CS709a: Algorithms and Complexity

Focus: Spatial Data Structures and Algorithms

Instructor: Dan Coming
dan.coming@dri.edu

Thursdays 5:30-8:15pm
Office hours after class
(or by appointment)

Today

- Syllabus
- Introduction
- Course Projects
- Class meeting time?

Syllabus

- See handout

Introduction

- Data
- Applications
- Organizing data

Data

- Multi-dimensional points (x_0, x_1, ... x_d)
  - In a space with dimension $d > 1$
  - Different data types: gender, age, height, weight
- Spatial data spans a continuous physical space
  - Location is interesting – relations among data
  - Generally *d* not too high
- Points vs objects (lines, triangles, cars)

Data Complications

- Curse of dimensionality
- Type of data (discrete, continuous, finite?)
- Type of operations to perform
- Organize the data or the space?
- Support adding/removing points dynamically?
- Do points move?
- Does the size of data require out-of-core storage?

Applications

- What kind of data?
- What is your space?
- What do you want to do?
- Examples: graphics, vision, robotics
- Your research area?

Organizing Data

- Many ways to organize data and/or space
  - How to choose?
- Constraints
  - Memory limitations
  - Build time vs access time
- More depth on this next class

Course Projects

- Three projects, and a final project
- Everyone has the same problem but a different data structure to implement
- Exchange projects between phases
  - Each time a new, common problem
  - Extend what you're given to solve the problem
- Result: shared library
- Final project – Apply shared library to some other research application

Project 1

- Data
  - Points
  - Template for N-dimensions, must work for 2D, 3D, extra credit if it works for ND
- Operations
  - Load
  - Range query (box and radius)
- Due Feb. 25, 11:59pm, present Feb. 26 in class

Spatial Data Structures for Project 1

- Range Tree
- Regular / Irregular Grid
- Quadtree / Octree
- kd-tree / Binary Space Partition (BSP)-Tree
- Bounding Volume Hierarchy

Later Projects

- Keep in mind
  - Exchanging projects
  - In your design, plan for:
    - Nearest neighbor / *k*-nearest neighbors
    - Non-point data: e.g., triangles
    - Ray cast

Regression Testing

- Automated scripts to test program
- Compare results against known gold standard
- Helps to find out when bugs are introduced by code modifications
- Good in practice (and required in class projects)
- Will help when you exchange projects

Class Meeting Time?

- Currently: Thursdays 5:30-8:15pm
- Better times?
- Changed to Thursdays 4:00-6:45pm

Next Time

- Basic spatial data structures for point data
- Reading: Chapter 1, Sections 1-5 (Pages 1-89)
- Find a partner for Project 1
  - Each group, email me group members/emails and which spatial data structures you prefer for Project 1, in order of preference