Denoising & Bilateral Filtering

Lecture 5

Rob Fergus

Many slides from Fredo Durand & Sylvain Paris

Admin stuff

- Start homework 2 early!!!!

- Have grader for course (Jiali Huang jiali.huang@nyu.edu)

- Come and see me about projects!!!!

Overview of today

- Denoising
  - Averaging
  - Wiener denoising
  - Median filtering
- Bilaterial filtering
  - Cross-bilateral filter
  - Flash applications

Noisy image

- Usually for dark conditions

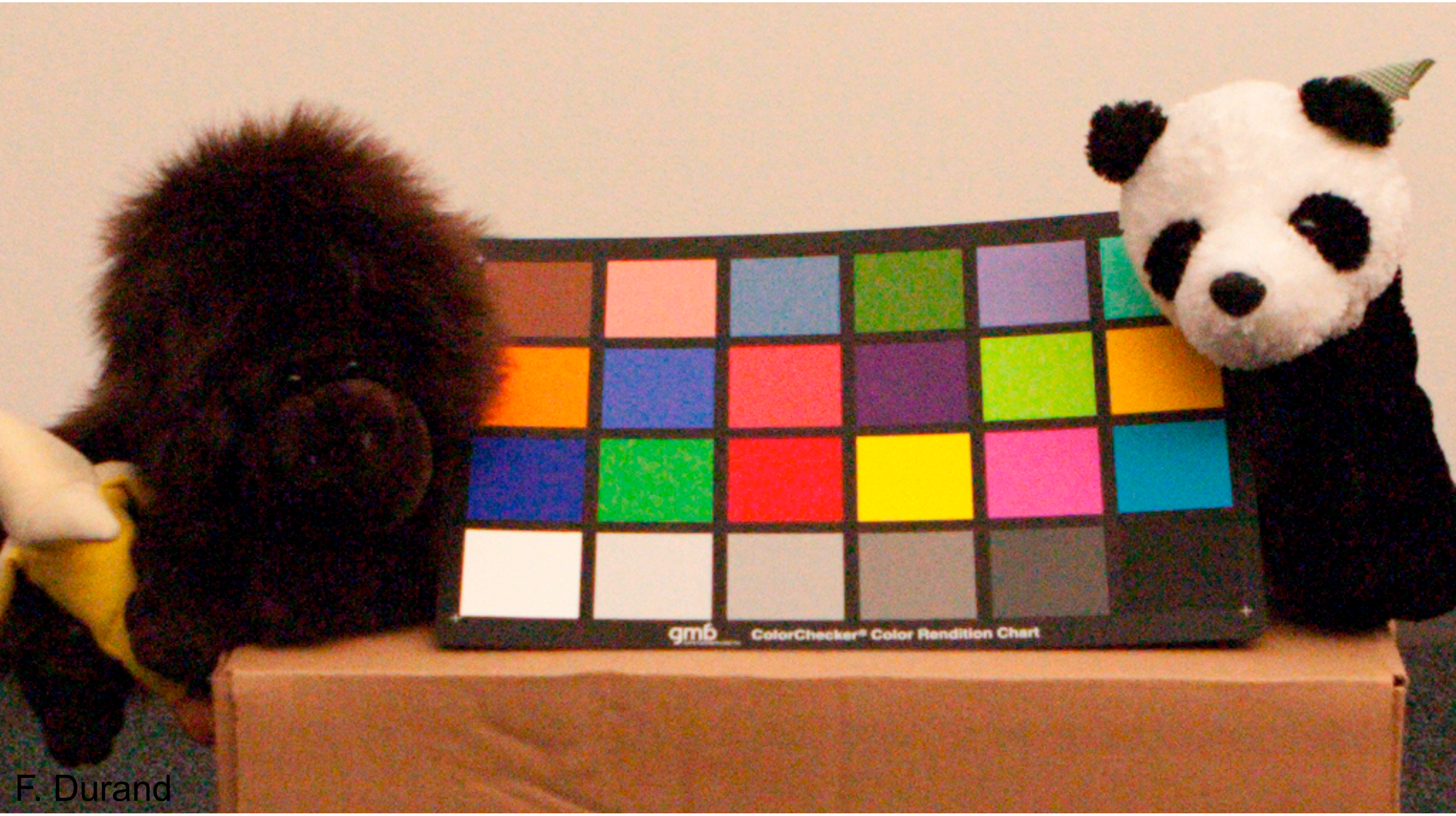
F. Durand

Noise

- Fluctuation when taking multiple shots

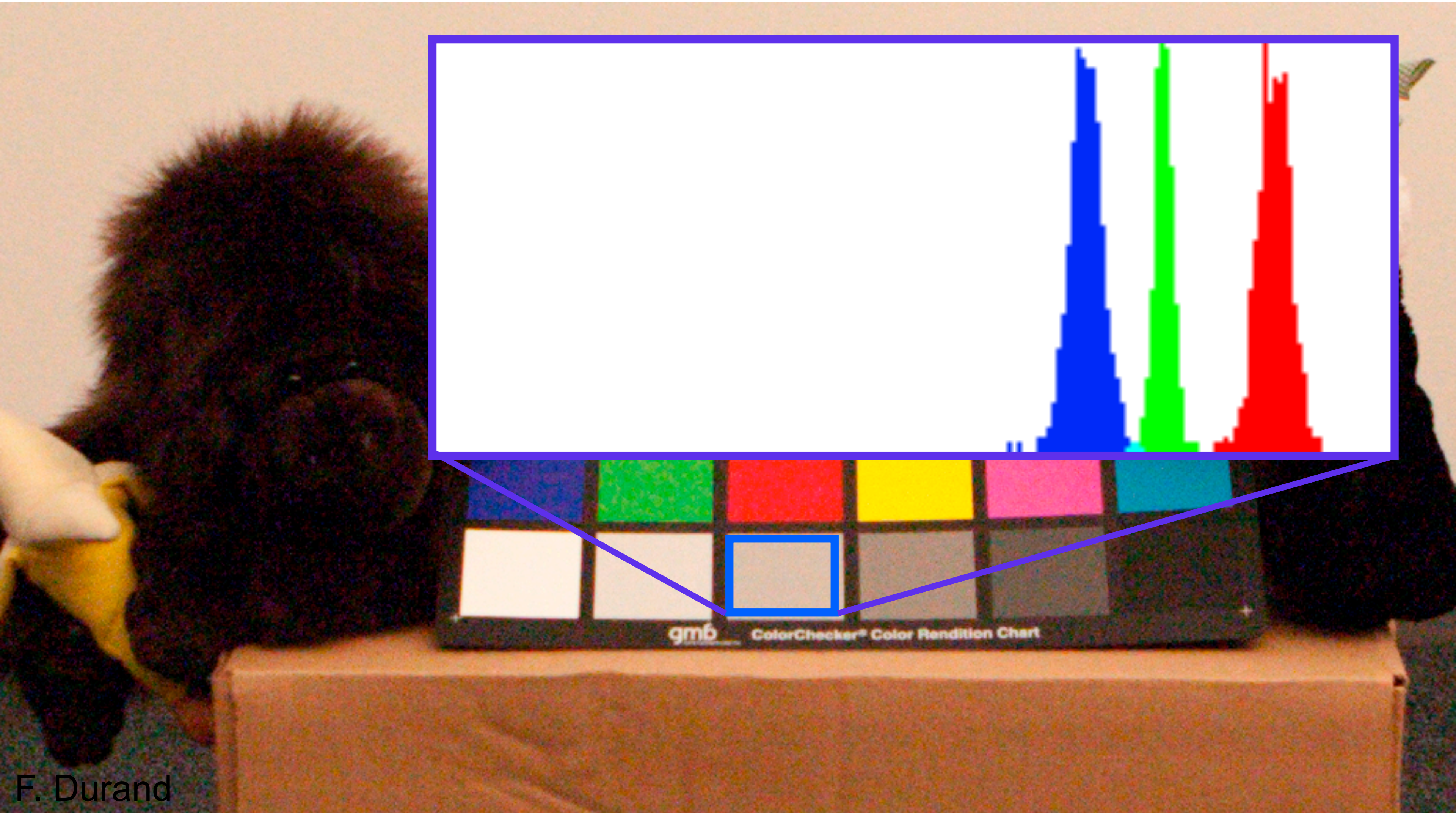
F. Durand

Canon 1D mark IIN at ISO 3200



Histogram of grey patch

- Should be single values for RGB (constant color)



Where Does Noise come from?

Digital pipeline

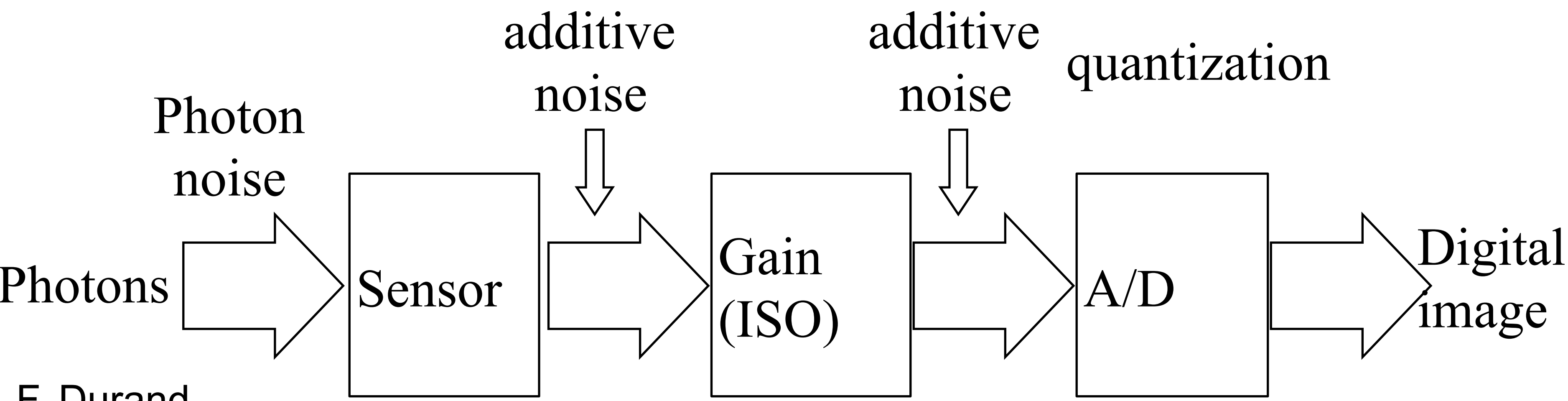
- Photosites transform photons into charge (electrons)
  - The sensor itself is linear
- Gets amplified (depending on ISO setting)
- Then goes through analog to digital converter
  - up to 14 bits/channel these days

- Stop here when shooting RAW

- Then demosaicing, denoising, white balance, a response curve, gamma encoding are applied
- Quantized and recorded as 8-bit JPEG

Pipeline & noise

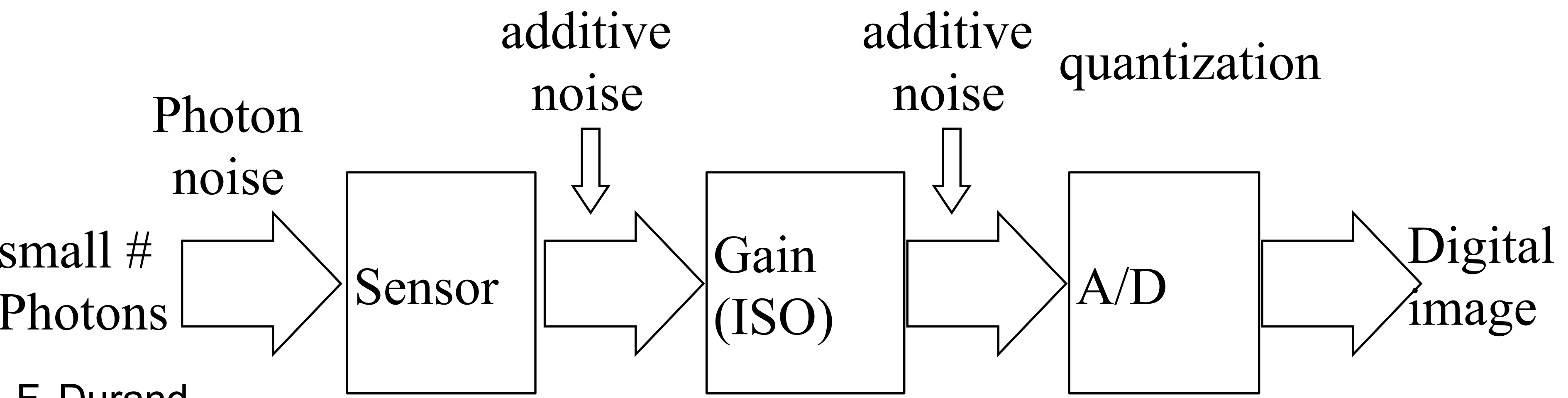
- This is a conceptual diagram, don't take it too literally
  - e.g. the AD converter is a serious source of noise, but usually electronic noise, not quantization artifacts
- Orders of magnitude:
  - # of photons per photosite : 10,000-100,000
  - Electronic noise 5-30 electrons per photosite



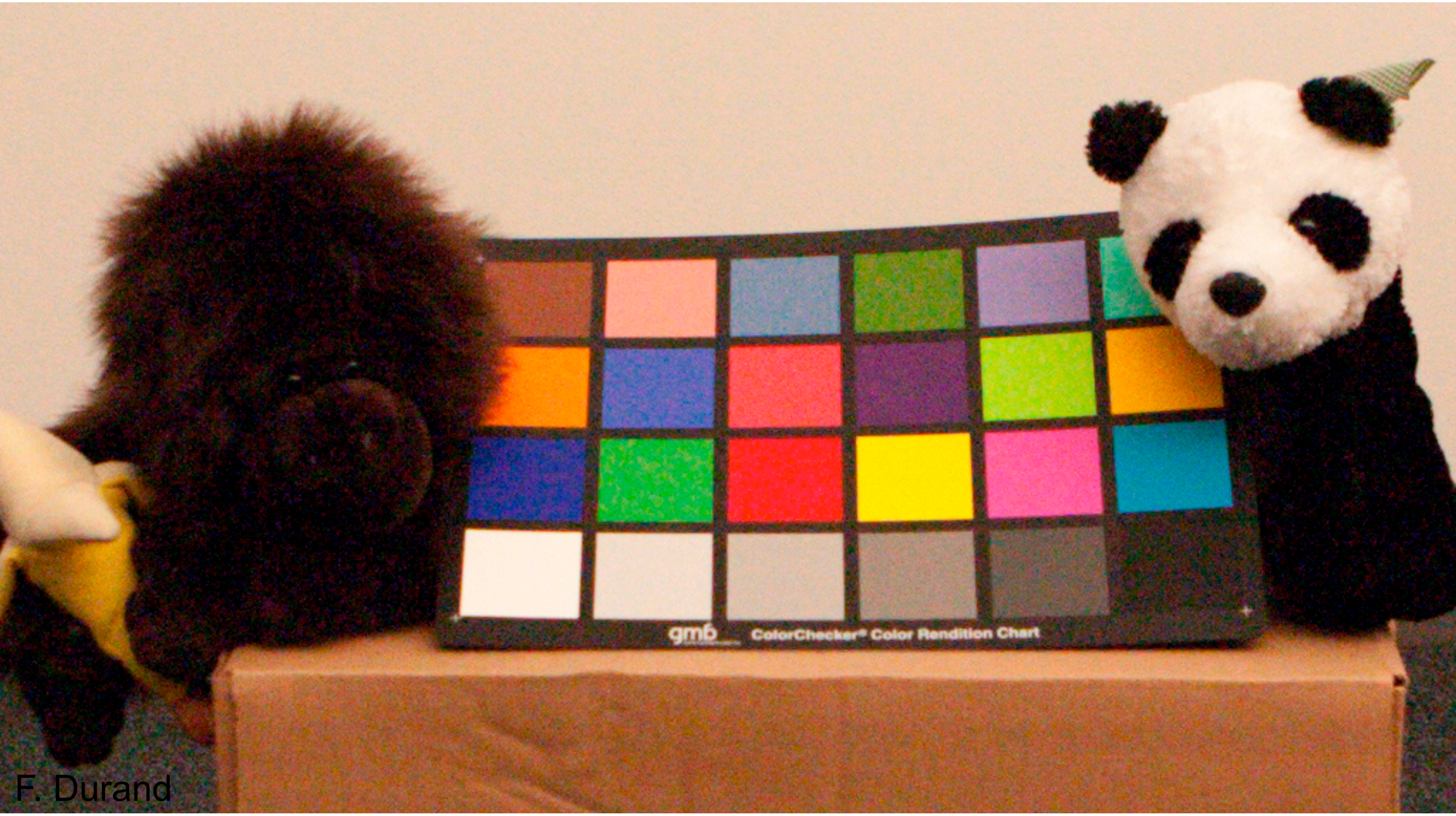
F. Durand

ISO amplifies

- e.g. going from ISO 100 to ISO 400 amplifies by 4
- both noise & signal
- usually use high ISO when signal is low
- => worse signal/noise ratio

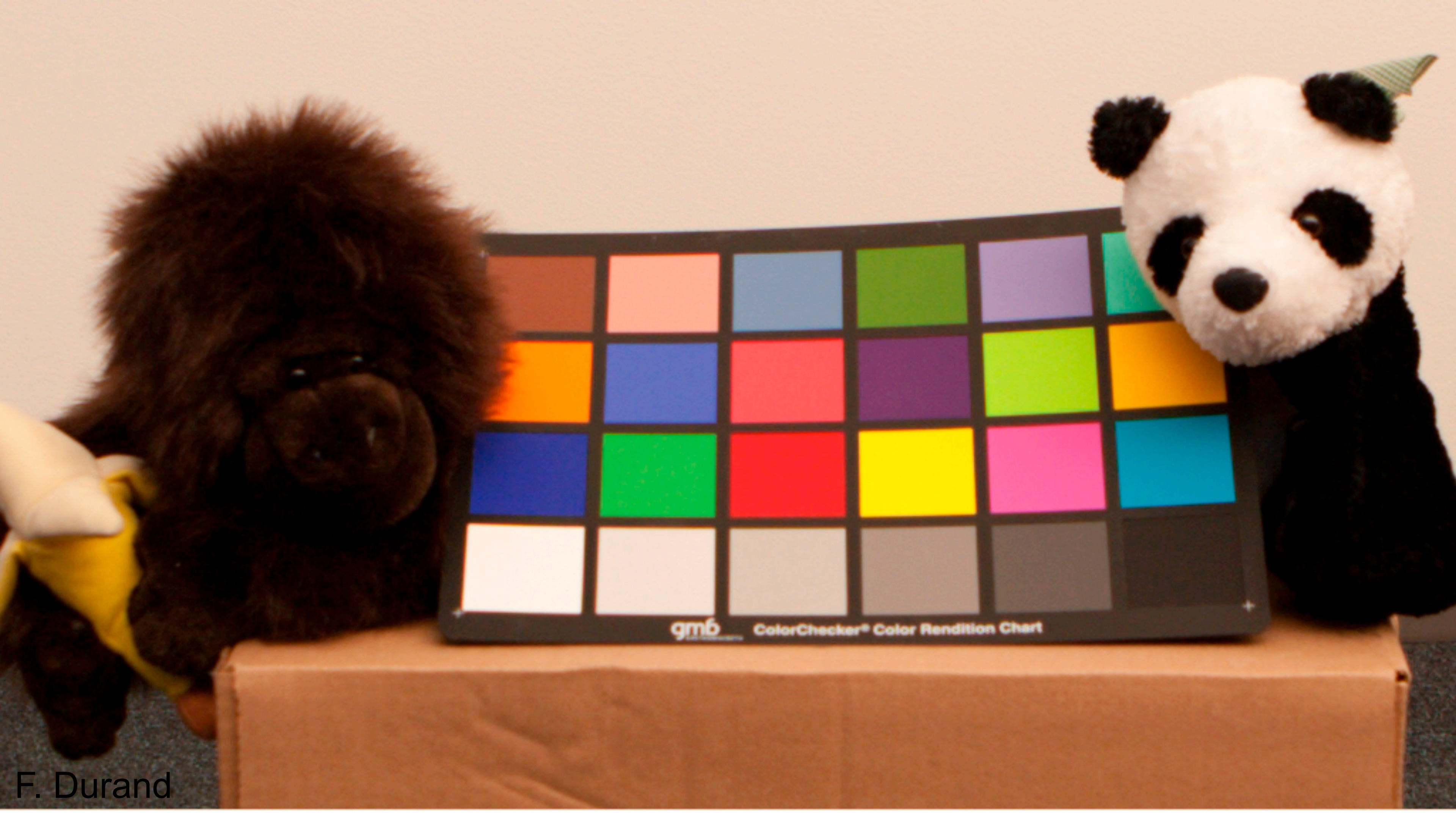


Canon 1D mark IIN at ISO 3200



Canon 1D Mark II, ISO 100

- Lot less noisy!



http://wiegaertnerfilms.com/tutorials/the-best-iso-settings-for-canon-video-dslrs/

Denoising by Averaging

1 image

F. Durand

3 images

5 images

Denoising a single image

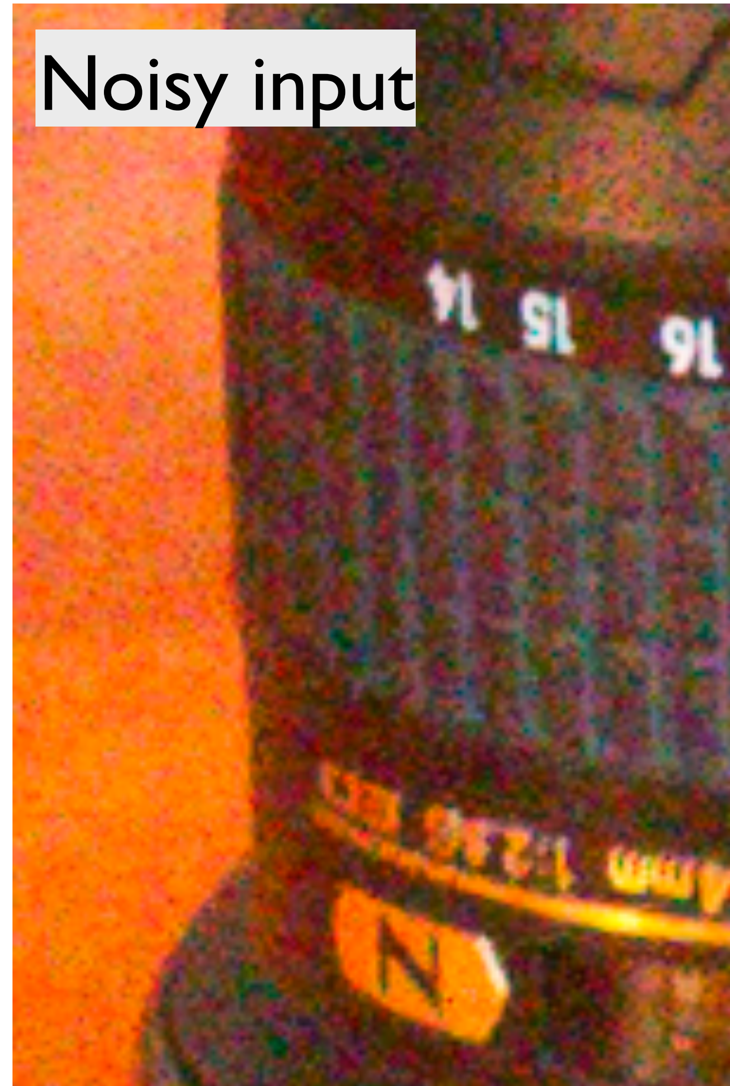
Denoising from 1 image

- We can't take average over multiple images

F. Durand

Denoising from 1 image

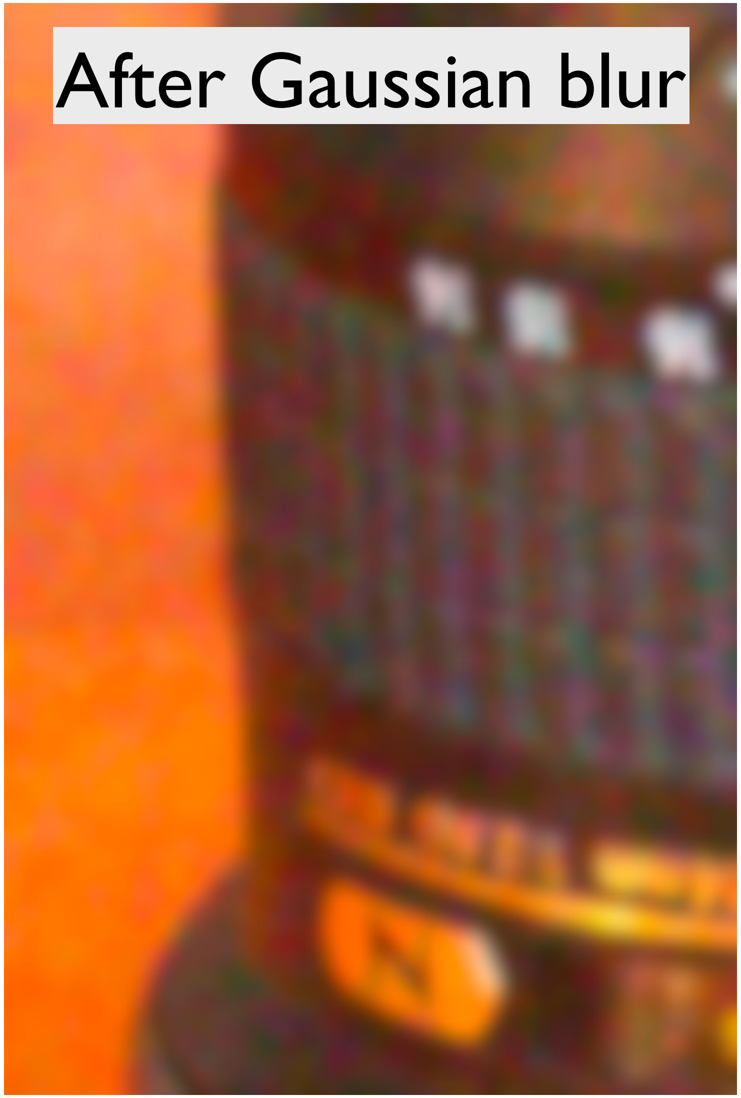
- We can't take average over multiple images
- Idea 1: take a spatial average
  - Most pixels have roughly teh same color as their neighbor
  - Noise looks high frequency => do a low pass
- Here: Gaussian blur



F. Durand

Gaussian blur

- Noise is mostly gone
- But image is blurry
  - duh!



Weiner Denoising

Wiener denoising derivation

- See http://www.cs.dartmouth.edu/farid/tutorials/fip.pdf
- Pages 57→59

Wiener denoising

- Simplest model **[Lee80]**: every neighborhood as a Gaussian vector sample with unknown variance

$$\mu = \sum_{n=1}^{N} x_n \qquad \sigma_s^2 = \left\lfloor \sum_{n=1}^{N} (x_n - \mu)^2 - \sigma_\omega^2 \right\rfloor$$

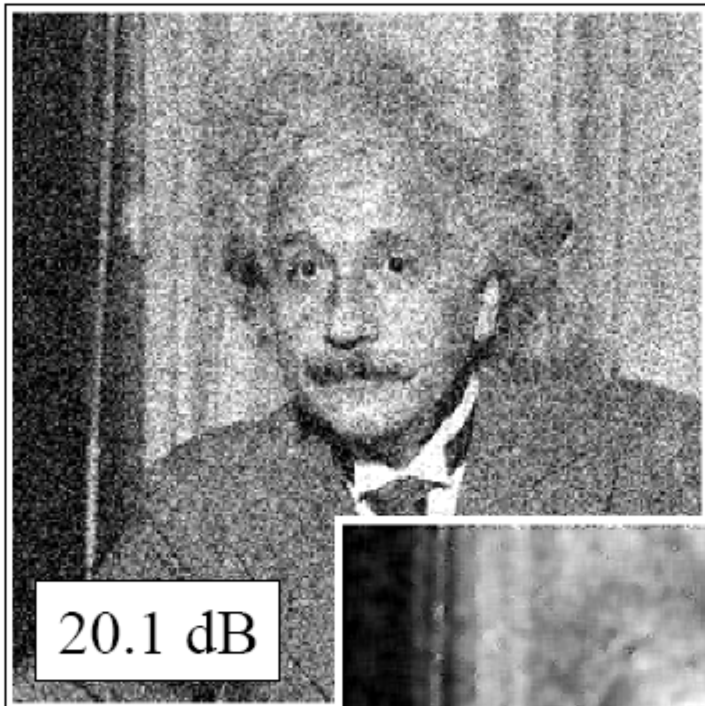
$$\boxed{\hat{\mathbf{x}} = \mu + \frac{\hat{\sigma}_s^2}{\hat{\sigma}_s^2 + \sigma_\omega^2} (\mathbf{x} - \mu)}$$

Local Wiener Estimation

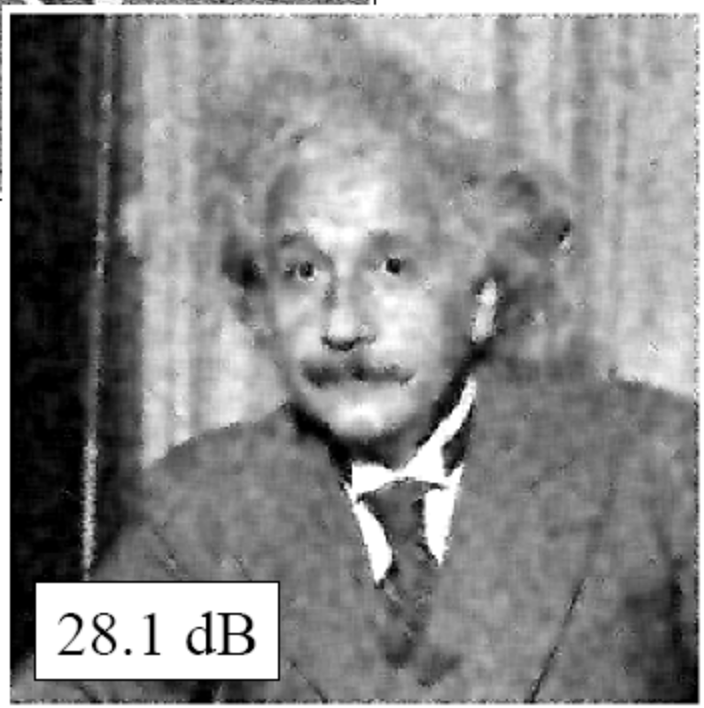
White Gaussian noise power (assumed to be known)

From J.Portilla ICIP' 01

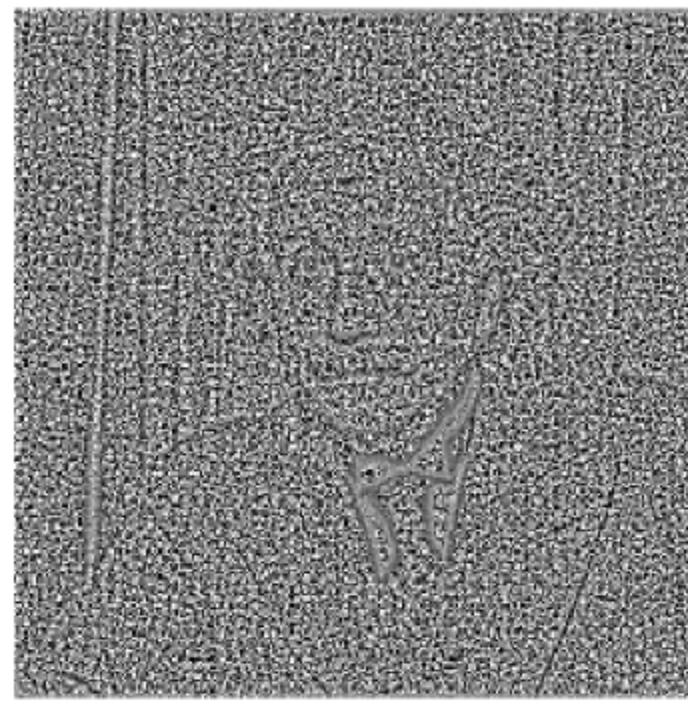
Observed Sample



- *Wiener2* Matlab function
- 5x5 pixels neighborhood



Local Wiener Filtered



Residual

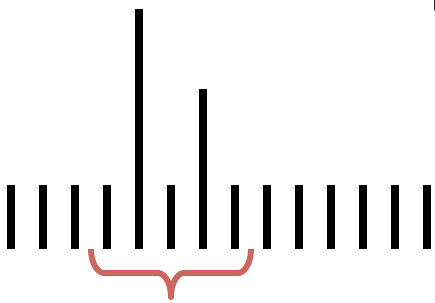
Denoising salt'n'pepper noise

Median filter

Replace each pixel by the median over N pixels (5 pixels, for these examples). Generalizes to “rank order” filters.

Median([1 7 1 5 1]) = 1
Mean([1 7 1 5 1]) = 2.8

In:

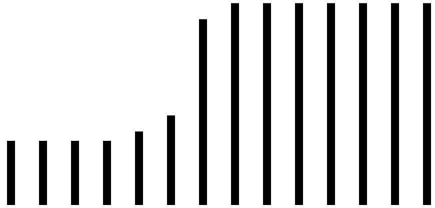


5-pixel neighborhood

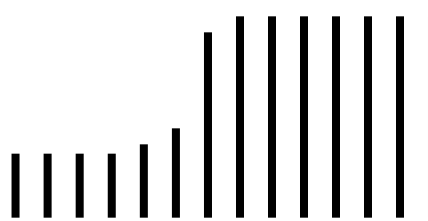
Out:

Spike noise is removed

In:



Out:



Monotonic edges remain unchanged

Median filtering results

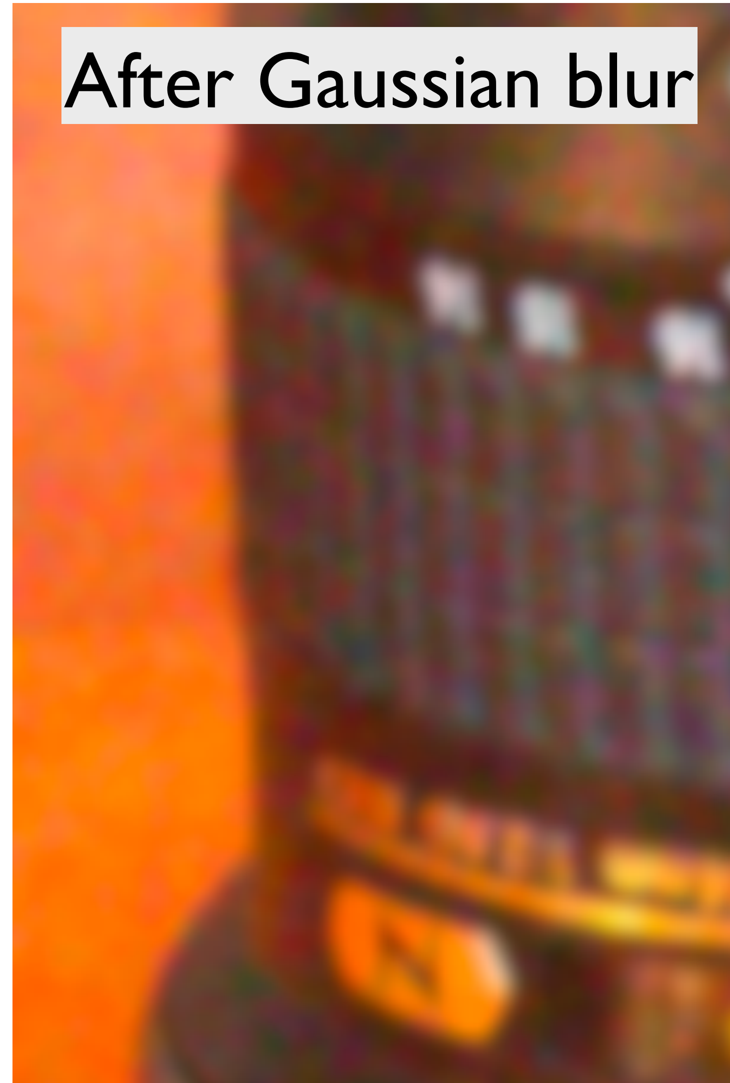
Best for salt and pepper noise

http://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm#guidelines

Bilateral filtering

Gaussian blur

- Noise is mostly gone
- But image is blurry
  - duh!
- Problem: not all neighbors have the same color
- Bilateral filter idea: only consider neighbors that have values similar enough



F. Durand

A Gentle Introduction to Bilateral Filtering and its Applications

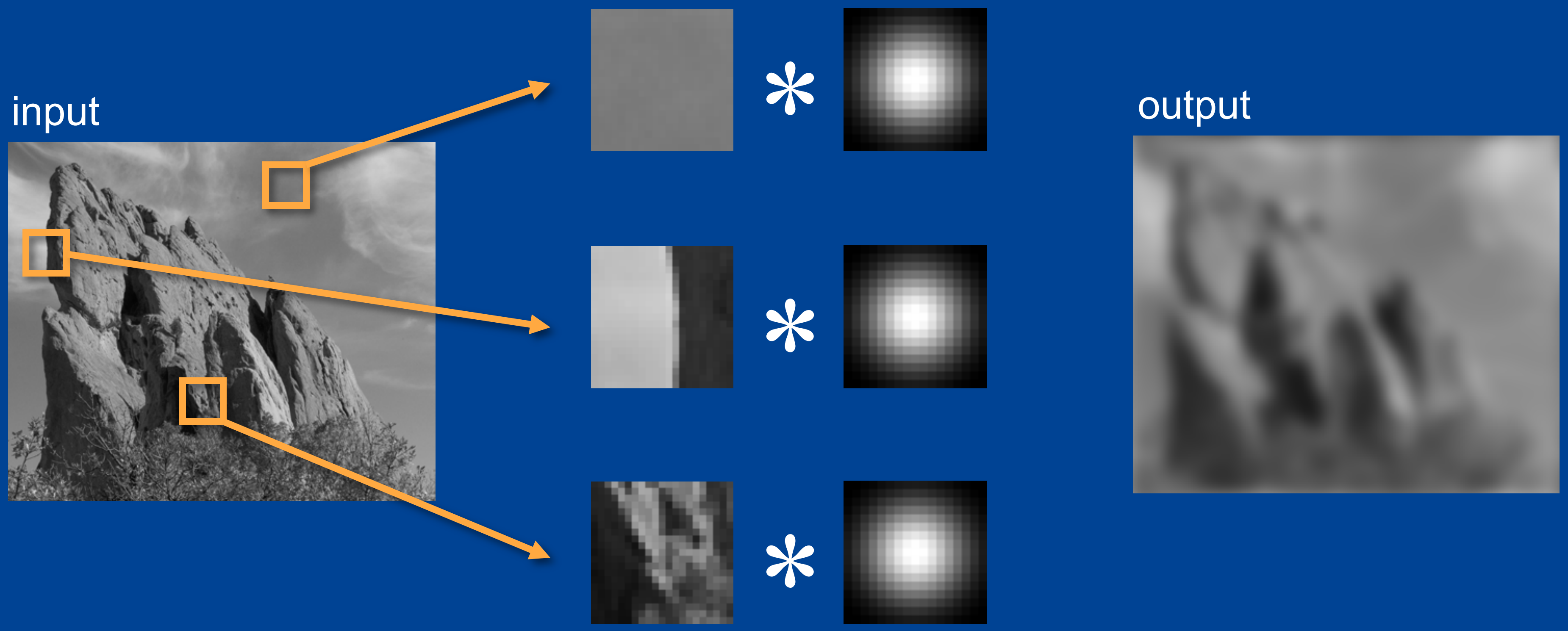
SIGGRAPH2007

“Fixing the Gaussian Blur”: the Bilateral Filter

Sylvain Paris – MIT CSAIL

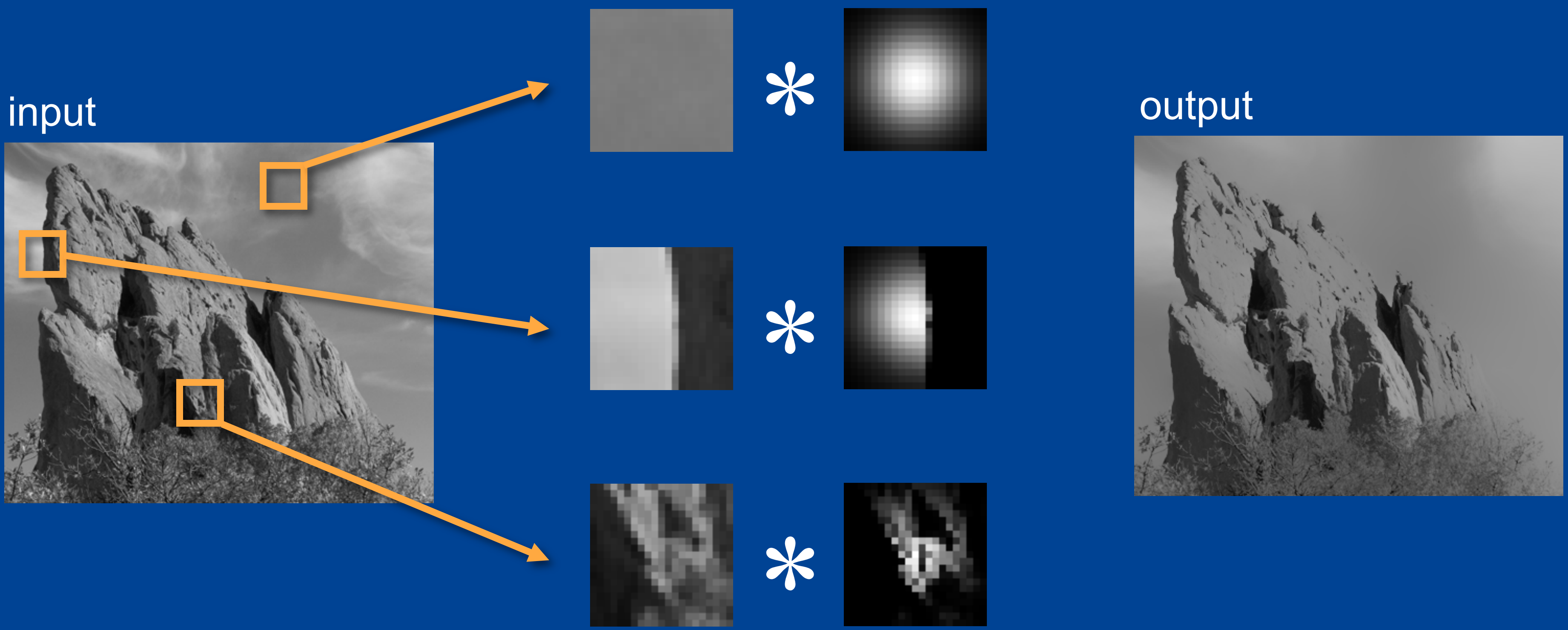
Fredo- Durand – MIT CSAIL

Blur Comes from Averaging across Edges



Same Gaussian kernel everywhere.

Bilateral Filter [Aurich 95, Smith 97, Tomasi 98]
No Averaging across Edges



The kernel shape depends on the image content.

Bilateral filter

- Tomasi and Manduci 1998]
  - http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf
- Developed for denoising
- Related to
  - SUSAN filter [Smith and Brady 95] http://citeseer.ist.psu.edu/smith95susan.html
  - Digital-TV [Chan, Osher and Chen 2001] http://citeseer.ist.psu.edu/chan01digital.html
  - sigma filter http://www.geogr.ku.dk/CHIPS/Manual/f187.htm
- Full survey: http://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf

Bilateral Filter Definition: an Additional Edge Term

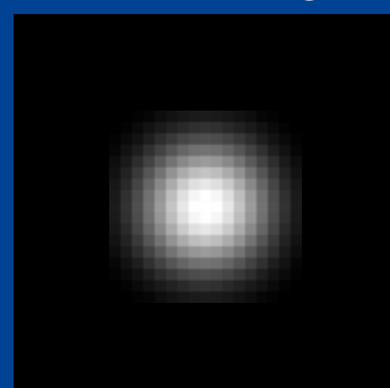
Same idea: **weighted average of pixels.**

new — not new — new

$$BF[I]_{\mathbf{p}} = \underbrace{\frac{1}{W_{\mathbf{p}}}}_{\text{new}} \sum_{\mathbf{q}\in S} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}_{\text{not new}}\, \underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}_{\text{new}}\, I_{\mathbf{q}}$$

normalization factor

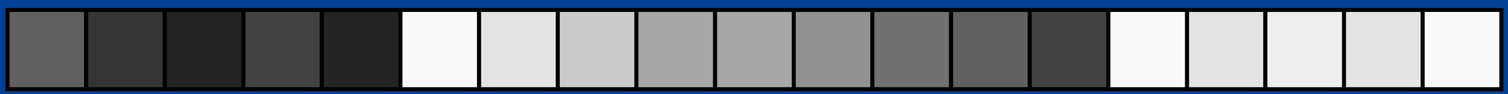
space weight



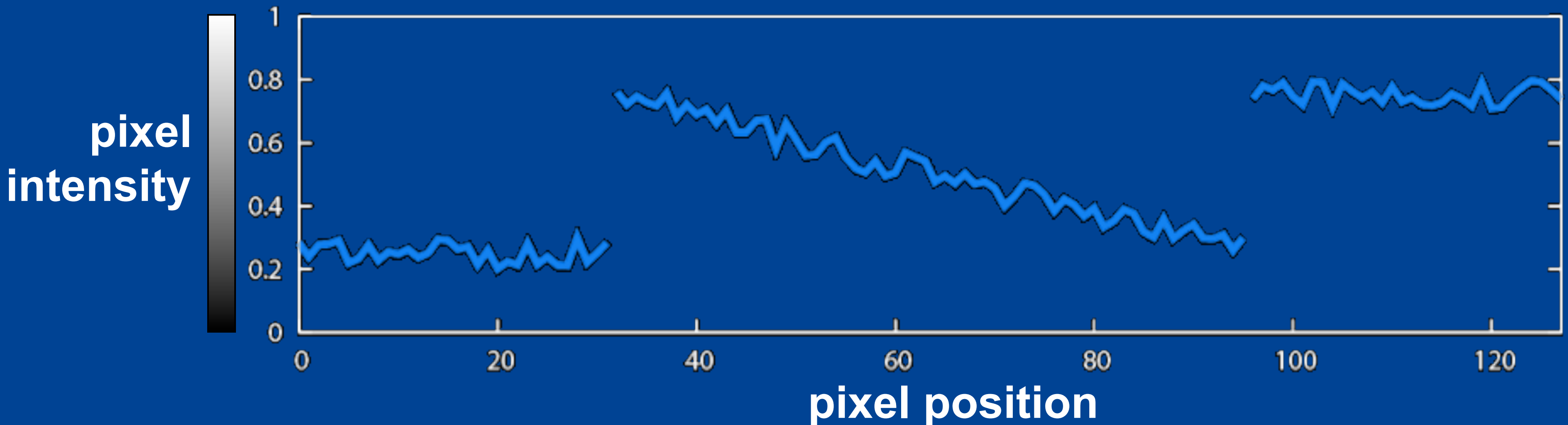
range weight

Illustration a 1D Image

- 1D image = line of pixels

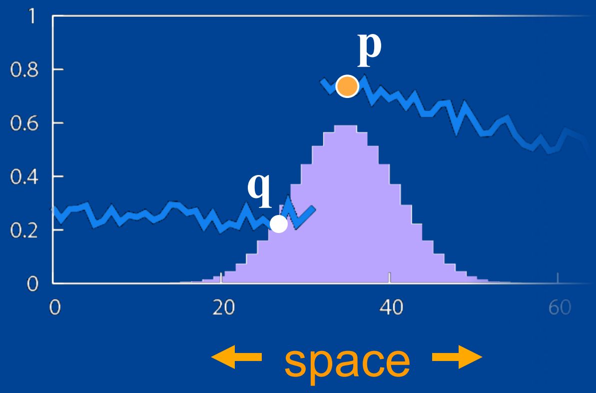


- Better visualized as a plot



Gaussian Blur and Bilateral Filter

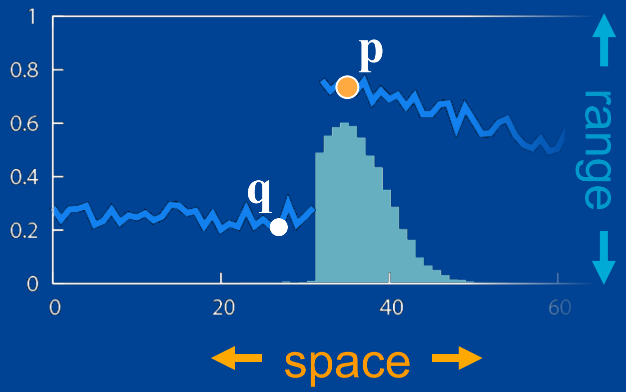
Gaussian blur



$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q}\in S} \underbrace{G_{\sigma}(\|\mathbf{p}-\mathbf{q}\|)}_{\text{space}} I_{\mathbf{q}}$$

Bilateral filter

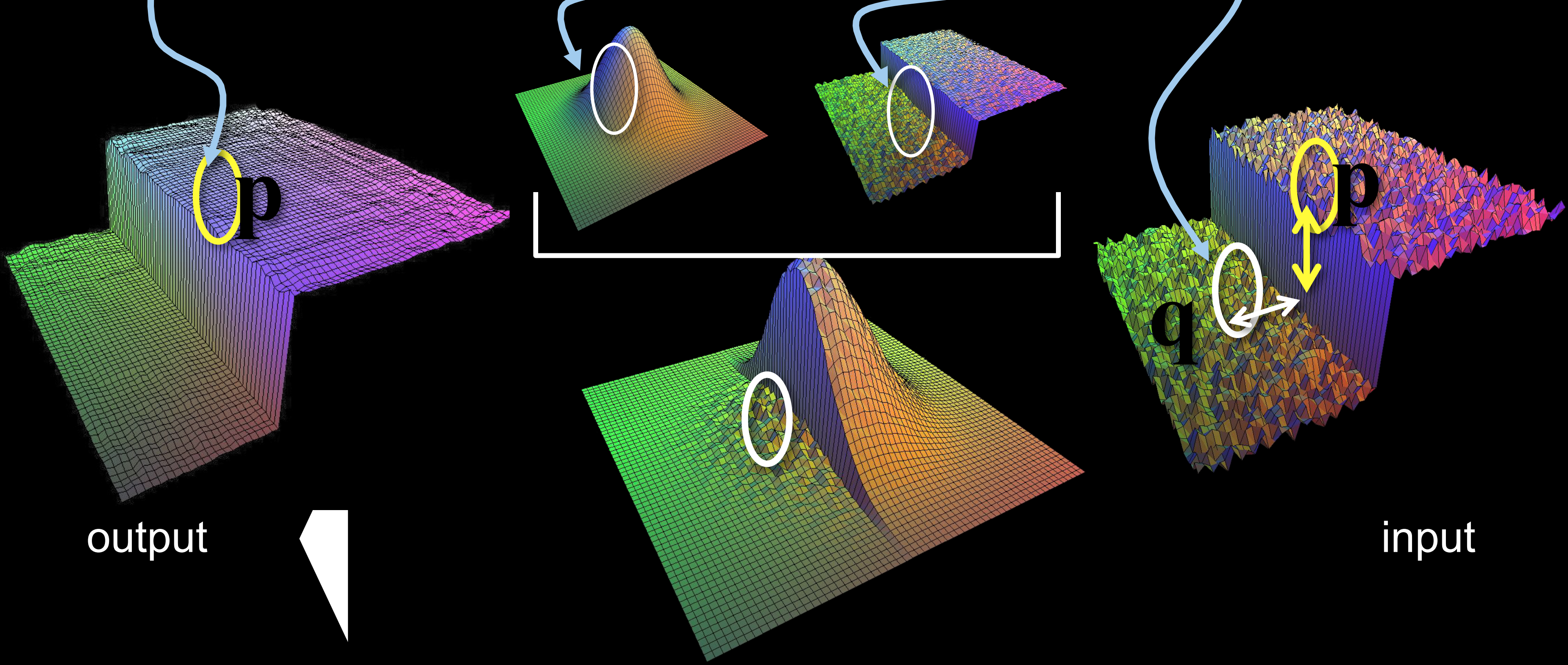
[Aurich 95, Smith 97, Tomasi 98]



$$BF[I]_{\mathbf{p}} = \underbrace{\frac{1}{W_{\mathbf{p}}}}_{\text{normalization}} \sum_{\mathbf{q}\in S} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}_{\text{space}} \underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}_{\text{range}} I_{\mathbf{q}}$$

Bilateral Filter on a Height Field

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$



reproduced from [Durand 02]

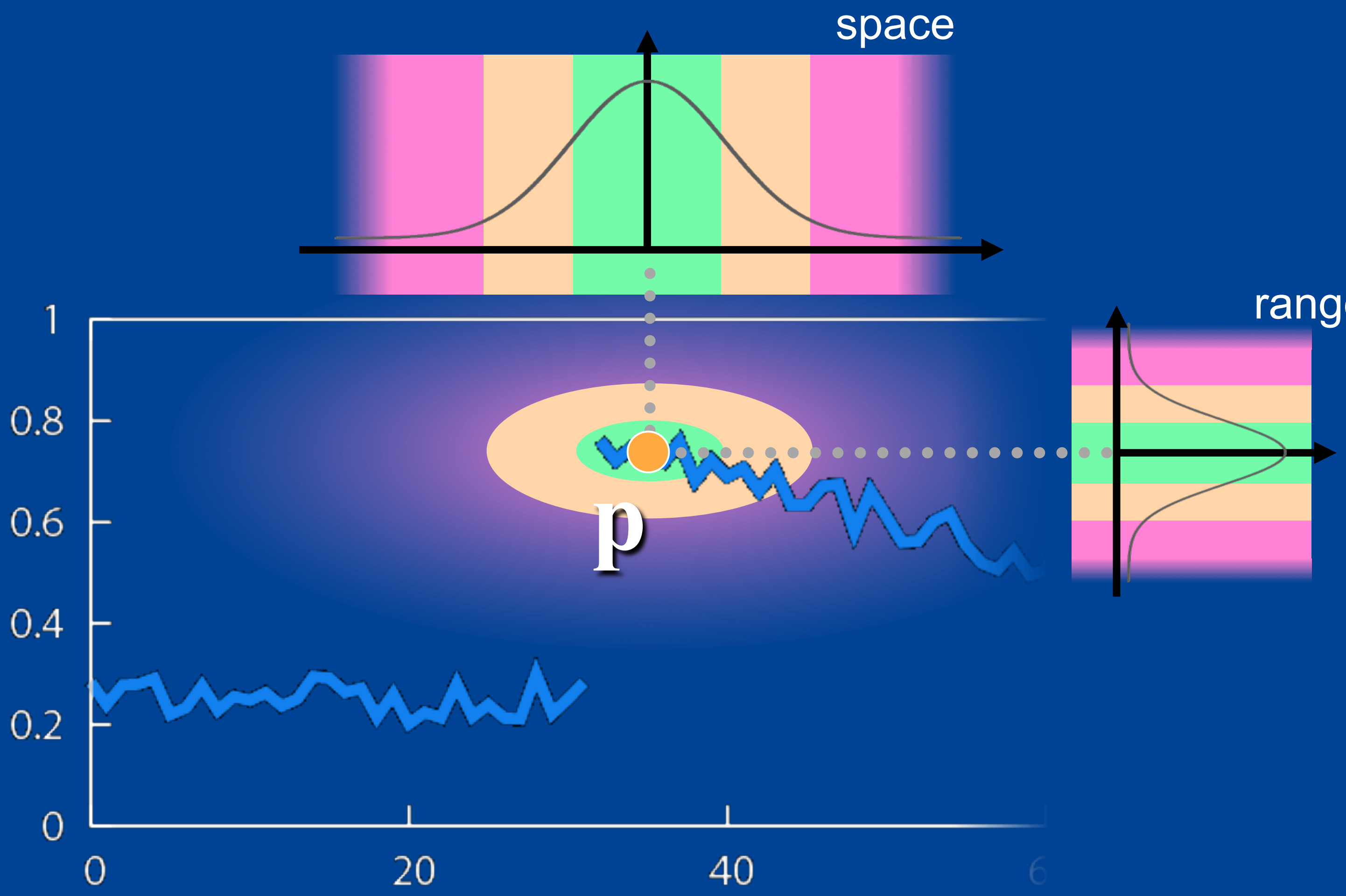
Space and Range Parameters

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_{\mathrm{s}}}\left(\| \mathbf{p} - \mathbf{q} \|\right) G_{\sigma_{\mathrm{r}}}\left(| I_{\mathbf{p}} - I_{\mathbf{q}} |\right) I_{\mathbf{q}}$$

- space σ_{s} : spatial extent of the kernel, size of the considered neighborhood.
- range σ_{r} : "minimum" amplitude of an edge

Influence of Pixels

Only pixels close in space and in range are considered.



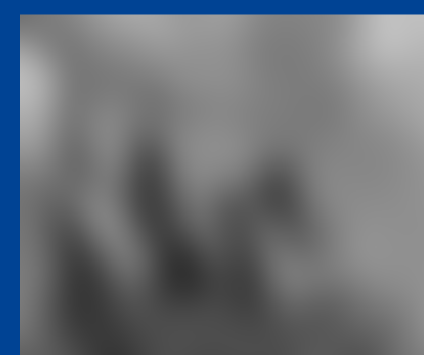
Bilateral filter

Noisy input

After bilateral filter

Exploring the Parameter Space

input

	$\sigma_r = 0.1$	$\sigma_r = 0.25$	$\sigma_r = \infty$ (Gaussian blur)
$\sigma_s = 2$			
$\sigma_s = 6$			
$\sigma_s = 18$			

Varying the Range Parameter

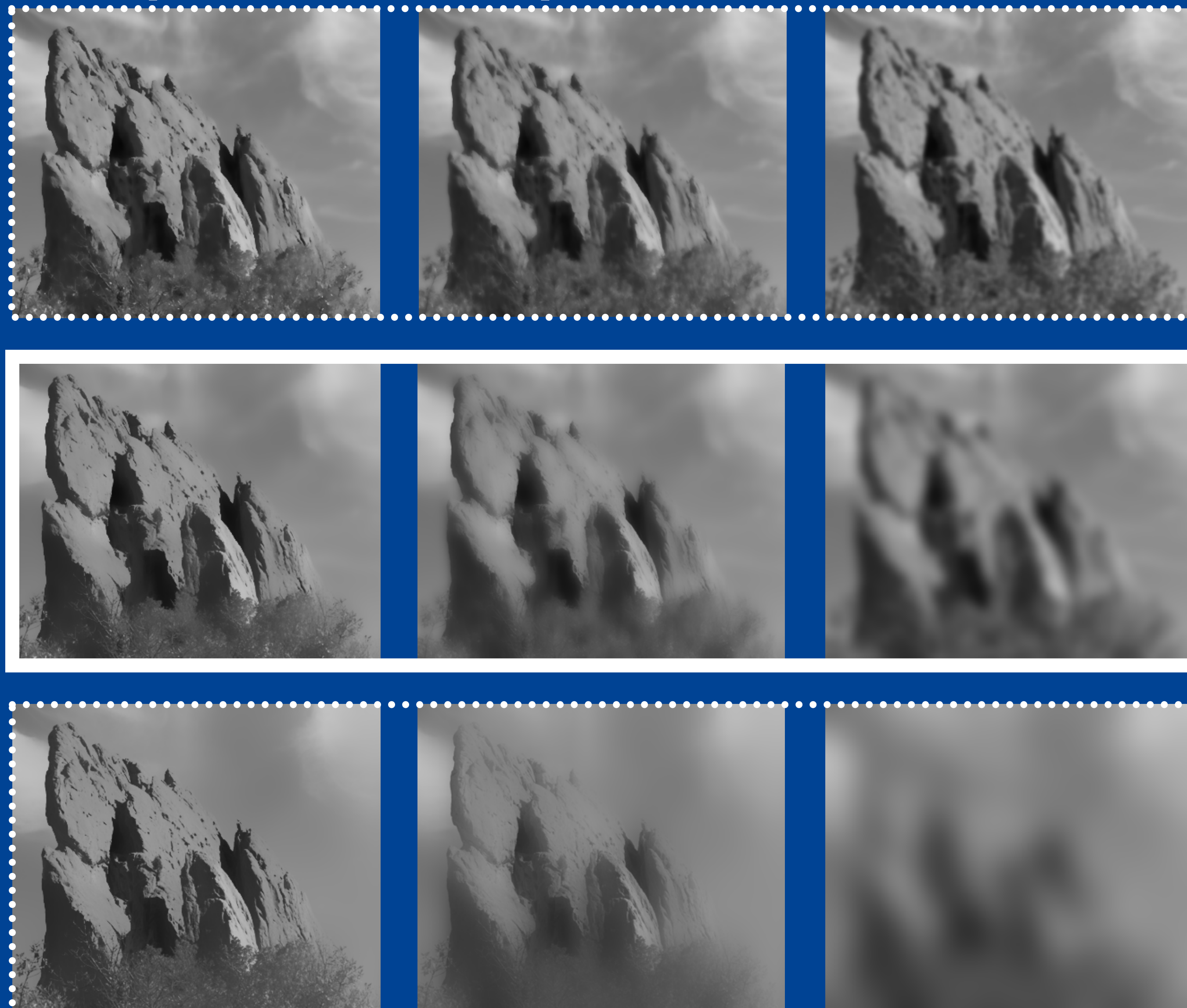
input

$\sigma_r = 0.1$ | $\sigma_r = 0.25$ | $\sigma_r = \infty$ (Gaussian blur)

$\sigma_s = 2$

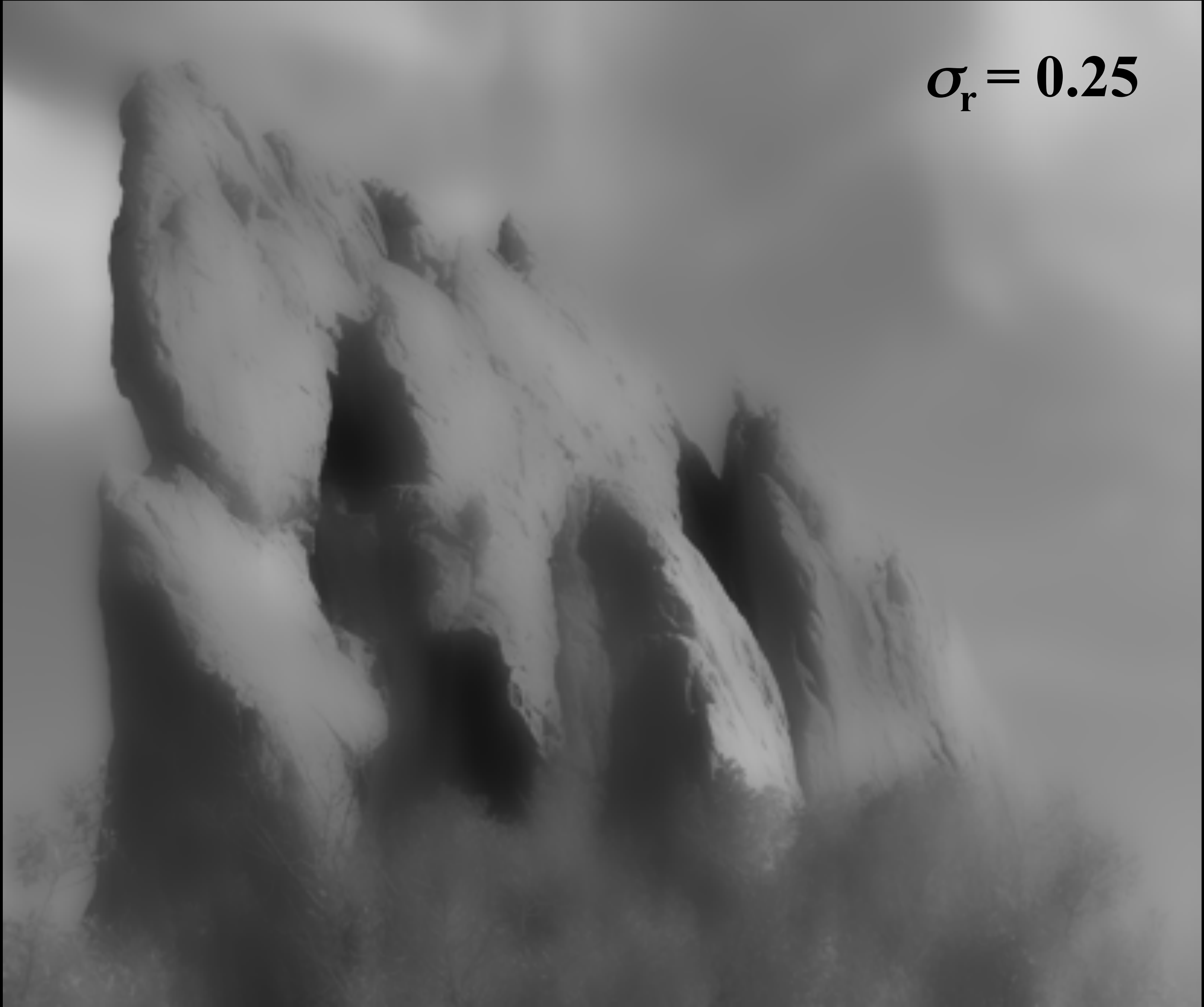
$\sigma_s = 6$

$\sigma_s = 18$




input


$\sigma_r = 0.1$


$\sigma_r = 0.25$


$\sigma_r = \infty$
(Gaussian blur)

Varying the Space Parameter

input

$\sigma_r = 0.1$

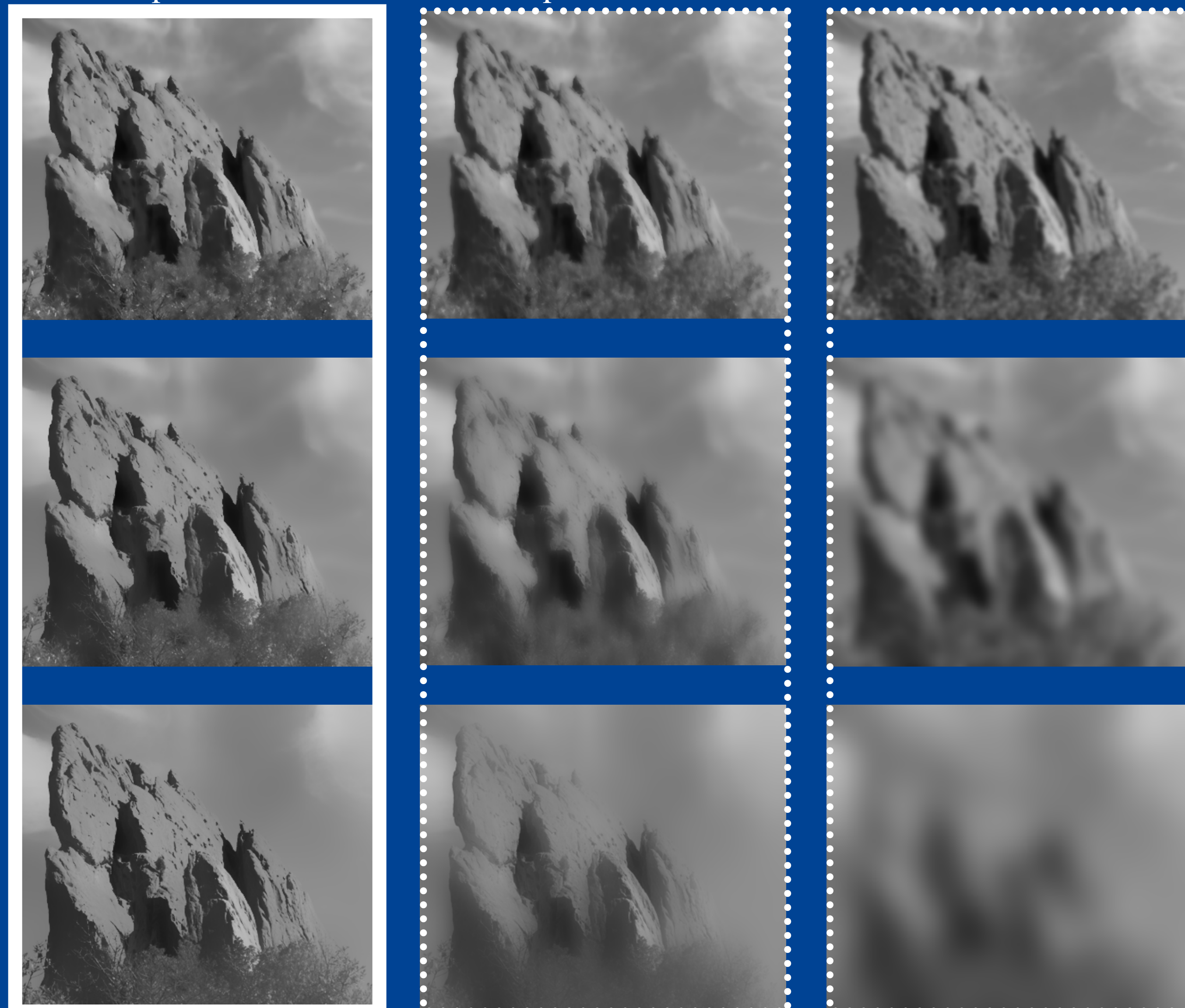
$\sigma_r = 0.25$

$\sigma_r = \infty$
(Gaussian blur)

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$




input


$\sigma_s = 2$


$\sigma_s = 6$


$\sigma_s = 18$

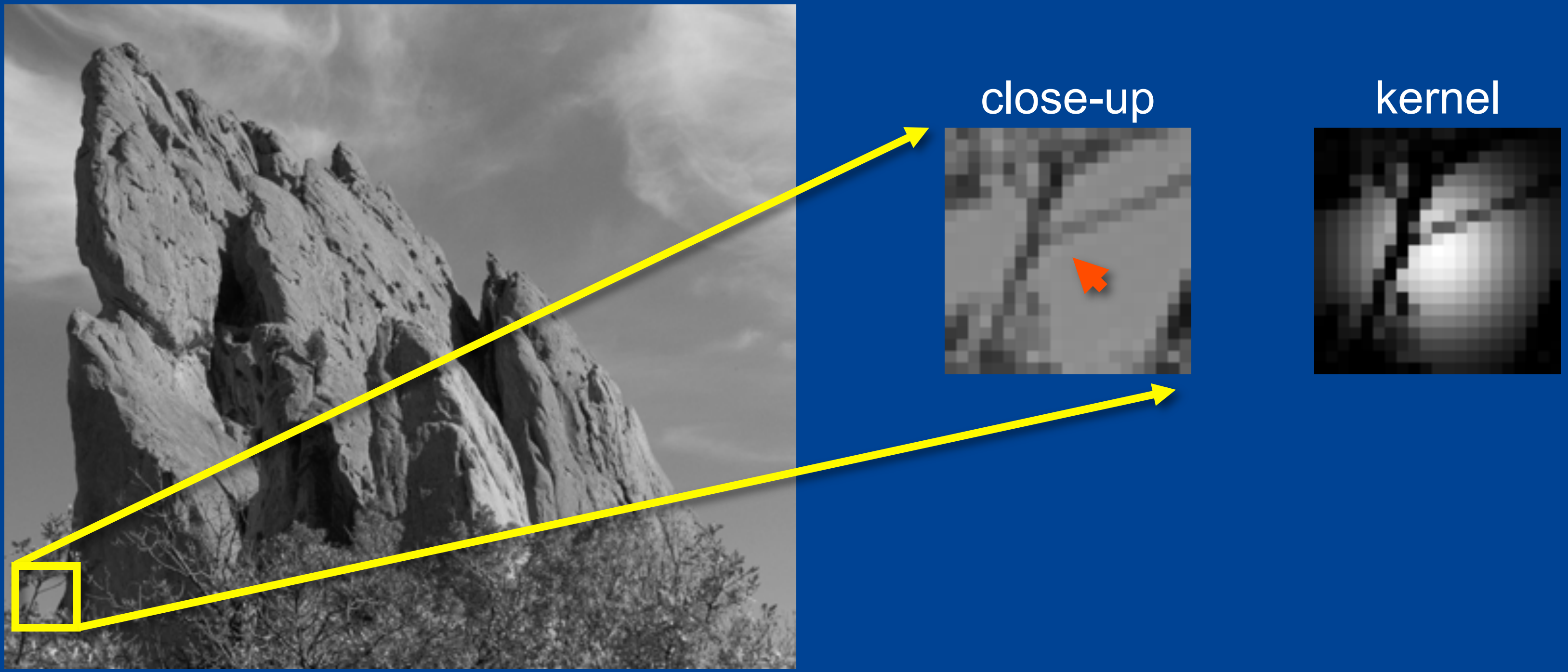
How to Set the Parameters

Depends on the application. For instance:

- space parameter: proportional to image size
  - e.g., 2% of image diagonal
- range parameter: proportional to edge amplitude
  - e.g., mean or median of image gradients
- independent of resolution and exposure

Bilateral Filter Crosses Thin Lines

- Bilateral filter averages across features thinner than ~$2\sigma_s$
- Desirable for smoothing: more pixels = more robust
- Different from diffusion that stops at thin lines



Bilateral Filtering Color Images

For gray-level images

intensity difference

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|) \, G_{\sigma_r}\left(\boxed{| I_{\mathbf{p}} - I_{\mathbf{q}} |} \right) \boxed{I_{\mathbf{q}}}$$

scalar

For color images

color difference

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|) \, G_{\sigma_r}\left(\boxed{\| \mathbf{C}_{\mathbf{p}} - \mathbf{C}_{\mathbf{q}} \|} \right) \boxed{\mathbf{C}_{\mathbf{q}}}$$

3D vector
(RGB, Lab)

Hard to Compute

- Nonlinear $$BF[I]_{\mathbf{p}} = \boxed{\frac{1}{W_{\mathbf{p}}}} \sum_{\mathbf{q}\in S} G_{\sigma_s}\left(\| \mathbf{p} - \mathbf{q} \|\right) \boxed{G_{\sigma_r}\left(| I_{\mathbf{p}} - I_{\mathbf{q}} |\right)} I_{\mathbf{q}}$$
- Complex, spatially varying kernels
  - Cannot be precomputed, no FFT…

- Brute-force implementation is slow > 10min

Basic denoising

Noisy input

Bilateral filter 7x7 window

Basic denoising

Bilateral filter

Median 3x3

Basic denoising

Bilateral filter

Median 5x5

Basic denoising

Bilateral filter

Bilateral filter – lower sigma

Basic denoising

Bilateral filter

Bilateral filter – higher sigma

Denoising

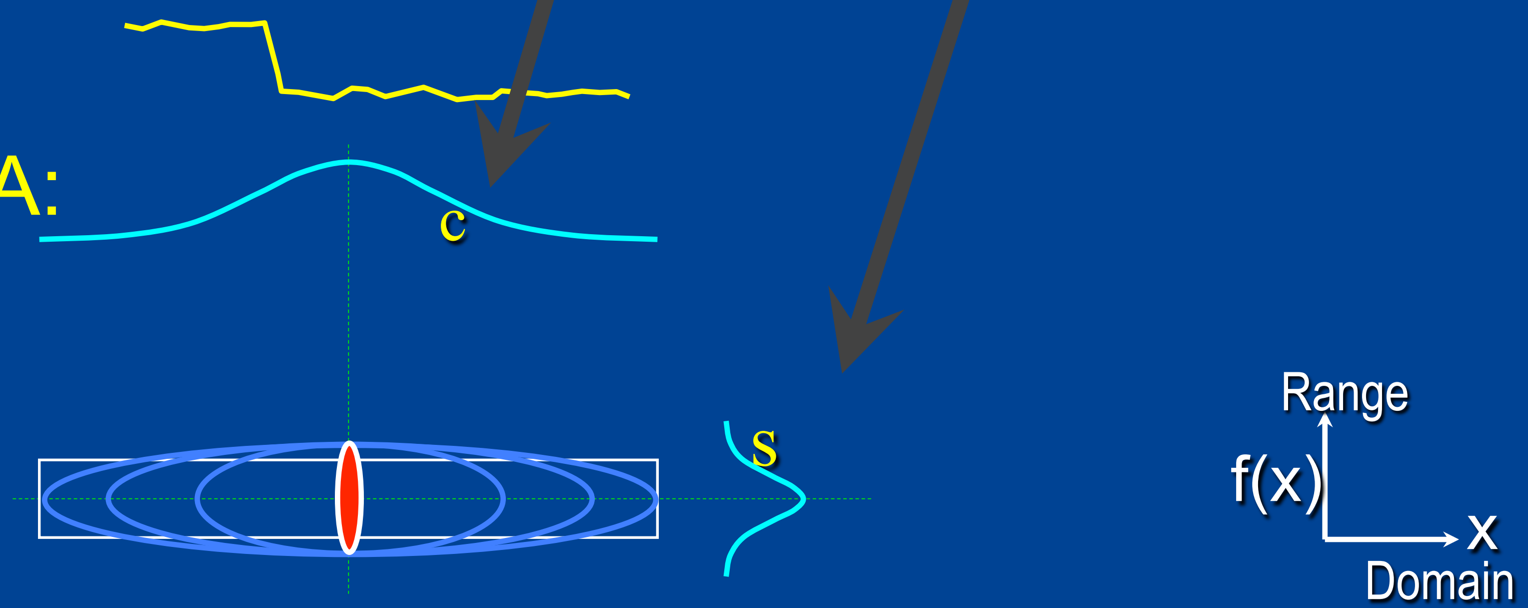
- Small spatial sigma (e.g. 7x7 window)
- Adapt range sigma to noise level
- Maybe not best denoising method, but best simplicity/quality tradeoff
  - No need for acceleration (small kernel)
  - But the denoising feature in e.g. Photoshop is better

Ordinary Bilateral Filter

Bilateral → *two kinds* of weights, one image A :

$$BF[A]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \boxed{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)} \boxed{G_{\sigma_r}(|A_{\mathbf{p}} - A_{\mathbf{q}}|)} \boxed{A_{\mathbf{q}}}$$

Image A:



'Joint' or 'Cross' Bilateral Filter

NEW: *two kinds* of weights, *two* images

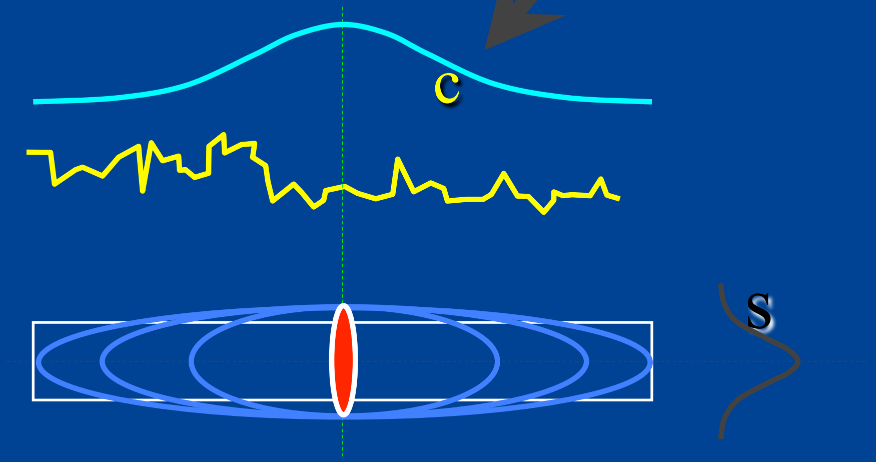
$$BF[A]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \boxed{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)} \boxed{G_{\sigma_r}(|B_{\mathbf{p}} - B_{\mathbf{q}}|)} A_{\mathbf{q}}$$

A: Noisy, dim

(ambient image)

B: Clean,strong

(Flash image)



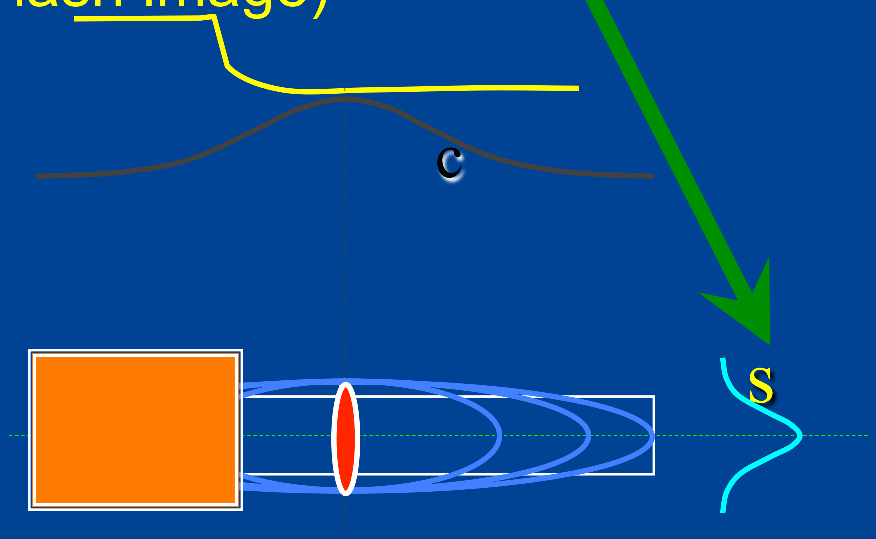


Image A: Warm, shadows, but too Noisy
(too dim for a good quick photo)

Image B: Cold, Shadow-free, Clean
(flash: simple light, ALMOST no shadows)

MERGE BEST OF BOTH: apply ‘Cross Bilateral’ or ‘Joint Bilateral’

(it really is *much* better!)

Dark Flash Photography

Dilip Krishnan
Rob Fergus

Dept. of Computer Science,
Courant Institute,
New York University

Our Camera & Dark Flash

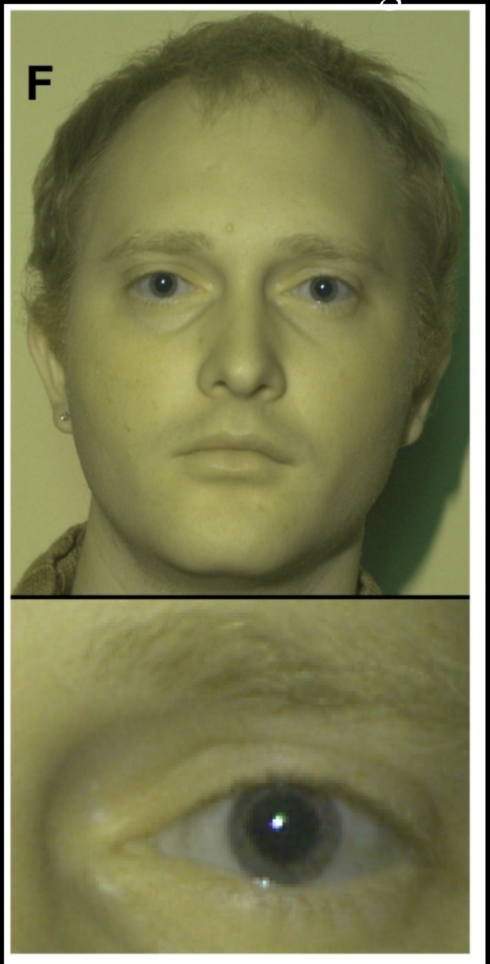
Dark Flash

Emits Ultraviolet (UV) and Infrared (IR) light just outside visible wavelength range

Dark Flash Photography

- Dark flash is ~200 times dimmer than conventional

1. Dark Flash image



Key Challenges

1. How to add light to the scene without it being perceived by people.
2. How to obtain an image with correct colors.

Key Challenges

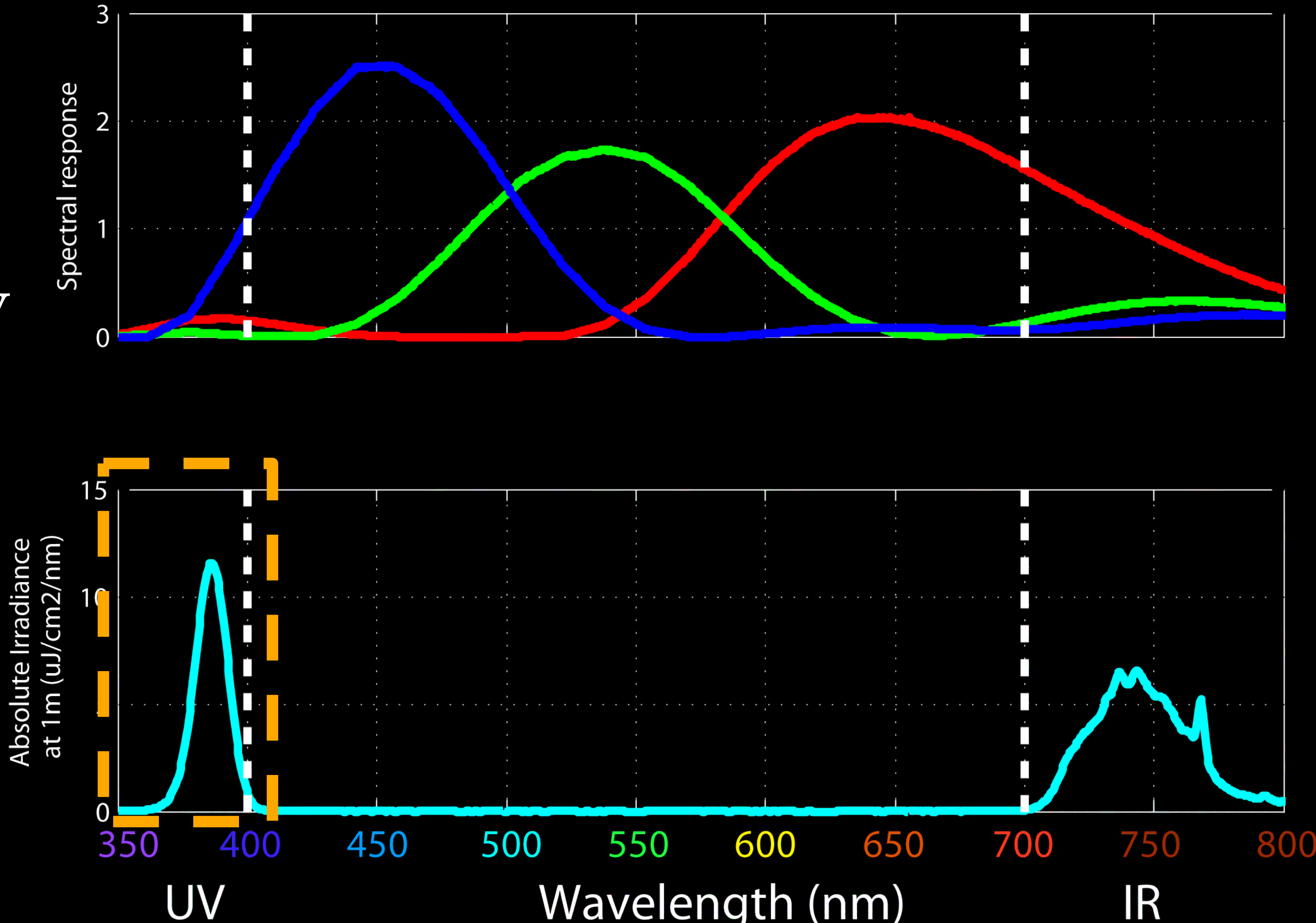
1. How to add light to the scene without it being perceived by people.

2. How to obtain an image with correct colors.

Dark Flash Emission Spectrum

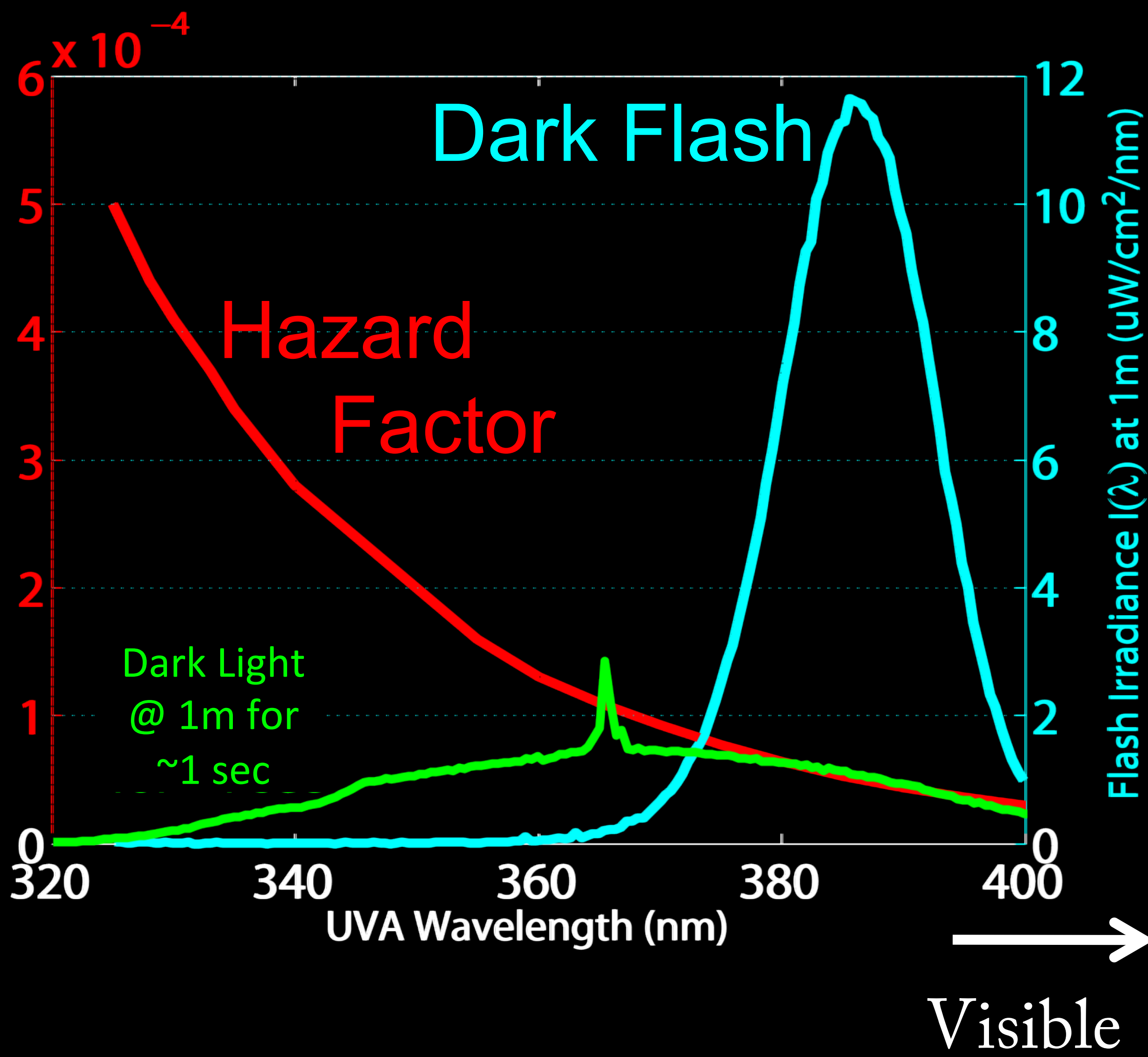
Camera Spectral Sensitivity

Dark Flash Emission



Flash Safety

- Government tables specify safe limits of exposure to UV (< 400nm)

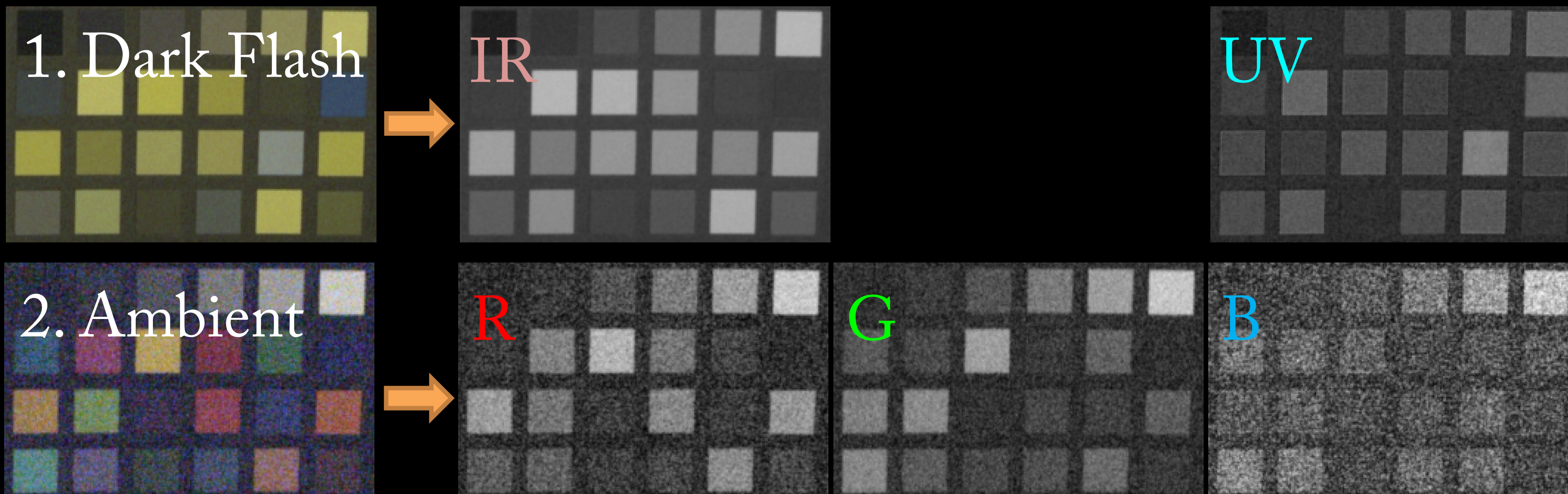


Key Challenges

1. How to add light to the scene without it being perceived by people.

2. How to obtain an image with correct colors.

Two Images: Five Spectral Bands

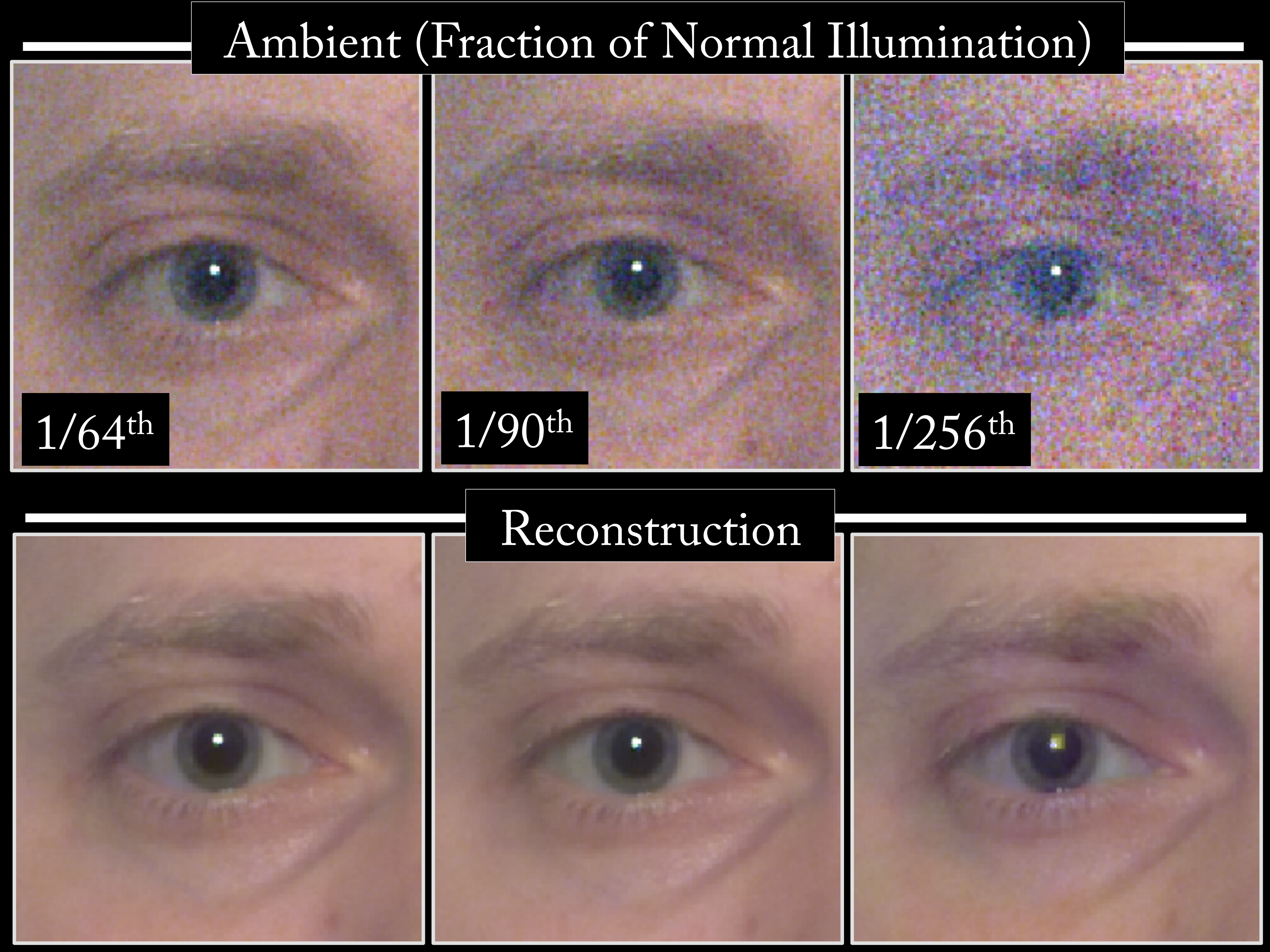


- In Dark Flash image:
  - “Blue” channel records UV
  - “Red” channel records IR

Assumptions

1. Little ambient UV and IR light
2. UV/IR flash dominates ambient visible light

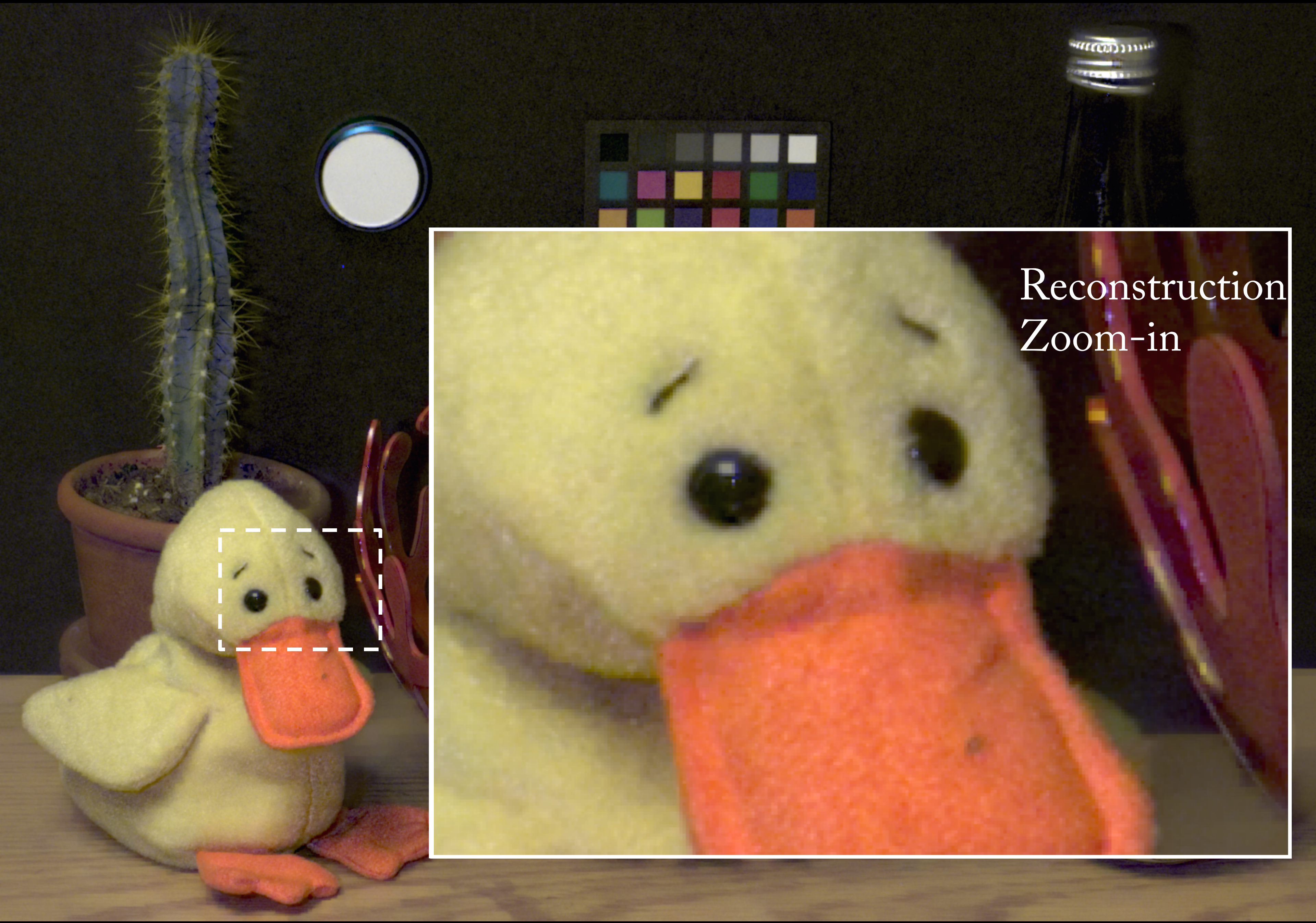
Ambient : $1/20^{\text{th}}$ sec Reconstruction Long exposure: 4 sec


Ambient (Fraction of Normal Illumination)
1/64th
1/90th
1/256th
Reconstruction

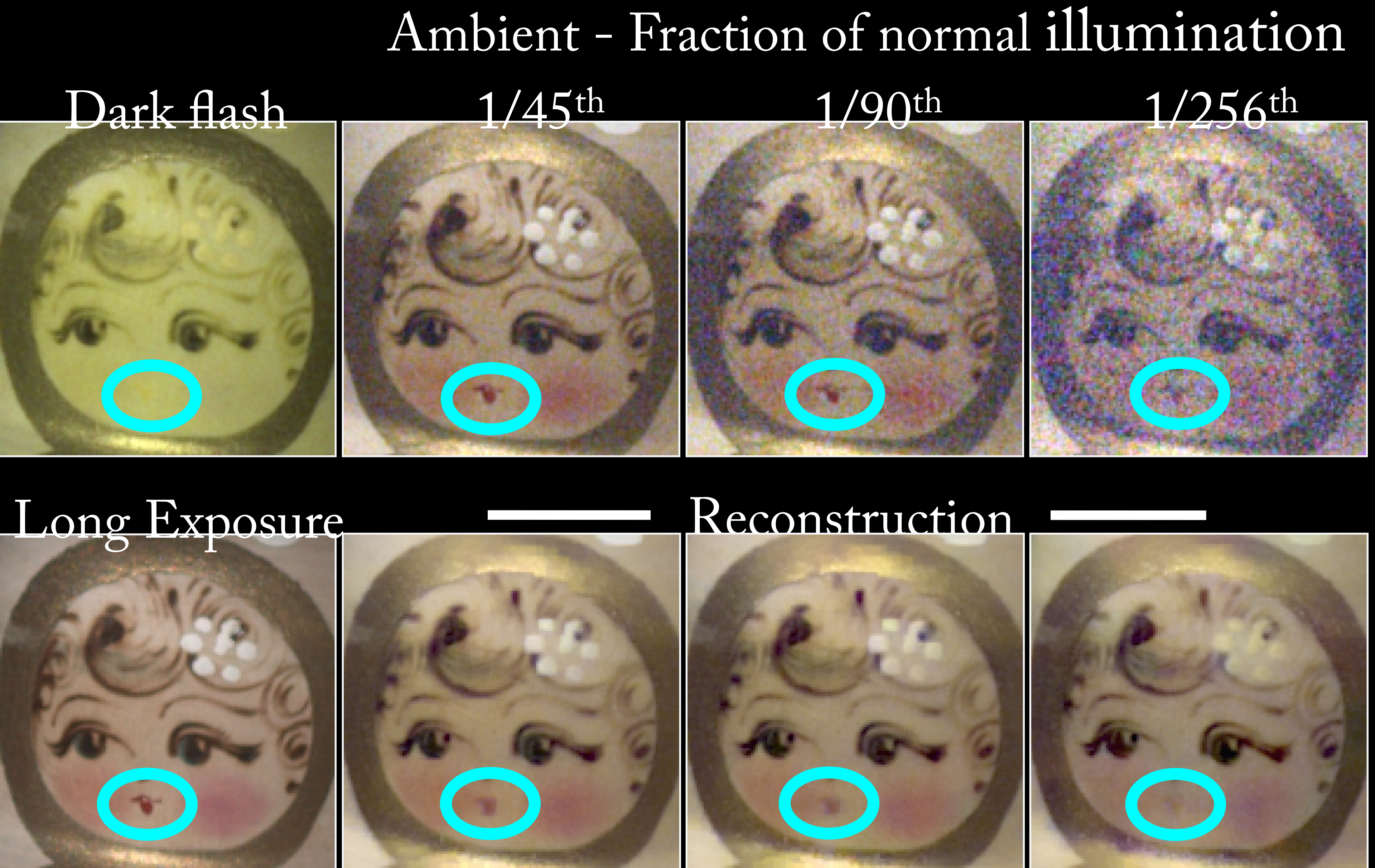
Ambient Illumination: 1/40th Normal Lighting

Reconstruction

Reconstruction



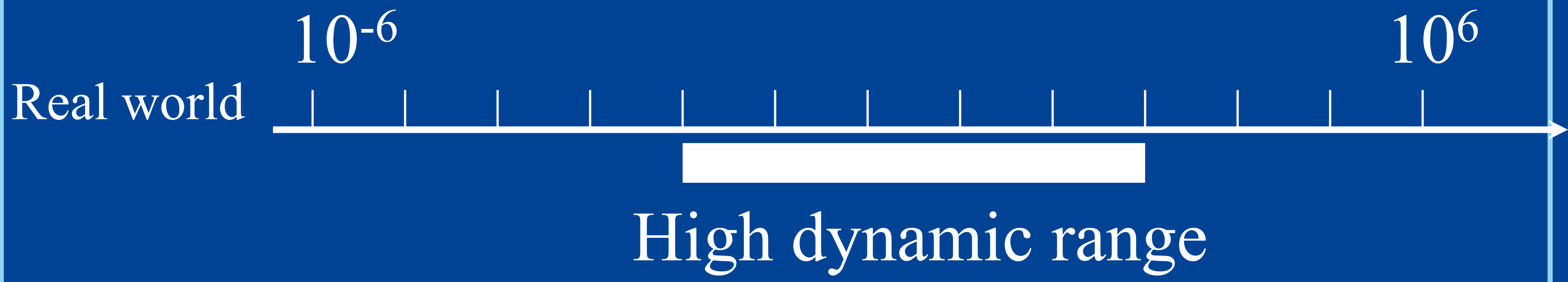
Limitations – Lack of edges in UV/IR



High Dynamic Range Imaging

Real world dynamic range

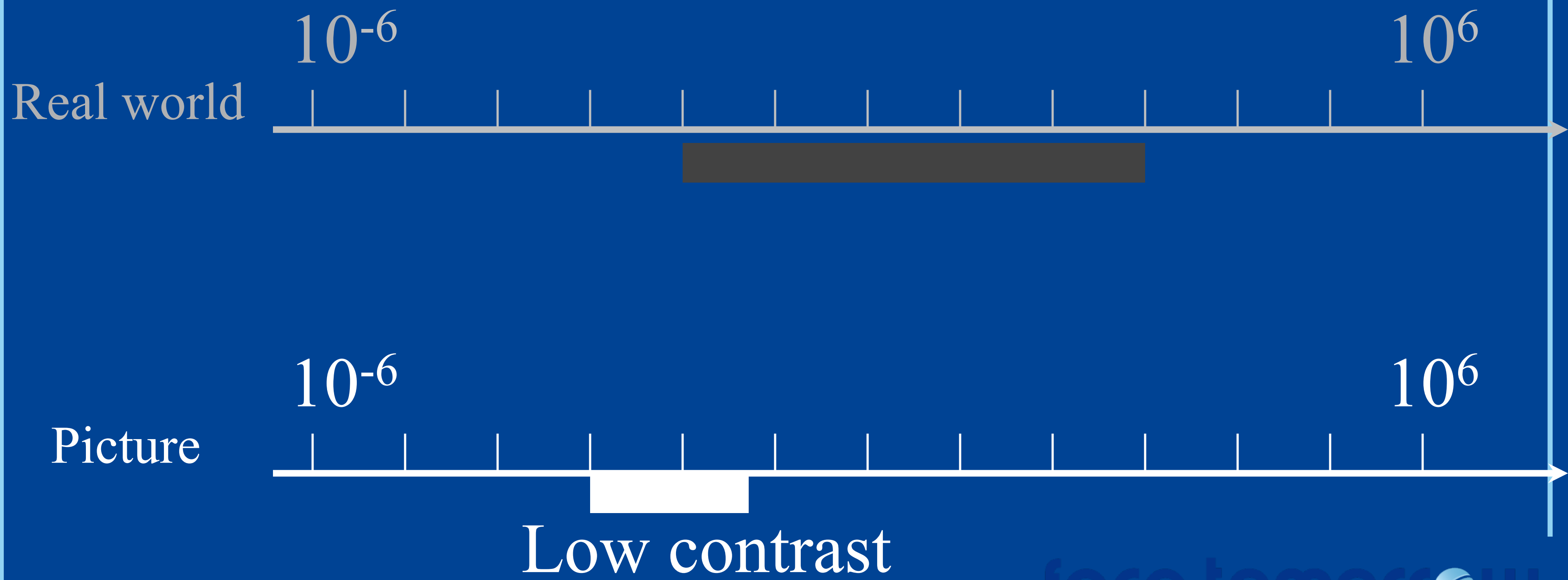
- Eye can adapt from ~ 10^{-6} to 10^{6} cd/m^2
- Often 1 : 10,000 in a scene



face tomorrow SIGGRAPH2007

Picture dynamic range

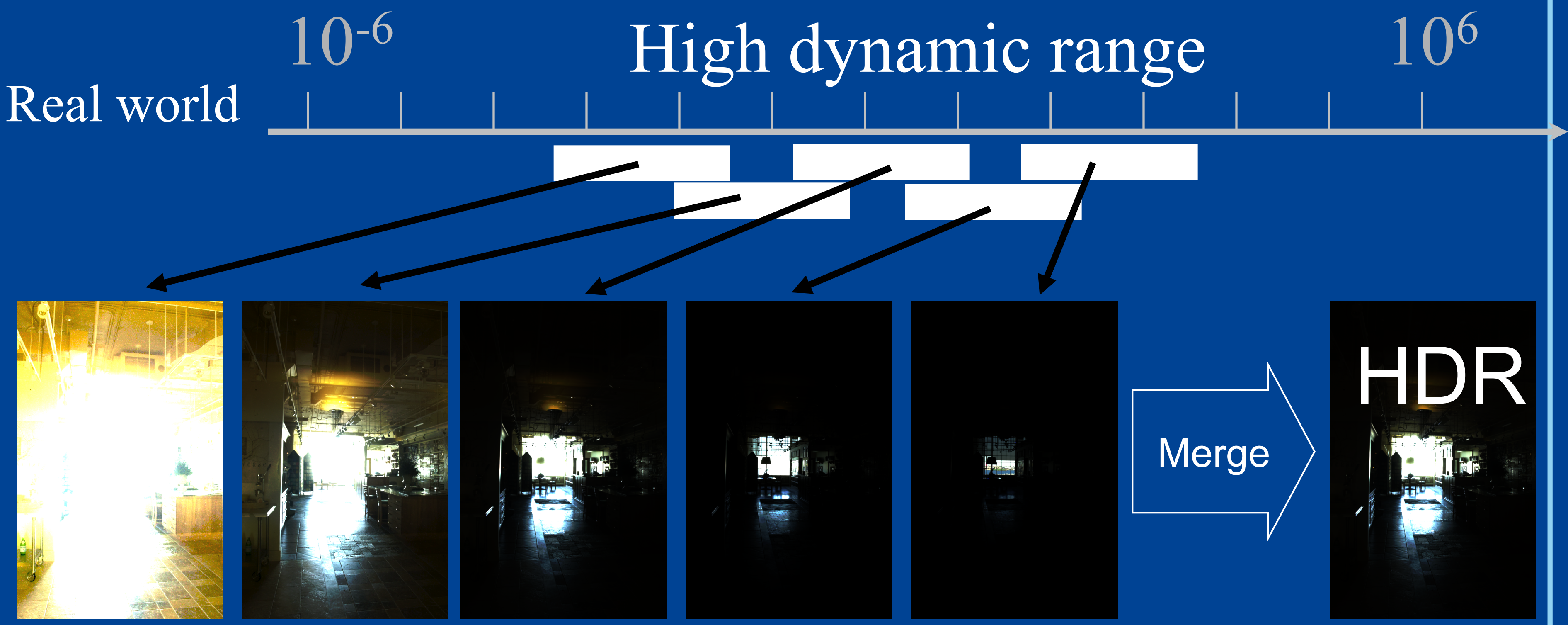
- Typically 1: 20 or 1:50
  - Black is ~ 50x darker than white



face tomorrow SIGGRAPH2007

Multiple exposure photography

- Merge multiple exposure to cover full range



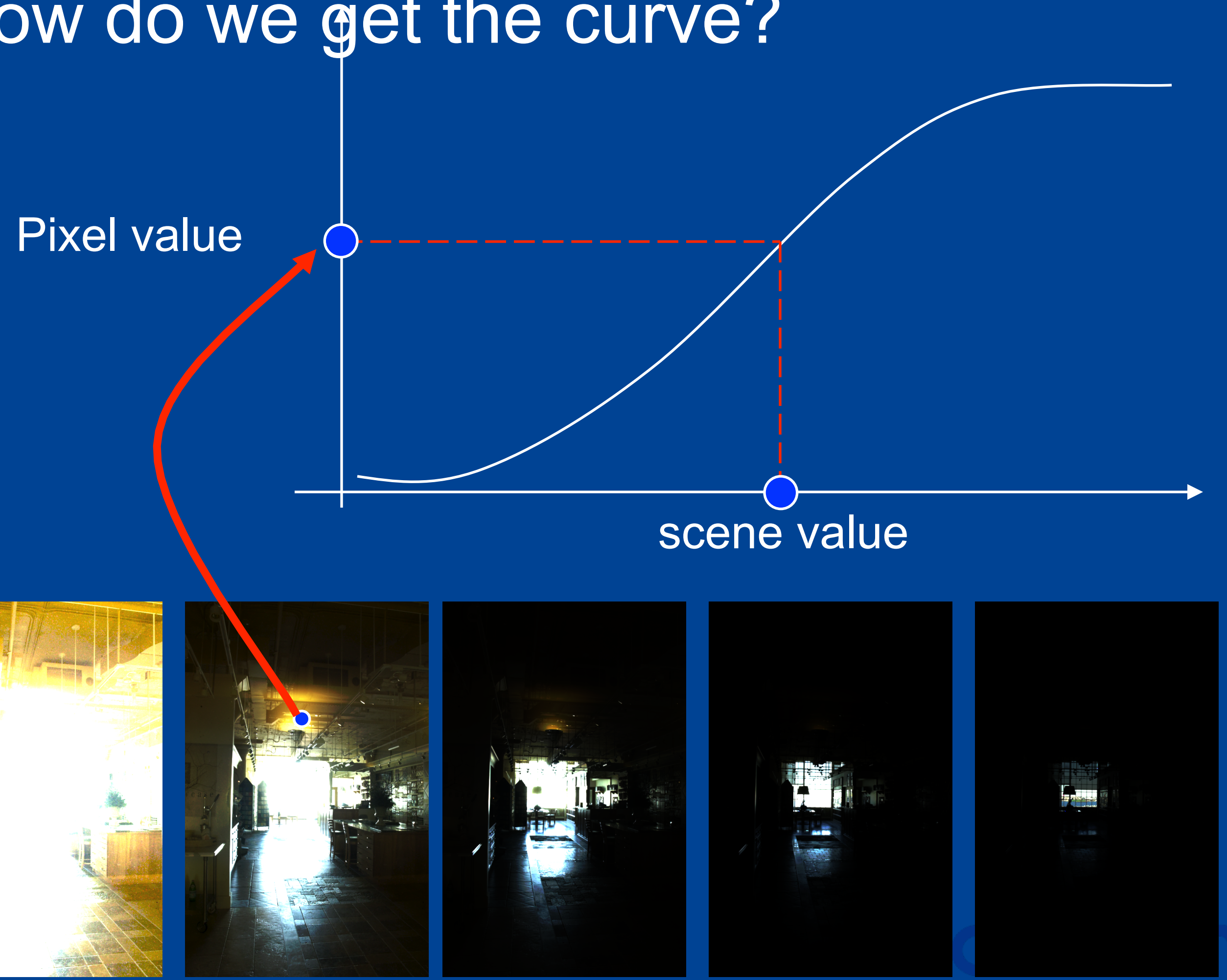
- We obtain one single image with floats per pixel
  - But we still can’t display it

HDR image using multiple exposure

- Given N photos at different exposure
- Recover a HDR color for each pixel

If we know the response curve

- Just look up the inverse of the response curve
- But how do we get the curve?

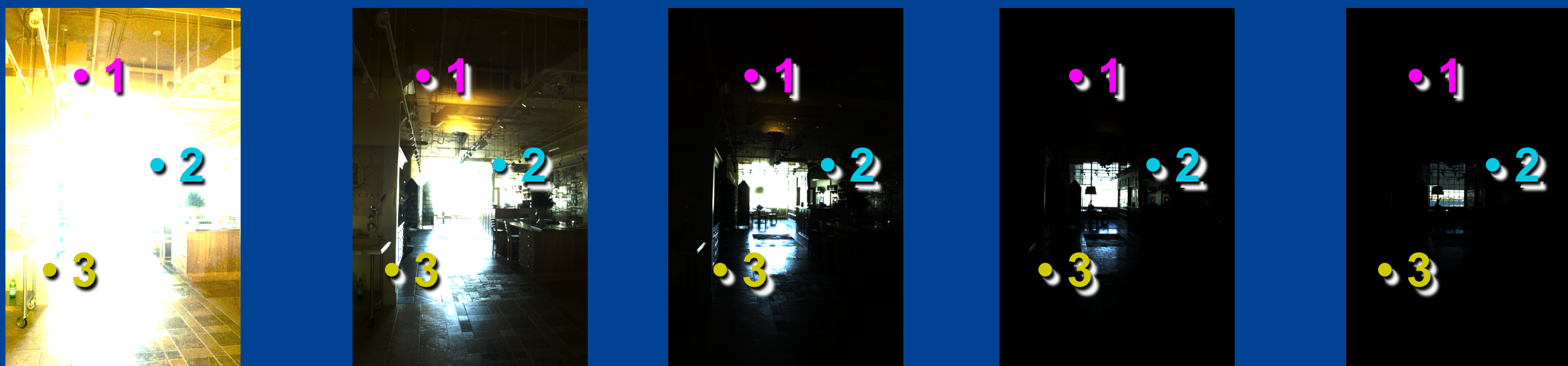


Calibrating the response curve

- Two basic solutions
  - Vary scene luminance and see pixel values
    - Assumes we control and know scene luminance
  - Vary exposure and see pixel value for one scene luminance
    - But note that we can usually not vary exposure more finely than by 1/3 stop
- Best of both:
  - Vary exposure
  - Exploit the large number of pixels

The Algorithm

Image series



$\Delta t =$ 10 sec | $\Delta t =$ 1 sec | $\Delta t =$ 1/10 sec | $\Delta t =$ 1/100 sec | $\Delta t =$ 1/1000 sec

Pixel Value Z = f(Exposure)

Exposure = Radiance $\times$ Δt

log Exposure = log Radiance + log Δt

Slide adapted from Alyosha Efros who borrowed it from Paul Debe

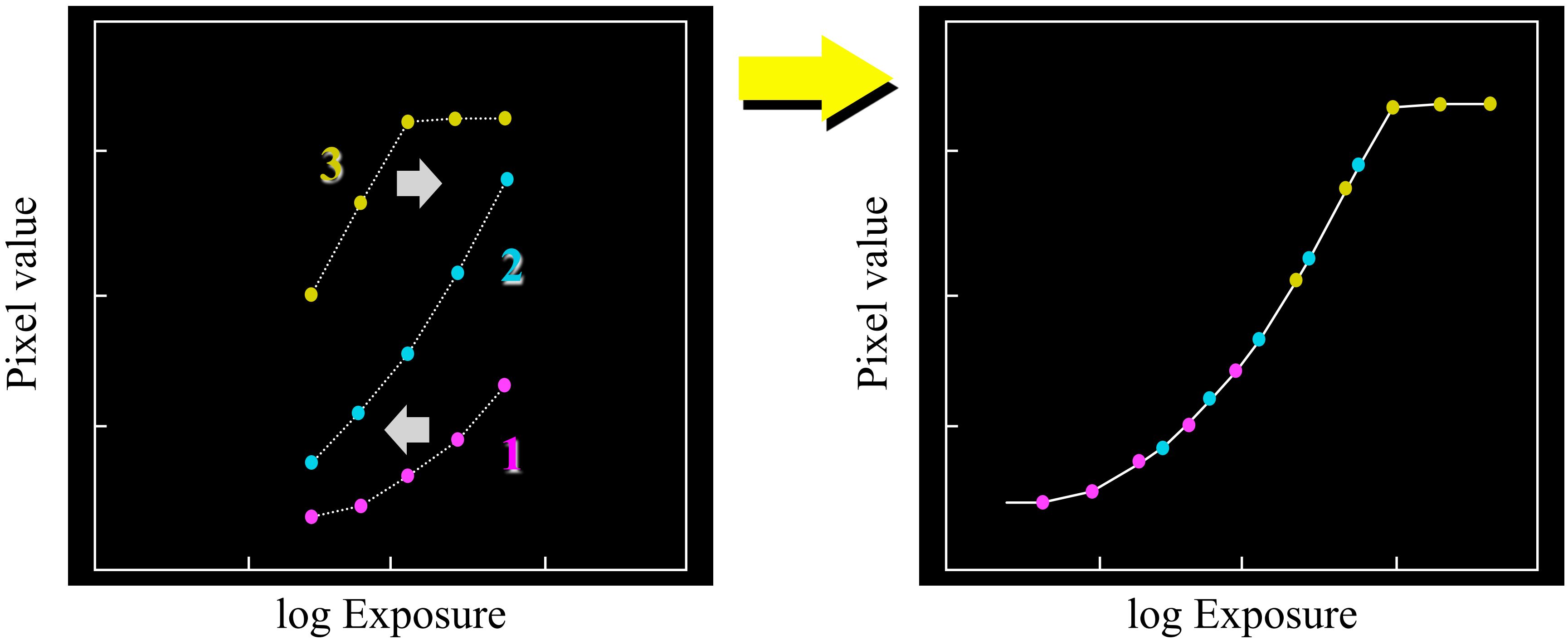
Δ t don't really correspond to pictures. Oh well.

Response curve

- **Exposure is unknown, fit to find a smooth curve**

Assuming unit radiance for each pixel

After adjusting radiances to obtain a smooth response curve



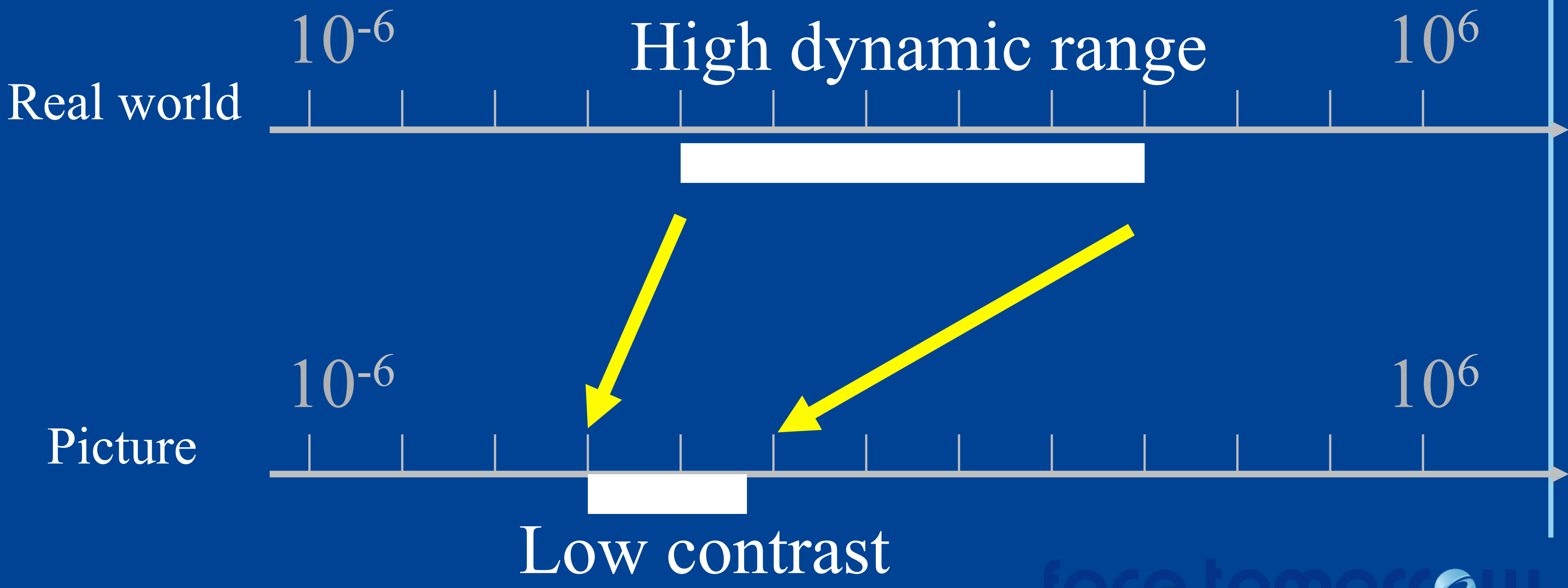
Slide stolen from Alyosha Efros who stole it from Paul Debevec

Reconstructed radiance map

Slide stolen from Fredo Durand who stole it from Alyosha Efros who stole it from Paul Debevec

Problem: Contrast reduction

- Match limited contrast of the medium
- Preserve details



face tomorrow
SIGGRAPH2007

Tone mapping

- Input: high-dynamic-range image
  - (floating point per pixel)

Naïve technique

- Scene has *1:10,000* contrast, display has *1:100*
- Simplest contrast reduction?

Naïve: Gamma compression

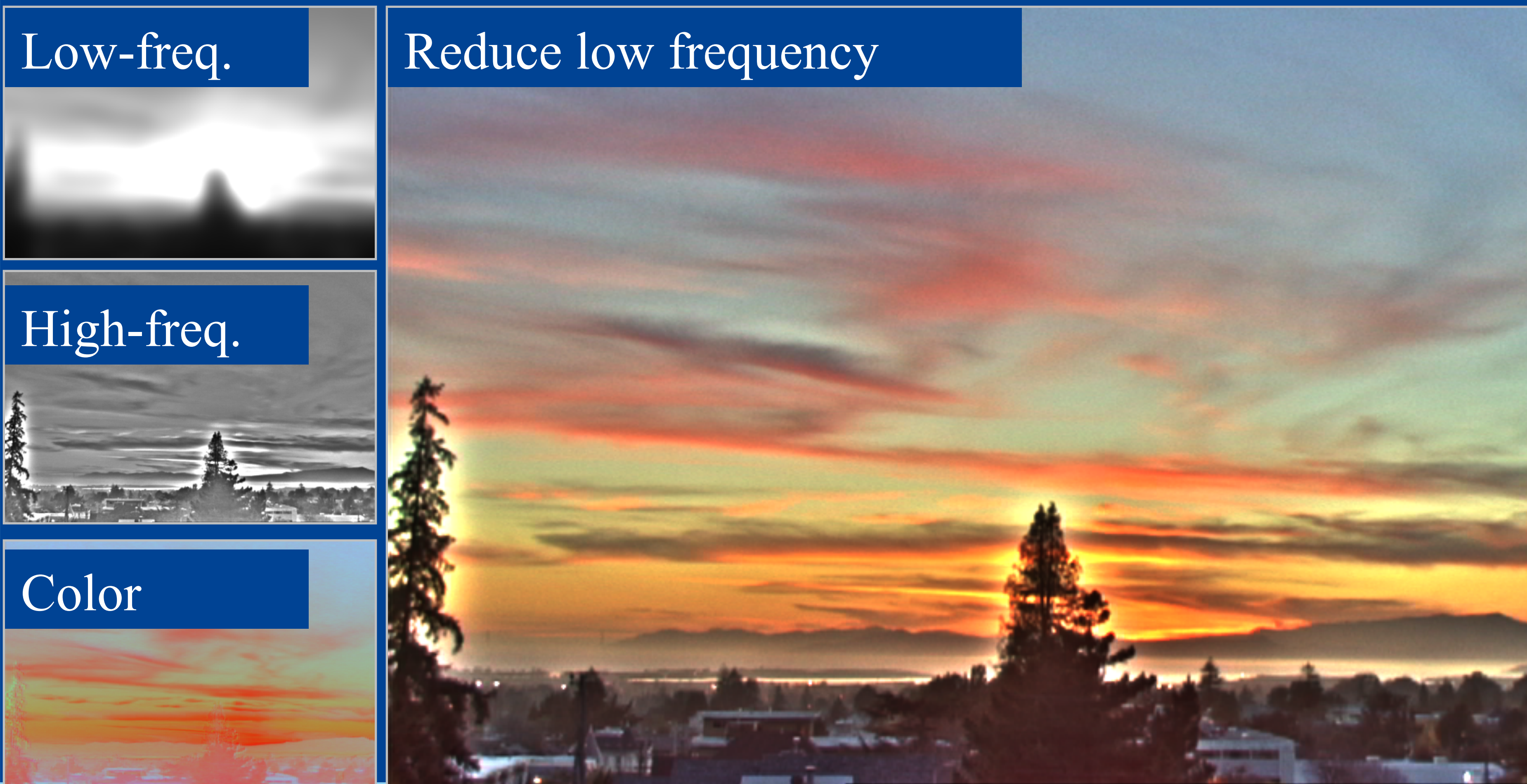
- $X \rightarrow X^{\gamma}$ (where γ=0.5 in our case)
- But… colors are washed-out. Why?

Gamma compression on intensity

- Colors are OK,
  but details (intensity high-frequency) are blurred

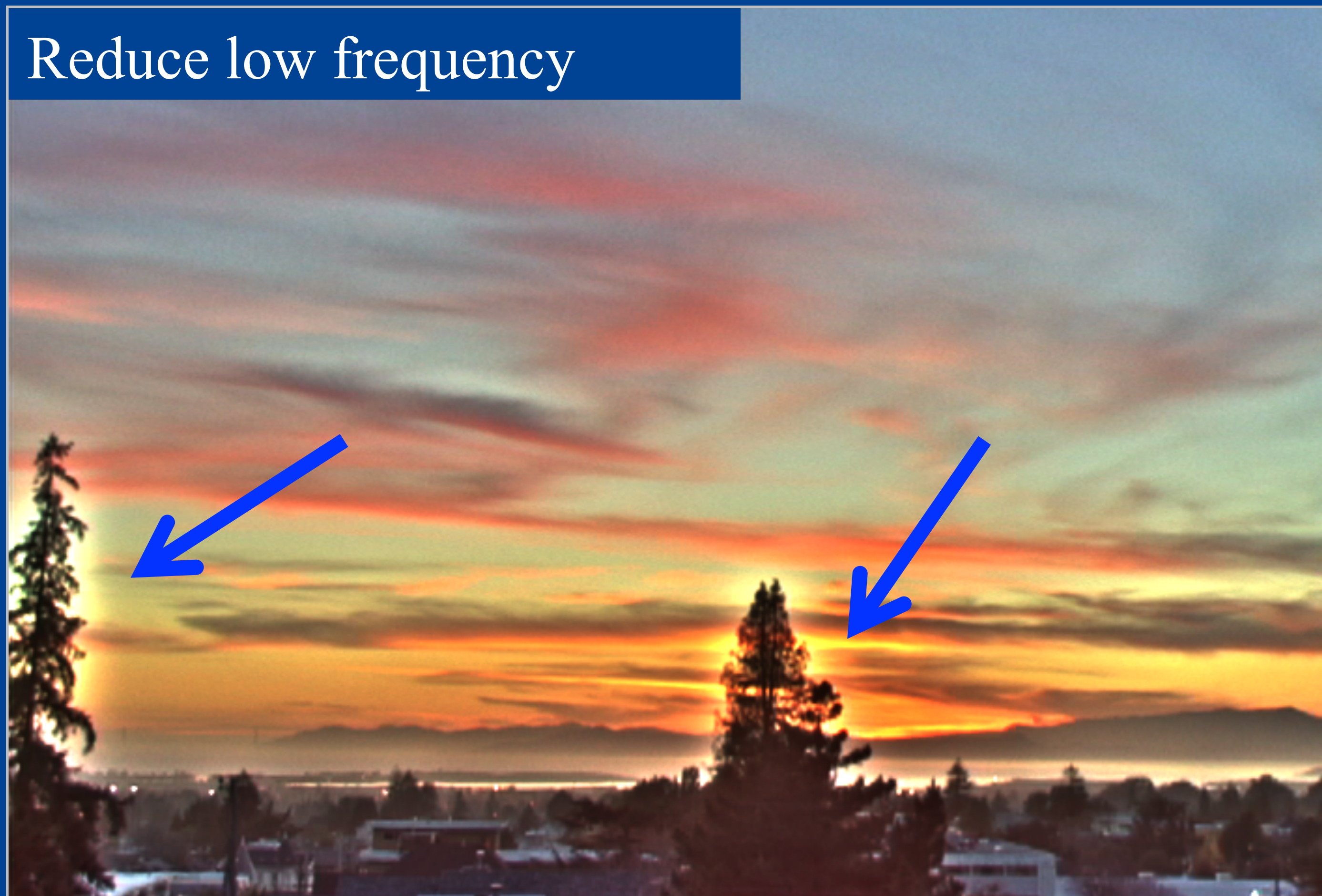
Oppenheim 1968, Chiu et al. 1993

- Reduce contrast of low-frequencies (log domain)
- Keep high frequencies



The halo nightmare

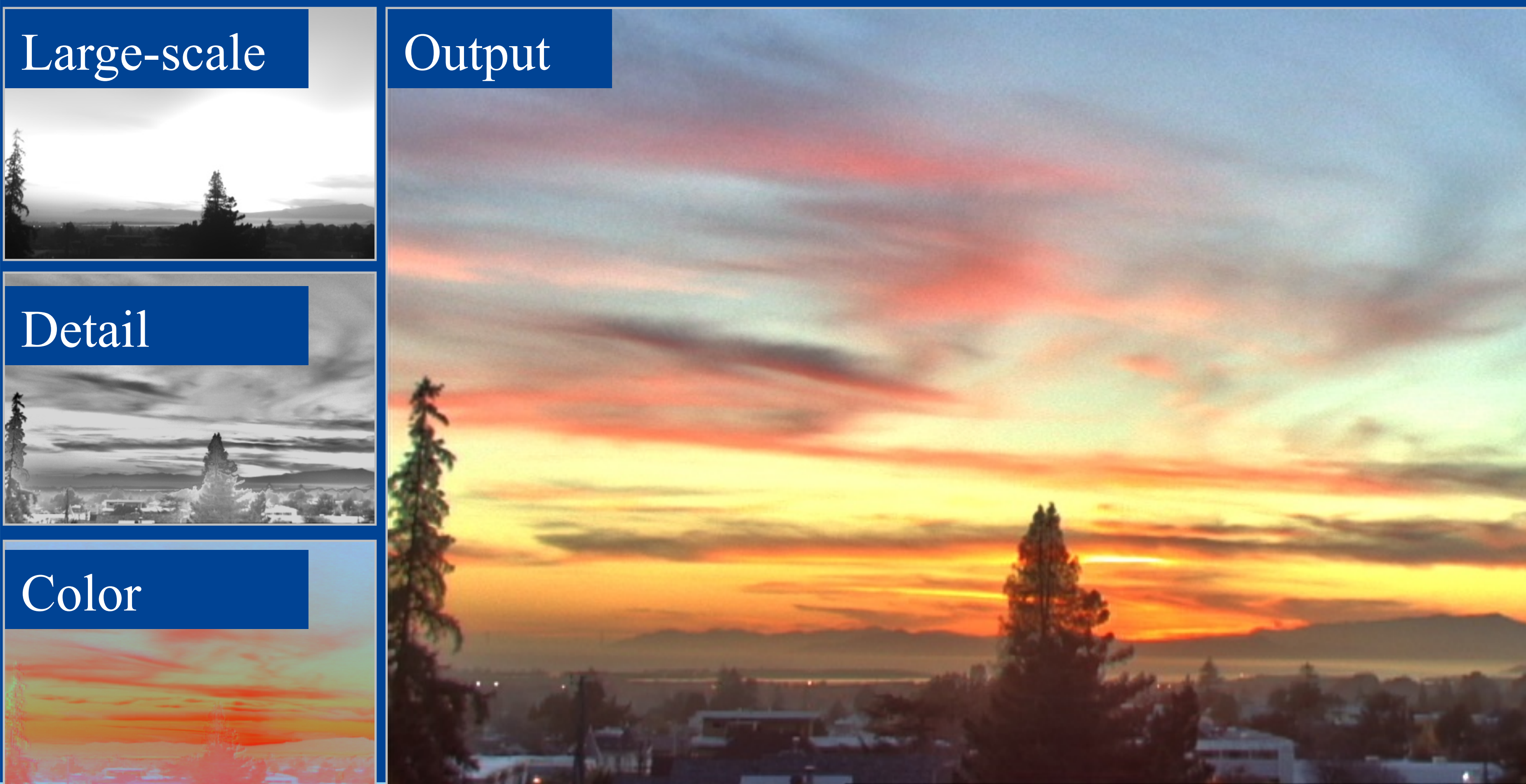
- For strong edges
- Because they contain high frequency



Bilateral filtering to the rescue

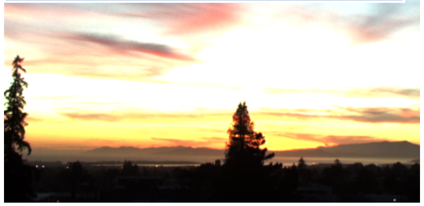
- Large scale = bilateral (log intensity)
- Detail = residual

[Durand & Dorsey 2002]



Contrast reduction

Input HDR image

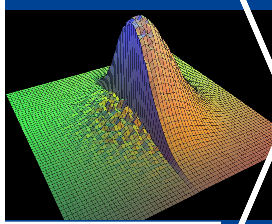


Contrast
too high!

Contrast reduction

face tomorrow
SIGGRAPH2007

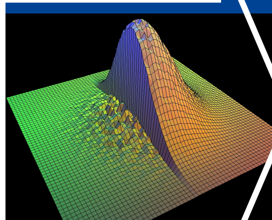
Contrast reduction



Bilateral
Filter
(in log domain!)

Spatial sigma: 2% image size
Range sigma: 0.4 (in log 10)

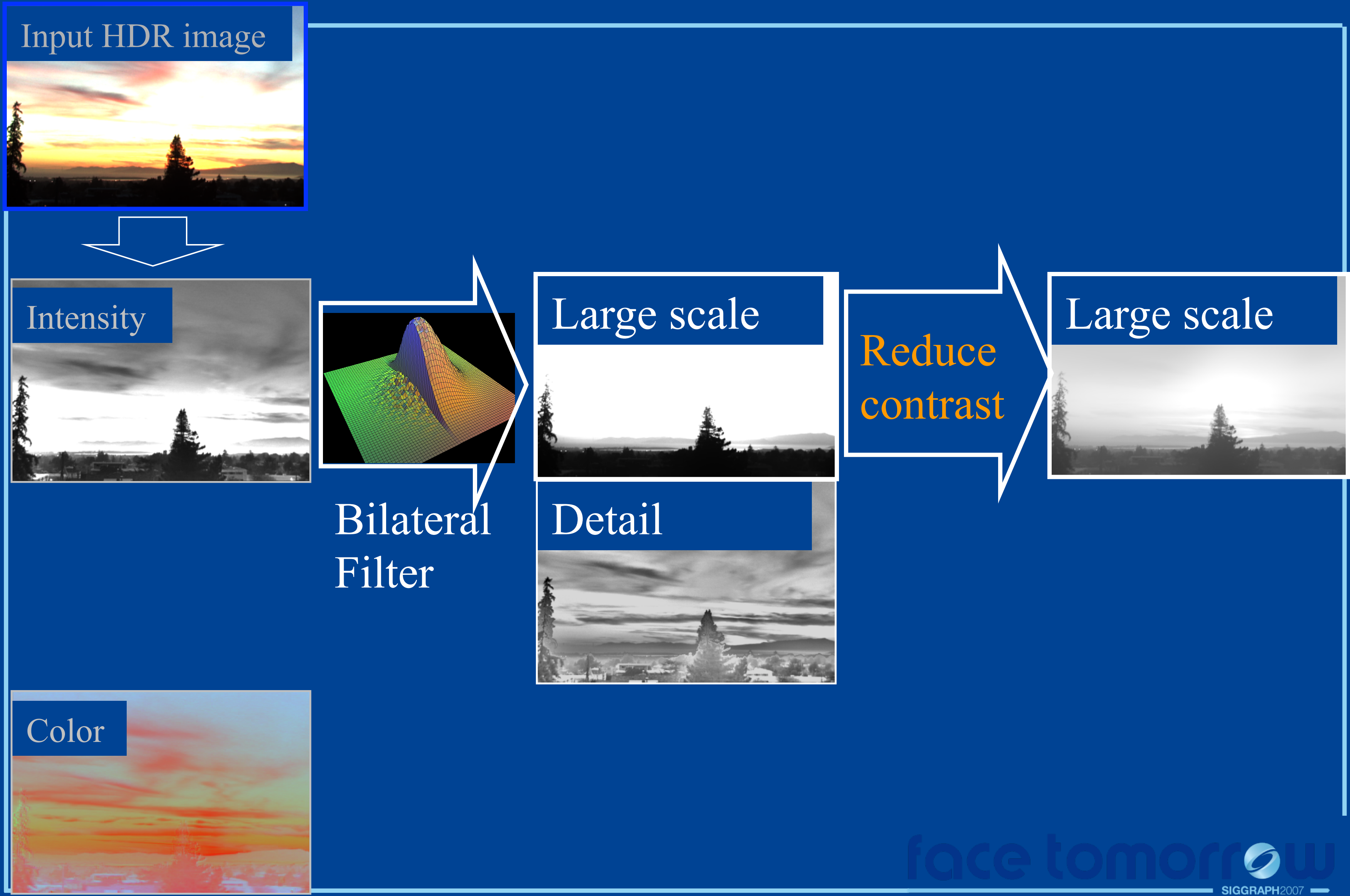
Contrast reduction



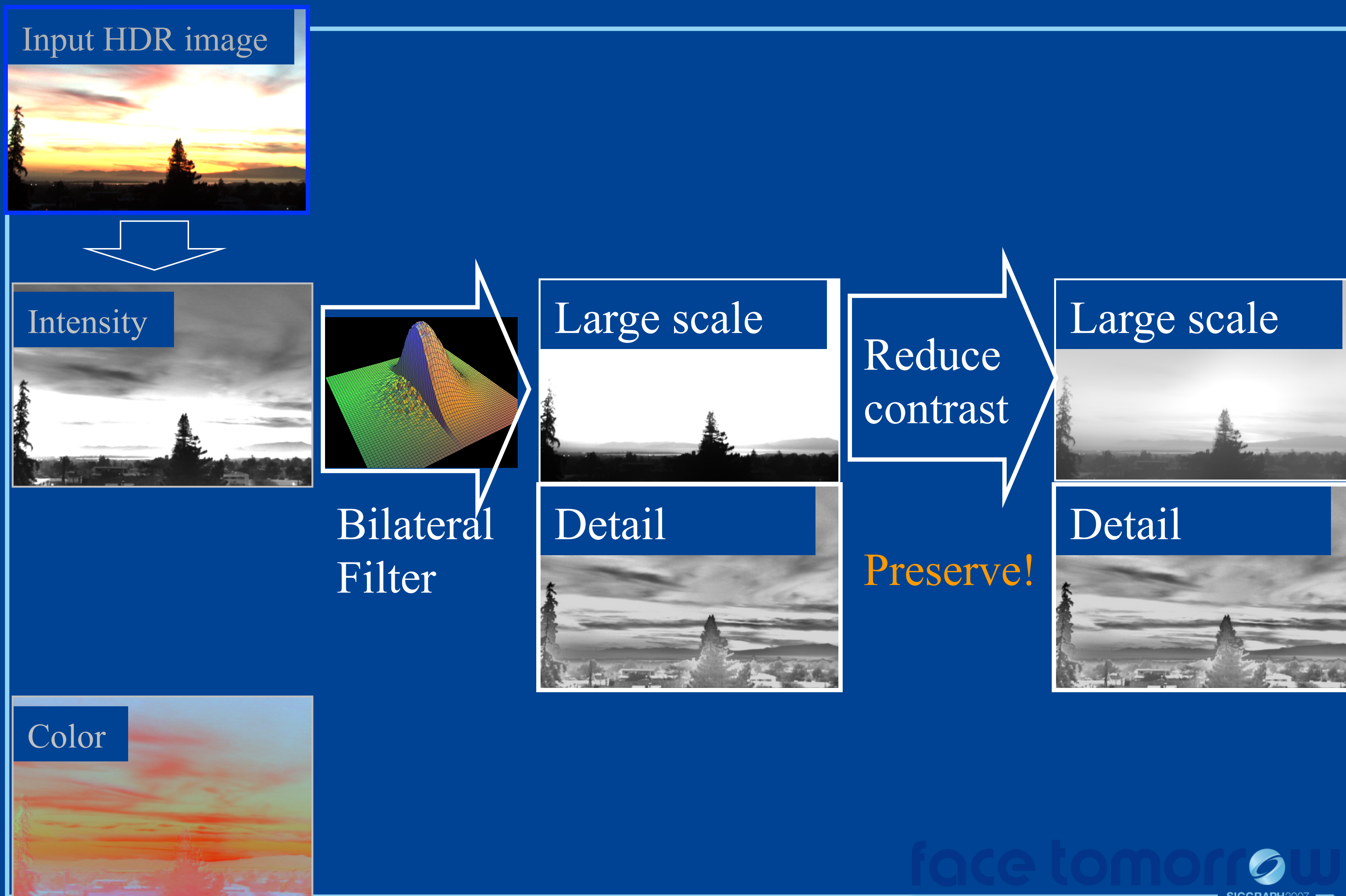
Bilateral Filter

Detail = log intensity –large scale (residual)

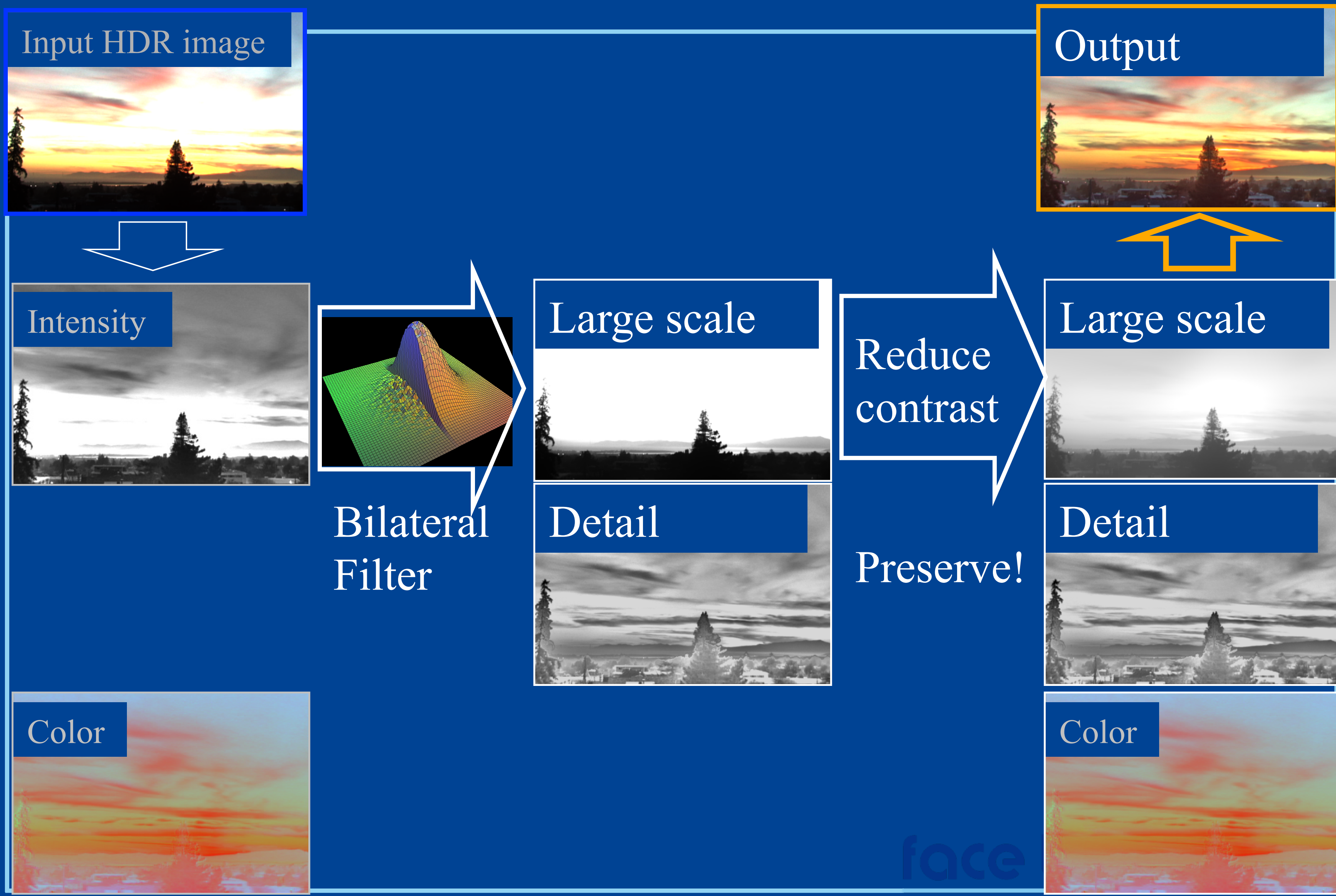
Contrast reduction



Contrast reduction



Contrast reduction



Contrast reduction in log domain

- Set target large-scale contrast (e.g. $\log_{10} 10$)
  - In **linear** output, we want 1:10 contrast for large scale
- Compute range of input large scale layer:
  - largeRange = max(inLogLarge) – min (inLogLarge)
- Scale factor $k = \log_{10}(10)$ / largeRange
- Normalize so that the biggest value is 0 in log

$$\boxed{\begin{aligned}\text{outLog} = {} & \text{inLogDetail} + \\ & \text{inLogLarge} * k - \max(\text{inLogLarge})\end{aligned}}$$

face tomorrow SIGGRAPH2007