

→ 3.4.2.10 Spezialfälle:

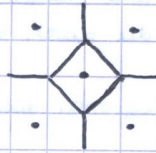
1) "normales" Voronoi-Diagramm: $VD(S) = VD_1(S)$. [closest point VD]

2) VD_{n-1} [farthest point VD]

↑ Voronoi-Regionen sind hier Mengen von Pkten, die von einem Ort weitest entfernt sind als von allen anderen.

Ab jetzt: $VD(S) = VD_1(S)$

Typisches Bsp:



→ 3.4.2.11 Lemma:

a) \forall Orte $x \in S$ gilt: $VR(x)$ ist unbeschränkt $\Leftrightarrow x$ ist Ecke von $CH(S)$ oder auf dem Rand von $CH(S)$

b) Voronoi-Diagramm für n Orte hat:

$\leq 2n-4$ Knoten

$\leq 3n-6$ Kanten.

(?)

Beweis:

a) \Rightarrow " sei $VR(x)$ unbeschränkt.

z.z. x ist Ecke von $CH(S)$ oder auf Rand von $CH(S)$

Ann.: nein.

$VR(x)$ konv u. unbeschr. n.V. $\Rightarrow \exists$ Strahl s , der in x startet

hier notwendig, dass x nicht auf Kante! \rightarrow und ganz in dieser Region $VR(x)$ verläuft.

Ann $\Rightarrow x$ im Inneren von $CH(S)$

S endlich \Rightarrow Strahl s schneidet den Rand von $CH(S)$ in einer Kante (y,z) oder in Pkt $z=y$. (da $CH(S)$ auch dann endlich)

$\exists p \in s$ (genügend weit entfernt von x), der näher zu y oder z ist als zu x

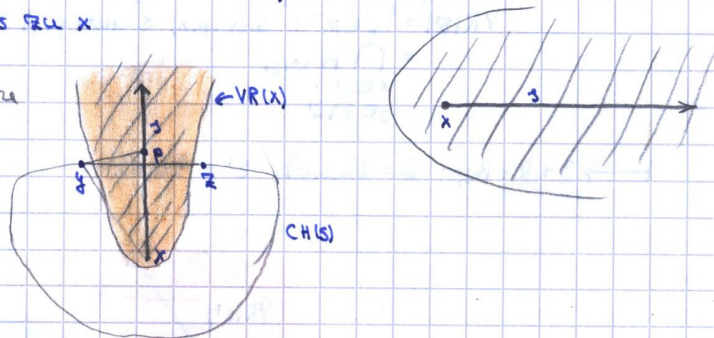
$\Rightarrow z$ zu $p \in VR(x)$

dh. z zu $VR(x)$ ist näher

ist $VR(x)$

dh. z zu $VR(x)$ ist

unbeschränkt.



\Rightarrow Beh

" \Leftarrow Sei nun x Ecke von $CH(S)$.

Betrachte Kegel K zwischen den Senkrechten auf dem zu x betrachteten Kanten.

Alle Pkte in K liegen näher zu x als zu allen anderen Orten.

$\Rightarrow K \subseteq VR(x)$

$\Rightarrow VR(x)$ unbeschränkt

(da K unbeschränkt)

Sei nun x auf dem Rand von $CH(S)$

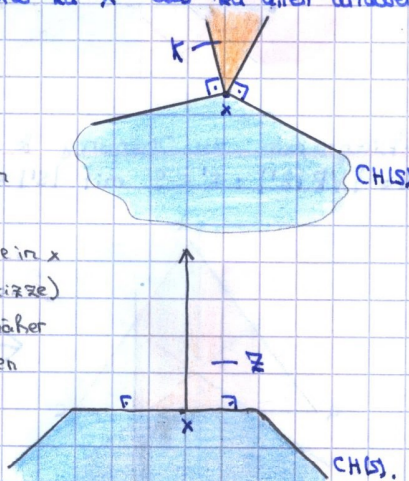
Betrachte senkrechte Gerade die in x startet und hoch zeigt (wie Skizze)

Alle Pkte auf Gerade G liegen näher zu x als zu allen anderen Orten

$\Rightarrow G \subseteq VR(x)$

$\rightarrow VR(x)$ unbeschränkt

(da G unbeschränkt)

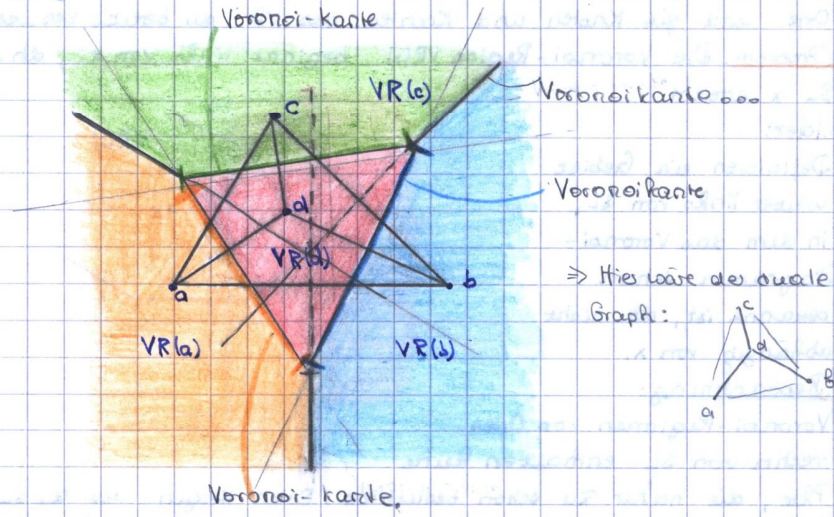


b). VD ist ein Graph, das \mathbb{R}^2 in Voronoi-Regionen unterteilt.

(\rightarrow Voronoi-Knoten + Voronoi-Kanten).

Sei $G=(V,E)$ der duale Graph zu VD(s), d.h.

$V=S$ (allg. Flächen (Regionen)) und $(x,y) \in E \Leftrightarrow VR(x)$ und $VR(y)$ haben gemeinsame Voronoi-Kante.



G ist ein planarer Graph und jede Kante $e \in E$ entspricht genau einer Voronoi-Kante (siehe Konstruktion von G).

(c,d) entspricht grün, (a,d) entspricht orange und (b,d) entspricht blau.

Graphentheorie \Rightarrow Jeder planare Graph mit n Knoten hat maximal $3n-6$ Kanten.

\Rightarrow Voronoi-Kanten $\leq 3n-6$.

Außerdem hat jeder Voronoi-Knoten mind. Grad 3 (da er durch mind. drei Orte definiert wird, von denen er gleiche Distanz hat).

$\forall v \in V: \text{Grad}(v) \geq 3$

$\Rightarrow \forall v \in V, |v|=1: \text{Grad}(v) \geq 3$

$\Rightarrow \forall v \in V: 3 \cdot |v| \leq \text{Grad}(v)$

$\Rightarrow \sum_{v \in V} 3 \cdot |v| \leq \sum_{v \in V} \text{Grad}(v)$

$\Rightarrow 3 \cdot \# \text{Knoten} \leq \sum_{v \in V} \text{Grad}(v)$

\Rightarrow Es gilt: $\sum_{v \in V} \text{grad}(v) \geq 3 \cdot \# \text{V-Knoten} \wedge \sum_{v \in V} \text{grad}(v) = 2 \cdot |E|$

$\Rightarrow 3 \cdot \# \text{V-Knoten} \leq \sum \text{grad}(v) = 2 \cdot |E| \leq 2 \cdot (3n-6) = 6n-12$

$\Rightarrow \# \text{V-Knoten} \leq 2n-4$

3.4.2.12 Bemerkungen:

1). VD für n Orte hat lineare Größe in n .

2). Sei $G=(V,E)$ dualer Graph zu VD(s), d.h. $V=S$ und $(x,y) \in E \Leftrightarrow VR(x)$ und $VR(y)$ haben gemeinsame Voronoi-Kante. \checkmark

G heißt Delaunay-Triangulierung.

Eigenschaften: Jeder Umkreis eines Dreiecks enthält keinen Ort in seinem Inneren.

Bea: Gemeinsame Rand von zwei VR'nen heißt Voronoi-Kante

Und Endpunkte von Voronoi-Kanten sind die Voronoi-Knoten

$\Rightarrow v$ Vor. Knoten $\Rightarrow \exists$ Orte a, b, c mit v Mittelpunkt von Kreis durch a, b, c



\Rightarrow Besser: andere Definition als 3.4.2.5!

B4.3. Konstruktion von Voronoi-Diagrammen durch einen Plane Sweep Algorithmus.

Zunächst: allg. Lage der Orte

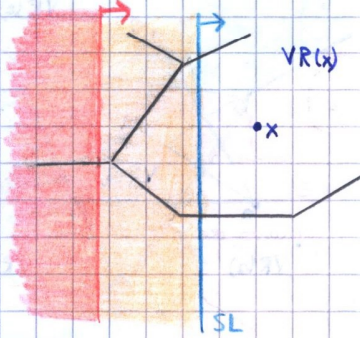
- keine 4 Orte liegen auf einem gemeinsamen Kreis
- paarweise verschiedene x-Koordinaten aller Orte und Events.

→ 3.4.3.1. Ziel: Bewege Senkrechte SL von links nach rechts über die Menge der Orte und gib Knoten und Kanten des VD ab bereits besuchten Orte aus.

→ 3.4.3.2. Problem: Die Voronoi-Region $VR(x)$ beginnt links von x , d.h. bevor SL x erreicht.

→ 3.4.3.3. Idee:

Definieren ein Gebiet weiter links von SL, in dem das Voronoi-Diagramm schon bekannt ist, d.h. nicht abhängig von x .



→ 3.4.3.4. Beobachtung:

Voronoi-Regionen von Orten rechts von SL enthalten keine Pkte, die näher zu schon besuchten Orten liegen als zu SL. Denn sonst würde ein solcher Pkt (unabh. von SL) näher zu einem anderen Ort liegen. Abstand zu SL ist nicht größer als Abstand zum nächsten Ort.

→ 3.4.3.5. Frage:

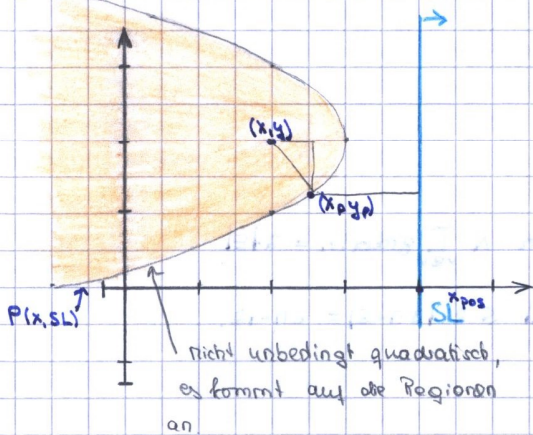
Betrachte das Gebiet $L := \{p \in \mathbb{R}^2 : \text{dist}(p, x) \leq \text{dist}(p, SL)\}$ für einen schon besuchten Ort x .

⇒ Voronoi-Diagramm von S im Gebiet L hängt nicht von Ort x rechts von SL ab.

Welche Form hat L ?

→ 3.4.3.6. Beispiele: $S' \leftarrow$ Menge der besuchten Orte.

1) S' besteht aus einem Ort x .



nicht unbedingt quadratisch, es kommt auf die Regionen an

→ Inneres der Parabel $P(x, SL)$

ist die Menge der Pkte mit gleicher Distanz zu x und SL .

(x, y) und SL bekannt

Wie sieht L aus?

→ Es muss gelten $\forall (x_p, y_p) \in L$:

$$(x_p - x_{pos})^2 \leq (x_p - x)^2 + (y_p - y)^2$$

weil es muss eigentlich gelten:

$$x_p - x_{pos} = \sqrt{(x_p - x)^2 + (y_p - y)^2}$$

→ $(x, y) = (2, 2)$, $SL \equiv 4$ bekannt

$$\Rightarrow (x_p - 4)^2 = (x_p - 2)^2 + (y_p - 2)^2$$

$$\Leftrightarrow x_p^2 - 8x_p + 16 = x_p^2 - 4x_p + 4 + y_p^2 - 4y_p + 4$$

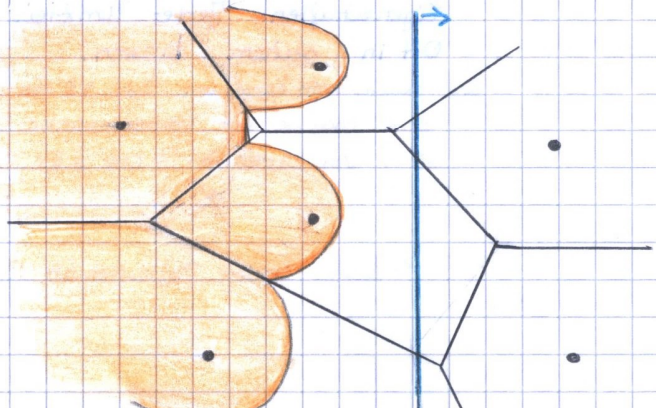
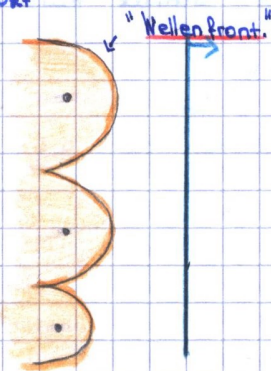
$$\Leftrightarrow -4x_p + 8 = y_p^2 - 4y_p$$

$$\Leftrightarrow 4x_p = -y_p^2 + 4y_p + 8$$

$$\Leftrightarrow x_p = -\frac{1}{4}y_p^2 + y_p + 2$$

→ dies ist eine Parabelgleichung mit der Variablen y

2) Im Allgemeinen wird L durch eine Folge von Parabelbögen nach rechts begrenzt



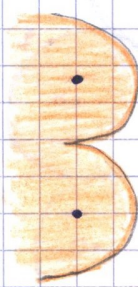
→ 3.4.3.7 Beobachtung: Schnittpunkte von benachbarten Parabeln in der Wellenfront überstreichen (33) die Kanten von V(DS).

→ 3.4.3.8 Idee: Verwalte die Wellenfront der Parabelbögen in einer Y-Struktur.

→ 3.4.3.9 Fragen: An welchen Positionen (Transaktionspunkten / Events) der SL ändert sich die Y-Struktur strukturell?
→ Welche Events gibt es?

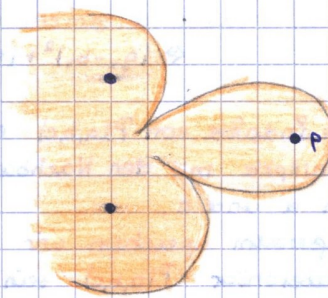
→ 3.4.3.10 Zwei Arten von Events:

1) SL überstreicht neuen Ort p vorher:



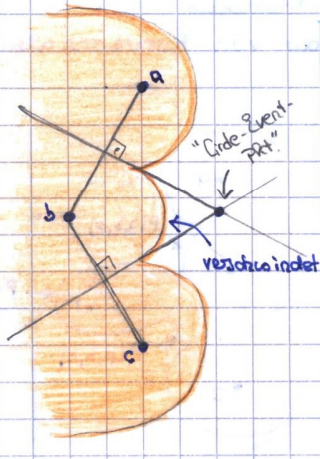
"Site Event"

nachher:



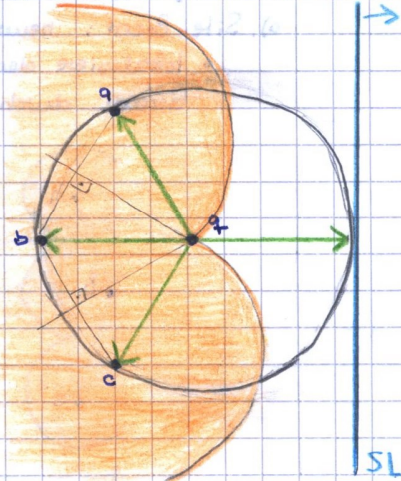
2) Ein Parabelbogen verschwindet aus der Wellenfront. "Circle Event"

vorher:



nachher:

bea: q gehört nicht zu S (i.d.) Es ist nur ein Schnittpunkt von VR'en und heißt Circle-Event-Pkt.



3) drei Parabeln schneiden sich in einem Pkt

↑ Hier entsteht ein neuer Voronoi-Knoten q = Mittelpunkt des Kreises durch a, b, c SL ist zu diesem Zeitpunkt Tangente an den Kreis (durch a, b, c).

→ 3.4.3.11 Lemma: Jeder Pkt auf der Kante von V(DS) kommt während des Sweep als Schnittpunkt zweier in der Y-Struktur benachbarter Parabelbögen vor.

(hier kontinuierlicher Sweep)

Beweis:

Sei e beliebige Voronoi-Kante von V(DS) und u ein beliebiger Pkt auf e. e trennt zwei Voronoi-Regionen $VR(p)$ und $VR(q)$. u liegt näher zu p und q (gleiche Distanz d) als zu allen anderen Orten.

Kreismittepunkt u enthält im Inneren keinen Ort.

Nun betrachte den Zeitpunkt des Sweep, in dem SL rechte Tangente an diesem Kreis K ist

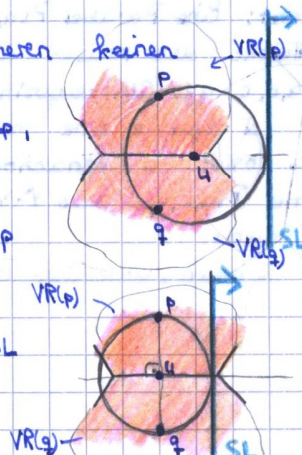
a) u hat den gleichen Abstand zu p und SL

⇒ u liegt auf Parabel von p

b) u hat gl. Abstand zu q und SL

⇒ u liegt auf Parabel von q

⇒ Beh.



3.4.3.12 Implementierungsdetails:

1. Darstellung der Parabelbögen:

Ein Parabelstück wird durch drei Orte a, b, c und die aktuelle Position x_{pos} der Sweepline definiert.

$P(a, b, c)$ bezeichnet Parabelstück von b , das an Parabeln von Ort a und von Ort c grenzt.

Gleichung der Parabel:

$$q \in P(a, b, c) \Leftrightarrow \text{dist}(q, b) = \text{dist}(q, SL)$$

auf Parabelstück, gemeint auf dem Rand $\Rightarrow "="$

$$\Leftrightarrow (q_x - b_x)^2 + (q_y - b_y)^2 = (q_x - x_{pos})^2$$

Die Endpunkte von $P(a, b, c)$ ergeben sich als Schnittpkte mit den Nachbarparabeln, dh. von a und c .

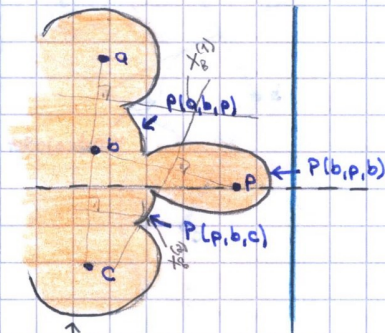
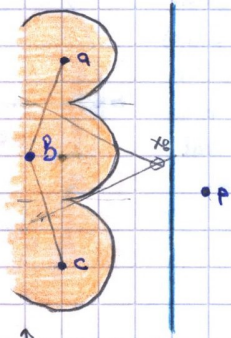
Außerdem speichern wir mit jedem Parabelstück $P(a, b, c)$ sein Circle-Event ab , dh. die Position von SL , bei der $P(a, b, c)$ aus Wellenfront verschwindet.

Aktionen: (Eventbehandlung).

a). Site-Event: neuer Ort p erreicht.

- lokalisieren den Parabelbogen $P(a, b, c)$, der von horizontalen Gerade durch p geschnitten wird.

i.d. Y-Struktur



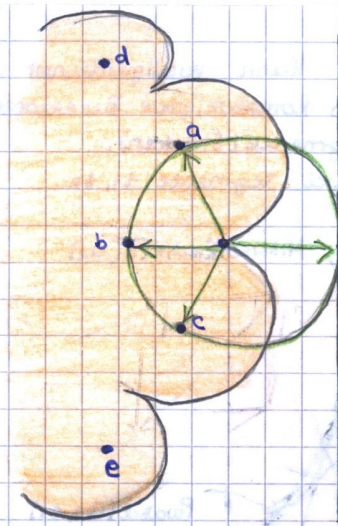
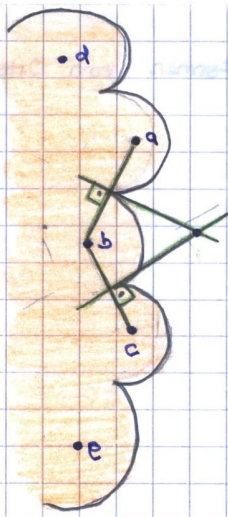
Y-Str.: In der Y-Str. $= \{P(-, a, b), P(a, b, c), P(b, c, -)\}$

$P(a, b, c)$ löschen aus Y-Str. und drei neue Bögen in die Y-Str. einfügen
 \Rightarrow Y-Str. $= \{P(-, a, b), P(a, b, p), P(p, b, c), P(b, c, -)\}$

- lösche $P(a, b, c)$ aus Y-Str.
- lösche Circle-Event ^{-PRt} von $P(a, b, c)$ aus X-Str.
- füge 3 neue Bögen in die Y-Str. ein
- berechne die Circle-Events ^{-PRt} der neuen Parabelstücke und füge sie in die X-Str. ein.

b). Circle-Event: Eine Parabel $P(a, b, c)$ verschwindet.

- Gib Mittelpkt des Kreises durch a, b, c als Voronoi-Knoten aus mit a, b, c gg. Uhrzeigersinn. (berechne zugleich Mittelpunkte von ab und bc und zeichne sie aus)
- Seien d und e Orte mit Parabelbögen, die mit Parabel von a bzw. c benachbart sind.
- Entferne $P(a, b, c)$ aus Y-Struktur und lösche den bearbeiteten Circle-EventPkt aus der X-Struktur
- berechne die neuen Circle-Event-Pkte



$\rightarrow P(d,a,b) \rightsquigarrow P(d,a,c)$ und $P(b,c,e) \rightsquigarrow P(a,c,e)$.

$X = \{C(d,a,b), C(a,b,c), C(b,c,e)\}$

$X = \{C(d,a,c), C(a,c,e)\}$

2) Initialisierung

3) Geometrische Primitive

a) Schnitt von Parabeln

b) Vergleichsoper für Ordnung der Parabeln in Y -Struktur für beliebigen Suchbaum

Y . locate(Y)

Y . insert($P(a,b,c)$)

Y . delete(...)

4) Konstruktion des Voronoi-Diagramms aus Folge der Circle-Events.

\rightarrow 2), 3), 4) das alles in Übungsaufgaben.

\rightarrow 3.4.3.13. Ausgabe:

Für jedes Circle-Event geben wir einen Mittelpunkt als Voronoi-Knoten v aus und die Orte auf dem Kreis a, b, c gegen Uhrzeigersinn.

(Können mehr als drei sein bei degenerierten Eingaben)

d.h. wir geben eigentlich die Dreiecke der Delaunay-Triangulierung aus (dualer Graph zum Voronoi-Diagramm).

\rightarrow 3.4.3.14. Übung:

1) Berechne aus dieser Ausgabe eine explizite Darstellung des Voronoi-Diagramms (d.h. Voronoi-Kanten)

2) Modifizieren sie den Alg. so, dass die Voronoi-Kanten direkt ausgegeben werden.

3) Degenerierte Eingaben

Circle-Event: mehr als drei Orte auf einem Kreis \Rightarrow mehr als ein Bogen verschwindet.

4) Alle geometrischen Primitive:

• Schnitt von Parabeln

• Vgl. von Parabeln in Y -Struktur.

\rightarrow können alle in 0(1) berechnet werden.

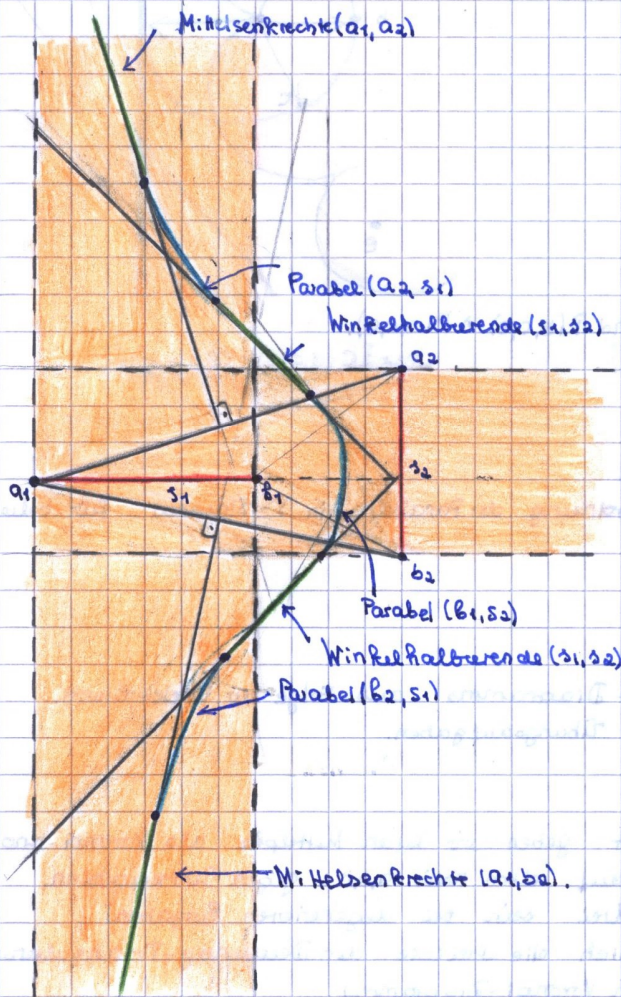
\rightarrow 3.4.3.15 Satz: Das Voronoi-Diagramm von n punktförmigen Orten kann durch plane-Sweep in Zeit $O(n \log n)$ berechnet werden.

Bew: Alle Operationen auf X - bzw. Y -Str. können in Zeit $O(\log n)$ ausgeführt werden, wenn sie durch binäre Suchbäume realisiert werden.

Außerdem hat VD lineare Größe in Zahl der Orte.

→ 3.4.3.16 Bem: Diese Alg. kann verallgemeinert werden für andere Formen von Orten, Dazu muss man komplexere Bisektoren betrachten.

→ 3.4.3.17 Bsp: Liniensegmente (Strecken).
Bisektor für zwei Segmente s_1, s_2 :



Bisektor von zwei Strecken ist eine Kurve, die sich aus Parabelbögen, Mittelsenkrechten und Winkelhalbierenden zusammensetzt.

Je nach Lage und Länge der Strecken können Kurvenarten fehlen.

→ 3.4.3.18 Bemerkung: z.B. Winkelhalbierende oder so nicht vorhanden.

Anm. Segmente schneiden sich nicht (sonst Zerlegung an Schnittpunkten)

Prinzipiell funktioniert die Sweep-Alg. zur Berechnung des VD von Segmenten genauso wie für Pkte.

Lediglich geometrische Primitive sind komplexer: Schnitt & Ordnung von Bisektoren.

3.4.4 Planar point Location:

→ 3.4.4.1 Aufgabe: Finde für beliebigen Pkt $p \in \mathbb{R}^2$ die Voronoi-Region, die p enthält. Dann ist der Ort dieser Region, der p am nächsten liegende Ort.

→ 3.4.4.2 Idee der Vorverarbeitung:

Verbrauche $O(n \log n)$ für Konstruktion von VD, um danach (viele) Anfragen effizient beantworten zu können.

→ 3.4.4.3 Ziel: Laufzeit $O(\log n)$ pro Frage.

Frage: ab wieviel Abfragen lohnt es sich VD zu konstruieren?

Abfragen: $= M < n \Rightarrow$ Konstruktion v. VD + anschließend Streifenmethode kostet für M Abfragen:

$$O(n \log n) + M \cdot O(\log n) = O(n \log n) + O(M \cdot \log n) = O(n \log n) \left. \vphantom{O(n \log n)} \right\} \text{ Falls } M < n, \text{ besser kein VD aufbauen!}$$

Suche nach Region durch Lineare Suche (ohne VD) kostet für M Abfragen: $O(M \cdot n) = O(n^2)$

Für $M > n \Rightarrow$ 1 Mögl. Kostet: $O(n \log n) + O(M \log n) = O(M \log n) \Rightarrow$ Besser VD aufbauen! (weil $\log n < M$).
2 Mögl. Kostet: $O(Mn) = O(M^2)$