# VO Rendering
# SS 2009

Unit 7: Post-processing: Tone Reproduction and White Balance

---

## Overview

- The Problem
  - Image Synthesis Pipeline
  - Different Image Types
  - Human visual system
  - Tone mapping
  - Chromatic Adaptation
- Tone Reproduction
  - Linear methods
  - Nonlinear and perceptual methods



---

## Image Synthesis Pipeline

- Modeling
- Rendering
  - Output can be RGB, XYZ, spectral images
  - Predictive rendering yields high dynamic range images
- **Display**
  - Typical devices have limited range for both luminance and colour



---

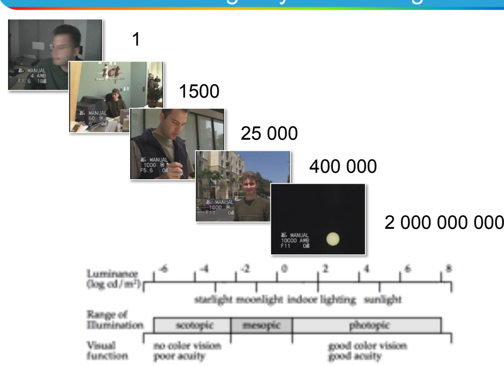
## What is HDR?

- The “dynamic range” of a scene is the contrast ration between its brightest and darkest parts



---

## Real World has High Dynamic Range






---

## In Theory...

- Captures of reality (or realistic rendering) leads to high dynamic range images
- These cannot be displayed directly on normal display hardware
- Special image formats are necessary
- A display representation which yields the same visual sensation as viewing of the real scene is also needed

## In Theory ...



## In Practice...



## The Problem

- It is usually impossible to solve the reproduction task perfectly
- It strongly depends on the output device
- Various heuristics of increasing complexity exist
- Full perception models difficult
- Animations pose additional challenges (frame to frame coherency)



## Image Types

- Relative Values
  - Measured in terms of some maximal output device capability
    - Computer screens: two orders of magnitude
    - Printouts: a range of roughly 10 luminance units
    - 8 bit images: 256 steps (!)
- Absolute Radiometric Values
  - Captures of reality - “scene reference images”
  - This is what digital cameras ought to capture!



## Storage: Image Formats

- Capture / output of rendering has to be stored for later processing
  - Conventional formats
  - High-dynamic range colour space formats
  - Spectral formats (possibly including polarization information)
- Any format except the last category destroys information gathered during rendering / image capture!



## Conventional Image Formats

- Usually RGB (TIFF, PNG, JPEG, …)
- TIFF: also CIE L*a*b*
- Normally: 8 bits per channel
- TIFF: 16 bit possible (JPEG 12 bit)
- „Brightness ends at 1“ → device dependent
- No physical meaning of values
- **Advantage**: compact size, standardised
- **Disadvantage**: large amounts of information are destroyed

## High Dynamic Range Formats

- Values have physical meaning
- Floating point components → large range
- **Advantage**: compact size, standardised, few quantization errors
- **Disadvantage**: compression can introduce artefacts
- (not understood by Photoshop et. al.)
- Radiance RGBE, Pixar Log and LogLuv TIFF, ART XYZ (uncompressed), **OpenEXR**



## OpenEXR HDR Image Format

- Developed by ILM for production use (2002)
  - Harry Potter, MIB 2, ...
- High dynamic range image format
  - Tailored to the needs of the movie industry
- Open source, freely available
  - www.openexr.net
  - BSD style license



## OpenEXR Motivation

- Formats with 8 bits per channel are fundamentally unsuitable for movie work
- 16 bit per channel formats have limitations with respect to post-processing
- 32-bit LogLuv TIFF is overkill for production use
  - Sufficient precision, but large size
  - With 3k x 2k pixel images size does matter
- Additional features (annotations) desirable



## OpenEXR Features 1

- 16 and 32 bit floating point colours
- 16 bits: $10^9$ range, 30 f-stops
  - 8-bit images: ~7-10 stops
- ~1000 colour steps per f-stop
  - 8-bit: ~70
  - No loss in accuracy even through repeated processing
- Lossless compression
  - ~ 35% - 55% for grainy images

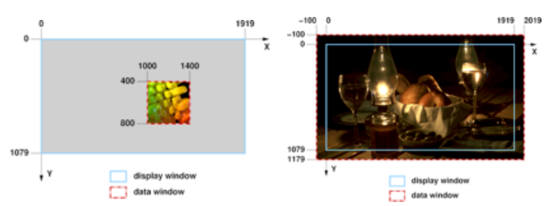


## OpenEXR Features 2

- 16 bit floating point data compatible with Nvidia CG data type HALF
  - EXR images can be directly used in hardware
- Arbitrary information can be stored alongside image data
  - Camera settings, colour timing information
- Arbitrary image channels
  - R, G, B, Alpha, Y, U, V, ...



## OpenEXR Display vs. Data




- Data which extends beyond display is needed for post-processing (wide filter kernels)
- Subimages can be useful for compositing

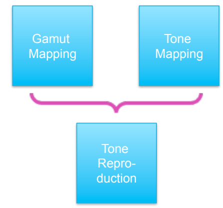
## Spectral Image Formats

- N spectral samples per pixel
- Floating point components ⇨ large range
- No widely standardized formats – FITS and ARTRAW are lone examples
- Values have physical meaning
- **Advantage**: no quantization errors, no compression errors, no information lost from rendering pass
- **Disadvantage**: huge filesizes (up to ~400MB for 640x480!), rarely any support outside originating package



## Image Post-Processing

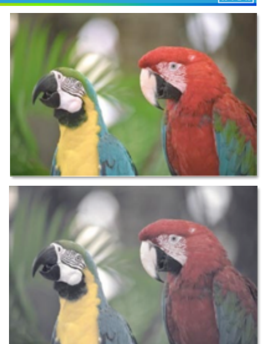
- Two Tasks
- **Gamut mapping:**
  - Getting all colours into the display gamut
- **Tone mapping:**
  - Fitting the luminance range to a given device





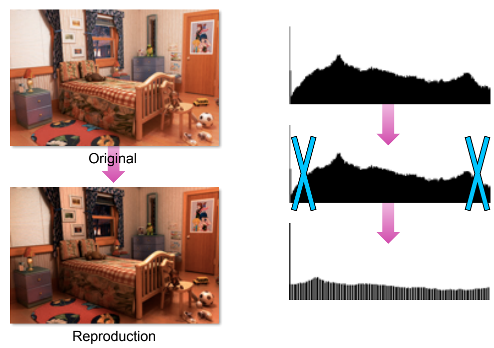

## Gamut Mapping: Approaches

- Move colour values into the displayable area
- **Local:** Outlying points are individually moved
  - Fast
  - Highlights may be lost
- **Global**: all points are analysed, and the point cloud is shrunk so that it fits into the gamut
  - Relation between colours is maintained
  - De-saturation of image





## Tone-Mapping Task

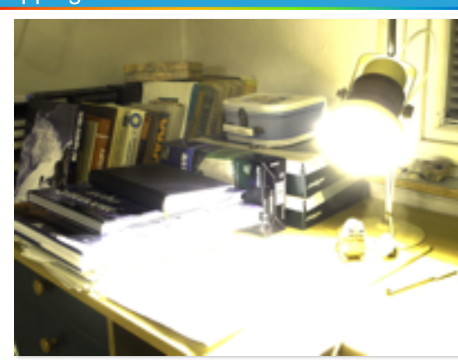





## Test Image Sequence



## Clipping on 1



## Tone Reproduction Operators

- Three different approaches
- Global Methods
  - Spatially uniform
  - Linear scale factor
  - Non-linear scale factor
- Local Methods
  - Spatially non-uniform
- Perceptual approaches



## Global Methods

- Scaling of all luminance values by a given factor
- Primitive & fast
- Automatic determination of the factor
- Sufficient for many scenes
- Will result in very dark images if the HDR image has a wide dynamic range
- Linear and non-linear operators



## Linear Solutions

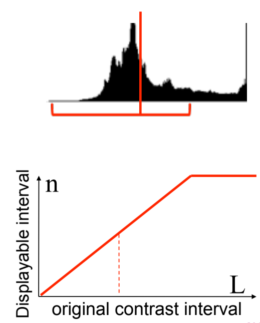
- All luminance values are scaled by the same linear factor
  - Mean value mapping
  - Interactive calibration
  - Ward´s contrast based scaling factor
  - …
- $L_d$ = device intensity, $L_w$ = world intensity



## Mean Value Mapping

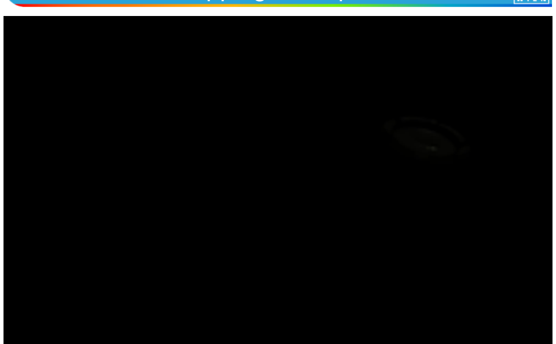
- Mean value of the histogram is mapped to 0.5
- Values outside the contrast interval are clipped (truncated)

$$L_d = 0.5 * L_w / L_{davg}$$





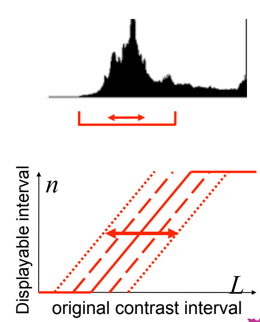

## Mean Value Mapping Example





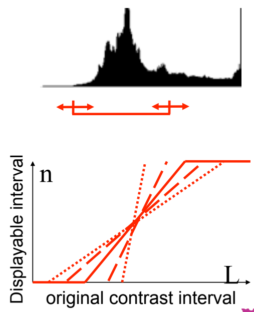
## Interactive Calibration

- Interactively define the ***area*** of the available contrast interval
- Interactively define the ***range*** of the available contrast interval



## Interactive Calibration

- Interactively define the ***area*** of the available contrast interval
- Interactively define the ***range*** of the available contrast interval






## Interactive Calibration Example



## Ward´s Contrast Based Scaling

$$L_d = \frac{1}{L_{dmax}}\left[\frac{1.219 + (\frac{L_{dmax}}{2})^{0.4}}{1.219 + L_{wa}^{0.4}}\right]^{2.5} L_w$$

$L_{dmax}$ ... maximum display luminance
$L_{wa}$ ... environment adaptation degree

- Good results: just visible differences remain
- But image has to be given in absolute units



## Ward Example



## Nonlinear Methods

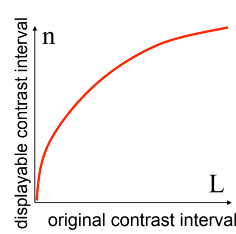
- Scaling factor nonlinear
  - Exponential Mapping
  - Schlick's Method
  - Mapping by Tumblin and Rushmeier
  - Visual Adaptation Model (Ferwerda et al)
- ...



## Exponential Mapping

- Exponential function corresponds to human perception
- Reduces the overproportional influence of a few very bright pixels

$$L_d = 1 - e^{\frac{-L_w}{L_{wavg}}}$$



## Exponential Mapping Example



## Schlick's Method

$$L_d = \frac{p \cdot L_w}{(p-1) \cdot L_w + L_{w\,\max}}$$

$$p = \frac{M \cdot L_{w\,\max}}{N \cdot L_{w\,\min}}$$

M ... darkest grey

N ... # of available color steps

- Exhibits behaviour similar to exponential mapping
- Well suited for images with high contrast
- Can fail completely!



## Schlick Example



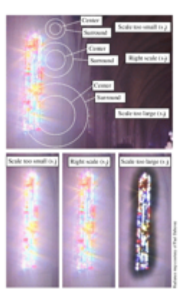
## Local Methods

- Differences between various parts of the image are taken into account
- Similar to techniques from photography, the image is separated into zones to determine brightness targets
- A local kernel of variable size is used for the final tone reproduction step
- Can look artificial



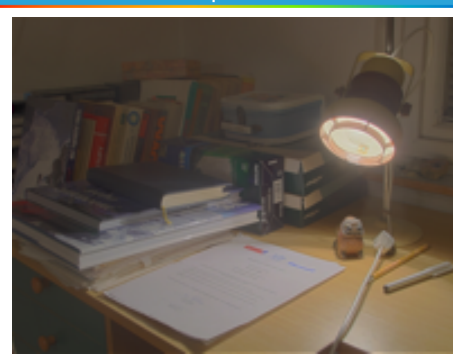
## Variable Parameters

- Size of kernel determines final image sharpness and blurring of bright areas
- Light is automatically selectively withheld („dodging and burning“) in areas with large adjacent illumination





## Local Method Examples #1



## Local Method Examples #2

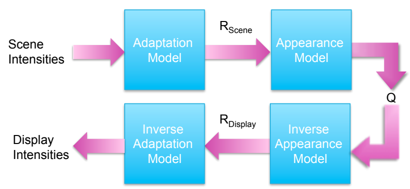


## Perceptual Methods

- Results from physiology and psychology are used in order to reproduce the behaviour of the human visual system
- Two-pronged approach:
  - one has to determine what a person would see if the scene were real
  - And then try to reproduce this sensation using a display device



## Perceptually Based Models




- The top row computes the viewed scene appearance
- The lower row attempts to reproduce this perception on the display device



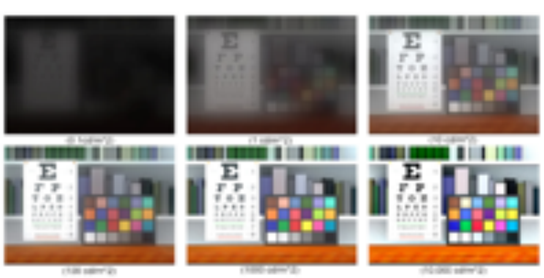
## Model of Visual Adaptation

- Pioneering work by Ferwerda et al. [1996]
- Based on physiological model
- Takes into account
  - Threshold sensitivity
  - Color appearance
  - Visual acuity
  - Light adaptation
  - Dark adaptation


threshold sensitivity

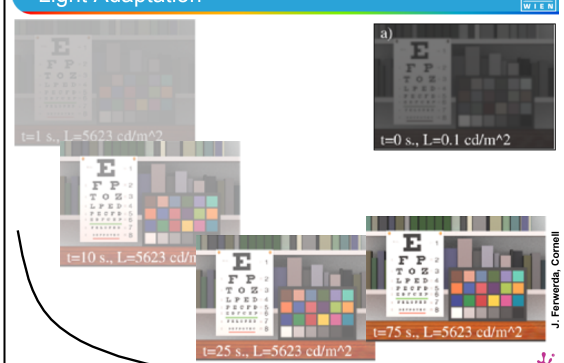


## Visual Acuity

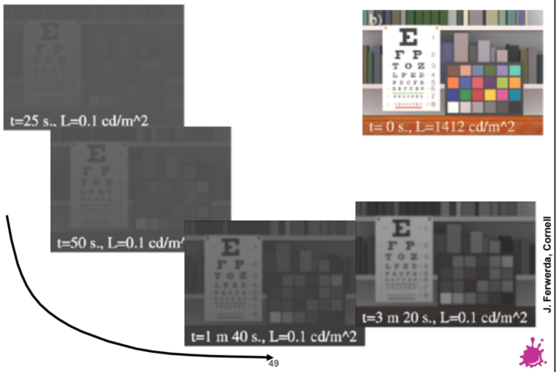




## Light Adaptation



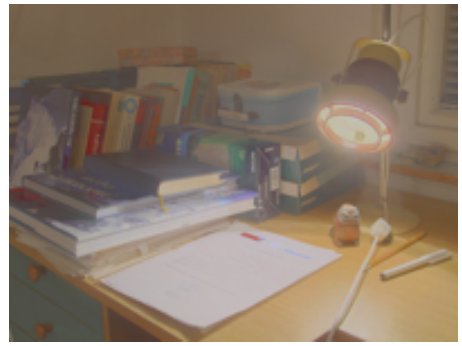
## Dark Adaptation




J. Ferwerda, Cornell



## Ferwerda Example





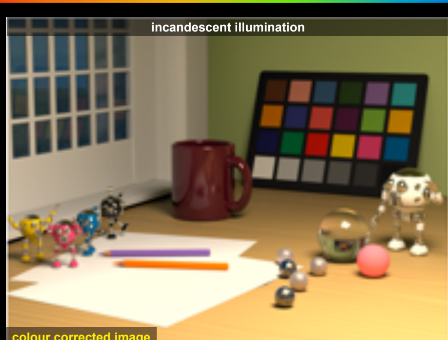
## Colour Constancy



## What is Chromatic Adaptation?



## Chromatic Adaptation: Example




## Colour Correction

- Alternative name: **white balance**
- Attempts to replicate the illuminant hue compensation of the human visual system
- Necessary to evoke identical viewer response for captured or synthetic scenes, and real images
- Problem: different viewing surrounds
  - Real scene: immersion
  - Captured scene (image): displayed on monitor

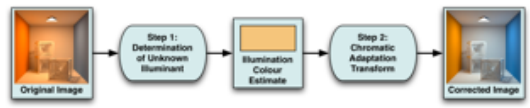
## CC Algorithms

- Still challenging task
- Many algorithms exist
  - Gray world
  - White patch
  - Neural networks
  - ...
- All algorithms are image based
- Only two exist that are scene driven



## Colour Correction Workflow

- Colour Correction is a two-step process:
  - Determining the illuminant colour
  - Applying a transform that compensates for the illuminant
- Step 1 is the tricky one if you only have image data at your disposal




## Gray World and White Patch

- Gray World
  - Assumption: Average of all pixels is gray
  - Average is mapped to gray
  - Fails if assumption is violated
- White Patch
  - Assumption: There is always a white object in the image (e.g. highlight)
  - Brightest pixel is mapped to white
  - Fails if no white object is in scene
- Better: Scene-based

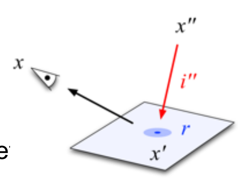


## CC State of the Art

- Practically all techniques only use image data
- Large research area in computer vision
- Very sophisticated methods available, but none are entirely robust
- **Idea:** A reliable CC method that uses additional information about an image that can be gathered during rendering
  - (Almost) free, simple, **robust**
  - End result just as good as with image-based methods

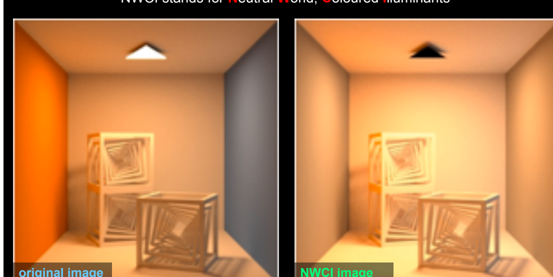
## Algorithm Overview

- Two additional images are computed during rendering
  - All *directly viewed* surfaces set to neutral
  - All lights set to neutral *on directly viewed surfaces*
- Cheap to compute as by-product of rendering
- Sub-sampling possible
- Images processed to ge illumination estimate




## NWCI Image

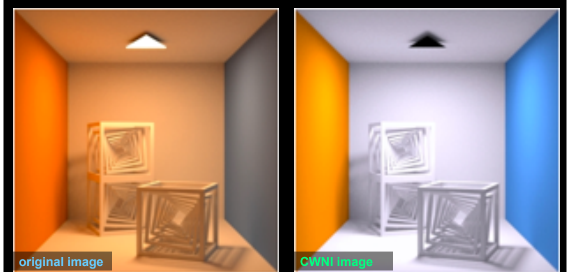
NWCI stands for Neutral World, Coloured Illuminants




All surfaces are set to neutral, while the illuminants retain their colours.

# CWNI Image

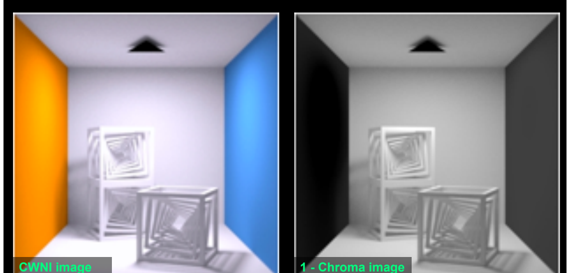
NWCI stands for Coloured World, Neutral Illuminants




All surfaces retain their colours, while the illuminants are set to neutral.

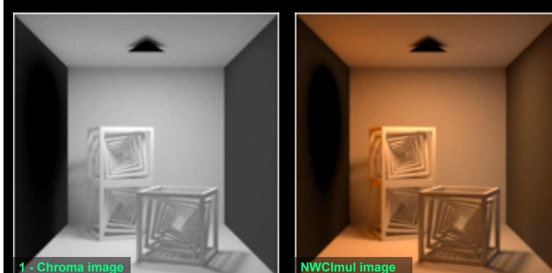
# RCA Algorithm

# 1 - Chroma Image



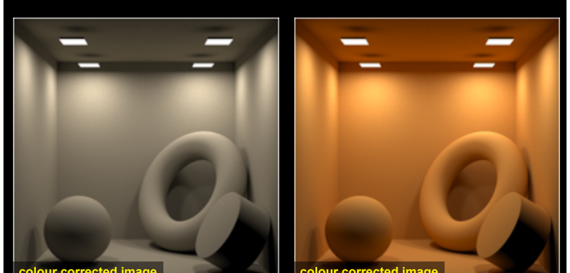

**1 - Chroma:** bright in those areas that have neutral surface colour, dark in those with high chroma

# NWCImul Image



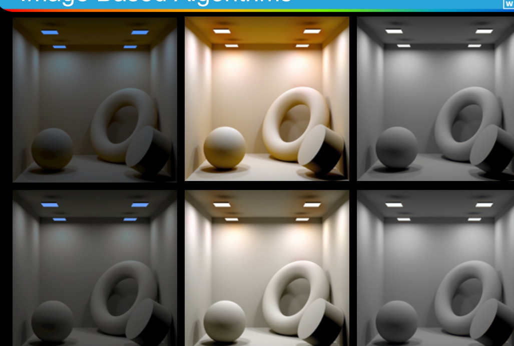

**NWCImul:** NWCI multiplied with the 1 - Chroma image

# White vs. Orange World



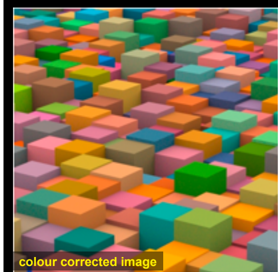

white box, orange light — orange box, white light

# Image Based Algorithms

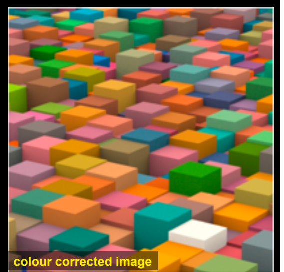


Gray World — Retinex — Local Shift

## 3D Mondrian, Yellow Illuminant

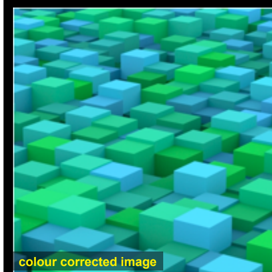



no white object present

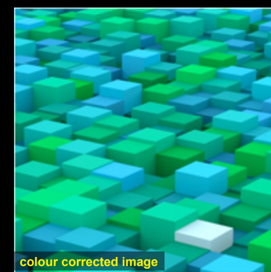



one white block in scene

## Green-Blue 3D Mondrian, Blue Light

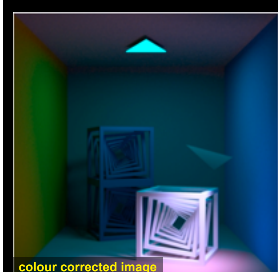


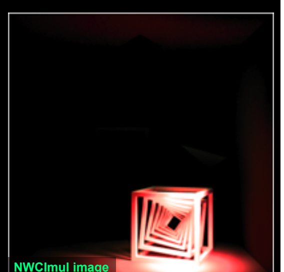

no white object present



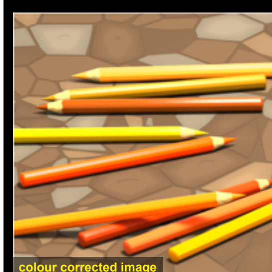

one white block in scene

## Adaptation for Reference White Objects

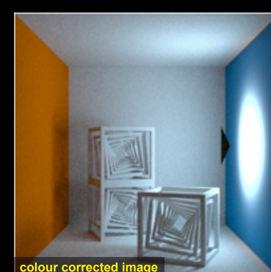



## Additional Examples




strong orange object colour cast, orange light




strong indirect illumination colour cast

## Realistic Scene, Various Illuminants


## Animation

## Interactive Tone Reproduction

- Interactive applications
  - Walkthroughs
  - Games
  - Flight/driving simulators, ...
- Additional effects
  - Dazzling
  - Slow dark adaptation
  - Other subtle effects of visual adaptation



## Rendering VO Unit 7

**The End**

*Thank you for your attention!*