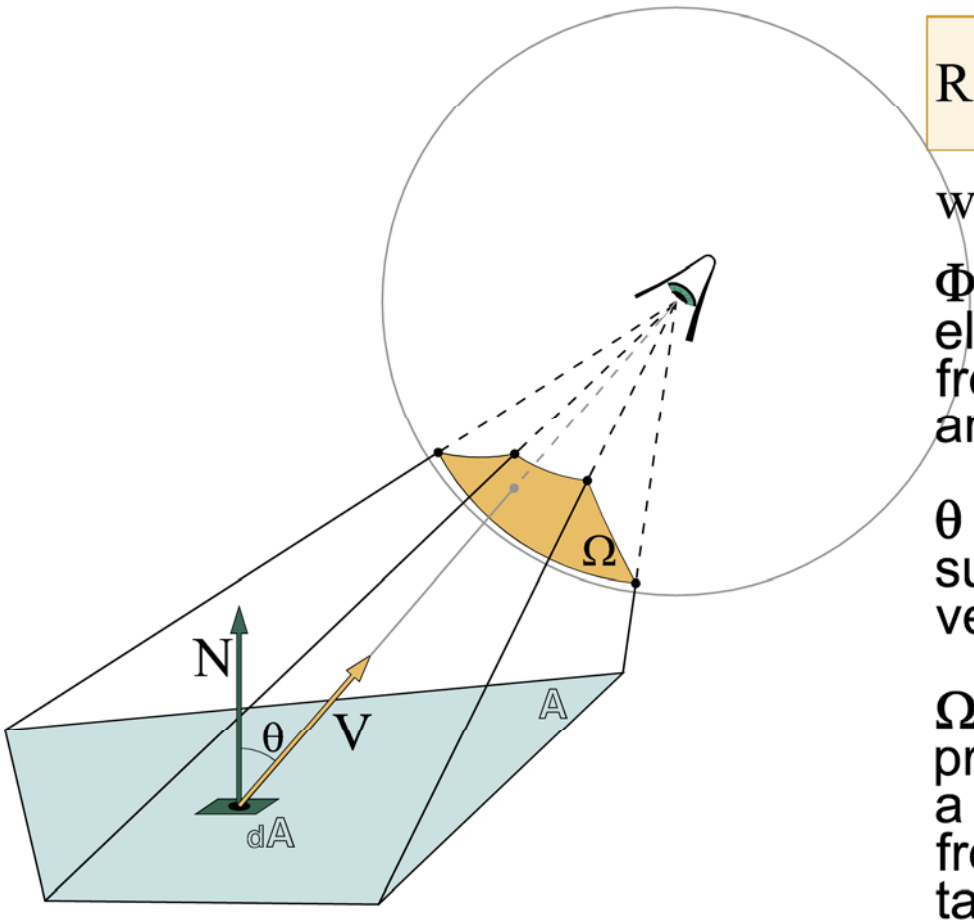
Gamma Correction

Galina Paskaleva

Institute of Computer Graphics and Algorithms

Vienna University of Technology



Radiance in the direction θ:

$$R(\theta) = \frac{d^2\Phi}{dA\,d\Omega\cos\theta}\,[W/m^2sr]$$

where:

Φ [W]... the total power of electro-magnetic radiation in all frequencies emitted by the surface w. area A;

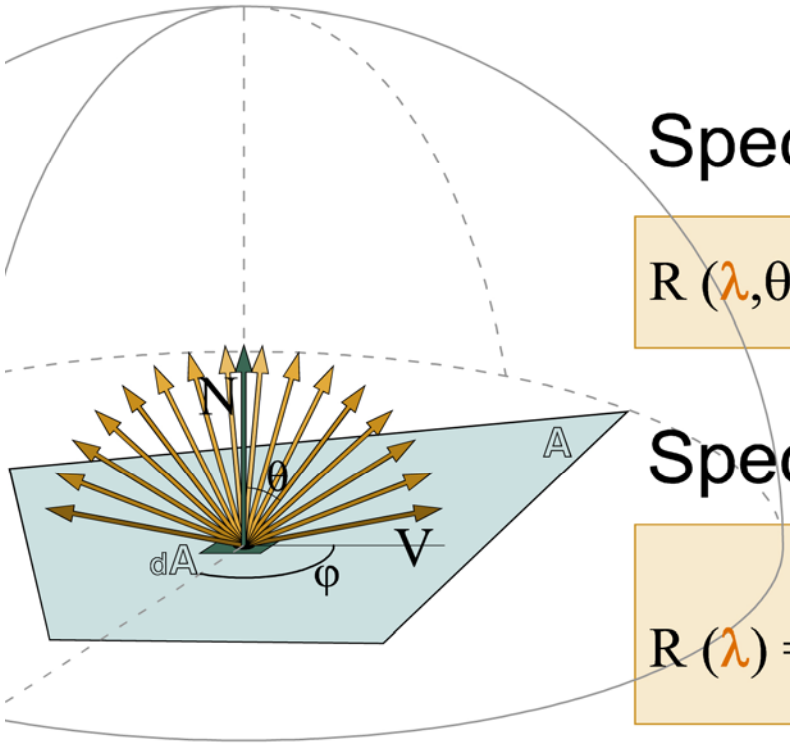
θ [1] ... the angle between the surface normal N and the viewing vector V;

Ω [sr] ... the solid angle obtained by projecting the observed surface onto a unit sphere with center at the point from which the measurement is taken.

Physics

Radiance:

$$R = \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi/2} R(\theta) \cos\theta \sin\theta \, d\theta \, d\varphi \quad [W/m^2]$$



Spectral Radiance in the direction θ:

$$R(\lambda,\theta) = \frac{d^2\Phi(\lambda)}{dA\,d\Omega\cos\theta} \quad [W/m^2sr]$$

Spectral Radiance:

$$R(\lambda) = \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi/2} R(\lambda,\theta) \cos\theta \sin\theta \, d\theta \, d\varphi \quad [W/m^2]$$

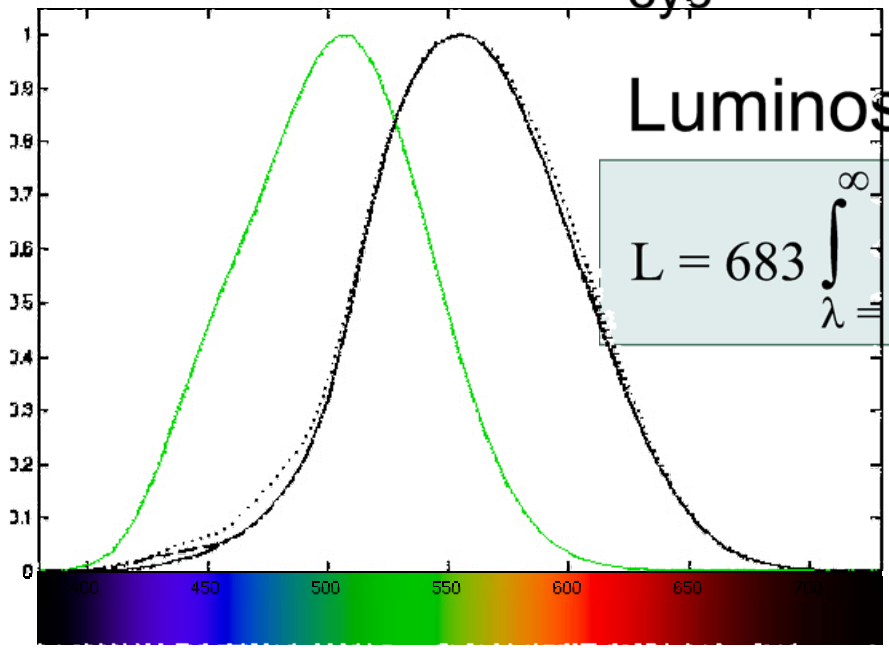
Psychophysics

Luminosity for monochromatic light:

$$L(\lambda) = 683\ f(\lambda)\ R(\lambda) \quad [cd/m^2]$$

where:

$f(\lambda)$ ... the standard luminosity function, describing the average sensitivity of the human eye



Luminosity:

$$L = 683 \int_{\lambda=0}^{\infty} f(\lambda)\ R(\lambda)\ d\lambda \quad [cd/m^2]$$

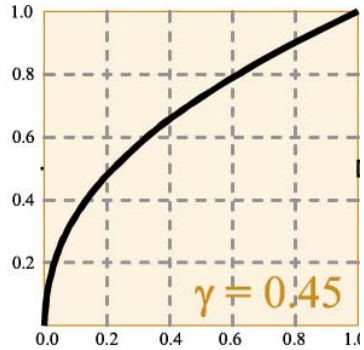
Steven's Law

- describes the relation between
  - Radiance (R) ~ Intensity (**I**) and
  - Luminosity (L) ~ Brightness (**B**)
  - by approximating the integral

$$L = 683 \int_{\lambda=0}^{\infty} f(\lambda)\, R(\lambda)\, d\lambda \quad [cd/m^2]$$

as

$dB/B = \gamma \cdot dI / I$ → $B = c \cdot I^{\gamma}$



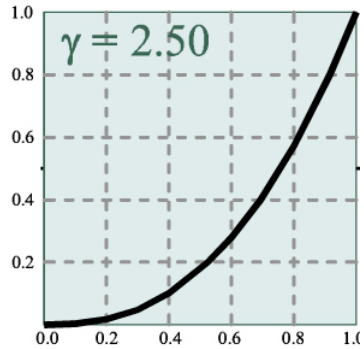
- in a PC $\gamma = 0.45$
- in a Mac $\gamma = 0.55$

CRT Display

- CRT has a non-linear response to the input signal, described as the relation between
  - Voltage (**V**) and
  - Intensity (**I**)

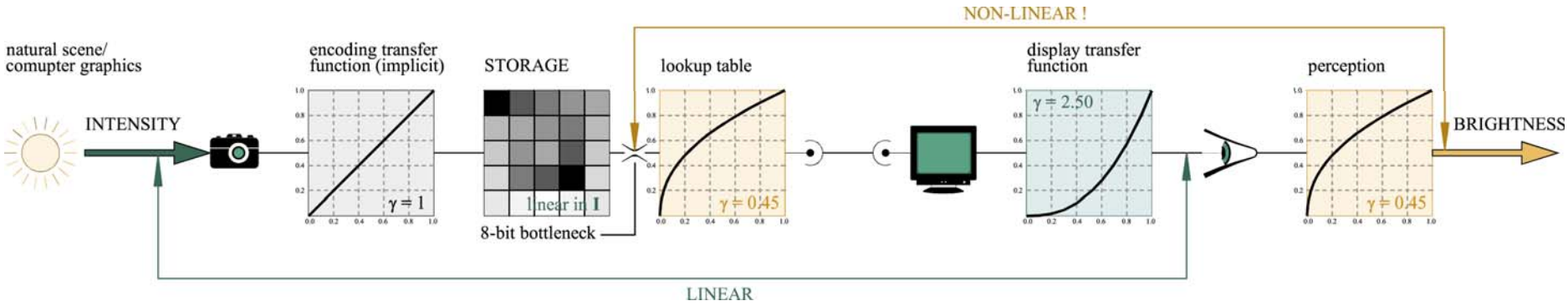
$$I = a \cdot V^{\gamma'} + b$$

where $\gamma' = 2.5$

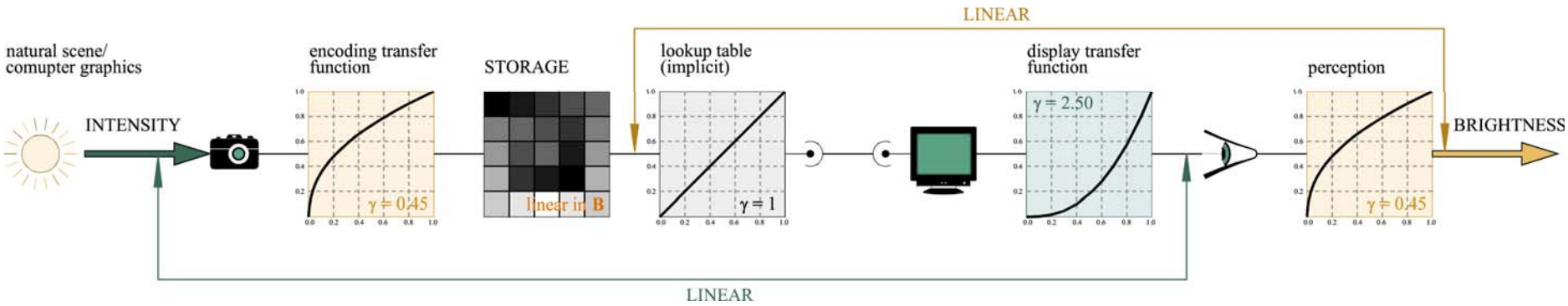


- incidentally 0.45 x 2.5 ~= 1.0

Transfer Functions

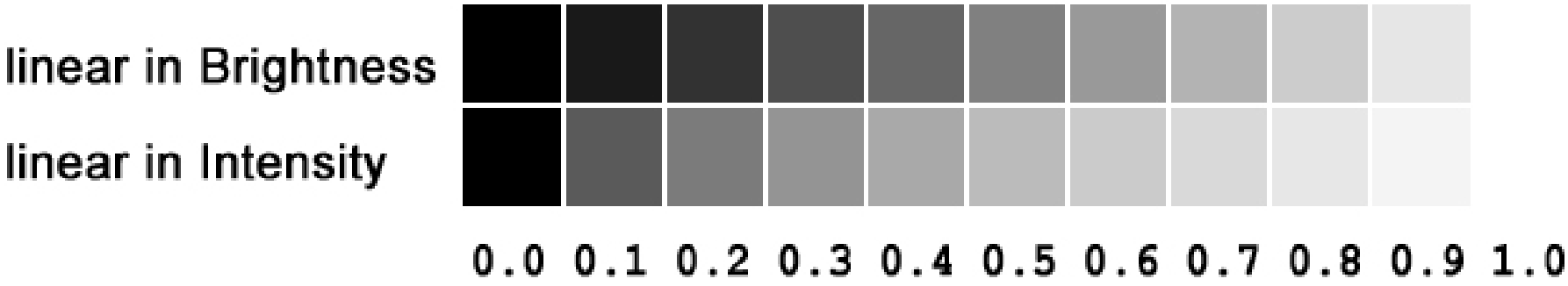


- storage linear in intensity (practically unused)



- storage linear in brightness (most common)

Storage Types



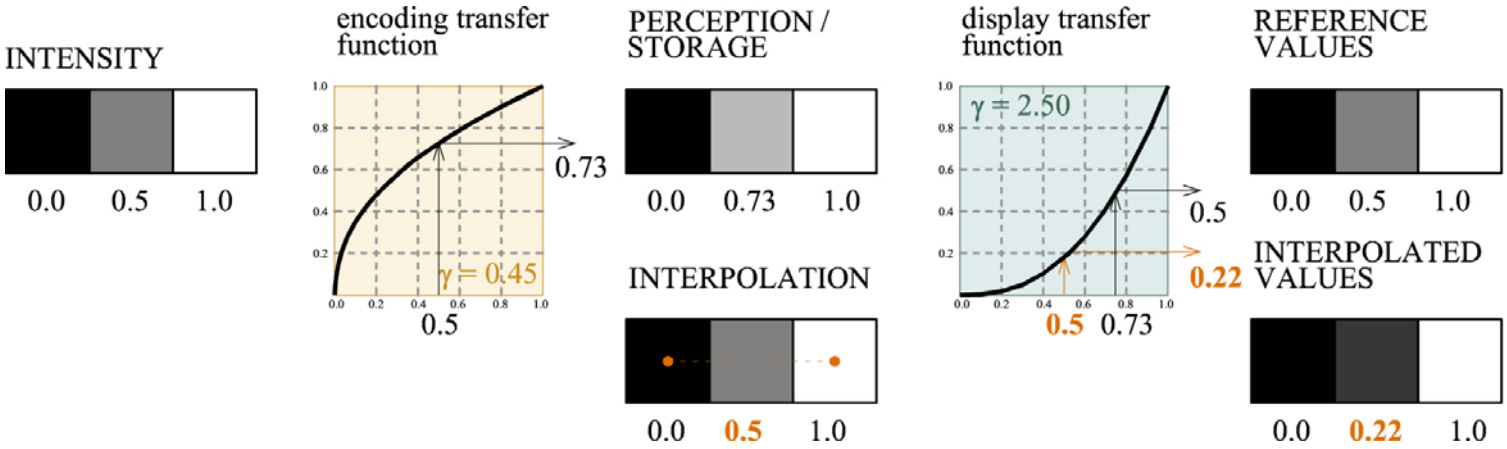
Digital Textures

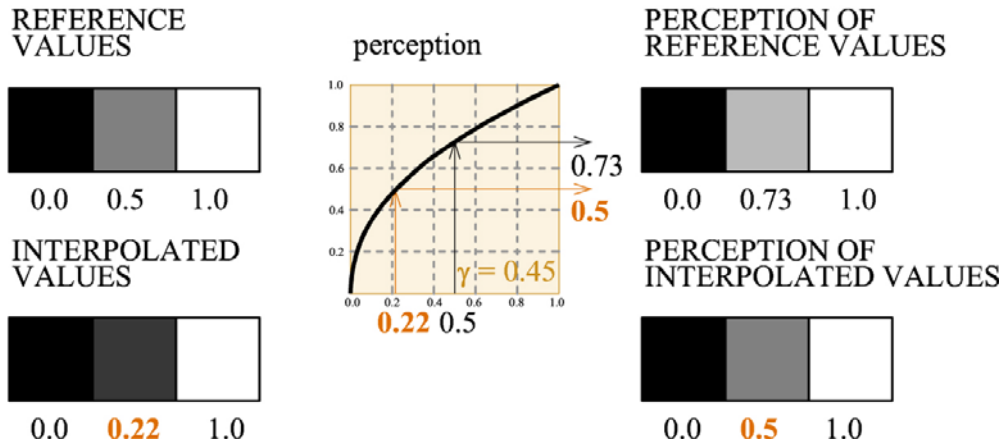
- If an image looks good in a web browser, it has, in all probability, already been gamma-corrected
- Alpha-channels, Height or Displacement Maps, on the other hand, are not
- JPEG files are pre-corrected for a $\gamma = 2.2$
  - I.e. they are linear in brightness but non-linear in intensity

Linearity

- a linear transformation is characterized by:
  - $f(x + y) = f(x) + f(y)$ and
  - $f(\lambda x) = \lambda f(x)$
- the transfer of intensity to brightness g is **not** linear:
  - $g^{-1}(g(I_1)) = I_1$ but
  - $g(I_1 + I_2) \neq g(I_1) + g(I_2)$ and
  - $g(\lambda I_1) \neq \lambda g(I_1)$
- do we perform lighting and filtering with the intensity I or with the brightness B = g(I)?

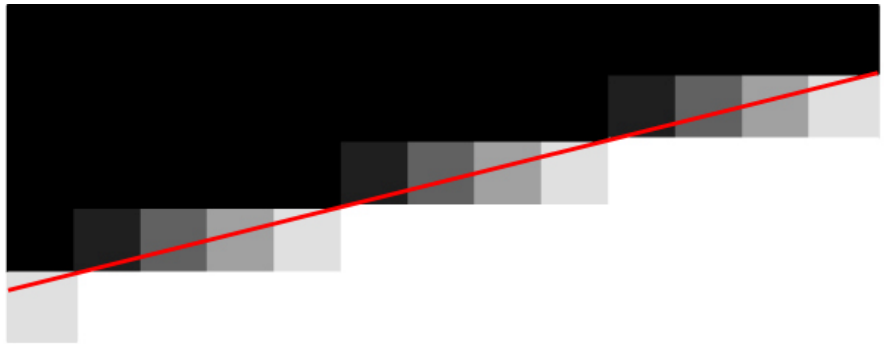
Interpolation w/o Gamma Correction



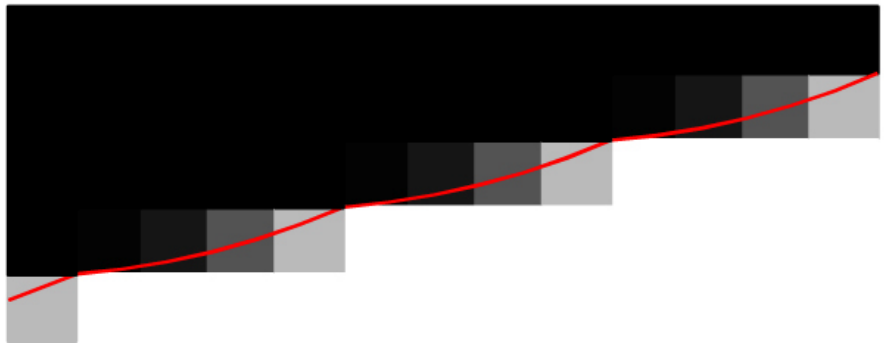


“Roping” Artefacts

- “Bending” of anti-aliased edges due to the non-linearity of brightness space
  - interpolated in intensity space

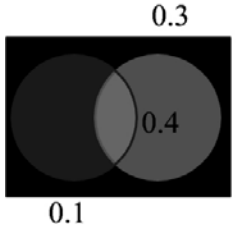


  - interpolated in brightness space



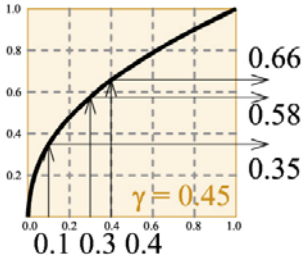
Lighting w/o Gamma Correction

INTENSITY CONTRIBUTION LIGHT 1

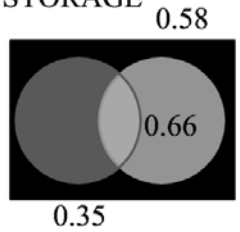


INTENSITY CONTRIBUTION LIGHT 2

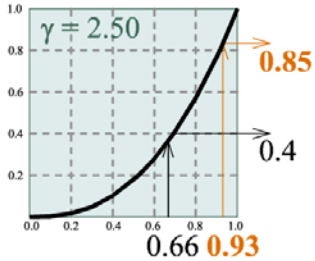
encoding transfer function



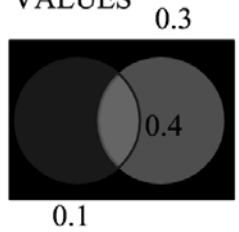
PERCEPTION / STORAGE



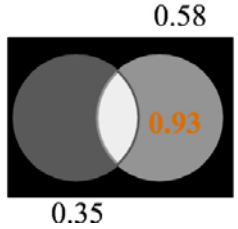
display transfer function



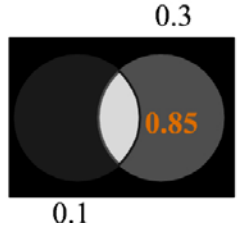
REFERENCE VALUES



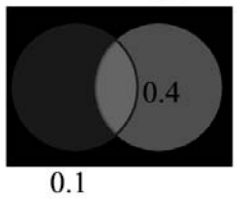
SUM



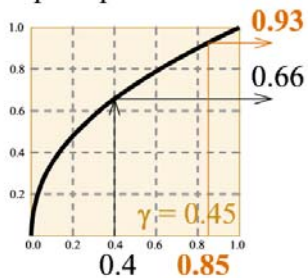
SUMMED VALUES



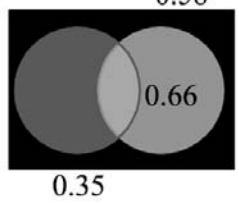
REFERENCE VALUES



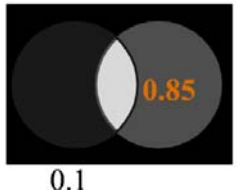
perception



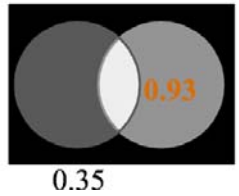
PERCEPTION OF REFERENCE VALUES



SUMMED VALUES



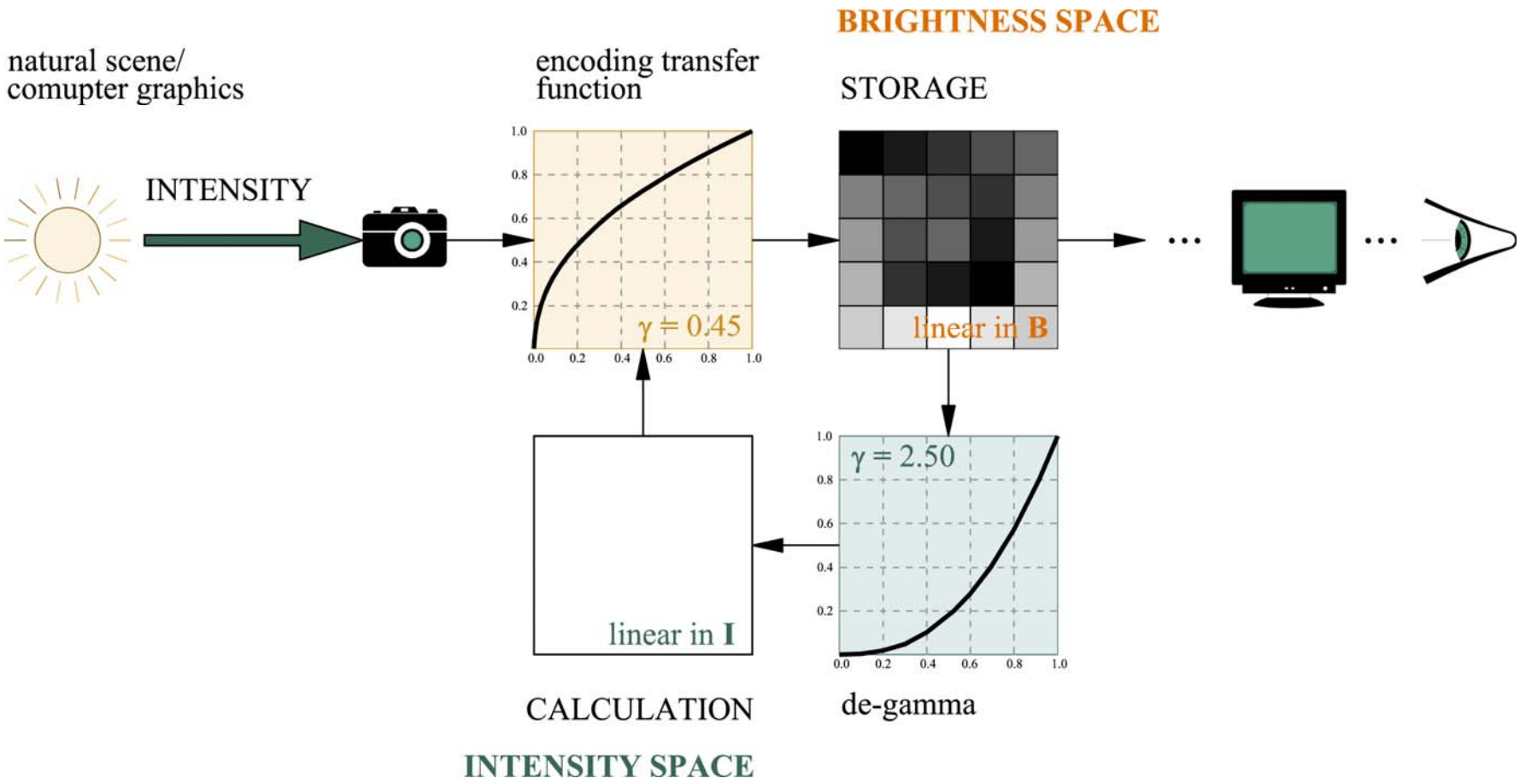
PERCEPTION OF SUMMED VALUES



Incorrect Lighting Calculation

- gamma-corrected / not gamma-corrected

Store and Calculate in **different spaces**



sRGB Texture

- sRGB texture format with $\gamma = 2.2$
  - proper gamma is applied by the **HW** before the results of texture fetches are used for shading
    - **can** be done manually
  - all samples used in a texture filtering operation are linearized by the **HW** before filtering
    - **cannot** be done manually
  - SRGB8 and SRGB8_ALPHA8 are stored as 8-bit unsigned fixed-point values

sRGB Frame Buffers

- final re-application of the gamma correction by the **HW** before display on the monitor
- any value returned in the Shader is gamma-corrected by the **HW** before storage in the frame buffer (or render-to-texture buffer)
- blending is automatically performed in linear intensity space and the result is also automatically gamma-corrected
  - Alpha values are not affected
- use sRGB intermediate color buffers

sRGB in OpenGL 4.1

- visible in the Application:

```
glActiveTexture(GL_TEXTURE1);
glBindTexture(GL_ARB_TEXTURE_SRGB, texture);

glEnable(GL_ARB_FRAMEBUFFER_SRGB);
glBindFramebuffer(GL_DRAW_FRAMEBUFFER, fbuffer);
```

- no influence in the Shader:

```
in vec2 ex_texCoord;
uniform sampler2D un_texture;
vec4 color = texture2D(un_texture, texCoord);
```


w/o sRGB in OpenGL 4.1

- visible in the Shader:
  - on fetch:

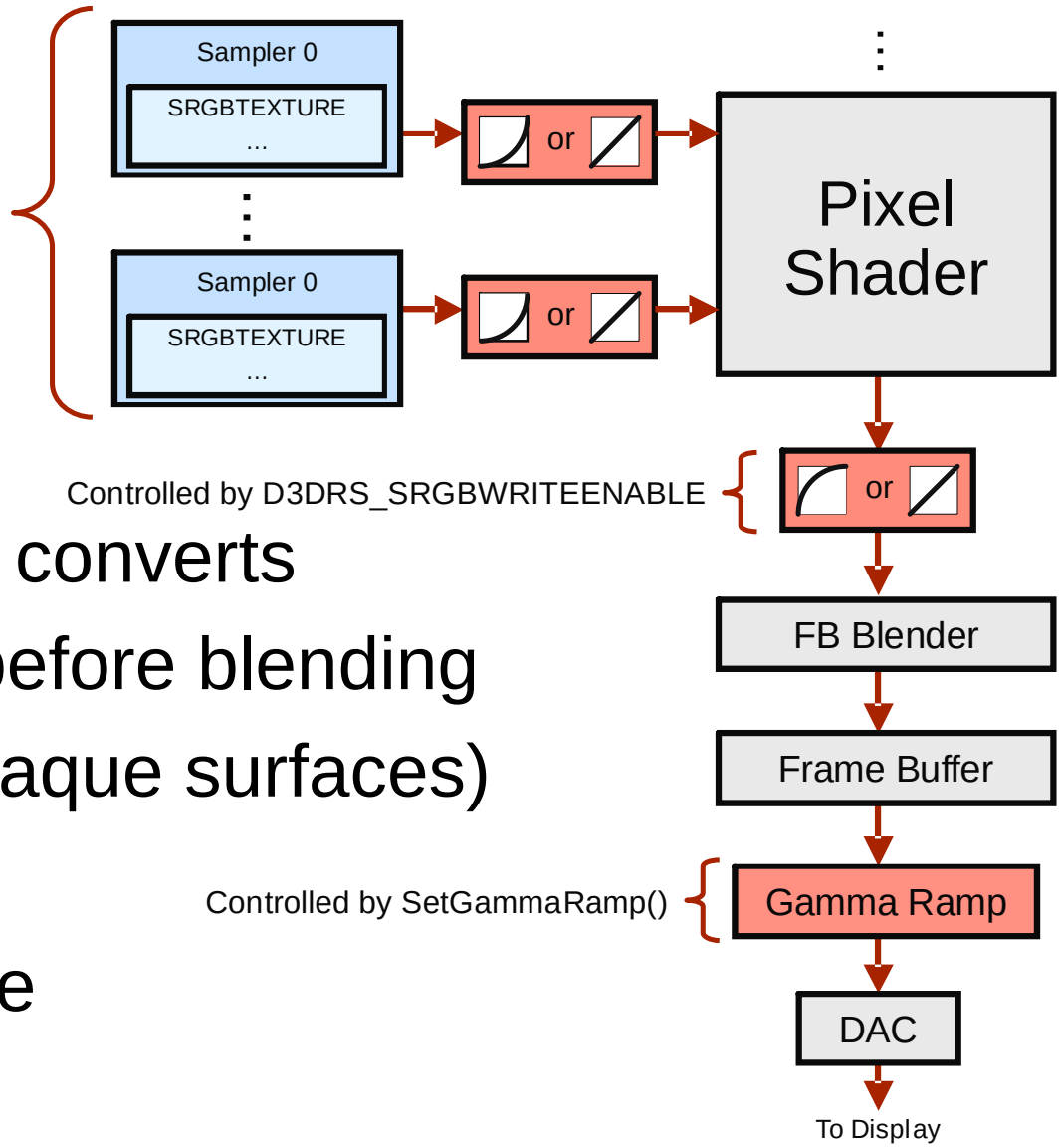
```
vec4 color = pow(texture2D(un_texture,
  texCoord),2.2);
```

  - on last-stage output:

```
return float4(pow(finalCol, 1.0 / 2.2),
  pixelAlpha);
```

  - **filtering** and **mipmapping** is performed in **non-linear** space!

DirectX 9 sRGB and Gamma



- DirectX 9 incorrectly converts into non-linear space before blending (not problematic for opaque surfaces)
- DirectX 10 performs blending in linear space

References and Further Reading

- [1] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, „Real-Time Rendering“, A. K. Peters, Ltd., Wellesley, MA 02482, 2008.
- [2] http://en.wikipedia.org/wiki/Gamma_correction
- [3] http://www.poynton.com/GammaFAQ.html
- [4] OpenGL Version 4.1 Core Profile Specification