STUDIEN

Effiziente Algorithmen

für Studenten der Mathematik und Informatik
an der Freien Universität Berlin

Semesterheft Sommer 2002

SCHWERPUNKT

Allgemeines

Das Gebiet *Effiziente Algorithmen* ist eines der Bindeglieder zwischen Informatik und Mathematik. Einerseits gehören Algorithmen und Datenstrukturen zum Kern der praktischen Informatik, andererseits bezieht die zugrundeliegende Theorie ihre Methoden im wesentlichen aus der diskreten Mathematik. Die Anwendungen reichen in zahlreiche Gebiete wie Computer–Grafik, Mustererkennung, Robotik, Computer Aided Design, Bioinformatik, Kartographie, Bildverarbeitung usw. Einige konkrete Beispiele werden anhand von Diplomarbeiten, Dissertationen und Projekten am Ende dieser Broschüre vorgestellt.

Das Gebiet ist in Berlin an allen drei Universitäten und am Konrad-Zuse-Zentrum stark vertreten. Diese Institutionen tragen gemeinsam das von der Deutschen Forschungsgemeinschaft geförderte Europäische Graduiertenkolleg *Combinatorics, Geometry, and Computation*, das in Zusammenarbeit mit der ETH Zürich durchgeführt wird. (Siehe auch die WWW-Seite:
`http://www.inf.fu-berlin.de/gk-cgc`.)
Neben zahlreichen Lehrveranstaltungen werden auch andere Möglichkeiten zur Weiterbildung angeboten. So finden zum Beispiel der Berliner Algorithmen-Tag oder mehrtägige Spezialschulen regelmäßig statt.

Der Studienschwerpunkt ist ein Vorschlag zur Vertiefung in dieses Fachgebiet im Rahmen der Studiengänge Mathematik und Informatik an der Freien Universität. Zum Beispiel wird eine abgestimmte Folge von Lehrveranstaltungen von den Grundlagen bis zu den Anwendungen angeboten. Darüberhinaus sollen den Studenten die zahlreichen Angebote in dieser Richtung in Berlin besser zugänglich gemacht werden. Dazu wird diese Informationsbroschüre jedes Semester aktualisiert. Neben Vorschlägen zur Studienplanung werden hier allgemeinere Informationen zum Gebiet *Effiziente Algorithmen* zusammengefasst. Die Broschüre gibt einen Überblick über die Lehrveranstaltungen zum Gebiet – auch an den anderen Berliner Universitäten – und die Lehrveranstaltungsplanung für die folgenden Semester. Zusätzlich gibt sie Informationen zu Tagungen und ähnlichen Veranstaltungen, zu den in der Arbeitsgruppe *Effiziente Algorithmen* tätigen Mitarbeitern und ihren Arbeitsgebieten sowie zu aktuell im Fachbereich behandelten Forschungsthemen (Diplomarbeiten, Dissertationen etc.).

Interessenten können sich im Sekretariat der Arbeitsgruppe Theoretische Informatik (Takustr. 9, Raum 111) in den Studienschwerpunkt einschreiben. Sie erhalten dann regelmäßig das Semesterheft und werden laufend über Veranstaltungen wie etwa Vorträge, Spezialschulen und Tagungen informiert.

1 Vorschlag zur Studienplanung

Für die Teilnahme am Studienschwerpunkt sind gewisse Grundkenntnisse aus dem Grundstudium unerlässlich. Es wird empfohlen, Vorlesungen und Kurse zu Kombinatorik, Wahrscheinlichkeitstheorie, Graphentheorie, Algorithmen und Datenstrukturen sowie Programmierung frühzeitig zu besuchen. Innerhalb des Hauptstudiums ist sowohl eine Orientierung in mathematisch-theoretischer Richtung als auch eine zu Anwendungsgebieten hin möglich. Im folgenden geben wir Empfehlungen zur Organisation des Studiums für beide Richtungen. Natürlich sind auch Mischformen und eine andere Organisation möglich.

Die Angebote des Hauptstudiums werden im Studienschwerpunkt mit einem Kürzel ([EA 1,2] Entwurf und Analyse von Algorithmen, [ADM] Angewandte Diskrete Mathematik, [ANW] Anwendungen, [PR] Praktikum) gekennzeichnet. Das Kürzel zeigt an, für welche Phase des Studiums die Veranstaltungen geeignet sind. Der Vorschlag sollte individuell durch begleitende Lehrveranstaltungen aus Mathematik und Informatik ergänzt werden.

Vertiefung in theoretischer Richtung.

[**EA1**] (3V + 2 Ü) im 5. Semester
Entwurf und Analyse von Algorithmen.

[**EA2**] (3V + 2 Ü) im 6. Semester
Themenbeispiele: *Algorithmische Geometrie* oder *Graphenalgorithmen* oder *Optimierung* oder *Parallele Algorithmen* oder ...

[**ADM**] Zumindest eine Vorlesung und ein Seminar im 7. & 8. Semester.
Themenbeispiele: *Polytope* oder *Pattern Matching* oder *Randomisierte Algorithmen* oder *Graphenalgorithmen* oder ...

⋆ anschließend *Diplomarbeit.*

Vertiefung in anwendungsbezogener Richtung.

[EA1] (3V + 2 Ü) im 5. Semester
Entwurf und Analyse von Algorithmen.

[EA2] (3V + 2 Ü) im 6. Semester
Themenbeispiele: *Algorithmische Geometrie* oder *Graphenalgorithmen* oder *Optimierung* oder *Parallele Algorithmen* oder ...

[ANW] Zumindest eine Vorlesung und ein Seminar im 7. & 8. Semester. Themenbeispiele aus den Anwendungsgebieten: *Mustererkennung* oder *Computer–Grafik* oder *Computer Aided Design* oder *Robotik* oder *Bildverarbeitung* oder *Bioinformatik* oder ...

[PR] (4 PR) Praktikum im 8. Semester.

⋆ anschließend *Diplomarbeit.*

Entsprechend den vorgehenden Vorschlägen sollen Zyklen von einander ergänzenden Veranstaltungen angeboten werden. Am Beginn jedes Zyklus werden den behandelten Themen Kürzel zugewiesen und der Plan im Semesterheft angekündigt.
In diesem Sommersemester gibt es mehrere mögliche Fortsetzungen zur Vorlesung *“Entwurf und Analyse von Algorithmen”*. Zum einem bietet sich die Vorlesung *“Algorithmische Geometrie”* an, eine Einführung in das Hauptarbeitsgebiet der Arbeitsgruppe “Theoretische Informatik”, an die sich die Vergabe von Diplom- oder Studienarbeiten anschließem kann. Gleiches gilt für die anderen Veranstaltungen des Hauptstudiums. Weitere geometrische Themen werden in den Vorlesungen *kombinatorische Geometrie* und *Gittererzeugung* behandelt, die erste Vorlesung hat eher einen theoretischen umd kombinatorischen Schwerpunkt, die zweite ist auch für Anwendungen wichtig.
Die Vorlesung über *Komplexitätstheorie* gibt einen verbreiterten und vertiefenden Einblick in dieses Gebiet über den bekannten Begriff der NP-Vollständigkeit hinaus. Die Vorlesung über *kombinatorische Optimierung* stellt dieses für Anwendungen wichtige und algorithmisch reizvolle Gebiet vor.
Die Vorlesung *Computergrafik* behandelt ein wichtiges Anwendungsgebiet geometrischer Algorithmen.
Es werden auch zwei Seminare angeboten, an die sich Studien und Diplomarbeiten anschließen können. Zum einen handelt es sich um ein Seminar über *Graphenalgorithmen*, in dem Themen der gleichnamigen Vorlesung aus dem

Wintersemester vertieft werden, das aber auch anderen Interessenten offensteht. Weiter gibt es ein Seminar über *Algorithmen für Quantencomputer*.
Im Praktikum über *effiziente Algorithmen* gibt es Gelegenheit zu praktischer Arbeit im Zusammenhang mit den Forschungsthemen der Arbeitsgruppe.
In diesem Semester wird auch die Vorlesung über *Graphentheorie* für Studierende der Mathematik von unserer Arbeitgruppe abgedeckt.

Realisierung im Studienplan.

Die Realisierung innerhalb der bestehenden Studiengänge wird für Mathematik und Informatik getrennt behandelt.

Diplomstudiengang Mathematik.
Bei Teilnahme am Studienschwerpunkt wird Studenten der Mathematik die Belegung des Nebenfachs Informatik empfohlen.

- Grundstudium.

  Auf jeden Fall sollte die *Elementare Wahrscheinlichkeitstheorie* besucht werden. Ein *Programmierkurs* ist Pflicht im Grundstudium. Die Vorlesungen *Informatik A* und *B* sind Teil der Nebenfachausbildung. Grundkenntnisse in diskreter Mathematik können durch einen möglichst frühen Besuch von *Kombinatorik* und/oder *Graphentheorie* erworben werden.

- Hauptstudium.

  [EA1] Anrechnung in A (Angew. Mathematik).

  [EA2] & [ADM] Anrechnung in B (Einarbeitung in Spezialgebiet).

  [ANW] & [PR] Anrechnung im Nebenfach Informatik.

  Seminare: Anrechnung im Rahmen der mindestens 2 Seminare.

Bei der Diplomprüfung aus Angew. Mathematik wird neben Algorithmentheorie auch ein weiteres Gebiet (Wahrscheinlichkeitstheorie oder Numerik) geprüft.

Diplomstudiengang Informatik.

- Grundstudium.

  Die Grundkenntnisse werden durch die Lehrveranstaltungen *Algorithmen und Programmierung III* und *Mathematik für Informatiker* abgedeckt.

- Hauptstudium.

  [EA1] & [EA2] Anrechnung in Theoretische Informatik.
  [ADM] & [ANW] & [PR] Anrechnung im Rahmen der 14 SWS im Studienschwerpunkt.
  Seminare: Anrechnung im Rahmen der mindestens 2 Seminare.

2 Lehrveranstaltungen im Sommer 2002

Vorlesungen

Algorithmische Geometrie [EA2]

Dozent: Alt; Vorlesungszeit: Di, Do 10–12 Uhr, 4-stündig,
Veranstaltungsort: Takustraße 9, SR 055.
Übungen Alt, Knauer, 2-stündig .
Beginn: 16.04.2002

INHALT: Generell effiziente Algorithmen für geometrische Probleme. Zum Beispiel Konstruktion der konvexen Hülle einer Menge von Punkten in der Ebene oder in drei Dimensionen; Konstruktion des Voronoi-Diagramms einer Menge von Punkten; geometrische Datenstrukturen, etwa um bei einer Unterteilung der Ebene effizient das Gebiet finden zu können, in das ein gegebener Punkt fällt. Allgemeine Techniken zum Entwurf effizienter geometrischer Algorithmen. Geometrische Algorithmen sind wegen ihrer Anwendungen z.B. in der Computer-Grafik, des CAD, der geographischen Informationssysteme, usw. wichtig.

Literatur: Preparata, F. P., and M. I. Shamos. Computational Geometry: An Introduction, New York: Springer, 1985.
Rolf Klein. "Algorithmische Geometrie", Addison-Wesley, 1997
J.-D. Boissonnat and M. Yvinec. Algorithmic Geometry. Cambridge University Press, UK, 1998.
M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. Computational Geometry - Algorithms and Applications" Zweite Auflage, Springer, 2000

Komplexitätstheorie [EA2]

Dozent: Knauer; Vorlesungszeit: Do 16–18 Uhr, 2-stündig,
Veranstaltungsort: Takustraße 9, SR 049.
Übungen Knauer, 2-stündig.
Beginn: 18.04.2002

INHALT: Die Komplexität eines Berechnungsproblems ist der Aufwand zu dessen algorithmischer Lösung in einem geeigneten Maschinenmodell, z.B.

dem Turing-Maschinenmodell. Sie wird üblicherweise durch den Zeit- oder Speicherbedarf eines Algorithmus, der dieses Problem auf einer entsprechenden Maschine löst, gemessen. Die Komplexitätstheorie untersucht systematisch die Komplexität von Berechnungsproblemen. Ziel ist es, diese in Komplexitätsklassen einzuordnen. Dazu versucht man einerseits effiziente Algorithmen für ein Problem zu finden, um eine obere Schranke für dessen Komplexität zu erhalten. Andererseits möchte man auch zeigen, dass zur Lösung eines Problems mindestens ein gewisser Aufwand erforderlich ist, also eine untere Schranke für dessen Komplexität finden. Ziel dieser Vorlesung ist es, nach einer kurzen zusammenfassenden Darstellung der Grundlagen auch auf neuere Resultate und Methoden einzugehen. Diese umfassen Fragestellungen die in den Standardvorlesungen normalerweise nicht behandelt werden:

- Gibt es Probleme für die man leicht die Korrektheit einer Lösung überprüfen kann, aber für die es beweisbar schwer ist, eine solche Lösung schnell zu berechnen? (Stichwort: P-NP Problematik, Exponentielle untere Schranken für die Größe monotoner Schaltkreise für das Cliquenproblem)
- Sind schwere Probleme auch zu etwas nützlich? (Stichwort: Kryptographie, Einwegfunktionen, Zero-Knowledge Beweise)
- Hilft der Zufall bei der Lösung von Problemen wirklich? Wenn ja, wieviel Zufall ist dazu notwendig? (Stichwort: Randomisierte Komplexitätsklassen, Recycling von Zufallszahlen, PCP-Theorem)
- Wie kann man Zufall quantitativ erfassen, was ist Zufall überhaupt? (Stichwort: Kolmogorov-Komplexität)

Kombinatorische Optimierung **[EA2,ADM]**

Dozent: Rote; Vorlesungszeit: Mi 16–18 Uhr, Do 14–16 Uhr, 3-stündig,
Veranstaltungsort: Takustraße 9, Mo: HS, Do: SR 055.
Übungen Rote, Meißner, 2-stündig.
Beginn: 17.04.2002

INHALT: In der kombinatorischen Optimierung wird aus einer endlichen Menge von Möglichkeiten die beste ausgewählt. Die Grundmenge hat dabei eine kombinatorische Struktur, zum Beispiel die Bäume oder die Färbungen eines Graphen. Kombinatorische Optimierungsprobleme treten in vielen Anwendungen

auf, zum Beispiel im Entwurf von Kommunikationsnetzen, bei Standortproblemen, oder bei der Frequenzzuteilung von Mobilfunknetzen. Die Vorlesung ist sowohl algorithmisch als auch theoretisch orientiert, weil die Algorithmen die Eigenschaften der zugrundeliegenden kombinatorischen Strukturen ausnützen. Die Vorlesung ergänzt die vorangegangene Vorlesung über Graphenalgorithmen, weil viele kombinatorischen Optimierungsaufgaben mit Graphen zu tun haben. Ebenso vertieft sie die Vorlesung über Entwurf und Analyse von Algorithmen in eine spezifische Richtung.
Inhaltsübersicht:

1. Grundlagen aus der linearen Optimierung
2. Matroide und der Greedy-Algorithmus
3. submodulare Funktionen
4. der primalduale Ansatz für kombinatorische Optimierungsprobleme
5. kürzeste Wege, Flüsse, Paarungen
6. vollständige Unimodularität
7. ganzzahlige Optimierung

Literatur:Bernhard Korte, Jens Vygen: Combinatorial Optimization: Theory and Algorithms. Springer, Berlin Heidelberg New York 2000.
C. H. Papadimitriou, K. Steiglitz: Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, A. Schrijver: Combinatorial Optimization, 1998, Wiley, New York .
Alexander Schrijver, Theory of Linear and Integer Programming, Wiley (Chichester, 1986).

Computergrafik **[ANW]**

Dozentin: Morin; Vorlesungszeit: Mi, Fr 12–14 Uhr, 3-stündig.
Veranstaltungsort: Takustraße 9, SR 051.
Übungen Morin, 2stündig.
Beginn: 17.04.2002

INHALT: 2 Dimensional Graphics: Fractal: Drawing (turtle geometry, Iterated function system), Dimension of self similar curves. Free-form curves: Parametric (Polynomials, B-splines, NURBS), Implicit and Subdivision curves.
3 Dimensional Graphics : 3D Geometric Transformations. Rendering: Viewing, hidden surfaces algorithms, shading algorithms, color. Free-form surface modeling: Parametric (Tensor product surfaces, triangular patches), Implicit and Subdivision surfaces. Programming and written assignments.

Literatur:Foley/vanDam/Feiner/Hughes: Computer Graphics: Principles and Practice.
Farin: Curves and Surfaces for Computer Aided Geometric Design
Warren/Weimer: Subdivision Methods for Geometric Design: A Constructive Approach.
Lectures will be given in English.

Gittererzeugung **[ADM,ANW]**

Dozent: Braß; Vorlesungszeit: Mo, 10.00–12.00 Uhr, 2-stündig.
Veranstaltungsort: Takustraße 9, SR 055.
Übungen Braß, Meißner, 2-stündig.

INHALT: Bei vielen praktischen Problemen ist es notwendig, ein großes Gebiet oder Volumen zu unterteilen, oder eine Fläche zu triangulieren. Hiermit beschäftigt sich die Gittererzeugung ("mesh generation"); sie ist ein notwendiger Vorverarbeitungsschritt für geometrische Objekte in der Numerik, der Computergrafik, und vielen anderen Bereichen. Dabei sollen die zu erzeugenden Triangulierungen viele spezielle Eigenschaften haben, etwa nur die gegebenen (oder wenige zusätzliche) Punkte verwenden, keine spitzen Dreiecke enthalten, keine Dreiecke besonders großer oder besonders kleiner Fläche enthalten und vieles andere. Daher ist die Erzeugung solcher Triangulierungen keineswegs trivial, und die Erzeugung einer "guten" Triangulierung ist ein notwendiger erster Schritt für gute Ergebnisse in den darauf aufbauenden Algorithmen. Wir werden in dieser Vorlesung Eigenschaften von Triangulierungen und Algorithmen zu ihrer Konstruktion behandeln, und einige dieser Algorithmen in den Übungen implementieren.

Literatur: Herbert Edelsbrunner: Geometry and Topology for Mesh Generation. Cambridge University Press 2001.

Kombinatorische Geometrie [ADM]

Dozent: Kortenkamp; Vorlesungszeit: Mi, Fr 10–12 Uhr, 4-stündig.
Veranstaltungsort: Takustraße 9, SR 055.
Beginn: 17.04.2002

INHALT: In dieser Veranstaltung werde ich eine Reihe offener Probleme im Bereich der Diskreten Geometrie vorstellen, die unter anderem Polygone, Polyeder, Zellzerlegungen, Packungen, Überdeckungen, Gitterpunkte und ähnliche Objekte betreffen. Falls Interesse besteht, werde ich auch eine Arbeitsgruppe anbieten, in der wir gemeinsam an einigen dieser Probleme arbeiten werden. Damit wird exemplarisch eine Einführung in die wissenschaftliche Arbeit gegeben.

Graphentheorie [ADM]

Dozent: Felsner; Vorlesungszeit: Di, Do 12–14 Uhr, 4-stündig.
Veranstaltungsort: Arnimallee 2-6, SR 007/008.

INHALT: Graphentheorie ist neben der "Einführung in die Diskrete Mathematik" die zweite Grundvorlesung im Bereich Diskrete Mathematik. Viele kombinatorische Aufgaben, Optimierungsaufgaben und algorithmische Probleme lassen sich graphentheoretisch formulieren.
In der Vorlesung werden die grundlegenden Konzepte wie Zusammenhang, Flüsse, Matchings, Touren, Planarität, Färbung, vorgestellt. Wo es sich ergibt, wird der algorithmische Aspekt hervorgehoben. An verschiedenen Stellen werden auch speziellere und aktuelle Themen behandelt.

Literatur: Aigner: Graphentheorie
Diestel: Graphentheorie

West: Introduction to Graph Theory.

Seminare, Praktika und sonstige Veranstaltungen

Seminar über Algorithmen für Quanten-Computer [EA2]

Dozent: Alt; Vorlesungszeit: n.V. , 2-stündig.
Veranstaltungsort: Takustraße 9, SR 049.
Vorbesprechung: Dienstag, 16.04., 16 Uhr c.t., SR 049

INHALT: Die Idee von Quanten-Computern erweitert unser bisheriges Rechnermodell erheblich. Während beim klassischen Modell nur auf einer Menge von Zahlen (Daten) gearbeitet werden kann, kann ein Quanten-Computer dies auf vielen gleichzeitig (und reversible), wobei diese verschiedenen Rechnungen auch überlagert werden können. Dadurch erhält man möglicherweise effiziente Algorithmen für Probleme, die bisher als nicht effizient lösbar galten. Bestes Beispiel ist die in der Kryptographie häufig eingesetzte Faktorisierung von Zahlen. Literatur: M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2001.
J. Gruska. Quantum Computing. McGraw-Hill, 1999.

Seminar über Graphenalgorithmen [ADM]

Dozent: Rote; Vorlesungszeit: Di 14–16 Uhr, 2-stündig.
Veranstaltungsort: Takustraße 9, SR 053
Vorbesprechung: Dienstag, 16.04.2002

INHALT: In diesem Seminar werden einige Themen aus meiner Vorlesung über Graphenalgorithmen im Wintersemester vertieft. Die Teilnehmerinnen halten auf der Grundlage von Spezialarbeiten einen Vortrag und fertigen eine kurze schriftliche Ausarbeitung an.
Einige der Themen sind:

1. neue (andere?) Algorithmen zum Planaritätstest;
2. Einbetten und Zeichnen von Graphen;
3. Zerlegung in dreifache Zusammenhangskomponenten;

4. dreifacher Zusammenhang mit PQ-Bäumen;
5. Zerlegung von Graphen mit höherem Zusammenhang;
6. Konstruktion einer Kontraktionsfolge für dreifach zusammenhängende Graphen;
7. schnelle kürzeste-Wege-Algorithmen;
8. Flüsse mit minimalen Kosten.

Weitere Informationen zu den Themen gibt es unter:
`http://www.inf.fu-berlin.de/~rote/Lere/2002-SS/`.

Diplomanden- und Doktorandenseminar
Theoretische Informatik **[EA2, ADM]**

Dozent: Alt, Rote, Felsner, Braß, Kriegel;
Vorlesungszeit: Di, Do, Fr 12–13 Uhr;
Veranstaltungsort: Takustraße 9, SR 055;

INHALT: Vorträge über eigene Forschung und Originalarbeiten aus der Theoretischen Informatik, insbesondere Algorithmen. Die Ankündigungen werden jeweils gesondert gegenüber Raum 111 in der Takustraße 9 ausgehängt.

Praktikum Effiziente Algorithmen **[PR]**

Dozent: Rote, Kortenkamp; Vorlesungszeit: Di 16–19 Uhr, 3-stündig;
Veranstaltungsort: Takustr. 9, SR 046;
Vorbesprechung: Dienstag, 16.04., 16 Uhr c.t.

INHALT: In Einzel- oder Gruppenprojekten soll die Anwendung von effizienten Algorithmen zur Realisierung praktischer Fragestellungen geübt werden. Dabei können sowohl eigene Programme geschrieben werden als auch vorhandene Software eingesetzt werden, wo es sich ergibt. Die Themen kommen aus der Forschung am Institut (im Bereich der Kombinatorik, der Optimierung oder der Geometrie) oder aus Anwendungen.
Ablauf: Zu Beginn werden die Themen kurz vorgestellt und ausgegeben. Die

Themen können einzeln oder in Gruppen zu zweit oder zu dritt bearbeitet werden. Nach einer Einarbeitungszeit wird das Ziel der Projekts und der Umfang für jedes Projekt im Einvernehmen mit den Betreuern genau festgelegt. Die Teilnehmer am Praktikum treffen sich regelmäßig und berichten über den Fortgang des Projektes, über Teilergebnisse oder über Schwierigkeiten. Am Ende werden die Ergebnisse des Praktikums präsentiert.
Themen:

- Auseinanderfalten von Polygonen
- Kanonische Pseudotriangulierungen
- Realisierung orientierbarer Matroide
- Das Assoziaeder
- Wege in Chirotopen
- Optimale Zufallsstrategien für Suchprobleme
- Der Expansionskegel in höheren Dimensionen
- Realisierung von dreidimensionalen Polytopen mit kleinen Koordinaten
- Optimierung von Roboterbahnen

Weitere Informationen zu den Themen gibt es unter:
`http://www.inf.fu-berlin.de/~rote/Lere/2002-SS/`.

Vorlesung des Europäischen Graduiertenkollegs
Combinatorics, Geometry and Computation **[ADM]**

Dozent: Alt, Rote u. Doz. des Kollegs, Vorlesungszeit: Mo 14–16 Uhr, 2-stündig;
Veranstaltungsort: Takustraße 9, SR 005.

INHALT: Die Dozenten und Gäste des Kollegs halten einführende Vorlesungen (in Blöcken von etwa 2–4 Stunden) zu speziellen Themen des Kollegs. Dazu gehören insbesondere algorithmische und diskrete Geometrie, algorithmische Kombinatorik, Codierungstheorie, Graphentheorie und Graphenalgorithmen, Gruppentheorie, kombinatorische Optimierung, konstruktive Approximation, Mustererkennung und zufällige diskrete Strukturen. Die Themen der Vorlesungen werden auch neben Raum 111 in der Takustraße 9 ausgehängt. (Interessenten können sich bei der Koordinatorin des Kollegs, Frau B. Felsner, auf einen Verteiler für das Verschicken der Ankündigungen setzen lassen.)

Kolloquium des Europäischen Graduiertenkollegs
Combinatorics, Geometry, and Computation [ADM]

Dozent: Alt u. Doz. des Kollegs; Vorlesungszeit: Mo 16–18 Uhr, 2stündig; Veranstaltungsort: Takustraße 9, SR 005;

INHALT: Mitglieder und Dozenten des Kollegs sowie Gäste halten Vorträge zum Thema des Kollegs. Die einzelnen Vorträge im Kolloquium werden gesondert angekündigt. (Interessenten können sich bei der Koordinatorin des Kollegt, Frau B. Felsner, auf einen Verteiler für das Verschicken der Ankündigungen setzen lassen.) Die Ankündigungen werden auch neben Raum 111 in der Takustraße 9 ausgehängt.

Die Vorlesungen des Graduiertenkollegs im Sommer 2002

Die Vorlesungen des Graduiertenkollegs werden durch Aushang an den einzelnen Universitäten (Fachbereiche und Arbeitsgruppen der Dozenten), neben Raum 111 in der Takustraße 9, sowie im Internet unter:
http://www.inf.fu-berlin.de/graduate-programs/cgc/ angekündigt.

Die Kolloquiumsvorträge des Graduiertenkollegs im Sommer 2002

Die Kolloquien des Graduiertenkollegs werden durch Aushang an den einzelnen Universitäten (Fachbereiche und Arbeitsgruppen der Dozenten), neben Raum 111 in der Takustraße 9, sowie im Internet unter:
http://www.inf.fu-berlin.de/graduate-programs/cgc/ angekündigt.

Weitere Veranstaltungen an der Freien Universität

- Ausgewählte mathematische Probleme (mit besonderer Berücksichtigung von Paul Erdős) (VL); Dozent: Lenz.
- Computeralgebra (VL); Dozent: Altmann.

3 Diplomarbeiten, Dissertationen, Projekte

Die von der Arbeitsgruppe Theoretische Informatik behandelten Forschungsthemen werden an Diplomarbeiten, Dissertationen und Projekten beispielhaft vorgestellt.

Diplomarbeit: Hausdorff-Abstand und Fréchet-Abstand von Spline-Kurven

Diplomandin: *Ludmila Scharf*, Betreuer: *Helmut Alt*

In der Arbeit geht es um Berechnen von Ähnlichkeiten zwischen durch Kurven modellierte Figuren in der Ebene. Als Ähnlichkeitsmaße werden der sogenannte Hausdorff-Abstandund der Fréchet-Abstandbetrachtet.

Zunächst sollen die Algorithmen für algebraische Kurven $(t \to (p_1(t), p_2(t)))$ erarbeitet und implementiert werden. Als zweiter Schritt soll die Verallgemeinerung auf stückweise algebraische Kurven, insbesondere Splines erfolgen.

Die Abstandsfunktionen finden ihre Anwendung als Qualitätsfunktion in Matching-Algorithmen, die meisten basieren auf dem Hausdorff-Abstand. Viele Arbeiten beschäftigen sich mit dem Hausdorff-Abstand für Polygone und er hat sich gut in der Praxis bewährt. Es gibt jedoch Fälle wo der Fréchet-Abstand ein besseres Kriterium dafür ist, wie ähnlich sich zwei Muster oder zwei Kurven sind.

Mögliche Anwendungsgebiete: Computergraphik, Mustererkennung, Kartographie.

Diplomarbeit: Implementieren von parametrisierten Kurven für CGAL

Diplomandin: *Ekaterina Langer*, Betreuer: *Helmut Alt*

CGAL ist eine C++ Bibliothek für Objekte und Datenstrukturen der algorithmischen Geometrie, deren Entwurf und Implementierung die Konzepte der generischen Programmierung zu Grunde liegen. Ziel der Diplomarbeit ist die Implementierung von parametrischen Kurven für die CGAL-Bibliothek.
Die parametrische Kurve ist eine Funktion $f : [a,b] \to \mathbb{R}^2$, die zu einem gegebenen Parameter t den entsprechenden Punkt in der 2D Ebene liefert. Um die Kurve allgemein zu halten und dem Benutzer die Möglichkeit zu geben, eigene Typen von Kurven zu definieren, werden dem Datentyp parametrische Kurve die benutzerdefinierte Funktionen $x(t)$ und $y(t)$ übergeben, die die Koordinaten eines Punktes auf der Kurve berechnen. Z.B. ensteht durch die Parametrisierung mit $x(t) = O_x + (R-r)\cos(t) + a\cos(\frac{R-r}{r}t)$ und $y(t) = O_y + (R-r)\sin(t) - a\sin(\frac{R-r}{r}t)$ eine Astroide (Abb.1).

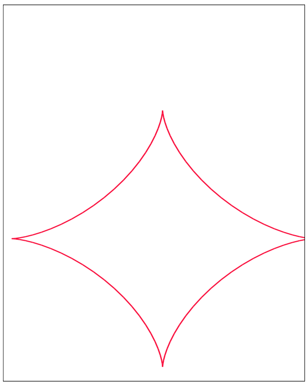


Abbildung 1: Astroide

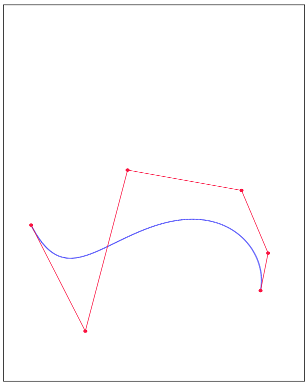


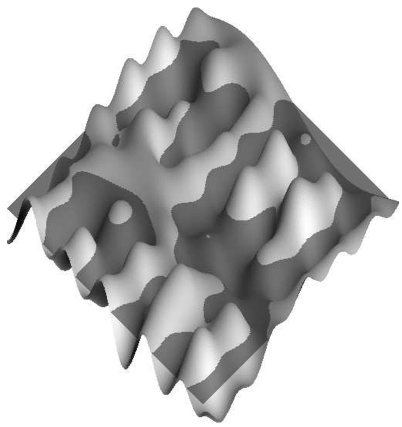
Abbildung 2: Bézier-Kurve

Neben der Funktion zum Berechnen eines Punktes auf der Kurve zu gegebenem Parameter, verfügt die Kurve auf Methoden zum Zeichnen, Bestimmen der Schnittpunkte zweier Kurven, ob ein Punkt auf der Kurve liegt, Berechnen der n. Ableitung und des stückweise linearen Approximants. Als Untertypen der parametrischen Kurve sind die Bézier-Kurven (Abb.2) und die B-Splines implementiert.

Diplomarbeit: Effiziente Konstruktion von Kontourbäumen in beliebigen Dimensionen

Diplomand: *Tobias Lenz*, Betreuer: *Günter Rote*

In vielen wissenschaftlichen Gebieten spielt die Visualisierung von Daten eine zunehmende Rolle. Dabei werden Werte an sehr vielen fixen Positionen gemessen, z.B. die Höhe über dem Meeresspiegel für einen bestimmten Landstrich, aus dem Körper austretende elektromagnetische Wellen in einem Kernspinresonanztomographen oder Hitze in einer Brennkammer. Die Daten liegen als Paare von Punkten in einer bestimmten Dimension und den dazugehörigen Messwerten vor und ihre Anzahl kann bei sehr detailierten Messungen durchaus Größenordnungen von $500^4 = 62,5$ Mrd. annehmen. Derartige Datenmengen können nicht in Echtzeit durchsucht werden, so dass man geeignete Datenstrukturen verwenden muss, um effizient bestimmte Teilmengen zu erhalten. Eine wichtige Teilmenge ist hierbei die Menge aller Punkte, die einen bestimmten Wert haben - sogenannte Isolinien bzw. Isoflächen oder auch Kontouren. Die Abbildung zeigt einen zweidimensionalen Datensatz, als Gebirge dargestellt, und eine Isolinie als Wasserspiegel einer bestimmten Höhe.



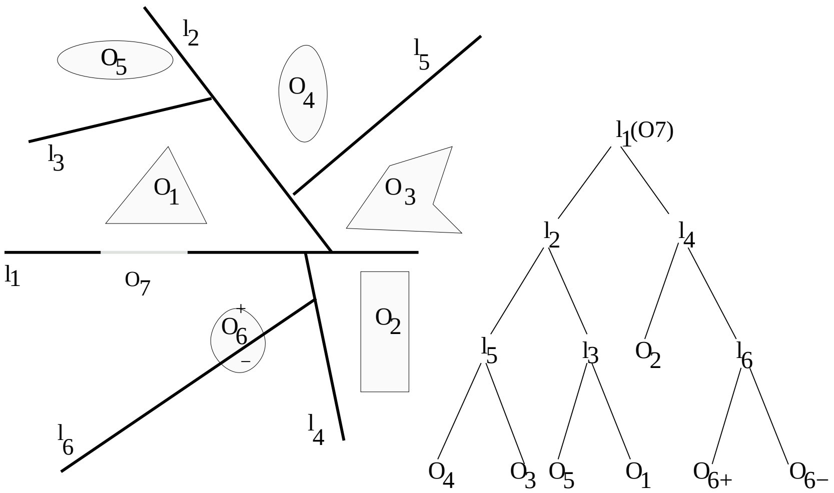
Bei der Erstellung einer Datenstruktur, die ein schnelles Zugreifen auf die Kontouren erlaubt, spielt der Kontourbaum eine wichtige Rolle. Er speichert alle relevanten "Ereignisse", die zu Änderungen der Kontouren führen und erlaubt das Erstellen minimaler sogenannter *seed sets*, aus denen dann eine Kontour effizient rekonstruiert wird.

Obwohl die Definition und die Eigenschaften des Baumes eine einfache Sweep-Technik zu dessen Erstellung implizieren, so kann doch unter bestimmten Annahmen über die gegebenen Daten ein Algorithmus, der monotone Wege läuft anstatt über alle Punkte zu streichen, in der Praxis eine beachtlich höhere Geschwindigkeit erzielen. Dieses Verfahren wird entwickelt und seine Laufzeit unter verschiedenen Bedingungen analysiert.

Diplomarbeit: Binäre Zerlegungen der Ebene

Diplomand: *Martin Mielich*, Betreuer: *Stefan Felsner*.

Es wird das Problem der Vergabe von Prioritäten betrachtet, das in Verbindung mit der Berechnung sichtbarer Objekte in der Computergrafik auftritt. Die darzustellenden Polygone werden so in Teilstücke zerlegt, dass es für beliebige Beobachterstandorte keinen überlappenden Zyklus gibt. Die resultierende Menge von Fragmenten sollte dabei so klein wie möglich sein und in einer Datenstruktur organisiert, die das Ziel der schnellen und korrekten Darstellung unterstützt.



Für eine Binary Space Partition (BSP) wird der Raum rekursiv entlang einer Hyperebene in je zwei Teilräume zerlegt, bis jeder Teilraum höchstens ein Objekt enthält. Im induzierten Binärbaum werden die Fragmente der Polygone verwaltet. Die Nutzung des Binärbaumes löst das angesprochene Problem der Prioritätenvergabe: Die Darstellung der Polygone entsprechend ihrer Anordnung im Baum führt zu einer korrekten Abbildung.

Jeder der Schnitte des Raumes zerteilt möglicherweise zahlreiche Polygone, die Größe des erzeugten Binärbaumes ist daher das wesentliche Maß für die

Effizienz der verwendeten Methode. Der Strategie der Teilung muss also besondere Aufmerksamkeit geschenkt werden, um eine unkontrollierte expansive Fragmentierung der Polygone zu verhindern.

In der Arbeit werden verschiedene Strategien der Erzeugung von BSP's in der Ebene vorgestellt. Mit bestimmten Forderungen an die Eingabemenge wird die Erzeugung von BSP's linearer Größe möglich, beispielsweise für orthogonale Segmente oder für Segmentmengen bei denen jedes Segment in der konvexen Hülle der Menge verankert ist.

Die lange bestehende Vermutung, dass für jede Eingabe im $\mathbb{R}^2$ eine BSP linearer Größe existiert konnte vor kurzem von David Csaba Tóth mittels einer Konstruktion widerlegt werden. Diese Arbeit wird in einem Kapitel vorgestellt.

Dissertation:

Doktorandin: *Britta Broser*, Betreuer: *Helmut Alt, Ulrich Kortenkamp*

Hinter den Kulissen der Geometriesoftware *Cinderella* verbirgt sich eine elegante mathematische Theorie, die sich aus verschiedenen Bereichen zusammensetzt. Aus ihr ergeben sich Fragen zwischen Komplexitätstheorie und Geometrie, die zum Teil noch ungelöst sind.
In *Cinderella* werden geometrische Konstruktionen durch geometrische Straight-Line Programme (GSP) repräsentiert. Diese setzen sich aus freien Punkten und abhängigen Elementen wie z. B.

- der Verbindungsgeraden zweier verschiedener Punkte,
- dem Schnittpunkt zweier verschiedener Geraden,
- einer der beiden Winkelhalbierenden zweier Geraden,
- einer der höchstens zwei Schnittpunkte einer Geraden mit einem Kreis

zusammen. Eine Instanz eines GSP ist eine Zuweisung von festen Werten zu allen freien Punkten und Wahlen. Ein GSP entspricht also einer formalen Konstruktionsbeschreibung und eine Instanz einer konkreten Zeichnung in der Ebene.

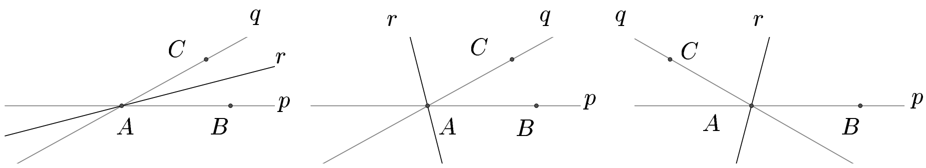


Abbildung 1: Drei verschiedene Instanzen des GSPs aus dem Beispiel

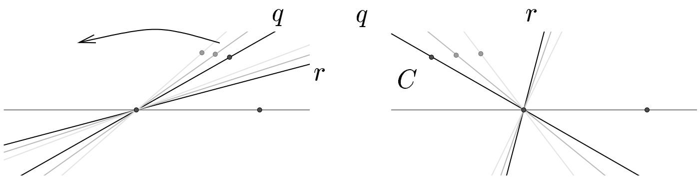


Abbildung 2: Die linke Instanz aus Abb. 1 kann „stetig" in die rechte überführt werden.

Beispiel für ein GSP:

$$\begin{array}{lcll}
A & \leftarrow & FREE & \backslash\backslash\ A \text{ ist ein freier Punkt.} \\
B & \leftarrow & FREE & \backslash\backslash\ B \text{ ist ein freier Punkt.} \\
C & \leftarrow & FREE & \backslash\backslash\ C \text{ ist ein freier Punkt.} \\
p & \leftarrow & JOIN(A,B) & \backslash\backslash\ p \text{ ist die Gerade durch } A \text{ und } B. \\
q & \leftarrow & JOIN(A,c) & \backslash\backslash\ q \text{ ist die Gerade durch } A \text{ und } C. \\
r & \leftarrow & BISECT(p,q) & \backslash\backslash\ r \text{ ist Winkelhalbierende von } p \text{ und } q.
\end{array}$$

Abbildung 1 zeigt drei Instanzen dieses GSPs. Man sieht leicht, daß die linke Instanz „stetig" in die rechte überführt werden kann (s. Abb. 2). Im allgemeinen ist es jedoch nicht immer möglich, eine vorgegebene Instanz „stetig" in eine weitere vorgegebene Instanz zu überführen. In [1] wird gezeigt, daß das sogenannte „Reachability Problem" NP-schwer ist.
Die Komplexität des selben Problems im Komplexen (d.h. die Koordinaten der freien Punkte und der abhängigen Elemente dürfen Werte aus $\mathbb{C}$ annehmen) ist hingegen noch unbekannt.
Ein weiteres Problem ist das „Tracing Problem", das mit dem Reachability Problem verwandt ist. Hier liegt die gleiche Situation vor: In [1] wird gezeigt, daß es im Reellen NP-schwer ist, und die Komplexität im Komplexen ist unbe-

kannt. Das sogenannte „Complex Tracing“ könnte z.B. für das automatische Beweisen oder das Umgehen von Singularitäten in *Cinderella* verwendet werden.

Literatur

[1] J. Richter-Gebert, U. Kortenkamp, *Complexity Issues in Dynamic Geometry*, Proceedings of the Smale Fest 2000, 2001

Dissertation: Nächste-Nachbar-Suche in hohen Dimensionen und Externe Algorithmen

Doktorandin: *Laura Heinrich-Litan*, Betreuer: *Helmut Alt*

Für viele Anwendungen, wie zum Beispiel Ähnlichkeitsanfragen in multimedialen Datenbanken, Mustererkennung, Data Mining und Video Kompression, werden Methoden benötigt, die zu einer gegebenen Menge von Punkten $P \subset \mathbb{R}^d$ und einem spezifizierten Punkt $q \in \mathbb{R}^d$, effizient den oder die nächsten Nachbarn aus P zu q finden. Der Abstand wird in einer der Minkowski-Metriken $L_1, L_2, \ldots, L_\infty$ definiert. Diese Suche nennt man Nächste-Nachbar-Suche, und das Problem, dafür effiziente Datenstrukturen und Algorithmen zu entwickeln, ist das Nächste-Nachbar-Problem. Bei vielen Methoden oder Datenstrukturen, die für das Nächste-Nachbar-Problem entwickelt wurden, wird angenommen, daß die Dimension d eine kleine Konstante ist. Die Laufzeit, der Speicherbedarf oder die Vorverarbeitungszeit dieser Methoden sind exponentiell in der Dimension d. In vielen Anwendungen ist aber die Dimension d des Suchraumes sehr groß und deswegen verbietet sich eine in d exponentielle Laufzeit. Im Prinzip konkurrieren alle Algorithmen mit der naiven *Bruteforcemethode*, die $\Theta(nd)$ Zeit für Minkowski-Metriken kostet, keine Vorverarbeitung und nur Speicherplatz für P benötigt. Die *Boxmethode* (entwickelt von Hoffmann und Alt 1998) bestimmt aus einer gegebenen Menge P von Punkten im reellen Einheitswürfel $[0,1]^d$ den nächsten Nachbarn zu einem Punkt $q \in [0,1]^d$ bezüglich der durch die Maximumnorm induzierten Metrik L_∞. Der Algorithmus hat unter der Voraussetzung, daß die Punkte aus P und der spezifizierte Punkt q gleichverteilt aus $[0,1]^d$ sind, die erwartete Laufzeit $O(\frac{nd}{\ln n})$. Varianten der Boxmethode habe ich für das k-Nächste-Nachbar-Problem, zur Bestimmung der k nächsten Nachbarn zu q, erweitert und analysiert. Zukünftig werde ich die Boxmethode bei Zugrundelegung von anderen Wahrscheinlichkeitsverteilungen untersuchen. Von Interesse bleibt auch eine experimentelle Untersuchung, ob diese Methoden in praktischen Anwendungen, wie zum Beispiel bei der Datenbanksuche, sinnvoll einsetzbar sind.

Bei der Entwicklung von Algorithmen für Anwendungen mit sehr großen Datenmengen (large-scale applications), die nicht in den Hauptspeicher passen, ist es wichtig die Input/Output (I/O) Kommunikation zwischen Hauptspeicher und dem langsameren, sekundären Speicher zu minimieren. Geographische Informationssysteme (GIS) und Datenbanksysteme sind gute Beispiele für solche Anwendungen, die immense Datenmengen verarbeiten. Externe Algorithmen (I/O Algorithmen) werden für das theoretische 2-Schichten-Berechnungsmodel entwickelt, wobei das Speichersystem aus einem Hauptspeicher und einer Anzahl von externen Speichergeräten (Disks) besteht. Die Kommunikation erfolgt blockweise, d.h. die Daten werden in Blöcken von zusammenhängenden Daten übertragen. Das von Aggarwal und Vitter eingeführte I/O-Berechnungsmodel ist durch folgende Parameter spezifiziert : M die Anzahl von Elementen (items), die in den Hauptspeicher passen, N die Anzahl der Elemente der Probleminstanz, B die Anzahl von Elementen pro Block und D die Anzahl der Disks, wobei $B \geq 1$ und $B \leq M < N$. Eine I/O-Operation ist eine Lese- oder Schreiboperation eines Blockes aus dem oder in den sekundären Speicher. Ziel ist die Entwicklung solcher Algorithmen, welche sowohl die Anzahl von I/O-Operationen als auch die Anzahl von internen Operationen (die CPU-Zeit) minimieren. Mein Interesse liegt in der Entwicklung eines effizienten, externen Algorithmus für das Nächste-Nachbar Problem in hohen Dimensionen.

Dissertation: Methoden der algorithmischen Geometrie in der Mustererkennung

Doktorand: *Christian Knauer*, Betreuer: *Helmut Alt*.

Die algorithmische Geometrie beschäftigt sich mit dem Entwurf und der Analyse von effizienten Datenstrukturen und Algorithmen zur Lösung von geometrisch motivierten Problemstellungen. Typische Beispiele für solche Problemstellungen sind etwa die Berechnung der konvexen Hülle einer Punktmenge oder die Berechnung aller Schnittpunkte einer Menge von Liniensegmenten.

Auch in der Mustererkennung trifft man auf solche geometrischen Fragestellungen: Zu zwei Mengen von Liniensegmenten in der Ebene, M (dem *Muster*) und B (dem *Bild*) der Kardinalität $m = \#M$ und $b = \#B$ suchen wir in einer Menge von *zulässigen Transformationen* $\mathcal{T}$ (z.B. Translationen oder starre Bewegungen) eine Abbildung t, die $\tilde{\delta}_H(t(M), B)$, den *einseitigen Hausdorff-Abstand* von $t(M)$ nach B, minimiert. Dabei ist $\tilde{\delta}_H(t(M), B)$ der größte Abstand, den ein Punkt aus $t(M)$ zu seinem nächsten Nachbarn in B hat

$$\tilde{\delta}_H(t(M), B) = \max_{p \in t(M)} \min_{q \in B} ||p - q||,$$

also ein Maß dafür, wie „ähnlich“ $t(M)$ zu einem Teilmuster von B ist.

Mit zunehmender Anzahl der Freiheitsgrade der zulässigen Transformationen wird auch die Lösung des Problems aufwendiger. So kann der einseitige Hausdorff-Abstand von M nach B auf einer arithmetischen RAM noch in $\mathcal{O}((b + m) \log(b + m))$ Schritten (uniformes Kostenmaß) berechnet werden ($\mathcal{T} = \emptyset$), während die Laufzeit des derzeit asymptotisch besten Algorithmus zum Auffinden einer optimalen Translation bereits von der Größenordnung $\mathcal{O}((bm)^2 \log^3 bm)$ ist.

Obwohl die Fragestellung sehr stark aus der Anwendung heraus motiviert ist, und die theoretischen Grundlagen bereits weitgehend vorhanden sind, fehlen Implementierungen der entwickelten Verfahren, mit deren Hilfe die Praktikabilität dieser Algorithmen beurteilt werden könnte. Die Realisierung der Algorithmen wird im wesentlichen durch die folgenden Probleme erschwert:

- Oftmals kommen komplexe Datenstrukturen (z.B. Voronoi-Diagramme von Liniensegmenten) und Methoden (z.B. parametrische Suche) zum Einsatz.
- Geometrische Daten müssen durch Zahlentypen beschränkter Genauigkeit im Rechner repräsentiert werden; dem gegenüber steht das idealisierte Rechnen mit reellen Zahlen - eine Annahme, die beim Entwurf der meisten Algorithmen zugrunde gelegt wird.

Dies wirft eine Reihe von Fragen auf, denen man bei der Implementierung von geometrischen Algorithmen immer wieder begegnet, die also nicht unbedingt spezifisch für aus der Mustererkennung motivierte Anwendungen sind:

- Können asymptotisch gute Verfahren auch effizient in die Praxis umgesetzt werden?

- Lohnt es sich, an Stelle von asymptotisch optimalen Verfahren, „schlechtere“, aber konzeptionell einfachere - und damit auch leichter zu implementierende - Methoden zu verwenden?
- Lohnt es sich, an Stelle von exakten Verfahren, approximative, aber schnellere und konzeptionell einfachere Methoden zu verwenden (z.B. Referenzpunktmethoden)?
- Mit welcher Genauigkeit muß gerechnet werden, um die topologische Korrektheit des Endresultates garantieren zu können, bzw. um eine vorgegebene Fehlerschranke nicht zu überschreiten?

Dissertation: Datenstrukturen zum Auffinden von Formen

Doktorand: *Lutz Meißner*, Betreuer: *Helmut Alt.*

Die Menge aller Polygonzüge in der Ebene wird, etwa durch den Hausdorff- oder den Fréchet-Abstand, zu einem metrischen Raum. Von besonderem Interesse sind die Räume, bei denen zur Abstandsmessung die einzelnen Polygonzüge “verschoben” werden können:

$$\delta(P,Q) = \min_{t \in \mathbf{R}^2} \tilde{\delta}(P, Q+t)$$

Es läßt sich nun, bei gegebenen Polygonzügen $P_1, P_2, \ldots, P_n$, die Frage stellen, welches dieser P_i einem weiteren Polygonzug P am“ähnlichsten” ist:

$$\mathrm{NN}(P) = \{P_i | \delta(P, P_i) \leq \delta(P, P_j) \text{ für } j = 1, \ldots, n\}$$

Gesucht ist also der nächste Nachbar (oder die nächsten Nachbarn) von P.

Ist man nicht nur an der Bestimmung des nächsten Nachbarn eines, sondern vieler Polygonzüge interessiert, ist es sinnvoll, $P_1, P_2, \ldots, P_n$ in einer Datenstruktur zusammenzufassen, um eine effiziente Abfrage zu ermöglichen. Aber wie muß diese Datenstruktur aussehen?

Für die Nächste-Nachbar-Suche von Punkten im $\mathbf{R}^d$ sind effiziente Datenstrukturen bekannt. Diese nutzen jedoch Eigenschaften, etwa die endliche Dimensionalität oder die Vektorraum-Eigenschaften, aus, die bei den Polygonzügen nicht vorhanden sind.

Bei den bekannten Methoden in allgemeinen "großen metrischen Räumen" werden Voraussetzungen sowohl an die zugrundeliegende Metrik als auch an die Verteilung der Datenmenge gestellt, die hier nicht oder nur bedingt zutreffen. Weiter gibt es kaum Aussagen zu den erwarteten Laufzeiten.

Die Berechnung des Abstands zweier Polygonzüge ist zeitaufwendig. Es stellt sich die Frage, ob zur Bestimmung von $\mathrm{NN}(P)$ tatsächlich P mit einzelnen P_i verglichen werden muß, oder ob ein effizienteres Vorgehen möglich ist.

Obwohl zunächst die theoretischen Aspekte untersucht werden, wird auch an die Implementierung von entwickelten Ansätzen gedacht.

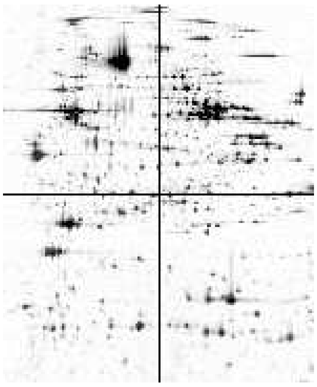
Projekt: Point-Pattern-Matching zur Analyse von Gelbildern

Helmut Alt, Darko Dimitrov, Frank Hoffmann, Klaus Kriegel.

Das derzeitige Projekt geht aus einer Forschungskooperation des Instituts für Informatik der FU Berlin und des Deutschen Herzzentrums Berlin hervor. Dieses ursprüngliche Projekt wurde bis Juni 2001 von der DFG gefördert. Für Teile der dabei entwickelten Software wurde ein Lizensierungsvertrag mit der Firma Bio-Rad Laboratries abgeschlossen, der eine 2-jährige Weiterfinanzierung der Forschung und Softwareentwicklung sichert.

Gegenstand der Untersuchung sind 2-dimensionale Gelbilder, die durch Gelelektrophorese - Techniken erzeugt werden. Die 1975 von O'Farrell eingeführte Gelelektrophorese hat sich als eine zentrale molekularbiologische Methode zur hochauflösenden Trennung von Protein-Gemischen und zur Analyse der Protein-Zusammensetzung von Gewebeproben etabliert. Jeder Punkt ("Spot") in einem so erzeugten Gelbild repräsentiert ein in der Probe auftretendes Protein. Das Original des hier verkleinert abgebildeten Herz-Ventricle-Gels enthält ca. 2000 Spots. Ziel der Analyse der Bilder ist es insbesondere, krankheitsassoziierte Proteinausprägungen zu erkennen. Zwar ist es möglich, einzelne

Proteine durch Sequenzierung zu bestimmen, dies ist aber sehr teuer und aufwendig und bei der Menge der Daten nicht realistisch. Deshalb basierte bis vor wenigen Jahren die Auswertung der Gelbilder zu großen Teilen auf der genauen (und zeitaufwendigen) Betrachtung durch erfahrene Spezialisten.

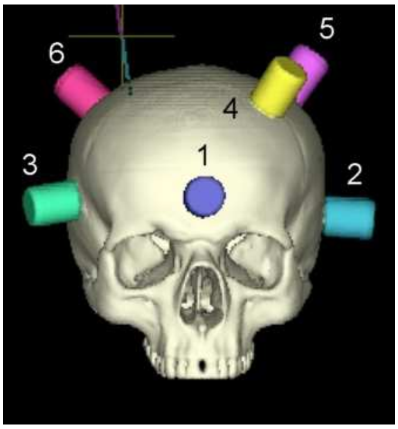


Inzwischen gibt es eine Reihe von Softwarepaketen zur Unterstützung dieser Arbeit, aber an einer hochzuverlässigen und vollautomatischen Lösung des Problems wird überall noch gearbeitet.

Im Projekt werden zwei der zentralen algorithmischen Probleme der Gelanalyse untersucht:
1) **Spotdetektion:** Im allgemeinen konzentrieren sich die Moleküle eines Proteins aus der Probe in einer achsenparllelen elliptischen Region des Gels - dem Spot des Proteins. Bei der Spotdetektion geht es um die Erkennung dieser Regionen. Das ist eine relativ einfache Bildverarbeitungsaufgabe, so lange die Spots gut separiert sind. Wenn sich mehrere Spots zu einer komplexen und übersättigten Region überlappen, ergibt sich ein schwieriges algorithmisches Problem, das mit Approximationsalgorithmen bearbeitet wird.
2) **Gelmatching:** Hier setzt man voraus, dass zwei zu vergleichende Bilder durch die Spotdetektion schon in geometrische Punktmuster umgewandelt wurden und nun ein geometrisches Matching dieser Punktmuster gesucht wird. Die besondere Schwierigkeit ergibt sich durch die technologisch bedingten, geometrischen Verzerrungen in den Bildern. Durch die Komplexität der Gelelektrophorese an sich, sind schon von ein und derselben Probe in einem Labor keine zwei identischen Bilder zu erwarten. Die zu entwickelnden Verfahren müssen daher sehr robust sein, um auch den Vergleich von Bildern aus verschiedenen Datenbanken zu ermöglichen. Mit Ansätzen aus der algorithmischen Geometrie konnte ein neuartiger Lösungsweg für dieses Problem entwickelt und implementiert werden, der den Kern des Programmsystem CAROL bildet (`http://gelmatching.inf.fu-berlin.de`).

Projekt: Elektromagnetische Navigation in der Gehirnchirurgie

G. Rote



Entwicklung eines elektromagnetischen Navigationsverfahrens zum daten- und bildgesteuerten intraoperativen neurophysiologischen Mapping (Ortung) und Monitoring (Überwachung) bei Operationen in eloquenten Hirnarealen, bei der Deep Brain Stimulation (Tiefe Hirnstimulation) und bei minimal-invasiven Eingriffen.

Dieses in Zusammenarbeit mit der neurochirurgischen Klinik, Universitätsklinikum Benjamin Franklin (Prof. Brock, Dr. Suess) geplante Projekt soll das obige Projekt fortsetzen und erweitern. Bei der Operation von Hirntumoren in der Nähe von funktionell wichtigen ("eloquenten") Hirnarealen können diese Zonen verletzt werden und dadurch bleibende neurologische Schäden entstehen. Deshalb ist es wichtig, vor einer Operation die Lage dieser Funktionszentren festzustellen und sie während der Operation zu überwachen. Dabei müssen räumliche Daten, die aus verschiedenen Quellen stammen oder zu verschiedenen Zeiten (präoperativ beziehungsweise während der Operation) gewonnen wurden, miteinander verknüpft werden. Zum Beispiel werden vor der Operation mit Hilfe von Computertomographie (CT) und Magnetresonanztomographie (MRT) dreidimensionale Bilder gewonnen. Während der Operation müssen diese mit der tatsächlichen Lage im Kopf des Patienten zur Übereinstimmung gebracht werden, wobei sich durch den Eingriff die Gehirnmasse verschiebt. Zusätzlich gibt es Informationen über die Lage funktionell wichtiger Areale beim Menschen in sogenannten Stereotaxie-Atlanten. Diese müssen an individuelle Unterschiede zwischen den Patienten angepasst werden müssen.

Das algorithmische Problem, das diesen Aufgaben zugrundeliegt, besteht in der Registrierung (Ausrichtung), bei der die verschiedenen Datensätze miteinander in räumliche übereinstimmung gebracht werden. Die hierbei verwendeten Transformationen waren bisher gewöhnlich starre Transformationen, oder

Transformationen, die eine gewisse Verzerrung berücksichtigen, die jedoch im gesamten Bereich gleichartig ist (affine Transformationen). Bei den Anwendungen dieses Projekts müssen jedoch allgemeinere Transformationen in Betracht gezogen werden, für die jedoch noch kein mathematisches Modell verfügbar ist.

Ein anderer Teil dieses Projekts betrifft die minimalinvasiven Endoskopie: Dabei wird durch durch die Hohlgänge und Hohlräume des Gehirns ein Instrument eingeführt, das man sich als dünnen starren Schlauch vorstellen kann, dessen Spitze jedoch lenkbar ist und verlängert werden kann. Das Problem ist hier, sich anhand des Kamerabildes an der Spitze des Schlauches im Gehirn zurechtzufinden.

4 Die Arbeitsgruppe Theoretische Informatik

Leiter

- Prof. Dr. Helmut Alt

Entwurf und Analyse von Algorithmen, insbesondere Algorithmische Geometrie mit Schwerpunkt Formanalyse.

- Prof. Dr. Günter Rote

Algorithmische Geometrie, kombinatorische Optimierung.

Mitglieder der Arbeitsgruppe

- Hosam Abdo

Algorithmische Geometrie.

- PD Dr. Peter Brass

Diskrete Geometrie, Algorithmische Geometrie.

- Britta Broser

Kombinatorik, Geometrie und Optimierung.

- Darko Dimitrov

Bildverarbeitung, Computersehen, Flächenrekonstruktion aus dreidimensionalen Punktdaten.

- PD Dr. Stefan Felsner

Algorithmen für Halbordnungen und Graphen, Algorithmische Geometrie, Kombinatorik.

- Laura Heinrich-Litan

Algorithmische Geometrie, Externe Algorithmen, Nächste-Nachbar- Suche.

- Dr. Frank Hoffmann

Algorithmische Geometrie, geometrische Online-Probleme, angewandte Matching-Probleme.

- Christian Knauer

Algorithmische Geometrie, Implementierung von geometrischen Algorithmen, Ähnlichkeitsbestimmung von polygonalen Figuren.

- Dr. Ulrich Kortenkamp
Dynamische Geometrie, Orientierte Matroide, Nachbarschaftliche Polytope, Java.

- PD Dr. Klaus Kriegel
Graphalgorithmen und graphentheoretische Methoden für geometrische Probleme.

- Lutz Meissner
Algorithmische Geometrie, Implementierung von geometrischen Algorithmen.

- Dr. Géraldine Morin
Computergrafik, Computer-Aided Geometrie Design.

- Ares Ribó Mor
Geometrie, Kombinatorik

Weitere Informationen

Prof. Dr. Helmut Alt Takustr. 9 Raum 112 Tel.: 838-75160 `alt@inf.fu-berlin-de`	Prof. Dr. Günter Rote Takustr. 9 Raum 110 Tel.: 838-75150 `rote@inf.fu-berlin.de`	Dr. Stefan Felsner Takustr. 9 Raum 117 Tel.: 838-75161 `felsner@inf.fu-berlin.de`
Dr Peter Braß Takustr. 9 Raum 124 Tel.: 838-75167 `brass@inf.fu-berlin-de`	PD Dr. Klaus Kriegel Takustr. 9 Raum 115 Tel.: 838-75156 `kriegel@inf.fu-berlin.de`	