsmodell

Real RAM Model: Jede Speicherzelle speichert eine reelle Zahl. Arithmetische Operationen und Vergleiche in konstanter Zeit. Ebenso ggf. Wurzeln, Logarithmen, analytische Funktionen. Indirektes Addressieren (ganzzahlig). Je nachdem: Floor, ceiling.

Algebraische Berechnungsbäume [Ben-Or 1983]: Berechnung als binärer Baum.

- Knoten mit einem Kind wird eine Operation der Form $+, -, *, /, \sqrt{}$ zugewiesen, die zwei Operanden unter seinen Ahnen im Baum verknüpft (oder eine Konstante oder einen Eingabewert).
- Knoten mit zwei Kindern entsprechen eine Verzweigung der Form > 0, ≥ 0 oder $= 0$.

- In den Blättern stehen die Ergebnisse.

Bemerkung: Basiert jede Verzweigung auf einer linearen Funktion in den Eingabewerten oder Konstanten, so handelt es sich um einen linearen Berechnungsbaum. (entspr. quadratisch, etc.)

3 Preliminaries

Betrachte $P \subset \mathbb{R}^d$, $n \in \mathbb{N}$, $p_i \in P$ und $\lambda_i \in \mathbb{R}$ für $i \in \{1, \ldots, n\}$.
Lineare Hülle

$$\mathrm{lin}(P) := \left\{ q \;\middle|\; q = \sum_{i=1}^{n} \lambda_i p_i \right\}$$

(kleinster linearer Teilraum, der P enthält, Bsp. für $p \in \mathbb{R}^2$: Gerade durch p und den Ursprung)
Affine Hülle

$$\mathrm{aff}(P) := \left\{ q \;\middle|\; q = \sum_{i=1}^{n} \lambda_i p_i \wedge \sum_{i=1}^{n} \lambda_i = 1 \right\}$$

(kleinster affiner Teilraum, der P enthält, Bsp. für $p, q \in \mathbb{R}^2$: Gerade durch p und q)
Konvexe Hülle

$$\mathrm{conv}(P) := \left\{ q \;\middle|\; q = \sum_{i=1}^{n} \lambda_i p_i \wedge \sum_{i=1}^{n} \lambda_i = 1 \wedge \forall\, i \,:\, \lambda_i \geq 0 \right\}$$

(kleinster konvexer Teilraum, der P enthält, Bsp. für $p, q \in \mathbb{R}^2$: Liniensegment durch p und q, $=: \overline{pq}$)

Definition 1 *Eine Menge $P \subseteq \mathbb{R}^d$ heisst* **konvex**, *falls $\overline{pq} \subseteq P$ für jedes Paar $p, q \in P$.*

Observation 1 $\mathrm{conv}(P)$ *ist konvex.*

Proof. Betrachte $p = \sum_{i=1}^{n} \lambda_i p_i$ und $q = \sum_{i=1}^{n} \mu_i p_i$, wobei $\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \mu_i = 1$ und $\lambda_i, \mu_i \geq 0$ für alle $i \in \{1, \ldots, n\}$. (Falls in den Konvexkombinationen unterschiedliche Punkte auftreten, setze den Koeffizienten für den anderen Punkt auf 0.)

Für beliebiges $\alpha \in [0, 1]$ ist dann

$$\begin{aligned} \alpha p + (1-\alpha) q &= \alpha \sum_{i=1}^{n} \lambda_i p_i + (1-\alpha) \sum_{i=1}^{n} \mu_i p_i \\ &= \sum_{i=1}^{n} (\alpha \lambda_i + (1-\alpha) \mu_i) p_i \,, \end{aligned}$$

wobei $\sum_{i=1}^{n} (\alpha \lambda_i + (1-\alpha)\mu_i) = \alpha + (1-\alpha) = 1$ und $\alpha \lambda_i \geq 0$ sowie $(1-\alpha)\mu_i \geq 0$ für alle $i \in \{1, \ldots, n\}$. □

Theorem 1 $\mathrm{conv}(P)$ *ist*

a) die kleinste konvexe Teilmenge des $\mathbb{R}^d$, *die* P *enthält.*

b) der Durchschnitt aller konvexen Obermengen von P.

c) der Durchschnitt aller abgeschlossenen Halbräume, die P *enthalten.*

Insbesondere: Ist P *endlich, so ist* $\mathrm{conv}(P)$ *der Durchschnitt endlich vieler Halbräume, d.h. ein konvexes Polytop. [McMullen-Shephard 1971]*

Theorem 2 (Carathéodory) *Für* $P \subset \mathbb{R}^d$ *und* $q \in \mathrm{conv}(P)$ *existieren* $k \leq d+1$ *Punkte* $p_1, \dots, p_k \in P$, *so dass* $q \in \mathrm{conv}(p_1, \dots, p_k)$.

4 Problemstellung

Konvexe Hülle

Eingabe: $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, $n \in \mathbb{N}$.

Ausgabe: Folge $(q_1, \dots, q_h)$, $1 \leq h \leq n$, der Randpunkte von $\mathrm{conv}(P)$ im Gegenuhrzeigersinn.

Extrempunkte

Eingabe: $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, $n \in \mathbb{N}$.

Ausgabe: Menge $Q \subseteq P$ der Eckpunkte von $\mathrm{conv}(P)$.

Degeneriertheiten. Drei Kollineare Punkte. Sind alle Randpunkte? Nein...

Definition 2 *Ein Punkt* $p \in P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ *ist* **Randpunkt** *von* $P \iff$ *es gibt eine gerichtete Gerade* g *durch* p, *so dass* $P \setminus \{p\}$ *links von* g *liegt.*

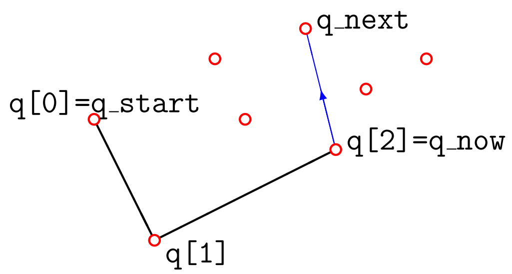
5 Triviale Algorithmen

Teste für jeden Punkt $p \in P$, ob es drei Punkte $q, r, s \in P \setminus \{p\}$ gibt, so dass p im Dreieck mit Eckpunkten q, r und s liegt. Laufzeit $O(n^4)$.

Teste für jedes Paar $(p, q) \in P^2$, ob alle Punkte $P \setminus \{p, q\}$ links der gerichteten Geraden durch p und q (oder auf dem Liniensegment $\overline{pq}$) liegen. Laufzeit $O(n^3)$.

6 Jarvis' Wrap

Suche einen Punkt p_1, der Randpunkt von $\text{conv}(P)$ ist (z.B. ein Punkt mit kleinster x-Koordinate). "Packe" P von p_1 aus "ein", d.h. finde immer jeweils den Punkt, der am weitesten rechts liegt.



Analyse. Für jeden Ausgabepunkt n rightturn Tests $\Rightarrow O(nh)$. (Im worst case $h = n$, d.h. $O(n^2)$.)

Degeneriertheiten.

- Mehrere Punkte mit kleinster x-Koordinate $\Rightarrow$ lexikographische Ordnung: $(p_x, p_y) < (q_x, q_y) \iff p_x < q_x \vee p_x = q_x \wedge p_y < q_y$.
- Mehrere Punkte identisch.
- Drei oder mehr kollineare Punkte $\Rightarrow$ Wähle den Punkt, der jeweils am weitesten vom aktuellen Punkt entfernt ist.

7 Entwurfszyklus Geometrischer Algorithmen

Modellierung: Formulieren der Problemstellung.

Beobachtungen: unmittelbare Folgerungen aus der Problemdefinition $\Rightarrow$ Triviale Algorithmen.

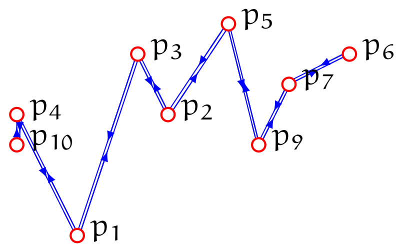
Analyse: Entwurf effizienter Algorithmen, oft für generische Eingaben (allgemeine Lage).

Spezialfälle: Behandlung von Degeneriertheiten (z.B. Kollinearitäten).

Implementation: Berechnung der geometrischen Primitiva (z.B. Orientierungtest).

8 Graham Scan (SLR)

Sortiere Punkte lexikographisch und entferne Duplikate: $(p_1, \ldots, p_n)$.



$$p_{10}\, p_4\, p_1\, p_3\, p_2\, p_5\, p_9\, p_7\, p_6\, p_7\, p_9\, p_5\, p_2\, p_3\, p_1\, p_4\, p_{10}$$

Solange $\exists$ ein (aufeinanderfolgende) Tripel (p, q, r) mit q links von oder auf der gerichteten Geraden $\overrightarrow{pr}$, entferne q aus der Folge.

Analyse.

1. Sortieren und Entfernen doppelter Punkte: $O(n \log n)$.
2. Zu Beginn: $2n - 2$ Punkte(/Kanten); am Ende: h Punkte. $\Rightarrow$ $2n - h - 2$ shortcuts/positive rightturn Tests. Dazu kommen noch $2n - 2$ negative Tests. Also insgesamt $4n - h - 4$ rightturn Tests.

Also $O(n \log n)$ Zeit insgesamt.

Grundsätzliche Idee: [Graham '72]

9 Untere Schranke

Sortieren $\Omega(n \log n)$ (algebr. Berechnungsungsbäume).

Gegeben n Zahlen $x_1, \ldots, x_n$. Berechne (in O(n) Zeit) die Punkte $p_i = (x_i, x_i^2)$, $1 \leq i \leq n$. Die konvexe Hülle der p_i ergibt die sortierte Reihenfolge.

Gilt auch für Berechnung der Extrempunkte im Modell der algebr. Berechnungsungsbäume [Avis '79 linear, Yao '81 quadratisch, Ben-Or '83 algebr.].

10 Chan's Algorithmus [1996]

Divide. Eingabe: Menge $P \subset \mathbb{R}^2$ von n Punkten und $1 \leq H \leq n$.

1. Teile P in $k = \lceil n/H \rceil$ Mengen $P_1, \ldots, P_k$ mit $|P_i| \leq H$.
2. Berechne $\mathrm{conv}(P_i)$ für alle i, $1 \leq i \leq k$.

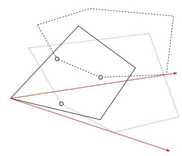
3. Berechne daraus H Punkte von $\text{conv}(P)$. (*conquer*)

Analyse: Schritt 1 braucht $O(n)$ Zeit. Schritt 2 kann mittels SLR in $O(H \log H)$ Zeit pro P_i, d.h. $O(n \log H)$ Zeit insgesamt, gelöst werden.

Conquer.

1. Finde den lexikographisch kleinsten Punkt in $\text{conv}(P_i)$ für alle i, $1 \leq i \leq k$.
2. Beginnend beim lexikographisch kleinsten Punkt aus P finde die ersten H Punkte von $\text{conv}(P)$ im Gegenuhrzeigersinn (Jarvis' Wrap auf den Folgen $\text{conv}(P_i)$).

Bestimme hierzu in jedem Schritt die Tangentialpunkte vom aktuellen Punkt von $\text{conv}(P)$ an die Polygone $\text{conv}(P_i)$, $1 \leq i \leq k$, durch lineare Suche.



Beobachtung: Alle bei der Suche übersprungenen Punkte (im Bild oben markiert) können nicht Randpunkte von P sein.

Analyse: erster Schritt in $O(n)$ Zeit. Im zweiten Schritt für jeden Punkt in $\text{conv}(P)$ Zeit $O(k + \#\text{übersprungener Punkte})$, also insgesamt $O(kH + n) = O(n)$.

Suche nach H. Ruf den Algorithmus mit Parameter $H = \min\{2^{2^t}, n\}$, für $t = 1, \ldots$ auf, bis im Conquer-Schritt alle Randpunkte von P gefunden werden (d.h., der Wrap kehrt zum Startpunkt zurück). (Doppelt exponentielle Suche).

Analyse: Sei 2^{2^s} der letzte Parameter, mit dem der Algorithmus aufgerufen wird. Da der Aufruf davor nicht erfolgreich war, folgt $2^{2^{s-1}} < h$, d.h. $2^{s-1} < \log h$, wobei h die Randpunkte von P bezeichne. Die Gesamtkosten sind damit

$$\sum_{i=1}^{s} cn \log 2^{2^i} = \sum_{i=1}^{s} cn2^i = cn(2^{s+1} - 2) < 4cn \log h = O(n \log h).$$

Bemerkungen: erster $O(n \log h)$ Algorithmus von Kirkpatrick und Seidel [1986] mit entsprechender unterer Schranke (sogar für Extrempunktanzahl Verifikation: geg. P und H, ist $|\text{conv}(P)| = h$?)