Excel@FIT 2024

HIGH DYNAMIC RANGE RENDERING OF VIRTUAL 3D SCENES

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Autor: Vadim Goncearenco   Supervisor: Ing. Jan Pečiva Ph.D.   Department: UPGM

Tone mapping operators

Global

- Photographic (*Reinhard et al. 2002*)

$$L_d(x,y) = \frac{L(x,y)\left(1+\frac{L(x,y)}{L^2_{\text{white}}}\right)}{1+L(x,y)}$$

- $L_{w(x,y)}$ is the *average luminance* of the scene.
- L_{white} *is the smallest luminance that will be mapped to pure white.*

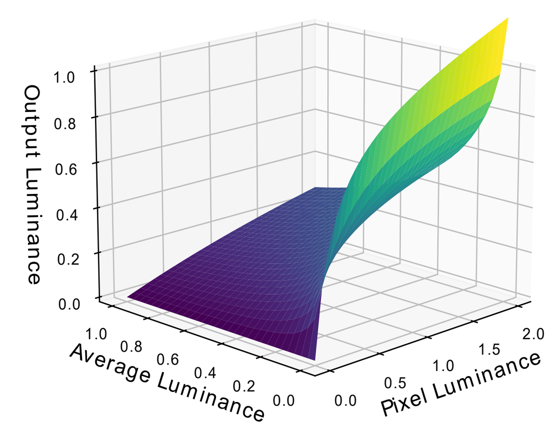

Figure 1: Graph of *Reinhard photographic tone mapping operator.*

- Filmic (*John Hable 2010*)

1. Compute *"S"-like curve*

$$f(x) = \frac{x(Ax+CB)+DE}{x(Ax+B)+DF} - \frac{E}{F}$$

A – F are adjustable coefficients for the curve.

2. Divide by *compressed white point (W)*

$$y = \frac{f(x)}{f(W)}$$

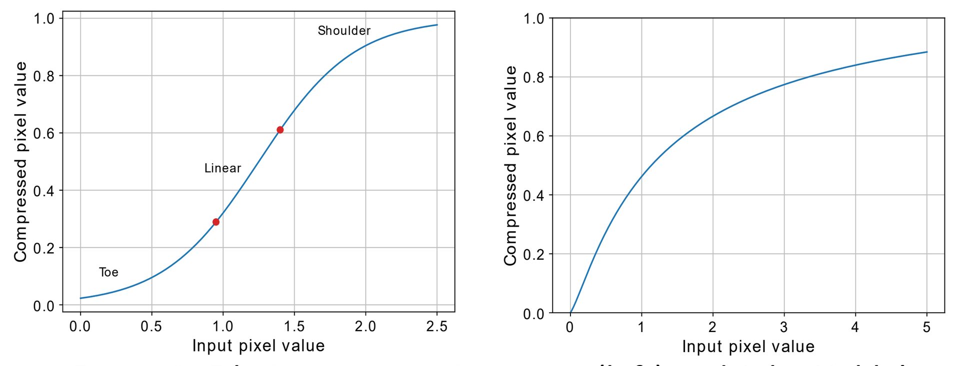

Figure 2: *Filmic* tone mapping curve (left) and *John Hable's Uncharted 2* operator (right).

- Based on ACES (*Krzysztof Narkowicz 2016*)

Filmic curve sampled from *ACES Output Transform*.

Academy Color Encoding System (ACES) is a standard that provides color accurate workflow for the motion picture industry.

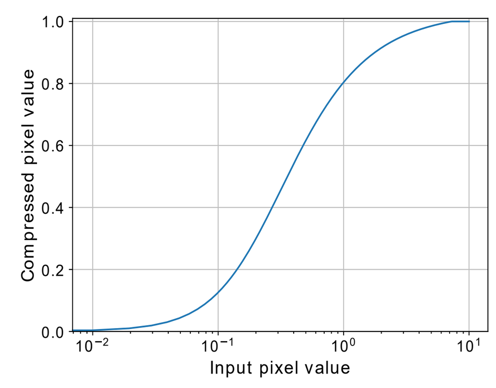

Figure 3: *Filmic* tone mapping operator based on *ACES*.

Local

Operator uses *Bilateral Filter (Durand and Dorsey 2002):*

1. Compute Bilateral filter kernel coefficients:

$$B(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma_d^2} - \frac{\|I_c - I(x,y)\|^2}{2\sigma_r^2}\right)$$

- I_c is pixel located at kernel center.
- $I_{(x,y)}$ is value of the current pixel.
- σ_d is a spatial sigma, an inverse weight for the distance from pixel (x, y) to center.
- σ_r is a range sigma, an inverse weight for intensity difference between pixel $I(x,y)$ and Ic.

2. Normalize each coefficient:

$$B(x,y)^{\text{norm}} = \frac{B(x,y)}{\sum_{x,y} B(x,y)}$$

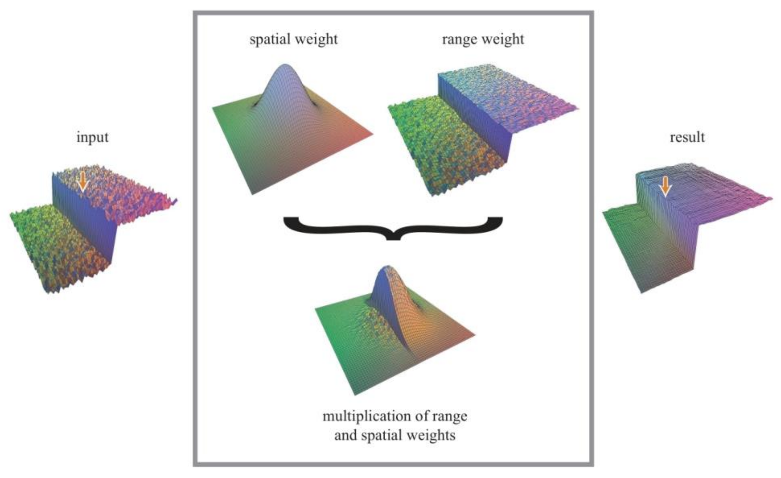

Figure 4: *Bilateral Filter* scheme. [1]

Figure 5: Results of *Bilateral Filter local tone mapping. HDR input (left) vs. tone-mapped (right).*

Eye adaptation

Exponential decay function:

$$L_{new} = L + (L_{avg} - L)\cdot(1 - e^{-\frac{T}{\tau}})$$

Luminance histogram:

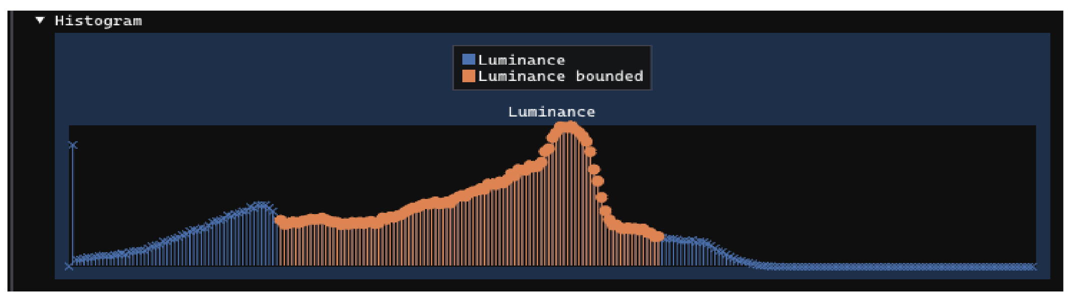

Figure 6: Real-time luminance histogram from my application.

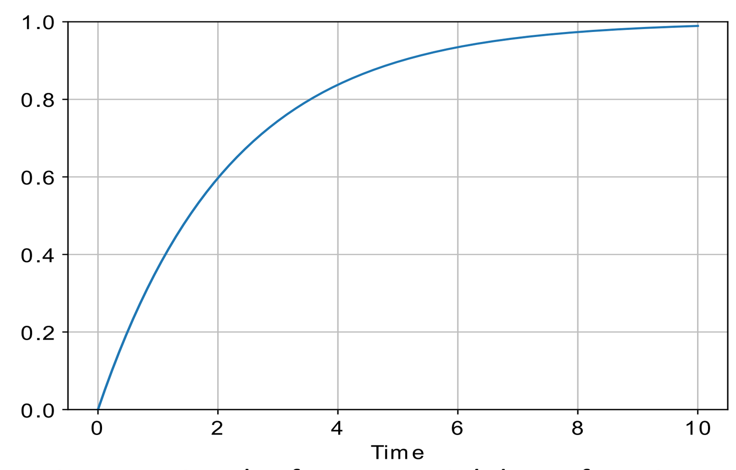

Figure 7: Graph of *Exponential decay* function.

Exposure Fusion

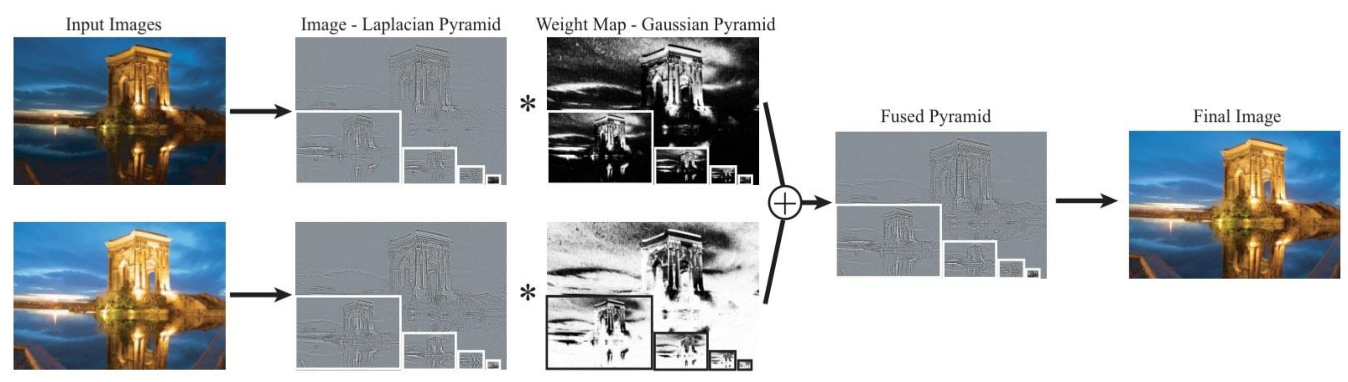

Figure 8: Scheme of Exposure Fusion technique. [2]

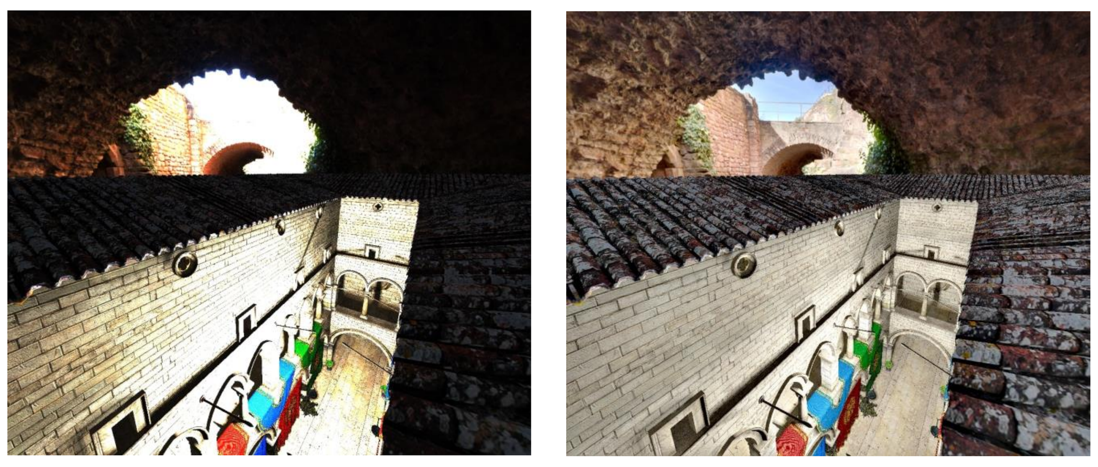

Figure 9: Results of *Exposure Fusion technique. HDR input (left) vs. tone-mapped (right).*

Bloom

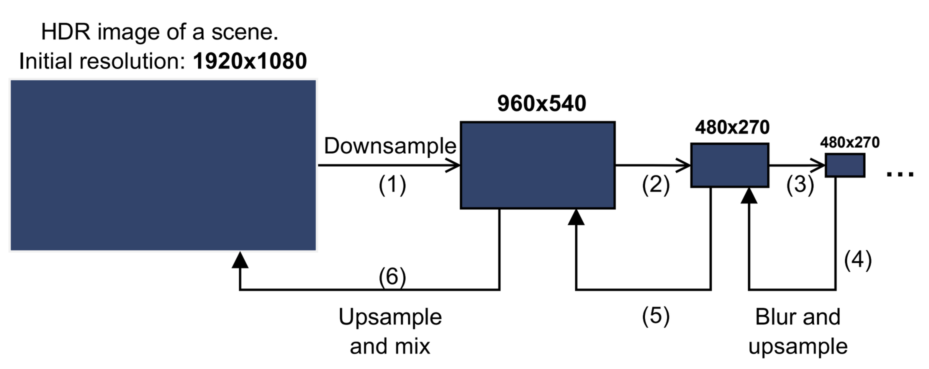

Figure 10: Scheme of *Bloom effect* algorithm.

1. Downsample using the *right kernel*:

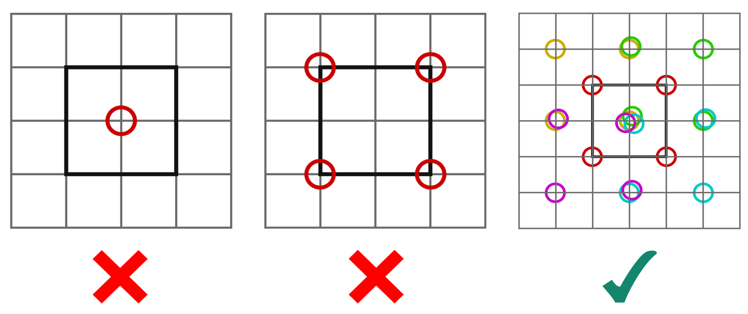

Figure 11: Worse vs. better method to downsample the image texture.

2. Upsample using the *"tent" filter*:

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Figure ?: Upsampling kernel for the *bloom effect*.

Figure 12: Results of *Bloom effect implementation.*

Demo Application (C++, ImGUI)

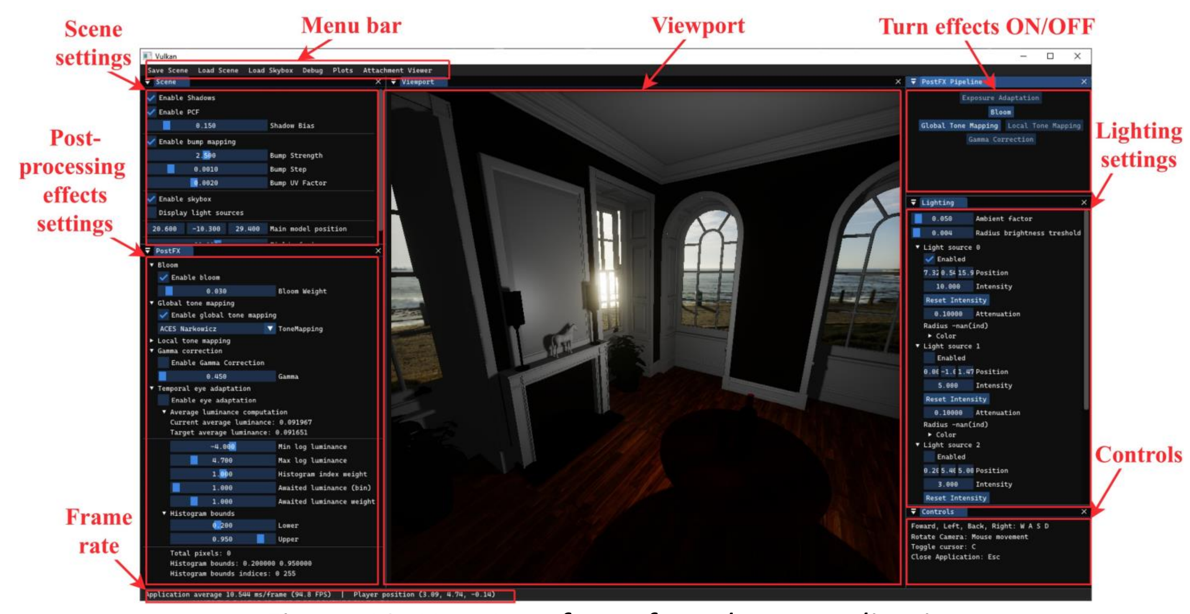

Figure 13: User Interface of my demo application.

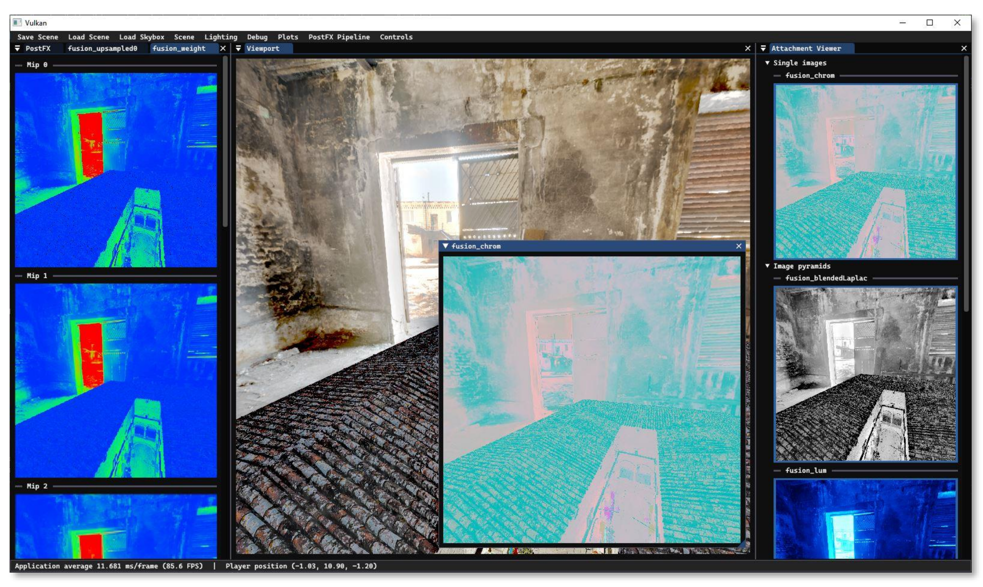

Figure 14: Texture viewer (for debugging and visuzalization).

References:

[1] Mertens, T.; Kautz, J. and Van Reeth, F. Exposure Fusion. In: 15th Pacific Conference on Computer Graphics and Applications (PG'07). October 2007, p. 382–390. ISSN 1550-4085. ISSN: 1550-4085.

[2] Paris, S. A gentle introduction to bilateral filtering and its applications. In: ACM SIGGRAPH 2007 courses. New York, NY, USA: Association for Computing Machinery, August 2007, p. 3–es. SIGGRAPH '07. ISBN 9781450318235. Available at: https://doi.org/10.1145/1281500.1281604.