Great Ideas in Computational Photography
HDR Imaging, Tone Mapping, Coded Imaging

CSC2529
David Lindell
University of Toronto
cs.toronto.edu/~lindell/teaching/2529

*slides adapted from Yannis Gkioulekas, Gordon Wetzstein, Fredo Durand, Marc Levoy, James Hays, Sylvain Paris, Sam Hasinoff

Announcements

- HW3 due Wednesday 18/10
- HW4 is out
- Project proposal due in 1 month!

- Problem session for HW4 tomorrow

wikipedia

-4 stops

wikipedia

exposure sequence

-2 stops

wikipedia

exposure sequence

2 stops

wikipedia

4 stops

Motivation

wikipedia

HDR
contrast
reduction
(scaling)

wikipedia

HDR
local tone
mapping

Exposure control

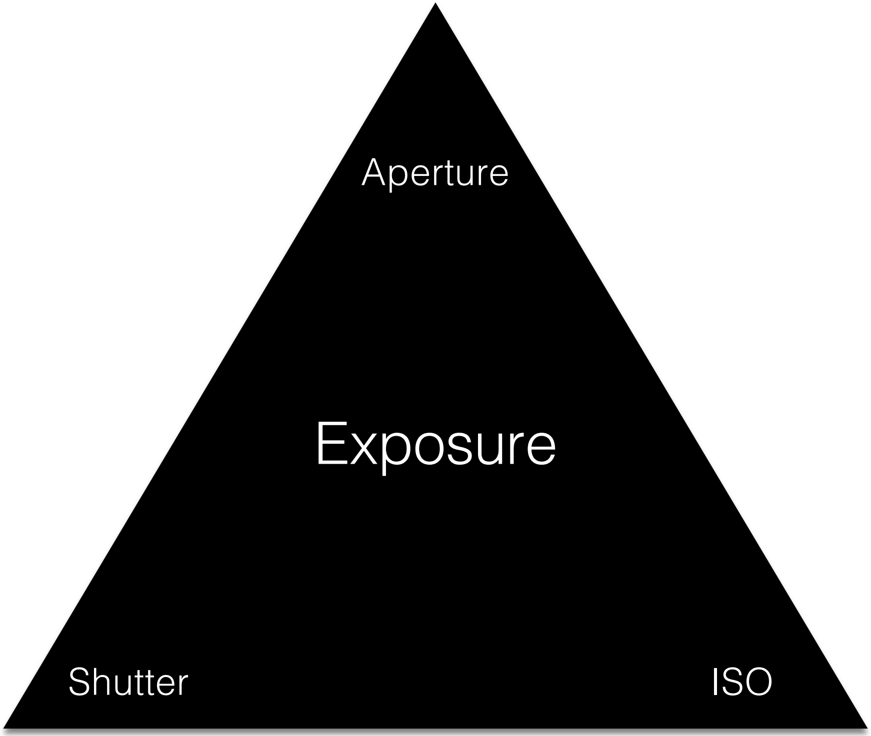
What is exposure?

Roughly speaking, the “brightness” of a captured image given a fixed scene.

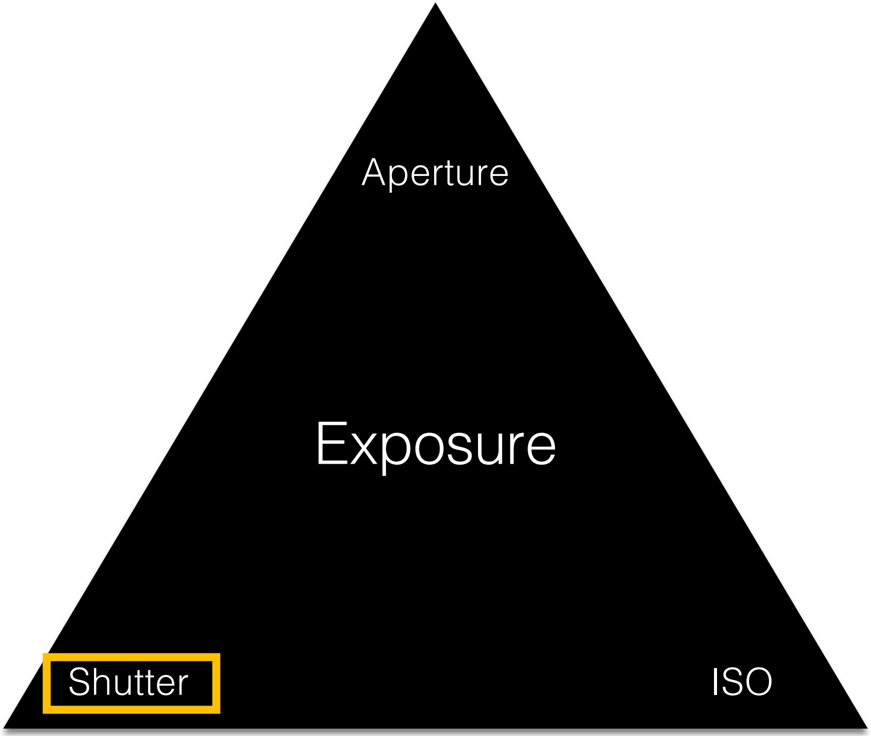
$$\text{Exposure} = \text{Gain} \times \text{Flux} \times \text{Time}$$

- Flux is controlled by the aperture.
- Time is controlled by the shutter speed.
- Gain is controlled by the ISO.

Exposure controls brightness of image

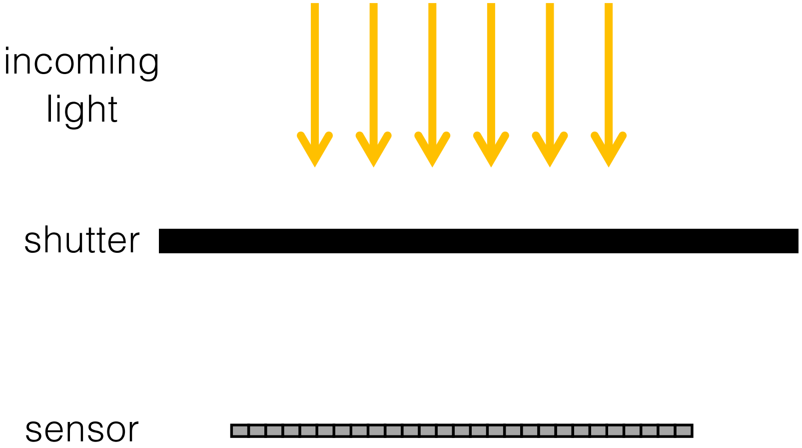


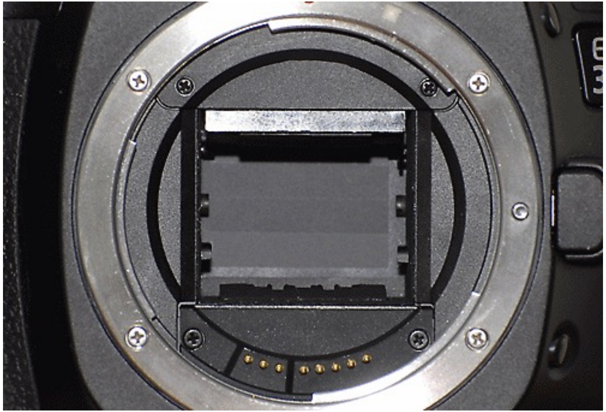
Exposure controls brightness of image



Shutter speed

Controls the length of time that shutter remains open.

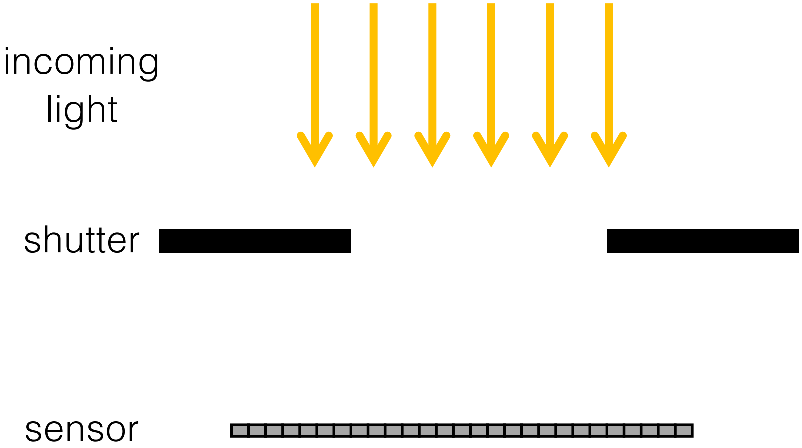


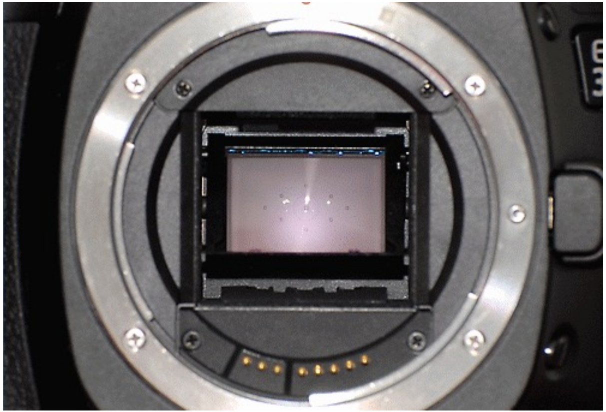


closed shutter

Shutter speed

Controls the length of time that shutter remains open.



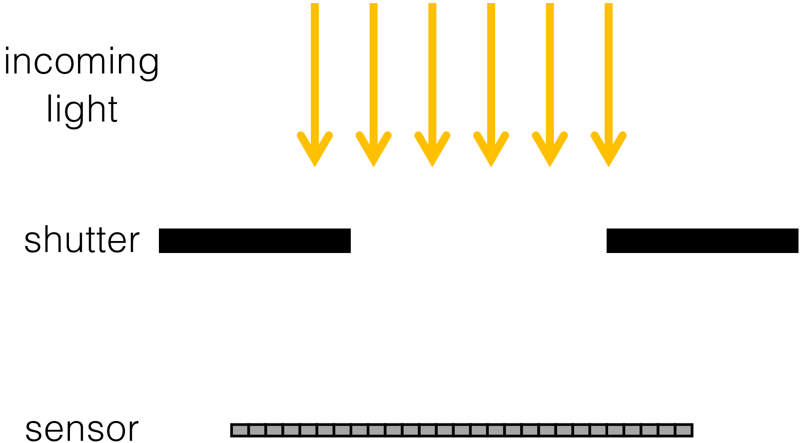


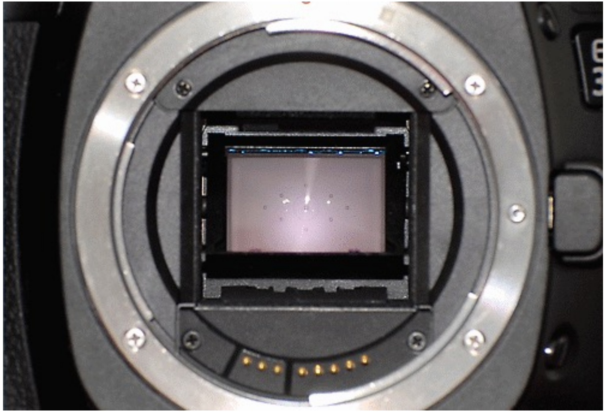
open shutter

Nikon D3s

Shutter speed

Controls the period of time that shutter remains open.



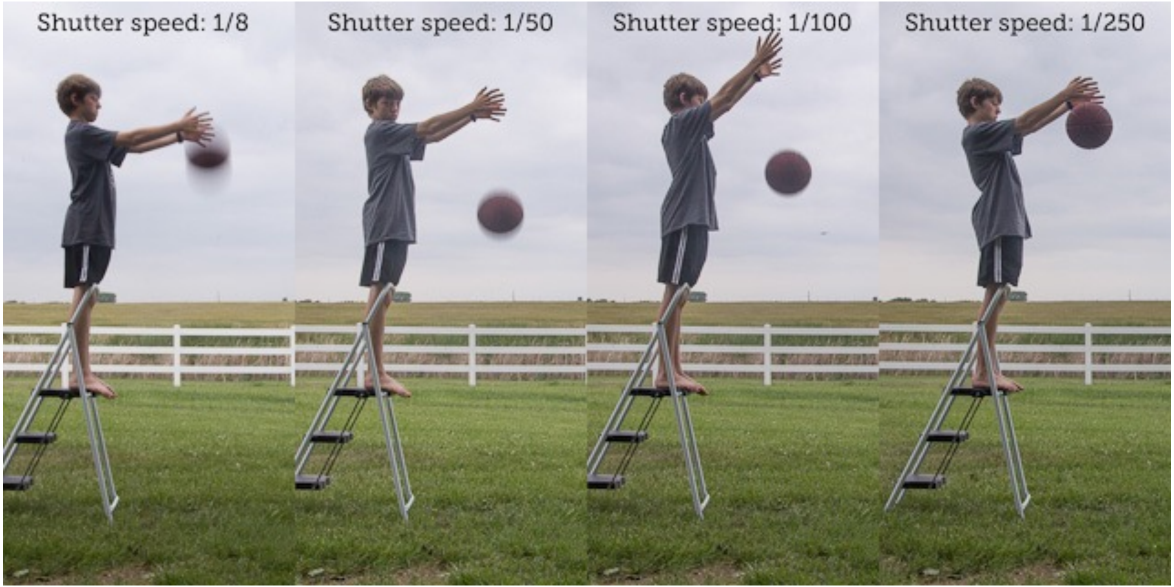


open shutter

What happens to the image as we increase shutter speed?

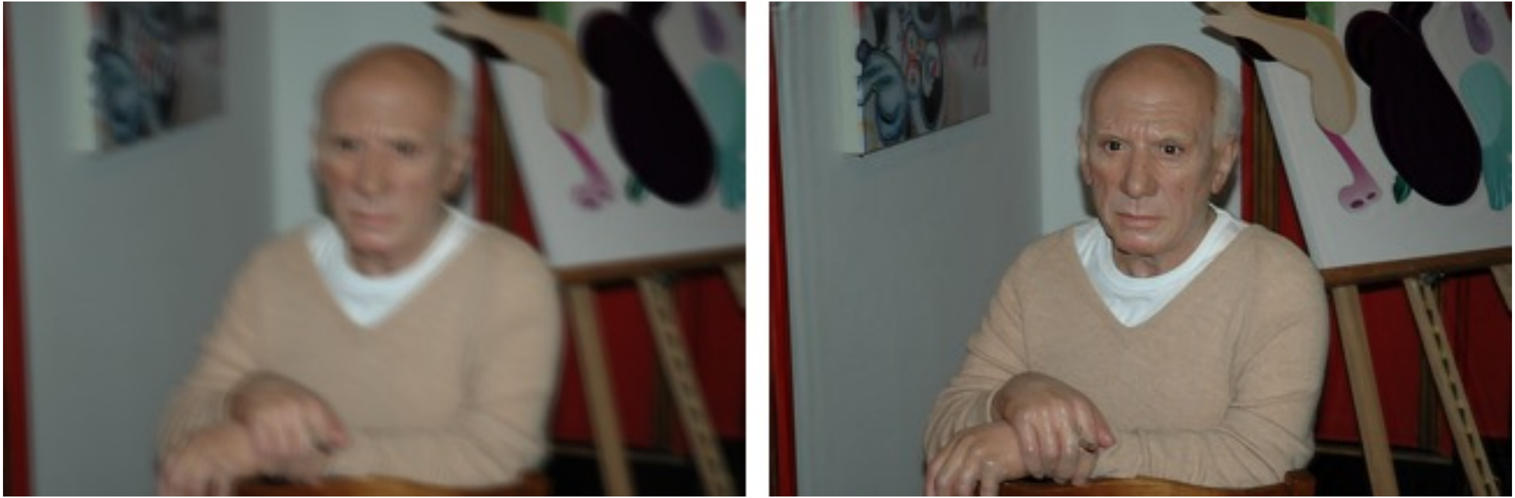
Side-effects of shutter speed

Moving scene elements appear blurry.



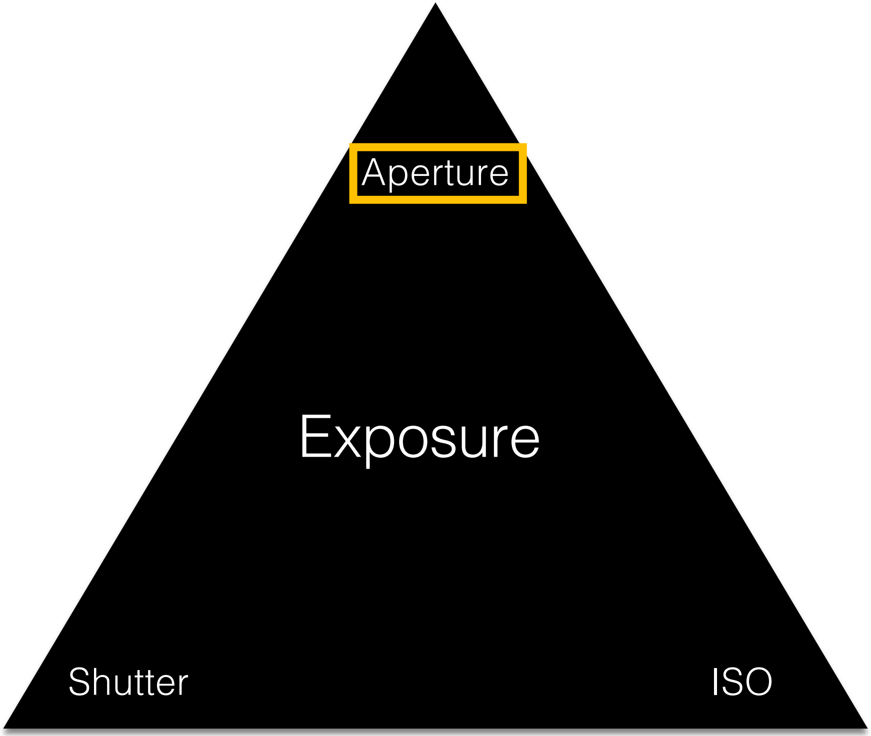
How can we “simulate” decreasing the shutter speed?

Motion deblurring



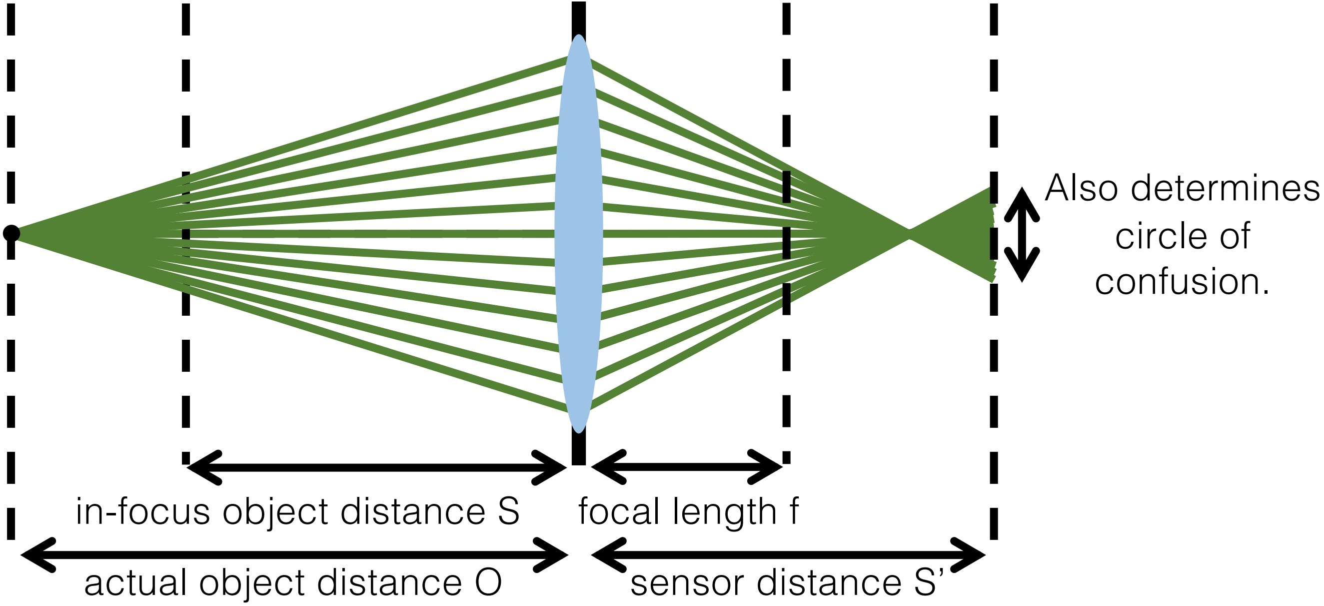
Shah et al. High-quality Motion Deblurring from a Single Image, SIGGRAPH 2008

Exposure controls brightness of image



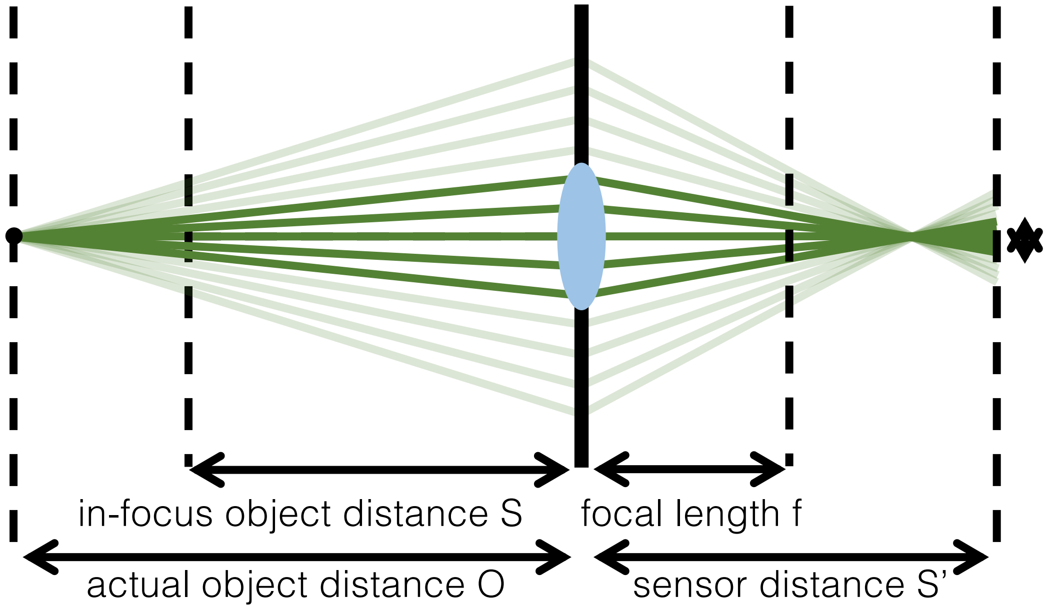
Aperture size

Controls area of lens that lets light pass through.



Aperture size

Controls area of lens that lets light pass through.



Also determines circle of confusion.

Aperture size

Most lenses have apertures of variable size.

- The size of the aperture is expressed as the “f-number”: The bigger this number, the smaller the aperture.

f / 1.4 f / 2.8 f / 4 f / 8 f / 16

You can see the aperture by removing the lens and looking inside it.

Side-effects of aperture size

Depth of field decreases as aperture size increases.

- Having a very sharp depth of field is known as “bokeh”.

How can we simulate bokeh?

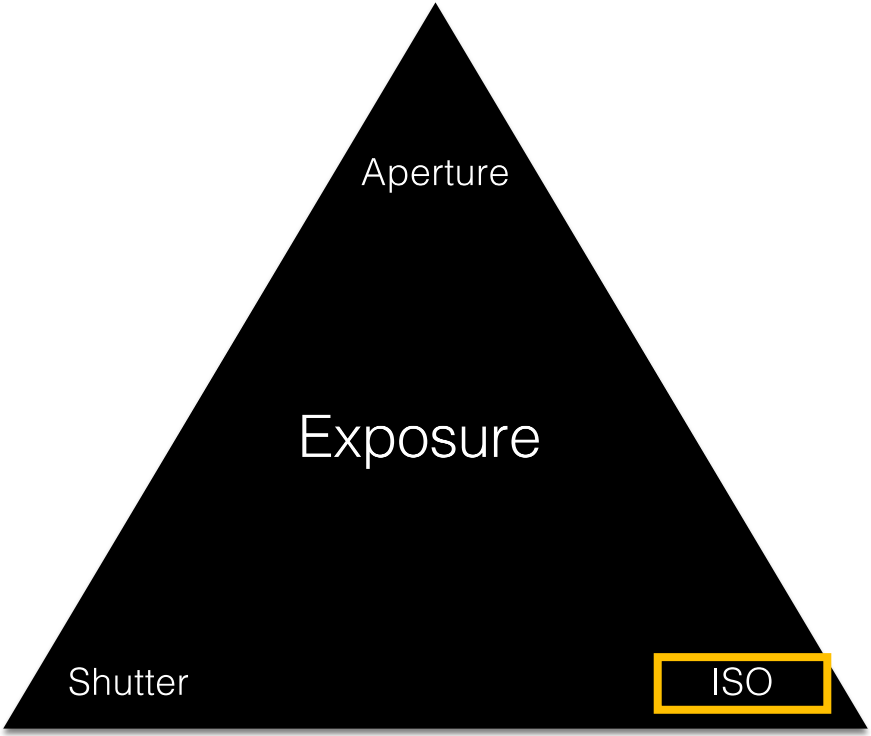
How can we simulate bokeh?

Infer per-pixel depth, then blur with depth-dependent kernel.

- Example: Google camera “lens blur” feature

Barron et al., “Fast Bilateral-Space Stereo for Synthetic Defocus,” CVPR 2015

Exposure controls brightness of image



Side-effects of increasing ISO

Image becomes very grainy because noise is amplified.

ISO 80

ISO 800

ISO 1600

Note about the name ISO

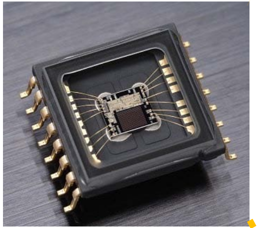
ISO is not an acronym.

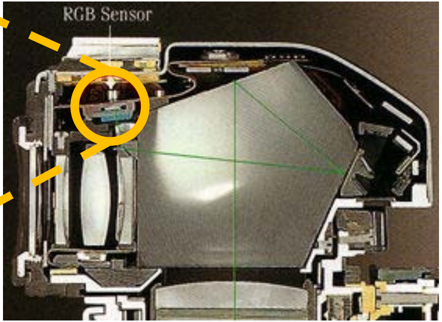
- It refers to the International Organization for Standardization.
- ISO comes from the Greek word ίσος, which means equal.
- It is pronounced (roughly) eye-zo, and should not be spelled out.

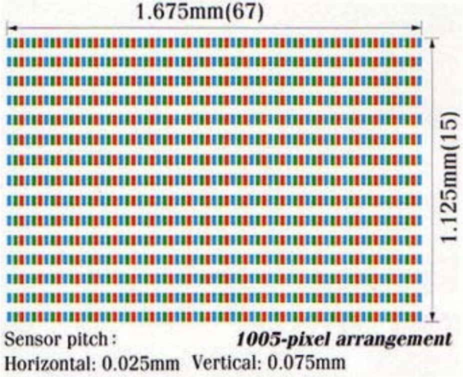
Light metering

Light metering in modern cameras

- SLR cameras use a separate low-resolution sensor that is placed at the focusing screen.







- Mirrorless cameras use measurements directly from the main sensor.

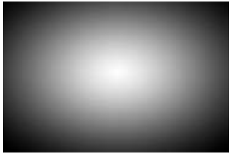
Light metering in modern cameras

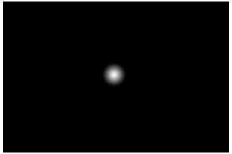
- Measurements are averaged to produce a single intensity estimate, which is assumed to correspond to a scene of 18% reflectance (the “key”).

- Exposure is set so that this average is exposed at the middle of the sensor’s dynamic range.

- Averaging can be done in many ways:

  1. Center-weighed.
  2. Spot.
  3. Scene-specific preset (portrait, landscape, horizon).
  4. “Intelligently” using proprietary algorithm.





Our devices do not match the world

The world has a high dynamic range

1

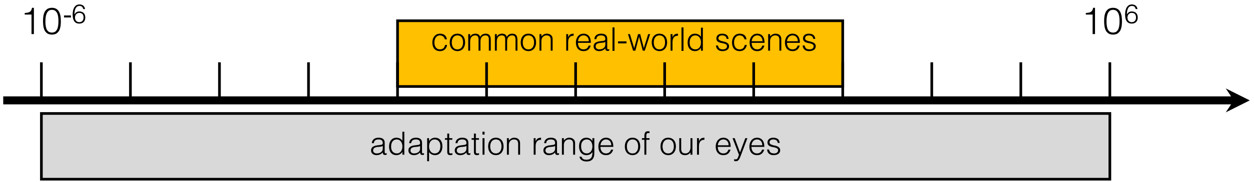
1500

25,000

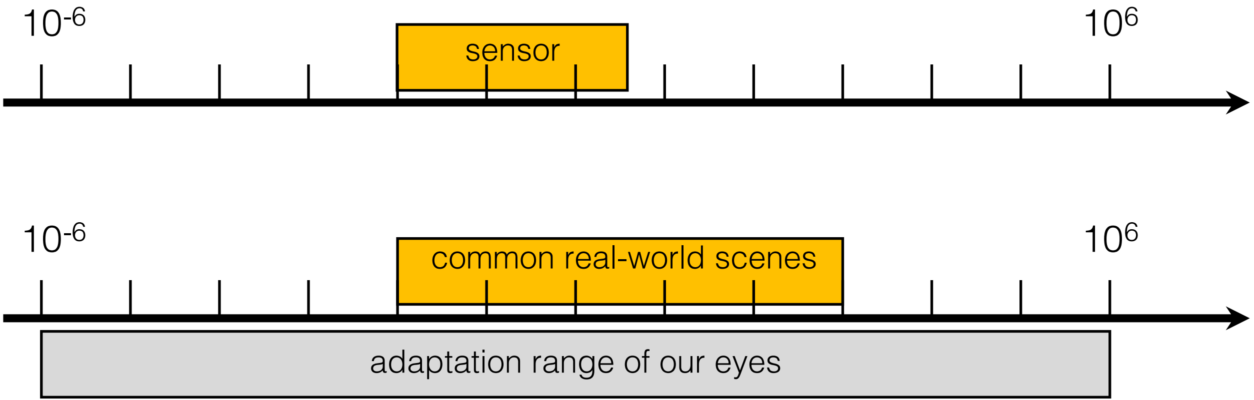
400,000

2,000,000,000

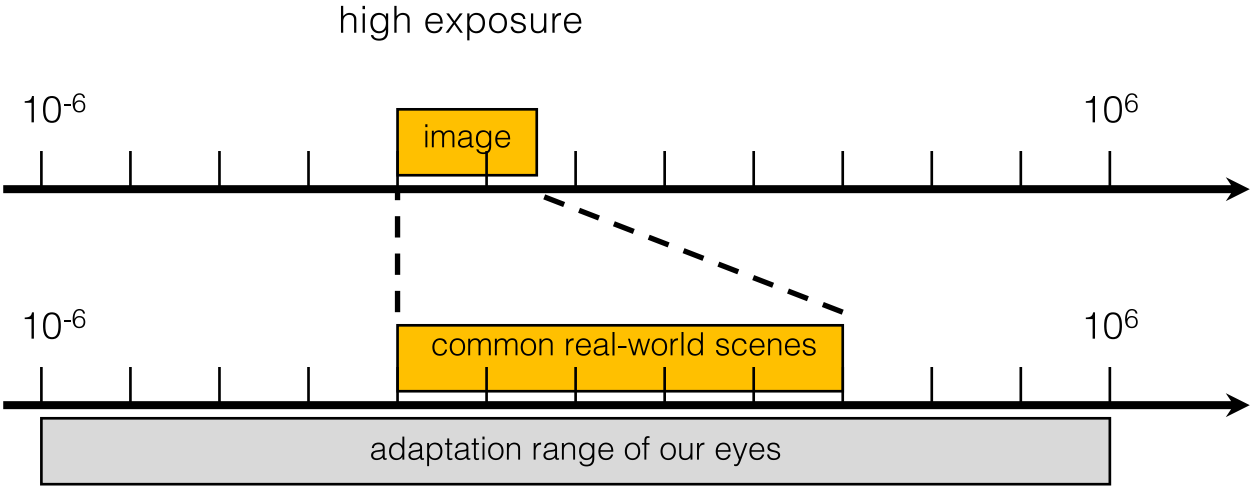
The world has a high dynamic range



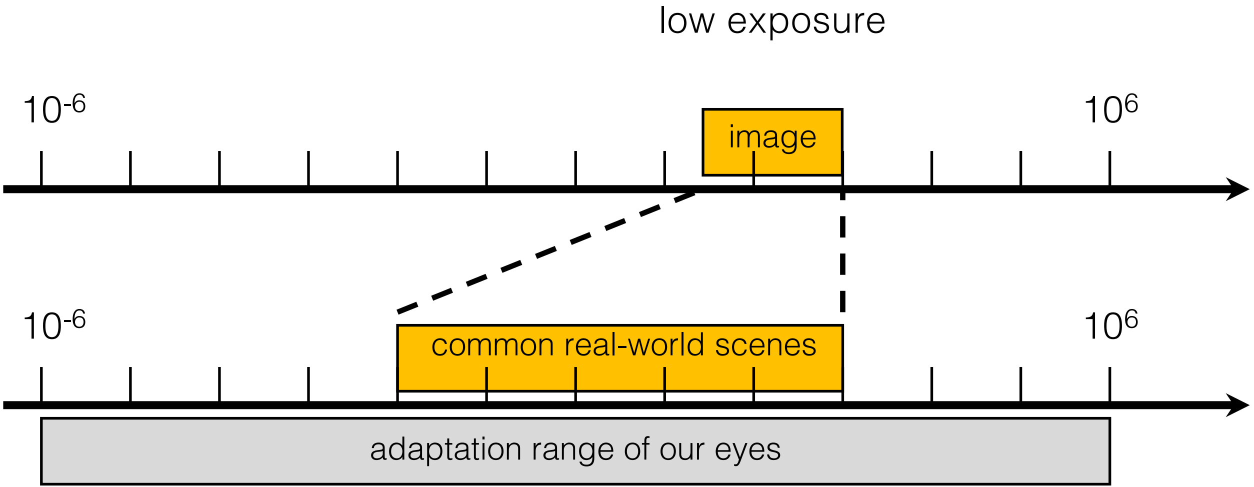
(Digital) sensors also have a low dynamic range



(Digital) images have an even lower dynamic range

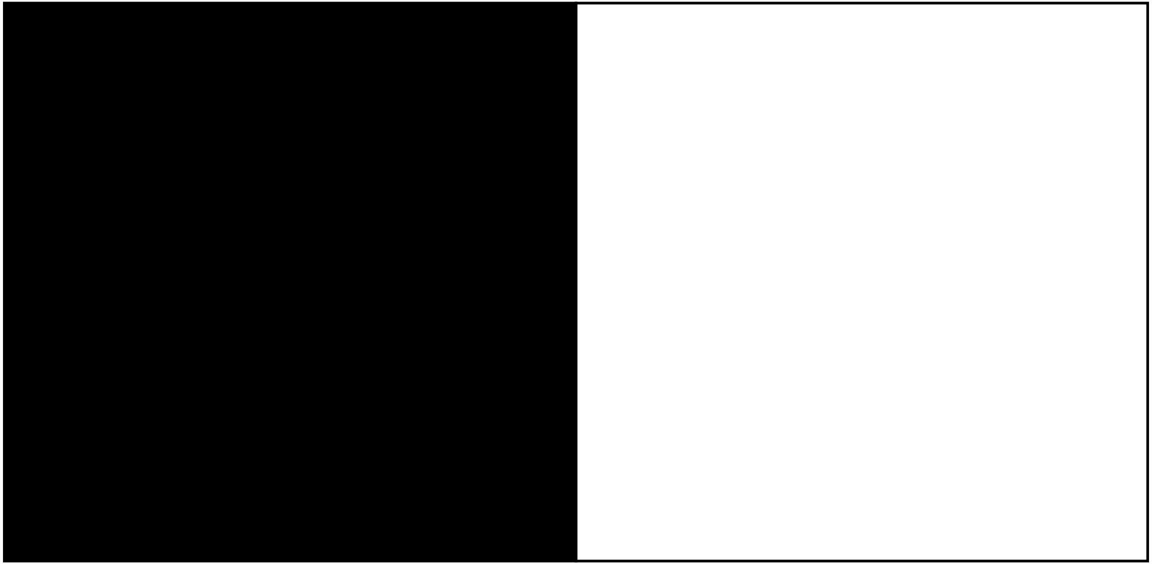


(Digital) images have an even lower dynamic range



(Digital) images have an even lower dynamic range

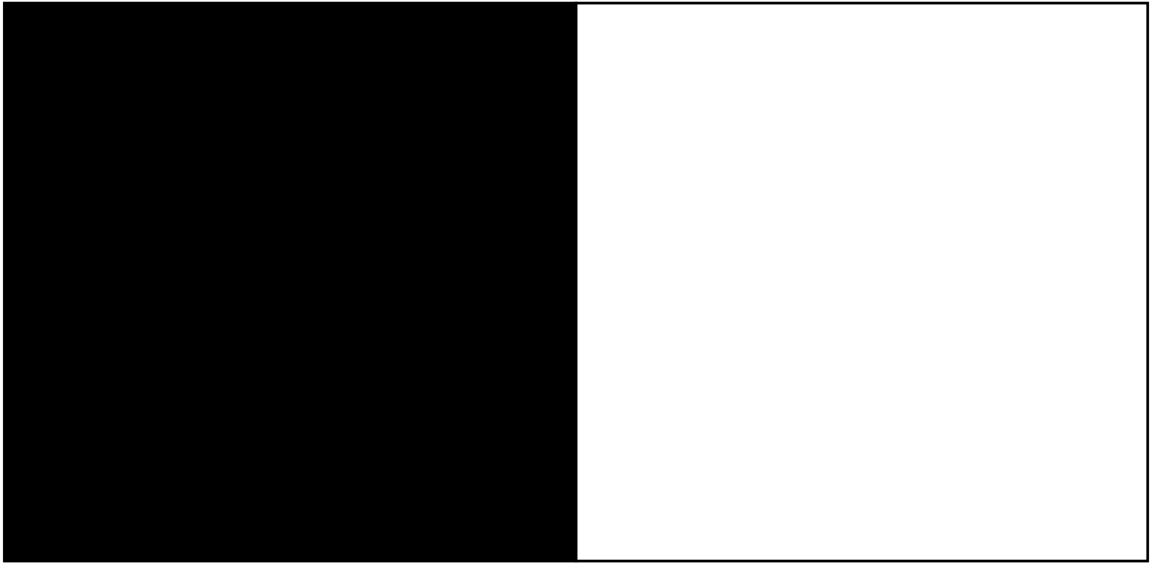
Any guesses about the dynamic range of a standard 0-255 image?



pure black pure white

(Digital) images have an even lower dynamic range

Any guesses about the dynamic range of a standard 0-255 image?



about 50x brighter

pure black

pure white

Our devices do not match the real world

• 10:1	photographic print (higher for glossy paper)
• 20:1	artist's paints
• 200:1	slide film
• 500:1	negative film
• 1000:1	LCD display
• 2000:1	digital SLR (at 12 bits)
• 100000:1	real world

Two challenges:

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

Our devices do not match the real world

• 10:1	photographic print (higher for glossy paper)
• 20:1	artist's paints
• 200:1	slide film
• 500:1	negative film
• 1000:1	LCD display
• 2000:1	digital SLR (at 12 bits)
• 100000:1	real world

HDR imaging and tonemapping are distinct techniques with different goals

HDR imaging compensates for sensor limitations

Two challenges:

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

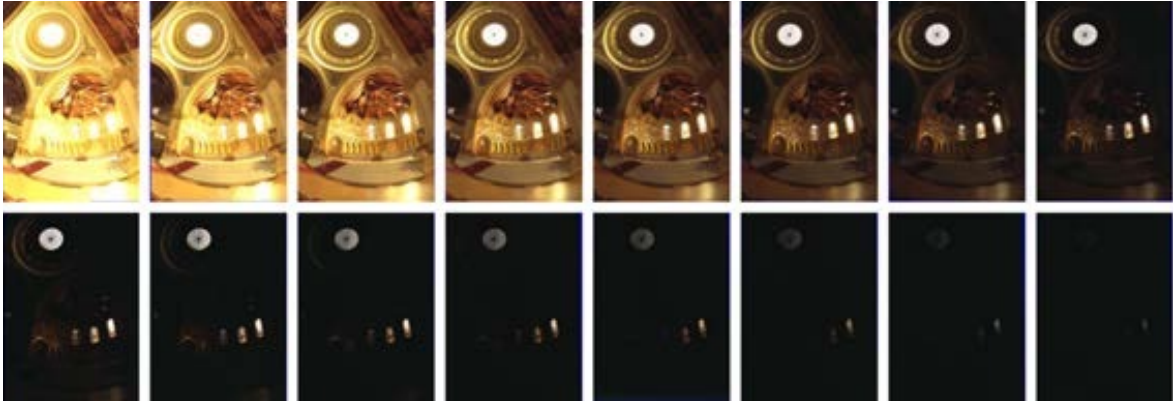
Tonemapping compensates for display limitations

High dynamic range imaging


HYATT

Key idea

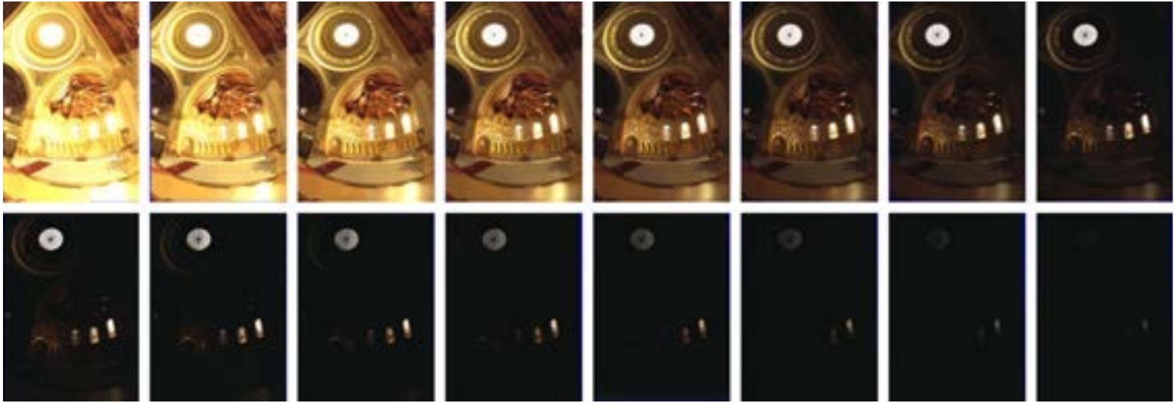
1. Exposure bracketing: Capture multiple LDR images at different exposures



2. Merging: Combine them into a single HDR image

Key idea

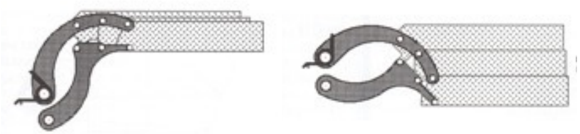
1. Exposure bracketing: Capture multiple LDR images at different exposures



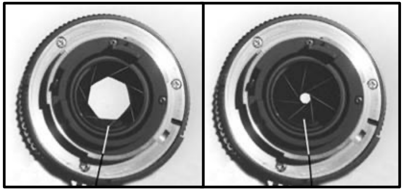
2. Merging: Combine them into a single HDR image

Ways to vary exposure

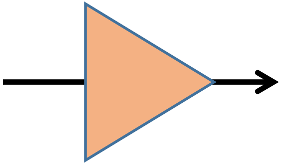
1. Shutter speed



2. F-stop (aperture, iris)



3. ISO



4. Neutral density (ND) filters

Pros and cons of each for HDR?

Ways to vary exposure

1. Shutter speed
   - Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   - Pros: repeatable, linear
   - Cons: noise and motion blur for long exposure

Ways to vary exposure

1. Shutter speed
   - Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   - Pros: repeatable, linear
   - Cons: noise and motion blur for long exposure
2. F-stop (aperture, iris)
   - Range: about f/0.98 to f/22 (3 orders of magnitude)
   - Pros: fully optical, no noise
   - Cons: changes depth of field

Ways to vary exposure

1. Shutter speed
   – Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   – Pros: repeatable, linear
   – Cons: noise and motion blur for long exposure
2. F-stop (aperture, iris)
   – Range: about f/0.98 to f/22 (3 orders of magnitude)
   – Pros: fully optical, no noise
   – Cons: changes depth of field
3. ISO
   – Range: about 100 to 1600 (1.5 orders of magnitude)
   – Pros: no movement at all
   – Cons: noise

Ways to vary exposure

1. Shutter speed
   - Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   - Pros: repeatable, linear
   - Cons: noise and motion blur for long exposure
2. F-stop (aperture, iris)
   - Range: about f/0.98 to f/22 (3 orders of magnitude)
   - Pros: fully optical, no noise
   - Cons: changes depth of field
3. ISO
   - Range: about 100 to 1600 (1.5 orders of magnitude)
   - Pros: no movement at all
   - Cons: noise
4. Neutral density (ND) filters
   - Range: up to 6 densities (6 orders of magnitude)
   - Pros: works with strobe/flash
   - Cons: not perfectly neutral (color shift), extra glass (interreflections, aberrations)

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each “stop” is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each "stop" is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Questions:

1. How many exposures?
2. What exposures?

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each “stop” is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

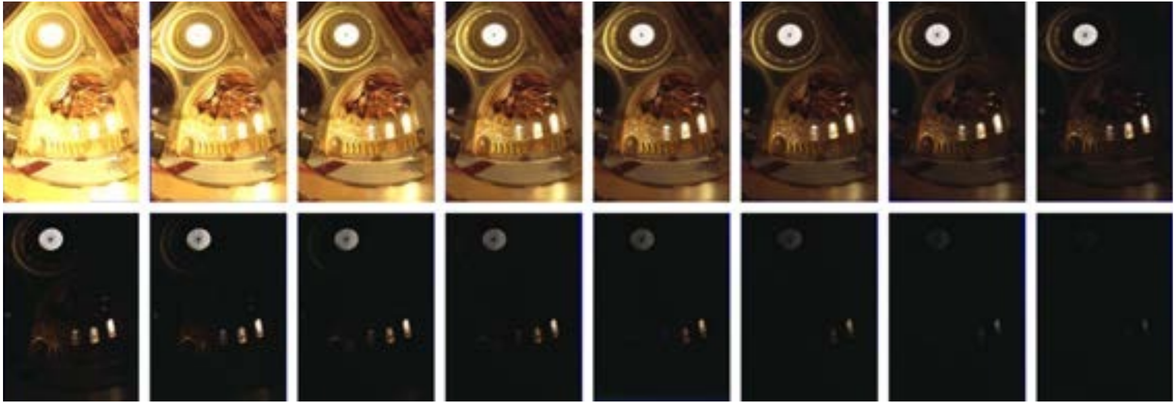
Questions:

1. How many exposures?
2. What exposures?

Answer: Depends on the scene, but a good default is 5 exposures, the metered exposure and +/- 2 stops around that.

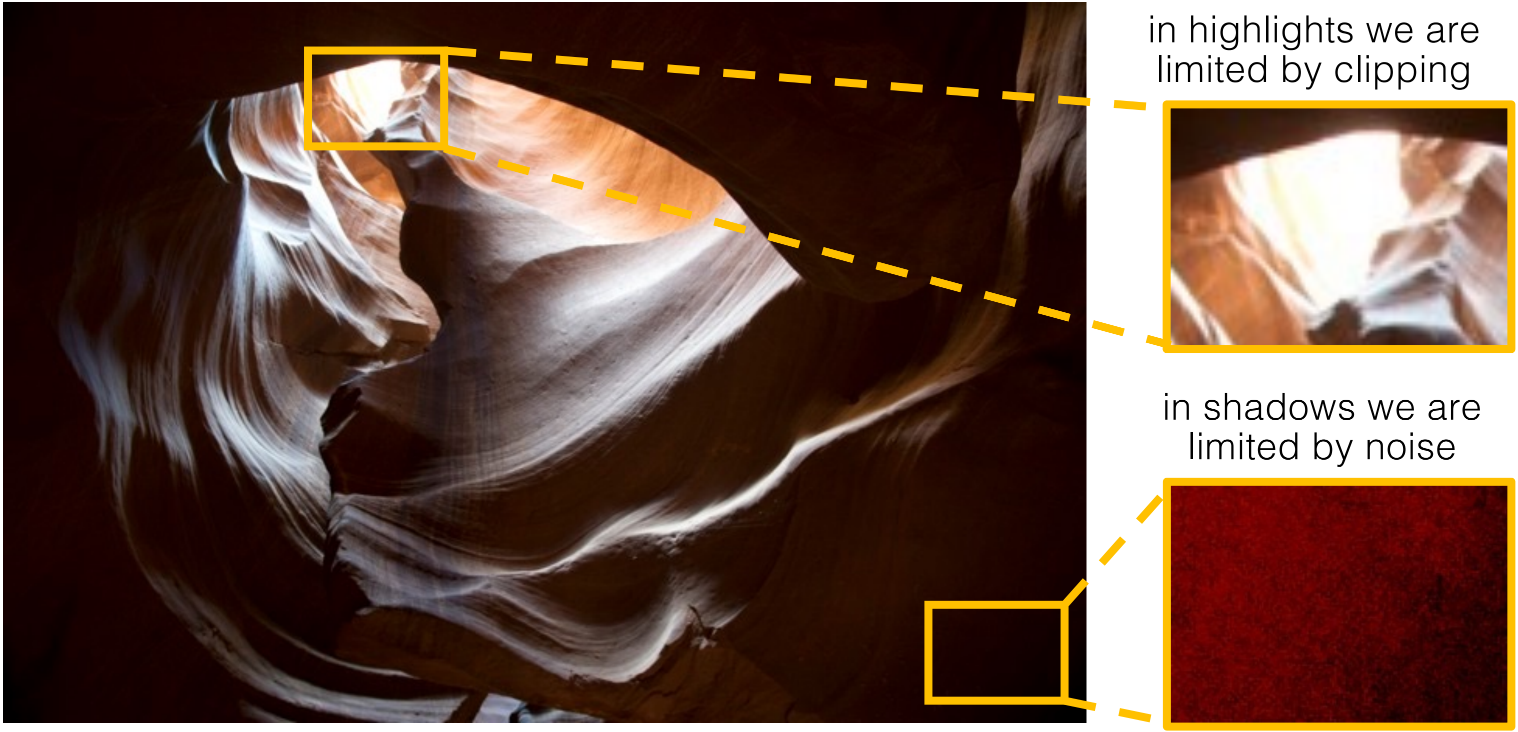
Key idea

1. Exposure bracketing: Capture multiple LDR images at different exposures



2. Merging: Combine them into a single HDR image

Over/under exposure



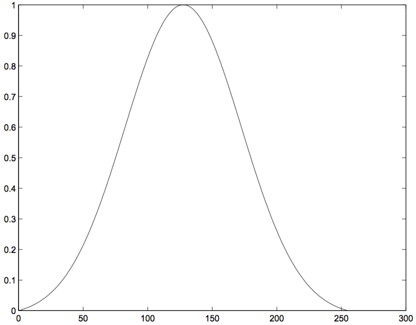
HDRI – Merging LDR Exposures

- compute a weight (confidence) that a pixel is well-exposed

  → (close to) saturated pixel = not confident, pixel in center of dynamic range = confident!

$$w_{ij} = \exp\left(-4 \frac{\left(I_{lin_{ij}} - 0.5 \right)^2}{0.5^2} \right)$$

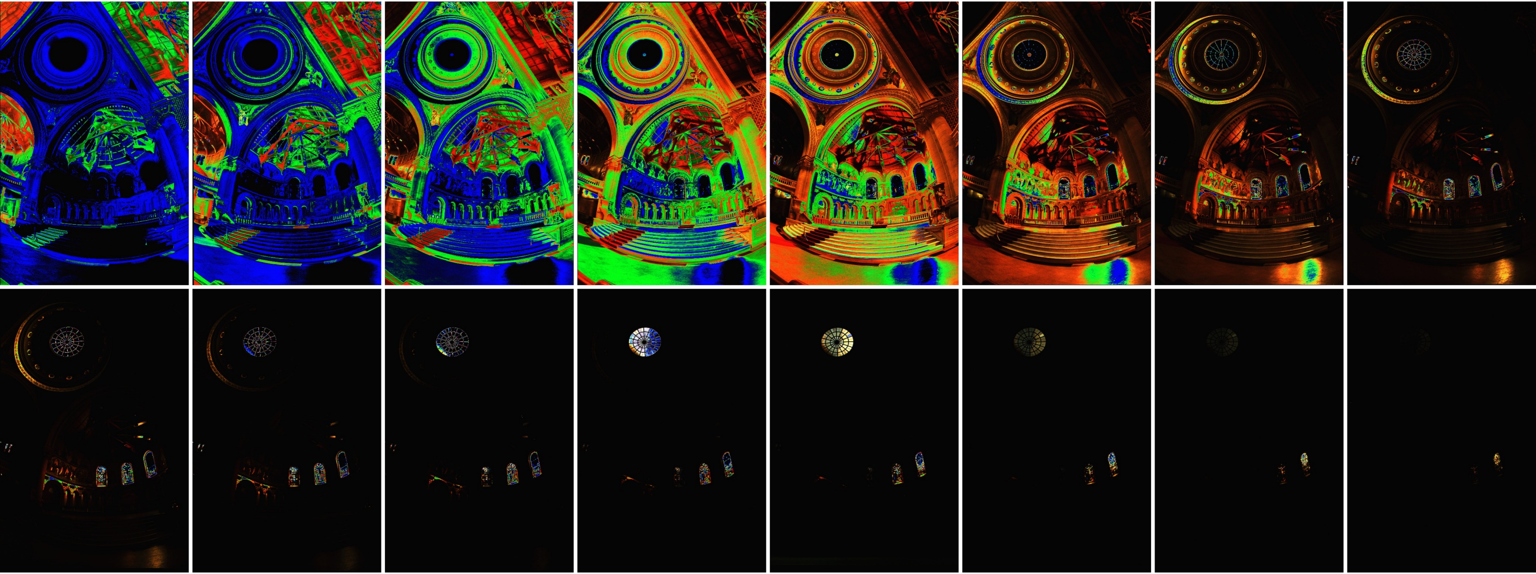
or mean pixel value, e.g. 127.5 if I in [0, 255]



HDRI – Merging LDR Exposures

- compute per-color-channel-per-LDR-pixel weights

$$w_{ij} = \exp\left(-4\frac{\left(I_{lin_{ij}} - 0.5\right)^2}{0.5^2}\right)$$



HDRI – Merging LDR Exposures

- define least-squares objective function in log-space → perceptually linear: $$\underset{X}{\text{minimize}} \quad O = \sum_i w_i \left(\log\left(I_{lin_i} \right) - \log\left(t_i X \right) \right)^2$$

- equate gradient to zero:

$$\frac{\partial O}{\partial \log\left(X\right)} = -2 \sum_i w_i \left(\log\left(I_{lin_i} \right) - \log\left(t_i \right) - \log\left(X \right) \right) = 0$$

- gives: $$\widehat{X} = \exp\left(\frac{\sum_i w_i \left(\log\left(I_{lin_i} \right) - \log\left(t_i \right) \right)}{\sum_i w_i} \right)$$

HDRI – Merging LDR Exposures

- define least-squares objective function in log-space → perceptually linear:

$$\underset{X}{\text{minimize}} \quad O = \sum_i w_i \left(\log\left(I_{lin_i}\right) - \log\left(t_i X\right) \right)^2$$

- equate gradient to zero:

$$\frac{\partial O}{\partial \log(X)} = -2 \sum_i w_i \left(\log\left(I_{lin_i}\right) - \log\left(t_i\right) - \log\left(X\right) \right) = 0$$

- gives:

$$\hat{X} = \exp\left(\frac{\sum_i w_i \left(\log\left(I_{lin_i}\right) - \log\left(t_i\right) \right)}{\sum_i w_i} \right)$$

What if I cannot use raw?

Radiometric calibration

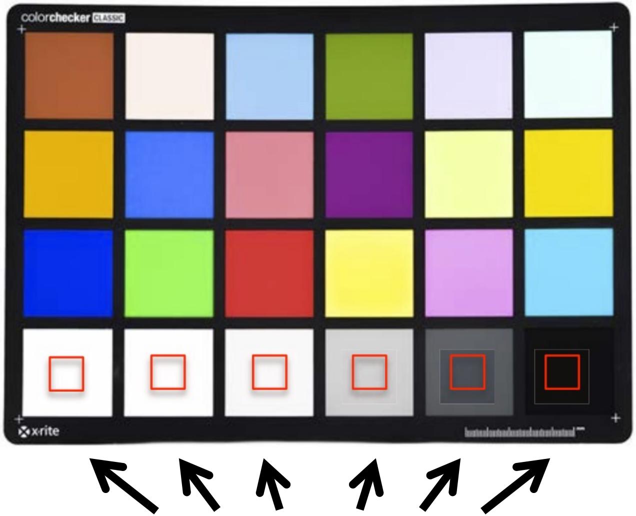
Radiometric calibration

The process of measuring the camera's response curve. Can be done in three ways:

- Take images of scenes with different flux while keeping exposure the same.
- Takes images under different exposures while keeping flux the same.
- Takes images of scenes with different flux and under different exposures.

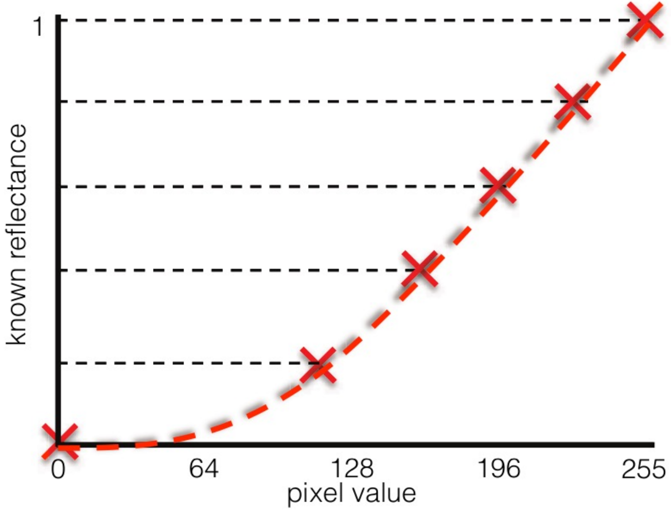
Same camera exposure, varying scene flux

Colorchecker: Great tool for radiometric and color calibration.



Patches at bottom row have log-reflectance that increases linearly.

e.g. JPEG



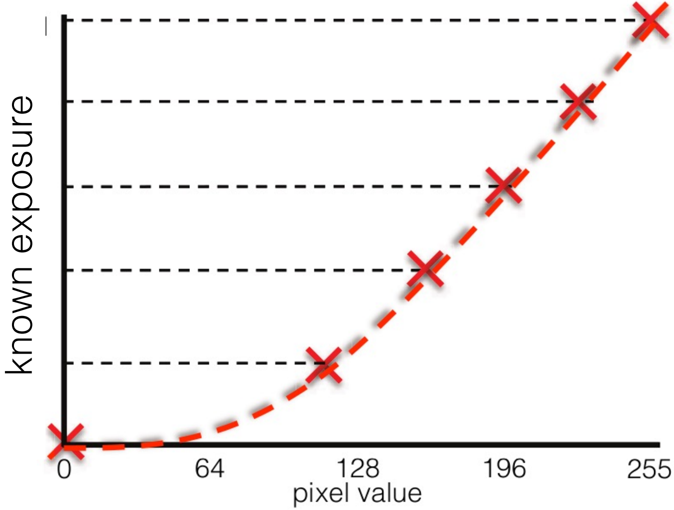
Different values correspond to patches of increasing reflected flux.

Same scene flux, varying camera exposure

White balance card: Great tool for white balancing and radiometric calibration.

All points on (the white part of) the target have the same reflectance.

e.g. JPEG

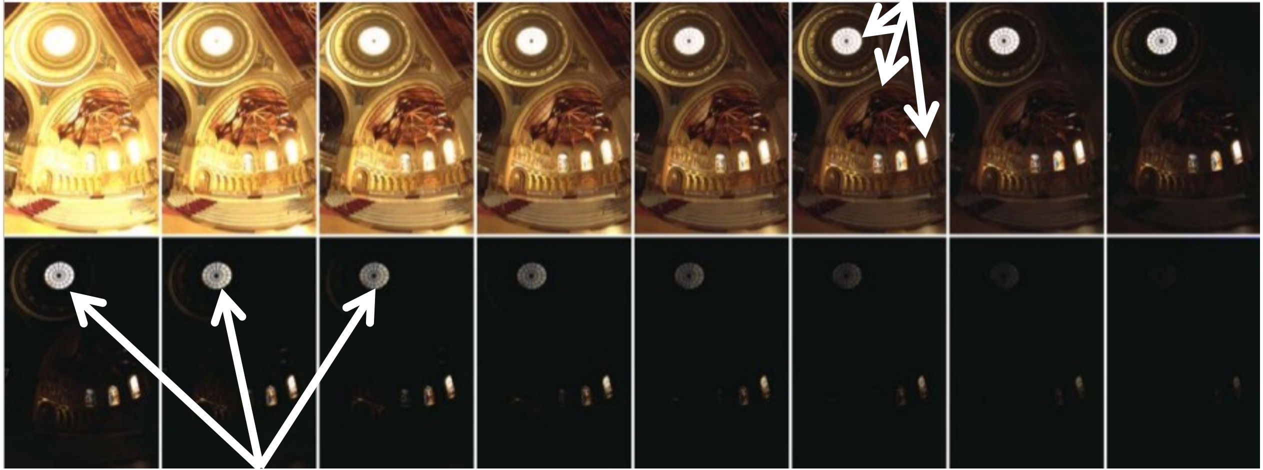


Different values correspond to images taken under increasing camera exposure.

Varying both scene flux and camera exposure

You can do this using the LDR exposure stack itself.

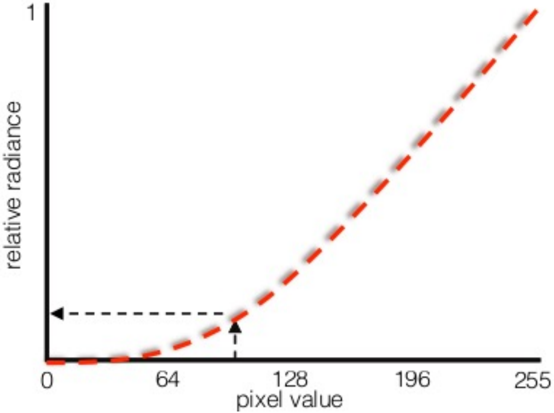
Different scene flux, same camera exposure



Same scene flux, different camera exposure

Non-linear image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$
Exposure time: t_i



$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$

$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$

How would you merge the non-linear images into an HDR one?

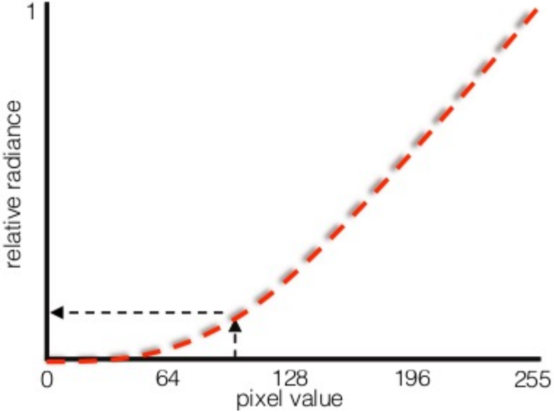
Non-linear image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$
Exposure time: t_i

$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$

$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$

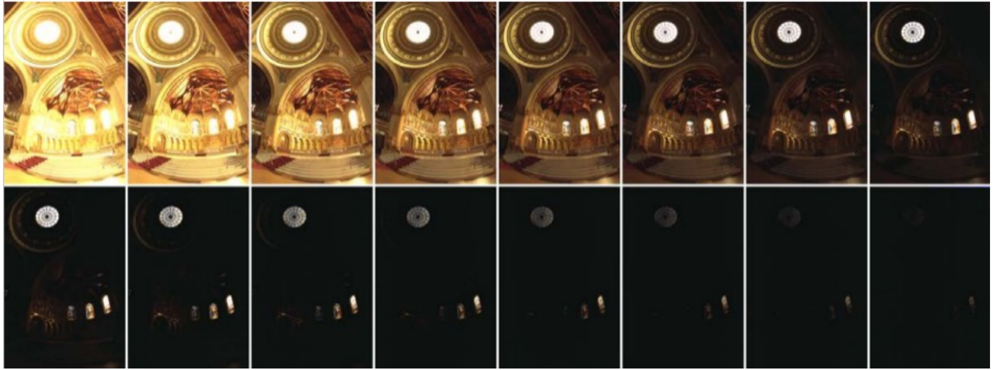
$I_{est}(x,y) = f^{-1}[\ I_{non\text{-}linear}(x,y)\ ]$



Use inverse transform to estimate linear image, then proceed as before

Linearization

$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$



$I_{est}(x,y) = f^{-1}[\ I_{non\text{-}linear}(x,y)\ ]$

Merging non-linear exposure stacks

1. Calibrate response curve

2. Linearize images

For each pixel:

3. Form a new pixel value as the weighted least squares solution

Same as in the RAW case.

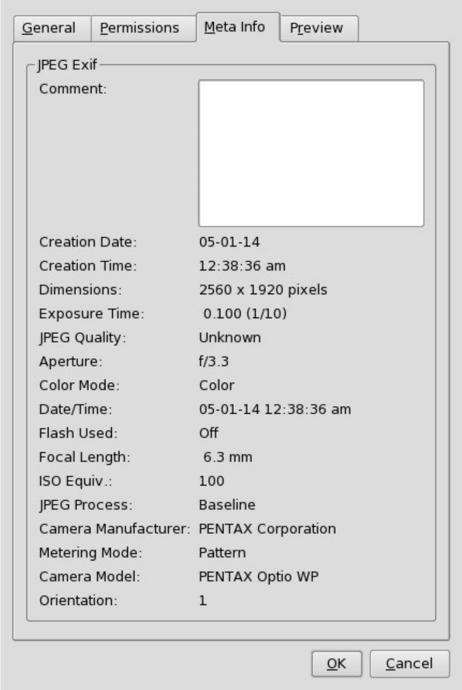
$$\hat{X} = \exp\left(\frac{\sum_i w_i \left(\log\left(I_{lin_i} \right) - \log\left(t_i \right) \right)}{\sum_i w_i} \right)$$

What if I cannot measure the response curve?

You may find information in the image itself

If you cannot do calibration, take a look at the image’s EXIF data (if available).

Often contains information about tone reproduction curve and color space.



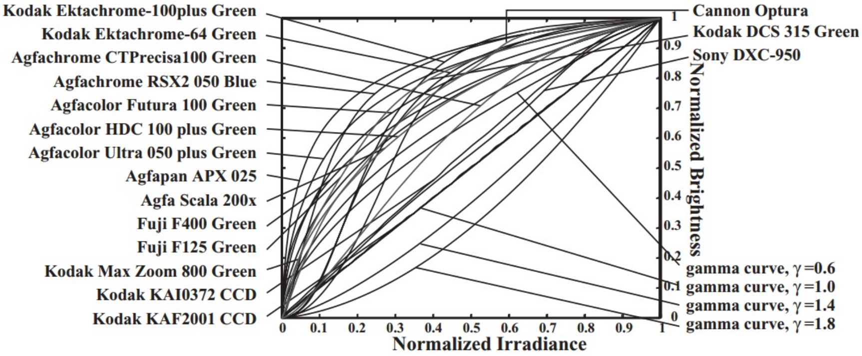
Tone reproduction curves

The exact tone reproduction curve depends on the camera.

- Often well approximated as L^{γ}, for different values of the power γ ("gamma").
- A good default is $\gamma = 1 / 2.2$.

before gamma

after gamma



If nothing else, take the square of your image to approximately remove effect of tone reproduction curve.

Other aspects of HDR imaging

Relative vs absolute flux

Final fused HDR image gives flux only up to a global scale

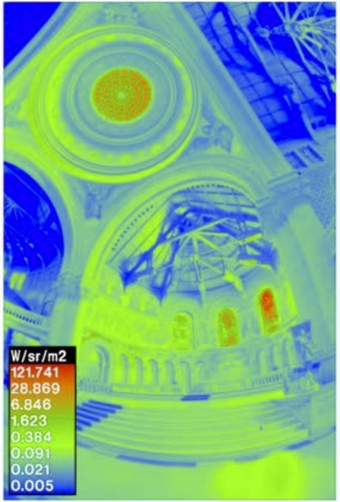
- If we know exact flux at one point, we can convert relative HDR image to absolute flux map

HDR image (relative flux)

→

spotmeter (absolute flux at one point)

→



absolute flux map

Basic HDR approach

1. Capture multiple LDR images at different exposures
2. Merge them into a single HDR image

Any problems with this approach?

Basic HDR approach

1. Capture multiple LDR images at different exposures
2. Merge them into a single HDR image

Problem: Very sensitive to movement

- Scene must be completely static
- Camera must not move

Most modern automatic HDR solutions include an alignment step before merging exposures

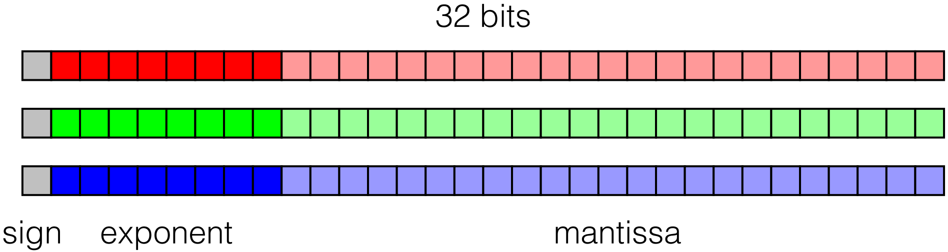
How do we store HDR images?

- Most standard image formats store integer 8-bit images
- Some image formats store integer 12-bit or 16-bit images
- HDR images are floating point 32-bit or 64-bit images

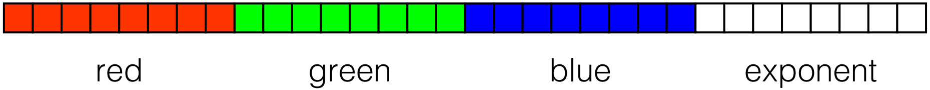
How do we store HDR images?

Use specialized image formats for HDR images

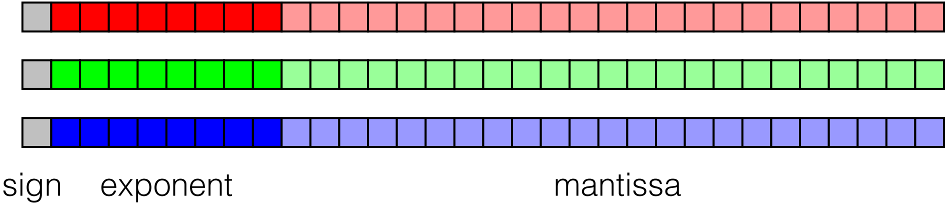
portable float map (.pfm)
- very simple to implement



Radiance format (.hdr)
- supported by Matlab



OpenEXR format (.exr)
- multiple extra features



Another type of HDR images

Light probes: place a chrome sphere in the scene and capture an HDR image

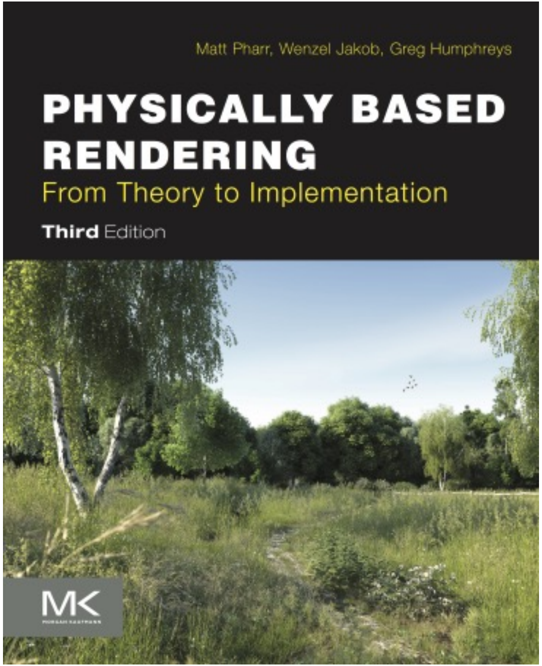
- Used to measure real-world illumination environments (“environment maps”)

Application: image-based relighting

Another way to create HDR images

Physics-based renderers simulate flux maps (relative or absolute)

- Their outputs are very often HDR images



Our devices do not match the real world

- 10:1 photographic print (higher for glossy paper)
- 20:1 artist's paints
- 200:1 slide film
- 500:1 negative film
- 1000:1 LCD display
- 2000:1 digital SLR (at 12 bits)
- 100000:1 real world

HDR imaging and tonemapping are distinct techniques with different goals

HDR imaging compensates for sensor limitations

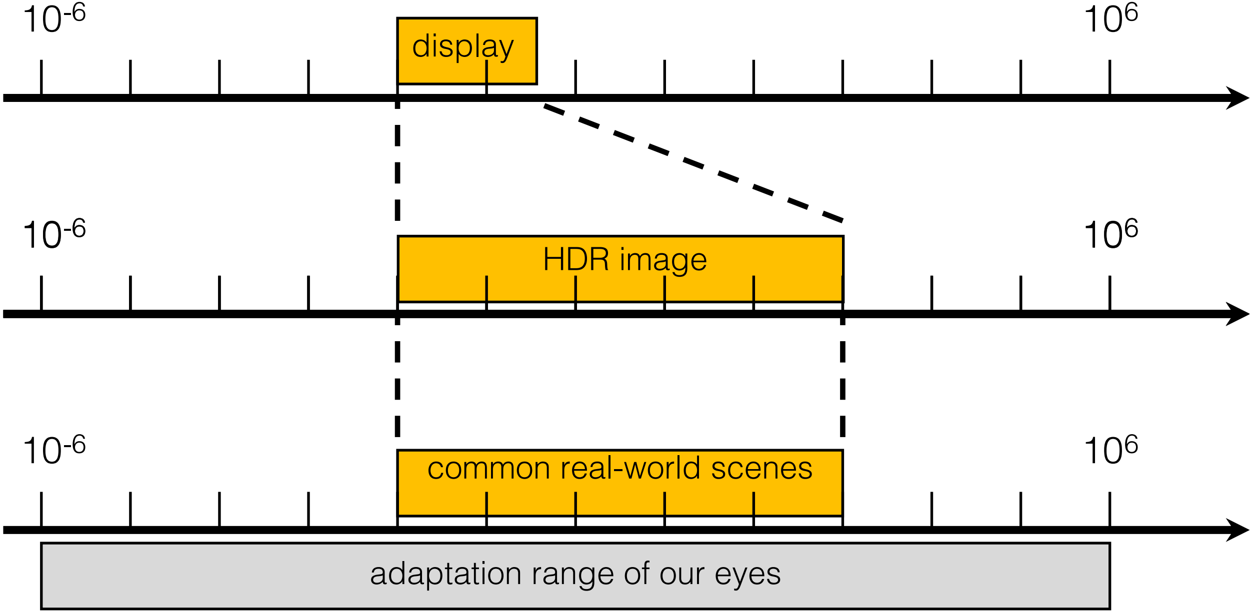
Two challenges:

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

Tonemapping compensates for display limitations

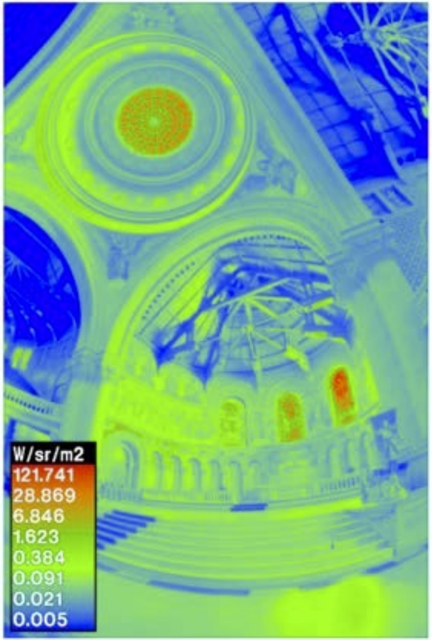
Tonemapping

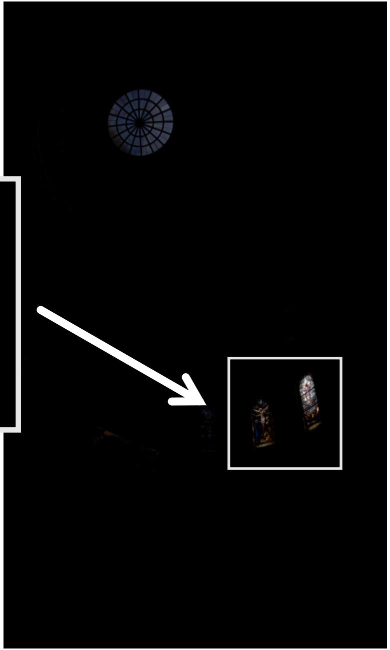
How do we display our HDR images?



Linear scaling

Scale image so that maximum value equals 1.

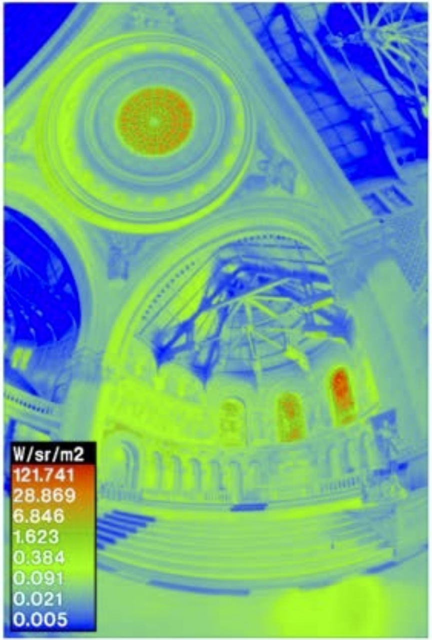




HDR image looks underexposed because of the display's limited dynamic range, but is not actually underexposed.

Linear scaling

Scale image so that 10% value equals 1.



→

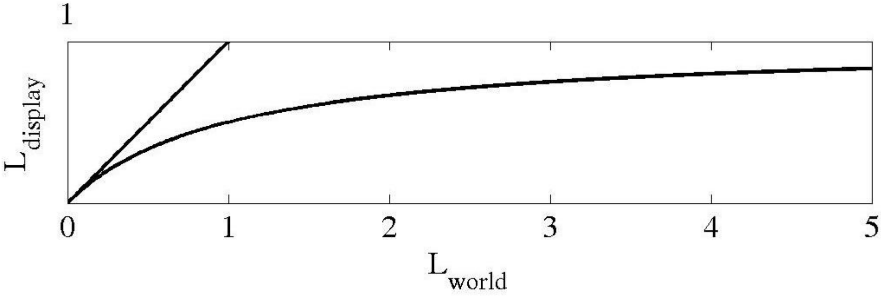
HDR image looks saturated because of the display's limited dynamic range, but is not actually saturated.

Can you think of something better?

Photographic tonemapping

Apply the same non-linear scaling to all pixels in the image so that:

- Bring everything within range → asymptote to 1
- Leave dark areas alone → slope = 1 near 0



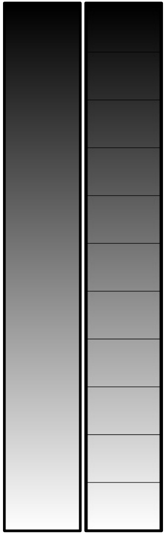
$$I_{display} = \frac{I_{HDR}}{1 + I_{HDR}}$$

(exact formula more complicated)

- Perceptually motivated, as it approximates our eye's response curve.

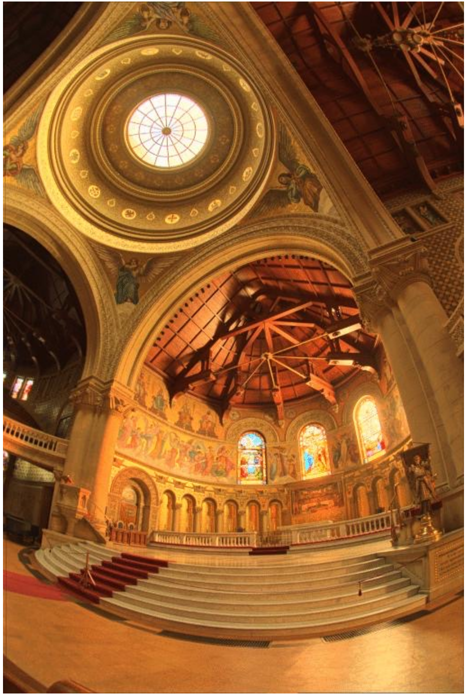
What is the zone system?

- Technique formulated by Ansel Adams for film development.
- Still used with digital photography.

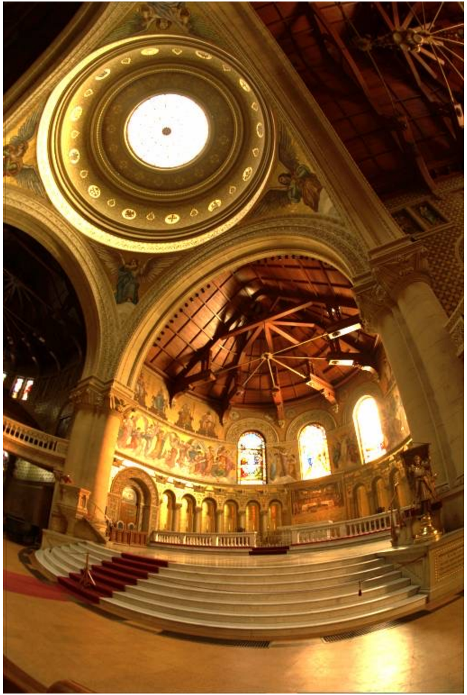


Zone	Description
0	Pure black
I	Near black, with slight tonality but no texture
II	Textured black; the darkest part of the image in which slight detail is recorded
III	Average dark materials and low values showing adequate texture
IV	Average dark foliage, dark stone, or landscape shadows
V	Middle gray: clear north sky; dark skin, average weathered wood
VI	Average Caucasian skin; light stone; shadows on snow in sunlit landscapes
VII	Very light skin; shadows in snow with acute side lighting
VIII	Lightest tone with texture: textured snow
IX	Slight tone without texture; glaring snow
X	Pure white: light sources and specular reflections

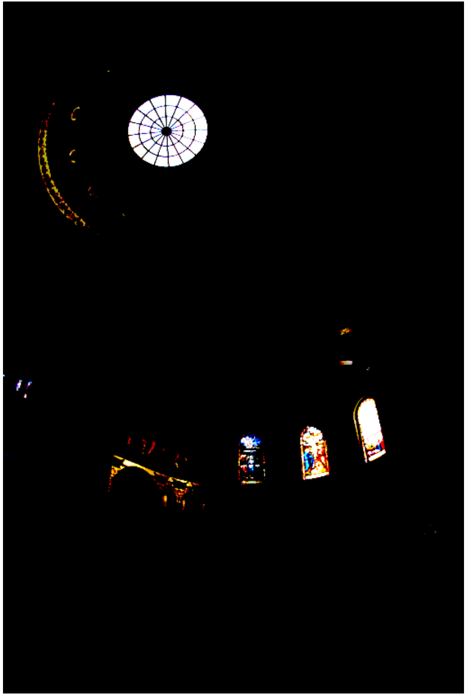
Examples



photographic tonemapping

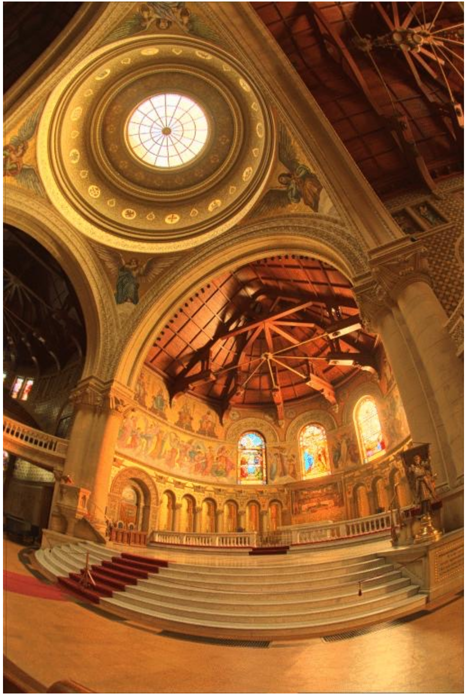


linear scaling (map 10% to 1)

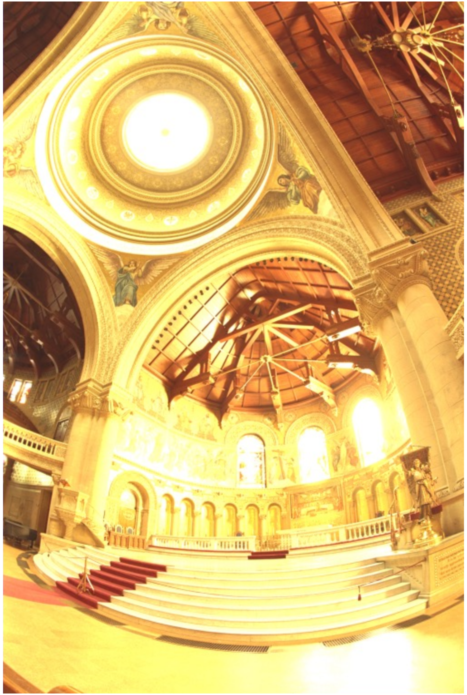


linear scaling (map 100% to 1)

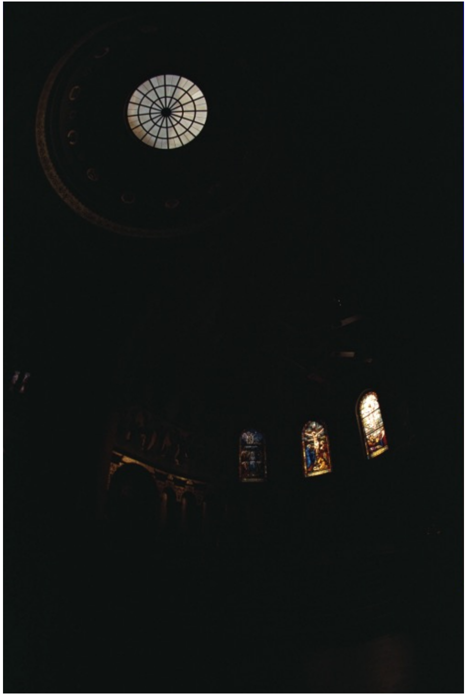
Compare with LDR images



photographic tonemapping



high exposure



low exposure

Dealing with color

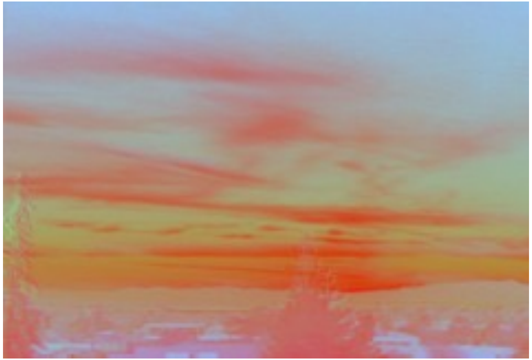
If we tonemap all channels the same, colors are washed out

Can you think of a way to deal with this?

Intensity-only tonemapping

tonemap intensity (e.g., luminance Y in xyY)

leave color the same (e.g., xy in xyY)



How would you implement this?

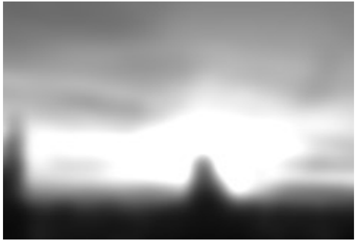
Comparison

Color now OK, but some details are washed out due to loss of contrast

Can you think of a way to deal with this?

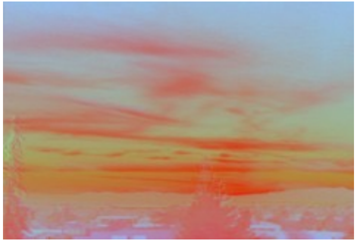
Low-frequency intensity-only tonemapping

tonemap low-frequency intensity component



leave high-frequency intensity component the same

leave color the same



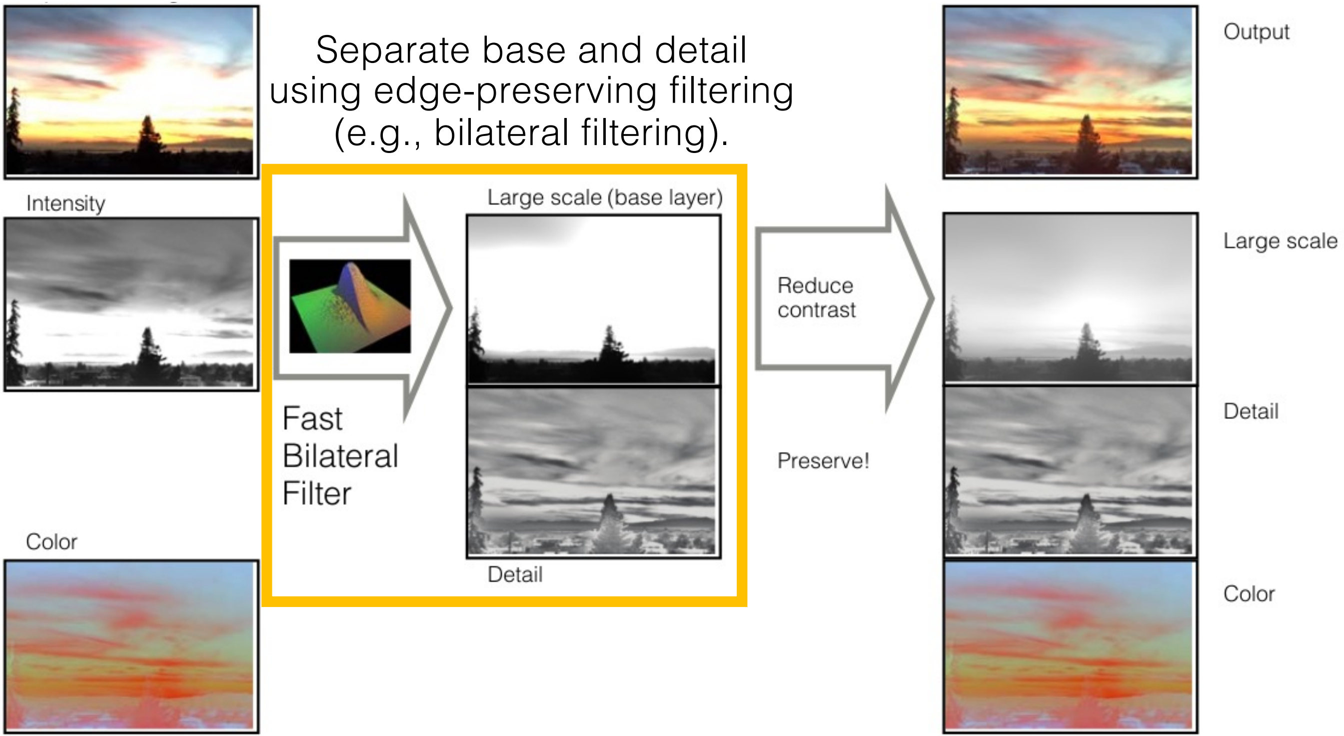
How would you implement this?

Comparison

We got nice color and contrast, but now we've run into the halo plague

Can you think of a way to deal with this?

Edge-aware filtering and tonemapping

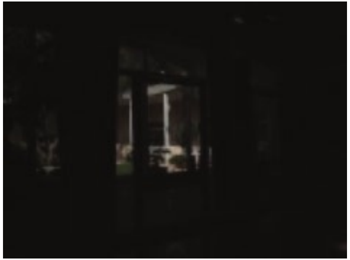


Comparison

We fixed the halos without losing contrast

Gradient-domain processing and tonemapping

Compute gradients, scale and merge them, then integrate (solve Poisson problem).








Tone Mapping w/ Local Laplacian Filters

- Many many more and more complicated tone mapping algorithms out there (too many to discuss here)
- Local Laplacian Filters is one of the state-of-the-art approaches

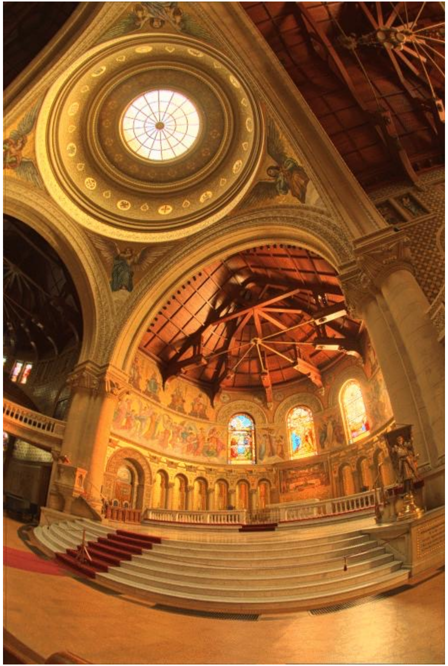
(a) input HDR image tone-mapped with a simple gamma curve (details are compressed)

(b) our pyramid-based tone mapping, set to preserve details without increasing them

(c) our pyramid-based tone mapping, set to strongly enhance the contrast of details

[Paris et al., 2011]

Comparison (which one do you like better?)



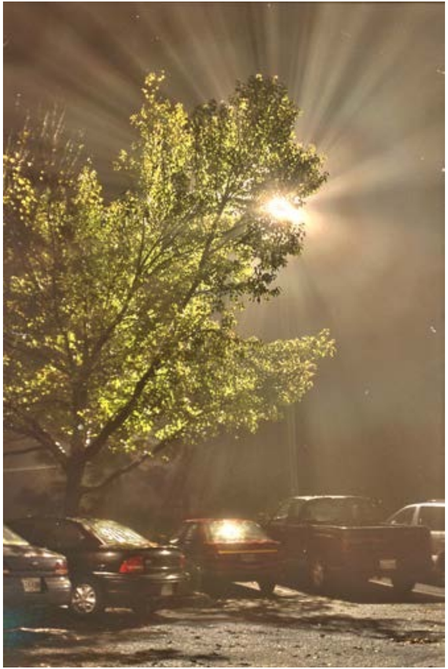
photographic

bilateral filtering

gradient-domain

Comparison (which one do you like better?)

photographic



bilateral filtering

gradient-domain

Comparison (which one do you like better?)



There is no ground-truth: which one looks better is entirely subjective



photographic | bilateral filtering | gradient-domain

Some notes about HDR imaging and tonemapping

A note about terminology

“High-dynamic-range imaging” is used to refer to a lot of different things:

1. Using single RAW images.
2. Performing radiometric calibration.
3. Merging an exposure stack.
4. Tonemapping an image (linear or non-linear, HDR or LDR).
5. Some or all of the above.

A note about terminology

"High-dynamic-range imaging" is used to refer to a lot of different things:

1. Using single RAW images.
2. Performing radiometric calibration.
3. Merging an exposure stack.
4. Tonemapping an image (linear or non-linear, HDR or LDR).
5. Some or all of the above.

Technically, HDR imaging and tonemapping are distinct processes:

- HDR imaging is the process of creating a radiometrically linear image, free of overexposure and underexposure artifacts. This is achieved using some combination of 1-3, depending on the imaging scenario.
- Tonemapping (step 4) process of mapping the intensity values in an image (linear or non-linear, HDR or LDR) to the range of tones available in a display.

A note about terminology

"High-dynamic-range imaging" is used to refer to a lot of different things:

1. Using single RAW images.
2. Performing radiometric calibration.
3. Merging an exposure stack.
4. Tonemapping an image (linear or non-linear, HDR or LDR).
5. Some or all of the above.

Technically, HDR imaging and tonemapping are distinct processes:

- HDR imaging is the process of creating a radiometrically linear image, free of overexposure and underexposure artifacts. This is achieved using some combination of 1-3, depending on the imaging scenario.
- Tonemapping (step 4) process of mapping the intensity values in an image (linear or non-linear, HDR or LDR) to the range of tones available in a display.

But:

- In consumer photography, "HDR photography" is often used to refer to both HDR imaging (steps 1-3) and tonemapping (step 4).

A note of caution

- HDR photography can produce very visually compelling results.

A note of caution

- HDR photography can produce very visually compelling results.
- It is also a very routinely abused technique, resulting in awful results.

A note of caution

- HDR photography can produce very visually compelling results.
- It is also a very routinely abused technique, resulting in awful results.
- The problem typically is tonemapping, not HDR imaging itself.

A note about HDR today

- Most cameras (even phone cameras) have automatic HDR modes/apps.
- Popular-enough feature that phone manufacturers are actively competing about which one has the best HDR.
- The technology behind some of those apps (e.g., Google's HDR+) is published in SIGGRAPH and SIGGRAPH Asia conferences.

Burst photography for high dynamic range and low-light imaging on mobile cameras

Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, Marc Levoy
Google Research

Figure 1: *A comparison of a conventional camera pipeline (left, middle) and our burst photography pipeline (right) running on the same cell-phone camera. In this low-light setting (about 0.7 lux), the conventional camera pipeline underexposes (left). Brightening the image (middle) reveals heavy spatial denoising, which results in loss of detail and an unpleasantly blotchy appearance. Fusing a burst of images increases the signal-to-noise ratio, making aggressive spatial denoising unnecessary. We encourage the reader to zoom in. While our pipeline excels in low-light and high-dynamic-range scenes (for an example of the latter see figure 10), it is computationally efficient and reliably artifact-free, so it can be deployed on a mobile camera and used as a substitute for the conventional pipeline in almost all circumstances. For readability the figure has been made uniformly brighter than the original photographs.*

Abstract

Cell phone cameras have small apertures, which limits the number of photons they can gather, leading to noisy images in low light. They also have small sensor pixels, which limits the number of electrons each pixel can store, leading to limited dynamic range. We describe a computational photography pipeline that captures, aligns, and merges a burst of frames to reduce noise and increase dynamic range. Our system has several key features that help make it robust and efficient. First, we do not use bracketed exposures. Instead, we capture frames of constant exposure, which makes alignment more robust, and we set this exposure low enough to avoid blowing out highlights. The resulting merged image has clean shadows and high bit depth, allowing us to apply standard HDR tone mapping methods. Second, we begin from Bayer raw frames rather than the demosaicked RGB (or YUV) frames produced by hardware Image Signal Processors (ISPs) common on mobile platforms. This gives us more bits per pixel and allows us to circumvent the ISP's unwanted tone mapping and spatial denoising. Third, we use a novel FFT-based alignment algorithm and a hybrid 2D/3D Wiener filter to denoise and merge the frames in a burst. Our implementation is built atop Android's Camera2 API, which provides per-frame camera control and access to raw imagery, and is written in the Halide domain-specific language (DSL). It runs in 4 seconds on device (for a 12 Mpix image), requires no user intervention, and ships on several mass-produced cell phones.

Keywords: computational photography, high dynamic range

Concepts: **•Computing methodologies → Computational photography; Image processing;**

1 Introduction

The main technical impediment to better photographs is lack of light. In indoor or night-time shots, the scene as a whole may provide insufficient light. The standard solution is either to apply analog or digital gain, which amplifies noise, or to lengthen exposure time, which causes motion blur due to camera shake or subject motion. Surprisingly, daytime shots with high dynamic range may also suffer from lack of light. In particular, if exposure time is reduced to avoid

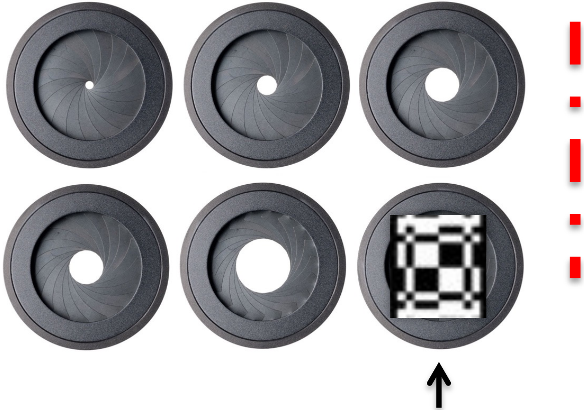
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.
SA '16 Technical Papers, December 05 - 08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2980254

Coded (Aperture) Computational Imaging

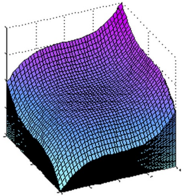
Camera Aperture Revisited

A camera aperture has (at least) two parts that can be "coded":

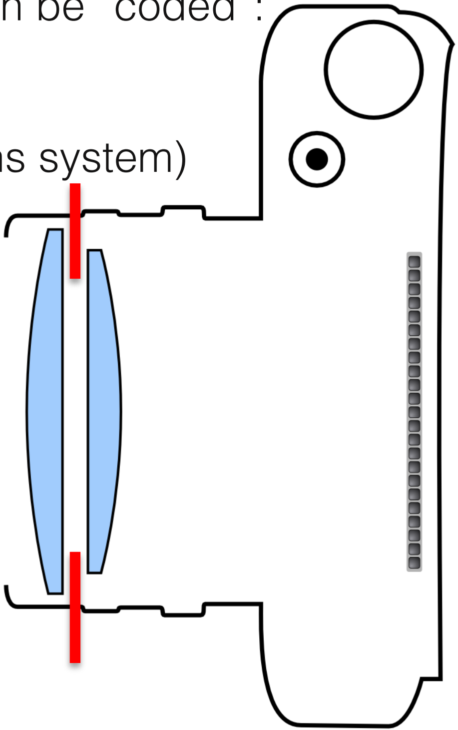
1. aperture stop – attenuating pattern
2. refractive elements (lens or compound lens system)



1. attenuating coded aperture

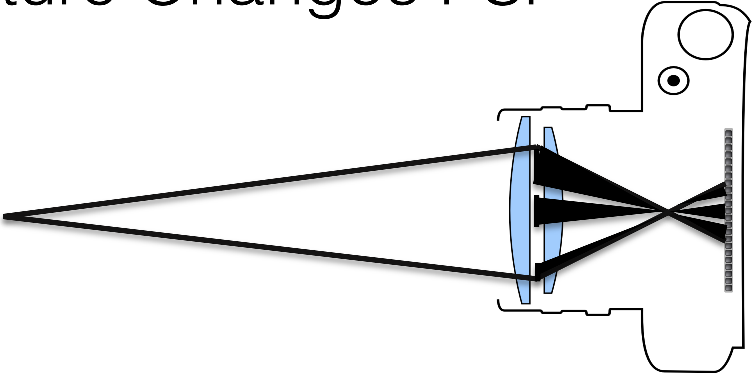


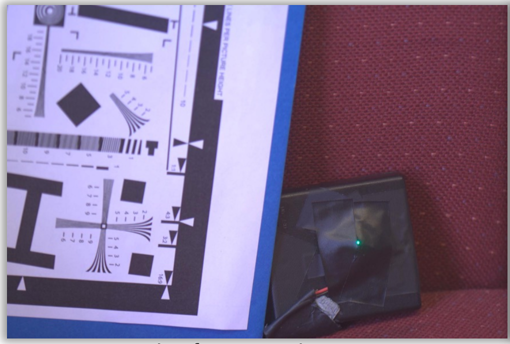
2. refractive or diffractive coded aperture or lens system



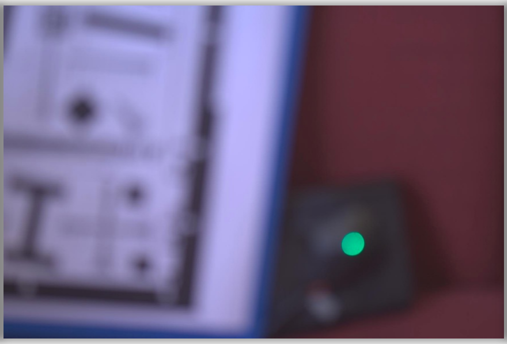
Coded Aperture Changes PSF

[Veeraragharavan et al. 2007]





in-focus photo

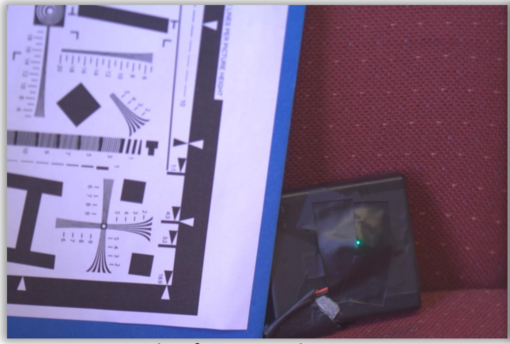


out-of-focus, circular aperture

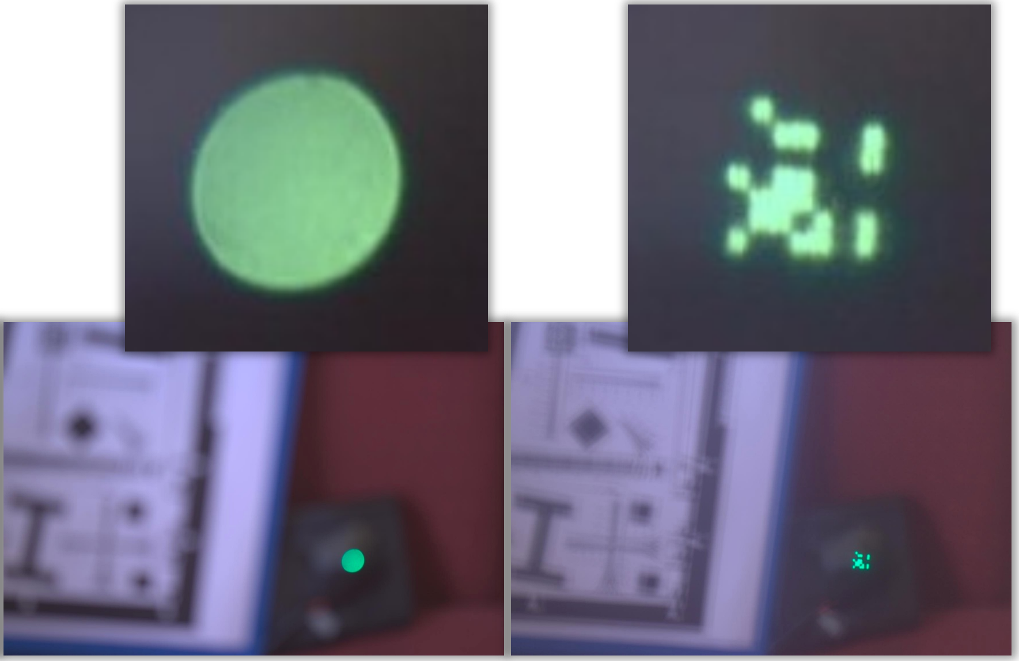
out-of-focus, coded aperture

Coded Aperture Changes PSF

[Veeraraghavan et al. 2007]



in-focus photo



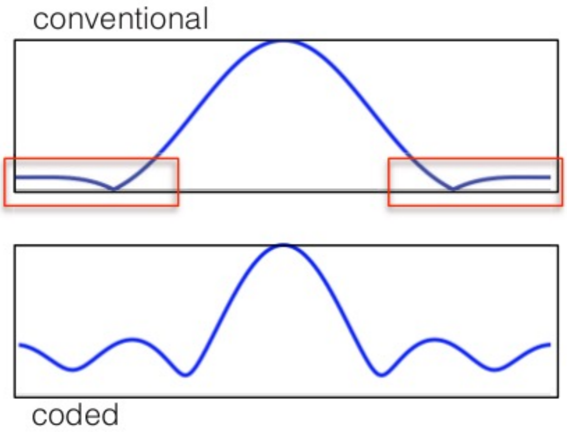
out-of-focus, circular aperture

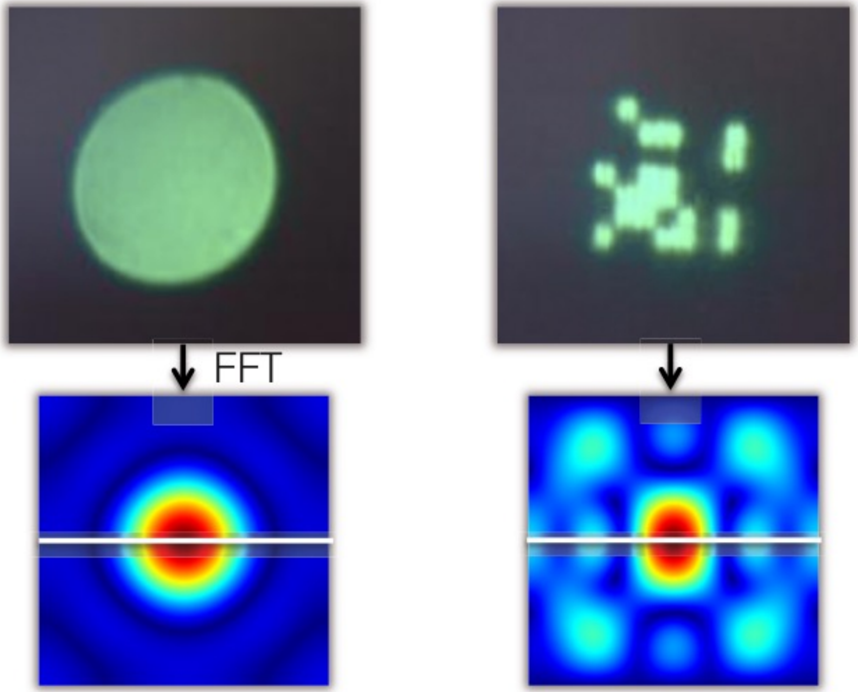
out-of-focus, coded aperture

Coded aperture changes shape of PSF

New PSF preserves high frequencies

- More content available to help us determine correct depth





Coded (Aperture) Imaging

Applications of *Coded Aperture Imaging*:

- Extended depth of field
- Monocular depth estimation

Applications of *Coded Imaging* in General:

- Motion deblurring
- High-speed, hyperspectral, light field, single-pixel imaging …

Coded (Aperture) Imaging

Applications of *Coded Aperture Imaging*:

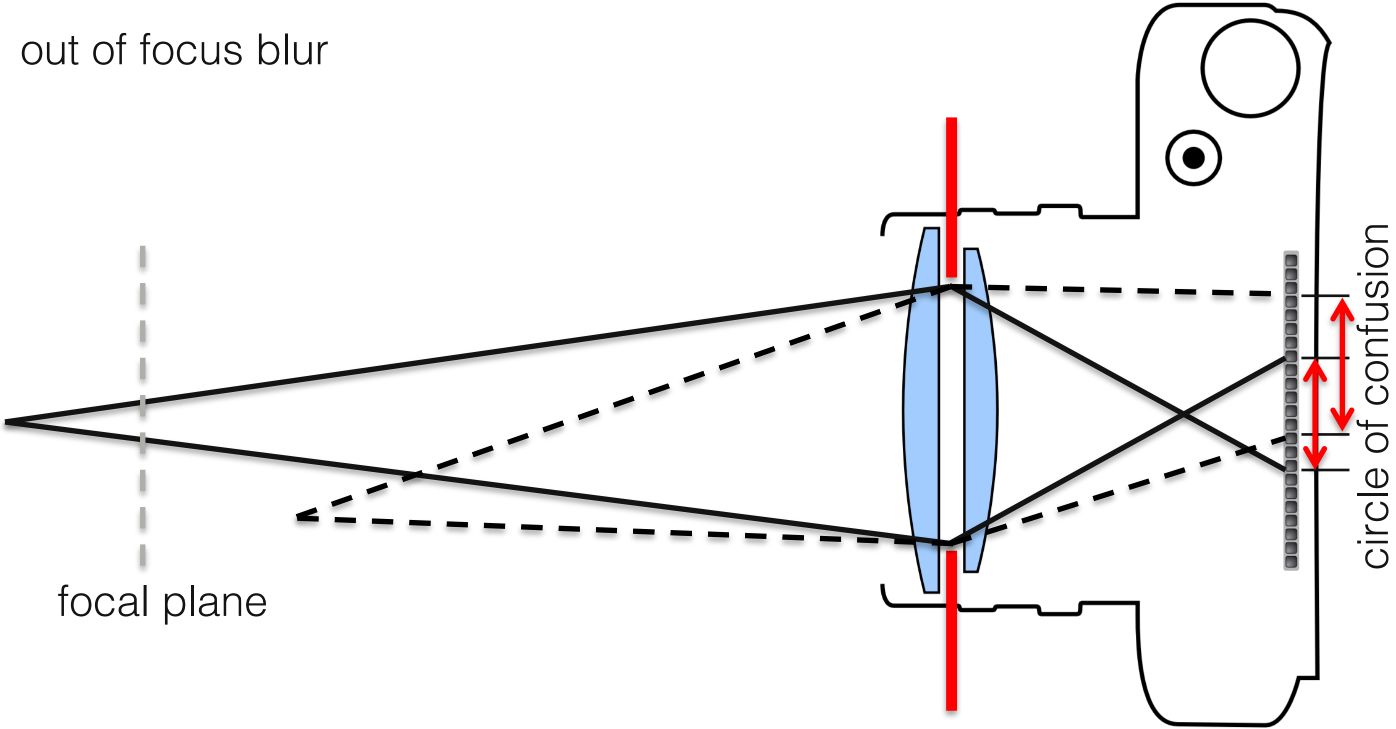
- Extended depth of field
- Monocular depth estimation

Applications of *Coded Imaging* in General:

- Motion deblurring
- High-speed, hyperspectral, light field, single-pixel imaging ...

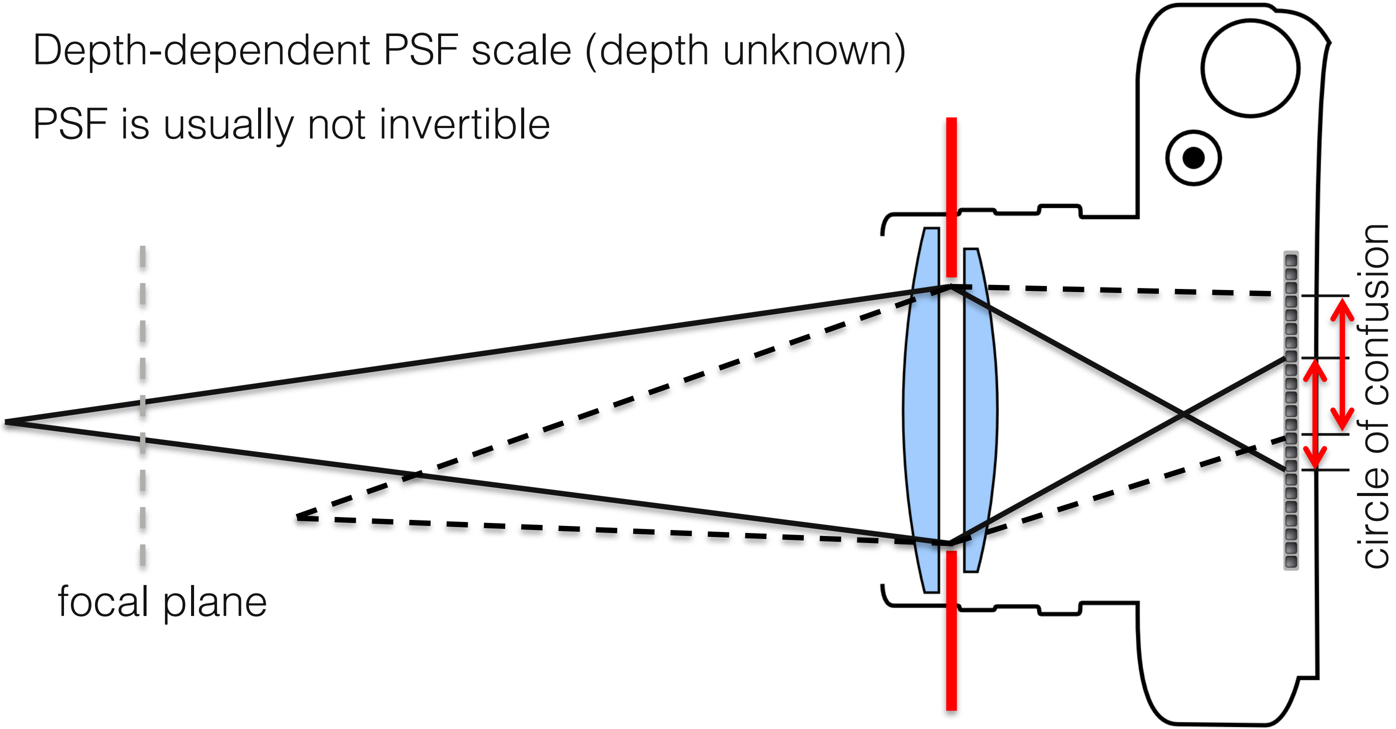
What makes Defocus Deblurring Hard?

- out of focus blur



What makes Defocus Deblurring Hard?

1. Depth-dependent PSF scale (depth unknown)
2. PSF is usually not invertible



Extended Depth of Field

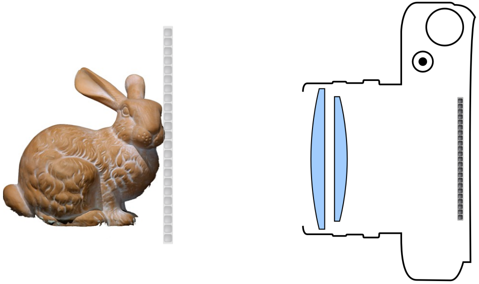
1. Problem: depth-dependent PSF scale (depth unknown)
    - engineer PSF to be depth invariant
    - resulting shift-invariant deconvolution is much easier!

2. Problem: circular / Airy PSF is usually not invertible: ill-posed problem
    - engineer PSF to be broadband (flat Fourier magnitudes)
    - resulting inverse problem becomes well-posed

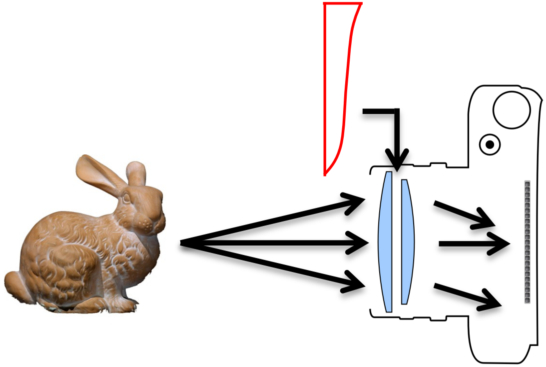
Extended Depth of Field

- Two general approaches for engineering depth-invariant PSFs:

1. move sensor / object
(known as focal sweep)



2. change optics
(e.g., wavefront coding)



Extended Depth of Field – Focal Sweep

captured focal sweep
always blurry!

EDOF image

conventional photo
(small DOF)

conventional photo
(large DOF, noisy)

[Nagahara et al. 2008]

Extended Depth of Field – Focal Sweep

- noise characteristics are main benefit of EDOF

- may change for different sensor noise characteristics

SNR should be evaluation metric!

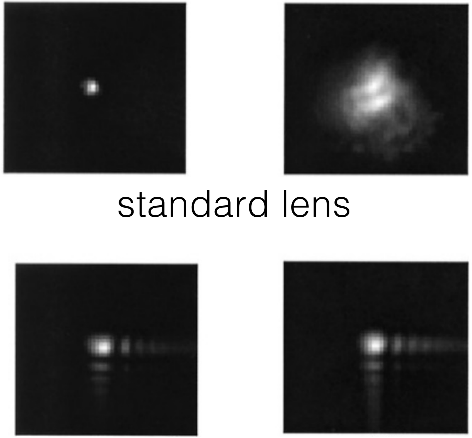
EDOF image

conventional photo (large DOF, noisy)

[Nagahara et al. 2008]

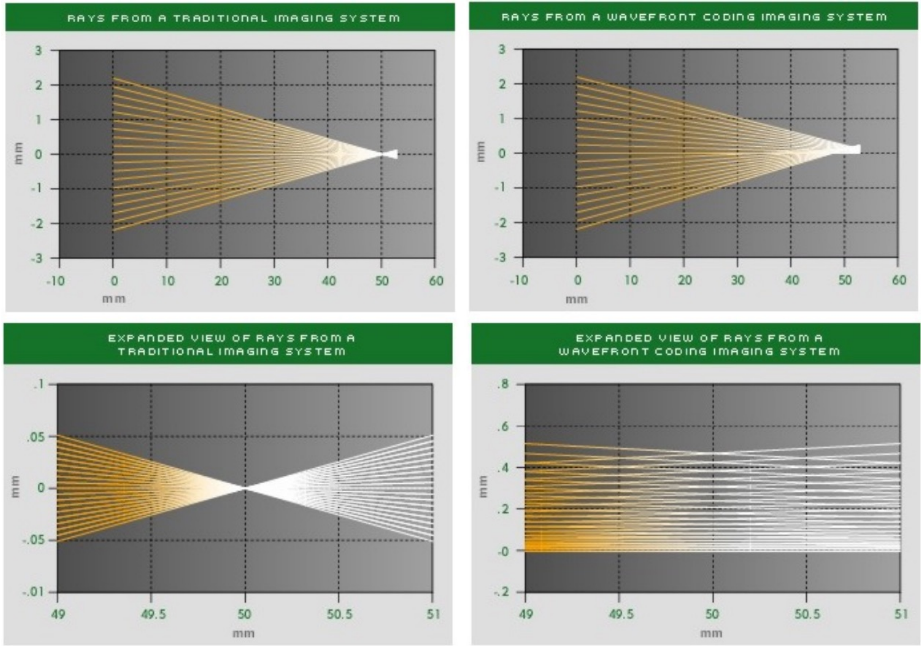
Wavefront coding

In focus Out of focus



standard lens

wavefront coding



- Rays no longer converge.
- Approximately depth-invariant PSF for certain range of depths.

Coded (Aperture) Imaging

Applications of *Coded Aperture Imaging*:

- Extended depth of field
- Monocular depth estimation

Applications of *Coded Imaging* in General:

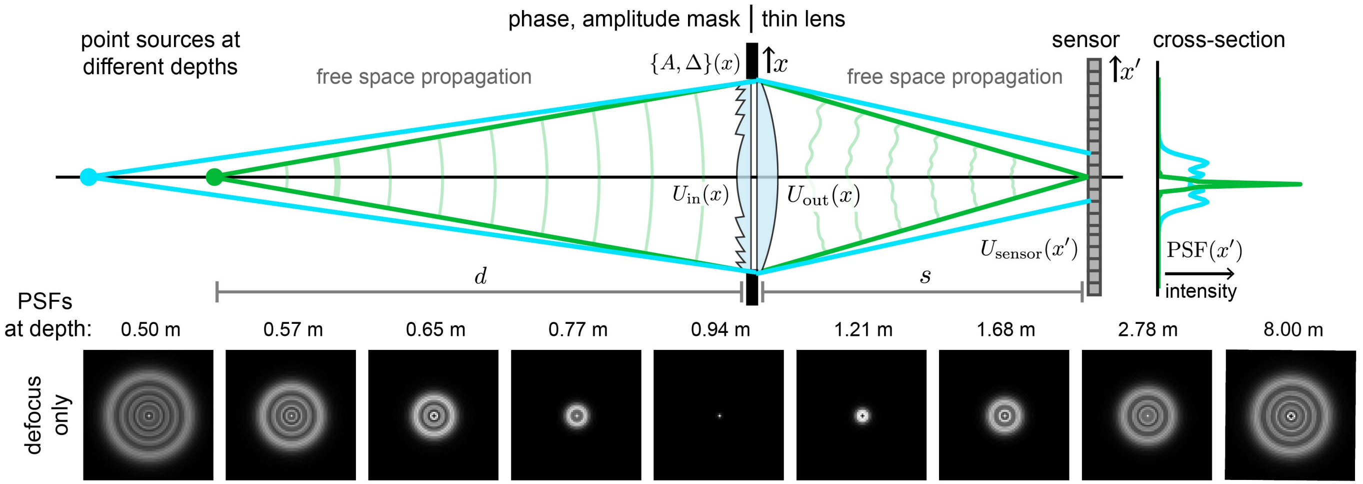
- Motion deblurring
- High-speed, hyperspectral, light field, single-pixel imaging …

Monocular Depth Estimation

- Problem: 3D/depth cameras are hard
- Solution: a single image contains a lot of depth cues – learn to use them for depth estimation (like humans)

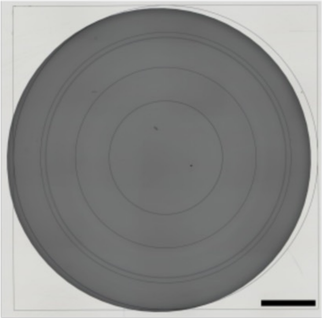
[Godard et al., 2017]

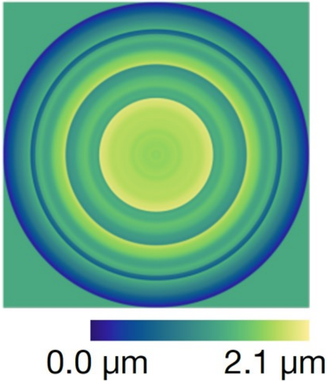
Coded Apertures for Depth Estimation



[Chang and Wetzstein, 2019]

Coded Apertures for Depth Estimation

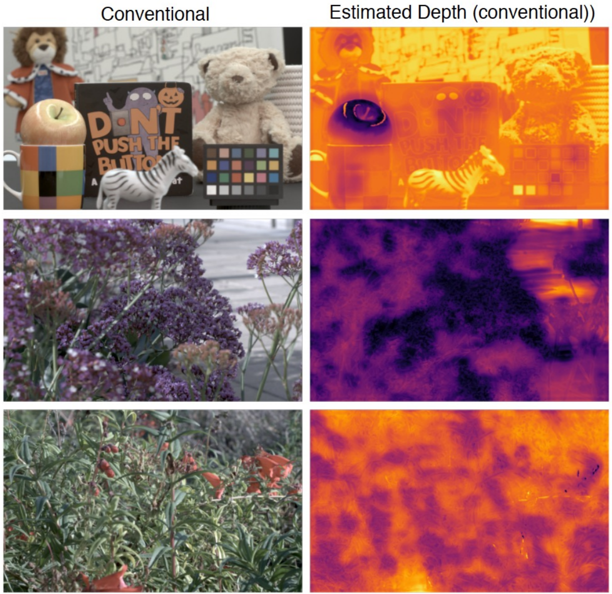




[Ikoma et al., 2021]

Coded Apertures for Depth Estimation

- PSF engineering can make depth estimation more robust by encoding low-level depth information in the PSF (rather than just pictorial cues)



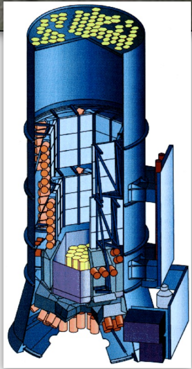
[Ikoma et al., 2021]

Coded Apertures in Astronomy

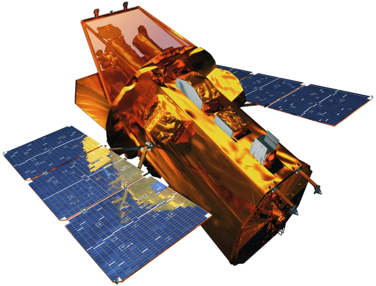
- some wavelengths are difficult to focus

→ no "lenses" available

- coded apertures for x-rays and gamma rays



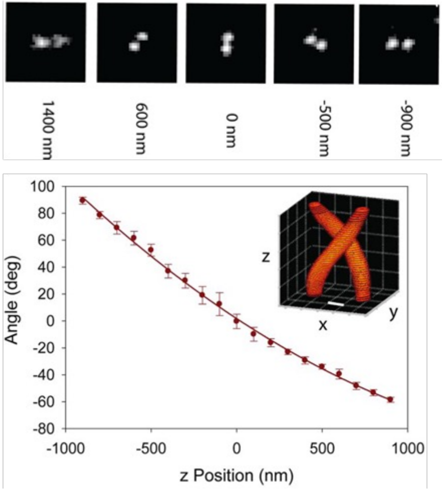
ESA SPI / INTEGRAL



NASA Swift

Coded Apertures in Microscopy

- for low-light, coding of refraction is better (less light loss)



e.g., rotating double helix PSF
Stanford Moerner lab

Coded (Aperture) Imaging

Applications of *Coded Aperture Imaging*:

- Extended depth of field
- Monocular depth estimation

Applications of *Coded Imaging* in General:

- Motion deblurring
- High-speed, hyperspectral, light field, single-pixel imaging ...

Motion Blur and Deblurring

- Problem: objects that move throughout exposure time will be blurred
- Motion deblurring is hard because:
  1. Motion PSF may be unknown and different for different object
  2. Motion PSF is difficult to invert

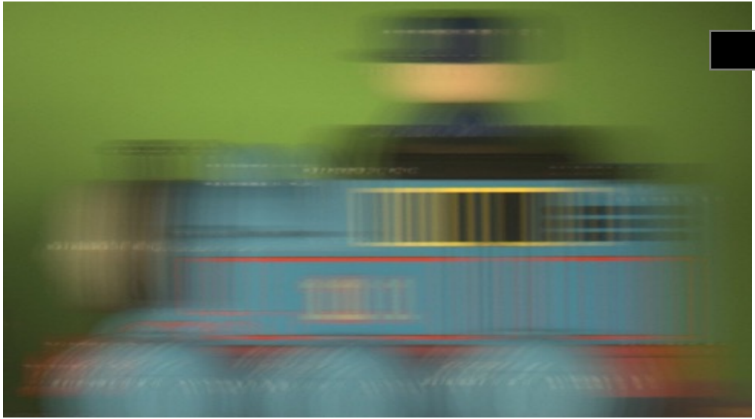
Blurred input image

Deblurred image

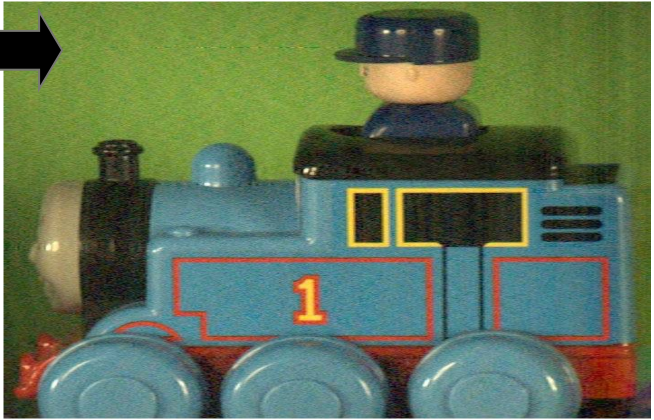
[Shan et al. 2008]

Motion Deblurring w/ Flutter Shutter

- engineer motion PSF (coding exposure time) so it becomes invertible!



Input Photo

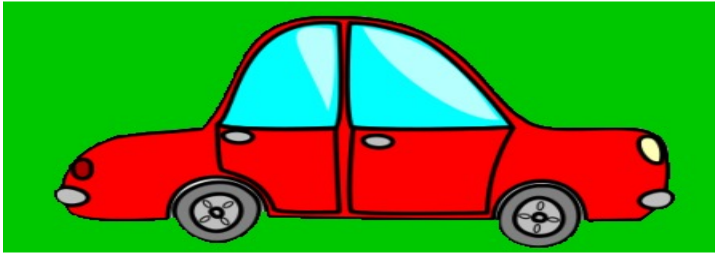


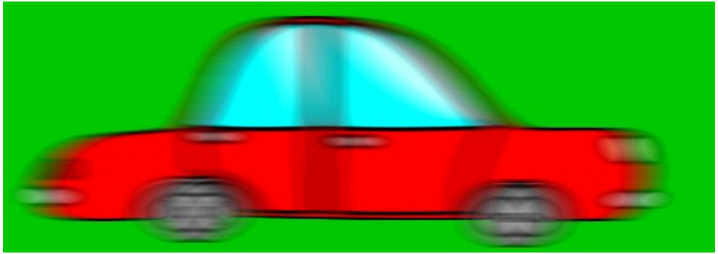
Deblurred Result

[Raskar et al. 2006]

Traditional Camera:

Shutter is OPEN

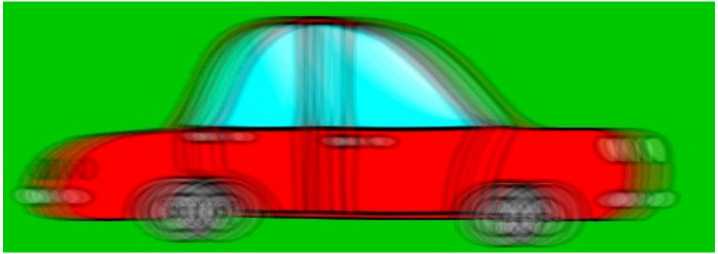




[Raskar et al. 2006]

Flutter Shutter Camera:

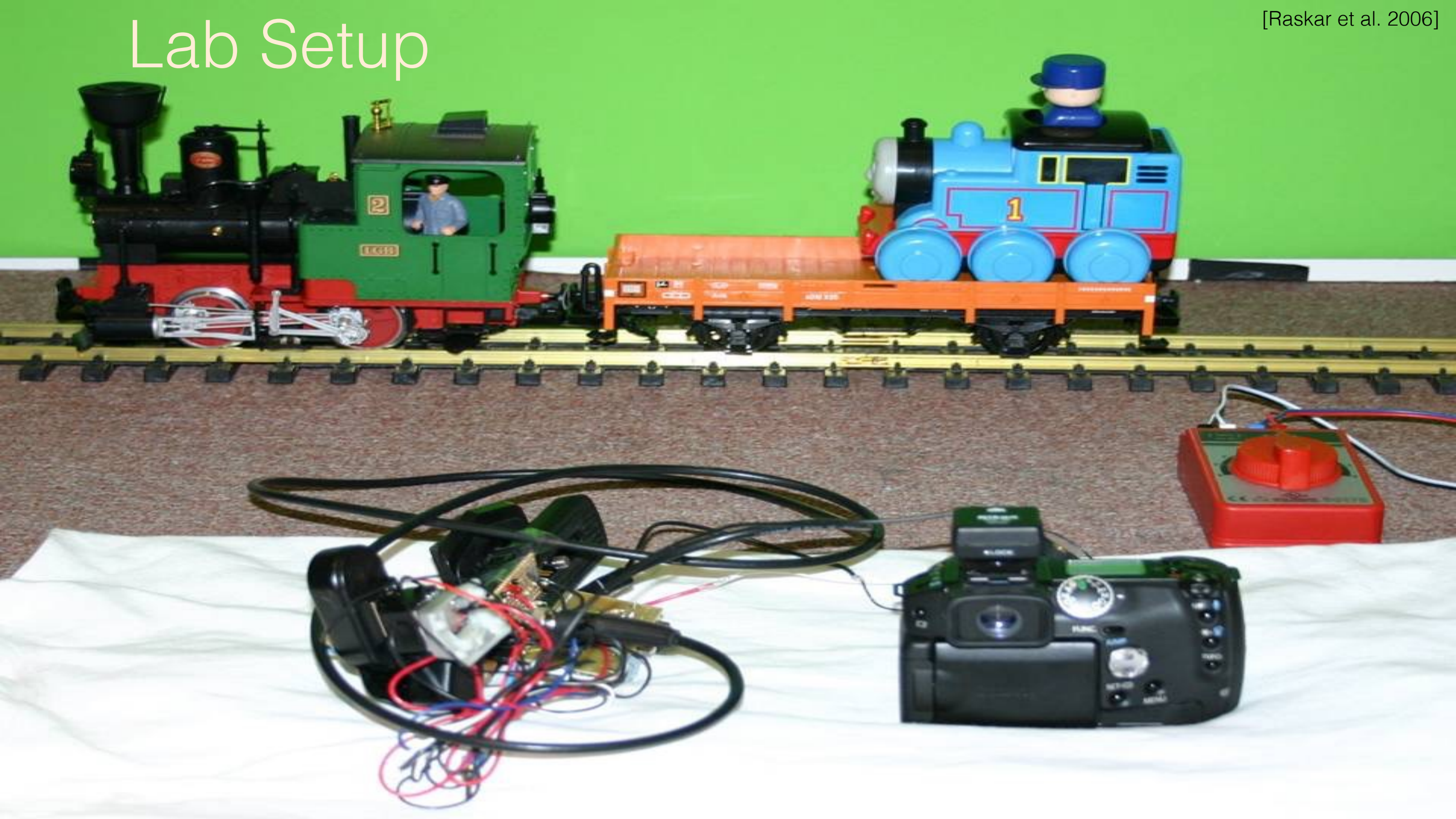
Shutter is OPEN & CLOSED



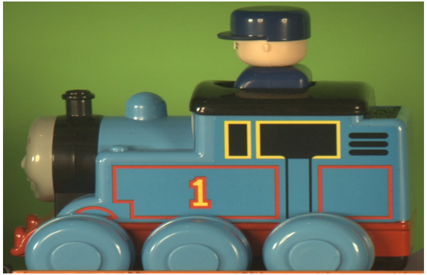
[Raskar et al. 2006]

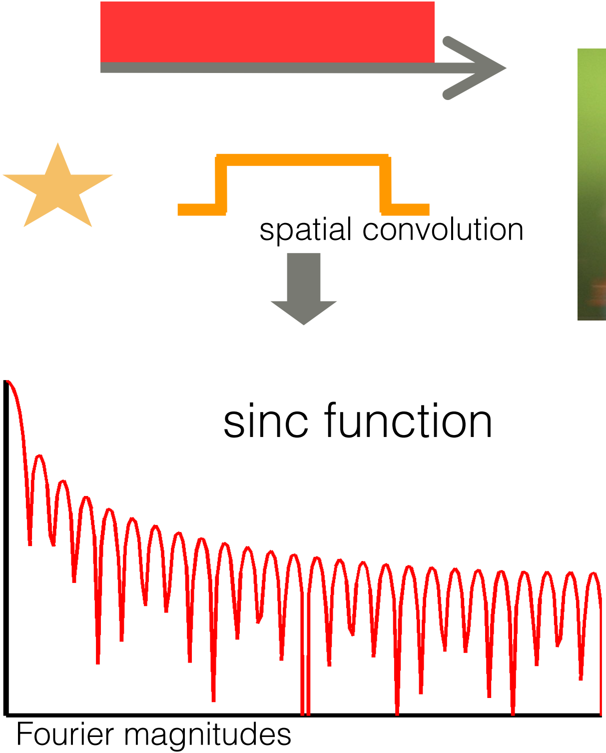
Lab Setup

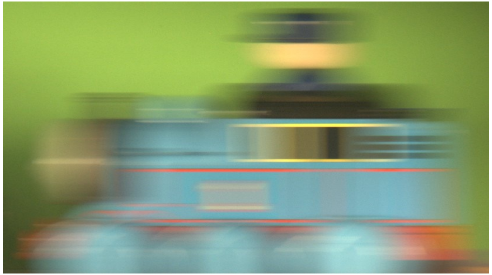
[Raskar et al. 2006]



[Raskar et al. 2006]







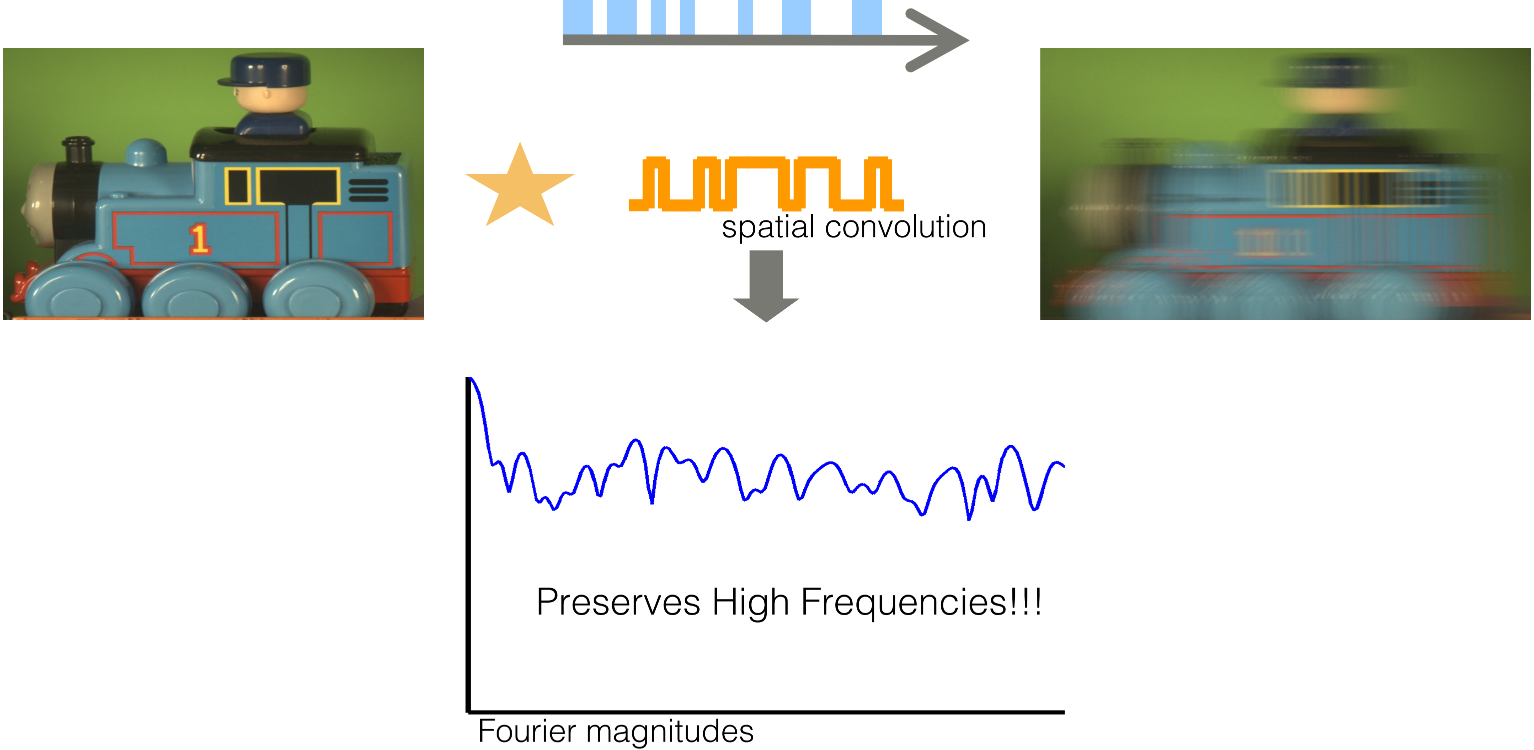
Blurring

=

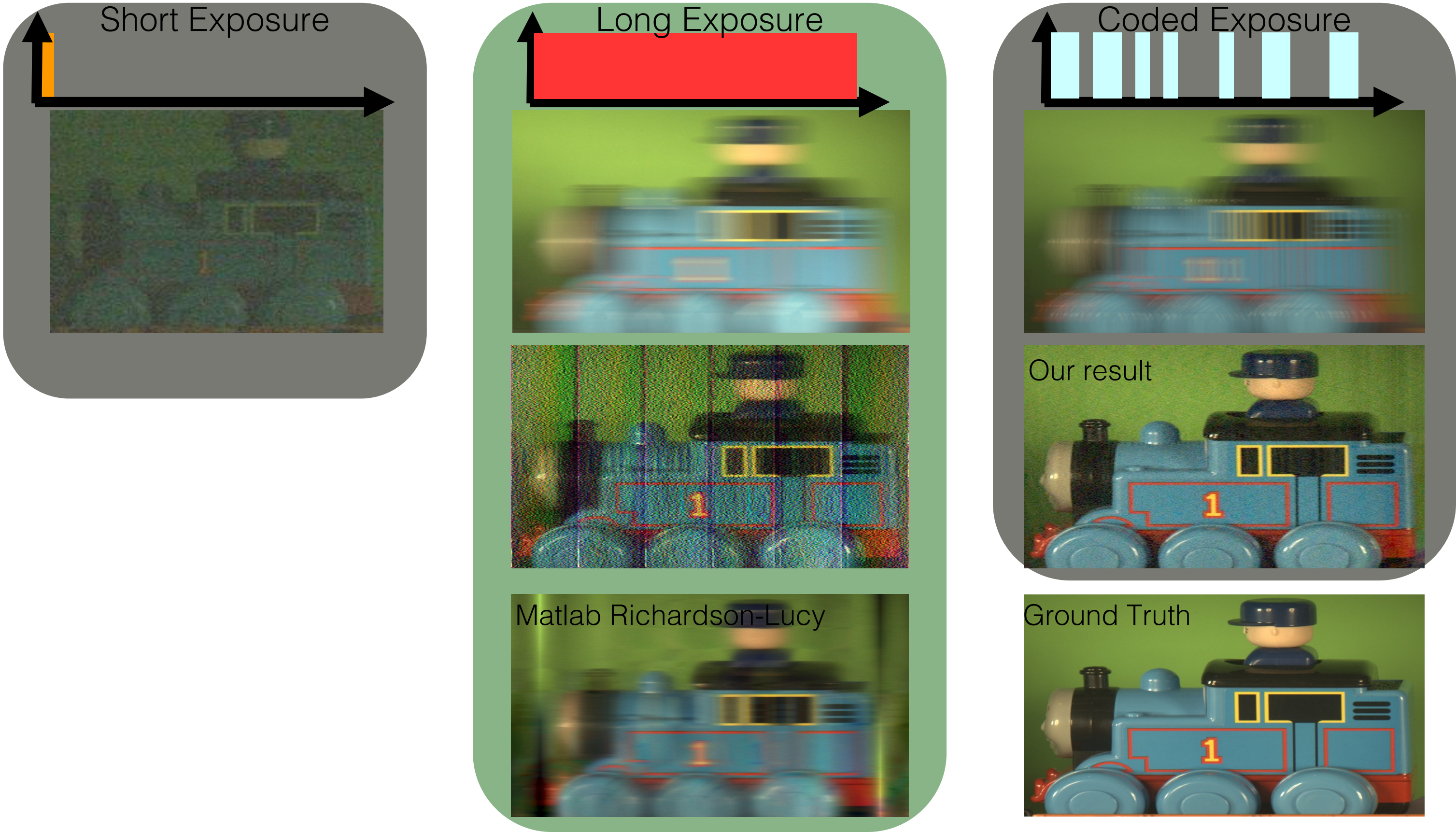
Convolution

Traditional Camera: Box Filter

[Raskar et al. 2006]



Flutter Shutter: Coded Filter



Short Exposure
Long Exposure
Coded Exposure
Our result
Matlab Richardson-Lucy
Ground Truth

[Raskar et al. 2006]

License Plate Retrieval

[Raskar et al. 2006]

License Plate Retrieval

parabolic sweep

Motion-invariant photography

Introduce extra motion so that:

- Everything is blurry; and
- The blur kernel is motion invariant (same for all objects).

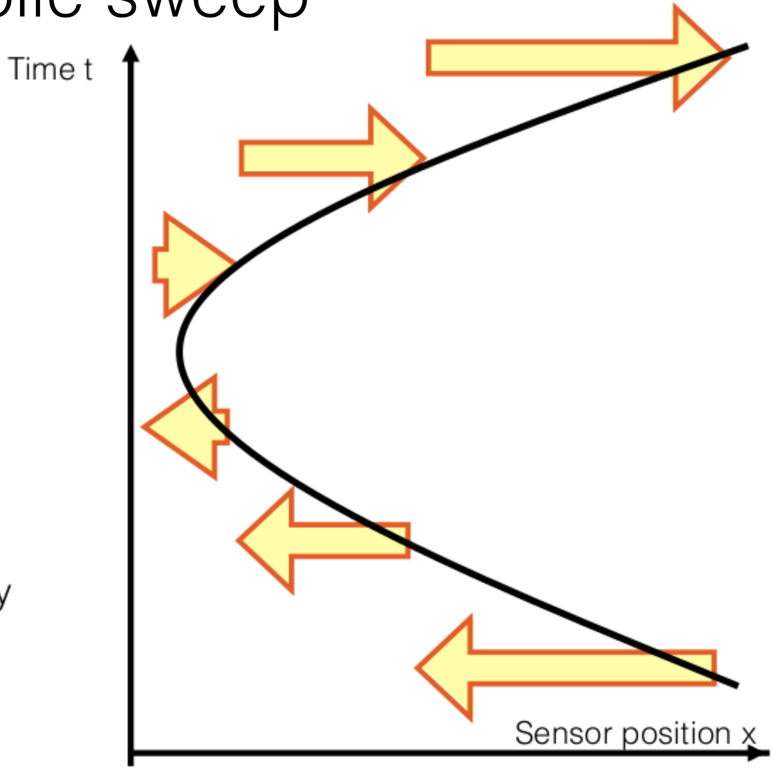
How would you achieve this?

Parabolic sweep

Sensor position $x(t)=a\ t^2$

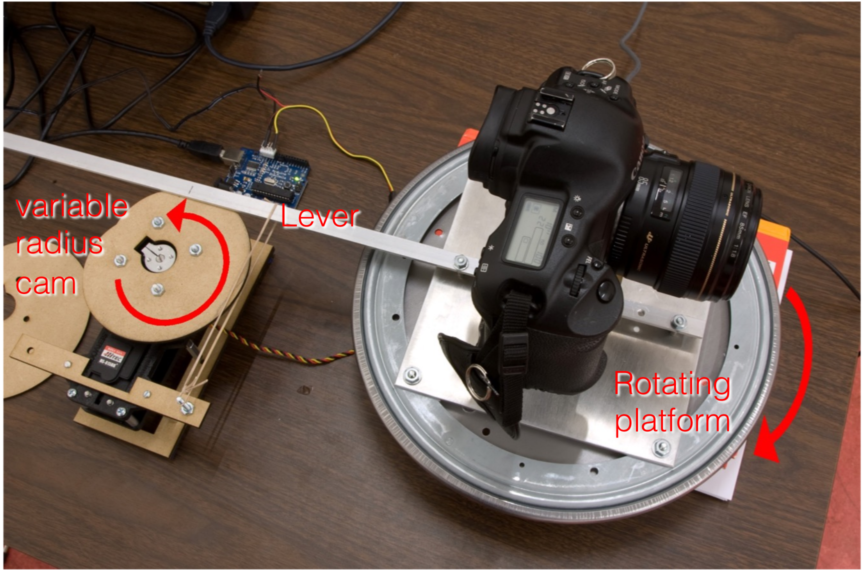
- start by moving very fast to the right
- continuously slow down until stop
- continuously accelerate to the left

- Intuition:
  - for any velocity, there is one instant where we track perfectly
  - all velocities captured same amount of time



Hardware implementation

Approximate small translation by small rotation



Some results

static camera input - unknown and variable blur

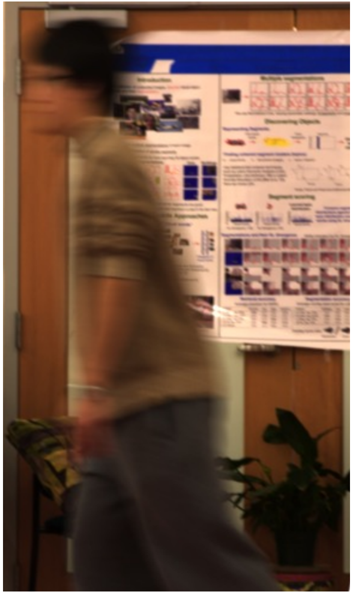
parabolic input - blur is invariant to velocity

Some results

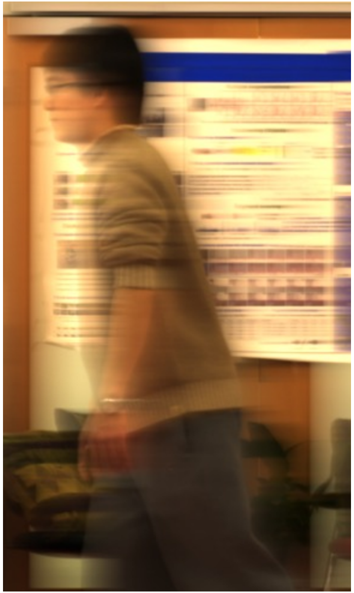
static camera input - unknown and variable blur

output after deconvolution

Some results



static camera input



parabolic camera input

deconvolution output

Some results

static camera input

output after deconvolution

Why does it fail in this case?

Coded (Aperture) Imaging

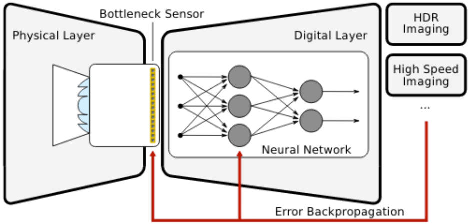
Applications of *Coded Aperture Imaging*:

- Extended depth of field
- Monocular depth estimation

Applications of *Coded Imaging* in General:

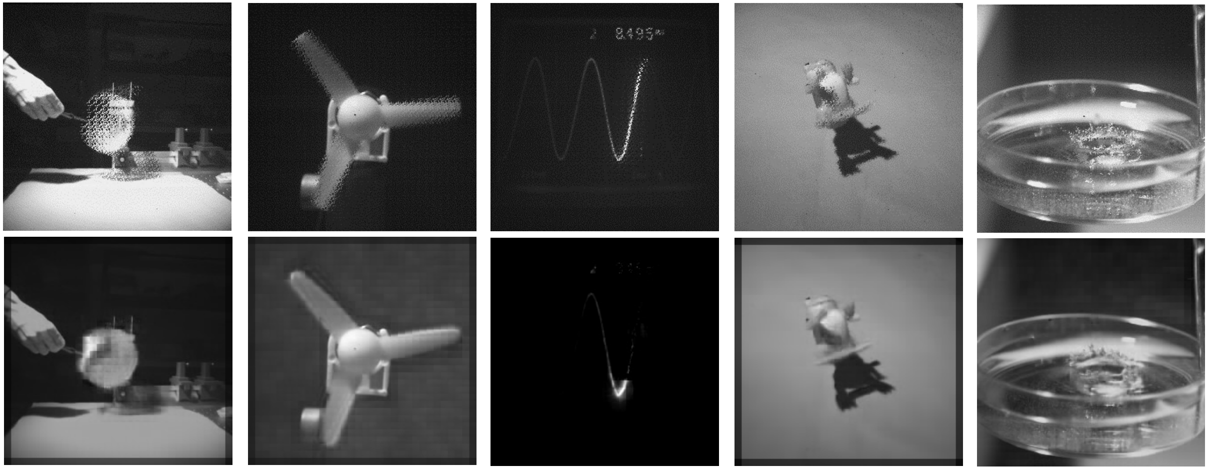
- Motion deblurring
- High-speed, hyperspectral, light field, single-pixel imaging …

Coded Imaging with Neural Sensors



Coded Measurements

Reconstructions



[Martel et al., 2020]

Next time: image processing with neural networks

References and Further Reading

HDR

- Mann, Picard "On Being 'Undigital' with Digital Cameras: Extending Dynamic Range by Combining Differently Exposed Pictures", IS&T 1995
- **Debevec, Malik, "Recovering High Dynamic Range Radiance Maps from Photographs", SIGGRAPH 1997**
- Reinhard, Ward, Pattanaik, Debevec (2005). High dynamic range imaging: acquisition, display, and image-based lighting. Elsevier/Morgan Kaufmann

Tone Mapping

- **Durand, Dorsey, "Fast Bilateral Filtering for the Display of High Dynamic Range Images", ACM SIGGRAPH 2002**
- Paris, Hasinoff, Kautz, "Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid", ACM SIGGRAPH 2011

Burst Photography/Denoising

- Hasinoff, Sharlet, Geiss, Adams, Barron, Kainz, Chen, Levoy "Burst photography for high dynamic range and low-light imaging on mobile cameras", SIGGRAPH Asia 2016
- Liba et al., "Handheld Mobile Photography in Very Low Light", ACM SIGGRAPH Asia 2019

Extended Depth of Field

- E. Dowski, W. Cathey, "Extended depth of field through wave-front coding", Appl. Opt. 34, 11, 1995
- H. Nagahara, S. Kuthirummal, C. Zhou, S. Nayar, "Flexible Depth of Field Photography", ECCV 2008
- Levin, Hasinoff, Green, Durand, Freeman, "4D Frequency Analysis of Computational Cameras for Depth of Field Extension", ACM SIGGRAPH 2009
- O. Cossairt, S. Nayar "Spectral Focal Sweep for Extending Depth of Field", ICCP 2010
- O. Cossairt, C. Zhou, S. Nayar, "Diffusion-Coded Photography", ACM SIGGRAPH 2012

Depth Estimation

- C. Godard, O. Aodha, G. Bostrow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency", CVPR 2017
- J. Chang, G. Wetzstein, "Deep optics for monocular depth estimation and 3d object detection", ICCV 2019
- H. Ikoma, C. Nguyen, C. Metzler, Y. Peng, G. Wetzstein, "Depth from Defocus with Learned Optics for Imaging and Occlusion-aware Depth Estimation", ICCP 2021

Motion Deblurring

- Q. Shan, J. Jia, A. Agrawal, "High-quality Motion Deblurring from a Single Image", ACM SIGGRAPH 2008
- **R. Raskar, A. Agrawal, J. Tumblin "Coded Exposure Photography: Motion Deblurring using Fluttered Shutter", ACM SIGGRAPH 2006**
- Levin, Sand, Cho, Durand, Freeman, "Motion-Invariant Photography", ACM SIGGRAPH 2008
- Bando, Holtzman, Raskar, "Near-Invariant Blur for Depth and 2D Motion via Time-Varying Light Field Analysis", ACM Trans. Graph. 2013

Other

- J. Martel, L. Mueller, S. Carey, P. Dudek, G. Wetzstein, "Neural Sensors: Learning Pixel Exposures with Programmable Sensors", IEEE T. PAMI 2020