Introduction to Computer Graphics

Richard (Hao) Zhang
CMPT 361 – Introduction to Computer Graphics
Lecture 1
How are they related to graphics?

- “Ratatouille”
- Robotics and autonomous driving
- 3D printing
- A triangle
- Wii and computer games
- Fourier transform
- OpenGL programming
- AR/VR
- Physics
- Geometry of lines, planes, and curved surfaces
- YouTube
- Matrices and linear algebra
How are they related to graphics?

- “Ratatouille”
- Robotics and autonomous driving
- 3D printing
- A triangle
- Wii and computer games
- Fourier transform
- OpenGL programming
- AR/VR
- Physics
- Geometry of lines, planes, and curved surfaces
- YouTube
- Matrices and linear algebra
But first thing first … in CMPT 361

- “Ratatouille”
- Robotics and autonomous driving
- 3D printing
- A triangle
- Wii and computer games
- Fourier transform
- OpenGL programming
- AR/VR
- Physics
- Geometry of lines, planes, and curved surfaces
- YouTube
- Matrices and linear algebra
In the fourth year …

- “Ratatouille”
- Robotics and autonomous driving
- 3D printing
- A triangle → Triangle meshes (CMPT 464)
- Wii and computer games
- Fourier transform
- OpenGL programming
- AR/VR
- Physics → Computer animation (CMPT 466)
- Geometry of lines, planes, and curved surfaces
- YouTube
- Matrices and linear algebra
Latest and future technologies

- “Ratatouille”
- Robotics and autonomous driving: 3D reconstruction
- 3D printing: Shape modeling and optimization
- A triangle
- Wii and computer games
- Fourier transform
- OpenGL programming
- AR/VR: 3D scene modeling, motion transfer
- Physics
- Geometry of lines, planes, and curved surfaces
- YouTube
- Matrices and linear algebra
Today

- Course expectation and evaluations
- What is computer graphics?
- Computer graphics applications
- Main topics in computer graphics
- Specifically, CMPT 361 course topics at a glance
- Summary: major goals in computer graphics

Disclaimer: the only image and video intensive lecture!
Basic prerequisites

- Good programming background – C++/C (Java not to be used)
- Good Unix exposure - Makefiles, etc.
  - You will most likely be working in a Linux/Unix environment (not Windows) – even if you worked in Windows, your program must compile on the CSIL machines (TA will grade on those machines)
- Basic linear algebra and geometry knowledge
  - Matrices, vectors and associated operations, e.g., dot/cross products
  - Surface normal, line-surface intersection, etc.
- Spatial reasoning
Basic prerequisites

- Basic computer science
  - Basic data structures, e.g., linked lists, trees, arrays, and associated algorithms
- Not afraid of doing some level of math
- Be good at reading (OpenGL) manuals, so that you can be proficient in OpenGL programming more or less by yourself (we will provide four lectures of help including tutorial and exercise sessions)
- 361 is NOT easy: lots of programming + non-trivial amount of math
I am not going to ...

- Teach C++/C
- Teach data structures
- Teach the really basic linear algebra, e.g., matrix vector multiplications
- Teach what each OpenGL function does (learn to read manuals)
- Questions about C++/C are low-priority, consult TA Wallace
- Lab procedures are your responsibility (Read the course web page on the relevant information)
Evaluations

- Three assignments (45%) – programming + written solutions
  - Assignment #1 (15%) – 2D displays and interaction – Tetris or Nibbles?
  - Assignment #2 (15%) – 2D + 3D Scene modeling & primitive animation
  - Assignment #3 (15%) – Ray tracer
- Midterm test (20%)
- Final exam (35%)
Past images created by CMPT 361 students
3D pacman is real!
Meow and Hai!
Cute!
What does not belong? 😊
Look really closely …
Dangerous and dark …
Fun and creative!
What is computer graphics (CG)?

Computer graphics is concerned with all aspects of producing and displaying images (possibly moving images) using a computer, i.e., computer-aided image synthesis and display.
More specifically, it is about …

the creation, storage, display, animation, and manipulation of models and images of real-world or imaginary data

www.eatoutgoout.com
Computer graphics applications

- Display of information
- Design and manufacturing
- Simulation and animation
- User interfaces

Follows treatment of the course text; there are many other ways of looking at the numerous applications
Display of information

Scientific visualization

Geographical studies
Hawaii island of Oahu (dvl.sdsc.edu)
– see Color Plates 25, 26 in text

Medical imaging
(Also see Color Plate 20)
Display of information

- Weather and fluid dynamics – e.g., Color Plate 19 in text
- Molecular biology
- Urban planning & advertising – e.g., Vancouver Olympic bid
- Art-historical study
- Information visualization (textural information using 2D or 3D graphics)

Michelangelo’s Pietà
[Bernardini et al. 01]

cubiceye™
www.2ce.com
Design and manufacturing

Polygonal modeling used for manufacturing of hearing aids
(Raindrop Geomagic: http://www.raindropgeomagic.com/products/eshell/)

SLA, SLS: Solid Imaging systems from 3dsystems Inc.

Solid Imaging systems: machineries that produce a solid 3D object of certain special material, given a digital representation of the model of the object, e.g., 3D printer
Laser scanning for data acquisition
Design and manufacturing

- Automobile engineering
- Aerospace engineering
- Architectural design
- Custom manufacturing, e.g., for injection molding
- Design of mechanical parts to VLSI circuits
- etc.

New Amsterdam Theater, NYC
[Jason Ardizzone, 1994]
Architectural geometry
Simulation and animation

Computer games
Simulation and animation

- Flight simulation (more serious)
- Animated or feature films
  - Lord of the Ring
  - Ice Age, Shrek
  - Monsters Inc.
  - Tin tin
  - Life of Pi
- Many other virtual reality applications
Graphical user interfaces

- Mac OS, Windows, CentOS Linux
- Chrome, Mozilla Firefox, Internet Explorer, etc.
- Practically all the software we are using today, maybe not some of the old purely text-based Linux/Unix ones

They are so ubiquitous that we are not even aware that we are working with computer graphics
Main topics of CG

Recall: computer graphics is about the creation, storage, display, animation, and manipulation of models and images of real-world or imaginary objects or data.

www.eatoutgoout.com
Main topics

- **Modeling** — the creation of digital representations (→ CMPT 464)
  - For real-world physical objects, e.g., engineering, architectural, or medical
  - Abstract or conceptual models, e.g., those from mathematical simulations
  - Natural phenomena, e.g., snow, fire

- **Rendering** — display (→ CMPT 461)
  - Image formation using projections, colors, shading, lighting effects, textures, composition, etc.
Main topics continued

- **Human computer interaction** — (CMPT 363)
  
  Manipulation of model and data through effective input devices and intelligent graphical interfaces

- **Computer animation** — (CMPT 466)
  
  Representation and control of motion to achieve physical realism and to tell an interesting story, but actually quite a bit of modeling too!

- **Storage and transmission** — (CMPT 464)
  
  Compact and efficient means to handle massive graphics data, e.g., JPEG, MPEG-4 — a relatively new topic due to tremendous growth of scene complexity, e.g., with the advent of 3D laser scanners
Related courses at SFU

- CMPT 363 – User Interface Design
- CMPT 340 – Computers in Biomedicine
- CMPT 365 – Multimedia Systems
- CMPT 461/761 – Image Synthesis (hopefully fall 2014)
- CMPT 467/767 – Visualization (probably not in very near future)
- CMPT 466 – Animation (likely 2015)
Related courses at SFU

- CMPT 414 – Model-Based Computer Vision – **image understanding**
- CMPT 412 – Computational Vision
- CMPT 406 – Computational Geometry

- **NEW in the works:** Professional Masters Program in Visual Computing, to be launched in September 2018
Computer graphics at SFU

Eugene Fiume  Yasu Furukawa  Ping Tan  Kangkang Yin  Hao Zhang
Computer graphics at SFU

- Fiume (2017): rendering; natural phenomena; computational physics
- Furukawa (2017): 3D vision; scene analysis and reconstruction; ML
- Tan (2014): 3D vision; SLAM; UAV; image-based modeling
- Yin (2017): computer animation and simulation; geometric modeling
- Zhang (2003): geometric modeling; shape analysis; fabrication; ML
### Computer Science Rankings

Click on a triangle (▲) to expand areas or institutions. Click on a name to go to a faculty member's home page. Click on a pie (the ● after a name or institution) to see their publication profile as a pie chart. Click on a Google Scholar Icon (●) to see publications, and click on the row number of publications to go to a DBLP entry.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Institution</th>
<th>Count</th>
<th>Faculty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Zhejiang University</td>
<td>20.0</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>Massachusetts Institute of Technology</td>
<td>29.2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Stanford University</td>
<td>27.8</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>ETH Zurich</td>
<td>24.2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Carnegie Mellon University</td>
<td>19.8</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>University College London</td>
<td>19.2</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Max Planck Institute</td>
<td>17.1</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>University of British Columbia</td>
<td>16.7</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>New York University</td>
<td>14.0</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>Cornell University</td>
<td>14.0</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>Columbia University</td>
<td>15.0</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>Tel Aviv University</td>
<td>15.0</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>University of California - San Diego</td>
<td>14.0</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>University of Toronto</td>
<td>13.6</td>
<td>6</td>
</tr>
<tr>
<td>15</td>
<td>Simon Fraser University</td>
<td>12.5</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>Texas A&amp;M University</td>
<td>12.0</td>
<td>6</td>
</tr>
<tr>
<td>17</td>
<td>IST Austria</td>
<td>11.4</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>KAUST</td>
<td>11.4</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>Princeton University</td>
<td>11.2</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>University of Southern California</td>
<td>11.0</td>
<td>5</td>
</tr>
<tr>
<td>21</td>
<td>University of Tokyo</td>
<td>10.3</td>
<td>4</td>
</tr>
</tbody>
</table>
Computer graphics and computer vision
More on modeling and rendering

- Modeling
- Rendering
- Interaction (CMPT 363)
- Animation (CMPT 466)
Modeling

- Information needed to represent a scene, real or imaginary
  - **Geometry**: curves, surfaces, volumes, etc.
  - **Photometric information**: light, color, reflectance, etc.

- How to build geometric representations
  - Declarative: conceptualize and write it down
  - Constructive: sculpt it, merge, intersect, etc., starting from simple shapes
  - Procedural: let it grow, rule-based, e.g., fractals
  - Acquired: via 3D laser scanning
Declarative geometric modeling

Functional representation

- Parametric surfaces:
  \((x(s, t), y(s, t)),\)
  
  \(-1 \leq s \leq 1, -1 \leq t \leq 1\)

- Implicit surface models, e.g.,
  \(f(x, y, z) = 0\)
Constructive geometric modeling

**Constructive solid geometry** (CSG): built from spheres, cylinders, ellipsoids, etc., through merging, intersection, differencing, etc.
Procedural models

... for objects that are hard to model using surfaces, e.g., trees, water, smoke, cloud, etc. [See Chapter 9 of text — not covered in CMPT 361]

- **Particle systems** for clouds, fire, etc.
- **Fractals** for trees, mountains, etc.

Particles are generated and fractals are grown according to specific rules, e.g.,

![Diagram of procedural models]

- Very interesting research problem: *inverse* procedural modeling
A fractal mountain

Google search on “fractal mountains”. Also see Exercise 2.1 of the Angel text.
Declarative & acquired: polygonal meshes

**Polygonal mesh** – a surface composed of polygons pasted along their edges

- **Declarative:**
  - Specify all vertex locations and their connectivity, or
  - Special all polygons/triangles

- **Acquired:**
  - Polygonal meshes are often acquired via laser scanning
Polygonal meshes

Polygonal mesh – the most popular free-form surface representations

Main reason: computers/GPUs are very fast at polygon rendering!
Acquisition of polygonal meshes

- Mostly from 3D scanning
- David:
  - 480 individually aimed scans
  - 2 billion polygons
  - 7,000 color images
  - 32 gigabytes
  - 30 nights of scanning
  - 22 people
- Think processing …
3D digitizer and scanner

InSpeck

Roland DGA LPX-250
($10K CDN)
www.3dtechnics.com

Cyberware

trimensional
3D Scanner for iPhone
Available now on the App Store Trimensional
Laser scanning for data acquisition
A hybrid: subdivision

**Subdivision surfaces**: recursively subdivide and refine the polygons according to specific rules

http://grail.cs.washington.edu/projects/subdivision/

It is really *procedural but in the form of a polygonal mesh.*
Geometric modeling in CMPT 361

- Concentrate on polygonal models for geometry
  - Start in 2D: points, lines, and polygons
  - 3D polygonal meshes and subdivision surfaces
- Coverage on parametric curves and surfaces (the smooth worlds)

OK, we have a geometric model, what now?
Have to display it on the screen – rendering
Add in viewing information:

- **Transformation**: transform our geometric model depending on position and orientation of camera
- **Visibility**: determine what can be seen (do not draw what cannot be seen)

Add in light: **illumination and shading**

Add in texture: **texture mapping**

Add in color: some coverage on **color models/representations**
Shading and texture mapping

Modeling

Act on these primitives

Rendering
The quest for visual realism

Model

Model + Shading

Model + Shading + Textures

At what point do things start looking real?

For more info on the computer artwork of Jeremy Birn see http://www.3drender.com/jbirm/productions.html
Summary: The graphics pipeline

What primitives are in the scene and how to represent them

Modeling

Transform

Visibility

Add in camera, lights, and arrangement of primitives

Hidden surface removal

Rasterization

Projection and clipping

Illumination and Shading, texture mapping

We will cover the basics of the whole process in CMPT 361
Topics of CMPT 361 (1)

- Image formation and two fundamental graphics algorithm used for image formation
- Graphics displays, input and interactions
- Programming basics using OpenGL – the graphics library we use
- 2D raster graphics – scan conversion (rasterization) of simple primitives (lines, circles, polygons) and clipping
- 2D and 3D geometric transformations – rotation, translation, use of homogenous coordinates, etc.
Topics of CMPT 361 (2)

- 3D viewing – transformations, hidden surface removal, clipping, etc.
- Illumination and shading – local models, global models, color models, texture mapping
- Curves and surfaces
- Polygonal meshes and subdivision surfaces
- Sampling issues and antialiasing
Overall goal of CG

- Create computer graphics models and make them appear and behave the way we want ...

“Mount meshmore” [Igor Guskov et al., 1999]

in a useful way (quality & usability) and via practical means (cost, e.g., speed and storage cost)
Quality goals of CG – visual realism

- **Photo-realistic** rendering (modeling of the real world)
Visual realism

- Realistic animation, e.g., digital actor walking the way we do
Another kind of visual realism

- **Non-photorealistic** rendering – the way it should be
  (see: www.red3d.com/cwr/npr/)

© 1996 *Barbara J. Meier*

Courtesy: Alejo Hausner
Why bother with computer graphics when we have photos, video, and traditional art?

Answer: With the right computer graphics models and techniques, we can do much more!
Great entertainment

- Computer games
- Animated films
- Flight simulation, etc.

- Realizing an **imaginary world** with images and animations
- Navigating and interacting with the imaginary world
Power of creativity and perception

- Playing around with different imaging effects (e.g., lighting, texturing), or other manipulations, without physical construction of the world

- Digitally edit and examine a machine part before manufacturing — great cost saving

- Display of information
  - Graphical display of non-visual information, e.g., stock charts — a picture is worth a thousand words
  - Visualize and better understand complex phenomena that are impossible or difficult to capture with a camera or be drawn, e.g., network traffic, wind flow over the body of an automobile, etc.
Goals of CG related to cost

- **Speed, speed, and more speed** (30 frames per second)
  - The complexity of computer graphics models will continue to grow
    - The Pieta model took 90 hours over 14 days using 800 separate scans
    - Raw data file size: 3 GB
  - Interactivity is essential in practically all applications, e.g., simulation, computer game, computer-aided design
- Memory and simplicity of programs also are concerns

Michelangelo’s Pietà
[Bernardini et al. 01]
Usability issues

- Make computer graphics **easy to use**, e.g.,
  - More user-friendly graphical user interfaces (HCI)
    - e.g., more intuitive visualization and navigation in 3D environment
  - More effective in conveying information (scientific and information visualization)
  - Easy-to-use animation software, so that everyone can make their own (short) movie
- Make computer graphics **ubiquitous**
Summary of goals

- **Quality**: photorealistic or non-photorealistic rendering and animation
- **Interactivity**: speed through innovative modeling techniques and algorithms
- **Ease of use**: intelligent user interfaces, input devices, and well-designed graphics software
- Finally, **new applications** of computer graphics and make computer graphics ubiquitous
Disclaimer: True physics vs. CG tricks

To achieve visual realism

- The rendering process tries to resemble the way physics and our visual system works
- Doing physics exactly is often too expensive, remember we want speed, speed, and more speed
- A lot of simplified models and computer graphics tricks (Look good? Ok!)
- In CMPT 361, we study the most basic of these models and tricks
Next: Image formation

- Overview of a graphics system
- The raster display
- Image formation
- Two fundamental image formation algorithms
  - Z-buffer
  - Ray tracing
Later on (week 4): OpenGL basics

- Two introductory lectures
  - OpenGL API overview and modern architecture
  - OpenGL functions and shaders (GLSL)
  - Program structure through a simple example
  - Input and interactions --- to prepare already for your 2D game
- Followed by two tutorial and exercise sessions
  - Given by TA
  - Conducted in the CSIL lab aimed to provide hands-on experience