

Abstracts des 6. Workshops über

Computational Geometry

CG '90

Klaus H. Hinrichs (Ed.)

Bericht Nr. 90 - 01

INFORMATIK-BERICHTE



Institut für Informatik
Postfach 10 12 40
D-5900 Siegen

Vorwort

Der vorliegende Bericht faßt die Kurzfassungen der Vorträge zusammen, die für den 6. Workshop über "Computational Geometry" am 22. und 23. März 1990 an der Universität - Gesamthochschule - Siegen angemeldet wurden. Die vorangehenden Workshops fanden 1983 in Zürich (J. Nievergelt), 1984 in Bern (W. Nef, H. Bieri), 1985 in Karlsruhe (A. Schmitt, H. Müller), 1988 in Würzburg (H. Noltemeier) und 1989 in Freiburg (T. Ottmann) statt. Der Workshop hat das Anliegen, die Kommunikation zwischen den deutschsprachigen Arbeitsgruppen zu fördern, die im Bereich der "Computational Geometry" und ihrer Anwendungen arbeiten. Insbesondere soll der Workshop den jüngeren Nachwuchswissenschaftlern Gelegenheit bieten, in zwangloser Umgebung Ideen und Ergebnisse auszutauschen und zu diskutieren. Die Vielzahl der Anmeldungen zeigt, daß für Workshops dieser Art - und dieser Thematik - ein großer Bedarf besteht.

Die Organisation des Workshops lag zu einem großen Teil in den Händen von Friederike Hoffmann und Liisa Wahler, denen ich an dieser Stelle herzlich für die geleistete Arbeit danke.

Siegen, im März 1990

Klaus H. Hinrichs

Liste der Vorträge

- | | |
|---|--|
| Agarwal, P.
Schwarzkopf, O.
Welzl, E. | Euclidean Minimum Spanning Trees and
Bichromatic Closest Pairs |
| Alt, H.
Behrends, B. | Approximative Kongruenz von Polygonen |
| Becker, B.
Widmayer, P. | Maßstabsunabhängige Verwaltung von Landkarten |
| Benninger, D. | R-Bäume und Abfragen auf skalarwertigen
Vektorkarten |
| Berg, M. de | Translating Polygons with Applications to Hidden
Surface Removal |
| Blankenagel, G.
Güting, R.H. | EST - Externe Segment Trees |
| Brüderlin, B. | Robustness of Computational Geometry Algorithms |
| Hagen, H. | Qualitätsanalyse - Algorithmen |
| Heep, S.
Kriegel, H. P. | Effiziente sekundärspeichergeeignete Randdarstellung
polygonaler Objekte und ihre algorithmische
Bearbeitung |
| Henrich, A. | Die Unterstützung von Nicht-Standardanfragen in geo-
metrischen Datenbanksystemen durch räumliche
Zugriffsstrukturen |
| Hertel, E. | Einige algorithmische Probleme in Polyeder- und
Konvexgeometrie |
| Huckenbeck, U. | Kreisscheiben als Orakel für geometrische
Random Access Machines |
| Kreveld, M.J. van
Overmars, M.H. | Concatenable structures for decomposable problems |
| Kriegel, H. P.
Schiwietz, M. | Eine effiziente und robuste Indexstruktur für
große Mengen polygonaler Objekte |

- Lasser, D. Kurven- und Flächen-Verschneidungsmethoden
- Müller, H. Bestimmung von Nullstellen von Polynomgleichungssystemen durch Basistransformation
- Overmars, M. New Methods for Hidden Surface Removal
- Rothe, C. Algorithmen in einem optimierten, hierarchischen Raumraster
- Schorn, P. Accuracy and Degeneracy in Geometric Computation
- Schuijjer, S. A General Theory of Visibility
- Stifter, S. Voronoi-Diagramme für Teilmengen des 3D und ihre Anwendung auf Kollisionsprobleme bei Robotern
- Vegter, G. Computational Complexity of Combinatorial Surfaces

Euclidean Minimum Spanning Trees and Bichromatic Closest Pairs*

Pankaj Agarwal^{††}

Otfried Schwarzkopf^{**}

Emo Welzl^{**}

Given a set S of N points in Euclidean d -space \mathbb{E}^d , a *Euclidean minimum spanning tree* (EMST) is a spanning tree of S whose edges have a minimum total length among all spanning trees of S , where the length of an edge is the Euclidean distance of its vertices. For $d = 2$, an $\mathcal{O}(N \log N)$ algorithm for the computation of the EMST was given by Shamos and Hoey. For $d \geq 3$, Yao obtained $\mathcal{O}(N^2)$ algorithms. In three dimensions, his algorithm runs in time $\mathcal{O}((N \log N)^{1.5})$, which can be reduced to $\mathcal{O}((N \log N)^{1.5})$ using recent results on the computation of Voronoi diagrams.

Our aim is to shed light on the relation between the EMST problem and the computation of *bichromatic closest pairs* (BCP). The latter problem can be formulated as follows: Given a set of n red and m blue points in \mathbb{E}^d , find a red point p and a blue point q such that the distance of p and q is minimum among all such bichromatic pairs.

It is not difficult to verify that an EMST for the union of the red and blue points contains at least one closest red-blue pair. It is thus possible to solve the BCP problem by computing an EMST. The first result of this paper is to show that the reverse is also true: We give an algorithm that computes an EMST by solving several BCP problems. If we can find a BCP for n red and m blue points in \mathbb{E}^d in time $\mathcal{T}_d(n, m)$, we can compute an EMST in \mathbb{E}^d in time $\mathcal{O}(\mathcal{T}_d(N, N) \log^d N)$.

Practically all EMST algorithms compute a set of edges which can be shown to be a superset of the EMST. In the two-dimensional case the Delaunay triangulation is a good choice for this superset. Unfortunately, in higher dimensions the complexity of the Delaunay triangulation is quadratic, so it does not lead to any improvement as compared to the complete graph. Another possible choice for a suitable superset is due to Yao: he divides the set of all possible edges into a constant number of groups, according to the slope of the edge. From each group only a linear number of edges needs to be selected. Our algorithm is based on a similar idea: we classify all possible edges into a set of $\mathcal{O}(N \log^d N)$ groups, each group forming a complete bipartite subgraph on two subsets P, Q of S . We are then able to show that for each group only the closest pair between P and Q can form an edge of the EMST.

We then turn our attention to the BCP problem. We show a randomized reduction to the problem of finding among n given unit balls those which are disjoint from given m points in \mathbb{E}^d . In three dimensions, this leads us to a randomized algorithm that runs in expected time $\mathcal{O}(n^{3/5} m^{4/5} \psi(\frac{m}{\sqrt{n}}) \log^2 n + n \log^2 n + m \log^2 n)$ where $\psi(t) = 2^{(\alpha^{-1}(t)/5)}$ and $\alpha(t)$ is the functional inverse of Ackermann's function. This implies a $\mathcal{O}(N^{1.4} \log^5 N \psi(\sqrt{N}))$ randomized algorithm for the EMST problem in \mathbb{E}^d .

* First and third author acknowledge support from the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), a National Science Foundation Science and Technology Center under NSF grant STC 88-09648. Second author's work was supported by the Deutsche Forschungsgemeinschaft under grant Al 253/1-2. Third author's work was also partially supported by the ESPRIT II Basic Research Actions Program of the EC under contract no. 3075 (project ALCOM)

[†]DIMACS Center, Hill Center — Busch Campus, P.O. Box 1179, Rutgers University, Piscataway, NJ 08855-1179, USA

[‡]Dept. of Computer Science, Duke University, Durham, NC 27 706, USA

^{**}Institut für Informatik, Fachbereich Mathematik, Freie Universität Berlin, Arnimallee 2-6, D-1000 BERLIN 33, West Germany

Approximative Kongruenz von Polygonen¹

Zusammenfassung

Helmut Alt, Bernd Behrends
FU Berlin

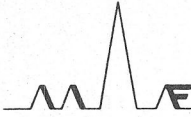
Eine typische Fragestellung aus der Mustererkennung ist, wie "ähnlich" zwei vorgegebene Figuren P, Q einander sind. Dieses Problem wird hier für Polygone P, Q mit Mitteln der algorithmischen Geometrie behandelt. Wir nennen P, Q ϵ -kongruent, falls P kongruent zu einem Polygon in der ϵ -Umgebung von Q ist und umgekehrt. Mit anderen Worten es existiert eine starre Bewegung (Translation und Rotation), die P in die ϵ -Umgebung von Q abbildet und deren Inverse das Umgekehrte bewirkt. Wir betrachten sowohl das *Entscheidungsproblem*: "Gegeben P, Q, ϵ ; sind P, Q ϵ -kongruent?" als auch das *Optimierungsproblem*: "Gegeben P, Q ; berechne das kleinste ϵ mit dieser Eigenschaft".

Das Problem läßt sich spezialisieren durch Einschränkung der starren Bewegung auf nur Translationen (Beispiel Schrifterkennung), auf Translationen entlang einer fest vorgegebenen Richtung oder darauf, daß überhaupt keine Bewegung erlaubt ist, dh. daß man lediglich den Hausdorff-Abstand zweier fest vorgegebener Polygone bestimmen will. Im folgenden bezeichne n die Gesamtanzahl der Ecken von P und Q .

Für die obigen Probleme in der dort angegebenen Reihenfolge erhalten wir Algorithmen mit Laufzeiten der Größenordnungen n^{10}, n^8, n^6, n^2 für das Optimierungs- und n^8, n^6, n^4, n^2 für das Entscheidungsproblem.

Eine bessere Laufzeit für das allgemeine Entscheidungsproblem läßt sich durch einen jnv. ("ja-nein-vielleicht") Algorithmus erzielen. Ein solcher Algorithmus erkennt eine ϵ -Kongruenz nur mit Sicherheit falls die Polygone c -kongruent sind; dabei ist $c < 1$ eine Konstante. Nicht- ϵ -Kongruenz wird auf jeden Fall erkannt. Wir erhalten für obiges Problem einen jnv-Algorithmus der Laufzeit $O(n^2 \log n)$.

¹Gefördert durch die DFG im Rahmen des Schwerpunktprogramms "Datenstrukturen und Effiziente Algorithmen", Projekt Al 253/1-3.



Universität Freiburg
Institut für Informatik
B. Becker
P. Widmayer

Rheinstr. 10-12
D-7800 Freiburg i. Br.
Tel. (0761) 203-3886

Maßstabsunabhängige Verwaltung von Landkarten

Landkarten enthalten je nach Maßstab verschiedene Informationen. So sind beispielsweise auf einer Karte der Bundesrepublik Autobahnen als ziemlich geradlinig verzeichnet; auf der Siegener Stadtkarte hingegen schlängelt sich die Sauerlandlinie deutlich um den Rothenberg.

Aus Gründen der Effizienz und der Vermeidung konsistenzbedrohender Redundanz ist eine integrierte Verwaltung von Landschaftsdaten aller Maßstäbe wünschenswert. Dabei sollte es durch immer stärkere Abstraktion von der Realität — die Generalisierung im Jargon der Kartographen — gelingen, aus einem einzigen Datenbestand geometrische Objekte wie beispielsweise Linienzüge mit unterschiedlichem Detaillierungsgrad zu extrahieren.

Wir stellen eine Datenstruktur vor, die geometrische Anfragen mit variablem Detaillierungsgrad effizient unterstützt. Insbesondere erläutern wir, wie Linienzüge so gespeichert werden können, daß Bereichsanfragen um so schneller beantwortet werden, je weniger detailliert die Antwort sein muß.

R-Bäume und Abfragen auf skalarwertigen Vektorkarten

Daniel Benninger
Mathematisches Institut*
Universität Bern

12. Januar 1990

Abstract

Nicht-Standard Anwendungen von Datenbank-Systemen, wie Computer Vision, Computer-aided Design (CAD) oder Geographische Informationssysteme, gewinnen stetig an Bedeutung. Eine zentrale Rolle spielt dabei die Verwaltung geometrischer Objekte (Punkte, Linien, Polygone u.a.) im d -dimensionalen Euklidischen Raum E^d . Hinzu kommt eine Vielfalt von Operatoren auf geometrischen Daten, die effizient unterstützt werden müssen.

Mengen-Operatoren Schnitt, Verein, Differenz

Such-Operatoren Punkt-Suche (gegeben eine Familie $\mathcal{O} = \{O_i | i \in I, O_i \subset E^d\}$ von Geo-Objekten im E^d und einen ausgezeichneten Punkt $P \in E^d$, finde alle Objekte in \mathcal{O} die P "enthalten".

Bereichs-Suche (gegeben eine Familie \mathcal{O} von Geo-Objekten und ein Referenz-Objekt $S \subset E^d$, bestimme alle Objekte aus \mathcal{O} , die mit dem vorgegebenen Objekt nicht-leeren Schnitt bilden)

Topologische-Operatoren Abstände, Volumina, "nächster" Nachbar

Es existiert eine ganze Palette von geometrischen Indexstrukturen (Datenstrukturen), welche eine effiziente Realisierung der geometrischen Such-Operatoren unterstützen, z.B. *Quadtree* [Finkel'84, Samet'85], *BSP-Baum* [Fuchs'80], *k-d-B-Baum* [Robinson'81], *Gridfile* [Nievergelt'84], *R-Baum* [Guttman'84], *R+-Baum* [Sellis'87], *Cell-Baum* [Günther'88].

In der praktischen Anwendung, etwa bei der Manipulation von Geo-Objekten in der Kartographie und in GISen, muss die jeweilige Indexstruktur die Sekundärspeicherverwaltung, auf Grund der mächtigen Datenmenge, in hohem Masse unterstützen.

Am Beispiel der *skalarwertigen Vektorkarten* [Molenaar'89], einer formalen Datenstruktur für vektororientierte GISe wollen wir die Anwendbarkeit des R-Baumes als Geo-Index diskutieren. Skalarwertige Vektorkarten umfassen Punkte, Linien und Flächen auf der Grundlage einer Knoten resp. Knoten-Kanten-Struktur, wobei jedem Objekt genau eine thematische Ausprägung zugeordnet ist. Wir untersuchen die Fragestellung wie bzw. wie gut R-Bäume den Ereignisraum, dh. die Menge aller Abfragen, von skalarwertigen Vektorkarten insbesondere aber die Such-Operatoren und die topologischen Operatoren, unterstützt.

*Sidlerstrasse 5, CH-3012 Bern, Schweiz (Tel. +41 31 65 88 27)

Translating Polygons with Applications to Hidden Surface Removal

Mark de Berg*

Abstract

In its most general form, the *separability problem* can be stated as follows. Given a set of objects in some space, separate them by a sequence of motions. During the motions, the objects should not collide with each other. These problems come in many different flavors, depending on the objects that are considered, the space they are in, and the type of motions that is allowed.

We consider the following restricted version of the separability problem. Given a set $S = \{P_1, \dots, P_m\}$ of non-intersecting polygons in the plane, translate them in some direction d to infinity, one at a time. Thus, every polygon has to be translated in the same direction. The problem now is to determine whether the polygons can be ordered such that, if the polygons are translated according to that ordering, no collisions occur and, if so, to compute such a *translation ordering*.

In this paper it is shown that a set of (arbitrary) polygons can be preprocessed in time $O(n \log n)$ and space $O(n)$, such that it is possible to determine, for any given direction d , in time $O(\log n)$ whether there exists a translation ordering and to compute one (if it exists) in time $O(n)$. We also show that all directions of separability can be computed in $O(n \log n)$ time.

One of the main applications of the translation problem is in computer graphics. To render a realistic picture of a scene, hidden surface removal has to be performed. One way to do this is to display the objects in the scene in a 'back to front' (with respect to the viewpoint) order. A moment's thought will make it clear that a valid displaying order for this so-called *painter's algorithm* corresponds to a translation ordering for the objects in the direction perpendicular to the viewing plane. However, computing translation orderings in three dimensions efficiently is difficult. Fortunately, for an important class of three dimensional scenes, the so-called *terrains* solutions to the two dimensional translation problem can be used.

The translation ordering for the set of polygonal faces of a scene corresponds to a parallel view of the scene. This is of course unwanted. One of the advantages of our method is that it can easily be adapted to yield a valid displaying order for perspective views within the same bounds.

*Dept. of Computer Science, University of Utrecht, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. Supported by the Dutch Organisation for Scientific Research (N.W.O.). Partially supported by the ESPRIT II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

EST - Externe Segment Trees

Gabriele Blankenagel

Ralf Hartmut Güting

Universität Dortmund
Postfach 500500
D-4600 Dortmund

FernUniversität Hagen
Postfach 940
D-5800 Hagen

Die effiziente Verwaltung von großen, dynamischen Intervallmengen ist ein grundlegendes Problem mit Anwendungen in vielen Bereichen. In der Computational Geometry hat man hierzu verschiedene worst-case effiziente Hauptspeicherstrukturen entwickelt wie den Segment Tree (Bentley), den Interval Tree (Edelsbrunner) und den Priority Search Tree (McCreight). Wir versuchen, die diesen Strukturen zugrundeliegenden Prinzipien und Suchstrategien in angemessener Weise auf Hintergrundspeicher zu übertragen, um auch extern effizient auf Intervallen suchen zu können. Externe Intervallstrukturen können auf vielfältige Weise als Indexstrukturen in Datenbanksystemen eingesetzt werden: Neben der Indizierung von Transaktions- und Gültigkeits-Zeitintervallen in temporalen Datenbanken können sie als eindimensionaler oder als mehrdimensionaler Index für eine Menge mehrdimensionaler geometrischer Objekte mit Ausdehnung verwendet werden - eine Menge von Gebieten in der Ebene wird in einem eindimensionalen Index beispielsweise anhand ihres x -Intervalles dargestellt; in einem mehrdimensionalen Index können geschachtelte eindimensionale Strukturen, etwa ein Segment-Priority Tree, zur Verwaltung ihrer x - und y -Intervalle eingesetzt werden.

Mit dem *externen Segment Tree (EST)* stellen wir eine voll dynamische Intervallstruktur vor, die auf dem Segment Tree basiert. Wie im internen Fall unterstützt der EST das Auffinden aller über einem Raster definierten Intervalle, die einen Suchwert beinhalten. Um die Struktur kurz zu skizzieren, gehen wir von einem vollständigen binären Baum aus, dessen Blättern die Elementarintervalle des Rasters und dessen inneren Knoten die Vereinigung der Intervalle der Sohnknoten zugeordnet sind. Der EST besteht aus einer *Skelettstruktur* - einem Teilbaum dieses Baumes, der die Wurzel enthält - in die die Intervalle der Menge I eingetragen werden. Jeder Knoten besitzt eine *Überdeckungsliste*, jeder Blattknoten zusätzlich eine *Blattliste*. Überdeckungslisten enthalten, wie im internen Fall, Intervalle aus I , die das Knotenintervall vollständig überdecken. Eine Blattliste enthält bis zu c Intervalle aus I , die nur einen Teil des Knotenintervalles überdecken. Anders als beim internen Segment Tree sind die Blattintervalle beim EST i. a. wesentlich größer als 1. - Mit dem Wachstum von I wächst die Skelettstruktur top-down durch Überläufe von Blattlisten: Beim Überlauf einer Blattliste erhält der bisherige Blattknoten zwei Söhne mit jeweils eigenen Überdeckungs- und Blattlisten, auf die die Intervalle der ursprünglichen Blattliste verteilt werden. (Filtert ein innerer Knoten bei kleiner werdendem I weniger als c Intervalle, wird er umgekehrt wieder zu einem Blattknoten.) - Die so beschriebene Struktur wird auf drei Arten von Seiten dargestellt: Die Skelettstruktur steht auf *Strukturseiten*; jede Strukturseite enthält maximal einen vollständigen binären Baum der Höhe h , i. a. einen Teilbaum desselben mit dessen Wurzel. Eine Blattliste steht normalerweise auf einer eigenen *Blattseite*. Ein Überlauf einer Blattliste, und damit einer Blattseite, führt zum oben beschriebenen Wachstum der Skelettstruktur. Blattseiten mit Füllungen unter 50% werden balanciert. Große Überdeckungslisten, die eine Seite zu mehr als 50% füllen, stehen auf eigenen *Überdeckungsseiten*(sequenzen). Sehr große Überdeckungslisten können auch zu einer beliebigen Sekundärstruktur ausgebaut werden, wodurch geschachtelte Strukturen entstehen. Kleine Überdeckungslisten werden gemeinsam mit anderen auf Überdeckungsseiten verwaltet. Hierbei tritt das interessanteste Problem für externe Segment Trees auf: Kleine Überdeckungslisten sind so auf Seiten zu verwalten, daß Punkteinschluß-Suchen möglichst gut unterstützt werden und eine Seitenauslastung von über 50% garantiert wird. Wir geben einen einfachen Algorithmus an, der für eine gegebene Intervallmenge I eine solche Seitenaufteilung statisch erzeugt und dynamisch aufrecht erhält, wobei er nur eine Anzahl von Balancieroperationen benötigt, die proportional zur Pfadlänge ist.

Insgesamt sind bei den Punkteinschluß-Suchen auf gleichverteilten Intervallmengen konstanter Länge höchstens 2 bis 4 Zugriffe auf Seiten zu erwarten, deren Inhalt nicht vollständig zur Ergebnismenge gehört (alle anderen betrachteten Seiten sind durchschnittlich zu mindestens 50% mit Ergebnisintervallen gefüllt). Allgemein ist der EST durch eine geringe durchschnittliche Anzahl von Suchzugriffen auf Überdeckungsseiten mit gemeinsam verwalteten kleinen Überdeckungslisten gekennzeichnet. Ein Intervall verursacht auf den Ebenen der Skelettstruktur, deren Knotenintervalle kürzer als die Intervalllänge sind, durchschnittlich nur einen Eintrag, auf der obersten dieser Stufen weniger.

Robustness of Computational Geometry Algorithms

Beat Brüderlin

Computer Science, University of Utah, Salt Lake City, UT 84112
bdb@cs.utah.edu

Efficient algorithm design is one of the major topics in computational geometry. The analysis of the complexity of most algorithms relies on the fact that certain basic operations, such as, for instance, finding the distance between two points, or intersecting two lines, can be done in constant time. However, this is only the case if the underlying geometric primitives are represented with constant precision. Most algorithms in computational geometry use a floating point representation because it is very flexible in representing large domains of real numbers, and also, because floating point arithmetic is efficiently implemented in the hardware of most computers. On the other hand, the floating point representation is only an approximation of the domain of real numbers. This causes severe difficulties with the interpretation of the data, especially, when branching decisions in an algorithm are based on the comparison of approximate data. The result of the algorithm is likely to be inconsistent.

Some recent publications have addressed the difficulties to find a representation of geometric objects by floating point numbers that is consistent with some intended model in a real Euclidean space. Some of the approaches taken will be presented in this talk. Most of them cause new problems by drastically increasing the complexity of the algorithm, and by restricting the domain of representable objects. It seems necessary to trade in speed and expressiveness for robustness.

We show a new, more pragmatic approach to this problem which does not have most of the aforementioned difficulties. It does some reasoning about the confidence we can have in a decision on approximate data. The method detects ambiguities that may lead to decisions that are inconsistent with decisions made earlier in the algorithm. This method can be applied to most computational geometry algorithms without changing their logic structure, and without affecting their asymptotic time complexity in most occurring cases. Moreover, it imposes almost no restriction on the intended data model.

Literature:

Beat Brüderlin: A Pragmatic Approach to Robustness and Accuracy Problems in Computational Geometry. Tech. Report, Computer Science, University of Utah, 1990.

Beat Brüderlin: Robust Regular Set Operations on Polyhedra. Tech. Report, Computer Science, University of Utah, 1990.

C.M. Hoffman: The Problems of Accuracy and Robustness in Geometric Computation. IEEE Computer, Vol. 22, No. 3, March 1989.

Qualitätsanalyse - Algorithmen

Hans Hagen

Fachbereich Informatik
Universität Kaiserslautern

Computer Aided Geometric Design ist aus dem Bedarf für Freiformflächen in der CAD/CAM-Technologie entstanden und hat sich zu einem Forschungsschwerpunkt der Informatik mit direkten Anwendungen in den Ingenieurwissenschaften entwickelt. In der letzten Zeit stellt sich, neben der reinen Konstruktion immer mehr die Frage nach Möglichkeiten die Qualität" einer Fläche vor dem eigentlichen NC-Prozeß beurteilen zu können. Derartige Algorithmen werden im Rahmen dieses Vortrags vorgestellt und an praktischen Beispielen erläutert.

Effiziente sekundär-speichergeeignete Randdarstellung polygonaler Objekte und ihre algorithmische Bearbeitung

S. Heep, H.-P. Kriegel
Praktische Informatik, Universität Bremen

Kurzfassung

In einigen Anwendungen, wie z.B. im CAD-Bereich, ist es unrealistisch zu jedem Zeitpunkt die gesamte Information über die Geometrie der betrachteten Bauteile im Hauptspeicher zu halten. Darüberhinaus sind bei Anfragen auch nur Teilobjekte von Interesse, so daß die gesamte Information über Objekte nicht erforderlich ist. Deshalb beschäftigen wir uns mit externen Algorithmen, die mit sublinearem bzw. konstantem Hauptspeicherplatz auskommen. Dazu wird zunächst eine Darstellung für 2-dimensionale einfache Polygone mit Löchern vorgestellt, die effiziente externe Algorithmen ermöglicht. Diese Darstellung kann sowohl auf nicht polygonal begrenzte, als auch auf 3-dimensionale Objekte (rigid solid objects) verallgemeinert werden.

Die bekannteste Darstellung geometrischer Objekte im 2- oder 3-dimensionalen Raum ist die Randdarstellung. Diese besteht aus den geometrischen Informationen über Punkte und (Begrenzungs-) Kanten (in 3D darüberhinaus Flächen) und den topologischen Beziehungen zwischen diesen. Der Zugriff auf die Geometrieinformationen erfolgt in herkömmlichen Systemen fast ausschließlich über die Topologie, die eine hierarchische Struktur besitzt. Im Datenbankbereich, genauer bei Zugriffsstrukturen die den Sekundär-speicher benutzen, bringt die hierarchische Struktur jedoch Effizienzprobleme mit sich. Daher benötigt man Darstellungen für geometrische Objekte, die den Zugriff über geometrische Informationen realisiert und eine effizientere Bearbeitung geometrischer Anfragen und Operationen zuläßt.

Eine bekannte Datenstruktur, welche die Geometrie anstelle der Topologie bei der Darstellung in den Vordergrund stellt, ist der quadtree. Während der region quadtree nur eine approximative Darstellung von Objekten zuläßt, gibt es auch Varianten wie den edge quadtree oder den nonminimal division quadtree, die eine exakte Darstellung von Objekten ermöglichen. Problem dieser Strukturen ist, daß bei sekundär-speichergeeigneten Varianten, wie dem linearen quadtree, im allgemeinen a priori eine maximale Auflösung vorgegeben werden muß. Dadurch geht entweder die Exaktheit verloren oder der Speicheraufwand wird sehr hoch.

In unserem Ansatz werden diese Nachteile dadurch überwunden, daß die Zerlegung des Objektes nicht bezüglich eines festen Gitters erfolgt, sondern nur durch den Verlauf der Begrenzungskanten dirigiert wird. Damit ist die Anzahl der Elemente der Zerlegung allein abhängig von der Struktur des Objektes. Im folgenden wird eine Objektdarstellung vorgestellt, die

- Objekte eindeutig zerlegt
- Objekte exakt repräsentiert und deren Speicheraufwand nicht von einem zugrunde liegenden Gitter abhängt
- auf effizienten Sekundär-speicherstrukturen zur Verwaltung von Rechtecken beruht
- Information über das Innere der Objekte explizit beinhaltet.

Grundlage ist eine Zerlegung in Trapezoide, wie sie 1985 von Asano und Asano vorgestellt wurde. Die Trapezoide werden in der hier vorgestellten Form mittels ihrer minimal umgebenden Rechtecke in einer Sekundär-speicherstruktur wie Grid File oder Buddy-Tree gespeichert.

In diesem Vortrag wird zunächst ein Algorithmus zur Zerlegung einfacher Polygone mit Löchern in die oben beschriebene Trapezoiddarstellung vorgestellt. Es folgen Algorithmen für die Operationen Punkt-im-Polygon Test und Rechteckschnittanfrage, die genau ein Objekt betreffen. Danach werden Algorithmen zum Identitätstest und zur Berechnung mengentheoretischer Operationen auf je zwei Objekten in dieser Darstellung präsentiert.

Die Unterstützung von Nicht-Standardanfragen in geometrischen Datenbanksystemen durch räumliche Zugriffsstrukturen*

Andreas Henrich, Praktische Informatik III, FernUniversität Hagen, Postfach 940, 5800 Hagen

In Datenbanken, die neben den Attributtypen *String*, *Real*, *Integer* und *Boolean* auch geometrische Attribute wie Punkte oder Polygonzüge (offen oder geschlossen) zulassen, müssen auch Anfragen, in denen derartige Attribute spezifiziert werden, effizient unterstützt werden. Ein wesentlicher Schritt in diese Richtung kann dabei durch den Einsatz geometrisch clusternder Zugriffsstrukturen getan werden.

Bisher beschränkte sich die Betrachtung derartiger Zugriffsstrukturen aber meist auf "einfache" Bereichsanfragen mit rechteckigen Anfragebereichen, bei denen all die Objekte gesucht werden, die innerhalb des Anfragebereiches liegen oder diesen schneiden. Es gibt aber noch zahlreiche andere Anfragetypen, von denen drei exemplarisch betrachtet werden sollen (Zur Formulierung der Beispielanfragen wird dabei die georelationale Algebra von Güting [1] verwendet).

Bereichsanfragen mit polygonalen Anfragebereichen

Eine solche Anfrage ergibt sich z.B. wenn alle Städte in NRW gesucht werden: `cities σ [center inside NRW]`. Die Frage ist nun, wie man solche Anfragen mit einer geometrisch clusternden Zugriffsstruktur unterstützen sollte. Sollte man (1) den komplexen Anfragebereich generell durch ein minimales umschließendes Rechteck approximieren, (2) bei der Suche im Directory das minimale umschließende Rechteck verwenden und vor jedem Bucketzugriff die zugehörige Bucketregion mit dem komplexen Anfragebereich vergleichen oder (3) während der ganzen Suche immer den komplexen Anfragebereich verwenden?

Unsere Simulationen haben gezeigt, daß bei kleinen Anfragebereichen die Variante (2) gegenüber der Variante (3) leichte Vorteile hat. Bei größeren Anfragebereichen ist aber die Variante (3) deutlich überlegen. Die Variante (1) ist immer am ungünstigsten.

Closest- und Distance-Sort-Anfragen

Eine Closest-Anfrage ergibt sich z.B. wenn man das Krankenhaus sucht, das am nächsten bei einem gegebenen Unfallort liegt: `hospitals Unfallort closest[location]`

Wir haben im LSD-Baum [2, 3] einen Algorithmus implementiert, der unter Ausnutzung der geometrischen Clustering die abgespeicherten Objekte nach ihrem Abstand zu einem vorgegebenen Punkt aufsteigend sortiert ausgibt. Es hat sich gezeigt, daß mit diesem Algorithmus Anfragen wie die obige ausgesprochen effizient bearbeitet werden können. Ein weiteres Beispiel für eine Anfrage, die mit diesem Algorithmus effizient unterstützt werden kann, ergibt sich z.B. wenn man die n Hotels sucht, die am nächsten bei einem vorgegebenen Tagungsort liegen, wobei die Ausgabe automatisch sortiert erfolgt.

Gemischte Anfragen

In einem geometrischen Datenbanksystem werden zwangsweise auch Anfragen auftreten, in denen gleichzeitig sowohl Standardattribute als auch geometrische Attribute spezifiziert sind. Wenn man z.B. alle Städte mit mehr als 50.000 Einwohnern in der Schweiz sucht, ergibt sich: `cities σ [center inside Schweiz and cpop \geq 50000]`

Wenn man nun bedenkt, daß in einer geometrischen Zugriffsstruktur ohnehin mehrdimensionale Objekte verwaltet werden, dann liegt es nahe, einzelne Standardattribute einfach als weitere Dimensionen zu betrachten und so einen Index über mehrere Attribute zu errichten. Dabei ergeben sich allerdings zahlreiche interessante Probleme, von denen wir u.a. folgende untersucht haben: (1) Wie wirkt es sich aus, daß die abzuspeichernden Objekte bezüglich einzelner Dimensionen intervallwertig sind, während sie in anderen Dimensionen punktartig sind? (2) Wie können UND- und ODER-Verknüpfungen zwischen Selektionsbedingungen effizient bearbeitet werden? (3) Wie können Selektionsbedingungen, in denen mehrere Attribute (standard und/oder geometrisch) gleichzeitig auftreten, effizient bearbeitet werden (die Anfrage `projects σ [return \geq (expense \times 1.05)]` wäre ein Beispiel hierfür)?

Es hat sich gezeigt, daß diese Probleme zwar nicht trivial, aber doch lösbar sind, und daß eine geometrische Zugriffsstruktur wie z.B. der LSD-Baum auch sinnvoll als Index über mehreren Attributen (standard und/oder geometrisch) eingesetzt werden kann.

Literatur:

- [1] Güting, R.H.: 'Gral An Extensible Relational Database System for Geometric Applications', Proc. 15th Int. Conf. on VLDB (1989), Amsterdam
- [2] Henrich, A., Six, H.-W., Widmayer, P.: 'The LSD tree: spatial access to multidimensional point and non-point objects', Proc. 15th Int. Conf. on VLDB (1989), Amsterdam
- [3] Henrich, A., Six, H.-W., Widmayer, P.: 'Paging binary trees with external balancing', Proc. Int. Workshop on Graphtheoretic Concepts in Computer Science (WG '89), Springer Lecture Notes in Comp. Science, to appear

* Diese Arbeit wurde von der Deutschen Forschungsgesellschaft unterstützt (Az.: Si 374/1)

Einige algorithmische Probleme in Polyeder- und Konvexgeometrie

E. Hertel (Jena)

1. Zwei Polyeder (eines n -dimensionalen Raumes konstanter Krümmung) heißen zerlegungsgleich, wenn sie sich in eine endliche Anzahl paarweise kongruenter Teilpolyeder zerlegen lassen. Für den Fall der euklidischen Ebene weiß man, daß zwei Polygone genau dann zerlegungsgleich sind, wenn sie inhaltsgleich sind, aber ein allgemeingültiger effektiver Algorithmus zur Realisierung der Zerlegungsgleichheit zweier Polygone mit möglichst wenig Teilen ("Zuschneideproblem") scheint nicht bekannt zu sein. Für den Fall der Zerlegungsähnlichkeit von Polygonen (die Teilpolygone sind paarweise ähnlich) kann ein solcher Algorithmus angegeben werden.
2. Eine minimale Datenstruktur für ein konvexes Polyeder P (mit weniger Seitenflächen als Eckpunkten) ist der "Vektorstern" von P . Seit Minkowski weiß man, daß umgekehrt jeder Vektorstern ein konvexes Polyeder Q bis auf Translation eindeutig bestimmt, aber ein effektiver Algorithmus zur "Konstruktion" von Q scheint nicht bekannt zu sein.
3. Ein wichtiges Modell für die automatische Bildverarbeitung ist die Konvexität in (endlichen) metrischen Räumen. Für das Studium dieses Modells interessieren Algorithmen für die Aufzählung (aller) konvexen Mengen und für konvextreue Einbettungen in Standardräume.

Kreisscheiben als Orakel für geometrische Random Access Machines

Ulrich Huckenbeck, Universität Würzburg,
Lehrstuhl für Informatik I, Am Hubland, 8700 Würzburg

Abstract: Eine *geometrische Random Access Machine* ist eine RAM, die mit Punkten und Geraden der euklidischen Ebene \mathbb{R}^2 arbeitet. Ihre wichtigsten Fähigkeiten sind

- Anwendung der vier Grundrechenarten auf Punktkoordinaten,
- Abfragen, ob eine reelle Zahl positiv ist,
- Ausdrucken eines Punktes, einer Geraden oder einer Antwort $\in \{ 'JA', 'NEIN' \}$.

Eine *A-RAM* ($A \subseteq \mathbb{R}^2$) ist eine abgewandelte geometrische RAM; anstelle des Vergleiches mit 0 kann sie abfragen, ob ein Punkt in A liegt oder nicht. — Sei M eine *A-RAM*; dann *entscheidet* M die Menge $B \subseteq \mathbb{R}^2$, falls sie jede Frage 't $\in B$?' (mit $t \in \mathbb{R}^2$) richtig beantwortet.

Im Rahmen dieses Vortrages beschäftigen wir uns hauptsächlich mit Orakeln der Form $A = \overline{B}(\mathcal{O}, a)$, das sind abgeschlossene Kreisscheiben um $(0,0)$ mit Radius $a > 0$; bei der Untersuchung von Entscheidbarkeitsbeziehungen betrachten wir die zugehörigen Kreislinien $S(\mathcal{O}, a)$.

Dann gilt: Seien $a, b > 0$, so sind folgende Aussagen äquivalent:

- (I) $\overline{B}(\mathcal{O}, b)$ kann durch eine $\overline{B}(\mathcal{O}, a)$ -RAM M in konstanter Zeit entschieden werden.
- (II) Es gibt eine rationale Funktion $R : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ mit ganzzahligen Koeffizienten, die den Rand von $\overline{B}(\mathcal{O}, b)$ in den Rand von $\overline{B}(\mathcal{O}, a)$ überführt; d.h. $R[S(\mathcal{O}, b)] \subseteq S(\mathcal{O}, a)$. — Falls b transzendent ist, so ist außerdem an der Stelle $x = b$ die euklidische Norm von $R(x, 0)$ streng monoton steigend.

Für transzendente und gewisse algebraische Zahlen b sind (I) und (II) äquivalent zu der Aussage

- (III) Es gibt rationale Funktionen $r, s : \mathbb{R} \rightarrow \mathbb{R}$ mit ganzzahligen Koeffizienten derart, daß

$$a^2 = r(b^2)^2 + s(b^2)^2 \text{ oder } a^2 = b^2 \cdot (r(b^2)^2 + s(b^2)^2).$$

Wenn b transzendent ist, so ist außerdem $f'(b^2) > 0$ mit $f(z) = r(z^2)^2 + s(z^2)^2$ bzw. $f(z) = z^2 \cdot (r(z^2)^2 + s(z^2)^2)$.

Concatenable structures for decomposable problems*

Marc J. van Kreveld Mark H. Overmars

December 11, 1989

Abstract

Given a data structure and an ordering on the objects it contains, we study methods for obtaining a modified data structure that allows for splits and concatenations with respect to that ordering. A new and general technique is given, which works for all data structures for decomposable searching problems and order decomposable set problems. The technique is based on a variant of the equal block method, used for making data structures dynamic. As a consequence, the modified structures also allow for insertions and deletions. For example, one could make an interval tree that allows for splitting and concatenating on the lengths of the intervals (in $O(\sqrt{n \log n})$ time), in addition to stabbing queries (in $O(\sqrt{n \log n} + k)$ time, where k is the size of the output), and insertions and deletions (in $O(\log n)$ time). Another application is a linear size data structure for maintaining the convex hull of a planar set of points, where a split or concatenation on first or second coordinate takes $O(\sqrt{n \log n} \log n)$ time. In the technique, a trade-off can be specified between the query time and the other operations.

An immediate consequence of being able to split and concatenate is a method for the addition of range restrictions. Among others, this yields a new version of the k -d tree, which is interesting in its own right. The two-dimensional structure allows for member queries, insertions and deletions in $O(\log n)$ worst-case time, range queries in $O(\sqrt{n \log n} + k)$ time (where k is the size of the output), and splitting and concatenating on first and second coordinate in $O(\sqrt{n \log n})$ time. The structure uses linear storage.

*Authors address: Dept. of Computer Science, Utrecht University, P.O. Box 80.089, 3508TB Utrecht, the Netherlands. This research was partially supported by the ESPRIT II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

Eine effiziente und robuste Indexstruktur für große Mengen polygonaler Objekte

H.P. Kriegel und M. Schiwietz
Praktische Informatik / FB3
Universität Bremen

Kurzfassung

Der Hauptaugenmerk bei der Anfragebearbeitung in Datenbanksystemen lag bisher auf sogenannten Punktzugriffsstrukturen, die geeignet waren, die Tupel eines relationalen Datenbanksystems (aufgefaßt als Punkte im entsprechend-dimensionalen Raum), effizient zu speichern. Diese Punktzugriffsstrukturen sind jedoch überfordert, um die in den sogenannten Nicht-Standard-Anwendungen auftretenden komplexen Objekte, fast immer räumliche Objekte, auch Geo-Objekte genannt, effizient zu unterstützen.

Ein Beispiel hierfür sind "Geographische Informationssysteme", wo im wesentlichen Polygone als Objekte auftreten. Bisherige Ansätze legen zur Organisation des Zugriffs auf diese Objekte zweidimensionale minimal umgebende achsenparallele Rechtecke als Approximation um die Polygone und verwalten diese mit Hilfe von Punktzugriffsstrukturen unter Verwendung bestimmter Techniken (Transformation, überlappende Regionen, Clipping). Die exakte Darstellung wird gesondert gehalten. Der Vorteil liegt in der leichten Erweiterbarkeit einer Punktzugriffsstruktur zur Handhabung beliebig-dimensionaler Geo-Objekte. Das Anfrageverhalten leidet jedoch unter der ungenauen Approximation, die sich negativ auf die Anzahl der bei Anfragen benötigten Sekundärspeicherzugriffe auswirkt.

Daher gehen neuere Anstrengungen u.a. in Richtung eines Minimierens der zusätzlichen Überlappung innerhalb des Directorys durch exaktere Approximation der Objekte. Einer der ersten Ansätze war der Zellbaum, der jedoch durch seinen binären Directory-Aufbau benachteiligt ist (BSP, "Clipping").

Unser Ansatz benutzt ein hierarchisches Directory zur Verwaltung der Objekte, wobei ein Vaterelement im günstigsten Fall das minimal umgebende konvexe Polygon seiner Söhne darstellt ("Überlappende Regionen"). Aus Gründen der beschränkten Kapazität einer Directory-Seite darf jedoch nicht eine beliebige Anzahl von Eckpunkten dieser polygonalen Directory-Elemente zugelassen werden. Auf der anderen Seite müssen diese Directory-Einträge möglichst komprimiert gespeichert werden, um einen hohen Verzweigungsgrad zu garantieren. Wir stellen ein Verfahren vor, das beide Punkte berücksichtigt und nur unwesentlich mehr Speicherbedarf für die Directory-Elemente benötigt als beispielsweise der R-Baum. Gegenüber dem R-Baum hat unser Verfahren, zusätzlich zur Funktionalität bei Polygonen, den Vorteil einer flexibleren Partitionierung und einer, hierdurch bedingten, wesentlich geringeren Überlappung der Directory-Elemente.

Ausgehend von der Struktur des R-Baumes zur Verwaltung von Polygonen über deren minimal umgebende Rechtecke, verallgemeinern wir dessen Partitionierungsmethode im Hinblick auf Minimierung der Überlappung innerhalb des Directorys. Wir stellen weiterhin eine Methode vor, die eine hohe Kapazität der Directory-Seiten bei nahezu optimaler Partitionierung gewährleistet. Abschließend verdeutlichen wir den Leistungsgewinn gegenüber dem R-Baum.

Kurven - und Flächen-Verschneidungsmethoden

Dieter Lasser

Fachbereich Informatik

Universität Kaiserslautern

Schnittpunkte von (ebenen) Kurven oder auch Schnittkurven von Flächen müssen bei vielen Anwendungen ermittelt werden, etwa

- bei der Erzeugung von Niveaulinienbildern zur graphischen Darstellung von vorgegebenen Flächen und Hyperflächen,
- beim Berechnen von Umrißlinien zur Verbesserung graphischer Darstellungen von Flächen,
- beim booleschen Verknüpfen von Körpern,
- beim Konstruieren glatter Übergangskurven bzw. -flächen zur Abrundung von durch Verschneidung entstandenen Ecken und Kanten zwischen Kurven bzw. Flächen oder auch
- beim Bestimmen von Offsetkurven bzw. -flächen, die Selbstdurchdringungen besitzen können.

Ein guter (Flächen-) Verschneidungsalgorithmus ist gekennzeichnet durch die Begriffe

- **Genauigkeit**, im herkömmlichen numerischen Sinne,
- **Zuverlässigkeit**, im Sinne, daß die Durchdringung vollständig und richtig, auch bei Zerfall in mehrere Teilkurven (Punkte) ermittelt wird,
- **Schnelligkeit**, insbesondere im Hinblick auf das interaktive Arbeiten und
- **Selbständigkeit**, im Sinne, daß die Durchdringung ohne interaktiven Eingriff des Benutzers ermittelt wird.

Dieser Beitrag gibt einen Überblick über und, soweit dies durch bereits ausgeführte Untersuchungen gefestigt ist, einen Vergleich vorhandener Verschneidungsalgorithmen und -methoden.

Bestimmung von Nullstellen von Polynomgleichungssystemen durch Basistransformation

Heinrich Müller
 Institut für Informatik
 Universität Freiburg

Polynomgleichungssysteme treten bei der graphischen Darstellung implizit definierter Flächen des geometrischen Modellierens, bei Bewegungsplanungsproblemen der Robotik und bei der Auswertung multidimensionaler Sensorsignale auf. Sowohl die numerische Mathematik als auch die Computeralgebra stellen Lösungsverfahren für solche Systeme zur Verfügung. Die Numerik ist dabei in der Lage, recht schnell lokale Lösungen zu finden, während die Computeralgebra hilft, sich eine Übersicht über den ganzen Lösungsraum zu verschaffen, um unter mehreren Lösungen die überhaupt gewünschte bzw. alle zu finden. Verschiedene numerische Verfahren sind aus dem Newton-Verfahren abgeleitet [Allgower et al., 1983]. Beispiele algebraischer Verfahren sind das auf der Variablenelimination mit Resultanten basierende Collins-Verfahren [Collins, 1975] sowie die Gröbner-Basen [Boege et al., 1989]. Diese Verfahren haben einen erheblichen Rechenaufwand, der sie nur für kleine bzw. größere Systeme spezieller Bauart praktikabel machen.

Ein Kompromiß, die lokale Numerik mit globalen Garantieaussagen zu verbinden, stellt die Intervallrechnung dar. Diese zerlegt den Variablenraum in Gebiete, in denen Lösungen liegen und solche ohne Lösungen. Bei hinreichend feiner Näherung können so die Lösungen isoliert werden. Die Notwendigkeit, die Rechenoperationen jeweils für die Intervallgrenzen durchführen zu müssen, erhöht jedoch den Rechenaufwand.

Bei Polynomgleichungssystemen kann die Mehrfachrechnung vermieden werden. Sei

$$\sum_{i_1=0}^{d_1} \dots \sum_{i_m=0}^{d_m} a_{i_1, \dots, i_m, j} x_1^{i_1} x_2^{i_2} \dots x_m^{i_m} = 0, \quad j = 1, \dots, n,$$

das Gleichungssystem. Gesucht werden Lösungen mit

$$(x_1, \dots, x_m) \in [0, 1]^m.$$

Durch Transformation des Gleichungssystems auf die Bernsteinbasis, bestehend aus Produkten der Bernsteinpolynome $B_d^i := \binom{d}{i} t^i (1-t)^{d-i}$ erhält man die sogenannte Bézierdarstellung

$$\sum_{i_1=0}^{d_1} \dots \sum_{i_m=0}^{d_m} b_{i_1, \dots, i_m, j} B_{d_1}^{i_1}(t_1) \dots B_{d_m}^{i_m}(t_m) = 0, \quad j = 1, \dots, n,$$

des Systems [Hoschek, Lasser, 1989]. Die Koeffizienten $(b_{i_1, \dots, i_m, 1}, \dots, b_{i_1, \dots, i_m, n})$, $i_1 = 0, \dots, d_1, \dots, i_m = 0, \dots, d_m$, der Bézierdarstellung definieren ein m -dimensionales Punktgitter im \mathbb{R}^n ,

dessen geometrisches Aussehen Aussagen über das Lösungsverhalten zuläßt. So hat das Punktgitter die Eigenschaft, daß $P([0, 1]^m)$ Teilmenge seiner konvexen Hülle ist, wobei P die linke Seite des Gleichungssystems bezeichnet. Enthält also die konvexe Hülle den Ursprung des \mathbb{R}^n nicht, dann hat das Gleichungssystem keine Lösung.

Das Béziérgitter kann auf einfache, stabile Weise auch für Teilintervalle von $[0, 1]^n$ berechnet werden. Damit ist die Suche nach den Nullstellen eines Polynomgleichungssystems auf ein geometrisches Suchproblem reduziert. Gegenstand des Vortrags sind algorithmische Teilprobleme, die mit der geschilderten Lösungsstrategie verbunden sind.

Zitate

E.L. Allgower, K. Georg (1983), Predictor-Corrector and Simplicial Methods for Approximating Fixed Points and Zero Points of Nonlinear Mappings, in: Mathematical Programming: The State of the Art, Springer-Verlag

W. Boege, R. Gebauer, H. Kredel (1989), Some Examples for Solving Systems of Algebraic Equations by Calculating Groebner Bases, J. Symbolic Computation 1, 83-98

G.E. Collins (1975), Quantifier elimination for real closed fields by cylindrical algebraic decomposition, 2nd GI Conference on Automata Theory and Formal Languages, LNCS 33, 134-183

J. Hoschek, D. Lasser (1989), Grundlagen der geometrischen Datenverarbeitung, B.G. Teubner, Stuttgart

New Methods for Hidden Surface Removal

Mark H. Overmars*

December 11, 1989

Abstract

An important problem in computer graphics is *hidden surface removal*. In a typical setting of the problem we are given a set of non-intersecting polyhedral objects in 3-space, and a viewing point v , and our goal is to construct the view of the scene, as seen from v .

Many solutions have been developed to date. Some of them use an *image-space* approach, in which one tries to calculate for each pixel which object is visible. These techniques often have hardware implementations. Other techniques have an *object-space* flavor. They compute a combinatorial representation of the view of the scene, called the *visibility map*, whose complexity does not depend on the screen size, but only on the combinatorial complexity of the scene. Such object space methods are also important for e.g. hidden line elimination or determining the parts of objects that are lighted by light sources.

Let n denote the number of edges of the given polyhedra. If we project all these edges onto the plane of view we obtain an arrangement of n (usually intersecting) segments. The visible portion of the scene yields a polygonal decomposition of the plane into regions bounded by portions of these edges. These well known observations lie at the basis of practically all object-space hidden surface removal algorithms. Many of these algorithms simply calculate the entire arrangement of the projected edges, and then determine which features of the arrangement are visible. Crude implementations of this approach run in time $O(n^2)$. More careful implementations run in time $O((n + I) \log n)$, where I denotes the number of intersections between the projected edges.

The main problem with these solutions is that they are not *output-sensitive*. When the combinatorial complexity of the viewed scene is small, these solutions can be very inefficient. In this talk we present an improved output-sensitive algorithm for the hidden surface removal problem that is conceptually very simple and works for various classes of objects. The only assumption we have to make is that the objects can be ordered by their nearness to the viewing point. Let k be the number of vertices in the projected visible scene. The method runs in time $O(n\sqrt{k} \log n)$. If we specialize this algorithm to the case of n horizontal discs, we obtain an algorithm whose time complexity is $O(n^{3/2} \log n + k)$.

We also present a technique for merging visibility maps of subsets of the objects to obtain the total visibility map of the whole set. This method has application in e.g. animation. It also yield efficient algorithms for special cases of the hidden surface removal problem.

*Department of Computer Science, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands.

Algorithmen in einem optimierten, hierarchischen Raumraster

Christoph Rothe

Zusammenfassung:

Das schnelle Auffinden im Raum verteilter Objekte stellt das zentrale Problem der bekannten Bilderzeugungsalgorithmen, wie beispielsweise des Ray-Tracings, dar. Die Raumrasterung ist ein inzwischen gängiges Verfahren um diese Aufgabe zu beschleunigen. Die bekannten Verfahren zur Raumrasterung lassen sich in zwei Gruppen einteilen: während bei einfachen Rastern homogen über den gesamten Szenebereich gerastert wird, kann mit Hilfe hierarchischer Strukturen eine ungleichmäßige Verteilung der Objekte besser berücksichtigt werden. Mit hierarchischer Rasterung ist es ohne großen Speicherverschnitt möglich, jeder Zelle des Raumrasters ungefähr die gleiche Anzahl von Objekten zuzuordnen. Somit ergibt sich eine Rasterung, die die Verteilung der Objekte in der Szene berücksichtigt, was für die Bilderzeugung vorteilhaft ist. Der größte Nachteil hierarchischer Verfahren liegt darin, daß die räumlichen Zusammenhänge in der Szene in hierarchischen Strukturen verloren gehen und nur unter relativ großem Aufwand rekonstruiert werden können: um von einer Zelle zu einer benachbarten zu gelangen ist es unter Umständen nötig, in der hierarchischen Struktur eine Vielzahl vertikaler Sprünge auszuführen.

Es soll eine Codierung für hierarchische Strukturen vorgestellt werden, die es erlaubt, räumliche Zusammenhänge einfach wiederherzustellen. Durch eine Nachbearbeitung des hierarchischen Rasters wird es möglich, die Vorteile der Baumstruktur zu nutzen und dabei die Informationen über räumliche Zusammenhänge in der Szene zu bewahren. Weiterhin wird anhand einiger Standardalgorithmen gezeigt, daß die vorgestellte Datenstruktur auch in der Praxis Vorteile bietet.

Accuracy and Degeneracy in Geometric Computation

Peter Schorn

Institut für Theoretische Informatik, ETH, CH-8092 Zürich

Achieving accurate results with imprecise primitives is an important goal in geometric computation. The example of the maximum function with an approximate comparison operation shows that the treatment of degeneracies heavily influences the accuracy of the computation. We present a divide & conquer algorithm which achieves the optimal accuracy using a given primitive. We define the concept of consistency and show that consistency is necessary but not sufficient to guarantee accurate results. Using examples we show how and when pure floating point computation, rational arithmetic and interval arithmetic can be used to guarantee accurate results.

A General Theory of Visibility

Sven Schuierer

Institut für Informatik, Rheinstr. 10 -12, D-7800 Freiburg

In this talk we pursue the notion of a convexity space as a unifying framework for the treatment of various notions of convexity in the plane. In particular, we suggest how to capture the notion of visibility within the general framework of convexity spaces, and investigate the relationship between visibility, kernels and skulls.

Convexity spaces represent an abstraction of the structure of convex sets in Euclidian space. Leaving aside all topological concerns we require convex sets to be closed under intersection, thus capturing their lattice-theoretic and algebraic properties.

The main aim of abstract convexity is to provide a unifying framework that goes beyond the concept of convexity based on line segments as in real vector spaces. In fact, there is an astonishing variety of "non-standard" notions of convexity in the plane that have been considered in computational geometry in the past few years: *restricted orientation convexity*, *NESW-convexity*, *rectangular convexity* and *geodesic convexity*, to name the most prominent ones.

There is a close relationship between questions concerning visibility and those concerning convexity. In most convexity spaces the definition of a convex set is based on the property that all points of a convex set can "see" each other. This leads to interesting questions and results about the relationship between maximal convex subsets which we call *skulls* and the kernel of a set. In this talk we prove the Kernel Theorem and the Cover Kernel Theorem, both of which relate kernels and skulls.

Voronoi-Diagramme für Teilmengen des 3D und ihre Anwendung auf Kollisionsprobleme bei Robotern.

Sabine Stifter

Research Institute for Symbolic Computation
J. Kepler Universität, A-4040 Linz

Voronoi-Diagramme wurden zuerst für endliche Punktmenzen \mathcal{A} der Ebene eingeführt, [Voronoi 1908]. Später wurden sie verallgemeinert für endliche Punktmenzen in höheren Dimensionen, z.B. [Edelsbrunner, Seidel 1986], und andere Mengen \mathcal{A} als nur Punktmenzen, z.B. [Yap 1987]. Im letzteren Fall wird das Voronoi-Diagramm von \mathcal{A} üblicherweise als das Voronoi-Diagramm des Komplements \mathcal{F} von \mathcal{A} bezeichnet. Es gibt aber kaum Ansätze, die Nicht-Punktmenzen in 3 und mehr Dimensionen betrachten.

Verallgemeinerungen in 3 und mehr Dimensionen sind von großer Bedeutung für anspruchsvolle geometrische/technische Anwendungen. Zum Beispiel im Bereich Robotics kann das Voronoi-Diagramm einer Menge \mathcal{F} verwendet werden, um einen kollisionsfreien Pfad für einen Punkt (oder eine Kugel) durch \mathcal{F} zu finden. Dazu muß das Voronoi-Diagramm eindimensional sein, und der folgende Satz gelten:

Für jedes p, q im Voronoi-Diagramm von \mathcal{F} : Wenn es eine zusammenhängende Kurve von p nach q in \mathcal{F} gibt, dann gibt es auch eine im Voronoi-Diagramm; für alle $p \in \mathcal{F}$ existiert eine zusammenhängende Kurve in \mathcal{F} von p zum Voronoi-Diagramm von \mathcal{F} .

Unser Ansatz zu Voronoi-Diagrammen ist verschieden von und allgemeiner als alle bekannten Ansätze. Um größtmögliche Allgemeinheit zu erreichen, definieren wir Voronoi-Diagramme für offene, zusammenhängenden Mengen im 3D, die einige zusätzliche Eigenschaften haben. Grob gesagt, muß eine endliche Menge $\text{contour}(\mathcal{F})$ mit folgenden Eigenschaften existieren:

- 1) $\text{contour}(\mathcal{F})$ ist eine Partition vom Rand von \mathcal{F} .
- 2) Bewegt man sich entlang einer Normalen weg von einem Element in $\text{contour}(\mathcal{F})$, so wächst der Abstand zu diesem Element.
- 3) Flächen mit gleichem Abstand zu zwei Elementen in $\text{contour}(\mathcal{F})$ erreichen maximale Distanzen zu diesen Elementen nur am Rand.
- 4) Jeder Punkt in \mathcal{F} ist eindeutig projizierbar auf jedes Element in $\text{contour}(\mathcal{F})$.
- 5) Alle Elemente in $\text{contour}(\mathcal{F})$ sind stetig differenzierbar.

Das verallgemeinerte Voronoi-Diagramm einer solchen Menge \mathcal{F} ist die Menge der Punkte, die gleichen Abstand zu drei Elementen in $\text{contour}(\mathcal{F})$ haben; man muß allerdings mit der Abstandsdefinition an den Rändern vorsichtig sein. Wir zeigen, daß so definierte Voronoi-Diagramme eindimensional sind und zur Konstruktion kollisionsfreier Pfade geeignet sind, d.h. daß obiger Satz gilt. Für Details wollen wir auf [Stifter 1988] verweisen.

Edelsbrunner, H., Seidel, R., 1986: Voronoi-Diagrams and Arrangements; Discrete and Computational Geometry, vol. 1, pp. 25-44.

Stifter, S., 1988: A Medley of Solutions to the Robot Collision Problem in Two and Three Dimensions; Ph.D. thesis, University Linz, Austria, RISC-Linz series 88-12. Also: VWGÖ, Dissertationen der Universität Linz, Bd. 81, 1989.

Voronoi, G., 1908: Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques; Deuxieme Memoire: Recherches sur les Paralleleodres Primitifs; Journal Reine und Angewandte Mathematik, vol. 134, pp. 198-287.

Yap, Ch.K., 1987: An $O(n \log n)$ Algorithm for the Voronoi-Diagram of a Set of Simple Curve Segments; Discrete and Computational Geometry, vol. 2, pp. 365-393.

Computational Complexity of Combinatorial Surfaces

Gert Vegter
Dept. of Computing Science,
University of Groningen,
P.O.Box 800, 9700 AV Groningen,
The Netherlands

(Abstract)

The principal problem in the topology of closed surfaces is the search for the topological invariants of closed surfaces so that we can tell if two arbitrarily given closed surfaces are or are not homeomorphic.

It follows then, that the principal *computational* problem is finding efficient methods to compute these invariants and for two closed homeomorphic surfaces, to *construct* homeomorphisms between them. In this talk we present such algorithms.

We show that, given a triangulated orientable closed surface, we can construct a canonical set of $2g$ curves on the surface such that cutting the surface along these curves yields a triangulated polygonal disc. For an orientable surface of genus g we construct such a canonical set in $O(n \log n + g^2 n)$ time, where n is the total number of vertices, edges and faces of the triangulation. These curves are obtained as edge-paths of a triangulation of the surface of size $O(g^2 n)$, that is obtained by subdivision of the original triangulation.

The method can be applied to solve topological problems in the realm of combinatorial surfaces, like

- obtaining a homeomorphism between closed orientable surfaces of the same genus;
- deciding whether two closed curves on a surface are homotopic and, if so, constructing a homotopy between them.

This is joint work with Chee Yap (New York University and Free University Berlin).