What is this class about?

This is a programming class.

It is about algorithms that are used to create computer graphics images.

We will not learn how to use animation or rendering software to create animations. Our goal is to learn the basics that are necessary to develop such software.

Computer Graphics

Prerequisites

- **Basic linear algebra and calculus**
- **Solid C/C++ programming**
- **no Java**

Hardware/Software

- **PC, Linux or Windows, MacOS X**
- **other requires permission**
- **need OpenGL and GLUT; instructions for installation will be available**

Grading

- 1-2 written assignments
- 5 programming assignments
- Programming
  - up to 30% quality of code
  - 50% off for crashes without assert

Computer Graphics

- Late policy:
  - 24 hours: 20% off
  - 48 hours: 40% off
  - 72 hours: 80% off

Extensions are for exceptional circumstances only and have to be requested in advance, meaning at least 4 days before the due date;

NO extensions on the last day.

Prerequisites

Programming:

Good programming skills are essential.

Good working knowledge of C++ (or at least C)

is assumed.

The programming load is high; the grade is primarily determined by programming.

Math:

Elementary geometry and linear algebra.

Topics

Emphasis on OpenGL rendering pipeline

Scan conversion

Image processing

Basic modeling

Lighting

Rendering algorithms

What is computer graphics?

Computer science: software and hardware systems, vision, computational geometry

Mathematics: transformations, curves, surfaces, PDEs, numerical integration

Physics: light, dynamics

Psychology: perception

Art

Applications

Entertainment

- **Animation and special effects**
- **Games**

CAD

Scientific Visualization

Medicine

System Visualization

Animation

Computer Animation

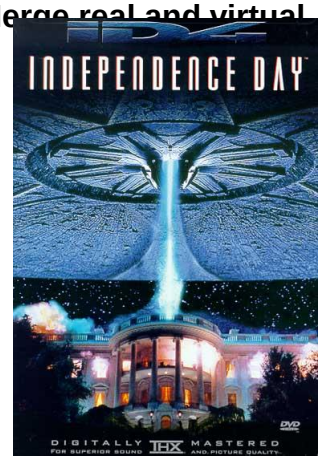
Games

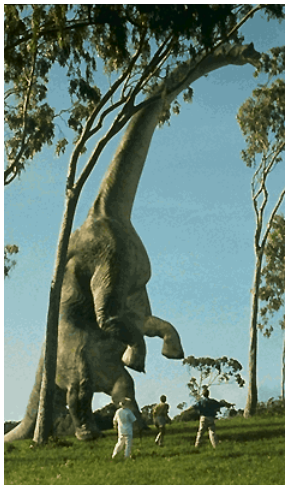
Games

Games

Special effects

Merge real and virtual



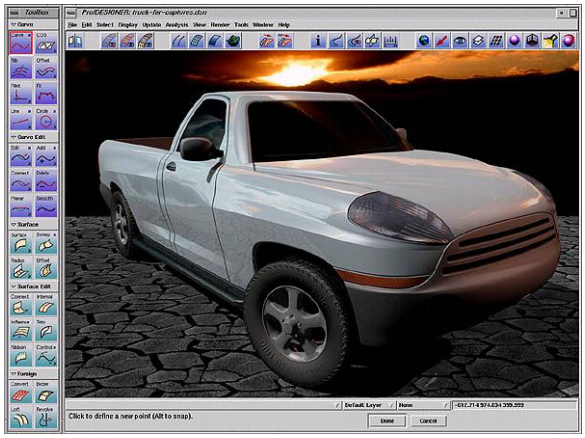


Special effects

Special effects

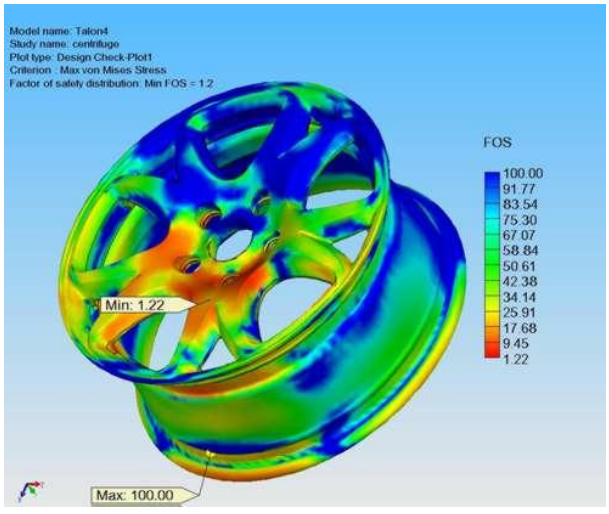
Computer-Aided Design

Conceptual design

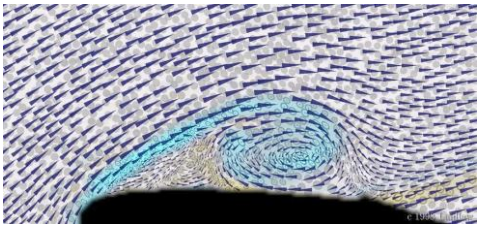


Computer-Aided Design

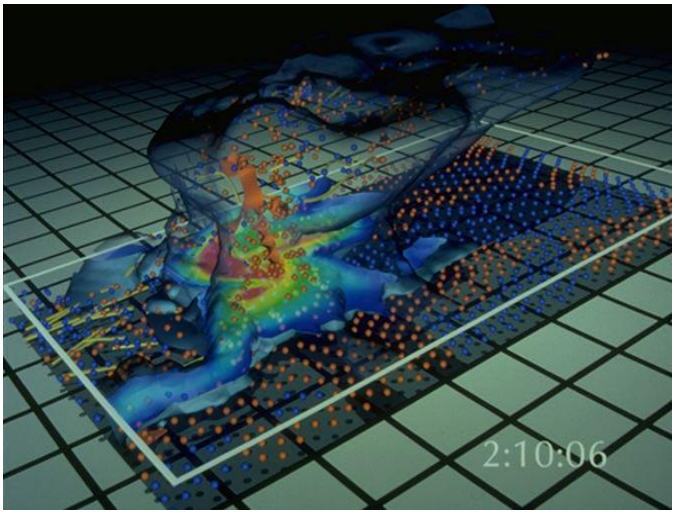
Simulation



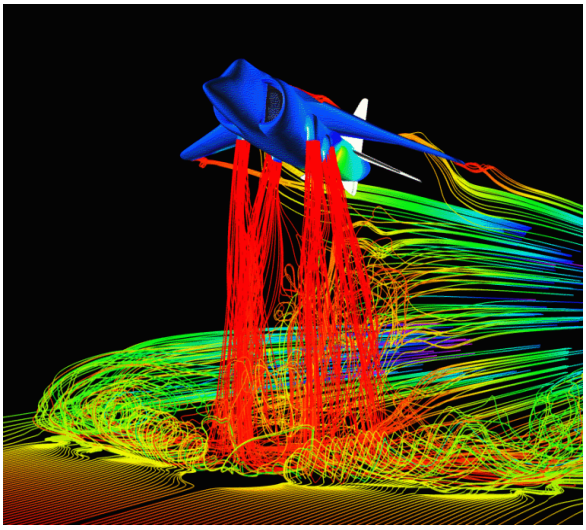
Visualization



Thunderstorm Simulation



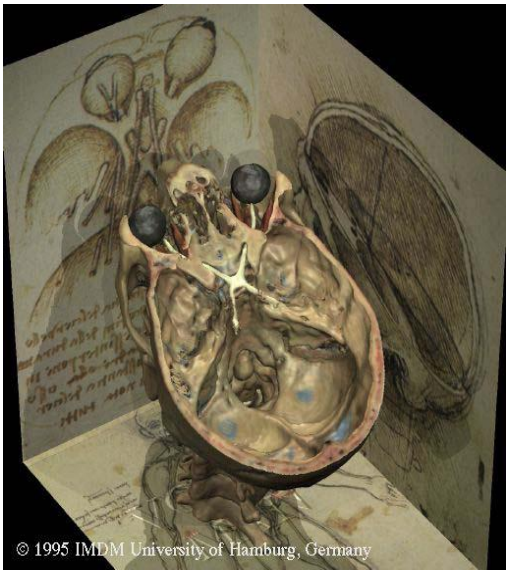
NASA's FAST



Airflow around a Harrier Jet
FAST System

Nasa Ames

Visible Human



Modeling

Create the environment

- **shapes**
- **appearance**
- **views**

make it move

- **define parameters**
- **compute how object shape, position, appearance changes**

Rendering

Physics-based: simulate light propagation

Empirical: use trial and error to get pictures that are good enough

Nonphotorealistic: imitate artistic styles

Image-based: generate directly from photos or video (no modeling)

Image Processing

Output: typically a raster device

(CRT, LCD, printer)

Discrete colors/intensities

Need to convert continuous data to discrete

Combine real and synthetic

Graphics hardware