

# 컴퓨터그래픽스

김준호

Visual Computing Lab.

국민대학교 소프트웨어학부

# Course Instructor

- Name: 김준호 (Junho Kim)
- Research interests
  - Computer Graphics / Neural Graphics
  - Computer Visions / 3D Vision
  - Deep Learning / Neural Algorithms
- Expertise
  - Generative Models
  - 3D reconstruction & understanding
- Contact Info.
  - ☎: 02-910-4826
  - E-mail: [junho@kookmin.ac.kr](mailto:junho@kookmin.ac.kr)



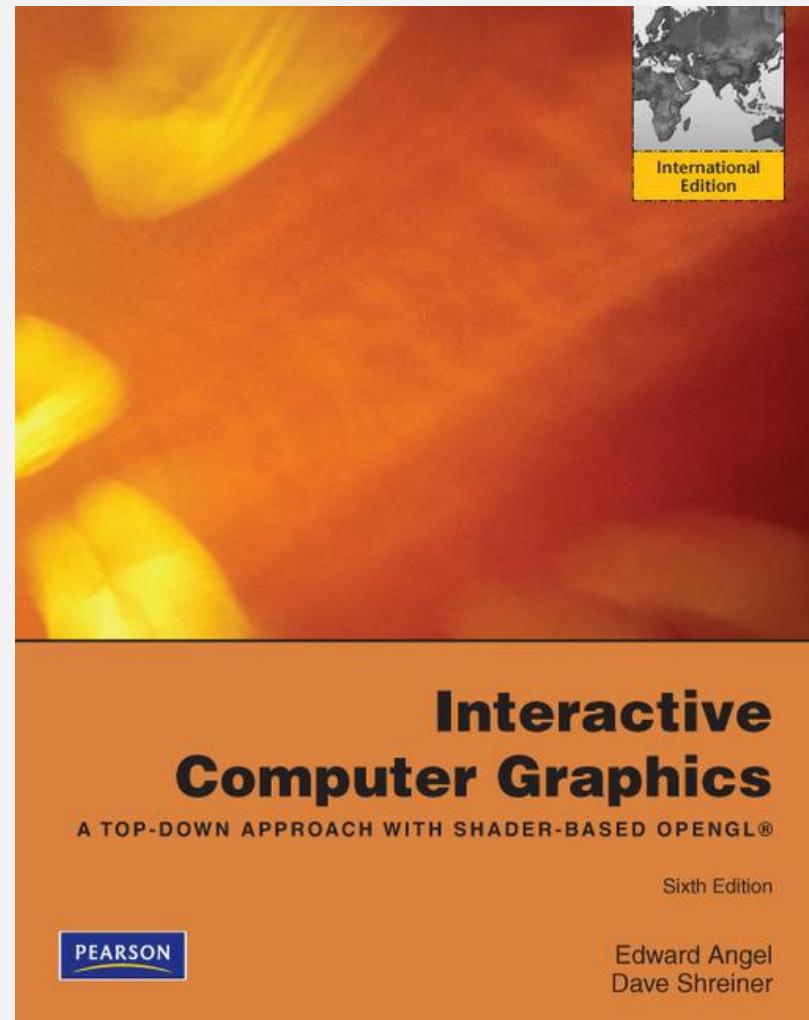
# Teaching Assistant

- 이영균
  - 비주얼 컴퓨팅 연구실
  - 미래관 7층 K-Lab
  - E-mail: [yglee981130@kookmin.ac.kr](mailto:yglee981130@kookmin.ac.kr)
- 김채현
  - 비주얼 컴퓨팅 연구실
  - 미래관 7층 K-Lab
  - E-mail: [chaehyun@kookmin.ac.kr](mailto:chaehyun@kookmin.ac.kr)



# Textbook

- Interactive Computer Graphics:  
A Top-Down Approach with  
**Shader-based OpenGL (6/E)**, by  
Edward Angel and Dave Shreiner.



# Course Description

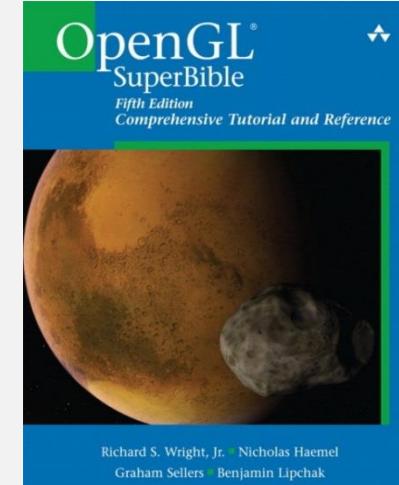
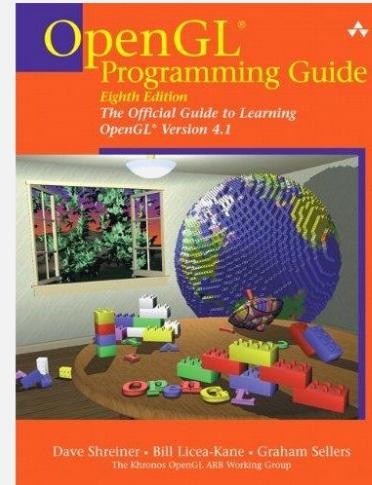
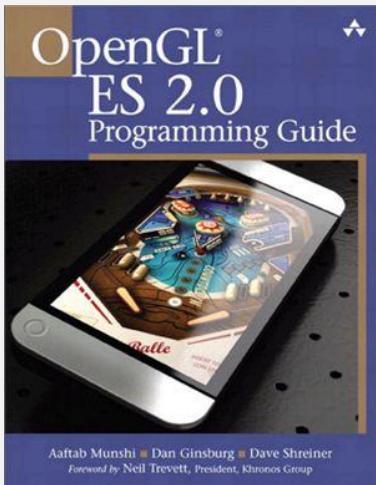
- Lectures hours
  - Mon. 13:30 ~ 14:45      Lecture
  - Wed. 13:30 ~ 14:45      Lecture & Lab.
- Course evaluation (**tentative**)
  - Midterm: 30%
  - Final: 30%
  - Assignments & Project: 30%
    - # of Assignments: ~ 5 or 6
    - # of Projects: 1
  - Attendance: 10% (incl., attitude)

# Course Description

- Objective
  - To learn the basic theories of computer graphics
    - Transformations, Viewing, Interaction
    - Modeling, Rendering, Animation
    - Fragments, Buffers, Buffer Objects
  - To understand the interactive graphics techniques with
    - Shader-based OpenGL: programmable rendering pipeline
    - ~~Non shader-based OpenGL: fixed rendering pipeline~~
  - To know about advanced topics in computer graphics
    - Desktop graphics basics: OpenGL 1.x & 2.x+
    - Mobile graphics basics: OpenGL ES 1.x & 2.x+

# References

- Useful books
  - OpenGL ES 2.0 Programming Guide
    - <http://opengles-book.com/>
    - Several figures in this slides come from this book
  - OpenGL Programming Guide (a.k.a., OpenGL Redbook)
    - Available in online: <http://glprogramming.com/red/>
    - Several figures in this slides come from this book
  - OpenGL Superbible
    - <http://www.starstonesoftware.com/OpenGL/>



# References

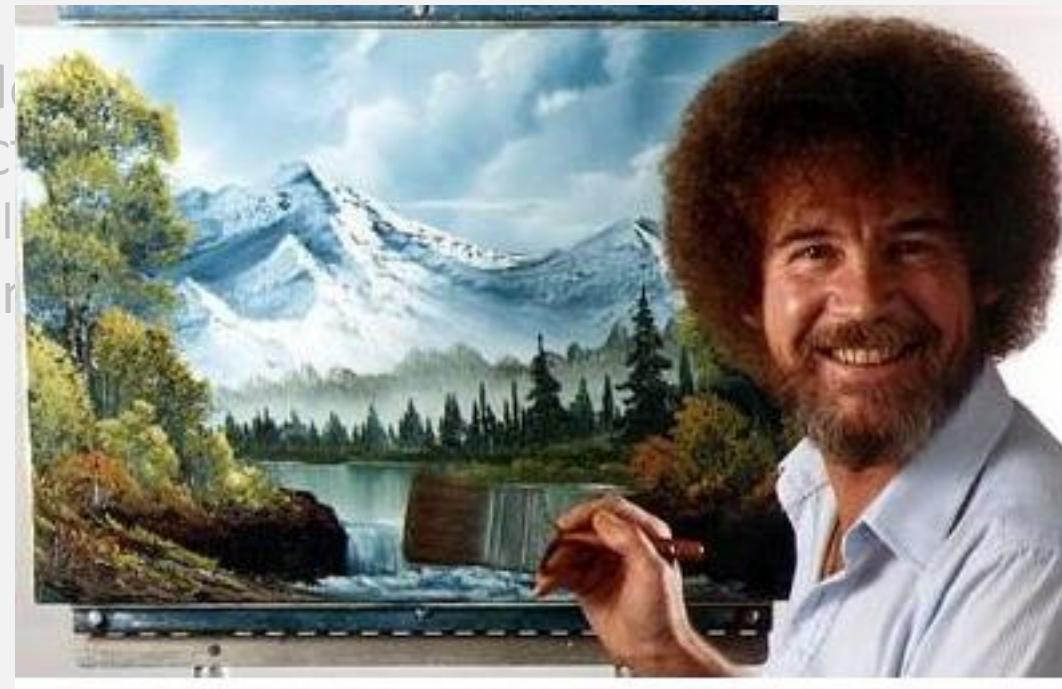
- Useful reference sites
  - OpenGL generals: <http://www.opengl.org>
  - OpenGL ES: <http://www.khronos.org>
  - WebGL: <http://www.chromeexperiments.com/webgl>
- Useful OpenGL tutorial sites
  - OpenGL tutorial: <http://www.opengl-tutorial.org/kr/>
  - Learn OpenGL: <https://learnopengl.com/>
  - NeHe OpenGL Tutorials: <http://nehe.gamedev.net/>
  - Lighthouse3D OpenGL Tutorials: <http://www.lighthouse3d.com/tutorials/>
  - Song Ho Ahn’s OpenGL Tutorial: <http://www.songho.ca/opengl/index.html>
- Useful Youtube tutorials
  - TheChernoProject ([link](#))

# Some Tips

- Please, make questions in the lectures
  - Q & A could be ok in Korean.
- Materials related to our textbook is available at the following author's homepage
  - <http://www.cs.unm.edu/~angel/BOOK/INTERACTIVE COMPUTER GRAPHICS/SIXTH EDITION/>
  - I strongly recommend you to print out the PPT slides therein and bind them.
- Use Google & wikipedia to get further information
  - Google: [www.google.com](http://www.google.com)
  - Wikipedia: [www.wikipedia.org](http://www.wikipedia.org)

# Some Tips

- Please, make questions in the lectures
  - Q & A could be ok in Korean.
- Materials related to our textbook is available
  - <http://www.cs.unm.edu/~angel/BOOK/INTERACTIVEBOOK.HTML>
  - I strongly recommend you to print out the PPT slides
- Use Google & wikipedia to get further information
  - Google: [www.google.com](http://www.google.com)
  - Wikipedia: [www.wikipedia.org](http://www.wikipedia.org)
- **걱정하지 마세요! 매우 쉽습니다.**
  - 단, 수업을 잘 따라오고 있으실 경우...

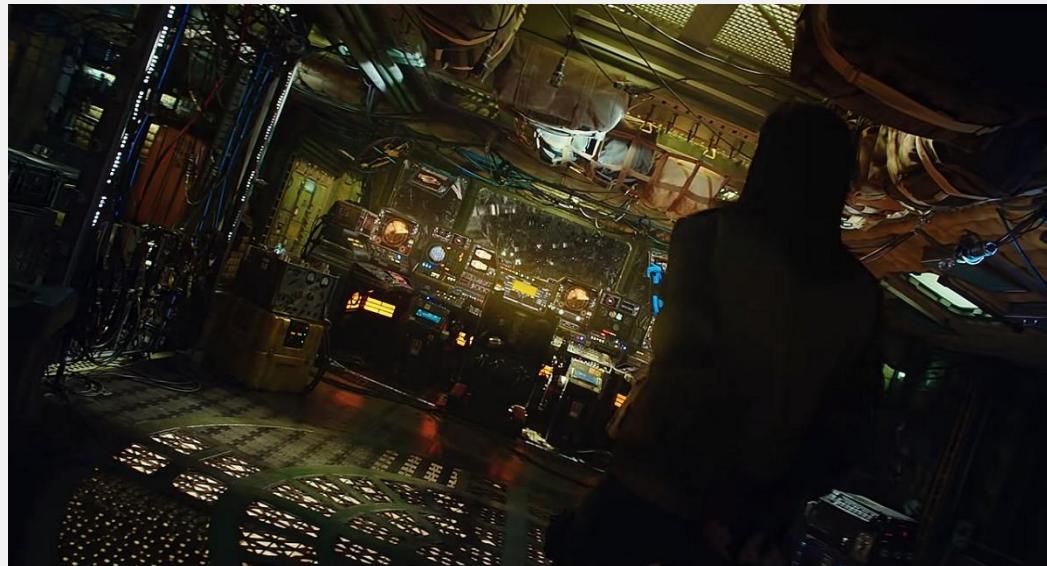


- **Computer Graphics**
- OpenGL & OpenGL ES

# Introduction

# Introduction to Computer Graphics

- Computer Graphics is ...
  - The representation and manipulation of pictorial data by a computer [from wikipedia.org]



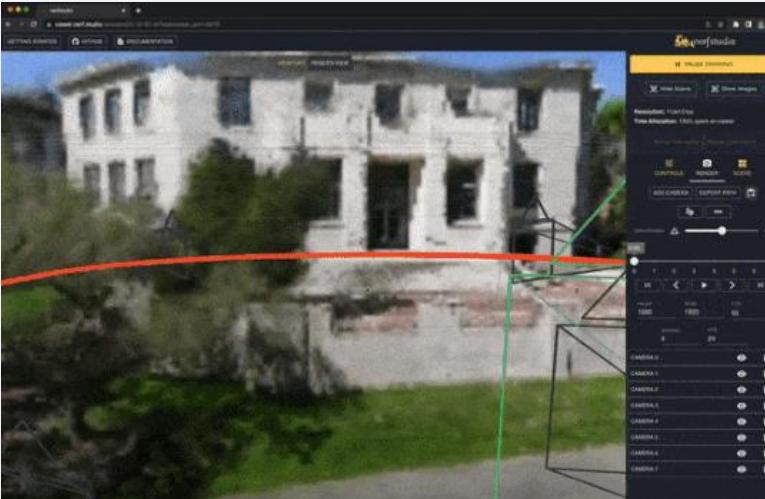
[Spaceship Victory – offline graphics]



[Red Dead Redemption 2 – interactive graphics]

# Introduction to Computer Graphics

- Computer Graphics is ...
  - The representation and manipulation of pictorial data by a computer [from wikipedia.org]



[StyleGAN, NeRF, Diffusion – Neural Graphics]

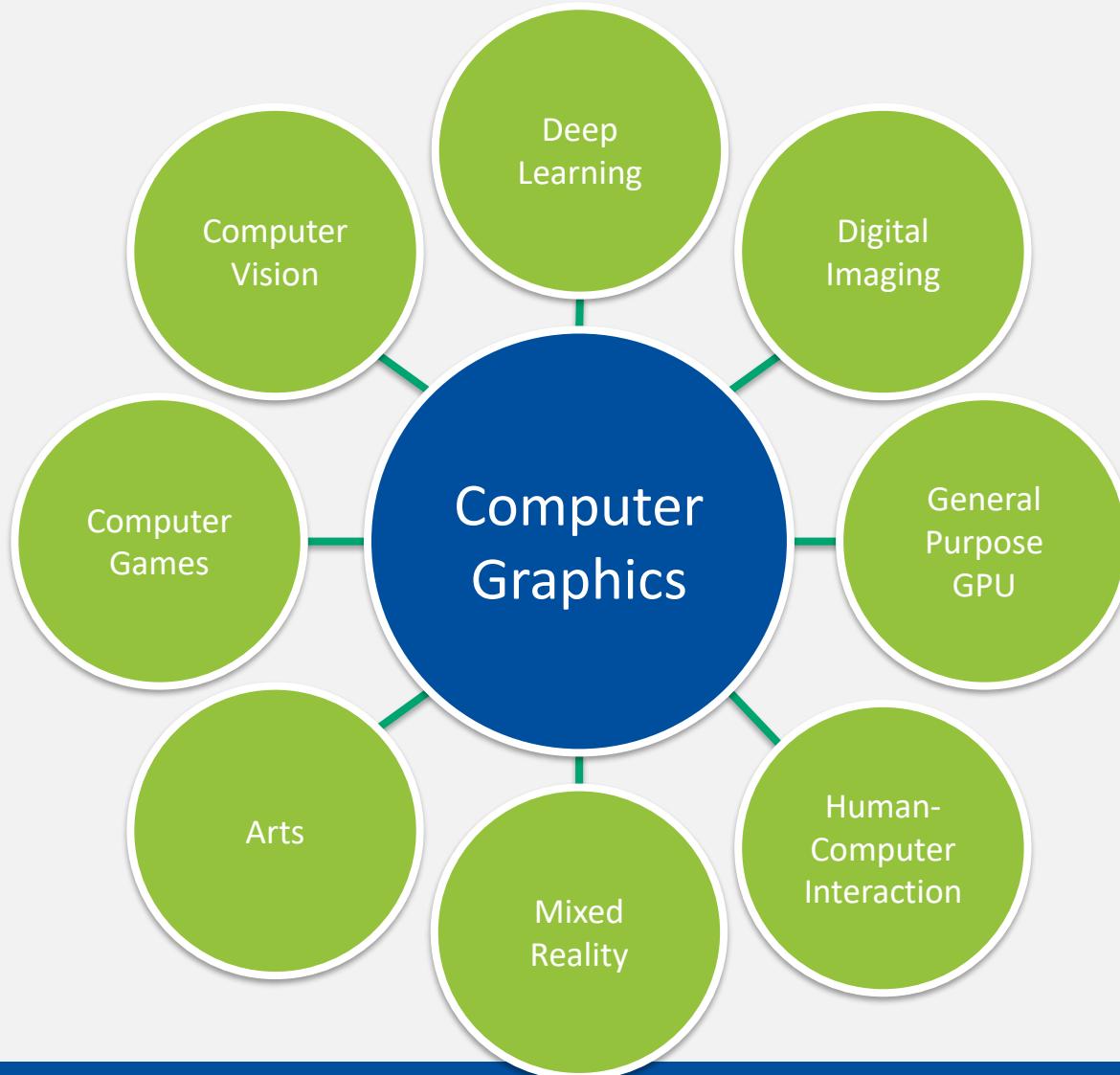
# Introduction to Computer Graphics

- Computer Graphics is ...
  - The representation and manipulation of pictorial data by a computer [from wikipedia.org]



[[OpenAI Sora – Neural Graphics](#)]

# Introduction to Computer Graphics

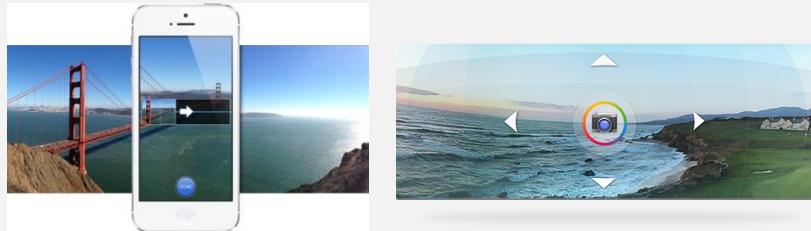


# Introduction to Computer Graphics

- Computer Graphics + Computer Vision

- Panoramic photos

- Panorama mode (iOS6 above)
    - PhotoSphere (Android 4.2 above)
    - cf) [QuickTime™ VR](#), Apple Inc. in 1995
      - Image-based rendering
      - Image stitching technology



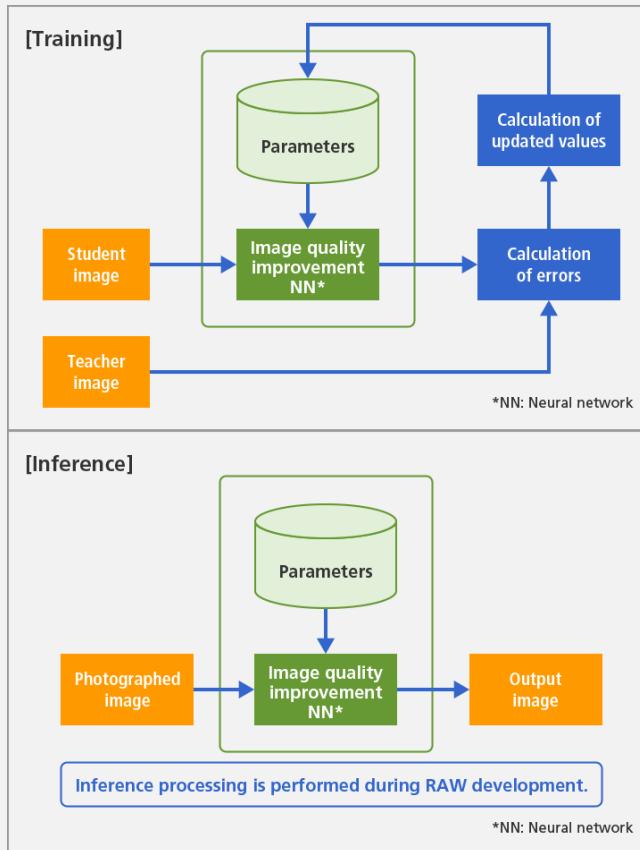
- 3D maps

- Apple Maps (iOS6 above)
    - Google Maps 3D ([app](#))
    - Naver Labs ([article](#), [www](#))
    - cf) [C3 Technology](#), [acute3D](#)
      - Multiple view geometry
      - Photogrammetry technology



# Introduction to Computer Graphics

- Computer Graphics + Digital imaging → Computational Photography



[Deep Learning Image Processing Technology @ Canon](#)

# Introduction to Computer Graphics

- Computer Graphics + Digital imaging
  - Making Celeb Photos
  - A Style-Based Generator Architecture for Generative Adversarial Networks, T. Karras et al., CVPR 2019 ([arXiv](#), [youtube](#))



# Introduction to Computer Graphics

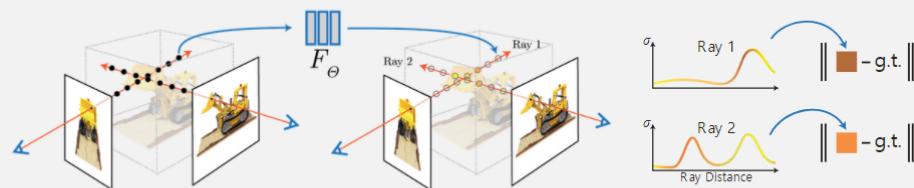
- Computer Graphics + Neural Rendering
  - Neural Rendering from Multi-View Photos, w/o Modeling
  - NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, Mildenhall et al., ECCV 2020 ([pdf](#), [youtube](#), [nerfstudio](#))

## Abstract & Method

We present a method that achieves state-of-the-art results for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views.

$$(x,y,z,\theta,\phi) \rightarrow \boxed{\text{ }} \rightarrow (RGB\sigma)$$
$$F_{\theta}$$

Our algorithm represents a scene using a fully-connected (non-convolutional) deep network, whose input is a single continuous 5D coordinate (spatial location  $(x, y, z)$  and viewing direction  $(\theta, \phi)$ ) and whose output is the volume density and view-dependent emitted radiance at that spatial location.



We synthesize views by querying 5D coordinates along camera rays and use classic volume rendering techniques to project the output colors and densities into an image. Because volume rendering is naturally differentiable, the only input required to optimize our representation is a set of images with known camera poses. We describe how to effectively optimize neural radiance fields to render photorealistic novel views of scenes with complicated geometry and appearance, and demonstrate results that outperform prior work on neural rendering and view synthesis.

# Introduction to Computer Graphics

- Computer Graphics + Self-driving Cars
  - CARLA ([www](#))
  - Video-to-Video Synthesis, T-C. Wang et al., NIPS 2018 ([www](#), [arXiv](#), [youtube](#), [video](#))



Video-to-Video Synthesis

Ting-Chun Wang<sup>1</sup> Ming-Yu Liu<sup>1</sup> Jun-Yan Zhu<sup>2</sup> Guilin Liu<sup>1</sup> Andrew Tao<sup>1</sup> Jan Kautz<sup>1</sup> Bryan Catanzaro<sup>1</sup>

<sup>1</sup>NVIDIA Corporation    <sup>2</sup>MIT

[Paper] [arXiv] [Video] [Code]

# Introduction to Computer Graphics

- Computer Graphics + Mixed Reality → Spatial Computing
  - Natural feature tracking
    - QR 코드와 같이 특별한 형식의 마커가 아닌 일반 영상을 마커로 인식
    - e.g.) [PTC® Vuforia™](#)
  - SLAM-based approach
    - 실시간으로 주변환경에 대한 지도를 작성고, 지도 내의 현재 위치를 인식
    - e.g.) [MS KinectFusion](#), [MS MobileFusion](#), [PTAM](#), [Meta Quest](#), [Apple Vision Pro](#)



[\(video, youtube\)](#)



[\(video, youtube\)](#)

# Introduction to Computer Graphics

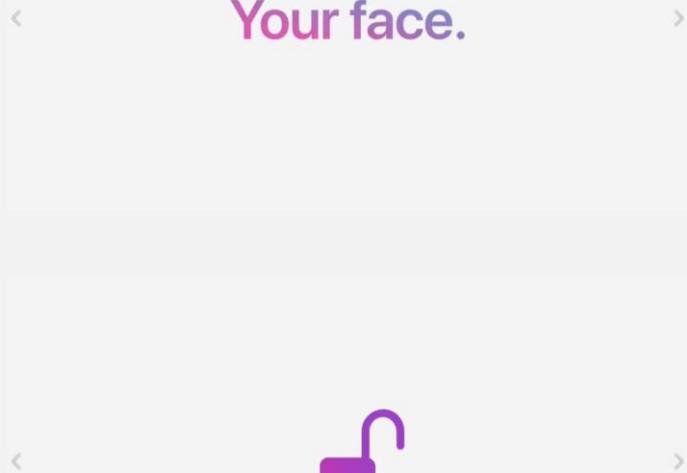
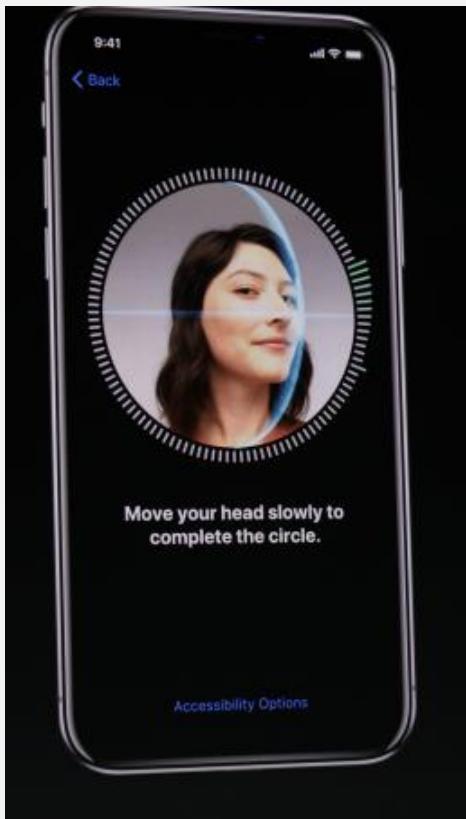
- Computer Graphics + Mixed Reality
  - Animoji on iPhone X



[[youtube](#), [video](#)]

# Introduction to Computer Graphics

- Computer Graphics + Mixed Reality
  - Face ID on iPhone X



# Introduction to Computer Graphics

- Computer Graphics + Mixed Reality
  - Microsoft Kinect



Kinect for Windows v1 Sensor



# Introduction to Computer Graphics

- Computer Graphics + Mixed Reality
  - Microsoft Kinect



Kinect for Windows v1 Sensor

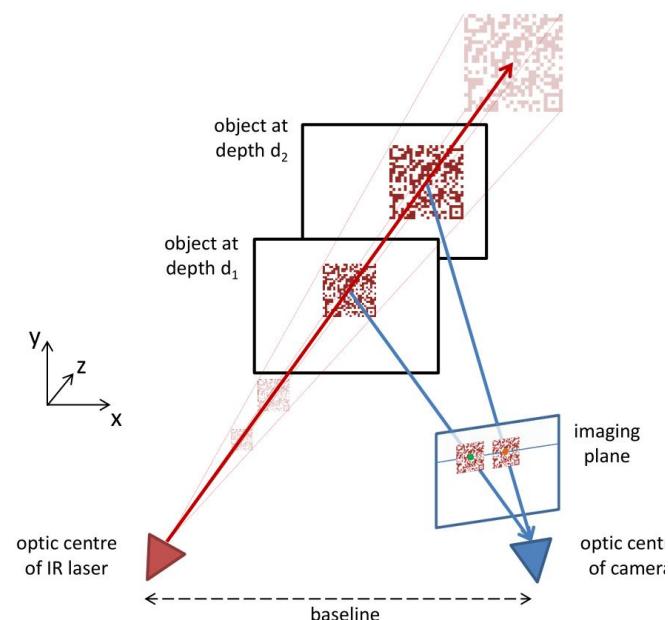


# Introduction to Computer Graphics

- Computer Graphics + Mixed Reality
  - Microsoft Kinect



Structured light



Kinect for Windows v1 Sensor



[Jamie Shotton's Slide @ CVPR Tutorial]

# Introduction to Computer Graphics



- iPhoneX FaceID
  - Similar tech. with MS Kinect
    - PrimeSense was [bought by Apple in 2013](#).
    - See more details in articles in [[techcrunch](#), [independent](#), [imore](#)]



# Face Tracking with ARKit in iPhone X

## Tech overview

- Real-time tracking for AR



## ARKit API overview [[www](#), [video](#)]

(3:40 – 5:40)

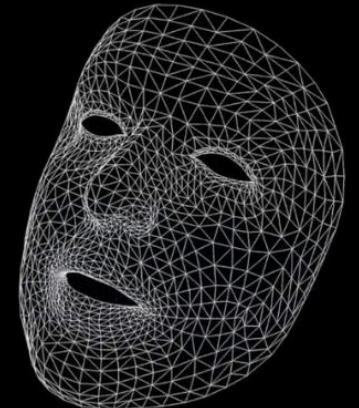
### ARFaceAnchor

Added when face detected

Position and orientation

3D topology and facial expression

Tracked in real-time



# Face Tracking with ARKit in iPhone X

## Tech overview

- Facial expression to Animoji



## ARKit API overview

- Blend Shape [[www](#), [video](#), [paper](#)]

### Blend Shapes

ARBlendShapeLocation Coefficients

BrowDownLeft	EyeLookInRight	MouthClose	MouthRollLower
BrowDownRight	EyeLookOutLeft	MouthDimpleLeft	MouthRollUpper
BrowInnerUp	EyeLookOutRight	MouthDimpleRight	MouthShrugLower
BrowOuterUpLeft	EyeLookUpLeft	MouthFrownLeft	MouthShrugUpper
BrowOuterUpRight	EyeLookUpRight	MouthFrownRight	MouthSmileLeft
CheekPuff	EyeSquintLeft	MouthFunnel	MouthSmileRight
CheekSquintLeft	EyeSquintRight	MouthLeft	MouthStretchLeft
CheekSquintRight	EyeWideLeft	MouthLowerDownLeft	MouthStretchRight
EyeBlinkLeft	EyeWideRight	MouthLowerDownRight	MouthUpperUpLeft
EyeBlinkRight	JawForward	MouthPressLeft	MouthUpperUpRight
EyeLookDownLeft	JawLeft	MouthPressRight	NoseSneerLeft
EyeLookDownRight	JawOpen	MouthPucker	NoseSneerRight
EyeLookInLeft	JawRight	MouthRight	

# BlendShape



## Blend Shapes

ARBlendShapeLocation Coefficients

BrowDownLeft	EyeLookInRight	MouthClose	MouthRollLower
BrowDownRight	EyeLookOutLeft	MouthDimpleLeft	MouthRollUpper
BrowInnerUp	EyeLookOutRight	MouthDimpleRight	MouthShrugLower
BrowOuterUpLeft	EyeLookUpLeft	MouthFrownLeft	MouthShrugUpper
BrowOuterUpRight	EyeLookUpRight	MouthFrownRight	MouthSmileLeft
CheekPuff	EyeSquintLeft	MouthFunnel	MouthSmileRight
CheekSquintLeft	EyeSquintRight	MouthLeft	MouthStretchLeft
CheekSquintRight	EyeWideLeft	MouthLowerDownLeft	MouthStretchRight
EyeBlinkLeft	EyeWideRight	MouthLowerDownRight	MouthUpperUpLeft
EyeBlinkRight	JawForward	MouthPressLeft	MouthUpperUpRight
EyeLookDownLeft	JawLeft	MouthPressRight	NoseSneerLeft
EyeLookDownRight	JawOpen	MouthPucker	NoseSneerRight
EyeLookInLeft	JawRight	MouthRight	

BlendShape 파라메터 실시간 분석

BlendShape 파라메터 이용 얼굴 리깅

[[youtube](#), [video](#)]



# Facial/Motion captures

## Personalized Facial/Motion captures



[[www](#), youtube ([1](#), [2](#), [3](#)), video ([1](#), [2](#), [3](#))]

## Production-level Facial/Motion captures



[Avatar 2009]

D-ID.com ([www](http://www.D-ID.com))



# Introduction to Computer Graphics

- Computer Graphics + Computer Games

- Physics engines

- 복잡한 물리 연산을 실시간으로 계산
    - 영화와 같은 연출 가능
    - E.g.) nVidia PhysX engine



([video](#), [youtube](#))

- Pre-computed radiance transfer

- 실시간 렌더링 계산이 용이하도록,  
오프라인으로 미리 처리



([video](#), [youtube](#))

- Stylized rendering

# Introduction to Computer Graphics

- Computer Graphics + Arts
  - Stylized rendering
    - 사진과 같은 사실적 렌더링 결과가 아닌, 화가가 그린 듯 스타일이 살아 있는 렌더링



Madame Palmyre with Her Dog  
– Henri de Toulouse-Lautrec



Stylized rendering of Cupid  
[\[Herzmann and Zorin 2000\]](#)



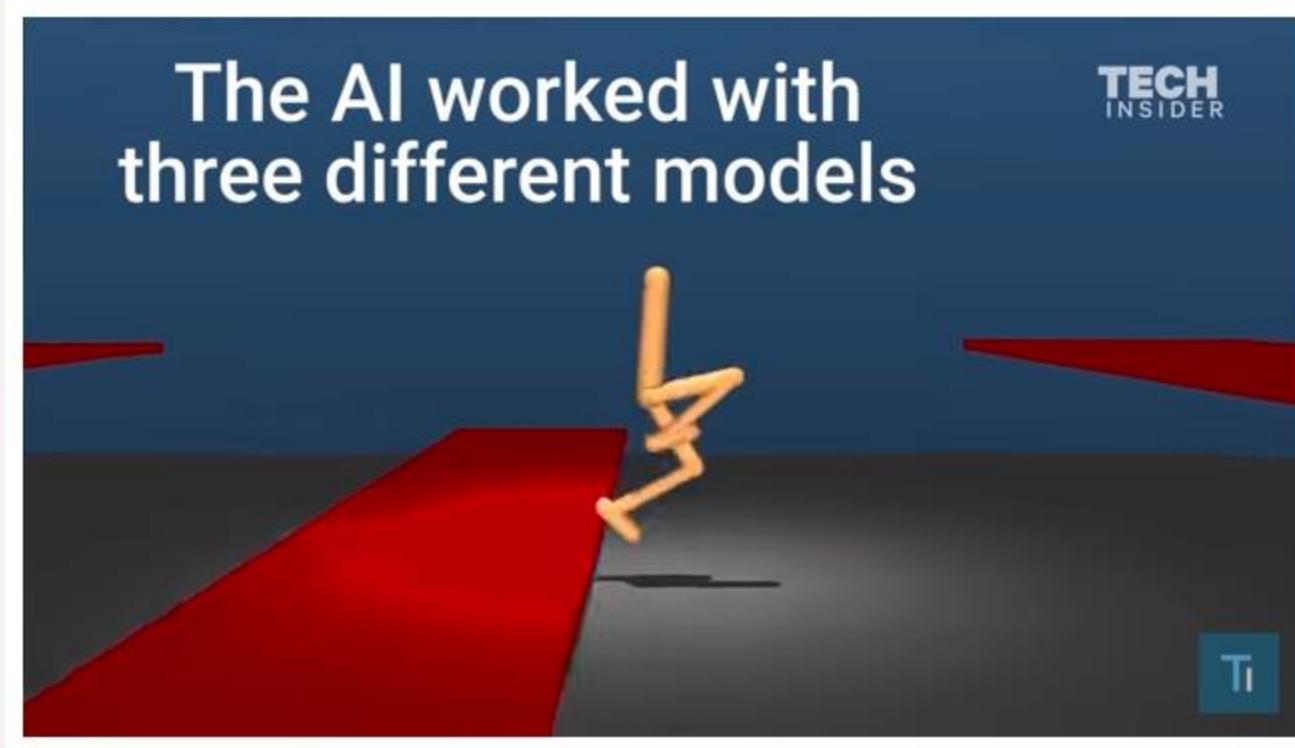
Neural Style Transfer  
[\[Gatys et al. 2016\]](#)



Inkpunk Diffusion  
<https://huggingface.co/Envii/Inkpunk-Diffusion>

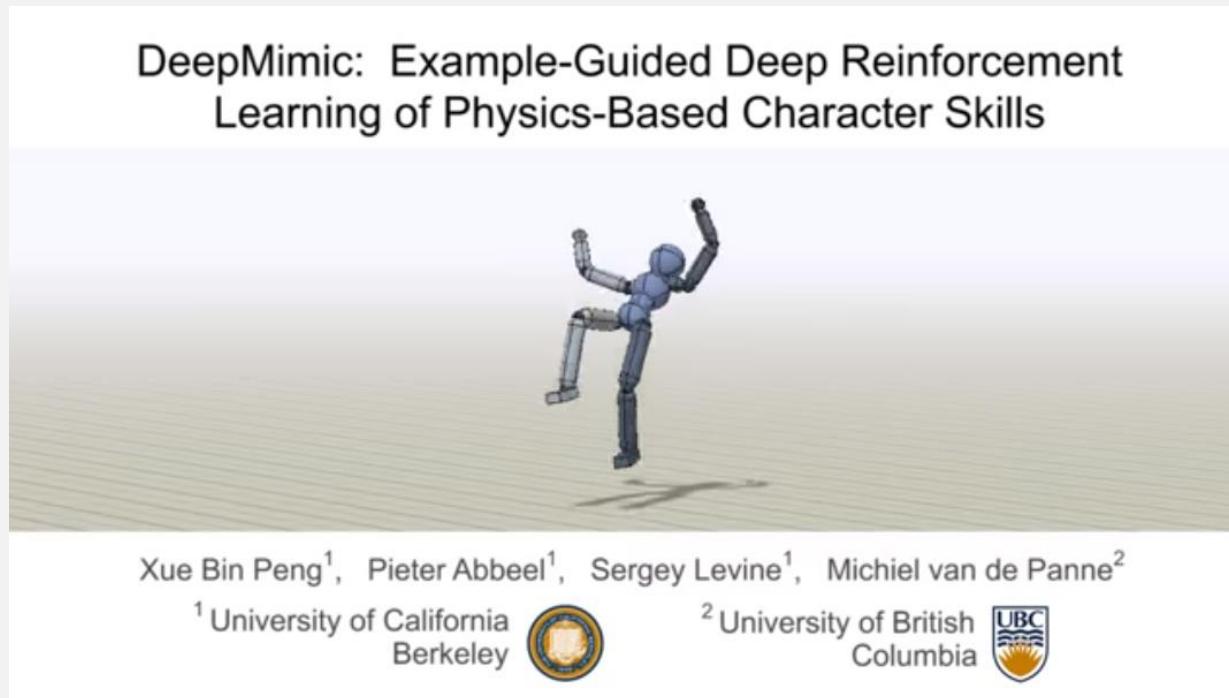
# Introduction to Computer Graphics

- Google's DeepMind AI Just Taught Itself To Walk ([youtube](#), [video](#))



# Introduction to Computer Graphics

- DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, X. B. Peng et al, Siggraph 2018 ([arXiv](#), [youtube](#), [video](#))



# Introduction to Computer Graphics

- Mode-Adaptive Neural Networks for Quadruped Motion Control, H. Zhang et al., Siggraph 2018 ([pdf](#), [youtube](#), [video](#))

## Mode-Adaptive Neural Networks for Quadruped Motion Control

HE ZHANG<sup>†</sup>, University of Edinburgh

SEBASTIAN STARKE<sup>†</sup>, University of Edinburgh

TAKU KOMURA, University of Edinburgh

JUN SAITO, Adobe Research



Fig. 1. A selection of results using our method for quadruped animation. We show some different modes for sitting, turning trot, pace, canter, jumping and standing from left to right. The locomotion gaits are not labeled individually, but naturally produced by the movement velocity control.

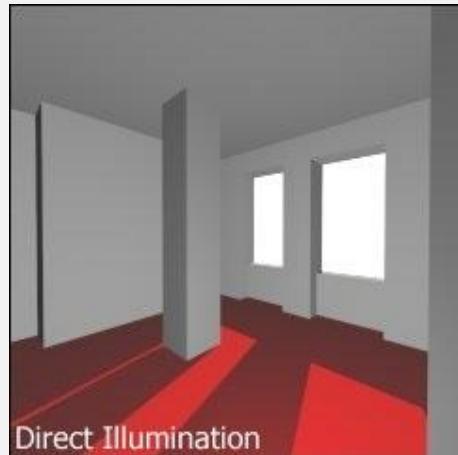
# Principles of Computer graphics

- Illumination
  - Global v.s. Local
    - Time – Quality tradeoff



## Global illumination

- Off-line graphics
- High quality
- E.g.) Animations, Cinematic effects

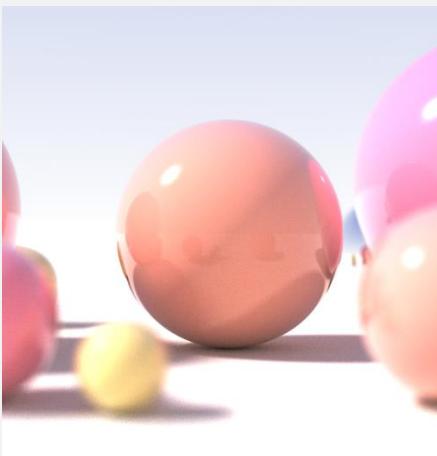
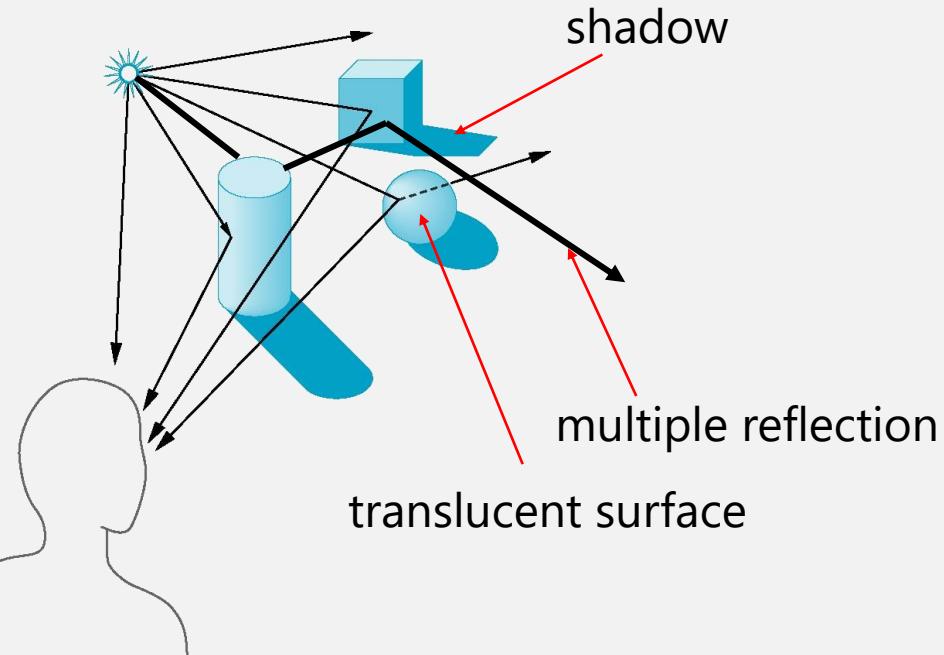


## Local illumination

- Interactive graphics
- Low quality
- E.g.) Mobile UI, Games

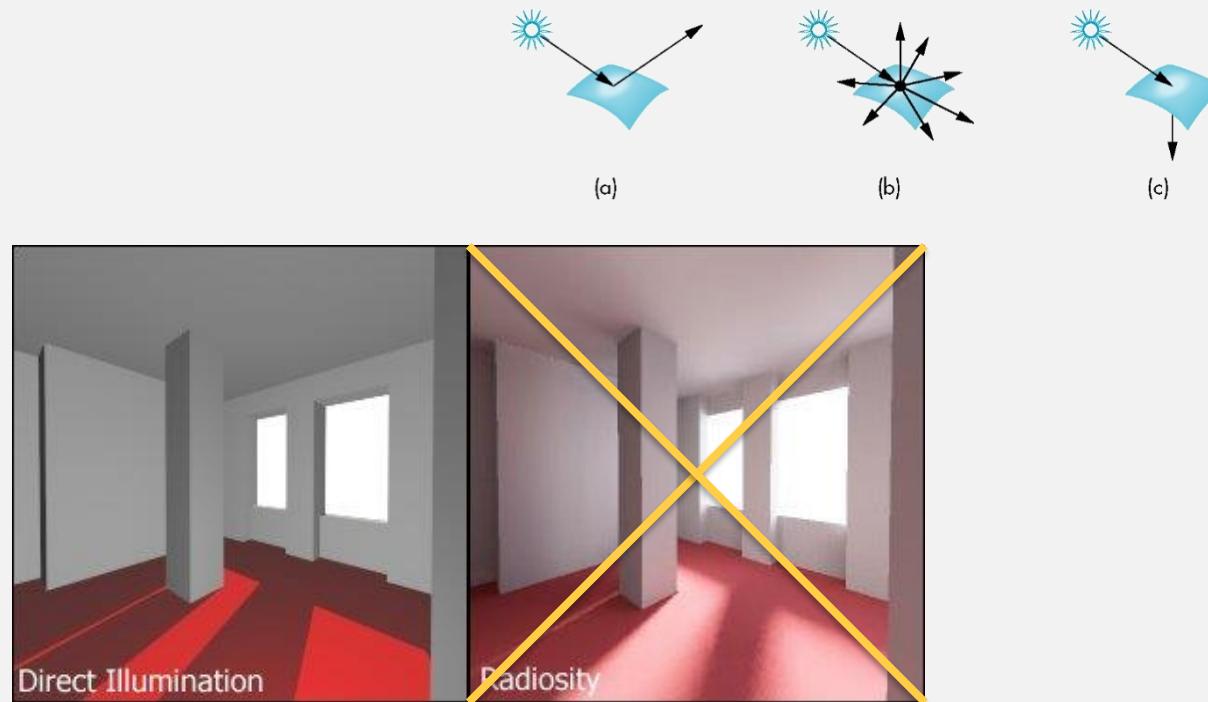
# Global Illumination

- Non-real-time in general
  - [Rendering equation](#)
  - [Radiosity, Ray tracing](#)
- Considering global light effects
  - Direct illumination
  - Indirect illumination



# Local Illumination

- Real-time in general
  - Phong reflection model
- Considering local effects only → H/W friendly
  - Direct illumination
  - ~~Indirect illumination~~



# Interactive Computer Graphics

- Real-time, real-time, real-time!!!
  - [Blender eevee](#), [Unity v.s. Unreal](#)
- H/W accelerated graphics
  - GPU
  - OpenGL, OpenGL ES, DirectX
- Interaction should be properly handled
  - Mouse/Touch interactions



- Computer Graphics
- OpenGL & OpenGL ES

# Introduction

# OpenGL

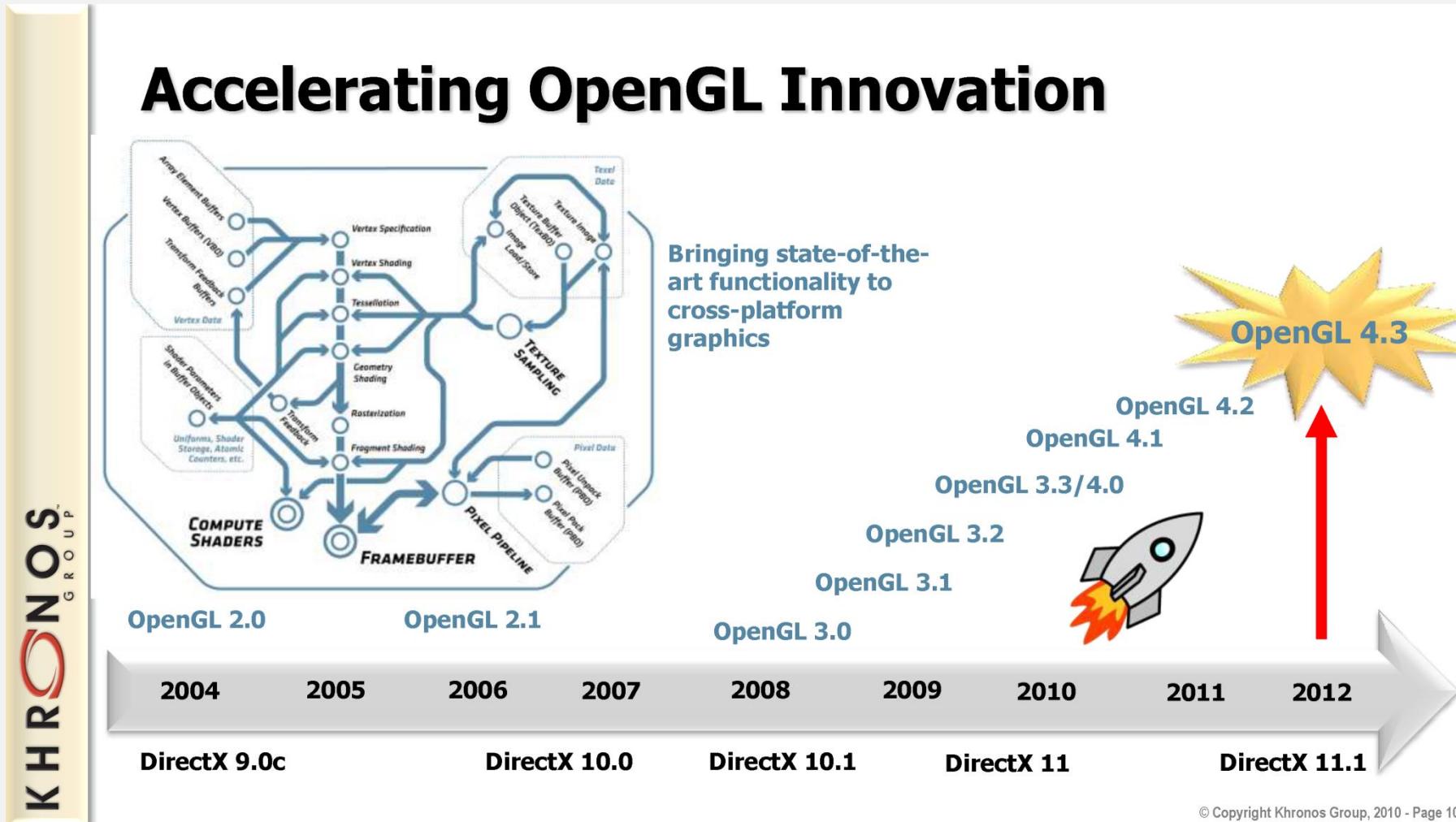
- ~ 1991
  - IRIS GL: a proprietary graphics API created by Silicon Graphics (SGI)
- OpenGL
  - Released in Jan. 1992
  - Procedural H/W accelerated 3D graphics
  - Maintenance
    - Maintained by OpenGL Architectural Review Board (OpenGL ARB)
    - Now, maintained by Khronos Group (2006 ~ Now)
- Strong competitor
  - Microsoft Direct3D (1998 ~ now)



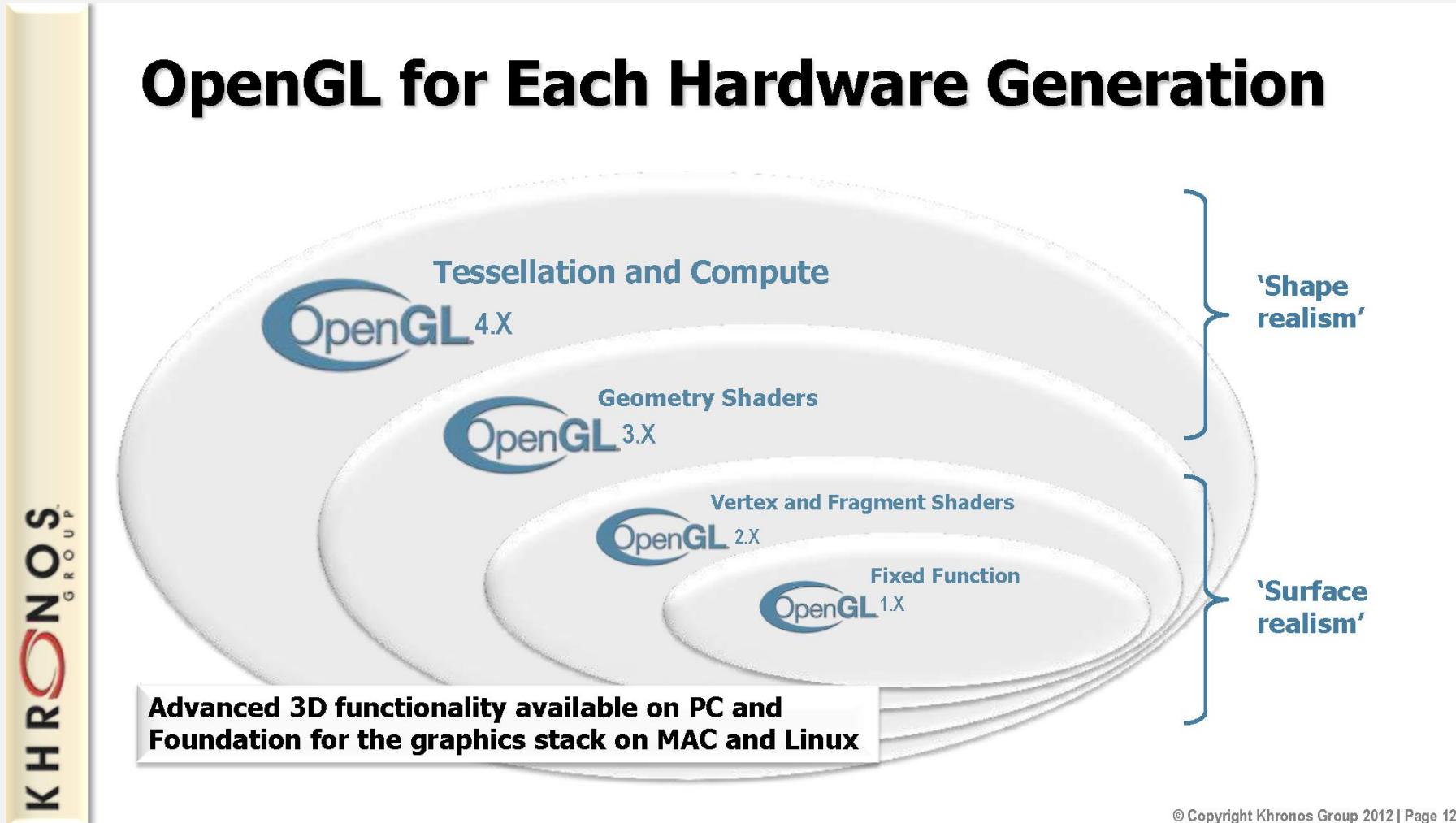


© Copyright Khronos Group 2012 | Page 6

# History of OpenGL



# History of OpenGL – Summary



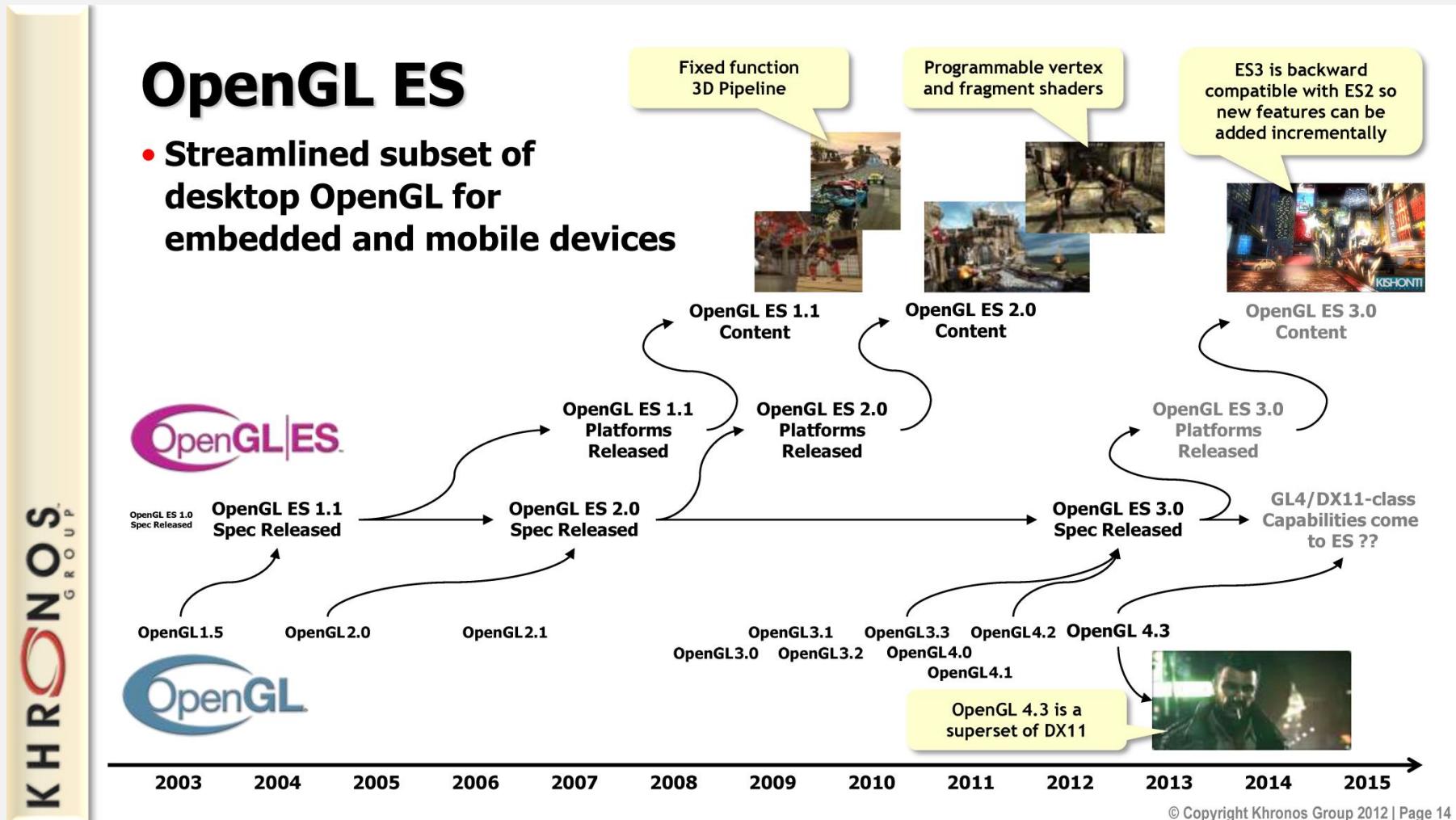
K H R O N O S<sup>®</sup>

# OpenGL ES

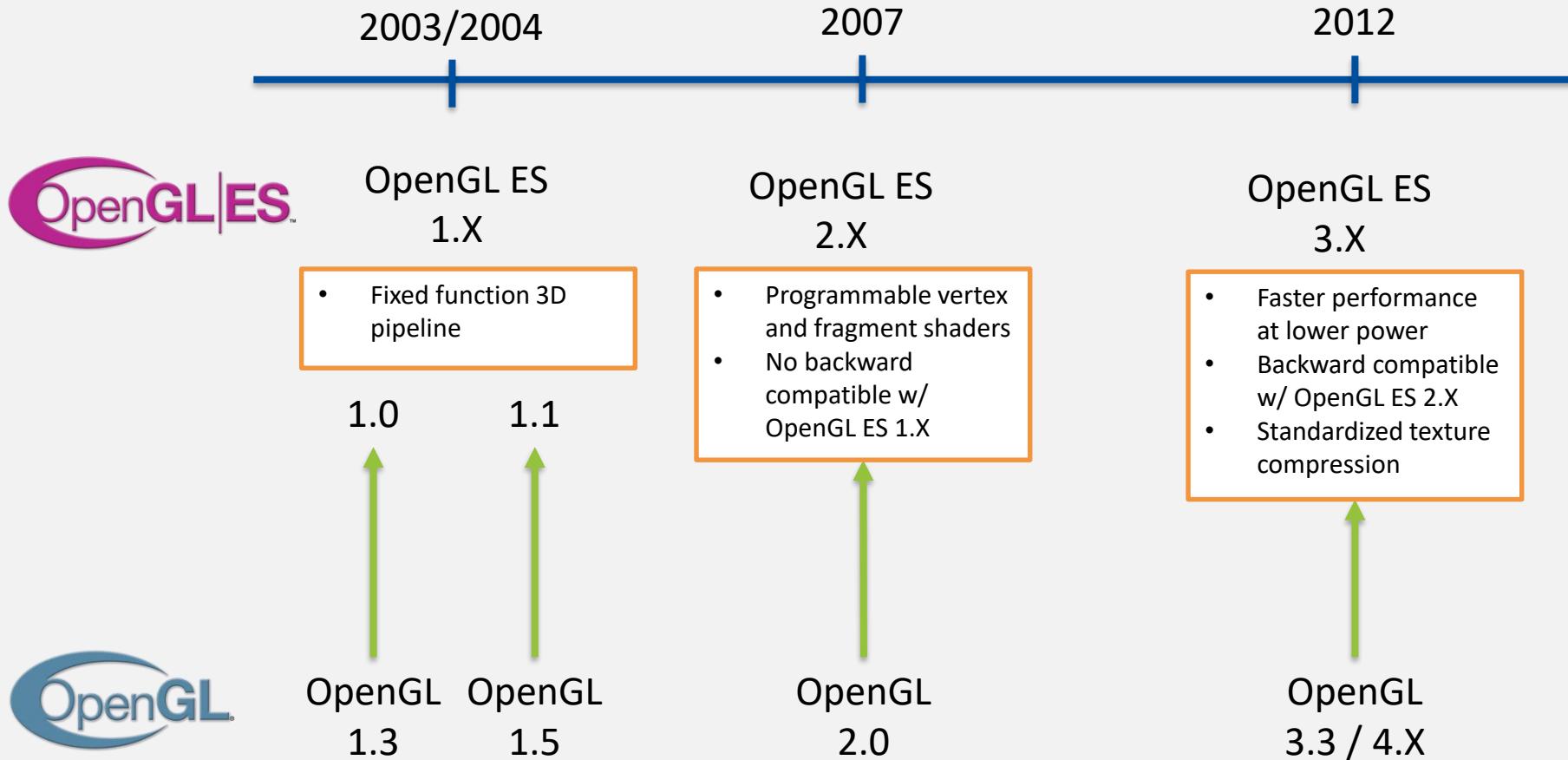
- OpenGL for Embedded Systems
  - A subset of OpenGL for embedded systems
    - Mobile phones, PDAs, video game consoles
  - Royalty-free, cross-platform API
  - Maintained by Khronos Group
- The leading 3D rendering API for mobile and embedded devices
  - OpenGL ES adopted by every major handset OS
  - OpenGL ES has become the most widely deployed 3D API



# History of OpenGL ES



# History of OpenGL ES – Summary



# Vulkan – New Generation Graphics & Compute API

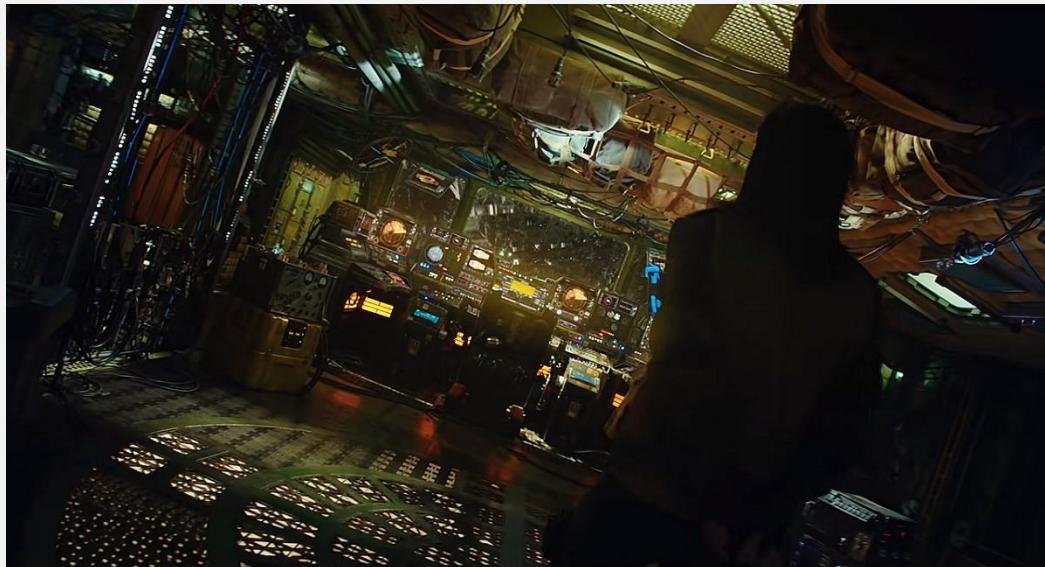
- Low-overhead, cross-platform 3D graphics and compute API
  - Drawn from & built upon AMD's [Mantle](#)
    - Explicit GPU control
    - Multi-threading efficiency
  - Royalty-free, cross-platform API
  - Maintained by Khronos Group
- Next generation GPU APIs
  - Microsoft DirectX 12 – Windows only
  - Apple Metal – Apple only
  - Vulkan – Cross platform ([MWC 2016 Demo](#))



# Computer Graphics Systems

# Introduction to Computer Graphics

- Computer Graphics is ...
  - The representation and manipulation of pictorial data by a computer [from wikipedia.org]



[Spaceship Victory – offline graphics]



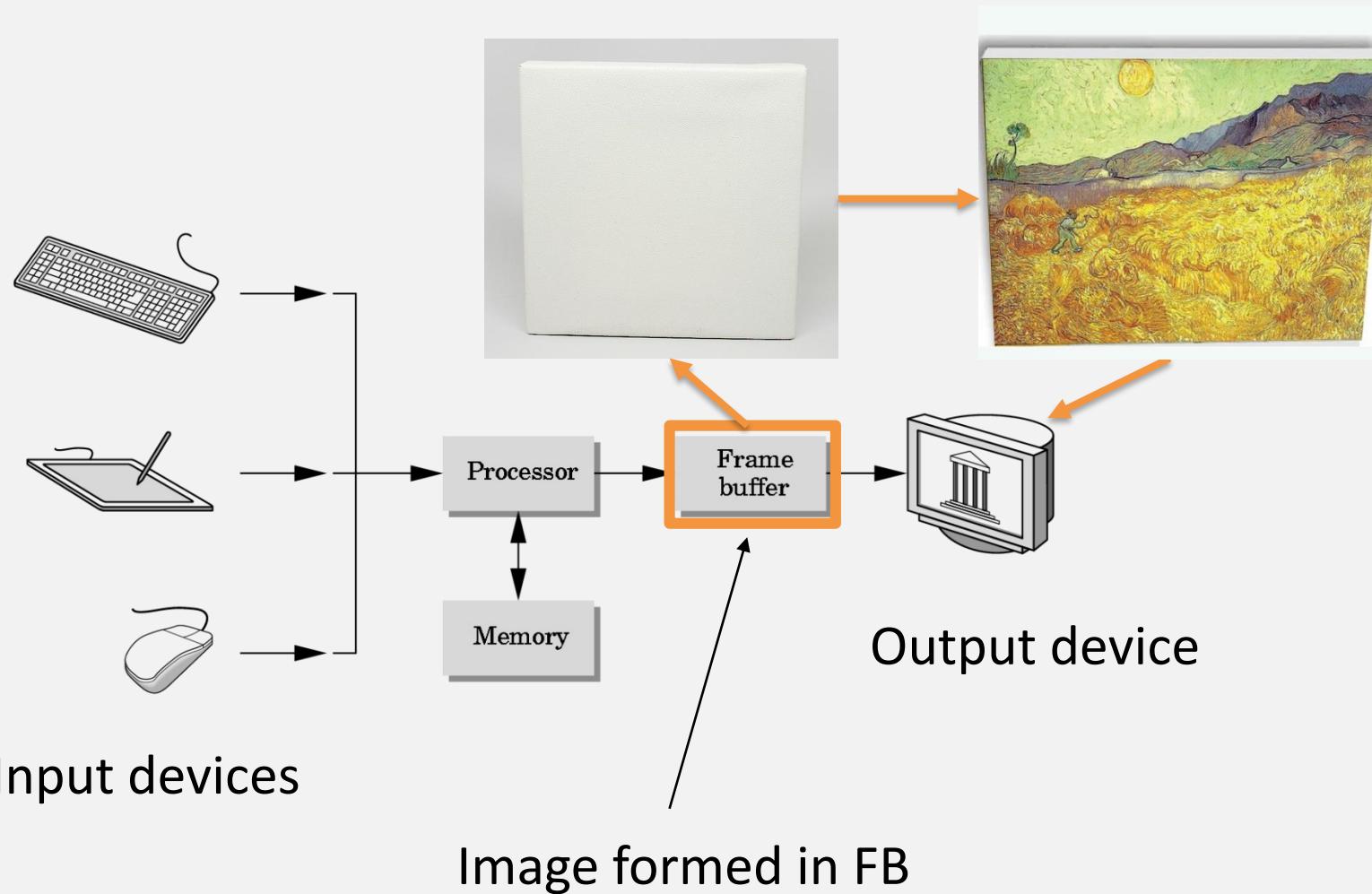
[Red Dead Redemption 2 – interactive graphics]

# Example

- Where did this image come from?
- What H/W and S/W did we need to produce it?

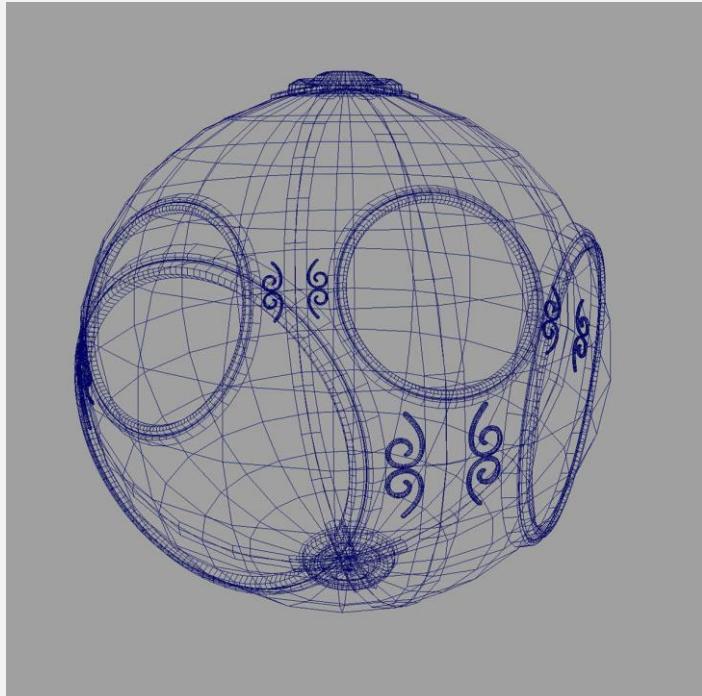


# Basic Graphics Systems



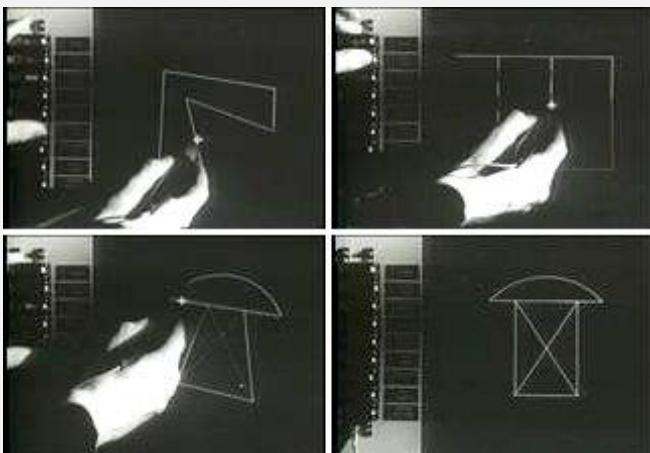
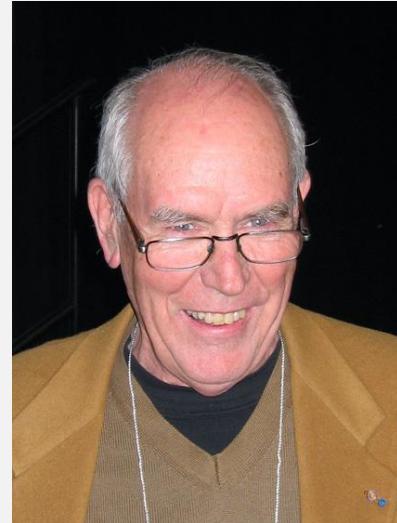
# Computer Graphics: 1960-1970

- Wireframe graphics
  - Draw only lines
  - Oscilloscope cathode ray tube (CRT): storage tubes



# Sketchpad (1963)

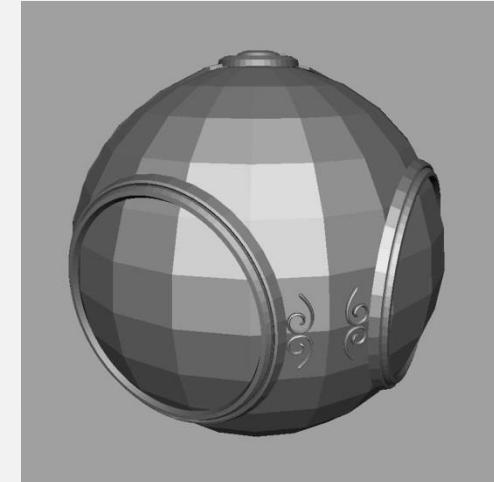
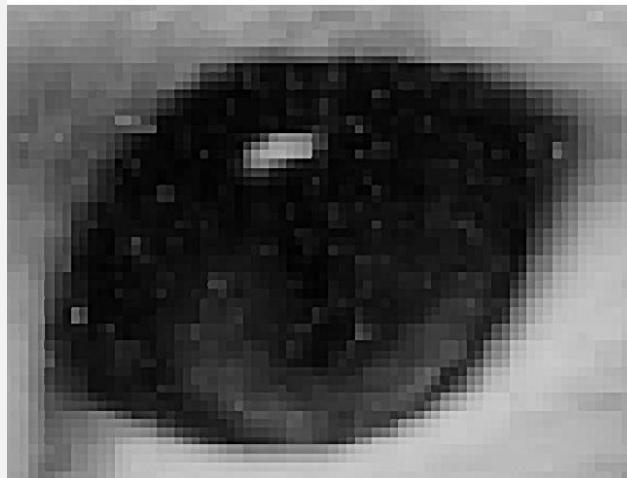
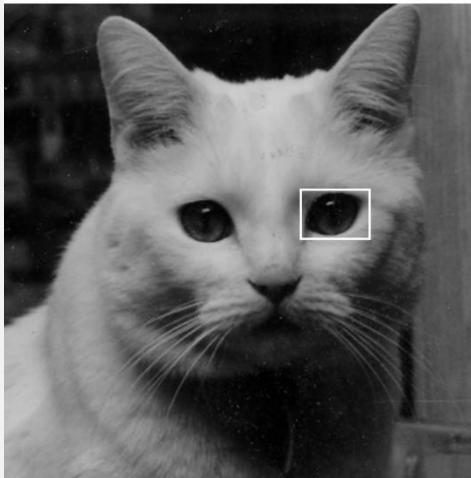
- Ivan Sutherland's PhD thesis at MIT
  - Recognized the potential of man-machine interaction
    - He received the Turing Award in 1988
  - Vector graphics – “line work”



([video](#), [youtube](#))

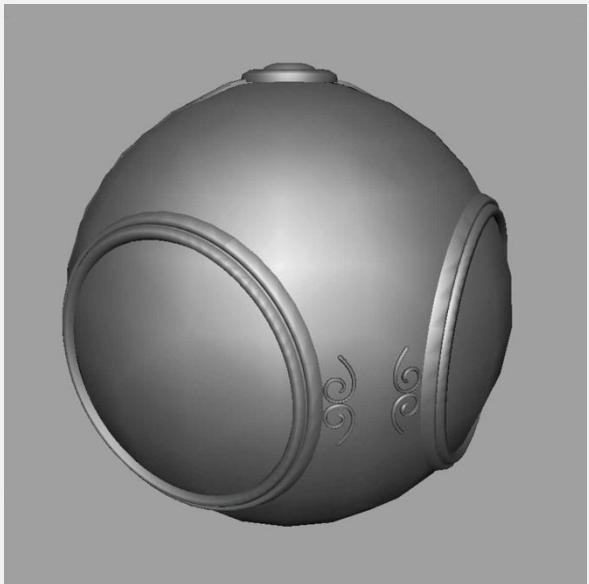
# Computer Graphics: 1970-1980

- Raster Graphics
  - Image produced as an array (the raster) of picture elements (pixels) in the frame buffer
  - It allows us to go from lines and wire frame images to filled polygons



# Computer Graphics: 1980-1990

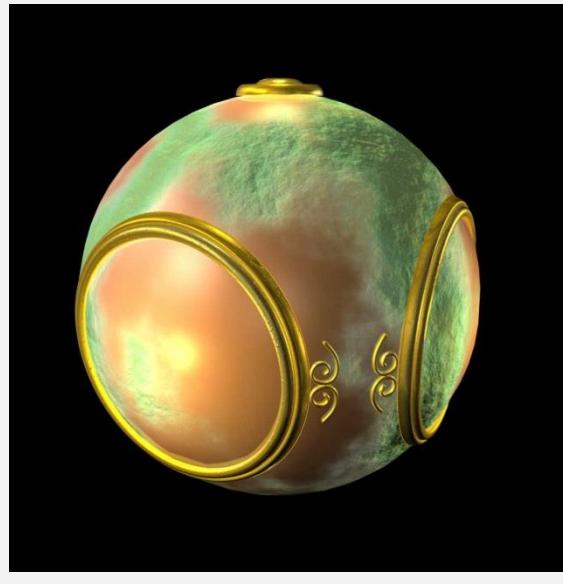
- Realism comes to computer graphics



smooth shading



environment  
mapping



bump mapping

# Computer Graphics: 1990-2000

- OpenGL API
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers



# Computer Graphics: 2000-2010

- Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, Lightwave
- Programmable pipelines



# Computer Graphics: 2010-2020

- Stylized rendering
- Precomputed Radiance Transfer (PRT)
  - Textures become much more important
- Image/Video processing
- General purpose GPU (GPGPU)

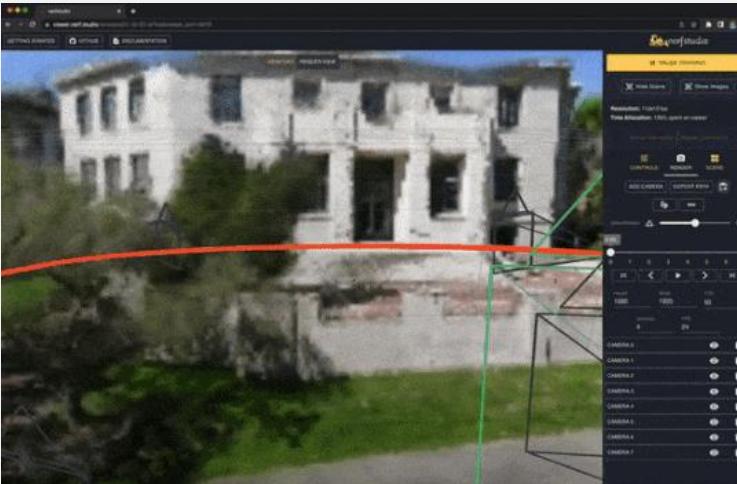


Researchers at NASA want to get a better understanding of what happens when galaxies collide.



# Computer Graphics: 2020-

- Stable Diffusion (SD)
- Neural 3D Rendering (NeRF)
- OpenAI Sora



# The Future is Here

- 3D Gaussian Splatting ([project](#))
- Edify 3D: Scalable High-Quality 3D Asset Generation ([project](#))

