Images

Display device

- Rendered images are shown on **display device**
  - e.g., monitor, television screen, etc.
- Display devices are adapted for showing **raster images** – arrays of pixels
- **Display devices are limited** in:
  - Resolution (number of pixels)
  - Brightness (intensity)
  - Contrast
  - Color (gamut)
- Rendered images may contain values which can not be directly shown on display devices.

Frames per second

- Number of rendered images per second:
  - Real-time graphics
  - Interactive graphics
  - Offline graphics

Image buffer

- Rendering algorithm uses virtual camera which defines film information
- Film is in fact raster image → it contains array of pixels also called image buffer.
- Rendering of each pixel produces a raster image

Reconstructing images

- Final image is reconstructed from samples per pixel
  - TODO
  - **RTR 5.4.**

Storing images

- Storing the rendered images

Two main types of images

- Raster images
- Vector images

Vector images

- Shapes in vector image are analytical defined
- To display an image, rendering is performed to evaluate analytically defines shapes on raster display

Raster images

- Raster images are made of array of pixels.
- Each pixel contains one color
- Discrete representation

Raster image file formats

- Popular raster image file formats are JPG, PNG, TIFF, etc.

Raster image display

- Rendering results in raster image where each pixel has certain color
- Due to display devices, using “original” pixel values is sometimes not desired due to limitations
- Remapping of colorspace is often done to achieve correct display
- Gamut plays an important role
- TODO

Display encoding

- All input values (e.g., texture image) for rendering and all values during rendering must be in **linear** colorspace.
  - Linear values are needed for correct addition and multiplication operations
- On the other hand, display devices have **nonlinear** (power law) relationship between input voltage and display radiance
  - Early display devices that were based on cathode-ray tube (CRT). As energy level applied to pixel is increased, the radiance emitted doesn't grow linearly but according to power law. For example, pixel set to 50% will emit 0.5^2 amount of light.
  - LCDs and other display devices mimic the CRT response, although have different intrinsic tone response curves.
  - Relationship between digital values in image buffer and radiance levels emitted from the display is described with electrical-optical transfer function (EOTF)*
- Problem 1: image which is rendered is in linear colorspace and will not display correctly if shown directly
  - Linear values will appear to dim on the screen <image>
- Problem 2: when working in modeling tools where we pick texture images or colors on screen, those colors are encoded for display device so we can see them properly
  - They can not be used in rendering computation directly because they are in non-linear space: shading computations are correct for linear values <image>
- When encoding linear color values for display, the goal is to cancel out the effect of display transfer function: inverse of display transfer function is applied to color values – **gamma correction**
  - Standard transfer function for personal computer display is called **sRGB**
  - **<image: texture → decoding → shading → encoding → display (decoding)>**

* also optical-electrical transfer function describe the other end of the process for image and video capture devices

Gamma correction

- Gamma correction can be seen as last step in rendering and post-processing – when everything is computed and image is ready for display.
  - To encode rendered image which is in linear colorspace (x) for sRGB display (y) the following formula is applied to all color values:
    - $y = x^{1 / gamma}$, where gamma = 2.2
- Everything we see on the screen (e.g., image textures or color-pickers in modeling tools) is display-encoded data and those values must be decoded to linear values for rendering.
  - To decode sRGB display encoded values (y) to linear colorspace (x) the following formula is applied to all color values:
    - $x = y^{gamma}$, where gamma = 2.2

High dynamic range display encoding

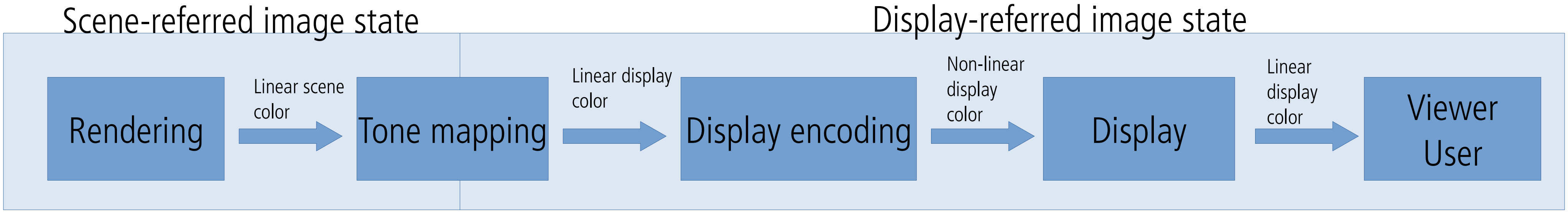
- By now we have discussed standard dynamic range (SDR):
  - SDR Monitors use sRGB display standard
  - SDR Televisions use Rec. 709 and Rec. 1886 display standards
- Both SDR monitors and televisions have same RGB **gamut** and **white point** (D65), also similar non-linear encoding curves and luminance levels
- HDR displays use Rec. 2020 and Rec. 2100 standards
  - Rec. 2020 has significantly wider color gamut and same white point (D65) as Rec. 709 and sRGB color spaces
  - Rec. 2100 defines two non-linear display encodings: perceptual quantizer (PQ) and hybrid-log gamma (HLG)
- <image comparing gamuts>
- Therefore, different methods must be used for transferring rendered images in linear colorspace to HDR display:
  - HDR10
  - scRGB
  - Dolby Vision

Tone mapping

- Reminder: **Rendering** is process of calculating image color based on 3D scene description.
  - The final results is – pixel values in display buffer – still needs to be determined.
- Reminder: **Display encoding** is process in which linear color (radiance) values are converted to nonlinear values for display hardware.
  - Inverse of display EOTF is applied so that linear values in image buffer match the linear radiance emitted by the display
- Between rendering and display encoding there is **tone mapping step.**
- But let's see the big picture of **imaging pipeline** first.

Imaging pipeline

- Tone mapping is step between **image states***:
  - **Scene-referred**
    - Defined in reference to scene color (radiance) values.
    - Physically-based rendering computations are correct for this state
  - **Display-referred**
    - Defined in reference to display color (radiance) values.



* Display limitations require non-linear transform between two states

Tone mapping

- Tone mapping (end-to-end or scene-to-screen transfer) is about converting scene color values to display radiance values
  - Goal 1: **image reproduction** - create display-referred image that reproduces, as closely as possible given display and viewing properties, perceptual impression that viewer would have if they were observing original scene.
  - Goal 2: **preferred image reproduction** – create display-referred image that looks better (to some criteria) than original scene.

Tone mapping: image reproduction

- Reproduce a similar perceptual impression as the original scene, problems:
  - Original scene **luminance** exceeds display capabilities
  - **Saturation** (purity) of original scene also exceeds display capabilities
- **Exposure** is another term critical to image reproduction
  - Exposure in photography refers to controlling the amount of light falling on film/sensor.
  - In rendering, exposure is a linear scaling operation performed on scene-referred image before tone mapping is applied.

Tone mapping: image reproduction

- <image: image with and without image reproduction tone mapping>

Tone mapping: image reproduction

- **Global tone mapping:**
  - Scaling by exposure
  - Then applying tone repoducton
- Local tone mapping:
  - Different mapping pixel-to-pixel based on surrounding pixels and other factors

Global tone mapping

- Tone reporduction
  - TODO - RTR 8.2
- Exposure
  - TODO - RTR 8.2

Tone mapping: preferred image reproduction

- Creative manipulation of image colors to obtain image desired artistic “look” - is called **color grading**.
- **Grading look up tables** (LUT) contain desired color transformations which are applied on image
  - e.g., baked color curves
- Color grading is possible on:
  - Scene-referred images: produce hi-fi results
  - Display-referred images: easier to set-up

Color grading

- <example of color graded images>

More into topic

- https://developer.nvidia.com/sites/default/files/akamai/gameworks/hdr/UHDColorForGames.pdf
- Color grading: http://filmicworlds.com/blog/minimal-color-grading-tools/
- https://cinematiccolor.org/
-

Texture image compression

- RTR 6.2.6