A Brief History of Graphics

Stephen J. Guy

Sep 11, 2017

Objectives

- History of Graphics (High level overview)
- Understand legacy left by historical developments
- Introduction to basic graphics program
- Ensure everyone feels comfortable with HW0

2

Notes:

- If you missed last class, HW0 is already posted
- No need to learn OpenGL (yet!), focus is mainly on making sure you download the libraries
  - 75 pts just for getting the sample code to compile
- I've posted readings online

3

Recap

- What is Computer Graphics?

Computer graphics: The study of creating, manipulating, and using visual images in the computer.

Computer graphics:
Mathematics made visible.

4

Recap

- What are the main areas of graphics discussed last time?
  - Imaging
  - Modeling
  - Rendering
  - Animation
  - Hardware
- What are their differences?

5

Outline

- History of Graphics
- Pre-History
- Early Work
- Rise of the GPU
- Today's Graphics & OpenGL

6

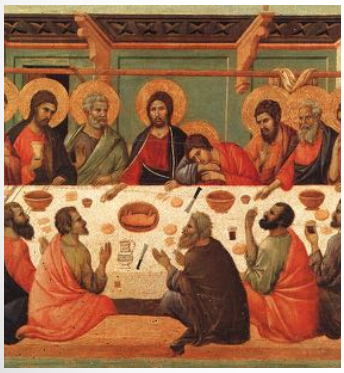
Outline

- History of Graphics
- **Pre-History**
- Early Work
- Rise of the GPU
- Today's Graphics & OpenGL

7

Renaissance Perspective

- Geometry of vision
  - Early theories by Greeks
  - Mastery in the Renaissance



Duccio c. 1308

Duccio di Buoninsegna

da Vinci c. 1498

8

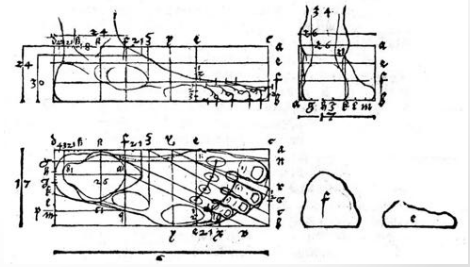
Perspective

- Optical Techniques, Mathematical Ideas, and Secret Knowledge spread through Renaissance Europe.
  - A quantum jump in image realism
- Painting is the rendering problem
  - More on the math of perspective latter
- How might you generate a prospectively correct image?

9

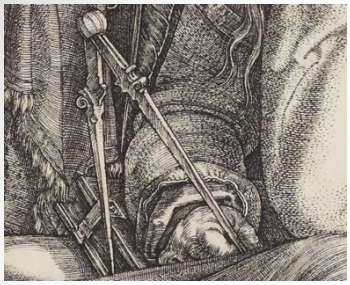
Albrecht Durër, 1471-1528

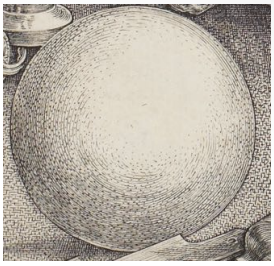
Self Portrait

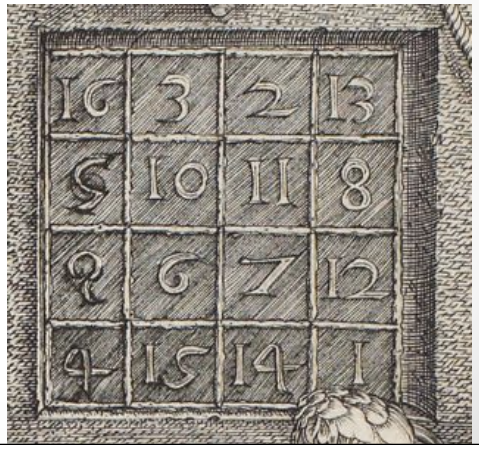


Durër, 1510

11

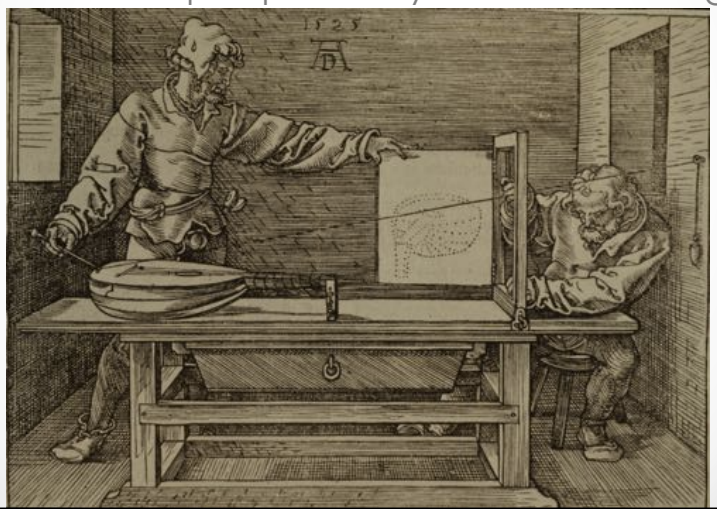





16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1

Durër Woodcuts

- 1525 Albrecht Durër made a wood cut showing how to make prospectively correct drawings



15

Outline

- History of Graphics
- Pre-History
- **Early Work**
- Rise of the GPU
- Today's Graphics & OpenGL

16

Graphics in the 60s & 70s

- Graphics starting seeing rapid progress in 60s
- Resources were highly limited
  - 1 display could be a years salary!
  - 1 display per institution was common
- 1970's era mainframe computers = ~1 MIPS
- 640x480 image = .3 mega pixels
  - There was not a lot of processing available to be done to each image

17

Resource Limitations

- Constrained resources cast a long shadow of the field
- Still a strong interest in:
  - Tiny optimizations
  - Fast programming languages
  - Clever tricks
  - Non-physical (but good looking) methods
- Graphics is a field with a long tradition of hacks!

18

"Hacks" in Graphics

- Light fall off
  - Light brightens close things more.
  - Real light falls as $1/d^2$. Why?
  - Graphics use 1/d. Because it looks better!?
- Light Transport
  - Can we see objects with no direct lighting? Why?
  - Light bounces off all surfaces in rooms.
  - Ehh, too hard / too slow…
  - …lets just add a bit of light to all pixel
  - "**Ambient**" component of lighting models

19

Outline

- History of Graphics
- Pre-History
- Early Work
- **Rise of the GPU**
- Today's Graphics & OpenGL

20

GPU

- Graphical Processing Unit
  - Special processor just to help with graphics
  - Very parallel
  - Special hardware for:
    - Linear algebra
    - Projections
    - Trig
    - Blending
    - Interpolation
- The process for rendering image was hardcoded into the GPU
  - Including common hacks!

21

(Old) Graphics Pipeline

- Input to GPU:
  - 3D Triangle positions & color at each vertex; 3D Camera position & fov, 3D light positions
- GPU:
  - Converts 3D triangles to 2D (projection)
  - Breaks 2D triangles on fragments (pixels)
  - [Finds closest fragment] (z-buffering)
  - Computes lighting of each fragment/pixel (shading)
- Output:
  - A picture!

22

(New) Graphics Pipeline

- Converts 3D triangles to 2D (projection)
  - Let users write their own projection code
- Breaks 2D triangles on fragments (pixels)
- Computes lighting of each fragment/pixel (shading)
  - Let users write their own lighting code
- Shaders:
  - Programs which run on the GPU
  - Historically to replace part of the graphics pipeline
  - Special languages used, e.g., GLSL

23

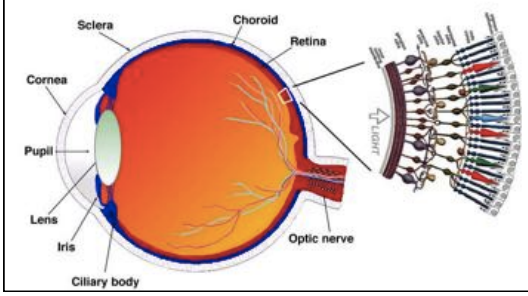
Outline

- History of Graphics
- Pre-History
- Early Work
- Rise of the GPU
- **Today's Graphics & OpenGL**

24

Recent Trends (Offline)

- Fully simulate the real physics
  - Physically-based rendering
- Understand the human visual system and display systems.
  - Gamma corrections
  - Tone mapping



25

GPUs

- Today: GPUs are FAST!

Graphics Programming Today

- Offline Graphics & 2D Graphics:
  - Largely based around writing or editing custom software
- Online 3D Graphics (e.g., games):
  - Need to access these powerful GPUs
  - Several libraries must be involved:
    - OpenGL
    - GLAD
    - SDL

27

OpenGL 2.0+

- OpenGL is an API for computing with the graphics card.
- OpenGL 2.0 was released in Sept, 2004.
  - centered around the **new pipeline** of having the users write shaders
  - More flexible, but harder to get started
- **Alternatives:** Microsoft DirectX API is very similar and used in many games and on the XBox system

28

GLAD

- Different drivers (e.g., nVidia vs ATI vs Intel) support slightly different version and custom extensions of OpenGL.
- Some extensions are "optional" and needed to be loaded specially
  - The extensions are needed to write your own shaders!
- GLAD is a tool to simplify this process:
  - Web interface to generate .c/.h files that load extensions
- **Alternatives:** Hard work, GLEW, OpenGL 1.0

29

SDL

- Ultimately we need to display images, take in input, & send output (e.g., sound)
  - Each OS has a different way of doing this…
- The Simple Direct Media Layer 2.0 (SDL2) provides a single, cross-platform way of doing this.
  - Widows, Keyboard & Mouse Input, Audio Output
- **Alternatives:** FreeGLUT, GLUT, QT, Windows API

30

Running the Code

- Download & Install:
  - SDL2 Binaries
- Include in project:
  - glad.c and glad.h

- SDL library will be installed soon on lab machines

- Be sure to link to the libraries at compile time
- On OSX:

```
> g++ hw0.cpp glad/glad.c -framework SDL2  -framework OpenGL
```

31