Algorithmen und Datenstrukturen
Einführung

Matthias Teschner
Graphische Datenverarbeitung
Institut für Informatik
Universität Freiburg

SS 12

Motivation

- Einführung in die Begriffe
  - Algorithmus
  - Datenstruktur
  - Entwurfskonzept
- Bedeutung von Algorithmen und Datenstrukturen
  - Beispiele
  - Anwendungen

Zur Person

- Informatik-Studium in Berlin
- Promotion in Erlangen
- Forschung und Lehre
  in Stanford und an der ETH Zürich
- seit 2005 in Freiburg
  Professur für Graphische Datenverarbeitung
  am Institut für Informatik

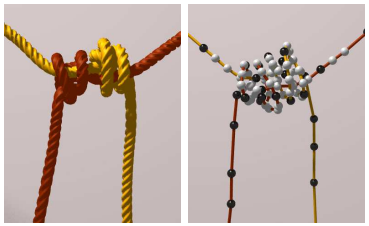
- Kontakt
  - http://cg.informatik.uni-freiburg.de/
  - teschner@informatik.uni-freiburg.de
  - 052 / 01-005

Graphische Datenverarbeitung innerhalb der Informatik

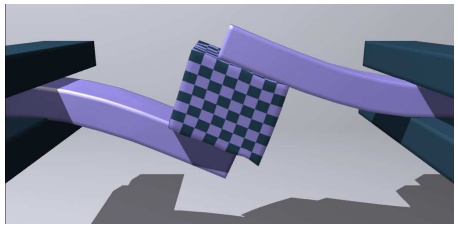
- Algorithmen und Datenstrukturen
- Rechnerarchitektur und Betriebssysteme
- Programmiersprachen und Softwaretechnik
- Künstliche Intelligenz
- Kommunikations- und Informationssysteme
- Graphik und Bildverarbeitung
  - Mustererkennung und Bildverarbeitung
  - Graphische Datenverarbeitung

Graphische Datenverarbeitung Forschung

- physikalisch-basierte Animation

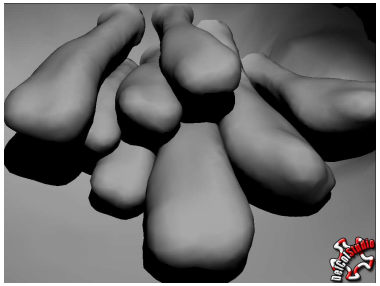


Seile

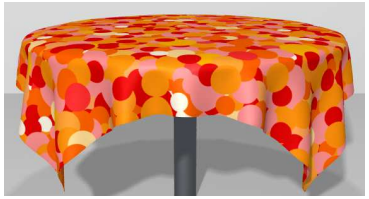


deformierbare Objekte

Operationsplanung



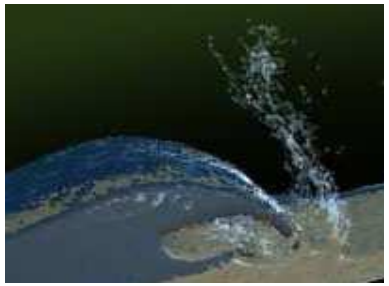
Trainingssysteme



Stoff

Kontaktbehandlung

Effekte



Flüssigkeiten

Effekte

Überblick

- Algorithmus
- Datenstruktur
- Entwurfskonzept
- Organisatorisches

Algorithmus

- wohldefinierte Rechenvorschrift, die eine Menge von Elementen als Eingabe verwendet und eine Menge von Elementen als Ausgabe erzeugt
- beschreibt eine Rechenvorschrift zum Erhalt einer durch die Formulierung eines Problems gegebenen Eingabe-Ausgabe-Beziehung

$$\text{Algorithmus} f : \text{Eingabe} \rightarrow \text{Ausgabe}$$

- Kochrezepte, Bedienungsanleitungen sind Algorithmen.
- Programme sind Algorithmen, die durch Computer ausgeführt werden können.

Beispiel

- Sortierproblem

- Eingabe: Folge von Zahlen $< a_1, a_2, \ldots, a_n >$
- Ausgabe: Sortierte Folge der Eingabe $< a'_1, a'_2, \ldots, a'_n >$
  mit $a'_1 \leq a'_2 \leq \ldots \leq a'_n$

- Sortieralgorithmen lösen das durch die Eingabe-Ausgabe-Relation beschriebene Sortierproblem.

Instanz eines Problems

- Eine Eingabe stellt eine Instanz eines Problems (Eingabeinstanz) dar, wenn
  - die an die Eingabe gebundenen Bedingungen erfüllt sind und
  - alle zur Ausführung des Algorithmus notwendigen Daten vorliegen

- Beispiele
  - $(4, 2, 3)$ ist eine Instanz des Sortierproblems
  - $(4, 2, \mathrm{Algorithmus})$ ist keine Instanz des Sortierproblems, solange keine entsprechende Ordnungsrelation definiert ist
  - $(2, 3, 4)$?

Formale Eigenschaften von Algorithmen

- Korrektheit
  - Ein korrekter Algorithmus stoppt (terminiert) für jede Eingabeinstanz mit der durch die Eingabe-Ausgabe-Relation definierten Ausgabe.
  - Ein inkorrekter Algorithmus stoppt nicht oder stoppt mit einer nicht durch die Eingabe-Ausgabe-Relation vorgegebenen Ausgabe.

- Effizienz
  - Bedarf an Speicherplatz und Rechenzeit
  - Wachstum (Wachstumsgrad, Wachstumsrate) der Rechenzeit bei steigender Anzahl der Eingabe-Elemente (Laufzeitkomplexität)

Motivation

- Bedeutung der Korrektheit intuitiv klar
- Bedeutung der Effizienz eines Algorithmus durch Begrenztheit von Ressourcen gegeben
  - Sortieralgorithmus A benötigt n^2 Schritte für n Elemente
  - Sortieralgorithmus B benötigt $n\log_2 n$ Schritte für n Elemente
  - Computer 1: 10^9 Schritte / s, Computer 2: 10^7 Schritte / s
  - Eingabegröße: 10^6 Elemente
  - Laufzeit:

	Algorithmus A	Algorithmus B
Computer 1	17 min	0.02 s
Computer 2	2 d 8 h	2 s

- → Wachstumsrate oder Laufzeitkomplexität dominiert die benötigte Berechnungszeit für große Eingaben

Weitere Eigenschaften von Algorithmen

- Ausführbarkeit: Jeder Schritt muss ausführbar sein.
- Finitheit: Algorithmus muss mit endlichem Text eindeutig beschreibbar sein.
- Dynamische Finitheit: Algorithmus benötigt zu jedem Zeitpunkt endlich viel Speicher.
- Determiniertheit (determiniertes Ergebnis): Die Ausgabe ist für jede Eingabe eindeutig definiert.
- Determinismus (deterministischer Ablauf): Der nächste Schritt des Algorithmus ist zu jedem Zeitpunkt eindeutig definiert.
- Terminierung: Der Algorithmus terminiert, wenn er nach endlich vielen Schritten stoppt.

Beispiele für Problemstellungen

- Sortieren
- Optimieren
  - effizienter Transport
  - Verwaltung knapper Ressourcen
- geometrische Fragestellungen
  - konvexe Hülle einer Punktmenge
  - n nächste Nachbarn eines Punktes innerhalb einer Menge
- mathematische Fragestellungen
  - Matrixoperationen für spezielle Matrizen

Zusammenfassung Algorithmus

- präzise, endliche Beschreibung eines Verfahrens
- realisiert eine gegebene Eingabe-Ausgabe-Beziehung
- Korrektheit und Effizient sind wichtige formale Eigenschaften zur Analyse von Algorithmen.
- Die Motivation für die Beschäftigung mit Algorithmen ergibt sich insbesondere aus der Bedeutung der Laufzeitkomplexität.
- Die Laufzeit eines Algorithmus ist dominiert durch den Wachstumsgrad der Laufzeit bei steigender Eingabegröße.

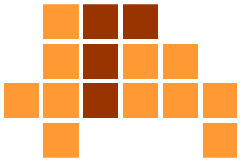
Überblick

- Algorithmus
- Datenstruktur
- Entwurfskonzept
- Organisatorisches

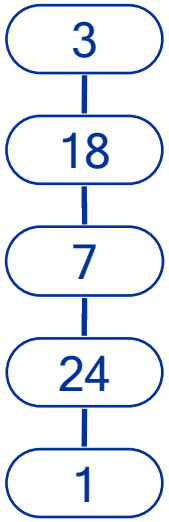
Datenstruktur

- Algorithmen manipulieren dynamische Mengen von Elementen (Eingabe → Ausgabe)
  - Suchen, Einfügen, Löschen
  - Minimum, Maximum, nächstkleinstes oder nächstgrößtes Element
- Datenstrukturen werden zur Realisierung (Repräsentation) dynamischer Mengen verwendet.
- Datenstrukturen sind unterschiedlich effizient in Bezug auf Manipulationen (Operationen).
- Sinnvolle Wahl einer Datenstruktur hängt von der Effizienz der darin implementierten Operationen ab, die für einen gegebenen Algorithmus relevant sind.

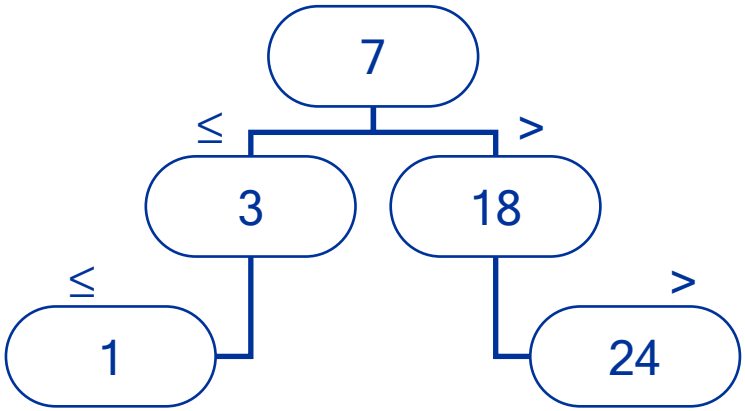
Motivation



- alternative Repräsentation einer Menge von Zahlen

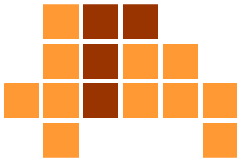


Liste

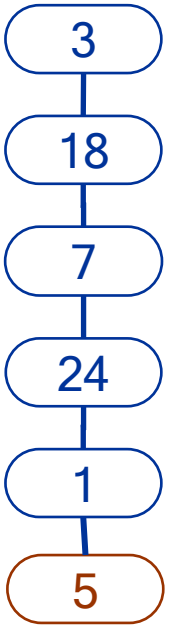


Binärbaum

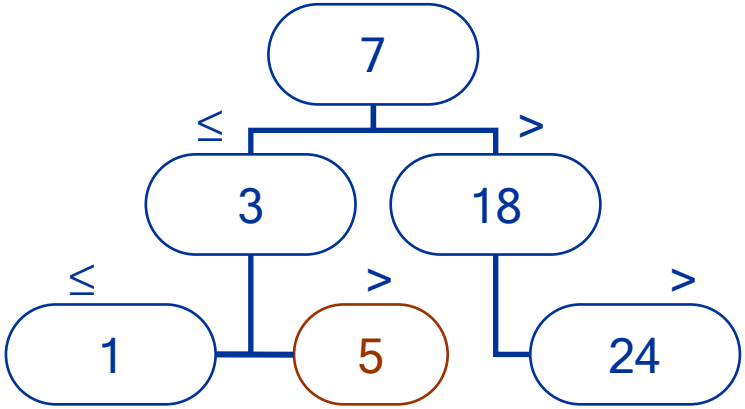
Motivation



- Einfügen eines weiteren Elements "5"



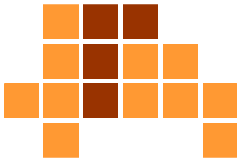
Liste



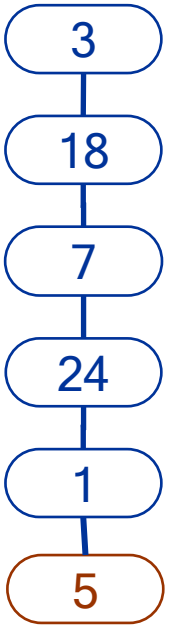
Binärbaum

- aufwändiger für den Binärbaum

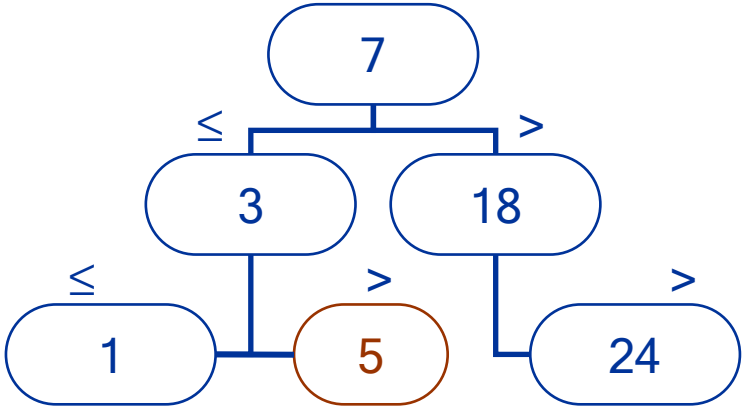
Motivation



- Suchen des Elements "5"



Liste



Binärbaum

- effizienter für den Binärbaum

Beispiele für Datenstrukturen

- Feld
  - Zugriff auf ein Element über einen Index
- Liste
  - Element besitzt Verweis auf das folgende Element
- Stapel
  - Elemente können nur in umgekehrter Reihenfolge des Einfügens gelesen oder gelöscht werden
- Warteschlange
  - Elemente können nur in gleicher Reihenfolge des Einfügens gelesen oder gelöscht werden
- Graphen, Bäume
  - Elemente besitzen variable Anzahl von Verweisen auf weitere Elemente

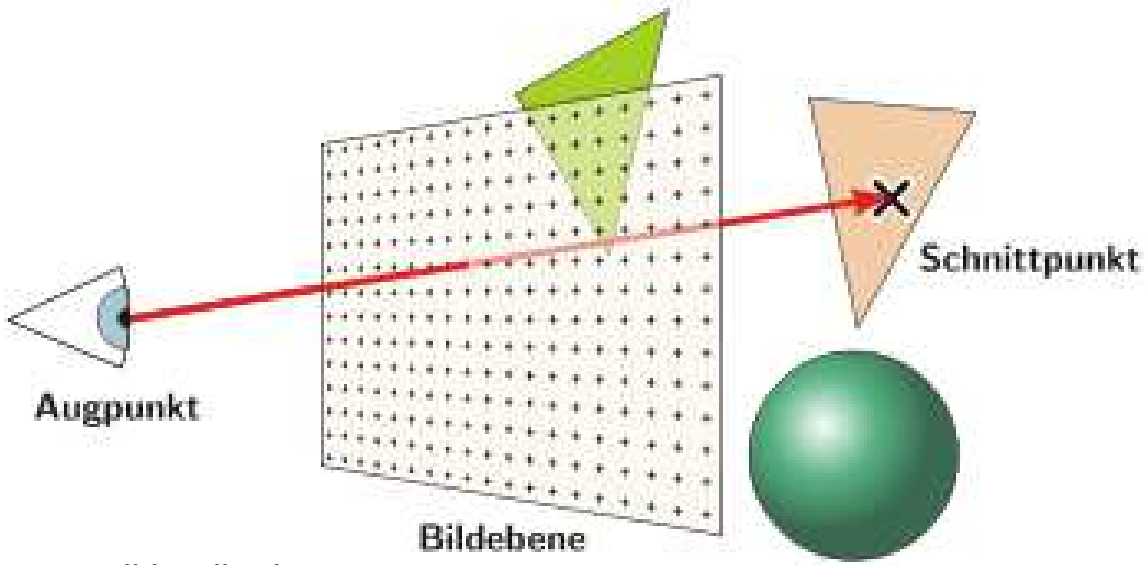
Anwendungen in der Graphischen Datenverarbeitung

- Ray tracing

Gilles Tran
www.oyonale.com

Ray tracing

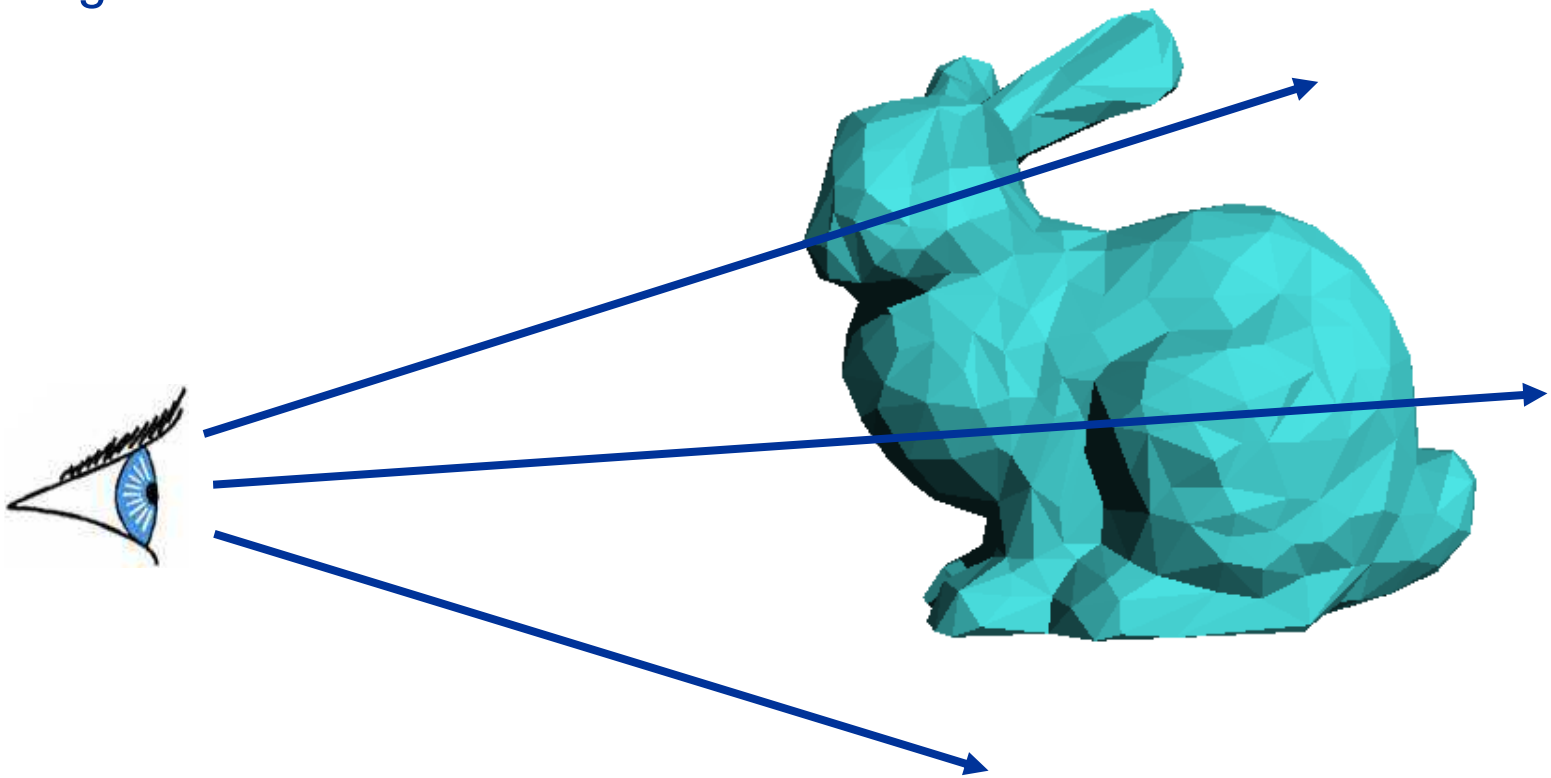
- betrachte einen Sehstrahl pro Bildpunkt
- werte ein Beleuchtungsmodell am nächsten Schnittpunkt zwischen Sehstrahl und Szene aus
- weise die resultierende Farbe dem Bildpunkt zu



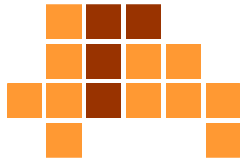
www.wikipedia.de

Ray tracing

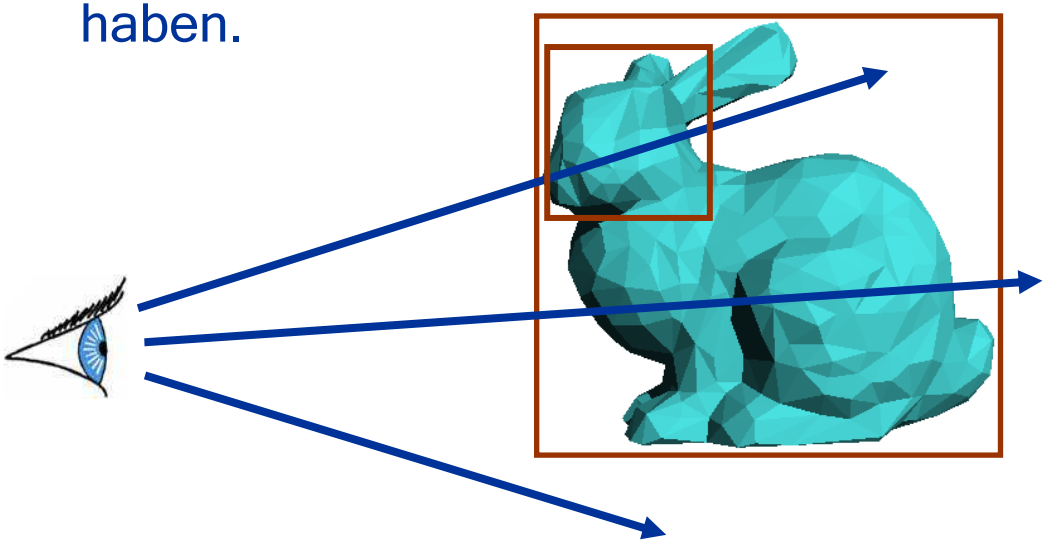
- Problem
  - Minimiere die Zahl der Schnitt-Tests, wenn die Szene durch eine große Zahl von Dreiecken beschrieben ist.



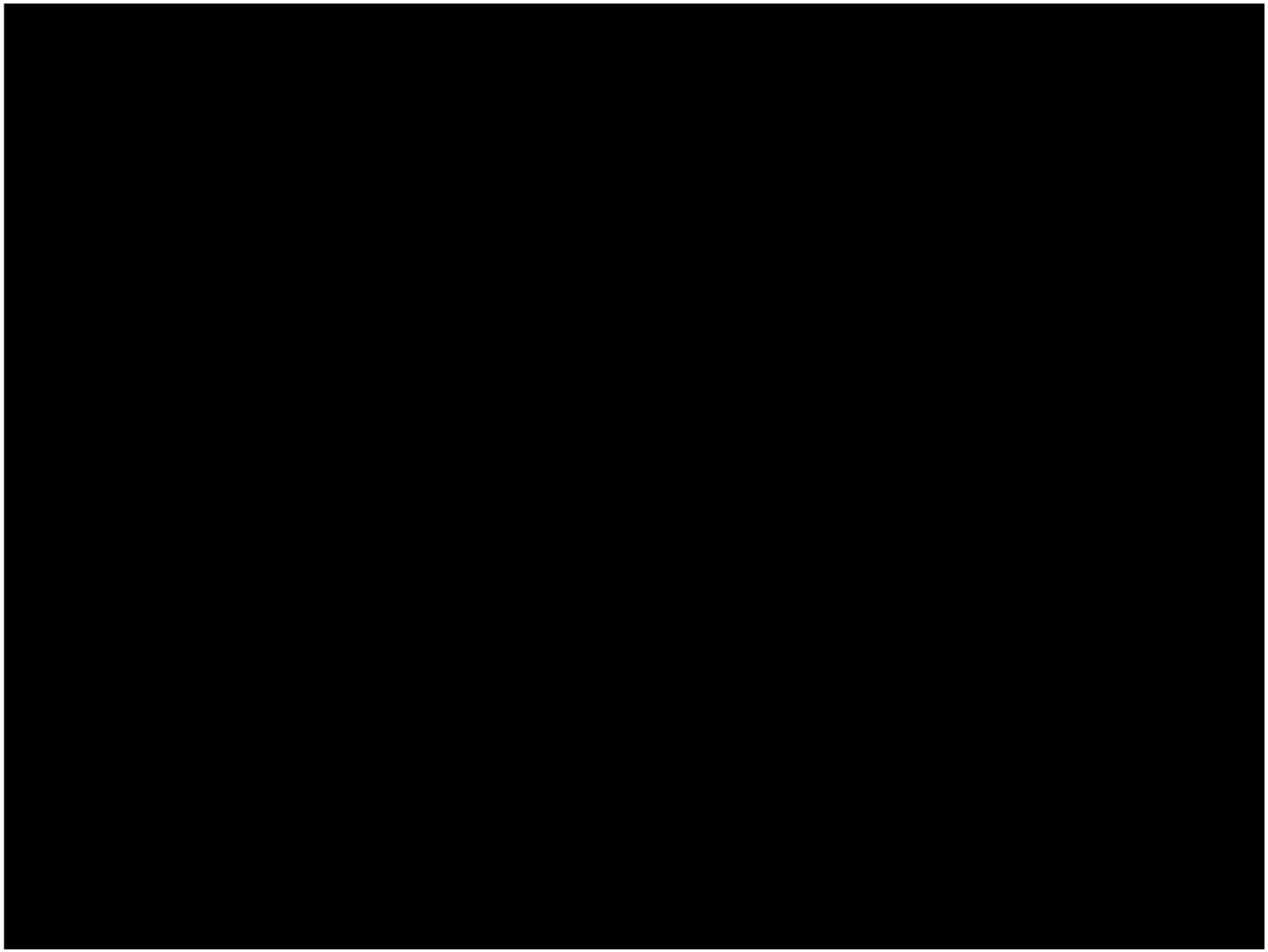
Ray tracing



- Lösungsansatz
  - räumliche Datenstrukturen (z. B. Begrenzungsvolumina)
- Idee
  - effizientes Ausschließen potentieller Schnitte
  - Ein Sehstrahl, der das Begrenzungsvolumen nicht schneidet, kann keinen Schnittpunkt mit einem eingeschlossenen Dreieck haben.



Anwendungen in der Graphischen Datenverarbeitung



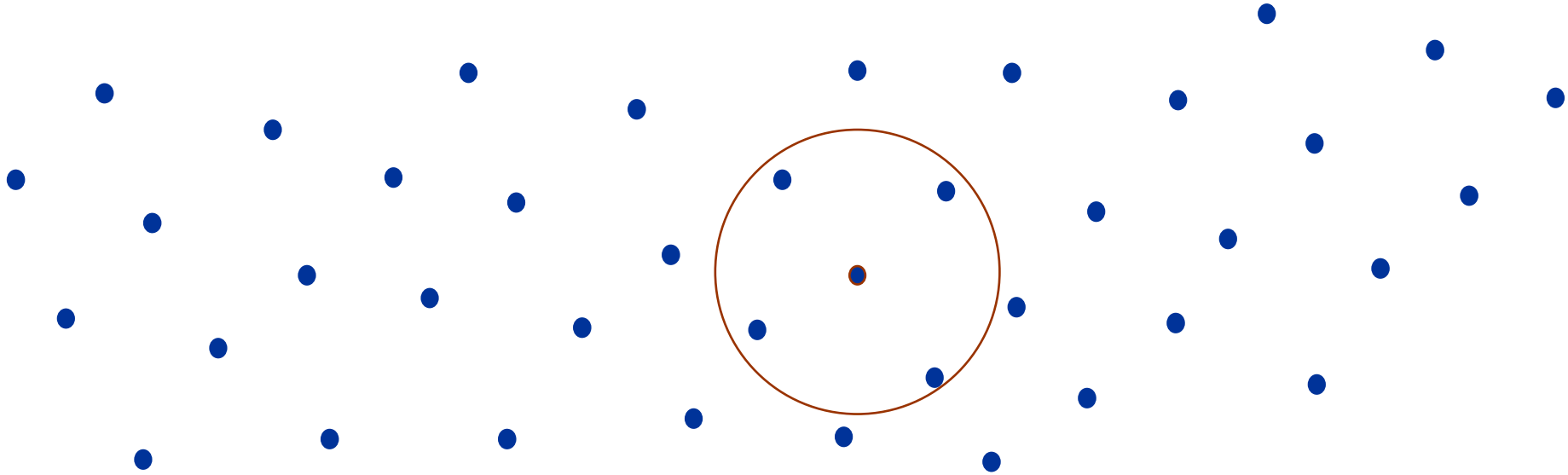
Fluid-Animation

Nadir Akinci
Markus Ihmsen
Gizem Akinci
Barbara Solenthaler
Matthias Teschner

Universität Freiburg / ETHZ

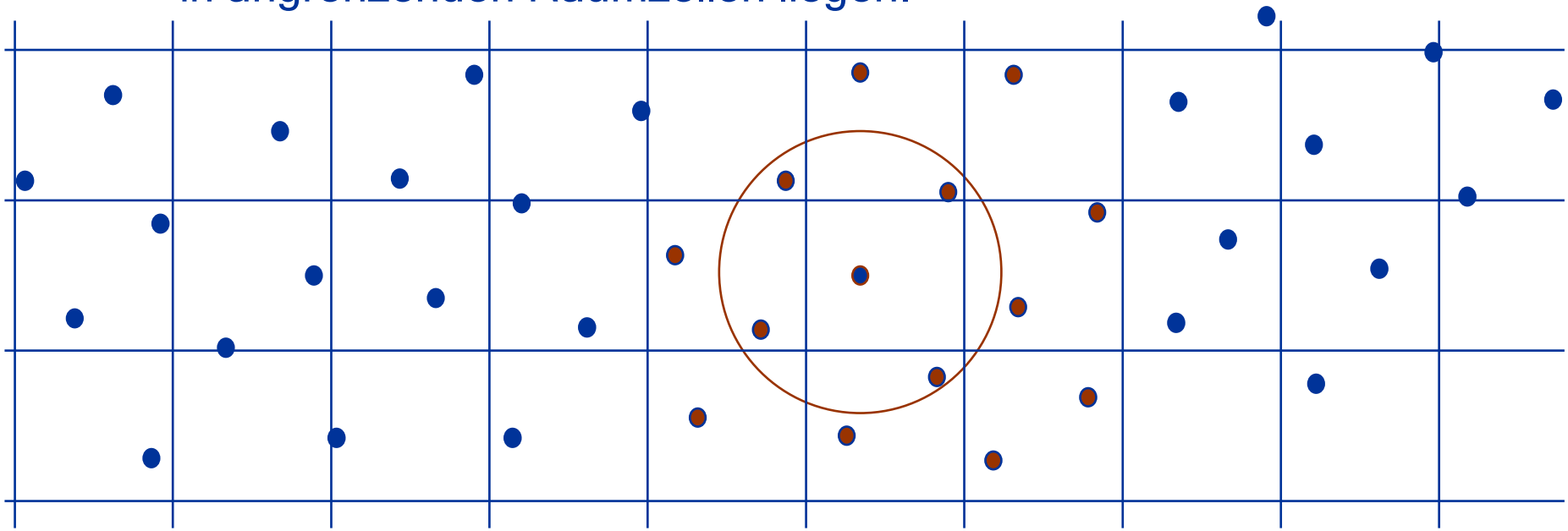
Partikelbasierte Fluid-Animation

- Problem
  - Finde alle Partikel in der Umgebung eines Partikels zur Berechnung von Kräften



Partikelbasierte Fluid-Animation

- Lösungsansatz
  - räumliche Datenstruktur (z. B. Raumunterteilung)
  - Partikel werden Raumzellen zugeordnet.
  - Nachbarpartikel können nur in der gleichen oder in angrenzenden Raumzellen liegen.



Zusammenfassung Datenstruktur

- Algorithmen manipulieren dynamische Mengen von Elementen (Eingabe → Ausgabe)
  - Suchen, Einfügen, Löschen
  - Suchen von Punkten in einer Umgebung
- Datenstrukturen werden zur Realisierung dynamischer Mengen verwendet.
- Datenstrukturen sind unterschiedlich effizient in Bezug auf Manipulationen (Operationen).
- Sinnvolle Wahl einer Datenstruktur hängt von der Effizienz der darin implementierten Operationen ab, die von der Anwendung / dem Algorithmus vorgegeben ist.

Überblick

- Algorithmus
- Datenstruktur
- Entwurfskonzept
- Organisatorisches

Entwurfstechniken

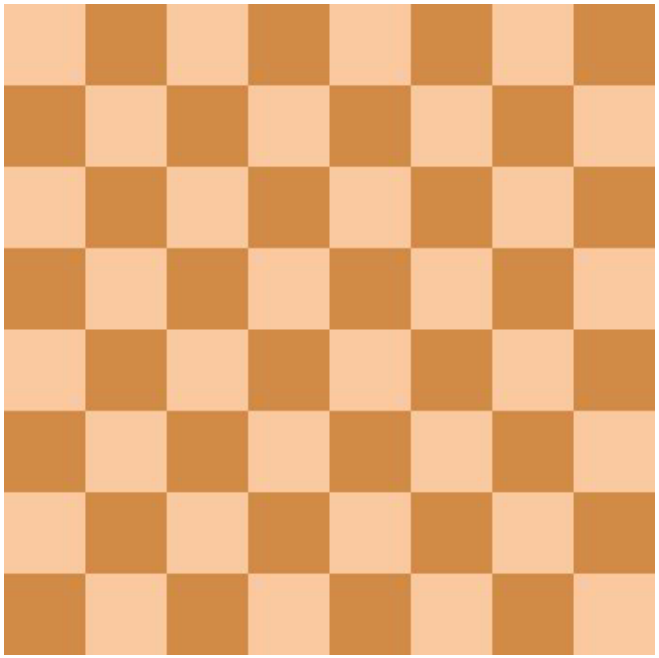
- umfassen grundlegende Ideen zum Entwurf eines Algorithmus
- sind generische algorithmische Muster für bestimmte Problemklassen
- Beispiele
  - Greedy
  - Backtracking
  - Divide-and-Conquer

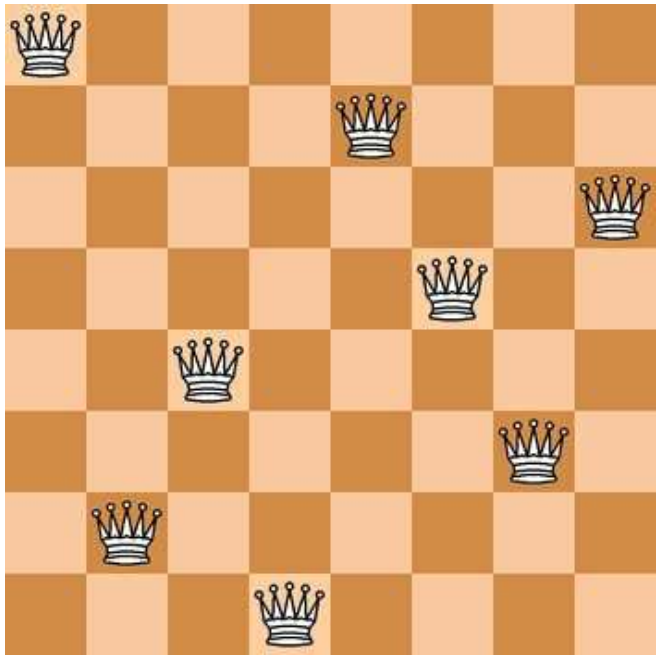
Greedy

- wähle jeweils den Folgezustand,
  - der den augenblicklich größten Zugewinn verspricht
  - der augenblicklich optimal ist
- Beispiel: Wechselgeld mit möglichst wenig Münzen
- Lösungsansatz: Wähle die jeweils größtmögliche Münze, um schnell (gierig, greedy) ans Ziel zu kommen
  Münzen: 50, 20, 10, 5, 2, 1
  Wechselgeld: 68 = 50 + 10 + 5 + 2 + 1 (optimal)
- Oft nicht optimal (lediglich lokales statt globales Optimum):
  Münzen: 11, 5, 1
  Wechselgeld: 15 = 11 + 1 + 1 + 1 + 1 (optimal: 15 = 5+5+5)

Backtracking

- systematische, vollständige Suche in einem vorgegebenen Suchraum
- Beispiel: Platziere n Damen, die sich nicht gegenseitig bedrohen

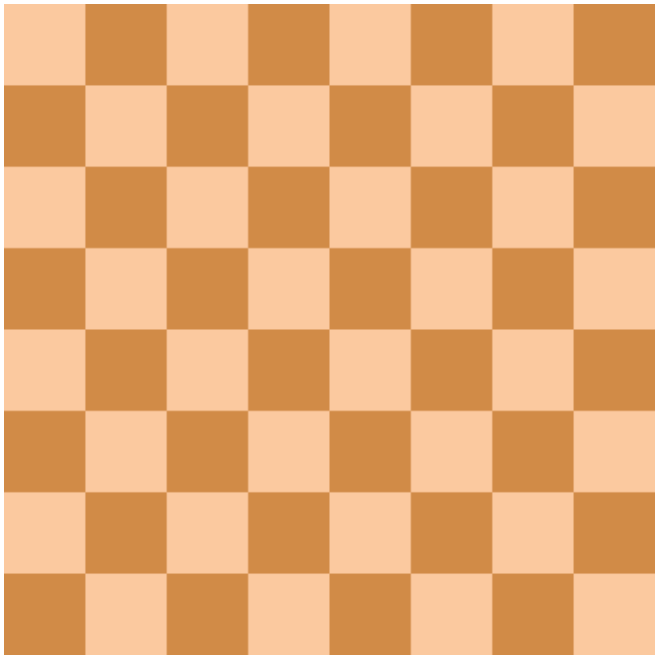


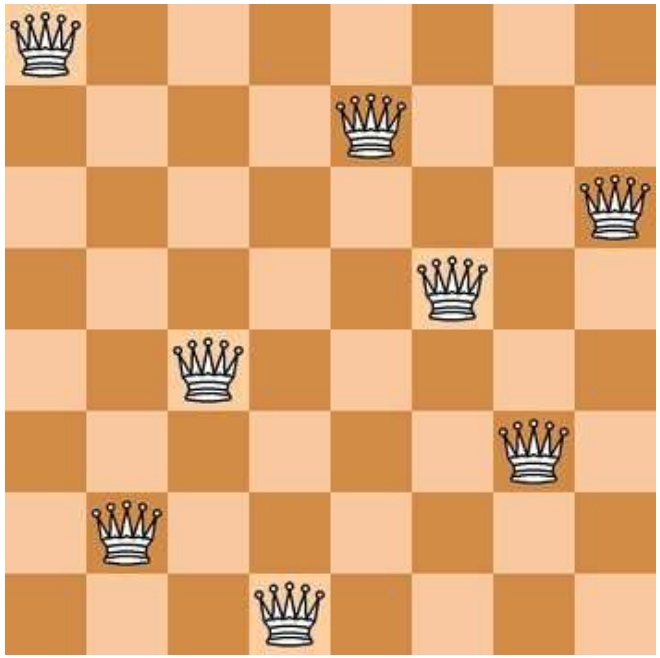


www.wikipedia.de

Backtracking

- systematische, vollständige Suche in einem vorgegebenen Suchraum
- Beispiel: Platziere n Damen, die sich nicht gegenseitig bedrohen





www.wikipedia.de

Zusammenfassung
Lernziele der Vorlesung

- Algorithmen
  - Sortieren, Suchen, Optimieren
- Datenstrukturen
  - Repräsentation von Daten
  - Listen, Stapel, Schlangen, Bäume
- Techniken zum Entwurf von Algorithmen
  - Algorithmenmuster
  - Greedy, Backtracking, Divide-and-Conquer
- Analyse von Algorithmen
  - Korrektheit, Effizienz

Literatur

- **Ottmann, Widmayer**, *"Algorithmen und Datenstrukturen"*, Spektrum Akademischer Verlag, Heidelberg, ISBN 3-8274-1029-0, 2002.
- **Saake, Sattler**, *"Algorithmen und Datenstrukturen: eine Einführung mit Java"*, dpunkt-Verlag, Heidelberg, ISBN 3-89864-122-8, 2002.
- **Cormen, Leiserson, Rivest, Stein**, *"Algorithmen – Eine Einführung"*, Oldenbourg Wissenschaftsverlag, München, ISBN 978-3-486-58262-8, 2007.
- **Cormen, Leiserson, Rivest, Stein**, *"Introduction to Algorithms"*, MIT Press, Boston, ISBN 0-262-53196-8, 2003.

- Knuth, Sedgewick, Baase / van Gelder, Kleinberg / Tardos, Goodrich / Tamassia, Nievergelt / Hinrichs, Güting / Dieker, Heun, Drozdeck, Standisch, Kruse, Wood u.v.a.

Überblick

- Algorithmus
- Datenstruktur
- Entwurfskonzept
- Organisatorisches

Übungen

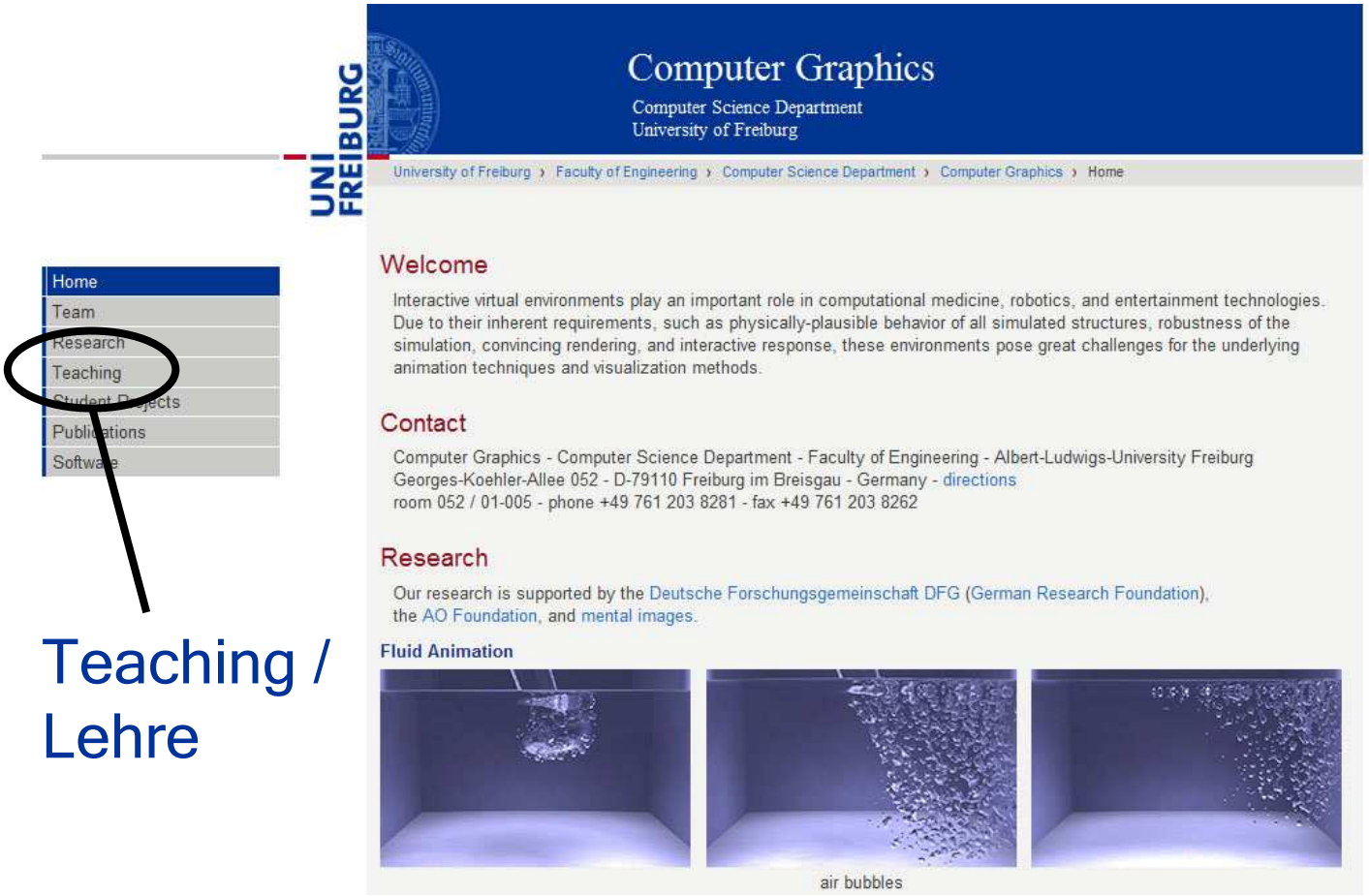
- Eintragung in die Übungsgruppen via Online-Portal
- Tutorate: wöchentlich Mo, Mi, Do, Fr ab 7. Mai
- Praktika: Pools im Geb. 082, Räume 021 und 028
- Übungsblätter
  - Ausgabe / Abgabe jeweils montags
  - eine Woche Bearbeitungszeit
  - Wechsel von Theorie und Praxis
  - Abgabe bis jeweils montags, 12 Uhr in den entsprechend gekennzeichneten Briefkästen im Erdgeschoss Geb. 051
  - Ausgabe 1. Übungsblatt: Montag, 30.4.

Übungen

- Mo 16-18 101 / 01-018
- Mi 16-18 051 / 00-006
- Do 14-16 051 / 00-006
- Do 16-18 078 / 00-014
- Fr 14-16 051 / 03-026

Web-Seite

- http://cg.informatik.uni-freiburg.de/



Teaching / Lehre

Web-Seite

- http://cg.informatik.uni-freiburg.de/teaching.htm

Übersicht
der Lehr-
veranstaltungen

Informationen zu
Lehrveranstaltungen
- Aufzeichnung 2008
- Vorlesungsfolien
- Übungsblätter
- Ankündigungen

Kontakt

- Tutoren
  - Christian Schöneweiß info2.christian@tutor.tf
  - Johannes Löffler info2.johannes@tutor.tf
  - Tobias Seufert tobias.seufert@web.de
  - Daniel Leinfelder info2.daniel@tutor.tf
  - Christoph Schötz schoetzc@informatik.uni-freiburg.de
- Koordination
  - Thorsten Schmidt tschmidt@informatik.uni-freiburg.de

Fragen

- Web
  - Informatik
  - Graphische Datenverarbeitung
  - Prüfungsordnung
- Kommilitonen
- Tutoren
- Thorsten Schmidt
- Matthias Teschner

Vermischtes

- "Wissen Sie schon genau wann die Klausur sein wird?"
- "Ist es notwendig sich für ohre Vorlesung anzumelden oder sich irgendwo für Übungen einzutragen oder erfährt man das alles in der erten Vorlesungsstunde?"
- "... komme jetzt ins zweite Semester. In diesem ist es geplant Informatik II zu belegen, kann ich mich auch jetzt noch bei Ihnen anmelden?"
- "Bitte vergessen Sie meine E-Mails."
- "Hallo, ich habe gehört ... - Klaus" (Absender: basildon1981@gmx.de)

Nächste Vorlesung

- Beschreibung von Algorithmen
- Pseudocode
- Java