http://www.radekaphotography.com/rainbow.htm

Last Lecture

Lecture 12
CS 212 – Fall 2007

Announcements

- Please participate in the online Course Evaluation
  - http://www.engineering.cornell.edu/CourseEval/
  - CourseEval is open for *one week only*
    - Monday, Nov 26, through Sunday, Dec 2
  - Evaluations are anonymous, but the CS212 staff receives a list of those who completed an evaluation
  - Note: This is worth 1% of your 212 grade
- Part 4 is due on Friday, Nov 30
  - But we won't count it as late as long as it's submitted by Tuesday, Dec 4

Quick Overview

- Introduction
  - Computer architecture
  - Machine language & assembly language
  - Intro to SaM
- Compilers
  - Lexical analysis & parsing
  - Abstract Syntax Trees
  - Recursive descent parsing & code generation
- Software Engineering
  - Use of *abstraction*
  - Specification & validation
  - Testing & debugging
  - Models for software development
  - Top-down vs. bottom-up design

More Overview

- Implementing recursive functions
  - Stack frames
- Implementing arrays
- Implementing objects
  - Use of the *Heap*
  - Dispatch vectors
- Software Engineering Tools
  - Unix
  - Programming languages
  - Scripting languages
  - Regular expressions
  - Version control
  - UML
  - Profiling

CS Undergrad Areas at Cornell

- "Core" courses
  - Algorithms
  - Data structures
  - Logic
  - Programming languages
  - Scientific computing
  - Systems
  - Theory
- Elective courses
  - Artificial intelligence
  - Computer graphics
  - Computer vision
  - Databases
  - Multimedia
  - Networks

Undergrad Core

- Programming:
  - 100, 211, 212, 312
- Architecture and Systems:
  - 314 or 316, 414
- Theory and Algorithms:
  - 280, 381, 482
- Scientific Computing:
  - 321 or 322 or 421 or 428

CS Research Areas at Cornell

- Architecture
  - processor architecture, networking, asynchronous VLSI, distributed computing
  - CS416, CS419, CS514, CS516, CS519
- Artificial Intelligence
  - machine learning, natural language processing, data mining, knowledge representation, planning, reasoning under uncertainty, search, vision
  - CS472/473, CS572, CS324, CS474, CS475, CS478
- Computational Biology
  - sequence analysis, structure analysis, protein classification, gene networks, molecular dynamics
  - CS426, CS428

More CS Research Areas at Cornell

- Databases and Digital Libraries
  - database systems, digital libraries, data mining
  - CS230, CS330, CS430, CS431, CS432, CS433, CS485, CS578
- Languages and Compilation
  - programming language design and implementation, optimizing compilers, type theory, formal verification, language-based security
  - CS400, CS411, CS412/413, CS501, CS514
- Graphics
  - interactive rendering, global illumination, modeling, measurement, image-based modeling, perception
  - CS465, CS467/468, CS565, CS567

Even More CS Research Areas at Cornell

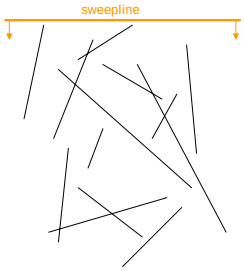
- Operating Systems, Networks, and Distributed Computing
  - operating systems, distributed computing, networking, wireless systems, security and protection
  - CS416, CS419, CS514, CS516, CS519
- Scientific Computing
  - numerical analysis, computational geometry, physically based animation
  - CS321, CS322, CS 421
- Security
  - secure network services, language-based security, mobile code, privacy, logic, verifiable systems
  - CS513
- Theory of Computing
  - algorithms, complexity, logic
  - CIS400, CS485, CS487

An Aside: If you're interested in undergraduate research

- Consider taking CS490 (Independent Research)
  - There is a link to it on the CS Dept *Courses* page (http://www.cs.cornell.edu/Courses/index.htm)
  - Basically, you need to
    - choose an area-of-interest,
    - find a faculty member working in that area, and
    - come to an agreement about what you'll do for the semester
- Take a look at the department's Research page (http://www.cs.cornell.edu/Research/index.htm)
  - Each topic links to faculty who work on that topic
- Also, take a look at websites for individual faculty members

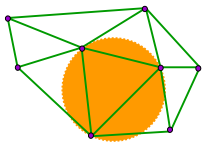
What I Do: Computational Geometry

- Using a computer to solve geometric problems
  - Get to use lots of data structure ideas
  - Example
    - Given n line segments in the plane, report all intersections
    - Uses both a PQ and a Balanced Tree
- Areas I work in
  - Motion Planning
  - Meshing
  - Shape Matching
    - computer vision
    - protein matching
  - Various theoretical questions



The Delaunay Triangulation

- Has the "Empty Circle Property" (each Delaunay triangle's circumcircle is empty)
- Is commonly used for *meshing*



initial sites

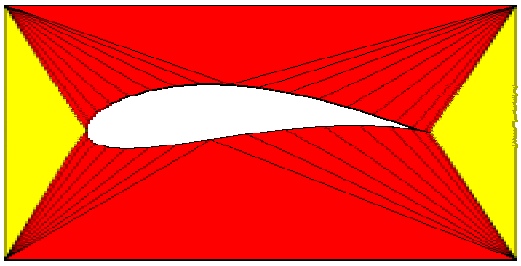
a Delaunay triangulation

an empty circle

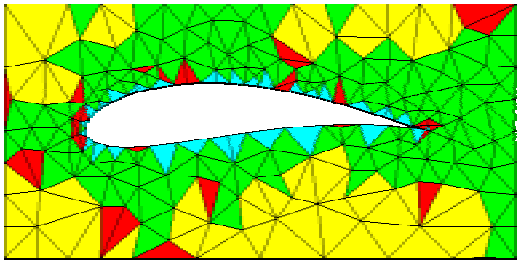
Meshing Requirements

- Control of element density
  - Small elements (in "interesting" regions) for accuracy
  - Large elements (elsewhere) for efficiency
- Allow internal boundaries
  - Needed to represent, e.g., a crack
- Ideally: guarantee of element quality
  - Nice, but unnecessary for a single mesh
  - For many applications, we need multiple meshes as geometry changes over time

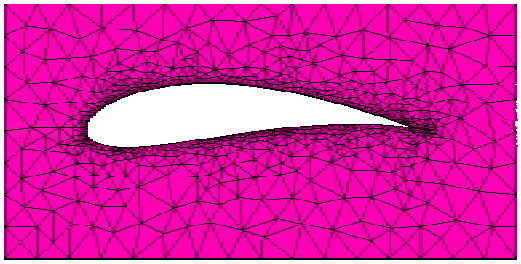
Initial Crude Mesh



During Improvement

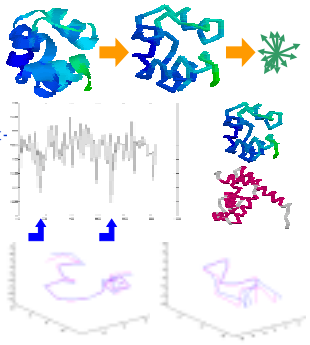


Final Mesh



Protein Shape and URMS

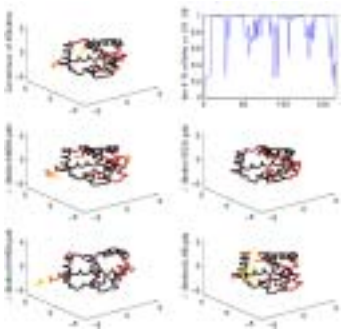
- Protein function is largely based on the protein's geometric shape
- How do we analyze protein shape?
- Our technique: URMS (Unit-vector Root Mean Square distance)
- Advantages
  - Insensitive to outliers
  - Efficient to compute
  - Equal weight for all portions



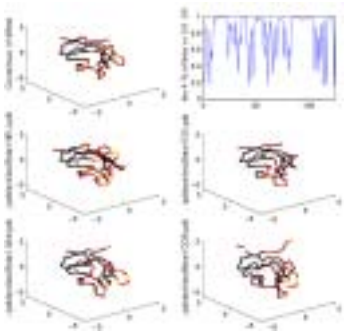
Protein Families & Consensus Shape

- Evolution theory ⇒ a protein ancient ancestor evolved into a family of proteins
- Membership in a protein family is expressed by sequence similarity, but is more strongly expressed by structure similarity
  - 25-30% sequence resemblance (almost always) ensures shape resemblance
- Goal: Create a *Consensus Shape Algorithm* that produces
  - a multiple alignment of structures, and
  - a single (core) structure that summarizes the structural information for a protein family

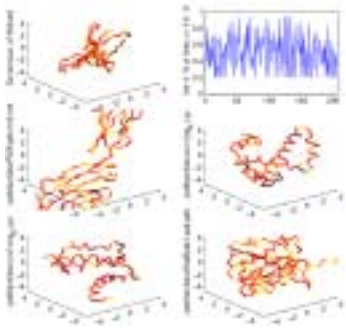
An Alpha Protein Family (Globins)



A Beta Protein Family



Unrelated Proteins



I hope Part 4 is going well

Thanks for an enjoyable semester

Good luck on any final exams!