Computer Graphics

- Tone Mapping -

Marcus Magnor
Philipp Slusallek

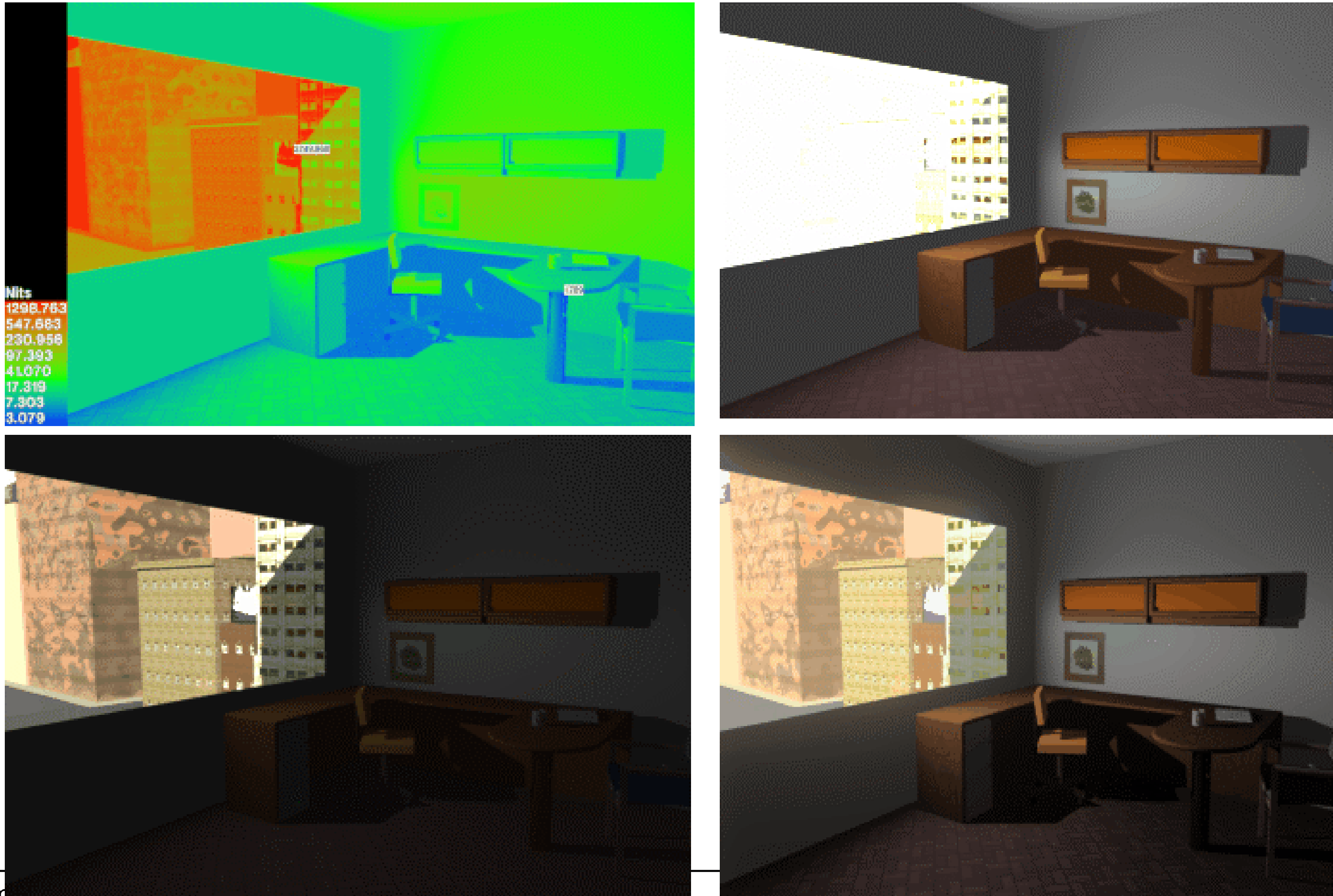
Overview

- **Last time**
  - Gamma Correction
  - Color spaces
- **Today**
  - Tone Mapping
- **Next lecture**
  - Transformations

Why Tone-Mapping?

- **Mapping radiance to pixel values?**
  - Luminance of displays
    - Up to about 300 cd/m2
  - Luminance range for human visual perception
    - Min 10^{-5} cd/m^2 Shadows under starlight
    - Max 10^{5} cd/m^2 Snow in direct sun light
- **Goal**
  - Compress the dynamic range of an input image
  - Reproduce human perception to closely match that of the real scene
    - Adaptation of the eye to environment
    - Contrast
    - Brightness
    - Other issues (glare, color perception, resolution)

Example

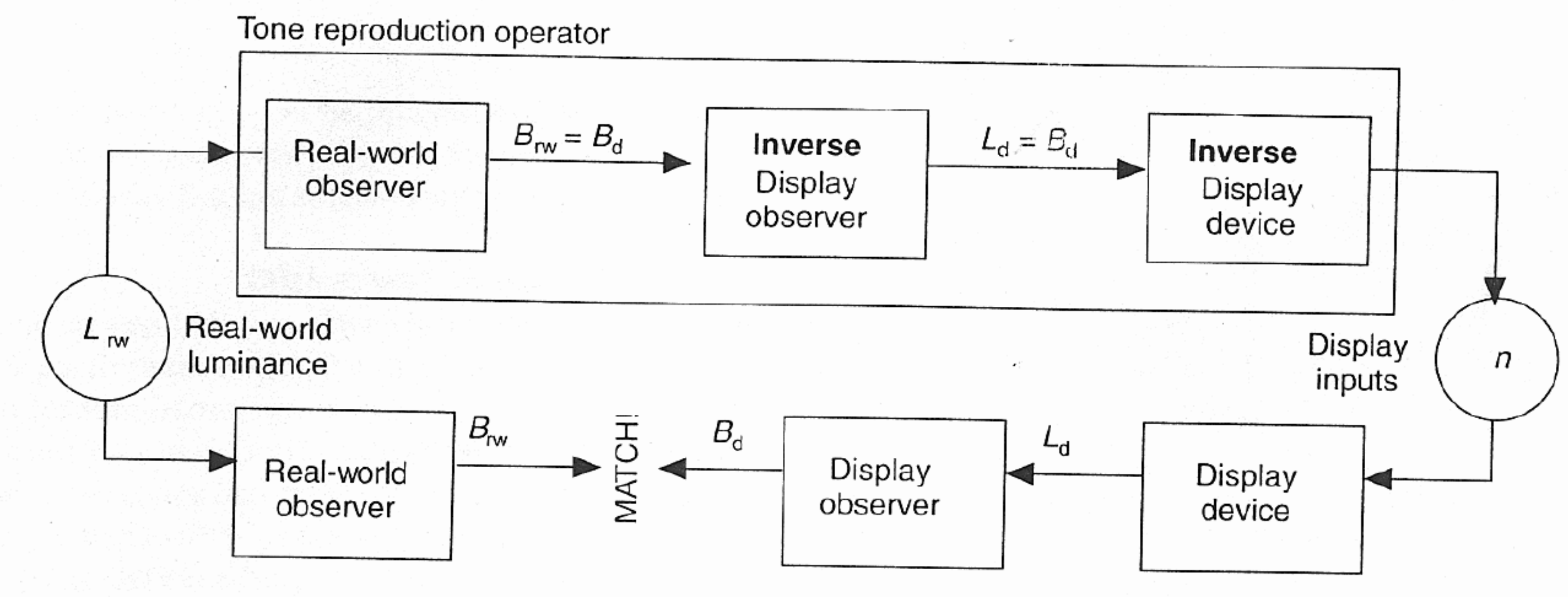


Heuristic Approaches

- **Scaling brightest value to 1 (in gray value)**
  - Problem: light sources are often several orders of magnitude brighter than the rest
    - ➔ Rest will be black
- **Scaling of brightest non-light-source value**
  - Scaling to a value slightly below 1
  - Capping light source values to 1
- **General problem of simple scaling**
  - Absolute brightness gets lost:
    - Scaling the intensity of light sources will have no effect

General Principle

- **Approach**
  - Create model of the observer
  - Requirement:
    - Observer should perceive same image from real and virtual display
  - Compute Tone-Mapping using concatenation and inversion of operators
  - Model usually operates only on luminance (no color)



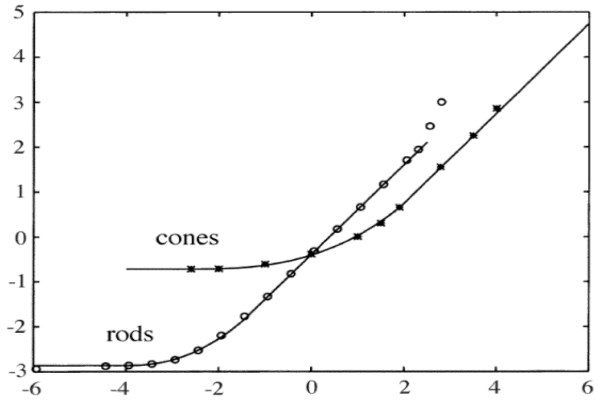
Maintaining Contrast

- **Contrast-based Scaling Factor [Ward `94]**
  - Maintain visible contrast differences in the image
    - Definition of contrast: (foreground-background)/background
  - Just noticeable contrast according to Blackwell [CIE`81] (subjective measurements)

$$\Delta L(L_a) = 0.0594(1.219 + L_a^{0.4})^{2.5}$$

  - L_a: Adaptation level of eye (luminance)

  - Goal: linear scaling factor $m(L_a)$
    - $L_d = m(L_a) L_w$
    - L_d: display luminance
    - L_w: world luminance



Maintaining Contrast

- **Approach using „Just noticeable difference (JND)“**
  - Assume JND for real and virtual image are the same
    - JND of real world: $\Delta L(L_{wa})$
    - JND of display: $\Delta L(L_{da})$

$$\Delta L(L_{da}) = m(L_{wa})\,\Delta L(L_{wa})$$

  - Substitution results in

$$m(L_{wa}) = \left[\frac{1.219 + L_{da}^{0.4}}{1.219 + L_{wa}^{0.4}}\right]^{2.5}$$

  - With $L_{da}=L_{dmax}/2$ and scaling factor sf in [0..1]

$$sf = \frac{1}{L_{d\max}}\left[\frac{1.219 + (L_{d\max}/2)^{0.4}}{1.219 + L_{wa}^{0.4}}\right]^{2.5}$$

Maintaining Contrast

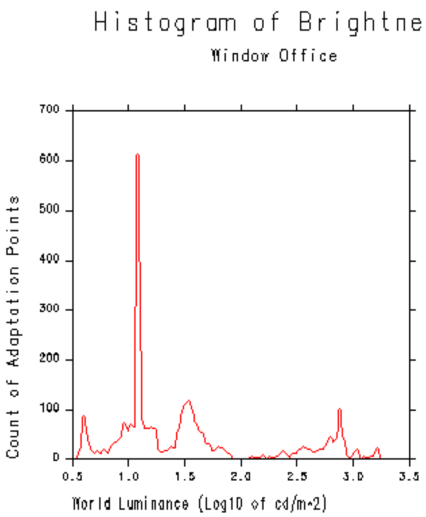
- **Deriving L_{wa}**
  - Depends on light distribution in field of view of observer
  - Simple approximation using a single value
    - Eyes try to adjust to average brightness
    - Brightness B:

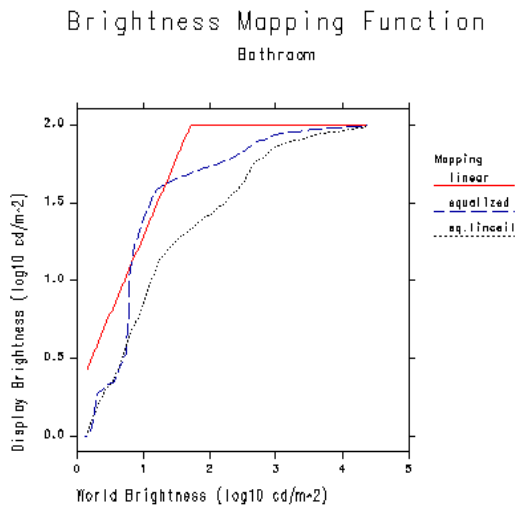
      $\log_{10}(B) = a(L_{in}) \log_{10}(L_{in}) + b(L_{in})$ Power-Law [Stevens`61]
    - Comfortable brightness

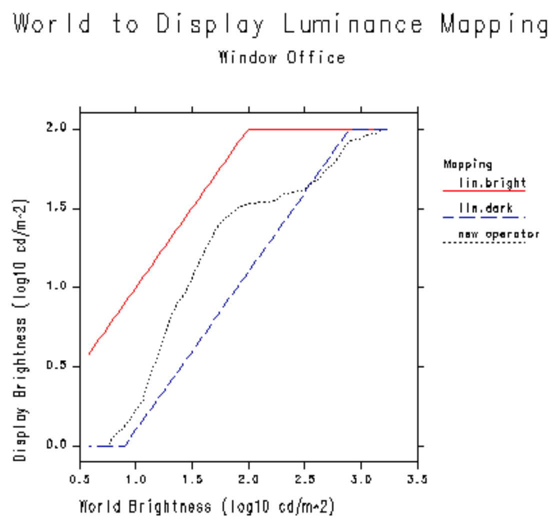
      $\log_{10}(L_{wa}) = E\{\log_{10}(L_{in})\} + 0.84$
- **Problems of this Approach**
  - Single factor for entire image
    - Different adaptation for different locations in image
    - We do not perceive absolute differences in luminance
  - Adaptation mainly acts on the 1 degree fov (fovea)
  - Results in clamping for too bright regions

Histogram-Adjustment

- **Optimal Mapping of the Dynamic Range [Ward`97]**
  – Computing an adjustment image
    - Averaging over 1 degree regions and reducing the resolution
  – Computing the histogram of the image
    - Binning of luminance values
  – Adjusting the histogram based on restrictions of human visual system
    - Limiting contrast enhancement







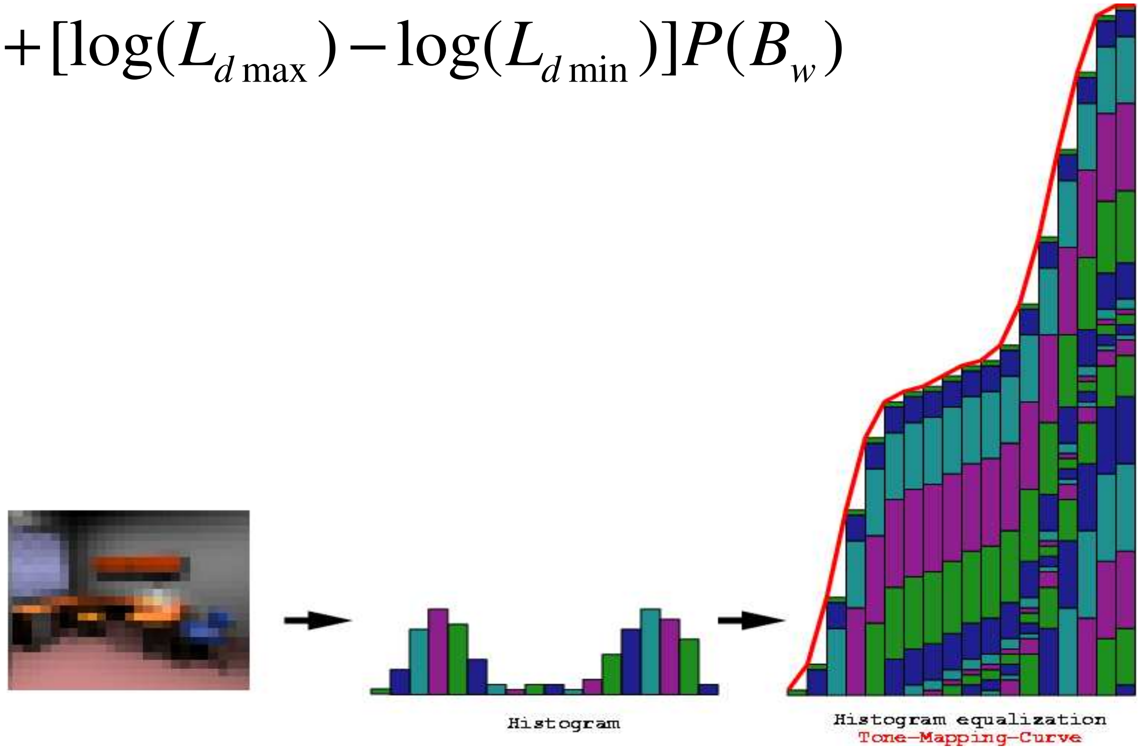
Histogram-Adjustment

- **Computing the Adjustment Image**
  - Assumes known view point
  - Average image
    - Filtering non-overlapping regions covering 1 degree fov
    - Reference uses simple box filter

Histogram-Adjustment

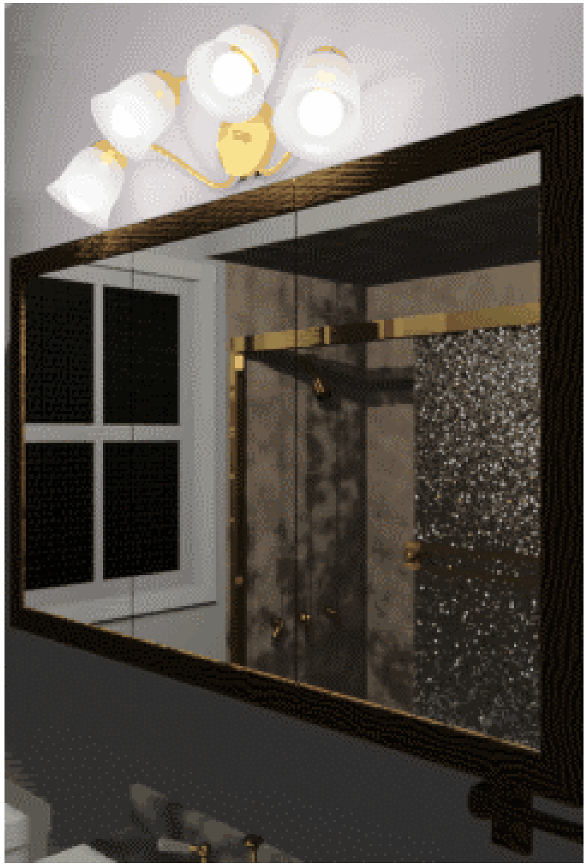
- **Naïve Histogram Adjustment (Equalization)**
  - f(Bw): Number of sample per bin
  - P(Bw): Acumulated probability (sum of sample counts)
  - T: Sum over all f(Bw)
  - Mapping

$$B_d = \log(L_{d\min}) + [\log(L_{d\max}) - \log(L_{d\min})]P(B_w)$$

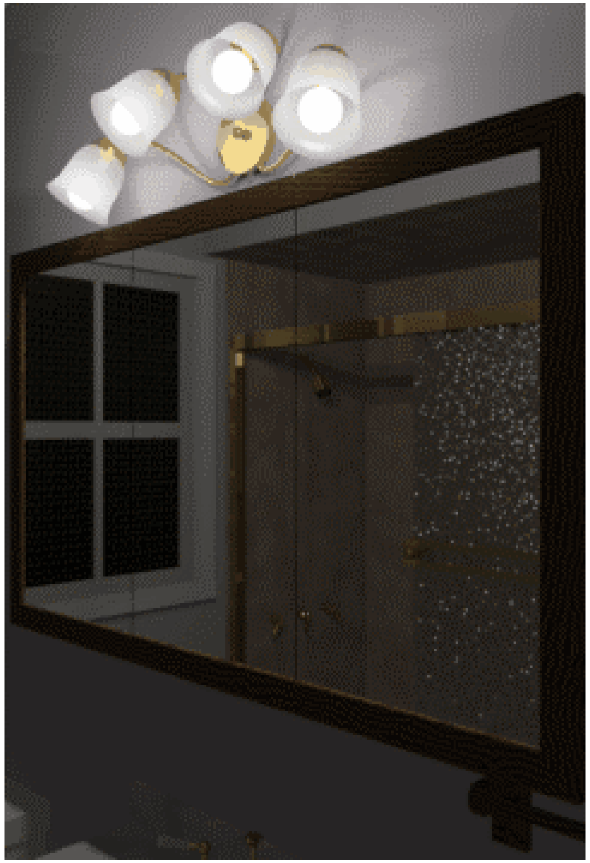


Histogram-Adjustment

Linear Mapping



Naïve Histogram-Adjustment



Histogram-Adjustment considering the human visual system

Histogram-Adjustment

- **Problem**
  - Too strong emphasis on contrast in highly populated regions of the dynamic range
  - Idea:
    - Limiting the contrast enhancement (linear scaling work well for low contrast images)

$$\frac{dL_d}{L_d} \le \frac{dL_w}{L_w} \Rightarrow \frac{dL_d}{dL_w} \le \frac{L_d}{L_w}$$

    - Differntiate $\exp(B_d) = L_d$ with respect to L_w

$$B_d = \log(L_{d\,\min}) + [\log(L_{d\,\max}) - \log(L_{d\,\min})]P(B_w)$$

leads to

$$\frac{dL_d}{dL_w} = \exp(B_d)\frac{f(B_w)}{T\Delta b}\frac{\log(L_{d\,\max}) - \log(L_{d\,\min})}{L_w} \le \frac{L_d}{L_w}$$

Histogram-Adjustment

- **Result**
  - Limiting the sample count per bin in histogram

$$f(B_w) \le \frac{T\Delta b}{\log(L_{d\,\max}) - \log(L_{d\,\min})} \qquad T = \sum f(b_i) \qquad \Delta b = \frac{\log(L_{w\max}) - \log(L_{w\min})}{N}$$

  - Implementation
    - Truncating too large bins with redistribution
    - Ditto without redistribution (gives better results)

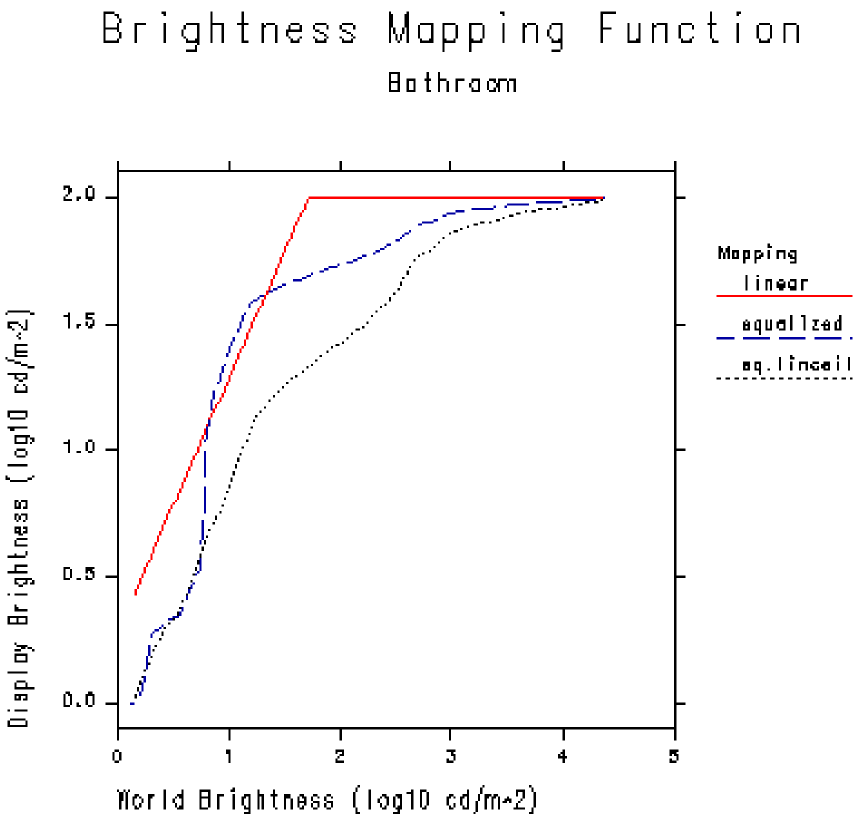
Histogram-Adjustment

- **Implementing the Limitation**

```
boolean function histogram_ceiling()
tolerance := 2.5% of histogram total
repeat {
      trimmings := 0
      compute the new histogram total T
      if T < tolerance then
            return FALSE
      foreach histogram bin i do
            compute the ceiling
            if ƒ(bi) > ceiling then {
                  trimmings += ƒ(bi) - ceiling
                  ƒ(bi) := ceiling
            }
} until trimmings <= tolerance
return TRUE
```

  – Fails for cases where no compression in necessary
    - Can easily be detected
  – Use modified $f(B_w)$ in naïve histogram equalization

Histogram-Adjustment



Histogram-Adjustment

- **Adjustment for JND**
  - Limiting the contrast to the ratio of JNDs (global scale factor)

$$\frac{dL_d}{dL_w} \le \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)}$$

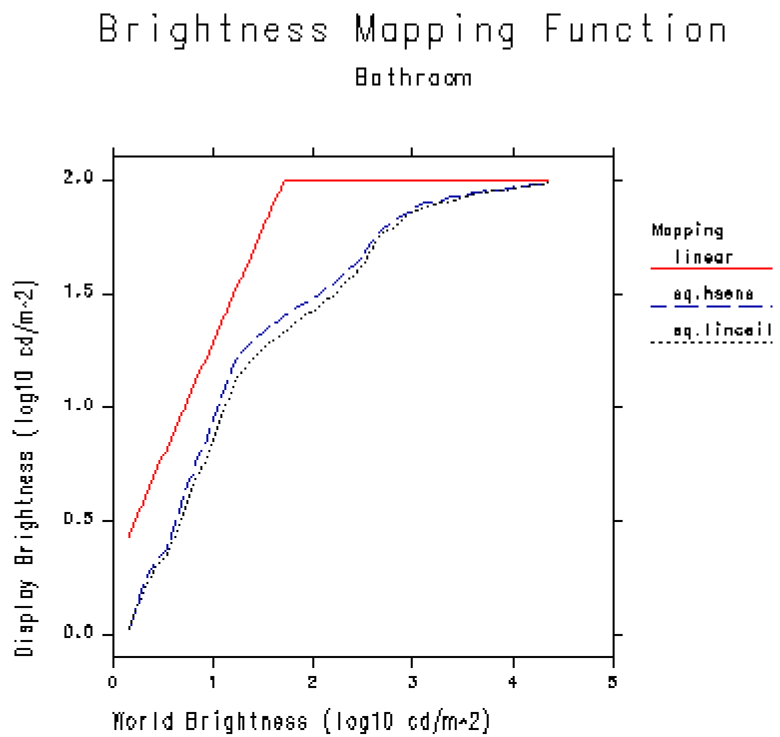
  - That results in

$$f(B_w) \le \frac{\Delta L_t(L_d)}{\Delta L_t(L_w)} \frac{T \Delta b L_w}{[\log(L_{d\max}) - \log(L_{d\min})] L_d}$$

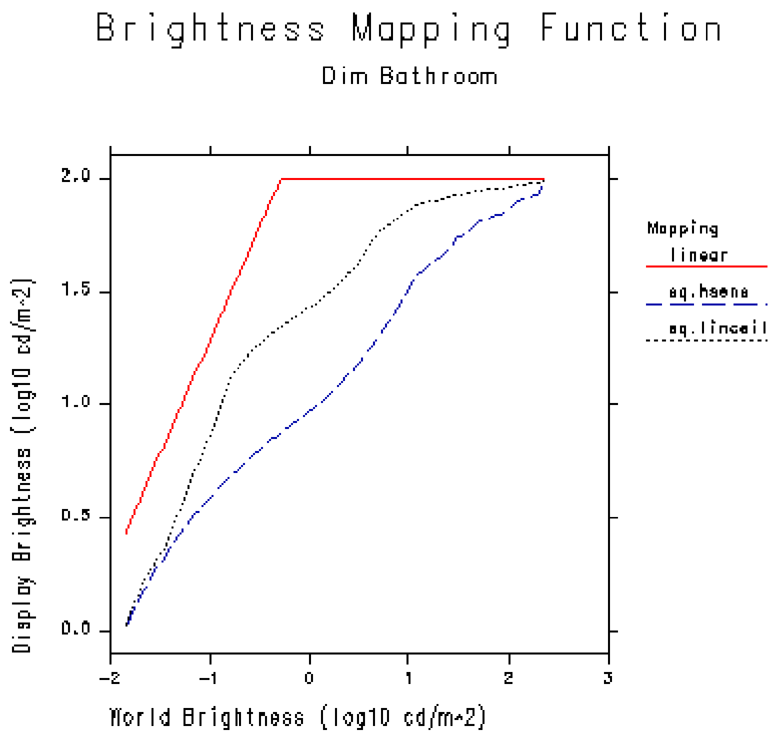
  - Implementation is the similar as for previous histogram limiting

Example: Darkening a Room

- **Reduction in Contrast Sensitivity in Dark Scenes**

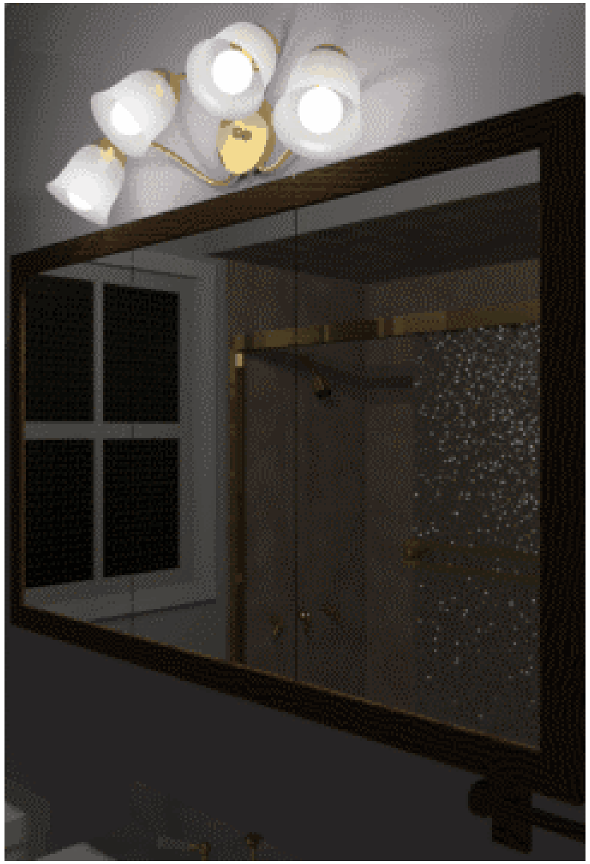


Bright Bathroom

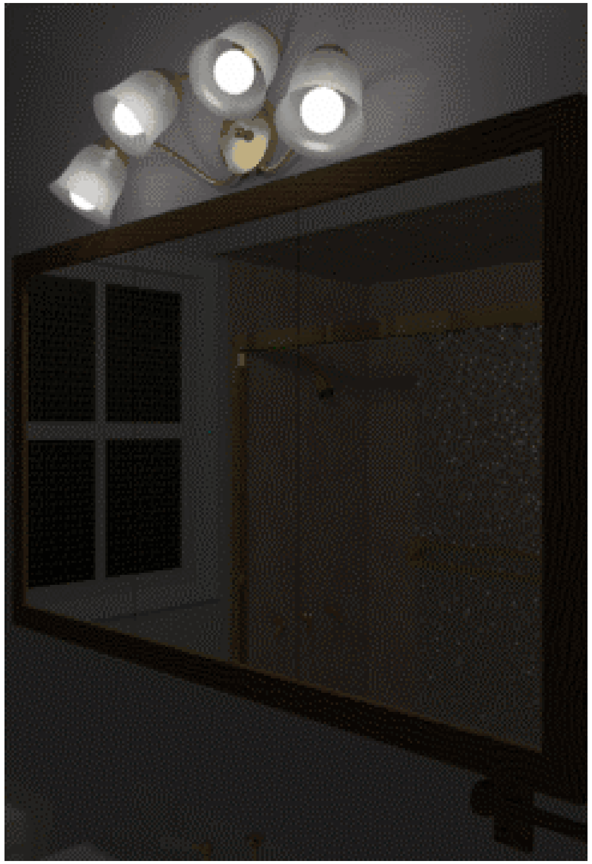


Dark Bathroom (1/100) with reduced contrast

Example: Darkening a Room



Bright Bathroom



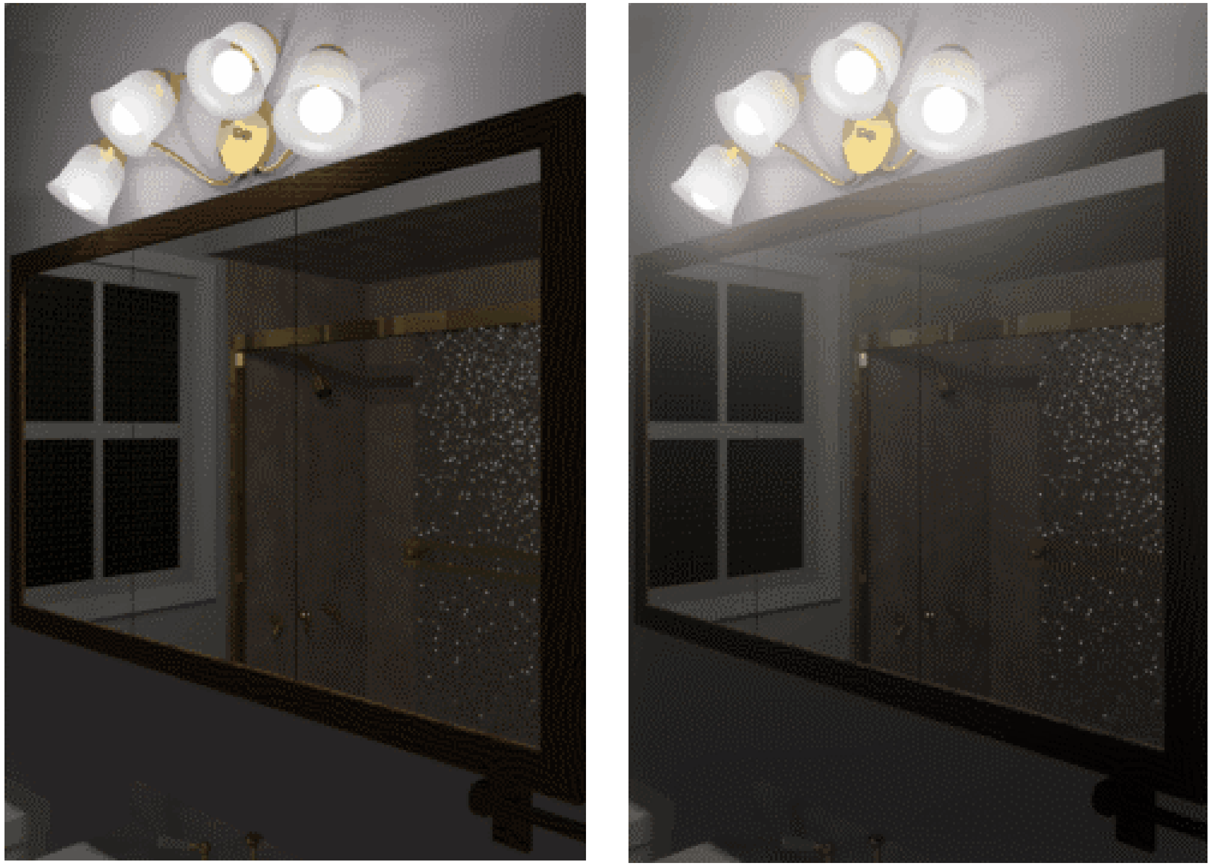
Dark Bathroom (1/100) with reduced contrast

Extensions: Glare

- **Considering Glare**
  - Bright light sources result in veiling (German: Schleier)
    - Due to scattering of bright illumination in the eye
  - Results in correction to adaptation level
- **Approach**
  - Moderate illumination in periphery does not contribute to adaptation
    - Depend exclusively on foveal region
  - But: glare in the periphery does change the adaptation
    - Scattered light is added even in foveal region
  - Compute a veiled image by filtering over peripheral region
    - Added to normal adaptation luminance L_f [Moon and Spencer,`45]

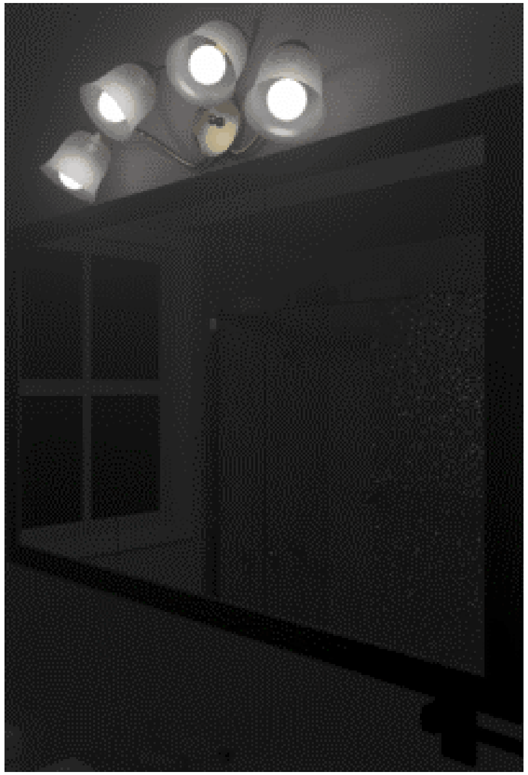
$$L_a = 0.913\, L_f + \frac{K}{\pi} \iint_{\theta > \theta_f} \frac{L(\theta, \phi)}{\theta^2} \cos(\theta) \sin(\theta)\, d\theta d\phi$$

Example: Veiling due to Glare



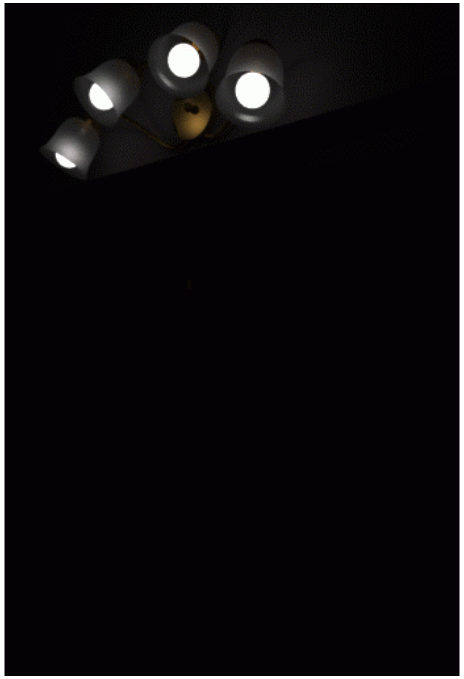
Extensions

- **Loss of color vision in dark areas**

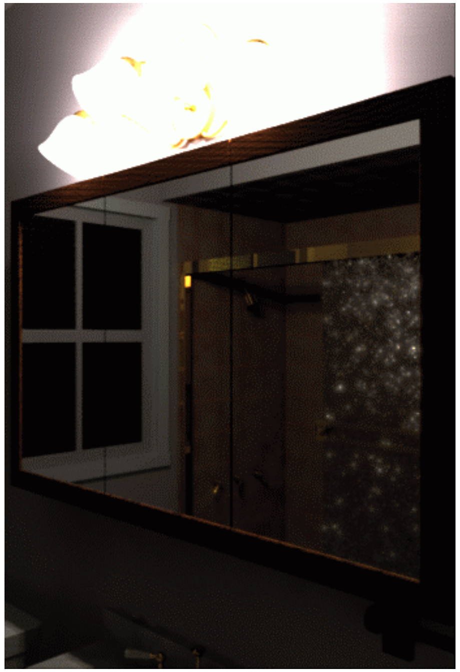


- **Loss of visual resolution in dark areas**
  - Simple blur filter

Comparison



Maximum Tone-Mapping

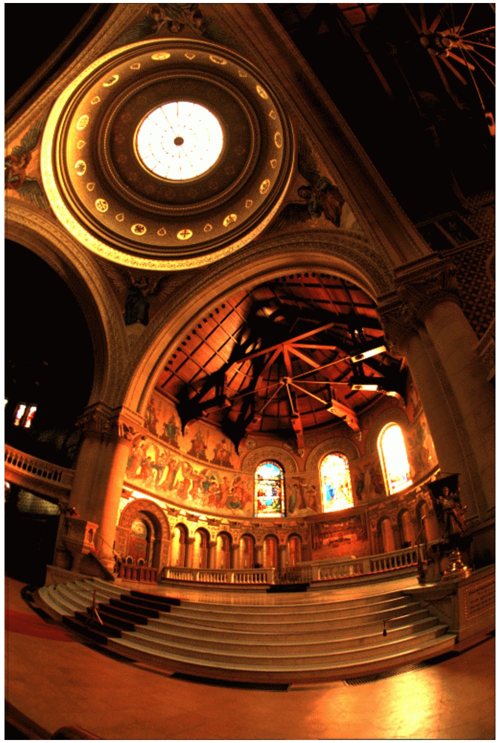


Tumblin/Rushmeier Tone-Mapping

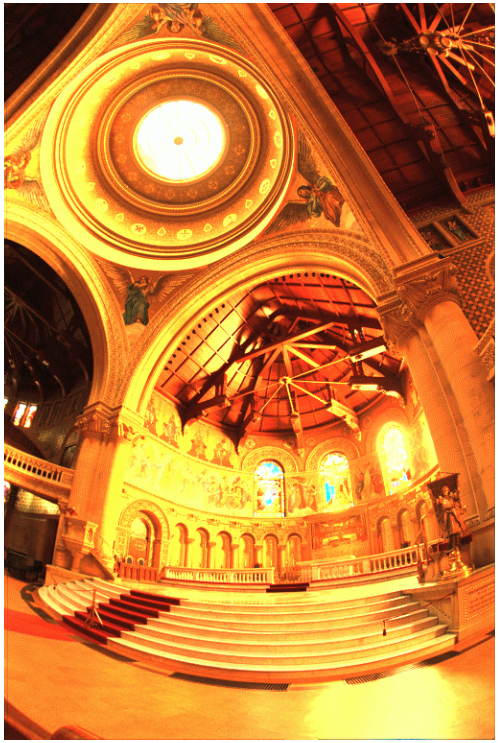
Ward`94 Tone-Mapping

Ward`97 Tone-Mapping

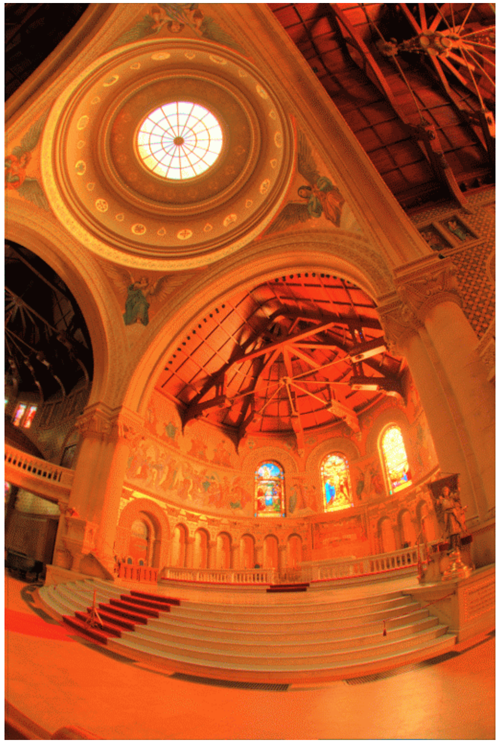
Comparison



Tumblin/Rushmeier Tone-Mapping



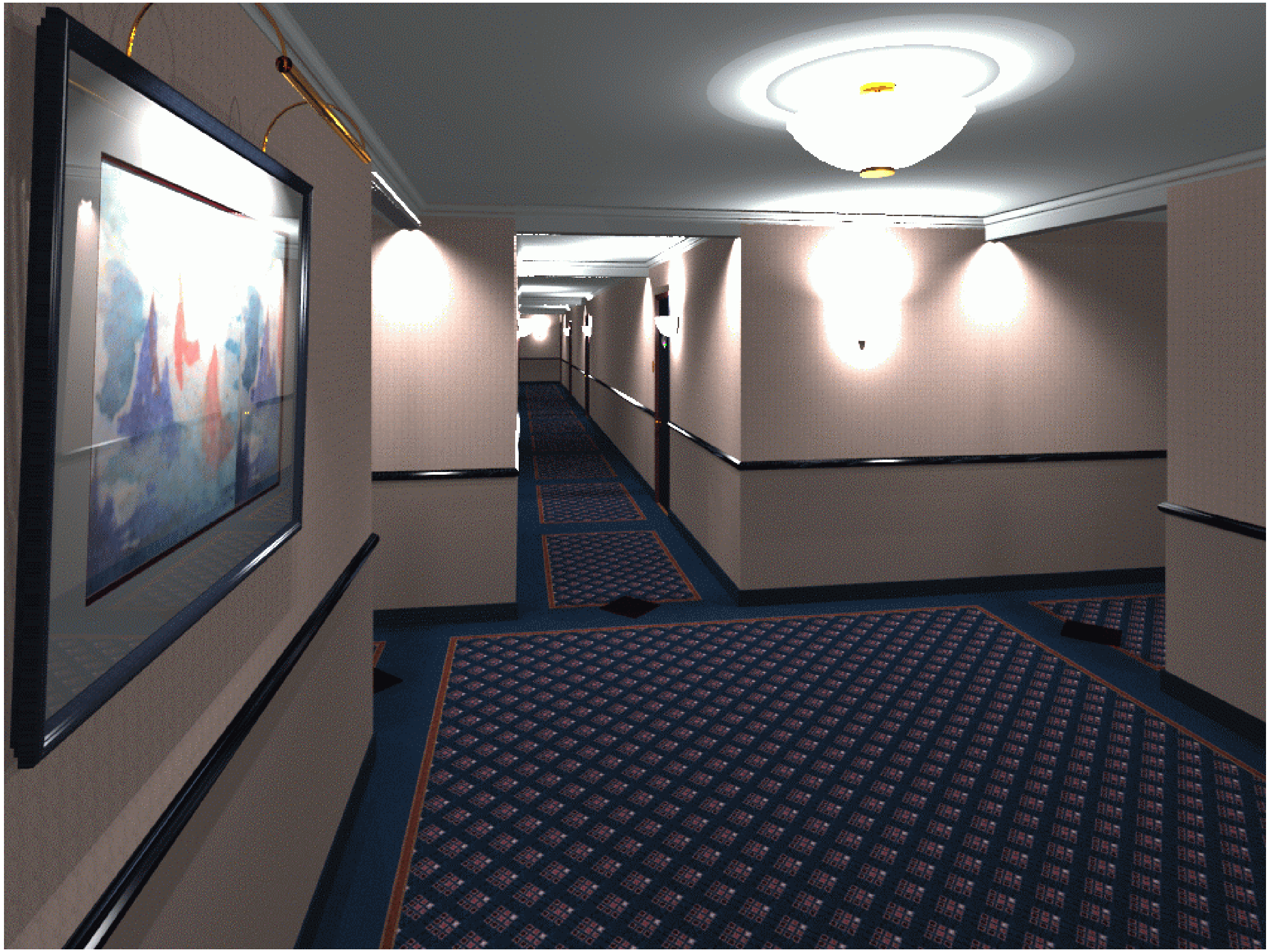
Ward`94 Tone-Mapping



Ward`97 Tone-Mapping

Comparison: Tumblin/Rushmeier

Comparison: Ward`94



Comparison: Ward`97