CAPITULO I

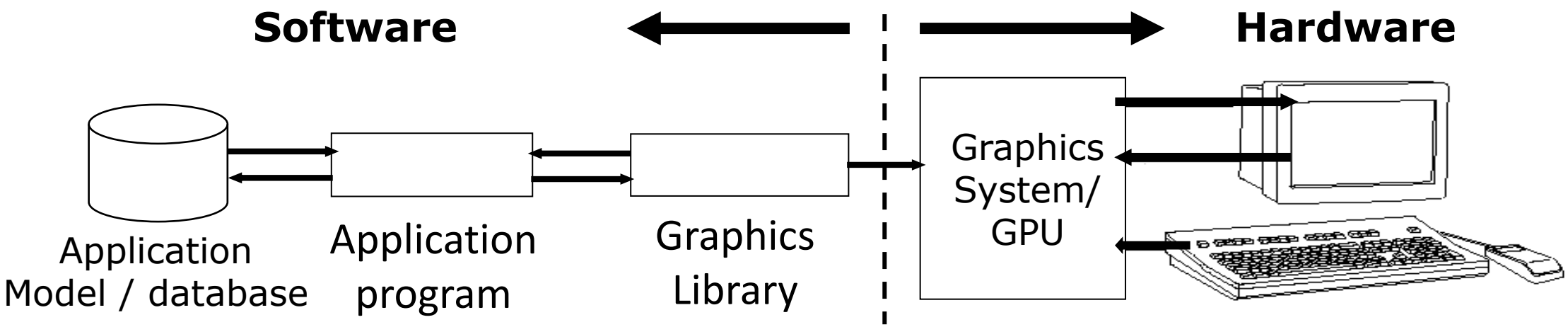
REPRESENTACIÓN VISUAL COMPUTARIZADA

1.2

SISTEMAS GRÁFICOS BÁSICOS

BASIC GRAPHIC SYSTEMS

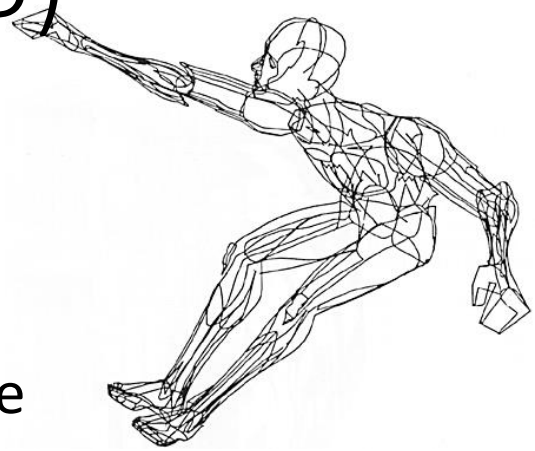
Conceptual Framework for Interactive Graphics



Hardware - 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

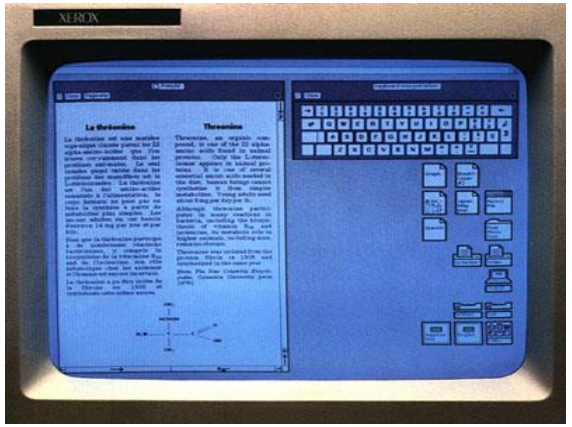
Hardware - Environmental Evolution (2/5)



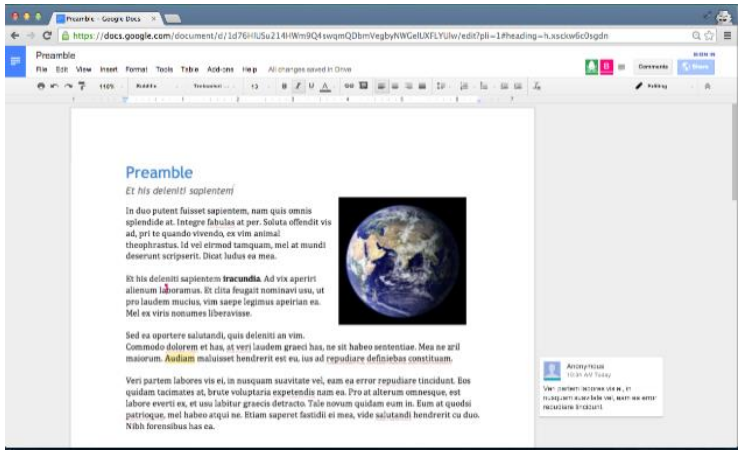
- Vector (Calligraphic, Line Drawing)
- Displays (1963 – 1980s)
- Display: line drawings and stroke text; 2D and 3D transformation hardware
- Object and command specification: command-line typing, function keys, menus
- Control over appearance: pseudo-WYSIWYG
- Application control: single or multitasked, 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

Hardware - 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”)

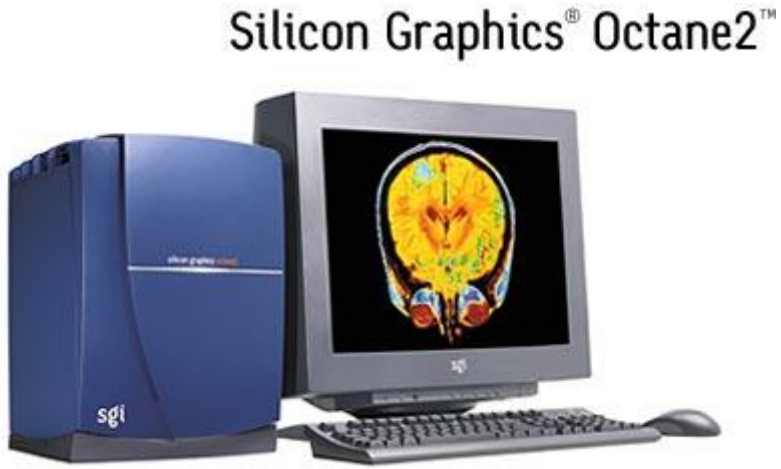


Above, a classic WIMP interface. The technology, at its core, remains largely the same today. Below, a modern WIMP interface.



Hardware - 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

Hardware - Environmental Evolution (5/5)

- High-end PCs with hot graphics cards (NVIDIA GeForce™, AMD Vega™) 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.

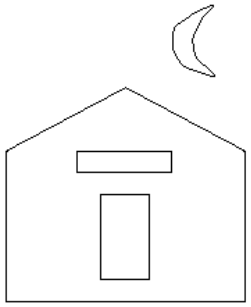
- Also build GPU-clusters as number crunchers, e.g., protein folding, weather prediction, machine learning
  - Now accessible to consumers via technologies like NVIDIA's **SLI (Scalable Link Interface) bridge**

SLI Image from http://www.overclock.net/t/1606562/official-nvidia-titan-x-pascal-owners-thread/3410

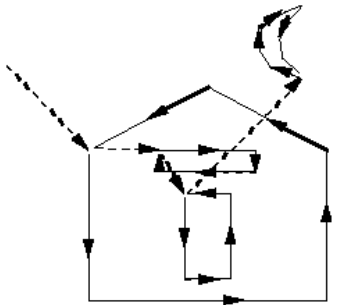
Graphics Display Hardware

Vector (calligraphic, stroke, random-scan)

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



Ideal Drawing



Vector Drawing

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

Outline

Filled

Graphics Display Hardware

- Many form factors
  - Smartphones/laptops/desktops/tablets
  - Smart watches
  - Head-mounted displays (HMDs)
  - Augmented Reality
  - Virtual Reality
    - AR vs VR: Different experiences!

Apple iPhone

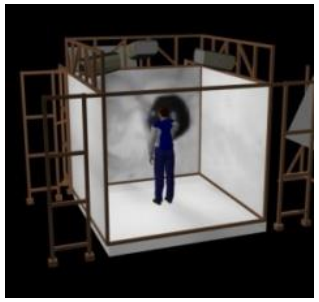
Android Phones

Tablets

Microsoft's first Surface

Apple Watch

Android Wear



Brown's old Cave

Microsoft Hololens

Vive Focus

Oculus Rift

Google Cardboard

Digression: Augmented Reality

- Easily accessible AR through smartphones

- Advanced AR

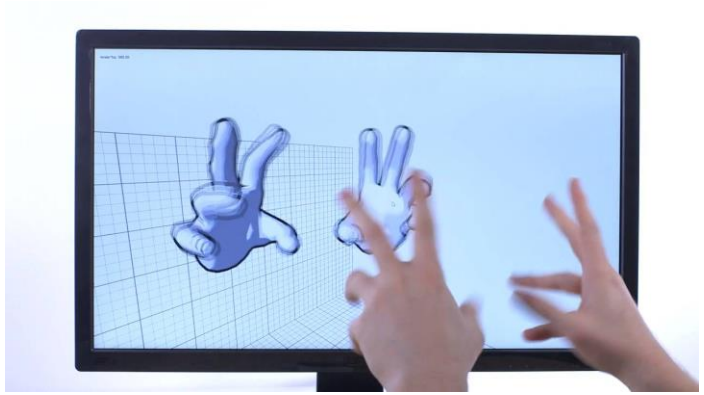
https://youtu.be/80IIEnSNwQc

Graphics Input Hardware

- Input Devices
  - Mouse, tablet & stylus, multi-touch, force feedback, and other game controllers (e.g., JoyCon for Nintendo Switch), scanner, digital camera (images, computer vision), etc.
  - Body as interaction device
    - https://youtu.be/7SEScSsG2Hg

Xbox Kinect:

Ultraleap:



Hardware – Graphics Subsystems / GPU

- Hardware revolution
  - **Moore's Law:** every 12-18 months, computer power improves by factor of 2 in price / performance as size shrinks
  - Newest CPUs are 64-bit with 4, 8, 16, even up to 56 cores
  - Intel Coffee Lake – consumer processor with up to 8 cores, 16 threads, and a fully featured graphics chip built in to the processor
  - Significant advances in commodity graphics chips every 6 months vs. several years for general purpose CPUs
    - NVIDIA GeForce RTX 3090… 10496 cores, 24GB memory GDDR6X, 1.7 Ghz in a single chip

Hardware – Graphics Subsystems / GPU

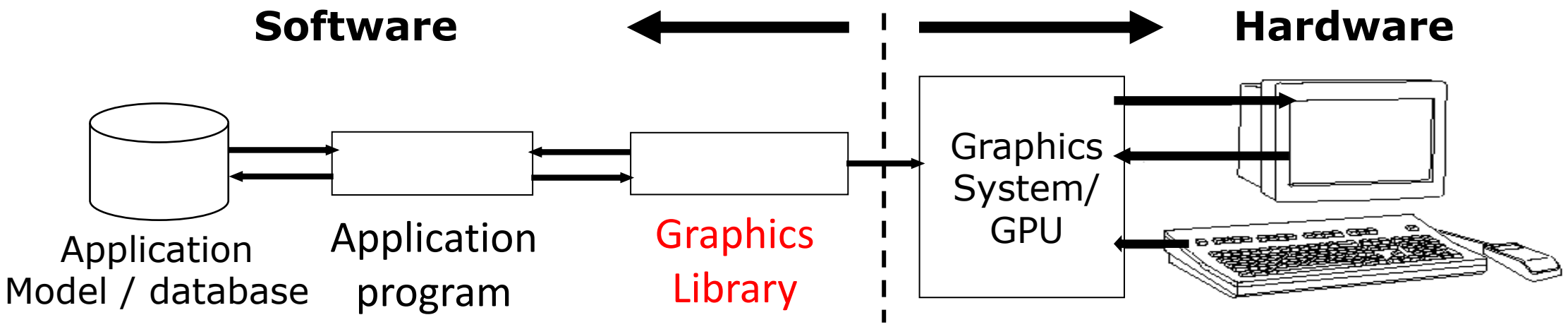
- Graphic subsystems
  - Offloads graphics processing from CPU to chip designed for doing graphics operations quickly
  - NVIDIA GeForce™, AMD Vega™, and Intel HD and Iris Pro Graphics
  - GPUs originally designed to handle special-purpose graphics computations
  - Increasingly, GPUs used to parallelize other types of computation (known as **GPGPU**, or General-Purpose Computing on the Graphics Processing Unit)

Which is the lastest GPU model?

Software

- 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: calculating value of one pixel is often independent of other pixels
  - Distributed and Cloud computing
    - Send operations to the cloud, get back results, don’t care how
    - Rendering even available as internet service!

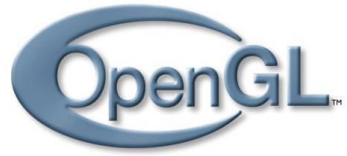
Conceptual Framework for Interactive Graphics



- 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

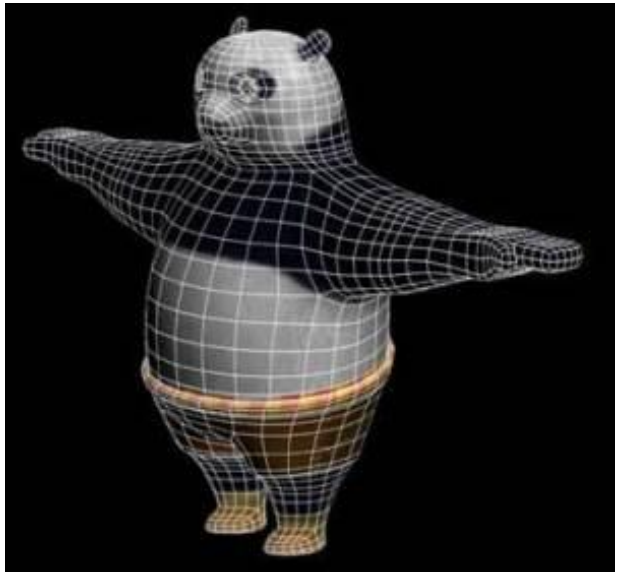
Graphics Library

- Examples: OpenGL™, DirectX™, Windows Presentation Foundation™ (WPF) accessed via XAML, 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 (Directed Acyclic Graph). It is a display-centered extract of the Application Model



Application Distinctions: Two Basic Paradigms

Sample-based graphics *vs* Geometry-based graphics

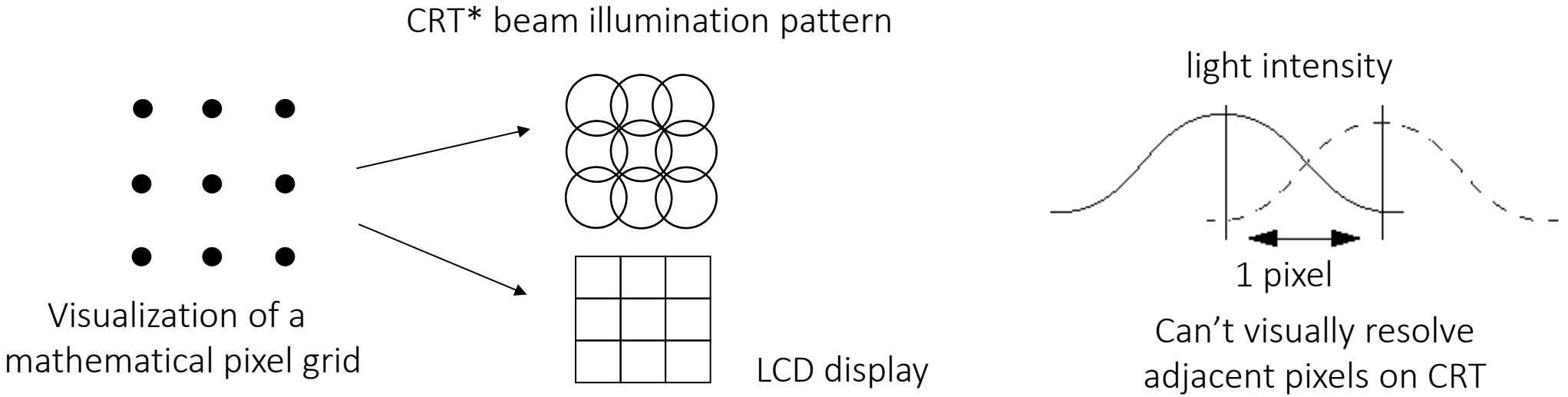


Sample-based Graphics (1/3)

- **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™

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



* Cathode Ray Tube, like those really old TVs

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
  - 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, Computational Photography

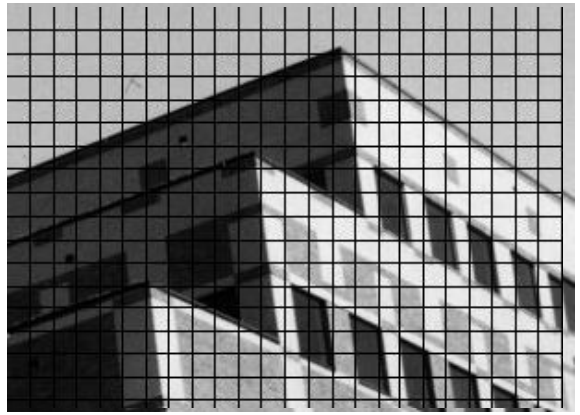
Sampling an Image

- 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; depth sensors can be used in conjunction
  - 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.

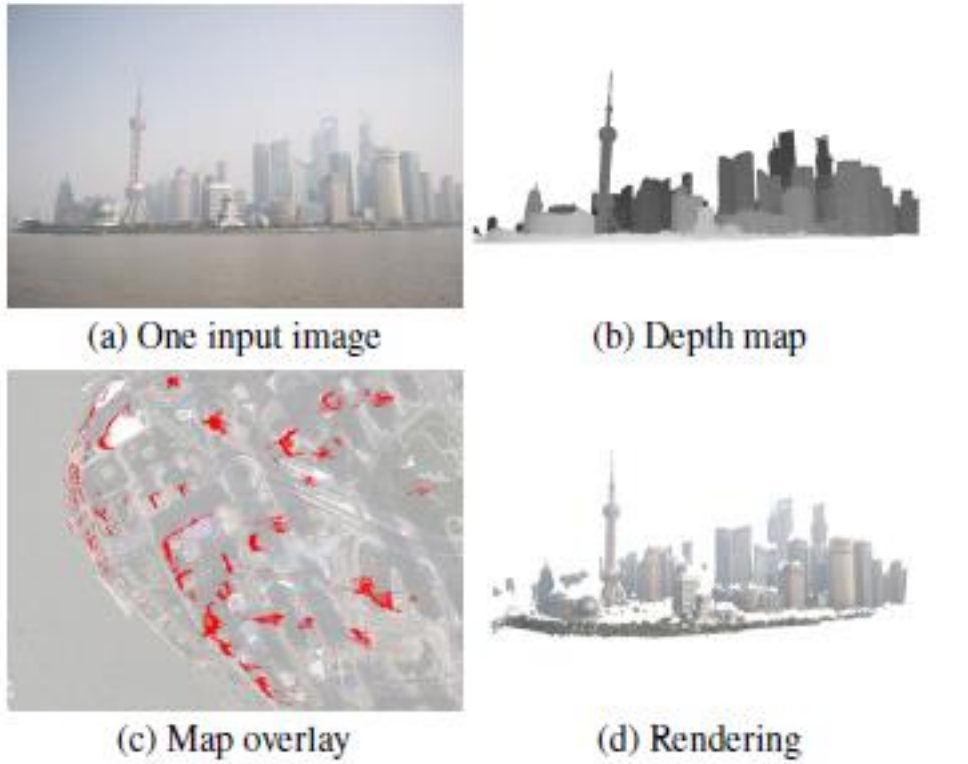
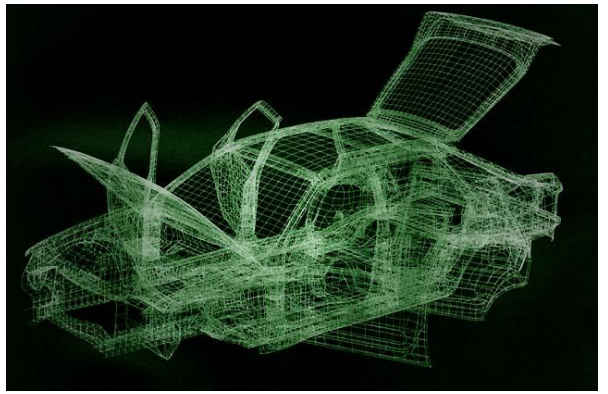


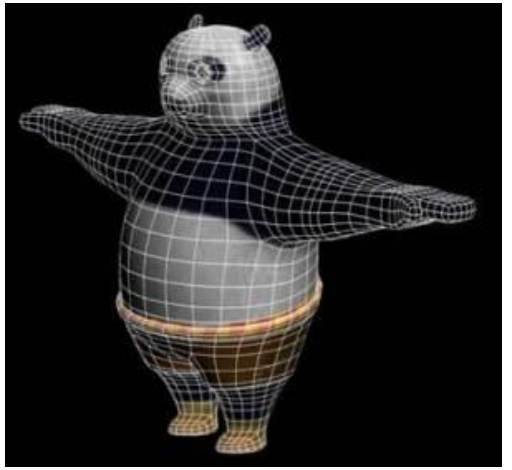
Figure 15: *Results on a challenging unstructured light field, obtained by hand-held capture (a) from a floating boat. (b) A resulting depth map. (c) Overlay of our reconstruction on a satellite image ©2013 DigitalGlobe, Google. (d) Rendering from a novel viewpoint.*

"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™ and 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 called 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
  - CG animated characters (geometry) on painted or filmed scene images (samples)

What is Geometric Modeling?

- What is a model?
- Captures salient features (data, behavior) of object/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, we have abstract, easily changeable 3D forms, for the first time in human history
    - 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.

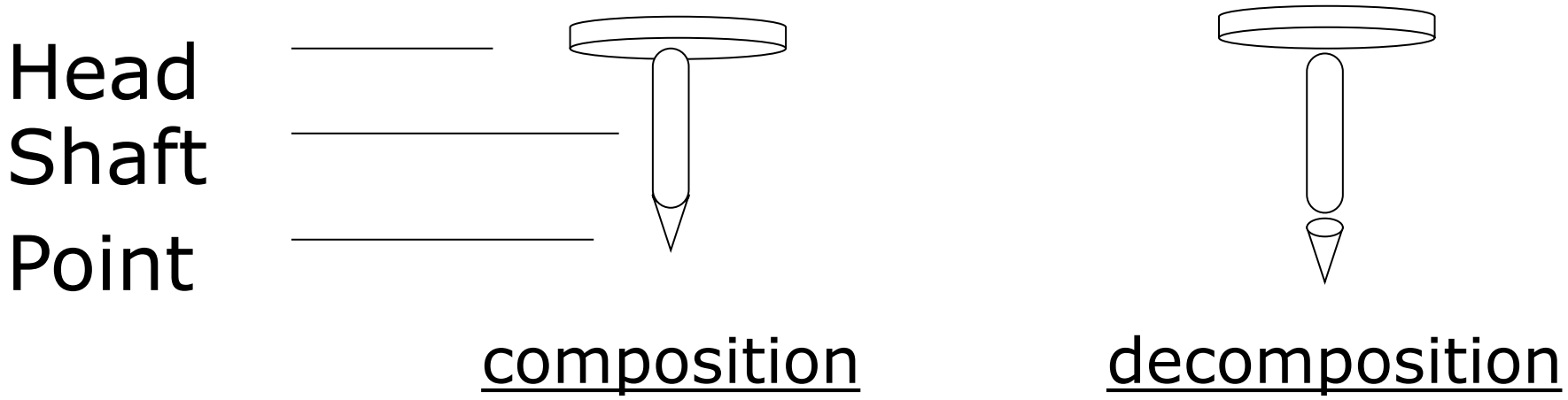
Demo: photorealistic model/render of A humaN

https://youtu.be/S3F1vZYpH8c

Andy Serkis MoCap

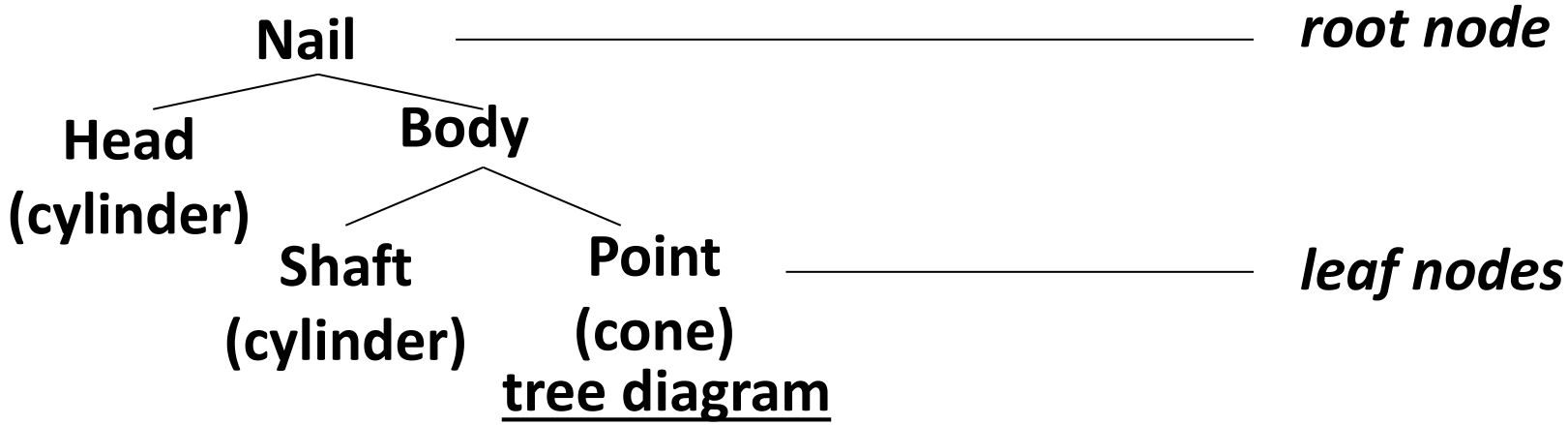
Decomposition of a Geometric Model

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



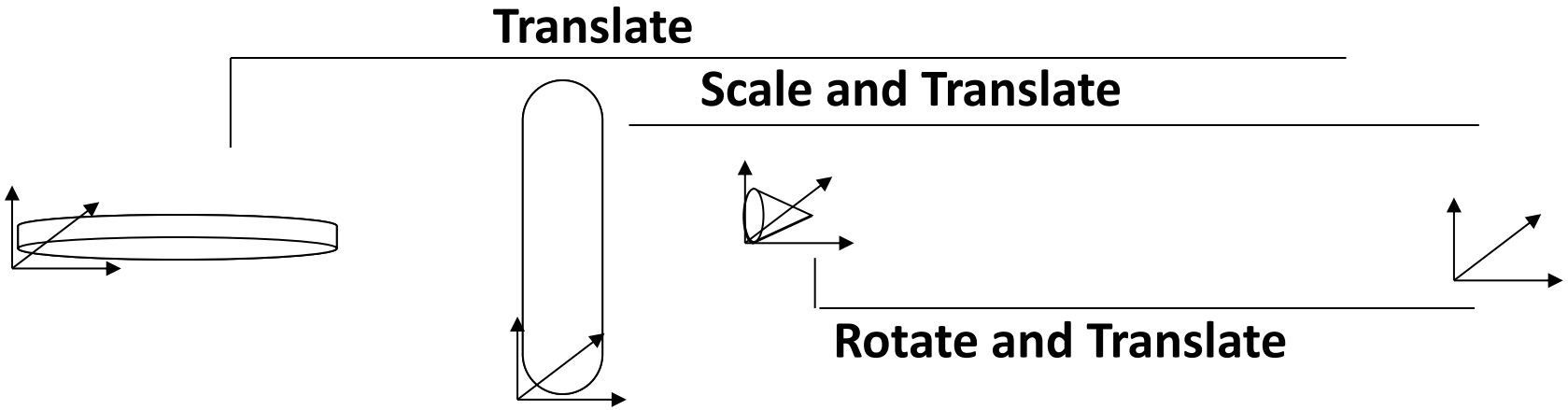
Hierarchical (Tree) Diagram of Nail

- 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



- 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
in their own modeling coordinate system

Composition
in world (root) coordinate system

- 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!