# 02562 Rendering - Introduction

## High Dynamic Range Imaging

Jeppe Revall Frisvad

November 2023

# Dynamic range

- Ambient luminance levels for some common lighting environments:

| **Condition** | **Illumination ($cd/m^2$)** |
|---|---|
| Starlight | $10^{-3}$ |
| Moonlight | $10^{-1}$ |
| Indoor lighting | $10^{2}$ |
| Sunlight | $10^{5}$ |
| Maximum intensity of common monitors | $10^{2}$ |


Reference

- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting, second edition, Morgan Kaufmann/Elsevier, 2010.

# High dynamic range imaging

- Why doesn't the camera see what I see?
  - The camera has a much smaller dynamic range (several orders of magnitude measured in $cd/m^2$).
  - The part of the visible dynamic range captured by the camera is determined by the size of the aperture and the exposure time.
- Exposure is usually changed in stops.
  - A stop is a power-of-two exposure step (halving the exposure time while keeping the aperture constant will decrease the exposure by 1 stop, for example).
- High dynamic range imaging:
  - Keep the camera still and take images at multiple exposures.
  - Combine several low dynamic range images into one high dynamic range image (HDR image capture).
  - Map the high dynamic range image to a low dynamic range display (tone reproduction).
- HDRI was once Hollywood's best kept secret [Bloch 2007].

References


- Bloch, C. *The HDRI Handbook: High Dynamic Range Imaging for Photographers and CG Artists*. Rocky Nook, 2007.
- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, second edition, Morgan Kaufmann/Elsevier, 2010.

# HDR image capture

- Exposure time from 30 s to 1 ms in 1-stop steps.

- Combining to get high dynamic range:

$$L_{ij} = \sum_{k=1}^{N} \frac{f^{-1}(Z_{ijk})w(Z_{ijk})}{\Delta t_k} \Bigg/ \sum_{k=1}^{N} w(Z_{ijk}) \ ,$$

where $Z_{ijk}$ is response-weighted radiant exposure of pixel $ij$ in capture $k$ of exposure time $\Delta t_k$, $w$ is a weighting function (to deal with overexposure), and $f$ is the camera response function.


References

- Debevec, P. E., and Malik, J. Recovering high dynamic range radiance maps from photographs. In *Proceedings of ACM SIGGRAPH 97*, pp. 369–378, August 1997.
- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, second edition, Morgan Kaufmann/Elsevier, 2010.

# Tone reproduction

left Linear mapping of all dynamic range.

middle Linear mapping of lower 0.1% of dynamic range.

right Histogram adjustment [Ward et al. 1997].


References

- Debevec, P. E., and Malik, J. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH '97*, pp. 369–378, August 1997.
- Ward, G., Rushmeier, H., and Piatko, C. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics 3*(4), pp. 291–306, 1997.

# RGBE encoding (the .hdr format)

- RGBE $\rightarrow$ RGBA


FIGURE 3.2 Bit breakdown for the 32-bit/pixel RGBE (and XYZE) encodings.

$$
\begin{aligned}
R_W &= \frac{R_M + 0.5}{256} 2^{E-128} \\
G_W &= \frac{G_M + 0.5}{256} 2^{E-128} \\
B_W &= \frac{B_M + 0.5}{256} 2^{E-128}
\end{aligned}
$$

References

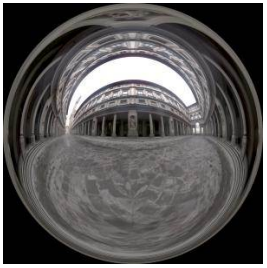

- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, second edition, Morgan Kaufmann/Elsevier, 2010.

# Light probes

- The angular map

$$
\begin{aligned}
r &= \frac{\arccos(-D_z)}{2\pi\sqrt{D_x^2 + D_y^2}} \\
(u, v) &= \left(\frac{1}{2} + rD_x, \frac{1}{2} + rD_y\right) ,
\end{aligned}
$$

where $(D_x, D_y, D_z)$ is the look-up direction into the environment map.




References

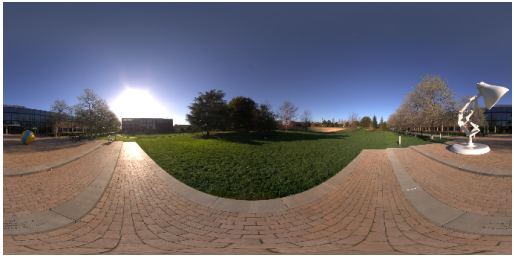
- Debevec, P. Image-based lighting. *IEEE Computer Graphics and Applications 22*(2), pp. 26–34, 2002.

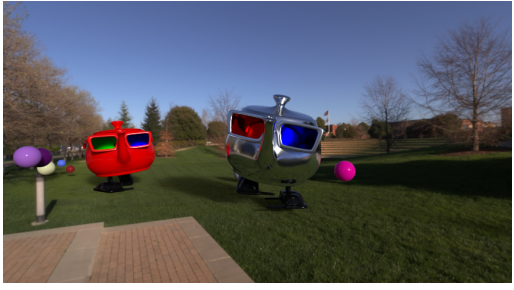
# Panoramic Format

- The latitude-longitude map

$$u = \frac{1}{2} + \frac{1}{2\pi} \arctan\left(\frac{D_x}{-D_z}\right)$$
$$v = \frac{1}{\pi} \arccos(-D_y),$$

where $(D_x, D_y, D_z)$ is the look-up direction into the environment map.

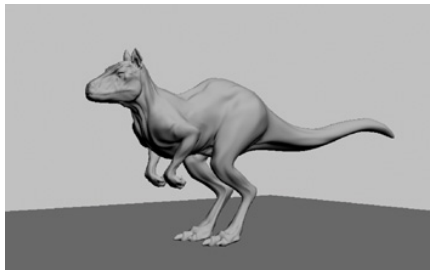




References

- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, second edition, Morgan Kaufmann/Elsevier, 2010.
- Pixar RenderMan Holdout Workflow: https://renderman.pixar.com/resources/RenderMan_20/risHoldOut.html.

# Ambient occlusion



- Using the Lambertian BRDF for materials, $f_r = \rho_d/\pi$; the cosine weighted hemisphere for sampling, $\text{pdf}(\vec{\omega}'_j) = \cos\theta/\pi$; and a visibility term $V$ for incident illumination, the Monte Carlo estimator for ambient occlusion is simply:

$$L_N(\boldsymbol{x}, \vec{\omega}) = \frac{1}{N}\sum_{j=1}^{N} \frac{f_r(\boldsymbol{x}, \vec{\omega}'_j, \vec{\omega}) L_i(\boldsymbol{x}, \vec{\omega}'_j) \cos\theta}{\text{pdf}(\vec{\omega}'_j)} = \rho_d(\boldsymbol{x}) \frac{1}{N}\sum_{j=1}^{N} V(\vec{\omega}'_j)\,.$$

# Environment illumination

$$L_N(\boldsymbol{x}, \vec{\omega}) = \frac{1}{N} \sum_{j=1}^{N} \frac{f_r(\boldsymbol{x}, \vec{\omega}'_j, \vec{\omega}) L_i(\boldsymbol{x}, \vec{\omega}'_j) \cos\theta}{\mathrm{pdf}(\vec{\omega}'_j)}$$

$$= \rho_d(\boldsymbol{x}) \frac{1}{N} \sum_{j=1}^{N} V(\vec{\omega}'_j) L_{\mathrm{env}}(\vec{\omega}'_j)\,,$$

- $L_{\mathrm{env}}(\vec{\omega}'_j)$ is the radiance received from an environment map by look-up using $\vec{\omega}'_j$.
- To cast shadows on the environment, one can use the concept of holdouts: inserting geometry to model objects seen in the environment.
- Holdout shading:

$$\rho_d(\boldsymbol{x}) = \frac{L_{\mathrm{env}}(\vec{\omega})}{\frac{1}{M} \sum_{k=1}^{M} L_{\mathrm{env}}(\vec{\omega}'_k)}\,.$$

- Importance sample the environment map to avoid noise. Then $L_N(\boldsymbol{x}, \vec{\omega}) = L_{\mathrm{env}}(\vec{\omega}) \frac{1}{N} \sum_{j=1}^{N} V(\vec{\omega}'_j)\,.$

# Loading an image as binary data and storing this in a WebGPU texture

```
async function load_texture(device, filename)
{
  const response = await fetch(filename);
  const blob = await response.blob();
  const img = await createImageBitmap(blob, { colorSpaceConversion: 'none' });

  const texture = device.createTexture({
    size: [img.width, img.height, 1],
    format: "rgba8unorm",
    usage: GPUTextureUsage.COPY_DST | GPUTextureUsage.TEXTURE_BINDING | GPUTextureUsage.RENDER_ATTACHMENT
  });
  device.queue.copyExternalImageToTexture(
    { source: img, flipY: true },
    { texture: texture },
    { width: img.width, height: img.height },
  );
  texture.sampler = device.createSampler({
    addressModeU: "clamp-to-edge",
    addressModeV: "clamp-to-edge",
    minFilter: "nearest",
    magFilter: "nearest",
  });
  return texture;
}
```