

Voronoi-Relaxierung allgemeiner Objekte

Stefan Hiller*, Oliver Deussen
Technische Universität Dresden

Zusammenfassung

Die vorliegende Arbeit beschreibt ein Verfahren zur effizienten und schnellen Berechnung zentroidalen Voronoi-Tesselierungen nahezu beliebig geformter zweidimensionaler Objekte. Die Besonderheit der hierfür entwickelten Berechnungsmethode ist dabei, daß alle Berechnungen durch grafische bzw. numerische Verfahren abgedeckt werden und nicht analytisch erfolgen. Hierdurch ist der Einsatz von Graphikhardware möglich und damit eine effiziente Behandlung allgemeiner Objekte. Anhand einiger Anwendungsbeispiele wird die Nützlichkeit der Verfahren demonstriert.

1 Motivation

In [DHOS00] wurde beschrieben, wie punktförmige Objekte aus einer zufälligen oder regelmäßigen Verteilung über Relaxierung auf der Basis von Voronoi-Diagrammen in eine sogenannte zentroidale Voronoi-Tesselierung überführt werden können, in der jedes Objekt im Schwerpunkt seines Voronoi-Gebietes sitzt. Die Anordnung der Punktobjekte entspricht dann einer Poisson-Disc Verteilung. Diese Verteilungen wurden anschließend verwendet, um Punkte in einer nichtrealistischen gepunkteten Computergraphik (engl. Stippling) anzuordnen. Es wurde dabei empirisch gezeigt, daß Künstler ihre Punkte beim Anfertigen solcher Darstellungen auf ähnliche Weise setzen. Schon vorher wurden dieselben Punktverteilungen für die Positionierung von Pflanzen auf einer Oberfläche verwendet [DHL⁺98], da auch in diesem Falle für manche Pflanzenspezies eine Poisson-Disc Verteilung angenommen werden kann.

In beiden Fällen stellt sich die Frage, ob man solche Verteilungen nicht auch für andersartige Objekte erzeugen kann. Im Fall der nichtrealistischen Computergraphik könnten dann neuartige nichtrealistische Darstellungen mit beliebigen Objekten generiert werden, im Falle der Modellierung von Ökosystemen könnte man Objekte wie Äste oder Baumstämme auf einer Oberfläche ansprechend verteilen. Für weitere Anwendungen genügt es ebenfalls nicht, die einzelnen Objekte lediglich als Punkte zu behandeln. Hier besteht die Notwendigkeit, zentroidale Voronoi-Verteilungen von beliebig geformten Objekten (d.h. in der Regel von beliebig geformten Polygonen) zu erzeugen.

Die primäre Zielstellung dieser Arbeit liegt daher in der schnellen Berechnung der Voronoi-Gebiete und der effizienten Durchführung der Relaxierungsschritte zur Erzielung einer zentroidalen Voronoi-Tesselierung allgemeiner Objekte. Hierbei hilft eine Arbeit von K. Hoff et al. [HCK⁺99], die eine vorher schon in [NDW93] vorgestellte Berechnungsmethode mit Graphikhardware nutzt.

*Fakultät für Informatik, Institut für Software und Multimediatechnik, D-01062 Dresden, Germany

2 Voronoi-Gebiete und deren graphische Bestimmung

Der von uns betrachtete Raum zur Ermittlung eines Voronoi-Diagrammen ist in der Regel der R^2 , die Objekte dieses Raumes seien die Elemente einer n-elementigen Menge $S = \{e_1, e_2, \dots\}$. Das Voronoi-Diagramm der Elementmenge S wird hier unter Zuhilfenahme des euklidischen Abstandsmaßes bestimmt. Für zwei gegebenen Elemente e_1, e_2 ist der Bisektor die Menge aller Punkte p aus R^2 , welche zu e_1 und e_2 den gleichen Abstand haben. Er teilt den Ebene/Raum in zwei offene Halbebenen.

$$\begin{aligned} \text{Bisektor:} & \quad B(e_1, e_2) = \{p \in R^2; |e_1p| = |e_2p|\} \\ \text{1. Halbebene:} & \quad D(e_1, e_2) = \{p \in R^2; |e_1p| < |e_2p|\} \\ \text{2. Halbebene:} & \quad D(e_1, e_2) = \{p \in R^2; |e_1p| > |e_2p|\} \end{aligned}$$

Das Voronoi-Gebiet zu einem Element e ist eine kompakte Region (für Punkte ein konvexes Polygon), welches immer auch e enthält.

$$\text{Voronoi-Gebiet: } VR(e_1, S) = \bigcap D(e_1, e_2), e_2 \in S \setminus e_1 \quad (1)$$

Zwei verschiedene Elemente e_1, e_2 haben disjunkte Voronoi-Regionen, da kein $p \in R^2$ gleichzeitig zu $D(e_1, p)$ und $D(e_2, p)$ gehören kann. Die Vereinigung der Voronoi-Gebiete ist eine Tesselierung der Ebene R^2 , wenn die Punkte der Bisektoren (Voronoi-Kanten) zwischen den Elementen hinzugenommen werden.

Zur Ermittlung von Voronoi-Diagrammen allgemeiner Objekte muss der euklidische Abstands begriff erweitert werden. Der kleinste Abstand zwischen einem Punkt p der Zeichenebene und einem Objekt e wird definiert als:

$$|ep| = \min |qp|, q \in e, p \in R^2 \quad (2)$$

Es wird also jeweils das Minimum der Abstände von Objektpunkten zu p verwendet. Der Bisektor zwischen einem Punkt und einem Liniensegment besteht dann aus einer Strecke und zwei Parabeln¹. Der Bisektor zwischen zwei Liniensegmenten aus vier Liniensegmenten und drei Kurven.

Solche Voronoi-Diagramme können mit Hilfe einer 3D-Grafikpipeline relativ einfach praktisch bestimmt werden. Die Anforderungen an die Pipeline sind dabei lediglich die Existenz eines Tiefenpuffer mit individuell gespeichertem Tiefenwert für jedes Pixel, eine Möglichkeit zum Zeichnen farbiger 3D-Dreiecke und eine orthogonale Projektion.

In [NDW93](S. 406) sowie [HCK⁺99] wird ein Ansatz zur Berechnung von Voronoi-Diagrammen allgemeiner Objekte mit Hilfe von OpenGL präsentiert. Dieser Ansatz ist sowohl effizient durch den Einsatz von Graphikhardware zu implementieren, als auch genügend flexibel um, eine Reihe von Effekten zu testen. Der grafische Berechnungsansatz kann auch mit einer anderen Grafikkbibliothek implementiert werden wie z.B. Direct3D, wichtig sind nur die genannten Anforderungen an die Grafikpipeline. Farbige Darstellung

¹Bildet die Linie die x-Achse in einem Koordinatensystem und liegt der Punkt p auf der y-Achse, so gehört jeder Punkt zum Bisektor, der die Gleichung $y = x^2/2a + a \cdot 2$ erfüllt.

benötigen wir zur Farbkodierung der einzelnen Objekte. Die 3D-Dreiecke, der Z-Puffer (Tiefenpuffer) und die orthogonale Projektion gestatten es, die euklidische Entfernungsbestimmung konstruktiv umzusetzen. Die orthogonale Projektion ist wichtig, um keine geometrische Verzerrung zu erzeugen die die Lage der Bisektoren verfälschen würde.

Die Beschreibung des grafischen Ansatzes wird deutlich wenn wir zunächst von Punkten zu Linien und anschließend zu Linienzügen/Polygonen übergehen. Wie aus der allgemeinen Definition für Voronoi-Diagramme hervorgeht liegt ein Hauptproblem in der Suche nach der Menge von Bisektoren. Für den trivialen Fall, bei dem lediglich zwei Punkte gegeben sind, ist der Bisektor gleich der Mittelsenkrechten zwischen den beiden Punkten.

Diese Mittelsenkrechte kann graphisch bestimmt werden, wenn an der Position der Punkte auf der Zeichenebene jeweils ein in die Tiefe gerichteter unendlicher Kegel gestellt wird, dessen Spitze zum Betrachter zeigt. Der Schnitt zwischen beiden Kegeln ist genau der gesuchte Bisektor (Blickrichtung senkrecht zur Anordnungsebene). Nutzt man den Tiefenpuffer zur Ermittlung des Bisektors, so besteht dieser aus genau den Pixeln, für die die Tiefe auf beiden Objekten dieselbe ist.

In Abbildung 1 ist die Seitenansicht des Tiefenpuffers dargestellt. An jedem Punkt ist ein Kegel befestigt.

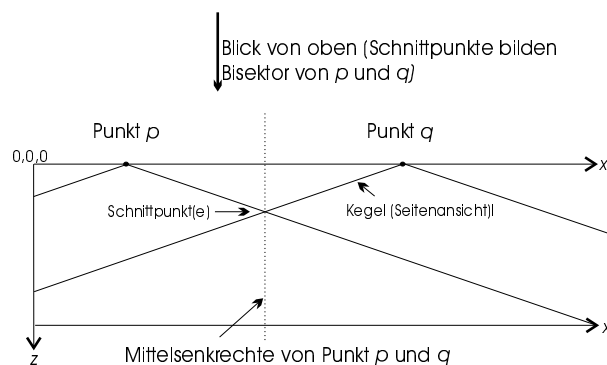


Abbildung 1: Geometrische Konstruktion für einen Fall mit zwei gegebenen punktförmigen Objekten

Dabei ist zu beachten, daß der Anstieg der Kegel gleich groß sein muß. Die resultierenden Schnittpunkte der Hilfslinien (Kegelmäntel) bestimmen die Lage der Mittelsenkrechten und damit die Lage der Bisektoren. Für mehrere Punkte ergibt sich die Konstruktion analog: Zuerst werden die Punkte so im dreidimensionalen Raum angeordnet, daß sie auf einer gemeinsamen Ebene liegen (in der Regel X-Y Ebene). Wieder wird an der Position jedes Punktes die Spitze eines Kreiskegels angesetzt. Ein dreidimensionaler Kegel wird unter OpenGL/Direct3D durch eine Menge von Dreiecken approximiert. Um die durch die Bildauflösung maximal mögliche Genauigkeit auszunutzen, sollte die optimale Anzahl der Kegelsegmente S bestimmt werden. Die Bestimmungsformel hat als Eingabegrößen den Radius der Kegelgrundfläche R sowie den max. Fehlerwert η in Pixel (z.B. $\eta = 0.5$). Ergebnis ist der spitze Winkel α der gleichschenkligen Dreiecke, die einen einzelnen Kegel

approximieren.

$$\alpha = 2 \cdot \cos^{-1} \left(\frac{(R - \eta)}{R} \right) \quad (3)$$

Die Bestimmung der Anzahl der Kegelsegmente ist damit trivial:

$$S = \frac{360^\circ}{\alpha} \quad (4)$$

Die Kegel sind gleich hoch und besitzen den gleichen Grundflächenradius. Ein optimaler Radius für die Kegelgrundfläche ist die größte Diagonale der Zeichenfläche, um im gesamten Z-Puffer zu jeder Zeit gültige Z-Werte zu erzeugen. Je nach Anzahl und Lage der Objekte kann der gemeinsame Radius verkleinert werden - was zu Performancesteigerung führen kann, weil weniger Clipping-Operationen notwendig sind.

Wählt man für jeden Punkt eine andere Kegelfarbe und betrachtet die Zeichenfläche senkrecht zur Punktebene (Kegelspitzen zeigen zum Betrachter) sowie eine orthogonale Projektion, so bilden sich durch die gegenseitige Überschneidungen genau die Voronoigebiete der entsprechenden Punkte. Die Schnittberechnung erfolgt damit vollständig durch die Grafikhardware.

Für Punktobjekte verwendet man durch Dreiecke approximierte Kreiskegel, für Linien setzt sich die Approximation aus zwei halben Kegeln an den Endpunkten und deren Verbindung durch Vierecke zusammen (siehe Abbildung 2).

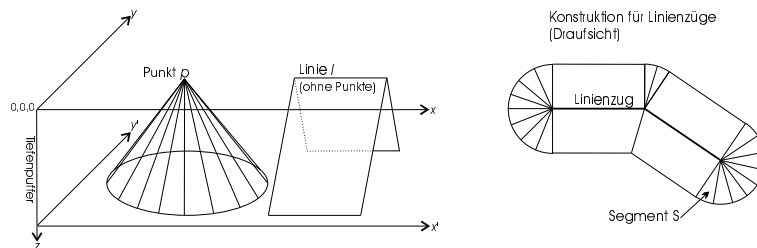


Abbildung 2: Visualisierung der geometrischen Konstruktion durch verallgemeinerte Kegel/Keglschnitte und Prismen

Abbildung 3 zeigt das Ergebnis des Algorithmus am Beispiel von Punktobjekten. Die Farbkodierung hat den Vorteil, daß die Voronoi-Gebiete direkt aus dem Farbbild bestimmt und dann den Objekten zugeordnet werden können. Der Tiefenpuffer muss somit gar nicht explizit ausgewertet werden.

Die Genauigkeit der Berechnung ist bei diesem Ansatz natürlich begrenzt. Sie hängt im wesentlichen von der räumlichen Auflösung der Zeichenfläche ab. Durch ausschnittsweises Berechnen (Zooming) kann jedoch die Genauigkeit lokal an gegebene Forderungen angepasst werden. Der Nachteil der geringen Auflösung wird durch die schnelle und einfache Berechnung in vielen Anwendungsfällen kompensiert. In den folgenden Abschnitten wird die Verknüpfung der hardware-basierten Bestimmung von Voronoi-Gebieten mit dem

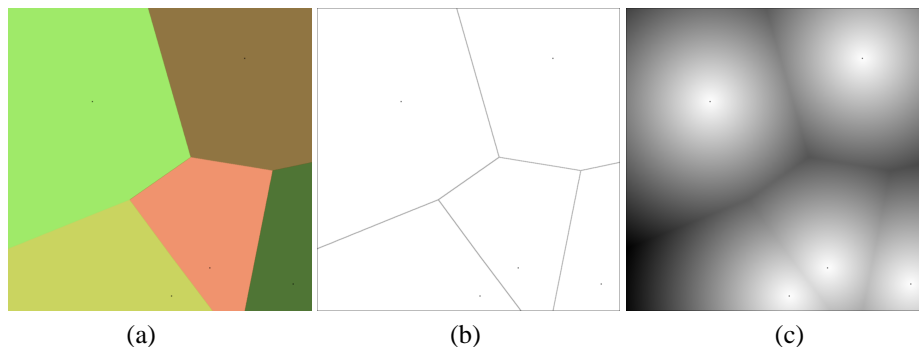


Abbildung 3: a) farbkodierte Voronoi-Gebiete, b) Voronoi-Kanten, c) grauwertkodierter Z-Puffer (Basis: fünf punktförmige Generatoren)

Relaxierungsverfahren vorgestellt, welches es ermöglicht, die gewünschten Anordnungen allgemeiner Objekte herzustellen.

3 Voronoi Iteration

Das Problem der lokalen Optimierung von Punktanordnungen wurde bereits Anfang des 19ten Jahrhunderts von Launhardt (1882) und Weber (1909) untersucht. Mitte des 20. Jahrhunderts schritt die Entwicklung entsprechender Lösungsverfahren weiter voran. Heute können Anordnungsprobleme mit einer großen Anzahl von Punkten auf der Ebene oder dem Raum mit Hilfe von Voronoi-Diagrammen gelöst werden. Beispiele sind in [OBS97] zu finden. Nachteil vieler Verfahren ist jedoch die relativ hohe Zeitkomplexität. Diese resultiert aus der aufwendigen Lösung der Zielfunktion:

$$F(x_1, \dots, x_n) = \sum_{i=1}^n \int_{V_i} f(\sqrt{\|x - x_i\|^2}) \phi(x) dx. \quad (5)$$

Die Funktion $\phi(x)$ beschreibt hierbei die Dichte der Ziel-Anordnung (Bewertungsfunktion). $f(\sqrt{\|x - x_i\|^2})$ ist eine einfache Kostenfunktion. Dabei ist der Ausdruck $\sqrt{\|x - x_i\|^2}$ der Euklidischen Abstand der Punkte x und x_i .

Diese allgemeine Optimierung kann über Voronoi-Gebiete lokal angenähert werden. Hierzu wird das Voronoi-Diagramm der Punktmenge bestimmt und jeder Punkt in den Schwerpunkt seines Voronoi-Gebietes verschoben. Das Verfahren wird auch Lloyd's Methode genannt [OBS97]. Seien V_i die Voronoi-Gebiete der Punkte p_i , aus Gleichung 1.

Der Grundalgorithmus der Voronoi-Relaxierung (Iteration) lautet wie folgt:

Globale Relaxierungs-Bedingung: Alle Operationen der Relaxierung verschieben die Objekte innerhalb ihrer Voronoi-Gebiete $V_g(O_i)$, $O_i \in O$ mit $O \subseteq F \subseteq R^2$, wobei F die verwendete Zeichenfläche ist.

Der entwickelte Relaxierungs-Operator wird folgendermaßen definiert:

Eingabe:	Objektmenge O (2D-Koordinaten der Objekte) Bedingung: Alle Objekte $O_i \in O$ liegen innerhalb von F .
Ausgabe:	Objektmenge O' mit verschobenen und gedrehten Basiskoordinaten.
Vorberechnung:	Bestimme Voronoi-Gebiete $V_g(O_i)$ aller $O_i \in O$, bestimme $O_{i_{sp}} = \text{Schwerpunkt}(O_i)$, bestimme $O_{i_{ta}} = \text{Hauptmomente}(O_i)$.
Verschiebung/Drehung:	Für alle $O_i \in O$, nimm $V_g(O_i)$ aus $V_g(O)$, bestimme $V_g(O_i)_{sp} = \text{Schwerpunkt}(V_g(O_i))$, bestimme $V_g(O_i)_{ta} = \text{Hauptmomente}(V_g(O_i))$, verschiebe $O_{i_{sp}}$ auf $V_g(O_i)_{sp}$, drehe O_i , so daß $O_{i_{ta}}$ auf $V_g(O_i)_{ta}$ zusammenfallen

Die Flächenabgrenzung der Zeichenfläche F bewirkt, daß die Objekte die Grenze von F nicht überschreiten. Damit sind alle Voronoi-Gebiete abgeschlossen. Die Schwerpunkte der so modifizierten Voronoi-Gebiete liegen damit immer innerhalb von F . Dadurch "verlassen" die Objekte $O_i \in O$ niemals die Zeichenfläche F . Die für die Relaxierung notwendigen Flächenmomente der einzelnen Voronoi-Gebiete werden beim hierbei verwendeten graphikhardware-basierten Ansatz durch Bildverarbeitungsverfahren für jede Farbe (und damit für jedes Voronoi-Gebiet) approximativ bestimmt.

Der oben beschriebene Algorithmus verwendet implizit eine konstante Bewertungsfunktion $\phi(x, y)$. Die Bewertungsfunktion kann, wie gesagt, als eine Beschreibung einer angestrebten Objektverteilungsdichte interpretiert werden. Bevor die Erweiterung für nichtkonstante Dichtefunktionen $\phi(x, y)$ beschrieben wird, demonstrieren wir die Berechnungsvorschrift an einfachen Beispielbildern.

In Abbildung 4 wird der Relaxierungsalgorithmus auf eine Menge von 30 Punkten angewendet. Das erzielte Ergebnis gleicht dem rein analytischen Ansatz, der noch in [Hil99] verwendet wurde.

Betrachten wir nun zufällig verteilte Linien unterschiedlicher Länge in einem Teilbereich der zu betrachtenden Ebene (Abbildung 5(a)). Diese werden auf zweierlei Arten relaxiert. Im ersten Fall werden sie nur verschoben (Abbildung 5(b)) im zweiten Fall auch gedreht (Abbildung 5(c)). In beiden Fällen entsprechen die entstehenden Resultate unseren Wünschen in Bezug auf die visuelle Qualität. Interessanterweise scheinen auch die initialen Überlappungen nicht zu stören, im Endergebnis erhält man jedesmal eine gleichmäßig verteilte Linienschar.

In Abbildung 6 und 9 liegt der Voronoi-Iteration einer konstante Dichtefunktion zu Grunde. Eine nahezu beliebige Bewertungsfunktion $\phi(x, y)$ kann auf einfachem Weg in das Rechen-schemata integriert werden: Dazu wird der Wert der Bewertungsfunktion für jeden Bildpunkt

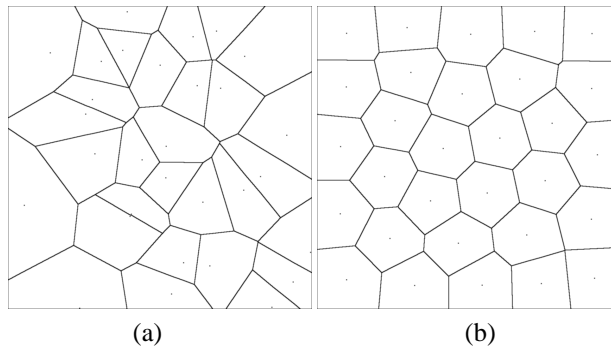


Abbildung 4: Punkt-Objekte und deren Voronoi-Diagramm a) ohne Relaxierung, b) nach 50 Iterationen

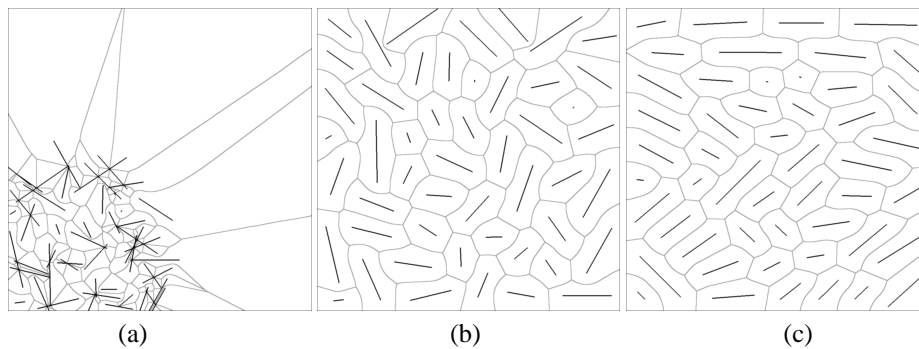


Abbildung 5: Linien-Objekte und deren Voronoi-Diagramme: a) keine Iteration, b) nach 150 Iterationen ohne Drehung, c) nach 150 Iterationen mit Drehung

vorberechnet und in einem zweidimensionalen Feld abgespeichert. Eine spezielle Bitmap könnte ebenfalls verwendet werden, um die Berechnungen nochmals zu beschleunigen. Während der grafischen Schwerpunkt-Bestimmung der Voronoi-Gebiete werden die benötigten Funktionswerte abgefragt und in Form der numerischen Integration von Flächenintegralen ausgewertet. Anschaulich kann man sich die Vorgehensweise als die Verschiebung von Schwerpunkten durch eine ungleichmäßige Massenverteilung auf den Flächen vorstellen.

4 Vorteile der grafischen Berechnung

Die analytische Bestimmung von Voronoi-Gebieten, besonders der von Polygonen ist rechnerisch zeitaufwendig und erfordert außerdem eine komplexe Implementierung durch viele mögliche Sonderfälle. Die bei der analytischen Berechnung zur erreichende Genauigkeit ist zwar beliebig genau, wenn entsprechende Zahlenformate verwendet werden, für viele

Anwendungen ist jedoch eine begrenzte Rechengenauigkeit ausreichend. Die neue Form der Berechnung bietet aber dafür den Vorteil, sowohl Dichtefunktionen als auch beliebige Objekte mit dem Algorithmus behandeln zu können.

Anwendungen für diese Erweiterung der Relaxierung lassen sich viele finden: So siedeln sich Pflanzen beispielsweise bevorzugt an Orten mit ganz speziellen Umgebungsparametern an, was zu ungleichmäßigen Anordnungen führt. Das in [Hil99] beschriebene Illustrationswerkzeug ordnet die Punkte einer Illustration nach wieder anderen Gesichtspunkten an. Andere Anwendungen sind in der Zellbiologie, der Statistik oder der Bildkompression zu finden [DFG99], wo ebenfalls Verteilungen mit unterschiedlichen Charakteristika benötigt werden.

Beispiele für die Optimierung beliebig geformter Objekte sind z.B. Objektanordnungen in Grundrissen von Häusern oder technischen Zeichnungen von Maschinenteilen oder aber die Verteilung von natürlichen Objekten wie Zweigen oder Steinen innerhalb von 3D-Modellen. Es gibt daher viele Problemstellungen, die im Kern auf eine optimale Anordnung von Objekten innerhalb eines eingeschränkten Raumes reduzierbar sind.

Bevor praktische Beispiele vorgestellt werden, soll noch einmal eine grobe Beschreibung des entwickelten Anordnungsverfahrens gegeben werden. Folgende Berechnungsschritte werden bei der hardwaregestützten Relaxierung durchlaufen:

1. Vorberechnung:

- Festlegen von Abmessungen der Zeichenfläche (X,Y,Z)
- Vorbestimmung der Objektapproximation/ Farbkodierung
- Vorberechnung der Dichtefunktion

2. Iterative Berechnung:

- Zeichnen der Objektapproximation
- Berechnen der Zwischenergebnisse
- Verschiebung der Objektkoordinaten
- Auswerten von Abbruchbedingungen

3. Ausgabe der Ergebnisse:

- Prüfung/Weiterverarbeitung/Anwendung

Der vollständige Algorithmus besitzt weitere Unterpunkte, die jedoch für das grundlegende Verständnis unwichtig sind. Die vorliegende Implementierung ist in C/C++ geschrieben und mit 25 KByte Sourcecode sehr kompakt.

Das Ergebnis fällt dann wie in Abbildung 9 dargestellt aus. Als zugrundeliegende Bewertungsfunktion diente hierbei die in Abbildung 8 gezeigte mehrdimensionale Exponentialfunktion. Die hier entwickelte Art der lokalen Optimierung erbringt die gleichen Ergebnisse wie [DFG99]. Der hier vorgestellte Algorithmus ist jedoch einfacher zu implementieren, schneller in der Ausführung und kann nicht nur Punktobjekte, sondern beliebig geformte Objekte optimal anordnen.

Eine modellhafte praktische Anwendung ist in Abbildung 10 dargestellt. Gegeben ist ein Maschinenteil (Stanzteil aus Blech) mit einer Menge von in der Lage festgelegten Bohrungen. Nun sollten, um Material einzusparen oder/und das Bauteil leichter auszuführen, Durchbrüche eingebracht werden. Durch die Anwendung der Relaxierung auf eine gegebene Menge von Durchbrüchen werden diese auf der gegebenen Fläche verteilt.

5 Zusammenfassung und Ausblick

In der vorliegenden Arbeit konnte gezeigt werden, wie mit einfachen Mitteln eine komplexe lokale Optimierungsaufgabe auf der Basis von Voronoi-Diagrammen grafisch/numerisch gelöst werden kann. Das vorgestellte Verfahren ist wesentlich flexibler als die analytische Variante, welche noch in [Hil99] angewendet wurde.

Wird die Voronoi-Relaxierung auf eine Menge von Linien oder Polygonen angewendet, ergeben sich einige spannende Fragen. Beispielsweise ist empirisch zu klären, ob das Verfahren in jedem Fall konvergiert, ein mathematischer Beweis steht selbst für Punktobjekte noch aus [DFG99] und ist daher für allgemeine Objekte nicht zu erwarten.

Ferner sind die statistischen Eigenschaften der resultierenden Verteilungen zu untersuchen. Der in [Hil99] benutzte Ansatz, die statistischen Eigenschaften im Frequenzraum zu charakterisieren, kann sicher nicht auf Objekte unterschiedlicher Ausdehnung angewendet werden. Bestimmte Eigenschaften wie z.B. Drehungsinvarianz durch Fouriertransformation zu untersuchen, ist allerdings eine einfache und elegante Methode. Für die hier behandelten Objekte müssen daher neue Bewertungsverfahren gefunden werden.

Da Linien und Polygone im Gegensatz zu Punkten eine Ausdehnung haben, ist auch zu untersuchen, was bei Überlappungen in der initialen Verteilung geschieht. Schließlich sollten die entstehenden Verteilungen auch ästhetisch dem entsprechen, was man sich unter einer gleichmäßigen zufälligen Verteilung vorstellt. Auch die Verteilung von nichtkonvexen dicht zusammenliegenden Objekten sollte weiter untersucht werden.

Für weiterführende Arbeiten könnten wir uns außerdem vorstellen, die Informationen im Tiefenpuffer zu nutzen. Wenn beispielsweise ein zusätzliches Objekt in eine bestehende Szene eingefügt werden soll, kann dies an der Stelle erfolgen, wo der Z-Puffer auf einer Voronoi-Kante den größten Wert besitzt, also die Kegelfläche am weitesten vom Betrachter entfernt ist. Weiterhin sollten weitere Beispielanwendungen untersucht werden, etwa in der Robotik, bei der Bewegungsplanung und der Bestimmung kürzester Abstände (Spanning-Trees) zu finden. Auch für die Planung von computergenerierten Kamerafahrten sind Voronoi-basierte Algorithmen anwendbar.

Literatur

- [DFG99] Q. Du, V. Faber und M. Gunzburger. *Centroideal Voronoi Tessalations: Applications*. In: *Siam review*, Vol. 41(4), S. 637–676, 1999.

- [DHL⁺98] O. Deussen, P. Hanrahn, B. Lintermann, R. Mech, M. Phar und P. Pusinkiewicz. *Realistic Modelling and Rendering of Plant Ecosystems*. In: Proc. SIGGRAPH 1998, Orlando, 1998.
- [DHOS00] O. Deussen, S. Hiller, K. Van Overfeld und T. Strothotte. *Floating Points: A Method for computing Stippel Drawings*. In: D. Duke S. Coquillart, Hrsg., Proc. Eurographics 2000, Interlaken, S. C41. Blackwell Publishers Ltd, 2000.
- [HCK⁺99] K. Hoff, T. Culver, J. Keyser, M. Lin und D. Manocha. *Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware*. In: Proc. SIGGRAPH 1999, Los Angeles, 1999.
- [Hil99] Stefan Hiller. *Einsatz von relaxierten Punktverteilungen in der Computergrafik*. Diplomarbeit Universität Magdeburg, 1999.
- [NDW93] Jackie Neider, Tom Davis und Mason Woo. *OpenGL Programming Guide*. Addison-Wesley Publishing Company, 1. Auflage, 1993.
- [OBS97] Atsuyuki Okabe, Barry Boots und Kokichi Sugihara. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, 1997. ISBN 0-471-93430.

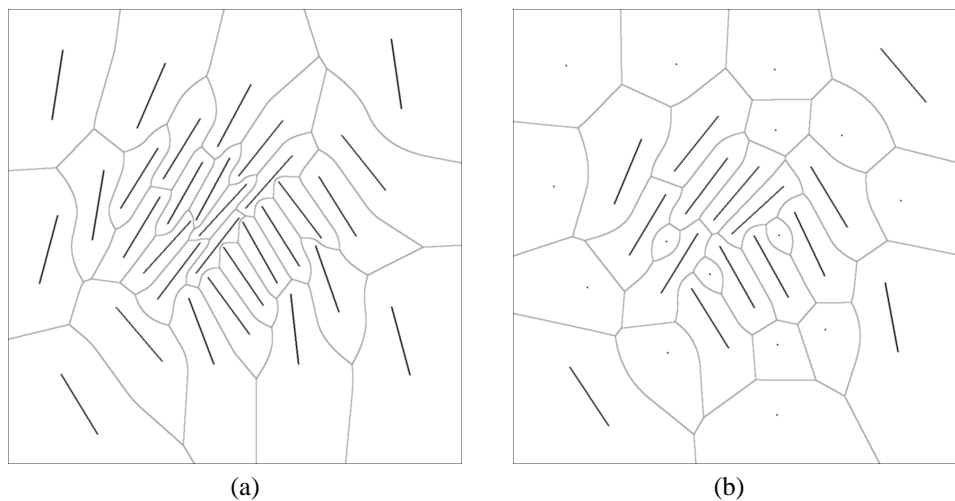
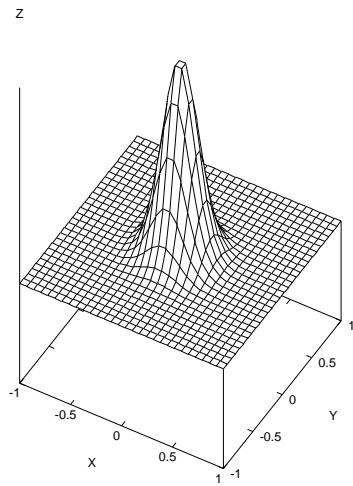
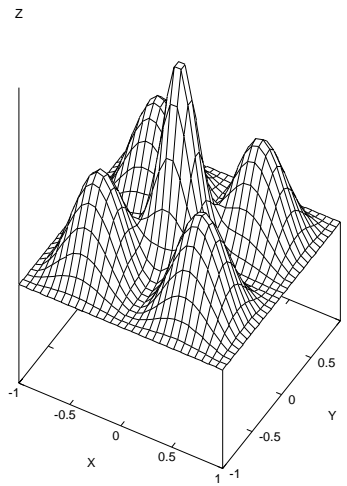


Abbildung 6: Relaxierung von a) Linien und b) Linien mit Punkten mit $\phi(x, y)_0 = e^{-8x^2-8y^2}$, nach jeweils 150 Iterationen

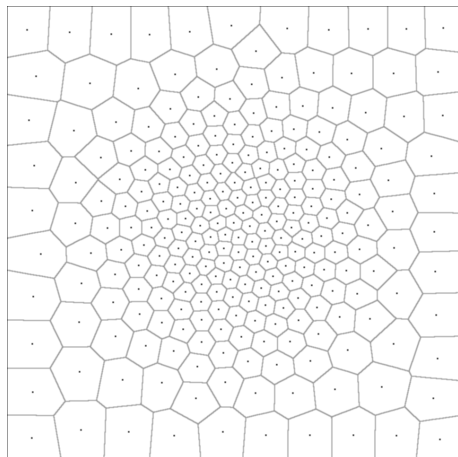


(a)

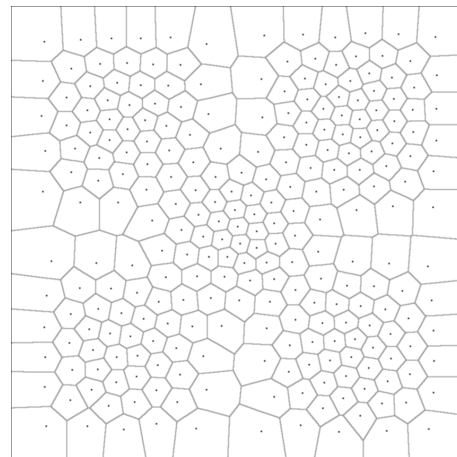


(b)

Abbildung 7: Visualisierung der verwendeten Bewertungsfunktionen: a) $\phi(x, y)_1 = e^{-8x^2 - 8y^2}$, b) $\phi(x, y)_2 = e^{-4x^2 - 4y^2 + 0.5 \sin^2(\pi x) \sin^2(\pi y)}$



(a)



(b)

Abbildung 8: Resultate von a) $\phi(x, y)_1$ und b) $\phi(x, y)_2$ nach der Optimierung einer identischen Ausgangspunktmenge (jeweils 150 Iterationsschritte)

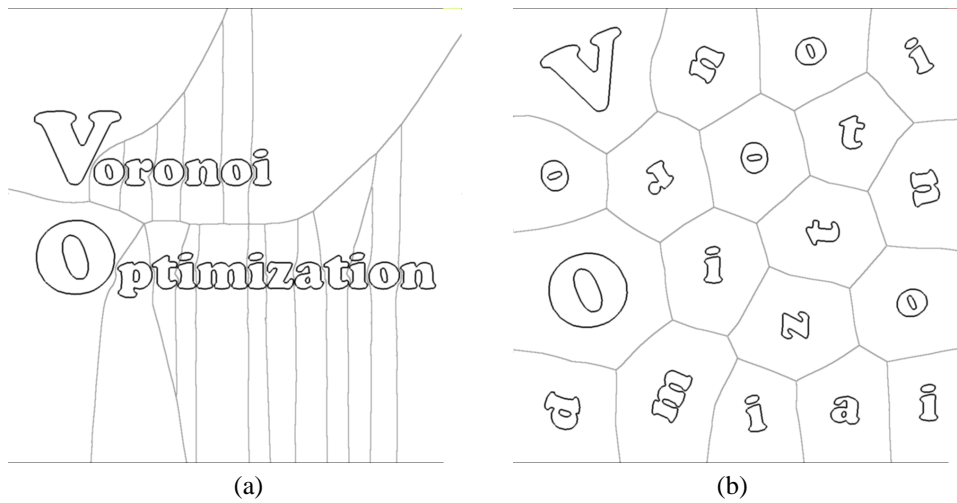


Abbildung 9: Relaxierung von Polygonen: Schriftzug nach a) 0, b) 150 Iterationen ($\phi(x, y)$ konstant)

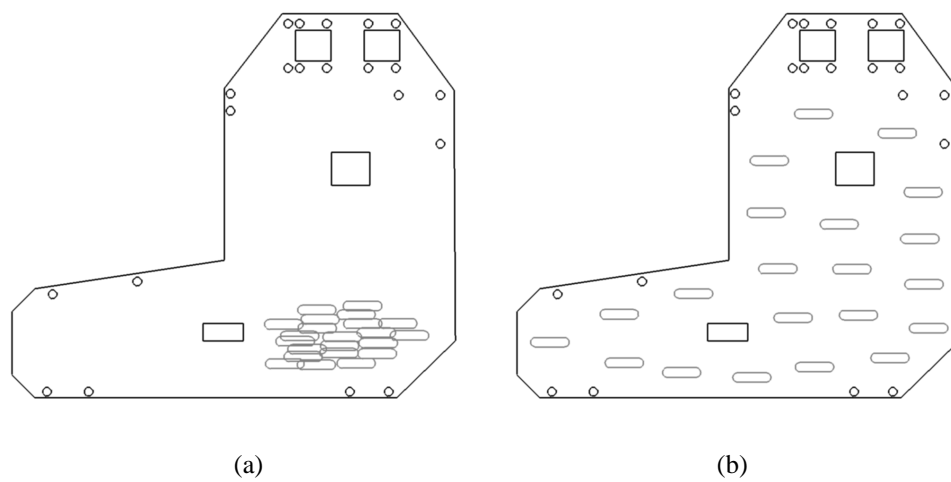


Abbildung 10: Ansicht Stanzteil nach a) 0 und b) 80 Iterationsschritten ($\phi(x, y)$ konstant).