Medrick
game studio

Real-Time Rendering

Session 1: Introduction

Mehrdad Jafari
Winter 2023

What we’ll do in this series

- Explain & visualize the rendering process
- Overview over features, challenges & solutions
- Understand the big picture, not a detailed one

Table of Contents

1. Intro
2. Before Rendering
3. Geometry Rendering
4. Rasterizing & GBuffer
5. Textures
6. Pixel Shaders & Materials
7. Reflections
8. Static Lighting/Shadows
9. Dynamic Lighting/Shadows
10. Fog & Transparency
11. Post Processing

Intro

- RTR is at its most efficient when there is nothing
- RTR is about managing losses and handling target framerate
- You can’t do RTR perfect

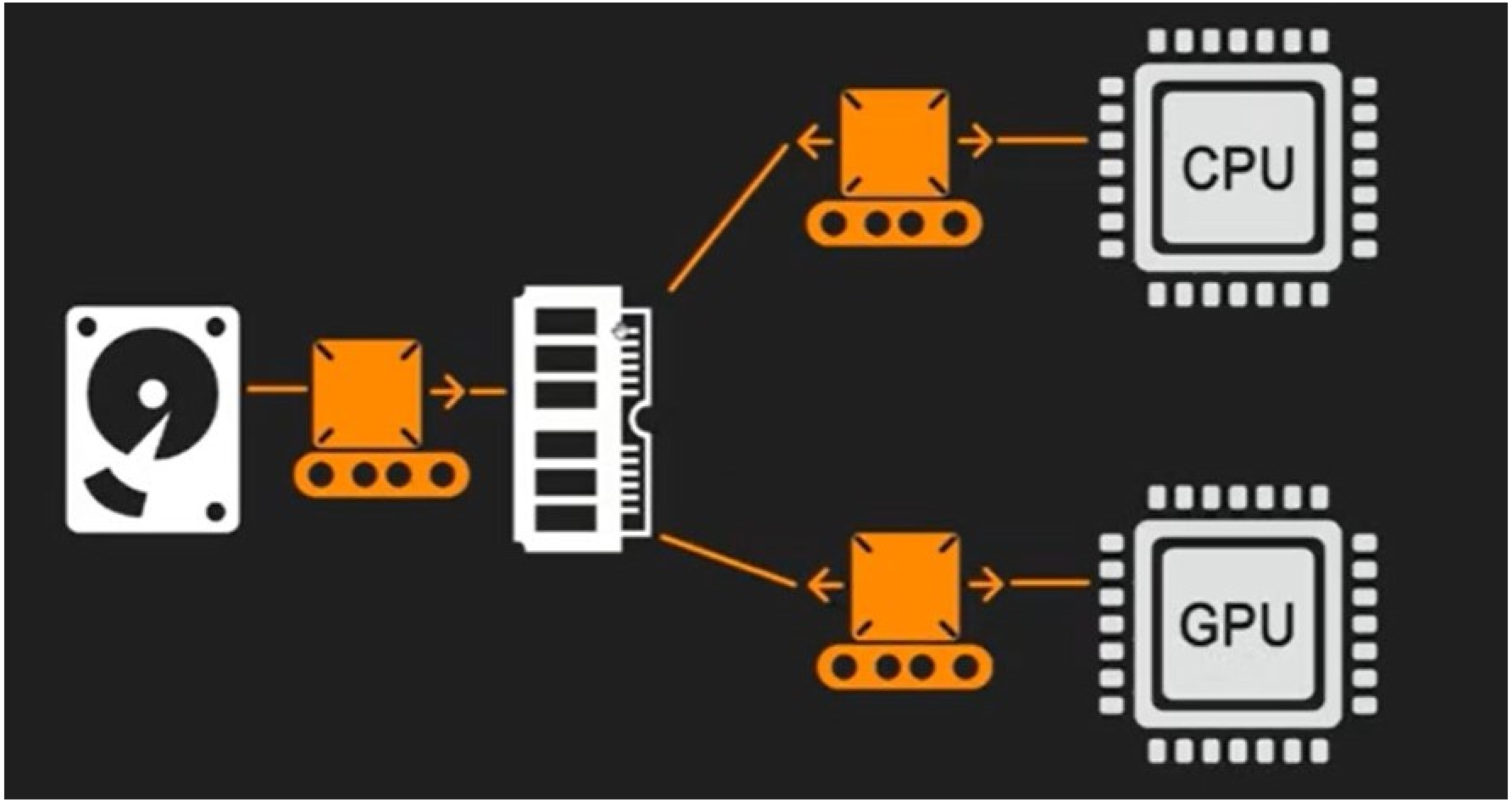
Intro

It’s a tradeoff between gains & losses. You always sacrifice something to gain something else.

1. Everything needs to be efficient as possible
2. We need rigid pipelines & restrictions
3. We need to offload parts to pre-calculations
4. We need a mix of solutions

Intro

Triangles & vertices



Intro

CPU vs GPU

1. Handle different parts of the rendering in sync
2. Can bottleneck each other
3. We should know how the load is distributed between the two

Intro

Deferred vs Forward Rendering

Deferred

1. Shading happens in deferred passes
2. Good at rendering dynamic lighting
3. Good at stable predictable high-end performance
4. More flexible with disabling features, less flexible when it comes to surface attributes
5. No MSAA, just TAA

Intro

Deferred vs Forward Rendering

Forward

1. Shading happens in the same pass as geometry/materials
2. More flexible in how lighting & materials are computed but less flexible when different features are mixed
3. Good at translucent surface rendering
4. Faster for simpler uses
5. Dynamic lighting has a big performance impact
6. MSAA possible

Coming up

Before Rendering & Occlusion

Thank you for your attention!