A B S T R A C T S

zum vierten Workshop über Computational Geometry

am 24. und 25. März 1988 an der

Universität Würzburg

CG'88

Prof. Dr. Hartmut Noltemeier
Dipl.-Math. Hugo Heusinger
Lehrstuhl für Informatik I
Am Hubland
D-8700 Würzburg

Das Zwillings-Prinzip zur Speicherplatzoptimierung geometrischer Zugriffsstrukturen

Andreas Hutflesz, Hans-Werner Six, Peter Widmayer

Zusammenfassung

Zugriffsstrukturen für Punkte, die räumliche Suche unterstützen, leiden oft an einer unerwünscht niedrigen Speicherplatzausnutzung. Wir zeigen, wie eine gegebene Punktmenge zwischen zwei Gridfiles so verteilt werden kann, daß die Speicherplatzausnutzung optimal ist. Das optimale Zwillings-Gridfile kann praktisch so schnell wie ein übliches Gridfile aufgebaut werden, d.h., die Speicherplatzoptimalität ist fast ohne zusätzliche Kosten erhältlich. Außerdem schlagen wir einen effizienten dynamischen Mechanismus vor, bei dem Einfügungen und Entfernungen die Umverteilung der Punkte zwischen den beiden Gridfiles auslösen. Wir vergleichen die Eigenschaften des optimalen statischen Zwillings-Gridfiles, des dynamischen Zwillings-Gridfiles, und des üblichen Gridfiles. Die Speicherplatzausnutzung des dynamischen Zwillings-Gridfiles ist fast optimal, abhängig vom Aufwand bei der Umverteilung von Punkten. Typische Bereichsanfragen - die wichtigsten räumlichen Zugriffsoperationen - können mindestens so schnell wie im üblichen Gridfile beantwortet werden.

ETH Zürich
Institut für Informatik
Dr. H. Hinterberger

Graphische Information zur Mustererkennung in kartesischen Produkte-Räumen

Das Suchen (Extrahieren und Visualisieren) von verständlichen Strukturen in multivariaten Datensätzen ist ein noch wenig erforschtes Gebiet und bildet eine wichtige Aktivität in der *erkundenden Datenanalyse*, eine sich vorwiegend auf Daten-Beschreibung stützende Methodik. Beschreibungen anhand graphischer Darstellungen erlauben der erkundenden Datenanalyse die Mustererkennungsfähigkeit der menschlichen visuellen Wahrnehmung auszunützen. Deshalb manipulieren die meisten traditionellen Methoden linear auf zwei- oder dreidimensionale Räume projizierte Datenwolken, d.h. zu Punkten im Euklidschen Raum reduzierte multivariate Daten. Die Verwendung von convex hull Algorithmen zur Clusteranalyse illustriert die dadurch entstandene Nachbarschaft zur Computational Geometry. Weil auf Projektionen basierende Methoden jedoch Schwierigkeiten bereiten wenn nicht-lineare Strukturen - typisch für wissenschaftliche und statistische Daten - vorhanden sind, muss weiterhin nach neuen Methoden gesucht werden. Diese neuen Methoden sollten ebenfalls mit den enormen Grössen heutiger Datensätze fertig werden.

Wir erläutern eine Methode, die auf der *Gitterdatei* (grid file) fusst. Diese Datei erlaubt schnelles Extrahieren von Informationen über Daten, ohne dass auf die Daten selbst zugegriffen werden muss. Konzeptionell basiert die Methode auf dem neuartigen Ansatz, dass das Verzeichnis (directory) einer Datei als komprimierte Darstellung eines gewissen Informationsinhaltes der verwalteten Daten aufgefasst werden kann. Diese Information im Verzeichnis kann selbst verwendet werden und dient nicht nur als versteckte zusätzliche Hilfsorganisation. Dieser Ansatz wird praktisch implementierbar, wenn das Verzeichnis der Gitterdatei so konzipiert ist, dass die Information über Regionen in k-dimensionalen Euklidschen Räumen leicht zugänglich ist. Wir stellen ein solches Verzeichnis in der Form des *Regionenverzeichnisses* (region directory) vor.

Der Gitterdatei wird oft vorgeworfen, dass ihre Anpassungsfähigkeit auf Kosten eines superlinearen Wachstums geht. Mit dem Regionenverzeichnis haben wir jedoch eine Datenstruktur gefunden, die den Gitterarray (grid array) der Gitterdatei so implementiert, dass sein Wachstum *linear* von der Anzahl der gespeicherten Daten abhängt, ungeachtet der (statistischen) Verteilung der Daten.

Datenverwaltung und Datenvisualisierung - beide ausgiebig erforschte Gebiete - werden selten gemeinsam betrachtet. Dies überrascht nicht, scheint doch ein gemeinsamer Nenner zu fehlen. Am Beispiel des Regionenverzeichnisses zeigen wir indes, dass ein elementares Mass, namentlich die *Datendichte*, der Datenverwaltung und der Datenvisualisierung dienen kann, indem sie die Basis für eine gemeinsame Struktur liefert. Das Regionenverzeichnis der Gitterdatei ist eine solche (Daten-) Struktur, die - ohne zusätzlichen Aufwand - graphische Information geeignet zur Visualisierung von Strukturen multivariater Datensätze zur Verfügung stellt, wobei Regionen variabler Dichte der verwalteten Daten angenähert angezeigt werden. Wir stellen mehrere Methoden zur Darstellung Regionen variabler Dichte in k-dimensionalen kartesischen Produkteräumen vor und zeigen damit, dass die Gitterdatei die Möglichkeit zur schnellen Ableitung von Abstraktionen aus grossen Datenmengen bietet.

Wir beschreiben erste Resultate einer Implementation der Gitterdatei, basierend auf dem Regionenverzeichnis, bezüglich graphischer Darstellungsmöglichkeiten zur Visualisierung von Dichten multivariater Daten.

Trennbare und rechteckige Clusterungen

H. Heusinger H. Noltemeier
Universität Würzburg

Zusammenfassung

Zunächst führen wir das Problem der trennbaren Clusterung ein: Gegeben ist eine endliche Menge S ($n = |S|$) von Punkten in der euklidschen Ebene. $A \subset S$ heißt trennbares Cluster, wenn es eine Gerade gibt, die A von $S \setminus A$ trennt. Eine Zerlegung der Menge S in C paarweise disjunkte trennbare Cluster S_i ($|S_i| = k_i$) heißt C-t-Clusterung. Dabei sind C und k_i ($1 \leq i \leq C$) vorher fixierte natürliche Zahlen.
Ist nun $f : (S_1, S_2, \ldots, S_C) \longmapsto r \in \mathbb{R}$ ($S_i \subset S$) eine Clusterfunktion, so bestimmt unser Algorithmus in

$$O(p_C \; [\; C \; (n^{3/2}\log^2 n + n^{3/2} U_f(n) \;) + P_f(n) \;] \;)$$

Schritten eine bezüglich f optimale C-t-Clusterung, wobei $P_f(n)$ (bzw. $U_f(n)$) die Zeit für die Berechnung (bzw. den update) der Funktion f ist, und p_C die Anzahl der nicht durch zyklische Vertauschungen ineinander überführbaren Permutationen der k_i ist. Wir geben auch ein Beispiel an, bei dem die Anzahl der C-t-Clusterungen exponentiell in n ist.

Danach betrachten wir das folgende Problem:
Bestimme alle Mengen von C (minimalen) achsenparallelen Rechtecken R_i, mit:

$$|R_i \cap S| = k_i \text{ und } R_i \cap R_j \cap S = \emptyset,$$

wobei wieder C und k_i vorgegebene natürliche Zahlen sind.
Die Idee, wie das obige Problem in $O(n^{C-1}\log n)$ Schritten gelöst werden kann, wird dann vorgestellt, und es wird ein Beispiel angegeben, bei dem es $\Omega(n^{C-1})$ solcher Mengen von Rechtecke gibt.

Workshop "Computational Geometry", Würzburg 24./25.3.88

H. Bieri, W. Nef (Bern)

Schneiden, Vereinigen, Subtrahieren von Polyedern (Zusammenfassung)

Ausgangspunkt für die von uns in den letzten Jahren entwickelte Polyeder-Theorie war die "ärgerliche" Tatsache, dass die Menge der wie üblich definierten Polyeder gegenüber den elementaren Operationen "Komplement" und "Differenz" nicht abgeschlossen ist. Unser Ziel war die Behebung dieses Mangels auf mathematisch strenge und in der Praxis der algorithmischen Geometrie brauchbare Weise. Dazu gibt es grundsätzlich zwei Wege: Anpassung der Operationen ("Regularisierung") oder des Polyederbegriffs. Wir haben den zweiten Weg gewählt, dabei allerdings gefunden, dass der erste sozusagen von selber darin enthalten ist.

Wir definieren ein Polyeder $P \subset \mathbb{R}^d$ als eine Menge, die aus endlich vielen Halbräumen mittels der Operationen "Durchschnitt" und "Komplement" erzeugt werden kann. (Damit ist die Menge der Polyeder auch gegenüber Vereinigungs- und Differenzbildung abgeschlossen). Polyeder im eben definierten Sinn brauchen weder abgeschlossen noch beschränkt zu sein. Die Polyeder im üblichen Sinn sind spezielle Polyeder nach unserer Definition.

Die "lokalen Eigenschaften" eines Polyeders P im Punkt $x \in \mathbb{R}^d$ werden durch die Pyramide $P^x = x + \mathbb{R}^+(P \cap U(x) - x)$ ausgedrückt (wo $U(x)$ eine genügend kleine Umgebung von x ist). Die Seiten von P werden als die Aequivalenzklassen der Relation $x \sim y \Longleftrightarrow P^x = P^y$ definiert.

Als Datenstruktur für die Beschreibung eines Polyeders P wählen wir die Menge der in geeigneter Form digitalisierten, den Seiten S von P zugeordneten Pyramiden $P^S := P^x$ (für $x \in S$). Es ist allerdings nicht trivial, dass P durch diese Pyramiden eindeutig bestimmt ist. Es sei deshalb erwähnt, dass sich aus dieser Struktur die affinen Hüllen der Seiten und ihre Inzidenzmatrix gewinnen lassen, womit der Zugang zu vertrauteren Strukturen (insb. "Boundary representation") eröffnet ist.

Da die Bildung des Komplements sehr einfach und wenig aufwendig ist, werden wir schwergewichtig einen Algorithmus für den Durchschnitt beschreiben. Er ist in dem Sinn rekursiv, dass der d-dim Fall auf den (d-1)-dim zurückgeführt wird (der 0-dim Fall ist trivial).

Es zeigt sich nebenbei, dass auch die Operationen "abgeschlossene Hülle" und "(relativ) Inneres" auf einfache Weise realisierbar sind, womit auch die "regularisierten" Operationen erfasst werden.

USING GALE TRANSFORMS IN COMPUTATIONAL GEOMETRY

by

Franz Aurenhammer[1)]

Abstract

Let P denote a set of $n \geq d+1$ points in d-space R^d. A Gale transform of P assigns to each point in P a vector in space R^{n-d-1} such that the resulting n-tuple of vectors reflects all affinely invariant properties of P. First utilized by Gale in the 1950s, Gale transforms have been recognized as a powerful tool in combinatorial geometry.

This paper introduces Gale transforms to computational geometry. It offers a direct algorithm for their construction and sketches applications to convex hull and visibility problems. An application to scene analysis is worked out in some more detail.

1) Institutes for Information Processing,
Technical University of Graz and Austrian Computer Society,
Schiesstattgasse 4a, A-8010 Graz, Austria.

Glättungsalgorithmen und ihre
Bedeutung für CAD - Systeme

Prof. Dr. Hans Hagen
Universität Kaiserslautern

Computer Aided Geometric Design ist aus dem Bedarf für Freiformkurven- und Freiformflächenkonzeptionen in der CAD/CAM - Technologie entstanden und hat sich zu einem der Hauptforschungsschwerpunkte der Informatik mit direkten Anwendungen in den Ingenieurwissenschaften entwickelt.
Im Rahmen dieses Vortrags werden zunächst Algorithmen vorgestellt, die Flächen erzeugen, die dem Standardsglattheitskriterium minimaler Biegungsenergie genügen. Danach wird auf Methoden eingegangen, die eine Qualitätsanalyse der Flächen ermöglichen. Neben dem Reflektionslinienverfahren wird auch die Isophotenmethode besprochen.

Raytracing von Freiformflächen mit dem Design-Patch

von
Alfred Schmitt
Universität Karlsruhe (TH)

Kurzfassung:

Das für die Erzeugung fotorealistischer Abbildungen von 3-dimensionalen Szenen und Objekten am besten geeignete Verfahren ist die Strahlverfolgungs-Methode (Raytracing). Mit keinem anderen Verfahren lassen sich neben schattierten Objekten mit echten Schatten auch Spiegel, Glas, Wasser und sonstige spezielle Materialien problemlos ins Bild setzen.

In der ursprünglich gegebenen (u,v)-Parameterdarstellung sind Freiformflächen jedoch aus verschiedenen Gründen für das Raytracing schlecht geeignet. Wir verfolgen daher in Karlsruhe das Ziel, beliebig gegebene Freiformflächen zunächst durch eine größere Zahl sogenannter Design-Patches zu approximieren und dadurch die Bilderzeugung zu vereinfachen, zu systematisieren und auch zu beschleunigen.

Ein einzelnes Design-Patch ist eindeutig definiert durch
- 3 Eckpunkte P1, P2, P3;
- 3 Normalvektoren N1, N2, N3 in den Eckpunkten;
- 6 Randkurventangenten in den Eckpunkten
  Tij ($i \neq j$, i, j $\in \{1,2,3\}$).

Die genaue Gestalt der durch diese wenigen Bestimmungestücke definierten 3-Eck-Freiformflächen wird durch einen in bestimmter Weise definierten Zerlegungsalgorithmus definiert, der eine beliebig feine 3-Ecks-Facettierung der Flächen liefert.

Wie die Bezeichnung bereits ausdrückt, ist das Design-Patch insbesondere geeignet, um auch die graphische Entwurfsarbeit entscheidend zu vereinfachen, da man nur Stützstellen, Flächennormale und Kurventangenten bestimmen muß, um eine dichte und glatte Fläche zu erstellen.

Durch die Implementierung des Design-Patches als elementares geometrisches Objekt in der Karlsruher Raytracing-Software *VERA* wird es zukünftig möglich sein, auch komplexere Szenen mit sehr vielen Freiformflächen der unterschiedlichsten Art in fotorealistische Bilder umzusetzen. Voraussetzung dafür ist allerdings eine Vorverarbeitungssoftware, die die gängigen Freiform-Flächentypen jeweils durch eine hinreichende Zahl miteinander verketteter Design-Patches approximiert.

Die Karlsruhe Metrik

Rolf Klein
Institut für Informatik
Universität Freiburg
Rheinstr. 10-12
7800 Freiburg

Viele Straßen in Karlsruhe laufen entweder fächerförmig auf das zentral gelegene Schloß zu oder sind Teile von Kreisbögen um das Schloß herum. Wir zeigen, daß Voronoi Diagramme für n Punkte in dieser Metrik in optimaler Zeit O(n logn) berechnet werden können. Zu diesem Zweck wird ein allgemeiner Satz bewiesen, der besagt, daß die Bisektoren geeignet separierter Punktmengen keine Schleifen enthalten, wenn, neben anderen Eigenschaften, die *Kreise* in der zugrundeliegenden Metrik keine Löcher aufweisen. In diesem Fall kann das Voronoi Diagramm mit einem Divide&Conquer-Verfahren in O(n logn) Schritten bestimmt werden. Dieser Satz gilt nicht nur für die Karlsruhe Metrik, sondern für eine ganze Klasse von Metriken, wie z. B. alle Metriken in der Ebene, die durch Zusammensetzen der euklidischen Metrik und der Manhattan Metrik in Teilen der Ebene entstehen.

Kollisionsvermeidung für nichtstarre Objekte

Stephan Abramowski – Universität Karlsruhe

Bei der Kollisionsvermeidung für nichtstarre Objekte geht es darum, ein gegebenes dreidimensionales Objekt zwischen einer Menge von Hindernissen (ebenfalls dreidimensionale Objekte) von einer Startposition in eine Zielposition zu überführen, ohne daß es dabei zu Kollisionen zwischen zu bewegendem Objekt und Hindernissen kommt. Dabei gehen wir davon aus, daß das zu bewegende Objekt aus einer Menge gekoppelter starrer Objekte besteht.

Ein gängiger Ansatz zur Lösung des Bewegungsplanungsproblems ist die Verwendung des "Konfigurationsraums". Dabei wird die Lage des zu bewegenden Objekts durch eine Folge von Parametern beschrieben, z.Bsp. die Folge der Winkel, die die aufeinanderfolgenden Glieder eines Roboterarms einschliessen. Ein Arm mit d Gliedern in der Ebene läßt sich dann durch ein d-Tupel von Winkeln beschreiben, der Konfigurationsraum wäre in diesem Fall die Menge aller d-Tupel. Ein Hindernis geht dabei über in die Menge aller Konfigurationen (d-Tupel), für die der Arm in der entsprechenden Lage mit dem Hindernis kollidiert.

Für das Problem in dieser allgemeinen Formulierung wurde in [SS83] ein Algorithmus vorgestellt. Die einzuhaltenden Bedingungen werden über Polynome formuliert. Die von diesen Polynomen induzierte Zerlegung des Konfigurationsraums wird explizit berechnet. Man erhält einen Graphen, dessen Knoten zulässigen Konfigurationen entsprechen; Kanten bestehen zwischen "elementar" ineinander überführbaren Konfigurationen. Zwei Konfigurationen sind ineinander überführbar genau dann, wenn sie in der selben Zusammenhangskomponente des Graphen liegen. Der Graph hat $O((2k)^{3^{d+1}} \cdot n^{2^d})$ Kanten, wenn man n Polynome in d Variablen und Grad (in jeder Variablen) kleiner gleich k hat. Das Verfahren scheidet in der Praxis wegen seiner hohen Komplexität aus.

Wir untersuchen das Problem in einer idealisierten Version: die gekoppelten Objekte sind Kugeln, die Kopplung zwischen den Kugeln wird durch Vorgabe des Abstandes zwischen einigen Kugelmittelpunkten erreicht. Auch die Hindernisse sind Kugeln. Dadurch gelangt man zu wesentlich einfacheren Zerlegungen des Konfigurationsraums.

Literatur

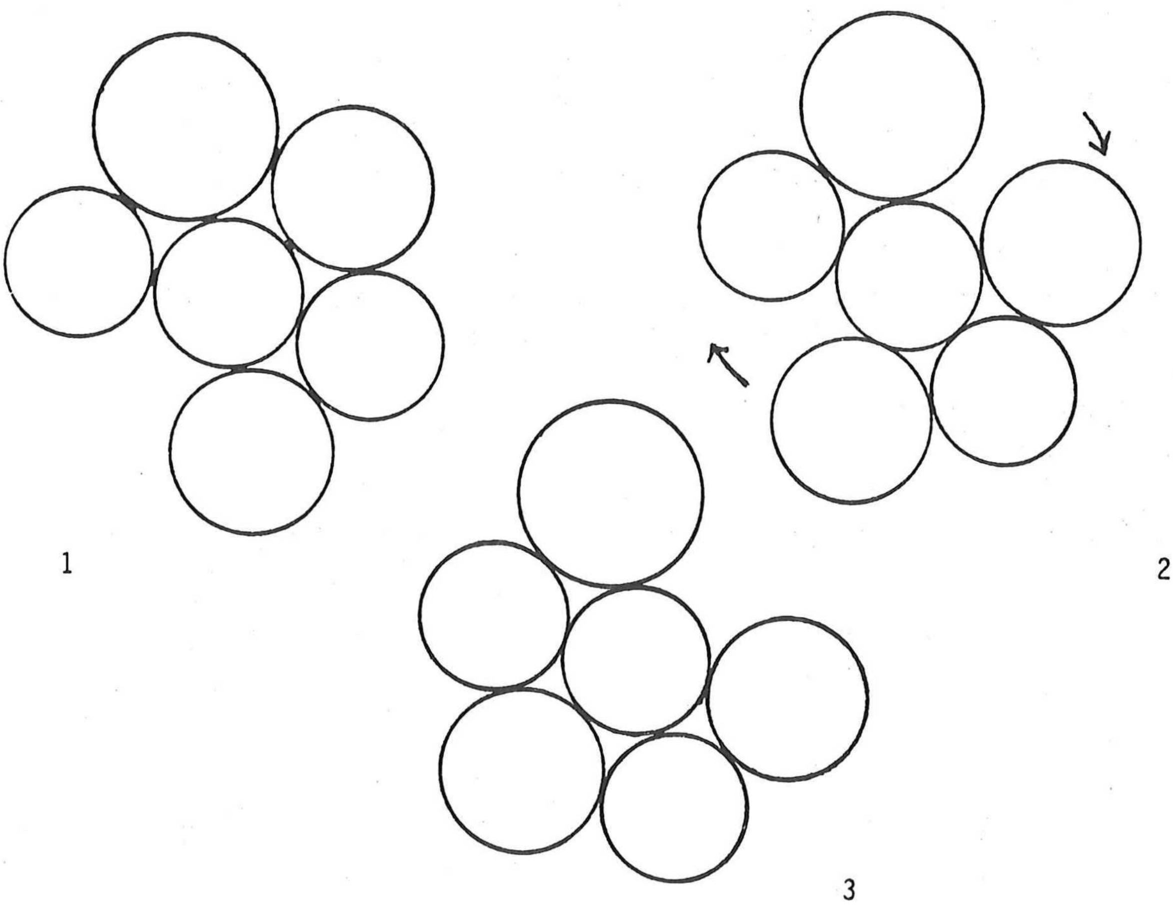
[SS83] Jacob T. Schwartz and Micha Sharir. On the piano movers problem. ii. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, (4):298–351, 1983.

On-line Kollisionsbestimmung

Heinrich Müller, Universität Karlsruhe

Gegeben sei eine Menge von Kreisscheiben in der Ebene (analog geht es mit Kugeln im Raum), die sich nicht überlappen, sich aber gegenseitig so berühren, daß die Konfiguration in sich unbeweglich ist, wenn kein Kontakt aufgelöst wird. Das Auflösen eines Kontaktes induziert eine Konfiguration, deren Deformationsmöglichkeiten eindeutig sind, vgl. Bild. Findet während der Bewegung eine Kollision zwischen Kreisen statt, ist die Konfiguration wieder unbeweglich. Es sollen nun eine Folge unbeweglicher Konfigurationen berechnet werden, die sich aus einer Folge von Kontaktauflösungen und Kollisionen ergibt. Dieses Aufgabenstellung tritt in der Bewegungsplanung von Robotorarmen auf, wenn Selbstkollisionen des Armes zu berücksichtigen sind. Wir geben eine linearisierte Näherungslösung dieses Problems an, die auf der Sweeptechnik und dem Lösen einer Folge von linearen Gleichungssystemen basiert. Die Linearisierung besteht darin, die Kreise durch n-Ecke zu ersetzen.

Das oben geschilderte Problem ist eine relativ komplexe Version der On-Line Kollisionsbestimmung. Ähnliche Probleme treten bei der Simulation von Fahrzeugen oder Flugkörpern auf, die sich interaktiv gesteuert oder autonom in einer Umwelt bewegen, oder bei der Simulation von Partikelsystemen in der Physik. Auch diese Probleme lassen sich durch Sweep effizient lösen.



Generierung von adaptiven Gittern für die FEM-Simulation unter Verwendung der Hexblock-Datenstruktur

Paolo Conti
Institut für Integrierte Systeme
ETH Zürich

Bei der 3-dimensionalen Simulation von Halbleiterbauelementen mit der Methode der finiten Elemente (FEM) ist die Gittergenerierung eines der grundlegenden Probleme. In gewissen Gebieten des Bauelementes ist ein sehr feines Gitter nötig, während in anderen Gebieten ein relativ grobes Gitter genügt. Langgezogene Elemente oder solche mit sehr stumpfen Winkeln fuehren zu schlecht konditionierten Gleichungssystemen und somit zu Stabilitätsproblemen. Problematisch ist daher die Gittergenerierung besonders am Übergang zwischen grob und fein tessellierten Regionen.

Eine wesentliche Rolle bei der Implementierung eines solchen Gittergenerators spielen die verwendeten Datenstrukturen. Die Anzahl benötigter Gitterpunkte und Elemente kann nicht a priori festgelegt werden; neue Punkte müssen nach Bedarf in gewissen Regionen eingefügt und ueberflüssige Punkte wieder aus zu fein tessellierten Regionen entfernt werden. Der Gittergenerator soll auch verschiedene Elementtypen erzeugen können, in unserem Fall Tetraeder, Viereckspyramiden, Prismen und Quader. Je nachdem, wie die Gleichungen gelöst werden, werden auch verschiedene Inzidenzinformationen benötigt. Die verwendete Datenstruktur soll daher insbesondere möglichst alle Nachbarschaftsinformationen in konstanter Zeit zur Verfügung stellen können.

Fuer die Generierung von Gittern verwende ich die Hexblock-Datenstruktur (C. E. Buckley). Es ist dies eine Grenzelementdarstellung, mit der sich beliebige ebenbegrenzte Tessellierungen darstellen lassen. Hexblocks sind eine 3d-Erweiterung der quadedges von Guibas und Stolfi, mit denen sich planare Graphen darstellen lassen. Ähnlich wie mit quad-edges lassen sich mit Hexblocks beliebige Tessellierungen mit nur einer konstruktiven Primitive und nur einem topologischen Operator konstruieren. Die Topologie wird separat von der Geometrie gespeichert und beliebige Inzidenzinformationen können in konstanter Zeit gewonnen werden.

Fuer die Generierung der Gitter gehe ich folgendermassen vor: zuerst wird ein grobes Gitter generiert. Anschliessend werden nach Bedarf Elemente verfeinert, indem alle Kanten halbiert werden und aus einem Tetraeder oder Kubus 8 neue Tetraeder oder Kuben entstehen. Im nächsten Schritt werden die Elemente am Rand der verfeinerten Gebiete so aufgeteilt, dass kein Knoten in der Mitte einer Kante endet. In der Regel wird dieser Schritt zu schlecht geformten Elementen fuehren. Zur Aufbesserung der Qualität dieser Randelemente kann beispielsweise eine Gauss'sche Glättung vorgenommen werden. Eckpunkte der schlechten Elemente werden dabei so verschoben, dass die Gesamtqualität der in diesen Eckpunkten inzidenten Elemente aufgebessert wird.

Im Vortrag werde ich erst die Hexblock-Datenstruktur vorstellen, und anschliessend meine Erfahrungen bei deren Verwendung zur Implementierung eines automatischen Gittergenerators zusammenfassen.

"Support–Set" Berechnung konvexer Hüllen mittels der Hexblock–Strukturen

Dr. Ch. Buckley
Institut für Integrierte Systeme
ETH Zürich

29. Februar 1988

Die besten Zeit– und Speicheraufwand–Ergebnisse bei der Berechnung von konvexen Hüllen höherer Dimension sind zur Zeit nur durch die Verwendung von Algorithmen, die gewissen Bedingungen auf ihren Eingaben genügen, erreichbar. Entweder:

- müssen die Punkte im voraus gesammelt werden, um als eine komplette Eingabe-Menge an den Algorithmus geliefert zu werden,
- oder es dürfen höchstens Einzelpunkte sequentiell eingeführt werden.

Bessere Lösungen sind schon bekannt in den Fällen der Dimensionen 2 und 3, wo man das leicht zu parallelisierende "divide–and–conquer"-Verfahren verwenden kann. Dabei erweist sich der Gebrauch einer Struktur, die die Randtopologie der Zwischen-Konvexen–Hüllen genau beschreibt, als unerlässlich.

Die vor kurzem von mir entwickelte Hexblock–Struktur erfüllt dieses Bedürfnis in vier Dimensionen, d. h. sie leistet die Beschreibung von vollständigen 3er–Kompexen. (Darum ist sie auch brauchbar, um räumliche Triangulationen zu beschreiben, wofur sie eigentlich entwickelt wurde.) Von den Eigenschaften dieser Struktur hörten Sie im Vortrag von Herrn Conti.

Dieser Vortrag betrifft die Verwendung der Hexblock–Struktur um genau so einen divide-and-conquer–Algorithmus zu unterstützen. Interessant ist, dass die Motivation dieses Algorithmus völlig im dualen Gauss'sche–Kugel–Raum begründet ist, wobei die Abbildung zwischen diesem und dem Punkt–Koordinaten–Raum durch die Berechnung von Support–Sets erfolgt. Interessant ist auch, das trotzdem alle Berechnungen aus Genauigkeitsgründen im primalen Raum durchgeführt werden müssen.

Ein Zeit– und Speicher–optimaler Algorithmus ist in Lisp implementiert worden. Dabei mussten lexikografische Perturbationen verwendet werden. Um die entsprechende Programmierungs–Komplexität zu minimieren, wurden alle numerischen Teile des Programms automatisch durch Macsyma generiert.

Über dieses Resultat, und auch über noch offene Fragen wird am Workshop berichtet.

Ein - Schicht - Verdrahtung in der Ebene

Shaodi Gao*, Marc Jerrum**, Michael Kaufmann*,
Kurt Mehlhorn*, Wolfgang Rülling*, Christoph Storb*

* FB10 - Informatik, Universität des Saarlandes, D 6600 Saarbrücken
** Dep. of Comp. Science, University of Edinburgh, Scotland.

Eine Eingabe für das ***Ein-Schicht-Verdrahtungs-Problem ESVP*** besteht aus einer Verdrahtungsskizze, d.h. einer Menge von ***Hindernissen*** und ***Pfaden*** (Drähten) in der Ebene. Pfade sind stetige Kurven, die sich nicht schneiden und von Hindernissen disjunkt sind.

Als Lösung suchen wir eine Menge von Pfaden, homotop zu den Eingabepfaden, so daß für alle Pfade p_i gilt: $U(p_i) = \{x \mid dist(x, p_i) < 1/2\}$ ist eine einfach zusammenhängende Teilmenge der Ebene und es gilt: $U(p_i) \cap U(p_j) = \emptyset$ für $i \neq j$ und $U(p_i) \cap U(o) = \emptyset$ für alle i und alle Hindernisse o.

Kriterium für die Lösbarkeit ist die ***Schnittbedingung***. Ein ***Schnitt*** C ist die geradlinige Verbindung zweier ***Features*** (Pfadendpunkt oder Hindernis) oder ein in einem Feature startender unendlicher Strahl. Seine ***Dichte*** $dens$ ist die Anzahl aller Pfade, die ihn bei gegebener Homotopie der Pfade schneiden müssen; seine ***Kapazität*** cap ist seine Euklidsche Länge. Die Schnittbedingung lautet: $dens(C) + 1 \leq cap(C)$.

Hauptaussage der Arbeit ist folgender Satz:

a.) Ein ESVP hat eine Lösung, genau dann wenn die Schnittbedingung für alle Schnitte erfüllt ist.

b.) Die Schnittbedingung kann in Zeit $O(n^2 \cdot \log n)$ getestet werden.

c.) Eine Lösung (, falls sie existiert,) kann in Zeit $O(n^3 \cdot \log n)$ und Platz $O(n^3)$ konstruiert werden.

Hierbei ist n die Größe der Eingabe.

Die Lösungsmethode basiert auf einer Verallgemeinerung des ***Plane - Sweeps*** auf den ***universellen Überlagerungsraum*** $\mathcal{C}$ der Ebene bzgl. eines Punktes s:
$\mathcal{C} = \{(x, p) \mid x \in \mathbf{R}^2$ und p ist eine Homotopieklasse von Pfaden von s nach $x\}$.

Durch einen Sweep konstruieren wir für jeden Draht den kürzesten Weg, der alle durch Hindernisse und andere Drähte erzeugten verbotenen Gebiete vermeidet.

Plane-Sweep Algorithmen lösen Nachbarschaftsprobleme elegant

Klaus H. Hinrichs

Plane-sweep ist ein einfaches algorithmisches Prinzip zur Lösung geometrischer Probleme in der Ebene, das bei der Suche nach einem neuen Algorithmus immer in Betracht gezogen werden sollte. Plane-sweep kann auf überraschend viele Probleme angewendet werden. Verglichen mit dem algorithmischen Prinzip des divide-et-impera löst plane-sweep etwa die gleiche Klasse von Problemen mit dem gleichen Zeitaufwand.

Die beiden bekanntesten Algorithmen, die das "Paar mit minimaler Distanz" Problem und das "Alle nächsten Nachbarn" Problem in optimaler Zeit lösen, basieren auf dem Prinzip des divide-et-impera. Wir zeigen, dass beide Probleme auch durch plane-sweep Algorithmen in optimaler Zeit gelöst werden können. Diese Algorithmen sind viel leichter zu verstehen und zu implementieren.

Wir haben diese beiden plane-sweep Algorithmen unter besonderer Berücksichtigung von Robustheit und auftretenden Sonderfällen implementiert.

Philips NSM 8082

Vergleich interner und externer Plane-Sweep und Divide-And-Conquer Algorithmen am Beispiel des Points-In-Regions Problems

Gabriele Blankenagel
Ralf Hartmut Güting

Fachbereich Informatik, Universität Dortmund,
D-4600 Dortmund 50, West Germany

Bei theoretischen Untersuchungen der algorithmischen Komplexität von elementaren geometrischen Problemen geht man typischerweise davon aus, daß die betrachteten Algorithmen vollständig intern ablaufen. Diese Annahme ist in vielen praktischen Fällen (VLSI-Design, geometrische Datenbanksysteme) unrealistisch und stellt ein Haupthindernis bei der praktischen Umsetzung der in der "Computational Geometry" gewonnenen Resultate dar. Aus diesem Grund ist es notwendig, zu den geometrischen Algorithmen systematisch effiziente Varianten zu entwickeln, die weniger als linearen Speicherplatzbedarf im Vergleich zur Größe der Eingabe benötigen.

In diesem Zusammenhang betrachten wir das "Points-In-Regions" Problem, das als geometrisches Mengenproblem in geometrischen Datenbanksystemen auftritt - es entspricht dem "Inside"-Join der "Geo-Relationalen Algebra", einem Modell zur Behandlung geometrischer Daten. Das "Points-In-Regions" Problem besteht darin, zu einer Menge von Punkten und einer Menge von disjunkten Gebieten in der Ebene alle aus einem Punkt und einem Gebiet bestehenden Paare zu ermitteln, bei denen der Punkt innerhalb des Gebietes liegt.

Wir beschreiben effiziente Algorithmen, die das Problem mittels der Plane-Sweep und der Divide-And-Conquer Technik in $O(n (\log n) + t)$ bzw. $O(n (\log^2 n) + t)$ Zeit und mit $O(n)$ Speicherplatzbedarf lösen, wobei n die Anzahl der Punkte und Gebiete und t die Anzahl der gefundenen Paare angibt. Neben dem jeweiligen internen Algorithmus geben wir bezüglich beider Techniken jeweils eine Version an, die intern abläuft, aber wesentlich weniger als linearen Speicherplatz benötigt. Außerdem stellen wir jeweils eine externe Version vor, die nur einen konstanten, von der Größe der Eingabe unabhängigen internen Speicherplatzbedarf erfordert. Bei einem Vergleich der mittels Plane-Sweep und Divide-And-Conquer erreichten Lösungen wird deutlich, daß man im externen Fall ein günstigeres Verhalten der Divide-And-Conquer Lösung erwarten kann, selbst wenn die entsprechende interne asymptotische Komplexität im worst-case höher liegt.

Literatur:

Blankenagel, G. und R. H. Güting,
Internal and External Algorithms for the Points-in-Regions Problem - the INSIDE Join of Geo-Relational Algebra. Universität Dortmund, Abteilung Informatik, Report 228, 1987.

хрущёв | Михаил Горбачёв | Чеварнадсе

Point Location in Arrangements

Stefan Meiser

FB 10, Informatik, Universität des Saarlandes, 6600 Saarbrücken

Eine Menge von Hyperebenen unterteilt den mehrdimensionalen euklidischen Raum E^d in Gebiete (engl. faces) unterschiedlicher Dimension, genannt das Arrangement dieser Hyperebenen. Diese Gebiete unterscheiden sich in ihrer Lage bezüglich der Hyperebenen. Wir stellen eine Datenstruktur vor, die es erlaubt, einen Suchpunkt innerhalb eines Arrangements von n Hyperebenen im E^d in Zeit $O(\log n)$ zu lokalisieren. Unter Lokalisieren verstehen wir hierbei die Bestimmung des faces im Arrangement, das den Suchpunkt enthält. Als Ergebnis der Suche liefern wir einen Pointer auf das face im Inzidenzgraphen des Arrangements zurück. Der Platzbedarf der Datenstruktur beträgt $O(n^{d+\kappa})$ für beliebiges $\kappa > 0$ und ist damit fast optimal, wenn man berücksichtigt, daß der Inzidenzgraph Größe $\Omega(n^d)$ hat.
Die Datenstruktur ist nach dem Prinzip "Teile und Herrsche" aufgebaut, d.h. wir teilen das Suchproblem durch die Auswahl einer konstanten Zahl (abhängig von d) von geeigneten Hyperebenen in kleinere Probleme auf, in denen wir rekursiv weitersuchen können. Zur Auswahl der Hyperebenen bedienen wir uns des Konzeptes der ε-Netze nach Haussler/Welzl [HW86]. Es stellt sich heraus, daß wir zur Aufteilung des Suchproblems das Arrangements triangulieren müssen. Eine ähnliche Vorgehensweise schlägt auch Clarkson [Cl86] vor. Er macht jedoch keine Angaben über die (von d abhängigen) Konstanten in seinen Abschätzungen. Die Konstante zur Suchzeit ist in unserer Arbeit polynomiell in d. Der Aufbau der Datenstruktur erfolgt in "probabilistischer" Zeit $O(n^{d+1+\kappa})$, $\kappa > 0$ beliebig. Da für die Bestimmung eines ε-Netzes kein deterministischer Algorithmus bekannt ist, erfolgt die Auswahl der Hyperebenen durch zufälliges Ziehen.

Literatur:

[HW86] D. Haussler, E. Welzl
Epsilon-Nets and Simplex Range Queries
ACM, Proc. of the 2. Symp. on Comp. Geometry (1986), pp. 61-71

[Cl86] K.L. Clarkson
Further Applications of Random Sampling to Computational Geometry
ACM, STOC (1986), pp. 414-423

[Cl85] K.L. Clarkson
A Probabilistic Algorithm for the Post Office Problem
ACM, STOC (1985), pp. 175-184

Ulrich Huckenbeck,
Universität Würzburg

Würzburg, 23.2.88

Betrifft: Workshop CG'88

Zusammenfassung des Vortrages

'Geometrische Maschinenmodelle'

von U. Huckenbeck

Eines der bekanntesten Maschinenmodelle der Computational Geometry ist die auf $\mathbb{R}^2$ arbeitende RAM; ihre Primitive sind arithmetischer Natur, nämlich die vier Grundrechenarten sowie Größenvergleiche reeller Zahlen.
Demgegenüber sollen hier nun solche Geometrieautomaten vorgestellt werden, deren Primitive **geometrischer** Natur sind:
Die **Zirkel-und-Lineal-Maschine (ZLM)** sowie die **Rechtwinkel-Lineal-Maschine (RLM)** simulieren den Gebrauch der entsprechenden Zeichengeräte, und die erweiterten Versionen **A-ZLM** und **A-RLM** können außerdem noch entscheiden, ob ein beliebiger Punkt in der (festen) Orakelmenge $A \subseteq \mathbb{R}^2$ liegt oder nicht.
Nach der Einführung der (A-)ZLM und der (A-)RLM erfolgt ein Überblick über die vielfältigen Probleme im Zusammenhang mit diesen Automaten. Dabei diskutieren wir kurz ein paar typische Resultate zu den folgenden Themen

- Berechenbarkeit von (partiellen) Funktionen $F: \mathbb{R}^2 \dashrightarrow \mathbb{R}^2$ mit Hilfe von (A-)ZLM und (A-)RLM,
- Entscheidbarkeit von Orakeln B mit Hilfe einer A-ZLM oder einer A-RLM,
- die A-RLM und das Convex Hull Problem.

Abfragen Verzw. Punkt ober/unterh. v. Gerade. ?

Beat Brüderlin, ETH Zürich

Titel: Automatisierung geometrischer Konstruktionen und Beweise.

Abstract

Das Finden einer Konstruktion zu einem gegebenen geometrischen Problem ist ein Spezialfall für den Beweis eines allgemeinen Theorems in der Euklid'schen Geometrie. Die Euklid'sche Geometrie ist bekanntlich durch die Axiomensysteme von Hilbert oder Tarski vollständig definiert. Es ist also naheliegend zu versuchen, solche Beweise mittels "Automated Theorem Proving" Methoden vom Computer selbständig aus den Voraussetzungen und Axiomen herleiten zu lassen. Die meisten bisherigen Ansätze basieren auf einer Übersetzung des geometrischen Problems in die Algebra (Tarski, Wu, etc.) Wir versuchen einen Ansatz direkt in einer geometrischen Spache, was insbesondere für interaktiven Systemen grosse Vorteile bringt. Der wichtigste Teil des Algorithmus' ist die sogenannte äquivalente Ersetzung von Formeln durch einfachere. Mit einer speziellen Formulierung des "critical pair" Algorithmus von Knuth und Bendix kann die Menge der Ersetzungsregeln vervollständigt werden.

Bilderzeugung für sehr große Szenen

Jörg Winckler, Universität Karlsruhe

Viele Bilderzeugungssysteme beschränken den Anwender in seiner Arbeit, weil sie die Bildszene als Ganzes im Hauptspeicher halten wollen. Eine triviale Lösung dieses Problems sind Paging-Algorithmen, die die jeweils benötigten Szenenobjekte einlagern. Der Nachteil dieser Lösung ist offensichtlich ein erhöhter I/O-Anteil in der Anwendung. Um diesen im vernünftigen Rahmen zu halten, müssen Algorithmen und Datenstrukturen entwickelt werden, die das Einlagern von Objekten minimiert. Hilfreich sind in diesem Zusammenhang Algorithmen, die räumliche (und bei Animationen zeitliche) Kohärenzen ausnutzen.

Neben der Anforderung, große Datenmengen verarbeiten zu können, steht der Wunsch nach möglichst naturgetreuer Abbildung. Auf diesem Gebiet hat sich das Strahlverfolgungsverfahren (Ray-tracing) einen Namen gemacht. Dieses Verfahren ist in der Lage, korrekte Spiegel- und Brecheffekte sowie Schatten zu berechnen. Bislang liegt der Preis für die hohe Qualität des Verfahrens in den langen Rechenzeiten.

In diesem Vortrag werden zwei Algorithmen vorgestellt. Der *strahlrelativierte Tiefenpufferalgorithmus* verallgemeinert den klassischen Tiefenpufferalgorithmus so, daß er das Strahlverfolgungsverfahren nachvollziehen kann. Der *szenenorientierte Bildinterpolationsalgorithmus* ermittelt aus zwei Stützframes einer Animation sichtbare Oberflächenteile und interpoliert diese in den Zwischenbildern. Neu zu berechnen sind nur noch die neu sichtbaren Oberflächenteile. Das reduziert die Anzahl der zu verfolgenden Strahlen und macht dadurch den strahlrelativierten Tiefenpufferalgorithmus, dessen Speicheraufwand proportional zur Anzahl der Strahlen ist, anwendbar.

Gribas / Seidel   2. CompG. 1986

Pixel matching

Prof. Dr. Ralf Hartmut Güting
Lehrstuhl Informatik VI
Universität Dortmund

16.2.88

Vortrag beim Workshop über Computational Geometry CG '88 am 24.-25.3.88 an der Universität Würzburg

Titel: Geo-Relationale Algebra: Ein Modell für die Integration geometrischer Algorithmen in Datenbanksysteme.

Zusammenfassung: Zur Verwaltung und Verarbeitung von Daten mit geometrischen Bestandteilen, etwa in der Geographie und Kartographie, im VLSI-Entwurf und allgemein im CAD, werden spezielle Datenbanksysteme, genannt *Geo-Datenbanksysteme*, benötigt. Die wesentlichen Probleme bei der Entwicklung solcher Systeme liegen im Entwurf eines Benutzermodells für Geodaten und ihre Manipulation sowie in der effizienten Implementierung dieses Modells. Das Benutzermodell sollte *einfach*, *präzise*, *ausdruckskräftig* und *effizient implementierbar* sein. - Im Vortrag wird ein Ansatz zur Verwirklichung dieser Ziele beschrieben, genannt *geo-relationale Algebra*. Die Idee besteht darin, relationale Datenbanktechnologie auf allen Ebenen zu erweitern: Auf der konzeptuellen Ebene wird relationale Algebra als die formale Grundlage relationaler Systeme erweitert um geometrische Datentypen und Operatoren. Dies ist auf einfache und saubere Weise möglich, wenn man von einer Sicht der relationalen Algebra als mehrsortige Algebra ausgeht. Eine konkrete Algebra für Anwendungen in zwei Dimensionen wird gezeigt, die als ausdrucksstarke Abfrage- und Datenmanipulationssprache benutzt werden kann. Auf der Implementierungsebene werden Standardtechniken relationaler Systeme integriert mit geometrischen Algorithmen und Datenstrukturen, wie sie in den letzten zehn Jahren im Bereich der "Computational Geometry" entwickelt worden sind.