CG apes from Dawn of the Planet of the Apes, 2014

Introduction

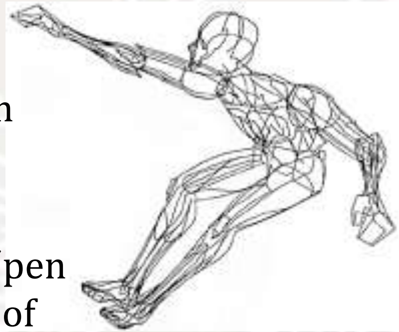
Introduction to Computer Graphics

What is Computer Graphics? (1/2)

- Computer graphics generally means creation, storage and manipulation of models and images
- Such models come from diverse and expanding set of fields including physical, biological, mathematical, artistic, and conceptual/abstract structures

Frame from animation by William Latham, shown at **SIGGRAPH 1992**. Latham creates his artwork using rules that govern patterns of natural forms.

What is Computer Graphics? (2/2)



- William Fetter coined term “computer graphics” in 1960 to describe new design methods he was pursuing at Boeing for cockpit ergonomics
- Created a series of widely reproduced images on “pen plotter” exploring cockpit design, using 3D model of human body.

“Perhaps the best way to define computer graphics is to find out what it is not. It is not a machine. It is not a computer, nor a group of computer programs. It is not the know-how of a graphic designer, a programmer, a writer, a motion picture specialist, or a reproduction specialist.

Computer graphics is all these – a consciously managed and documented technology directed toward communicating information accurately and descriptively.”

Computer Graphics, by William A. Fetter, 1966

What is **Interactive** Computer Graphics? (1/3)

- User controls content, structure, and appearance of objects and their displayed images via rapid visual feedback
- Basic components of an interactive graphics system
  - input (e.g., mouse, tablet and stylus, multi-touch…)
  - processing (and storage)
  - display/output (e.g., screen, paper-based printer, video recorder…)
- First truly interactive graphics system, **Sketchpad**, pioneered at MIT by Ivan Sutherland for his 1963 Ph.D. thesis
- Used TX-2 transistorized "mainframe" at Lincoln Lab

Note CRT monitor, light pen and function-key panel

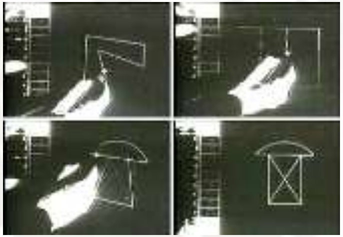
DEMO: SKETCHPAD

http://youtu.be/J6UAYZxFwLc

What is **Interactive** Computer Graphics? (2/3)

- Almost all key elements of interactive graphics system are expressed in first paragraph of Sutherland's 1963 Ph.D. thesis, *Sketchpad, A Man-Machine Graphical Communication System:*

> *The Sketchpad system uses drawing as a novel communication medium for a computer. The system contains input, output, and computation programs which enable it to interpret information drawn directly on a computer display. Sketchpad has shown the most usefulness as an aid to the understanding of processes, such as the motion of linkages, which can be described with pictures. Sketchpad also makes it easy to draw highly repetitive or highly accurate drawings and to change drawings previously drawn with it…*



What is **Interactive** Computer Graphics? (3/3)

- Today, still use non-interactive *batch mode* for final production-quality video and film (special effects – FX). Rendering a single frame of Monsters University (a 24 fps movie) averaged 29 hours on a 24,000-core render farm!

Still from Monsters University

Render farm

Enabling Modern Computer Graphics (1/5)

- Hardware revolution
  - **Moore's Law:** every 12-18 months, computer power improves by factor of 2 in price / performance as feature size shrinks
  - Significant advances in commodity graphics chips every 6 months vs. several years for general purpose CPUs
    - NVIDIA GTX Titan Z... 8122 gigaflops
  - Newest CPUs are 64-bit, 2, 4, 6, 8, or 10 core
    - Intel Core i7 – consumer, up to 6 cores hyperthreaded to provide 12 threads
    - Intel Haswell – industrial, 8 cores HT, 16 threads

Enabling Modern Computer Graphics (2/5)

- Graphic subsystems
  - Offloads graphics processing from CPU to chip designed for doing graphics operations quickly
  - nVidia GeForce™, AMD Radeon™
  - GPUs originally designed to handle special-purpose graphics computations
  - Increasingly, GPUs used to parallelize other types of computation (known as **GPGPU**, or General-Purpose Computing on the Graphics Processing Unit)
- Hardware show and tell: Dept's NVIDIA GeForce GTX 460s
  - 1.35 GHz clock, 1GB memory, 37.8 billion pixels/second fill rate
  - Old cards: GeForce 7300 GT: 350 MHz clock, 256 MB memory, 2.8 billion fill rate

Enabling Modern Computer Graphics (3/5)

- Input Devices
  - Mouse, tablet & stylus, multi-touch, force feedback, and other game controllers (e.g., Wii), scanner, digital camera (images, computer vision), etc.
  - Body as interaction device
    - http://youtu.be/zXghYjh6Gro

Xbox Kinect

Leap Motion

Nimble UX

Enabling Modern Computer Graphics (4/5)

- Many form factors
  - Smartphones/laptops/desktops/tablets
  - Microsoft PPI display
  - Smart watches
  - Head-mounted displays (HMDs)
    - [Oculus bird simulator video]
  - 3D immersive virtual reality spaces

Apple iPhone

Samsung Galaxy S5

Samsung Galaxy Tab

Microsoft's first Surface

Microsoft PPI display

Android Wear

Brown's old Cave

Google Glass

Oculus Rift

Google Cardboard

Digression: Cave Redesign

- Old Cave:
  - 4 1024 x 786 projectors on 8' x 8' walls (8-10 pixels per inch)
  - Too low resolution and brightness for many applications, and got worse (brightness, contrast deteriorated over time)
- New Cave:
  - 69 projectors onto cylindrically curved screen 8' radius, floor, ceiling
  - 140 million pixels
  - Powered by a ~69 gpu cluster
  - No right angles, up to 40 pixels per inch (can't see individual pixels at normal viewing distance)

Brown's new Cave under construction in 2013

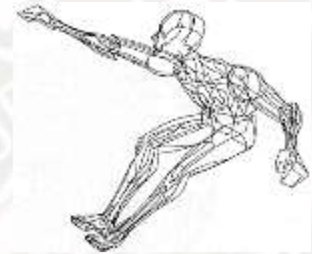
Enabling Modern Computer Graphics (5/5)

- Software Improvements
  - Algorithms and data structures
    - Modeling of materials
    - Rendering of natural phenomena
    - "Acceleration data structures" for ray tracing and other renderers
  - Parallelization
    - Most operations are embarrassingly parallel: changing value of one pixel is often independent of other pixels
  - Distributed and Cloud computing
    - Send operations into 'cloud', get back results, don't care how
    - Rendering even available as internet service!

Environmental Evolution (1/5)

- Character Displays (1960s – now)
- Display: text plus alphamosaic pseudo-graphics (ASCII art)
- Object and command specification: command-line typing
- Control over appearance: coding for text formatting
  (.p = paragraph, .i 5 = indent 5)
- Application control: single task

Environmental Evolution (2/5)



- Vector (Calligraphic, Line Drawing)
- Displays (1963 – 1980s)
- Display: line drawings and stroke text; 2D and 3D transformation hardware
- Object and command specification: command-line typing, function keys, menus
- Control over appearance: pseudo-WYSIWYG
- Application control: single or multitasked, distributed computing pioneered at Brown via mainframe host <-> minicomputer satellite
- Term “vector” graphics survives as “scalable vector graphics” SVG library from Adobe and W3C – shapes as transformable objects rather than just bitmaps

Environmental Evolution (3/5)

- 2D bitmap raster displays for PCs and workstations (1972 at Xerox PARC - now)
- Display: windows, icons, legible text, “flat earth” graphics
  - Note: late 60’s saw first use of raster graphics, especially for flight simulators
- Minimal typing via WIMP GUI (Windows, Icons, Menus, Pointer): point-and-click selection of menu items and objects, direct manipulation (e.g., drag and drop), “messy desktop” metaphor
- Control over appearance: WYSIWYG (which is really WYSIAYG, What You See Is All You Get – not pixel-accurate or controllable)
- Application control: multi-tasking, networked client-server computation and window management (even “X terminals”)

Above, a classic WIMP interface. The technology, at its core, remains largely the same today. Below, a modern WIMP

Environmental Evolution (4/5)

- 3D graphics workstations (1984 at SGI – now)
  - could cost up to $1M for high-end!
- Display: real-time, pseudo-realistic images of 3D scenes
- Object and command specification: 2D, 3D and N-D input devices (controlling 3+ degrees of freedom) and force feedback haptic devices for point-and-click, widgets, and direct manipulation
- Control over appearance: WYSIWYG (still WYSIAYG)
- Application control: multi-tasking, networked (client/server) computation and window management

Graphics workstations such as these have been replaced with commodity hardware (CPU + GPU),
e.g., our MaxBuilts + Nvidia cards

Environmental Evolution (5/5)

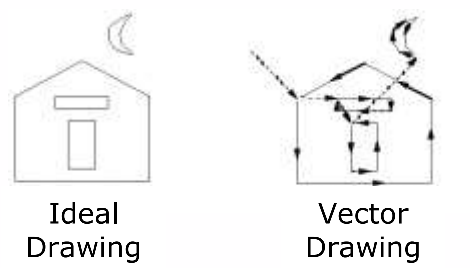
- High-end PCs with hot graphics cards (nVidia GeForce™, AMD Radeon™) have supplanted graphics workstations
- Such PCs are clustered together over high speed buses or LANs to provide “scalable graphics” to drive tiled PowerWalls, Caves, etc.
- Also build GPU-clusters as number cruncl e.g., protein folding, weather prediction
  - Now accessible to consumers via technologies like NVIDIA’s **SLI (Scalable Link Interface) bridge**

You can put multiple GPUs together in your computer using SLI.

Graphics Display Hardware

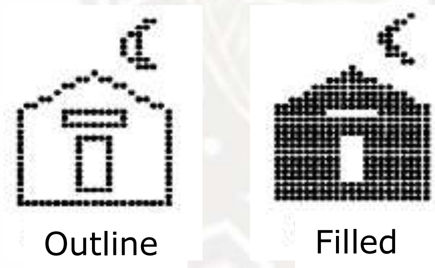
Vector (calligraphic, stroke, random-scan)

- Driven by display commands
  - (move (x, y), char("A") , line(x, y)…)
- Survives as "scalable vector graphics"



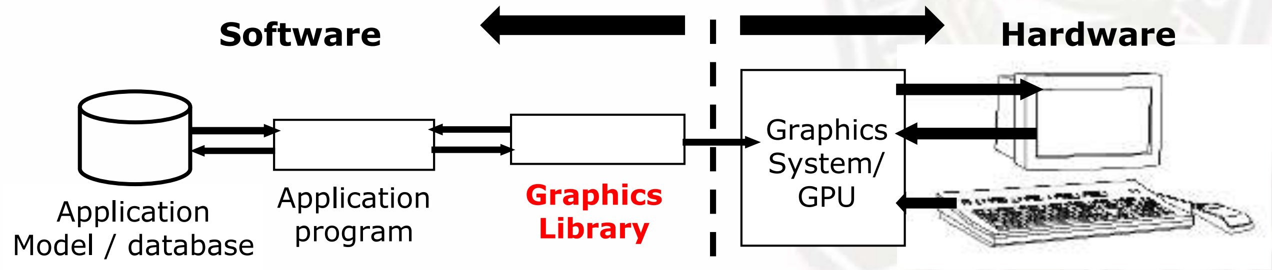
Raster (TV, bitmap, pixmap) used in displays and laser printers

- Driven by array of pixels (no semantics, lowest form of representation)
- Note "jaggies" (aliasing errors) due to discrete sampling of continuous primitives



Conceptual Framework for Interactive Graphics

- Graphics library/package is **intermediary** between application and display hardware (Graphics System)
- Application program maps application objects to views (images) of those objects by calling on graphics library. Application model may contain lots of non-graphical data (e.g., non-geometric object properties)
- User interaction results in modification of image and/or model
- This hardware and software framework is 5 decades old but is still useful

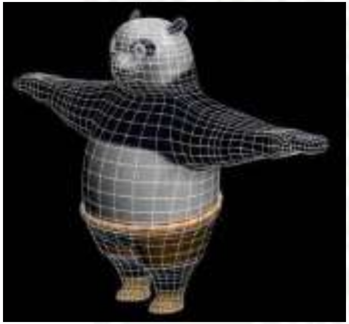


Graphics Library

- Examples: OpenGL™, DirectX™, Windows Presentation Foundation™ (WPF), RenderMan™, HTML5 + WebGL™
- Primitives (characters, lines, polygons, meshes,…)
- Attributes
  - Color, line style, material properties for 3D
- Lights
- Transformations
- Immediate mode vs. retained mode
  - immediate mode: no stored representation, package holds only attribute state, and application must completely draw each frame
  - retained mode: library compiles and displays from scenegraph that it maintains, a complex DAG. It is a display-centered extract of the Application Model

Application Distinctions: Two Basic Paradigms

Sample-based graphics *vs* Geometry-based graphics



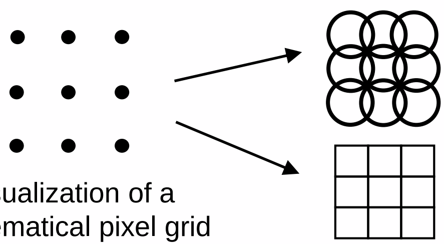
Sample-based Graphics (1/3)

- **Sample-based graphics:** Discrete samples are used to describe visual information
  - pixels can be created by digitizing images, using a sample-based "painting" program, etc.
  - often some aspect of the physical world is sampled for visualization, e.g., temperature across the US
  - example programs: Adobe Photoshop™, GIMP™ , Adobe AfterEffects™ (which came out of CS123/CS224!)

Sample-based Graphics (2/3)

- **Pixels** are point locations with associated sample values, usually of light intensities/colors, transparency, and other control information
- When we sample an image, we sample the point location along the *continuous signal* and we *cannot* treat the pixels as little circles or squares, though they may be displayed as such

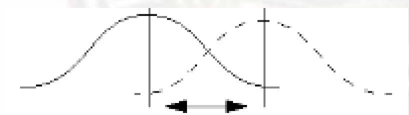
CRT beam illumination pattern



Visualization of a mathematical pixel grid

LCD display

light intensity



1 pixel

Can't visually resolve adjacent pixels on CRT

Sample-based Graphics (3/3)

- Samples created directly in paint-type program, or by sampling of continuous (analog) visual materials (light intensity/color measured at regular intervals) with many devices including:
  - flatbed and drum scanners (e.g., http://www.luminouslandscape.com/reviews/scanners/drum_scans.shtml)
  - digital still and motion (video) cameras
- Sample values can also be input numerically (e.g., with numbers from computed dataset)
- Once an image is defined as pixel-array, it can be manipulated
  - **Image editing:** changes made by **user**, such as cutting and pasting sections, brush-type tools, and processing selected areas
  - **Image processing:** algorithmic operations that are performed on image (or pre-selected portion of image) without user intervention. Blurring, sharpening, edge-detection, color balancing, rotating, warping. These are front-end processes to Computer Vision.

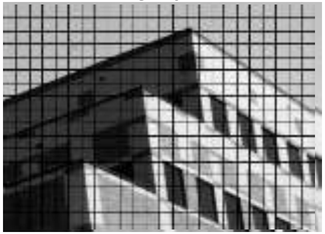
Sampling an Image

- Lets do some sampling of CIT building

3D scene

- A color value is measured at every grid point and used to color corresponding grid square

0 = white, 5 = gray, 10 = black



- Crude sampling and image reconstruction method creates blocky image

What’s the Advantage?

- Once image is defined in terms of colors at (x, y) locations on grid, can change image easily by altering location or color values
- E.g., if we reverse our mapping above and make 10 = white and 0 = black, the image would look like this:
- Pixel information from one image can be copied and pasted into another, replacing or combining with previously stored pixels

What's the Disadvantage?

- WYSIAYG (What You See Is All You Get): No additional information
  - no depth information
  - can't examine scene from different point of view
  - at most can play with the individual pixels or groups of pixels to change colors, enhance contrast, find edges, etc.
  - But increasingly great success in image-based rendering to fake 3D scenes and arbitrary camera positions. New images constructed by interpolation, composition, warping and other operations.
  - Take James Hays's *Computational Photography* (CS1950G)

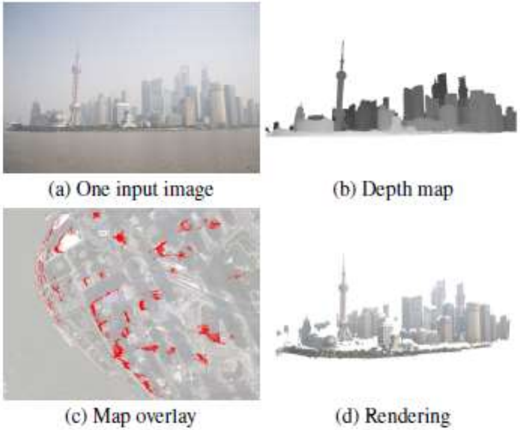
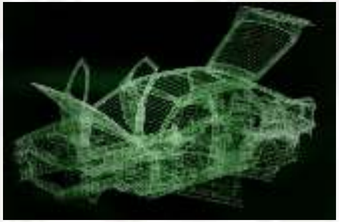


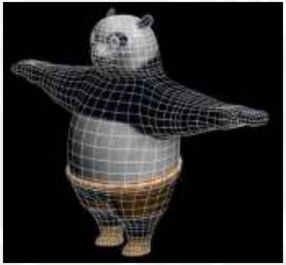
Figure 15: *Results on a challenging unstructured light field, obtained by hand-held capture (a) from a floating boat. (b) A resulting depth map. (c) Overlay of our reconstruction on a satellite image ©2013 DigitalGlobe, Google. (d) Rendering from a novel viewpoint.*

"Scene Reconstruction from High Spatio-Angular Resolution Light Fields" by Kim, Zimmer et al., 2013

Geometry-Based Graphics (1/2)

- **Geometry-based graphics** (also called scalable vector graphics or object-oriented graphics): geometrical model is created, along with various appearance attributes, and is then sampled for visualization (rendering, a.k.a image synthesis)
  - often some aspect of physical world is visually simulated, or "synthesized"
  - examples of 2D apps: Adobe Illustrator™, Adobe Freehand™, Corel CorelDRAW™
  - examples of 3D apps: Autodesk's AutoCAD™, Autodesk's (formerly Alias|Wavefront's) Maya™, Autodesk's 3D Studio Max™





Geometry-Based Graphics (2/2)

- Geometry-based graphics applications
  - Store mathematical descriptions, or "**models**," of geometric elements (lines, polygons, polyhedrons, polygonal meshes…) and associated attributes (e.g., color, material properties).
  - Geometric elements are primitive shapes, **primitives** for short.
  - Images are created via sampling of geometry for viewing, but not stored as part of model.
  - Users cannot usually work directly with individual pixels in geometry-based programs; as user manipulates geometric elements, program resamples and redisplays elements
- Increasingly rendering combines geometry- and sample-based graphics, both as performance hack and to increase quality of final product
  - geometric characters on painted or scanned scene images

What is Geometric Modeling?

- What is a model?
- Captures salient features (data, behavior) of thing/phenomenon being modeled
  - data includes geometry, appearance, attributes…
  - note similarity to OOP ideas
- Modeling allows us to cope with complexity
- Our focus: modeling and viewing simple everyday objects
- Consider this:
  - Through 3D computer graphics, first time in human history we have abstract, easily changeable 3D forms.
    - Has revolutionized working process of many fields – science, engineering, industrial design, architecture, commerce, entertainment, etc. Profound implications for visual thinking and visual literacy
    - “Visual truth” is gone in the Photoshop and FX-saturated world (but consider painting and photography…) – seeing no longer is believing…(or shouldn’t be!)

Modeling vs. Rendering

- **Modeling**
  - Create models
  - Apply materials to models
  - Place models around scene
  - Place lights in scene
  - Place the camera

- **Rendering**

  Take “picture” with camera

- Both can be done with commercial software:

  Autodesk Maya™ ,3D Studio Max™, Blender™, etc.

Spot Light

Ambient Light

Point Light

Directional Light

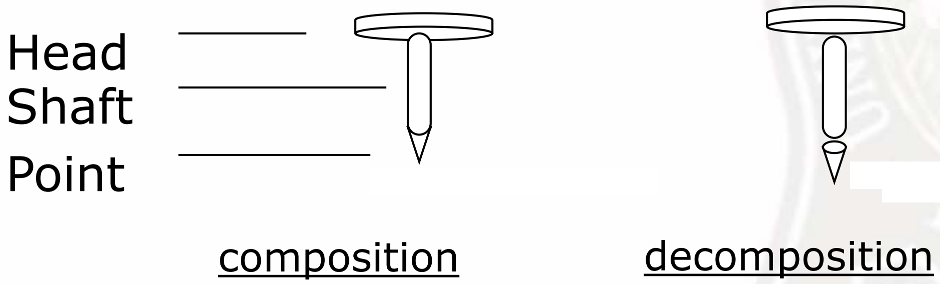
CS128 lighting assignment by Patrick Doran, Spring 2009

DEMO: PHOTOREALISTIC MODEL/RENDER OF HUMAN EYE

http://youtu.be/TAZIvyAJfeM

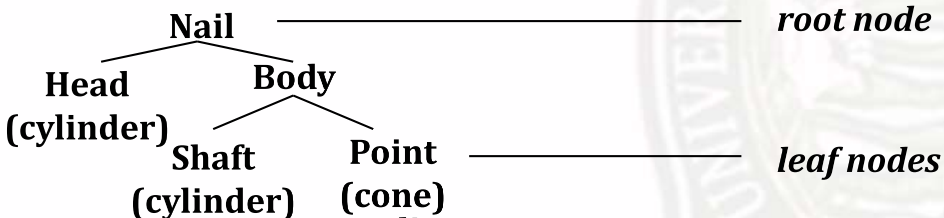
Decomposition of a Geometric Model

- Divide and Conquer
- Hierarchy of geometrical components
- Reduction to primitives (e.g., spheres, cubes, etc.)
- Simple vs. not-so-simple elements (nail vs. screw)



Hierarchical (Tree) Diagram of Nail

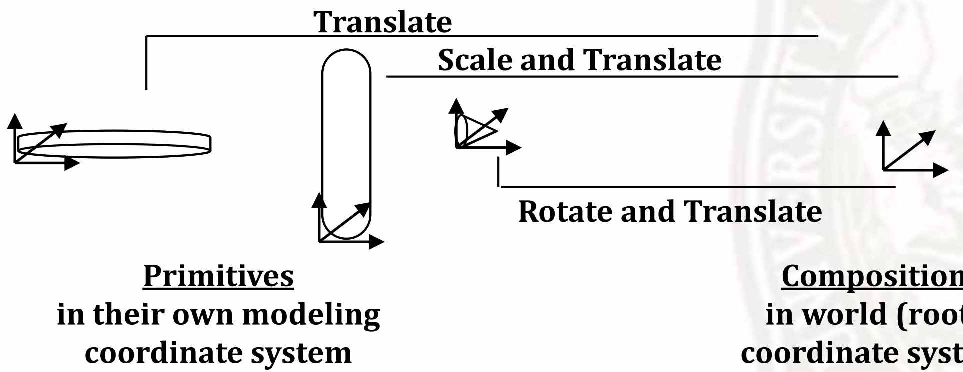
- Object to be modeled is (visually) analyzed, and then decomposed into collections of primitive shapes.
- Tree diagram provides visual method of expressing “composed of” relationships of model



tree diagram

- Such diagrams are part of 3D program interfaces (e.g., 3D Studio MAX, Maya)
- As a data structure to be rendered, it is called a **scenegraph**

Composition of a Geometric Model



- Primitives created in decomposition process must be assembled to create final object. Done with **affine transformations**, T, R, S (as in above example). Order matters – these are not commutative!

Upcoming Topics

- We manipulated primitive shapes with geometric **transformations** (translation, rotation, scale). These transformations are essential for model organization, process of composing complex objects from simpler components.
- Hierarchical models and geometric transformations are also essential for animation – create and edit **scenegraphs**
- Once object's geometry is established, must be **viewed** on screen: map from 3D geometry to 2D projections for viewing, and from 2D to 3D for 2D input devices (e.g., the mouse or pen/stylus, or touch)
- While mapping from 3D to 2D, object (surface) material properties and lighting effects are used in **rendering** one's constructions. This rendering process is also called **image synthesis**.