
Hong Qin
Rm. 151, NEW Computer Science Building (My office)
Department of Computer Science
Stony Brook University (SUNY Stony Brook)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu; or qin@cs.sunysb.edu
http://www.cs.stonybrook.edu/~qin
Course Website

• http://www.cs.stonybrook.edu/~cse328
Lecture Information

- **WHEN:** Mon & Wed 2:30pm - 3:50pm
- **WHERE:** OLD CS Rm.2129
- **OFFICE HOURS:** Mon & Wed 12:50pm – 2:20pm, or by appointment!
- **Teaching Assistant(s):** Computer Science PhD candidate, Yicheng LIN, yiclin@cs.stonybrook.edu
- **TA office hours:** Tuesdays 1-2:30pm, Thursdays 1-2:30pm, NCS, Rm.132, or by appointment!!!
TA Issue

• TBA later when I have a confirmed TA
My Contact Information

Hong Qin
Rm.151, NEW Computer Science Building (My office)
Department of Computer Science
Stony Brook University (SUNY Stony Brook)
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu; or qin@cs.sunysb.edu
http://www.cs.stonybrook.edu/~qin
OpenGL Tutorials

• Tutorials for Modern OpenGL (3.3+)


OpenGL Help for CSE328

- My TAs in previous years have collected many examples:
  - http://www3.cs.stonybrook.edu/~dozhang/cse528/
- TA Website (for openGL tutorials):
Course Facts

• **Students are expected to**
  - Take midterm-1 (10%) on Wednesday Feb 27 (tentative);
  - Take midterm-2 (10%) on Wednesday Apr 17 (tentative);
  - Take final exam (take-home, 10%) on May 9 (tentative);
  - Finish 2 programming assignments (10% each, 20% total);
  - Class attendance (5%)
  - Complete one course project, present your project in the class, and submit the final report for your course project (45%);

• **What projects are appropriate?**
  - Talk to the instructor and suggest possible topics of interest
  - Projects also available from the instructor
Programming Assignments

- Programming Assignment One (10%): due at 2:20pm, Monday, March 4, 2019
- Programming Assignment Two (10%): due at 2:20pm, Monday, April 1, 2019
Course Project (45%)

- One-page project proposal (March 6 Wed): 5%
- Mid-term demo with preliminary software results and ppt presentation (April 8 Mon): 5%
- Final project technical report + ppt file (May 10 Friday 9am): 2%
- A working system + software codes (May 10 Friday 9am): 30%
- Final software demo & oral ppt presentation (May 10 Friday 9am-7pm): 3%
Possible Bonus

• Up to 10%

• Possible ways to gain up to 10% (extra)
  – Projects on much more challenging computer graphics topics
  – Extra functionalities being implemented (beyond what the paper was originally suggesting, e.g., extra works suggested in the future work subsection)
  – You should discuss this option with the instructor before the submission of your project proposal
Project Proposal

- **HARD Deadline (no extension):** March 6 Wednesday at 2:30pm
- Hardcopy submission in class
- One-page proposal
- Project is based paper entitled “XYZ”
- Detailed plan with weekly activities
Course Project

• Electronic submission to My TA (TBA), including software, project report, ppt file for your presentation, and any supporting documents such as readme files

• Hardcopies submission to my office, including project report, ppt file for your presentation, and any supporting documents such as readme files
Course Prerequisites

- **Mathematical skills:** fundamental knowledge on calculus, linear algebra, analytic geometry, etc. (Basic mathematical training at the undergraduate level, Appendix A Mathematics for Computer Graphics is a good starting point to refresh our memory)

- **Computer science background:** programming skills at the basic undergraduate level (C/C++, OpenGL (graphics library))
Course Prerequisites

- Essentially, you need to have an undergraduate education in computer science or engineering (at the senior level) with basic knowledge on computer programming.
- You need to speak to the instructor if you are not sure about your background knowledge and course prerequisites.
Course Prerequisites

• Please note that, just like other CSE3xx level courses in our department, this is an senior-level under-graduate course!!

• I expect that you are having the ability to learn a programming library on your own (OpenGL)!!!
Course Prerequisites

- I do NOT expect that we are going to teach you C/C++ during my rather precious lecturing hours!
- This is NOT a game-programming course!
- We are NOT going to teach a particular package
Computer Graphics Course

Not about!

Paint and Imaging packages (Adobe Photoshop)

Cad packages (AutoCAD)

Rendering packages (Lightscape)

Modeling packages (3D Studio MAX)

Animation packages (Digimation)

Not about!

Game programming and/or Graphic design courses!!!
My Expectations

• Time and efforts dedicated to this course, you have to spend time on reading the textbook, reviewing notes, attending my lectures, working on your programming assignments/projects, and taking exams!

• Problem-solving skills: what are the right and most effective approaches, taking advantages of online resources, etc.

• Interaction with the instructor, TA, and your fellow students, etc.
My Expectations

• Your works should be your OWN!
• NEVER share code with your fellow student or debug code together
• Reference examples from the web is an effective way to learn and you are encouraged to do so
• When using open sources, you should explicitly point them out
• NOT a course about graphic/game design, NOT using graphics packages like PhotoShop / Maya
Two Critical Issues

• (1) Course pre-requisites: Do I have sufficient amount of background knowledge???

• (2) Satisfying all course requirements: can I succeed in this course???

• Assuming we have all the required background, but why we are still failing in a course like this one: key elements include studying what are required, spending enough amount of time in programming assignments, final project, starting early, time management, seeking help from instructor/TA, etc.
Textbook Coverage (for Midterm-1)

- Chapters 1, 2, and 3 (Introduction, Hardware, Software)
- Chapters 4, 5, and 6 (2D Graphics Primitives and Their Attributes)
- Chapters 7, and 8 (2D Graphics Pipeline involving Multiple Coordinate Systems and Transformations)
- Chapters 9, and 10 (3D Concepts, 3D Graphics Pipeline involving Multiple Coordinate Systems and Transformations)
- Plane Equation, Normal, and their Math are Chapter 4.7 (pages 67-70)!!!
- Appendix A: Mathematics for CG
Textbook Coverage

- Chapters 1, 2, and 3 (Introduction, Hardware, Software)
- Chapters 4, 5, and 6 (2D Graphics Primitives and Their Attributes)
- Chapters 7, and 8 (2D Graphics Pipeline involving Multiple Coordinate Systems and Transformations)
- Chapters 9, and 10 (3D Concepts, 3D Graphics Pipeline involving Multiple Coordinate Systems and Transformations)
- Plane Equation, Normal, and their Math are Chapter 4.7 (pages 67-70)!!!
- Appendix A: Mathematics for CG
Lecture Information

- **WHEN:** Mon & Wed 2:30pm - 3:50pm
- **WHERE:** OLD CS Rm.2129
- **OFFICE HOURS:** Mon & Wed 12:50pm – 2:20pm, or by appointment!
- **Teaching Assistant(s):** Computer Science PhD candidate, Yicheng LIN, yiclin@cs.stonybrook.edu
- **TA office hours:** Tuesdays 1-2:30pm, Thursdays 1-2:30pm, NCS, Rm.132, or by appointment!!!
Textbook Coverage

- Chapter 6.1 Line-drawing Algorithms
- Chapter 6.4 Circle-generating Algorithms
- Chapter 6.5 Ellipse-generating Algorithms
- Chapter 6.6 Other Curves
- Chapter 6.10 General Polygon-fill Algorithm
- Chapter 6.11 Convex Polygons
- Chapter 6.12 Region Fill with Curved Boundaries
- Chapter 6.13 Area Fill with Irregular Boundaries
Textbook Coverage

- Chapter 7.1 Basic 2D Geometric Transformations
- Chapter 7.2 Matrix Representations and Homogeneous Coordinates
- Chapter 7.3 Inverse Transformations
- Chapter 7.4 2D Composite Transformations
- Chapter 7.5 Other 2D Transformations
- Chapter 7.8 Transformations between (Multiple) 2D Coordinate Systems
- Chapter 7.9 OpenGL Functions
- Chapter 7.10 OpenGL Programming Examples
Textbook Coverage

- Chapter 8.1 2D Viewing Pipeline
- Chapter 8.2 Window and Viewport
- Chapter 8.3 Viewport Transformations and Normalization
- Chapter 8.4 OpenGL 2D Viewing Functions
- Chapter 8.5-8.8 Clipping
- Chapter 8.8-8.10 Curve Clipping & Text Clipping
The Course Objectives

• Provide our undergraduate students a comprehensive knowledge on fundamentals of computer graphics, including basic concepts, theory, algorithms, techniques, and applications for modeling, simulation, rendering, animation, human-computer interactions, and other key elements of graphics-driven visual computing.

• Demonstrate the significance of these mathematical and computational tools and graphics algorithms in visual computing and relevant areas.

• Emphasize a "hands-on" approach to both the better understanding of graphics concept/theory/algorithms and the effective use of graphics techniques in various applications.
The Course Load and Strategies

• Reading our required textbook (we will be covering about 70-80% contents of this book during this semester) and learn knowledge about background, theory, algorithms, techniques, system components and architecture, software and hardware elements, applications, etc.

• Practice on exercises documented at the end of each chapters (two types of exercises: problem-solving questions, and graphics programming examples)
The Course Load and Strategies

- All concepts, theories, algorithms, techniques, system matters, software and hardware elements, and applications relevant to computer graphics are well within the boundary of our textbook, so please DO read the book and practice on exercises.

- At the same time, many programming examples throughout this book (in C and with the help of OpenGL, graphics library), so practice on those programming examples as well.
Our Mechanism to Grade This Course

- Programming assignments for us to get familiar with graphics programming basics
- Two midterm exams (in classroom) and one take-home final exam to test our abilities in understanding the course material, including concepts, theories, algorithms, techniques, software and hardware elements, applications, etc.
- But, ultimately, we are doing a course project that allows us to integrate theory with programming capabilities
Improving our Problem-Solving Skills

• Understand the problem via paper reading

• (Re)-implement a published paper by developing a graphics system that demonstrate that you are: learning graphics knowledge, and gaining graphics programming skills

• A strategy leading to success: reading the book and the paper, thinking about your interest, proposing your project (based on a published paper), and delivering results
How to Get a “A”?

- Two programming assignments (10% each, 20% total), roughly speaking, one programming assignment every five weeks!!!
- Two midterm exams, tentatively on 2/27 (10%), and 4/17 (10%)
- One take-home final exam, tentatively on 5/9 (10%)
- Class attendance (5%)
- NO final exam during the final-exam week!
- Course final project (45%)
- The above percentages will collectively comprise 100% of this course’s requirements!!!
- Up to 10% bonus (e.g., much more challenging project topics, extra functionalities, should discuss this option with Hong)
Key Components

- **Computer graphics pipeline**, basic concepts, theory, algorithms, and techniques
- **Modeling**: representation choices of different models
- **Rendering**: simulating light and shadow, camera control, visibility, discretization of models
- **HCI (human-computer interface)**: specialized I/O devices, graphical user interfaces
- **Animation**: lifelike characters, natural phenomena, surrounding virtual environments
- **Advanced topics**
Main Concentrations

- Mathematical concepts, modeling and rendering theory, and computational tools
- Fundamental algorithms in representation, modeling, simulation, rendering, animation, etc.
- Geometric (and graphical and visual) modeling and simulation techniques, and geometric processing and analysis tools
- A large variety of applications in graphics and visualization as well as other visual computing areas
- Several advanced topics and they are all research-oriented, representing the most sophisticated ones
Our Course

• A subset of key concepts, theory, algorithms, techniques, and applications
• Extensive topics with a main focus on our unique course mission
• Comprehensive lectures (focusing on geometric intuition, good ideas, and application needs)
• Numerous slides, figures, images, and videos for easy understanding (after all, this is the nature of graphics and visualization)
• Active students’ involvements
Course Facts

• This is a senior-level undergraduate course for both CSE and ISE students!!!

• Can I take this course? YES, if YOU
  – are a undergraduate student (at the senior-level) with CS background, have basic mathematical skills in calculus, linear algebra, and analytic geometry, have BASIC knowledge on computer programming, or talk to the instructor

• One required textbook, several suggested references

• Lecture notes are important!!! Class attendance in critical!!!
Course Facts

• Students are expected to
  – Take two midterm exams (10% + 10%): tentatively on 2/27 and 4/17
  – Take a take-home final exam (10%): tentatively on 5/9
  – Finish 2 programming assignments (10% each, 20% total);
  – Class attendance (5%)
  – Complete one course project, present your project in the class, and submit the final report for your course project (45%);
  – Up to 10% bonus (extra)

• What projects are appropriate?
  – Talk to the instructor and suggest possible topics of interest
  – Projects also available from the instructor
Programming Assignments and Course Project

- Two programming assignments: 10% each, 20% total
- Due: 3/4 and 4/1
- Course project: 45%
- Basic programming and course project requirements
  - Interactive interface (graphics-based)
  - Intuitive and easy to understand
  - Efficient (fast, high-performance)
  - Basic functionalities
  - Examples
  - Flexible and easy to generalize
Course Project Plan and Deadlines

- Study 1-2 papers throughout the semester
- Submit your own one-page proposal on course project (Mar 6 Wednesday is the deadline)
- Implement basic functionalities and user interface before our mid-term check point (April 8 Monday is the deadline)
- Final project and report (Electronic submission, May 10 Friday 9am, this is a HARD deadline, no extension!!!)
- Class presentation & final project demonstration (all show up please, each will present 15 minutes in class, May 10 Fri 9am-7pm)
- Individual project or group project (up to 2 students) is okay!!!
- Office hours / individual meetings
- Penalty for late submission (25% each day)
If You are Serious about this Course

• Study my on-line, electronic course notes, and read the textbook
• Prepare for two midterm tests that cover fundamentals of computer graphics in the aforementioned aspects
• Successfully complete two programming assignments (Details TBA)
• Start to think about your course project by trying to read a few research papers
• Start to think about how to implement your course project
• Write a proposal on your project and start to work on it immediately
• Finish your project by the end of this semester and submit your final course project
• You are welcome to communicate with me via emails, call me, or come to meet with me during my office hours in my office!
• Feel free to make appointments with me!
Important Deadlines for CSE328

- Two programming assignments: deadlines (3/4 and 4/1)
- Three exams: tentatively scheduled on 2/27, 4/17, and 5/9)
- March 6 (Wednesday): one-page proposal for the course project
- April 8 (Monday): mid-term check point for project with demo
- May 10 9am final project due (code + report)
- May 10 9am-7pm: final course project presentation in classroom
Course Project Grading Requirements

1. Meet with the instructor for (at least) 30 minutes to decide your study plan for this course, review your research experiences in the past, and plan for the future, and this should be done during the next four weeks.

2. Upon the individual meeting with the instructor, select 1-2 research papers and start to read them immediately.

3. Write and submit a one-page technical proposal on what you are planning to do during the next 2 months, roughly 10-11 weeks between the proposal deadline and the end of the semester (programming-driven research projects, re-implementation of at least one part from one paper, etc.)

4. Finish all the course requirements for all check points.

5. Give a final presentation (up to 15 minutes) based on your final technical report for your project (5 pages).

6. Submit your software code, ppt presentation file, and project report at the end of the semester.
My Goals for this Course

• My bottom-line is that everybody in this class will learn something by the end of this semester, so that people are NOT wasting their time here

• My strategy: breadth (I will make the slides available to everybody) + depth (I will pay attention to several important topics)

• In order to realize these goals, I would like to get everybody involved, and I very much encourage INTERACTION!

• Students must finish their assignments (mid-term exams, two programming assignments, project proposal, various check points for the course project, final project demo, final project report, etc.) and course projects and they should give presentations to the entire class

• Success in our undergraduate education: a good idea (project with a research goal) + technical writing (putting together technical reports) skills + communication skills (oral presentations in our class)
How to Get a “A”?

• Finish all the course requirements, and I will issue a “A” grade
Questionnaire

1. List your background courses/knowledge/education related to graphics/visualization, programming language, mathematical requirements, and your current education level

2. What is the main goal/purpose for you to take this course (e.g., learn the knowledge, pursue a career in this area)

3. How does this course help your future professional career

4. Your expectations on the course

5. Your studying plan

6. Other important issues that you can think of about the course
What is Computer Graphics

The creation of, manipulation of, analysis of, and interaction with pictorial representations of objects and data using computers

- Dictionary of Computing

Computer Graphics is also called Image Synthesis

A picture is worth a thousand words

- Chinese Proverb
Computer Graphics Components
Computer Graphics

- (Realistic) pictorial synthesis of real and/or imaginary objects from their computer-based models (datasets)
- It typically includes modeling, rendering (graphics pipeline), and human-computer interaction
- So, we are focusing on computer graphics hardware, software, and mathematical foundations
- Computer Graphics is computation
  - A new method of visual computing
- Why is Computer Graphics useful and important?
- Course challenges: more mathematics oriented, programming requirements, application-driven, inter-disciplinary in nature, etc.
Basic Elements of Computer Graphics

- Graphics modeling: representation choices
- Graphics rendering: geometric transformation, visibility, discretization, simulation of light, etc.
- Graphics interaction: input/output devices, tools
- Animation: lifelike characters, their interactions, surrounding virtual environments
Earlier Days of Computer Graphics

- Visual display of data (graphs and charts)
Mathematical Function Plots
Computer Graphics Components
Mathematical Background

- **Computer Graphics has a strong 2D/3D geometry component**
- **Basic linear algebra is also helpful** – matrices, vectors, dot products, cross products, etc.
- **More continuous math (vs. discrete math) than in other typical computer science courses**
- **Function plots, curves, and surfaces**
Primary Topics

- Overview, applications
- Basic components, history development
- Hardware, system architecture, raster-scan graphics
- Line drawing, scan conversion
- 2D transformation and viewing
- 3D transformation and viewing
- Hierarchical modeling
- Interface
- Geometric models
- Color representations
- Hidden object removal
- Illumination models
- Advanced topics
A Very Good Textbook for General Issues in Computer Graphics

OpenGL Reference Books


Why Graphics and Visualization

• A Chinese proverb: “a picture is worth a thousand words.”

• “A picture is worth more than a thousand words.” – ancient proverb
One Picture
Many Words...

1000 words (or just 94 words), many letters though...

It looks like a swirl. There are smaller swirls at the edges. It has different shades of red at the outside, and is mostly green at the inside. The smaller swirls have purple highlights. The green has also different shades. Each small swirl is composed of even smaller ones. The swirls go clockwise. Inside the object, there are also red highlights. Those have different shades of red also. The green shades vary in a fan, while the purple ones are more uni-color. The green shades get darker towards the outside of the fan.
Computer-Aided Design
Scientific Visualization/Simulation

velocity vectors and absolute velocity

Flowfield around a Highrise Building
Digital Ocean
Geosciences/GIS
Biology (Protein on DNA)
Medical Imaging and Processing

Multiple fragments

An x-ray is a photo taken with a machine which passes electromagnetic radiation through the body, capturing an image of the internal structures.
Digital Entertainment
Graphic Arts
Graphics Examples
What is Visualization

Visualization is a method of extracting meaningful information from complex or voluminous datasets through the use of interactive graphics and imaging.
Why Graphics and Visualization

- Enable scientists (also engineers, physicians, general users) to observe their simulation and computation
- Enable them to describe, explore, and summarize their datasets (models) and gain insights
- Offer a method of SEEING the UNSEEN
- Reason about quantitative information
- Enrich the discovery process and facilitate new inventions
Why Graphics and Visualization

- Analyze and communicate information
- Revolutionize the way scientists/engineers/physicians conduct research and advance technologies
- About 50% of the brain neurons are associated with vision
- The gigabit bandwidth of human eye/visual system permits much faster perception of visual information and identify their spatial relationships than any other modes
  - Computerized human face recognition
More Examples

Images  

Points  

Volumes
More Examples
Terrain Modeling and Rendering
Medicine and Health-care
Entertainment
Virtual Environment
National Security
Virtual Tourism
Design and Manufacturing
What are Happening Now

• Network Graphics

3D Advertisement

Server → Virtual Museum → Client

Live Sports Broadcast
What are Happening Now

- Wireless Graphics
What Are Our Ultimate Goals?

- A large variety of datasets (acquired via scanning devices, super-computer simulation, mathematical descriptions, etc.)

- A pipeline of data processing that consists of data modeling (reconstruction), representation, manipulation (rigid transformation or deformation), classification (segmentation), feature extraction, simulation, analysis, visual display, conversion, storage, etc.

- Visual information processing in the intelligent way (Intelligent Information Processing)
What Are Our Ultimate Goals?

- Datasets that are huge, multi-dimensional, time-evolving, unstructured, multi-attributes (geometric info. + material distributions), scattered (both temporal and spatial)...

- We are investigating mathematical tools and computational techniques for data modeling, reconstruction, manipulation, simulation, analysis, and display
Challenges

- TOO MUCH data
- The number of data sources keeps increasing
- Sensor quality and resolution are increasing
- Existing instruments are still available
- The speed of supercomputer is faster than ever
- We must do something (besides collecting and storing the datasets)
- We must deal with the huge datasets effectively
- Visual communication, improve our visual interaction with data
Challenges

- Data-driving, scientific computing to steer calculations
- Real-time interaction with computer and data experimentation
- Drive and gain insight into the scientific discovery process
Related Fields

- **Computer graphics (image synthesis)**
  - Generate images from complex multivariate datasets
- **Image processing, signal processing**
- **Image understanding (pattern recognition)**
  - Interpret image data
- **Computational vision**
- **Human-computer interaction**
  - Mechanisms to communicate, use, perceive visual information
- **Computer-aided design**
- **Neurological/physiological studies on human brain and our visual system**
sensors, scanners, cameras

data

super-computers

sampling/scanning

polygonization

discretization

computation/simulation

geometric model (structures)

image (signal)

computer graphics computer vision

image processing

display device

film recorder

sampling/scanning
Computer Graphics Pipeline

- Data acquisition and representation
- Modeling data and their (time-varying) behaviors (e.g., physical experiments or computational simulations)
- Graphics system and software environments for data rendering
- Image-based techniques
Data Sources

- Scanned, computed, modeled data
- The first process is data-gathering
- Large variety of data sources and attributes
- Extremely large-scale datasets
- Require real-time processing
Data Acquisition and Processing

- Pixels and voxels
- Regular & irregular grids
- Numerical simulations
- Surface or volumetric data
- Scalar, vector, tensor data with multiple attributes
- Higher-dimensional and/or time-varying data
- Popular techniques
  - Contouring, iso-surfaces, triangulation, marching cubes, slicing, segmentation, volume rendering, reconstruction
- Image-based processing techniques
  - Sampling, filtering, anti-aliasing, image analysis and manipulation
Information Domain

- Sciences (e.g., statistics, physics)
- Engineering (e.g., empirical observations for quality control)
- Social events (e.g., population census)
- Economic activities (e.g., stock trading)
- Medicine (e.g., computed tomograph (CT), magnetic resonance imaging (MRI), X-rays, ultrasound, various imaging modalities)
- Geology
Information Domain

- Biology (e.g., electronic microscopes, DNA sequences, molecular models, drug design)
- Computer-based simulations (e.g., computational fluid dynamics, differential equation solver, finite element analysis)
- Satellite data (e.g., earth resource, military intelligence, weather and atmospheric data)
- Spacecraft data (e.g., planetary data)
- Radio telescope, atmospheric radar, ocean sonar, etc.
- Instrumental devices recording geophysical and seismic activities (e.g., earthquake)
Graphics and Visualization

• Data acquisition, representation, and modeling
• Imaging processing
• Visualization (displaying) methods and algorithms
• More advanced research topics
Pathway to Success

- Highly-motivated
- Hard-working
- Start as soon as possible
- Communicate with the instructor on a regular basis
- Actively interact with your fellow students
- Visit libraries and internets frequently for papers and software system
- Read as many papers as possible
- Work on your course project
Computer Graphics

• “The purpose of scientific computing is insight, not numbers,” by Richard Hamming many years ago

• These fields are all within computer science and engineering, yet computer graphics spans multidisciplines

• Computer Graphics (another definition)
  – Application of computers to the disciplines of sciences/engineering
Computer Graphics

• Computer Graphics is application-driven, so what are its applications?
Applications

- Simulation and training: flight, driving
- Scientific visualization: weather, natural phenomena, physical process, chemical reaction, nuclear process
- Science: Mathematics, physics (differential equations), biology (molecular dynamics, structural biology)
- Environments sciences
- Engineering (computational fluid dynamics)
- Computer-aided design/manufacturing (CAD/CAM): architecture, mechanical part, electrical design (VLSI)
Applications

- Art and Entertainment, animation, commercial advertising, movies, games, and video
- Education, and graphical presentation
- Medicine: 3D medical imaging and analysis
- Financial world
- Law
- WWW: graphical design and e-commerce
- Communications, interface, interaction
- Military
- Others: geographic information system, graphical user interfaces, image and geometric databases, virtual reality, etc.
Journals and Conferences

- Computer Graphics (proceedings of ACM SIGGRAPH)
- ACM Transactions on Graphics
- IEEE Transactions on Visualization and Computer Graphics
- IEEE Computer Graphics and Applications
- Computer-Aided Design
- Computer Aided Geometric Design
- Others!!!
Why Graphics and Visualization

- A Chinese proverb: “a picture is worth a thousand words.”
- “A picture is worth more than a thousand words.” – ancient proverb
Key Components

• **Modeling**: representation choices of different models
• **Rendering**: simulating light and shadow, camera control, visibility, discretization of models
• **HCI (human-computer interface)**: specialized I/O devices, graphical user interfaces
• **Animation**: lifelike characters, natural phenomena, surrounding virtual environments
Course Projects

- Sampled course projects from previous years:
- Terrain rendering (rendering of heightmaps)
- Realistic animation of liquids
- Raytracing based on KD-trees
- SPH (Smoothed Particle Hydrodynamics) Simulation of water
- Modeling and rendering of cloud
- Modeling plants with L-systems
- Subdivision surfaces in character animation
Course Projects

- Shadow mapping techniques
- Modeling and rendering of seashells
- Particle-based modeling system for cloth animation
- 3D non-photorealistic rendering
- Hidden surface removal
- Particle-based fluid simulation for interactive applications
- Interactive ray tracing
Course Projects

- Procedural modeling for terrain generation
- Distributed ray tracer
- Teddy: A sketching interface for 3D freeform design
- As-rigid-as-possible shape manipulation
- Physical wave simulation
- Real-time shape illustration using Laplacian lines
- Cellular automata for cloud simulation
- Interactive technical illustration
Course Projects

- 3D terrain design
- Image deformation based on moving least squares
- Shape sculpting system based on vector field deformation
- Surface skinning techniques
- Video textures
- Simulating Chinese painting effects
- Digital inpainting and image restoration
Flow Simulation (Navier-Stokes Equation)
Fluid Simulation
Natural Phenomena
Simulation of Bubble Flow
Geometry Synthesis of Human Hair

(a) curly + long straight  
(b) short spikes + wavy  
(c) puffy + straight  
(d) combing
Tree Simulation
Facial Expression Acquisition and Synthesis
Computer Art with Physical Interface
Non-Photorealistic Rendering
Collision Handling

Figure 5: Screenshots of (a) Object and (b) Plane Simulation.
Shape Deformation and Editing
Shape Deformation
Motion Synthesis (Animation)
Shape Matching
Urban Modeling
Architectural Geometry
Biomedical Applications
Organ Deformation
Fig. 13. The soft tissue simulator produces realistic deformations of (a) the visualization geometry, and (b) embedded volumetric muscles.
Model Segmentation
Building Reconstruction

Figure 11: Additional reconstruction results using SmartBoxes. From left to right: real photograph, LiDAR scan, 3D reconstruction, and its textured version for a visual comparison with the photograph. The examples show reconstruction of complex buildings with some irregularity. Grouping and contextual forces during drag-and-drop allow the reconstruction to deal with large-scale missing data (bottom row).
PDE-driven Texture Synthesis
Geometry Texture Synthesis

High genus scales
Generating New Models from Examples
Augmented Reality in Neurosurgery
More Suggested Topics

- Realistic animation of clouds
- Multi-scale line drawings from 3D meshes for shape abstraction
- Simulation of smoke
- Stylized rendering techniques for scalable real-time 3D animation
- Image-based tree-modeling using particle flows
- Animation of fire
• Real-time 3D fluid simulation on GPU with complex obstacles
• Real-time non-photorealistic rendering
• Volcano animation
• Real-time procedural terrain generation
• Real-time collision handling in graphics games
• Image quilting for texture synthesis and transfer
• Sketch-based user interface for procedural terrain
• Interactive 3D terrain sketching
• 3D concept sketches using cross-section curves
• Particle-based fluid simulation for interactive graphics applications
• Automatic video texture synthesis
• Modeling and reconstructing objects with single photo
• Practical model for light transport
• Automatic and interactive lighting preview system
• Automatic generation of surface crack patterns
• Sketching solid models using blob trees
• Interactive architectural modeling using procedural extrusions
• Mesh simplification
• Out-of-core ray casting and ray tracing
• Fluid surface simulator
• Double-sided 2.5D graphics
• Cloth simulation
• Interactive global illumination
• K-means algorithm for image and geometry shape classification
• Fluid simulation with bubble generation
• Automatic construction of 3D models from line drawings
• Liquid interaction with deformable models
• Fluid dynamics for games
• Multiresolution dynamic deformations
• Rendering heightmaps based on GPU
• SPH simulation for water in pool
• KD-tree based ray tracing
• An extended toon shader
• Interactive technical illustration
• Image segmentation
• Progressive refinement for rapid radiosity
• Crowd simulation for game development
• Feature-sensitive bas relief generation
• Fast façade acquisition using line arrangements
• 3D shape retrieval based on visual similarity
• Video face replacement
• Poisson vector graphics
• Real-time oil painting
• Modeling of clouds using single photograph
• A curved ray camera for handling occlusions
• Realistic animation of liquids
• Subdivision surfaces in character animation
• Teddy
• Facial animation
• Simulation of cloud dynamics on GPU
• Simulating and animating liquids
• Shadow mapping based on ray tracing
• Modeling snow fall and accumulation
• Modeling sea shells
• Particle system for cloth modeling
• Real-time hidden surface removal
• Non-photorealistic rendering
• Digital inpainting and image restoration
• Real-time ray tracing based on graphics hardware
• Photon mapping and rendering
• Procedural modeling of cities
• 3d particle system for real-time video game
• Creation of maze art
• Real-time distributed ray tracing
• Procedural terrain based on scene voxelization
• 3D lego modeling based on shape voxelization
• As-rigid-as-possible shape manipulation
• Physical wave simulation for ocean scene production
• Real-time Eulerian water simulation
• Geometric skinning for human animation
• Video textures
• Real-time shape illustration using Laplacian lines
• A procedural watercolor engine for polygons
• Simple data-driven modeling of brushes
• Discrete element textures
• Photo watercolorization
• Computer generated floral ornament
• Silhouettes and outlines of arbitrary 3D models
• Simulating and modeling lichen growth
• Modeling plants using L-systems
• Procedural generation of road network
• Texture mapping for hand-drawn cartoons
• Qsplat: a multiresolution point rendering system for large meshes
• Image deformation using moving least squares
• Treakable light and shade for cartoon animation
• 3D mesh descriptors
• 3D model reconstruction from cross-sectional curve drawings
• Perlin noise generator
• 3D shape co-segmentation and clustering
• Extracting tree skeletal structures from point clouds
• Urban scene reconstruction using raw LiDAR data
• Image retargeting
• Creation of the impossible
• Multiresolution volume rendering
• Cross-boundary brushes for interactive shape segmentation
• Image color style transfer and colorization
• Photo-sketcher: Interactive sketch based image synthesis
• Snow cover generation
• Real-time realistic ocean lighting
• Face sketch synthesis via sparse representation
• Mesh denoising and smoothing via anisotropic diffusion of surfaces
• Adaptive ray-tracing
• Learning 3D model segmentation
• Graphical model simplification
• Multiscale rendering of 3D models
• Efficient image blending
• Synthesizing high-resolution smoke animation
• Large-scale sketch-based recognition
Journals and Conferences

- Siggraph (Siggraph Asia)
- Eurographics
- Pacific Graphics
- ACM Transactions on Graphics
- IEEE Transactions on Visualization and Computer Graphics
- Computer Graphics Forum
• Geometry-oriented journals and conferences (GMP, SPM, SMI, SGP, Computer-aided Design, CAGD, GMOD, Computers & Graphics)
• Computer Vision (CVPR, ICCV, ECCV)
• Image processing
• VR
• HCI