STUDIEN

Effiziente Algorithmen

für Studenten der Mathematik und Informatik
an der Freien Universität Berlin

Semesterheft Winter 2002/03

STUDIENSCHWERPUNKT

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.

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* 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 Winter 2002/03

Vorlesungen

Algorithmen und Programmierung I **[Grundstudium]**

Dozent: Alt; Vorlesungszeit: Mo 12–14 Uhr, Do 16–18 Uhr, 4-stündig,
Veranstaltungsort: Takustraße 9, HS.
Übungen Alt, 2-stündig .
Beginn: 14.10.2002

Inhalt: Zentraler Gegenstand des Bereichs Algorithmen und Programmmierung ist die Entwicklung und Beschreibung von Algorithmen. Dazu gehören theoretische Grundlagen wie Berechenbarkeit, Verifikation und Komplexität ebenso wie die praktische Programmierung.
Behandelt werden Spezifikationen und Implementierung von Algorithmen und Datenstrukturen und grundlegende Prinzipien von Programmiersprachen und Programmiermethodik. Während ab dem 2. Semester in einer imperativen Sprache (Java) programmiert wird, werden in dieser Veranstaltung Funktionen zur Formulierung von Algorithmen verwendet. Zur Einführung in die Funktionale Programmierung benutzen wir die Programmiersprache Haskell und zwar die Implementierung HUGS, die kostenlos zur privaten Nutzung von "http://haskell.cs.yale.edu/hugs" bezogen werden kann. Sie enthält eine Unix- und eine Windows 95/NT-Version sowie eine umfangreiche Dokumentation. Das Handbuch der aktuellen Version ist Teil der unverzichtbaren Literatur zur Veranstaltung. Bitte beachten: Aktuelle Informationen zur Vorlesung finden Sie jeweils auf der AlPI-Webseite.

Literatur: Thompson, S.: Haskell, the craft of functional programming, Addison-Wesley. Bird, R./Wadler, Ph.: Einführung in Funktionale Programmierung, Hanser Verlag, 1982. Zusätzliche Literatur wird in der Vorlesung angegeben.

Entwurf und Analyse von Algorithmen **[EA1]**

Dozent: Kortenkamp; Vorlesungszeit: Mo 10–12 Uhr, Mi 10–12 Uhr, 3-stündig,
Veranstaltungsort: Takustraße 9, HS.
Übungen Kortenkamp, 2-stündig.

Beginn: 21.10.2002

INHALT: Der Entwurf von Algorithmen bildet einen Kernbereich der Informatik. Diese Vorlesung ist eine einführende Veranstaltung zur Algorithmik und Grundlage für die meisten anderen Veranstaltungen in der Theoretischen Informatik. Inhalt ist der Entwurf und die Analyse von Algorithmen und Datenstrukturen für viele grundlegende Probleme wie Suchen, Sortieren, Graphenprobleme, Arithmetik, goemetrische Probleme usw.
Literatur: Cormen/Leiserson/Rivest: Introduction to Algorithms, MIT Press, 1990.

Externe Algorithmen und Datenstrukturen [EA2]

Dozent: Knauer; Vorlesungszeit: Do 16–18 Uhr, 2-stündig.
Veranstaltungsort: Takustraße 9, SR 046.
Übungen Knauer, 2-stündig.
Beginn: 17.10.2002

INHALT: In vielen Informatikanwendungen, z.B. der Analyse von biologischen Massendaten, treten enorm große Datenmengen auf, die auf einem externen Speichermedium, wie z.B. Magnet-Plattenspieler, gespeichert werden müssen. Da Externspeicher wesentlich langsamer als Hauptspeicher ist, hat die Entwicklung von effizienten externen Datenstrukturen und Algorithmen an Bedeutung gewonnen, die speziell für die Verwaltung von Daten auf dem Externspeicher entworfen werden. In dieser Vorlesung werden wichtige externe Daten- strukturen vorgestellt. Es wird unter anderem auf externe Sortierverfahren, B-Bäume und mehrdimensionale Index- strukturen eingegangen.
Literatur: Cormen, Leierson, Rivest: Introduction to Algorithms, MIT Press Ottmnan, Widmayer: Algorithmen und Datenstrukturen, BI Wissenschaftsverlag Abello (Ed.): External Memory Algorithms, AMS

Seminare, Praktika und sonstige Veranstaltungen

Seminar Ausgewählte Kapitel der Graphentheorie [ADM]

Dozent: Felsner; Vorlesungszeit: Fr 14–16 Uhr oder als Blockseminar, Block, 2-stündig.

Veranstaltungsort: Takustraße 9, SR.
Vorbesprechung: n.V. Vorbesprechung in Raum 117
Inhalt: In diesem Seminar wollen wir uns mit Arbeiten zu topologischer Graphentheorie und graphentheoretischer Topologie beschäftigen. Anstelle einer Inhaltsangabe haben wir eine Liste von Schlagwörtern zusammengestellt: Planare Graphen, Kreuzungszahlen, Knoten, Polytope und Skelett- graphen, Graphenklassen jenseits von Planarität. Genaueres in der Vorbesprechung.

Seminar über Algorithmen [EA2]

Dozent: Alt, Kortenkamp; Vorlesungszeit: Do 14–16 Uhr, 2-stündig.
Veranstaltungsort: Takustraße 9, SR 037.
Vorbesprechung: erste Semesterwoche Di 16–17 Uhr
Inhalt: Hauptsächlich algorithmische Geometrie mit einem Schwerpunkt auf der Rekonstruktion von Kurven und Oberflächen aus Mengen von Punkten einem in den letzten Jahren wegen seiner praktischen Relevanz sehr intensiv bearbeitetem Thema. Zielgruppe: Informatiker, Mathematiker und andere einschlägig Vorgebildete im Hauptstudium. Voraussetzungen: Vorlesung Ëntwurf und Analyse von Algorithmen" Perspektiven: Vergabe von Studien-, Examens- und Diplomarbeiten möglich.

Literatur: zur Einführung in die algorithmische Geometrie: R. Klein, Algorithmische Geometrie, Addison-Wesley, 1997; sonst Originalliteratur

Seminar Komplexitätstheorie [EA2]

Dozent: Knauer; Vorlesungszeit: Mi 16–18 Uhr, 2-stündig.
Veranstaltungsort: Takustraße 9, SR 053.
Inhalt: Das Seminar soll die gleichnamige Vorlesung aus dem letzten Semester vertiefen und neuere Resultate und Methoden vorstellen die in den Standardvorlesungen normalerweise nicht behandelt werden:
o Exponentielle untere Schranken für die Größe monotoner Schaltkreise für das Cliquenproblem
o Zero-Knowledge Beweise
o Randomisierte Komplexitätsklassen
o PCP-Theorem
o Kolmogorov-Komplexität
Literatur: Bovet, Crescenzi: Introduction to the Theory of Complexity, Prentice Hall, 1994 Cormen, Leierson, Rivest: Introduction to Algorithms, MIT

Press, 1990 Garey, Johnson: Computers and Intractability - A Guide to the Theory of NP-Completeness, Freeman, 1979 Papadimitriou: Computational Complexity, Addison-Wesley, 1994 Reischuk: Einführung in die Komplexitätstheorie, Teubner, 1990 Sipser: Introduction to the theory of computation, PWS, 1997 Wegener: Kompendium theoretische Informatik, Teubner, 1996

Seminar Algorithmen zur Analyse von 2DE-Gelbildern [ANW]

Dozent: Hoffmann, Kriegel; Vorlesungszeit: n.V. , 2-stündig.
Veranstaltungsort: Takustraße 9, SR..., n. V.
Inhalt: Die Gelelektrophorese hat sich als eine wichtige moleku- larbiologische Methode zur Analyse der Protein- und DNA-Kompo- nenten von Gewebeproben etabliert. In den bei diesem Verfahren hergestellten 2-dimensionalen Elektrophorese-Gelbildern (kurz 2DE-Bilder) konzentriert sich jede Proteinkomponente aus der untersuchten Probe in einer kleinen, ellipsenförmigen Region des Bildes, einem sogenannten Spot. Die Analyse und der Vergleich der Bilder hilft, molekulare und genetische Krankheitsursachen aufzudecken oder den Einfluss von Umweltbedingungen auf den Organismus zu studieren. Im Mittelpunkt des Seminars stehen algorithmischen Fragen, die bei Bildanalyse und -Vergleich zu lösen sind. Während die Probleme bei der Spoterkennung in das Gebiet der Bildverarbeitung führen, werden wir uns beim Bildvergleich mit geometrischen Algorithmen und Datenstrukturen beschäftigen. Zielgruppe: Studenten der Informatik (Hauptstudium) und Bioinformatik (Master)

Diplomanden- und Doktorandenseminar [EA2]

Dozent: Alt, Felsner, Kriegel; Vorlesungszeit: Di, Do, Fr 12–13 Uhr, 3-stündig.
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. Siehe auch Semesterheft zum Studienschwerpunkt Effiziente Algorithmen.

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 Winter 2002/03

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 Winter 2002/03

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

- Algorithmische Bioinformatik (VL); Dozent: Vingron.
- Spatial Databases (VL); Dozent: Voissard.

Europäisches Graduiertenkolleg
Berlin ——————————— Zürich
Combinatorics, Geometry, and Computation

In Berlin und Zürich sind Stellen für

Doktorand(inn)en

mit überdurchschnittlichem Studienabschluss in Mathematik, Informatik oder einem verwandtem Gebiet für zwei Jahre zu vergeben.
Das Kolleg ist eine gemeinsame Initiative der ETH Zürich, der drei Berliner Universitäten – Freie Universität, Technische Universität und Humboldt-Universität –, und dem Konrad-Zuse-Zentrum für Informationstechnik Berlin. Die deutschen Partner finanzieren sich aus Mitteln der Deutschen Forschungsgemeinschaft (DFG).
Die Stipendien in Berlin werden nach den Richtlinien der DFG bemessen und betragen monatlich bis zu 1468,– EURO steuerfrei (Familienzuschlag 205,– EURO). Das wissenschaftliche Programm reicht von theoretischen Grundlagen bis hin zu Anwendungen. Die Forschungsgebiete sind Kombinatorik, Geometrie, Optimierung, Algorithmen und Berechnung, sowie Computer Graphik und Vision.
Am Studienort Berlin werden die Stipendiaten von den Professoren Aigner, Alt, Rote und Schulz (FU), Möhring, Ziegler (TU), Prömel (HU) und Grötschel (ZIB) betreut.
Zu den Bewerbungsunterlagen der Doktorand(inn)en gehören Lebenslauf, Zeugniskopien, Examensarbeit, Gutachten des letzten Betreuers und Vorstellungen zum gewünschten Promotionsvorhaben. Alle Unterlagen sind bei dem Sprecher des Kollegs in Berlin einzureichen. Sprecher des Kollegs

Prof. Dr. Helmut Alt
Institut für Informatik
Freie Universität Berlin
Takustraße 9
D-14195 Berlin

b.w.

Weitere Informationen sind erhältlich bei
Bettina Felsner
Tel. ++49–30–838 75 104
bfelsner@inf.fu-berlin.de

Internet: http://www.inf.fu-berlin.de/gk-cgc

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).

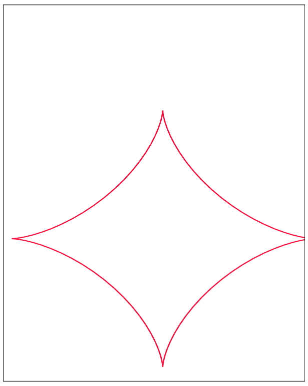


Abbildung 1: Astroide

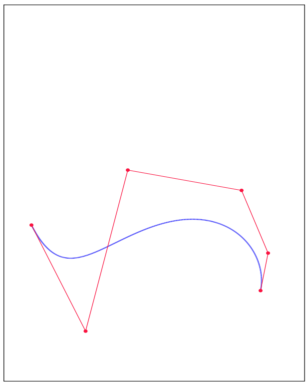


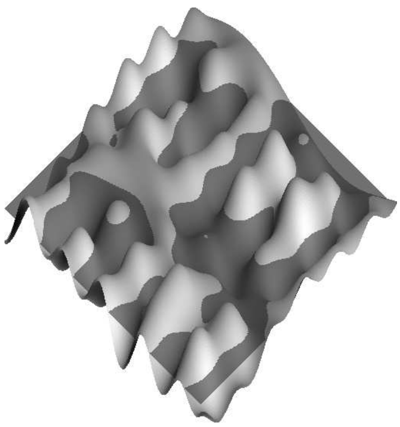
Abbildung 2: Bézier-Kurve

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.



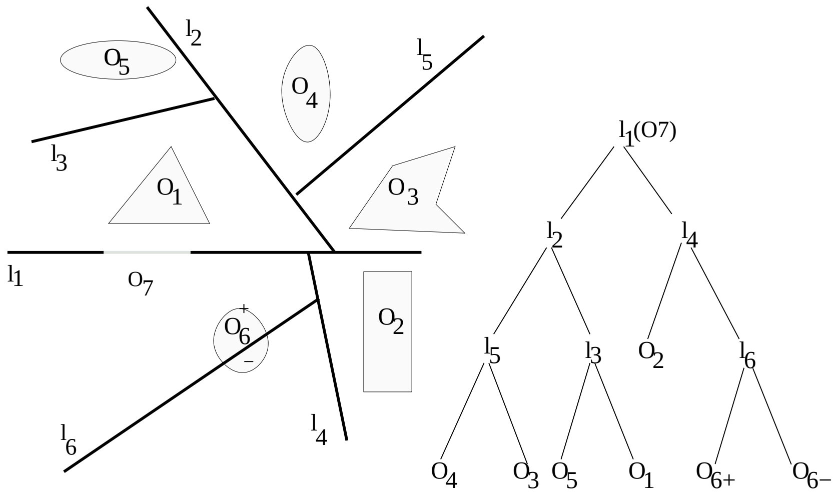
Bei der Erstellung einer Datenstruktur, die ein schnelles Zugreifen auf die Kontouren erlaubt, spielt der Kontourbaum eine wichtige Rolle. Er speichert alle relevanten "Ereignisse", die zu Änderungen der Kontouren führen und erlaubt das Erstellen minimaler sogenannter *seed sets*, aus denen dann eine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

Diplomarbeit: Zufällige Euler-Touren

Diplomand: *Miguel Domingo-Vecchioni*, Betreuer: *Stefan Felsner*

Ein Eulerscher Graph $G = (V, E)$ kann eine Vielzahl von Euler-Touren enthalten. Es stellt sich die Frage, wie man eine *zufällige Euler-Tour* von G erzeugen kann. Ist G gerichtet, so kann dieses Problem auf dem der Erzeugung eines *zufälligen Spannbaums* von G zurückgeführt werden, wofür mehrere effiziente Algorithmen bekannt sind. Das von Aldous und Broder z.B. simuliert eine Irrfahrt auf G bis alle Knoten besucht wurden; all die Kanten, die beim ersten Besuch eines Knotens verwendet wurden, bilden zusammen einen zufälligen Spannbaum. Für ungerichtete Graphen ist aber noch kein effizienter Algorithmus für die Erzeugung von zufälligen Euler-Touren bekannt.

In enger Verbindung dazu steht das Problem der Erzeugung von *zufälligen Eulerschen Orientierungen*, d.h. Orientierungen der Kanten von G mit $d^+(v) = d^-(v)$ für alle $v \in V$. Im Fall eines ebenen Graphen G kann auf der Menge $\mathcal{E}$ der Eulerschen Orientierungen von G eine Halbordnung definiert werden, die $\mathcal{E}$ zu einem distributiven Verband macht. Die von Propp und Wilson eingeführte Technik (*coupling from the past*) für die gleichverteilte Erzeugung von kombinatorischen Objekten ist dann besonders effizient. Es wird eine monotone Markov-Kette auf $\mathcal{E}$ konstruiert, deren stationäre Verteilung die Gleichverteilung ist. Kritisch für die Laufzeit des Algorithmus ist die Kopplungszeit der Markov-Kette, d.h. die Zeit bis die Trajektorien des Maximum und des Minimum des Verbands kollabieren. Die gilt es zu analysieren.

Algorithmen für die Erzeugung von zufälligen Euler-Touren können beim Test der statistischen Signifikanz von DNA-Matchings verwendet werden. Zufällige Eulersche Orientierungen finden Anwendung in der Berechnung von Partitionsfunktionen in der statistischen Physik.

Dissertation: Complex Tracing

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.

Beispiel für ein GSP:

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

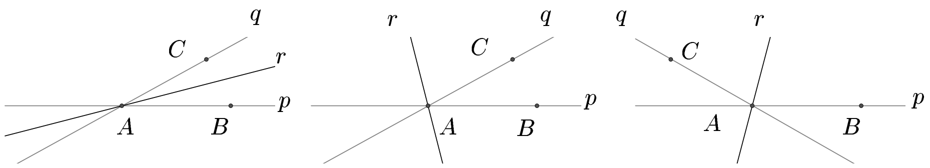


Abbildung 1: Drei verschiedene Instanzen des GSPs aus dem Beispiel

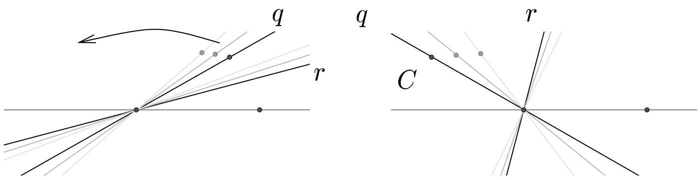


Abbildung 2: Die linke Instanz aus Abb. 1 kann „stetig“ in die rechte überführt werden.

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

Dissertation: Nächster-Nachbar-Suche in hohen Dimensionen

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 Nächster-Nachbar-Suche, und das Problem, dafür effiziente Datenstrukturen und Algorithmen zu entwickeln, ist das Nächster-Nachbar-Problem. Die Dimension d des Suchraumes ist in vielen Anwendungen dieses Problems sehr groß; sie liegt in der Grössenordnung von Hunderten bis Tausenden. Die bekannten Algorithmen für das exakte Nächster-Nachbar-Problem benötigen eine in d exponentielle Laufzeit oder einen in d exponentiellen Speicherplatz. Sie sind daher ineffizient für hochdimensionale Anwendungen, und können mit der brute-force Methode nicht konkurrieren, welche alle n Distanzen zu dem Anfragepunkt berechnet und den Punkt mit minimaler Distanz auswählt.

In meiner Dissertation werden effiziente Algorithmen für die exakte Nächster-Nachbar-Suche in einem hochdimensionalen $(\mathbb{R}^d, L_\infty)$ Raum entwickelt. Ich untersuche die erwartete Laufzeit dieser Algorithmen unter der Voraussetzung, dass die Punkte aus P gleichverteilt aus dem d-dimensionalen Einheitswürfel sind. Die Algorithmen sind einfach zu implementieren und verbessern wesentlich die brute-force Methode bezüglich der erwarteten Laufzeit.

Die *Boxmethode* (entwickelt von Alt und Hoffmann 1998) löst das exakte L_∞-Nächster-Nachbar Problem, benötigt keine Vorverarbeitung und nur Speicherplatz für die n Punkte. Deren erwartete Laufzeit ist $O(\frac{nd}{\ln n} + n)$. In meiner Dissertation untersuche ich Erweiterungen der Boxmethode, die auf der Basis einer einfachen Vorverarbeitung eine effizientere Suche ermöglichen. Für hohe Dimensionen $d > \ln n$ hat ein von mir entwickelte Suchalgorithmus eine erwartete Laufzeit von $O\left(n \ln\left(\frac{d}{\ln n} + 1\right) + n\right)$, einen linearen Speicherbedarf und $O(nd \ln n)$ Vorverarbeitungszeit. Weiterhin betrachte ich einen Suchalgorithmus, der eine in der Vorverarbeitung berechnete Zerlegung der Punktmenge P benutzt. Die Zerlegung besteht aus Folgen von Punkten, die monoton in

$\mathbb{R}^d$ sind. Die erwartete Laufzeit beträgt $O(\sqrt{d}n^{1-\frac{1}{\sqrt{d}}}\ln n)$ für Dimensionen $d < \left(\frac{\ln n}{\ln\ln n}\right)^2$.

Die entwickelten Methoden und deren Laufzeitanalyse habe ich für das k-Nächste-Nachbarn-Problem verallgemeinert, und bei Zugrundelegung von anderen Wahrscheinlichkeitsverteilungen analysiert. Außerdem betrachte ich das Externer-Speicher-Berechnungsmodell und einen externen Algorithmus für die L_∞-Nächster-Nachbar-Suche.

Weiterhin habe ich eine Methode entwickelt, die einen Tradeoff zwischen dem erwarteten Speicherbedarf der Datenstruktur und der erwarteten Laufzeit des dazugehörigen Suchalgorithmus erlaubt.

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.

Dissertation: Locked and Unlocked Self-Touching Linkages

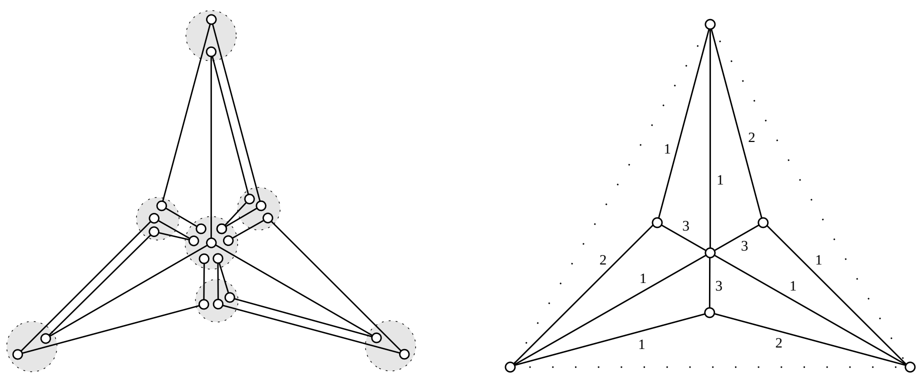
Doktorandin: *Ares Ribó Mor*, Betreuer: *Günter Rote*

In the last years, there has been much progress in the study of planar linkages that are *locked*, in the sense that there exists no motion into some other configuration preserving the bar lengths and without bars crossing. A *linkage* is a graph where edges are *rigid bars* with fixed length and vertices are flexible joints. A *configuration* of a linkage in $\mathbb{R}^2$ is a mapping of the vertices to points in $\mathbb{R}^2$. In the plane, the combinatorial planar embedding is specified because this cannot change by a motion that avoids crossings.

The problem is to analyze which planar linkages are locked. The tools used for this topic are geometric planar properties combined with techniques from rigidity theory, like first-order rigidity and equilibrium stresses.

It is known that a polygonal tree can be locked, and that a polygonal arc, a polygonal cycle or a disjoint union of nonnested polygonal arcs and cycles is always unlocked. The key distinction is that arc and cycles have maximum degree 2, but a tree may have vertices of higher degree. In fact, a single degree-3 vertex can prevent opening.

We have shown that every monotone tree is unfoldable, using geometric relations between segments with disjoint interiors in the plane. Other related questions are: is every arbitrarily flattened tree unfoldable? And, more general: which conditions must be satisfied by a tree for being unlocked?



STC, which is locked

In a *self-touching configuration* (STC) bars can touch and even lie along each other, but not properly cross. The set of feasible motions can be described by linear equations and inequalities, which are stable at least in some neighbourhood of the STC. A *δ-perturbation* of a STC is a repositioning of the vertices within disks of radius δ consistent with the combinatorial embedding in $\mathbb{R}^2$. Locked linkages are often based on approximations to STC.

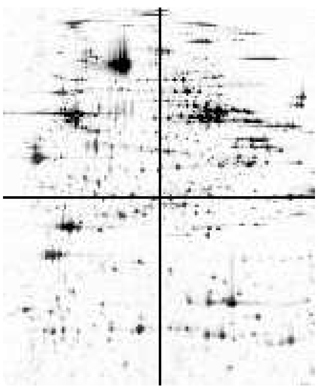
We have the following conjecture: for all STC and for all $\delta > 0$, there is a δ-perturbation that is a configuration without bars crossing. The idea for the proof is to model the configuration by a linear problem, and to show that the dual has a bounded solution. We treat the dual problem in terms of equilibrium stresses and study the properties of the geometric representation of the proportional distribution of this stresses.

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.



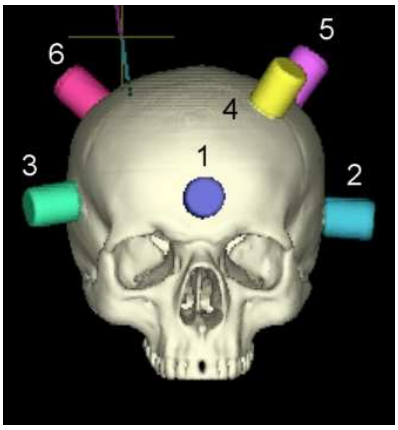
Inzwischen gibt es eine Reihe von Softwarepaketen zur Unterstützung dieser Arbeit, aber an einer hochzuverlässigen und vollautomatischen Lösung des Problems wird überall noch gearbeitet.

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.

Projekt: Three-Dimensional Dynamic Geometry

Enno Brehm, Ulrich Kortenkamp

Dynamic Geometry deals with the constructional aspects of drawings in a dynamic setup. Several dynamic geometry software systems (DGS) have been developed during the last decade that can handle two-dimensional geometric constructions, e.g. as done with ruler and compass. As opposed to traditional, non-computer-aided, drawings, a dynamic aspect is added to a construction: all steps of a construction are stored by the software and can be recalled for other positions of the base elements, making it possible to move points to other positions while maintaining the mathematical restrictions encoded in the construction.

The overwhelming success of DGS for two-dimensional geometry and its intimate relation to fields like computational kinematics, parametric CAD, and virtual reality demand for a genuine three-dimensional implementation of Dynamic Geometry. The goal of this project will be to eliminate several of the obstacles that have been identified on the way to this implementation.

The first step will be the proper identification of mathematical problems that arise in the transition from two to three dimensions. The concepts developed in [**?**] have to be extended where possible and will be replaced by alternative approaches where necessary. At the same time a reference implementation is required that demonstrates the feasibility of the new methods for 3D Dynamic Geometry. This includes extending work on user interfaces for 3D visualization respecting the special requirements of interactive manipulation of geometric constructions. The relations to industrial applications like CAD have to be investigated. It will be necessary to adapt the constructive mathematical model to the implicit formulations in constraint-based systems as customary today. Also the usability of a 3D-DGS has to be ensured for applications that involve PDE solving or physics simulation.

We will cooperate with J. Richter-Gebert (TU München), the EU funded project *Effective Computational Geometry for Curves and Surfaces*, and a subproject of the BMBF-Project *Entwicklung einer dezentralen internetunterstützten Lehr-Lernumgebung für das Lehramtsstudium Mathematik*.

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.

- Enno Brehm

Dynamische 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.

- Dr. 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.

- Tobias Lenz
Algorithmische Geometrie.

- Ares Ribó Mor
Geometrie, Kombinatorik

- Astrid Sturm
Algorithmische Geometrie.

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

PD Dr. Stefan Felsner
Takustr. 9
Raum 117
Tel.: 838-75161
felsner@inf.fu-berlin.de

PD Dr. Klaus Kriegel
Takustr. 9
Raum 115
Tel.: 838-75156
kriegel@inf.fu-berlin.de