

Konvexe Hüllen

Inkrementeller Algorithmus für die konvexe Hülle

Dabei heißt inkrementeller Algorithmus, daß wir Punkte p_1, \dots, p_n in dieser Reihenfolge nach und nach hinzufügen. Dieses Vorgehen wird in der Literatur auch online-Algorithmus genannt. In den vorangegangenen Algorithmen mußten wir vorher die gesamte Punktmenge kennen, dies ist hier nicht der Fall. Wir erhalten jedoch folgendes aufrecht: Wie beim Graham-Scan-Algorithmus betrachten wir Strahlen, die von einem Punkt p_0 im Innern der konvexen Hülle zu allen bisher eingefügten Punkten verlaufen. Weiter nehmen wir an, daß sich die Punkte in allgemeiner Lage befinden, das heißt, daß keine drei Punkte auf einer Geraden liegen. Ist dies nicht der Fall, d.h. drei Punkte liegen auf einer Geraden, ist es trotzdem machbar. Jedoch wollen wir diesen Fall im weiteren nicht berücksichtigen.

1) Initialisierung:

Dreieck p_1, p_2, p_3

$p_0 := \frac{1}{3}(p_1 + p_2 + p_3)$, liegt im Inneren der konvexen Hülle (siehe Abbildung 1)

for $i = 4, \dots, n$

- 2) Bestimme den Kegel zwischen zwei Strahlen, der p_i enthält,
die zugehörige Kante von $CH(p_1, \dots, p_{i-1})$ sei e_i
- 3) if $p_i \in$ Äußeren von $CH(p_1, \dots, p_{i-1})$
füge p_i hinzu,
suche ausgehend von Enden von e_i die Tangenten von p_i an $CH(p_1, \dots, p_{i-1})$,
entferne das Zwischenstück $p_j \dots p_k$ aus der konvexen Hülle (einschließlich Strahlen)
- 4) ersetze durch die Kanten: p_i nach p_j und p_i nach p_k
füge den Strahl $\overline{p_0 p_i}$ ein

Laufzeit:

- (1) benötigt $O(1)$ Zeit.
 - (2) Durch Binärsuche zwischen den Strahlen, falls diese in einem balancierten binären Suchbaum bezüglich der Ordnung im Uhrzeigersinn gegeben sind, benötigen wir hier $O(\log n)$ Zeit.
 - (3),(4) Insgesamt wird jeder Punkt höchstens einmal eingefügt und höchstens einmal entfernt.
das kostet jeweils $O(\log n)$ Zeit.
- Insgesamt benötigt der Algorithmus also $O(n \log n)$ Zeit.

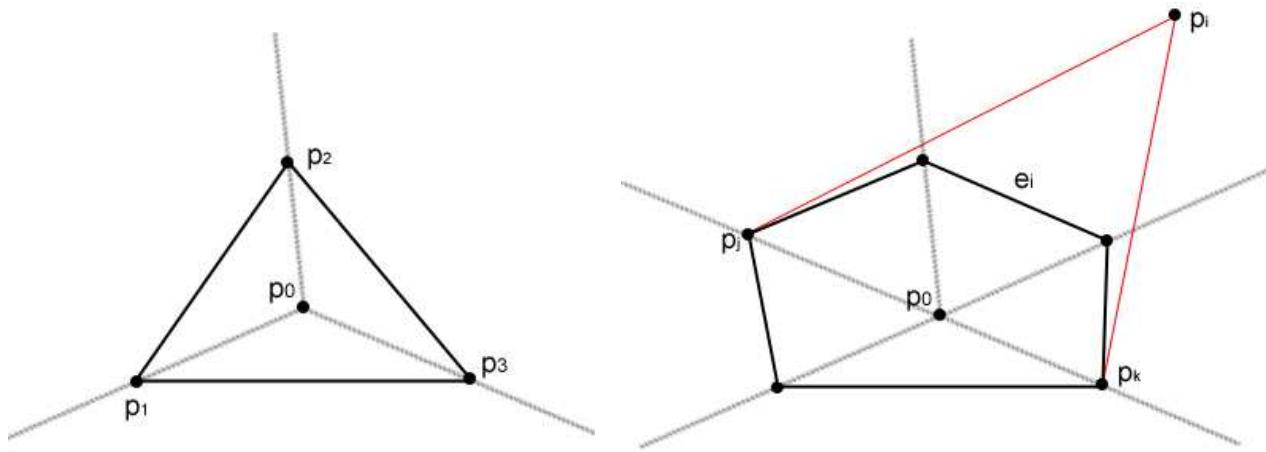


Abbildung 1: links: Schritt 1 rechts: Schritt 3 des inkrementellen Algorithmus für die konvexe Hülle

Kapitel 2: Voronoi-Diagramme und Suchen in ebenen Unterteilungen

Beispiel. Ein Beispiel ist das sogenannte post-office-problem.

Für dieses Problem stelle man sich eine Menge von Punkten als Postämter vor. Nun möchte man, egal wo man sich in diesem Augenblick befindet, wissen welche Post die Nächste ist.

Wir verarbeiten also eine Menge S von Punkten in eine Datenstruktur vor, so daß effizient zu einem gegebenen Punkt der nächstgelegene aus S bestimmt werden kann. Der Abstand ist hier der Euklidische Abstand $\|x - y\|$, wobei $\|a\| = \sqrt{a_x^2 + a_y^2}$ für $a = (a_x, a_y)$

Definition. Sei S Menge von n Punkten $\{p_1, \dots, p_n\}$ in der Ebene. Dann heißt

$$VR(p_i) = \{p \in \mathbb{R}^2 \mid \|p - p_i\| < \|p - p_j\|, j \neq i\}$$

die Voronoi-Region von p_i .

Beispiel siehe Abbildungen 2 und 3

Bemerkung. Offensichtlich ist $VR(p_i)$ der Durchschnitt p_i von offenen Halbebenen

$$H_j = \{p \mid \|p - p_i\| < \|p - p_j\|\}, \text{ also } VR(p_i) = \bigcap_{j \neq i, j=1, \dots, n} H_j.$$

Damit ist $VR(p_i)$ ein konvexes Gebiet, berandet durch endlich viele Strecken oder Strahlen.

Definition. Voronoi-Kanten sind die Kanten der Voronoi-Regionen.

Voronoi-Knoten oder Voronoi-Ecken sind die Ecken der Voronoi-Regionen. Insgesamt bilden die Voronoi-Regionen, Voronoi-Kanten und Voronoi-Ecken das Voronoi-Diagramm, eine Partition der Ebene.

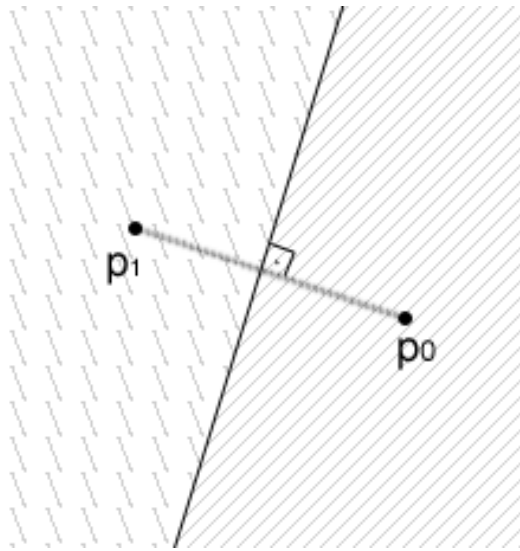


Abbildung 2: Bsp. mit $n=2$ Punkten, somit entstehen zwei Halbebenen

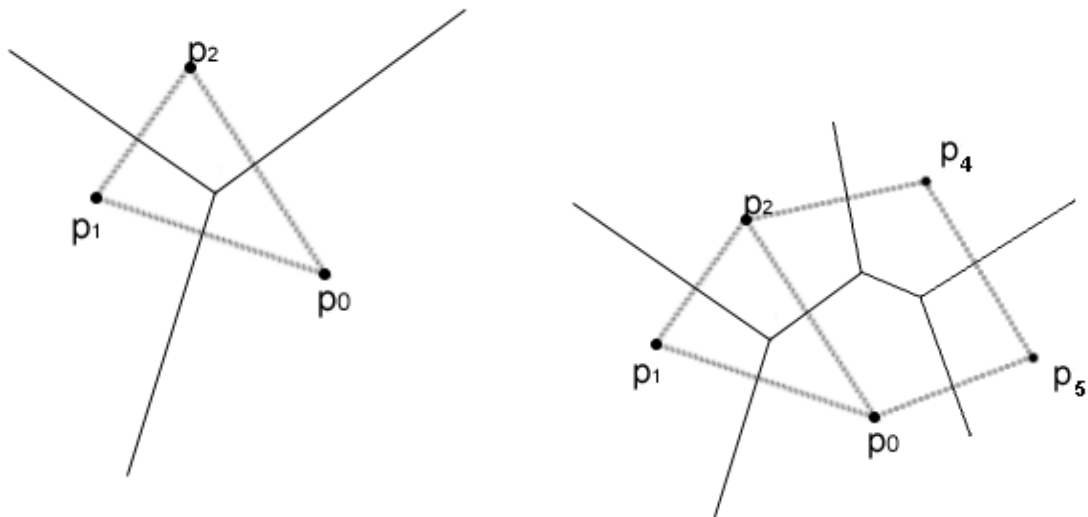


Abbildung 3: links: $n=3$ Punkte rechts: $n=5$ Punkte

Anwendungen von Voronoi-Diagrammen gibt es nicht nur beim post-office-Problem, sondern auch in diversen anderen Bereichen wie Robotik, Biologie, Mustererkennung,...

Bemerkung. 1. Punkte im Innern von Voronoi-Kanten haben gleichen Abstand zu 2 Punkten in S , und einen größeren Abstand zu allen anderen

2. Voronoi-Knoten sind genau die Punkte, die gleichen Abstand zu 3 oder mehr Punkten in S haben, und einen größeren zu allen anderen

Lemma. Für $p \in S$ gilt:

$VR(p)$ ist unbeschränkt genau dann, wenn p auf dem Rand der konvexen Hülle von S liegt, also wenn $p \in \partial CH(S)$.

Beweis. " \Rightarrow ": Falls $VR(p)$ unbeschränkt ist, dann existiert ein Strahl s von p ausgehend, der ganz in $VR(p)$ liegt. Dies gilt wegen der Konvexität von $VR(p)$. Sei die Gerade g durch p senkrecht auf s . (siehe Abbildung 4). Dann liegt ganz S auf einer Seite von g , und zwar in der, die s nicht enthält, denn: Angenommen, dies ist falsch, das heißt, es existiert ein Punkt $q \in S$, der in der gleichen offenen Halbebene wie s ist. Dann würde die Mittelsenkrechte zwischen q und p den Strahl s schneiden. Punkte jenseits des Schnittpunktes auf s wären näher an q als an p , was ein Widerspruch ist, da s ganz in $VR(p)$ liegt.

□

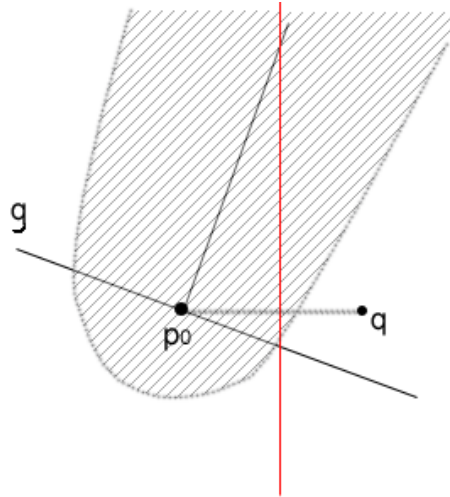


Abbildung 4:

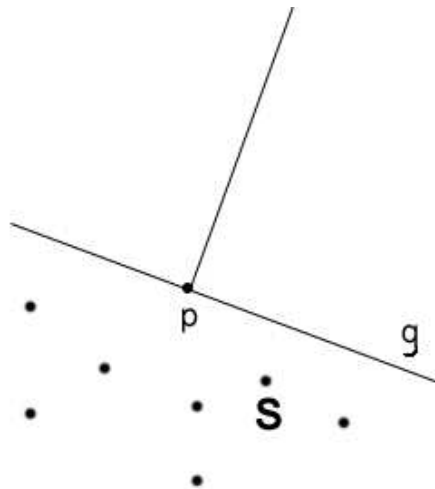


Abbildung 5:

Beweis. " \Leftarrow ": Aus $p \in \partial CH(S)$ folgt, daß es eine Gerade g durch P gibt, so daß alle Punkte von S in der abgeschlossenen Halbebene auf der anderen Seite von p liegen. Sei s der Strahl orthogonal zu g , von p ausgehend in die andere Halbebene. (siehe Abbildung 5). Dann folgt, daß jeder Punkt auf s näher an p als an einem der anderen Punkte von S ist. Also gehört s zur Voronoi-Region von p , und $VR(p)$ ist somit unbeschränkt.

□

Wie groß (in Abhängigkeit von $n = |S|$) ist das Voronoi-Diagramm von S ?

Anzahl der Regionen (oder auch “Zellen”): n .

Anzahl der Kanten: $\leq \binom{n}{2} \approx \frac{1}{2}n^2$

Anzahl der Knoten: $\leq \binom{n}{3} \approx \frac{1}{6}n^3$, d.h. jeder ist durch ein Tripel von Punkten aus S bestimmt.

Die Anzahl der Kanten und Knoten wurde hier pessimistisch abgeschätzt.

Einschub. planare Graphen

Definition. Ein Graph $G = (V, E)$ heißt planar (plättbar), wenn er eine ebene Einbettung besitzt, das heißt, es gibt eine Abbildung $V \rightarrow \mathbb{R}^2$, $E \rightarrow$ einfache Kurven im \mathbb{R}^2 , die verträglich ist mit den Endpunkten, so daß sich keine zwei Bilder von Kanten (im Innern) schneiden.

Nach dieser Definition sind die ersten beiden Graphen in Abbildung 6 planare Graphen. Der letzte Graph ist nicht planar.

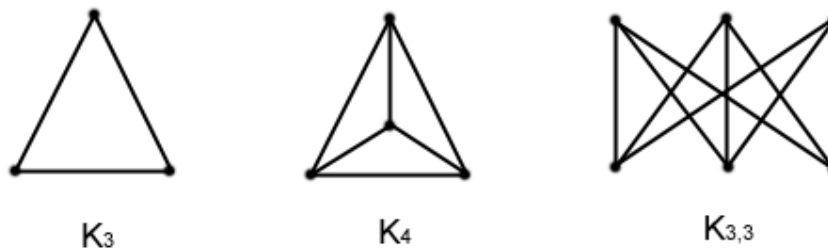


Abbildung 6: