Günter PAULINI

# Convex Shape Delaunay Triangulierungen

## DIPLOMARBEIT

zur Erlangung des akademischen Grades eines
Diplom-Ingenieurs

Diplomstudium Technische Mathematik


Graz University of Technology

Technische Universität Graz

Betreuer:
Univ.-Prof. Dipl.-Ing. Dr.techn. Franz AURENHAMMER

Institut für Grundlagen der Informationsverarbeitung

Graz, im Februar 2012

# Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ..... .....

(Unterschrift)

# Inhaltsverzeichnis

# Abstract

Eine konvexe Distanzfunktion (in der Ebene) erhält man, indem man den euklidischen Einheitskreis durch ein konvexes Objekt (Shape) ersetzt. Die vorliegende Arbeit ist dem Studium von Delaunay Triangulierungen und verwandten Strukturen auf der Basis von konvexen Distanzfunktionen gewidmet.
Zielsetzung ist es, solche sogenannten *Convex Shape Delaunay Triangulierungen* daraufhin zu untersuchen, inwieweit die zahlreichen Eigenschaften der klassischen (euklidischen) Delaunay Triangulierungen analog für konvexe Distanzfunktionen erhalten bleiben (zumindest in adaptierter Form), und im Zuge dessen noch ungeklärte Fragen zu beantworten.
Im Zuge einer gründlichen Studie der strukturellen Eigenschaften von Shape Delaunay Triangulierungen (abhängig von der Form des gegebenen Shapes), welche im Allgemeinen keine vollständigen Triangulierungen mehr darstellen (und im Extremfall kein einziges Dreieck beinhalten), werden auch relevante Teilgraphen untersucht. Während die klassische Delaunay Triangulierung immer den minimalen Spannbaum und den Gabrielgraphen als Teilgraphen enthält, wird für Shape Delaunay Triangulierungen durch Symmetrisierung eine dem Shape zugeordnete neue Metrik definiert (konvexe Distanzfunktionen entsprechen i.A. keiner Metrik mehr), mit welcher verwandte Ergebnisse erzielt werden können.
Insbesondere ist der Gabriel Graph für Dreiecks-Shapes identisch mit der Delaunay Triangulierung.

Die elementaren Umformungsoperationen in Triangulierungen (die sogenannten Flips) werden für den Fall konvexer Shapes sorgfältig untersucht. Es wird gezeigt, dass (wie im klassischen Fall) durch verbessernde Flips die Shape Delaunay Triangulierung stets erzeugbar ist, auch wenn es sein kann, dass sie im eigentlichen Sinne keine Triangulierung mehr ist (die vollständige Triangulierung wird jedoch als Datenstruktur beibehalten, damit man es nachwievor nur mit Dreiecken zu tun hat, die ungültigen Kanten müssen allerdings gesondert berücksichtigt werden). Winkel-basierte Optimalitätskriterien sind nicht mehr sinnvoll, stattdessen muss auf passende Umkreiskriterien ausgewichen werden.
Als "Nebenprodukt" wurde für die Ausführung des Incircle-Tests im euklidischen Fall (Punkt-Inklusion im Kreis) eine neue effiziente Methode entwickelt, welche im Vergleich zu bekannten Verfahren ein paar Rechenoperationen (speziell Multiplikationen) einspart.
Ein Einfüge-Algorithmus und ein Flip-Algorithmus zur Erzeugung von Shape Delaunay Triangulierungen werden in adaptierter Form gezeigt.

Zur Darstellung von Shape Delaunay Triangulierungen und entsprechenden Voronoi Diagrammen, sowie Gabriel Graphen und minimalen Spannbäumen etc., wurde zusätzlich ein Testprogramm implementiert.

# 1 Einleitung

## 1.1 Voronoi Diagramm

Ein *Voronoi Diagramm* (benannt nach dem russischen Mathematiker Georgi Feodosjewitsch Woronoi, 1868-1908) ordnet jedem Punkt $p$ einer gegebenen Punktmenge $S$ jene Region des Raumes zu (bzw. der Fläche im 2-dimensionalen Fall), für welche gilt, dass alle Punkte der Region dem Punkt $p$ näher sind, als allen anderen Punkten aus $S$. Die Bisektoren (mit gleichem Abstand zwischen 2 Punkten aus $S$) bilden die Grenzlinien zwischen den Regionen.

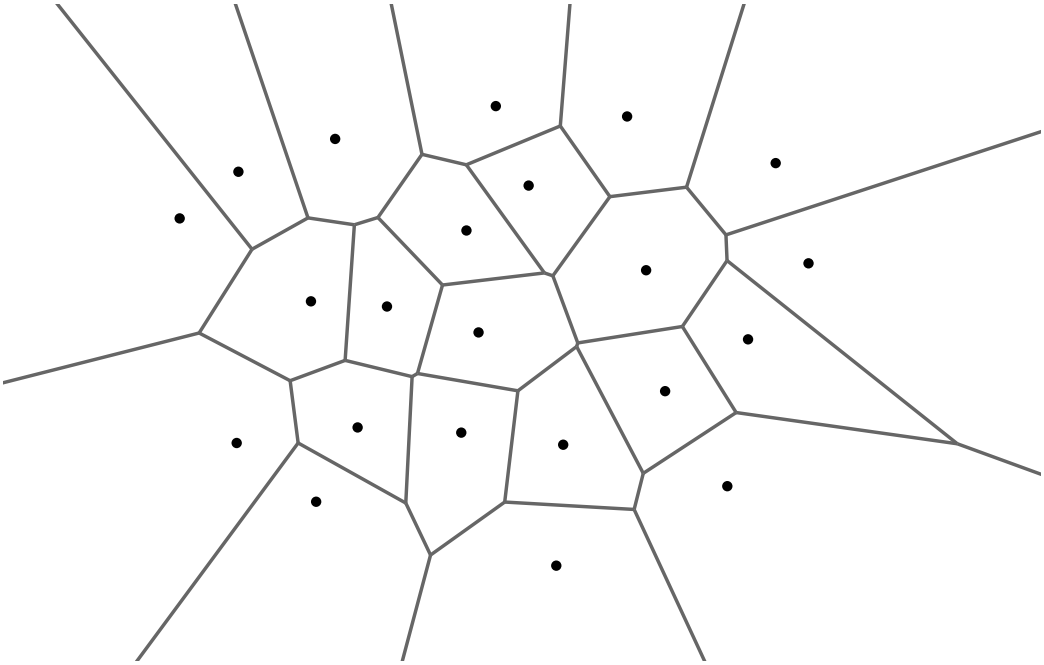

Abbildung 1: Beispiel eines Voronoi-Diagramms zu einer Punktmenge $S$ in der Ebene.

Voronoi-Diagramme (auch Thiessen-Polygone oder Dirichlet-Zerlegungen genannt) kommen abgesehen von der Algorithmischen Geometrie auch in weiteren unterschiedlichsten wissenschaftlichen Bereichen zum Einsatz, wie z.B. in der Biologie, Chemie, Kristallographie und Meteorologie.
Es gibt zudem zahlreiche Verallgemeinerungen bzw. Erweiterungen (vergl. F. Aurenhammer[4] bzw. F. Aur./R. Klein[5]), wie z.B. gewichtete Abstände, die Verwendung von allgemeineren Objekten statt Punkten (z.B. Liniensegmente), oder allgemeinere Metriken/Distanzfunktionen an Stelle der euklidischen

Metrik.

## 1.2 Delaunay Triangulierung

Die *Delaunay Triangulierung* (benannt nach dem russischen Mathematiker Boris Nikolajewitsch Delone [engl. Delaunay], 1890-1980) ist zum einen der duale Graph zum Voronoi-Diagramm (d.h. jeweils 2 Punkte, deren Regionen im Voronoi-Diagramm eine gemeinsame Grenze aufweisen, sind in der Delaunay-Triangulierung mit einer Kante verbunden), zum anderen ist sie auch völlig unabhängig davon durch zahlreiche interessante Eigenschaften definierbar (z.B. *Leerer-Kreis-Eigenschaft*, Maximierung des minimalen Dreieckswinkel uvm.) und findet dementsprechend ihre eigenen Anwendungsgebiete (vergl. S. Fortune[7]), wie z.B. bei der *Finite Elemente Methode* (FEM) zur Lösung von partiellen Differentialgleichungen, oder bei 3D-Modellen in der Computergrafik etc.

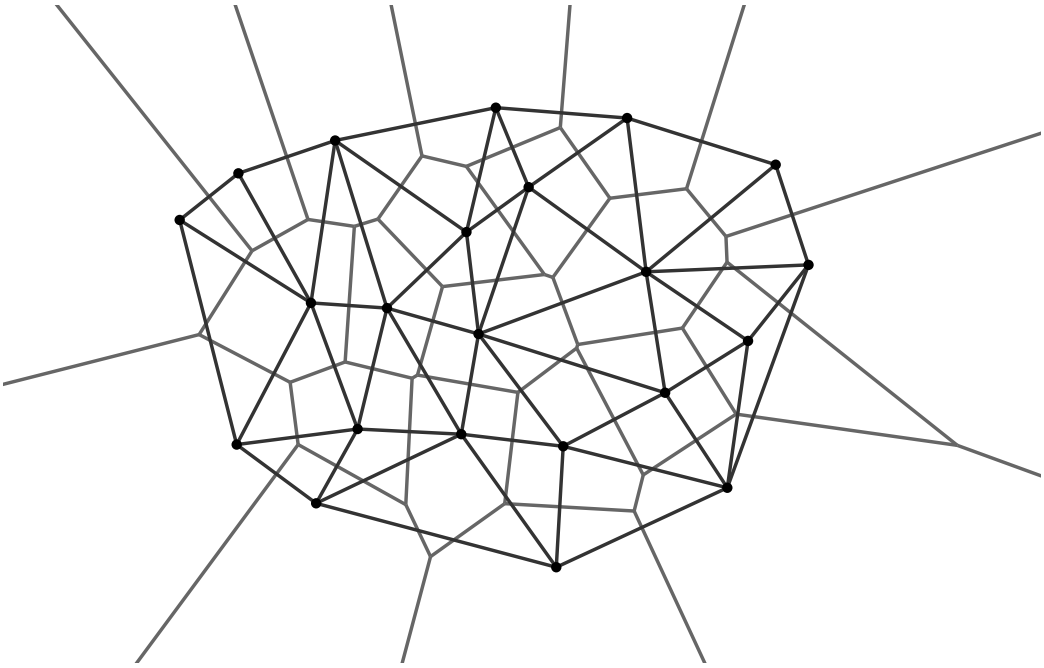

Abbildung 2: Das Beispiel aus Abbildung 1 mit hinzugefügter Delaunay-Triangulierung.

Auf diverse Eigenschaften der Delaunay-Triangulierung wird in Folge

noch näher eingegangen, und zwar im Zuge der Untersuchung, inwieweit diese Eigenschaften für *Convex Shape Delaunay Triangulierungen* (welche auf konvexen Distanzfunktionen basieren) erhalten bleiben.

## 1.3 Konvexe Distanzfunktionen

Während der übliche euklidische Abstand zwischen 2 Punkten in der Ebene wie folgt definiert ist: $\|p - q\| := \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$, kann man die Distanz zweier Punkte auch wesentlich allgemeiner definieren.

Sei $C$ eine konvexe kompakte Menge in der Ebene, die als "Zentrum" den Koordinatenursprung enthält (in Folge auch als konvexer Shape bezeichnet). Für 2 Punkte $p$ und $q$ verschiebe man den Shape $C$ um den Vektor $p$, und schneide die Gerade durch $p$ und $q$ mit dem Rand von $C$ ($r$ sei der Schnittpunkt). Dann ist die konvexe Distanzfunktion $d_C$ von $p$ nach $q$ folgendermaßen definiert:

$$d_C(p, q) := \frac{\|p - q\|}{\|p - r\|}$$

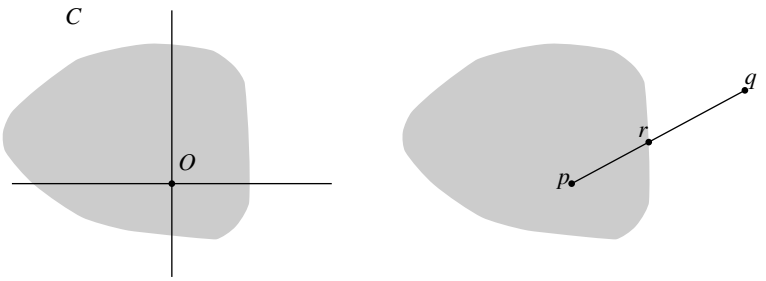


Abbildung 3: Konvexe Distanzfunktion $d_C(p, q)$

Die Funktion $d_C$ erfüllt 2 Bedingungen einer Metrik, zum einen die positive Definitheit ($d_C(p, q) \geq 0$ und $d_c(p, q) = 0 \Leftrightarrow p = q$), und zum anderen die Dreiecksungleichung ($d_C(p, q) \leq d_C(p, r) + d_C(r, q)$). Die Symmetrie ($d_C(p, q) = d_C(q, p)$) als 3. Bedingung gilt im Allgemeinen nicht mehr, sie ist nur dann erfüllt, wenn der Shape $C$ bezüglich des Ursprungs symmetrisch ist (nur in diesem Fall ist $d_C$ eine Metrik).
Eine konvexe Distanzfunktion $d_C$ heißt *glatt*, wenn der Rand von $C$ an jeder Stelle eine eindeutige Tangente hat (also keine Ecken besitzt), und *strikt konvex*, wenn der Rand keine Liniensegmente besitzt.

Ein bekanntes Beispiel von konvexen Distanzfunktion sind die $L_p$-Metriken ($1 \leq p \leq \infty$), definiert durch: $\|v\|_p = \sqrt[p]{|v_x|^p + |v_y|^p}$, wobei die $L_2$-Metrik als Spezielfall der euklidischen Metrik entspricht.
Die $L_p$-Metrik ist für $1 < p < \infty$ strikt konvex und glatt, für $L_1$ und $L_\infty$ gilt hingegen weder das eine noch das andere.

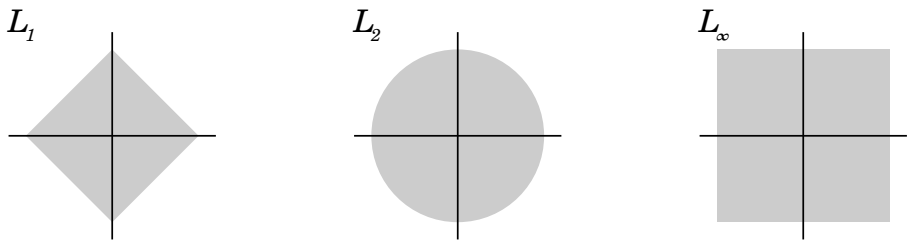


Abbildung 4: Die $L_1$-Metrik entspricht einer $d_C$ mit karo-förmigem Shape $C$, die $L_2$-Metrik entspricht dem Kreis und die $L_\infty$-Metrik dem Quadrat.

Auf Basis konvexer Distanzfunktionen kann man jetzt allgemeinere Voronoi-Diagramme und Delaunay-Triangulierungen betrachten (siehe L.P. Chew/R.L. Drysdale[6] bzw. L. Ma[2]).

Die Definition von gewöhnlichen Delaunay-Triangulierungen erfolgt üblicherweise entweder über den Dualgraph des Voronoi-Diagramms (d.h. in DG(S) sind genau dann 2 Punkte mit einer Kante verbunden, wenn die 2 entsprechenden Regionen in VD(S) einen gemeinsamen Bisektor haben, also benachbart sind) oder über die *Leerer-Kreis-Eigenschaft* (d.h. für jede Delaunay-Kante existiert ein im Inneren punkteleerer Kreis, der die Kanten-Endpunkte am Rand enthält). Für Dreiecke der Triangulierung ist dann der Umkreis im Inneren punktefrei (*Umkreiseigenschaft*).
Falls die Punktmenge $S$ in allgemeiner Lage ist, also keine 4 Punkte kozirkular liegen, sind beide Definitionen äquivalent. Für konvexe Distanzfunktionen ist das im Allgemeinen nicht mehr der Fall, da die Symmetrie-Eigenschaft nicht mehr gegeben ist ($d_C(p,q) \neq d_C(q,p)$).

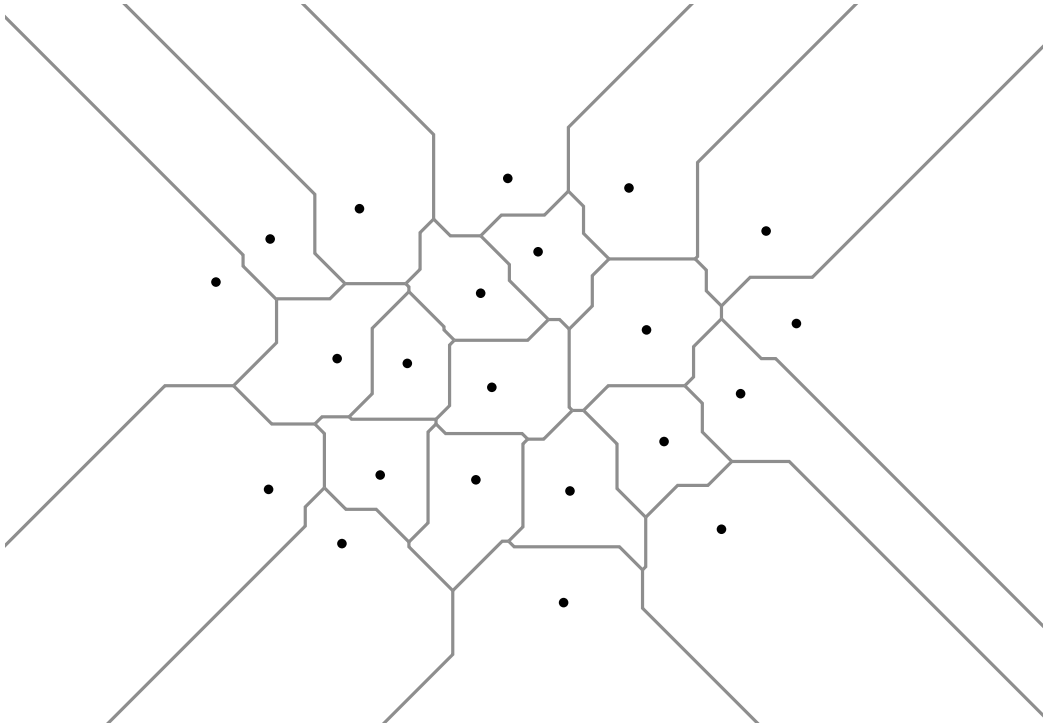

Abbildung 5: Beispiel eines Convex-Shape-Voronoi-Diagramms $VD_C(S)$ mit der $L_\infty$-Metrik. Im Gegensatz zu gewöhnlichen Voronoi-Diagrammen sind hier die Regionen i.A. nicht mehr konvex, sondern "sternförmig".

Sei $VD_C(S)$ das Voronoi-Diagramm bezüglich des konvexen shapes $C$, dann sind dessen Bisektoren definiert durch Punktmengen, für die gilt: $d_C(a_1, p) = d_C(a_2, p)$ mit $a_1, a_2 \in S$. Die Distanzfunktion geht beim VD also von den Knoten aus, während für in die Punktmenge passende Homotheten des Shapes gilt: $d_C(p, a_1) = d_C(p, a_2)$, d.h. die Distanzfunktion geht vom Zentrum des Shapes aus, hin zu den Knoten der Punktmenge.
Da die Gleichheit $d_C(p, q) = d_C(q, p)$ nur für (bezüglich des Zentrums) symmetrische Shapes gilt, und ansonsten $d_C(p, q) = d_{C_R}(q, p)$ ($C_R$ sei der reflektierte Shape), gilt also nur mehr folgende Äquivalenz: $DT_C(S) = DG_{C_R}(S)$ ($DG$ sei der Dualgraph zu $VD$, und $DT$ die über die Kreiseigenschaft definierte Triangulierung).

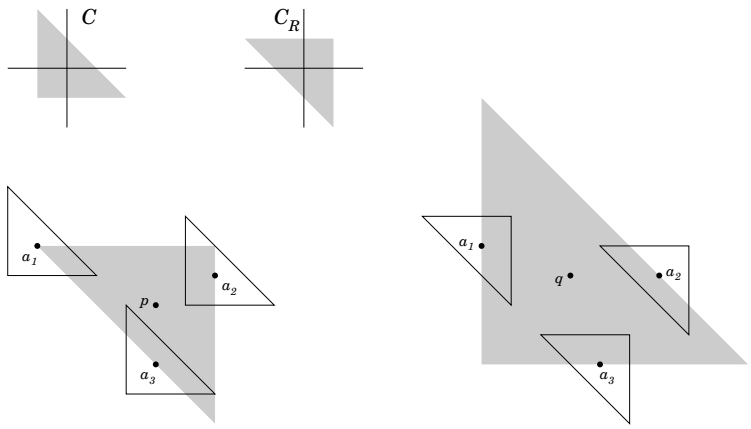


Abbildung 6: $d_C(a_1, p) = d_C(a_2, p) = d_C(a_3, p)$, $d_{C_R}(p, a_1) = d_{C_R}(p, a_2) = d_{C_R}(p, a_3)$, $d_{C_R}(a_1, q) = d_{C_R}(a_2, q) = d_{C_R}(a_3, q)$, $d_C(q, a_1) = d_C(q, a_2) = d_C(q, a_3)$

Falls sich die Punktmenge S nicht in allgemeiner Lage befindet, ist die Äquivalenz nicht mehr vollständig gegeben, da beim Dualgraph DG(S) „Lücken“ entstehen, die im Falle der DT(S) beliebig austrianguliert werden. Einfachstes Beispiel: 4 quadratisch angeordnete Punkte.

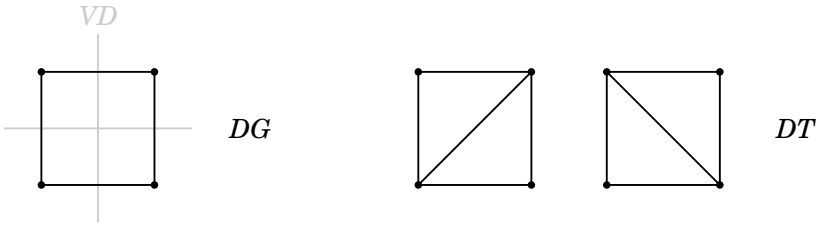


Abbildung 7: Im Falle von Kozirkularität ist DG nicht austrianguliert, während DT nicht mehr eindeutig ist.

Wie von Drysdale[1] definiert, ist die $DT_C(S)$ (*Convex Shape Delaunay Triangulation*) eine maximale Menge kreuzungsfreier Kanten, welche die *Leerer-Kreis-Eigenschaft* erfüllen. Im folgenden Abschnitt 2 werden wir sehen, dass diese Kantenmenge für beliebige Shapes im Allgemeinen die Punktmenge nicht mehr vollständig trianguliert.

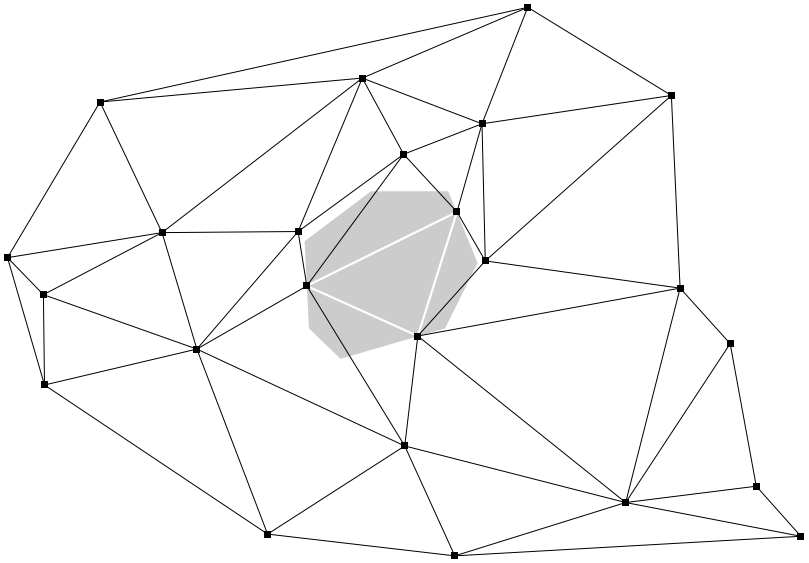

Abbildung 8: Umkreiseigenschaft für Dreiecke

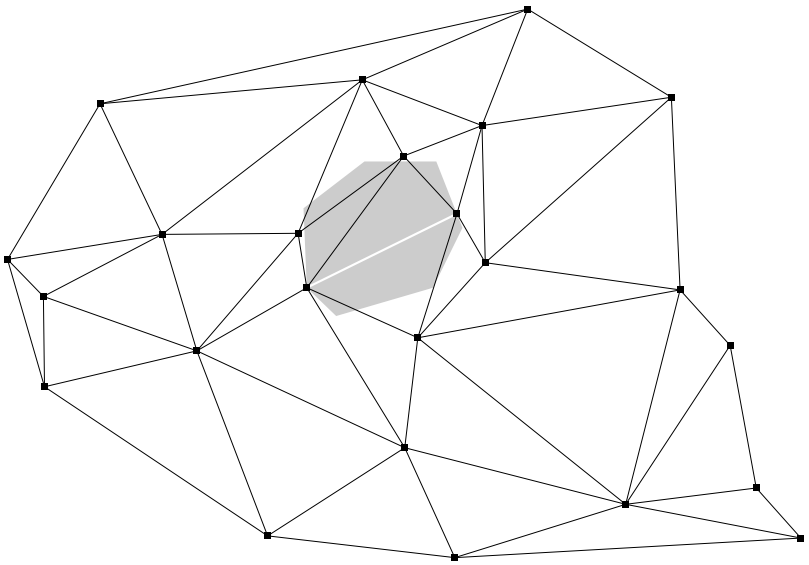

Abbildung 9: Leerer-Kreis-Eigenschaft für Kanten

## 1.4 Ergebnisse der Arbeit

Während die Einführung in erster Linie eine Zusammenfassung von bereits Bekanntem darstellt, liegt der Schwerpunkt der vorliegenden Diplomarbeit in der eigenständigen Erforschung und Bearbeitung von Fragen bezüglich der $DT_C(S)$ (speziell im Vergleich zur gewöhnlichen $DT(S)$), die hier kurz zusammengefasst werden.
Zunächst geht es in Abschnitt 2 darum, Struktureigenschaften der $DT_C(S)$ zu klären, z.B. wie genau und warum sich deren Hülle von der konvexen Hülle der Punktmenge $S$ unterscheidet, und wie sie von der Form des Shapes $C$ abhängt.
Im darauf folgendem Abschnitt wird aus Ermangelung einer Metrik für allgemeine konvexe Shapes eine solche in Form der symmetrischen und zentrumsunabhängigen Distanzfunktion $m_C$ neu definiert, auf deren Basis es möglich wird, einen minimalen Spannbaum bezüglich des Shapes C zu definieren und zu zeigen, und mit dem gewöhnlichen Fall der euklidischen Metrik zu vergleichen. Gleichzeitig kann auch ein Gabrielgraph definiert werden, wo an die Stelle des Durchmesserkreises zwischen zwei Punkten $p$ und $q$ der entsprechende Shape mit Skalierungsfaktor $m_C(p, q)$ tritt. Es wird gezeigt, dass mit $MST_C(S) \subseteq GG_C(S) \subseteq DT_C(S)$ die analogen Eigenschaften zum Fall der gewöhnlichen Delaunay-Triangulierung erhalten bleiben.
Was den Incircle-Test angeht (welcher entscheidet, ob ein Punkt innerhalb des von drei anderen Punkten aufgespannten Umkreises liegt), so bringt die übliche Vorgangsweise, die Punkte auf ein Paraboloid zu projizieren (und den Punkt daraufhin zu testen, auf welcher Seite der von den drei anderen Punkten aufgespannten Ebene er liegt), nicht nur unnötigerweise eine zusätzliche Dimension ins Spiel (welche mit Matrixumformungen wieder weggekürzt werden kann), sondern ist auch wenig geeignet für ein Analogon bezüglich allgemeiner konvexer Shapes. Auf der Suche nach Letzterem ergibt sich ein Lösungsansatz, welcher nicht nur für konvexe Shapes verallgemeinert werden kann, sondern für den gewöhnlichen Fall der euklidischen Metrik im Vergleich zum üblichen Weg effizienter ist, bzw. mit weniger Rechenoperationen auskommt (insbesondere werden Multiplikationen eingespart, die asymptotisch langsamer sind als Additionen/Subtraktionen).
In Abschnitt 5 werden daraufhin die Kanten-Flips näher unter die Lupe genommen. Um zu entscheiden, ob zur Erfüllung der lokalen Delaunay-Eigenschaft ein Flip notwendig ist, ist im Fall von eckigen (also nicht glatten) Shapes oft kein eigentlicher Incircle-Test nötig, bzw. auch gar nicht möglich, denn wenn ein Dreieck ungültig ist (d.h. durch drei gegebene Punkte kann kein Homothet des gegebenen Shapes gelegt werden), so erübrigt sich der

Inklusionstest des vierten Punktes. Speziell, wenn eine betrachtete Kante selbst ungültig ist, kann sie natürlich keine Delaunay-Kante sein und muss geflippt werden, außer die geflippte Kante ist ebenfalls ungültig.

In weiterer Folge werden die Optimierungseigenschaften $DT_C(S)$ untersucht, und es kann die Erhaltung der wichtigen Eigenschaft gezeigt werden, dass für Triangulierungen, welche die lokalen Delaunay-Bedingungen erfüllen, dies auch global gilt, was bedeutet, dass durch lokale Verbesserungen immer das globale Optimum erreicht werden kann (und damit z.B. der Flip-Algorithmus funktioniert, siehe Abschnitt 7.4).
Während das Winkelkriterium als Optimierungseigenschaft für die $DT_C(S)$ nicht mehr einsetzbar ist, hat sich das weniger bekannte Kriterium der Umkreisradien als brauchbar herauskristallisiert. Es wird zuerst gezeigt, dass gewöhnliche Delaunay-Triangulierungen den maximalen (und minimalen) Dreiecks-Umkreisradius minimieren (sowie die Summe der Radien), und dann dargelegt, in welcher Form dieses Kriterium auch für die $DT_C(S)$ bestehen bleibt.

In Abschnitt 7 werden ein paar kleine eigens entwickelten Algorithmen präsentiert, zum einen die Distanzfunktion für konvexe Polygone, wobei man z.B. sehen kann, dass bei ganzzahligem (bzw. rationalem) Input auch der Output eine rationale Zahl bleibt (ohne bei euklidischen Distanzen üblicher Wurzelberechnung), somit ist jeder Entscheidungsalgorithmus, der auf einem Vergleich solcher Distanzen beruht (z.B. Inklusionstest) exakt berechenbar (ohne Rundungsfehler). Zum anderen wird gezeigt, wie z.B. der symmetrische Vereinigungs-Shape $\widehat{C}$ aus $C$ gebildet werden kann (indem man quasi den polygonalen Shape entlang der Kanten um den Ursprung "schiebt", und die neu entstandenen Ecken mitspeichert).
Da $m_C(p,q) = d_{\widehat{C}}(p,q)$ gilt, kann man also die neu definierte Metrik $m_C$ mit Hilfe von $\widehat{C}$ über die übliche Distanzfunktion $d_C$ berechnen. Des weiteren kann man auf diesem Wege auch direkt den minimalen Shape $\overline{C}(p,q)$ ermitteln, denn dessen Skalierungsfaktor ist gleich $m_C(p,q)$, es fehlt also nur der "Ankerpunkt", der in $O(1)$ eruierbar ist, sofern man die Zuordnung der Ecken von $\widehat{C}$ zu jenen von $C$ mitgespeichert hat.
Für den Einfüge-Algorithmus und den Flip-Algorithmus zur Erstellung von Delaunay-Triangulierungen wird eine Adaption gezeigt, die auch für die $DT_C(S)$ funktioniert.
Abschließend wird noch kurz auf die Implementierung des Testprogramms eingegangen (welches vom Zeitaufwand her einen Großteil der Arbeit eingenommen hat), und auf ein paar Fragen, die noch offen geblieben sind.

# 2 Randbereich/Hülle

Während bei gewöhnlichen Delaunay Triangulierungen der Rand immer der konvexen Hülle entspricht, da für deren Kanten zumindest der äußere unendliche Kreis in Form einer Halbebene immer die *Leerer-Kreis-Eigenschaft* erfüllt, ist das im Allgemeinen für konvexe Shapes nicht mehr der Fall, wenn sie nicht glatt sind, sondern Ecken aufweisen (nicht stetig differenzierbar).

Abbildung 10: Unendliche Shapes

Für 2 Punkte aus der Menge $S$, deren Verbindungskante wegen einer Ecke nicht als Tangente an den Shape gelegt werden kann, ist der unendliche Shape keine Halbebene, sondern die von 2 der Ecke entsprechenden Halbstrahlen begrenzte Fläche. Der auf einer Seite der Kante unendliche Shape dringt also mit der Ecke auf die andere Seite der Kante vor - das Innere dieses Dreiecks ist damit ein "verbotener" Bereich, d.h. wenn sich darin ein weiterer Punkt aus S befindet, ist die Kante automatisch ungültig, da sie unmöglich die *Leerer-Kreis-Eigenschaft* erfüllen kann.

Wenn dies auf eine Kante der konvexen Hülle zutrifft, ist sie nicht mehr Teil der Triangulierung, und man benötigt eine allgemeinere Definition der Hülle; Drysdale macht dies mit der support hull ("Stützhülle").
Streng genommen ist die konvexe Hülle per Definition Teil jeder Triangulierung, man könnte aber $DT_C(S)$ als Sub-Triangulierung definieren, da sie zumindest innerhalb der support hull eine vollständige Triangulierung darstellt.
Im Folgenden wird $DT_C(S)$ der Einfachheit halber auch weiterhin als Triangulierung bezeichnet, um sie insbesondere vom Delaunay Graphen $DG_C(S)$ (dem Dualgraph zum Voronoi-Diagramm) zu unterscheiden, auch wenn sie im Extremfall kein einziges Dreieck enthält. Jedoch entspricht sie in jedem Fall einem zusammenhängenden Graphen, da sie mindestens einen Spannbaum enthält (siehe Abschnitt 3.2).

## 2.1 Support hull

Sei die Hülle gegen den Uhrzeigersinn orientiert, dann gelten für die Kanten der *support hull* folgenden Eigenschaften:

- Der nach rechts (bzw. außen) offene unendliche Shape durch die Kanten-Endpunkte ist im Inneren punktefrei (siehe Abb. 10).
- Falls der unendliche Shape einer Halbebene entspricht, dürfen keine Punkte direkt auf der Kante zwischen den Endpunkten liegen, andernfalls dürfen auf den Halbstrahlen rechts von der Kante keine Punkte liegen

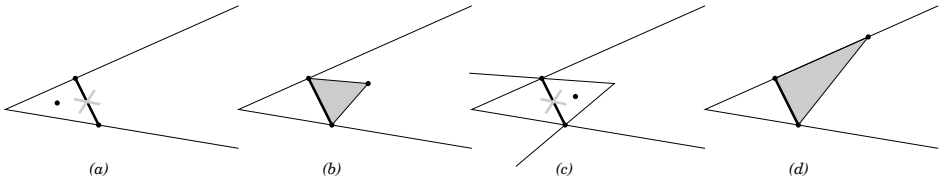


Abbildung 11: ($a$) Liegt ein Punkt im Inneren links der Kante, so ist die Kante ungültig. ($b$) Liegt ein Punkt im Inneren rechts der Kante, so gibt das entweder ein gültiges Dreieck (d.h. die Kante ist nicht Teil der Hülle), oder ($c$) der Punkt liegt im verbotenen Bereich des linken unendl. Shapes (d.h. die Kante ist ungültig). ($d$) Liegt ein Punkt am Halbstrahl rechts der Kante, so gibt dies wiederum ein gültiges Dreieck (keine Hüllenkante).

Ein durch zumindest einen Punkt gehender unendlicher Shape kann nicht von einer gültigen Kante überquert werden, da der Punkt dann im verbotenen Bereich dieser Kante liegen würde.

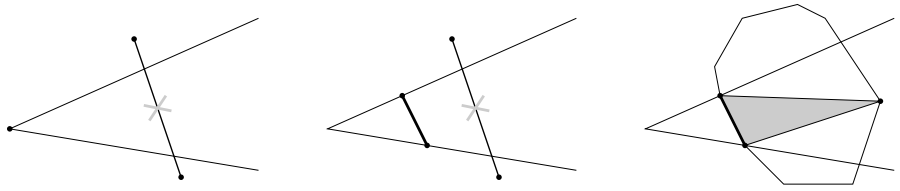

Abbildung 12: Eine direkte Querung ist nicht möglich, ein innerer Punkt ergibt ein gültiges Dreieck.

Daraus folgt, dass die $DT_C(S)$ innerhalb der support hull vollständig trianguliert ist, d.h. es gibt kein Polygon aus gültigen Kanten größer als ein

Dreieck, welches keine gültigen Diagonalkanten besitzt und somit nicht austrianguliert werden könnte.
Denn angenommen, es gäbe ein solches Polygon, dann gibt es eine Kante inklusive zweier Nachbarkanten, mit welchen kein Dreieck gebildet werden kann, d.h. diese liegen außerhalb des unendlichen Shapes durch die Kante, des Weiteren muss sich das Polygon schließen, also noch einmal den unendlichen Shape queren, was nicht möglich ist. Denn die direkte Querung ist ausgeschlossen (siehe Abbildung 12), und gäbe es einen Punkt im Inneren, dann ist entweder die ursprüngliche Kante ungültig, oder sie bildet ein gültiges Dreieck → Widerspruch zur Annahme.

Grundsätzlich kann man die Umgebung einer Kante in 3 Bereiche unterteilen:

I. Der ungültige Bereich, der dem Inneren des Durchschnitts der zwei unendlichen Shapes entspricht.

II. Der äußere Bereich, der von den unendlichen Shapes nicht abgedeckt wird.

III. Der reguläre Bereich, die Vereinigung der unendlichen Shapes ohne I.

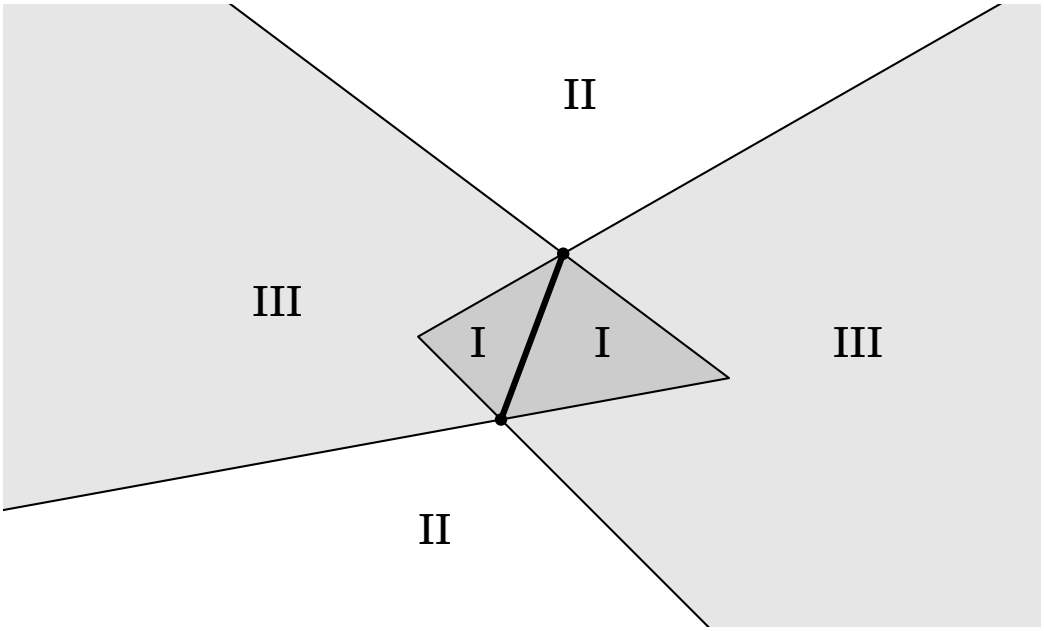


Abbildung 13: Kantenbereiche

Liegt ein Punkt im Bereich I, so ist die Kante ungültig, ein Punkt im Bereich II kann mit der Kante kein gültiges Dreieck bilden, während ein Punkt aus Bereich III ein gültiges Dreieck liefert.
Falls für eine Kante alle anderen Punkte aus der Menge S im Bereich II liegen, so ist die Kante ein Brücke, bzw. beidseitig Teil der support hull. Im Extremfall kann dies für alle gültigen Kanten zutreffen, dann ist $DT_C(S)$ ein Baum und besitzt keine (gültigen) Dreiecke.
Ist der Shape glatt, so sind die Bereiche I und II leere Mengen, und III der einzige Bereich.

## 2.2 Zusammenhang zwischen Hülle und Shape

Bei Abweichungen der support hull von der konvexen Hülle stellt sich die Frage, inwieweit man von diesen "Einkerbungen" Rückschlüsse auf die Form des der Triangulierung zugrundeliegenden Shapes ziehen kann. Während man beim Fehlen solcher Abweichungen auf einen glatten Shape schließen kann (bzw. zumindest auf einen mit entsprechend flachen Ecken), gilt allgemein: Je spitzere Einkerbungen der Rand von $DT_C(S)$ aufweist, desto spitzere Ecken hat auch der zugehörige Shape $C$, welcher auch in die Kerben passt (in Form von Homotheten beliebiger Skalierung), ohne Kanten oder Punkte von $DT_C(S)$ zu überdecken.

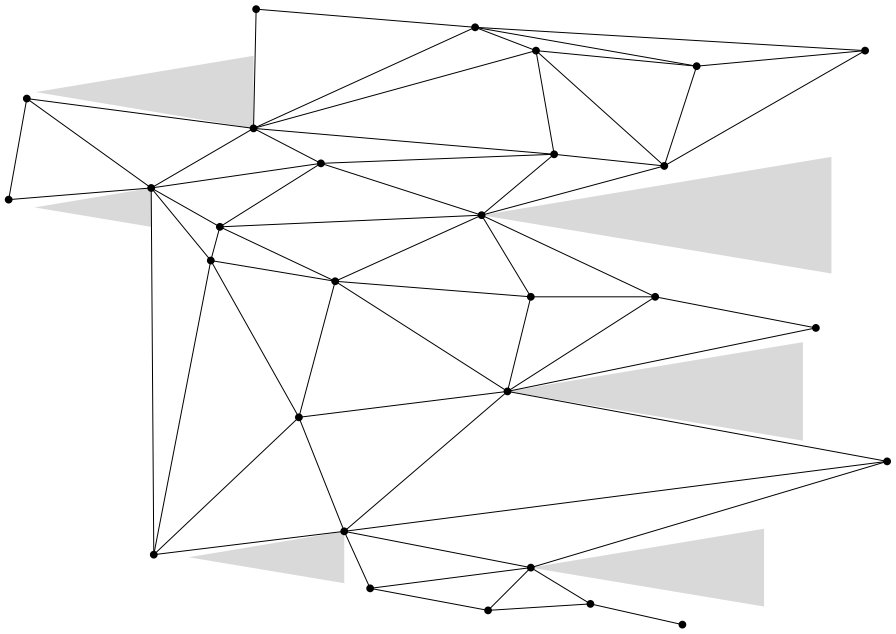

Abbildung 14: $DT_C(S)$ mit "eingepassten" Homotheten des Shapes $C$

Hat $DT_C(S)$ eine Einkerbung mit Winkel $\alpha$, so hat der Shape $C$ eine Ecke mit einem entsprechendem Winkel kleiner als $\alpha$. Wäre dem nicht so, dann wäre die entsprechende Abdeckungskante gültig. Es kann auch nicht sein, dass die Abdeckungskante durch ein Eck aus einer anderen Richtung ungültig wird, denn dann wäre die entsprechende Seitenkante der Einkerbung ebenfalls ungültig.
Man kann also vom Rand der $DT_C(S)$ auf den Rand von $C$ schließen, nicht aber auf die Proportionen von $C$, dafür benötigt man das Innere der Triangulierung, wo man aus den Proportionen der Dreiecke entsprechende Schlüsse ziehen kann. Details dazu sind allerdings nicht mehr Bestandteil dieser Diplomarbeit.

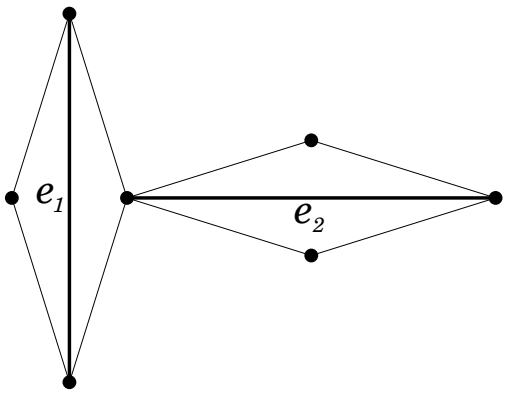


Abbildung 15: Ist $e_1$ eine Delaunay-Kante aus einer gegebenen $DT_C(S)$, dann folgt aus deren Lage, dass der Shape $C$ deutlich höher als breit sein muss, während aus der Lage von $e_2$ folgen würde, dass $C$ breiter als hoch ist. Sind beide Kanten Teil einer Triangulierung, so handelt es sich um keine $DT_C(S)$, da kein konvexer Shape $C$ existiert, der für beide die Delaunay-Eigenschaft erfüllen kann.

# 3 Minimale Abstände

In gewöhnlichen Delaunay-Triangulierungen ist bekanntermaßen immer auch der minimale Spannbaum enthalten, bzw. ist ein nächstes Paar von Punkten immer mit einer Kante in $DT(S)$ verbunden. Bei der Frage, inwieweit diese Eigenschaft auf den Fall allgemeiner konvexer Shapes übertragen werden kann, steht man vor dem Problem, dass konvexe Distanzfunktionen für (bezüglich des Zentrums) nicht symmetrische Shapes keiner Metrik entsprechen, somit ist für zwei Punkte kein eindeutiger Abstand definiert.


Abbildung 16: $d_{C_1}(p,q) = 2.5$, $d_{C_1}(q,p) = 4$, $d_{C_2}(p,q) = 5$, $d_{C_2}(q,p) = 2$ Welcher Abstand zwischen p und q ist der "richtige" bezüglich Shape $C$ (unabhängig vom Zentrum)?

Wie Lihong Ma[2] zeigt, ist $DG_C(S)$ unabhängig vom gewählten Zentrum von $C$, was aus der Sicht der nach Drysdale definierten $DT_C(S)$ wenig überraschend ist, denn für die "Leerer Kreis"-Bedingung zählt bei einem gewählten Shape $C$ mit Zentrum z nur, ob für Punkte $p \in S$ gilt: $d_C(z,p) < 1$ (p innerhalb von $C$), $d_C(z,p) = 1$ ($p$ am Rand von $C$) oder $d_C(z,p) > 1$ ($p$ außerhalb), was offensichtlich unabhängig vom gewählten Zentrum ist.
Folglich ist es naheliegend, eine konvexe Distanzfunktion zu suchen, die für allgemeine konvexe Shapes zentrumsunabhängig eine Metrik liefert, welche analog zu gewöhnlichen Delaunay Triangulierungen die erwähnten Eigenschaften erfüllt. Da es um minimale Abstände geht, ist ein erfolgversprechender Ansatz, das Minimum des Abstands über alle möglichen Zentren eines Shapes zu ermitteln.

**Definition:** Sei $C$ ein konvexer Shape mit Zentrum im Nullpunkt, so sei $C_v$ mit $v \in C$ der gleiche Shape, nur mit einer Translation des Zentrums um $v$ ($C_0 = C$).
**Definition:** Minimale Distanz zweier Punkte über alle möglichen Zentren von Shape C

$$m_C(p,q) := \min_{v \in C} d_{C_v}(p,q)$$

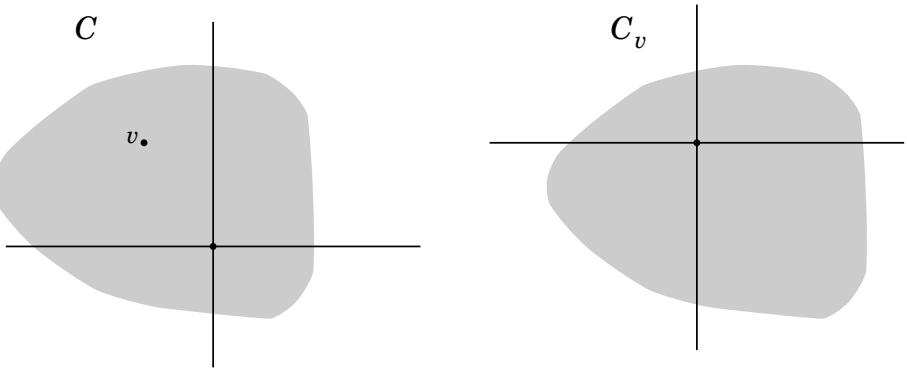


Abbildung 17: $C \to C_v$

Für diese Definition sind auch Zentren am Rand von C zugelassen, tatsächlich wird das Minimum ausschließlich für "Randzentren" auftreten.

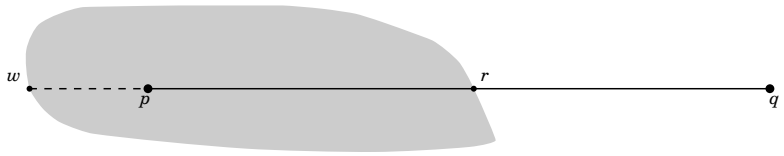


Abbildung 18: Angenommen, das Zentrum sei im Inneren von C, dann kann $d_C(p,q) = \frac{\|p-q\|}{\|p-r\|}$ nicht minimal sein, weil $\|w-r\| > \|p-r\|$ und damit auch $d_{C_w}(p,q) = \frac{\|p-q\|}{\|w-r\|} < d_C(p,q)$.

Der durch Duldung von Randzentren theoretische Fall von Division durch Null kann vernachlässigt werden, da für das gesuchte Minimum nur gegenüberliegende Randpunkte relevant sind. Dies führt zu einer alternativen Definition:

$$m_C(p,q) := \frac{\|p-q\|}{\delta_C(p,q)}$$

(mit $\delta_C(p,q) :=$ maximaler Durchmesser von $C$ parallel zu $\overline{pq}$)

Naturgemäß gilt die Symmetrie $m_C(p,q) = m_C(q,p)$, und $m_C$ ist unabhängig vom ursprünglich gewählten Zentrum von $C$, man geht also ab vom Konzept des Zentrums mit (im Allgemeinen nicht symmetrischen) "Radius", hin zum Konzept des (symmetrischen und zentrumsunabhängigen) Durchmessers.

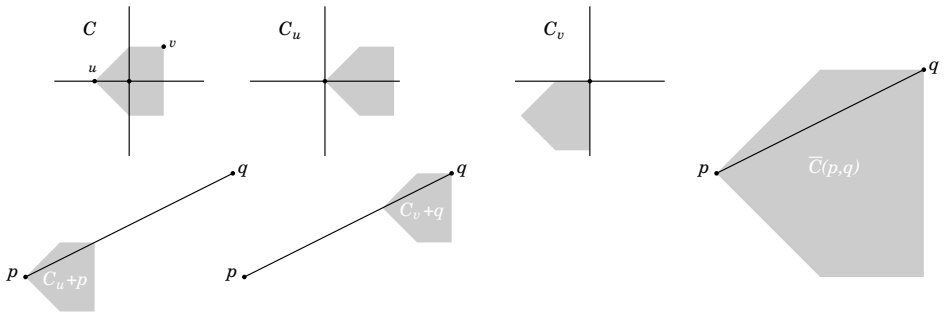


Abbildung 19: $m_C(p,q) = d_{C_u}(p,q) = d_{C_v}(q,p) = m_C(q,p) = \frac{\|p-q\|}{\|u-v\|}$

$m_C(p,q)$ entspricht weiters dem Skalierungsfaktor des minimalen Shapes $\overline{C}(p,q)$, welcher die Punkte $p$ und $q$ am Rand enthält.
Da allerdings die Definition von $m_C(p,q)$ mit "Randzentren" bzw. Zentrumsunabhängigkeit diverse Abweichungen einführt, stellt sich die Frage, inwieweit dies noch mit der ursprünglichen Definition von konvexen Distanzfunktionen vereinbar ist. Dies führt zu einer weiteren Alternativdefinition:

$$m_C(p,q) := d_{\widehat{C}}(p,q) \qquad (mit \quad \widehat{C} := \bigcup_{v \in C} C_v = C \ominus C = C \oplus C_R)$$

$C \ominus C$ sei hier die Minkowski-Differenz des Shapes $C$ mit sich selbst, das entspricht der Menge $\{a - b \mid a \in C, b \in C\}$, bzw. der Minkowski-Summe des Shapes mit dem reflektieren Shape $C_R$.

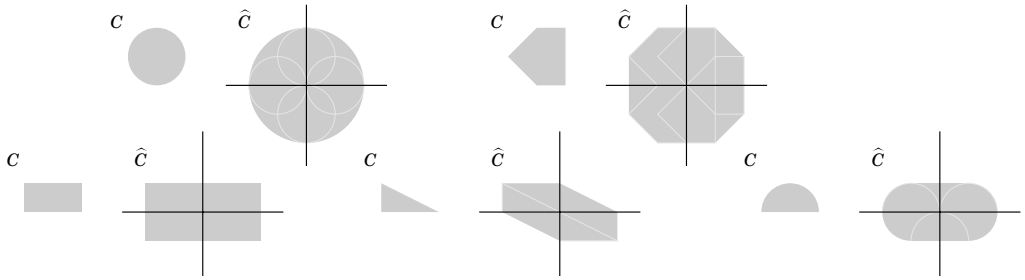


Abbildung 20: Beispiele für den symmetrischen Vereinigungs-Shape $\widehat{C}$

Für bereits (bezügl. dem Ursprung) symmetrische Shapes gilt einfach $\widehat{C} = 2 \cdot C$, so wie ein Durchmesser zweimal dem Radius entspricht. Nicht symmetrische Shapes werden zu symmetrischen erweitert $\Longrightarrow$ $d_{\widehat{C}}(p,q)$ ist eine Metrik.

Sei $\overline{C}(p,q)$ der minimale Homothet des Shapes $C$ durch die Punkte $p$ und $q$. Dieser entspricht dem Pendant zum minimalen Kreis durch $p$ und $q$ bei gewöhnlichen Delaunay-Triangulierungen. Bei Shapes mit parallelen Kanten ist $\overline{C}(p,q)$ wegen möglicher Parallelverschiebung oft nicht eindeutig.

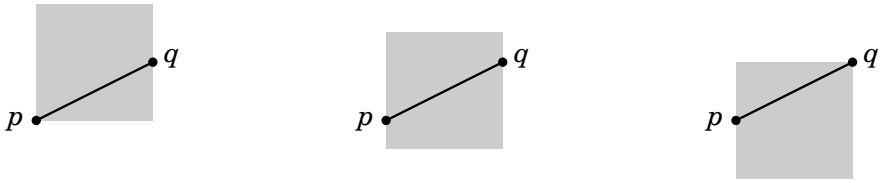


Abbildung 21: $\overline{C}(p,q)$ uneindeutig

Sei ein Punkt $r$ im Inneren von $\overline{C}(p,q)$, dann gilt $m_C(p,r) < m_C(p,q)$ und $m_C(q,r) < m_C(p,q)$, weil $\overline{C}(p,q)$ innerhalb der entsprechend skalierten Shapes $\widehat{C}$ um $p$ bzw. um $q$ liegt (inklusive Parallelverschiebung).

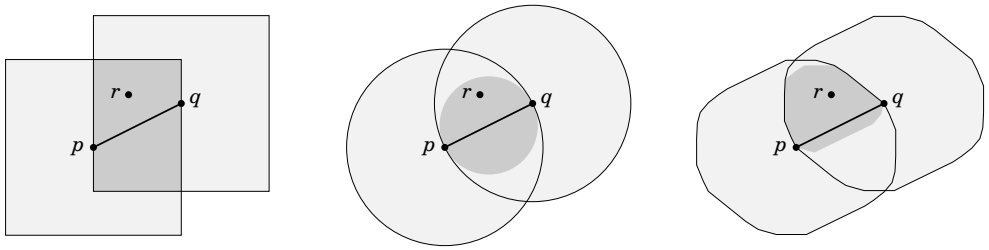


Abbildung 22: Für $r \in \overline{C}(p,q)$ gilt $d_{\widehat{C}}(p,r) < d_{\widehat{C}}(p,q)$ bzw. $d_{\widehat{C}}(q,r) < d_{\widehat{C}}(p,q)$

Damit sind bezüglich der Metrik $m_C$ alle Voraussetzungen erfüllt, um die bekannten Delaunay-Eigenschaften bezüglich *Nächstem Paar* bzw. *Minimalem Spannbaum* zu gewährleisten.

Im Folgenden wird von einer Punktmenge in allgemeiner Lage ausgegangen, da $DT_C(S)$ im Falle von Kozirkularität auf Grund frei wählbarer Kanten nicht eindeutig wäre.

## 3.1 Nächstes Paar

Sei C ein konvexer Shape und sei $(p, q)$ aus der Punktmenge S das nächste Paar bezüglich der Metrik $m_C$, dann gilt: $\overline{pq} \in DT_C(S)$, da der minimale Shape $\overline{C}(p, q)$ punkteleer ist. Andernfalls wäre $(p, q)$ kein nächstes Paar.

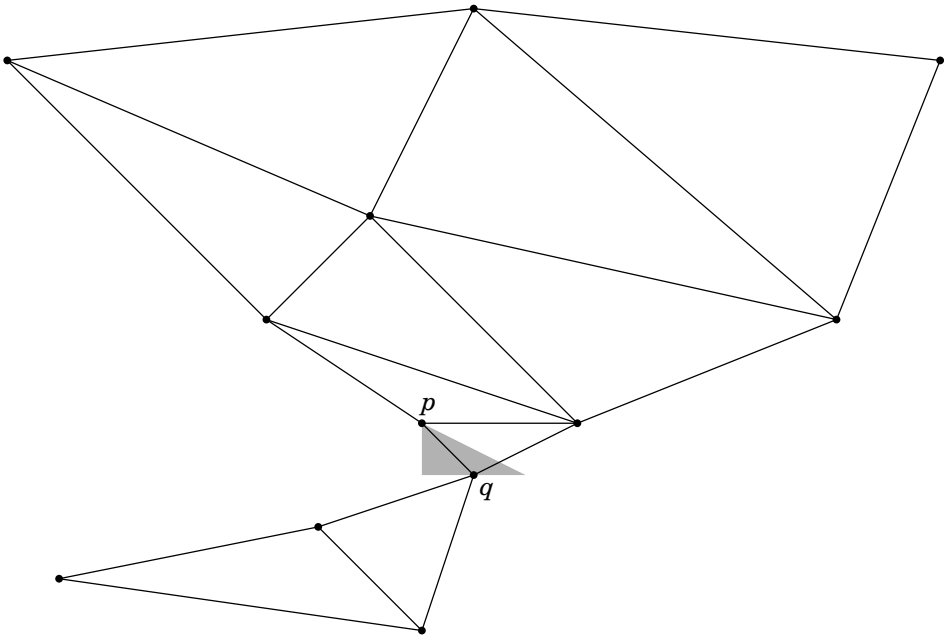


Abbildung 23: Beispiel einer $DT_C(S)$ mit keilförmigem Shape C, nächstem Paar $(p, q)$ und $\overline{C}(p, q)$

## 3.2 Minimaler Spannbaum

Analog zu gewöhnlichen Delaunay-Triangulierungen enthält $DT_C(S)$ bezüglich der Metrik $m_C$ den minimalen Spannbaum $MST_C(S)$.

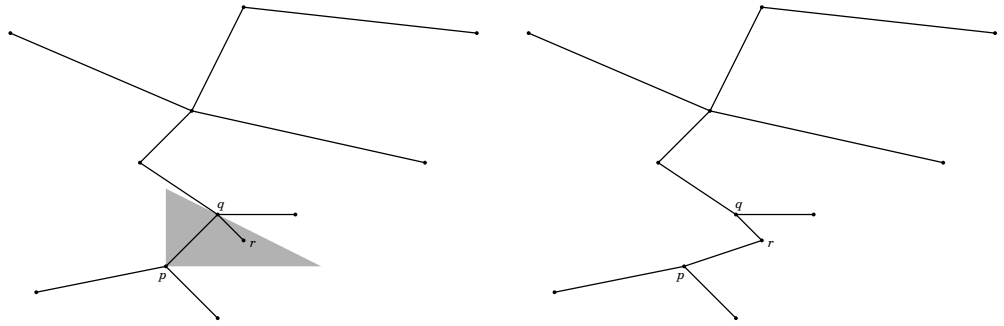


Abbildung 24: Links ungültiger, rechts korrekter $MST_C(S)$

Angenommen, $MST_C(S)$ enthält eine Nicht-Delaunay-Kante $\overline{pq}$, d.h. es gibt für $p$ und $q$ keinen punkteleeren Shape, insbesondere ist dann der minimale Shape $\overline{C}(p,q)$ nicht punkteleer, es sei also ein Punkt $r$ enthalten.
Damit ist sowohl $\overline{pr}$ als auch $\overline{qr}$ kürzer als $\overline{pq}$ (bezüglich $m_C$), und ein entsprechender Austausch führt zu einem kürzeren Spannbaum $\Rightarrow$ Widerspruch zur Annahme, dass der Spannbaum minimal war.

Für bezüglich des Ursprungs symmetrische Shapes $C$ ist $d_C$ eine Metrik, und es gilt $m_C(p,q) = \frac{1}{2} \cdot d_C(p,q)$, demzufolge liefert die Metrik $m_C$ natürlich den selben minimalen Spannbaum wie $d_C$.
Verallgemeinert gilt für symmetrische Vereinigungs-Shapes $\widehat{C}$: $m_{\widehat{C}}(p,q) = \frac{1}{2} \cdot m_C(p,q)$, das bedeutet $MST_{\widehat{C}}(S) = MST_C(S)$, $DT_{\widehat{C}}(S)$ beinhaltet also den selben $MST$ wie $DT_C(S)$.

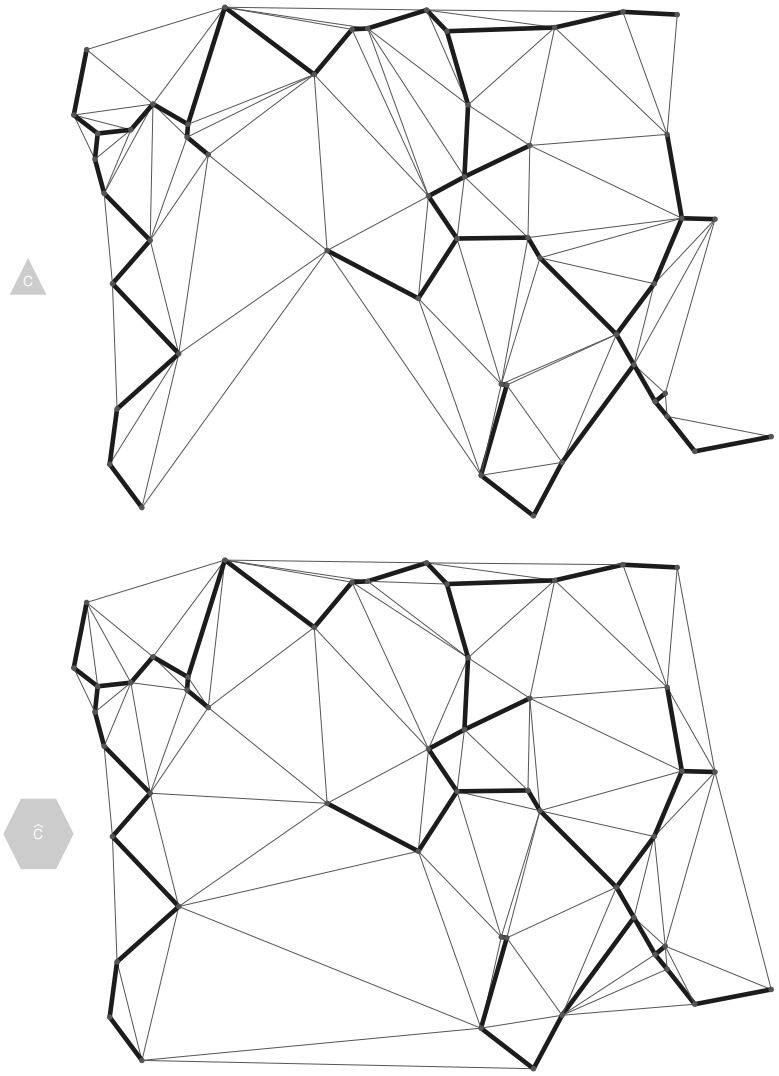


Abbildung 25: So unterschiedlich die Triangulierung für die Shapes $C$ und $\widehat{C}$ im Allgemeinen sein mag, der minimale Spannbaum bleibt der gleiche

## 3.3 Gabrielgraph

Der Gabrielgraph für eine Punktmenge $S$ ist (bei euklidischer Metrik) so definiert, dass jeweils ein Punktepaar $(p, q)$ genau dann mit einer Kante verbunden ist, wenn der Kreis mit Durchmesser $\overline{pq}$ (= minimaler Kreis, welcher $p$ und $q$ am Rand enthält) im Inneren punkteleer ist.
Natürlich ist dieser ein Subgraph der Delaunay-Triangulierung, da die Leerer-Kreis-Bedingung per Definition erfüllt ist, des Weiteren enthält er den minimalen Spannbaum als Subgraphen.

Wieder können wir für konvexe Shapes ein Pendant $GG_C(S)$ (bei Metrik $m_C$) definieren, wo 2 Punkte $p$ und $q$ dann mit einer Kante verbunden sind, wenn alle minimalen Shapes $\overline{C}(p, q)$ leer sind.
Es gilt $MST_C(S) \subseteq GG_C(S) \subseteq DT_C(S)$, denn offensichtlich erfüllen die Gabriel-Kanten die Delaunay-Bedingung und der minimale Spannbaum die Gabriel-Bedingung (siehe voriger Abschnitt).

Für den Fall, dass der konvexe Shape $C$ ein Dreieck ist, gilt $GG_C(S) = DT_C(S)$, weil das Innere von $\overline{C}(p, q)$ mit dem ungültigen Bereich der Kante $\overline{pq}$ zusammenfällt, d.h. wäre ein weiterer Punkt im Inneren, könnte die Kante nicht gültig sein, und damit sind alle Delaunay-Kanten automatisch Gabriel-Kanten.

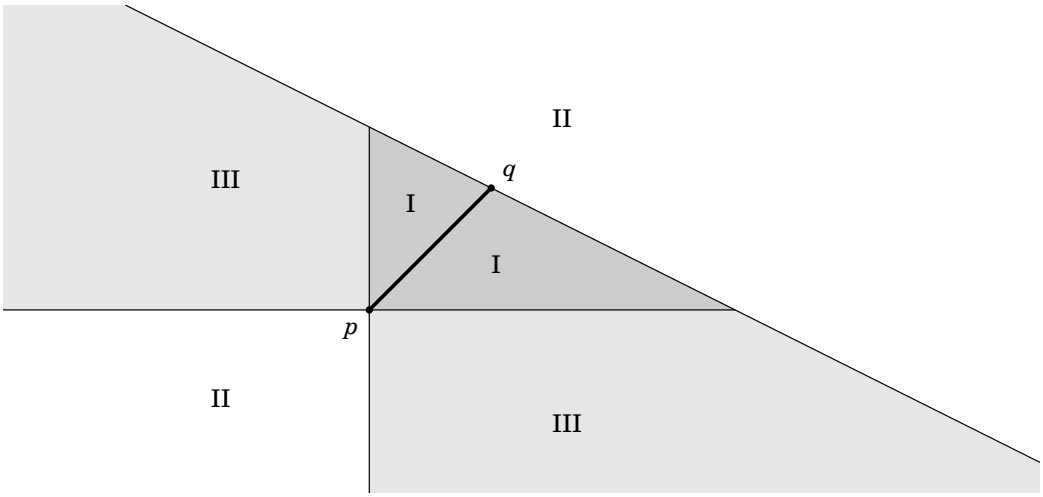


Abbildung 26: Der Bereich I ist gleich dem Inneren von $\overline{C}(p, q)$

# 4 Incircle-Test

Der Incircle-Test ist die essentielle Funktion in jedem Delaunay-Algorithmus, welche darüber entscheidet, ob eine lokale Delaunay-Eigenschaft erfüllt ist bzw. ob ein Kantenflip nötig ist, deswegen ist Effizient und Exaktheit gefragt (Algorithmen siehe J.R. Shewchuk[9] bzw. O. Devillers/S. Pion[10]).
Es seien 4 Punkte gegeben $(a, b, c, d)$, die gegen den Uhrzeigersinn orientiert sind, und es soll ermittelt werden, ob der Punkt $d$ innerhalb des Umkreises von Dreieck $abc$ liegt. Zu dieser Berechnung findet man in der Literatur für gewöhnlich folgende Matrix (*Orientation matrix* der auf ein Paraboloid projizierten Punkte, siehe z.B. *D-Z Du/F.K.Hwang*[3]), deren Determinante

$$\begin{pmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{pmatrix}$$

die gewünschte Auskunft liefert, und zwar ist sie größer 0, wenn der Punkt innerhalb liegt, gleich 0, wenn der Punkt genau auf dem Kreis liegt, und kleiner 0, wenn er außerhalb liegt.
Die Determinante lässt sich durch Umformungen vereinfachen, und liefert dann folgendes Ergebnis, welches 15 Multiplikationen (und 14 Additionen/Subtraktionen) benötigt.

$$\begin{vmatrix} a_x - d_x & a_y - d_y & (a_x - d_x)^2 + (a_y - d_y)^2 \\ b_x - d_x & b_y - d_y & (b_x - d_x)^2 + (b_y - d_y)^2 \\ c_x - d_x & c_y - d_y & (c_x - d_x)^2 + (c_y - d_y)^2 \end{vmatrix} =$$

$$= (a-d)^2 \cdot (b-d) \times (c-d) - (b-d)^2 \cdot (a-d) \times (c-d) + (c-d)^2 \cdot (a-d) \times (b-d)$$

Scheinbar weniger bekannt ist, dass hier noch eine weitere Vereinfachung möglich ist,

$$\begin{vmatrix} a_x - c_x & a_y - c_y & (a_x - c_x)(a_x - d_x) + (a_y - c_y)(a_y - d_y) \\ b_x - c_x & b_y - c_y & (b_x - c_x)(b_x - d_x) + (b_y - c_y)(b_y - d_y) \\ c_x - d_x & c_y - d_y & 0 \end{vmatrix} =$$

$$= (a - c)(a - d) \cdot (b - c) \times (b - d) - (b - c)(b - d) \cdot (a - c) \times (a - d)$$

welche mit 10 Multiplikationen (und 13 Additionen/Subtraktionen) auskommt, und z.B. bei der CGAL (Computational Geometry Algorithms Library) verwendet wird.
Anmerkung: $ab := a_x b_x + a_y b_y, \quad a \times b := a_x b_y - a_y b_x$

Dies ist allerdings noch nicht der Weisheit letzter Schluss, da durch Fallunterscheidung noch Reduktionen möglich sind und auch die geometrische Bedeutung der Terme klarer wird.
Wir gehen wieder von den obigen 4 Punkten aus und betrachten $b$ und $d$ in Relation zur Kante $ac$ und deren Durchmesserkreis.

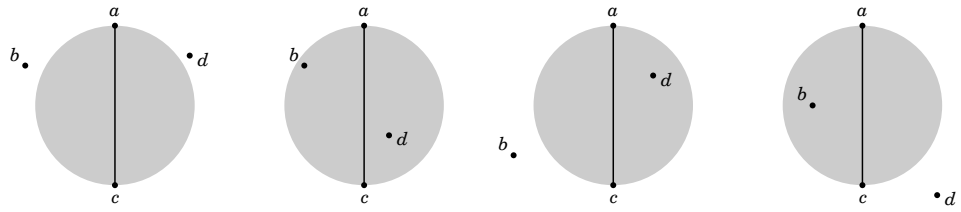


Abbildung 27: 4 Fälle bezüglich des Durchmesserkreises.

Wenn beide Punkte außerhalb liegen, haben wir eine Gabriel-Kante, und $d$ liegt offensichtlich auch nicht im Umkreis von $abc$. Sind hingegen beide Punkte im Inneren, so liegt $d$ definitiv auch im Umkreis von $abc$, nur die Fälle mit einem Punkt innen und einem außen bleiben noch unklar.
Der Test für den Durchmesserkreis ist simpel, dafür braucht man nur die Skalarprodukte $(a-b)(c-b)$ und $(a-d)(c-d)$. Sind beide kleiner 0, so liegen beide Punkte innen, sind beide größer 0, so sind beide Punkte außen, und damit hat man (mit 4 Multiplikationen) bereits die Hälfte der Fälle abgedeckt.
Um zu zeigen, dass es tatsächlich die Hälfte ist, definiert man einfach $P$ als Wahrscheinlichkeit, dass ein getesteter Punkt innerhalb liegt, und erhält dann mit $P^2+(1-P)^2 = 2P^2-2P+1$ die Wahrscheinlichkeit, dass entweder beide Punkte innen oder beide Punkte außen liegen. Diese Funktion nimmt bei 0.5 ihr Minimum an, empirische Tests beim Einfüge-Algorithmus mit ca. 1000 Zufallspunkten ergaben einen Durchschnittswert von 55%.
Theoretisch könnte man zusätzlich auch $a$ und $c$ bezüglich des Durchmesserkreises von $bd$ testen, dies erhöht die durchschnittlich abgedeckten Fälle allerdings nur auf ca. 66% und zahlt sich somit kaum aus.
Was die restlichen Fälle angeht, so kann man die Umkreise der beiden Dreiecke $abc$ und $acd$ als Konkurrenten bezüglich der Lage ihrer Mittelpunkte auf dem Bisektor von $a$ und $b$ betrachten.
Der Ansatz zur Berechnung startet mit dem Schnitt der Bisektoren, um die Mittelpunkte zu ermitteln, welche allerdings nicht explizit berechnet werden, sondern nur deren relative Lage zueinander.

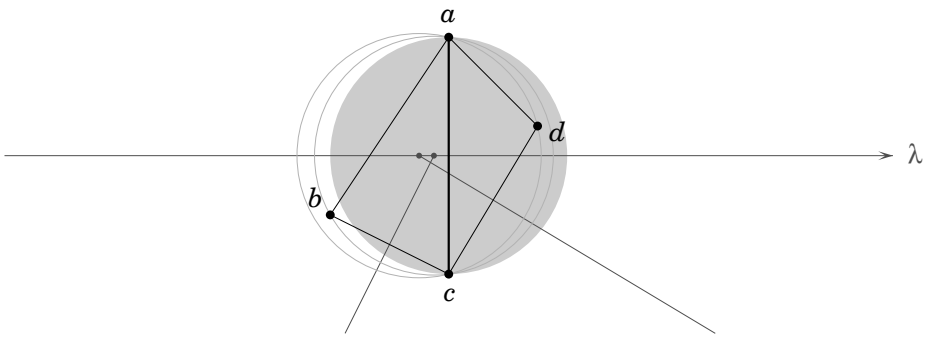


Abbildung 28: Liegt der Umkreismittelpunkt von *acd* weiter links als der von *abc* (wie hier z.B. der Fall), dann liegt $d$ im Umkreis von *abc* (und umgekehrt).

$$\frac{a+c}{2} + \lambda_1 \frac{(a-c)^\vdash}{2} = \frac{d+c}{2} - \mu_1 \frac{(d-c)^\vdash}{2} \qquad | \circ (d-c)$$

$$\lambda_1 (a-c)^\vdash (d-c) = (d-a)(d-c)$$

$$\lambda_1 = \frac{(a-d)(c-d)}{(a-d) \times (c-d)}$$

$$\frac{a+c}{2} + \lambda_2 \frac{(a-c)^\vdash}{2} = \frac{b+c}{2} + \mu_2 \frac{(b-c)^\vdash}{2} \qquad | \circ (b-c)$$

$$\lambda_2 (a-c)^\vdash (b-c) = (b-a)(b-c)$$

$$\lambda_2 = \frac{(a-b)(c-b)}{(a-b) \times (c-b)}$$

$$\lambda_1 < \lambda_2 \Leftrightarrow (a-d)(c-d) \cdot (a-b) \times (c-b) - (a-b)(c-b) \cdot (a-d) \times (c-d) > 0$$

Anmerkung: $a^\vdash = \begin{pmatrix} a_x \\ a_y \end{pmatrix}^\vdash := \begin{pmatrix} a_y \\ -a_x \end{pmatrix}$(Normalvektor im Uhrzeigersinn)

Nicht ganz zufällig entspricht der durch den Vergleich gewonnene Ausdruck dem Ergebnis der *orientation matrix* (nur anders umgeformt). Der Unterschied ist, dass hier die geometrische Bedeutung der einzelnen Bestandteile klar und transparent ist (damit ist die Berechnung auch robuster gegenüber etwaiger Fehlerfortpflanzung). Zum einen sind beide Skalarprodukte enthalten, die für den Durchmesserkreis-Test nötig sind, somit ist ein Vorabtest kein zusätzlicher Aufwand (und die nötigen Multiplikationen reduzieren sich auf durchschnittlich knappe 7), zum anderen sind mit den Kreuzprodukten auch die Orientierungen der Dreiecke *abc* und *acd* enthalten, d.h. auch wenn die Orientierung der 4 Punkte nicht gegeben ist, sind keine zusätzlichen Berechnungen nötig (nur weitere Fallunterscheidungen), und man braucht in keinem Fall mehr als 10 Multiplikationen.
Nebenbei gilt naturgemäß $\lambda_1 = cot\measuredangle(a-d, c-d)$ und $\lambda_2 = cot\measuredangle(a-b, c-b)$, sowie $|\lambda_1| < |\lambda_2| \iff r_{acd} < r_{abc}$.

Der Vollständigkeit halber wäre noch anzumerken, dass man nach dem Vorbild des Gauß'schen Algorithmus für komplexe Zahlen, welcher 4 Multiplikationen auf 3 reduziert (auf Kosten von 3 zusätzlichen Additionen/Subtraktionen), auch für Skalarprodukt und Kreuzprodukt vorgehen kann.

$$\begin{aligned} s &= a_x b_x \\ t &= a_y b_y \\ ab &= s + t \\ a \times b &= (a_x - a_y)(b_x + b_y) - s + t \end{aligned}$$

Damit würde man für den Incircle-Test insgesamt nur 4-8 Multiplikationen (knappe 6 im Durchschnitt) benötigen, dies würde sich allerdings nur dann auszahlen, wenn Multiplikationen auch wirklich entsprechend langsamer als Additionen wären, was bei aktuellen Prozessoren immer weniger der Fall ist (außer für sehr große Zahlen), außerdem sinkt hierbei wieder die Robustheit der Berechnung.

Wenn wir nun wieder zu allgemeinen konvexen Shapes übergehen, bleibt von diesen Ergebnissen nicht viel übrig, man kann aber auch hier als Vorabtest für eine Kante den minimalen Shape $\overline{C}$ berechnen, und damit bereits wieder eine gute Hälfte der Fälle abdecken.
Hat man einen Shape in Form eines konvexen Polygons $P_n$ mit $n$ Ecken gegeben, so kann man bei bekanntem symmetrischen Vereinigungs-Shape $\widehat{P_n}$ für jede Kante $e$ den minimalen Shape $\overline{P_n}(e)$ in O(log n) Zeit ermitteln.

Sind nun beide verbleibende Punkte innerhalb bzw. beide außerhalb von $\overline{P_n}(e)$, so sind wir bereits wieder fertig, andernfalls kann man per "sweep line" entlang der Ecken den Polygons weitersuchen.

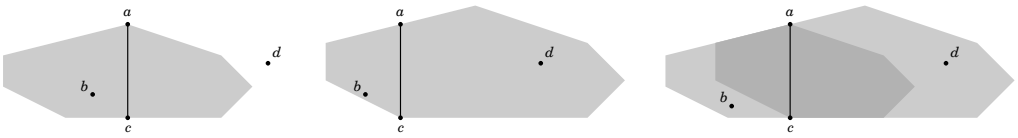


Abbildung 29: $\overline{P_n}(a, c)$ mit einem Punkt innerhalb und einem außerhalb. Findet man nun per Ecken-Test einen Shape mit beiden Punkten innerhalb (bzw. beiden außerhalb), ist der Incircle-Test wieder abgeschlossen, nur falls für 2 benachbarte Shapes gilt, dass jeweils ein Punkt in einem aber nicht im anderen liegt, ist eine genauere Vergleichsberechnung nötig.

Grundsätzlich gilt, dass wenn sowohl die Punktmenge als auch das Polygon ganzzahlig gegeben ist, der Incircle-Test auch exakt ganzzahlig entscheidbar ist, die exakte Berechnung eines Umkreis-Shapes durch 3 Punkte ist dafür nicht nötig.
Laut L.Ma[2] ist der Schnitt zweier Bisektoren algorithmisch in $O(log\, n)$ Zeit möglich, folglich gilt dies auch für den Incircle-Test.
Natürlich wäre hier ebenfalls wieder der analoge Ansatz wie in Abbildung 28 möglich (mit einem Vergleich zweier Schnittpunkte, ohne die Schnittpunkte explizit zu berechnen), inwieweit das in diesem Fall die optimale Vorgangsweise ist, sei allerdings dahingestellt.

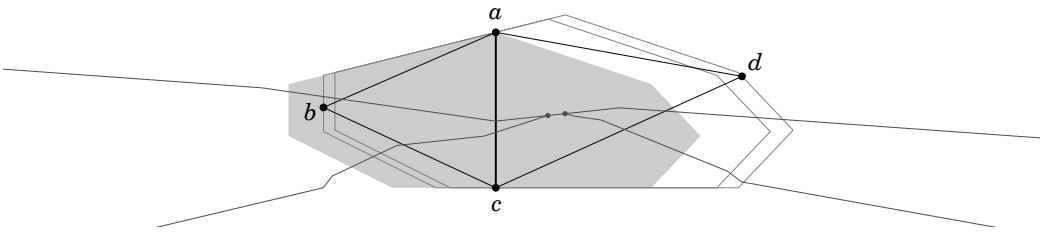


Abbildung 30: Ein Vergleich der Bisektoren-Schnittpunkte (bezüglich eines beliebig angenommenen Zentrums des Shapes) entspricht dem Incircle-Test.

# 5 Flips

Als Flip wird die elementare Transformation innerhalb von Triangulierungen bezeichnet, die eine Diagonale eines konvexen Vierecks durch die andere Diagonale ersetzt (vergl. S. Fortune[7] bzw. T. Lambert[8]).

Wenn wir eine Kante von $DT_C(S)$ auf Flipbarkeit untersuchen, erhalten wir mehrere mögliche Fälle, abhängig von der Lage der Punkte bezüglich des "Einzugsgebietes" der Kante (siehe Abbildung 13).
Zur Veranschaulichung wird im Folgenden ein quadratischer Shape angenommen.

(1) Beide Punkte befinden sich in Zone I → FLIP

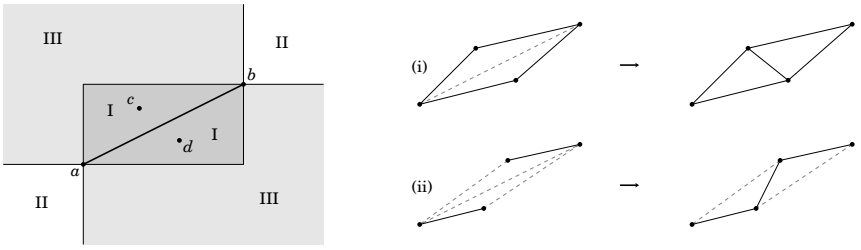


Abbildung 31: Im Fall (1) ist die Kante *ab* ungültig, so wie die beiden Dreiecke *abc* und *abd*. Ein Flip bringt jedenfalls eine Verbesserung, da die Kante *cd* hier zwangsweise gültig ist. Die neuen Dreiecke *acd* und *bcd* werden je nach Lage entweder gültig (i) oder bleiben ungültig (ii).

(2) Ein Punkt in Zone I, der andere in II → kein FLIP

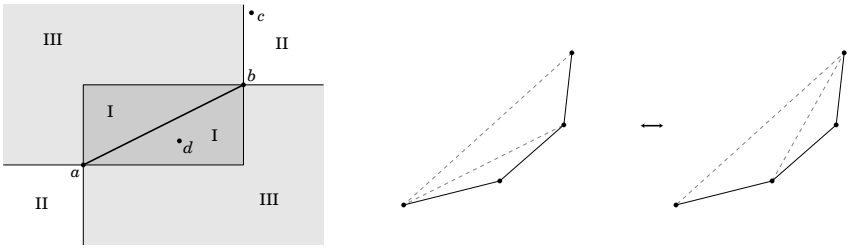


Abbildung 32: Im Fall (2) ist ebenfalls die Kante *ab* ungültig, so wie beide Dreiecke, allerdings gilt das Gleiche für die Kante *cd* und man kann sich den Flip sparen (andernfalls besteht die Gefahr einer Endlosschleife).

(3) Ein Punkt in Zone I, der andere in III → FLIP

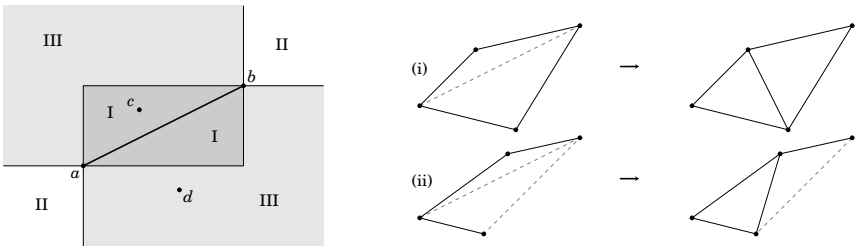


Abbildung 33: Im Fall (3) ist wieder die Kante *ab* ungültig, ebenso das Dreieck *abc*, während *abd* gültig ist. Es wird geflippt, denn man erhält mit *cd* eine gültige Kante, von den neuen Dreiecken sind entweder beide gültig (i) oder eines (ii).

(4) Ein Punkt in Zone II, der andere in III → kein FLIP

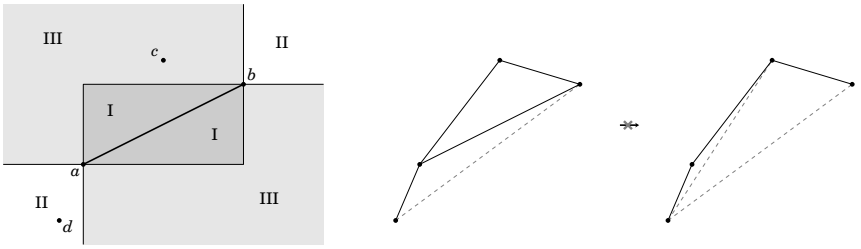


Abbildung 34: Umgekehrter Fall wie bei (3)(ii).

(5) Beide Punkte befinden sich in Zone II → kein FLIP

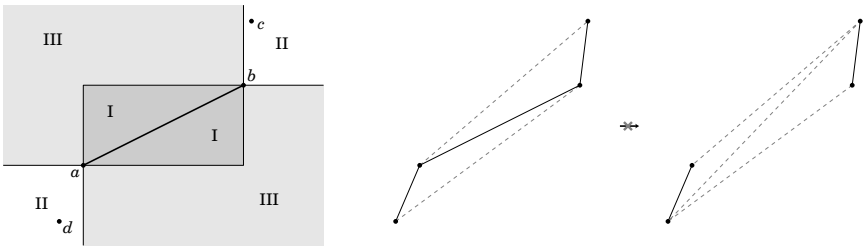


Abbildung 35: Umgekehrter Fall wie bei (1)(ii).

(6) Beide Punkte befinden sich in Zone III → FLIP abhängig vom Incircle-Test

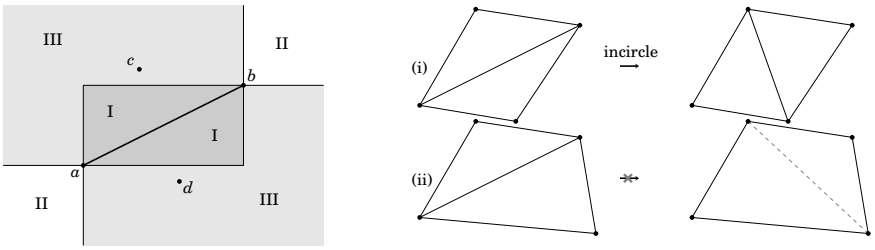


Abbildung 36: Im Fall (6) sind Kante *ab* und beide Dreiecke gültig, hier entscheidet der Incircle-Test über den Flip, bzw. falls die Kante *cd* ungültig ist (ii), kann man sich den Flip direkt sparen.

Zusammenfassend kann man sagen, sofern sich ein Punkt im "Out" befindet (Zone II), kann man direkt auf den Flip verzichten, falls sich ansonsten ein Punkt im ungültigen Bereich befindet (Zone I), wird geflippt, und andernfalls bleibt Fall (6) mit dem Incircle-Test, der bei glatten Shapes den einzigen Fall darstellt.
Eine gleichwertige Folgerung ist, dass ungültige Kanten immer geflippt werden, außer die andere Diagonale ist auch ungültig (hier würde ein Algorithmus, der so lange flippt als es flipbare Kanten gibt, zu einer Endlosschleife führen, dies gilt allerdings nicht für den Einfüge-Algorithmus), während gültige Kanten nicht geflippt werden, außer die Gegenkante ist auch gültig, dann entscheidet der Incircle-Test über die Delaunay-Eigenschaft.

# 6 Lokal Delaunay - Global Delaunay

## 6.1 Maximierende Eigenschaft

Bekanntlich maximieren gewöhnliche Delaunay Triangulierungen den minimalen Kantenwinkel über allen Triangulierungen. Da allgemeine konvexe Shapes allerdings quasi beliebige Proportionen haben können, ist hier keine allgemeingültige Aussage mehr über die Winkel der entsprechenden $DT_C$ möglich. Es gilt also, ein verwandtes Kriterium zu finden, welches die Analogie zwischen $DT$ und $DT_C$ möglichst aufrecht erhält.
Hier bietet sich die Größe der Umkreise an, da es naheliegend ist, dass größere Umkreise eines Dreiecks eher dazu neigen, den 4. Punkt zu beinhalten als kleinere, zumindest wenn sie nicht auf der zum 4. Punkt abgewandten Seite liegen.
Wenn 4 Punkte kein konvexes Viereck bilden, so sind die Dreiecke immer "voneinander abgewandt" und können trotz großer Umkreise nie den 4. Punkt erreichen.

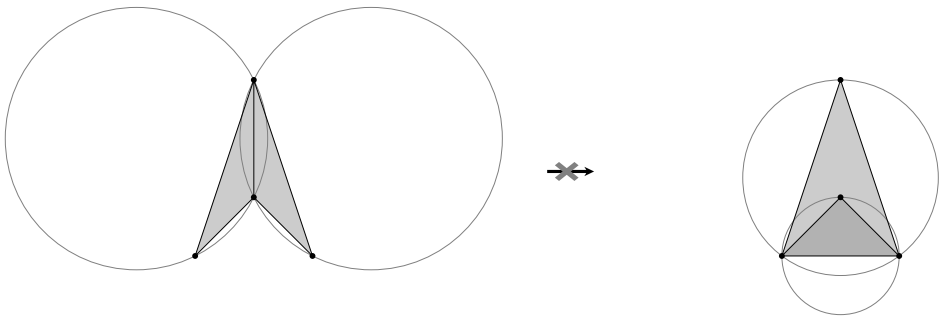

Abbildung 37: Wenn 2 benachbarte Dreiecke zusammen kein konvexes Viereck bilden, ist kein gültiger Flip möglich (auch wenn die Umkreise kleiner wären).

Dies gilt nicht nur für Kreise, sondern für alle konvexen Shapes. Wäre von einem durch 3 Punkte gehenden konvexen Shape der 4. Punkt erreichbar, dann würden die 4 Punkte ein konvexes Viereck bilden. Somit kann man sich bei der Betrachtung der Dreiecksumkreise auf konvexe Vierecke beschränken.

Haben wir also ein konvexes Viereck gegeben, so gibt es 4 mögliche Dreiecke, deren Umkreis wir untersuchen.

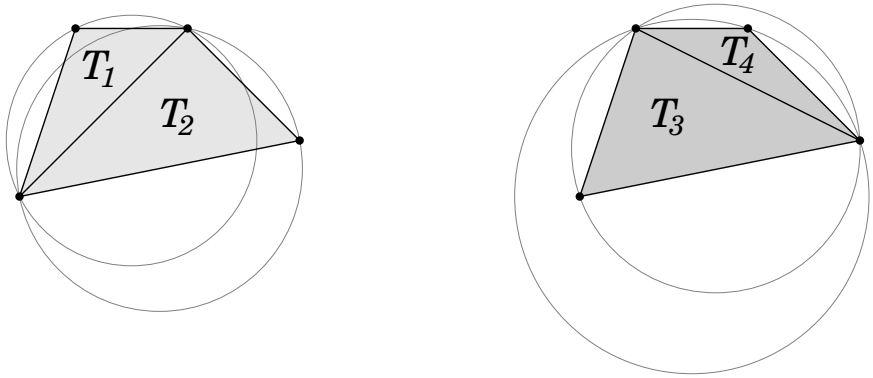


Abbildung 38: Triangulierung eines konvexen Vierecks (links Delaunay).

Sind die 4 Punkte kozirkular, so gilt für die Umkreisradien $r_1 = r_2 = r_3 = r_4$ und alle Dreiecke wären Delaunay. Ist das nicht der Fall und seien $T_1$ und $T_2$ die Delaunay-Dreiecke, so kann man zeigen:

$$max(r_1, r_2) < max(r_3, r_4)\ ,\ min(r_1, r_2) < min(r_3, r_4)$$

Daraus folgt u.a. auch: $r_1 + r_2 < r_3 + r_4$, d.h. die Delaunay-Triangulierung minimiert sowohl den maximalen und minimalen Umkreis als auch die Summe über alle Umkreise der Triangulierung.
Somit hat man geeignete (winkelunabhängige, aber zum Winkelkriterium gleichwertige) Kriterien zur Verfügung, die man darauf untersuchen kann, inwieweit sie für allgemeine konvexe Shapes Bestand haben, wobei sich insbesondere die Minimierung des maximalen Umkreises anbietet.
Zu Umkreiskriterien ist anzumerken, dass man es im Falle von entarteten Dreiecken (die 3 Punkte liegen auf einer Geraden) mit unendlich großen Radien zu tun bekommt, dies kann man vermeiden, indem man entweder entartete Dreiecke ausschließt oder den Kehrwert der Umkreisradien maximiert.

Um die Beziehungen der Umkreisradien zu zeigen, kann man sich am Verhalten eines Kreises auf seinem Weg entlang einer Voronoi-Kante orientieren, während er sich durch eine Seitenkante des konvexen Vierecks "zwängt". Vom Unendlichen kommend wird der Radius naturgemäß immer kleiner und nimmt im Durchmesserkreis der Kante sein Minimum an, bevor er auf der anderen Seite wieder wächst.

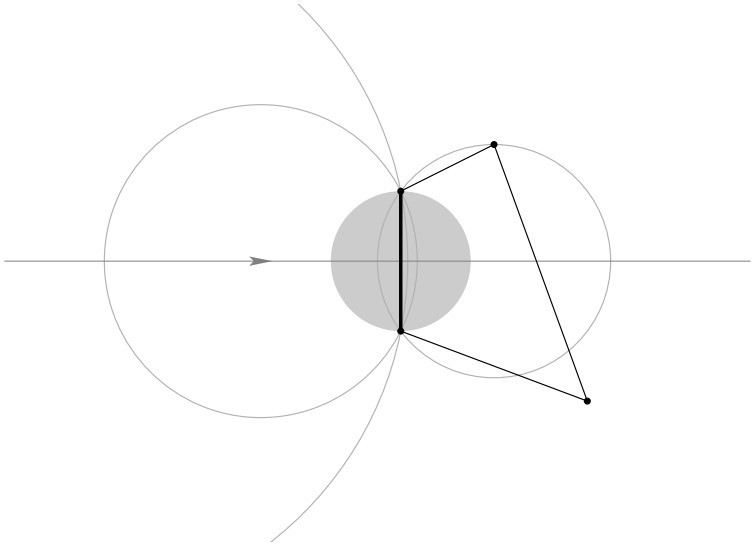

Abbildung 39: Weg des Kreises durch eine Seitenkante.

Logischerweise ist derjenige von den restlichen 2 Punkten des Vierecks, der als erster vom Kreis getroffen wird, auch derjenige, welcher mit der Kante ein Delaunay-Dreieck bildet. Ist nun die Kante bezüglich des Vierecks eine Gabriel-Kante, d.h. keiner der 2 Punkte liegt im Durchmesserkreis, so folgt daraus automatisch, dass das Delaunay-Dreieck den kleineren Umkreisradius hat als das Nichtdelaunay-Dreieck.
Liegt hingegen ein Punkt im Durchmesserkreis, so kann der Radius auf dem Weg vom ersten getroffenen Punkt zum zweiten Punkt noch sinken (in dem Fall hat das Nichtdelaunay-Dreieck den kleineren Radius), deswegen gilt im Allgemeinen: $max(r_1, r_2) \not< min(r_3, r_4)$.

Man kann nun zeigen, dass ein konvexes Viereck immer zwei gegenüberliegende Gabriel-Kanten besitzt, daraus folgend gibt es immer eine Indizierung (mit $r_1, r_2$ Delaunay und $r_3, r_4$ nicht Delaunay), sodass gilt: $r_1 < r_3$ sowie $r_2 < r_4$, und somit $max(r_1, r_2) < max(r_3, r_4)$ usw.
Angenommen, es gäbe ein konvexes Viereck, welches keine 2 gegenüberliegende Gabriel-Kanten besitzt, dann gibt es zumindest 2 Nachbarkanten, die keine Gabriel-Kanten sind, d.h. der 4. Punkt liegt innerhalb beider Durchmesserkreise, was unmöglich ist $\rightarrow$ Widerspruch zur Annahme.
Es stellt sich nun die Frage, ob die Eigenschaft der Gabriel-Kanten auch für allgemeine konvexe Shapes gültig bleibt? Wie sich herausstellt, gilt es zwar für die meisten Fälle, allerdings gibt es Ausnahmen, wo es zwei benachbarte Nicht-Gabriel-Kanten gibt, und das Dreieck mit dem kleinsten

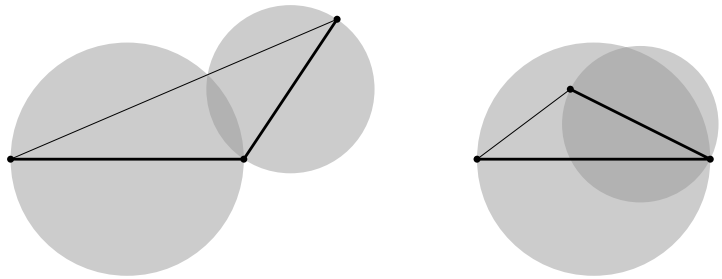

Abbildung 40: Eine Schnittmenge der Kreise gibt es nur innerhalb des Dreiecks (bzw. auf einer abgewandten Seite), wo ein 4. Punkt kein konvexes Viereck bilden kann.

Umkreisradius nicht delaunay ist $\rightarrow min(r_1, r_2) > min(r_3, r_4)$.
Für diese Ausnahmefälle kann man aber zeigen, dass zumindest immer noch gilt: $max(r_1, r_2) < max(r_3, r_4)$ (und allen Tests zufolge scheint für diesen Fall auch das Summenkriterium $r_1 + r_2 < r_3 + r_4$ zu halten, dies dürfte allerdings schwierig zu beweisen sein).

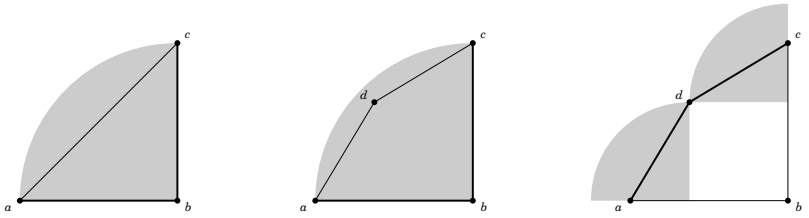


Abbildung 41: Beispiel mit einem Viertelkreis als Shape und einem Dreieck *abc*, für das gilt: $\overline{C}(a, b) = \overline{C}(b, c)$. Hier kann ein weiterer Punkt $d \in \overline{C}$ ein konvexes Viereck bilden, mit 2 benachbarten Nicht-Gabriel-Kanten *ab* und *bc*, sowie den Gabriel-Kanten *ad* und *cd*.

Sei *abcd* ein konvexes Viereck mit 2 benachbarten Nicht-Gabriel-Kanten *ab* und *bc* (bezüglich Shape $C$), so gilt $m_C(a, d) < m_C(a, b)$ und $m_C(c, d) < m_C(c, b)$, sowie beide kleiner als $m_C(a, c)$.
Das heißt, die anderen beiden Kanten *ad* und *cd* sind auf jeden Fall Gabriel-Kanten, und damit wird das Nicht-Delaunay-Dreieck *acd* erst mit größerem Radius (bzw. Skalierungsfaktor) erreicht.
Mit der Minimierung des maximalen Umkreis-Shapes haben wir allerdings für allgemeine konvexe Shapes noch kein fertiges Kriterium, da wir die ungültigen Kanten bzw. Dreiecke berücksichtigen müssen. Ist ein Dreieck ungültig, d.h. der gegebene Shape kann in keiner Skalierung durch die 3

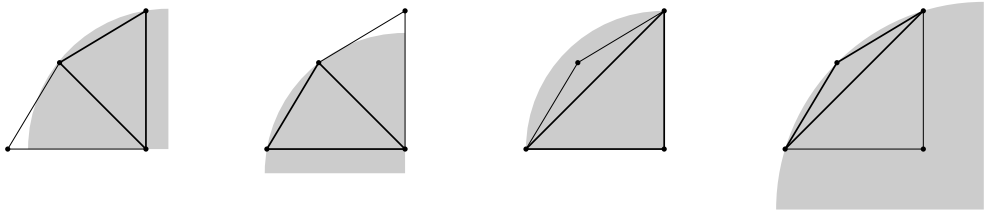

Abbildung 42: Hier hat ein Nicht-Delaunay-Dreieck den kleinsten Umkreis-Shape, d.h. $min(r_1, r_2) > min(r_3, r_4)$ und $max(r_1, r_2) < max(r_3, r_4)$.

Punkte gelegt werden, so müsste man dem "Umkreisradius" (bzw. Skalierungsfaktor) den Wert "unendlich" zuweisen.
Da man ungern mit unendlichen Werten hantiert, sofern es Alternativen gibt, bietet sich die Verwendung des Kehrwerts an ($\varrho := \frac{1}{r}$). Wir minimieren also nicht mehr den maximalen "Umkreis", sondern maximieren den minimalen Kehrwert. Hierbei sind endliche Werte garantiert, da wir natürlich von disjunkten Punktmengen ausgehen.
Ein weiteres Problem ergibt sich dadurch, dass bei allgemeinen konvexen Shapes eine Delaunay-Kante nicht mehr automatisch einem Delaunay-Dreieck angehört. Als Beispiel hätten wir Fall 4 aus dem letzten Abschnitt (Abbildung 34), wo die Delaunay-Kante $ab$ als Nachbarn ein gültiges Dreieck $abc$ und ein ungültiges Dreieck $abd$ hat, genauso wie die Nachbarn der in diesem Fall ungültigen Vergleichskante $cd$ aus einem ungültigen Dreieck $acd$ und einem gültigen Dreieck $bcd$ bestehen. Damit reicht der Vergleich des maximalen "Umkreisradius" bzw. des minimalen Kehrwerts (wegen $r_{abd} = r_{acd} = \infty$ bzw. $\varrho_{abd} = \varrho_{acd} = 0$) nicht mehr aus, und man muss die anderen 2 Werte mit einbeziehen, dies kann man mit der lexikographischen Ordnung realisieren: $\begin{pmatrix} \varrho_{abd} \\ \varrho_{abc} \end{pmatrix} \succ \begin{pmatrix} \varrho_{acd} \\ \varrho_{bcd} \end{pmatrix}$.

Damit sind wir aber auch noch nicht fertig, denn wie man bei Fall 2 (Abbildung 32) und Fall 5 (Abbildung 34) sehen kann, ist es auch möglich, dass alle 4 Dreiecke ungültig sind, damit wäre obiger Vergleich immer noch unzureichend.
Abhilfe schafft die zusätzliche Einbindung der Kante selbst, entweder in Form ihrer Gültigkeit ($\sigma_{ab}$ sei gleich 0, wenn $ab$ ungültig ist, d.h. $c$ oder $d$ befindet sich in Zone I bezüglich der Kante $ab$ und dem Shape C, ansonsten gleich 1) oder ihrer Länge ($\varrho_{ab} := \frac{1}{m_C(a,b)}$), denn im Fall der Ungültigkeit aller Dreiecke ist trivialerweise die gültige Kante (sofern es sie gibt) die kürzere.

Seien $\varrho_1$ und $\varrho_2$ die Werte $\varrho_{abc}$ und $\varrho_{abd}$ derartig zugeordnet, sodass gilt: $\varrho_2 \leq \varrho_1$, und sei $\varrho_0$ wahlweise gleich $\sigma_{ab}$ oder gleich $\varrho_{ab}$, so entspricht die lexikographische Maximierung von $\begin{pmatrix} \varrho_2 \\ \varrho_1 \\ \varrho_0 \end{pmatrix}$ dem Delaunay-Kriterium.

## 6.2 Global Delaunay

Um zu zeigen, dass die Erfüllung der lokalen Delaunay-Eigenschaften auch gleichzeitig global zur Delaunay-Triangulierung führt, kann man sich auch für allgemeine konvexe Shapes wieder am üblichen Beweis für gewöhnliche Delaunay-Triangulierungen orientieren.

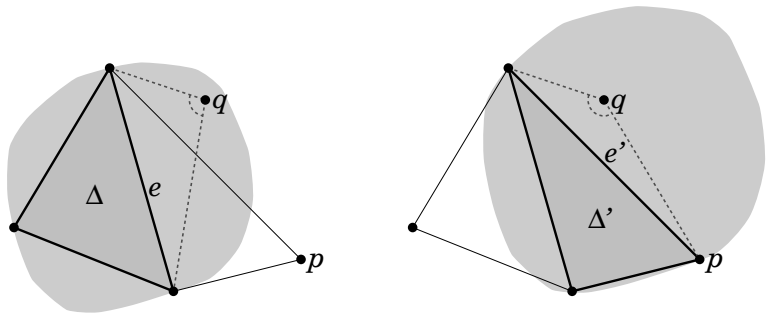


Abbildung 43: $(\Delta, e, q) \rightarrow (\Delta', e', q)$

Angenommen, die Triangulierung wäre lokal Delaunay, aber nicht global Delaunay, dann gibt es mindestens ein Dreieck $\Delta$, eine Kante $e$ und einen Punkt $q$, sodass sich $q$ im Umkreis-Shape von $\Delta$ befindet, mit $e$ als zugewandte Seite des Dreiecks. Zwar könnte man einwenden, dass für allgemeine konvexe Shapes die Existenz von Dreiecken gar nicht mehr garantiert ist (schlimmstenfalls bilden ja die Delaunay-Kanten nur einen Baum), allerdings kann man die ungültigen Dreiecke dazu nehmen und hat damit als Datenstruktur immer eine vollständige Triangulierung (als Umkreis-Shape wird einem ungültigen Dreieck der unendliche Shape durch die Kante $e$ zugeordnet).

Es wird nun jenes Tripel $(\Delta, e, q)$ betrachtet, wo $q$ die Kante $e$ unter dem größten Winkel sieht. Hier könnte man meinen, dass für allgemeine konvexe Shapes (wie beim maximierenden Kriterium) der Winkel kein taugliches Mittel darstellt, das ist aber nicht der Fall, weil der Winkel hier

nicht als Größenindikator für den Umkreis gebraucht wird.
Wegen lokaler Delaunay-Eigenschaften kann der Punkt $q$ nicht die Ecke des Nachbardreiecks von $\Delta$ sein, dieses sei $\Delta'$ mit Kante $e$' und neuem Eckpunkt $p$. Punkt $p$ liegt nicht im Umkreis-Shape von $\Delta$ (lokal Delaunay), wohl aber $q$, und damit liegt $q$ auch im Umkreis-Shape von $\Delta'$. Nun liefert aber das Tripel $(\Delta', e', q)$ den größeren Winkel $\rightarrow$ Widerspruch zur Annahme.

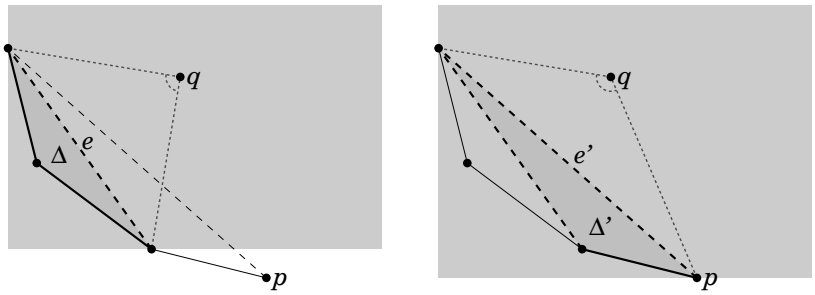


Abbildung 44: Beispiel mit quadratischem Shape und ungültigen Kanten/Dreiecken, die es nur am Rand geben kann (andernfalls nicht lokal Delaunay).

Zu beachten ist in diesem Zusammenhang noch der von Drysdale erwähnte Spezialfall mit Parallelverschiebung, welcher bei Shapes mit 2 parallelen Kanten auftreten kann, und der quasi ein Gegenbeispiel zum eben gezeigten Beweis darstellt, was sich aber leicht bereinigen lässt.

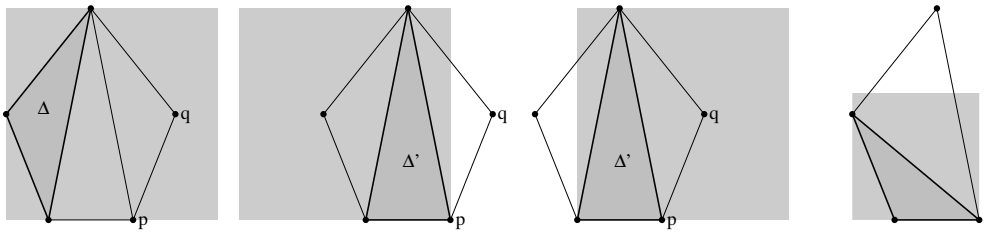


Abbildung 45: Beispiel mit quadratischem Shape und Parallelverschiebung.

Offensichtlich handelt es sich im Beispiel links nicht um eine (globale) Delaunay-Triangulierung, da sich Punkt $q$ im "Umkreis" von $\Delta$ befindet, es gibt aber wegen der Parallelverschiebung eine Variante vom $\Delta'$-Umkreis, die $q$ nicht enthält, scheint also lokal Delaunay. Das gleiche gilt auch für den linken Punkt von $\Delta$, allerdings ist $\Delta \cup \Delta'$ alleine gesehen auch schon fraglich, denn es gibt schließlich auch einen $\Delta'$-Umkreis, der den Punkt sehr wohl enthält.

Glücklicherweise gibt es keinen Grund, den Fall zu dulden, denn es handelt sich hier schließlich um Kozirkularität von 4 Punkten, wo man einfach die andere Diagonale wählen kann, bei der das Problem nicht auftritt (Abbildung 45 rechts).
Das heißt, bei einem Algorithmus, der nur auf Incircle-Tests beruht, muss man im Fall von Parallelverschiebung einfach den ganzen Bereich testen. Wenn man hingegen auf Größenunterschiede der Umkreise testet, ist das Problem irrelevant, da man in dem Fall mit kleineren Umkreisen automatisch die bessere Wahl trifft.

# 7 Algorithmen

## 7.1 Konvexe Distanzfunktion

Da sich jeder konvexe Shape beliebig nahe durch ein konvexes Polygon approximieren lässt, kann man sich im Folgenden auf die Betrachtung von Polygonen beschränken.
Gegeben sei ein konvexes Polygon P $(p_1, p_2, ...p_n)$ im Uhrzeigersinn mit dem Ursprung im Inneren, und gesucht sei die Distanzfunktion $d_P(a, b)$. Die Polygonkante, welche von der Geraden durch die Punkte a und b geschnitten wird, kann mit Hilfe von binärer Suche in $O(log\ n)$ Zeit ermittelt werden.

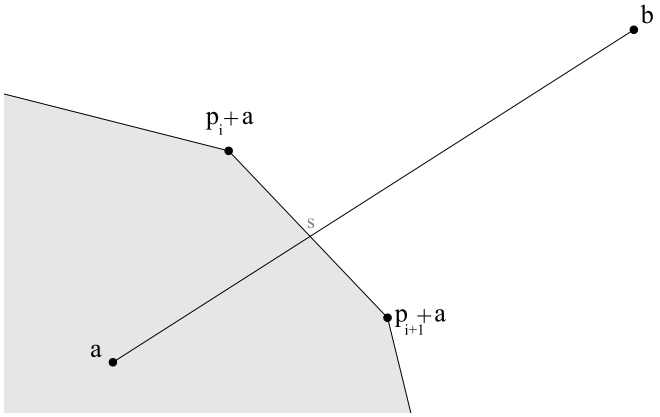


Abbildung 46: $d_P(a, b)$

Für die konvexe Distanz gilt dann: $d_P(a, b) = \frac{(p_{i+1}-p_i)\times(b-a)}{p_{i+1}\times p_i}$,
wobei das 2-dimensionale Kreuzprodukt definiert sei mit $u \times v = det[u\ v] = u_x v_y - u_y v_x$.
Wie man sieht, müssen für das Distanzverhältnis $\frac{\|b-a\|}{\|s-a\|}$ weder der Schnittpunkt $s$, noch die Distanzen $\|b-a\|$ und $\|s-a\|$ explizit berechnet werden.

Im Folgenden wird ein Beispielalgorithmus zur Ermittlung von $d_P(a, b)$ gezeigt.

$i \leftarrow 1$
$j \leftarrow n + 1$
**while** $j - i > 1$ **do**
  $k \leftarrow (i + j) \div 2$
  $c_1 \leftarrow (b - a) \times p_i$
  $c_2 \leftarrow (b - a) \times p_k$
  $c_3 \leftarrow p_k \times p_i$
  **if** $(c_1 > 0)$ and $(c_2 \leq 0)$ **then**
    **if** $k = i + 1$ **then**
      $d_P(a, b) \leftarrow \frac{c_1 - c_2}{c_3}$
    **end if**
    $j \leftarrow k$
  **else**
    **if** $((c_1 > 0)$ or $(c_2 \leq 0))$ and $(c_3 < 0)$ **then**
      $j \leftarrow k$
    **else**
      $i \leftarrow k$
    **end if**
  **end if**
**end while**

## 7.2 Symmetrischer Vereinigungsgraph

Die neu definierte Metrik $m_C$ ist schwieriger zu berechnen als die Distanzfunktion $d_C$, weil die Lage des gesuchten Durchmessers unklar ist, und auf direktem Weg kein $O(log\, n)$-Algorithmus möglich zu sein scheint.
Wegen $m_C(a, b) = d_{\widehat{C}}(a, b)$ kann man hier Abhilfe schaffen, indem man zuerst einmalig den symmetrischen Vereinigungs-Shape $\widehat{C}$ ermittelt, danach erhält man wie gewohnt die konvexe Distanz $d_{\widehat{C}}(a, b)$ in $O(log\, n)$ Zeit.

Folgend wird ein einfacher Algorithmus gezeigt, um aus einem gegebenen konvexen Polygon $p_1, p_2, ...p_n$ (im Uhrzeigersinn, mit zyklischer Addition $p_{n+1} := p_1$ etc.) das entsprechende symmetrische Vereinigungspolygon $q_1, q_2, ...q_m$ zu erstellen, mit $n \leq m \leq 2n$ und Laufzeit O(m).

$m \leftarrow 0$
$j \leftarrow 2$
**while** $(p_1 - p_n) \times (p_{j+1} - p_j) \leq 0$ **do**
  $j \leftarrow j + 1$
**end while**
**for** $i = 1$ to n **do**
  **repeat**
    $m \leftarrow m + 1$
    $q_m \leftarrow p_j - p_i$
    $c \leftarrow (p_{i+1} - p_i) \times (p_{j+1} - p_j)$
    **if** $c \leq 0$ **then**
      $j \leftarrow j + 1$
    **end if**
  **until** $c \geq 0$
**end for**

Sofern man bei der Ermittlung von $\widehat{C}$ neben den Ecken ($q_m \leftarrow p_j - p_i$) auch deren Ursprungs-Ecken aus $C$ mitspeichert ($r_m \leftarrow p_j$), kann man in Folge als Nebenprodukt der Berechnung von $d_{\widehat{C}}(a, b)$ auch direkt den minimalen Shape $\overline{C}(a, b)$ eruieren, den man für einen Gabriel-Test (oder als Basis für einen Incircle-Test) brauchen kann.

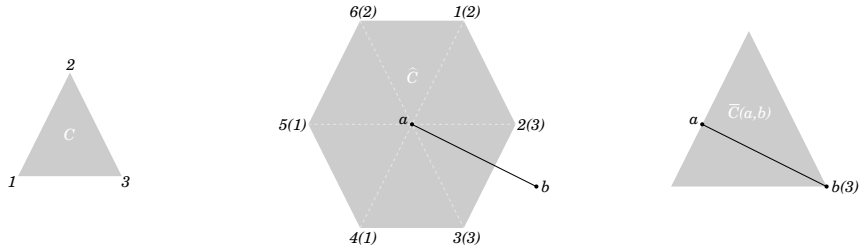


Abbildung 47: Beispiel mit dreieckigem Shape $C$ und dem mit obigem Algorithmus ermittelten sechseckigen Shape $\widehat{C}$, wobei in Klammern die Ursprungs-Ecken von $C$ angeführt sind. Berechnet man nun $d_{\widehat{C}}(a, b)$, so hat man bereits den Skalierungsfaktor von $\overline{C}(a, b)$, es fehlt nur noch der "Ankerpunkt". Haben die beiden Endpunkte der geschnittenen Polygonkante die selbe Urspungs-Ecke, so wird diese in den Punkt $b$ gelegt (wie hier im Beispiel Ecke 3). Andernfalls haben entweder die beiden Endpunkte der gegenüberliegenden Polygonkante (wegen Symmetrie ist diese bekannt) die selbe Ursprungs-Ecke, und diese wird in Punkt $a$ gelegt, oder es gibt eine Parallelverschiebung mit zwei unterschiedlichen Ankerpunkten (siehe Abb. 22).

## 7.3 Inkrementeller Algorithmus

Den inkrementellen Einfüge-Algorithmus zum Aufbau einer Delaunay-Triangulierung (siehe z.B. *D-Z Du/F.K.Hwang*[3]) kann man auch für konvexe Shapes adaptieren, indem man als Datenstruktur die vollständige Triangulierung beibehält und die ungültigen Kanten, die nicht Teil der $DT_C(S)$ sind, einfach als solche markiert und mit jedem Einfügeschritt aktualisiert.

Abbildung 48: Beispiel einer gewöhnlichen $DT(S)$ im Vergleich zu einer $DT_C(S)$ mit dreieckigem Shape, wo die ungültigen (strichlierten) Kanten das eigentliche Gerüst der Delaunay-Kanten zu einer vollständigen Triangulierung auffüllen.

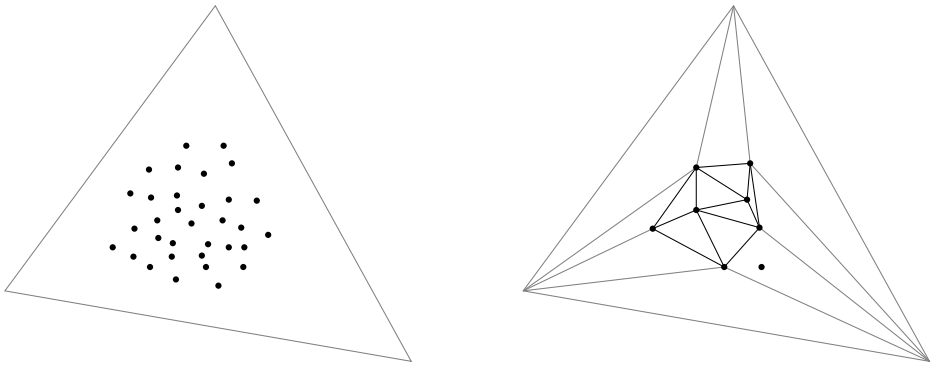

Abbildung 49: Gestartet wird der Algorithmus wie üblich mit einem großen Hilfsdreieck, welches die gesamte Punktmenge umschließt, sodass man von Anfang an per *point location* jeden nacheinander eingefügten Punkt in einem Dreieck finden kann.

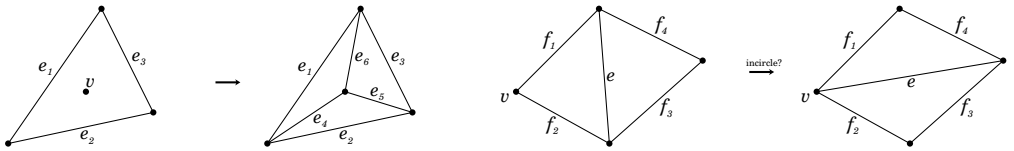


Abbildung 50: Für einen neu eingefügten Punkt wird das umgebende Dreieck lokalisiert, 3 neue Kanten werden erzeugt, und die Triangulierung wird (falls nötig) per Flips angepasst. Der Unterschied beim adaptierten Algorithmus ist nur, dass die betrachteten Kanten auch ungültig sein können.

Im folgenden Algorithmus sind *push* und *pop* die üblichen Stack-Operationen, *incircle* steht allgemein für die Entscheidungsfunktion, ob eine Flip notwendig ist, und *validate* ist die Prozedur, welche eine Kante in Abhängigkeit von Bereich I (Abbildung 13) als gültig/ungültig markiert.
Bei der Laufzeitanalyse gibt es keinen Unterschied zum originalen Einfüge-Algorithmus, das bedeutet für die Erzeugung einer $DT_C(S)$ aus $n$ Punkten $O(n^2)$ im worst-case, und bei randomisiertem Einfügen erwartete $O(n \log n)$.

push($e_1$)
push($e_2$)
push($e_3$)
validate($e_4$)
validate($e_5$)
validate($e_6$)
**while** stack not empty **do**
  pop($e$)
  **if** incircle($e$) **then**
    flip($e$)
    validate($f_1$)
    validate($f_2$)
    push($f_3$)
    push($f_4$)
  **end if**
  validate($e$)
**end while**

## 7.4 Flip-Algorithmus

Ziel des Flip-Algorithmus ist es, aus einer beliebigen Triangulierung $T(S)$ nur durch Flips die gesuchte $DT_C(S)$ zu erreichen. Grundvoraussetzung ist, dass Quell- und Ziel-Triangulierung die gleiche Kantenanzahl aufweisen, was durch die interne Beibehaltung einer vollständigen Triangulierung mit Hilfe ungültiger Kanten gewährleistet wird.
Wie in Abschnitt 6.2 zu sehen, gilt lokal Delaunay gleich global Delaunay, d.h. wenn keine lokale Verbesserung mehr möglich ist, hat man automatisch ein korrektes Ergebnis. Damit der Algorithmus terminiert, muss darauf geachtet werden, dass im Zuges des Incircle-Tests nicht bei gleichwertigen Situationen (Kozirkularität oder beide Diagonalen ungültig) geflippt wird, und der Test muss exakt sein, da ansonsten auch Rundungsfehler zu einer Endlosschleife führen können.
Empirische Tests legen eine lineare Abhängigkeit der Anzahl benötigter Flips von der Anzahl der Kanten nahe, aus $O(n)$ Flips würde dann eine Laufzeit von $O(n^2)$ im worst-case folgen, da die Anzahl der Durchgänge, in denen alle Kanten getestet werden, nicht höher sein kann, als die der Flips.

```
repeat
  for all edges do
    if incircle(e) then
      flip(e)
    end if
    validate(e)
  end for
until edges remain unchanged
```

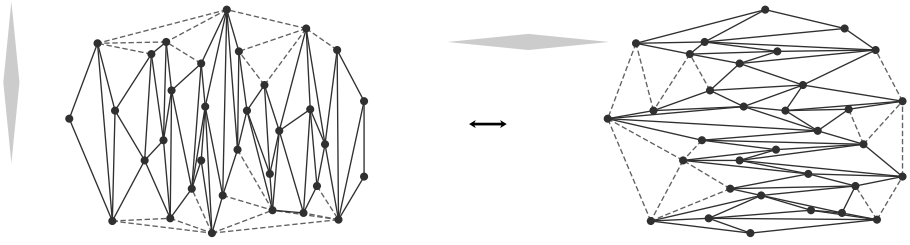

Abbildung 51: Am meisten Flips sind beim Wechsel von langen hochgestellten Shapes zu langen quergestellten Shapes nötig, hier kann die Anzahl der Flips jene der Kanten übertreffen (bleibt aber auch bei Tests mit mehreren Tausend Kanten zumindest unter der doppelten Kantenanzahl).

# 8 Testprogramm

Im Zuge dieser Diplomarbeit wurde auch ein Programm (unter Lazarus/Freepascal) erstellt, zum einen, um theoretische Ergebnisse visualisieren zu können, zum anderen, um noch unklare Zusammenhänge in der Praxis testen zu können bzw. empirische Daten zu erhalten, und nicht zuletzt auch deswegen, um die schriftliche Arbeit mit selbst erstellten korrekten Abbildungen von Convex-Shape-Delaunay-Triangulierungen etc. bereichern zu können.
Als zusätzlicher Bonus hat die Visualisierung unter anderem auch zur Entdeckung von Zusammenhängen geführt, wie z.B. dass für dreieckige Shapes Delaunay-Kanten mit Gabriel-Kanten zusammenfallen, was zwar in der Theorie trivial zu zeigen ist, dem aber ansonsten vermutlich gar keine Beachtung zu Teil geworden wäre.
Während Drysdale[1] einen Merge-Algorithmus beschreibt, wurde hier ein Einfüge-Algorithmus implementiert, sowie ein Flip-Algorithmus, welcher von einer beliebigen Triangulierung ausgehend mit Flips die Delaunay-Triangulierung erreicht. Als Datenstruktur wurde eine (doppelt verkettete) Liste von Knoten mit den Koordinaten plus ein Liste von Kanten verwendet, wobei eine Kante jeweils auf ihre beiden Endpunkte verlinkt ist, sowie auf die 2 Knoten und 4 Kanten, die das rechte und linke Nachbardreieck bilden. Ungültige Kanten werden markiert und nicht angezeigt (bzw. nur optional), bleiben aber in der Datenstruktur erhalten, dadurch hat man intern immer ein vollständige Triangulierung, was zum einen Grundvoraussetzung für den genannten Flip-Algorithmus ist, zum anderen bleiben Dreiecks-Suchalgorithmen und Laufzeitanalysen im Vergleich zu gewöhnlichen Delaunay-Triangulierungen unverändert.

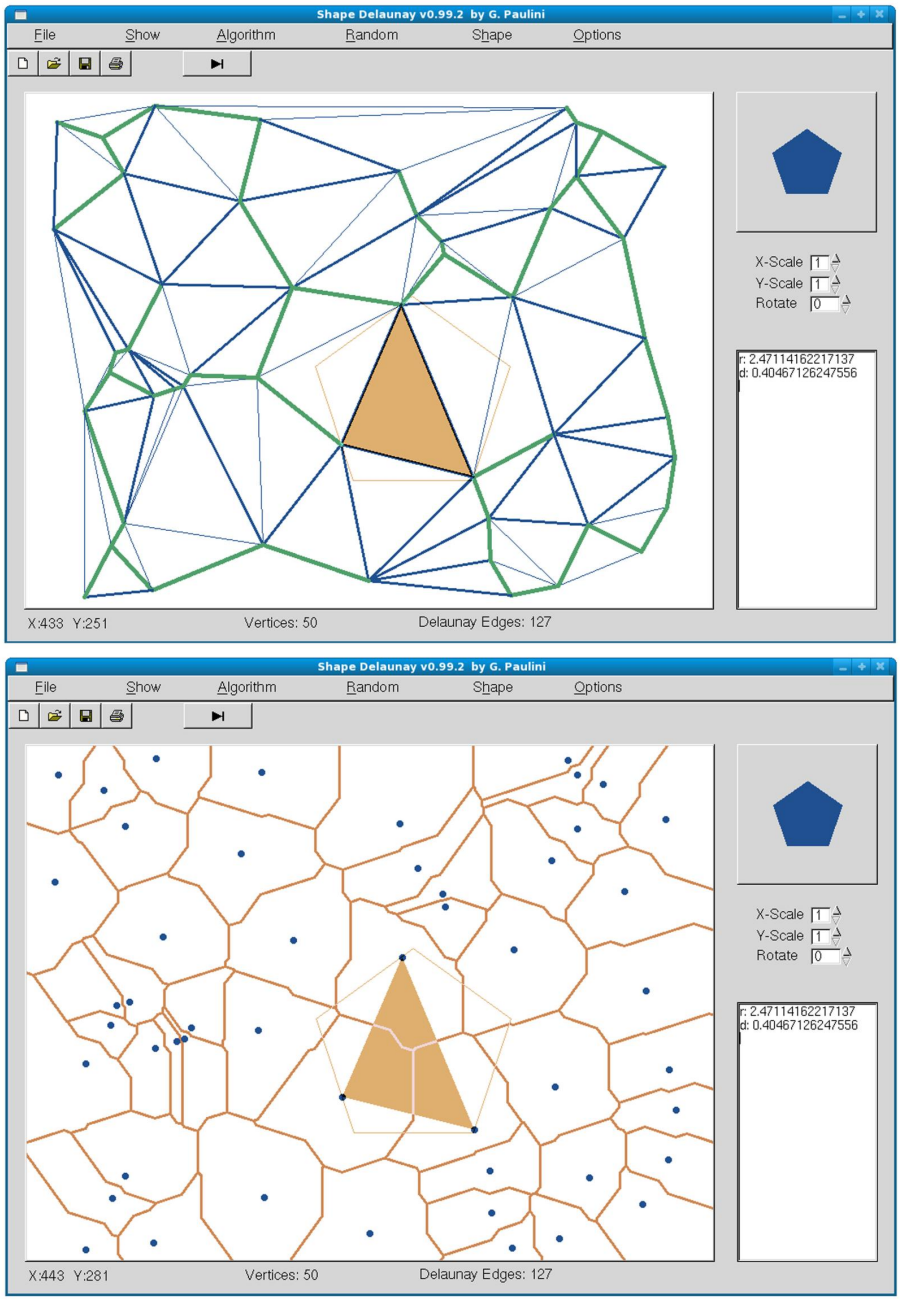


Abbildung 52: Beispiel zur Darstellung des Testprogramms, oben Delaunay-Kanten, Gabrielkanten und minimaler Spannbaum eines fünfeckigen Shapes $C$, unten ein entsprechendes Voronoi-Diagramm (jeweils mit selektiertem Dreieck).

# 9 Offene Problemstellungen

Im Zuge dieser Arbeit sind manche Fragen offen geblieben bzw. wurden neue aufgeworfen.

- Wieviele Convex-Shape-Delaunay-Triangulierungen gibt es eigentlich für eine gegebene Punktmenge $S$, und kann man einen Algorithmus finden, der entscheidet, ob eine gegebene Triangulierung einer $DT_C(S)$ entspricht oder nicht, und wenn ja, einen entsprechenden Shape $C$ findet?
  Es ist zumindest naheliegend, dass es wesentlich weniger Triangulierungen gibt, welche eine $DT_C(S)$ darstellen, als solche, für die es keinen passenden Shape gibt, denn wie in Abschnitt 2.2 (Abbildung 15) zu sehen ist, reichen bereits 2 Kanten in entsprechender Lage für einen Widerspruch zur Annahme, eine Triangulierung sei eine $DT_C(S)$. Unklar ist hingegen, wie zuverlässig so ein Widerspruch gefunden werden kann, und falls es keinen gibt, wie genau ein passender Shape bestimmt werden kann. Immerhin gibt es wie erwähnt diverse Ansätze, wie man vom Rand der Triangulierung auf den Rand des Shapes schließen kann und vom Inneren der Triangulierung auf Länge bzw. Ausrichtung des Shapes.
  Allerdings gibt es für eine $DT_C(S)$ unendlich viele Shapes $C$, welche die nötigen Anforderungen erfüllen, es stellt sich also die zusätzliche Frage, inwieweit man den einfachsten Shape (ein konvexes Polygon mit der Minimalanzahl von Ecken) finden kann.

- Für den Fall, dass für die Umkreisradien gilt: $min(r_1, r_2) > min(r_3, r_4)$ (siehe Abschnitt 6.1), kann man beweisen, dass das Summenkriterium $r_1 + r_2 < r_3 + r_4$ hält?

- Wieviele Flips sind notwendig, um per Flip-Algorithmus von einer beliebigen Triangulierung zu $DT_C(S)$ zu gelangen?
  Der Beweis dafür $O(n)$ Flips und $O(n^2)$ Zeit ist noch ausständig, es gibt allerdings wenig Grund zur Annahme, dass es hier ordnungsmäßig ein anderes Verhalten gibt, als im Falle der gewöhnlichen Delaunay-Triangulierung. In dem Zusammenhang wäre vielleicht noch interessant, inwieweit man für $DT_{C_1}(S) \rightarrow DT_{C_2}(S)$ eine genauere Abschätzung für die Anzahl der Flips finden kann, in Abhängigkeit von der Ähnlichkeit von $C_1$ und $C_2$.

# Literatur


[1] R. L. Drysdale, III. *A practical algorithm for computing the Delaunay triangulation for convex distance functions.*

[2] L. Ma. *Bisectors and Voronoi Diagrams for Convex Distance Functions.*

[3] D-Z Du und F.K.Hwang. *Computing in Euclidean Geometry.*

[4] F. Aurenhammer. Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Computing surveys* 23 (1991), 345–405.

[5] F. Aurenhammer und R. Klein. Voronoi diagrams. *Handbook of Computational Geometry*, Chapter V , J. Sack and G. Urrutia, editors, pages 201-290.

[6] L.P. Chew and R.L.S. Drysdale. Voronoi diagrams based on convex distance functions. *Proc. 1st Ann. ACM Symposium on Computational Geometry*, 1985, 235–244.

[7] S. Fortune. Voronoi diagrams and Delaunay triangulations. In D.-Z. Du and F. K. Hwang (eds), *Computing in Euclidean Geometry*, Lecture Notes Series on Computing 1, World Scientific, Singapore, 1992, 193–233.

[8] T. Lambert. Systematic local flip rules are generalized Delaunay rules. *Proc. 5th Ann. Canadian Conference on Computational Geometry*, 1993, 352–357.

[9] J.R. Shewchuk. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. *Discrete & Computational Geometry* *18*:305-363, 1997.

[10] O. Devillers/S. Pion. Efficient Exact Geometric Predicates for Delaunay Triangulations. *Thème 2 - Génie logiciel et calcul symbolique.* Projets Prisme, INRIA Sophia Antipolis. France.