Vorlesung

Geometrische Algorithmen

Generierung von Nicht-uniformen Netzen

Sven Schuierer

Überblick

1. Anwendung
2. Anforderungen an Netze
3. Quadrantenbäume
   - Quadrantenbäume für Punktemengen
   - Bestimmung von Nachbarn
   - Balancierung von Quadrantenbäumen
4. Von Quadrantenbäumen zu Netzen

1 Anwendung

Beispiel:

Wärmeleitung in Platinen für elektrische Schaltkreise

Methode der finiten Elemente:

- Unterteile Platine in Elemente (Drei- oder Vierecke)
- Simuliere Wärmeaustausch zwischen den Elementen

Qualität der Lösung:

- Feinheit des Netzes
- Anzahl der Elemente
- Art der Elemente
  - Berücksichtigen der Eingabe
  - Form der Elemente

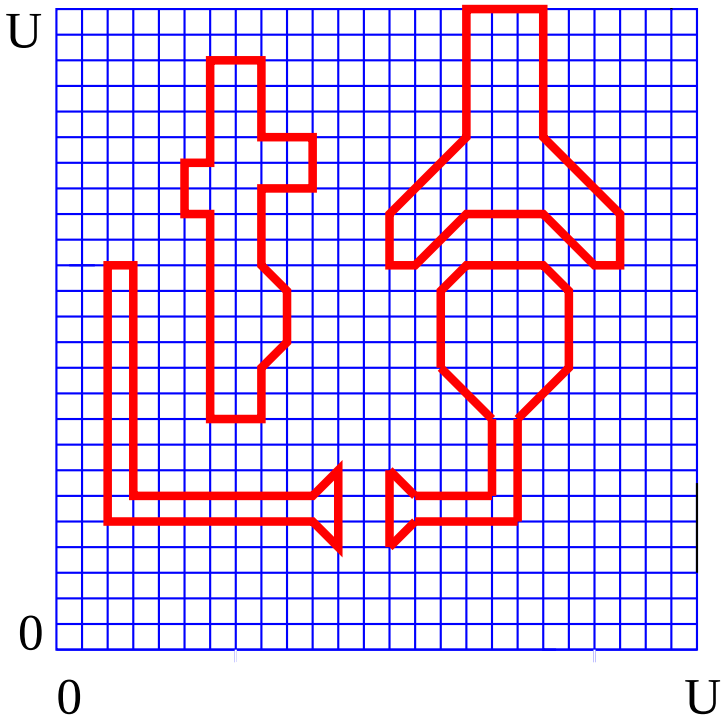
Eingabe

Eingabe:

- Quadrat Q^* (Platine, Gebiet)
- Menge von disjunkten einfachen Polygonen in Q^* (Komponenten)

$$Q^* = [0, U] \times [0, U]$$

mit $U = 2^k$.



Eckpunkte der Komponenten:
ganzzahlige Koordinaten in $[0, U] \times [0, U]$

Kanten der Komponenten:
axen-parallel oder diagonal

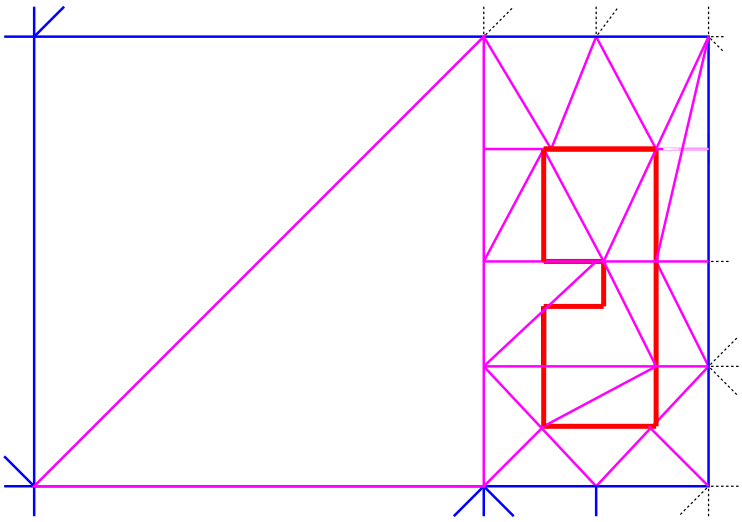
2 Anforderungen an Netze

Ziel:

Erzeugung eines guten Netzes in Q^* aus Dreiecken

Anforderungen an ein gutes Netz:

- Konformität
- Respektieren der Eingabe
- Wohlgeformte Dreiecke
- Nicht-Uniformität



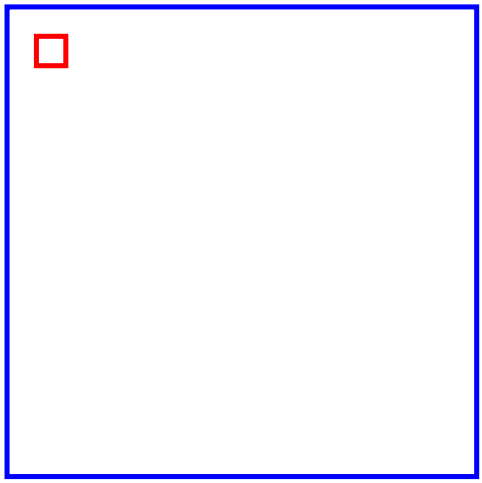
Triangulationen

Bisherige Triangulationen:

- Triangulation eines Polygons
- Triangulation einer Punktmenge (Delaunay-Triangulation)

Probleme:

- Triangulation der Eckpunkte: Eingabe wird nicht respektiert
- zu kleine Winkel ($< 5°$)

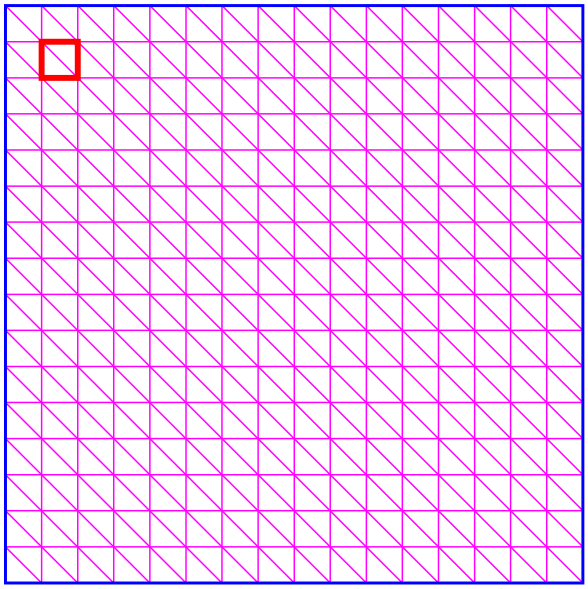


Abhilfe:

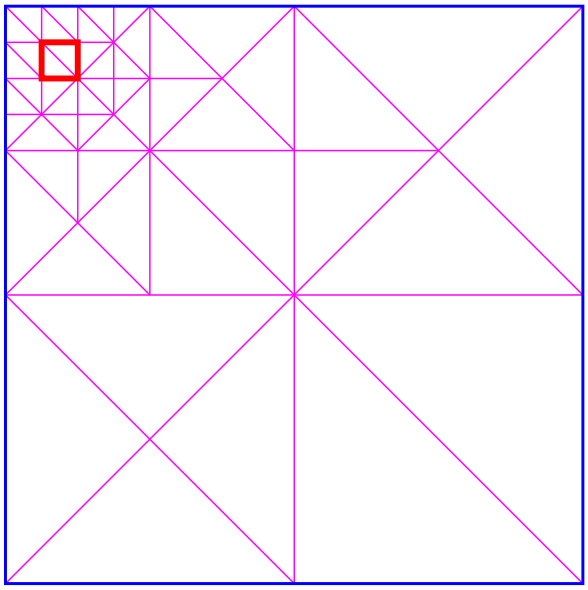
Einfügen zusätzlicher Eckpunkte (Steiner Punkte)

Steiner Punkte

Uniforme und nicht-uniforme Netze



512 Dreiecke



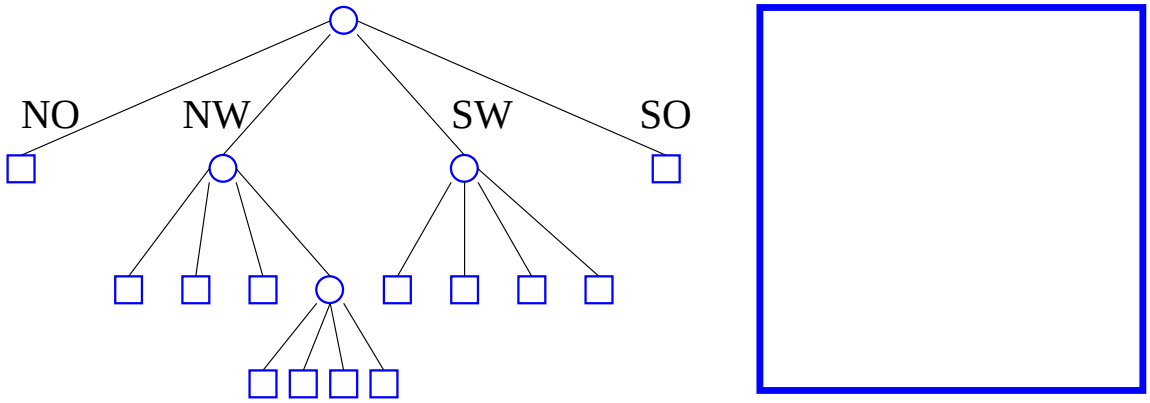
52 Dreiecke

3 Quadrantenbäume

Definition

Ein Quadrantenbaum ist ein Baum mit einer Wurzel, in dem jeder innere Knoten vier Söhne besitzt.

Jedem Knoten v im Quadrantenbaum ist ein Quadrat $Q(v)$ zugeordnet, das ein Quadrant des Quadrats des Vaters ist (sofern vorhanden).



Die Söhne eines Knotens werden mit NO, NW, SW und SO bezeichnet.

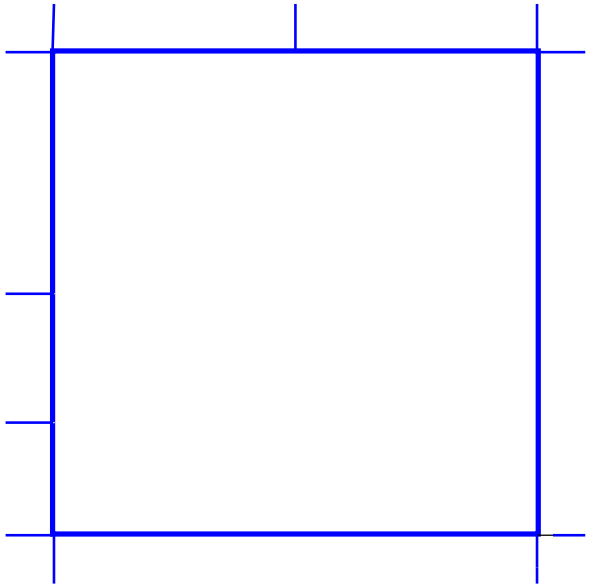
Beobachtung:
Die zugehörigen Quadrate der Blätter bilden eine Unterteilung des zur Wurzel gehörigen Quadrates Q^*

Seiten und Kanten

Definition

Eine Seite eines Quadrates Q ist ein Liniensegment zwischen zwei aufeinanderfolgenden Eckpunkten von Q

Eine Kante eines Quadrates Q ist ein maximales Liniensegment auf dem Rand von Q, das keine Eckpunkte eines Quadrates im Innern enthält



Bemerkungen:

- Eine Seite beinhaltet mindestens eine Kante, aber möglicherweise mehrere.
- Quadrate, die eine Kante teilen, heißen benachbart

Quadrantenbäume für Punktemengen

Quadrantenbaum zur Speicherung einer Unterteilung für eine Punktmenge P in der Ebene

Idee der Konstruktion :

Unterteile Quadrate, so lange noch mehr als ein Punkt in einem Quadrat enthalten ist

Quadrat Q enthalte Punktemenge P

$$Q = [x_Q, x'_Q] \times [y_Q, y'_Q]$$

$$x_{mid} =$$

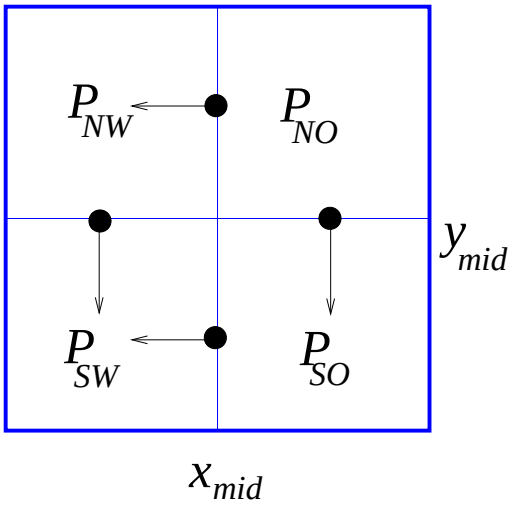
$$y_{mid} =$$

Quadrantenbäume für Punktemengen

Unterteilung von P:

Q_{NO}, Q_{NW}, Q_{SW} und Q_{SO} = die vier Quadranten von Q

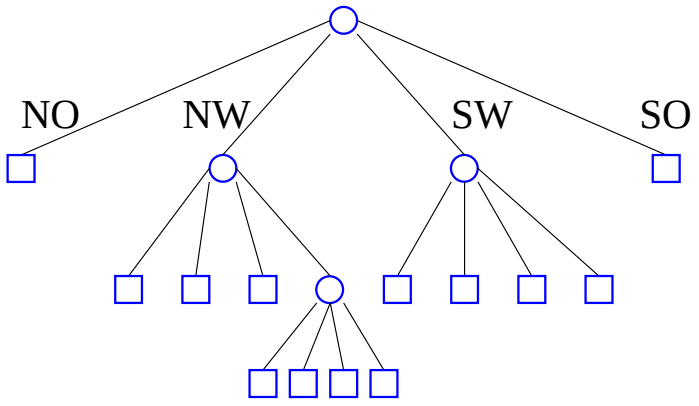
$$P_{NO} \leftarrow \{p \in P \mid p_x > x_{mid} \text{ und } p_y > y_{mid}\}$$
$$P_{NW} \leftarrow \{p \in P \mid p_x \leq x_{mid} \text{ und } p_y > y_{mid}\}$$
$$P_{SW} \leftarrow \{p \in P \mid p_x \leq x_{mid} \text{ und } p_y \leq y_{mid}\}$$
$$P_{SO} \leftarrow \{p \in P \mid p_x > x_{mid} \text{ und } p_y \leq y_{mid}\}$$

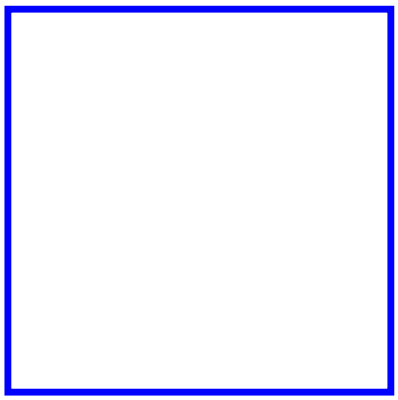


Quadrantenbaumkonstruktion

Algorithmus *Quadrantenbaum*(P, Q)
Input: eine Punktmenge P und ein Quadrat Q
Output: Quadrantenbaum für P und Q
if $card(P) \leq 1$
  then return Blatt v, das die Menge P und das Quadrat Q enthält
  else erzeuge Knoten v mit vier Söhnen v_{NO}, v_{NW}, v_{SW} und v_{SO}

$v_{NO} \leftarrow Quadrantenbaum(P_{NO}, Q_{NO})$
$v_{NW} \leftarrow Quadrantenbaum(P_{NW}, Q_{NW})$
$v_{SW} \leftarrow Quadrantenbaum(P_{SW}, Q_{SW})$
$v_{SO} \leftarrow Quadrantenbaum(P_{SO}, Q_{SO})$
return v



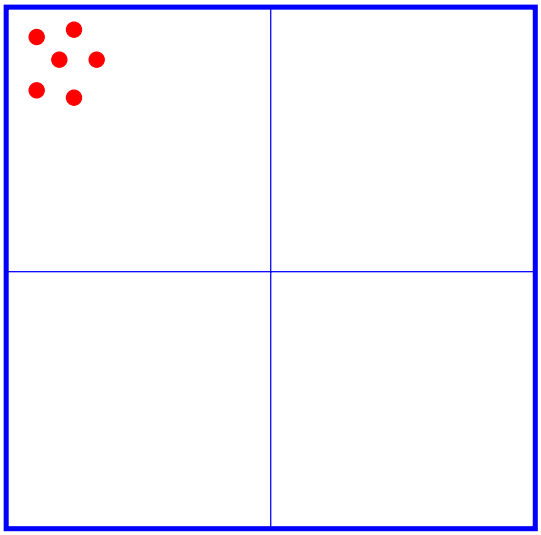


Startquadrat Q^***:**
Kleinstes Quadrat, das P enthält

Konstruktion eines Quadrantenbaums

Problem:

Es können alle Punkte in demselben Quadrat liegen.



Beobachtung:

Größe und Tiefe des Quadrantenbaums hängen nicht von Anzahl der Punkte von P ab

Tiefe eines Quadrantenbaums

Lemma

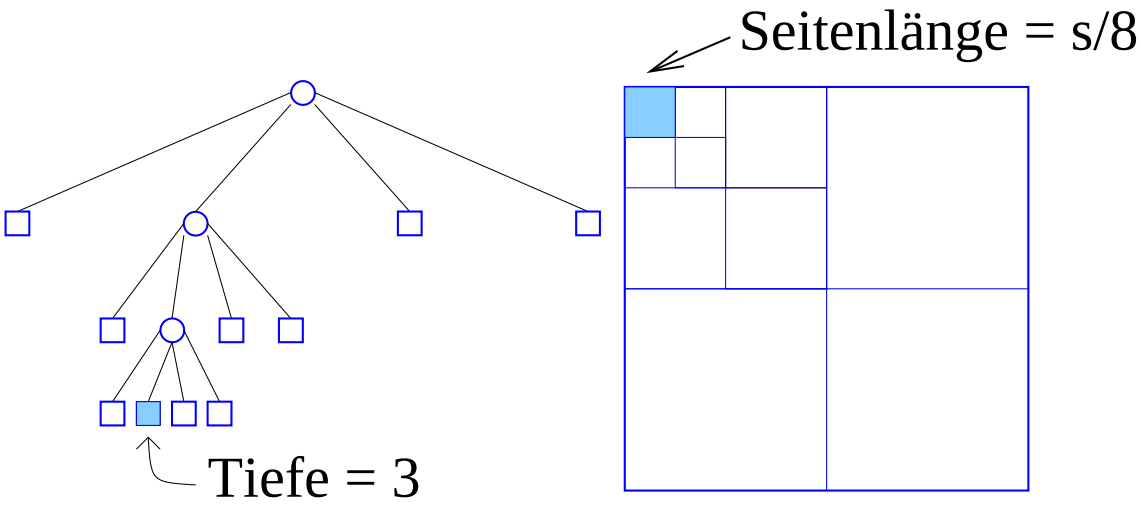
Sei P eine Menge von Punkten in der Ebene.
d_{min} = die kleinste Distanz zwischen zwei Punkten aus P
s = die Seitenlänge von Q^*

Die Tiefe eines Quadrantenbaums für P ist höchstens

$$\log \frac{s}{d_{min}} + \frac{3}{2}$$

Beweis:

1. Seitenlänge s_i eines Quadrates eines Knotens der Tiefe i:



Tiefe eines Quadrantenbaums

2. Maximale Distanz d_i zwischen zwei Punkten innerhalb eines Quadrates der Tiefe i:

Quadrat eines inneren Knoten der Tiefe i enthält mindestens 2 Punkte:

Tiefe des Baumes = 1 + maximale Tiefe eines inneren Knotens

Größe eines Quadrantenbaums

Satz:
Ein Quadrantenbaum einer Tiefe d, der eine Punktmenge von n Punkten enthält, hat $O(dn)$ Knoten und kann in der Zeit der Größenordnung $O(dn)$ konstruiert werden.

Beweis:

1. Knotenanzahl

   Jeder innere Knoten eines Quadrantenbaums besitzt vier Söhne.

   $\Rightarrow$ #(Blätter) = 3#(innere Knoten) + 1

   $\Rightarrow$ #(innerer Knoten) genügt

   #(innerer Knoten der Tiefe i) $\leq n$

   $\Rightarrow$ Anzahl innerer Knoten $O(dn)$

Konstruktionszeit für einen Quadrantenbaum

Beweis: (fortgesetzt)

2. Konstruktionszeit

Verteilung der Punkte auf die zugehörigen Quadrate benötigt die meiste Zeit.

$\Rightarrow$ Zeit an innerem Knoten v linear in der Anzahl der zu verteilenden Punkte

#(Punkte in Quadraten von Knoten der Tiefe i) $\leq n$

$\Rightarrow$ Konstruktionszeit $O(dn)$

Bestimmung eines Nachbarn

Gegeben:

- Knoten v
- Richtung D (N, O, S oder W)

Gesucht:

Knoten v' mit

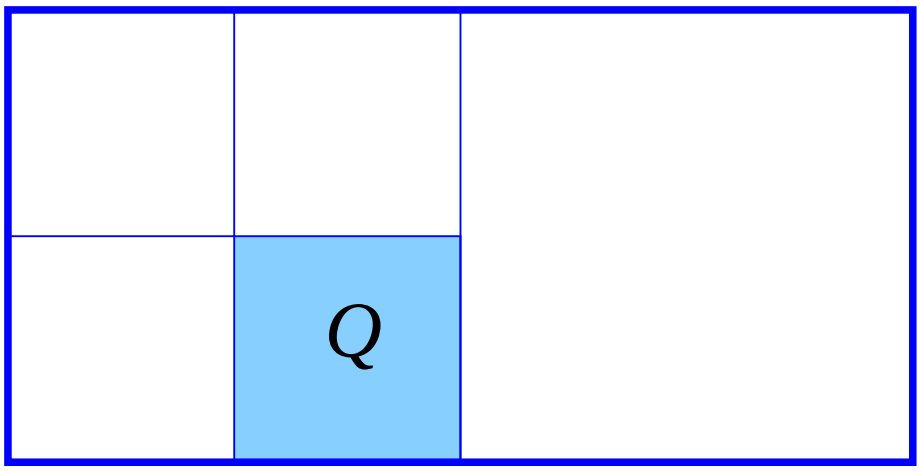
- $Q(v)$ und $Q(v')$ sind in Richtung D benachbart und
- v und v' haben dieselbe Tiefe

Falls kein solcher Knoten existiert:

Ausgabe des tiefsten Knotens, dessen Quadrat in Richtung D mit $Q(v)$ benachbart ist.

Falls kein angrenzendes Quadrat existiert:

Ausgabe **nil**



Bestimmung eines Nachbarn

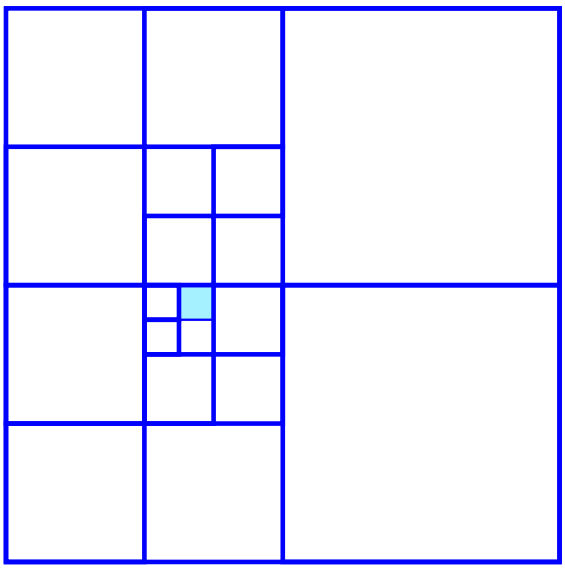
Algorithmus $NordNachbar(v, T)$

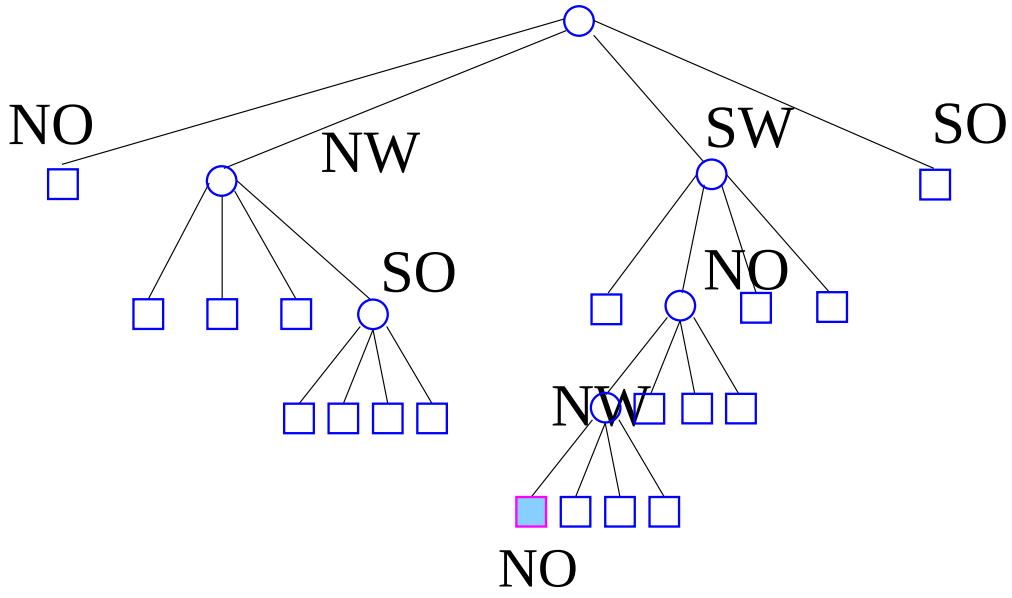
Input: Knoten v eines Quadrantenbaums T.

Output: Der tiefste nördliche Nachbar von v mit $tiefe(v) \leq tiefe(v')$ und **nil**, falls es keinen solchen Knoten gibt.

```
 1 if v = Wurzel von T then return nil
 2 if v = SW-Sohn von vater(v)
 3    then return NW-Sohn von vater(v)
 4 if v = SO-Sohn von vater(v)
 5    then return NO-Sohn von vater(v)
 6 u ← NordNachbar(vater(v), T)
 7 if u = nil or u ist ein Blatt
 8    then return u
 9    else  if v = NW-Sohn von vater(v)
10             then return SW-Sohn von u
11             else  return SO-Sohn von u
```

Bestimmung eines Nachbarn





Bestimmung eines Nachbarn

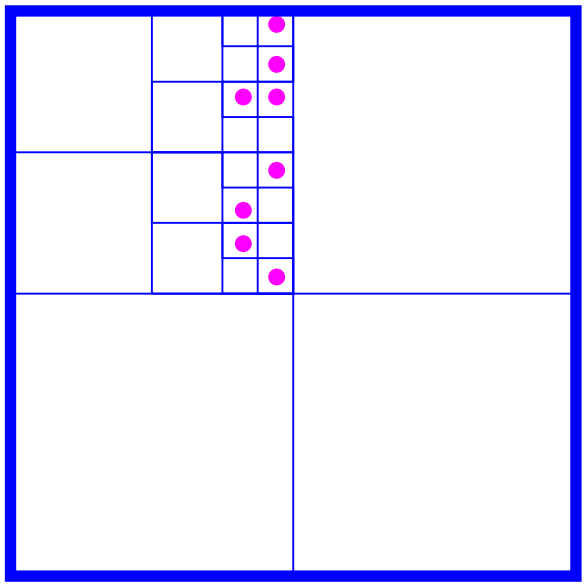
Satz

Sei T ein Quadrantenbaum der Tiefe d. Der Nachbar eines gegebenen Knotens v in T in einer vorgegebenen Richtung kann in $O(d)$ Zeit gefunden werden.

4 Balancierung von Quadrantenbäumen

Die Tiefe benachbarter Blätter ist beliebig unterschiedlich:

$\Rightarrow$ Quadrantenbaum ist nicht balanciert

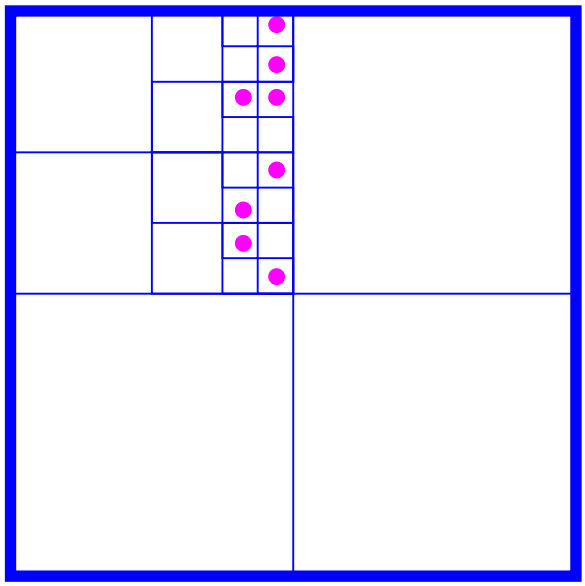


Balancierung von Quadrantenbäumen

Definition

Eine Quadrantenbaum-Unterteilung heißt balanciert, falls sich zwei benachbarte Quadrate in der Größe höchstens um den Faktor 2 unterscheiden.

Ein Quadrantenbaum heißt balanciert, falls die zugehörige Unterteilung balanciert ist.



Balancierte Quadrantenbäume

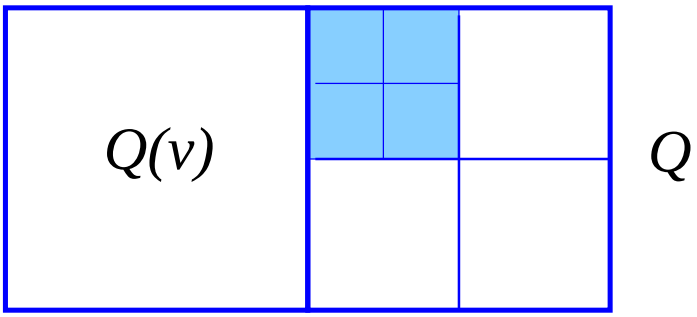
Algorithmus $BalanceQuadrantenbaum(T)$
Input: Quadrantenbaum T
Output: Eine balancierte Quadranten-baum-Version von T.

```
 1  Füge alle Blätter von T in eine Liste L ein.
 2  while L ist nicht leer do
 3      v = remove-first(L)
 4      if Q(v) muß aufgeteilt werden
 5          then erzeuge vier Blätter als Söhne
                  von v
 6               if v enthält einen Punkt p
 7                   then bringe p in neuem Blatt
                           unter
 8               Füge die vier neuen Blätter in L
                    ein.
 9               L' ← Liste von Nachbarn von
                    Q(v), die nun aufgeteilt
                    werden müssen.
10               L ← L ∪ L'
11  return T
```

Balancierte Quadrantenbäume

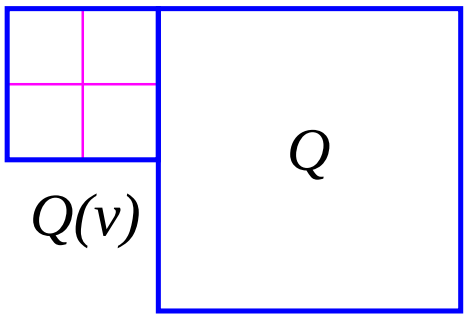
Schritt 4: $Q(v)$ muß aufgeteilt werden

$\Leftrightarrow$ Nachbar hat einen an $Q(v)$ angrenzenden Sohn, der kein Blatt ist



Schritt 9: Nachbar Q von $Q(v)$ muß aufgeteilt werden

$\Leftrightarrow$ Q ist größer als $Q(v)$



Balancierte Quadrantenbäume

Frage: Größe eines balancierten Quadrantenbaums?

Satz
Sei T ein Quadrantenbaum mit m Knoten.

Dann hat die balancierte Version des Baumes $O(m)$ Knoten, und dieser Baum kann in $O(dm)$ Zeit konstruiert werden.

Beweis:

Wir zeigen:

Es werden höchstens $O(m)$ Knoten unterteilt, um T zu balancieren.

$\Rightarrow$

1. Mit jedem Unterteilungschritt werden nur 4 weitere Knoten erzeugt

2. Anzahl betrachtete Knoten (Knoten in L):

   $\leq m + 4$ #(Unterteilungschritte)

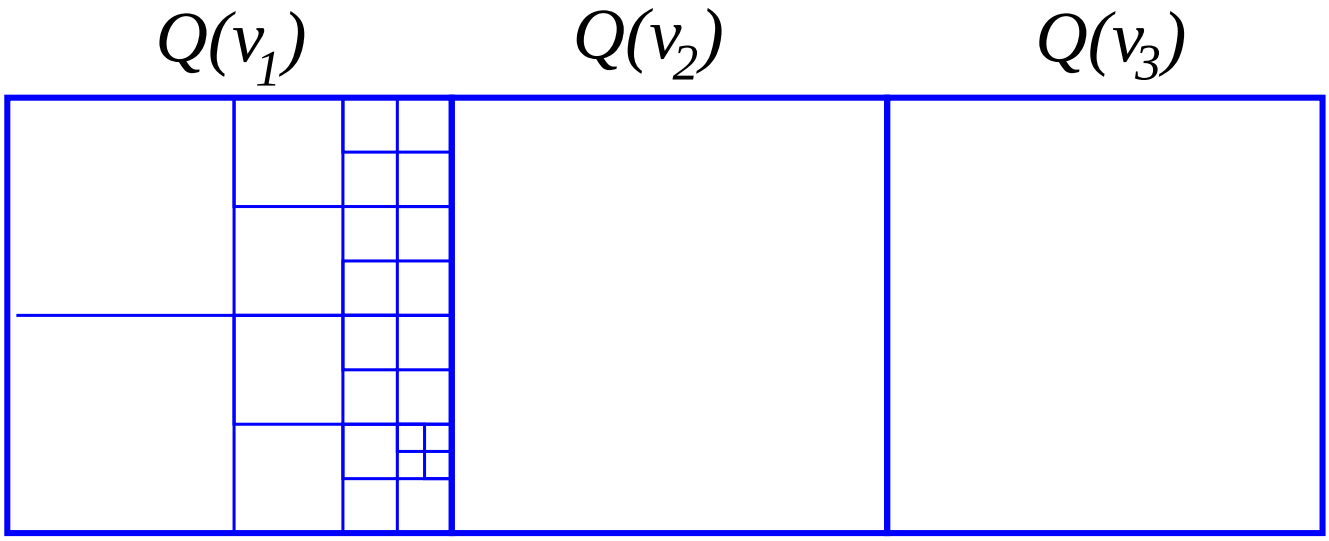
   pro betrachteten Knoten:

   4 Nachbarsuch-Operationen $O(d)$

Balancierte Quadrantenbäume

Seien v_1, v_2 und v_3 Knoten derselben Tiefe in T mit

- $Q(v_1), Q(v_2)$ und $Q(v_3)$ benachbart und
- v_2 und v_3 sind Blätter



Wir zeigen:
Die Unterteilung von $Q(v_1)$ setzt sich nicht nach $Q(v_3)$ fort

Balancierte Quadrantenbäume

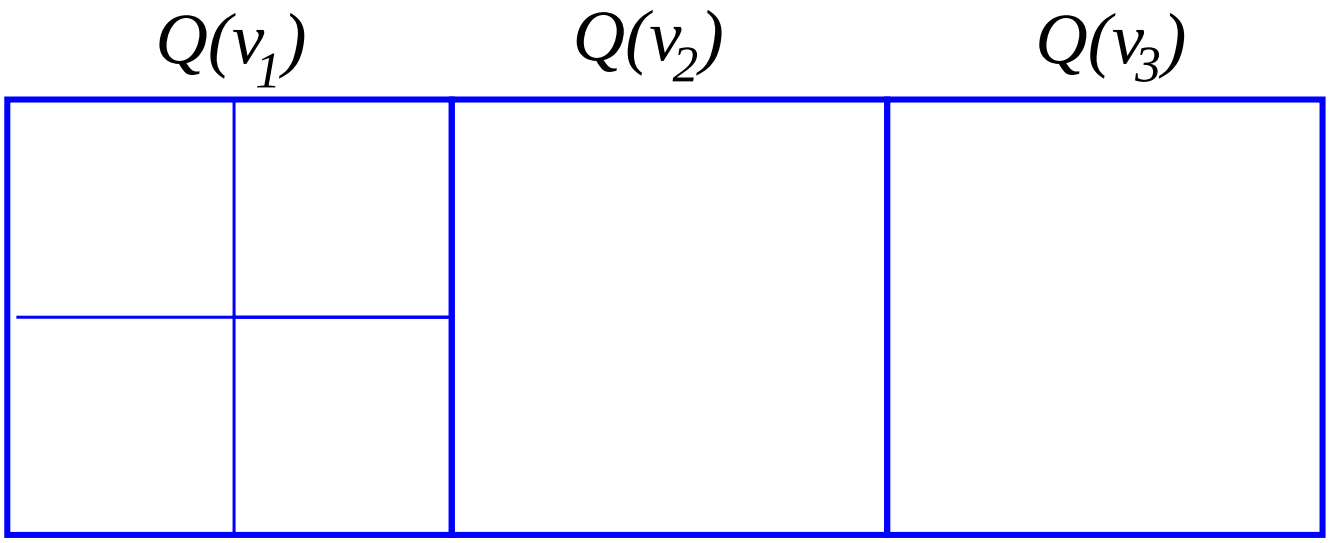
Induktion über Tiefe d des Teilbaumes mit Wurzel v_1:

$d = 1$:

$\Rightarrow Q(v_1)$ wurde einmal geteilt

$\Rightarrow Q(v_2)$ wird nicht geteilt

$\Rightarrow Q(v_3)$ wird nicht geteilt



Balancierte Quadrantenbäume

$d > 1$:

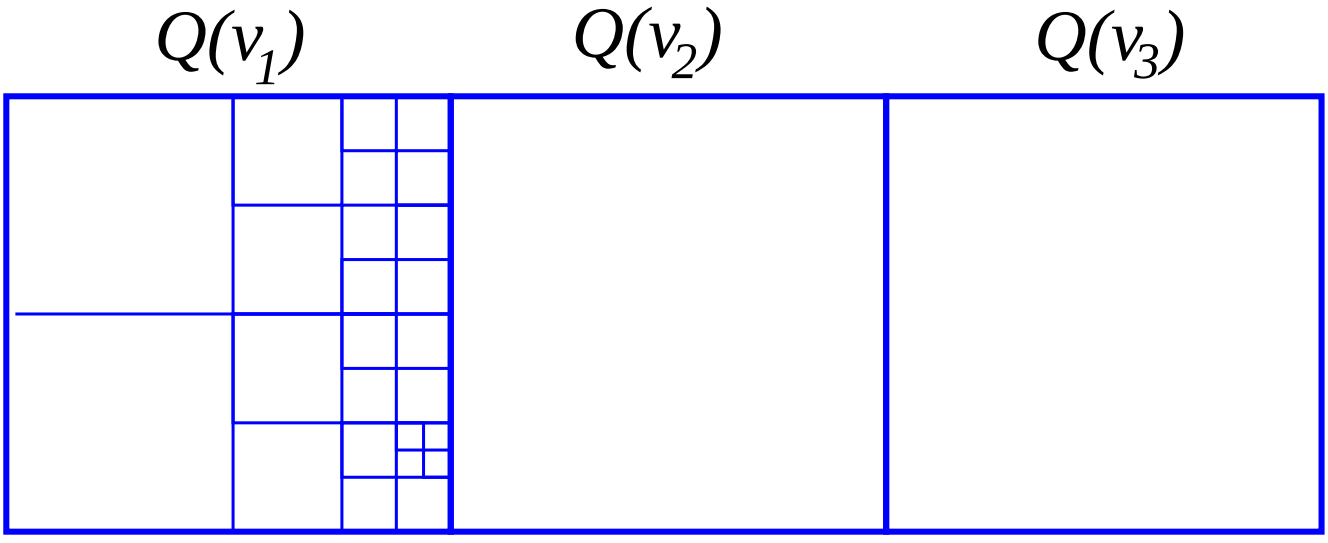
$\Rightarrow$ $Q(v_2)$ in vier Quadrate teilen

$\Rightarrow$ Unterteilung im NO-Quadranten von $Q(v_1)$ wird nicht im NO-Quadranten von $Q(v_2)$ fortgesetzt (Induktionsvoraussetzung)

$\Rightarrow$ Unterteilung im SO-Quadranten von $Q(v_1)$ wird nicht im SO-Quadranten von $Q(v_2)$ fortgesetzt (Induktionsvoraussetzung)

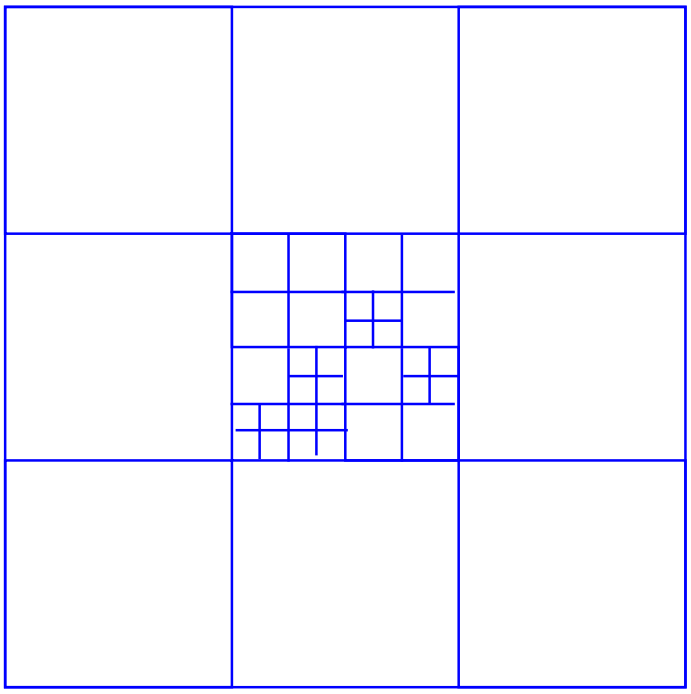
$\Rightarrow$ SO-, NO-Quadranten von $Q(v_2)$ werden nicht unterteilt

$\Rightarrow$ $Q(v_3)$ wird nicht unterteilt.



Balancierte Quadrantenbäume

Folgerung:
Die Unterteilung eines Quadrates Q in Unterquadrate kann nur dann eine Unterteilung eines Quadrates derselben Größe auslösen, falls dieses Quadrat eines der acht umliegenden Quadrate von Q ist.



Balancierte Quadrantenbäume

T' = balancierte Version von T.

Alte Quadrate: Quadrate der Knoten in T
Neue Quadrate: Quadrate der Knoten in
$T' \setminus T$

Annahme: Quadrat Q (alt oder neu) wird während des Balancierungsprozesses geteilt.

$\Rightarrow$ einer der acht Nachbarn von Q ist ein unterteiltes altes Quadrat

$\Rightarrow$ weise diesem alten Quadrat eine Kosteneinheit für die Unterteilung von Q zu.

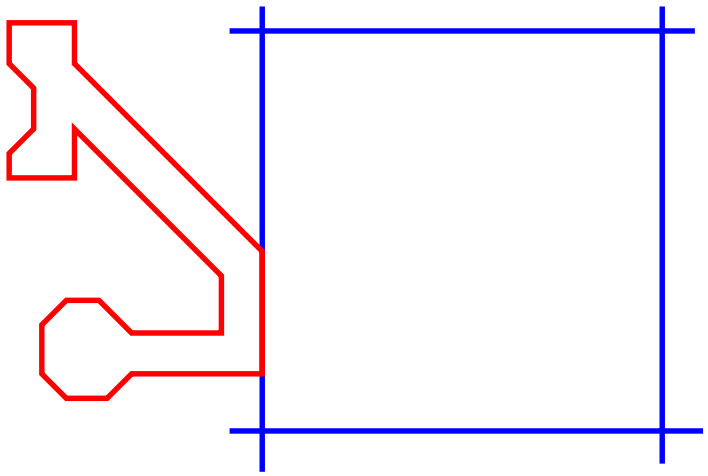
$\Rightarrow$ Jedes alte Quadrat erhält höchstens 8 Kosteneinheiten.

$\Rightarrow$ höchstens $8m$ Knoten werden unterteilt

5 Von Quadrantenbäumen zu Netzen

Ziel:
Eine gute Netztriangulation des Quadrates für die Eingabe.

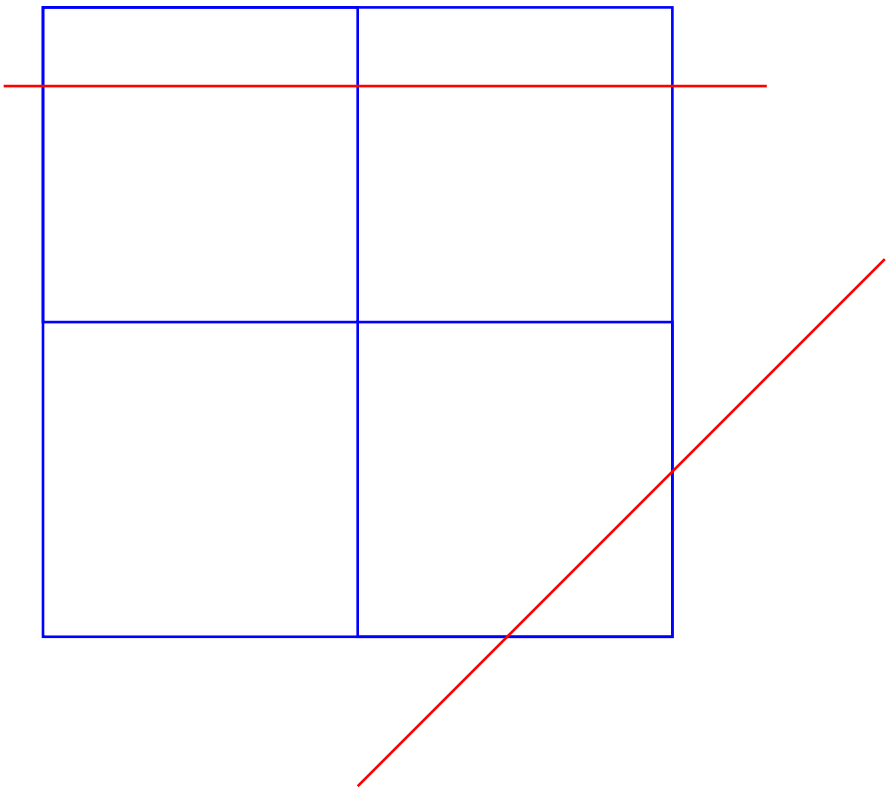
Idee:
- Verwende Quadrantenbaum
- Neues Haltekriterium:
  Halte Unterteilung an,
  falls das Innere des Quadrats nicht länger von einer Kante geschnitten wird, oder
  falls das Quadrat Einheitslänge aufweist



Quadrantenbaum-Unterteilung ist nicht-uniform

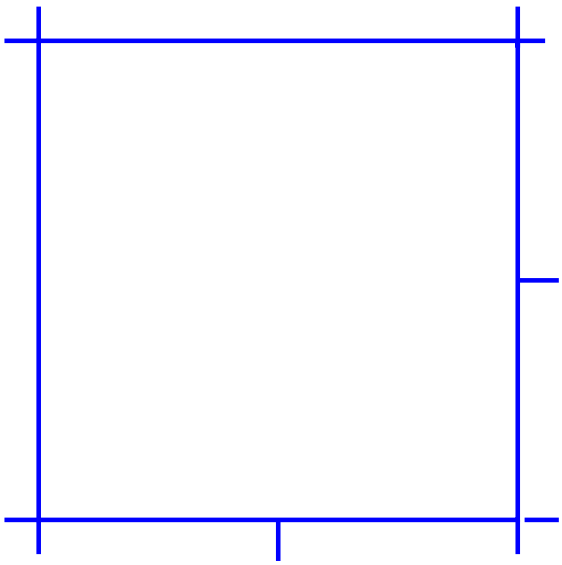
Lemma

Der Schnitt einer Kante mit dem Inneren des Quadrates eines beliebigen Blattes ist die Diagonale des Quadrates (oder es existiert kein Schnitt).



Triangulierung von Quadrantenbäumen

1. Balanciere Quadrantenbaum.
2. Quadrate, die keine Ecke im Inneren einer ihrer Seiten haben:
   Triangulation durch Diagonale
3. Sonstige Quadrate:
   Setze einen Steinerpunkt in das Zentrum



Triangulierung

Algorithmus $GenerateMesh(S)$

Input: Eine Menge S von Komponenten im Inneren des Quadrates $Q^* = [0, U] \times [0, U]$

Output: Ein gutes trianguliertes Netz M

```
 1 Konstruiere einen Quadrantenbaum T auf
     der Menge S innerhalb von Q* nach dem
     neuen Haltekriterium
 2 T ← BalanceQuadrantenbaum(T)
 3 M ← Unterteilung von Q* durch T
 4 for all Q ∈ M do
 5     if Komponentenkante schneidet Inneres
         von Q
 6       then füge Diagonale von Q in M ein
 7     else if Q hat nur Knoten an den Ecken
 8       then füge Diagonale von Q in M ein
 9       else  ergänze Mittelpunkt p von Q
10             verbinde p mit allen Knoten auf
               dem Rand von Q und füge die
               neuen Kanten in M ein.
11 return M
```

Erzeugung von Netzen

Satz

Sei S eine Menge von disjunkten polygonalen Komponenten innerhalb des Quadrates $Q^* = [0, U] \times [0, U]$.

Dann existiert für diese Eingabe ein nicht-uniformes trianguliertes Netz, das konform ist, die Eingabe beachtet und nur wohlgeformte Dreiecke hat.

Die Anzahl der Dreiecke ist $O(\mathit{Umfang}(S) \log U)$, wobei $\mathit{Umfang}(S)$ die Summe der Umfänge der Komponenten aus S ist.

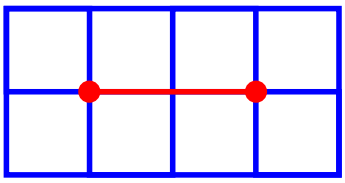
Das Netz kann in der Zeit $O(\mathit{Umfang}(S) \log^2 U)$ konstruiert werden.

Erzeugung von Netzen

Beweis:

Eigenschaften des erzeugten Netzes

Größe der Quadrantenbaum-Unterteilung:
Liniensegment der Länge l schneidet
höchstens $2(l+2)$ Einheitsquadrate



$\Rightarrow$ Anzahl der Einheitsquadrate, deren Abschluß von einer Kante einer beliebigen Komponente geschnitten werden:

$$O(\mathit{Umfang}(S))$$

$\Rightarrow$ Anzahl innerer Knoten des Quadrantenbaum in einer festen Tiefe:

$$O(\mathit{Umfang}(S))$$

Tiefe des Quadrantenbaum: $O(\log U)$

$\Rightarrow$ Anzahl der Knoten des Quadrantenbaums

$$O(\mathit{Umfang}(S) \log U)$$

Erzeugung von Netzen

- Balancieren erhöht Komplexität nicht asymptotisch
- Triangulieren der balancierten Quadrate erhöht die Komplexität nicht asymptotisch (höchstens 8 Dreiecke pro Quadrat)

$\Rightarrow$ Zahl der Dreiecke des endgültigen Netzes ist

$$O(\mathit{Umfang}(S) \log U)$$

Konstruktionszeit $T(S, U)$

1. Konstruktion des Quadrantenbaums:
   Linear in der Anzahl der Knoten:

$$O(\mathit{Umfang}(S) \log U)$$

2. Balancieren: zusätzlicher Faktor

$$d \leq \log U$$

3. Doppelt verkettete Kantenliste
4. Triangulieren

$\Rightarrow$

$$T(S, U) = O(\mathit{Umfang}(S) \log^2 U)$$

Platine Beispiel