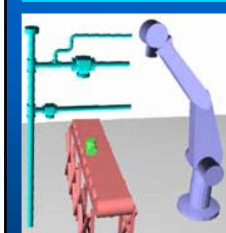
Robot Algorithms for Medical Simulation & Virtual Prototyping

Ming C. Lin
Department of Computer Science
University of North Carolina at Chapel Hill
http://www.cs.unc.edu/~lin
http://gamma.cs.unc.edu/

Motion

- Ubiquitous in physical world and virtual environments
- Arising from dynamical systems: biological entities or man-made structures or mechanisms

Robotics & Automation

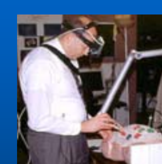

Assembly Planning (UNC)

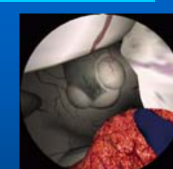
Robot/Computer Assisted Surgery(JHU)



Dream Robot (Sony)

Medical Applications


AR for Surgery (UNC/Ultrasound)

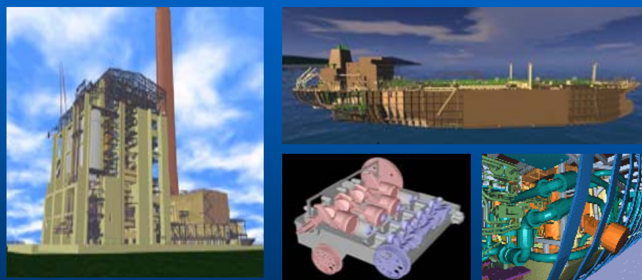

Third Ventriculostomy (HT/Immersion Medical)


da Vinci Surgical System (Intuitive Surgery, Inc.)


Imaging (UNC/MIDAG)

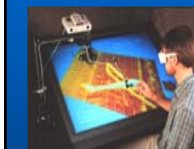

Human Simulation (BDI)

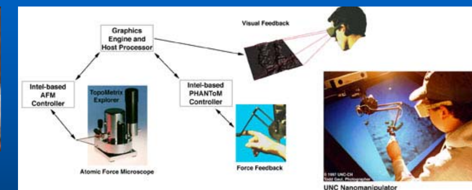
Engineering Design

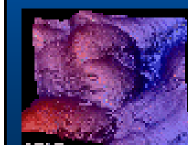


Interactive Prototyping of Massive Structures
Walkthrough Research Group (UNC)

Scientific Visualization







Interactive Visualization & Manipulation of nanoStructures
nanoManipulator Research Group (UNC)

Virtual Environments

Interactive Agents and Avatars (UPENN/HMS)

Walking Experiment & the Pit
EVE Research Group (UNC)

Training & Education

VET (ISI/USC)

Team Training (NRL)

Games & Entertainment

Medal of Honor™ (Pacific Assault)

Monster Inc. (Pixar Animation Studio)

IPAM-VS 2008 Ming C. Lin

Robot Algorithms

- Sensing and vision
- Motion control
- Dynamics and kinematics
- Motion planning
- Manipulation
- Others….

IPAM-VS 2008 Ming C. Lin

Organization

- Articulated Bodies
- Deformable Models

IPAM-VS 2008 Ming C. Lin

Organization

- Articulated Bodies
- Deformable Models

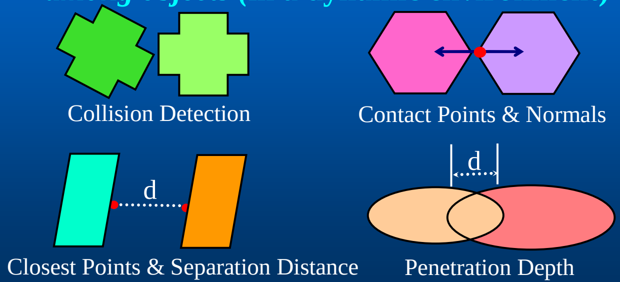
IPAM-VS 2008 Ming C. Lin

Algorithms for Articulated Bodies

- Continuous Collision Detection for Articulated Models
- Adaptive Forward Dynamics (AD)
- Planning of Highly Articulated Body Using AD + Contact Handling

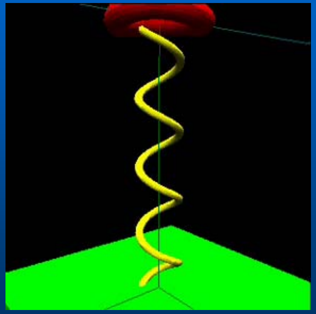
Collision & Proximity Queries

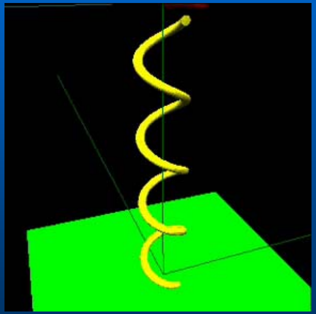
Geometric reasoning of spatial relationships among objects (in a dynamic environment)



⇒ Can take a ***high fraction*** of the running time in simulations

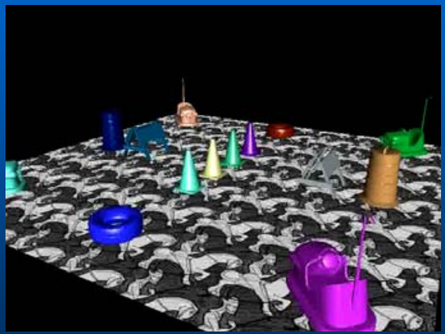
SWIFT++: Distance & Collision Queries





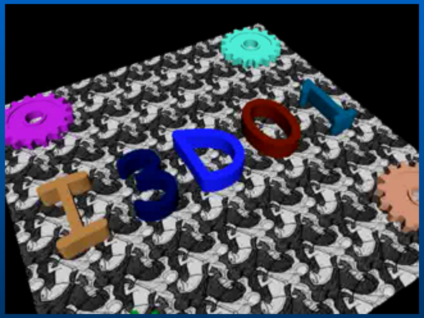
Use of Multi-resolution Reps. & Coherence
[Ehmann & Lin, EG 2001]

PIVOT: Proximity Computations



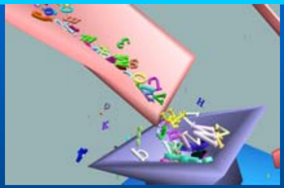
Proximity Queries Using Graphics Hardware Acceleration
[Hoff, et al, I3D 2001]

Real-Time Simulation of Complex Objects Using PIVOT on GPU

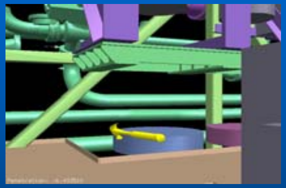


http://gamma.cs.unc.edu/PIVOT

Penetration Depth Computation

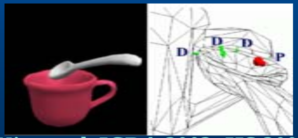


Dynamic Simulation



Virtual Prototyping

Haptic Rendering

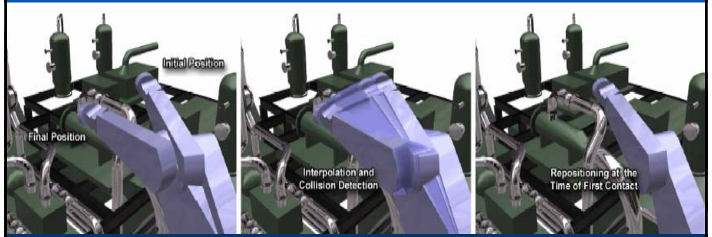


DEEP: [Kim et al, ICRA 2002; HS 2002]

PD: [Kim et al, SCA 2002]

Continuous Collision Detection for Articulated Models Using GPUs

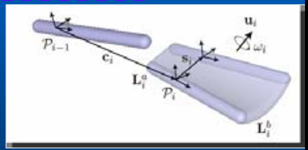
Motivation: Prior discrete algorithms only check for collision at each time instance, not in between steps. Collisions can be missed for fast moving and/or thin objects, such as an avatar rapidly moving around in a VE.



http://gamma.cs.unc.edu/Articulate

[Redon, Kim, Lin, Manocha; SM 2004]

Overview

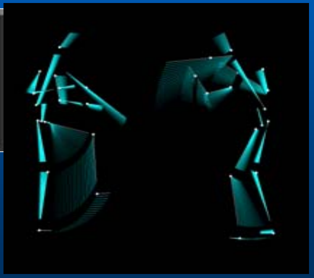


$$\mathbf{M}_i^{i-1}(t) = \begin{pmatrix} \mathbf{P}_i^{i-1}(t) & \mathbf{T}_i^{i-1}(t) \\ (0,0,0) & 1 \end{pmatrix}$$

$$\mathbf{T}_i^{i-1}(t) = \mathbf{c}_i + t\mathbf{s}_i$$

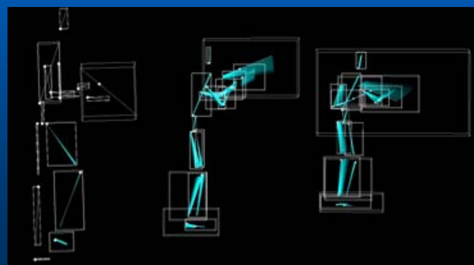
$$\mathbf{P}_i^{i-1}(t) = \cos(\omega_i t).\mathbf{A}_i + \sin(\omega_i t).\mathbf{B}_i + \mathbf{C}_i$$

$$\mathbf{M}_i^{0}(t) = \mathbf{M}_1^{0}(t).\mathbf{M}_2^{1}(t)...\mathbf{M}_i^{i-1}(t)$$



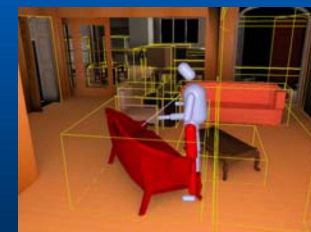
Overview

Motion interpolation → BVH Construction → BVH Culling → SV Generation → Collision Checking and TOC Estimation



Overview

Motion interpolation → BVH Construction → BVH Culling → SV Generation → Collision Checking and TOC Estimation



We use the dynamic AABBs to cull away the links far from the environment

Overview

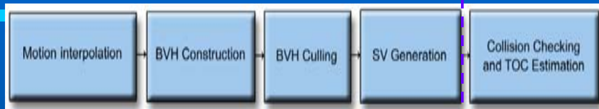
Motion interpolation → BVH Construction → BVH Culling → SV Generation → Collision Checking and TOC Estimation

We use a line-swept sphere volume to represent each link and compute its swept volume for all links of the avatar, as shown on the right (blue/green indicating initial/final position).

Overview

Motion interpolation → BVH Construction → BVH Culling → SV Generation → Collision Checking and TOC Estimation

Overview



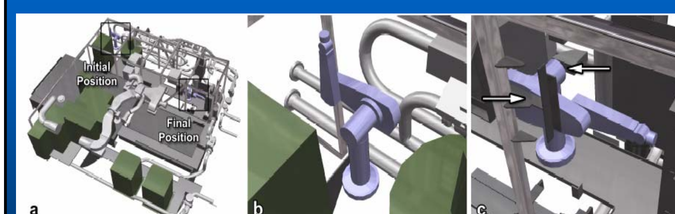
Overview

Mobile Interaction in VE

An avatar moving around rapidly in a living room; Real-time continuous collision detection is used to check for possible interference with the furniture.

[IEEE VR 2004]

Local Planning in Contact Space



Planning the motion of a Puma Robot (800 triangles, 7 dofs) in a partial Auxiliary Machine Room model (117,000 triangles)

Two orders of magnitude performance improvement

Adaptive Forward Dynamics

- Adaptive computation of dynamic (or quasi-static) simulations of complex linkages or articulated bodies:
  - Error-bounded Approximation
    Introducing "total acceleration equations" to approximate Linkage Acceleration
  - Progressive Refinement
    Using hierarchical reference frames to lazily update the linkage states
  - Efficiency
    Avoiding linear-time update of all linkages

[Redon et al, SIGGRAPH 2005]

IPAM-VS 2008 Ming C. Lin

Main Results

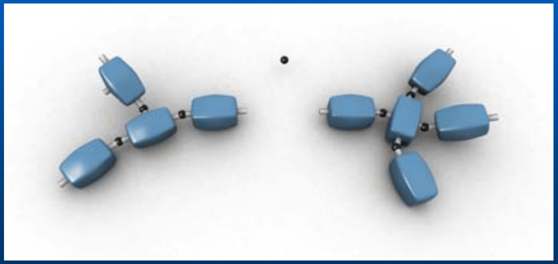
- **Hybrid bodies**
  - Novel articulated-body representation
  - To reduce the no. of degrees of freedom (DOFs)
- **Adaptive joint selection**
  - Novel customizable motion metrics
  - To determine the most important DOFs
- **Adaptive update mechanisms**

[Redon, Galoppo, Lin; SIGGRAPH 2005]

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA [1999]

- Recursive definition



An articulated body is formed by assembling two articulated bodies

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

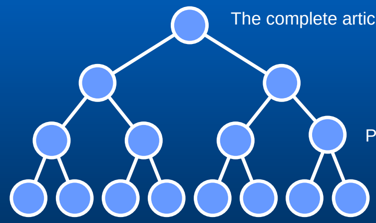
- Recursive definition

An articulated body is formed by assembling two articulated bodies

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

- Recursive definition



The assembly tree of an articulated body

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

- Recursive definition
- Articulated-body equation

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_{12} & \cdots & \Phi_{1m} \\ \Phi_{21} & \Phi_2 & \cdots & \Phi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{m1} & \Phi_{m2} & \cdots & \Phi_m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}$$

Body Accelerations — Inverse inertias and cross-inertias — Applied Forces — Bias accelerations

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

- Recursive definition
- Articulated-body equation

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_{12} & \cdots & \Phi_{1m} \\ \Phi_{21} & \Phi_2 & \cdots & \Phi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{m1} & \Phi_{m2} & \cdots & \Phi_m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}$$

The cross-coupling inverse inertia Φ_{12} describes the effect of a force $\mathbf{f}_2$ applied to body 2, on the acceleration $\mathbf{a}_1$ of body 1

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

- Recursive definition
- Articulated-body equation

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_{12} & \cdots & \Phi_{1m} \\ \Phi_{21} & \Phi_2 & \cdots & \Phi_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{m1} & \Phi_{m2} & \cdots & \Phi_m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_m \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \end{bmatrix}$$

The bias acceleration $\mathbf{b}_1$ is the acceleration of body 1 when no forces are applied

IPAM-VS 2008 Ming C. Lin

Hybrid bodies
Featherstone's DCA

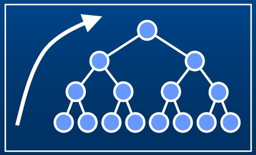
- Recursive definition
- Articulated-body equation
- Two main steps
  - Compute the articulated-body coefficients (↑)

Inverse inertias

$$\Phi_1^C = \Phi_1^A - \Phi_{12}^A \mathbf{W} \Phi_{21}^A$$
$$\Phi_2^C = \Phi_2^B - \Phi_{21}^B \mathbf{W} \Phi_{12}^B$$
$$\Phi_{21}^C = \Phi_{21}^B \mathbf{W} \Phi_{21}^A = (\Phi_{12}^C)^T$$

Bias accelerations

$$\mathbf{b}_1^C = \mathbf{b}_1^A - \Phi_{12}^A \gamma$$
$$\mathbf{b}_2^C = \mathbf{b}_2^B + \Phi_{21}^B \gamma$$



Hybrid bodies
Featherstone's DCA

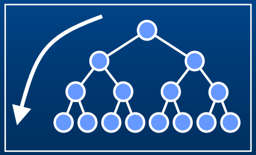
- Recursive definition
- Articulated-body equation
- Two main steps
  - Compute the joint accelerations and forces (↓)

Joint acceleration

$$\ddot{\mathbf{q}} = (\mathbf{S}^T \mathbf{V} \mathbf{S})^{-1} (\mathbf{Q} - \mathbf{S}^T \mathbf{V} (\Phi_{21}^A \mathbf{f}_1^A - \Phi_{12}^B \mathbf{f}_2^B + \beta))$$

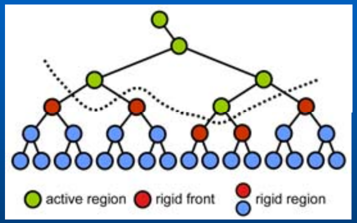
Kinematic constraint forces

$$\mathbf{f}_1^B = \mathbf{W} \Phi_{21}^A \mathbf{f}_1^A - \mathbf{W} \Phi_{12}^B \mathbf{f}_2^B + \gamma$$
$$\mathbf{f}_2^A = -\mathbf{f}_1^B$$



Hybrid bodies
Definitions

- Active region



The active region contains the mobile joints

Hybrid bodies
Definitions

- Active region 
- Hybrid-body coefficients

$$\Phi_1^C = \Phi_1^A - \Phi_{12}^A \mathbf{W} \Phi_{21}^A \qquad \mathbf{b}_1^C = \mathbf{b}_1^A - \Phi_{12}^A \gamma$$
$$\Phi_2^C = \Phi_2^B - \Phi_{21}^B \mathbf{W} \Phi_{12}^B \qquad \mathbf{b}_2^C = \mathbf{b}_2^B + \Phi_{21}^B \gamma$$
$$\Phi_{21}^C = \Phi_{21}^B \mathbf{W} \Phi_{21}^A = (\Phi_{12}^C)^T$$

Articulated-body coefficients

↓ Rigidify joint

$$\Phi^C = \Phi_i^C = \Phi_{ij}^C = \Phi^B (\Phi^A + \Phi^B)^{-1} \Phi^A$$
$$\mathbf{b}^C = \mathbf{b}_i^C = \mathbf{b}^A - \Phi^A (\Phi^A + \Phi^B)^{-1} (\mathbf{b}^A - \mathbf{b}^B)$$

Hybrid-body coefficients

Hybrid bodies
Definitions

- Active region 
- Hybrid-body coefficients
- Hybrid-body simulation
  - Same steps as articulated-body simulation
  - Computations restricted to a sub-tree (*cf.* paper)

Adaptive joint selection
Motion metrics

- Acceleration metric

$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i$$

- Velocity metric

$$\mathscr{V}(C) = \sum_{i \in C} \dot{\mathbf{q}}_i^T \mathbf{V}_i \dot{\mathbf{q}}_i$$

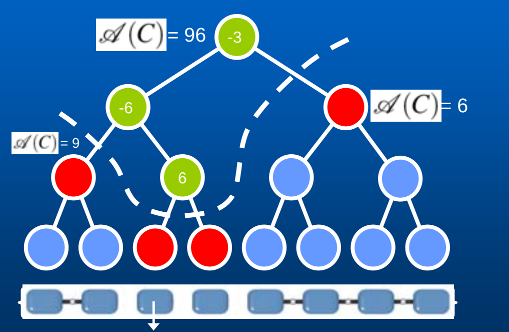
Adaptive joint selection
Motion metrics

Theorem

The acceleration metric value of an articulated body can be computed before **computing its joint accelerations:**

$$\mathscr{A}(C) = \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix}^T \begin{bmatrix} \Psi_1^C & \Psi_{12}^C \\ \Psi_{21}^C & \Psi_2^C \end{bmatrix} \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix} + \begin{bmatrix} \mathbf{f}_1^A \\ \mathbf{f}_2^B \end{bmatrix}^T \begin{bmatrix} \mathbf{p}_1^C \\ \mathbf{p}_2^C \end{bmatrix} + \eta^C$$

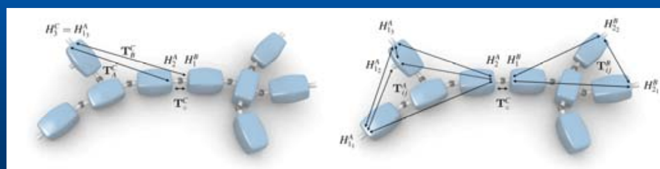
Adaptive joint selection
Acceleration simplification



Four subassemblies with joint accelerations implicitly set to zero

Adaptive update mechanisms

- Position-dependent coefficients
- Hierarchical state representation



[Redon and Lin; SPM 2005]

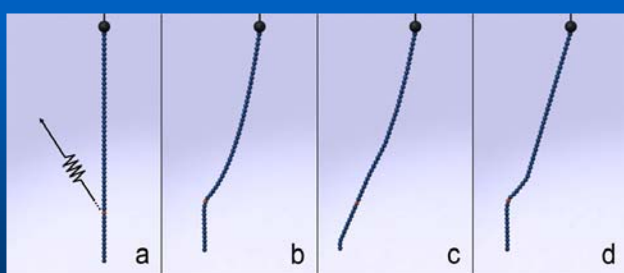
Adaptive update mechanisms

- Velocity-dependent coefficients
- Linear coefficients tensors

$$(\mathbf{b}_1^C)_a = (\mathbf{B}_1^C)_{abc}(\mathbf{v}_1^C)_b(\mathbf{v}_1^C)_c \qquad (\mathbf{p}_1^C)_a = (\mathbf{P}_1^C)_{abc}(\mathbf{v}_1^C)_b(\mathbf{v}_1^C)_c$$
$$(\mathbf{b}_2^C)_a = (\mathbf{B}_2^C)_{abc}(\mathbf{v}_2^C)_b(\mathbf{v}_2^C)_c \qquad (\mathbf{p}_2^C)_a = (\mathbf{P}_2^C)_{abc}(\mathbf{v}_2^C)_b(\mathbf{v}_2^C)_c$$
$$\eta^C = (\mathbf{E}^C)_{abcd}(\mathbf{v}^C)_a(\mathbf{v}^C)_b(\mathbf{v}^C)_c(\mathbf{v}^C)_d$$

Results

Adaptive joint selection



Adaptive determination of the active region

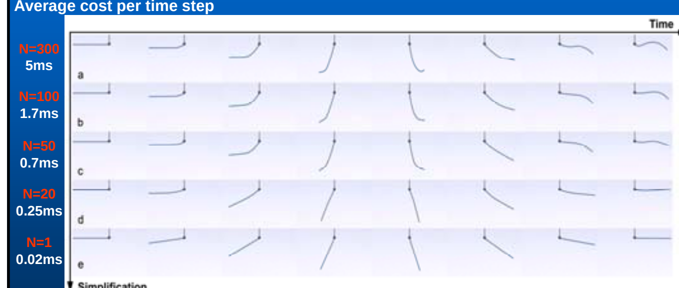
Achieved 10x speed-up

Results

Progressive dynamics

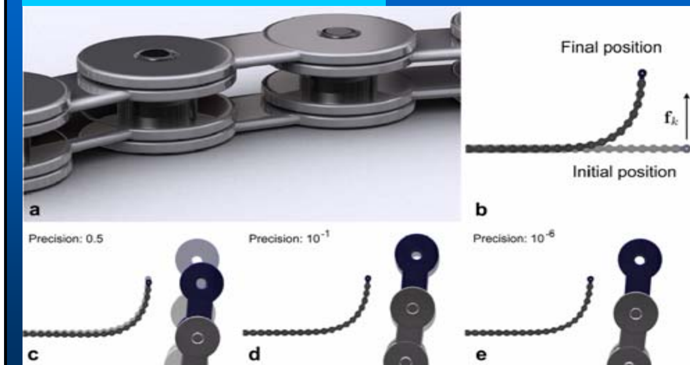
N = Number of active joints

Average cost per time step



Progressive dynamics of a 300-link pendulum

Design of Mechanical Chains

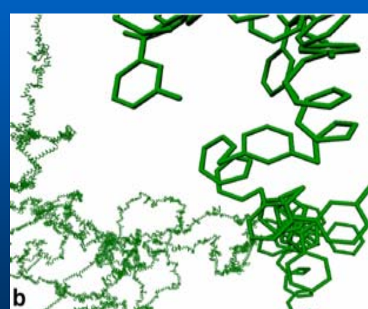


Simulation of Articulated Robots

L: Individual component
R: The millipede robot

Interactive Manipulation of Molecular Chains

L: The entire chain
R: A close-up view



Results

Test application

Achieved **two orders of magnitude** performance gain
For 200 avatars with 17,800 rigid bodies and 19,000 dofs

Adaptive Dynamics (AD) with Contact Handling

- Use impulse-based dynamics
- Solve *p* analytical constraints:

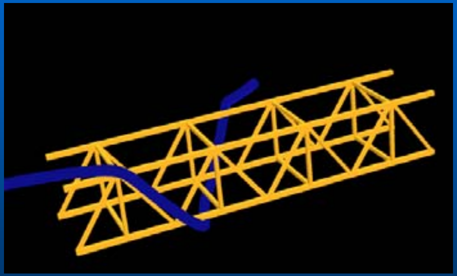
$$h(\ddot{q}^f) - h(\ddot{q}^o) = k f_i \quad \Rightarrow \quad \mathbf{h} - \mathbf{h}^o = \mathbf{K}\mathbf{f}$$

- Derive a new hybrid-body Jacobian:

$$J_P(q(t)) = \begin{pmatrix} \frac{\delta x}{\delta q_1} & \cdots & \frac{\delta x}{\delta q_{d_n}} \\ \frac{\delta y}{\delta q_1} & \cdots & \frac{\delta y}{\delta q_{d_n}} \\ \frac{\delta z}{\delta q_1} & \cdots & \frac{\delta z}{\delta q_{d_n}} \end{pmatrix}$$

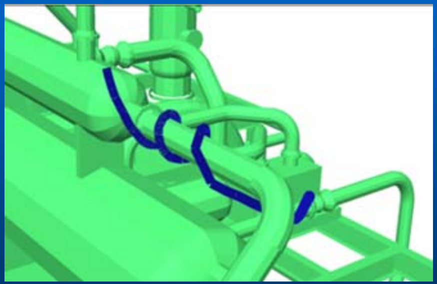
[Gayle et al; RSS 2006]

Cable Routing on Bridge



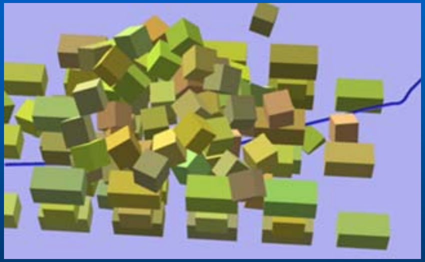
A snake robot of 500 links and 500 DOFs with only 70 DOFs used in this bridge scene

Pipe Inspection



A snake robot of 2000 links with only 200 DOFs used

Search & Rescue in Debri



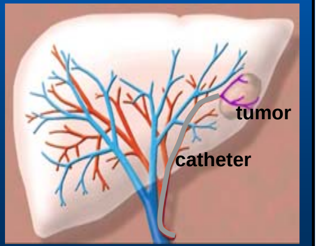
A snake robot of 2000 links with only 200 DOFs used

Catheterization Procedures

- In medical and surgical procedures, flexible catheters are often inserted in human vessels to
  - Obtain diagnostic information (blood pressure or flow)
  - Enhance imaging with the injection of contrast agents
  - Provide a mechanism to deliver treatment to a specific area

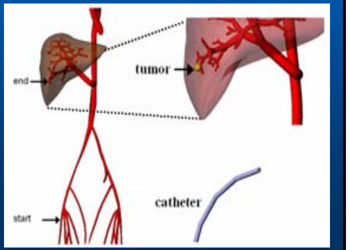
Liver Chemoembolization

- Catheter is used to inject chemotherapy drugs directly to the blood vessel supplying a liver tumor
- Catheter is inserted into the femoral artery (near the groin) and advanced into the selected liver artery
  - A fluoroscopic display and the resistance felt from the catheter are used to determine how it should be advanced, withdrawn, or rotated
- Chemotherapy drugs followed by embolizing agents are injected through the catheter into the liver tumor

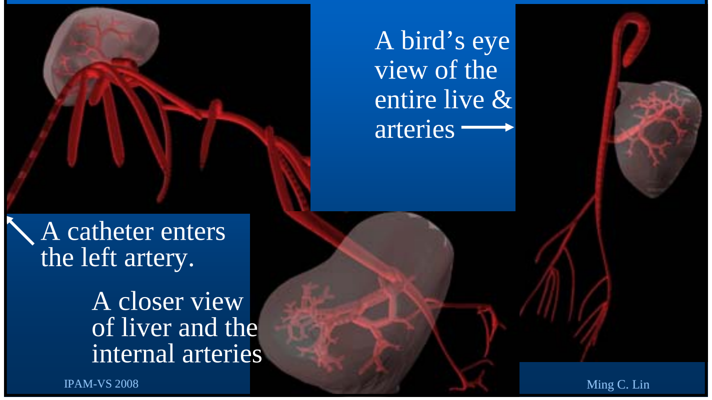


Motion Planning Application

- Application to plan the path of a flexible catheter, inserted at the femoral artery, to a specific liver artery supplying a tumor
  - Environment: 3D models of the liver and blood vessels obtained from the 4D NCAT phantom, a realistic computer model of the human body
  - Catheter was modeled as a snake robot with 2,500 joints with only 10% of joints simulated to achieve 10x speed up.



Benchmark: Liver (Courtesy of JHU)



A bird's eye view of the entire live & arteries →

A catheter enters the left artery.

A closer view of liver and the internal arteries

Catheter Insertion

System Demonstration

[Gayle et al; ICRA 2007]

Organization

- Articulated Bodies
- Deformable Models

Algorithms for Deformable Models

- Collision Detection for Deformable Models using Chromatic Decomposition
- Constraint-based Planning of Deformable Robots
- Modeling of Soft Articulated Bodies in Contact Using Dynamic Deformation Textures

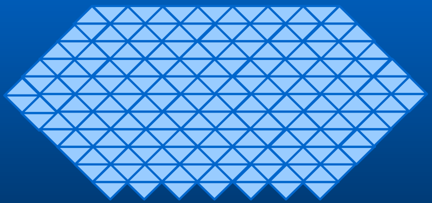
Collision Detection for Deformable Models using Chromatic Decomposition

- General collision detection algorithm
  - Robust and efficient
  - Continuous self-collision detection
  - Works on all triangulated meshes with fixed mesh connectivity
- Up to an order of magnitude speedup over prior algorithms

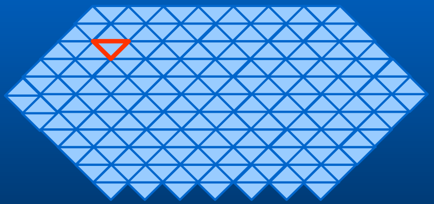
Hierarchical Approaches

- Spatial partitioning or bounding volume hierarchies
  - **AABBs** [Bridson et al. 02, Baraff et al. 03, DeRose et al. 98], **OBBs, k-DOPs** [Mezger et al. 02, Volino and Thalmann 00]
- Issues
  - Trade-off between speed and culling efficiency
  - Overlap tests are conservative, resulting in many false positives

Self-Collision Detection: Complexity

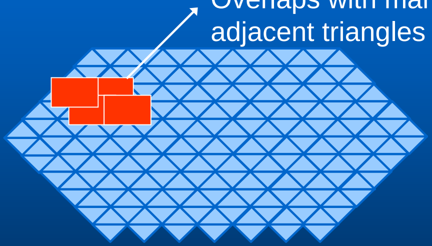


Self-Collision Detection: Complexity



Self-Collision Detection: Complexity

Overlaps with many adjacent triangles



Overlaps using AABBs
(Axis Aligned Bounding Box)

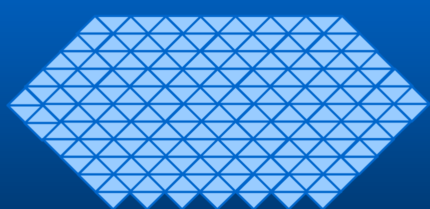
GPU-based Algorithms

- Use rasterization and involve no pre-processing
  - Applicable to deformable models
    [Baciu and Wong 2002, Hiedelberger et al. 04, Govindaraju et al. 05, Rossignac et al. 92, Vassilev et al. 01]
- Issues
  - Self-collision computations at image resolution

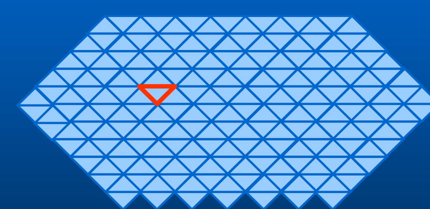
Algorithm Overview

- Problem decomposition
  - Non-adjacent collision detection (NACD)
  - Adjacent collision detection (ACD)

Problem Decomposition

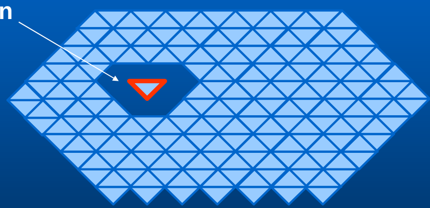


Problem Decomposition: NACD



Problem Decomposition: NACD

Separation



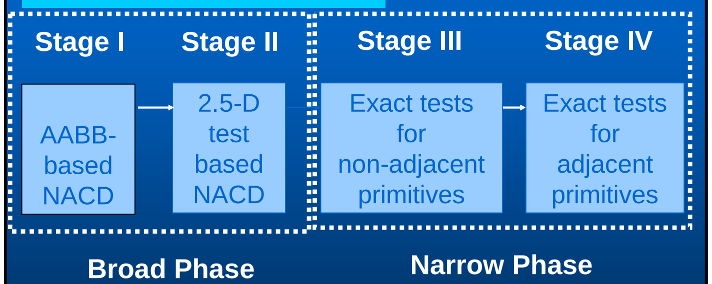
Problem Decomposition: ACD

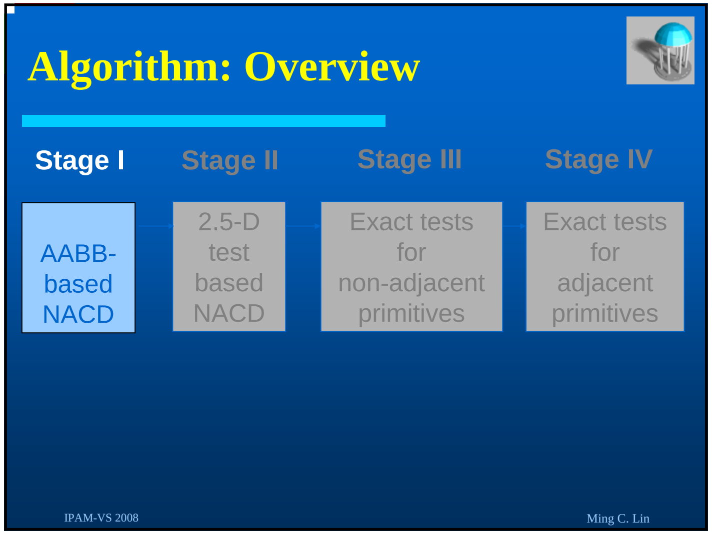
Based on mesh connectivity

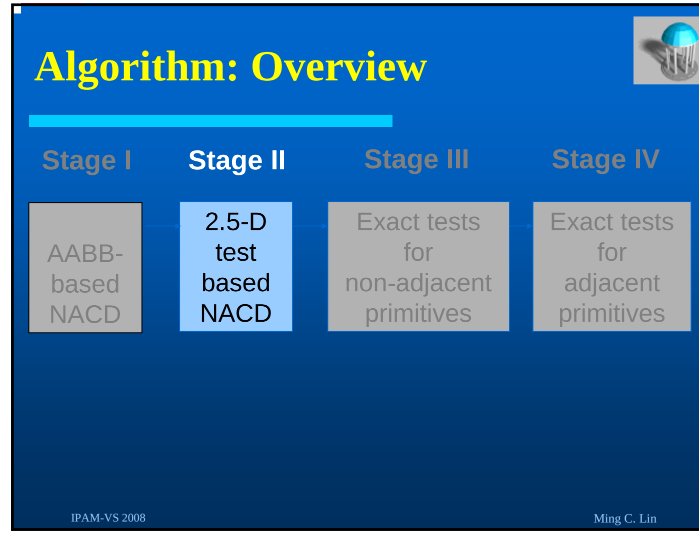
Problem Decomposition

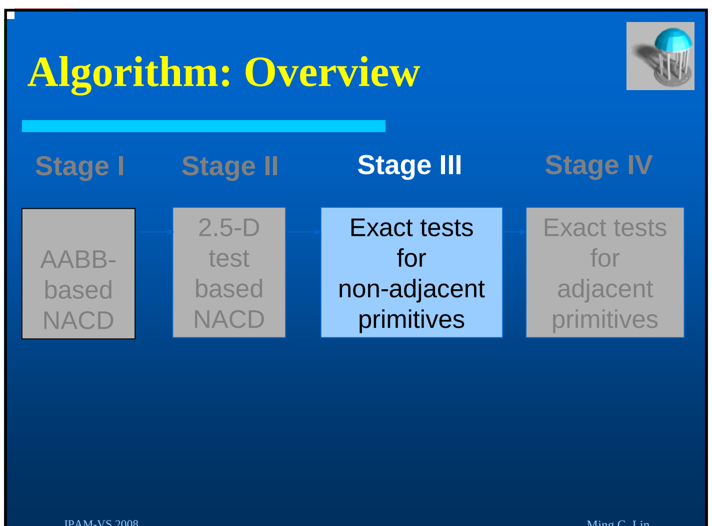
- **Decomposition into NACD and ACD**
  - **Significant collision culling**
  - **Fewer false positives**

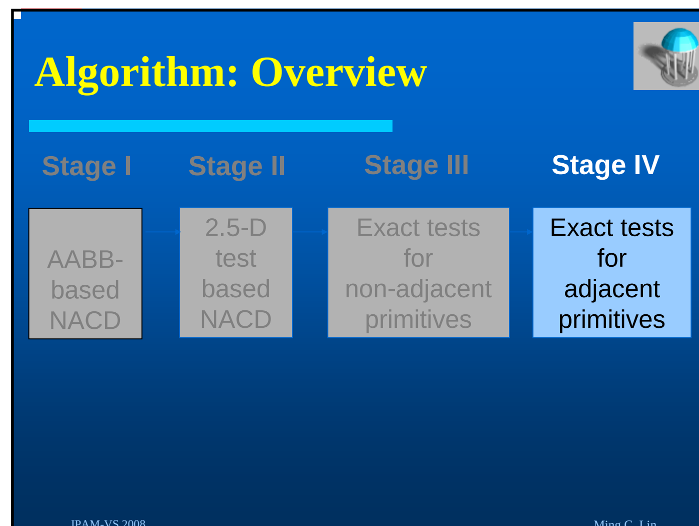
Algorithm: Overview




Algorithm: Overview
Stage I
Stage II
Stage III
Stage IV
AABB-based NACD
2.5-D test based NACD
Exact tests for non-adjacent primitives
Exact tests for adjacent primitives
IPAM-VS 2008
Ming C. Lin


Algorithm: Overview
Stage I
Stage II
Stage III
Stage IV
AABB-based NACD
2.5-D test based NACD
Exact tests for non-adjacent primitives
Exact tests for adjacent primitives
IPAM-VS 2008
Ming C. Lin


Algorithm: Overview
Stage I
Stage II
Stage III
Stage IV
AABB-based NACD
2.5-D test based NACD
Exact tests for non-adjacent primitives
Exact tests for adjacent primitives
IPAM-VS 2008
Ming C. Lin


Algorithm: Overview
Stage I
Stage II
Stage III
Stage IV
AABB-based NACD
2.5-D test based NACD
Exact tests for non-adjacent primitives
Exact tests for adjacent primitives
IPAM-VS 2008
Ming C. Lin

Algorithm: Overview

Runtime:

Stage I	Stage II	Stage III	Stage IV
AABB-based NACD	2.5-D test based NACD	Exact tests for non-adjacent primitives	Exact tests for adjacent primitives

Pre-process:

- AABB-hierarchy construction (→ Stage I)
- Mesh decomposition (→ Stage II)

Mesh Decomposition

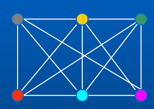
- Mesh decomposition is similar to vertex coloring
- Vertex Coloring: Color vertices such that adjacent vertices have different colors

Mesh Decomposition vs. Coloring

	Mesh Decomposition	Vertex Coloring
Partitions	Triangles	Vertices
Sets are	disjoint	disjoint
Partitioned Sets have	Non-adjacent triangles	Non-adjacent vertices

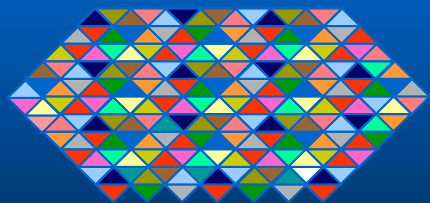
Mesh Decomposition

Mesh



Extended Dual Graph

Mesh Decomposition



Vertex Coloring

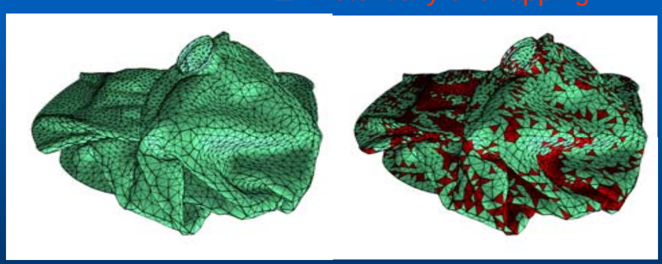
- Minimum vertex coloring is NP-complete
- Use greedy heuristic algorithms based on vertex degree (DSATUR)
  - Typically around 10-20 sets

Stage I : AABB-based NACD

- Update AABB hierarchy
- Check AABB hierarchy against itself
  - Only check for NACD

Culling Efficiency of Stage I

Potentially overlapping

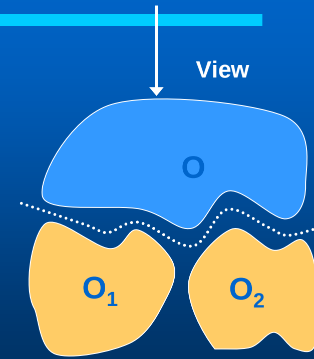


With Stage I AABBs:
20K elementary tests

Stage II : 2.5-D based NACD

- Uses 2.5-D visibility tests on GPUs
  [Govindaraju et al. 2003]
- 2.5-D tests: Objects are tested for full visibility against a set of objects
- Used for set-based collision culling

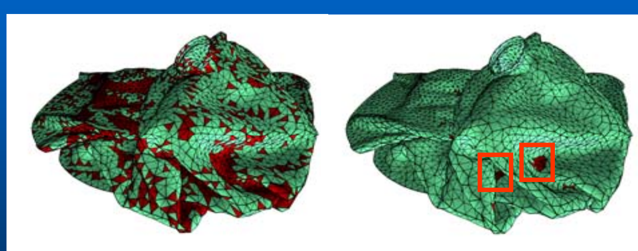
2.5-D Tests: Geometric Interpretation



Checks for existence of separating surface

Performed on GPU
Sufficiently *fatten* to account for precision errors [**Govindaraju et al; VRST 04**]

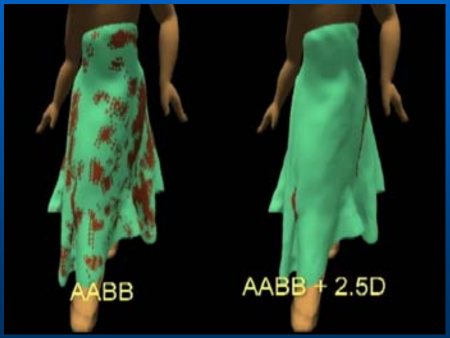
Culling Efficiency after Stage II



After Stage I:
20K elementary tests

After Stage II:
1K elementary tests

Improved Culling: 2.5D tests



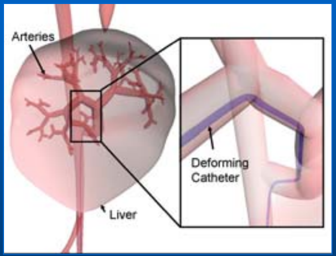
Dancing Girl: Cloth Simulation

- Skirt – 12.5 K triangles
- 0.55 *sec* self-collision detection time

Walking Girl: Cloth Simulation

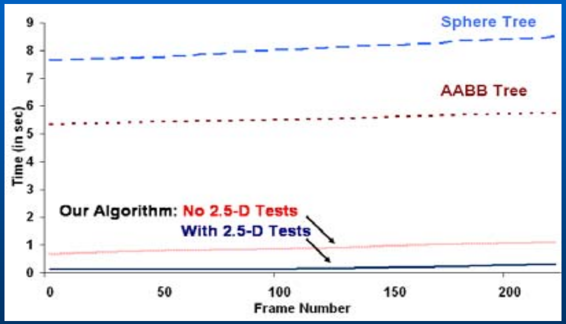
- Skirt – 12.5 K triangles
- 0.55 *sec* self-collision detection time

Application: Surgical Simulation

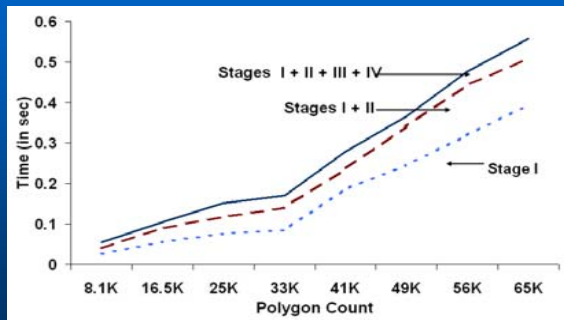


- Path planning for a deformable catheter
- Liver – 83K triangles : Catheter – 10K triangles
- 60~90 *msec* collision detection time between catheter and arteries

Speed Improvement



Speed Scales Linearly with Polygon Count

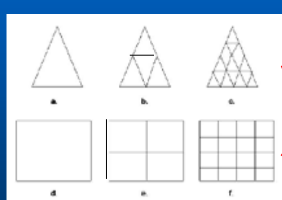


Multiresolution Collision Handling

- Precompute multiresolution meshes
  - Subdivision Hierarchy
  - Chromatic Decomposition
  - Use of Graphics Processor Units (GPUs)
- Adaptively select appropriate CLODs
- Refine simulation on the fly
- Accelerate overall computation by 7x

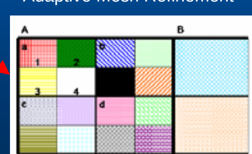
[Jain et al; CASA 2005]

Multiresolution Hierarchy



Subdivision Hierarchy

Adaptive Mesh Refinement



Benchmarks

Folding curtains (33 K tris total)

Squatting human with a pair of pants (10K tris)

Cloth (33 K triss) draping over a bunny

Planning of Deformable Robots

Motivations

– Surgical planning

– Layout for mechanical/electrical systems in complex structures

– Planning of reconfigurable robots

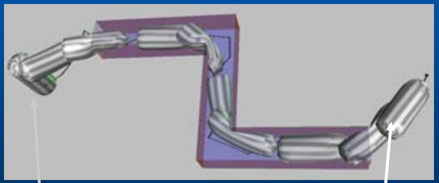
– Character animation

Motion Planning to Aid Catheterization Procedure

- **Accurate motion planning studies with deformable models can provide a vital tool to aid in the catheterization procedure**
- **Preoperatively, they may be used as part of surgical planning techniques to help choose the size and properties of the catheter used**
- **During the actual procedure, they can be used to compute the optimal path of the catheter to the targeted area, ensuring the best possible outcome for the patient**

Problem Description

- Extend the classical motion planning problem by allowing the robot to deform in order to follow a path while maintaining physical constraints



An example planning solution. Note that the robot must deform in order to successfully navigate the turns in the tunnel.

Starting position

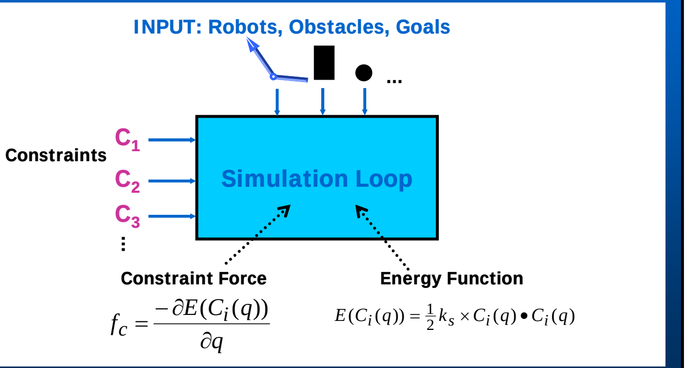
Final position

Constraint-based Planning for Deformable Robots

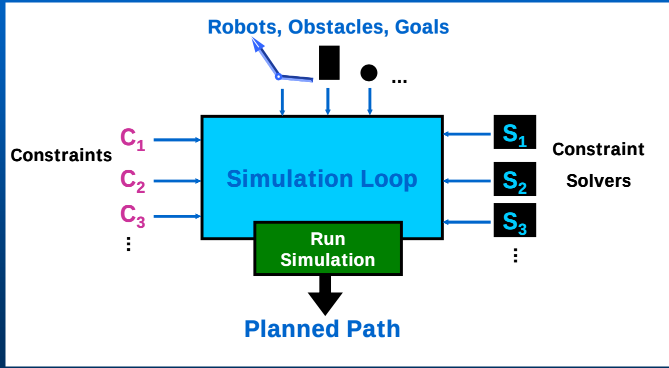
- **Formulate motion planning as a** *constrained dynamical system*
  - **Constraints are transformed to virtual forces acting on the simulated system**
  - **Solve the problem as a constrained minimization**
- **Introduce both *hard* and *soft* constraints**
  - **Goal seeking**
  - **Collision avoidance**
  - **Volume preservation**
  - **Non-penetration constraint (hard)**

[Gayle, Lin, Manocha; ICRA 2005]

General Framework



General Framework



Algorithm Overview

- Roadmap Generation
- Path Estimation
- Path Query
  - Advance simulation while satisfy constraints
    - *min E(x) subject to ∇V(x) ≤ ε*

$$E_s(X) = \sum_j \frac{k}{2}(||d_j|| - L_j)^2$$

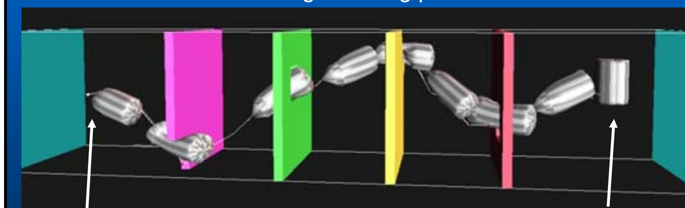
    - *Non-penetration*
    - *Volume Preservation*

Benchmark

The Serial Walls benchmark. Note that the robot is slightly larger than the holes through which it must pass

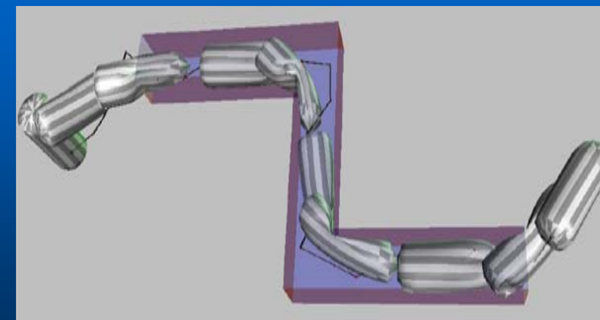
Benchmark

Steps along the planned path. We capture the robot at various stages during path traversal.



Final position   Starting position

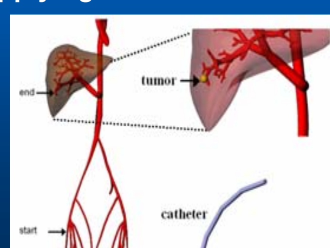
Benchmark



The Tunnel

Motion Planning Application

- **Application to plan the path of a flexible catheter, inserted at the femoral artery, to a specific liver artery supplying a tumor**
  - **Environment: 3D models of the liver and blood vessels obtained from the 4D NCAT phantom, a realistic computer model of the human body**
  - **Deformable robot: Catheter was modeled as a cylinder with a length of 100 cm and a diameter of 1.35 mm**



System Demonstration

[Gayle et al; ICRA 2005]

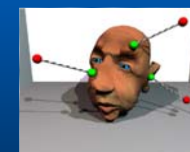
Modeling of Soft Articulated Bodies in Contact Using Dynamic Deformation Textures

IPAM-VS 2008 Ming C. Lin

Motivation

Global deformations ⟷ Detailed deformations

Barbič & James '05



Müller et al. '05

Ours

IPAM-VS 2008 Ming C. Lin

Motivation

- **Dynamic simulation of deformable** solids
- **Highly detailed** surface **geometry**
- **Large** contact area**: objects bounce, roll, slide,…**

IPAM-VS 2008 Ming C. Lin

Layered Models Motivation

- **Detailed, small-scale deformations**
- **Global (skeletal) deformations**
- **Dynamic interplay between skeletal motion and surface deformation during contact**

Ming C. Lin

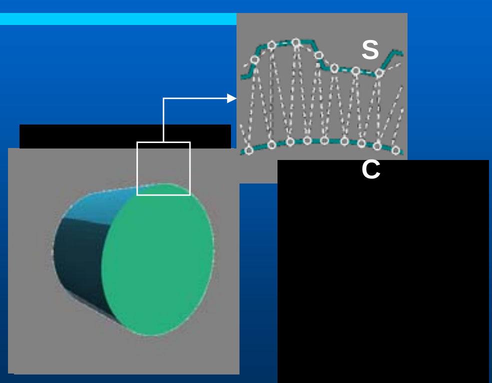
Layered Models

- Overview
  - Dynamic Deformation Textures
- Fast Coupled Layered Dynamics
- Fast Condensed Skeleton Dynamics

Layered Models

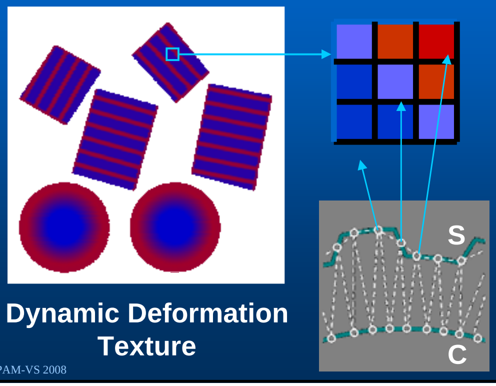
- Overview
  - Dynamic Deformation Textures
- Fast Coupled Layered Dynamics
- Fast Condensed Skeleton Dynamics

Layered Models



Layered Models

Dynamic Deformation Textures



Dynamic Deformation Texture

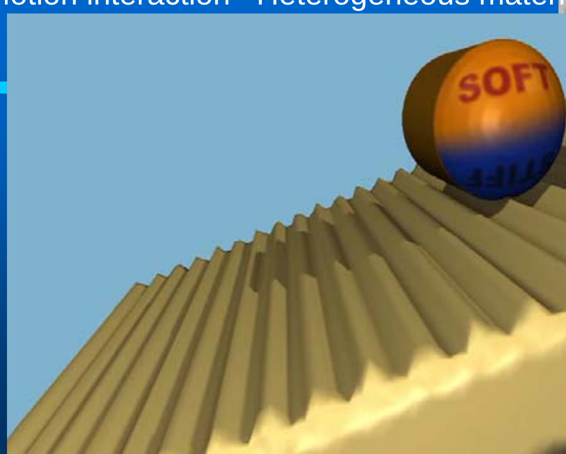
Layered Models

- Overview
  - Dynamic Deformation Textures
- **Fast Coupled Layered Dynamics**
- Fast Condensed Skeleton Dynamics

Fast Coupled Layered Dynamics

- Generalized Lagrangian framework
- Linear elasticity in moving frame of reference [Terzopoulos&Witkin'88]
- FEM discretization
- Implicit integration
- Physically based approximations

Motion interaction Heterogeneous material

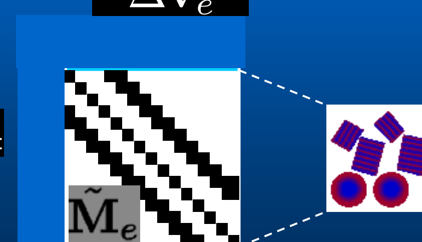


Cylinder: 161K tetrahedra

Dynamic Formulation

$$\tilde{M}\begin{bmatrix}\Delta \mathbf{v}_c \\ \Delta \mathbf{v}_e\end{bmatrix}$$

$$\tilde{M} =$$



- Exploit matrix structure + parallelize

Dynamic Formulation
Decoupled Solution

1. Project surface forces to update core velocity
2. Distribute core velocity to update surface velocities

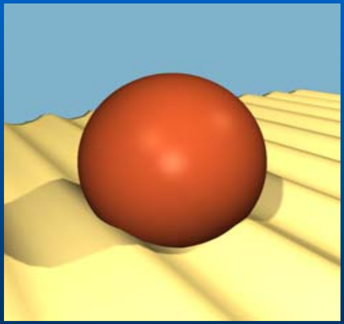
Fast & Responsive Contact

- Hierarchical Collision Queries
  - Texture-based Collision Detection
- Responsive Contact Handling
- Fast Coupled Contact Response
  - Skeleton Response Anticipation

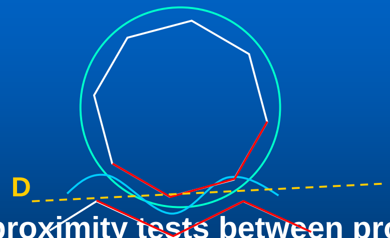
Fast & Responsive Contact

- Hierarchical Collision Queries
  - Texture-based Collision Detection
- Responsive Contact Handling
- Fast Coupled Contact Response
  - Skeleton Response Anticipation

Texture-based Collision Detection

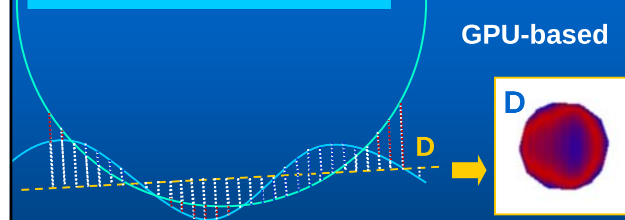


Texture-based Collision Detection
Stage 1: Low-resolution



- Fast proximity tests between proxies
- Contact plane D per contact region

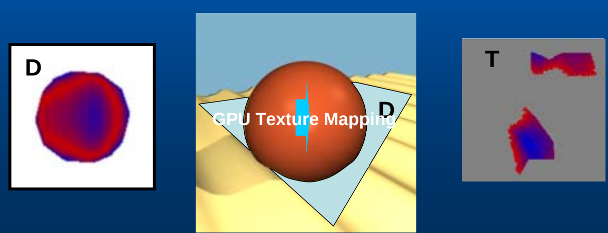
Texture-based Collision Detection
Stage 2: Detailed Surface Interference



- Surface projection onto contact plane D
- Identify contacts in D from separation distances

Texture-based Collision Detection
Collision Information Transfer

- Contact response in dynamic deformation texture
- Transfer collision information



Hierarchical Pruning

- Exploit skeletal nature of deformation
  - Locality

Fast & Responsive Contact

- Hierarchical Collision Queries
  - Texture-based Collision Detection
- Responsive Contact Handling
- Fast Coupled Contact Response
  - Skeleton Response Anticipation

Contact Response

$$\mathbf{J}\tilde{\mathbf{M}}^{-1}\mathbf{J}^T\lambda = -\mathbf{J}\mathbf{v}^-$$

Dense!

- Implicit matrices, hard to parallelize
- Decoupled solution

Responsiveness!

Fast & Responsive Contact

- Hierarchical Collision Queries
  - Texture-based Collision Detection
- Responsive Contact Handling
- Fast Coupled Contact Response
  - Skeleton Response Anticipation

Articulated Contact Response

Snake: 3,102 surface points 16 bones

Articulated Contact Response

- Joint constraint complications

$$\begin{pmatrix} \tilde{\mathbf{M}}_b & \tilde{\mathbf{M}}_{bs} & -\mathbf{J}_j^T & -\mathbf{J}_b^T \\ \tilde{\mathbf{M}}_{bs}^T & \tilde{\mathbf{M}}_s & 0 & -\mathbf{J}_s^T \\ -\mathbf{J}_j & 0 & 0 & 0 \\ -\mathbf{J}_b & -\mathbf{J}_s & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \mathbf{v}_b \\ \delta \mathbf{v}_s \\ \mu \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{b}_\mu^- \\ \mathbf{b}_\lambda \end{pmatrix},$$

Complexity: *O(mnk)*

Articulated Contact Response

- Contact-consistent Articulated Dynamics

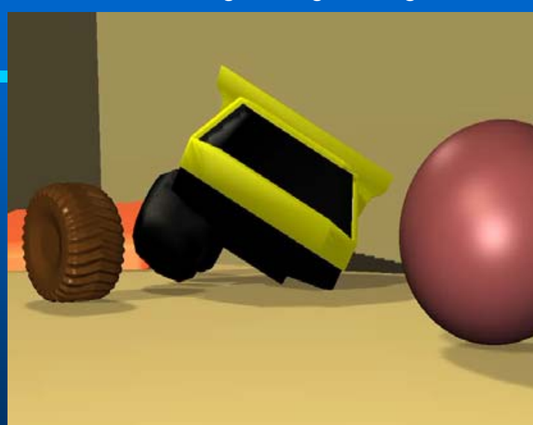
$$\begin{pmatrix} \mathbf{M}_b^* & -\mathbf{J}_j^T \\ -\mathbf{J}_j & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \mathbf{v}_b \\ \mu \end{pmatrix} = \begin{pmatrix} \mathbf{b}_b^* \\ \mathbf{b}_\mu^- \end{pmatrix},$$

$$\text{with } \mathbf{M}_b^* = \hat{\mathbf{M}}_{\text{cond}} + \mathbf{J}_{\text{cond}}^T \mathbf{M}_\lambda^{-1} \mathbf{J}_{\text{cond}}$$

$$\mathbf{b}_b^* = -\mathbf{J}_{\text{cond}}^T \mathbf{M}_\lambda^{-1} \mathbf{b}_\lambda$$

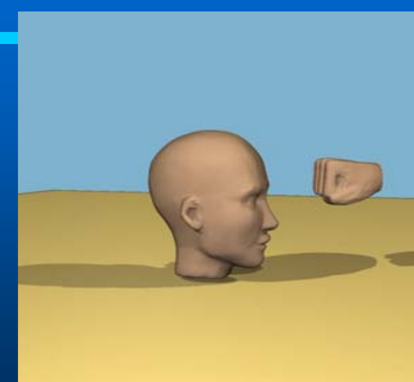
Complexity: *O(m + n + k)*

Rubbing, rolling, sliding



Tire: 162K tetrahedra

Highly detailed, dynamic deformations



Head: 240K tetrahedra 40K vertices

Exploiting Parallelism

- All surface-related computations on GPU
- Communication between core and surface computations minimized
- Dynamic deformation textures for immediate rendering

Fast Contact Modeling for Soft Characters

Deer: 2,755 surface points, 34 bones @ 10 FPS

Future Research Directions

- Multi-scale Dynamics Simulator for Virtual Prototyping and Design Automation
- Modeling and Simulation of Nano-structures and Nano-systems
- Functional Endoscopic Sinus Surgical Training System with Haptic Interfaces
- Planning and Control of Multiple Intelligent Agents using GPU Accelerated Computations
- Enabling Real-Time Interaction between Avatars and Complex Virtual Environments

Collaborators

- Dinesh Manocha
- Russell Gayle
- Nico Galoppo
- Naga Govindaraju
- Markus Gross
- Nitin Jain
- Kenneth Hoff
- Young Kim
- David Knotts
- Miguel Otaduy
- Stephane Redon

Research Sponsors

- Army Modeling and Simulation Office
- Army Research Office
- Defense Advanced Research Projects Agency
- Department of Energy ASCI Program
- Honda
- IBM
- Intel Corporation
- National Science Foundation
- Office of Naval Research
- RDECOM
- UNC Chapel Hill

Fellowship Sponsors

- ATI/AMD
- Government of the Basque Country
- Intel Corporation
- Link Foundation
- National Science Foundation
- NVidia Corporation
- University of North Carolina