

# Introduction to Interactive Computer Graphics

Dr. V Masilamani

[masila@iiitdm.ac.in](mailto:masila@iiitdm.ac.in)

Department of Computer Science and Engineering  
IIITDM Kancheepuram  
Chennai-127



Welcome

Course Logistics

What?

Why?

Applications

CG vs DIP vs CV

Pipeline

Modeling and Display

History of Computer Graphics

Computer Graphics in Industry

Acknowledgements



# Click!



► For Theory(COM524T)

- 2 Quizes; each one for 15 marks
- End sem: 40 marks
- Project: 20 marks
- Seminar: 10 marks

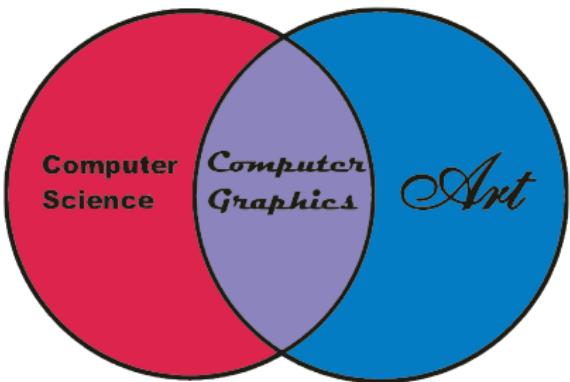
► For Practice(COM524P)

- Regular Assignments: 50 marks
- Midsem: 20 marks
- Endsem: 30 marks

# What is Computer Graphics?



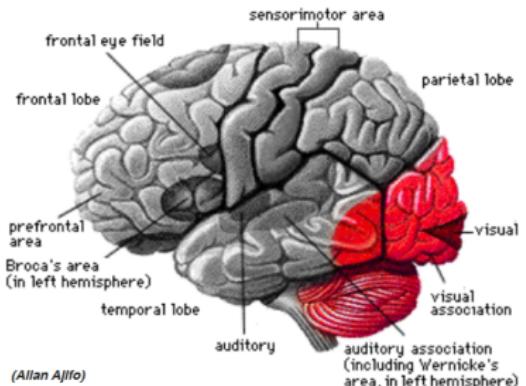
- ▶ **Def:** Synthesizing and displaying image/video using computer
- ▶ Synthesizing Image is termed as Rendering Image



# Why Computer Graphics?



- ▶ About 30% of brain dedicated to visual processing.
- ▶ Eyes are highest-bandwidth port into the head!



# Why Computer Graphics? (cont.)



- ▶ The CGI can cost from a fifth up to a half of the overall budget, that is to say between 40 and over 100 million dollars.

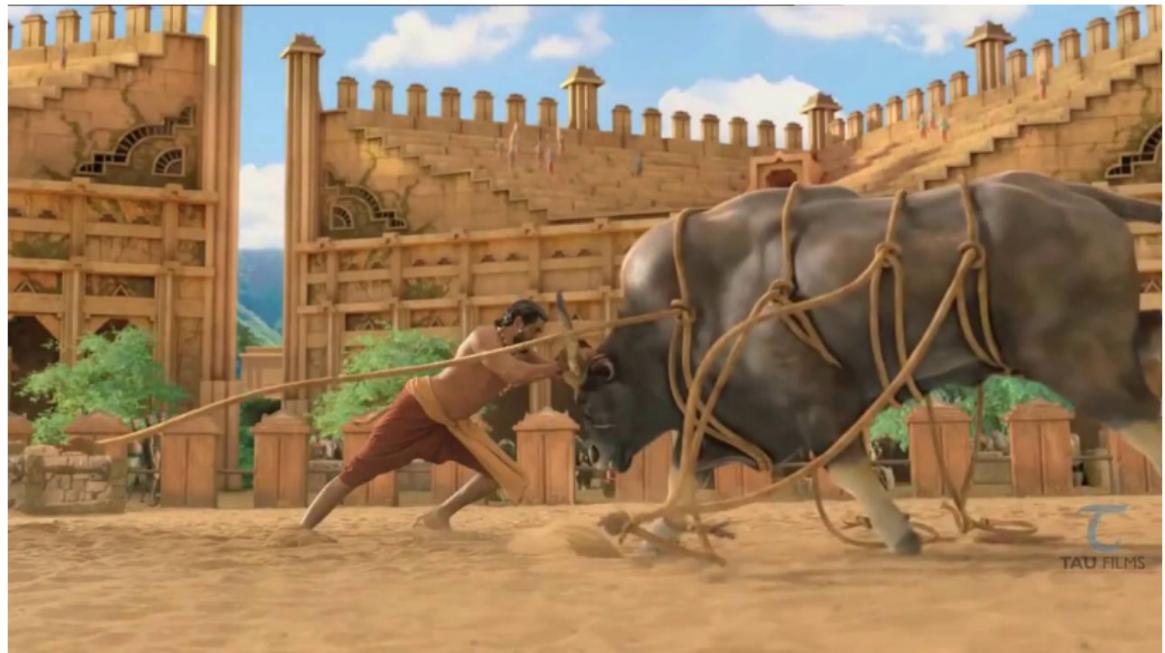




Figure 1: Robot 2.0 (2018)

# MOVIES (cont.)



Figure 2: Avengers(2018)

# MOVIES (cont.)



Figure 3: Avatar(2009)

# MOVIES (cont.)



Figure 4: Jurrasic Park(1993)



Figure 5: PUBG

# Games (cont.)

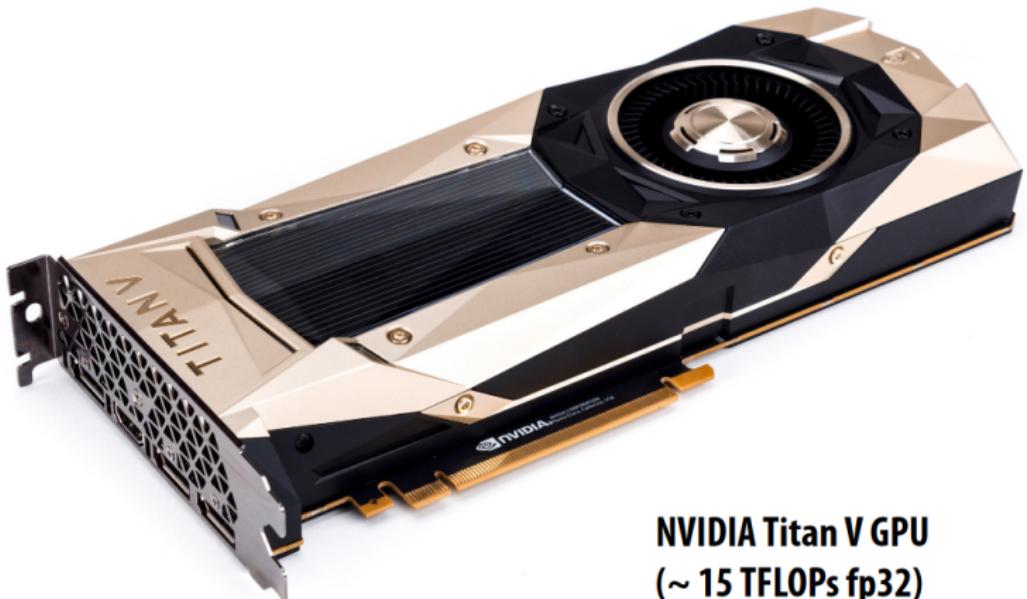


Figure 6: Call Of Duty

# Super Computing for Games



- ▶ For graphics intensive tasks such as video games and video editing, GPU is useful



# Augmented Reality



- ▶ AR superimposes a computer-generated image on a user's view of the real world, thus providing a composite view.
- ▶ By looking through the screens of mobile devices, such as smartphones and tablets, we can see things that are not really there.

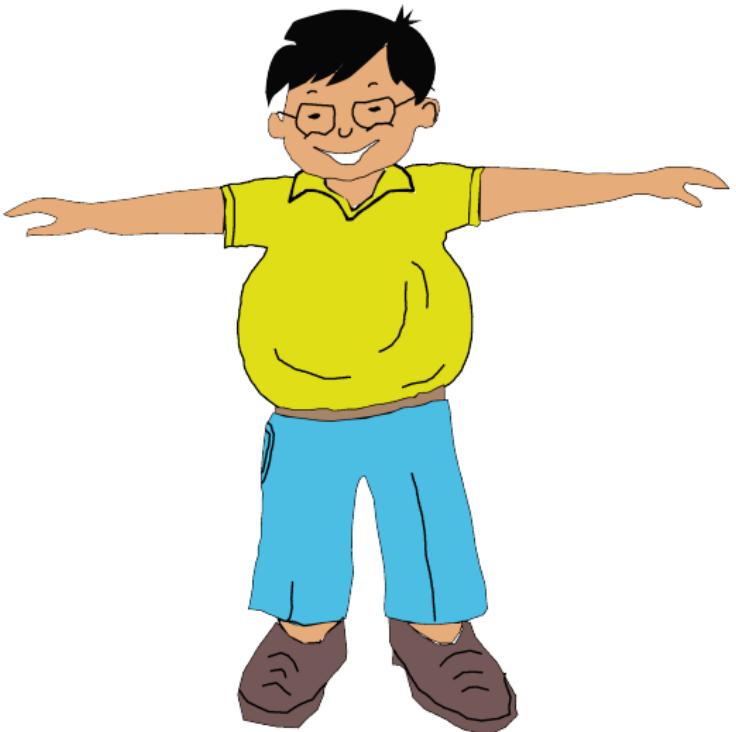


Figure 7: Microsoft Hololens augmented reality headset concept

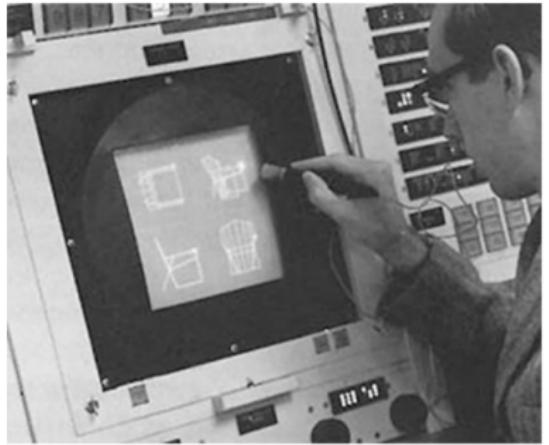
# Virtual Reality



# Digital Illustrations



# Graphical User Interfaces



Ivan Sutherland, "Sketchpad" (1963)



Doug Engelbart  
Mouse

# Modern graphical user interfaces



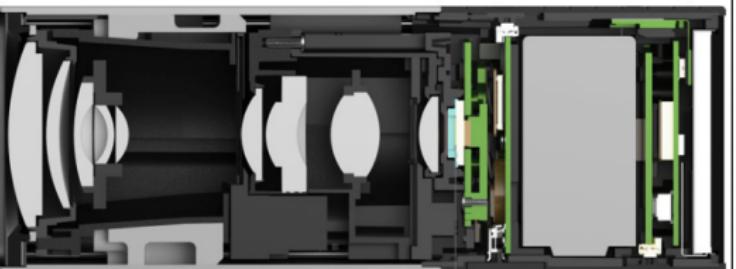
- ▶ 2D drawing and animation are ubiquitous in computing.  
Typography, icons, images, transitions, transparency, ... (all rendered at high frame rate for rich experience)



# Computational cameras

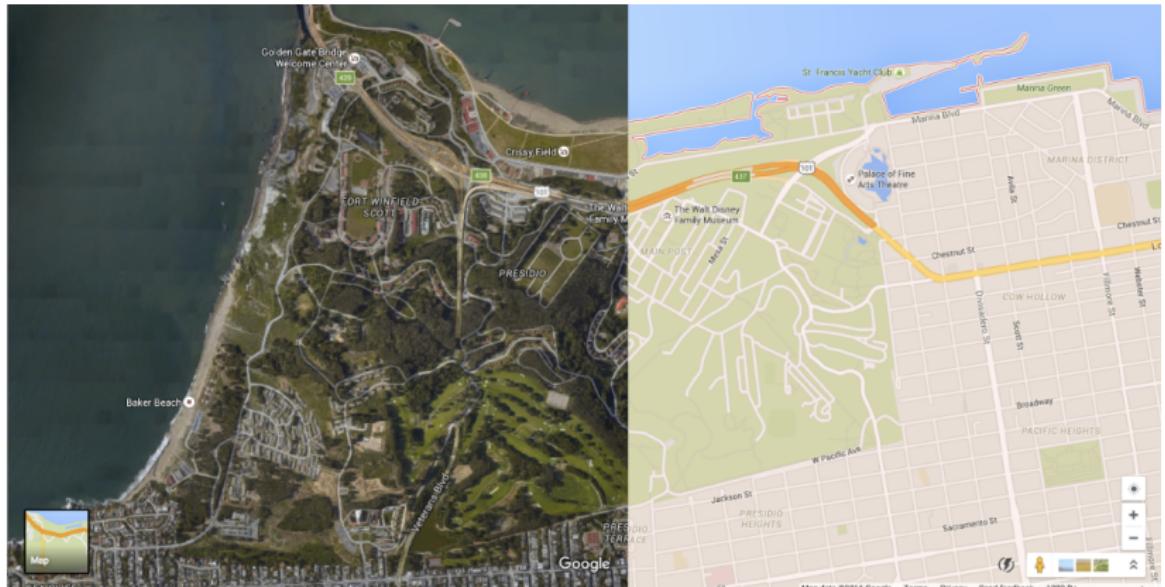


- ▶ Panoramic stitching, HDR photos, light field cameras, ...



# Imaging for mapping

- ▶ Maps, satellite imagery, street-level imaging,...



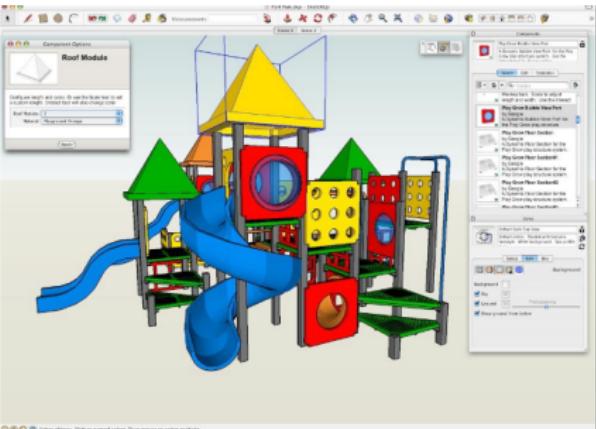
# Computer aided design



- ▶ For mechanical, architectural, electronic, optical, ...



SolidWorks



SketchUp

# Architectural design

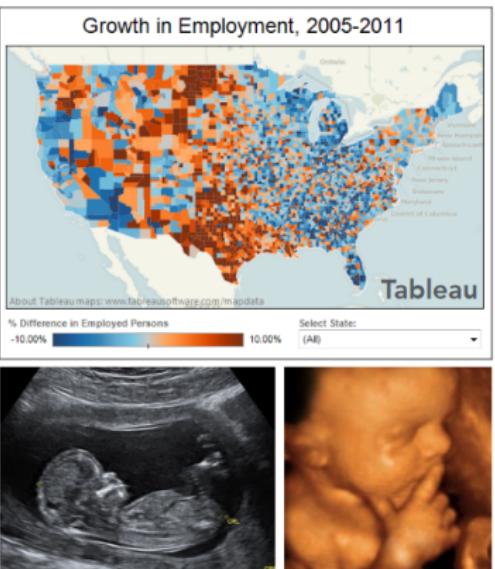
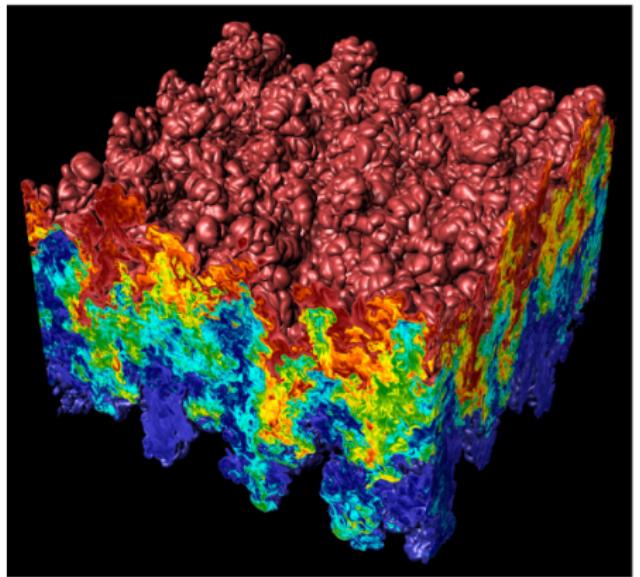


Figure 9: Bilbao Guggenheim, Frank Gehry

# Visualization



- ▶ Science, engineering, medicine, journalism, ...



# Simulation



**Driving simulator**  
**Toyota Higashifuji Technical Center**



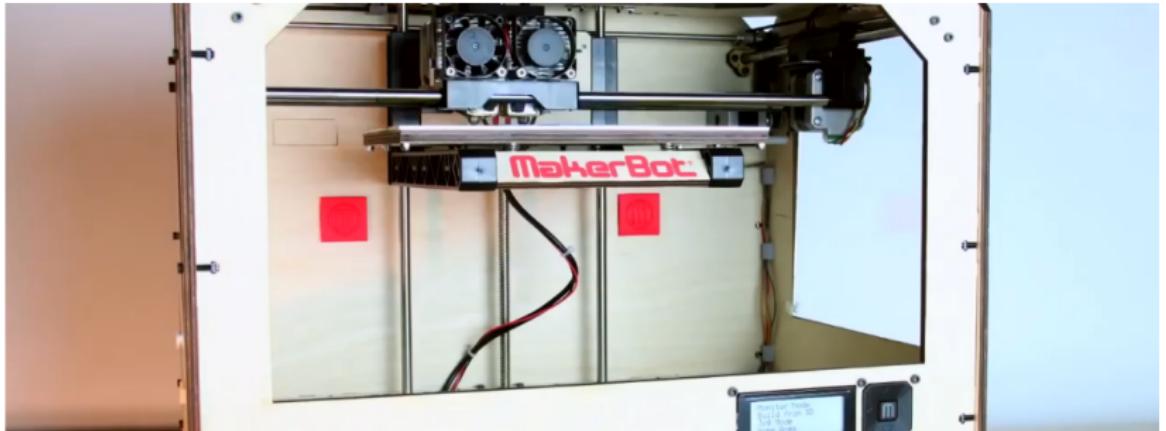
**da Vinci surgical robot**  
**Intuitive Surgical**

# Simulation for training models



Figure 10: autonomous driving simulator

# 3D Fabrication



- ▶ The porn industry replaces the faces in the porn videos by celebrity faces to have better business by using a software called DeepFakes which creates the fake face which are hard to differentiate.



(a)



(b)

Figure 11: Deepfakes examples

# Generative Adversarial Networks



<https://thispersondoesnotexist.com/>





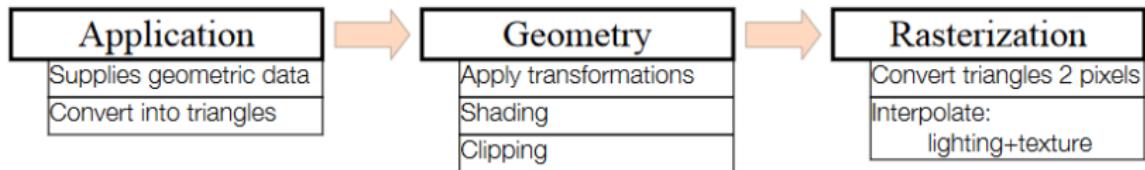
- ▶ All these applications demand sophisticated theory and systems
- ▶ Science and mathematics
  - Physics of light, color, optics
  - Math of curves, surfaces, geometry, perspective, ...
  - Sampling
- ▶ Systems
  - Parallel, heterogeneous processing
  - Graphics-specific programming systems
  - Input/output devices
- ▶ Art and psychology
  - Perception: color, stereo, motion, image quality, ...
  - Art and design: composition, form, lighting, ...

# Computer Graphics vs DIP vs Computer Vision



- ▶ Digital Image Processing takes images as input and outputs processed image or feature/information in the image
- ▶ Computer Graphics deals with generating or synthesising images, but does not consider camera captured images
- ▶ Computer Vision deals with analyzing images to get information about the scene which was imaged.
  - Computer Vision is often seen as part of Image Processing (High level Image Processing)

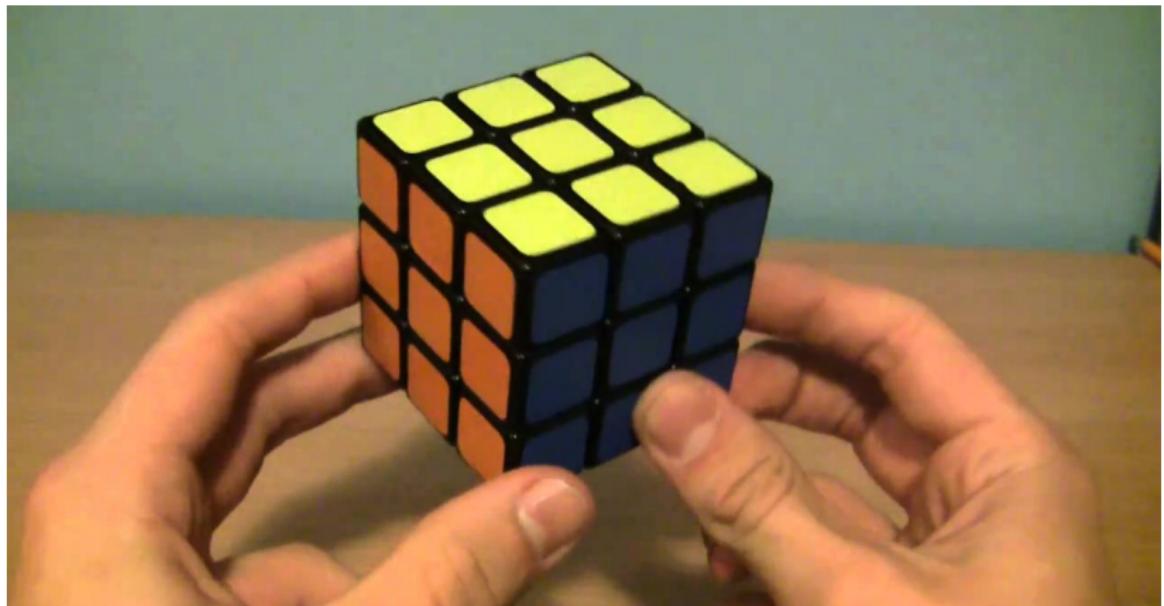
- ▶ Geometry + Transformations
- ▶ Cameras and viewing
- ▶ Lighting and shading
- ▶ Rasterization
- ▶ Texturing



# Modeling and drawing a cube



- ▶ **Goal:** generate a realistic drawing of a cube
- ▶ Key questions:
  - **Modeling:** how do we describe the cube?
  - **Rendering:** how do we then visualize this model?



# Modeling the cube



- ▶ Suppose our cube is...
  - centered at the origin  $(0,0,0)$
  - has dimensions  $2 \times 2 \times 2$
- ▶ QUESTION: What are the coordinates of the cube vertices?
  - A:  $(1, 1, 1)$  E:  $(1, 1, -1)$
  - B:  $(-1, 1, 1)$  F:  $(-1, 1, -1)$
  - C:  $(1, -1, 1)$  G:  $(1, -1, -1)$
  - D:  $(-1, -1, 1)$  H:  $(-1, -1, -1)$
- ▶ QUESTION: What about the edges?
  - AB, CD, EF, GH,
  - AC, BD, EG, FH,
  - AE, CG, BF, DH

# Drawing the cube

- ▶ Now have a digital description of the cube:

Vertices

A: ( 1, 1, 1 ) E: ( 1, 1,-1 )  
B: (-1, 1, 1 ) F: (-1, 1,-1 )  
C: ( 1,-1, 1 ) G: ( 1,-1,-1 )  
D: (-1,-1, 1 ) H: (-1,-1,-1 )

Edges

AB, CD, EF, GH,  
AC, BD, EG, FH,  
AE, CG, BF, DH

- ▶ How do we draw this 3D cube as a 2D (flat) image?

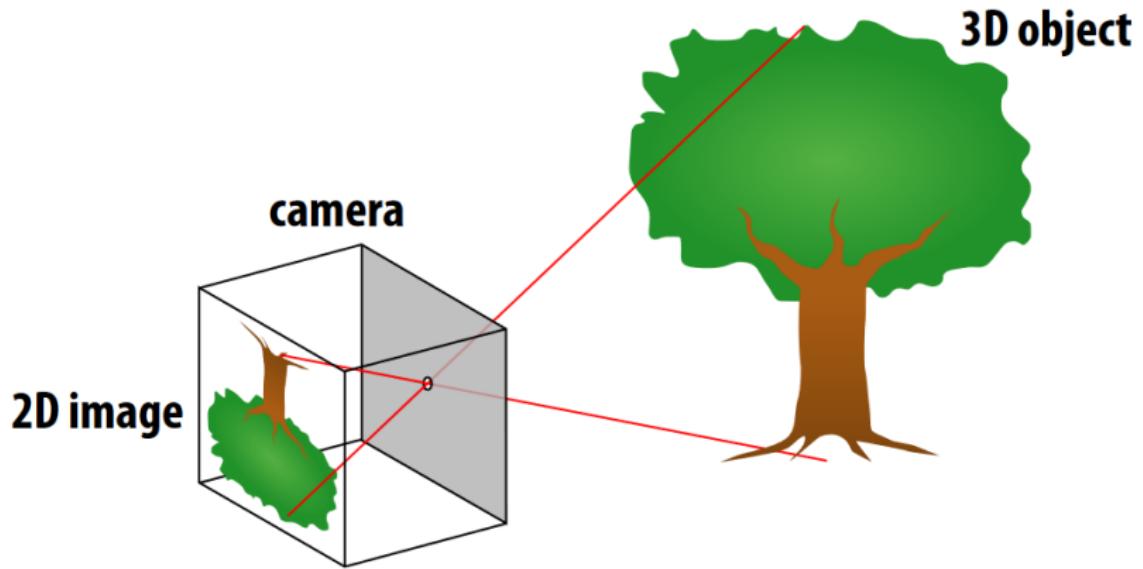
- ▶ Basic strategy:

1. Project 3D vertices to 2D points in the image
2. Connect 2D points with straight lines

# Perspective projection

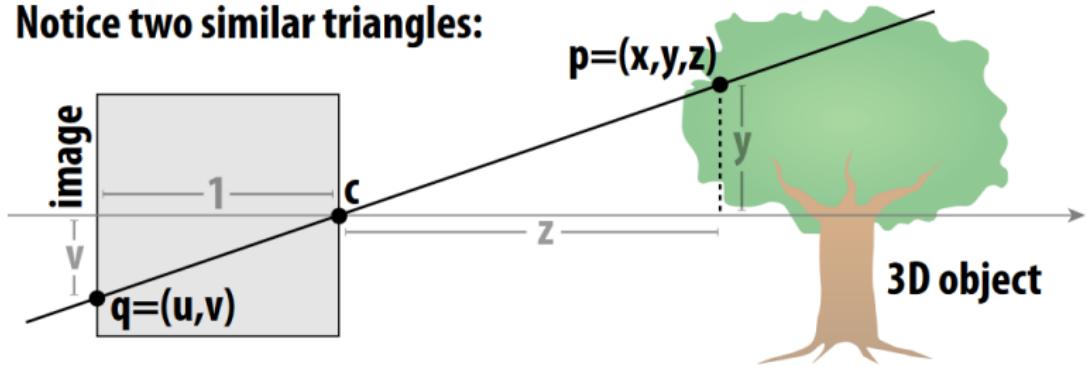


- ▶ Objects look smaller as they get further away ("perspective")
- ▶ Why does this happen?
- ▶ Consider simple ("pinhole") model of a camera:



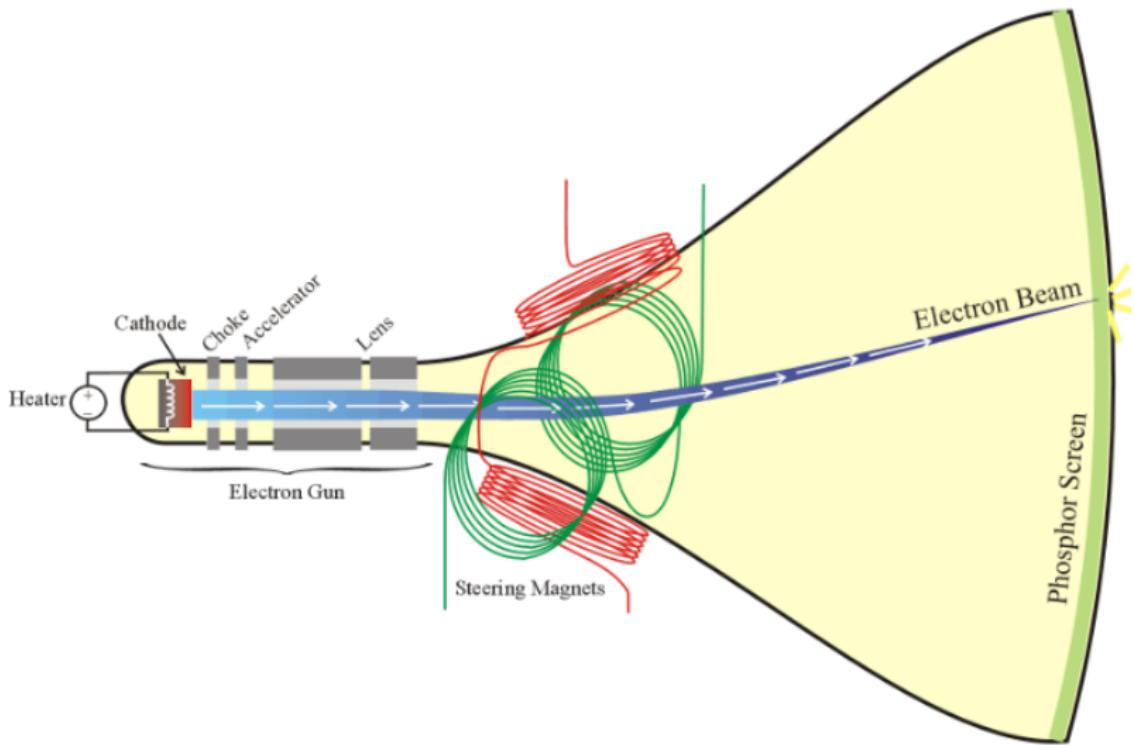
- ▶ Where exactly does a point  $p = (x, y, z)$  end up on the image?
- ▶ Let's call the image point  $q = (u, v)$

**Notice two similar triangles:**



- ▶ Assume camera has unit size, coordinates relative to pinhole  $c$
- ▶ Then  $\frac{v}{1} = \frac{y}{z}$ , similarly  $u = \frac{x}{z}$

# Displays: Cathod Ray Tube(CRT)



# Frame buffer: memory for a raster display

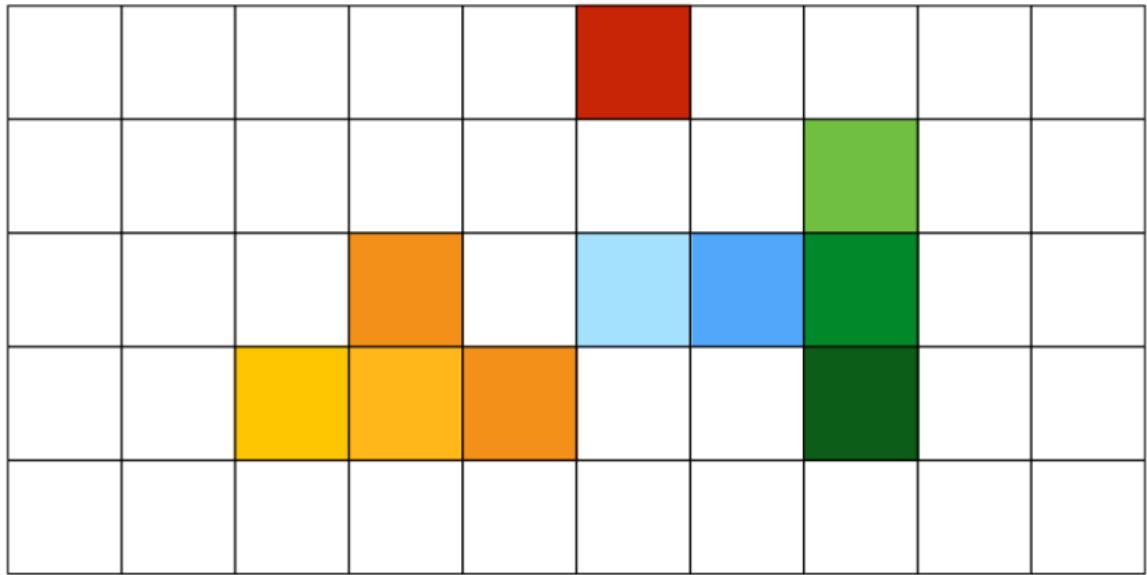


**image = “2D array of colors”**

# Output for a raster display



- ▶ Common abstraction of a raster display:
  - Image represented as a 2D grid of “pixels”
  - Each pixel can take on a unique color value



# Flat panel displays

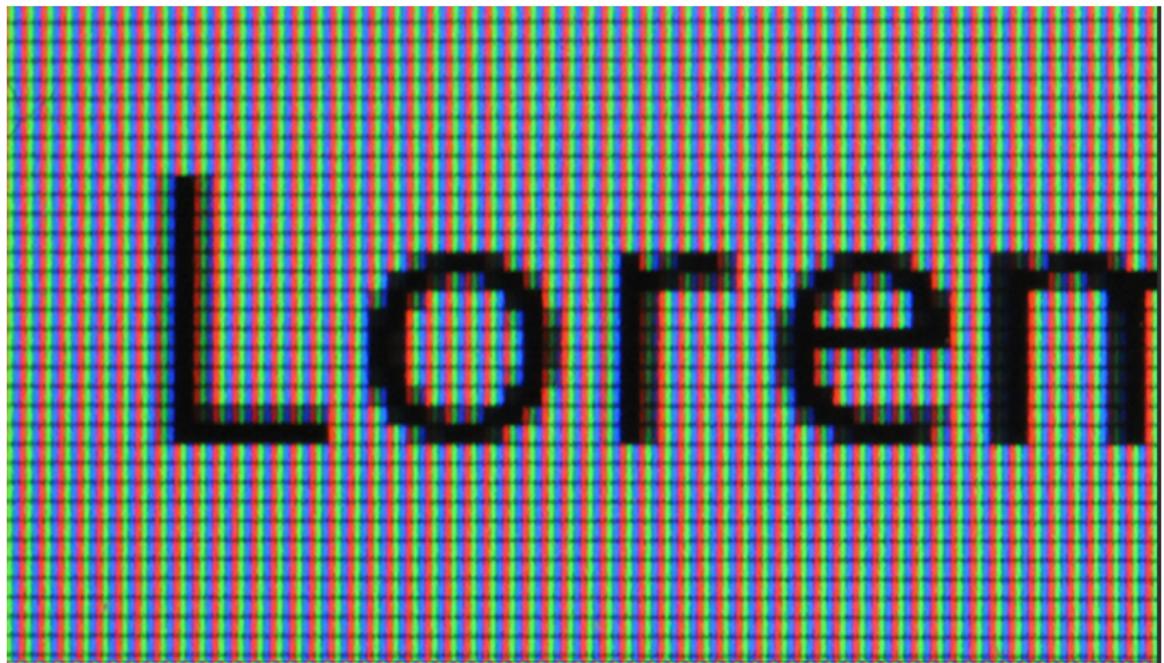


Low-Res LCD Display

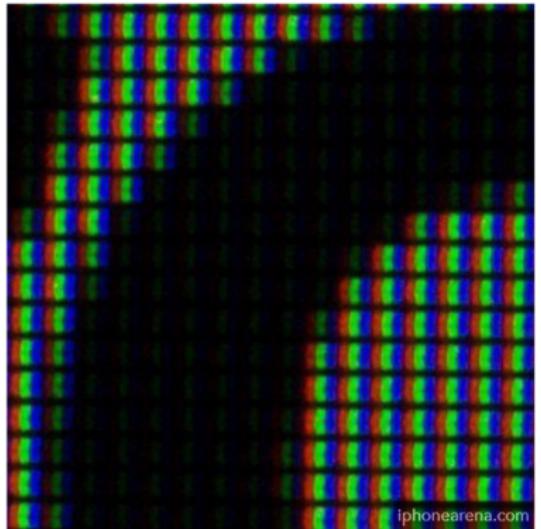


High resolution color LCD, OLED, ...

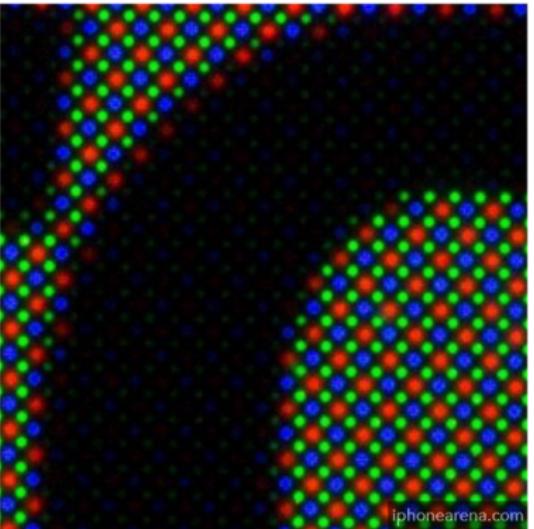
# Close up photo of pixels on a modern display



# LCD screen pixels (closeup)



**iPhone 6S**

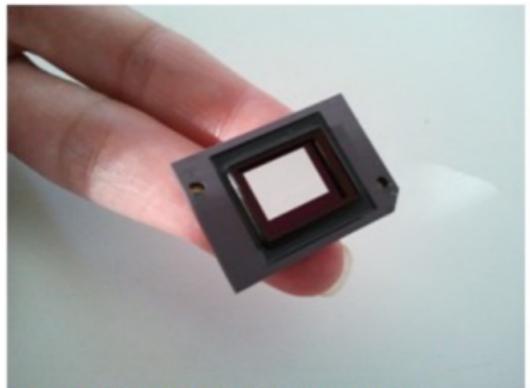


**Galaxy S5**

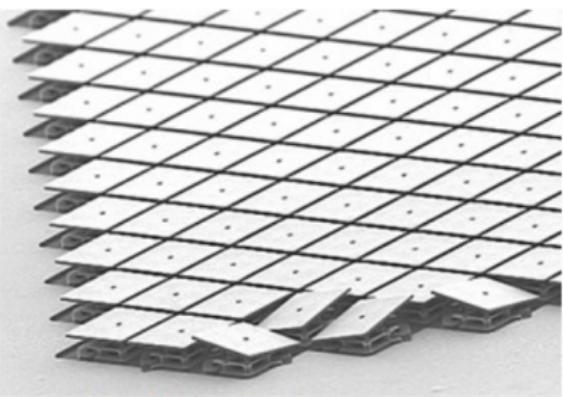
# DMD projection display



- ▶ Array of micro-mirror pixels
- ▶ DMD = Digital micro-mirror device



DIGITAL MICRO MIRROR DEVICE (**DMD**)  
(**SLM** - Spatial Light Modulator)



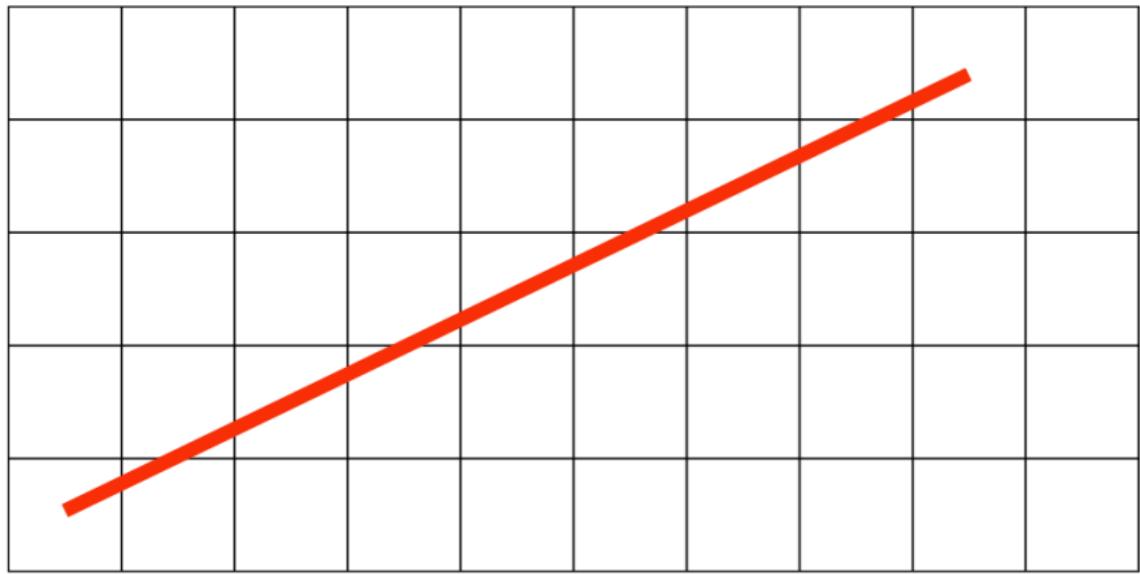
MICRO MIRRORS CLOSE UP

[Y.K. Rabinowitz; EKB Technologies

# What pixels should we color in to depict a line?



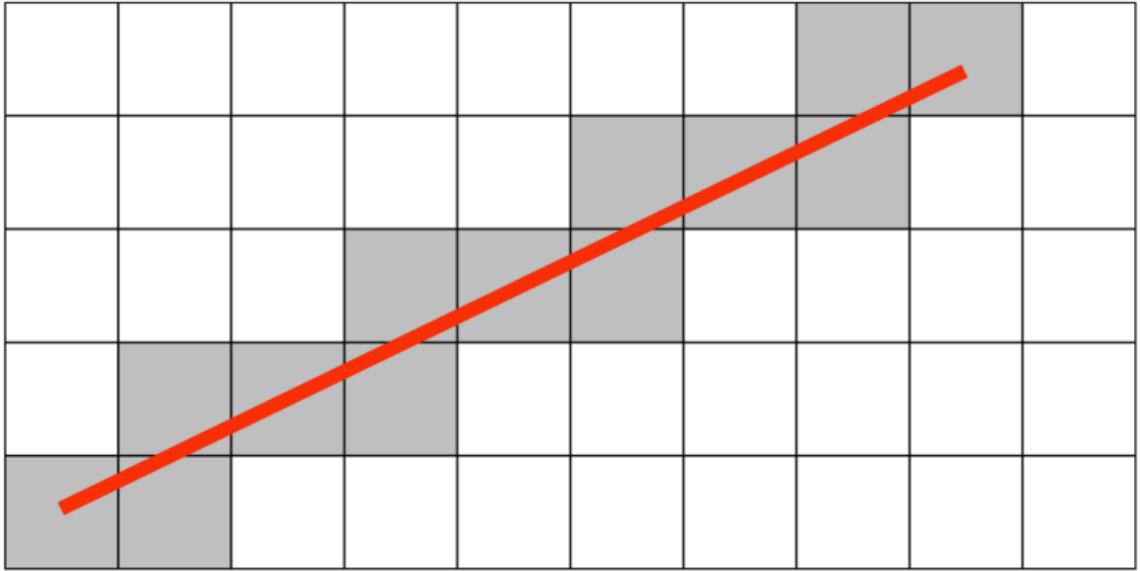
- ▶ “Rasterization”: process of converting a continuous object (a line, a polygon, etc.) to a discrete representation on a “raster” grid (pixel grid)



# What pixels should we color in to depict a line? (cont.)



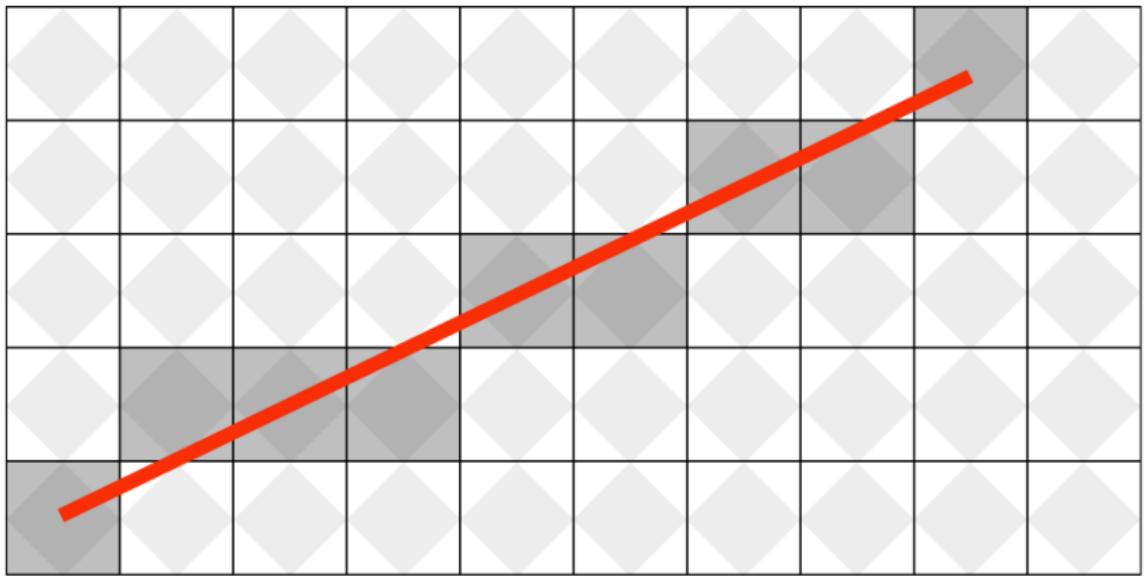
Light up all pixels intersected by the line?



# What pixels should we color in to depict a line? (cont.)



- Diamond rule (used by modern GPUs): light up pixel if line passes through associated diamond





- ▶ Could check every single pixel in the image to see if it meets the condition...
  - $O(n^2)$  pixels in image vs. at most  $O(n)$  "lit up" pixels
  - must be able to do better! (e.g., seek algorithm that does work proportional to number of pixels in the drawing of the line)

# Incremental line rasterization



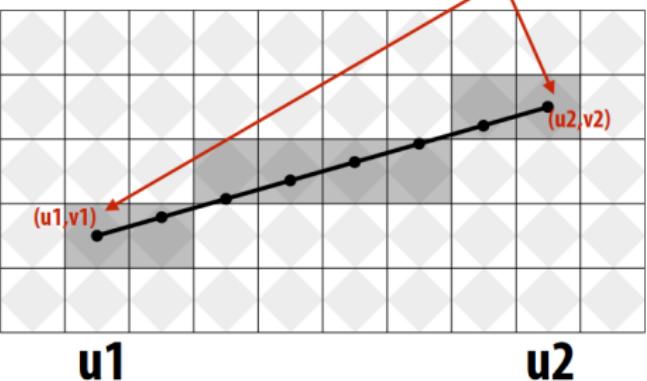
- ▶ Let's say a line is represented with integer endpoints:  $(u_1, v_1)$ ,  $(u_2, v_2)$
- ▶ Slope of line:  $s = (v_2 - v_1) / (u_2 - u_1)$

- Consider an easy special case:

- $u_1 < u_2, v_1 < v_2$  (line points toward upper-right)
  - $0 < s < 1$  (more change in x than y)

Assume integer coordinates  
are at pixel centers

```
v = v1;  
for( u=u1; u<=u2; u++ )  
{  
    v += s;  
    draw( u, round(v) )  
}
```



- ▶ Common optimization: rewrite algorithm to use only integer arithmetic (Bresenham algorithm)



- ▶ To render more realistic pictures (or animations) we need a much richer model of the world.
  - surfaces
  - motion
  - materials
  - lights
  - cameras

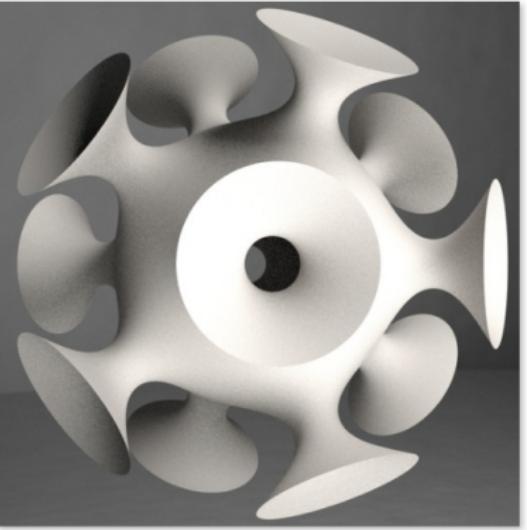
# 2D shapes



# Complex 3D surfaces



(Stanislav Oreshkov)



**Platonic noid**



[Kaldor 2008]

# Modeling material properties



# Realistic lighting environments



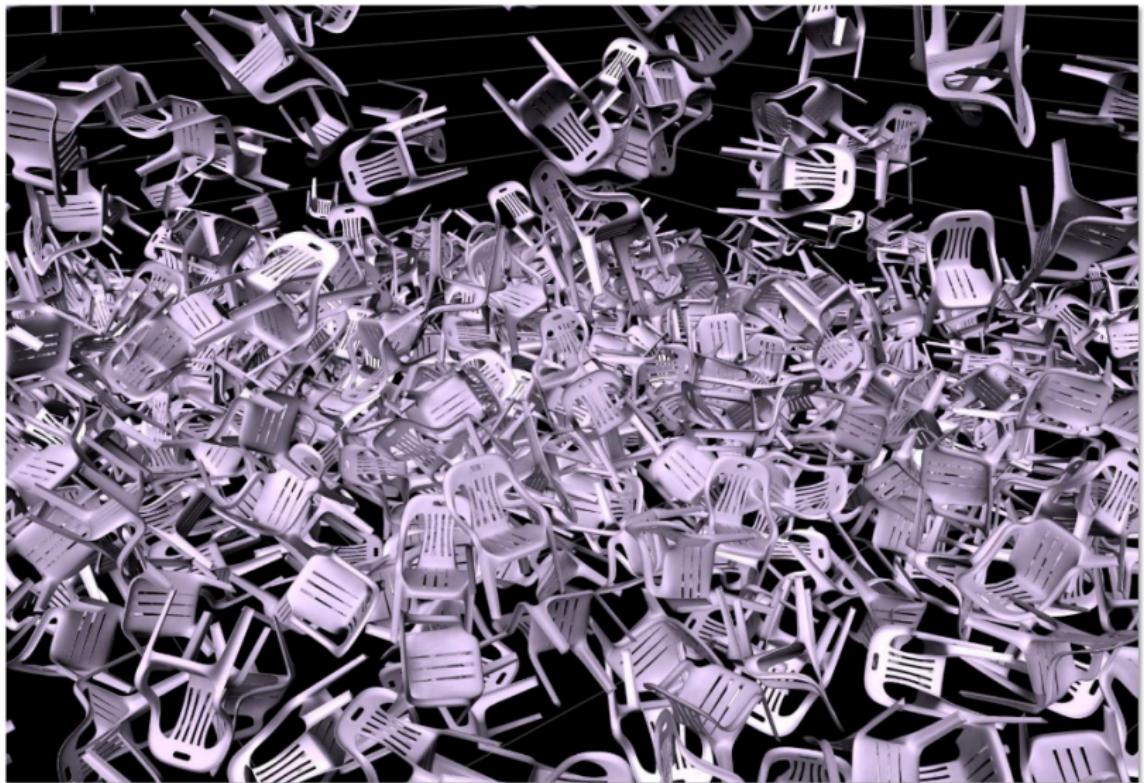
Wall-E, (Pixar 2008)



# Simulation of motion



[https://www.youtube.com/watch?v=tT81VPk\\_ukU](https://www.youtube.com/watch?v=tT81VPk_ukU)

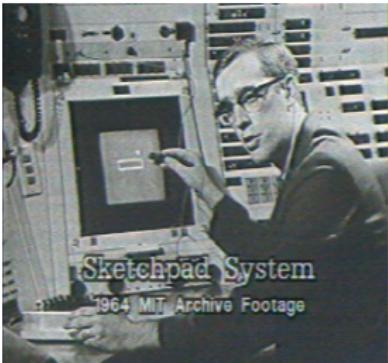


# History of Computer Graphics



## ► Ivan Sutherland (1963) - SKETCHPAD

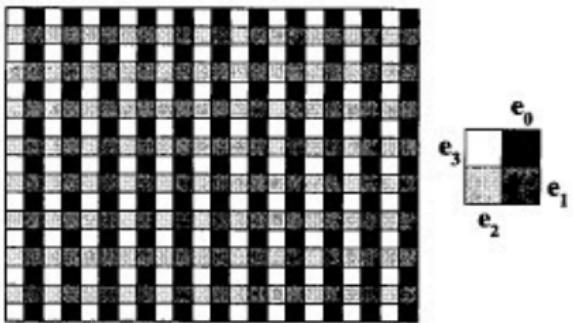
- pop-up menus
- constraint-based drawing
- hierarchical modeling



- ▶ vector displays
  - 1963 – modified oscilloscope
  - 1974 – Evans and Sutherland Picture System
- ▶ raster displays
  - 1975 – Evans and Sutherland frame buffer
  - 1980s – cheap frame buffers bit-mapped personal computers
  - 1990s – liquid-crystal displays laptops
  - 2000s – micro-mirror projectors digital cinema
- ▶ other
  - stereo, head-mounted displays
  - autostereoscopic displays
  - tactile, haptic, sound

## ► 2D

- light pen, tablet, mouse, joystick, track ball, touch panel, etc.
- 1970s 80s - CCD analog image sensor + frame grabber
- 1990s 2000's - CMOS digital sensor + in-camera processing
  - high-X imaging (dynamic range, resolution, depth of field,...)





► 3D

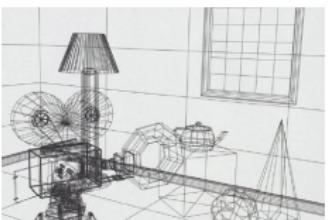
- 3D trackers
- multiple cameras
- active rangefinders

► data gloves

► voice

► 1960s - the visibility problem

- Roberts (1963), Appel (1967) - hidden-line algorithms
- Warnock (1969), Watkins (1970) - hidden-surface algorithms
- Sutherland (1974) - visibility = sorting



## ► 1970s - raster graphics

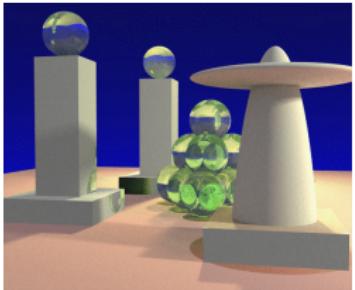
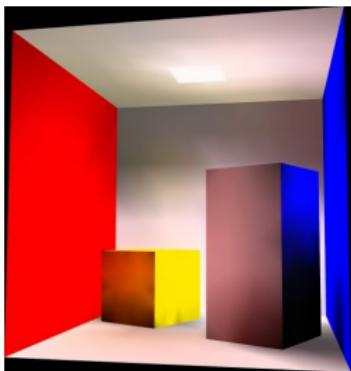
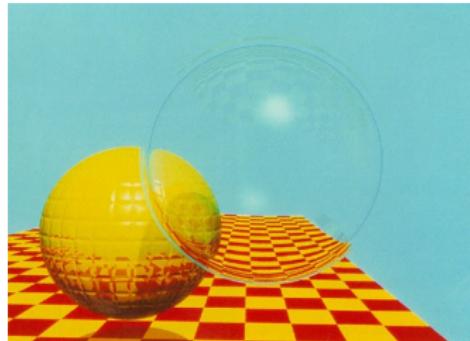
- Gouraud (1971) - diffuse lighting
- Phong (1974) - specular lighting
- Blinn (1974) - curved surfaces, texture
- Crow (1977) - anti-aliasing



# Rendering (cont.)



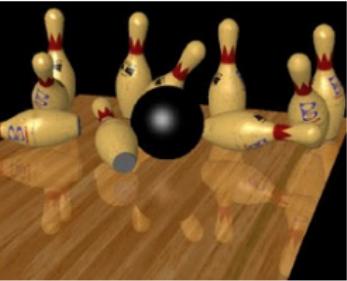
- ▶ early 1980s - global illumination
  - Whitted (1980) - ray tracing
  - Goral, Torrance et al. (1984), Cohen (1985) - radiosity
  - Kajiya (1986) - the rendering equation



# Rendering (cont.)



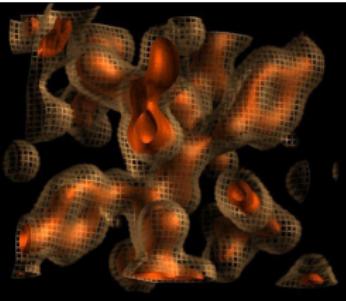
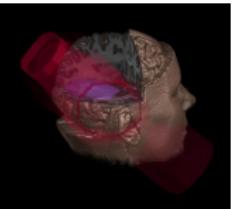
- ▶ late 1980s - photorealism
  - Cook (1984) - shade trees
  - Perlin (1985) - shading languages
  - Hanrahan and Lawson (1990) - RenderMan



# Rendering (cont.)



- ▶ early 1990s - non-photorealistic rendering
  - Drebin et al. (1988), Levoy (1988) - volume rendering
  - Haeberli (1990) - impressionistic paint programs
  - Salesin et al. (1994-) - automatic pen-and-ink illustration
  - Meier (1996) - painterly rendering



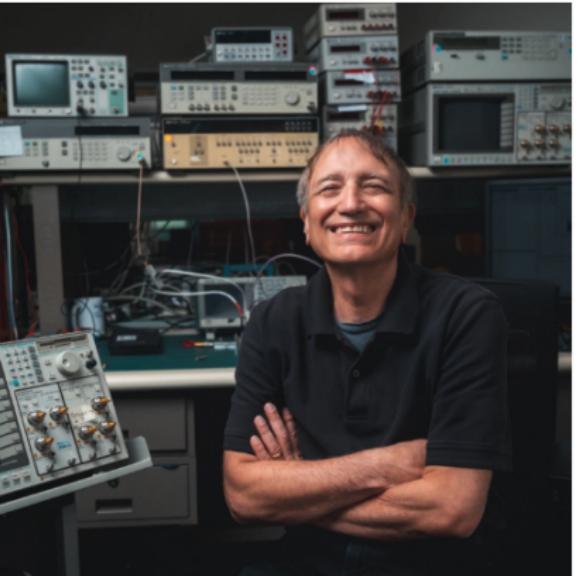


- ▶ CG has been nearly ubiquitous in video generation
- ▶ Pre-rendered graphics are nearly scientifically photorealistic
- ▶ Real-time graphics on a suitably high-end system may simulate photorealism to the untrained eye.
- ▶ Texture mapping has matured into a multistage process with many layers
- ▶ Experiments into the processing power required to provide graphics in real time at ultra-high-resolution modes like 4K Ultra HD are beginning.

# 2019 Turing Award for CG Pioneers



(a) Edwin E. Catmull( US – 2019), former president of Pixar and Disney Animation Studios For fundamental contributions to 3D computer graphics and the impact of computer-generated imagery (CGI) in filmmaking and other applications.



(b) Stanford computer scientist and engineer Patrick M. “Pat” Hanrahan is co-recipient of the 2019 Turing Award.



- ▶ Key players profiled in the global computer graphics market include
  - Advanced Micro Devices (AMD), Inc.
  - ARM Ltd.
  - Intel Corporation
  - Imagination Technologies Limited
  - Nvidia Corporation
  - Sony Corporation
  - Adobe Systems Ltd.
  - Siemens PLM Software
  - Autodesk Inc.
  - Microsoft Corporation
  - Dassault Systemes SA.

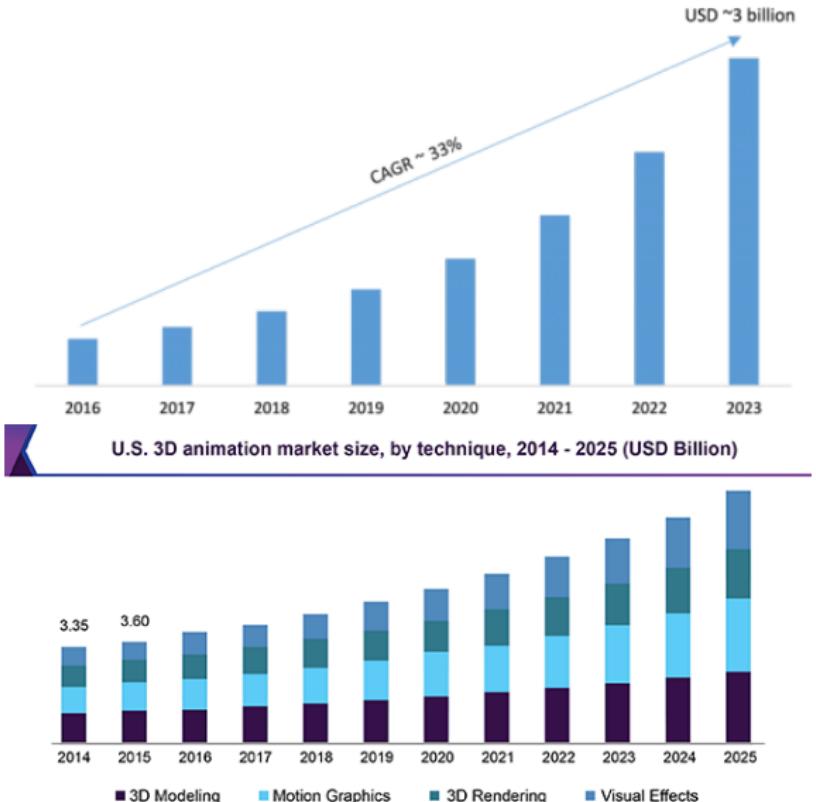
# Computer Graphics in Industry (cont.)



- ▶ Other significant players in the global computer graphics market include Matrox, Mentor Graphics, Inc., and 3D PLM-related software providers including, PTC, SAP PLM, and Oracle PLM
- ▶ The global computer graphics market is projected to reach the value of US \$ 308.6 Bn by 2030.
- ▶ The computer graphics market is projected to expand at a CAGR of 6% from 2020 to 2030.



# Computer Graphics in Industry (cont.)



Source: [www.grandviewresearch.com](http://www.grandviewresearch.com)

# Acknowledgements



- ▶ Some of the slides have been adopted from different universities like Stanford and different internet sources. The due credits are acknowledged.



Thank You! :)