

**Ausarbeitung**

# **Verfeinerte Delaunay- Triangulationen**

im Rahmen des Seminars  
Algorithmische Geometrie

Moritz Rieger und Christian Weber

Betreuer: Dr. Lihong Ma

# Inhaltsverzeichnis

<b>1</b>	<b>Anwendungen für Delaunaygraphen</b>	<b>1</b>
1.1	Finite Elemente Methoden . . . . .	1
1.2	Computergrafik . . . . .	1
1.2.1	3D Mesh Generation . . . . .	2
1.2.2	Global Illumination . . . . .	2
1.2.3	Generative Kunst . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Planar straight line graph (PSLG) . . . . .	3
2.2	Delaunay Eigenschaften . . . . .	4
2.2.1	Constrained Delaunay Triangulation (CDT) . . . . .	4
2.3	Delaunay Qualitätskriterien . . . . .	5
2.3.1	Skinny Triangles . . . . .	5
2.3.2	Encroached Subsegments . . . . .	6
<b>3</b>	<b>Delaunay Verfeinerungsalgorithmus von Ruppert</b>	<b>7</b>
3.1	Beschreibung des Algorithmus . . . . .	7
3.2	Terminierungsnachweis . . . . .	10
<b>4</b>	<b>Delaunay Verfeinerungsalgorithmus von Chew</b>	<b>15</b>
4.1	Beschreibung des Algorithmus . . . . .	15
4.2	Terminierungsnachweis . . . . .	16
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>20</b>

# Abstract

Verfeinerung von Delaunay-Graphen ist nötig, um eine Vernetzungsqualität zu erhalten, die z. B. bei der finiten Elemente Methode gefordert wird. Durch das *Verhältnis von Umkreisradius zur kürzesten Kante eines Dreiecks* lässt sich eine obere Schranke  $B$  festlegen, bis zu welcher die Dreiecke als qualitativ ausreichend anzusehen sind. Die im Anschluss vorgestellten Algorithmen haben das Ziel, alle Dreiecke des Graphen, die dieser Bedingung nicht standhalten, durch Einfügen neuer Knoten zu eliminieren und dabei die Delaunay-Bedingungen zu erhalten. Außerdem garantieren die Algorithmen eine untere Schranke für den kleinsten neu erzeugten Winkel. Da durch das Einfügen neuer Knoten immer wieder aufs Neue die Delaunay Bedingungen verletzt werden können, wird anhand der in [1] eingeführten Flussgraphen bewiesen, weshalb für beide Algorithmen eine untere Grenze der Länge neu erzeugter Kanten existiert und somit ein Terminieren garantiert ist.

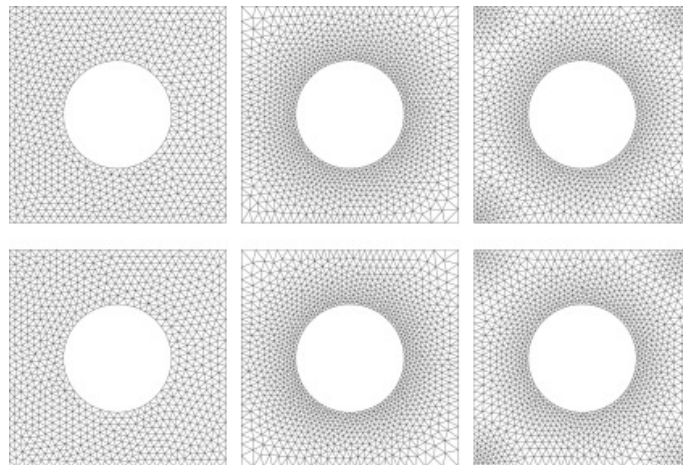
# 1 Anwendungen für Delaunaygraphen

Zunächst wollen wir eine Motivation für die folgenden Kapitel schaffen, indem wir die vielfältigen Anwendungsfälle von Delaunaygraphen aufzeigen.

## 1.1 Finite Elemente Methoden

FEM ist ein Berechnungsverfahren, welches häufig in Ingenieurwissenschaften genutzt wird, um z. B. Kräfte zu berechnen und deren Wirkung auf einzelne Bauteile oder ganze Bauteilgruppen zu simulieren.

Die Vernetzung der zu betrachtenden Geometrie spielt dabei eine zentrale Rolle. Das Netz zerlegt komplexe Geometrien in einzelne Teilstücke. An den Knotenpunkten, welche die einzelnen Elemente miteinander verbinden, werden die Kräfte von einem Netzelement auf das nächste Element übertragen. Für jedes Element ergeben sich Teillösungen, welche dann zu einer Lösung des Gesamtsystems aufsummiert werden. Je nach Problemstellung ist die Qualität der Vernetzung von großer Bedeutung. Um zum Beispiel Spannungen im Bauteil effektiv zu bestimmen, bedarf es einer hohen Netzauflösung im Einflussbereich. Abbildung 1 zeigt verschiedene Triangulationen einer Geometrie.



**Abbildung 1:** Delaunay Triangulation eines Bauteiles mit unterschiedlichen Auflösungen aus [2].

## 1.2 Computergrafik

In der Computergrafik steht man oft vor der Herausforderung eine dreidimensionale Punktwolke durch ein Netz zu verbinden, um eine durchgehende Oberfläche zu erhalten.

### 1.2.1 3D Mesh Generation

Extrahiert man aus mehreren zweidimensionalen Bildern gemeinsame Merkmalspunkte, kann man diese mit einer Delaunay Triangulation zu einem Netz verbinden. So erhält man eine Oberfläche, welche im Anschluss mit einer Textur versehen werden kann, wie in Abbildung 2(a) zu sehen ist.



(a)



(b)

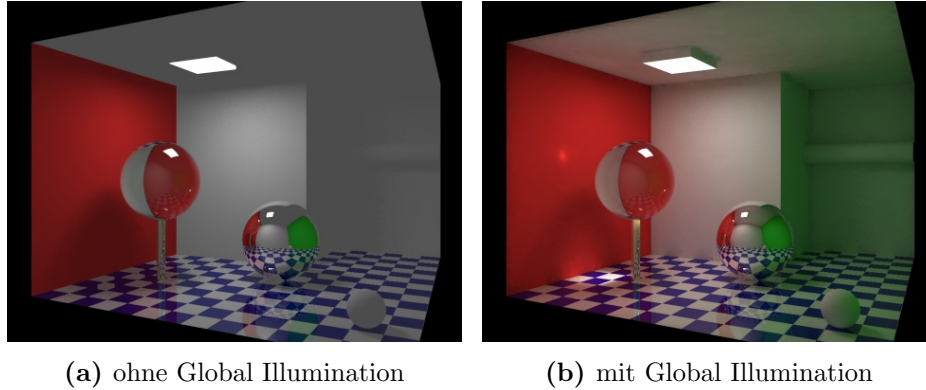
**Abbildung 2:** (a) Delaunay Triangulation über gemeinsame Merkmalspunkte eines Clowns aus [3]. (b) Trianguliertes Pixelbild mit Triangulate7 siehe [4].

### 1.2.2 Global Illumination

Unter Global Illumination versteht man die Nachahmung der Lichtwechselwirkung von Objekten in der 3-D-Computergrafik. Es wird dabei versucht einen besonders natürlichen Beleuchtungseffekt zu erzielen, indem das Umgebungslicht an jeder Stelle der Szene berechnet wird. Um die Beleuchtungsintensität eines Punktes in der Szene zu bestimmen, sendet man sogenannte *Raytraces* von den Lichtquellen aus. Diese werden je nach Oberflächenbeschaffenheit der Objekte mit bestimmter Intensität und Farbe reflektiert. Die Aufprallpunkte der Photonen in der Szene werden dann zu einem Delaunay Netz trianguliert. Die Dichte des Delaunaynetzes gibt nun, wie in [5] gezeigt, Auskunft über die Beleuchtungsintensität aller Punkte im Raum.

### 1.2.3 Generative Kunst

In letzter Zeit erlebt die Dreiecksform auch im künstlerisch, kreativen Bereich immer größere Beliebtheit. Es ist ein eigener Stil, die Low-Poly Art entstanden. Mit frei angebotenen Tools, wie z. B. Triangulate 7 von [4], können aus Pixelvorlagen durch



**Abbildung 3:** In der Abbildung aus [6] ist die globale Beleuchtung durch die Reflexion der farbigen Flächen gut zu erkennen.

das Setzen von Punkten im Handumdrehen abstrakte Kunstwerke erzeugt werden. Dabei werden die gesetzten Punkte zu einem Delaunay Netz verbunden, die Dreiecke des Delaunay-Netzes werden mit dem Durchschnitt der darunter liegenden Farbe gefüllt, siehe Abbildung 2(b).

## 2 Grundlagen

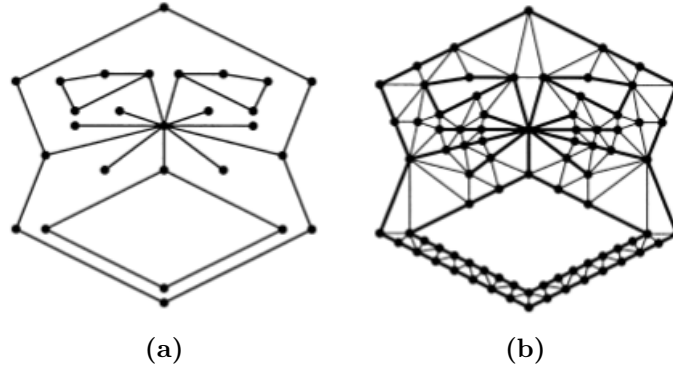
Wie im Kapitel 1 gezeigt, gibt es eine Vielzahl von Anwendungen, welche eine qualitativ hochwertige Vernetzung der gegebenen Geometrie voraussetzen. Die hier vorgestellte Delaunay Verfeinerung ist eine Methode, um solche hochwertigen Netze zu erzeugen. Das Problem ist es, eine passende Triangulation zu finden, welche die gesamte Geometrie abdeckt und dabei nur Dreiecke enthält, welche die gewünschten Vorgaben erfüllen:

- Die im Dreieck auftretenden Winkel sollten nicht zu klein, aber auch nicht zu groß sein, da dies zu sehr spitzen und ungleichmäßigen Dreiecken führt.
- Die Dreiecke sollten nicht viel kleiner oder größer sein, als es für den gewünschten Anwendungsfall notwendig ist.

### 2.1 Planar straight line graph (PSLG)

Um im Folgenden genauer auf die Details der Triangulation eingehen zu können, definieren wir zunächst unsere Eingabe als einen *planar straight line graph (PSLG)*. Ein PSLG besteht aus Knoten und Segmenten wie in Abbildung 4(a) zu sehen. Ein *Segment* ist eine Kante, welche auch im Graphen nach der Triangulation durch eine

Verkettung von Kanten enthalten sein muss. Nach Definition muss ein PSLG beide Endknoten eines Segments enthalten. Außerdem dürfen sich Segmente nur an Knoten bzw. Endpunkten anderer Segmente schneiden. Bei einer Vernetzung des PSLGs werden die *Segmente* in mehrere kleine Kanten, genannt *Subsegmente*, unterteilt, welche in Abbildung 4(b) durch die breiten Kanten dargestellt werden. Das



**Abbildung 4:** Ein PSLG und ein trianguliertes Netz nach Anwendung von Rupperts Algorithmus aus [1].

zu *triangulierende Gebiet* ist der Bereich der trianguliert werden soll. Zum Erstellen eines Netzes aus einem gegebenen PSLG muss dieser *segment-gebunden* sein, d.h. die Segmente müssen das gesamte zu triangulierende Gebiet umschließen. Der PSLG darf zwar Löcher enthalten, diese müssen jedoch auch gänzlich von Segmenten umschlossen sein, damit die zu triangulierende Domäne gänzlich vom *Außengebiet* getrennt ist.

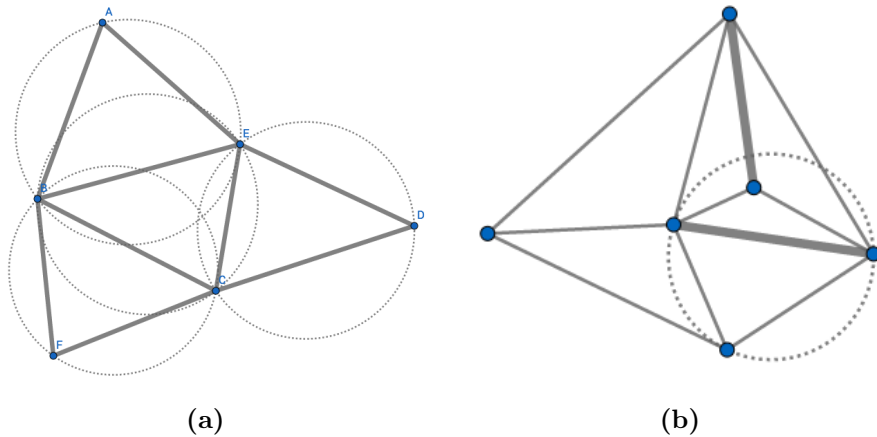
## 2.2 Delaunay Eigenschaften

Eine Triangulierung erfüllt dann die Delaunay Eigenschaft, wenn die sogenannte *Umkreisbedingung* erfüllt ist. Das bedeutet, dass der Umkreis eines jeden Dreiecks der Triangulierung keinen anderen Punkt des Graphen enthalten darf, siehe Abbildung 5(a).

Durch diese Eigenschaft weisen die Dreiecke möglichst große Innenwinkel auf, was gleichmäßigere Dreiecke zur Folge hat.

### 2.2.1 Constrained Delaunay Triangulation (CDT)

Eine *constrained Delaunay triangulation* ist ähnlich wie eine Delaunay Triangulation, berücksichtigt jedoch die Segmente und Knoten des PSLG  $G$ . Zwei Punkte sind dabei zu beachten:



**Abbildung 5:** (a) *Delaunay Graph* - kein Knoten liegt innerhalb eines Umkreises. (b) *Constrained Delaunay Graph* - kein sichtbarer Knoten liegt innerhalb eines Umkreises

- Kein Segment schneidet das Innere eines Dreiecks, da die Triangulation die Segmente berücksichtigen muss.
- Der Umkreis jedes Dreiecks enthält keine Knoten, die vom Inneren des Dreiecks aus sichtbar sind. Zwei Punkte sind *sichtbar* zueinander, wenn die Strecke zwischen diesen Punkten von keinem Segment aus  $G$  geschnitten wird.

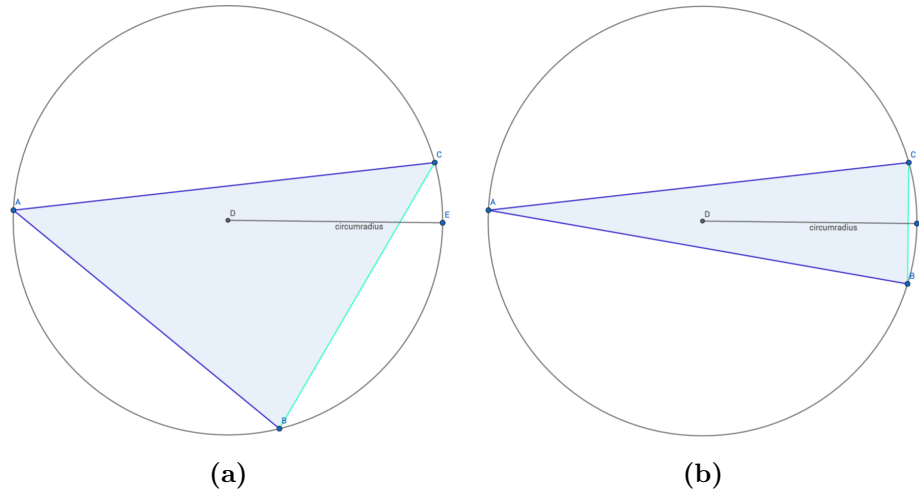
Es werden also Teile des Umkreises nicht berücksichtigt, wenn diese durch ein Segment aus  $G$  abgeschnitten werden wie in Abbildung 5(b) zu sehen. Enthält der PSLG keine Kanten, gleicht der *Constrained Delaunay Graph* dem Delaunay Graph.

## 2.3 Delaunay Qualitätskriterien

### 2.3.1 Skinny Triangles

Um die Qualität eines Delaunay-Graphen zu beschreiben, bedarf es aussagekräftiger Kriterien. Ein sehr elegantes Kriterium ist das *Verhältnis von Umkreisradius zur kürzesten Kante eines Dreiecks*. Eine obere Grenze  $B$  (Boundary) für diesen Wert garantiert eine gute Qualität des Delaunay Netztes, denn je kleiner dieser Wert ist, desto gleichmäßiger sind die Winkel im Dreieck, wie in Abbildung 6 zu sehen ist. Liegt dieser Quotient überhalb der festgelegten Grenze  $B$ , nennt man das Dreieck *skinny*. Ein häufiger Wert als obere Grenze  $B$  ist z. B. 1, oder  $\sqrt{2}$ .

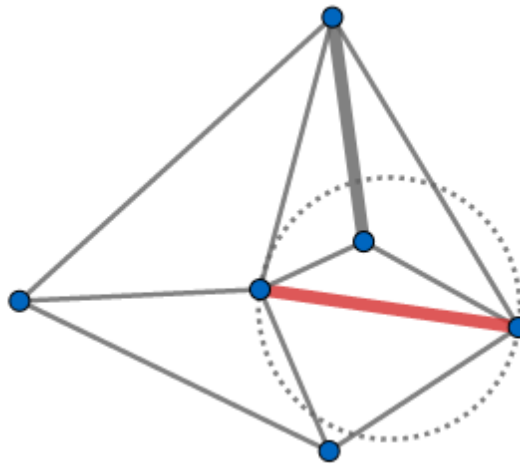




**Abbildung 6:** Bei  $B = 1$  gilt das Dreieck (a), bei welchem das Verhältnis von Umkreisradius zur kürzesten Kante 0,7 beträgt, als akzeptabel. Das Dreieck (b), mit einem Verhältnis von 1,8, ist hingegen als *skinny* zu bezeichnen.

### 2.3.2 Encroached Subsegments

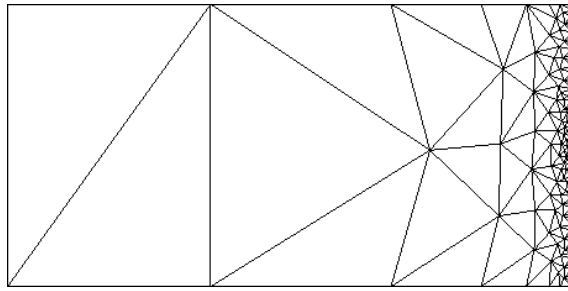
Ein Subsegment wird *encroached* (beeinflusst) genannt, wenn innerhalb des Durchmesserkreises des Subsegments ein sichtbarer Knoten des zu triangulierenden Graphen liegt. Das rot eingefärbte Subsegment in Abbildung 7 stellt ein solches *encroached subsegment* dar.



**Abbildung 7:** Dieser Delaunay Graph enthält ein *encroached subsegment* (rot), da ein Punkt innerhalb des Durchmesserkreises der roten Strecke liegt.

### 3 Delaunay Verfeinerungsalgorithmus von Ruppert

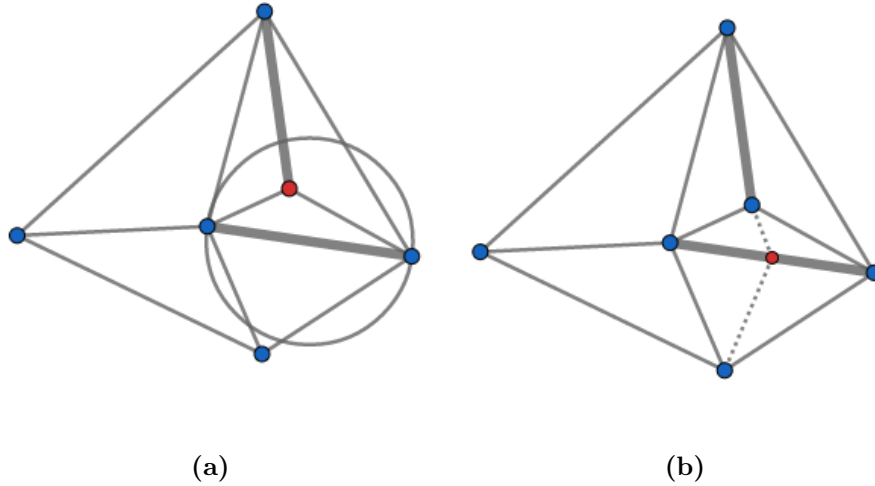
Ruppert's Algorithmus ist der erste praxistaugliche Algorithmus zur Verbesserung der Netzqualität eines gegebenen Delaunay Graphen. Er basiert auf Chews erstem Algorithmus und produziert schön abgestufte Dreiecksnetze, wobei eine große Varianz der Dreiecksgrößen möglich ist, vgl. Abbildung 8. Außerdem generiert er weniger Dreiecke als seine Vorgänger und bietet die Möglichkeit eine Grenze für kleinste Winkel - unabhängig von den Winkeln des Eingangsgraphen - zu setzen. Voraussetzung damit der Algorithmus terminiert ist, dass nur die Dreiecke welche die Schranke  $B = \sqrt{2}$  überschreiten, als spitz angesehen werden. Außerdem darf der Winkel zwischen zwei inzidenten Segmenten im PSLG nicht kleiner als  $60^\circ$  sein. Wir werden in Kapitel 3.2 genauer darauf eingehen, weshalb diese Beschränkungen gelten müssen.



**Abbildung 8:** Die Abbildung aus [1] ein *schön abgestuftes* Netz und dessen Übergang von großen zu kleinen Dreiecken nach Anwendung von Rupperts Algorithmus.

#### 3.1 Beschreibung des Algorithmus

Wir beschreiben Rupperts Algorithmus hier, wie in [1] erläutert, ausgehend von einem *Constrained Delaunay* Graphen eines *segment-gebundenen PSLG*. Der Algorithmus fügt so lange Knoten in den gegebenen Graphen ein, bis die vom Nutzer gesetzten Bedingungen hinsichtlich der Qualität und Größe der Dreiecke erfüllt sind. Zwei Regeln legen fest, wann und wo ein neuer Knoten eingefügt werden soll. Zuerst werden alle *encroached subsegments* behandelt. Im zweiten Schritt werden dann alle *skinny triangles* eliminiert. Ist ein Subsegment encroached, wird ein neuer Knoten in der Mitte der betroffenen Kante eingefügt. Alle Knoten, die mit dem neuen Knoten verbunden werden können ohne eine Kante bzw. Subsegment zu schneiden werden mit diesem verbunden, vgl. Abbildung 9. Ein neu eingefügter Knoten kann wiederum zu neuen *encroached subsegments* führen.



**Abbildung 9:** Einfügen eines neuen Knotens aufgrund eines *encroached subsegments*.

Erst, wenn es in dem Graphen keine *encroached subsegments* mehr gibt, geht der Algorithmus in den 2. Schritt über. Beim Beseitigen der *skinny triangles* müssen zwei Fälle unterschieden werden:

**Fall 1:** Durch den neuen Knoten würden wieder *encroached subsegments* entstehen.

Der Knoten wird nicht eingefügt (*rejected vertex*), sondern es werden die Subsegmente, welche von dem Knoten beeinflusst würden, mit einem Knoten in der Mitte geteilt.

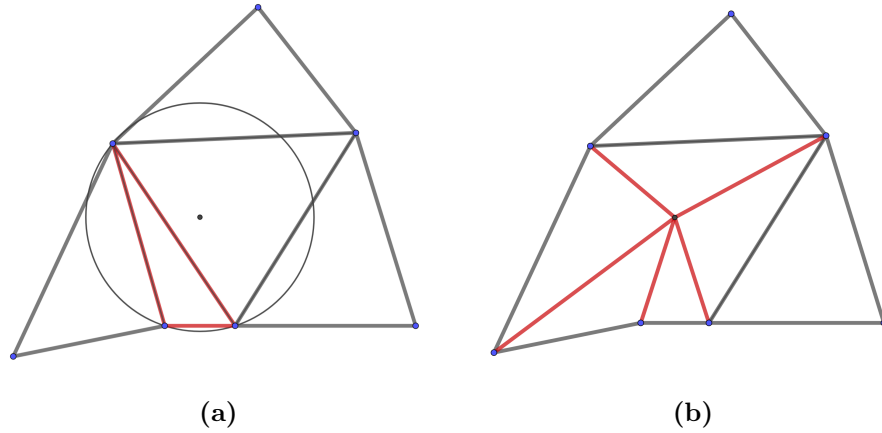
Anschließend wird, wie bei der Behandlung der *encroached subsegments* weiter verfahren.

**Fall 2:** Der neue Knoten ruft keine *encroached subsegments* hervor. Dann darf der Knoten eingefügt werden und das *skinny triangle* wird eliminiert. Der Übergang von Abbildung 10(a) zu Abbildung 10(b) veranschaulicht diesen Vorgang.

Diese Fallunterscheidung ist notwendig, damit der Algorithmus nicht in eine Endlosschleife gerät, indem er im 2. Schritt wieder neue *encroached subsegments* produziert.

Die Reihenfolge der beiden Regeln garantiert, dass keine Knoten außerhalb des Graphen eingefügt werden.

Angenommen, die segment-gebundene Triangulation  $T$  enthält keine *encroached subsegments*. Außerdem sei  $v$  der Umkreismittelpunkt eines Dreiecks  $t$  aus  $T$  und  $c$  der Schwerpunkt von  $t$  welcher offensichtlich innerhalb von  $t$  liegt, so liegt  $v$  immer innerhalb von  $T$ .

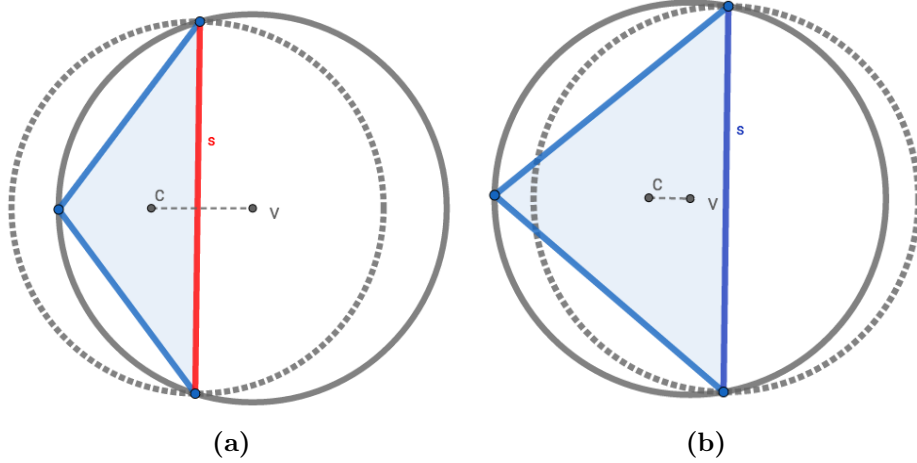


**Abbildung 10:** Entfernen eines *skinny triangles* durch das Einfügen eines Knotens am Umkreismittelpunkt

Um einen Widerspruchsbeweis zu führen, nehmen wir an  $v$  würde außerhalb von  $T$  liegen. So schneidet die Kante  $\overline{cv}$  zwangsläufig ein Subsegment  $s$ , wie in Abbildung 11(a) zu sehen ist. Da  $\overline{cv}$  komplett vom Umkreis von  $t$  umschlossen ist, muss auch ein Teil von  $s$  im Umkreis enthalten sein. Da nach Annahme keine *encroached subsegments* in  $T$  enthalten sind, können die Endpunkte von  $s$  nicht im Umkreis von  $t$  enthalten sein. Da  $c$  und  $v$  auf unterschiedlichen Seiten von  $s$  liegen, muss der Teil des Umkreises von  $t$ , welcher auf der selben Seite von  $s$  wie  $c$  liegt komplett im Durchmesserkreis von  $s$  enthalten sein. Da jeder Knoten von  $t$  auf dem Umkreis von  $t$  liegt, ist ein Knoten entweder Endknoten von  $s$ , oder er liegt auf der selben Seite von  $s$  wie  $c$ . Maximal zwei Knoten können Endknoten von  $s$  sein, aber mindestens ein Knoten liegt dann im Durchmesserkreis von  $s$ , was einen Widerspruch zur Annahme darstellt, dass die Triangulation keine *encroached subsegments* mehr enthält.

Neben  $B$  kann auch eine obere Schranke für die Größe der resultierenden Dreiecke festgelegt werden. Dabei bezieht sich die Schranke auf die Fläche des Dreiecks oder auf dessen Kantenlänge. Dies spielt z. B. bei der finiten Elemente Methode eine wichtige Rolle. Enthält das Delaunay-Netz sehr große Dreiecke, wird dadurch die Berechnung ungenau. Zu große Dreiecke werden analog zu *skinny triangles* behandelt.

Wie garantiert Ruppert mit seinem Algorithmus ein schön abgestuftes Netz? Um diese Frage zu beantworten, führt Ruppert eine neue Funktion ein, die er *local feature size (lfs)* nennt. Die *lfs* eines Punktes ist der kleinste Kreis, welcher zwei nicht inzidente Elemente des Graphen schneidet wie, in Abbildung 12 zu sehen. Damit formalisiert Ruppert den kleinstmöglichen Abstand eines Punktes, um zu zeigen,



**Abbildung 11:** (a) Ein Umkreismittelpunkt  $v$  kann nur außerhalb der Triangulationsdomäne  $T$  liegen, wenn diese *encroached subsegments* enthält. (b) Da dies nach Definition des Algorithmus ausgeschlossen ist, kann es keine neuen Knoten außerhalb von  $T$  geben.

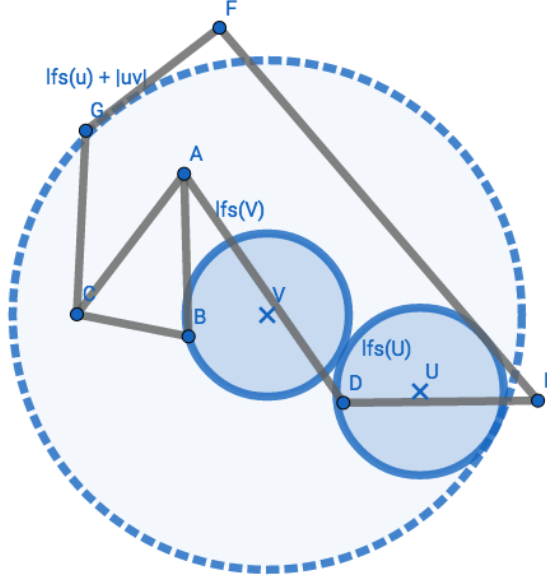
dass der Abstand der Knoten an jeder beliebigen Stelle des Netzes innerhalb eines konstanten Faktors des kleinstmöglichen Abstandes liegt.

Diese *local feature size* ist relativ zum PSLG definiert und proportional zum kleinstmöglichen Abstand der Knoten. Dies gilt in jeder Triangulation, die keine spitzen Dreiecke hat. Die Eigenschaft, dass die Funktion kontinuierlich ist und ihre Richtungsableitungen im Bereich  $[1, -1]$  begrenzt sind, führt zu einer unteren Grenze der Schrittweite mit welcher die Kantenlängen von groß nach klein steigen, während man sich von einem kleinen *local feature* entfernt. Weitere Details dazu findet man bei [7].

Dies kann bewiesen werden, indem wir für zwei beliebige Punkte  $u$  und  $v$  in der Ebene zeigen, dass die *local feature size* eines Punktes  $v$  durch  $lfs(u) + |uv|$  begrenzt ist. Der Kreis mit dem Radius  $lfs(u)$ , der auf  $u$  zentriert ist, schneidet zwei nicht inzidente Merkmale von PSLG. Der Kreis mit dem Radius  $lfs(u) + |uv|$  zentriert auf  $v$  enthält den vorherigen Kreis und damit auch die gleichen beiden Merkmale. Daher hat der kleinste Kreis mit Radius  $lfs(v)$ , zentriert in  $v$ , einen Radius der nicht größer als  $lfs(u) + |uv|$  ist.

### 3.2 Terminierungsnachweis

Wie wir gesehen haben, kann die Behandlung eines Problems, ein neues hervorrufen. Zum Beispiel kann beim eliminieren eines *skinny triangles* ein Neues entstehen. Es stellt sich die Frage, wie garantiert werden kann, dass dieser Prozess jemals zum Ende kommt.



**Abbildung 12:** Der Radius der dunklen Kreisflächen entspricht der *local feature size* des Kreismittelpunktes. Es gilt dabei stets  $lfs(v) \leq lfs(u) + |uv|$

Der Algorithmus terminiert nur unter den bereits erwähnten Bedingungen, dass für die obere Schranke  $B \geq \sqrt{2}$  gilt, sowie, dass kein Winkel zwischen zwei inzidenten Segmenten im PSLG kleiner als  $60^\circ$  sein darf. Um den Beweis zu führen, benötigen wir allerdings noch weitere Definitionen.

Ein *mesh vertex* (Netzknoten) ist ein Knoten, der erfolgreich in den Graphen eingefügt wurde, oder als *input vertex* (Eingabeknoten) schon im *input mesh* (CDT des PSLG) vorhanden war.

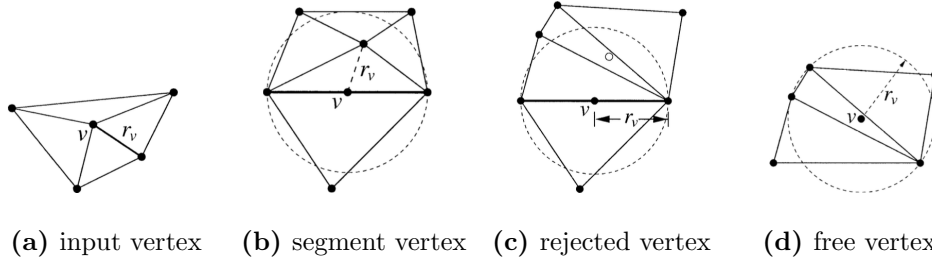
Ein *rejected vertex* (verworfenen Knoten) ist ein Knoten, welcher aufgrund eines *skinny triangles* eingefügt werden sollte, jedoch nicht eingefügt wurde, da er ein Subsegment *encroached* (beeinflusst) hätte.

Für jeden Netzknoten bzw. verworfenen Knoten  $v$ , gibt es einen sogenannten *insertion radius*  $r_v$  (Einfügeradius) dieser ist gleich der Länge der kürzesten Kante an  $v$ , nachdem  $v$  in den Graphen eingefügt wurde. Es ist wichtig zu betonen, dass hier nur die Knoten betrachtet werden, die zu genau dem Zeitpunkt existieren, als  $v$  eingefügt bzw. verworfen wurde. Man unterscheidet dabei folgende Fälle:

- Der Knoten  $v$  ist ein Eingabeknoten. Dann ist  $r_v$  die euklidische Distanz von  $v$  zum nächsten sichtbaren Eingabeknoten, siehe Abbildung 13(a).
- Der Knoten  $v$  wurde in der Mitte eines *encroached subsegments* eingefügt. Dann ist  $r_v$  die Distanz zum nächsten beeinflussenden Knoten im Graphen zu diesem Zeitpunkt, siehe Abbildung 13(b). Ist dieser nichtmehr vorhanden, weil

er verworfen wurde, so entspricht  $r_v$  der Länge einer der beiden Subsegmente, die durch  $v$  erzeugt wurden, wie in Abbildung 13(c) zu sehen.

- Der Knoten  $v$  wurde als Umkreismittelpunkt eines *skinny triangles* eingefügt, um dieses zu eliminieren. So ist  $r_v$  der Umkreisradius dieses *skinny triangles* vgl. Abbildung 13(d).



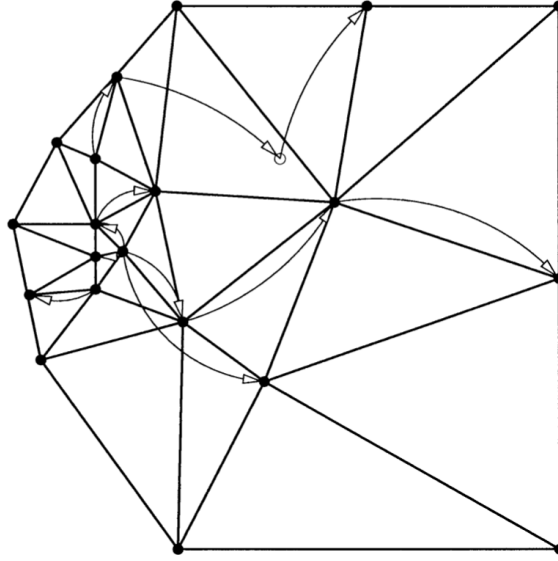
**Abbildung 13:** Der *insertion radius*  $r_v$  eines Knoten  $v$  ist abhängig von seiner Herkunft. [1]

Alle Knoten, außer Input Knoten, haben einen *parent vertex*  $p(v)$  (Elternknoten vom Knoten  $v$ ). Auch hier wird wieder über die Ursache, weshalb ein Knoten in den Graphen eingefügt wurde, entschieden, welcher Knoten sein *parent vertex* ist:

- Wird ein Knoten  $v$  wegen eines *encroached subsegments* eingefügt, so ist  $p(v)$  der beeinflussende Knoten. Gibt es mehrere Knoten in diesem Kreis, ist der Elternknoten der, welcher als nächstes zu  $v$  liegt.
- Wird ein Knoten  $v$  als Umkreismittelpunkt eines *skinny triangles* eingefügt, so ist  $p(v)$  der zuletzt eingefügte Knoten auf der kürzesten Kante des Dreiecks. Sind beide Knoten Eingabeknoten des Netzes, kann einer der beiden willkürlich gewählt werden.

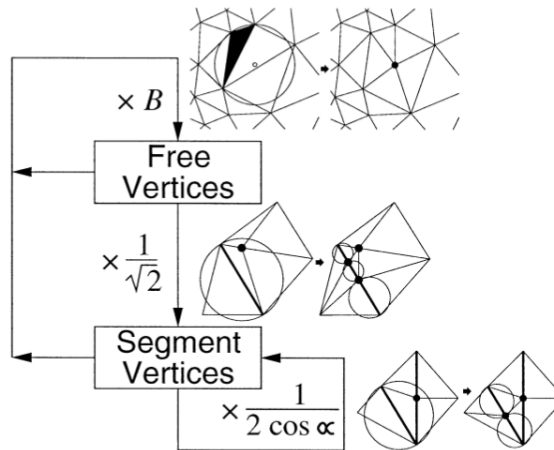
Durch diese Beziehung entsteht an jedem Input Knoten ein Baum, durch welchen sich der Ursprung aller enthaltenen Knoten zurückverfolgen lässt, wie in Abbildung 14 zu sehen ist.

Da wir nun mit dem nötigen Wissen gerüstet sind, wollen wir uns dem Beweis dieser Aussage annehmen. Wir wollen zeigen, dass es nicht möglich ist, immer kleinere Kanten zu produzieren, wodurch der Algorithmus aus Platzgründen terminieren muss. Dazu betrachten wir die Beziehung zwischen dem *insertion radius*  $r_v$  eines Knotens mit dem  $r_p$  seines *parent vertex* genauer. Eine Ausnahme stellen *input vertices* dar, da sie keine Elternknoten haben. Ein gutes Hilfsmittel zur Darstellung dieser Beziehungen ist das Flussdiagramm in Abbildung 15 aus [1]. Ein Pfeil zeigt



**Abbildung 14:** *Parent vertices* zeigen die Geschichte eines Graphen wie in [1] dargestellt.

jeweils vom Elternknoten auf seinen Nachfahren. Der Wert auf dem Pfeil gibt den Faktor an mit dem sich  $r_v$  maximal verändert. Beachten wir nun die Bedingungen die wir zu Beginn an den Eingangsgraphen gestellt haben -  $B \geq \sqrt{2}$  und kein Winkel zwischen inzidenten Segmenten  $\leq 60^\circ$  - so wird klar, dass in dem Flussdiagramm keine Schleife möglich ist, bei welcher der Faktor  $< 1$  ist. Dies bedeutet, dass keine kleineren Kanten, als die kleinste schon im Eingangsgraphen (CDT) vorhandene Kante entstehen können. Dies garantiert, dass der Algorithmus zum Stehen kommt.

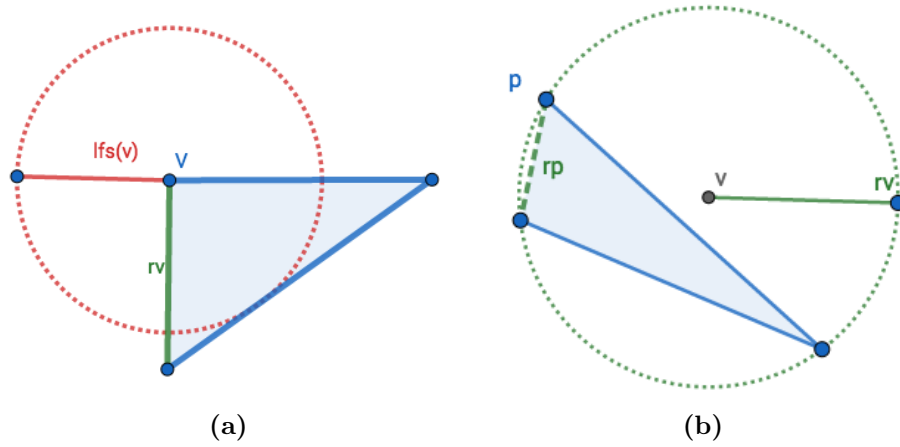


**Abbildung 15:** Unter den geltenden Bedingungen:  $B \geq \sqrt{2}$ , kein Winkel ist  $\leq 60$ , kann keine Schleife mit Faktor  $< 1$  entstehen.

Für alle Kanten zu Kindern von *input vertices* ist eine untere Grenze durch  $lfs(v)$ , wie in Abbildung 16(a), gegeben.

Woher diese Werte aus dem Flowgraph stammen, schauen wir uns jetzt genau-





**Abbildung 16:** (a) Für alle *input vertices*  $v$  gilt  $lfs(v) \leq r_v$ , d.h. der *insertion radius* eines neuen Knotens ist durch  $lfs(v)$  nach unten begrenzt. (b) Ist  $v$  ein freier Knoten, hat jede neue Kante mindestens  $r_v \geq Br_p$  als Kantenlänge. Zur Erinnerung:  $B$  ist die obere Grenze des Verhältnisses von Umkreisradius zur kürzesten Kante eines Dreiecks. Dieser Fall wird im Flowchart aus Abbildung 15 durch den Pfeil auf *free vertices* abgebildet.

er an. Für alle aufgrund von *skinny triangles* eingefügten Knoten (*free vertices*) gilt:  $r_v \geq Br_p$ , siehe Abbildung 16(b). Ist  $v$  ein aufgrund eines *encroached subsegments*  $s$  eingefügter Knoten (*segment vertex*), so muss man 4 Fälle unterscheiden. Allerdings sind die ersten zwei ausreichend, um die Endlichkeit des Algorithmus zu beweisen, falls im Input-Graphen alle Winkel  $\geq 90^\circ$  sind.

- Der Elternknoten  $p$  ist Input Knoten oder auf einer Kante, die nicht inzident zu  $s$  ist. Nach Definition gilt:  $lfs(v) \leq r_v$ .
- Der Elternknoten  $p$  ist ein rejected vertex. Da der zu triangulierende Graph constrained Delaunay ist, kann der Umkreis von  $p$  keinen Endpunkt von  $s$  enthalten, daher gilt:  $r_v \geq \frac{1}{\sqrt{2}} * r_p$ , siehe Abbildung 17(a).
- Die Knoten  $v$  und  $p$  liegen auf inzidenten Segmenten mit einem Winkel  $45^\circ \leq \alpha < 90^\circ$ . Dann gilt:  $r_v > \frac{r_p}{2 \cos \alpha}$ .  $r_v$  wird dann am kleinsten, wenn  $p$  auf dem Durchmesserkreis von  $s$  liegt, vgl. Abbildung 17(b).
- Die Knoten  $v$  und  $p$  liegen auf inzidenten Segmenten mit einem Winkel  $\alpha \leq 45^\circ$ . So gilt:  $r_v \leq r_p \sin \alpha$ .  $r_v$  ist dann minimal, wenn  $v$  die orthogonale Projektion von  $p$  auf  $s$  ist, vgl. Abbildung 17(c).

Somit haben wir gezeigt, dass unter den vorausgesetzten Bedingungen keine Schleife mit Faktor  $< 1$  im Flussdiagramm Abbildung 15 möglich ist.

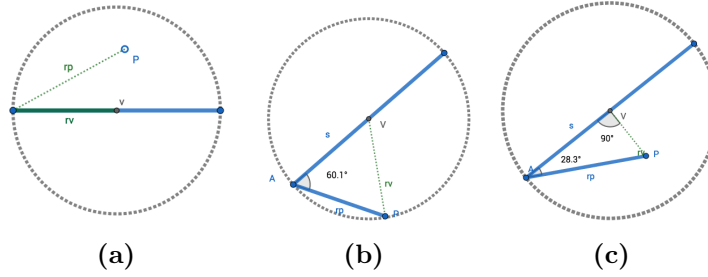


Abbildung 17: Ist  $v$  ein *segment vertex* werden vier Fälle unterschieden.

## 4 Delaunay Verfeinerungsalgorithmus von Chew

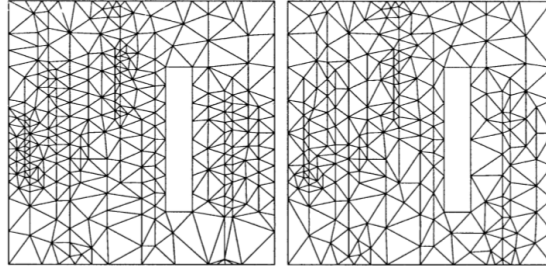
Chews zweiter Algorithmus ist dem von Ruppert grundsätzlich sehr ähnlich. Im direkten Vergleich bietet dieser jedoch Vorteile, welche nochmals zu einer Verbesserung der Vernetzungsqualität führen können.

Voraussetzung für eine garantierte Terminierung ist auch hier, dass im PSLG kein Winkel zwischen zwei inzidenten Segmenten kleiner als  $60^\circ$  sein darf. Für den Wert der in Abschnitt 2.3.1 definierten Schranke muss  $B \geq \sqrt{5}/2$  gelten. Dadurch beträgt der kleinstmögliche, in der Triangulation auftretende Winkel  $26,5^\circ$ . Das ist gegenüber Rupperts Verfahren, welches einen Mindestwinkel von  $20,7^\circ$  garantiert, eine Verbesserung von fast  $5^\circ$ . Daraus ergibt sich auch eine höhere Qualität der Triangulation. Denn wie bereits in Abbildung 6 gezeigt wurde, gilt, dass je größer die Innenwinkel eines Dreiecks sind, desto gleichschenkliger und somit homogener werden die erzeugten Dreiecke. Zum anderen werden bei Chews Algorithmus manche Segmentzerteilungen vermieden, wodurch je nach Geometrie die Gesamtzahl der erzeugten Dreiecke deutlich geringer ausfallen kann. Das macht sich vor allem bei PSLG mit langen Kanten bemerkbar, da hier häufiger *encroached subsegments* auftreten. In Abbildung 18 welche einen Vergleich der Triangulation zwischen Ruppert und Chew zeigt, wird deutlich, dass Chews zweiter Algorithmus unter den gegebenen Umständen deutlich weniger Dreiecke erzeugt.

Ein möglicher Nachteil von Chew ist, dass nur eine Constrained Delaunay Eigenschaft garantiert werden kann. Wird die Delaunay Eigenschaft zwingend benötigt, sollte deshalb mit Rupperts Algorithmus gearbeitet werden.

### 4.1 Beschreibung des Algorithmus

Wie in [1] beschrieben beginnt der Algorithmus mit einer Constrained Delaunay Triangulation eines PSLG. Anschließend wird diese Triangulation verfeinert, indem schrittweise alle als *skinny triangle* klassifizierten Dreiecke aus der Triangulation



**Abbildung 18:** Wie in [1] dargestellt, zeigt die linke Abbildung eine Triangulation mittels Ruppert's Algorithmus, während die rechte von Chew's zweitem Algorithmus erzeugt wurde.

entfernt werden. Wie in Abbildung 10(a) und (b) dargestellt, wird in jedem Schritt der Umkreismittelpunkt  $c$  des *skinny triangles* als Knoten in die Triangulation eingefügt. Anschließend wird das *skinny triangle* entfernt. Der neue Knoten wird dann mit allen Knoten verbunden, welche nicht durch eine Kante bzw. Subsegment von ihm getrennt sind.

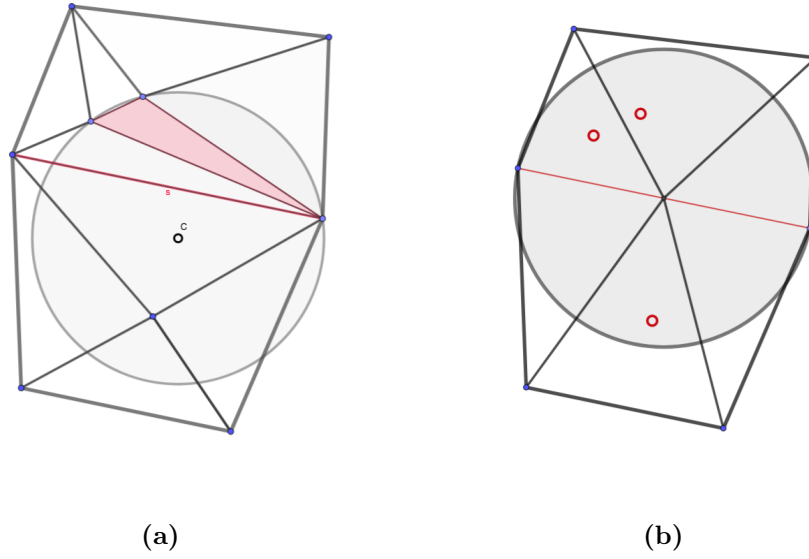
Anders als Ruppert prüft Chew vor dem Abarbeiten der *skinny triangles* nicht, ob *encroached subsegments* vorliegen. So kann es beim Beseitigen der *skinny triangles* vorkommen, dass sich der Umkreismittelpunkt  $c$  eines *skinny triangles*  $t$  auf der anderen Seite eines Segments befindet, wie das in Abbildung 19(a) der Fall ist.

In diesem Fall kann das *skinny triangle*  $t$  nicht auf herkömmliche Weise aufgelöst werden. Stattdessen wird der Umkreismittelpunkt  $c$  als neuer Knoten zurückgewiesen und alle freien Knoten, welche sich innerhalb des Durchmesserkreises des Segments  $s$  befinden und gleichzeitig von dessen Mittelpunkt aus sichtbar sind, werden entfernt. In Abbildung 19(b) sind diese als rote Kreise gekennzeichnet. Anschließend wird ein neuer Knoten am Mittelpunkt des Subsegments  $s$  eingefügt und die übrigen Knoten damit verbunden.

Falls mehrere Subsegmente zwischen dem *skinny triangle* und dem Umkreismittelpunkt liegen, so wird nur das Subsegment, welches am nächsten zu  $t$  liegt gesplittet. Befindet sich kein Subsegment zwischen  $t$  und  $c$ , aber liegt  $c$  genau auf einem Subsegment, so wird dieses wie ein *encroached subsegment* bei Ruppert behandelt. Hier wird also der neue Knoten im Mittelpunkt des betroffenen Segments eingefügt.

## 4.2 Terminierungsnachweis

Auch bei Chews Algorithmus stellt sich die Frage, wie sichergestellt wird, dass der Algorithmus garantiert zu einem Ende kommt. Wie in Abschnitt 3.1 gezeigt, gilt für den Fall, dass wenn ein Dreieck und sein Umkreismittelpunkt auf gegensätzlichen



**Abbildung 19:** Triangulation, wenn sich *skinny triangle*  $t$  und Umkreismittelpunkt  $c$  auf unterschiedlichen Seiten eines Segments  $s$  befinden

Seiten eines Subsegments liegen, mindestens ein Knoten des Dreiecks, welcher nicht auch Endpunkt des betrachteten Subsegments ist, auf, oder innerhalb des Durchmesserkreises des Subsegments liegt. Im Beispiel aus Abbildung 19 liegen sogar zwei Knoten innerhalb des Durchmesserkreises.

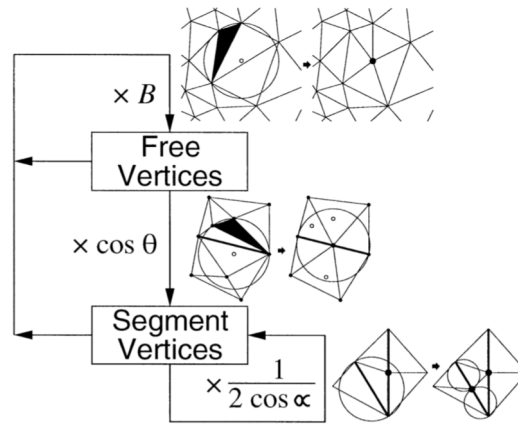
Das bedeutet, dass Chews Algorithmus niemals ein Subsegment zerteilt, welches Ruppert nicht auch zerteilen würde. Daraus folgt auch, dass die für Ruppert's Algorithmus in Abbildung 17 dargestellten Verhältnisse für Chew's Algorithmus genauso gelten. Mit einer Ausnahme, denn im Vergleich zu Rupperts Algorithmus wird Chew oft Zerteilungen zurückweisen und somit insgesamt weniger Subsegmente zerteilen. Als Konsequenz kann das Verhältnis zwischen *insertion radius*  $r_v$  des neuen Knotens und dem seines Elternknotens  $r_p$  enger gestaltet werden:

Sei  $\theta = \arcsin \frac{1}{2B}$  der Winkel, ab dem ein Dreieck als *skinny triangle* klassifiziert wird und  $s$  ein encroached subsegment, weil ein *skinny triangle* und sein Umkreismittelpunkt  $c$  auf verschiedenen Seiten von  $s$  (oder genau darauf) liegen. Dabei ist  $v$  der Knoten, welcher im Mittelpunkt von  $s$  eingefügt wird, vgl. Abbildung 19b. Dann gilt eine der folgenden vier Aussagen:

1.  $r_v \geq lfs(v)$ , entsprechend zu Abbildung 20(a).
2.  $r_v \geq \frac{r_p}{2 \cos \alpha}$ , für  $\alpha \geq 45^\circ$  entsprechend zu Abbildung 17(c).
3.  $r_v \geq r_p \sin \alpha$  für  $\alpha \leq 45^\circ$ , entsprechend zu Abbildung 17(d).



die unterste Grenze, die diese Voraussetzung erfüllt, mit  $B = \sqrt{5/2} = 1.12$  gegeben. Das entspricht einem Winkel von  $\arcsin 1/\sqrt{5} = 26.56^\circ$ .



**Abbildung 21:** Flowchart von Chew's Algorithmus

## 5 Zusammenfassung und Ausblick

Geometrische Triangulationen sind für eine Vielzahl von Anwendungen von großer Bedeutung. Die hier gezeigten Algorithmen von Ruppert und Chew gehören zu den etabliertesten Verfahren zur Delaunay bzw. Constrained Delaunay Triangulation. Mit Rupperts Algorithmus wurden wichtige Konzepte eingeführt, auf denen auch Chews zweiter Algorithmus aufbaut. Dabei ist es essenziell, das Konzept der Beziehungen zwischen *insertion radien* der Eltern- und Kindsknoten zu verstehen.

Durch das Entfernen von Knoten aus der Triangulation hat es Chew geschafft, dass kein Winkel kleiner als  $26,7^\circ$  in der Triangulation auftreten kann. Damit hat er den Mindestwinkel im Vergleich zu Rupperts Algorithmus um knappe  $5^\circ$  angehoben und kann so gleichmäßigere Triangulationen erreichen. Dadurch, dass zusätzlich weniger Segmente zerteilt werden, schafft Chew unter gegebenen Umständen eine deutlich bessere Netzqualität.

Die hier vorgestellten Algorithmen setzen voraus, dass die Eingabegeometrie keine Winkel kleiner als  $60^\circ$  beinhaltet. In der Praxis sind kleine Winkel ein großes Problem für eine qualitativ hochwertige Triangulation. In [1] werden Methoden vorgestellt, wie man mit solchen kleinen Winkeln umgehen kann um trotzdem gute Ergebnisse zu erzielen. Zusätzlich wird auch ein Algorithmus namens *the Terminator* eingeführt, der diese Methoden nutzt, um bessere Triangulationen mit größeren Mindestwinkel von  $30^\circ$  zu erzeugen.

# Literatur

- [1] Shewchuk JR. Delaunay Refinement Algorithms for Triangular Mesh Generation. Computational Geometry. 2002;22:21–74.
- [2] Wang B, Khoo BC, Xie ZQ, Tan ZJ. Fast centroidal Voronoi Delaunay triangulation for unstructured mesh generation. Journal of Computational and Applied Mathematics. 2015;280:158–173.
- [3] Peters G. Interaktive Systeme II. FernUniversität Hagen. 2016;p. 204.
- [4] Floussov I, Kelly S; [cited 05. Februar 2017]. Available from: <https://www.conceptfarm.ca/our-work/#image-triangulator>.
- [5] Kobbelt LP, Stamminger MF, Seidel HP. Density Estimation on Delaunay Triangulations. IEEE conference proceedings. 1998;p. 307–310.
- [6] Tanski G; [cited 20. Dezember 2017]. Available from: [https://en.wikipedia.org/wiki/Global\\_illumination](https://en.wikipedia.org/wiki/Global_illumination).
- [7] Ruppert J. A Delaunay refinement Algorithm for quality 2-dimensional mesh generation. Journal of Algorithms. 1995;18(3):548–585.