

Q & A

- Sign up on our BBS for discussion (<http://games-cn.org/forums/forum/games202/>)
- And QQ group!

计算机图形学与混合现实研讨会
GAMES: Graphics And Mixed Environment Seminar

首页 活动通知 往期报告PPT&视频 在线课程 ▾ GAMES线下会议 ▾ 招聘信息 ▾ 讨论区 更多资源 ▾ 其他信息 ▾

主页,Forums,Games202-高质量实时渲染 Sort topics by votes

This forum has 2 topics, 2 replies, and was last updated 4 hours, 52 minutes ago by George.

Viewing 2 topics - 1 through 2 (of 2 total)

FOLLOW:

NEXT STORY
GAMES在线课程 (现代计算机图形学入门) 讨论区 >



Assignments

- Assignments
 - 5 programming assignments (excluding assignment 0)
 - About 1.5 week for each assignment
 - Language: OpenGL Shader Language (GLSL)
 - Code skeleton will be provided (Javascript + WebGL)
- Submission
 - Submit your project by 11:59PM AoE (Anywhere on Earth) on/before the due dates
 - Feedback will be provided in a week

Assignments

- A quick look at our assignment skeleton



Assignments

- Assignment Submission Website
(<http://smartchair.org/GAMES202>)
- No Exams
- Course Project / Final Project
 - Starting midway of this course
 - References will be provided, but you decide the topic
 - Best work will be posted online for showing off

The image contains two screenshots of a web application. The top screenshot is the login page for 'GAMES202'. It features a logo with red, green, and blue squares, followed by the text '思澈会议系统'. Below this is a form with fields for 'Email:' (labeled 'Email address') and '密码:' (labeled 'Password'), both with placeholder text. A blue '登录' (Login) button is below the fields. To the right of the fields, there are three links: '请使用思澈会议系统(SmartChair)帐号登录系统注册相应会议及课程。 关于思澈会议系统(SmartChair)帐号' (Please use the SmartChair account to log in and register corresponding meetings and courses. About SmartChair account), '注册新的思澈会议系统(SmartChair)帐号' (Register a new SmartChair account), and '忘记密码?' (Forgot password?). The bottom screenshot shows a course project page titled '高质量实时渲染'. It includes the URL <https://sites.cs.ucsb.edu/~lingqi/teaching/games202.html>, the duration '在线 2021年3月10日 ~ 6月30日' (Online March 10, 2021 ~ June 30, 2021), the instructor '主讲老师: 闫令琪, UCSB (<https://sites.cs.ucsb.edu/~lingqi>)', and the course homepage '课程主页: <https://sites.cs.ucsb.edu/~lingqi/teaching/games202.html>'.

No Need to Use An IDE!

- IDE: Integrated Development Environment
- An IDE Helps you parse a entire project
 - And gives hints on syntax / usages of member functions, etc.
- Since you'll be focusing on writing shaders most of the times in this course
 - No need to use an IDE this time
 - A text editor is perfectly fine
 - Sublime Text, Vi / Vim, Emacs, etc.
 - Online text editors are also great

Academic integrity

- Work alone for regular assignments
 - no copy-pasting from any other sources
- Do not publish your code (on Github, etc.) for assignments using our skeleton code
- Do not post your solution online
 - Discussion / explanation is welcomed

Questions?

Today's Lecture

Outline

- Motivation
- Evolution of real-time rendering
- Technological and algorithmic milestones
 - Programmable graphics hardware
 - Precomputation-based methods
 - Interactive Ray Tracing

Motivation

- Today, Computer Graphics is able to generate **photorealistic** images
 - Complex geometry, lighting, materials, shadows
 - Computer-generated movies/special effects (difficult or impossible to tell real from rendered...)



[Artist: Teruyuki and Yuka]



[Artist:
Hyun Kyung]

Motivation

- But accurate algorithms (esp. ray tracing) are **very slow**
 - So they are called **offline rendering** methods
 - Remember how long it takes to render **1 frame** in Zootopia?



Zootopia, Disney Animation

Motivation

- With proper approximations, we can generate **plausible** results but runs much faster



Toyota 2000GT, from TurboSquid
(offline rendering)



Final Fantasy XV
(real-time rendering)

Evolution of Real-Time Rendering

- Interactive 3D graphics pipeline as in OpenGL
 - Earliest SGI machines (Clark 82) to today
 - Most focus on more geometry, texture mapping
 - Some tweaks for realism (shadow mapping, accum. buffer)



SGI Reality Engine 93
(Kurt Akeley)

Evolution of Real-Time Rendering

- 20 years ago
 - Interactive 3D geometry with simple texture mapping, fake shadows (OpenGL, DirectX)



Final Fantasy VII
(1997)



Counter Strike
(1999)

Evolution of Real-Time Rendering

- 20 -> 10 years ago
 - A giant leap since the emergence of programmable shaders (2000)
 - Complex environment lighting, real materials (velvet, satin, paints), soft shadows



Assassin's Creed II
(2009)

DARK OILY



Resident Evil 5
(2009)

Evolution of Real-Time Rendering

- Today
 - “Stunning graphics”



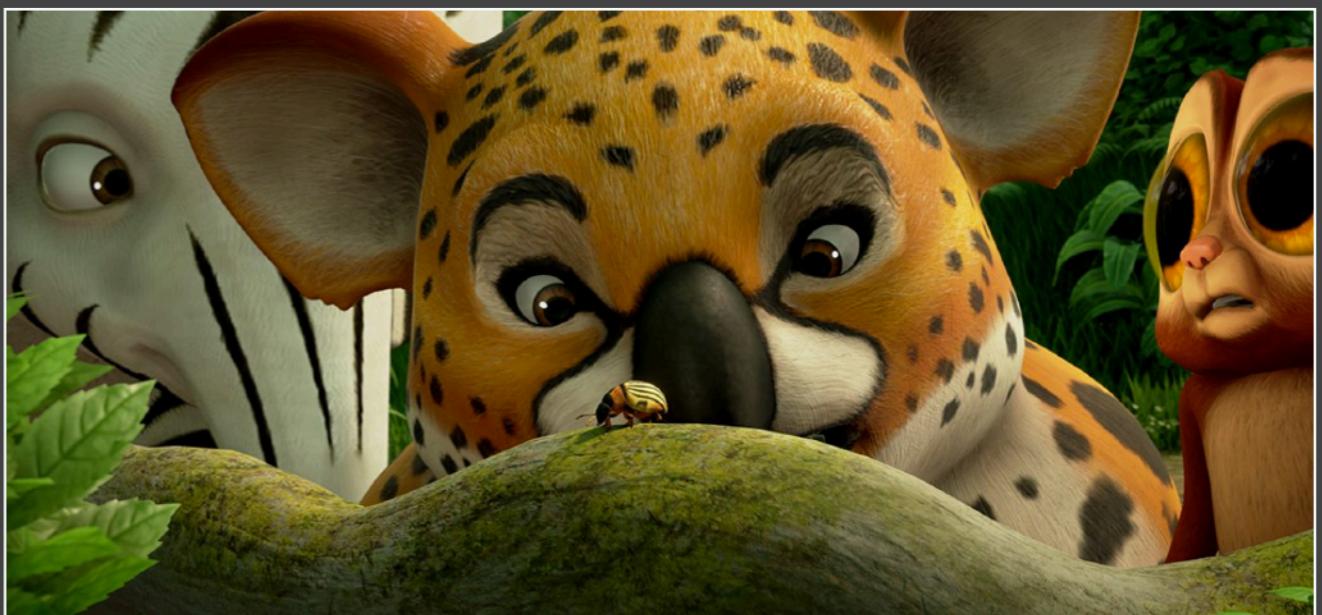
God of War (2018)

Evolution of Real-Time Rendering

- Today
 - Extended to Virtual Reality (VR) and even movies



Beat Saber, VR Game



Zafari, animation series rendered completely using Unreal game engine

Evolution of Real-Time Rendering

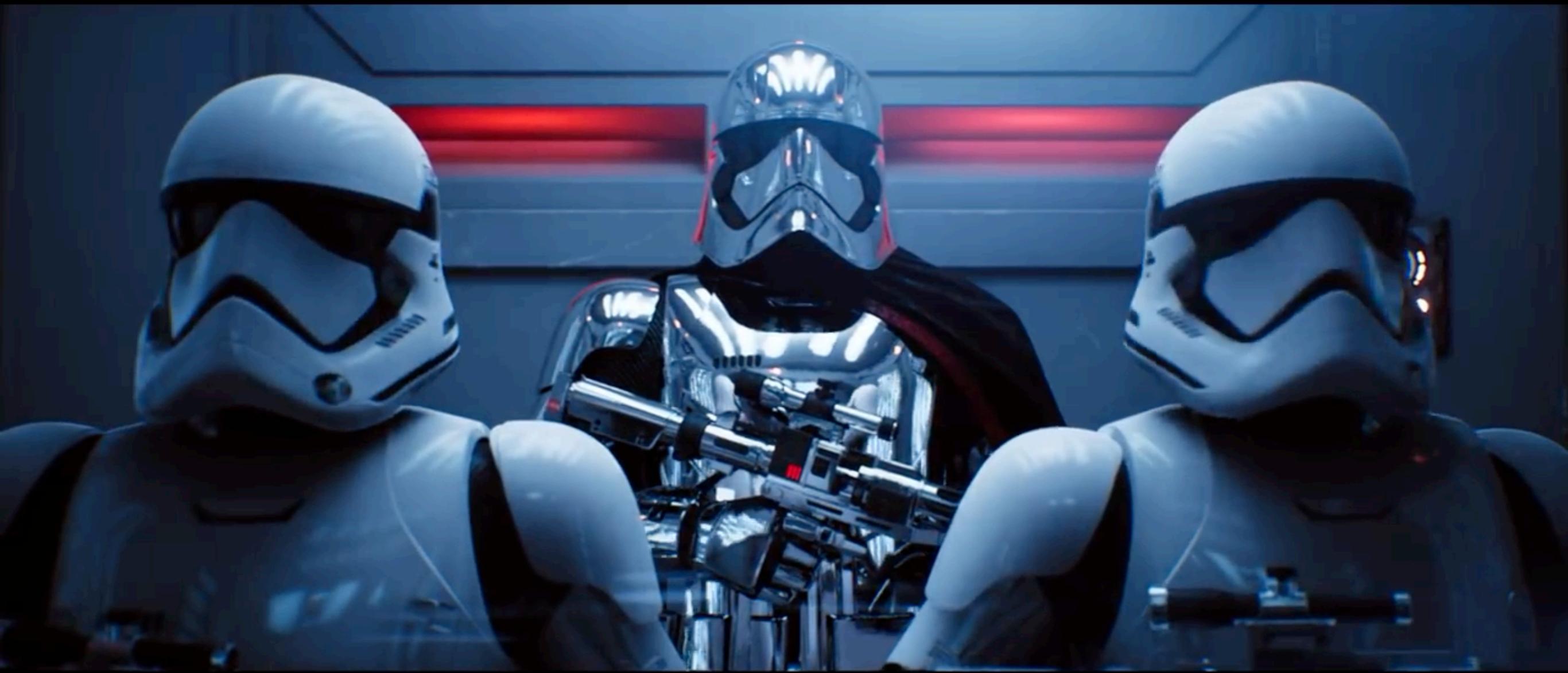
- Today



(2018) - Photorealistic Forests with Unreal Engine 4

Evolution of Real-Time Rendering

- Today



(2018) - Real-Time Ray Tracing Demo, NVIDIA

Evolution of Real-Time Rendering

- In the future



The Matrix (1999 movie)

Evolution of Real-Time Rendering

- In the future

A dark, futuristic scene from the movie Ready Player One. In the center, there is a digital projection of text. The text reads "COLUMBUS, OHIO" on the first line and "2045" on the second line. The text is in a blue, glowing font. There are some blurred lights and shapes in the background, suggesting a high-tech environment.

COLUMBUS, OHIO
2045 —

Ready Player One (2018 movie)

Technological and Algorithmic Milestones

- Programmable graphics hardware (shaders) (20 years ago)



A New Dawn demo, NVIDIA

<https://www.geforce.com/games-applications/pc-applications/a-new-dawn/videos>

Technological and Algorithmic Milestones

- Programmable graphics hardware (shaders)

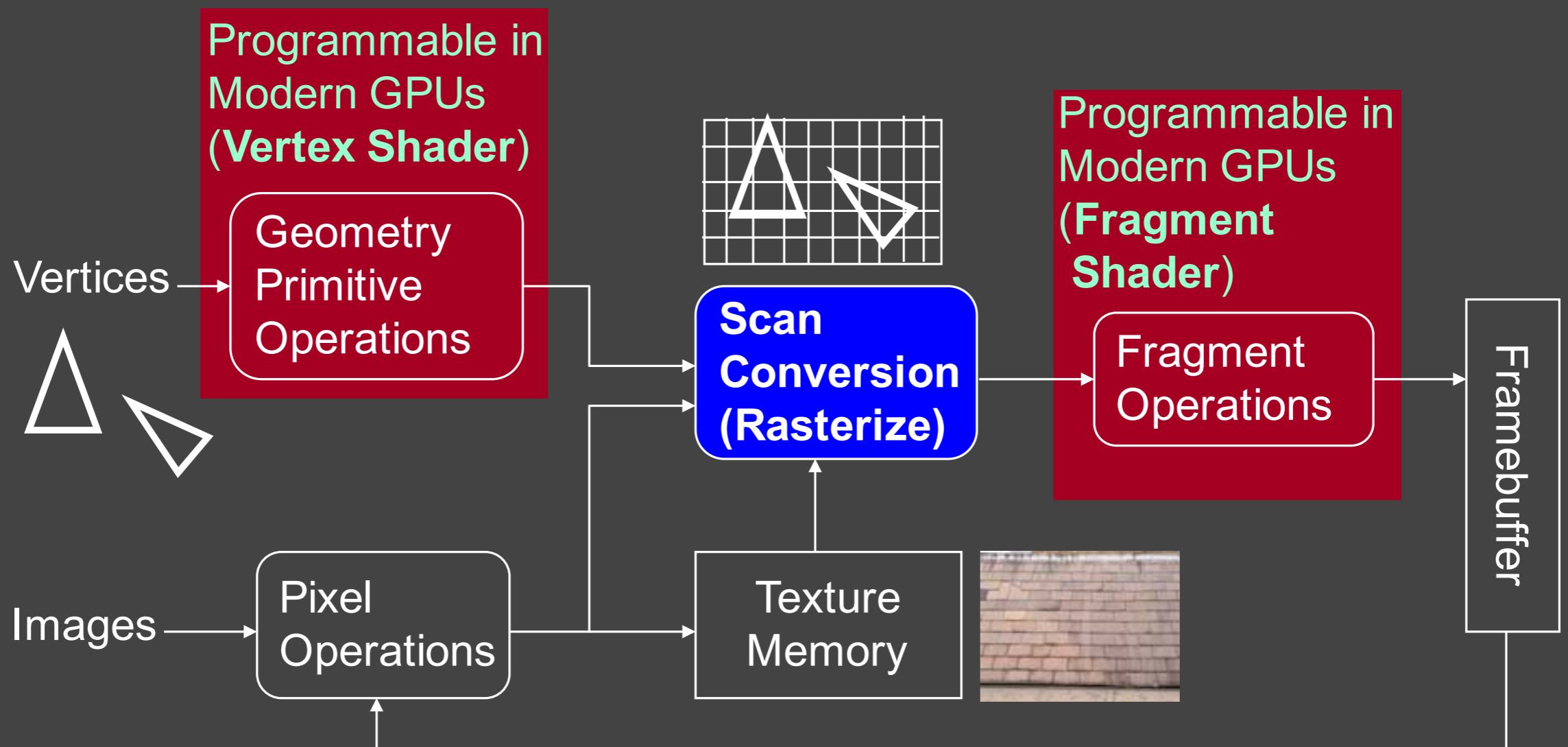


Image courtesy of Prof. Ravi Ramamoorthi

Technological and Algorithmic Milestones

- Precomputation-based methods (15 years ago)
 - Complex visual effects are (partially) pre-computed
 - Minimum rendering cost at run time

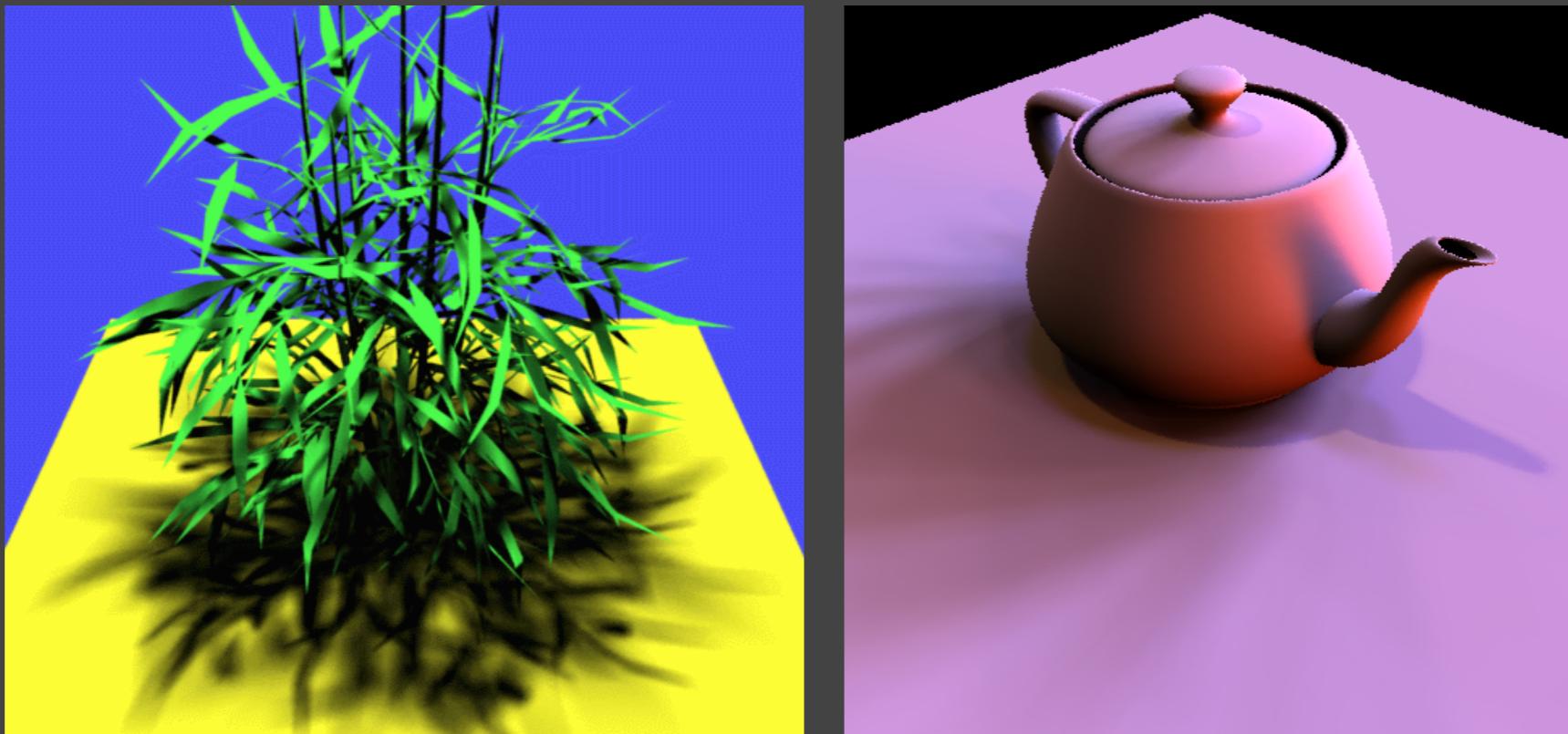


Image courtesy of Prof. Ravi Ramamoorthi

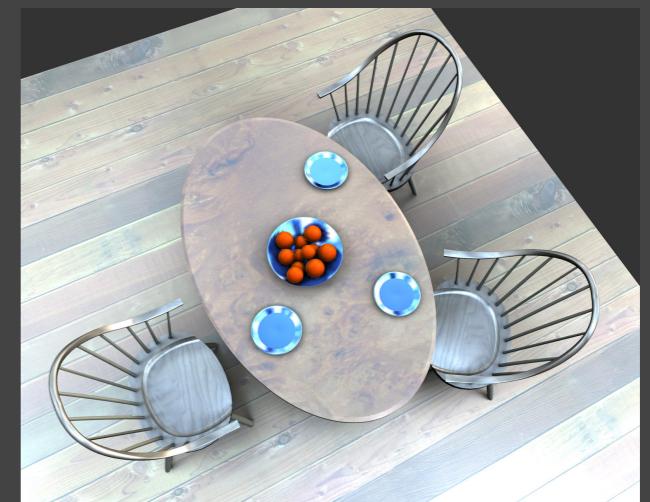
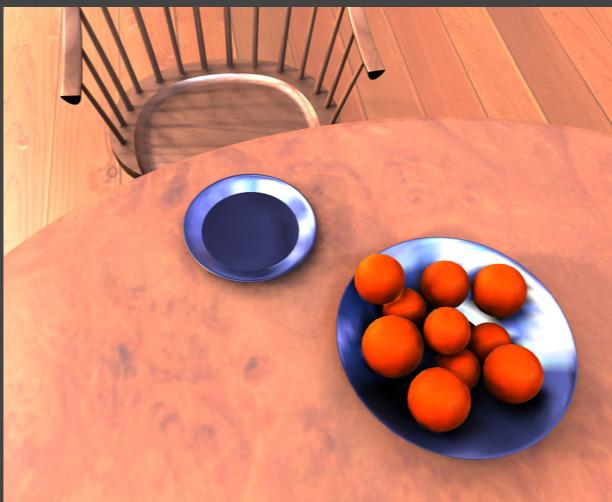
Technological and Algorithmic Milestones

- Precomputation-based methods



Technological and Algorithmic Milestones

- Precomputation-based methods: Relighting
 - Fix geometry
 - Fix viewpoint
 - Dynamically change lighting



[Ng, Ramamoorthi, Hanrahan 04]

Technological and Algorithmic Milestones

- Interactive Ray Tracing (8-10 years ago: CUDA + OptiX)
 - Hardware development allows ray tracing on GPUs at low sampling rates (~1 samples per pixel (SPP))
 - Followed by post processing to denoise



Car interactively rendered
using NVIDIA OptiX



Pixar's real-time
previewer

Questions?

Next Lecture

- A swift and brutal recap of some important concepts
 - Graphics Pipeline
 - Shader Language
 - Rendering Equation
 - Calculus
 - etc.

Thank you!