Tone Mapping

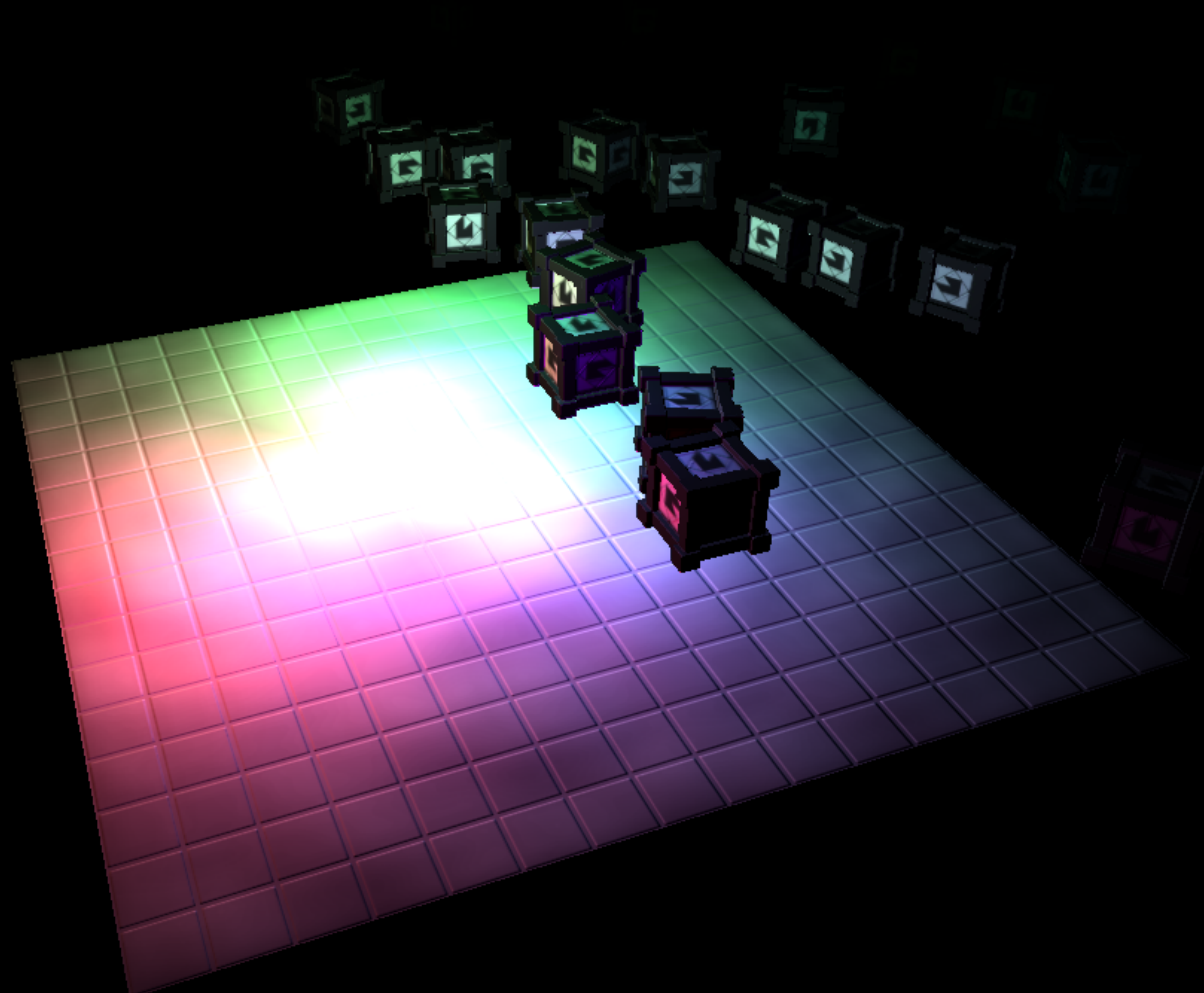
+ a little background on High Dynamic Range Imaging

High Dynamic Range Imaging

- Historically, limited computational resources meant that it was only possible to represent a small range of color values
- In the real world however, the range of luminance values can be many orders of magnitude.
- Now that computers have improved (in terms of memory and processing power) we can create and store images with these high ranges in luminance value.
- But we are still stuck in terms of displaying them, as there is not wide spread adoption of HDR displays

Where does HDR data come from?

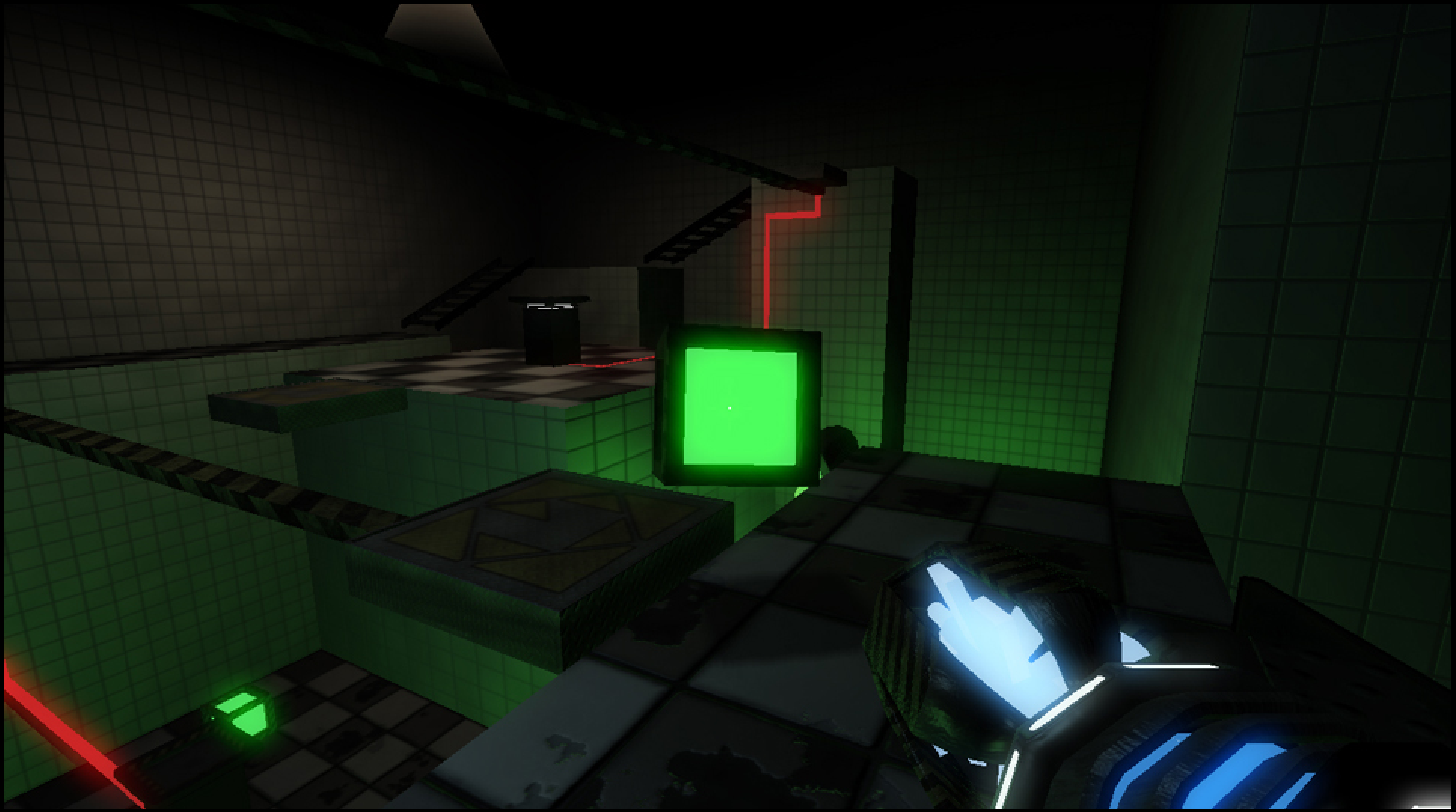
- HDR data comes from two primary sources:
    - Synthetically generated images
        - Modern renderers handle lighting calculations in HDR in order to improve the realism of rendering
    - Captured images
        - Some highend cameras have sensors which can automatically capture HDR images
        - Even a L(ow)DR camera can take a series of "bracketed" exposure images, which can be computed into a single HDR image




HYATT

Bloom

- Its <u>NOT</u> HDR!
- This is a trick often used in video games and other synthetic rendering that exploits the human visual system to give the impression of a greater luminance range than is actually presented



Glasshouse by Rocktopus Games

RGBE

- An image format developed by Greg Ward in order to handle the storage and processing of HDR image data
- Made up of 4 bytes:
  - (1 byte R, 1 byte G, 1 byte B, 1 byte E(xponent))
- By using a shared exponent value a significant space savings can be achieved
  - Compare with using a 12 or 16 bit floating point value for each R, G and B component
- In practice, this solution handles colors and very wide range of luminance values quite well

"Hey, wait a minute, I thought you said that we can't display HDR images on a regular computer monitor...?"

TONE MAPPING

..to the rescue!

Tone Mapping Operator Categories

Spatially Uniform / Global - A single function is used for all pixel in the image

Spatially Varying / Local - Function varies based upon each pixel's brightness and the brightness of nearby pixels

Note that a spatially varying operator is possible because the human eye is more sensitive to local contrast than overall luminance.

Local Contrast Preservation

Our basic goal is to preserve the local contrast in the image

This is even more important than preserving our overall brightness

Maximum To White

- The simplest tone mapping operator
- Two step process:
  - Loop through all the pixels in the image and find the greatest luminance
  - Loop through all the pixels and apply a uniform scaling such that the largest luminance maps to the maximum luminance value of the display

Maximum To White: Issues

- Two main disadvantages to this simplistic approach:
  - Doesn't account for the human visual system at all: if the lights in the scene are made 100 times brighter, this tone mapping operation will still produce the same image
  - A small number of bright pixels can cause the output to be too dark to see.

- Useful only in images with very limited dynamic range

Contrast-Based Scale Factor

- This tone mapping operator focuses on preserving contrast in the displayed image
- Developed by Greg Ward
- Based upon actual HVS research regarding the smallest change in luminance that is noticeable to a human observer at a given adaption luminance
- The larger the adaption luminance, the larger the change in luminance that is required to be noticeable

Ward, G. J. (A contrast-based scalefactor for luminance display. In P.Heckbert (Ed.) Graphics Gems IV, pp.415-421.)

Contrast-Based Scale Factor

- This operator then tries to preserve *contrast visibility* - given a region of the original image that would be noticeably different than its neighbor, the operator tries to scale these so they will still be just noticeably different

- Research has shown that given an adaption luminance in the photopic range (relatively well lit allowing color perception with our cone cells) Y^a, a reasonable model of the minimum change in luminance required to be visible is:

$$\Delta Y(Ya) = 0.0594(1.219 + (Y^a)^{0.4})^{2.5}$$

Contrast-Based Scale Factor

Based upon the previous formula, we can calculate the scale factor as follows:

$$\Delta Y(Y^a_d) = s\Delta Y(Y^a_w)$$

where Y^a_d is the display adaption luminance and Y^a_w is the world adaption luminance (for someone observing the real scene)

Some mathematical manipulation to get out our scale factor:

$$s = (\ (\ 1.219 + (Y^a_d)^{0.4}\ )\ /\ (\ 1.219 + (Y^a_w)^{0.4}\ )\ )^{2.5}$$

Contrast-Based Scale Factor

Obvious question: How do we compute the world adaption luminance Y^a_w?

Ideally, we could base this on what the viewer was looking at and for how long

However, since this isn't practical, we just use a log average of all the luminance values in the original image

By using the log average, we mitigate the effect of small, bright regions on the overall luminance average

Varying Adaption Luminance

- This approach calculates the local adaption luminance which varies smoothly over the image
- Once the local adaption luminance is calculated we can calculate a local scale factor in a manner similar to before
- Developed by Michael Ashikhmin

- Main issue here is preventing artifacts at boundaries between high and low luminance regions

Ashikhmin, M. 2002. A tone mapping algorithm for high contrast images. In The Proceedings of 13th Eurographics Workshop on Rendering, Pisa, Italy, pp. 145-55

Varying Adaption Luminance

- For example, if we apply a scaling factor that is based on high luminance to dim pixel we produce a dark halo
- In order to avoid this we can make the simple modification which only dims pixels based upon nearby low luminance areas

Spatially Varying Non-linear Scale

- Similar to varying adaption luminance, with the addition of applying the scaling in a non-linear fashion
- Logarithmic response curve causes darker pixels to require less chance in brightness to cause a given change in the output image
- Developed by Reinhard et al.

Reinhard, E., M. Stark, P. Shirley and J. Ferwerda. 2002. Photographic tone reproduction for digital images. ACM Transactions on Graphics 21(3), 267-76. Proceedings of ACM SIGGRAPH 2002.

Reinhard

Drago

http://www.mpi-inf.mpg.de/resources/tmo/NewExperiment/TmoOverview.html

Fattal

Retinex

Tumblin

Durand

Ward

Ashikhman

Demo

Tone mapping *IN ACTION* !