UNIVERSITY OF DERBY

5CM507 Graphics

Lecture 10 Image Space Effects

Dr Youbing Zhao

2025 年 12 月 1 日

Last Week

Shadow Maps

- Render to Texture
- Light Space Projection
- Shadow Map Artefacts

Contents

Post Processing

Post Processing

- Render to image/texture
- Multi-pass rendering
- Additional info: depth, texture coordinates, ...

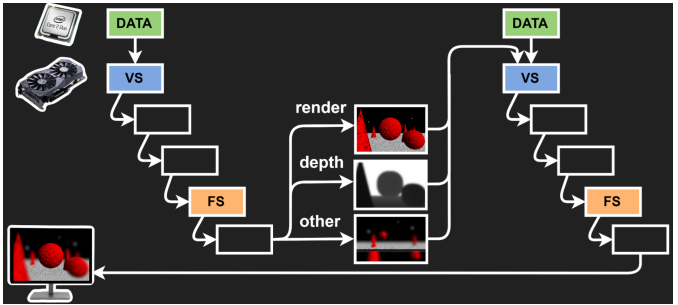


Image Processing

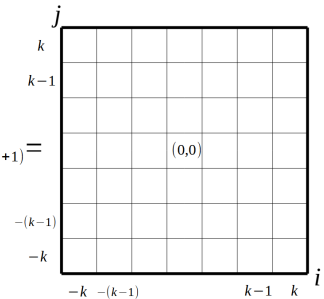
- Pixel
- RGBA channels

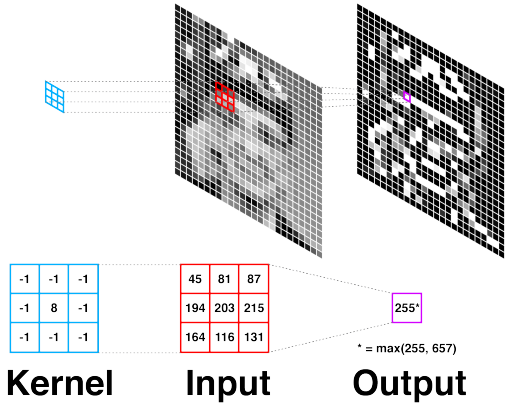
- Smooth, Sharpen, Emboss, Edge Detect, ...

Image Processing : Kernel and Convolution

$$(\mathrm{K} * \mathrm{I})(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} \mathrm{K}(i, j) \cdot \mathrm{I}(x - i, y - j)$$

$K_{(2k+1)\times(2k+1)} =$

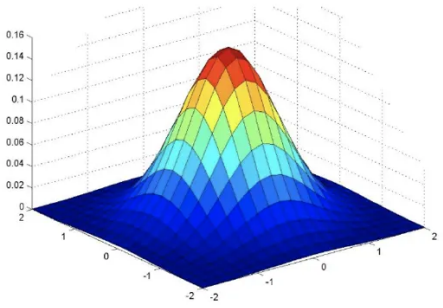




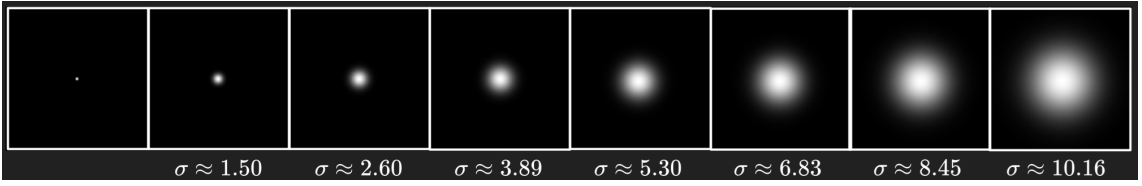
A Sharpening Kernel

Blur

- Box: $\underset{n\times n}{K(i,j)} = \frac{1}{n^2} \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix}}_{n} \Big\} n$

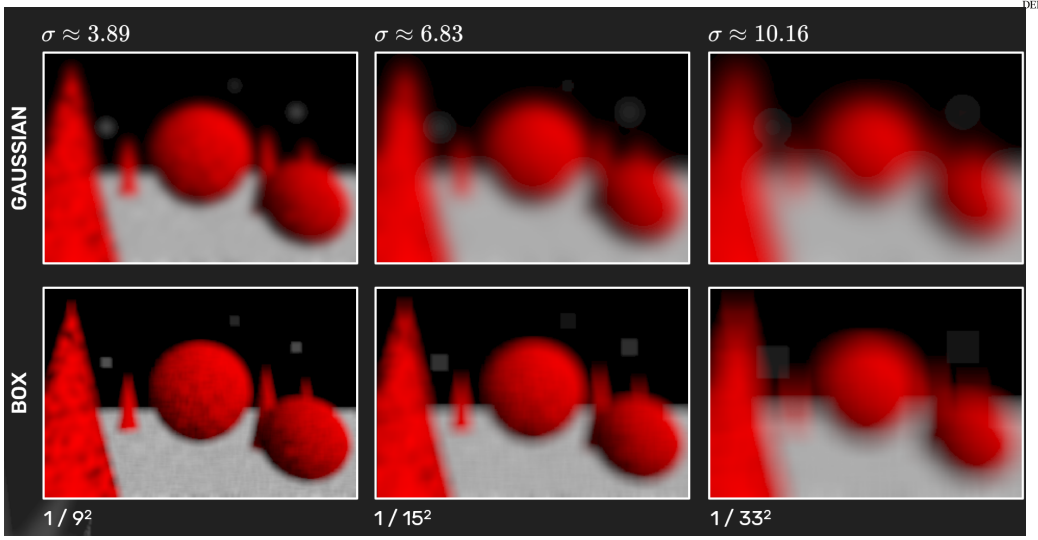


- Gaussian:
$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$



Gaussian kernels (Source: Raimond Tunnel)

Gaussian Blur vs Box Blur



Gaussian vs Box blur (Source: Raimond Tunnel)

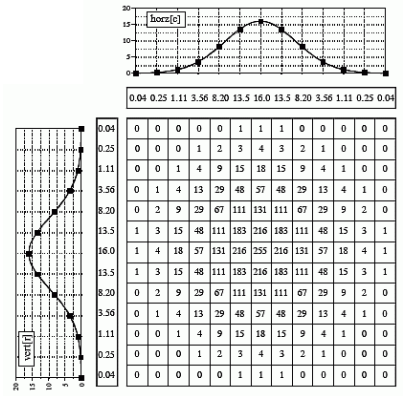
Gaussian Separation

1D Gaussian:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

2D Gaussian:

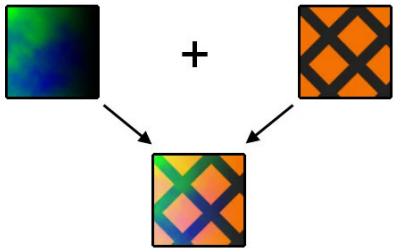
$$\begin{aligned} G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \\ &= g_\sigma(x) \cdot g_\sigma(y) \end{aligned}$$



Gaussian Separation

Blending

- Additive blending: $\mathrm{I} = \mathrm{A} + \mathrm{B}$
- Alpha blending: $\mathrm{I} = \alpha\mathrm{A} + (1 - \alpha)\mathrm{B}$



Fog

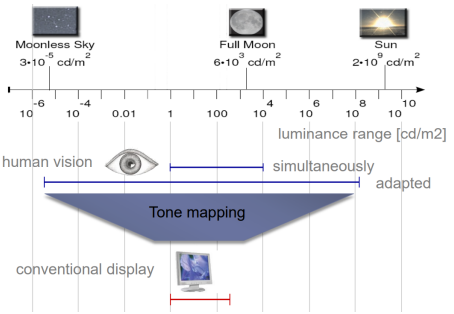
- Distance-based blending
- Fog factor $f \in [\overbrace{0}^{end}, \overbrace{1}^{start}]$

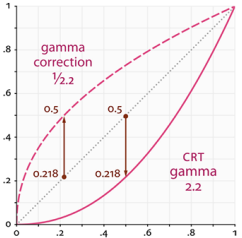
$$C_{blend} = fC_{scene} + (1 - f)C_{fog}$$

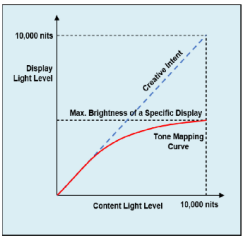
- Linear fog: $f = \frac{end - d}{end - start}$
  - Not physically accurate
- Exponential fog: $f = e^{-kd}$
  - More realistic

HDR Tone Mapping and Gamma Correction

- high-dynamic range (HDR) Tone mapping
  - HDR $\Longrightarrow$ LDR ([0-1.0])
  - Optimises bright and dark parts
  - Reinhard : $C_{out} = \frac{C_{in}}{C_{in}+1.0}$
  - Exposure : $C_{out} = 1.0 - exp(-\frac{C_{in}}{Exposure})$
  - Filmic tone mapping: toe, linear, shoulder
    - Ucharted 2: $C_{out} = \frac{C_{in}(0.22C_{in}+0.03)+0.002}{C_{in}(0.22C_{in}+0.2)+0.06} - \frac{0.01}{0.3}$
    - ACES (Academy Color Encoding System): $C_{out} = \frac{C_{in}(2.51C_{in}+0.03)}{C_{in}(2.43C_{in}+0.59)+0.14}$
- Gamma correction: adjusts the brightness
  - $C_{out} = C_{in}^{\frac{1}{\gamma}}$







Tone Mapping Example

More Interactive Examples

Source: John Hable @ FilmicWorlds

Bloom

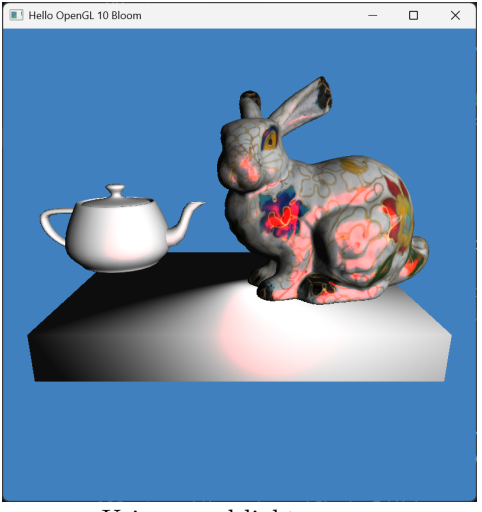
Bloom: Stages

1. Render to Texture
2. Highlight filtering
3. Guassian Blur
4. Blending

Source: Microsoft Spotlight

Pass 1: Render the Scene to Texture

- Create the texture ID
- Create the framebuffer object
- Attach the texture to the framebuffer
- Render the scene



Using a red light source

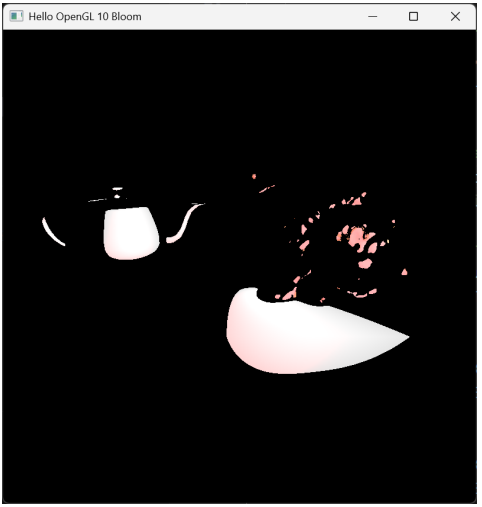
Pass 2: Highlight Filtering

Rendering:

- Read from the texture of Pass 1 rendering
- Render on a rectangle
- Ideally to render to a lower resolution, such as $\frac{1}{4}$ width/height, equiv. to box blur

The filtering fragment shader:

- RGB to intensity :
  $\mathrm{I} = 0.2126\mathrm{R} + 0.7152\mathrm{G} + 0.0722\mathrm{B}$
- Threshold based highlight filtering
  - $C_{in} \geq threshold : C_{out} = C_{in}$
  - Otherwise: $C_{out} = vec3(0.0)$



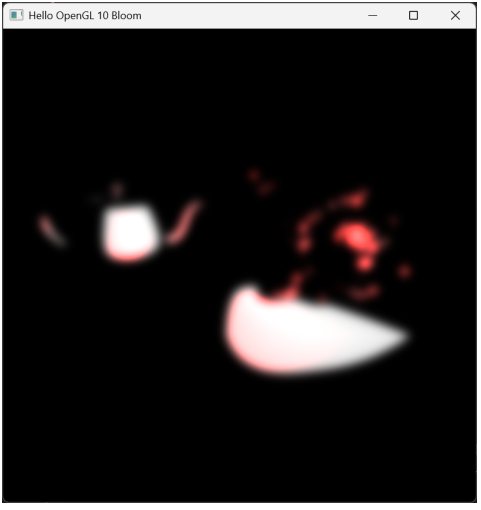
2D Gaussian Blur

Rendering:

- Read from the texture of Pass 2 rendering
- Render on a rectangle
- The same resolution as Pass 2

The Gaussian blur fragment shader:

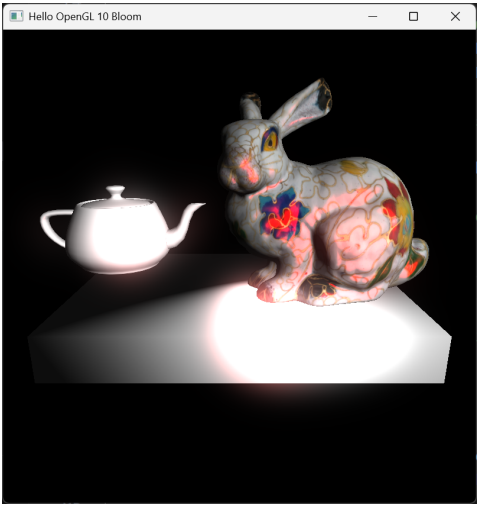
- Use 1d Gaussian blur
- Horizontal and vertical alternatively



Blending

Rendering:

- Read from the texture of Pass 1 and Pass 3
- Render on a rectangle
- The same resolution as of Pass 1
- Additive blending
- Tone mapping and Gamma correction



More Post-processing Effects

More Image Space Effects

- Motion Blur
- Depth of Field

Many variations of algorithms

Requirements of games

- Efficient
- Robust
- At a predictable cost

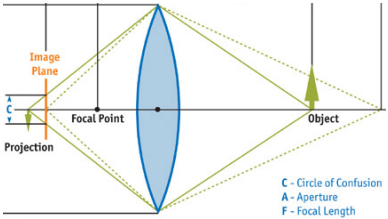
Motion Blur

Sense of Speed, esp. in racing games

- Directioned blur of on-screen rendering
- Pass 1 : render scene and velocity
  - Vertex shader : Project with PVM of both the current and previous frame
  - Fragment shader : Calculate per-pixel velocity and save to velocity texture
- Pass 2 : Directioned blur of Pass 1 rendering
  - Sampling pixels along velocity vector
- Static scenes (camera only) can use only 1 pass
  - Use the velocity vector on their own textures
- More complex with occlusions and speed transparency

Depth of Field

- Compute circle of confusion radius
- Separate scenes
  - Near field
  - Focus field
  - Far field
- Blur non-focused fields in separate images
- Blending



Depth of Field in Witcher 3

Summary

- Image Processing: Kernel and Convolution
- Gaussian blur
- High-dynamic Range (HDR) Tone mapping
- Blending
- Bloom
- Motion blur and depth of field

Questions?