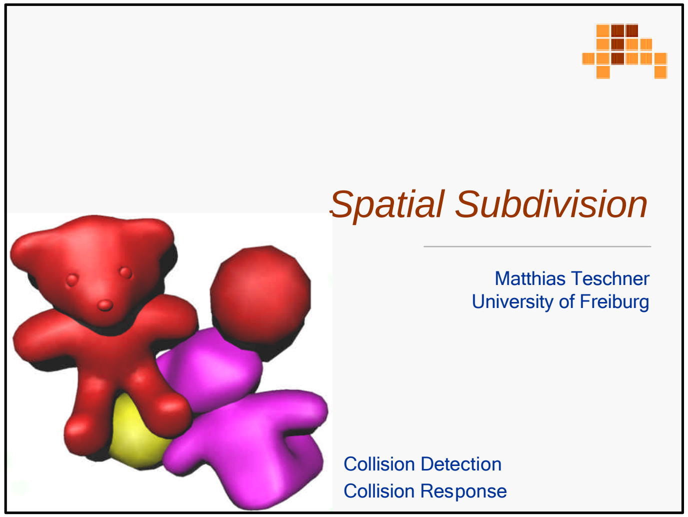
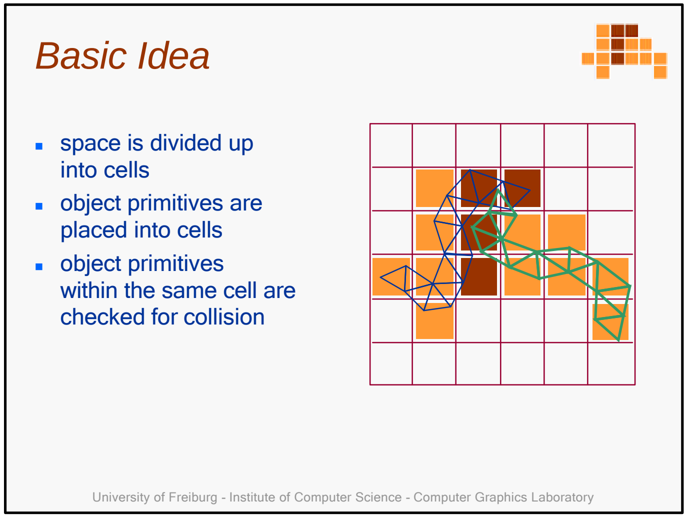


Spatial Subdivision

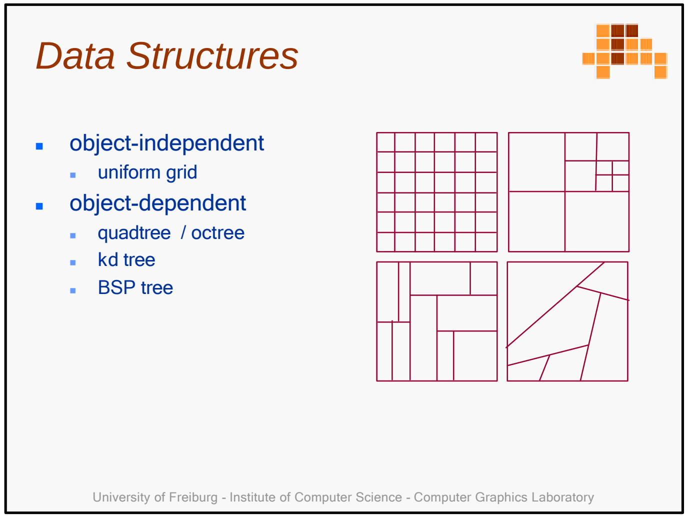
Matthias Teschner
University of Freiburg

Collision Detection
Collision Response



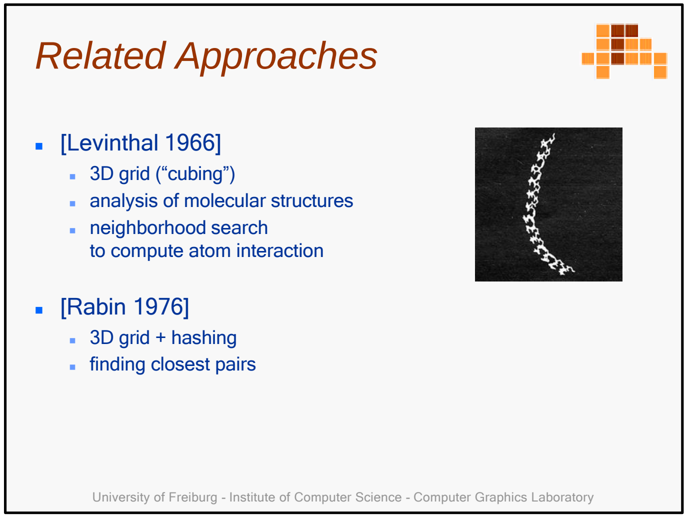
Basic Idea

- space is divided up into cells
- object primitives are placed into cells
- object primitives within the same cell are checked for collision



Data Structures

- object-independent
  - uniform grid
- object-dependent
  - quadtree / octree
  - kd tree
  - BSP tree



Related Approaches

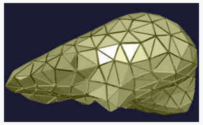
- [Levinthal 1966]
  - 3D grid (“cubing”)
  - analysis of molecular structures
  - neighborhood search to compute atom interaction
- [Rabin 1976]
  - 3D grid + hashing
  - finding closest pairs

Rigid-Body Collision Detection

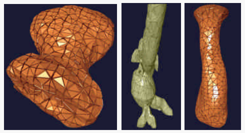
- [Turk 1989,1990]
  - 3D grid + hashing
- [Bandi, D. Thalmann 1995]
  - object-dependent spatial subdivision, octree
- [Gregory, Lin et al. 1999]
  - hybrid approach: spatial and object subdivision
- ...

Deformable Collision Detection

- [Teschner, Heidelberger et al. 2003]
  - collisions and self-collisions for deformable tetrahedral meshes
  - uniform 3D grid
  - non-uniform distribution of object primitives → hashing
  - no explicit 3D data structure
  - analysis of optimal cell size



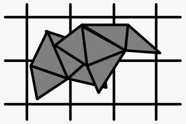
Epidaure, INRIA



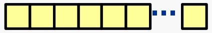
NCCR Co-Me

Algorithm - Setup

- implicit uniform grid:



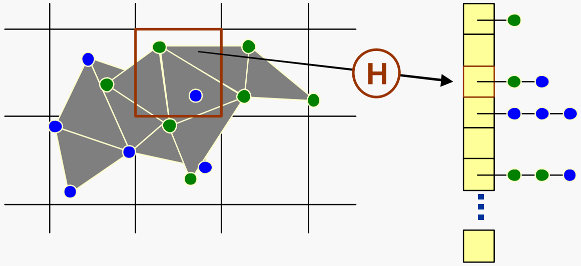
- hash function: H(cell) → hash table index
- hash table:

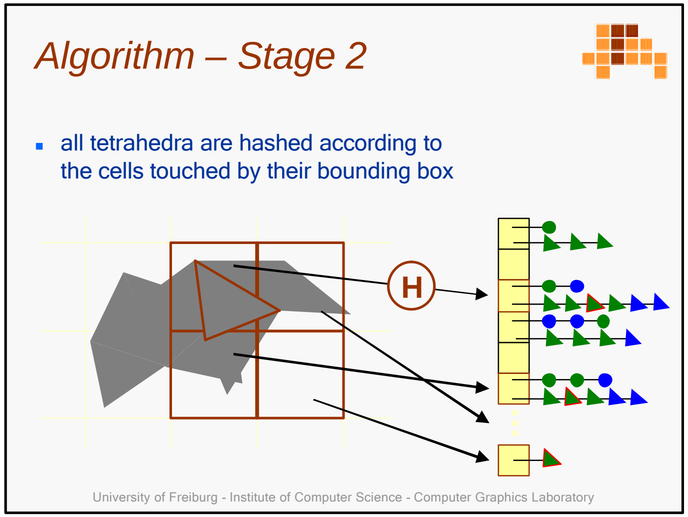


- no explicit 3D data structure
- non-uniform, sparse distribution of object primitives → hashing

Algorithm – Stage 1

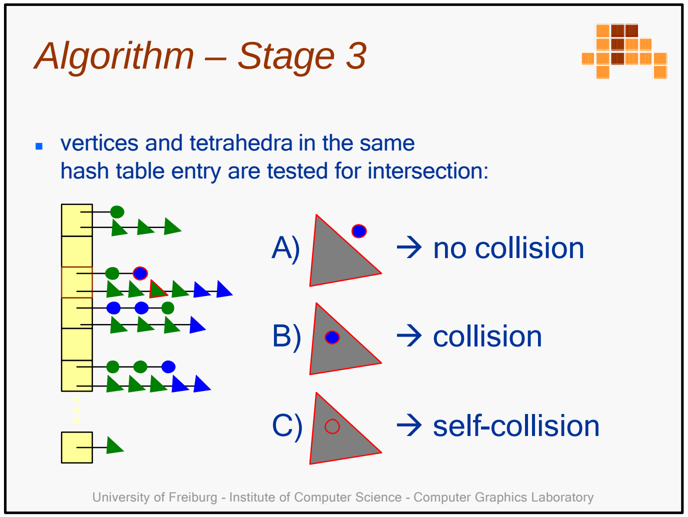
- all vertices are hashed according to their cell:





Algorithm – Stage 2

- all tetrahedra are hashed according to the cells touched by their bounding box



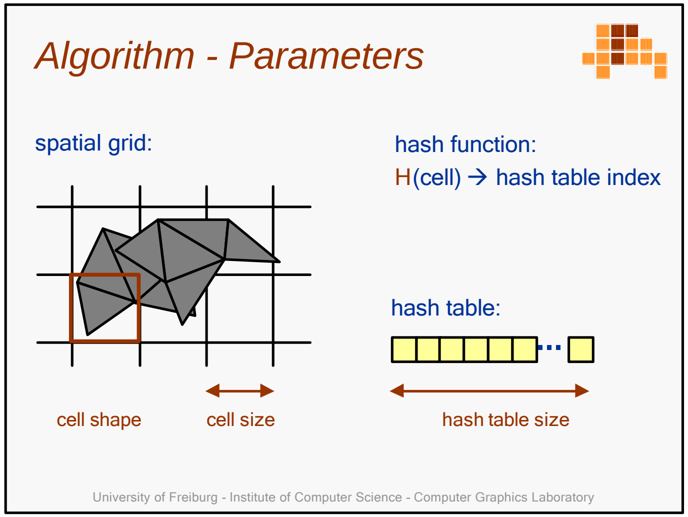
Algorithm – Stage 3

- vertices and tetrahedra in the same hash table entry are tested for intersection:

A) → no collision

B) → collision

C) → self-collision

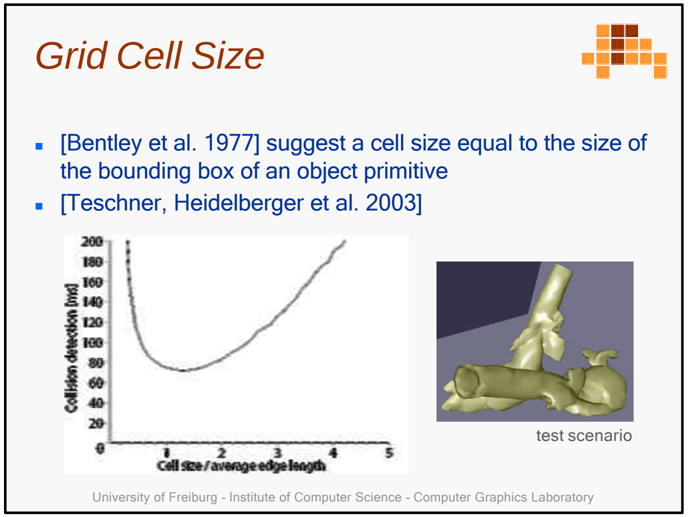


Algorithm - Parameters

spatial grid:

hash function:
H(cell) → hash table index

hash table:



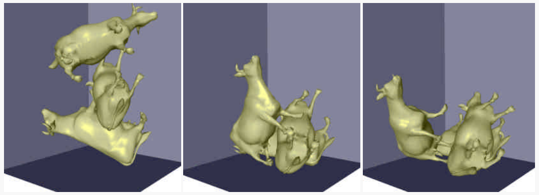
Grid Cell Size

- [Bentley et al. 1977] suggest a cell size equal to the size of the bounding box of an object primitive
- [Teschner, Heidelberger et al. 2003]

test scenario

Spatial Hashing - Summary

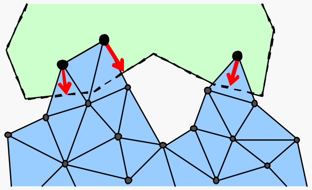
- rigid and deformable objects
- collisions, self-collisions, n-body environments
- memory efficient, interactive



Collision Response

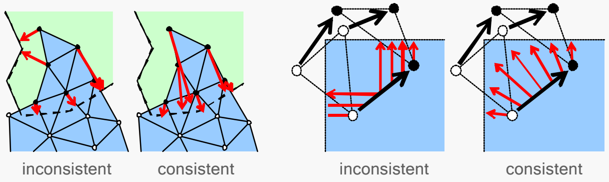
Introduction

- computation of penalty forces based on the penetration depth of intersecting vertices

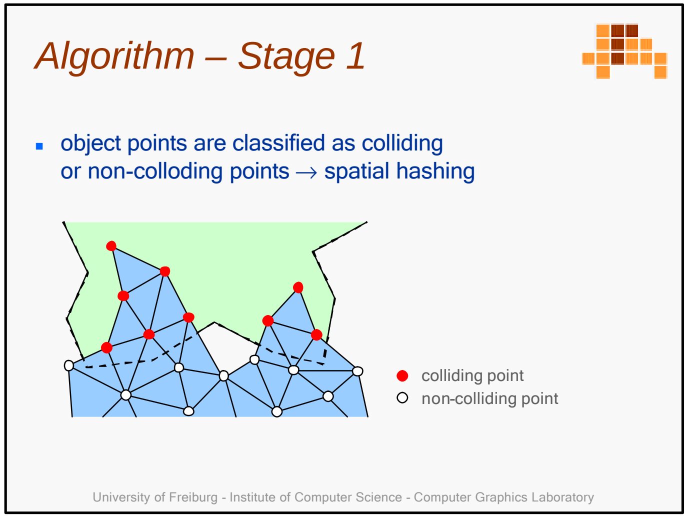


Challenges

- inconsistent penetration depth information due to discrete simulation steps and object discretization

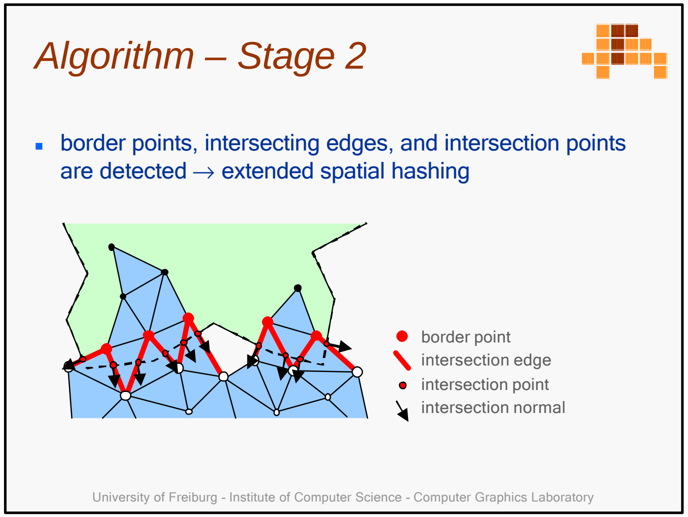


- [Heidelberger, Teschner et al. 2003]



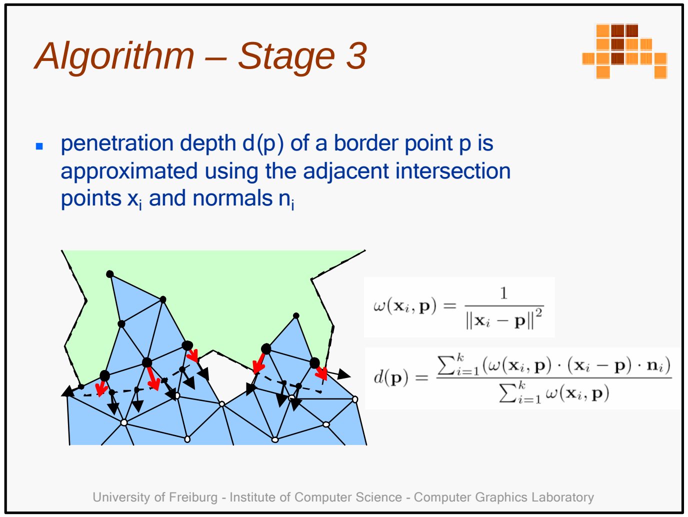
Algorithm – Stage 1

- object points are classified as colliding or non-colloding points → spatial hashing



Algorithm – Stage 2

- border points, intersecting edges, and intersection points are detected → extended spatial hashing

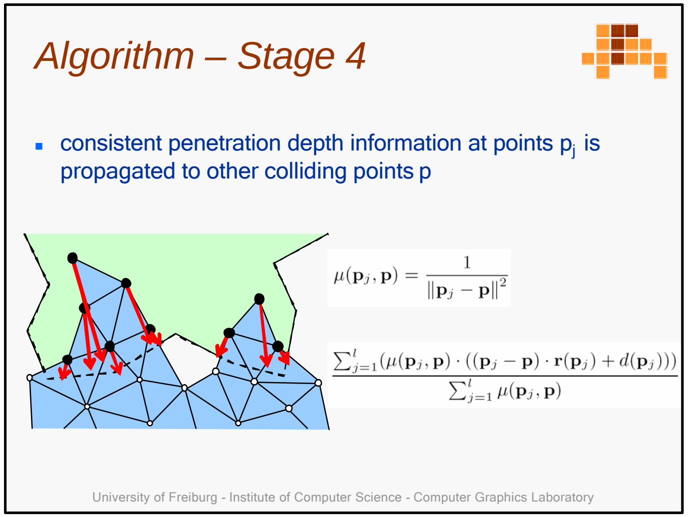


Algorithm – Stage 3

- penetration depth d(p) of a border point p is approximated using the adjacent intersection points x_i and normals n_i

$$\omega(\mathbf{x}_i, \mathbf{p}) = \frac{1}{\|\mathbf{x}_i - \mathbf{p}\|^2}$$

$$d(\mathbf{p}) = \frac{\sum_{i=1}^{k} (\omega(\mathbf{x}_i, \mathbf{p}) \cdot (\mathbf{x}_i - \mathbf{p}) \cdot \mathbf{n}_i)}{\sum_{i=1}^{k} \omega(\mathbf{x}_i, \mathbf{p})}$$



Algorithm – Stage 4

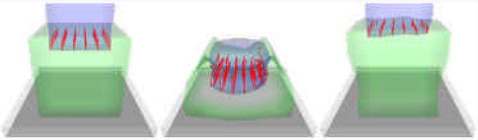
- consistent penetration depth information at points p_j is propagated to other colliding points p

$$\mu(\mathbf{p}_j, \mathbf{p}) = \frac{1}{\|\mathbf{p}_j - \mathbf{p}\|^2}$$

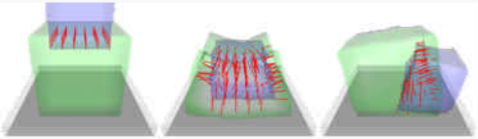
$$\frac{\sum_{j=1}^{l} (\mu(\mathbf{p}_j, \mathbf{p}) \cdot ((\mathbf{p}_j - \mathbf{p}) \cdot \mathbf{r}(\mathbf{p}_j) + d(\mathbf{p}_j)))}{\sum_{j=1}^{l} \mu(\mathbf{p}_j, \mathbf{p})}$$

Collision Response - Results

- consistent collision response

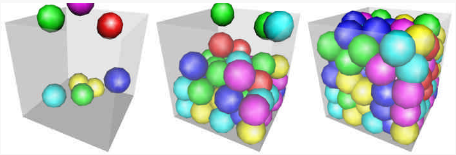


- inconsistent collision response



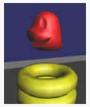
Collision Response - Summary

- penalty-based
- consistent penetration depth information
  - discrete object representation
  - discrete time simulation
- n-body collision response



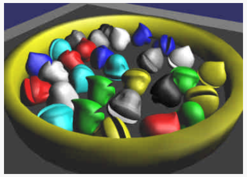
Interacting Deformable Objects

- collision detection based on spatial hashing
- collision response based on consistent penetration depth computation



squeeze

trampoline



References

- M. Teschner, B. Heidelberger et al., “Optimized Spatial Hashing for Collision Detection of Deformable Objects,” Proc. VMV, Munich, Germany, 2003.
- B. Heidelberger, M. Teschner et al., “Consistent Penetration Depth Estimation for Deformable Collision Response,” Proc. VMV, Stanford, USA, 2004.
- M. Teschner, B. Heidelberger et al., “A Versatile and Robust Model for Geometrically Complex Deformable Solids,” Proc. CGI, Crete, Greece, 2004.

- http://cg.informatik.uni-freiburg.de/
- http://graphics.ethz.ch/~brunoh/