Spring: A Case Study of an Open-Source Surgical Simulation Platform

Kevin Montgomery PhD

STANFORD UNIVERSITY MEDICAL CENTER

NATIONAL BIOCOMPUTATION CENTER

Surgical Simulation Benefits

- Broader training: Easily provide different scenarios
  - Anatomical variations (gender, age)
  - Pathologies (diseases, trauma)
  - Operating environments (ER, battlefield, space)
- Objective quantification of performance:
  - Simulate results
  - Certification
- Accelerated acquisition of baseline skills
- No risk to real patients

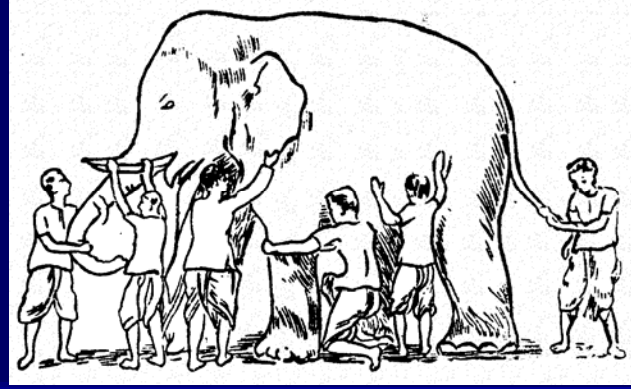
Difficulties: (It is hard to do)

- Cross-disciplinary: clinical/ engineering
- Spans many technical subdisciplines: graphics, algorithms, numerical methods, collision detection, networking, user interface, mechanical engineering, etc.
- Replicating this breadth is difficult:
  - Clinical departments alone can not develop native technical expertise to produce own simulators and must buy/use whatever is available
  - Engineering departments alone do not know what/how to build clinically useful systems and often can only focus in a few areas
- Represents major barrier to entry, production, and deployment of simulators and to the realization of those benefits

Difficulties: (the controversial part)

- Limited commercial incentive:
  - Many simulation companies are gone
  - Those that exist doing only moderately well
- Academic groups disincentivised toward building and integrating useful, real simulators
  - “We are here to train [CS] students, not do engineering”
  - Publish/push ahead in own area of research, not in others
- Therefore:
  - Better way of doing things, can save lives, no one will pay
  - We will not get rich on surgical simulation
  - Field will continue largely on government funding
  - Ex: Try to commercialize the Internet in 1970

Solution



- What if the entire community had a common framework of shared code?
  - Less time to realization of a working simulator
  - Shared individual expertise/contribution
  - Barrier to entry, deployment, and proliferation lessened
  - Accelerate the production and adoption of simulators
- Risks:
  - Trying to be *everything* to *everyone* can end up providing *nothing* to *anyone*
  - Technical challenges of building such a framework are great in themselves
  - Specific funding for such a task is difficult to obtain

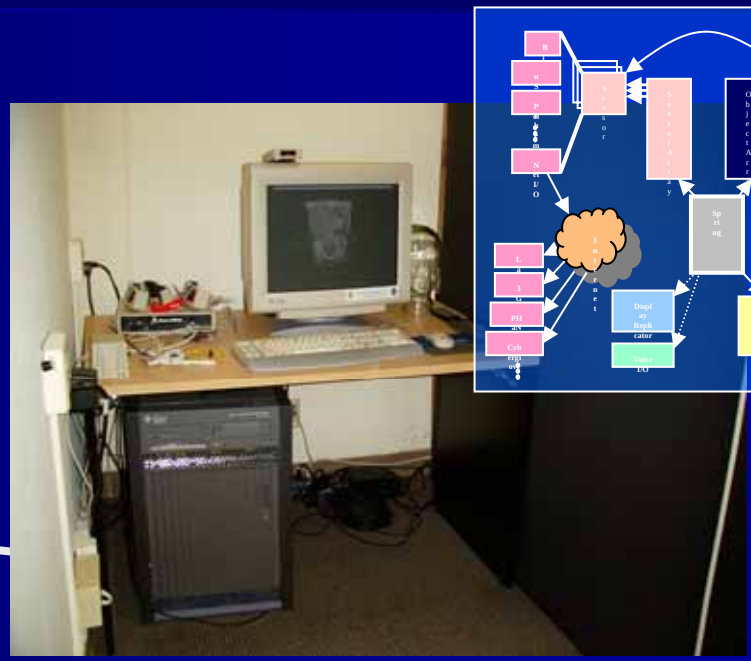
Overview

- Describe mass-spring physical simulation system built over past 6 years: ***Spring***
- Features:
  - Platform: (Sun, SGI, PC, Linux), C++/OpenGL, Parallelized
  - Models: Relatively easy introduction of patient-specific anatomy
  - Simulation: Soft tissue modeling, rigid body dynamics
  - Interfaces: Many devices, multi-user, multi-instrument
  - Haptics: networked, latency dependent or independent
  - Instruments: many surgical/nonsurgical produced
  - Collision Detection/Response: BSP-tree with enhancements
  - Display: stereo CRT, HMDs, projection, anything
  - Misc: Voice I/O, video input, stereo, replicated display (image, geom)
- Applications: Produced during development
- Emphasis on real-time (haptic rate) performance and generality

System Overview

Haptic Device/User

TCP/IP

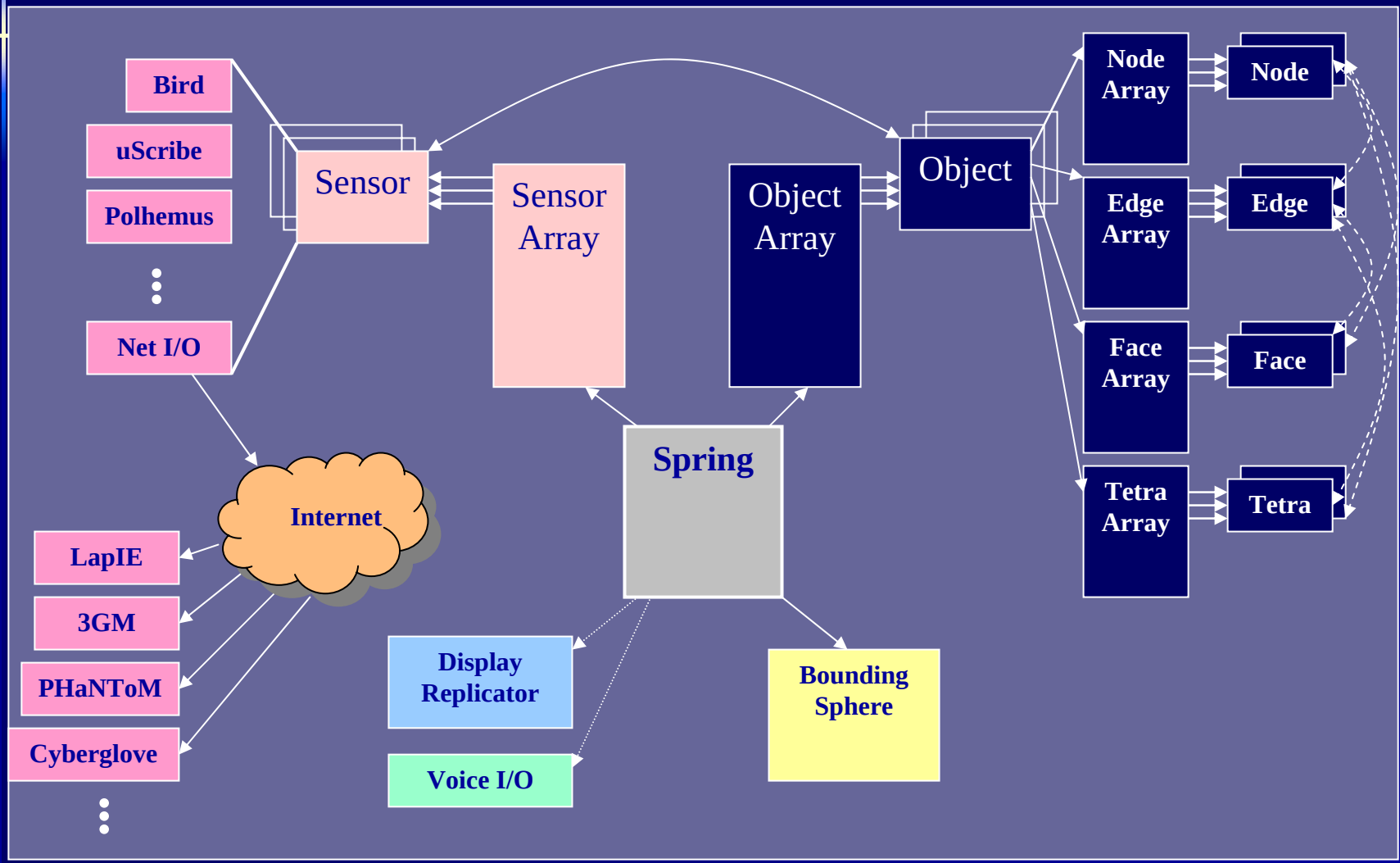


Simulation Engine

Proprietary

Haptic Device Controller

Architecture



Simulation

- Soft-Tissue Modeling:
  - Linear, piece-wise linear, nonlinear 1D springs/dampers
- Other Modeling Dynamics:
  - Suture dynamics, limited rigid body kinematics
- Numerical Methods:
  - Euler, Runge-Kutta, Quasistatic, Fast Summation
- Deformation Region Tracking:
  - Orders nodes from point of interaction, only processes in tracked region of deformation
- Parallelization:
  - Graphics/Simulation threads and/or per-object simulation threads

Interfaces

- Many devices supported:
  - Non-haptic:
    - Ascension
    - Polhemus
    - Microscribe
    - Cyberglove
    - Mouse
  - Haptic:
    - Immersion: LapIE, 3GM, Bimanual
    - Sensable: PHaNToM





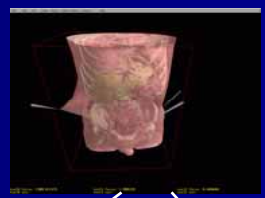


Networked Haptics/Sensors

- Devices can be a local connection (ISA, PCI, Serial) or over network to microcontroller (*hapticserver*)
  - Latency dependent: device send position/orientation with activation values, simulation processes, returns force
  - Latency independent: local microcontroller caches local region, performs initial contact processing and continuous contact interpolation
  - Enables multiperson, multi-instrument simulation
  - Also supports replication of display over network (streaming video)

NASA Ames Research Ctr

Mt View, CA



Stanford

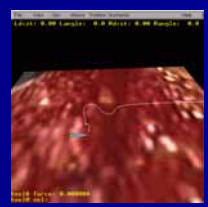
Palo Alto, CA

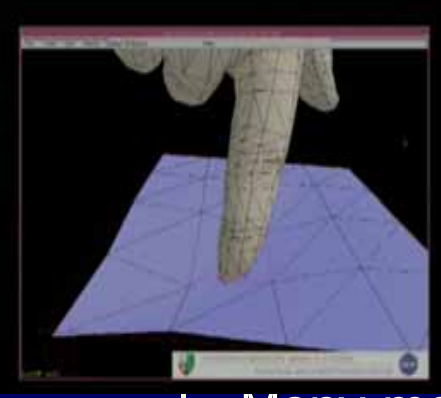
Instruments / Interactions

- Many types of interactions/instruments supported:
  - Probing: pick, dilator, hand
  - Grasping: grabby, forceps, endoscopic grasper
  - Piercing: needle, syringe
  - Cutting: scalpel, scissors, endoscopic scissors
  - Ablator/Cautery: roller ablator, loop cautery
  - Multitools:
    - Endoscopic tools with exchangable working channel for tools
    - Loop cautery, grasper, etc

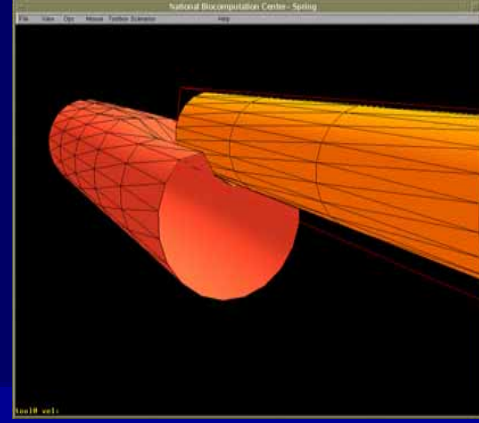






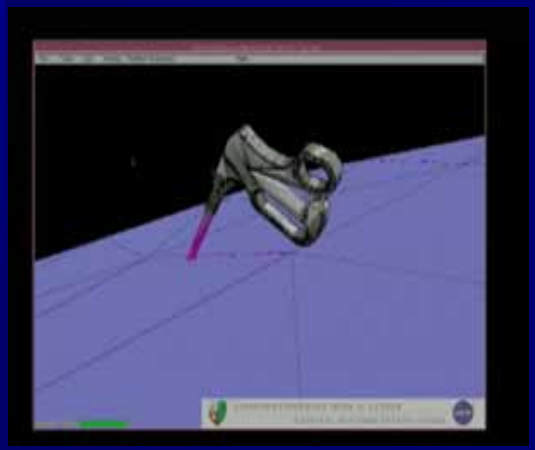


Collision Detection



Many methods implemented:

- Brute-force node-node based
- Axis-Aligned Bounding Boxes (AABBs/SBBs)
- Oriented Bounding Boxes (OBBs)
- Bounding Spheres (Quinlan) with Sorkin's enhancement for deformation recomputation and other performance enhancements



Coupling of simulation and collision detection is a big win

Haptic resolution: triangle-triangle based with averaging to calculate resulting haptic

Display Technologies Supported

	Noncollaborative	Collaborative
Nonimmersive		
Immersive		

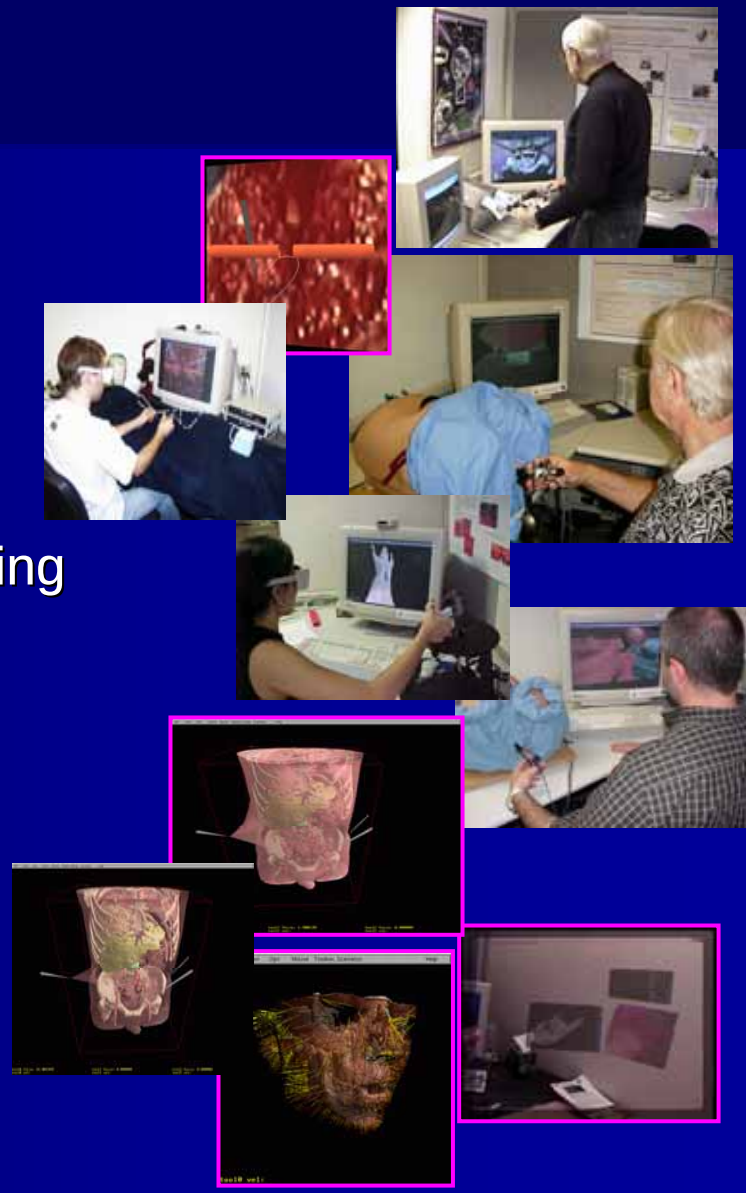
Also supports:

•Display Replication

•Geometry Replication/ Distributed Rendering

Applications

- In development:
  - Microsurgery: suture dynamics
  - Hysteroscopy/Gyn Laparoscopy
  - Colonoscopy/polyp extraction
  - Astronaut training/rat dissection
  - Stent placement simulator/planning
  - Cleft-lip surgical simulator
- Surgical planning
- Integrated into intraoperative assistance system



Summary

- Have built a generalized, useful/used framework for surgical simulation
- Supports many devices, multiple users and instruments, with generalized collision detection/ resolution
- Runs in real-time at haptic rates
- Provides for networked haptics with moderated latency for wide-area use

The Big Questions

- “This sounds great! Why isn’t it everywhere?”
  - Agencies must not want it- they will pay to build a specific simulator, but not take an existing platform and build a development community- hasn’t happened yet so not a priority
  - So therefore, we steal time from existing projects to try to do this on our own time- never enough time
  - Therefore, we’re left with okay documentation and no support- so we give it away to our friends that wont mind our blemishes
- “So where has Spring been?” It’s in 9 labs now.
- “Can I get a copy?” Watch *this*.

This is open source

The Deal:

- Non-commercial, academic use only
  - If you're a company, it gets more complicated
- Acknowledgement
  - Hey- be cool. Acknowledge us if you publish/present it.
- Mods
  - Please- do things generally so everyone can use it
  - If you make mods, send `em so everyone can benefit
- Support
  - Worth every penny you paid for it- a labor of love

"If you don't like the news, go out and make some of your own"

Final Thoughts

- Do I want *Spring* to be the defacto standard?
  - Ehhh, just want to help field
- It's worth every penny you paid for it
- Standards to not imply Interoperability
  - Which is most important?
  - Or do we really want to just accelerate development?
- Geez guys, let's just work together on all this stuff

Acknowledgements

- Developers of *Spring*
  - Cynthia Bruyns: Interactions, graphics, ...
  - Joel Brown: Soft-tissue modeling, ...
  - Frederic Mazzella: Haptics, ...
  - Guillaume Thonier: IOP application, video/voice I/O
  - Arnaud Tellier: Graphics, video/display
  - Suresh Sainath, Aneesh Sharma, Manju Boraiah: Applications
- Past Developers:
  - Stephen Sorkin, Benjamin Lerman, Jeremie Roux, Tyler Kohn
  - Anil Menon, Bobby Barnes, Bharath Beedu
- Collaborators/Contributors:
  - Simon Wildermuth, SUMMIT, NASA Ames Research Center, Michael Stephanides, JC Latombe

Contact Information

http://biocomp.stanford.edu

http://biocomp.stanford.edu/spring