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

Computer Graphics

Introduction to Computer Graphics

Andreas Aristidou
andarist@ucy.ac.cy
http://www.andreasaristidou.com

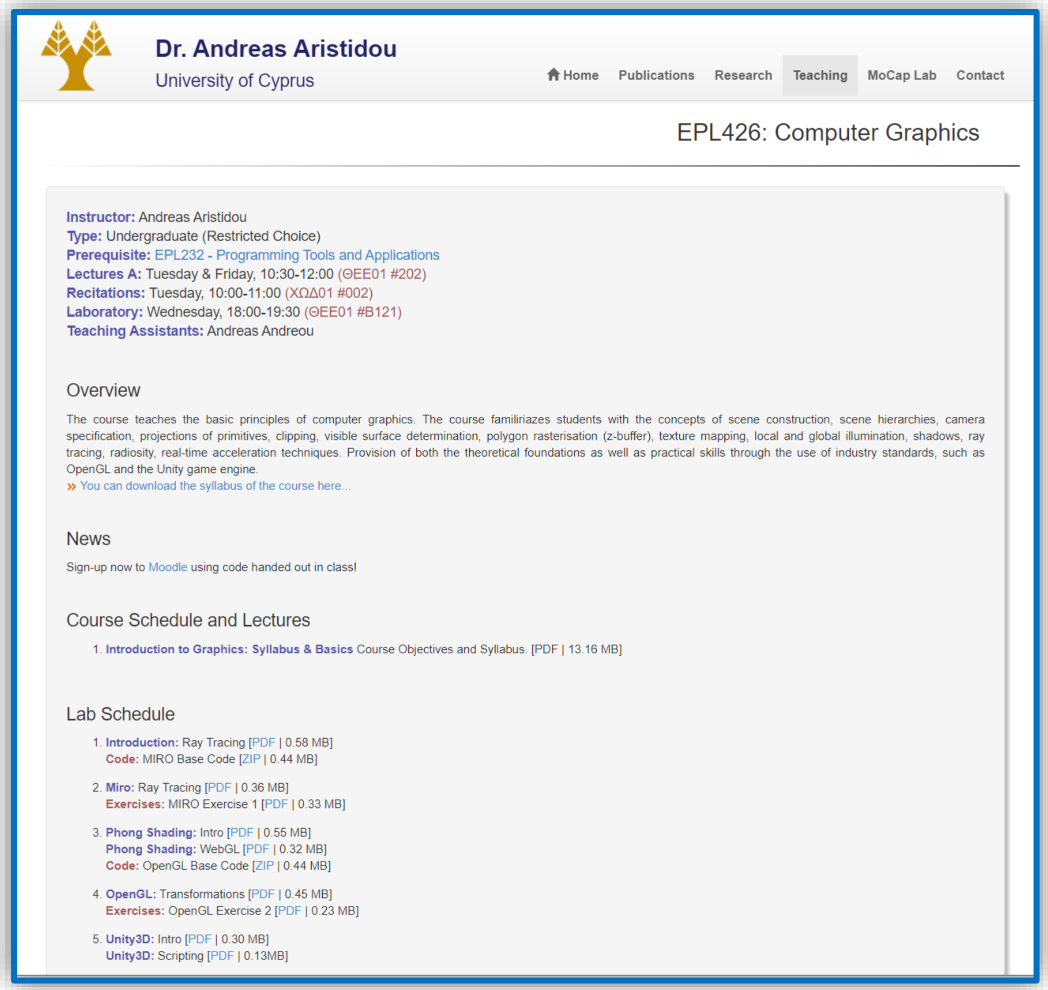
Hi!

- **Andreas Aristidou**

Assistant Professor
Office: FST01, Room B113
Office hours: Tuesday 09:00– 10:30
Friday 09:00 – 10:30
Only after appointment
email: andarist@ucy.ac.cy

- **Andreas Andreou**

Teaching Assistant
Office hours: Only after appointment
email: aandre01@cs.ucy.ac.cy



Dr. Andreas Aristidou
University of Cyprus

Home Publications Research Teaching MoCap Lab Contact

EPL426: Computer Graphics

Instructor: Andreas Aristidou
Type: Undergraduate (Restricted Choice)
Prerequisite: EPL232 - Programming Tools and Applications
Lectures A: Tuesday & Friday, 10:30-12:00 (ΘΕΕ01 #202)
Recitations: Tuesday, 10:00-11:00 (ΧΩΔ01 #002)
Laboratory: Wednesday, 18:00-19:30 (ΘΕΕ01 #B121)
Teaching Assistants: Andreas Andreou

Overview

The course teaches the basic principles of computer graphics. The course familiriazes students with the concepts of scene construction, scene hierarchies, camera specification, projections of primitives, clipping, visible surface determination, polygon rasterisation (z-buffer), texture mapping, local and global illumination, shadows, ray tracing, radiosity, real-time acceleration techniques. Provision of both the theoretical foundations as well as practical skills through the use of industry standards, such as OpenGL and the Unity game engine.
» You can download the syllabus of the course here...

News

Sign-up now to Moodle using code handed out in class!

Course Schedule and Lectures

1. **Introduction to Graphics: Syllabus & Basics** Course Objectives and Syllabus. [PDF | 13.16 MB]

Lab Schedule

1. **Introduction:** Ray Tracing [PDF | 0.58 MB]
   Code: MIRO Base Code [ZIP | 0.44 MB]
2. **Miro:** Ray Tracing [PDF | 0.36 MB]
   Exercises: MIRO Exercise 1 [PDF | 0.33 MB]
3. **Phong Shading:** Intro [PDF | 0.55 MB]
   Phong Shading: WebGL [PDF | 0.32 MB]
   Code: OpenGL Base Code [ZIP | 0.44 MB]
4. **OpenGL:** Transformations [PDF | 0.45 MB]
   Exercises: OpenGL Exercise 2 [PDF | 0.23 MB]
5. **Unity3D:** Intro [PDF | 0.30 MB]
   Unity3D: Scripting [PDF | 0.13MB]

https://www.cs.ucy.ac.cy/courses/EPL426/

Computer Graphics: *Introduction*

- **Computer Graphics**
  - What's It All About?
  - Application Areas
  - Interactive Computer Graphics

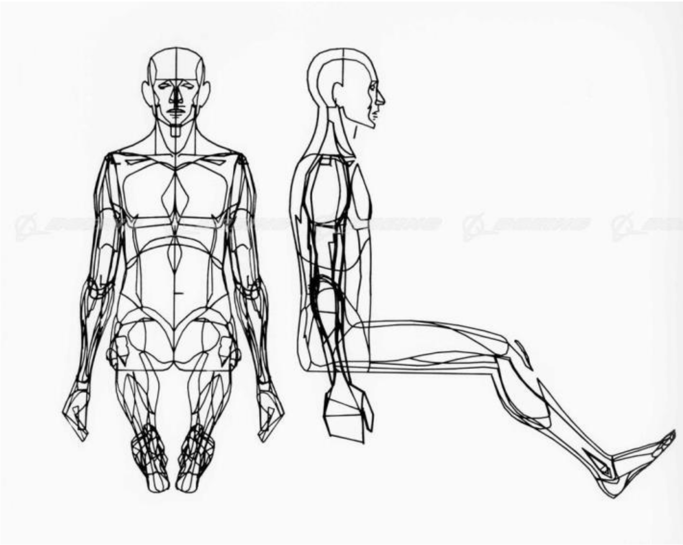
- **How do graphics work**
  - Modeling
  - Animation
  - Rendering

- **Course Outline**
  - Lectures, Exams, Evaluation
  - Books
  - Miscellanea

Computer Graphics: *Introduction*

- **com•put•er graph•ics** /kəmˈpyo͞odər ˈgrafiks/ *n.* The use of computers to synthesize and manipulate **visual information**.

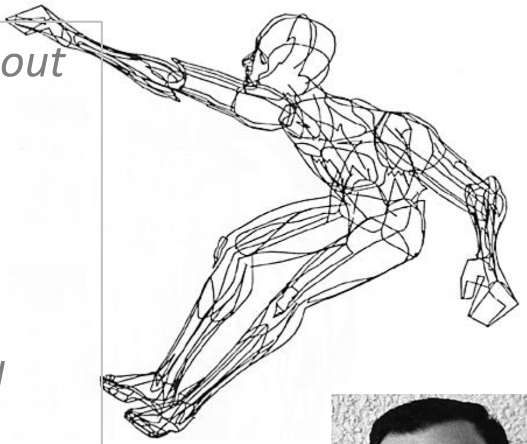
- The creation, storage and manipulation of models and images. Such models come from diverse and expanding set of fields including physical, biological, mathematical, artistic, and conceptual/abstract structures.

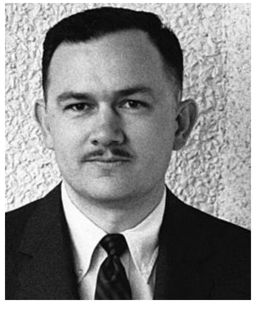


Perhaps the best way to define computer graphics is to find out what it is not. It is not a machine. It is not a computer, nor a group of computer programs. It is not the know-how of a graphic designer, a programmer, a writer, a motion picture specialist, or a reproduction specialist.

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

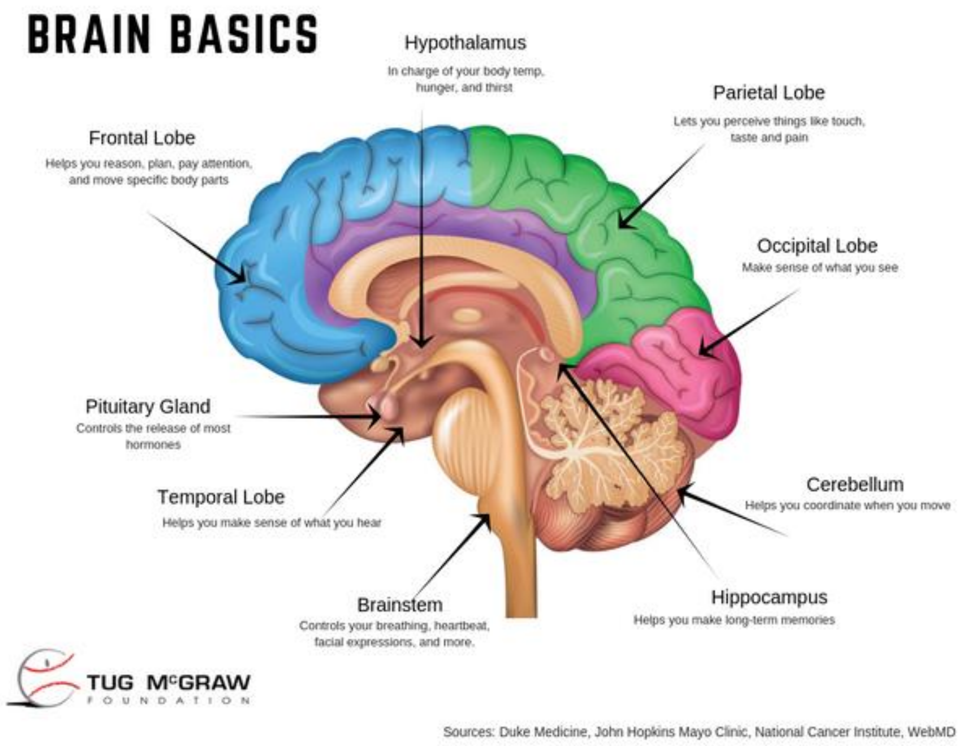
Computer Graphics, by William A. Fetter, 1966





Computer Graphics: *Why Visual Information?*

- About 30% of brain dedicated to visual processing...



© Petar Milošević

- ...eyes are highest-bandwidth port into the head!

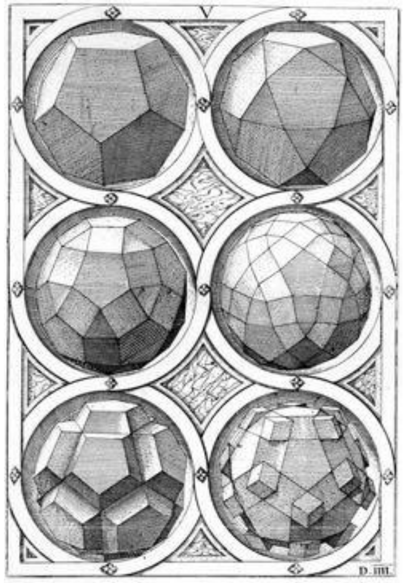
Computer Graphics: *History of visual depiction*

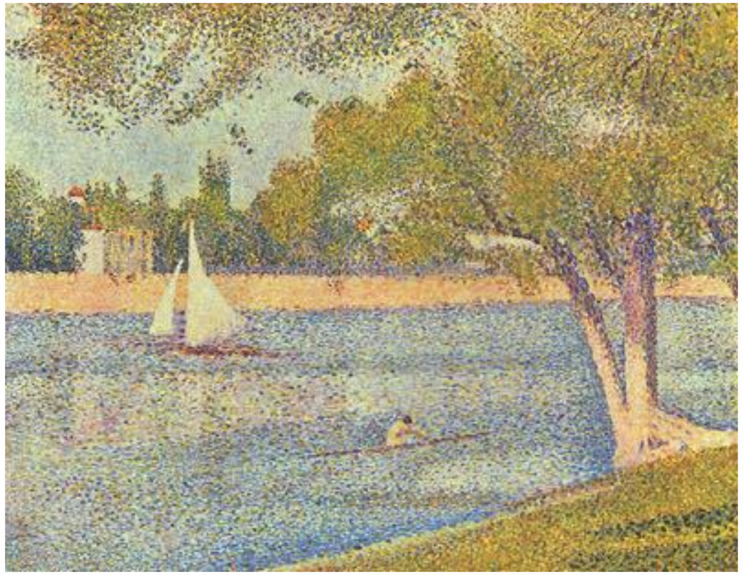
- Humans have always been visual creatures!

Indonesian cave painting (~38,000 BCE)

Computer Graphics: *History of visual depiction*

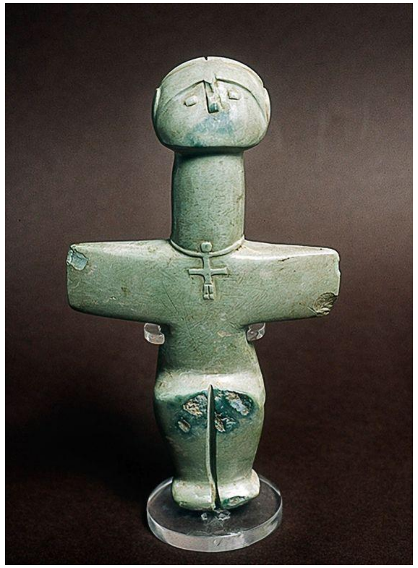
- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...





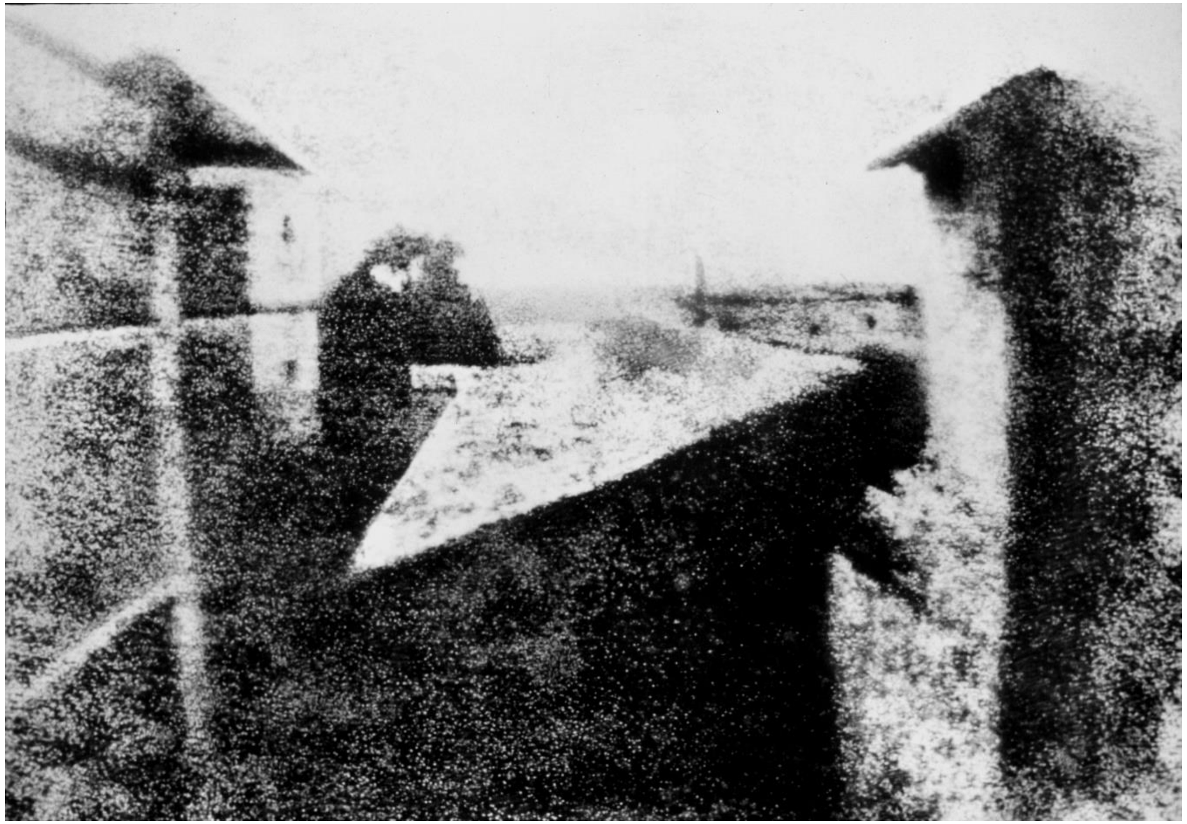
Computer Graphics: *History of visual depiction*

- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...
- Carving / sculpture



Computer Graphics: *History of visual depiction*

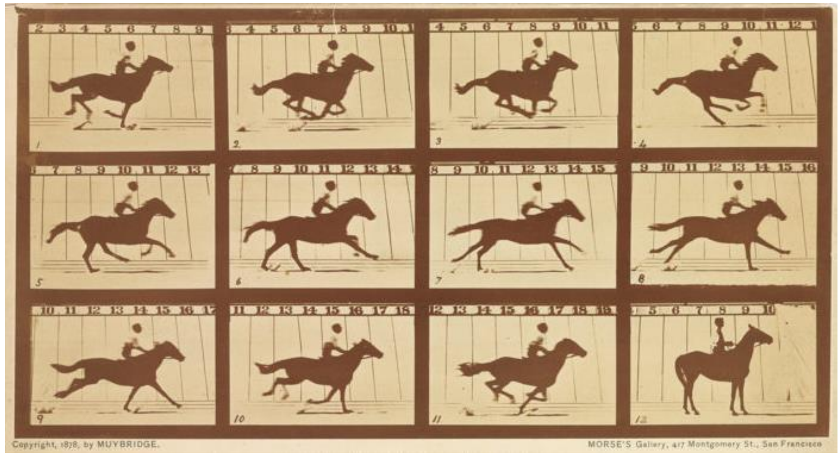
- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...
- Carving / sculpture
- Processing of visual data no longer happening in the head!



Joseph Niépce, “View from the Window at Le Gras” (1826)

Computer Graphics: *History of visual depiction*

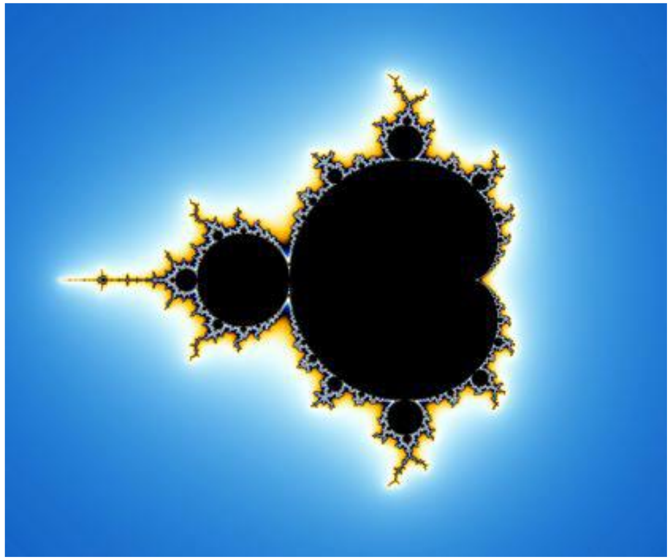
- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...
- Carving / sculpture
- Processing of visual data no longer happening in the head!

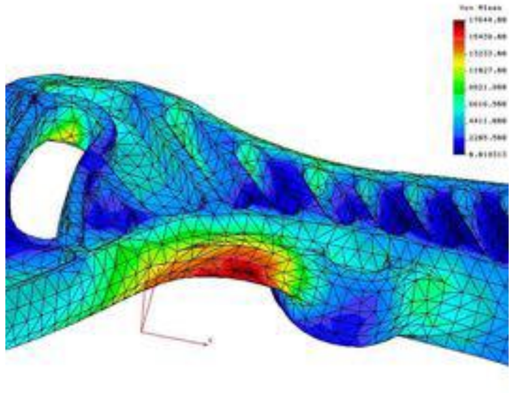





Computer Graphics: *History of visual depiction*

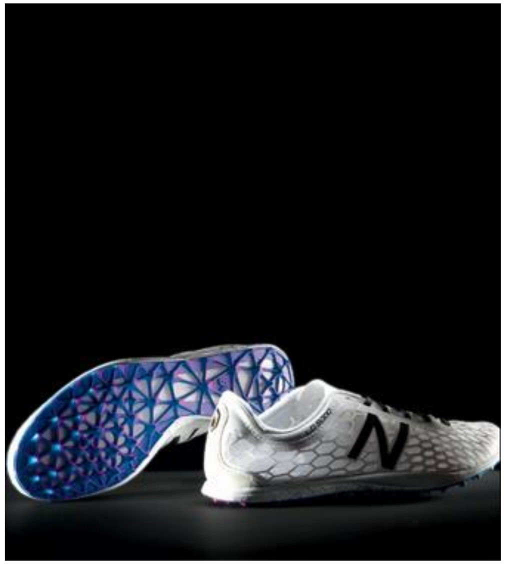
- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...
- Carving / sculpture
- Processing of visual data no longer happening in the head!
- **Digital imagery**
  - **Intersection of visual depiction & computation**

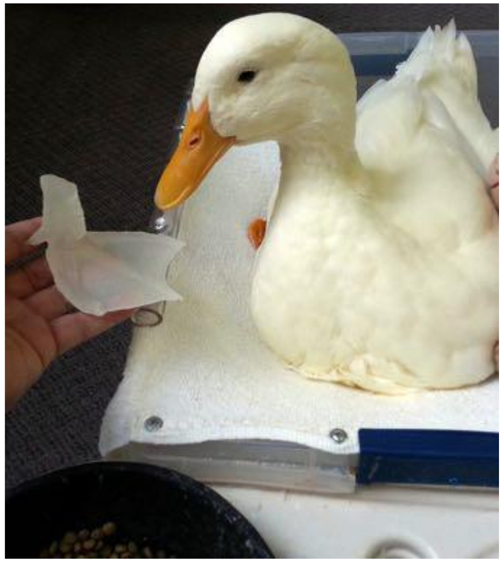




Computer Graphics: *History of visual depiction*

- Humans have always been visual creatures!
- Not purely representational: ideas, feelings, data, ...
- Carving / sculpture
- Processing of visual data no longer happening in the head!
- **Digital imagery**
- **3D fabrication**
  - **Create physical realization of digital shape**





Computer Graphics: *Introduction*

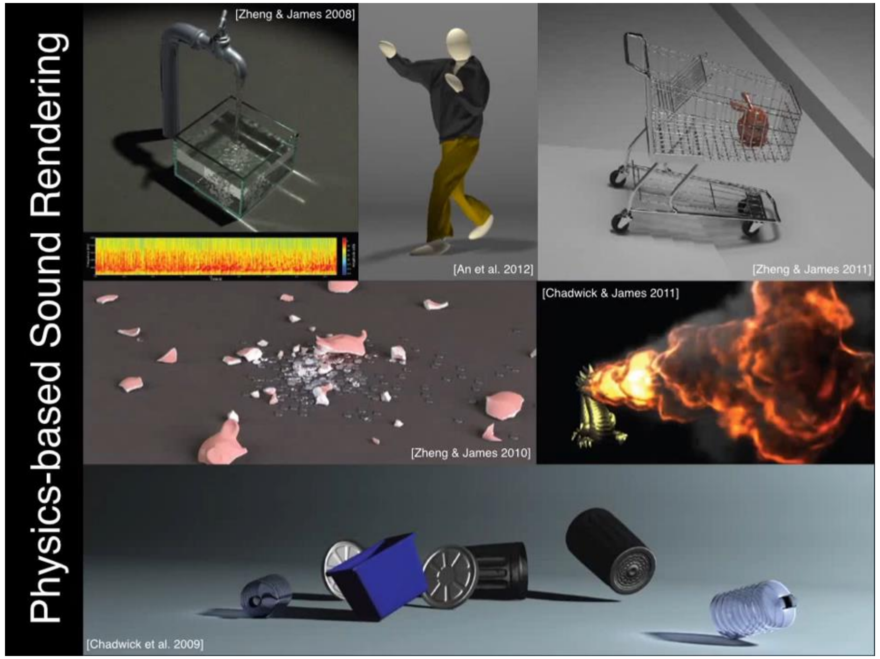
- **com•put•er graph•ics** /kəmˈpyo͞odər ˈgrafiks/ *n.* The use of computers to synthesize and manipulate **visual information**.

3D Computer Graphics: Not image processing!

Computer Graphics: *Introduction*

- **com•put•er graph•ics** /kəmˈpy… . The use of computers to synthesize and manipulate **visual information**.

Why only visual?



© TeslaSuit

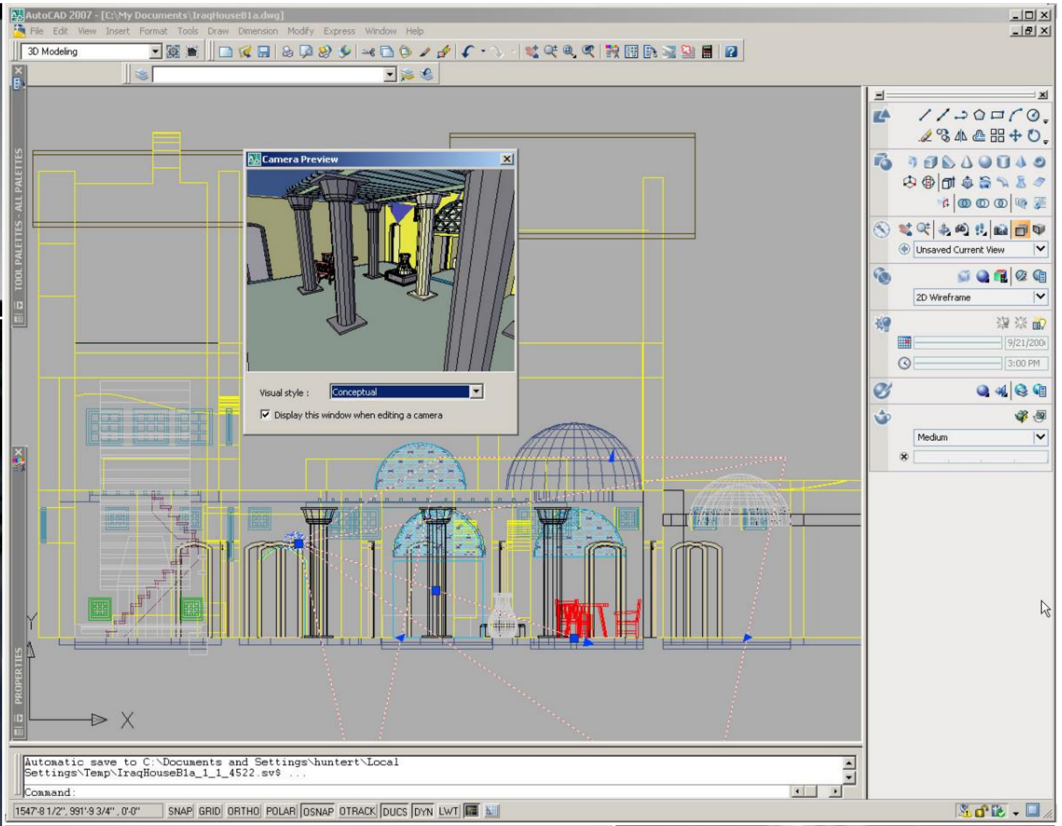
9D Cinemas | (…What about taste? Smell?!)

Computer graphics are everywhere!

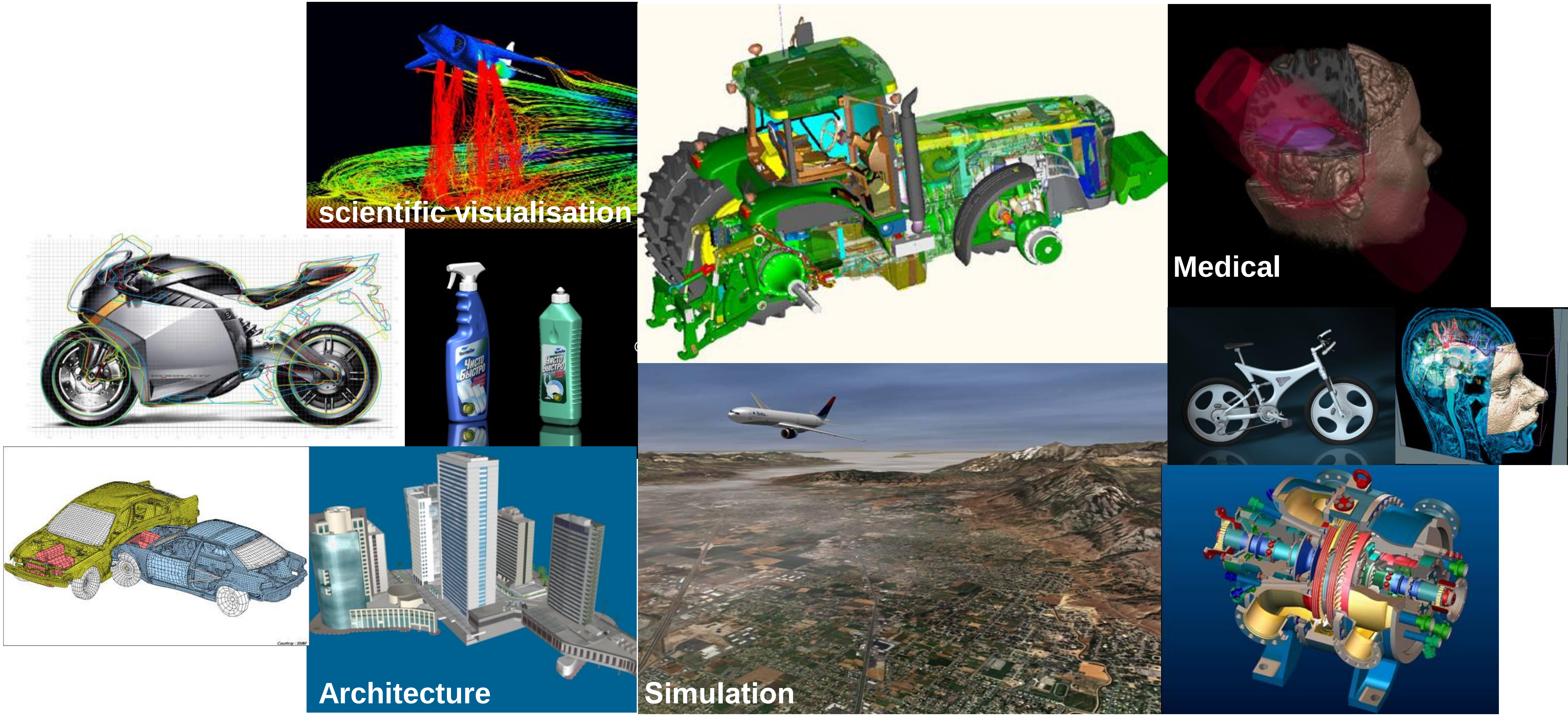
Some Applications of Computer Graphics

- Some of the application areas which make heavy use of computer graphics are:
  - Computer aided design
  - Scientific visualisation
  - Films
  - Games
  - Virtual/Augmented Reality
- **NOTE:** There are lots more and there is huge overlap between these different areas

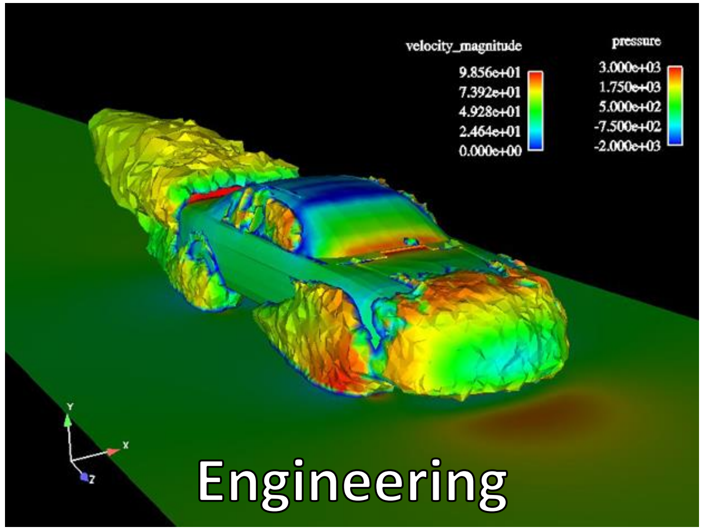
Computer Aided Design

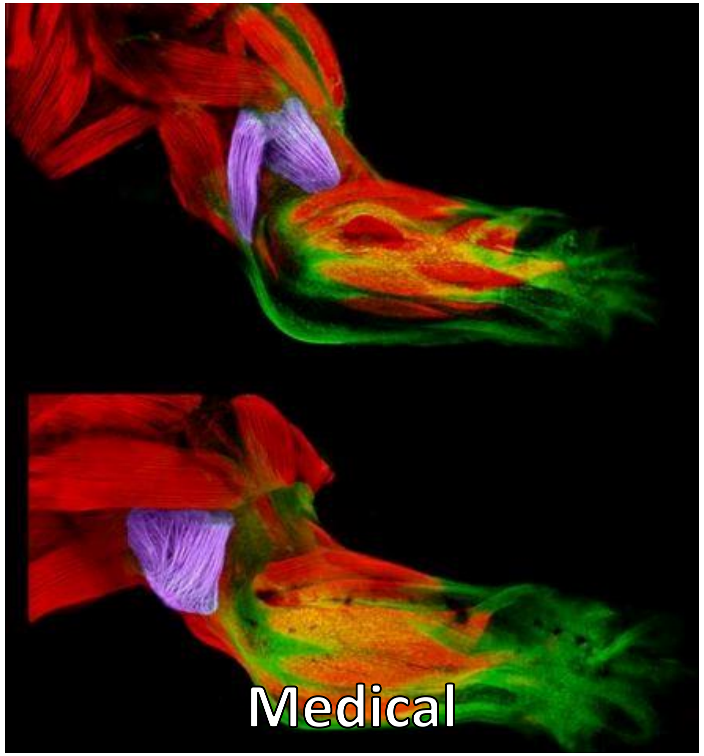


Computer Aided Design

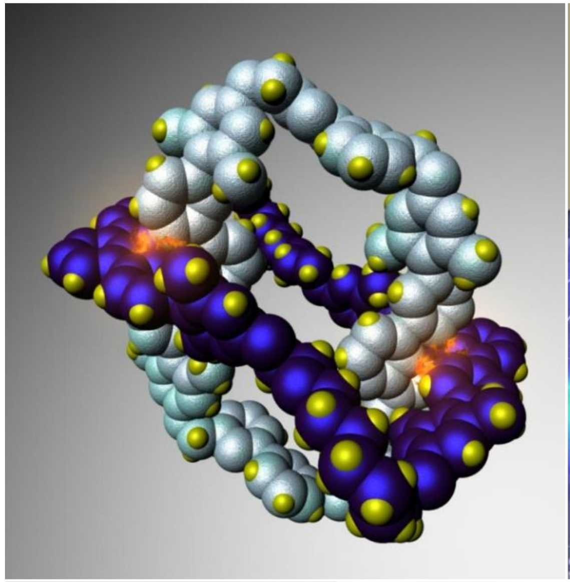


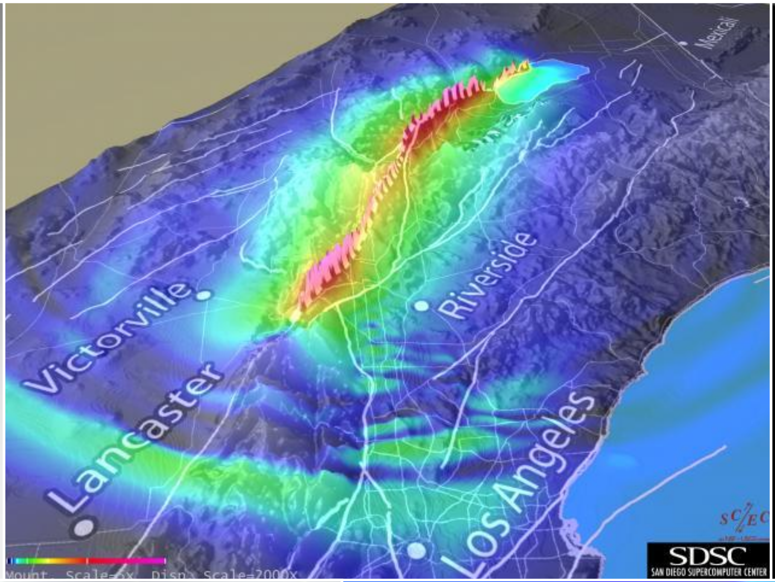
Computer Aided Design

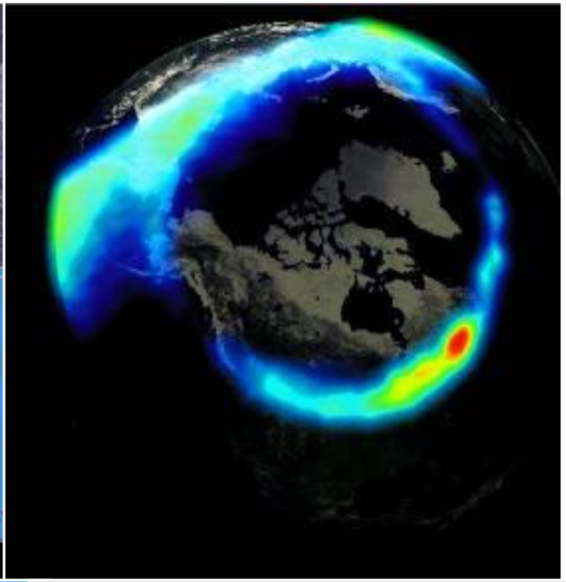


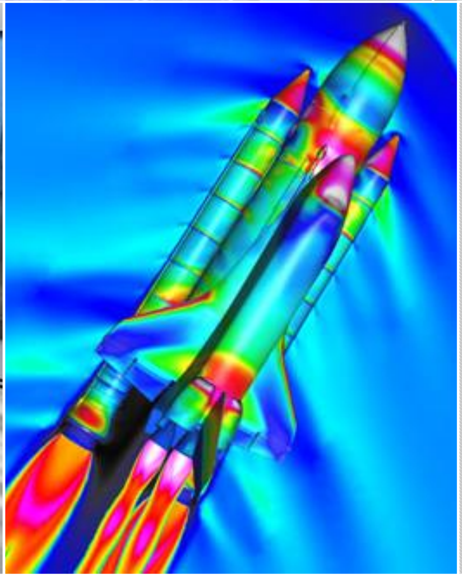


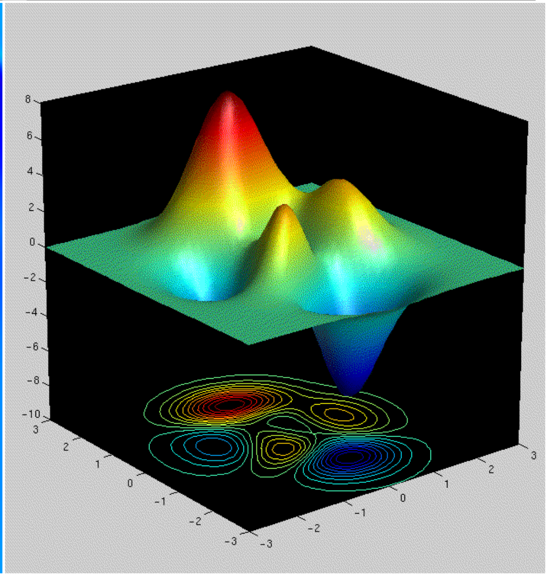
Scientific Visualisation











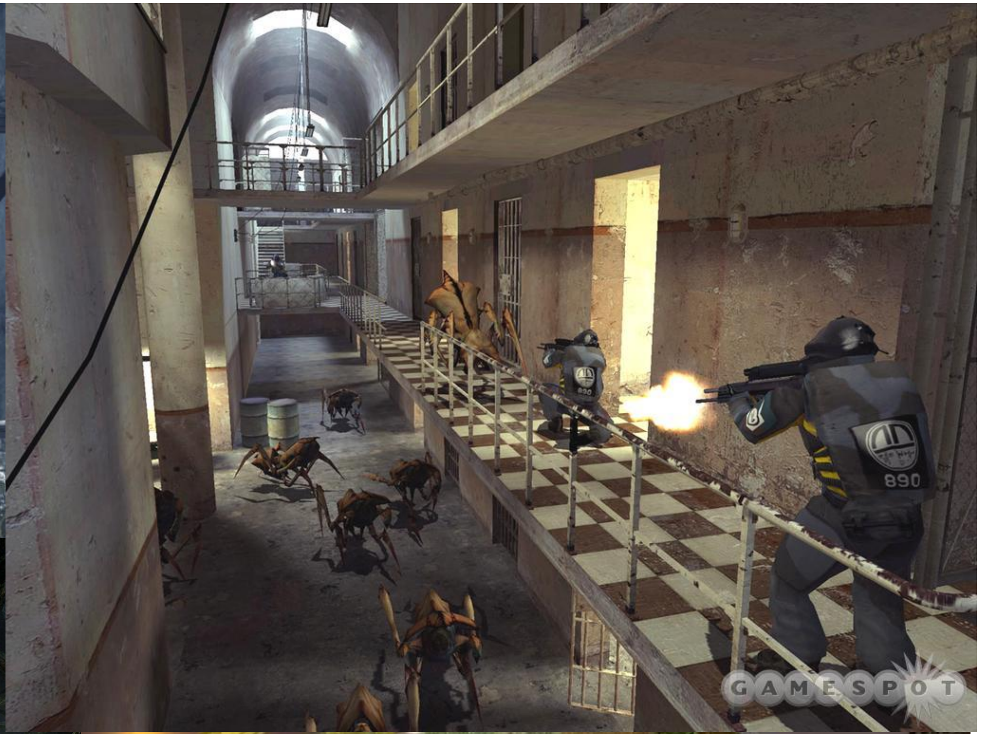
Navigation

Films

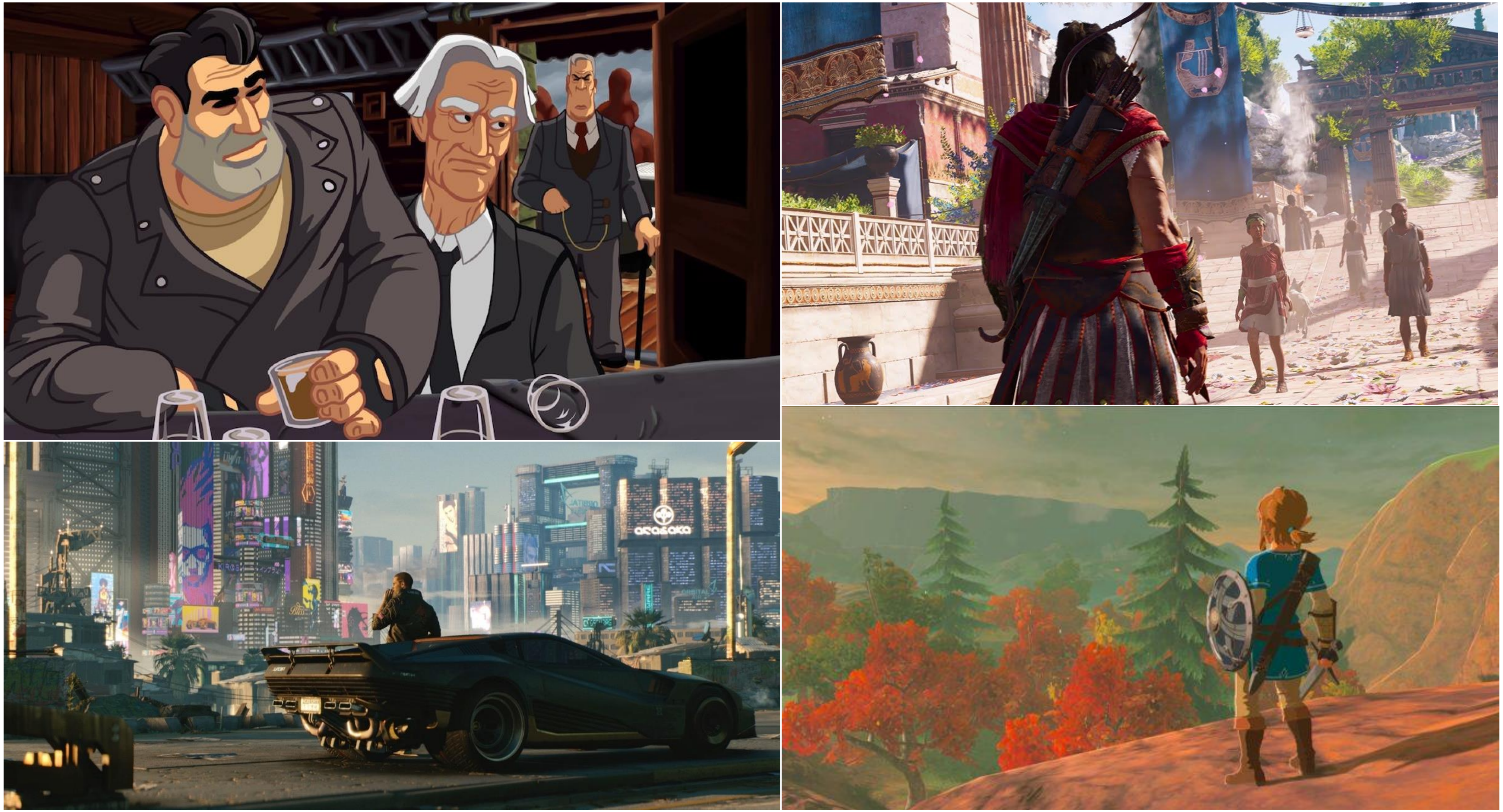
Films

Games

Games

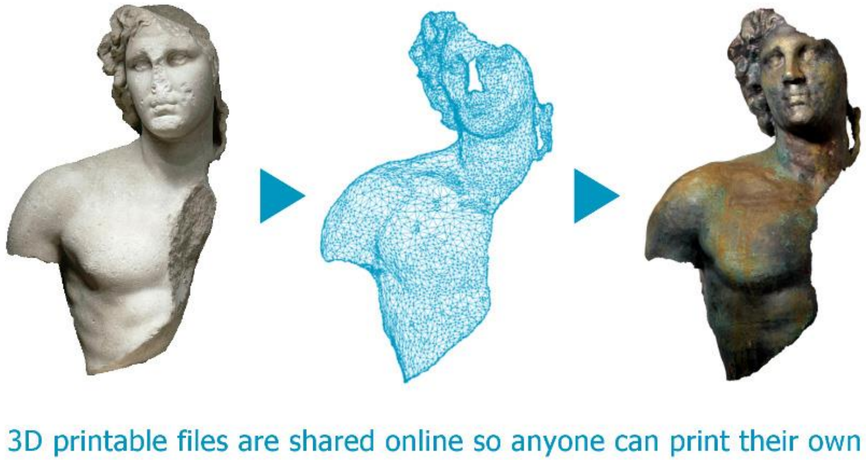


Games



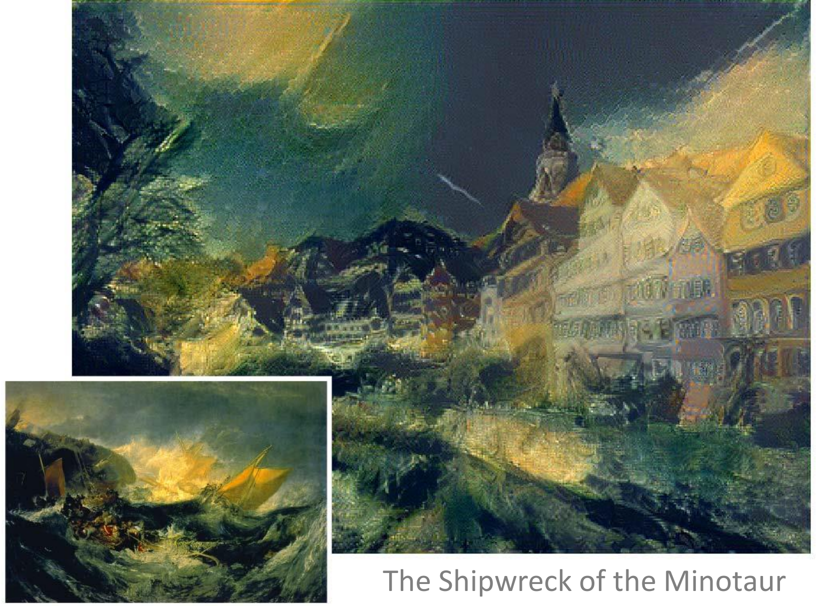
Virtual/Augmented Reality

Cultural Heritage



Entertainment

Neckarfront in Tubingen, Germany
©Andreas Praefcke

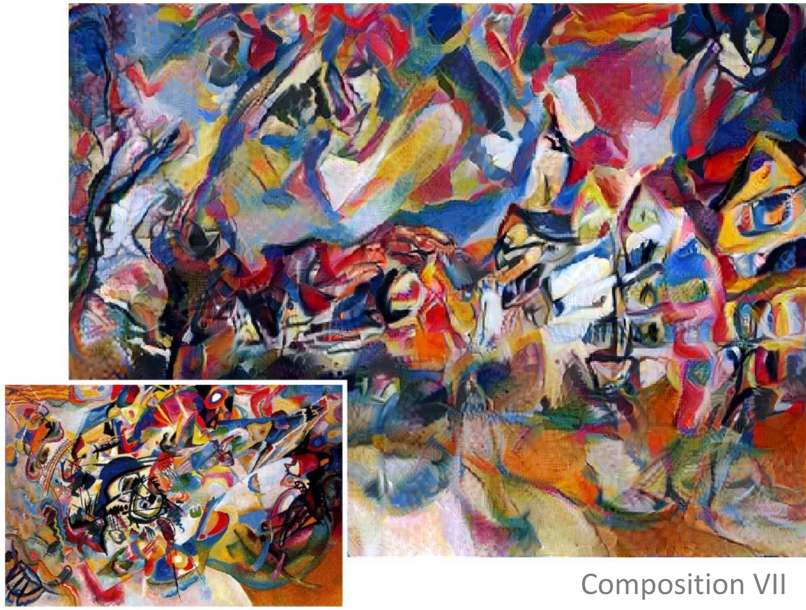


The Shipwreck of the Minotaur
by J.M.W. Turner, 1805

Femme nue assise
by Pablo Picasso, 1910

The Starry Night
by Vincent van Gogh, 1889

Der Schrei
by Edvard Munch, 1893

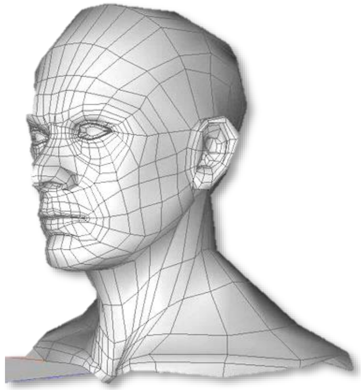


Composition VII
by Wassily Kandinsky, 1913

Gatys et al. 2016. Image Style Transfer Using Convolutional Neural Networks. Proc. CVPR 2016.

Foundations of computer graphics

Visual Computing



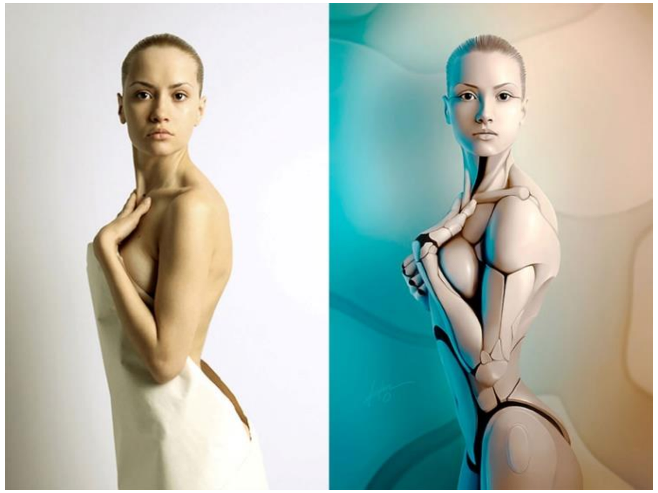
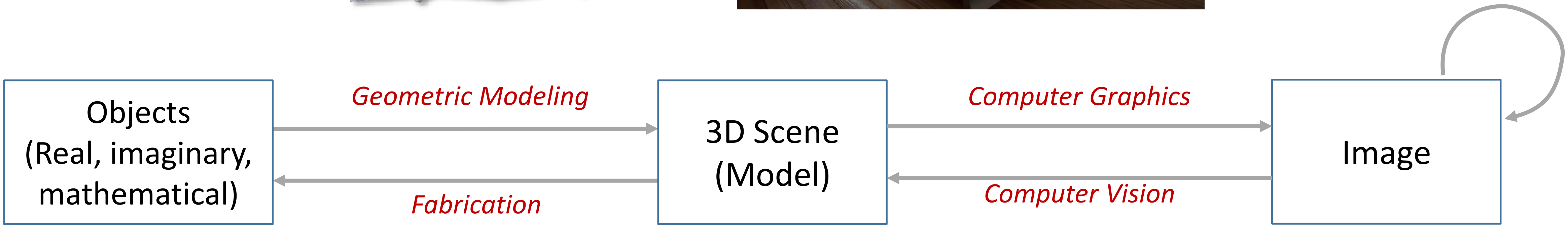
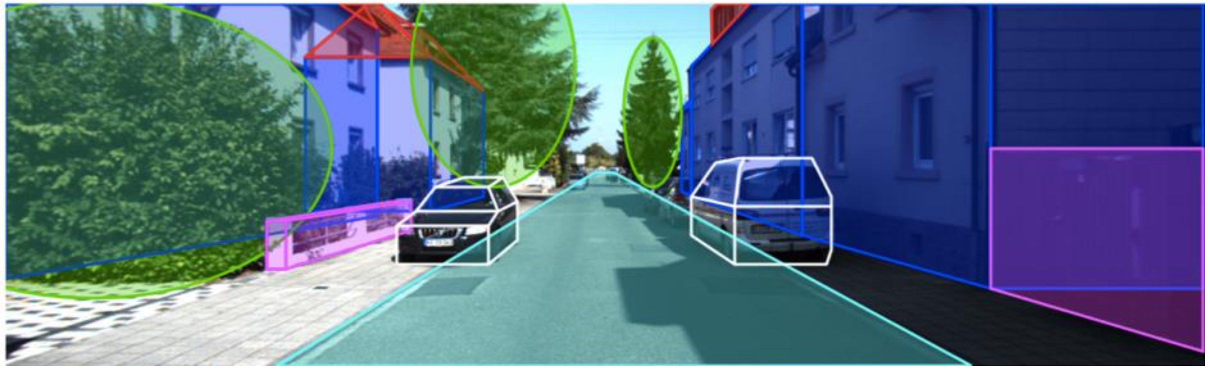


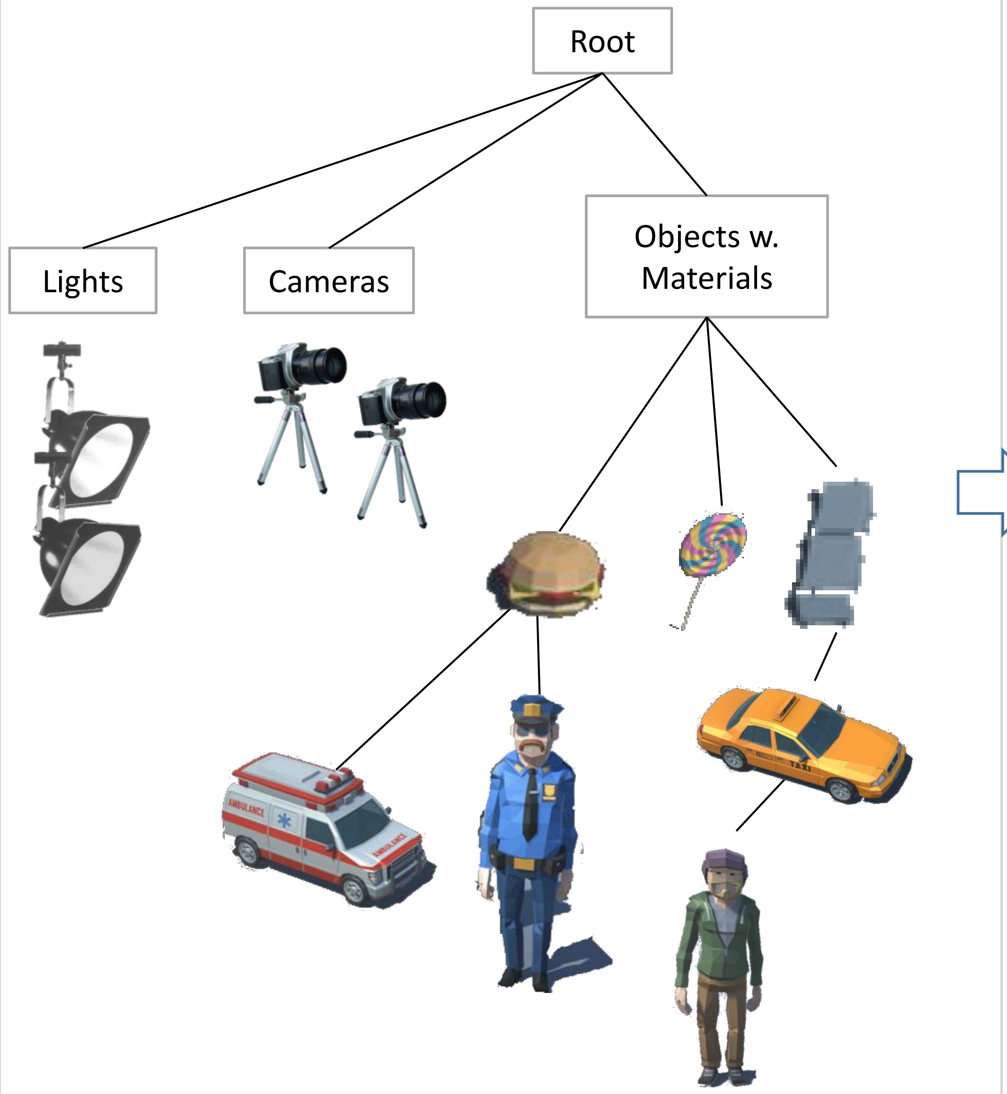
Image Processing





Computer Graphics

Scene Description



Rendering Algorithm

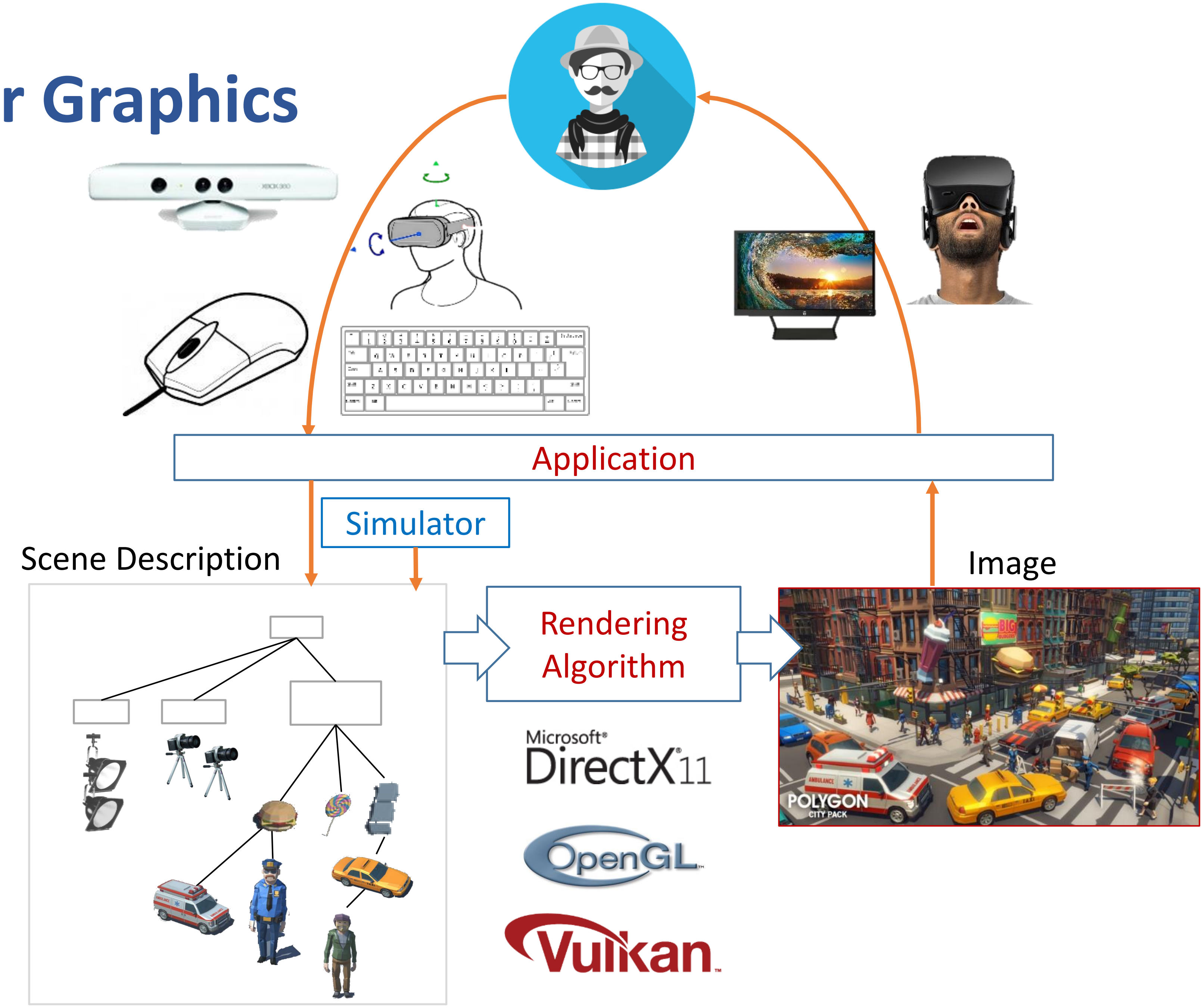
Image

Polygon City Pack for Unity

https://www.assetstore.unity3d.com/en/#!/content/95214

Interactive Computer Graphics

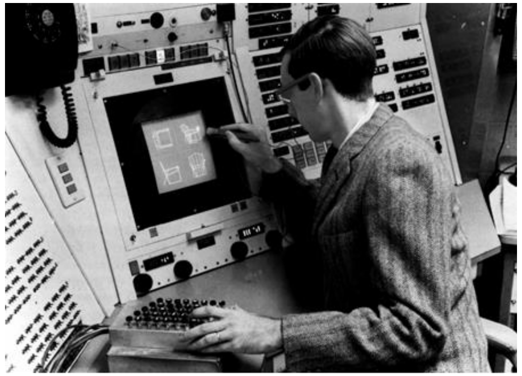
- User controls content, structure, and appearance of objects and their displayed images via rapid visual feedback
- Interactive: 15-120 frames per second depending on application
- Users and/or simulations modify the scene state

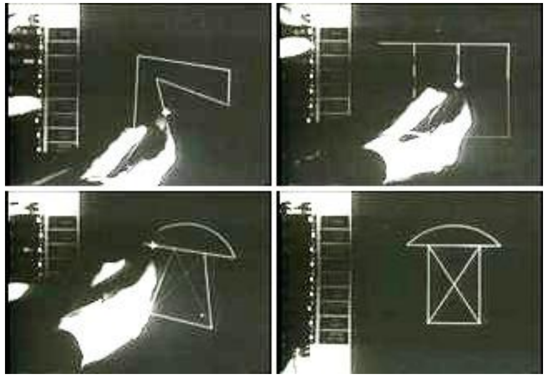


Basic components of an interactive graphics system

Interactive Computer Graphics

- a.k.a. Real-time Computer Graphics, Real-time Rendering.
- 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, stylus, multi-touch, in-air fingers…)
  - **processing** (and storage of the underlying representation/model)
  - **display/output** (e.g., screen, paper-based printer, video recorder…)





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…

Ivan Sutherland (1963)

Ph.D. thesis: *Sketchpad, A Man-Machine Graphical Communication System*
http://youtu.be/546ADZFMBT8

Enabling Modern Computer Graphics

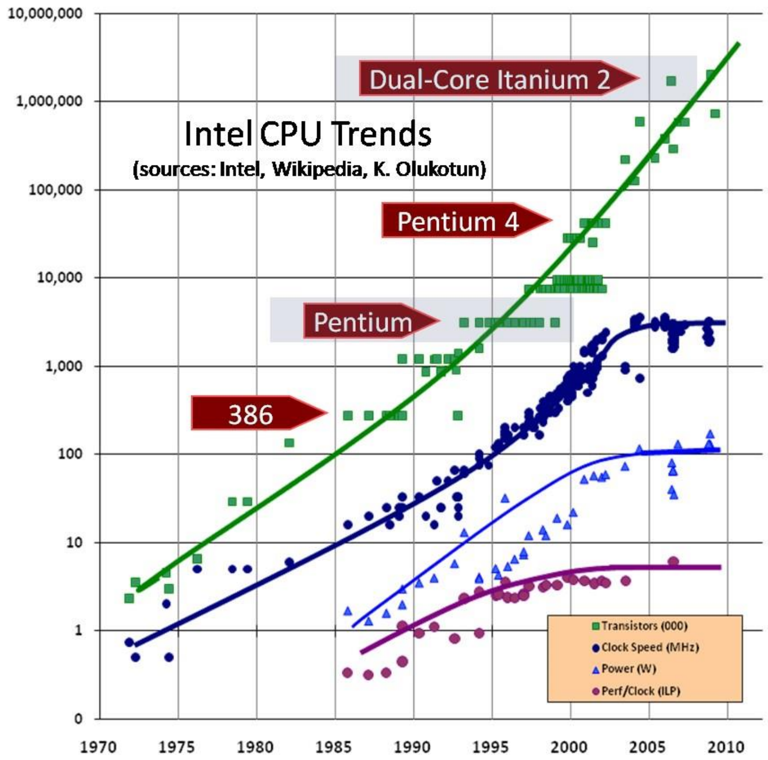
Hardware revolution

- **Moore's Law:** every 18-24 months, computer power improves by factor of 2 in price / performance as feature size shrinks
  - Newest processors are 64-bit and many-core

Enabling Modern Computer Graphics

Hardware revolution

- Significant advances in commodity graphics chips (GPUs) every 6 months vs. several years for general purpose CPUs
  - NVIDIA Titan XP… 3840 shaders (cores)
  - Graphic subsystems (GPUs)
    - Offloads graphics processing from CPU to chip designed for doing graphics operations fast
    - nVidia GeForce™, ATI Radeon™
    - GPUs used for special purpose computation, also being ganged together to make supercomputers
      - You can put multiple GPUs together in your computer using SLI.
  - GPUs has led to development of other dedicated subsystems
    - Physics: nVidia PhysX PPU (Physics Processing Unit), standard on many NVIDIA GPUs
    - Artificial Intelligence: AIseek Intia Processor (as of 2008)

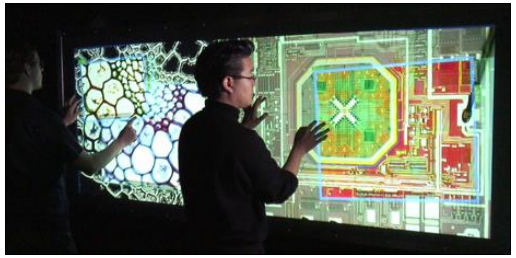


nVidia GeForce™ chip

Enabling Modern CG

- **Many form factors**
  - Cell Phones/PDAs (smartphones), Laptop/Desktops,
  - Jeff Han's Perceptive Pixel, Microsoft Surface
  - 3D immersive virtual reality systems

- **Software Improvements**
  - 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!
  - Algorithms and data structures
    - Rendering of natural phenomena
    - Acceleration data structures for ray tracing



Perceptive Pixel

Microsoft Surface

Brown's Cave™

Enabling Modern CG

- **Input Devices**
  - Mouse, tablet & stylus, multi-touch, force feedback, and other game controllers (e.g., Wii), scanner, digital camera (images, computer vision), etc.
  - Whole body as interaction device:
    - http://www.xbox.com/kinect

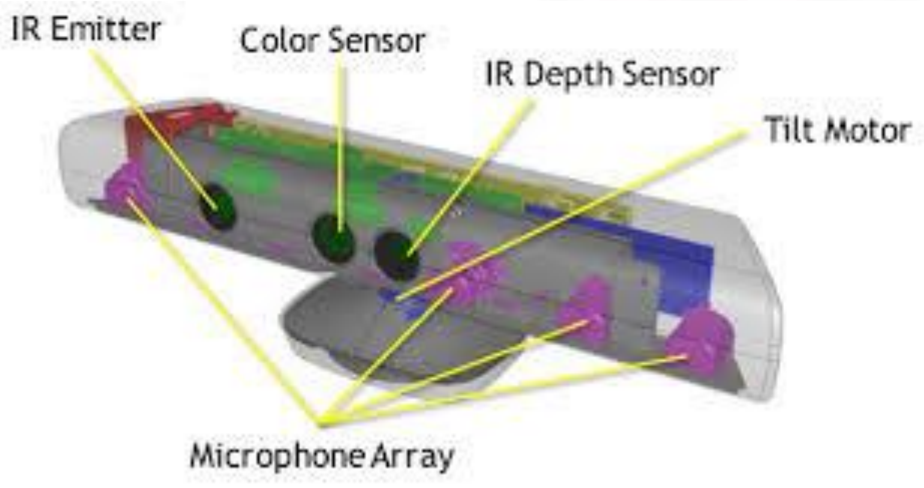
Xbox Kinect

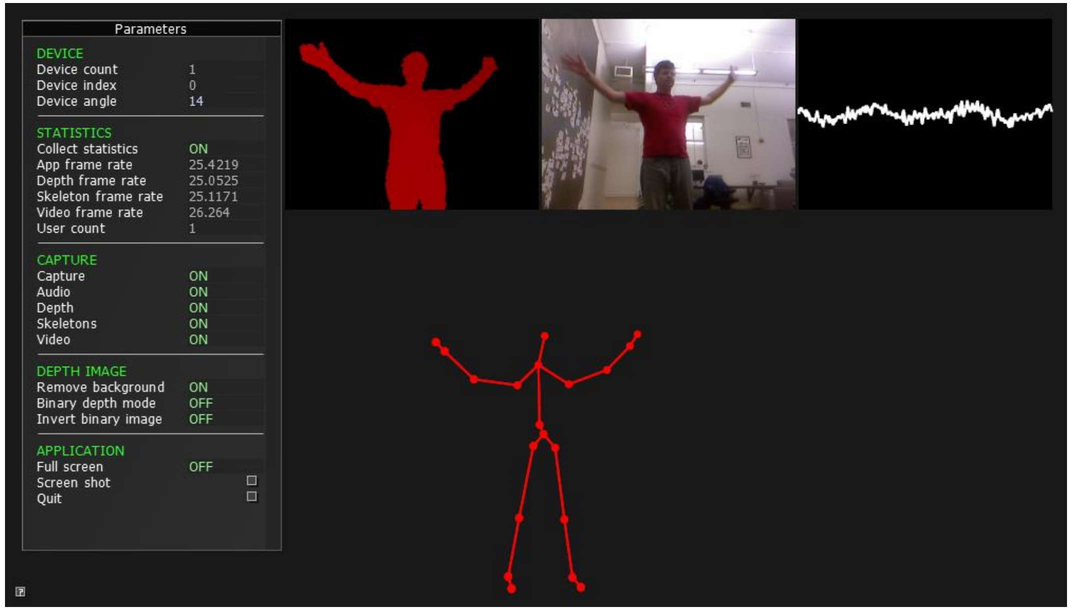

Leap Motion


Nimble UX

How do we interact with graphics images?

How do we interact with graphics images?





Immerse the user

- Stereo vision
- Tracking
- Haptics
- Surround sound
- Smell & taste (??)

How do we interact with graphics images?

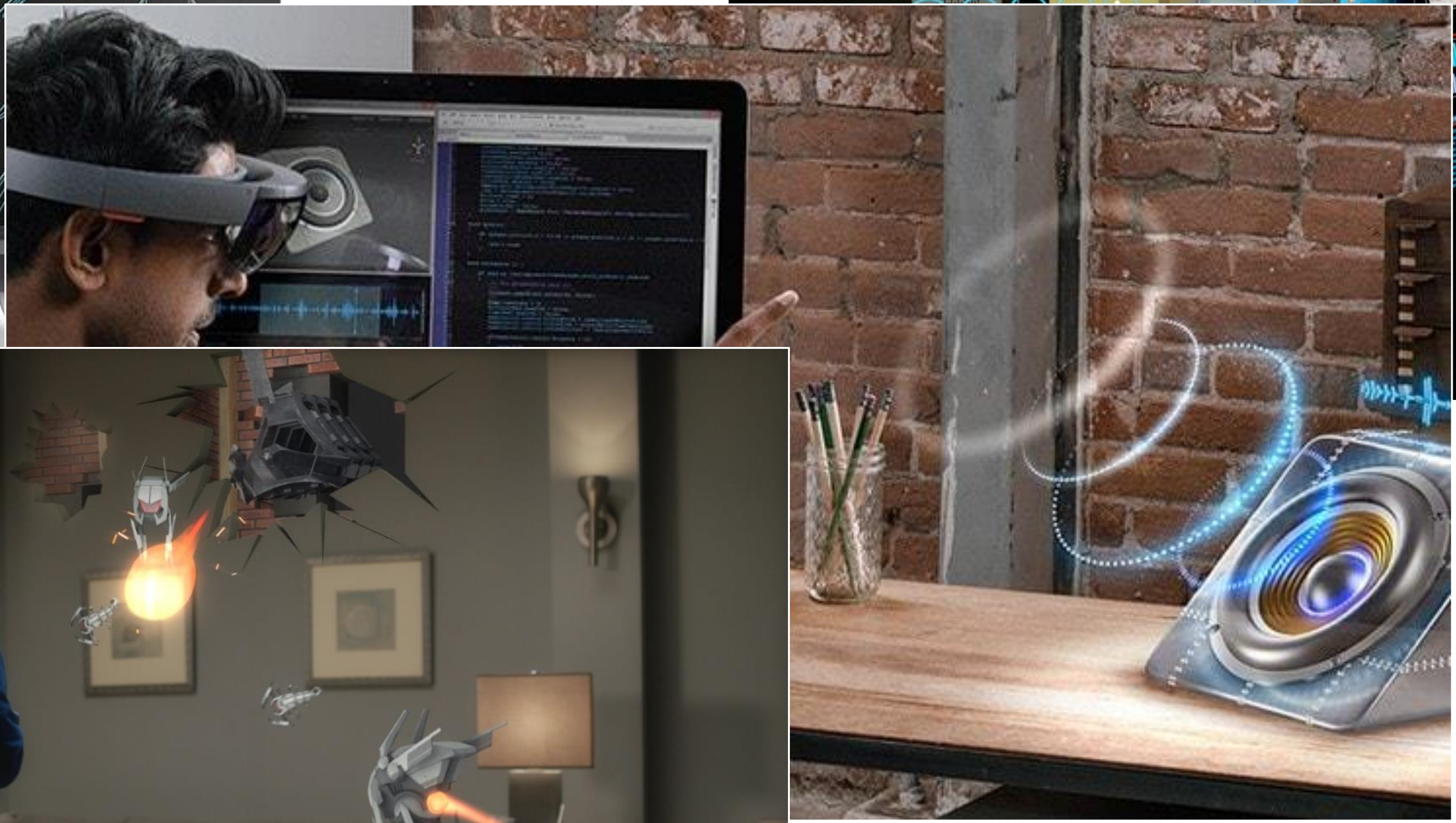
How do we interact with graphics images?

Stereo – Head Mounted Displays


htc
EVERESTVR
VIVEPORT
SUBSCRIPTION
Tilt Brush
by Google


Oculus

Augmented Reality Goggles



Computer Graphics: *Introduction*

- **Computer Graphics**
  - What's It All About?
  - Application Areas
  - Interactive Computer Graphics
- **How do graphics work**
  - Modeling
  - Animation
  - Rendering
- **Course Outline**
  - Lectures, Exams, Evaluation
  - Books
  - Miscellanea

How do we make this image?

Uncharted 4: https://youtu.be/zL46dpNEPPA

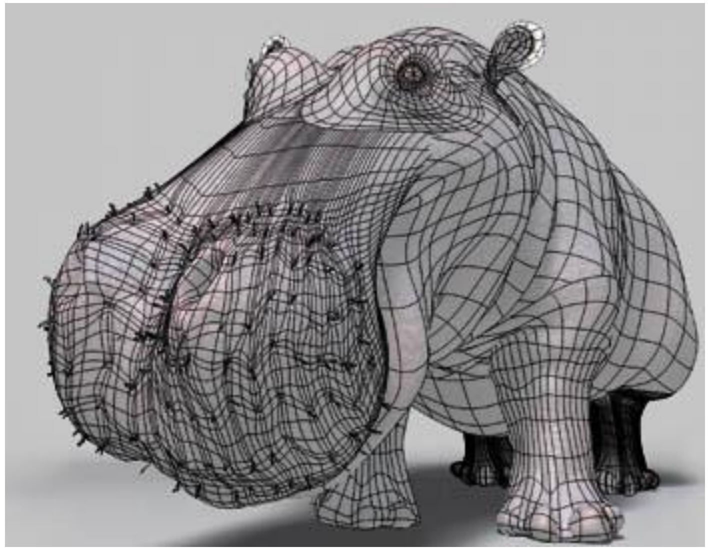
Making of Uncharted 4: https://youtu.be/3uKia6kb1fk

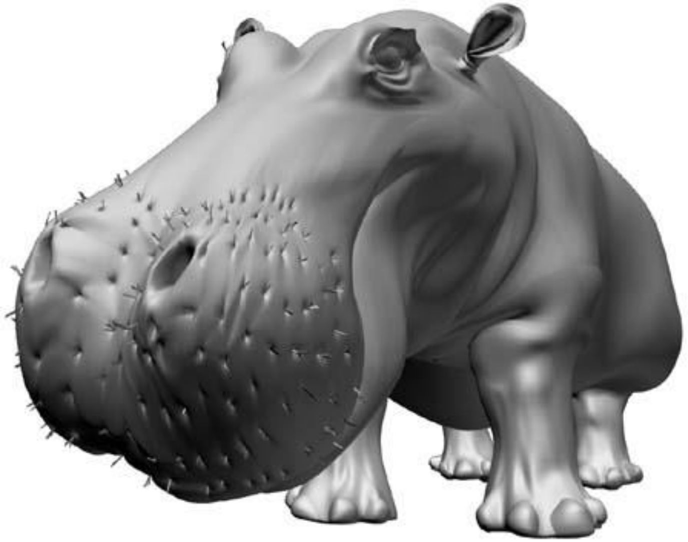
- Modelling
  - Geometry
  - Materials
  - Lighting
- Animation
  - How characters move?
- Rendering
  - Render light, shadows
  - Camera
  - Special Effects
  - Post-processing

How graphics images are made

- **Important factors:**
  1. Geometric Modeling: Create mathematical models of 2D and 3D objects.
  2. Animation: Definition/Representation of temporal behavior of objects.
  3. Rendering: Export images.

Geometric Modelling

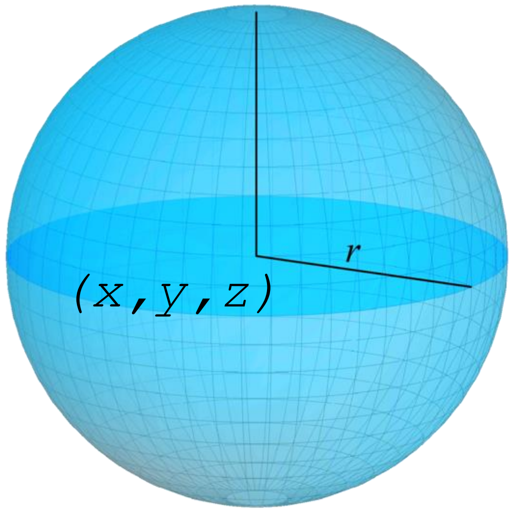


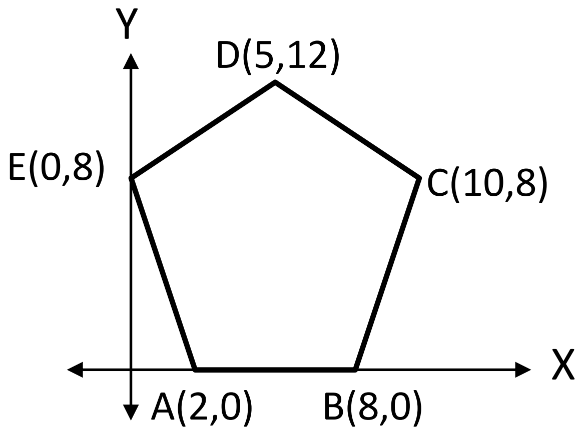


http://www.3drender.com/

Geometric Modelling

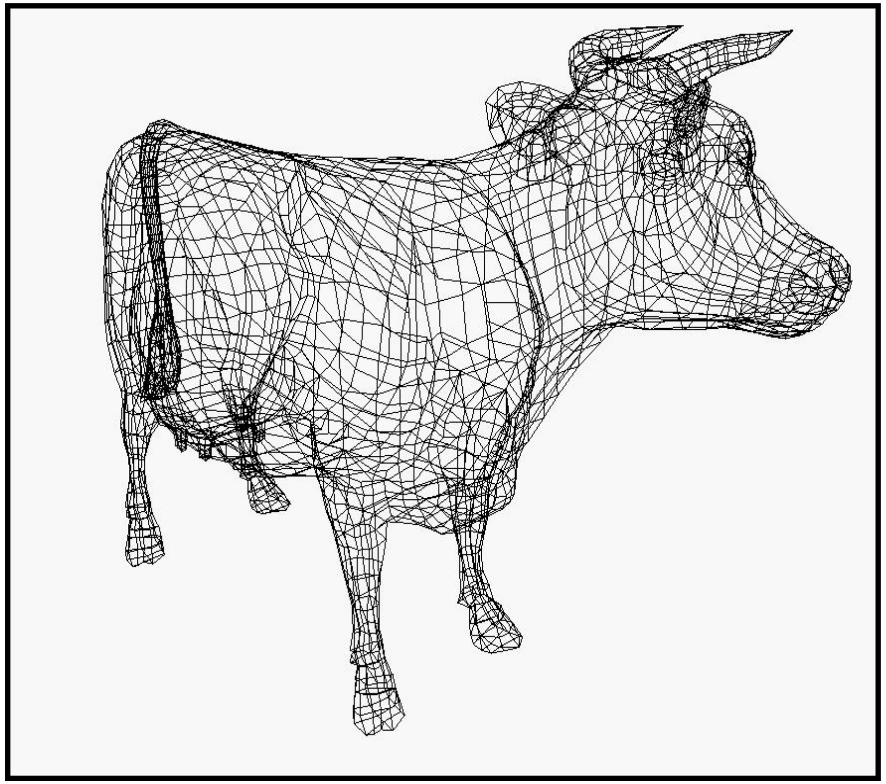
- Simple objects (primitives) can be easily defined:
  - E.g. For a sphere you only need 4 values (x,y,z,r).
  - E.g. For a polygon you need its vertices.

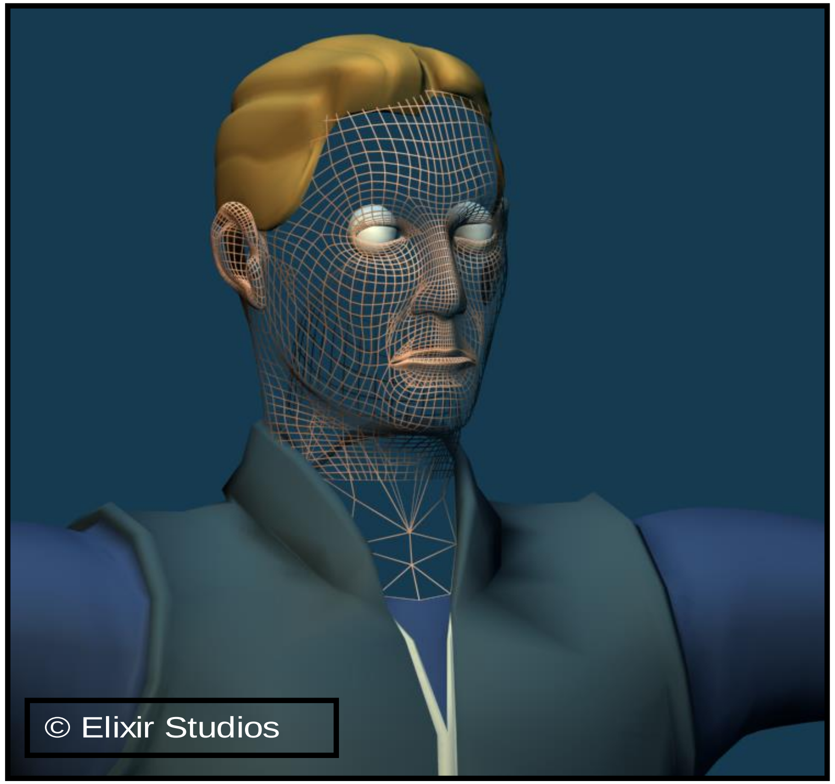




Geometric Modelling

- We combine polygons to define any shape.

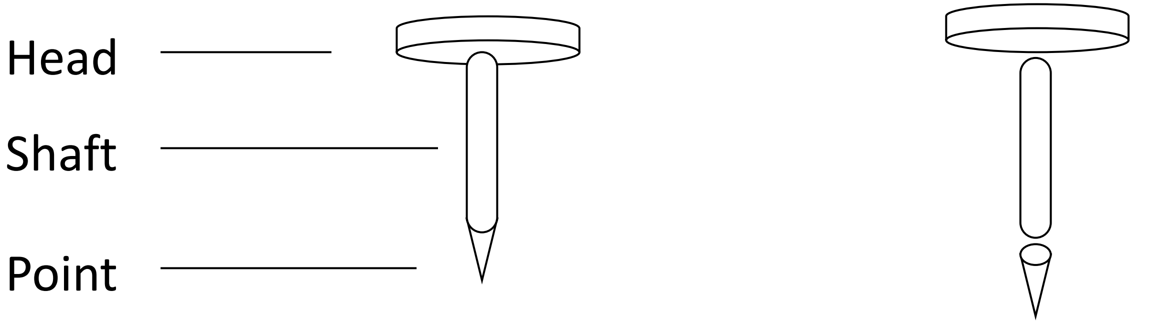




However, the number of polygons may be huge!

Decomposition of a Geometric Model

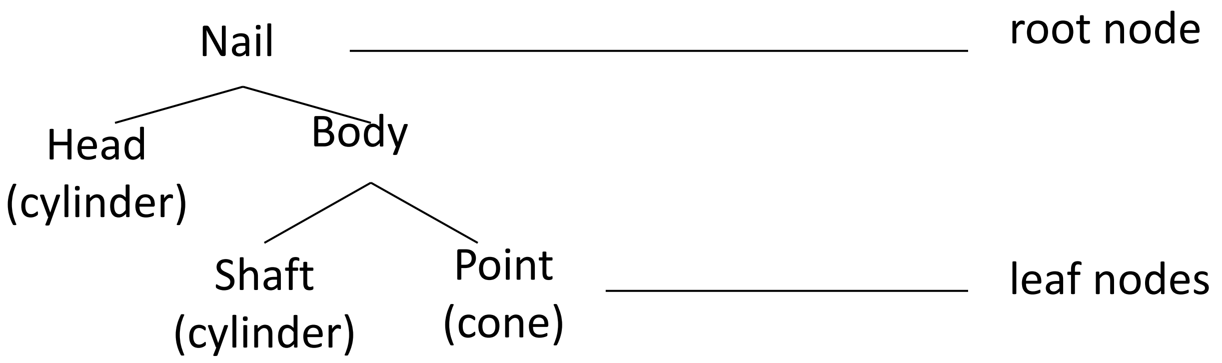
- Divide and Conquer
- Hierarchy of geometrical components
- Reduction to primitives (e.g., spheres, cubes, etc.)
- Simple vs. not-so-simple elements (nail vs. screw)



composition   decomposition

Hierarchical (Tree) Diagram of Nail

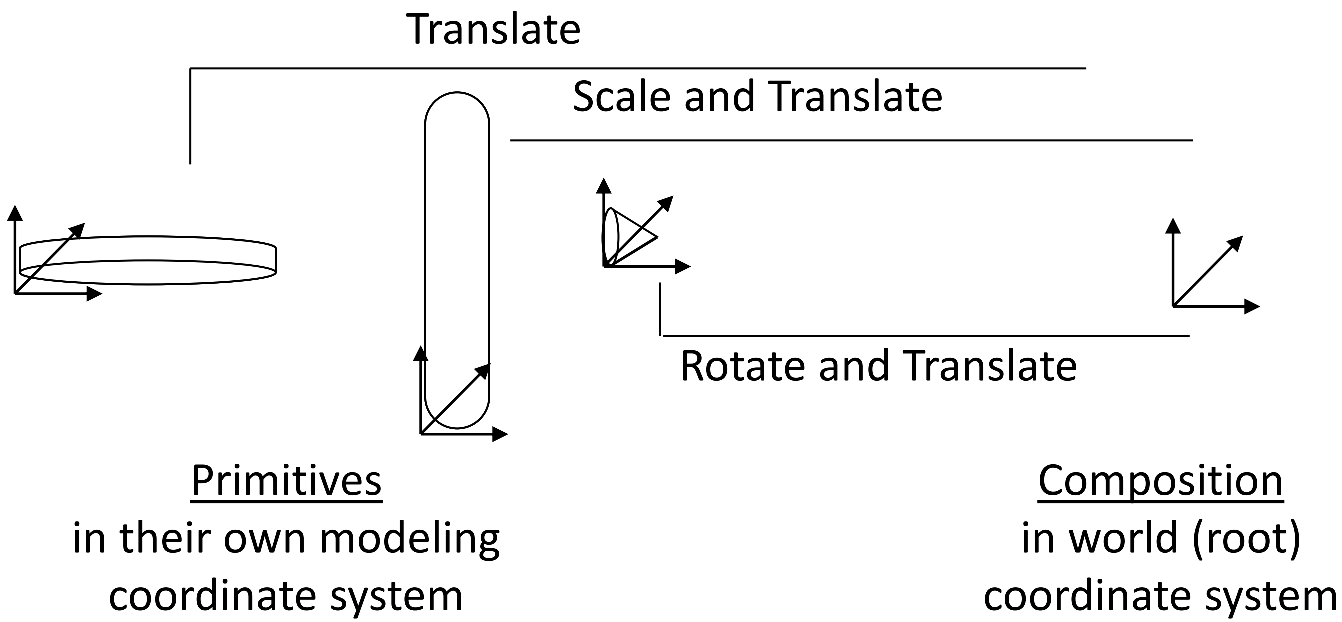
- Object to be modeled is (visually) analyzed, and then decomposed into collections of primitive shapes.
- Tree diagram provides visual method of expressing "composed of" relationships of model



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**

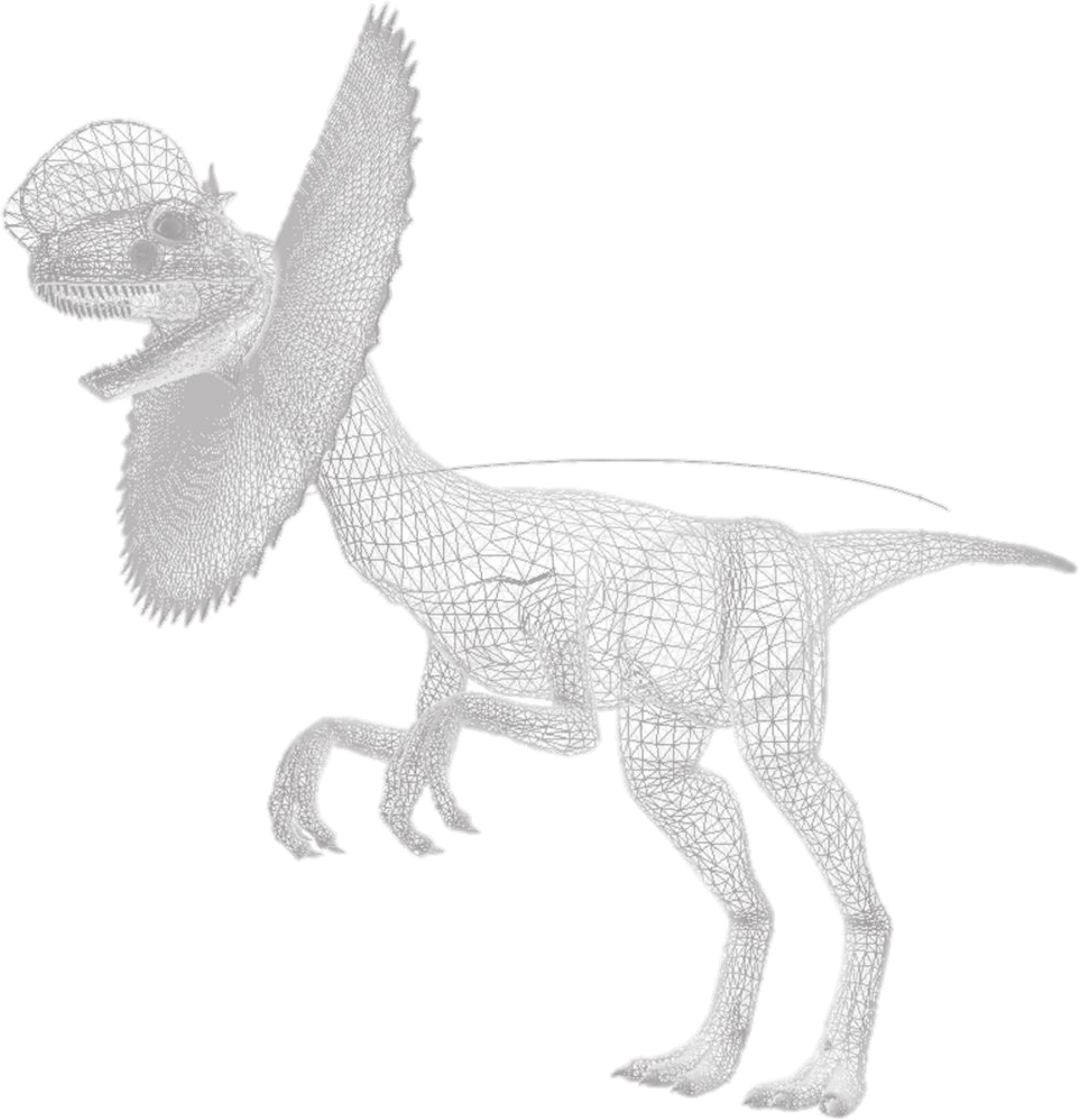
Composition of a Geometric Model



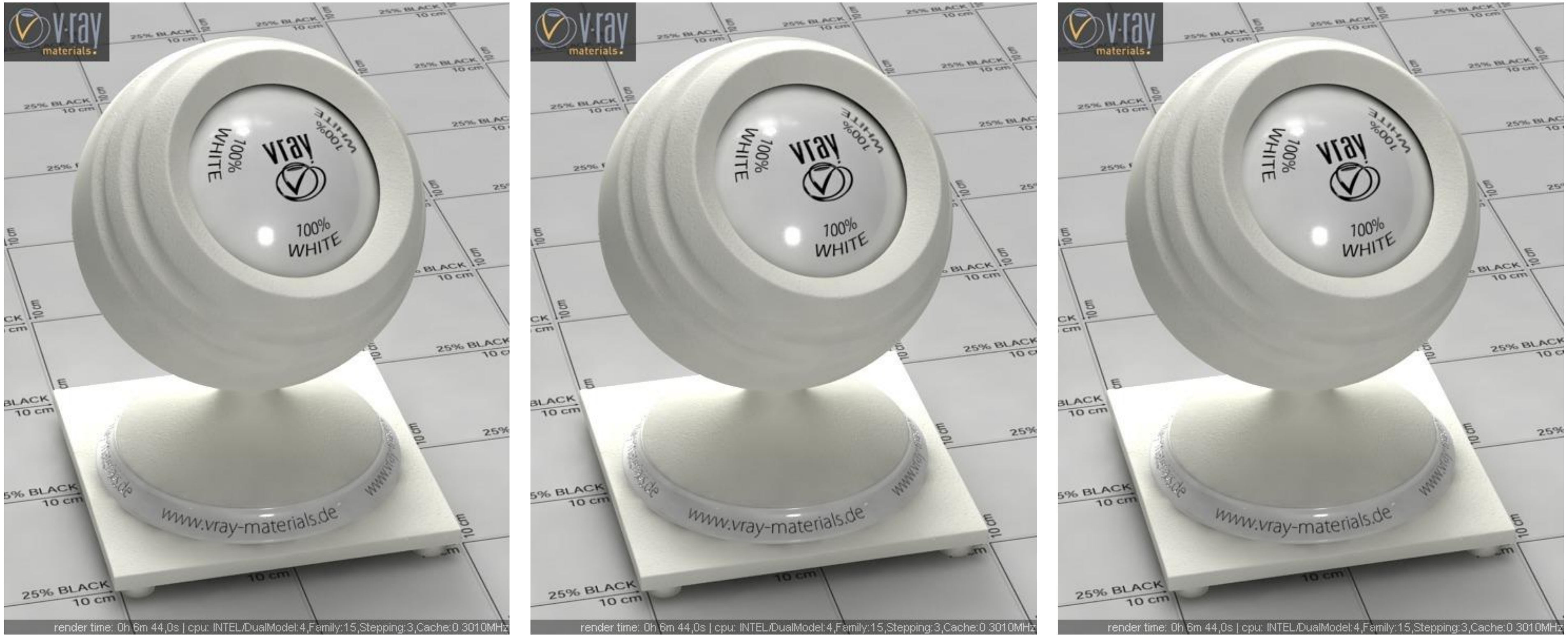
- Primitives created in decomposition process must be assembled to create final object. Done with **affine transformations**, T, R, S (as in above example). Order matters – these are not commutative!

Objects

- Objects consist of geometry + materials
- Geometry - typically a 3D Mesh
  - Approximates a continuous surface with a set of polygons (triangles + quads)
  - In offline rendering, we can also trace mathematical objects and volumes
- Material – describes how light interacts with the object

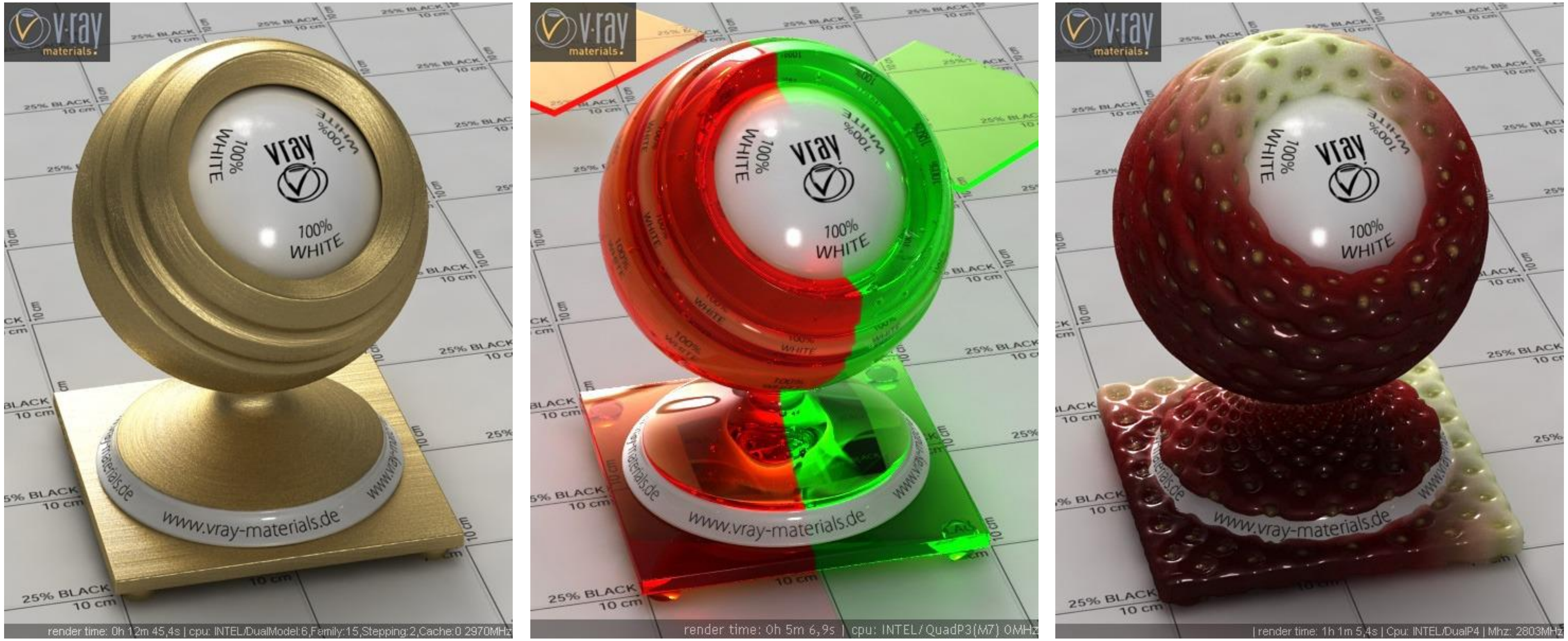


Modelling of materials



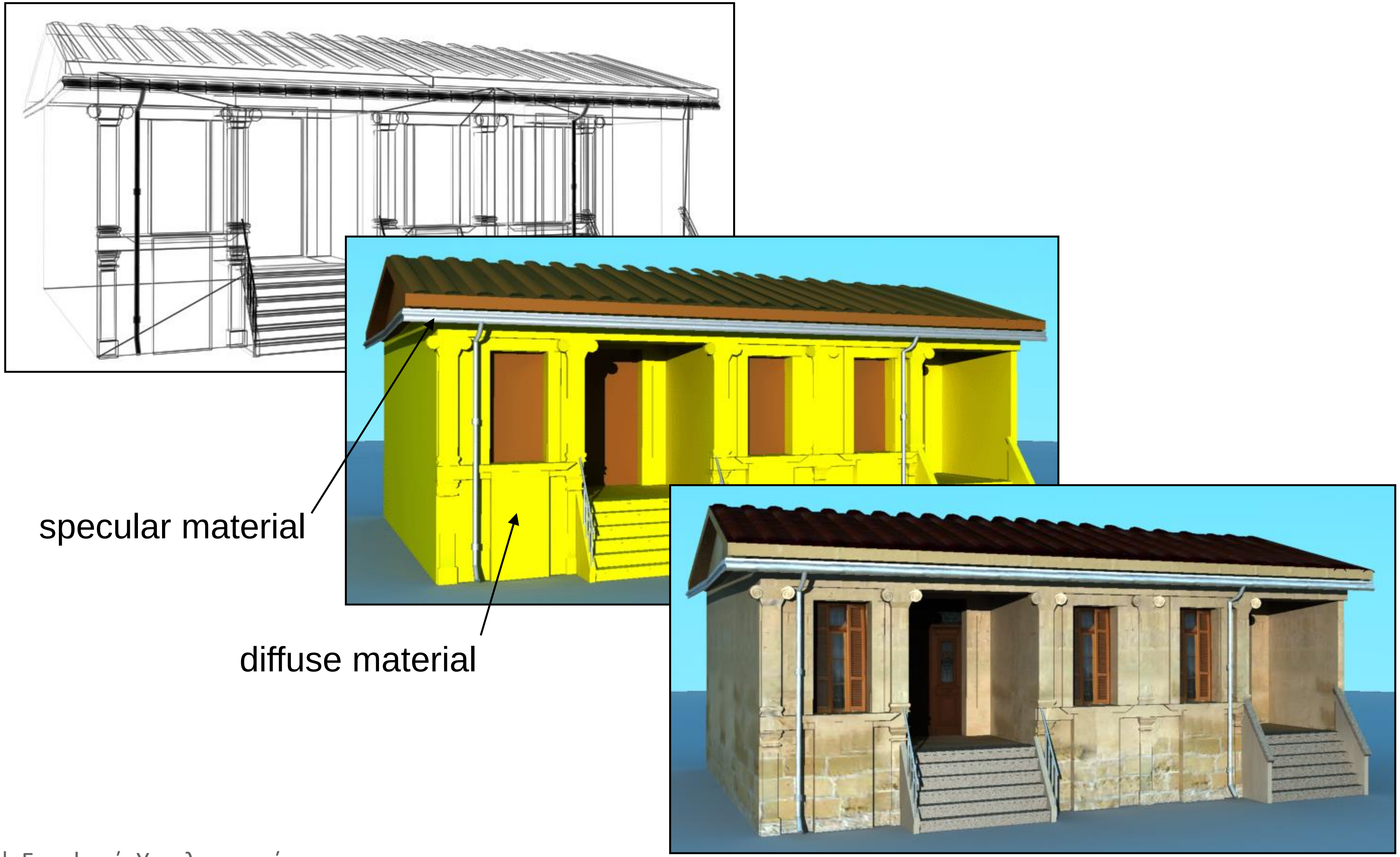
Identical Light Transport Algorithm, Geometry and Material descriptions

Modelling of materials



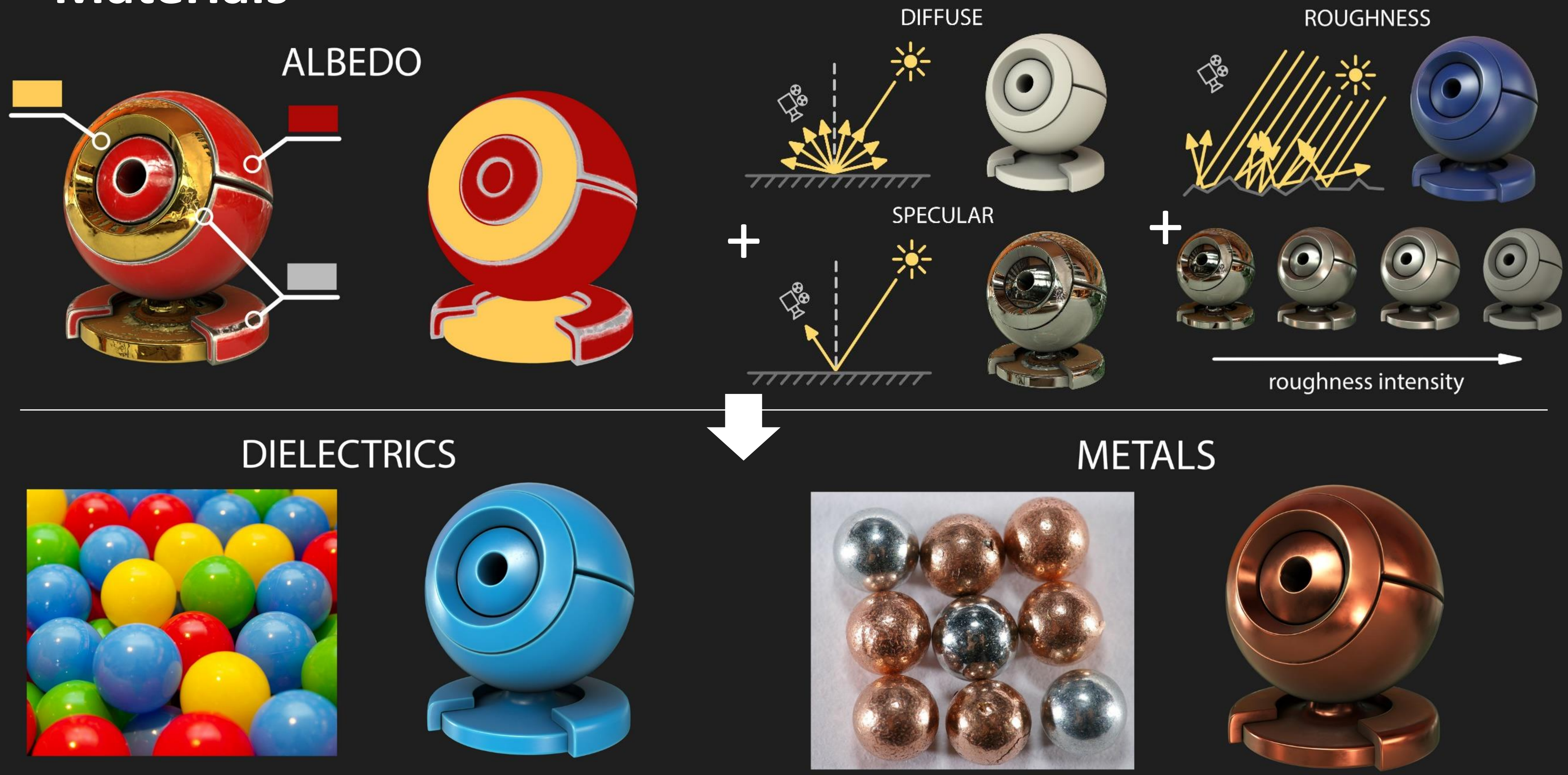
Identical Light Transport Algorithm and Geometry but different Material descriptions

Materials – reflective properties

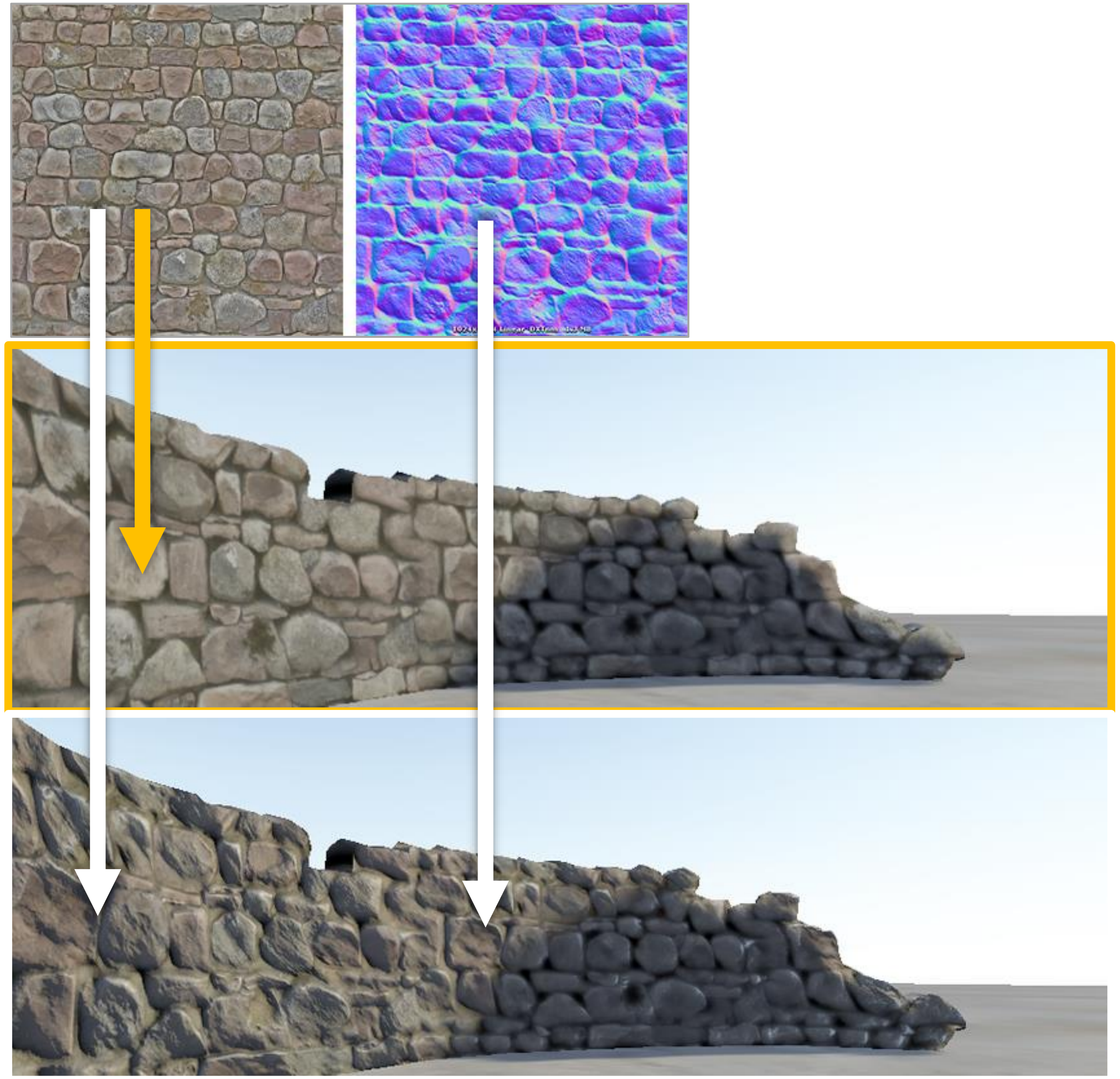


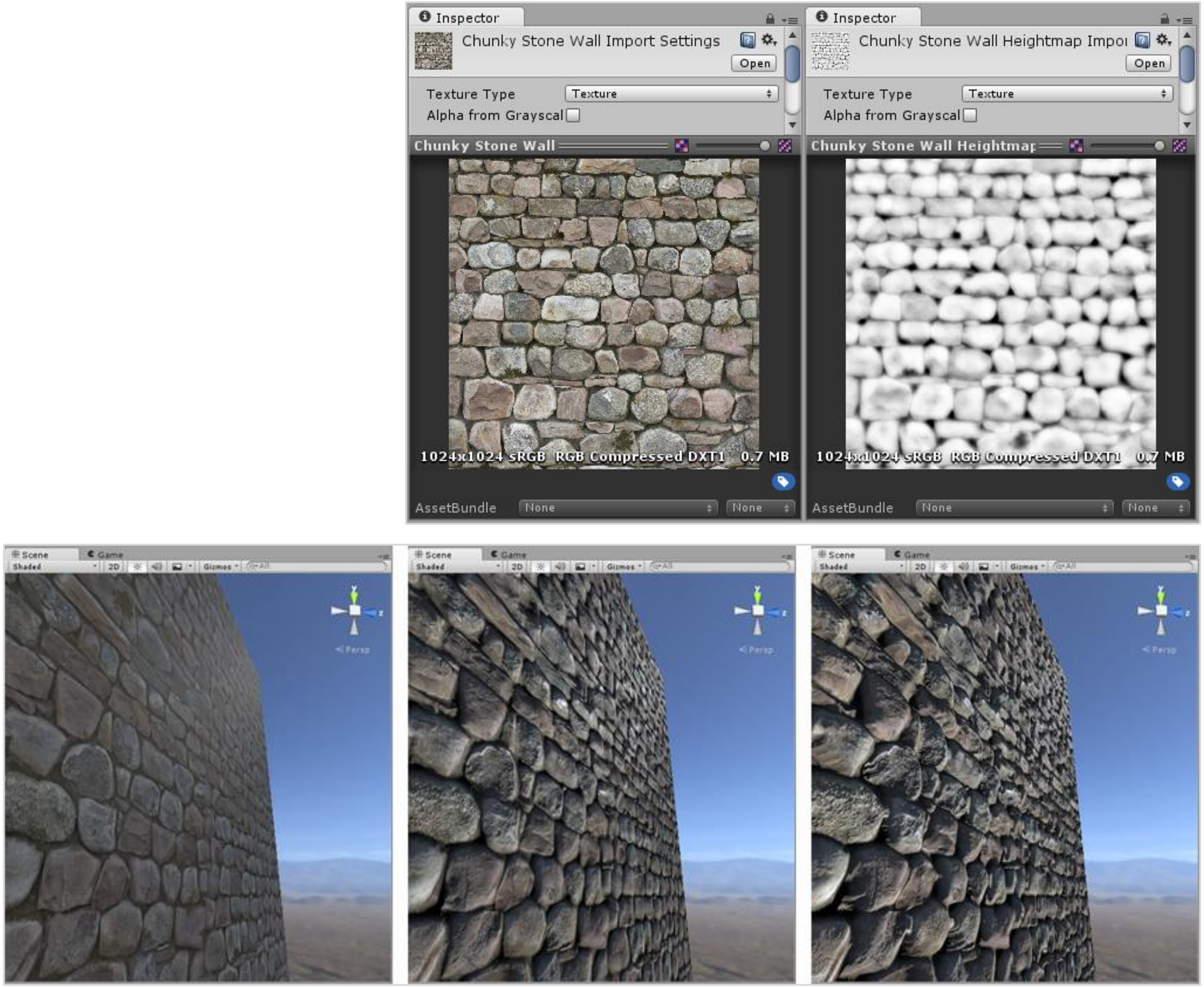
Materials

Images from: https://3dcoat.com/pbr/



Materials: Roughness without the geometry

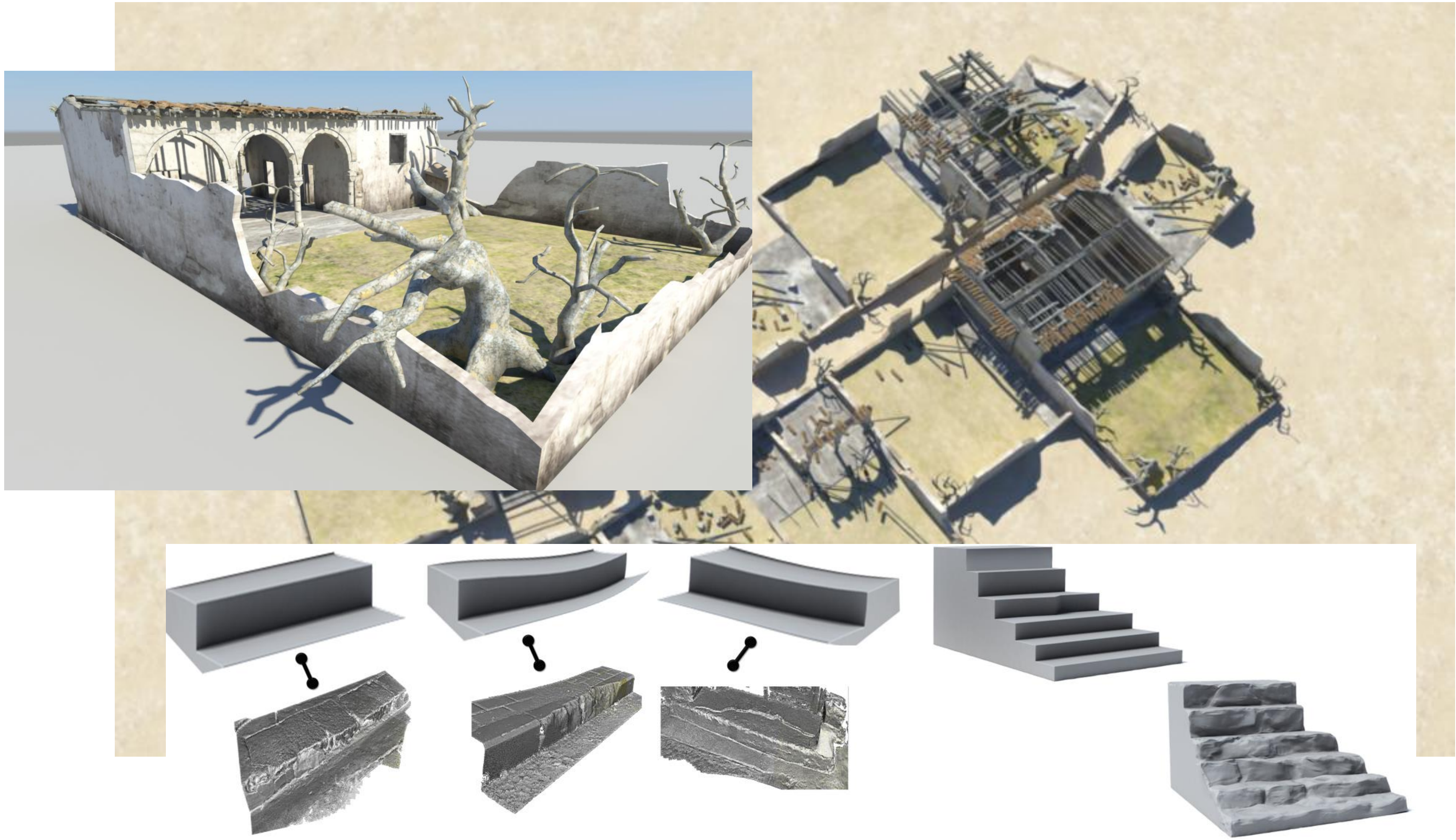




Materials

Your imagination is the limit

Materials: *Aging*

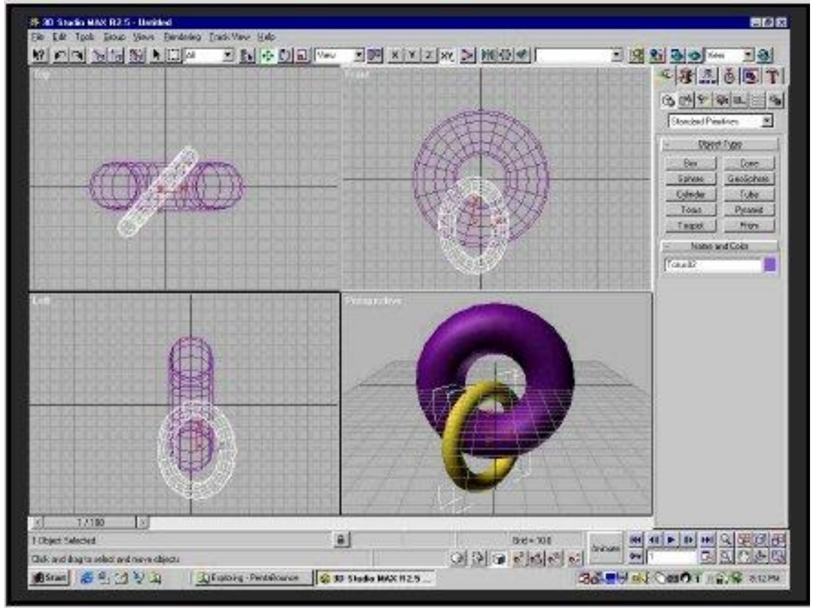


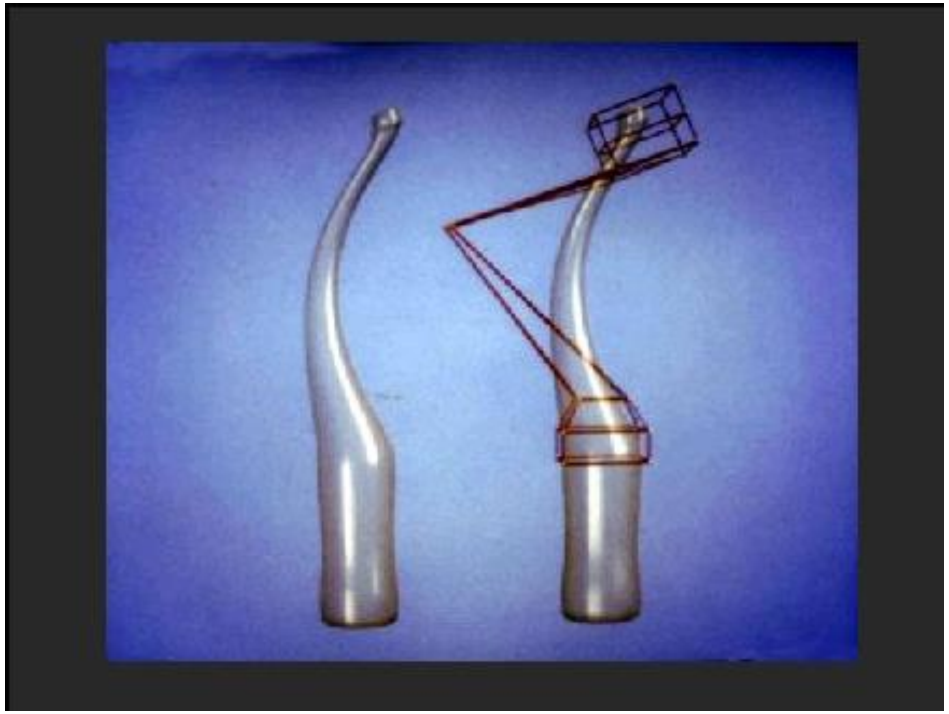
Modelling methods

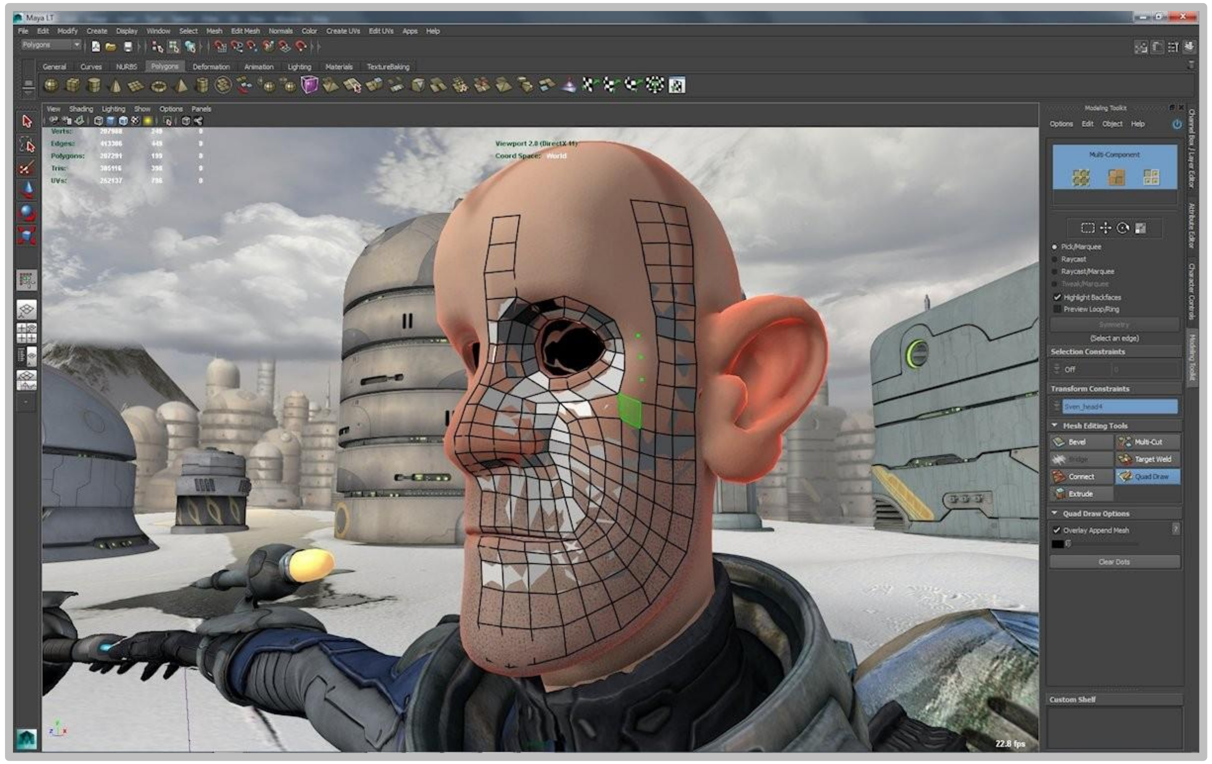
- Model library

Modelling methods

- Model library
- **Modeling software**

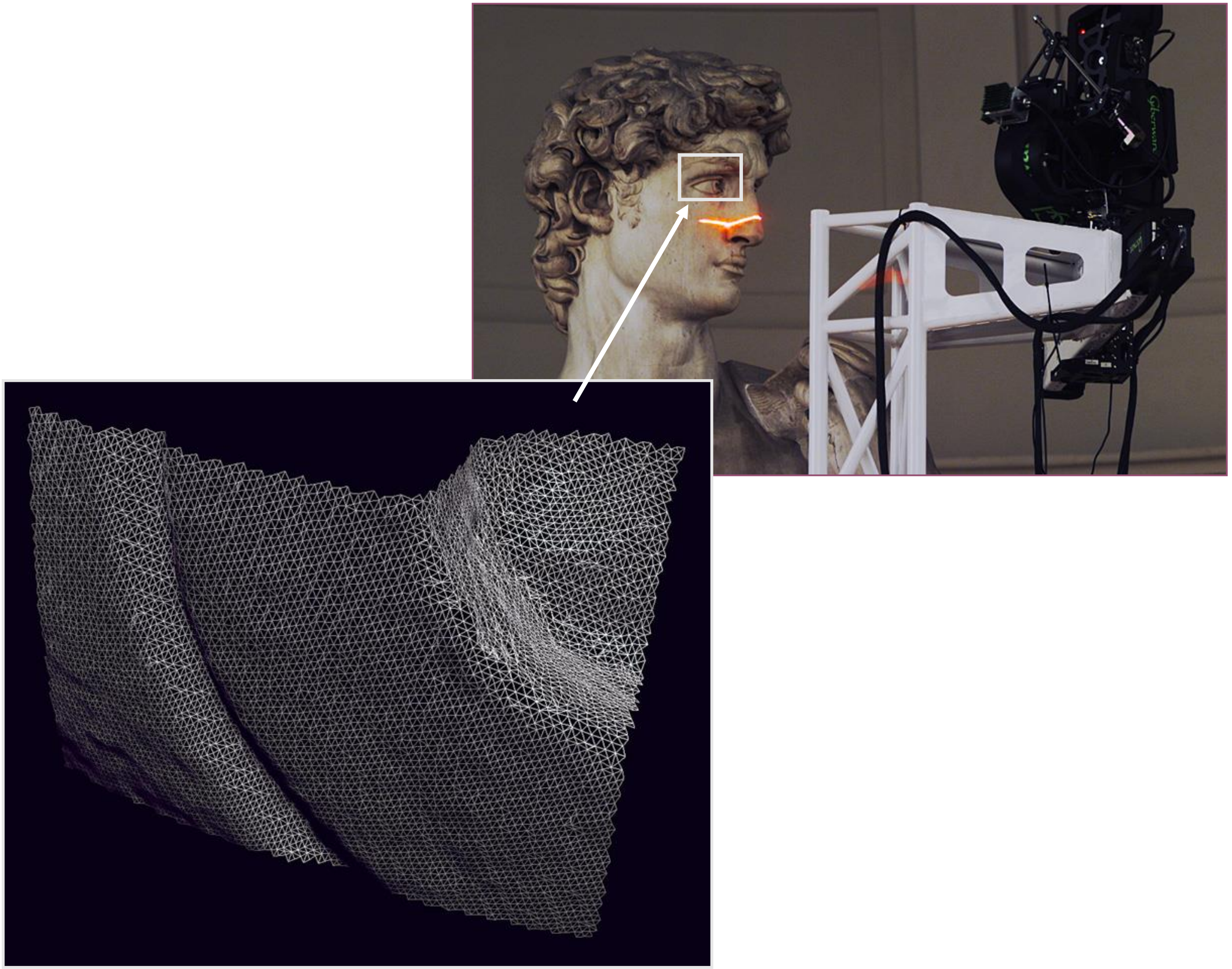






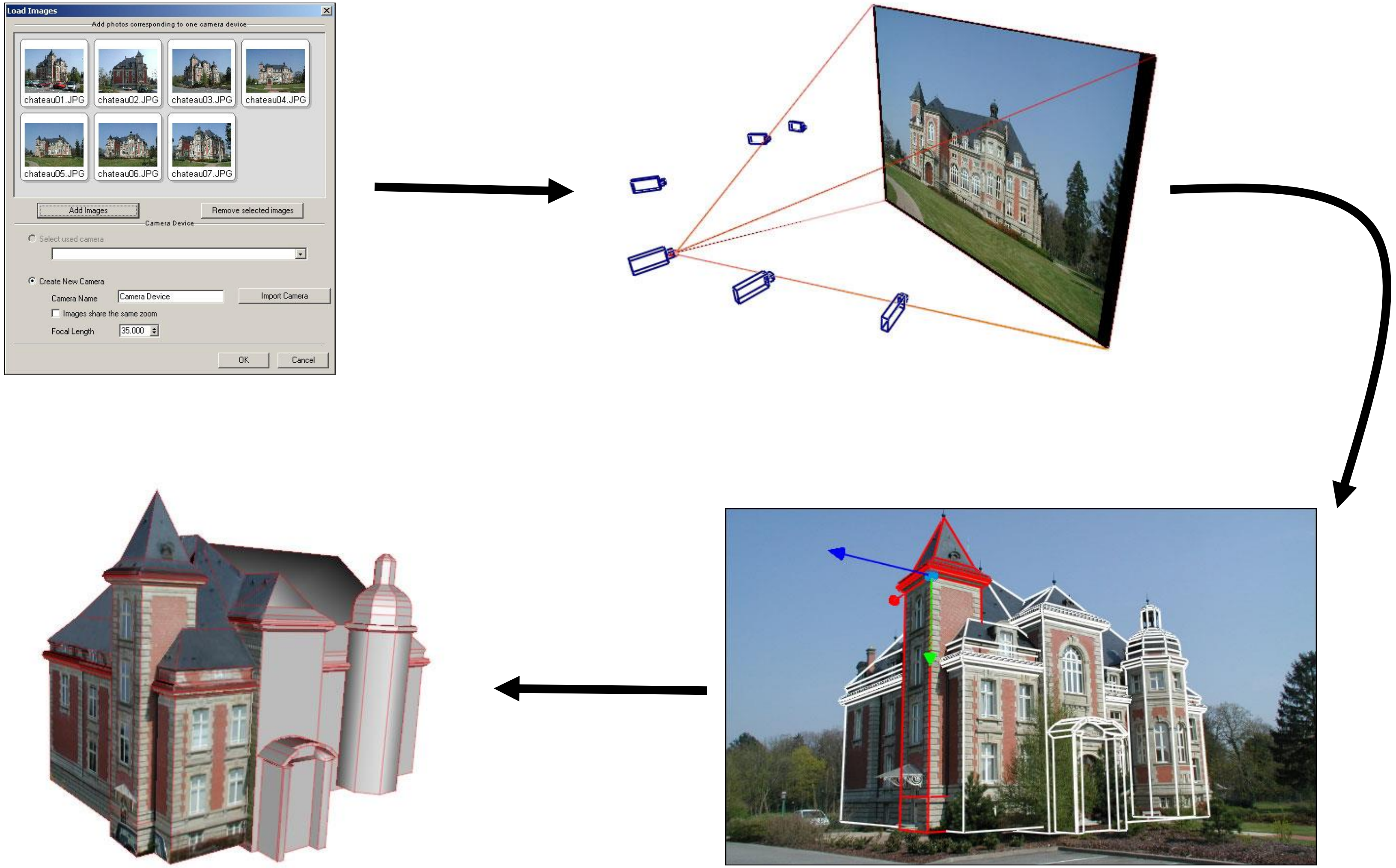
Modelling methods

- Model library
- Modeling software
- **3D scanner**



Modelling methods

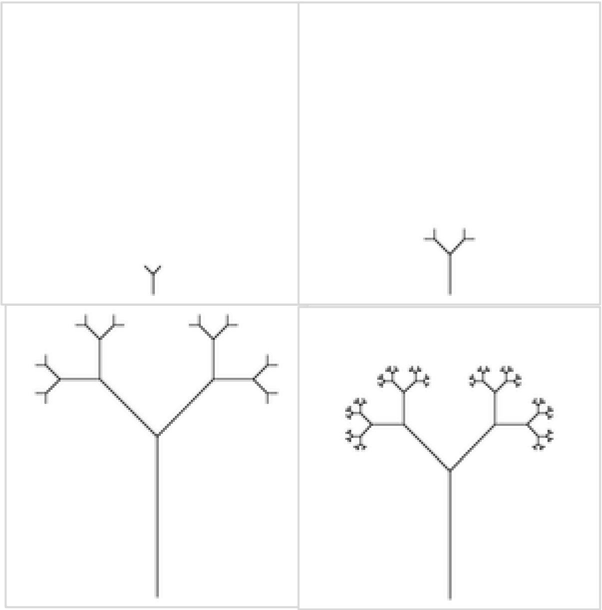
- Model library
- Modeling software
- 3D scanner
- **Image based modeling**

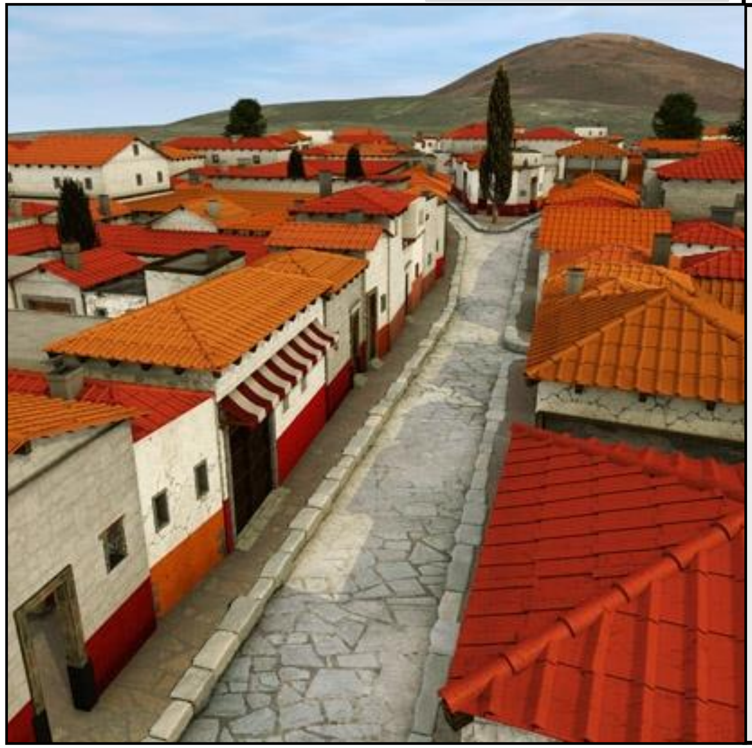


Paphos Gate

Modelling methods

- Model library
- Modeling software
- 3D scanner
- Image based modeling
- **Procedural**



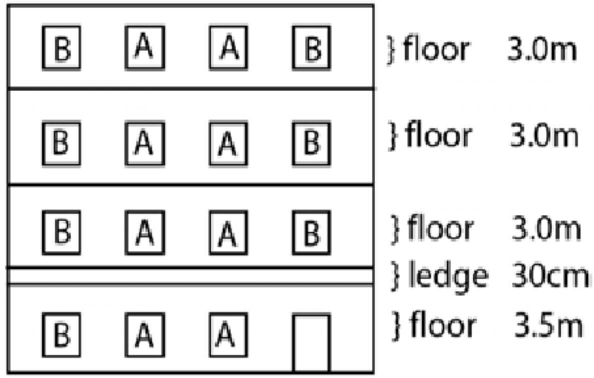


Virtual Pompeii
https://youtu.be/dQs9h3YurOk

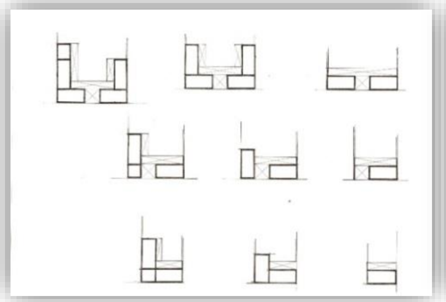
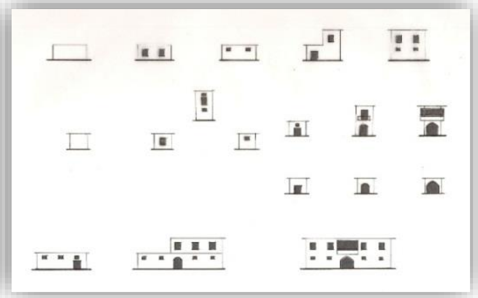
Procedural modeling (using Esri's CityEngine)

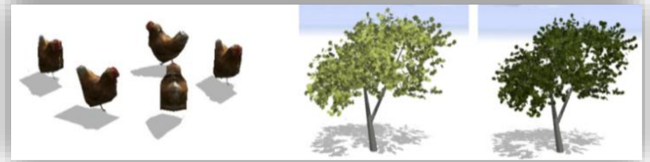
- Use CGA shape grammar
- A set of rules that describe recursively the shape and details of the buildings

1: fac $\rightsquigarrow$ Subdiv("Y",3.5,0.3,3,3,3){ floor | ledge | floor | floor | floor }

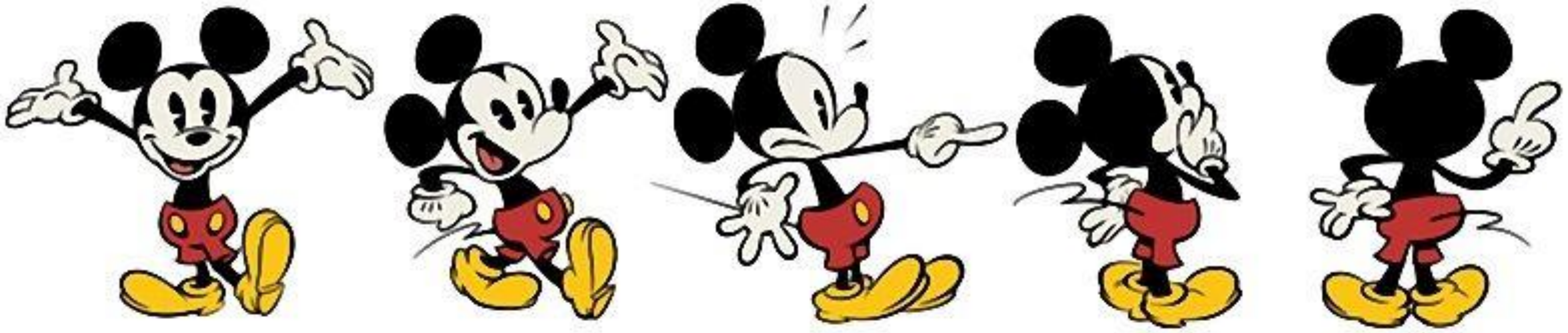


Procedural modeling

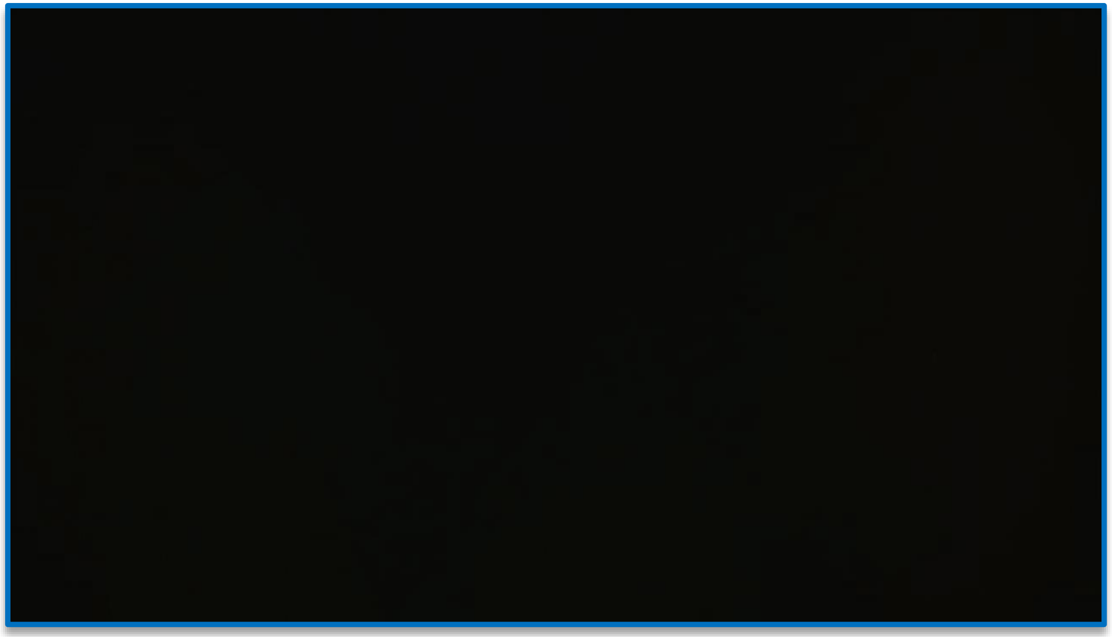



Animation

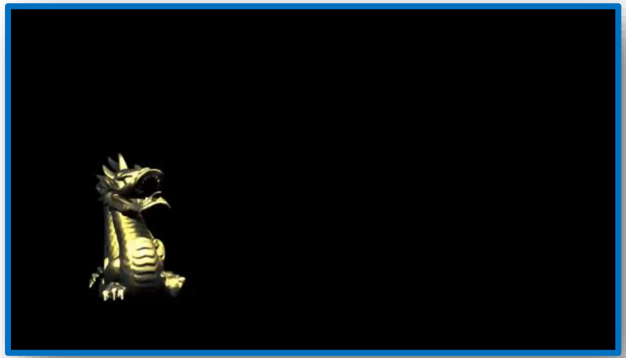


Animation

- It allows us to bring life into virtual characters
- There are many applications that require motion of **characters** or motion of **objects**.
  - Entertainment industry (e.g., movies, games)
  - Virtual / Augmented Reality.
  - Applications in Education or for Simulations.
- Other animation types?
  - Trees, liquids, fire, animals, clouds, etc …
- Any other important factors for animation production?
  - Lighting, Rendering, etc …



© Pixar Animation

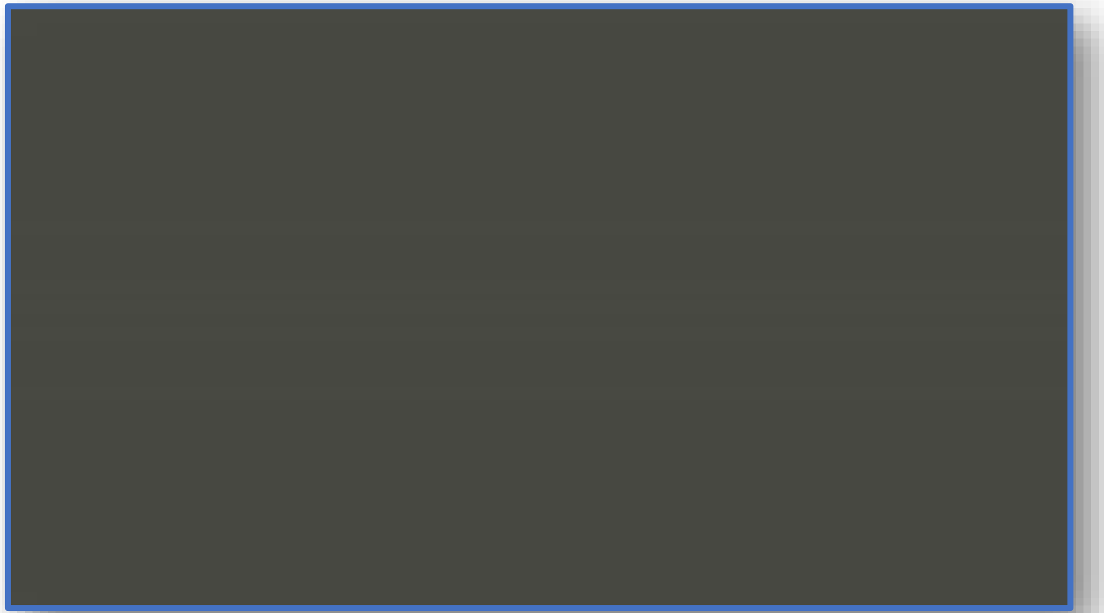


Character Animation

- 3 main categories:
  - **Skeleton motion** – motion at the main parts of the body.

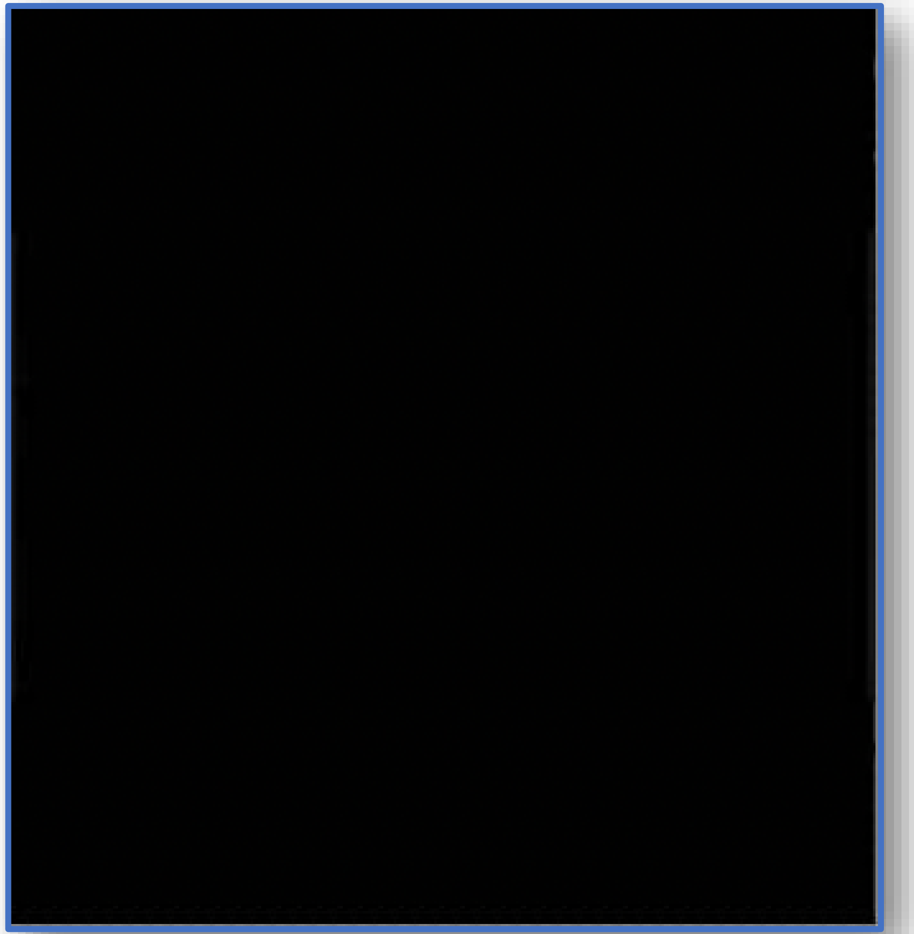
Character Animation

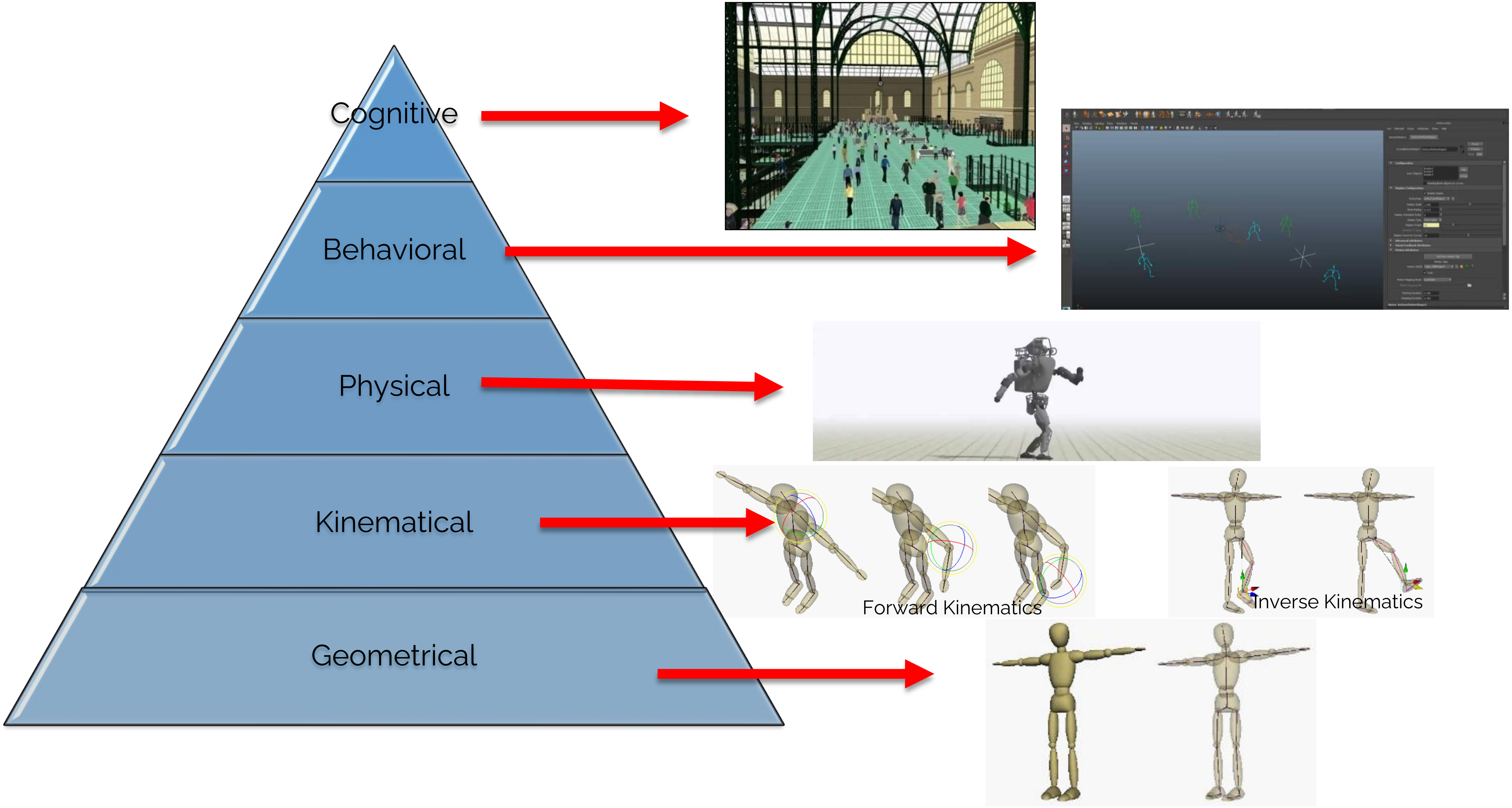
- 3 main categories:
  - **Skeleton motion** – motion at the main parts of the body.
  - **Facial motion** – motion at the main characteristics of the face.

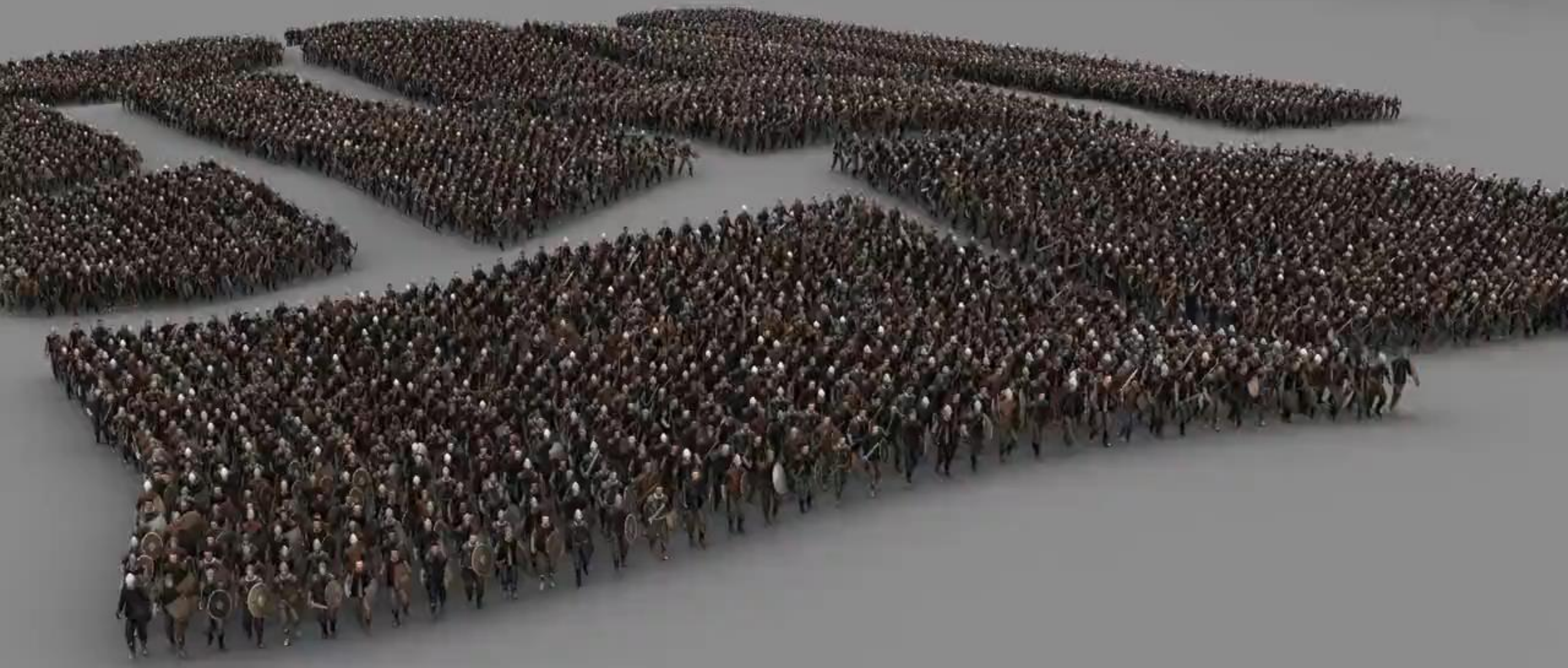


Character Animation

- 3 main categories:
  - **Skeleton motion** – motion at the main parts of the body.
  - **Facial motion** – motion at the main characteristics of the face.
  - **Hair** motion**, skin** motion, and **clothing** motion




Cognitive
Behavioral
Physical
Kinematical
Geometrical
Forward Kinematics
Inverse Kinematics



Crowd behavior


FRAMESTORE

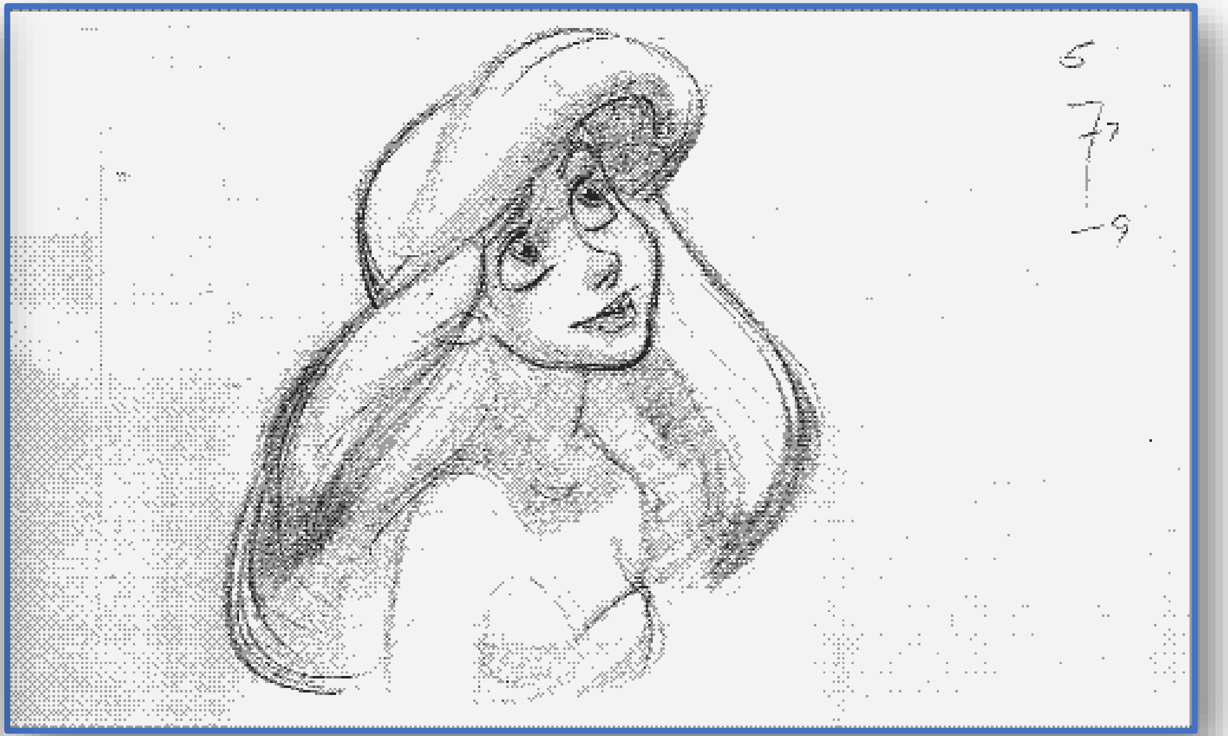

STARGATE
STUDIOS

Animation: *Introduction*

- What is the most common way of creating animated characters?

Animation: *Animated Manually: key-framing*

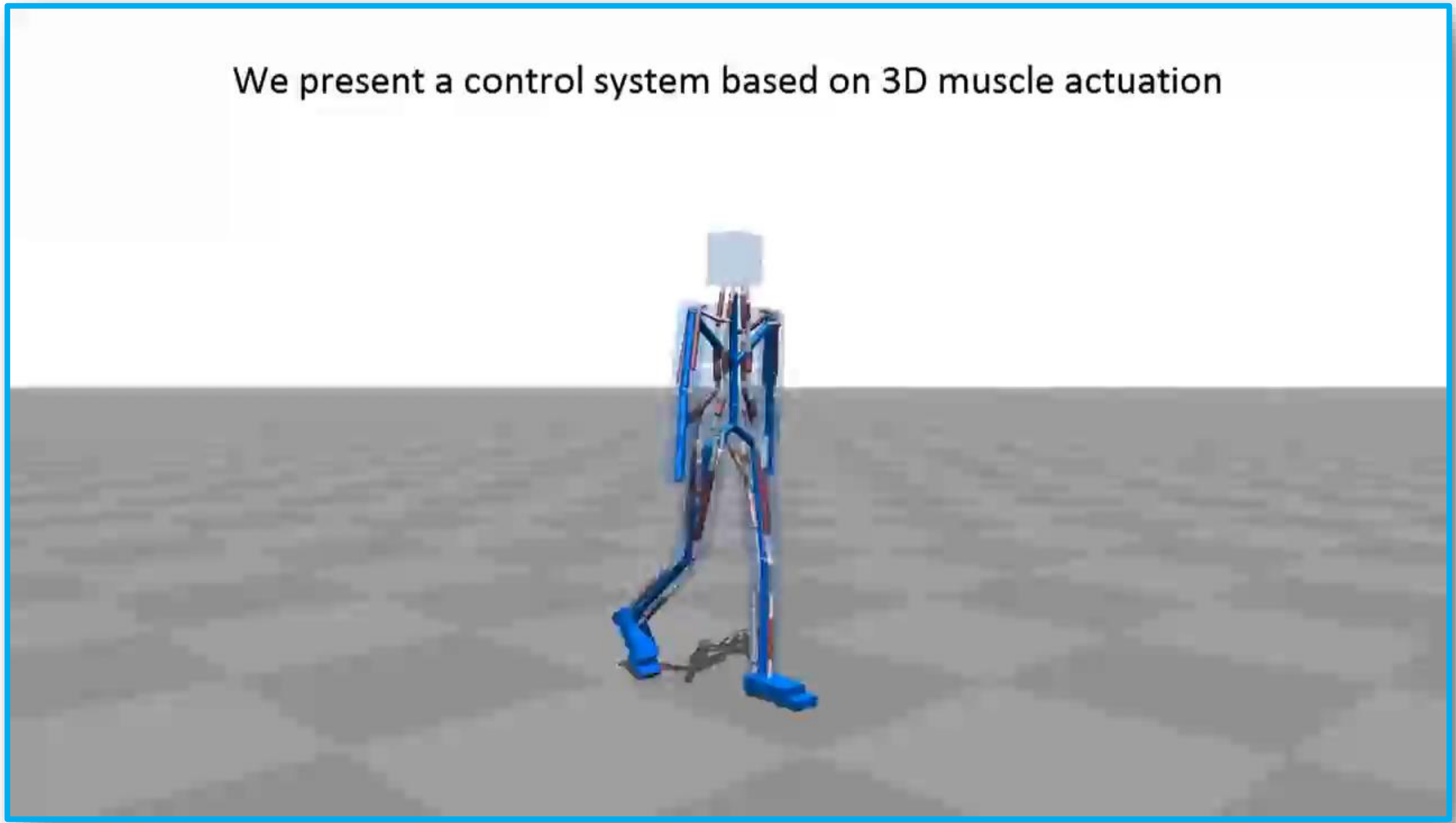
Flipbook Animation

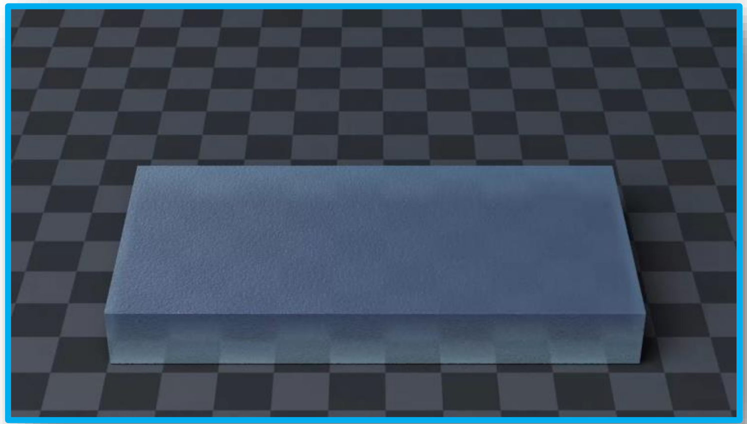


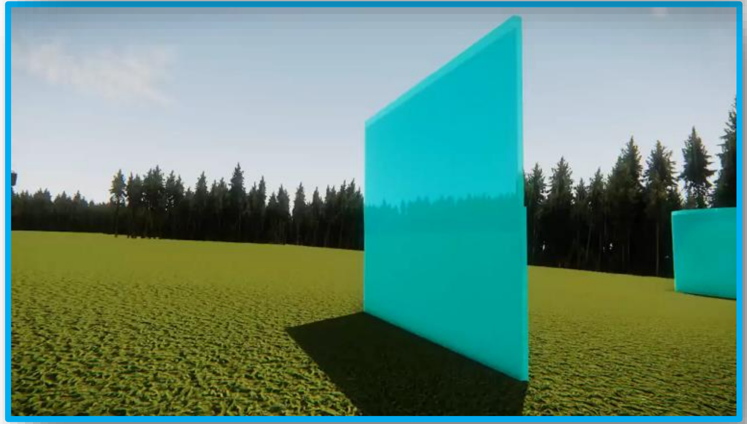
@ Disney Studios

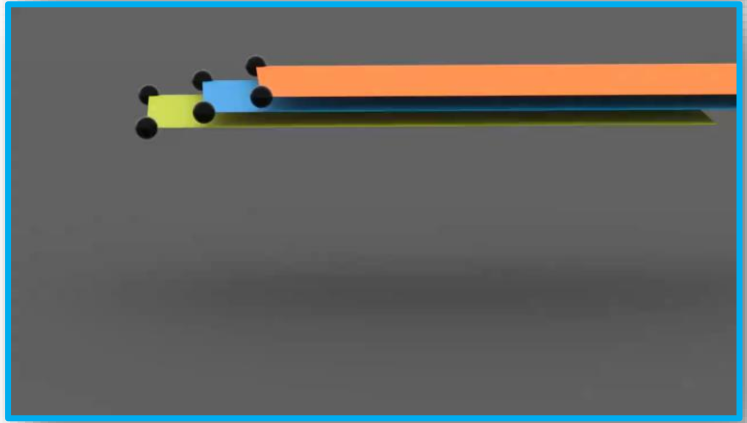
Animation: *Animated Manually: key-framing*

Character Animation: *Physics Simulation*

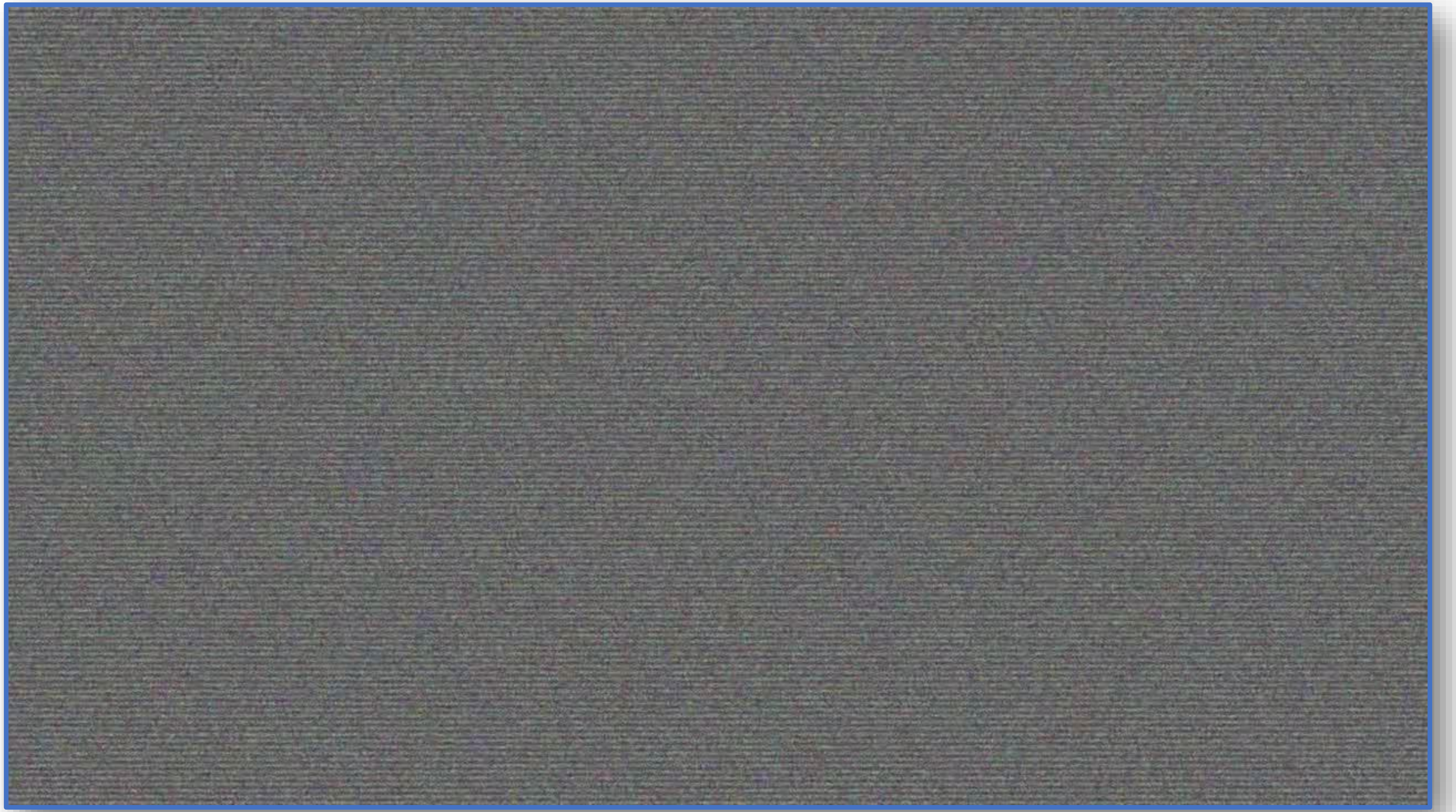








Character Animation: *Motion Capture*

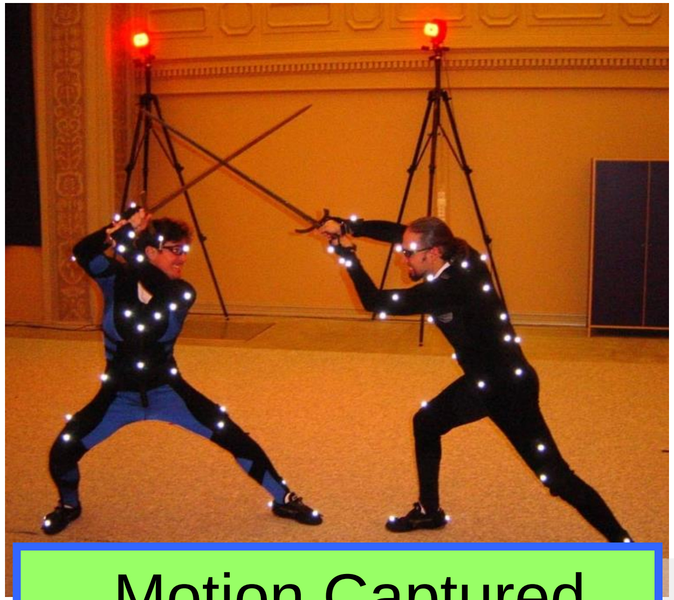


Character Animation: *Motion Capture*

Animation

Animating Characters is the most difficult.

Animated Manually



Motion Captured

Synthesize in some Manner

3. Rendering

Modelling Vs Rendering

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

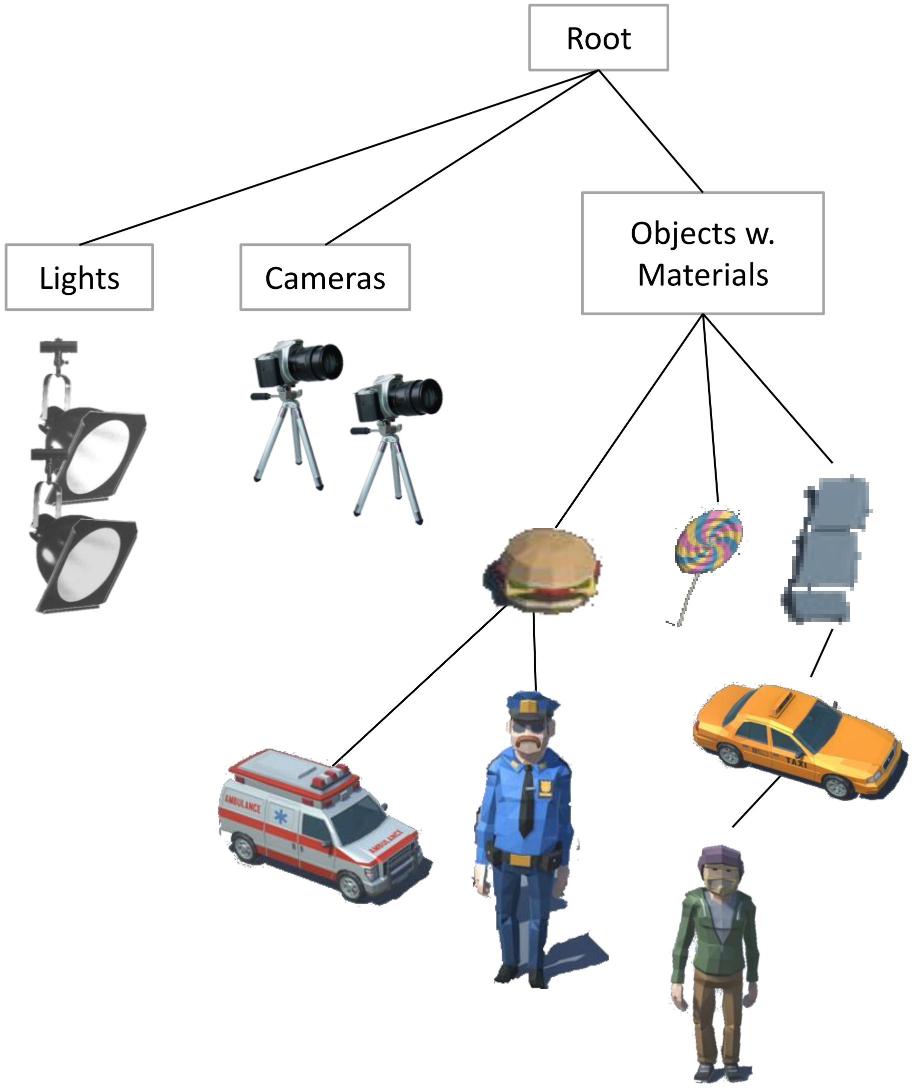
- **Rendering**
  - Take "picture" with camera

- Both can be done with commercial software: Autodesk Maya™ ,3D Studio Max™, Blender™, etc.

CS128 lighting assignment by Patrick Doran, Spring 2009

Computer Graphics

Scene Description



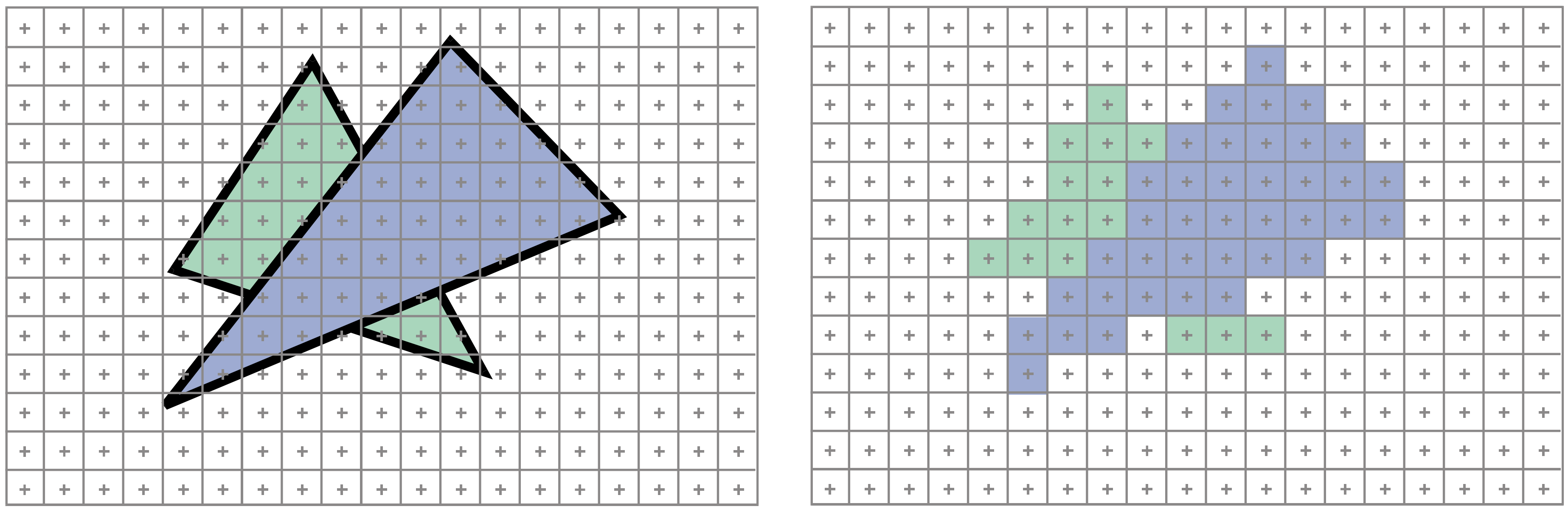
Rendering Algorithm

Image

Polygon City Pack for Unity

https://www.assetstore.unity3d.com/en/#!/content/95214

What is a digital image?

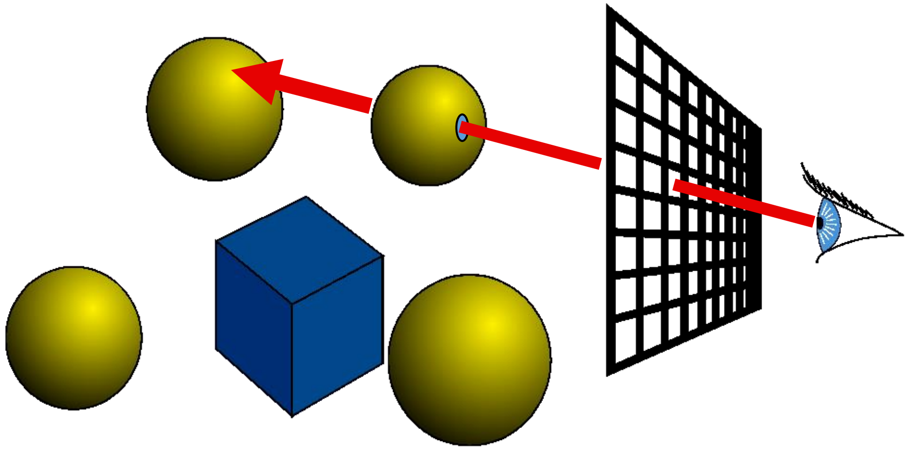


Two main approaches for creating images.

A. Ray Casting

- For each pixel
  - For each object

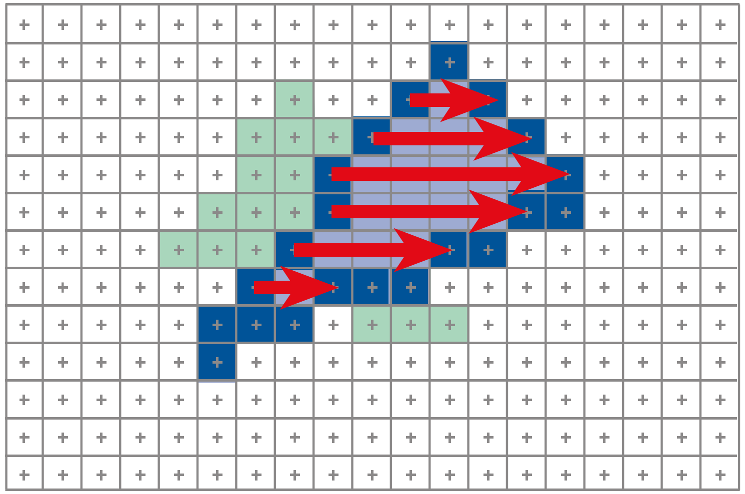
Send pixels to the scene



B. Rendering Pipeline

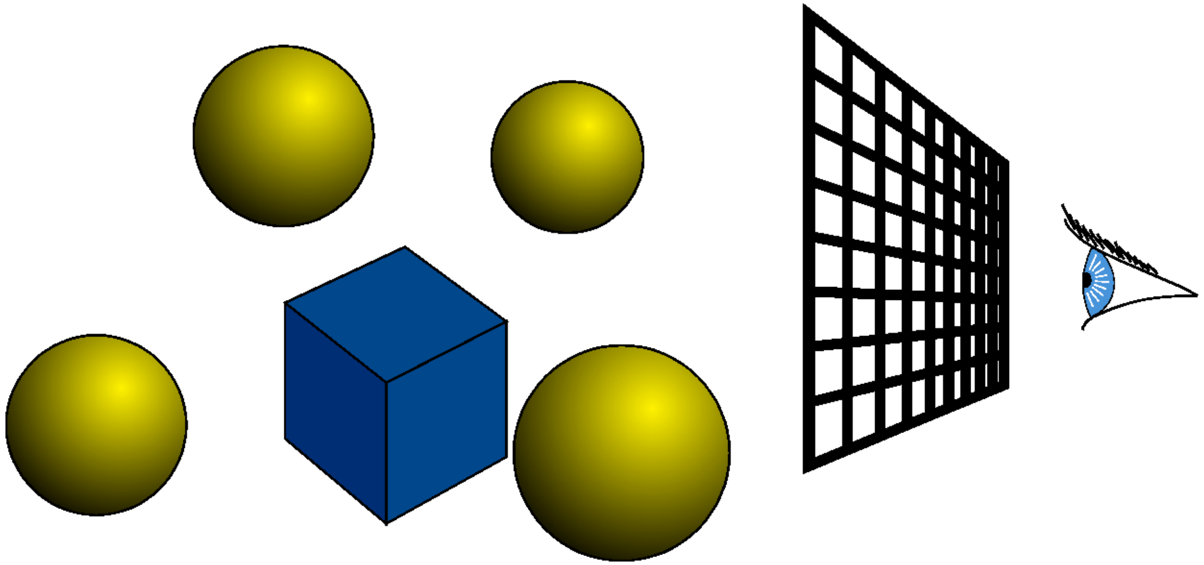
- For each triangle
  - For each projected pixel

Project scene to the pixels



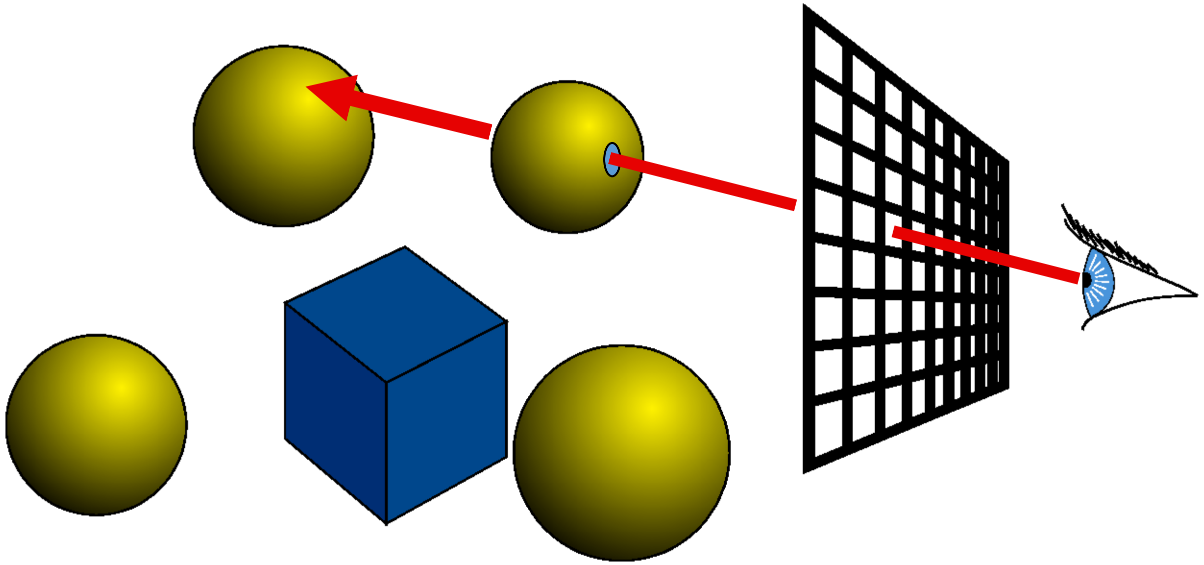
Ray Casting

- For every pixel
  construct a ray from the eye
  - For every object in the scene
    - Find intersection with the ray
    - Keep if closest



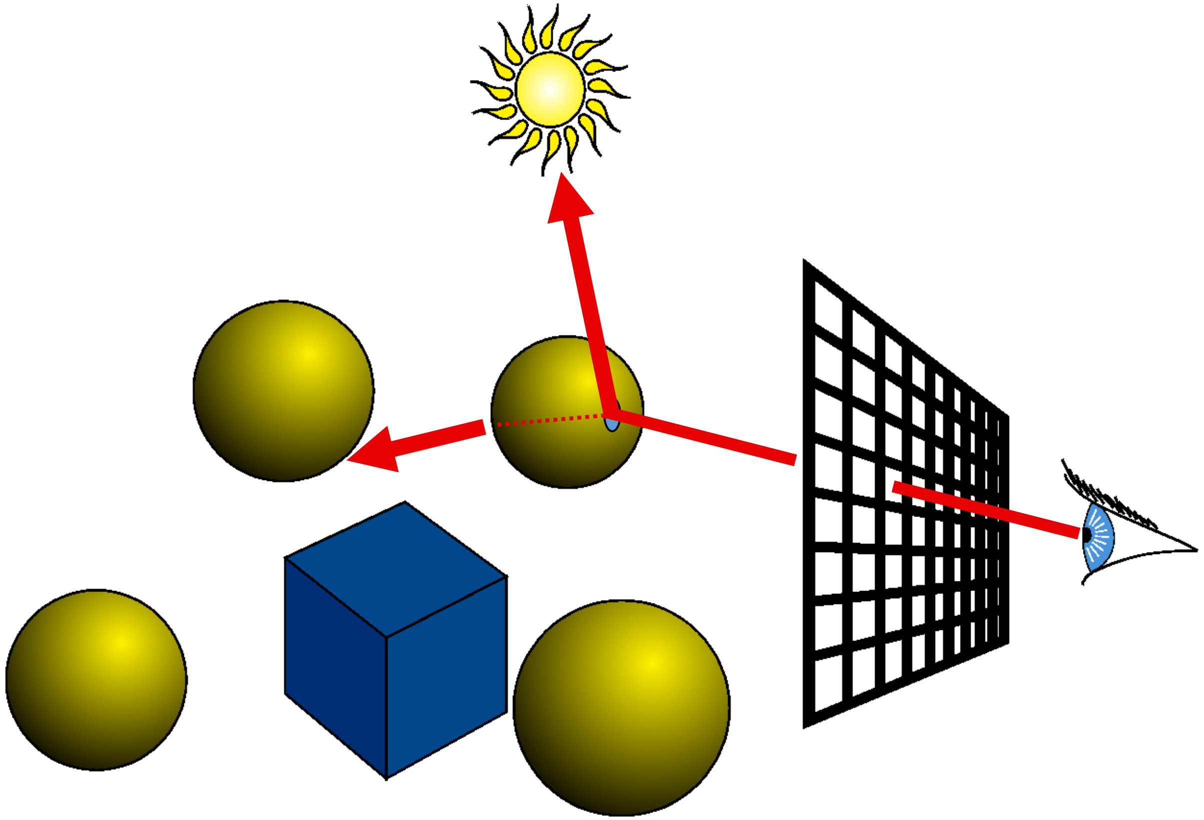
Ray Casting

- For every pixel
  construct a ray from the eye
  - For every object in the scene
    - Find intersection with the ray
    - Keep if closest



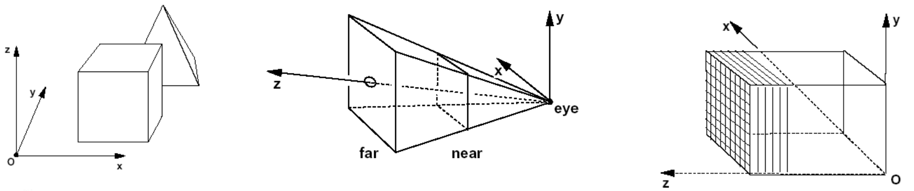
Ray Casting – Ray Tracing

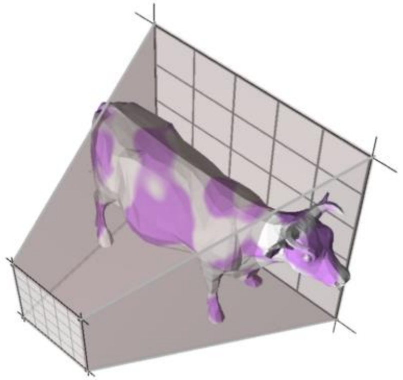
- Shade (interaction of light and material)
- Secondary rays (shadows, reflection, refraction

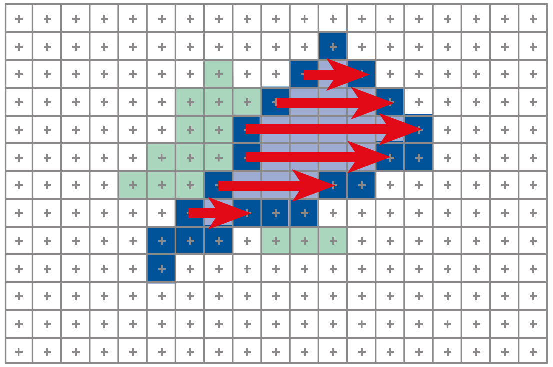


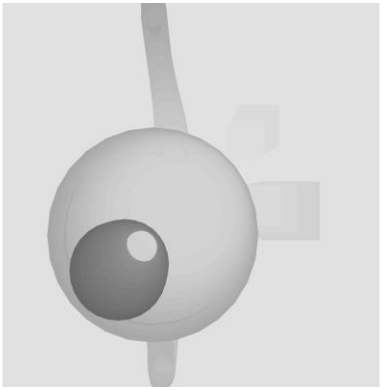
Rendering Pipeline

- Transformations
- Clipping
- Scanning
- Visibility



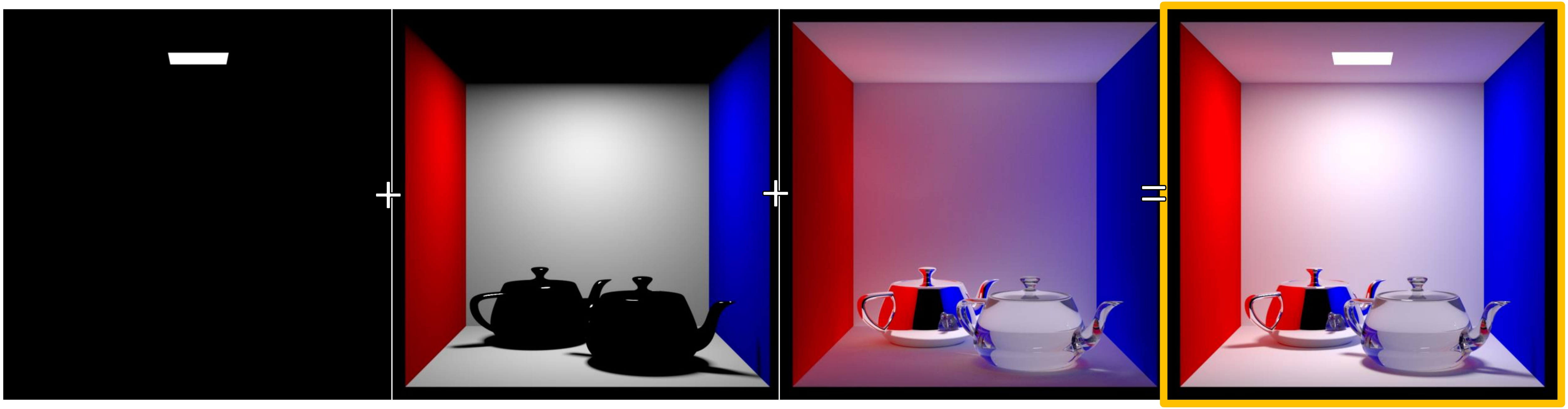






illumination – *maybe the most important part of the process*

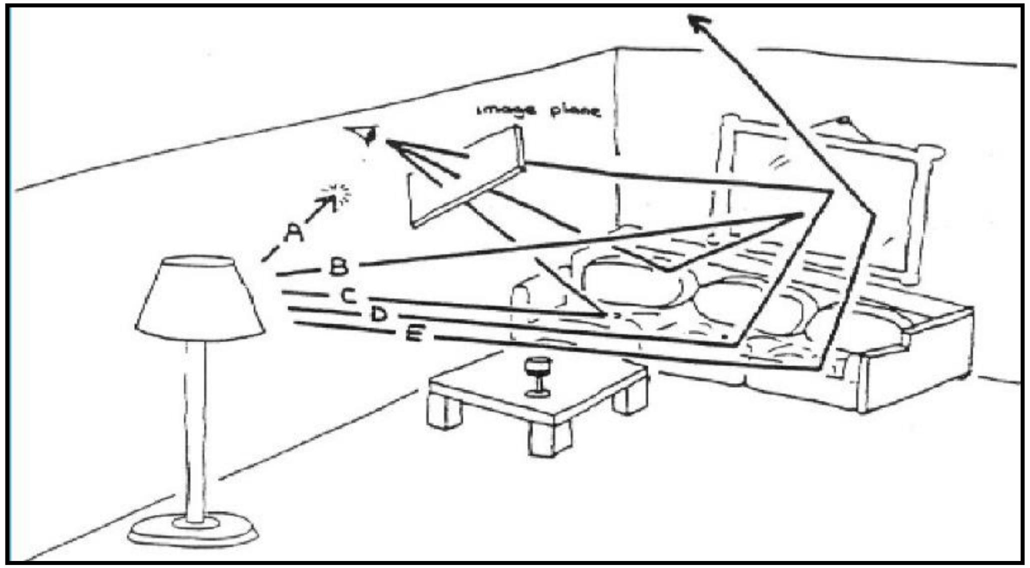
Rendering algorithms split illumination in several parts



Light Source | Direct Illumination | Indirect Illumination | Full Illumination

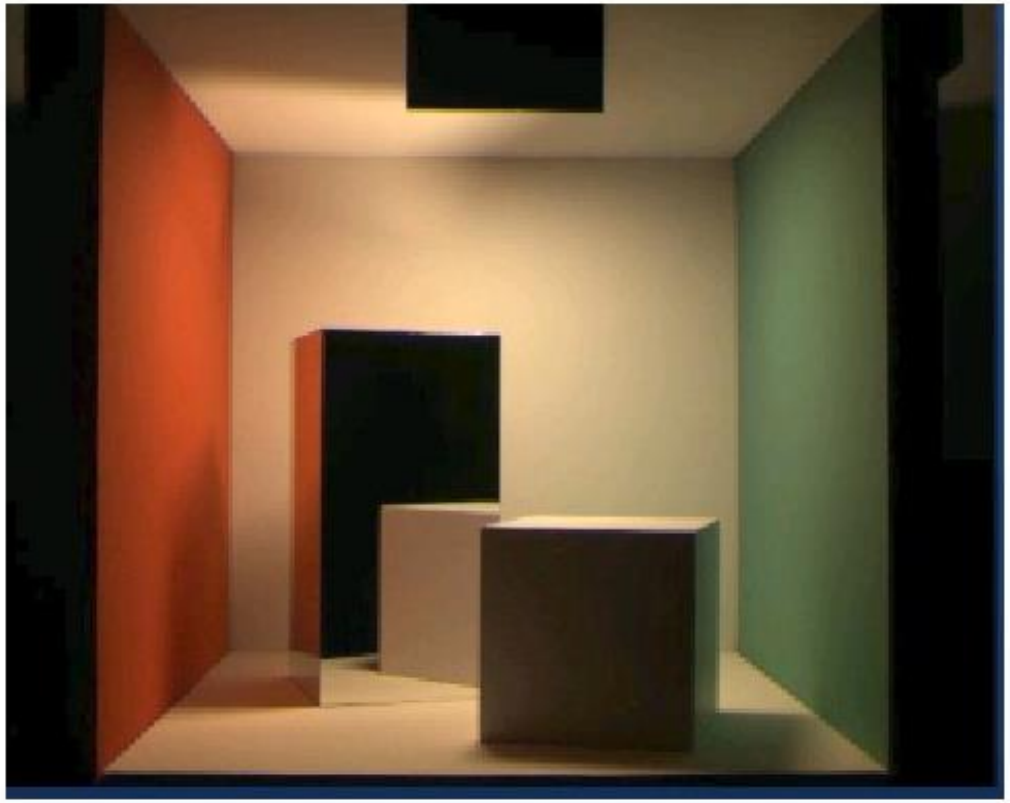
Photorealism

The rendering equation



$$
\begin{aligned}
L(\mathbf{r},\vec{\omega},\lambda,\mathbf{e},t) = \mu(\mathbf{r},\mathbf{s}) \Big[& L^e(\mathbf{s},\vec{\omega},t,\lambda) \\
& + m_p(\vec{\omega}) \int_{-\infty}^{t} d(t-\tau) P_p(\mathbf{s},\lambda) \int_{\Theta_i^i} L(s,\vec{\omega}',\lambda,\mathbf{e},\tau) \cos\theta' \, d\vec{\omega}' \, d\tau \\
& + \int_{\Theta_i^i} f(\mathbf{s},\lambda,\vec{\omega}' \to \vec{\omega}) \int_{\mathcal{R}_\mathcal{V}} P_f(\mathbf{s},\lambda' \to \lambda) L(\mathbf{s},\vec{\omega}',\lambda',\mathbf{e},t) \, d\lambda' \, \cos\theta' \, d\vec{\omega}' \Big] \\
+ \int_0^{h(\mathbf{r},\vec{\omega})} \mu(\mathbf{r},\mathbf{a}) \Big[& L^e(\mathbf{a},\vec{\omega},t,\lambda) \\
& + m_p(\vec{\omega}) \int_{-\infty}^{t} d(t-\tau) P_p(\mathbf{a},\lambda) \int_{\Theta_i^i} L(s,\vec{\omega}',\lambda,\mathbf{e},\tau) \cos\theta' \, d\vec{\omega}' \, d\tau \\
& + \int_{\Theta_i^i} f(\mathbf{a},\lambda,\vec{\omega}' \to \vec{\omega}) \int_{\mathcal{R}_\mathcal{V}} P_f(\mathbf{a},\lambda' \to \lambda) L(\mathbf{a},\vec{\omega}',\lambda',\mathbf{e},t) \, d\lambda' \, \cos\theta' \, d\vec{\omega}' \Big] \, d\alpha
\end{aligned}
$$

Global illumination



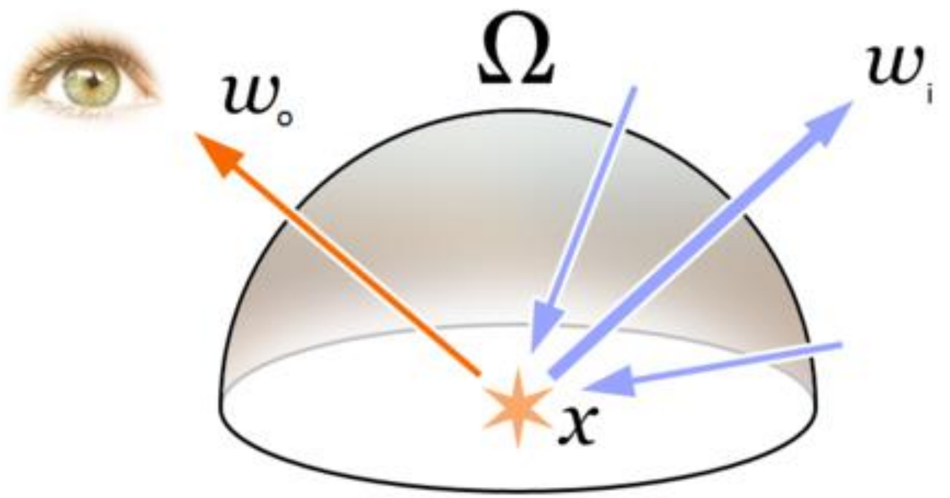
Measured

Simulated

The Rendering Equation

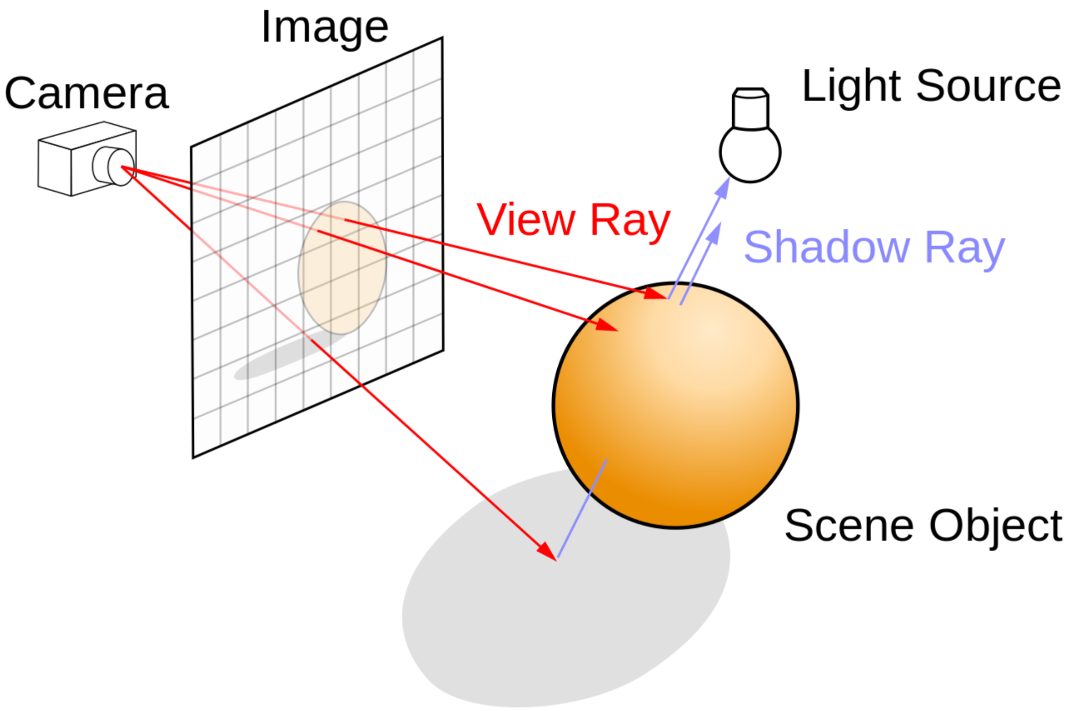
- Rendering methods approximate the following equation:

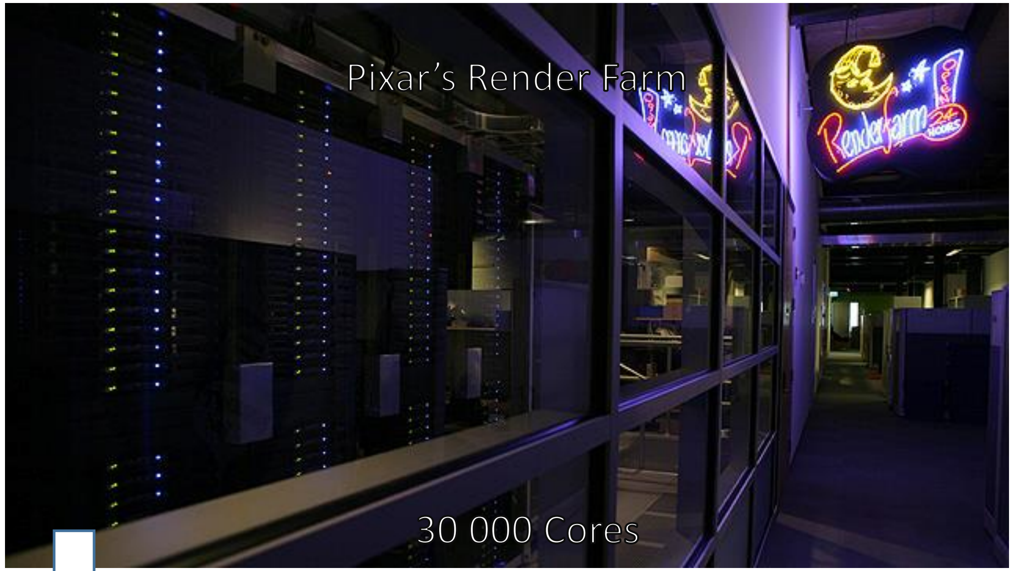
*Outgoing Light = Emitted Light + Reflectance Function * Incoming Light*



Offline Computer Graphics

- Aka, Batch Computer Graphics, for final production-quality video and film (special effects – FX).
- Realistic but computationally expensive
- Typically based on tracing rays of light to the eye/camera



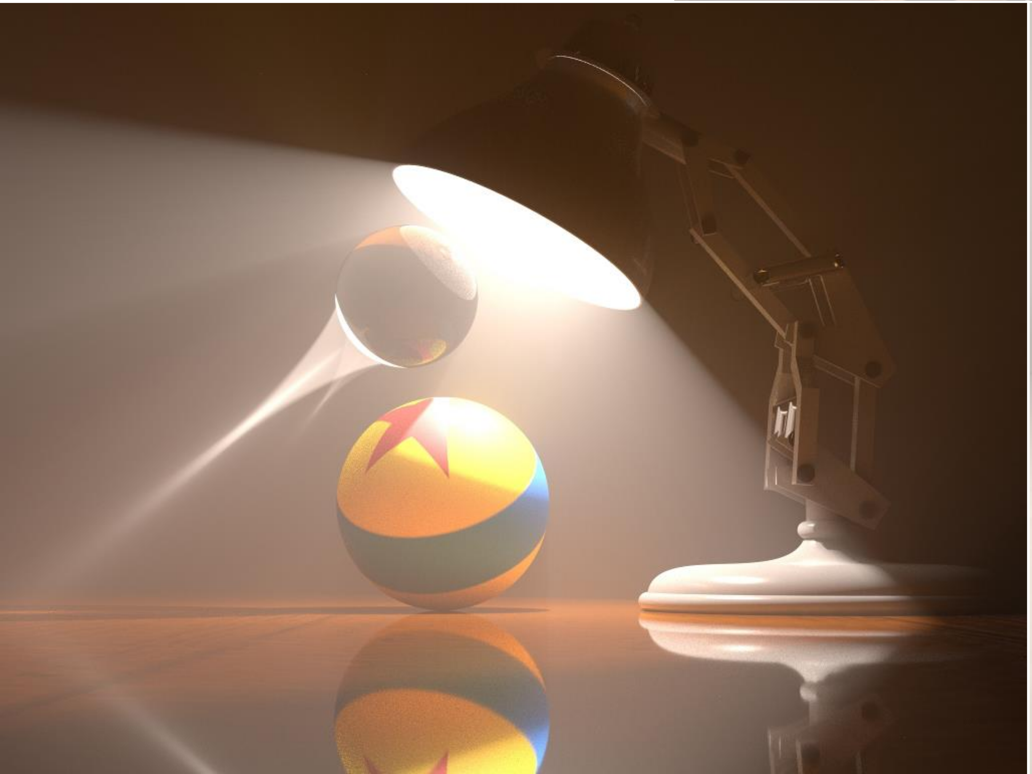


Rendering a single frame of The Good Dinosaur (a 24 fps movie) averaged 48 hours on a 30,000-core render farm!

Graphics Library

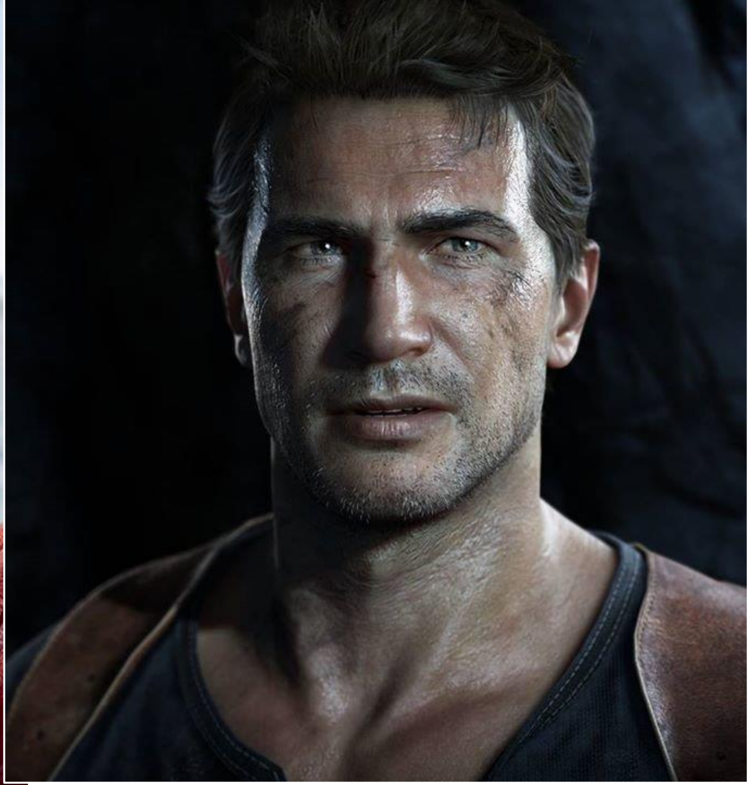
- Examples: OpenGL™, DirectX™, Windows Presentation Foundation™ (WPF), RenderMan™, HTML5 + WebGL™
- Primitives (characters, lines, polygons, meshes,…)
- Attributes
  - Color, line style, material properties for 3D
- Lights
- Transformations
- Immediate mode vs. retained mode
  - immediate mode: no stored 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

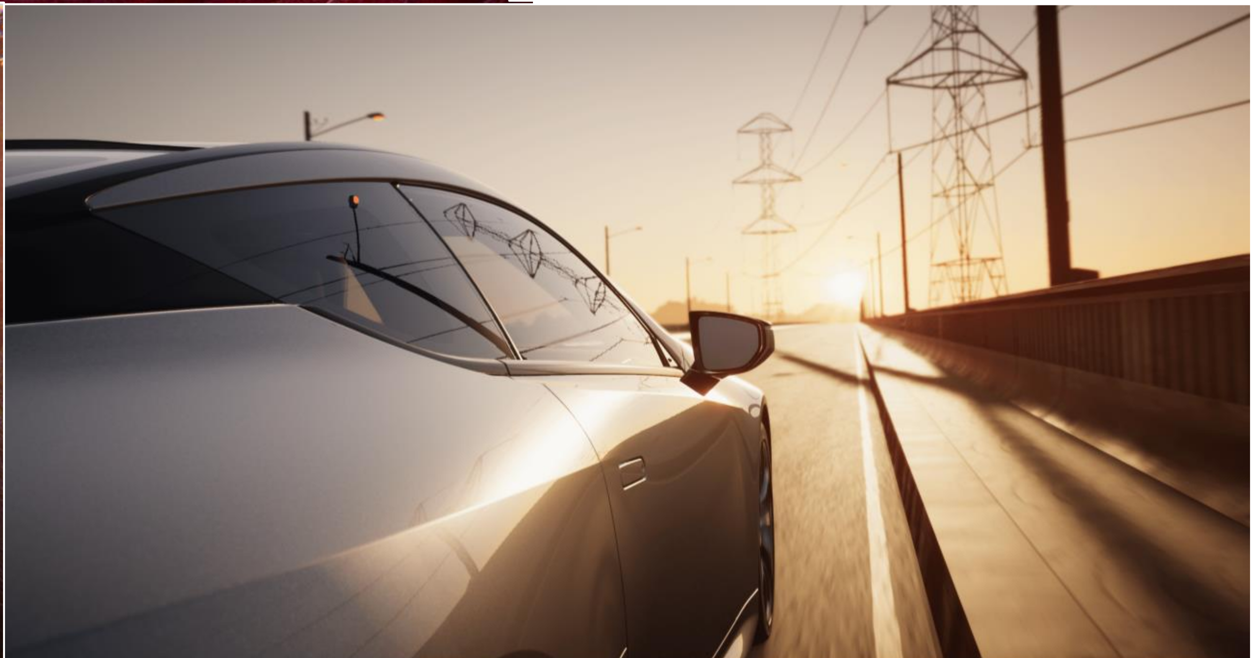
Some eye candy



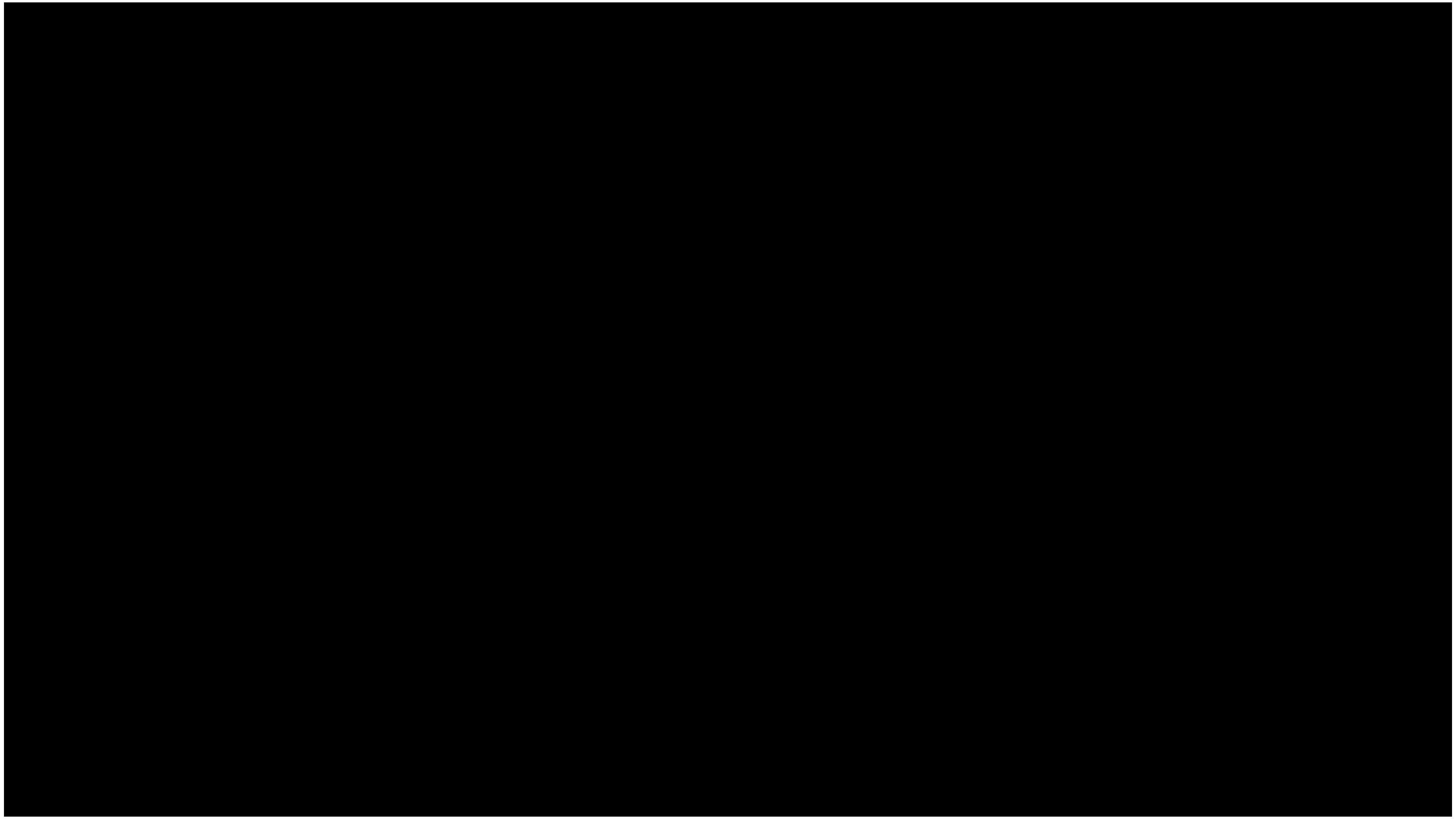
From: https://www.nvidia.com/en-us/design-visualization/iray/features/







Photorealistic Rendering



Computer Graphics: *Introduction*

- **Computer Graphics**
  - What's It All About?
  - Application Areas
  - Interactive Computer Graphics

- **How do graphics work**
  - Modeling
  - Animation
  - Rendering

- **Course Outline**
  - Lectures, Exams, Evaluation
  - Books
  - Miscellanea

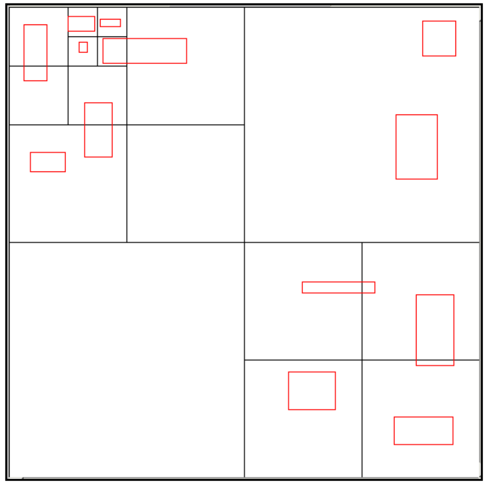
What we will learn at CS426 (ΕΠΛ426)

- The basic principles and algorithms of computer graphics.
- Enough practical knowledge so that we can implement most of the above.

  - **Geometrical Modelling – Basic knowledge**
  - **Animation – Quick Introduction**
  - **Rendering – In depth knowledge**

What we will learn at CS426 (ΕΠΛ426)

- Camera definition
- Modelling
  - Polygons, polyhedron
  - Scene graph
- Rendering Pipeline
  - crop, hide, scan, shadows…
- Global illumination with Radiosity
- Acceleration methods
  - Acceleration data structures
- Ray tracing
- GPU programming
- OpenGL & WebGL & Unity3D game engine
- Introduction to animation



hierarchical data structures

Τι ΔΕΝ θα μάθουμε στο ΕΠΛ426

- Tools for 2D image processing
  - Photoshop and other painting tools
- Artistic skills
- Game design

Prerequisites:

- Good programming skills
  - We will use C/C++ for lab assignments (Javascript για WebGL)
  - CS132 (ΕΠΛ 132), CS232 (ΕΠΛ 232)

- Geometric Algebra
  - Vectors, matrices, Θα χρειαστεί να ξέρουμε για διανύσματα, πίνακες, system of linear equations, basic knowledge on geometry
  - During the first two weeks we will cover the basics to refresh your memory.

Evaluation

- Find the Syllabus on the website:

  http://www.cs.ucy.ac.cy/courses/EPL426/

- The website will be updated frequently.
  - Notes will be available the previous day of each lecture.

- Evaluation:

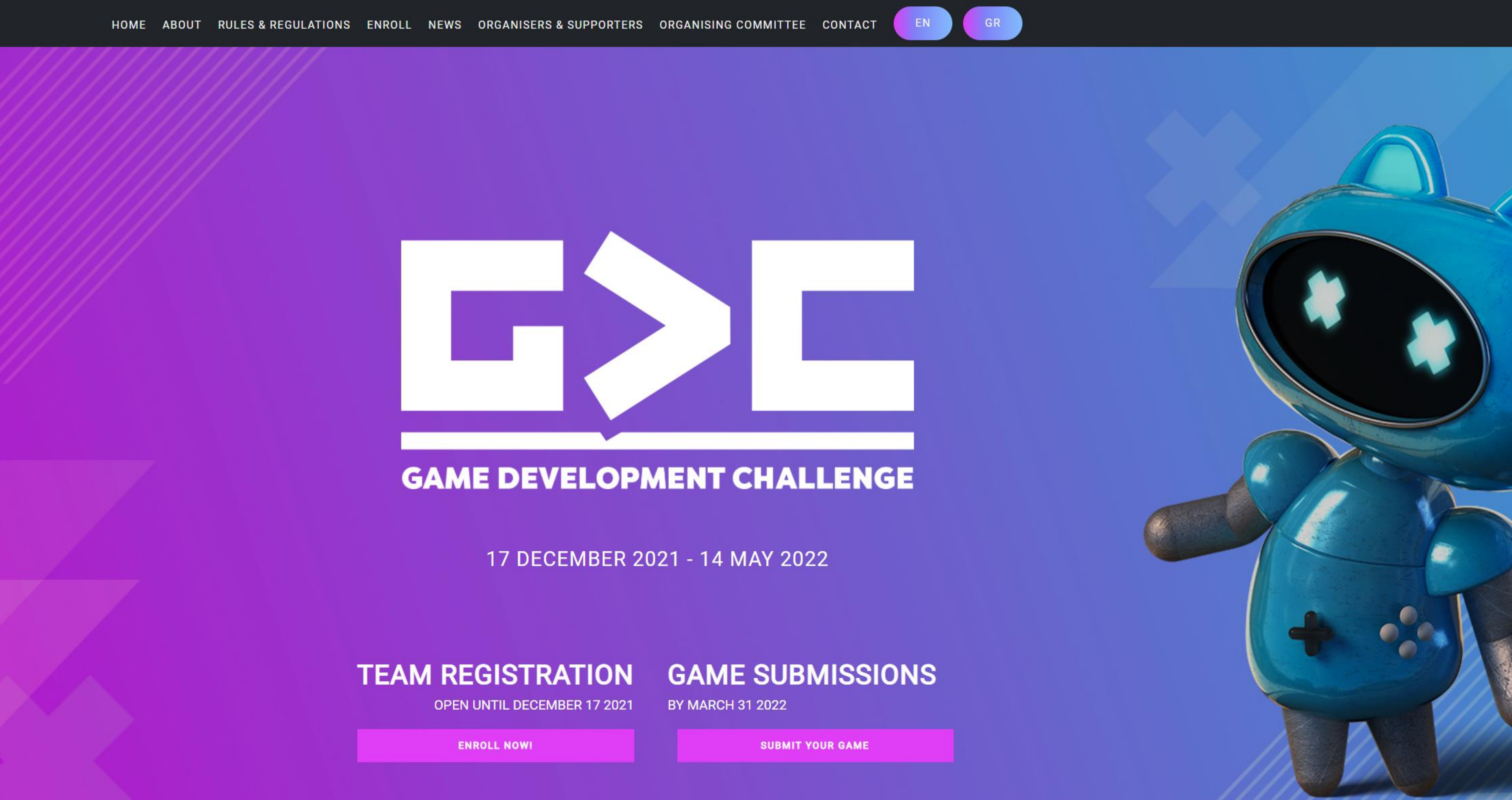
Evaluation	Weight
Assignments	40%
Mid-term exams	20%
Final exams	40%
Logipaignion	Up to 10%

Our philosophy

- **We want active students in the class:** you are adviced to be present in all of the lectures and participate:
  - Cameras should be enabled and I will note your attendance.
  - Lectures will NOT be recorded.
- **Required assignments.**
- **Required examinations.**
- Reasonable and lenient assessment (at least that is what the examiner believes)

Practical part

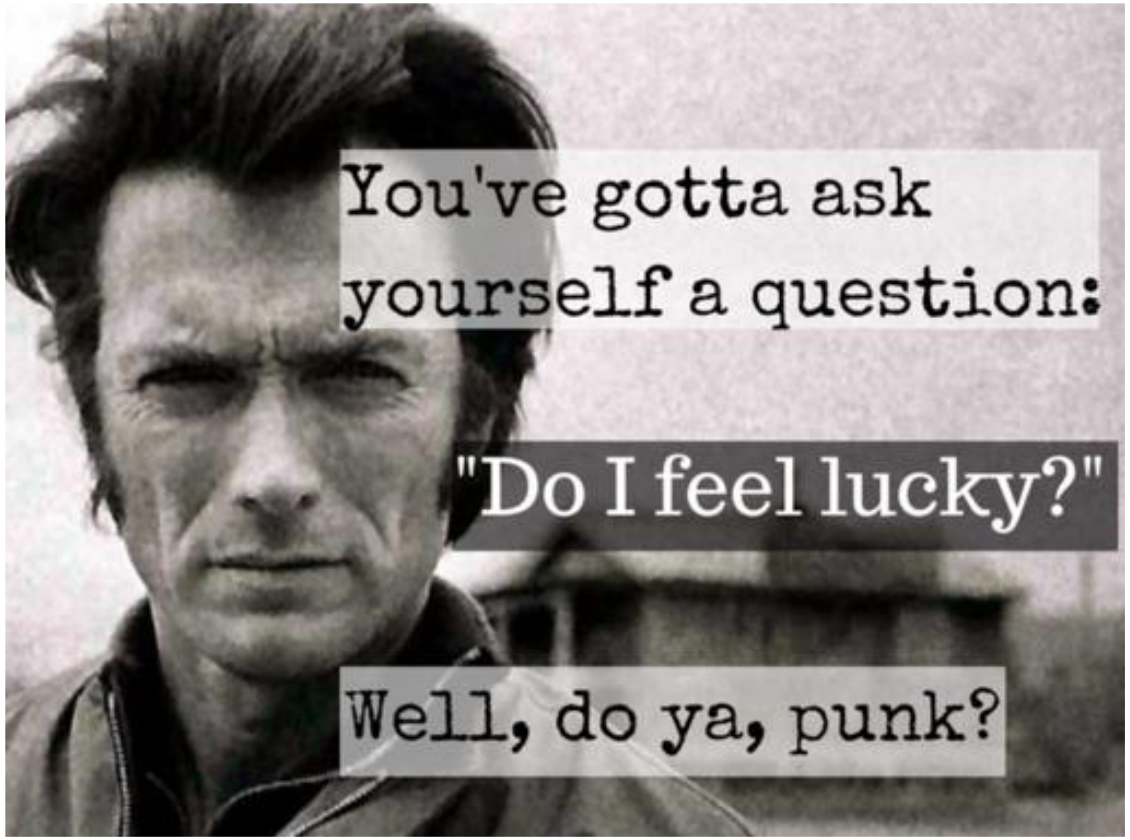
- A big part of the final grade will be given by the practical work:
  - Various small **exercises/assignments** for implementing concepts that we will learn in theory
- **Lab assignment** in openGL / webGL (15%)
- **Final project** in groups of 2 people for the implementation of an "impressive" application (25%)


HOME ABOUT RULES & REGULATIONS ENROLL NEWS ORGANISERS & SUPPORTERS ORGANISING COMMITTEE CONTACT EN GR
GAME DEVELOPMENT CHALLENGE
17 DECEMBER 2021 - 14 MAY 2022
TEAM REGISTRATION
OPEN UNTIL DECEMBER 17 2021
ENROLL NOW!
GAME SUBMISSIONS
BY MARCH 31 2022
SUBMIT YOUR GAME

Late hand-in policy

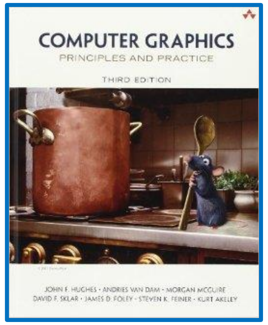
- Programming assignments
  - No late assignments will be accepted!
  - No extensions will be given!

Cheating Policy

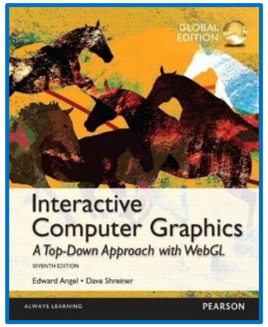


- Let’s keep it simple: if you are caught cheating, you will get a **zero** for the entire course (not just the assignment).

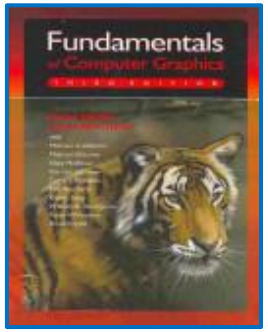
Books



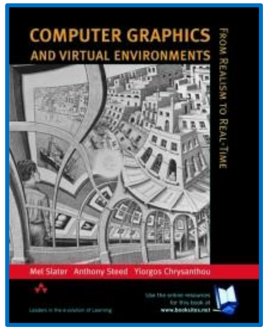
- “Computer Graphics: Principles and Practice”, J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, K. Akeley, Addison-Wesley, 2013
  Great for really in-depth theory



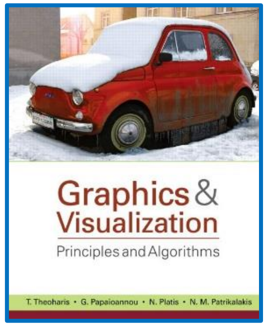
- “Interactive Computer Graphics with WebGL, E. Angel and D. Shreiner, Pearson; 7th edition, ISBN-13: 978-1292019345. 2014.



- “Fundamentals of Computer Graphics”, P. Shirley, M.l Ashikhmin, M. Gleicher, W. B. Thompson, P. Willemsen, E. Reinhard, S. R. Marschner, K. Sung, Taylor & Francis, 2009



- Computer Graphics and Virtual Environments: From Realism to Real-Time, M. Slater, A. Steed and Y. Chrysanthou, Addison Wesley publishers, ISBN 0-201-62420-6, 2001.



- “Graphics and Visualization: Principles & Algorithm”, T. Theoharis, G. Papaioannou, N. Platis, N. M. Patrikalakis, A K Peters, 2007.

We have no textbook for this class —the lecture slides are the primary course reference

Next Lecture

- Next time, we'll do a math review & preview
  - Linear algebra, vector calculus
  - Help make the rest of the course easier!

Thank you!