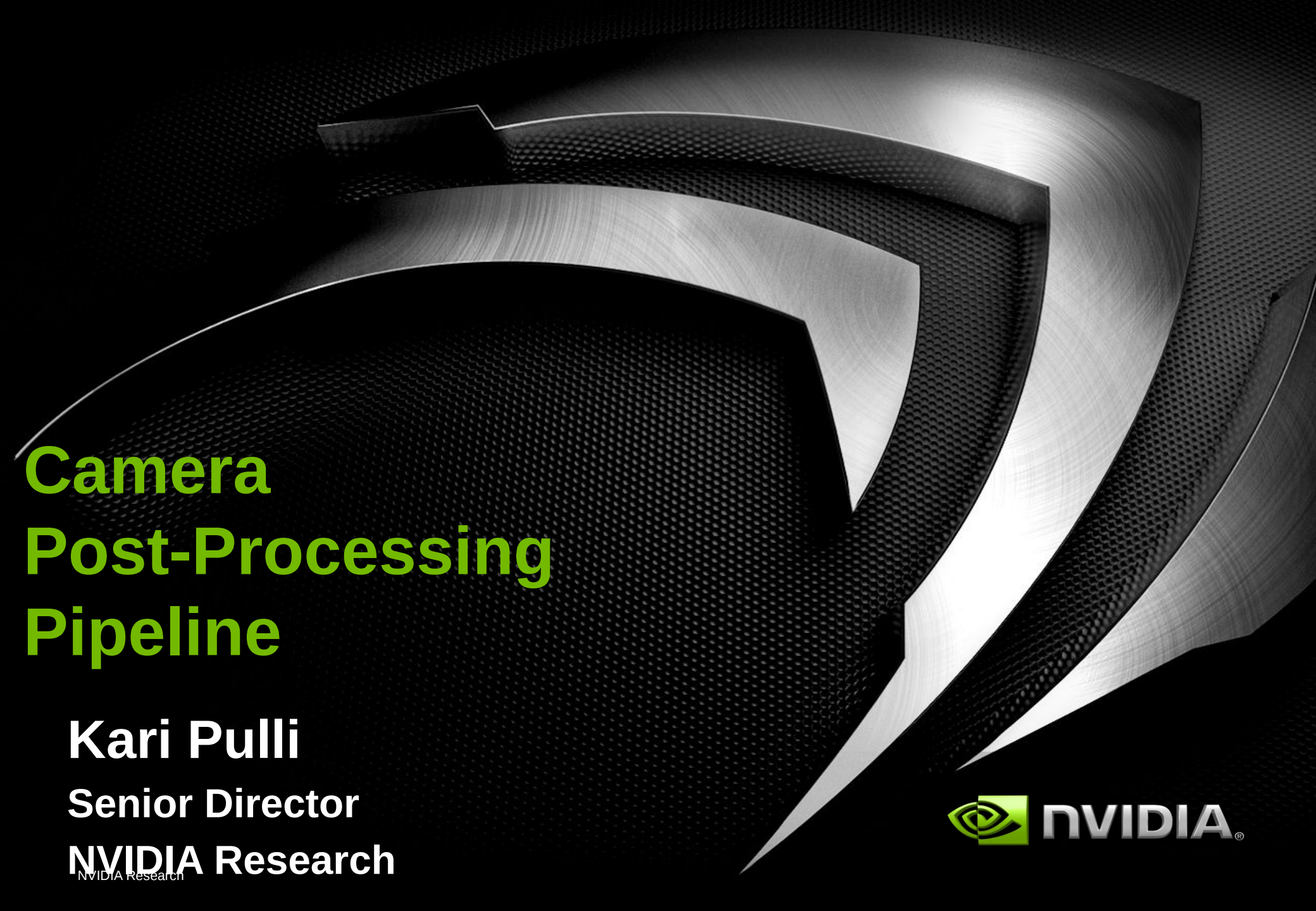


Camera Post-Processing Pipeline

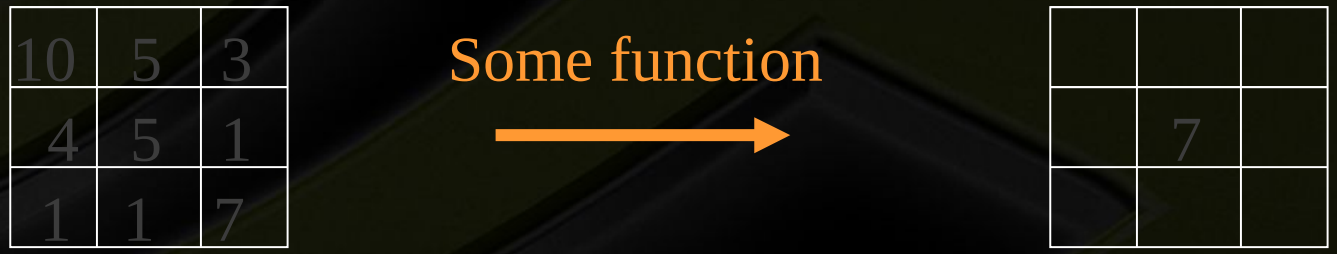
Kari Pulli
Senior Director
NVIDIA Research

Topics

- Filtering
  - blurring
  - sharpening
  - bilateral filter
- Sensor imperfections (PNU, dark current, vignetting, …)
- ISO (analog – digital conversion with amplification)
- Demosaicking and denoising
- Gamma correction
- Color spaces
- Response curve
- White balance
- JPEG compression

What is image filtering?

- **Modify the pixels in an image based on some function of a local neighborhood of the pixels**



Local image data

Modified image data

Linear functions

- **Simplest: linear filtering**
  - **replace each pixel by a linear combination of its neighbors**
- **The prescription for the linear combination is called the "convolution kernel"**

Local image data

kernel

Modified image data

Convolution

$$f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l]\, g[k,l]$$

Linear filtering (warm-up slide)

original

?

Linear filtering (warm-up slide)

original

Filtered
(no change)

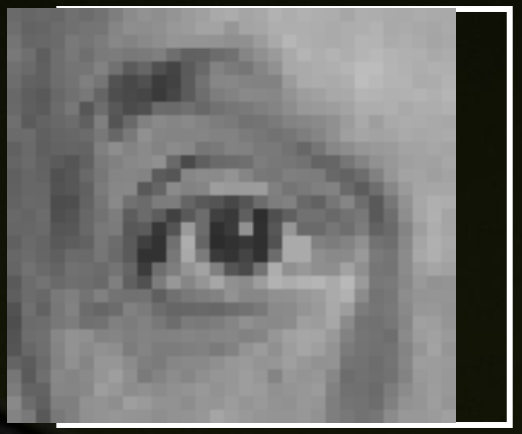
Linear filtering

original

?

Shift

original



shifted

Linear filtering

original

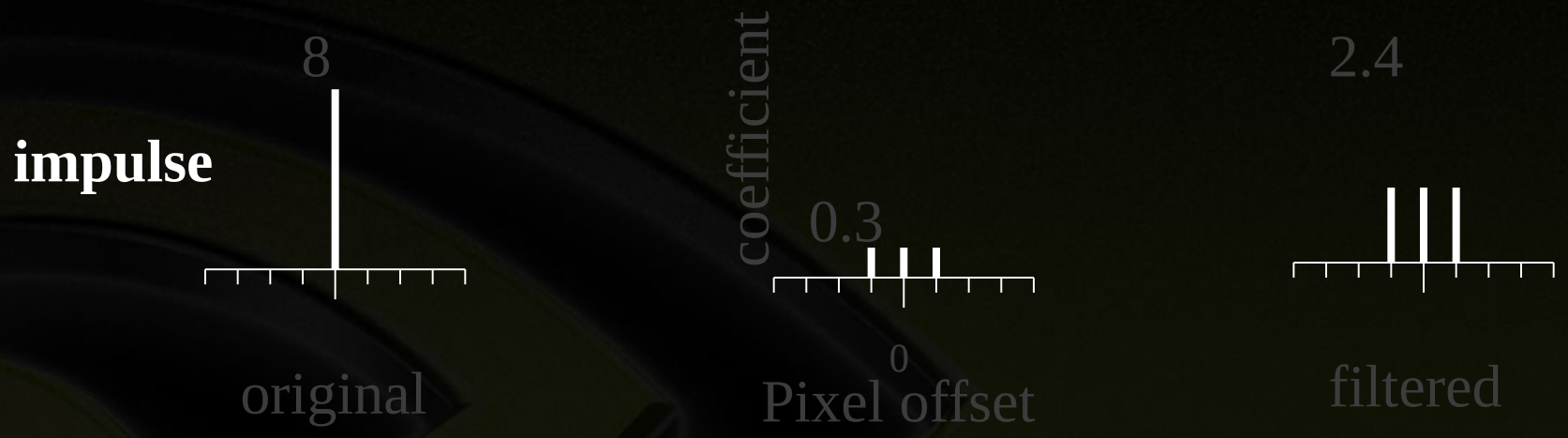
?

Blurring

original

blurred
(filter applied in
both dimensions)

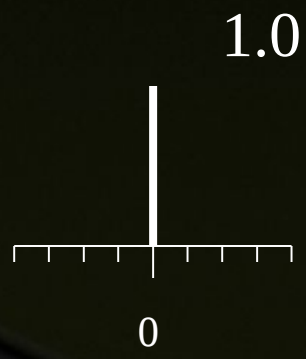
Blur examples



Blur examples

Linear filtering (warm-up slide)

original



?

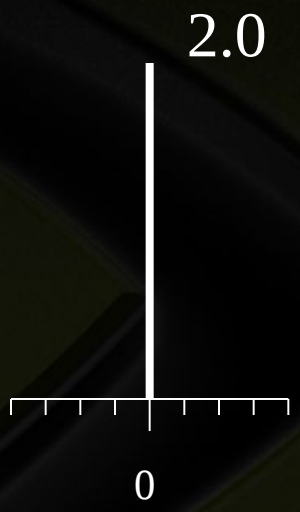
Linear filtering (no change)

original

Filtered
(no change)

Linear filtering

original



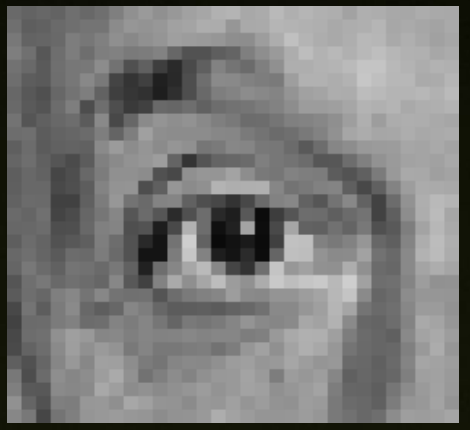
(remember blurring)

original

Blurred
(filter applied in
both dimensions)

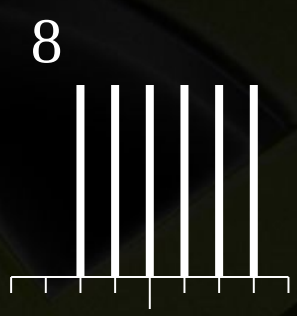
Sharpening

original



Sharpened original

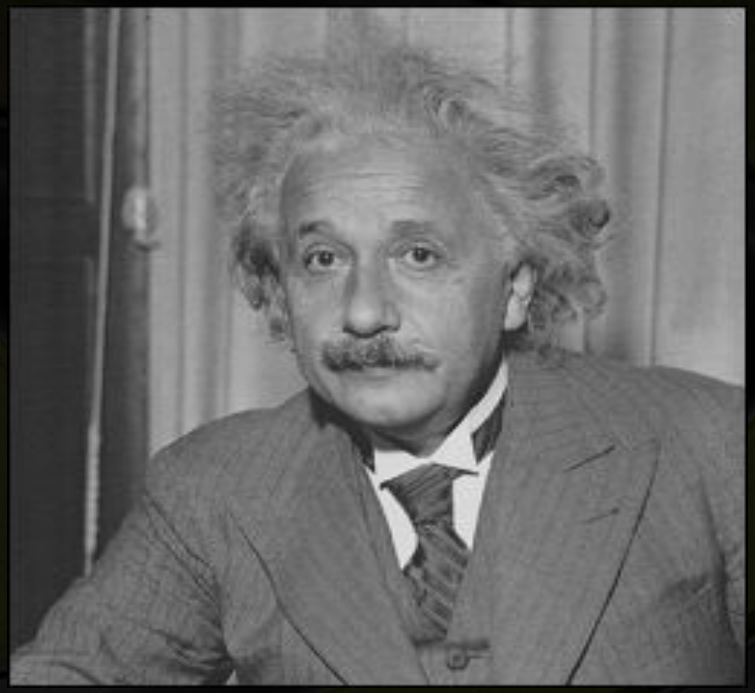
Sharpening example



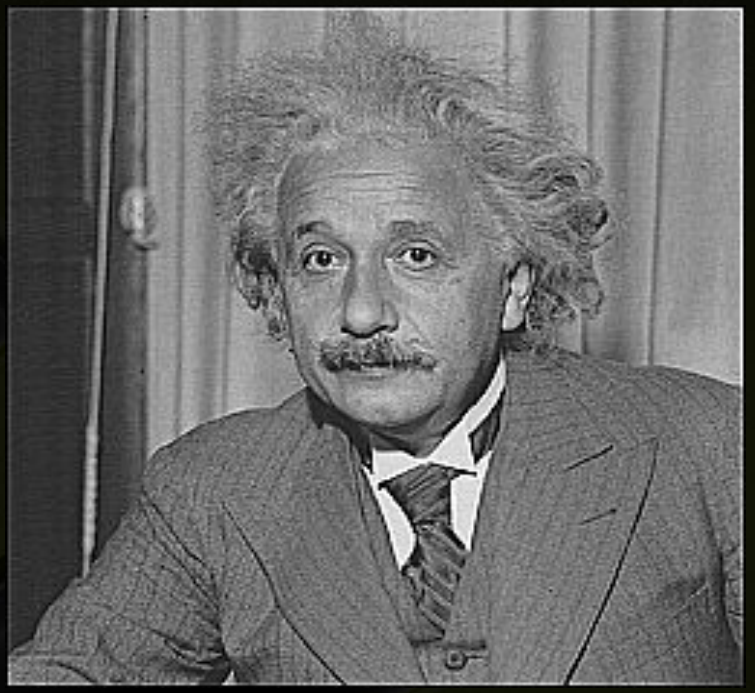
original

sharpened
(differences are accentuated;
constant areas are left untouched)

Sharpening



before



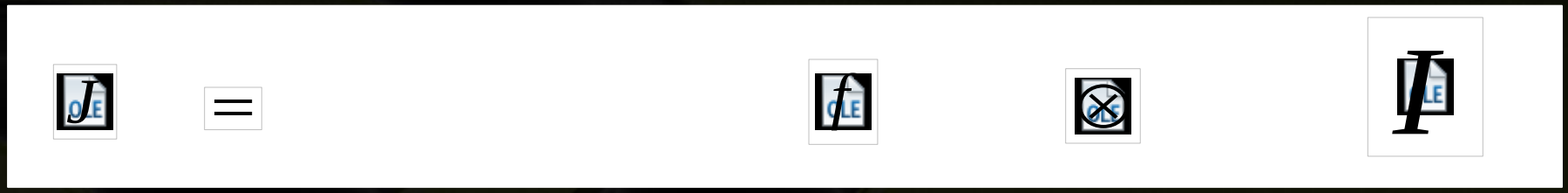
after

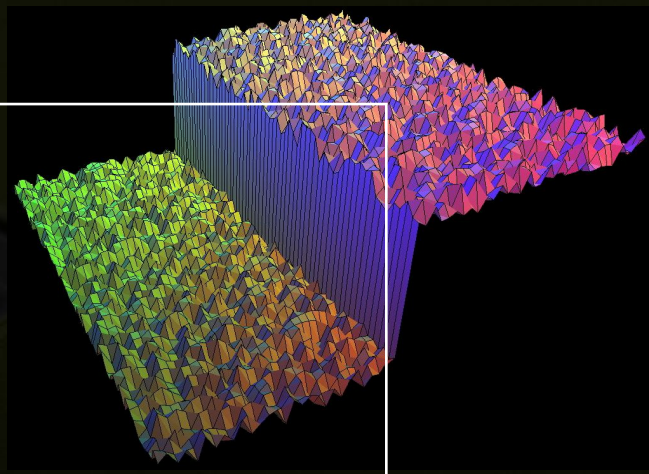
Bilateral filter

- Tomasi and Manduchi 1998
  http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf
- 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

Start with Gaussian filtering

- Here, input is a step function + noise



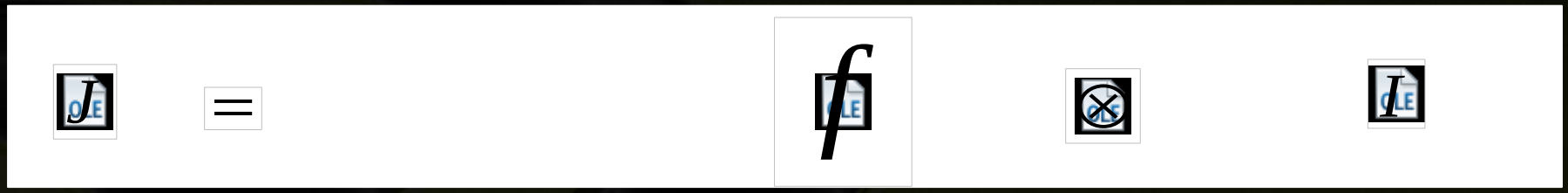


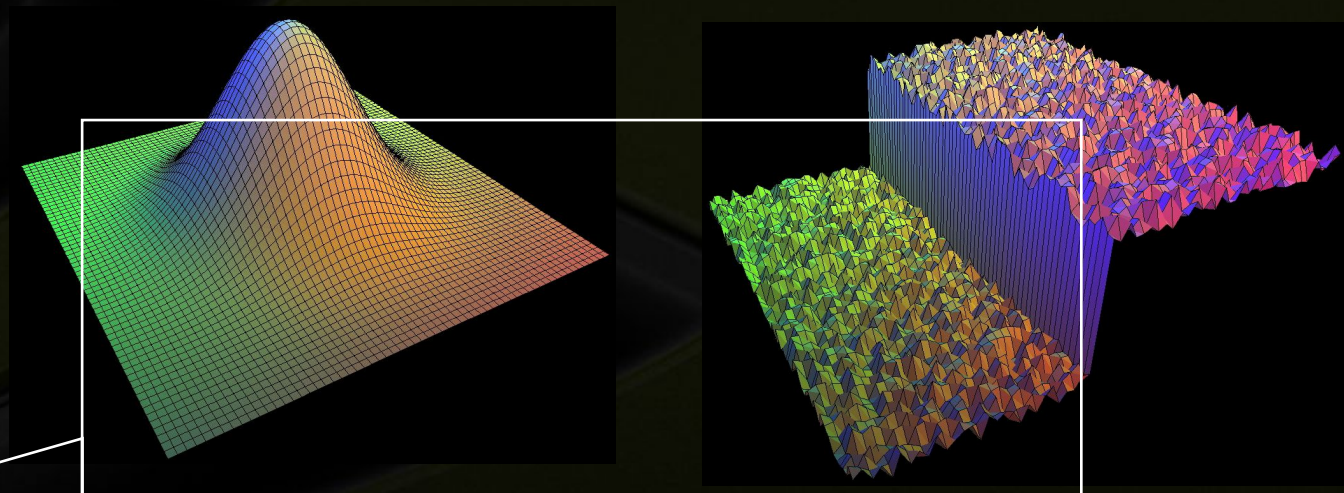
output

input

Start with Gaussian filtering

- Spatial Gaussian f



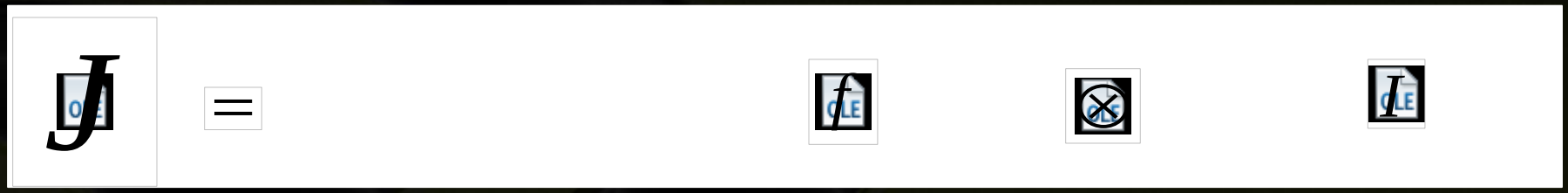


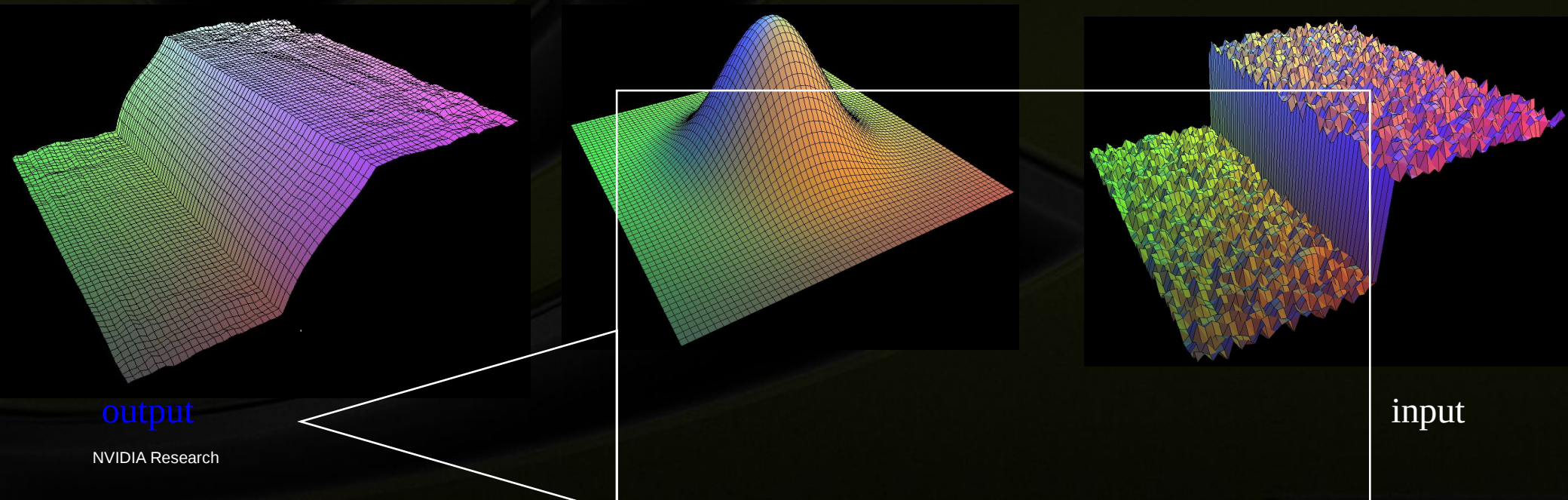
output

input

Start with Gaussian filtering

- Output is blurred

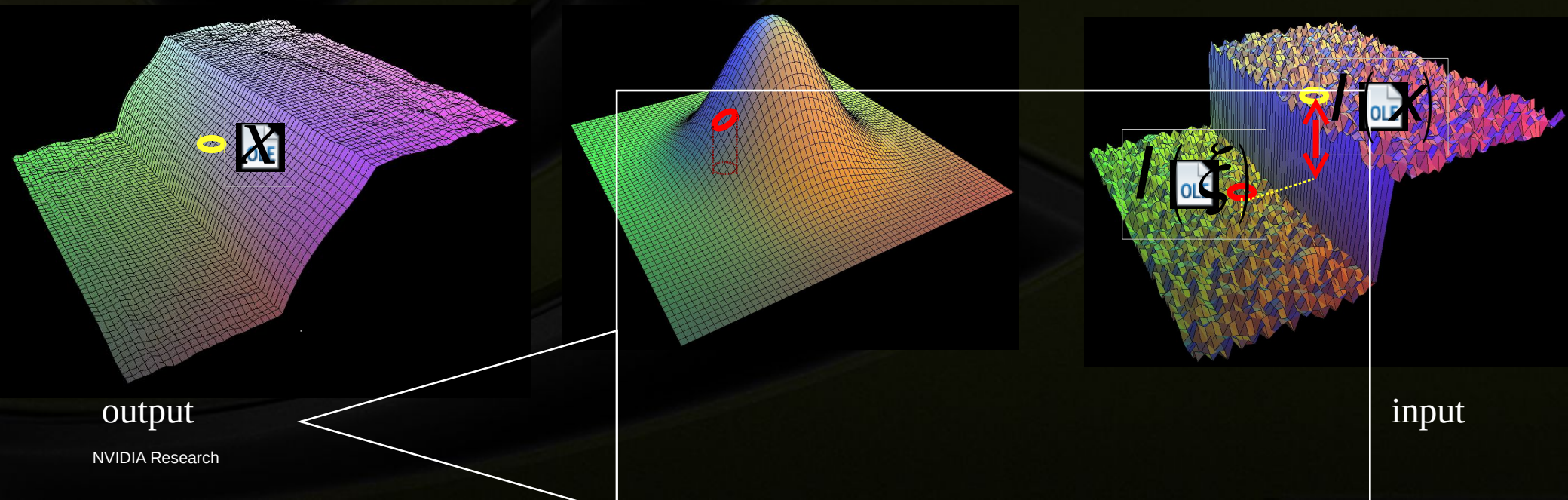




The problem of edges

- Weight f(x, ξ) depends on distance from ξ to x
- Here, $I(\xi)$ "pollutes" our estimate $J(x)$
  - It is too different

$$J(x) = \sum_{\xi} f(x,\xi)\, I(\xi)$$

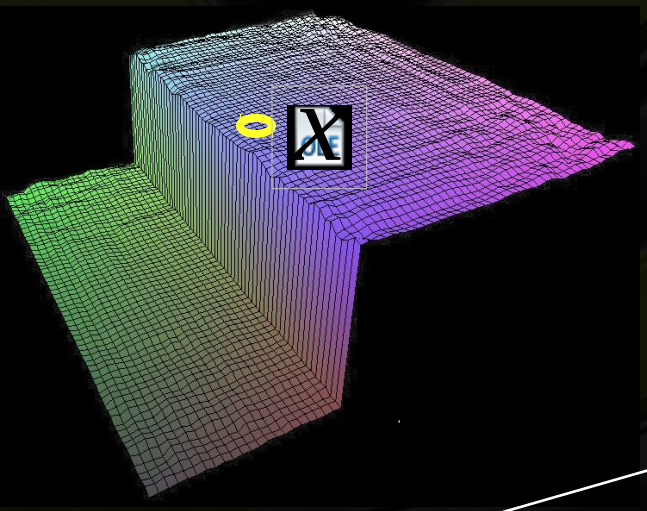


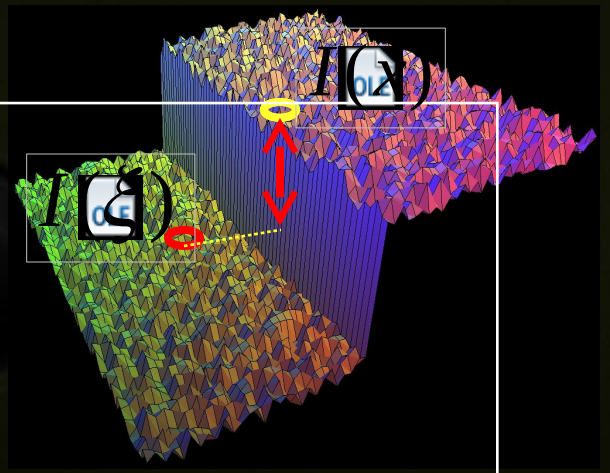
Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- Penalty g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi)\, g(I(\xi) - I(x))\, I(\xi)$$





output

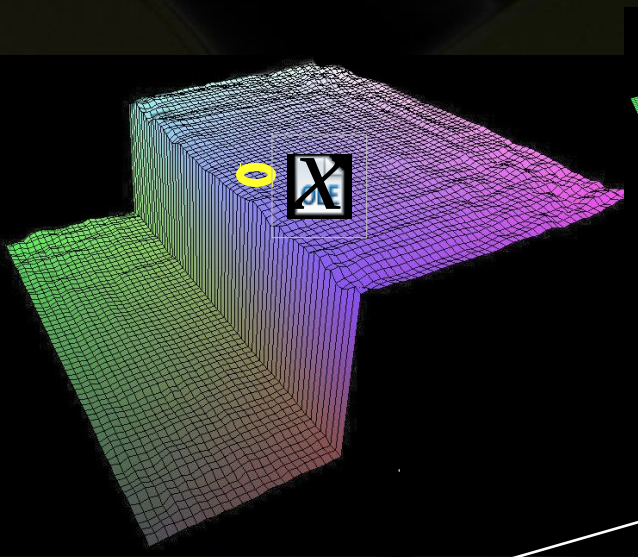
input

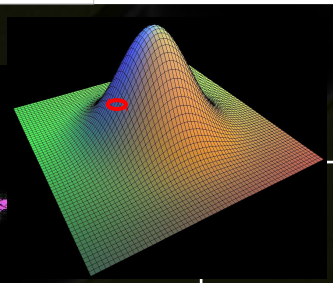
Bilateral filtering

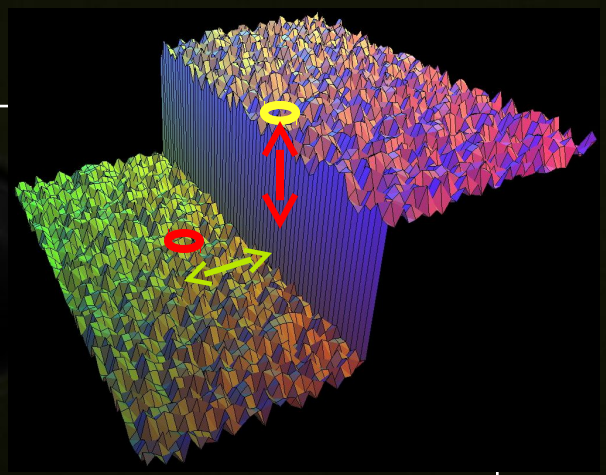
[Tomasi and Manduchi 1998]

- Spatial Gaussian f

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi)\, g(I(\xi) - I(x))\, I(\xi)$$







output

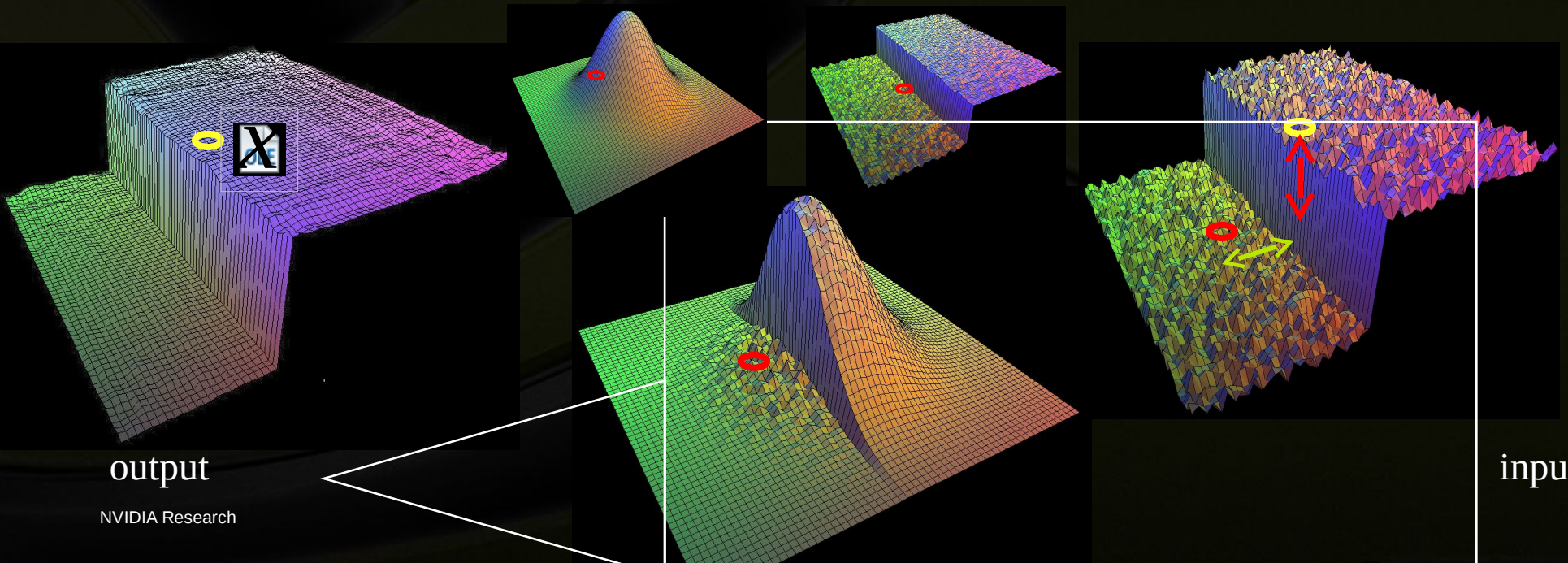
input

Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian f
- Gaussian g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi)\, g(I(\xi) - I(x))\, I(\xi)$$

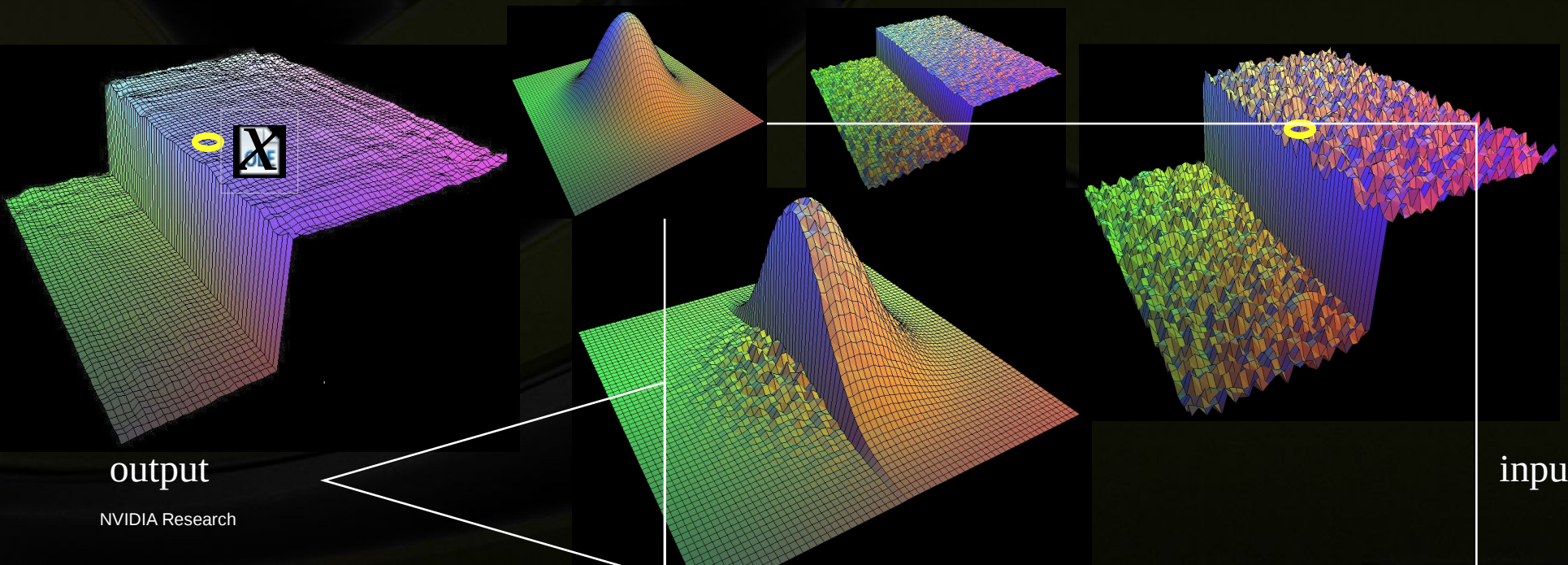


Normalization factor

[Tomasi and Manduchi 1998]

- k(x) = $\sum_{\xi} f(x,\xi)\, g(I(\xi) - I(x))$

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\, g(I(\xi) - I(x))\, I(\xi)$$

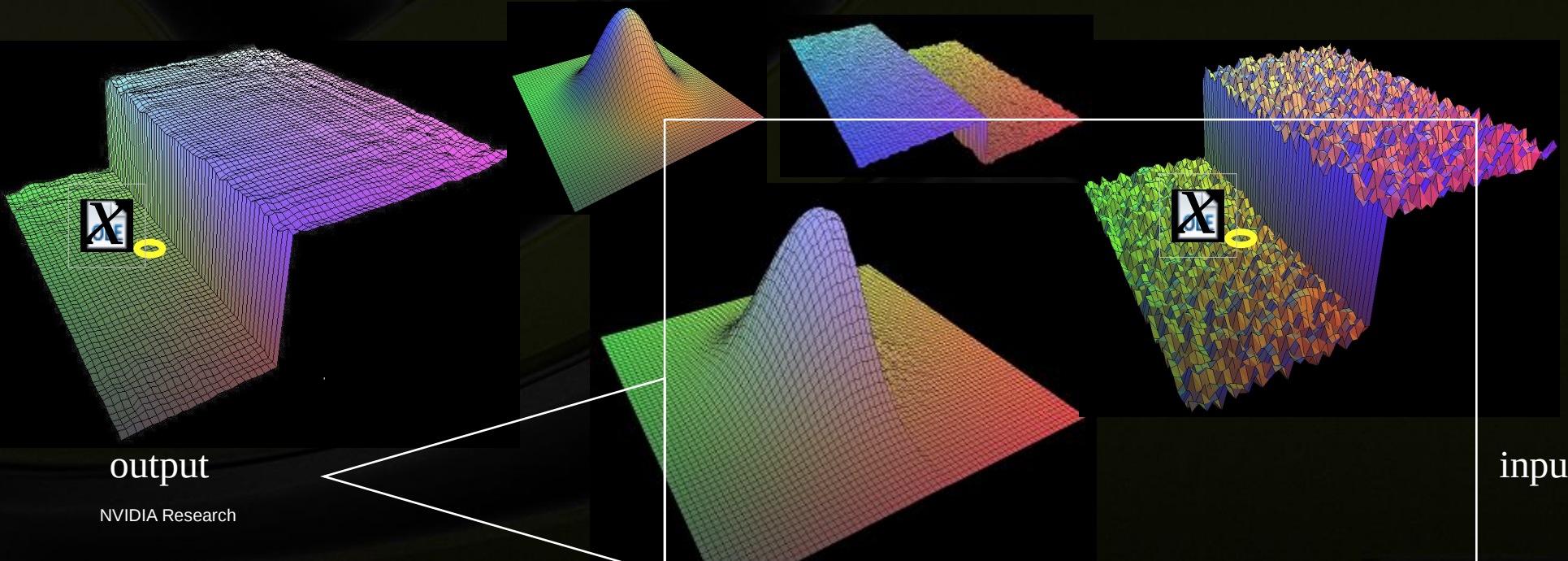


Bilateral filtering is non-linear

[Tomasi and Manduchi 1998]

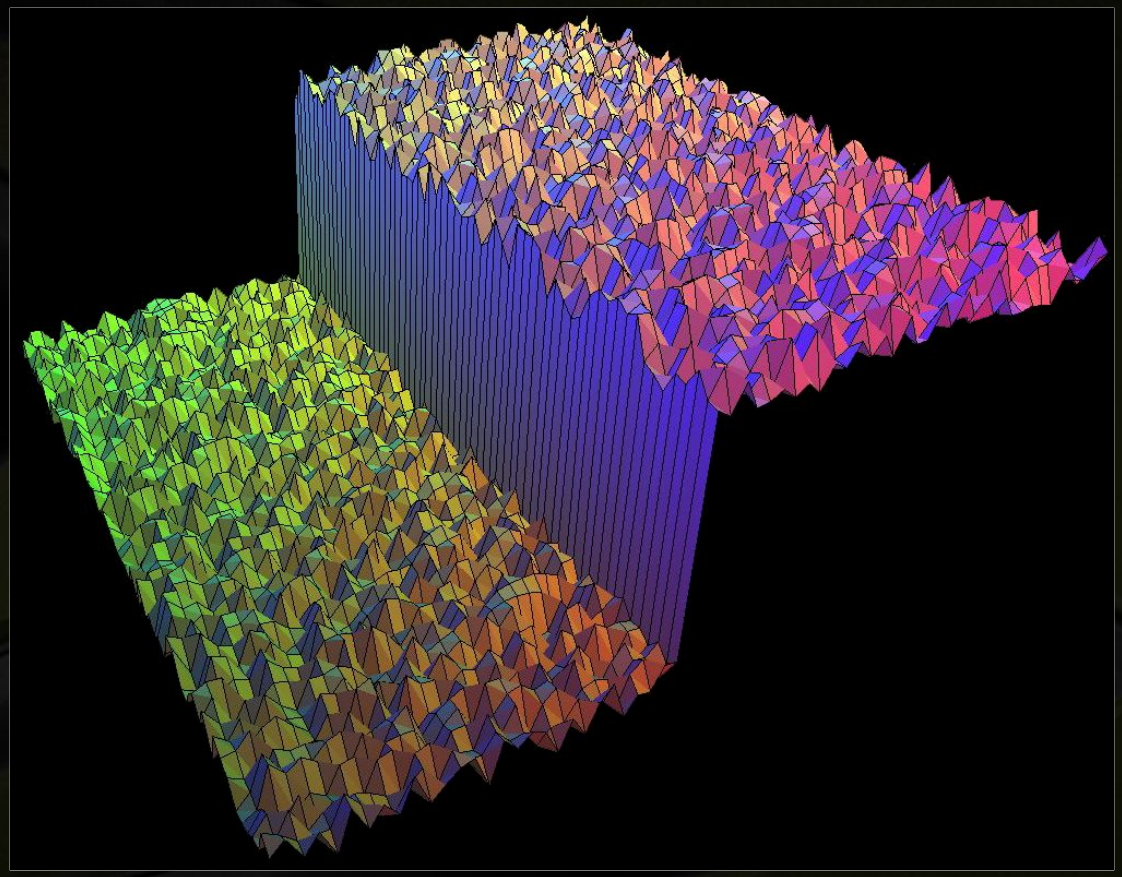
- The weights are different for each output pixel

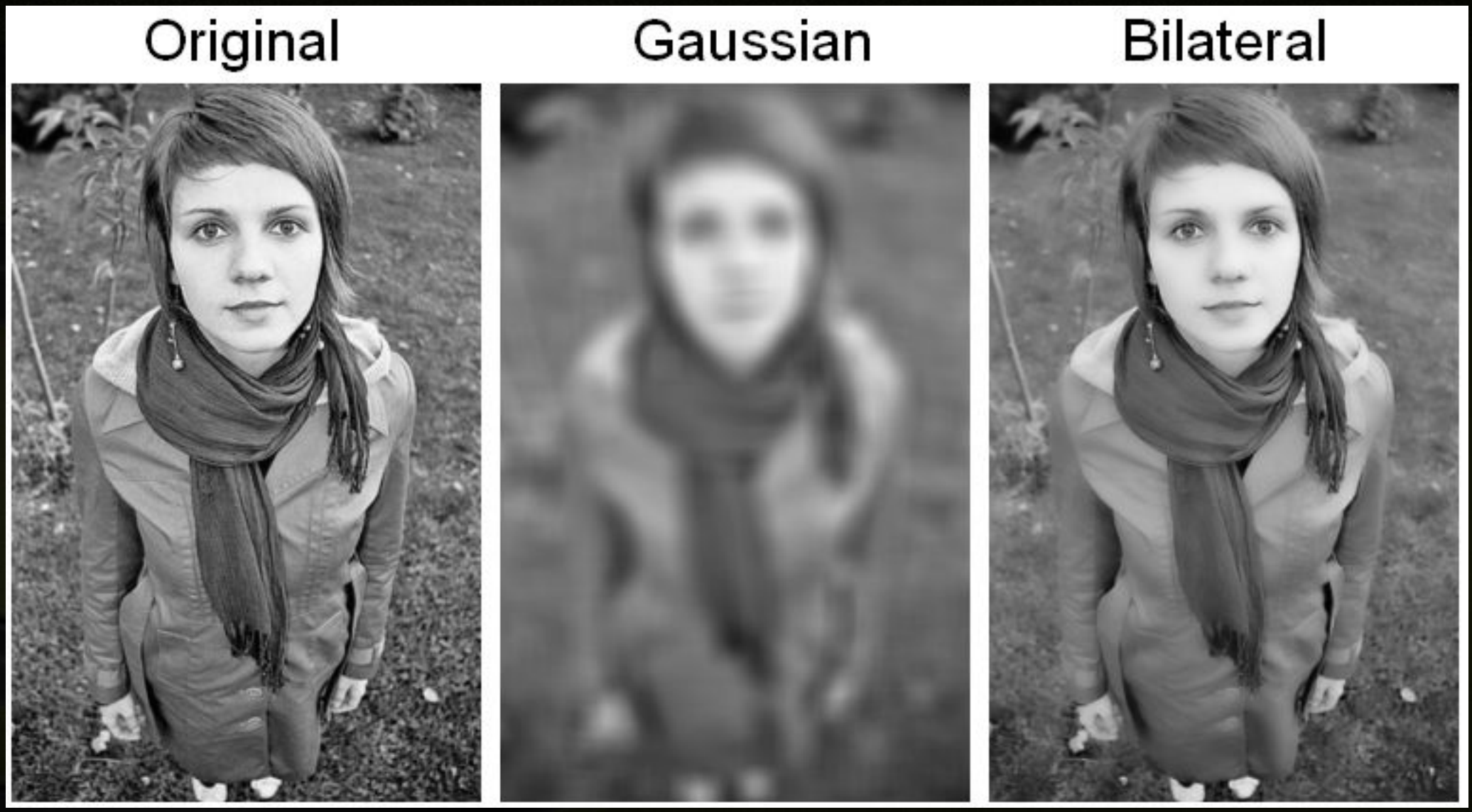
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi)\, g(I(\xi) - I(x))\, I(\xi)$$



Other view

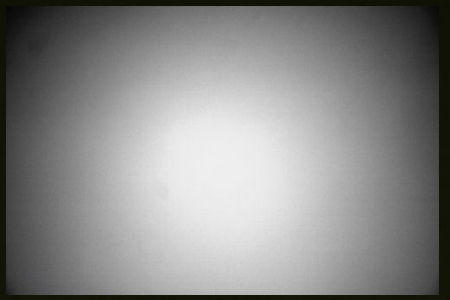
- The bilateral filter uses the 3D distance

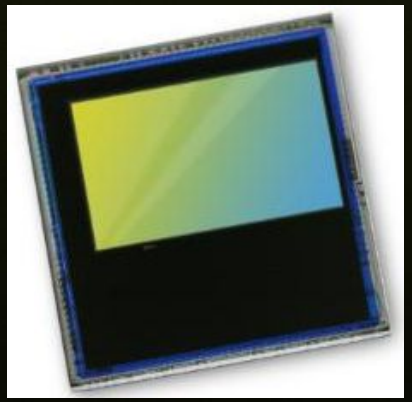



Original
Gaussian
Bilateral

From “raw-raw” to RAW

- Pixel Non-Uniformity
  - each pixel in a CCD has a slightly different sensitivity to light, typically within 1% to 2% of the average signal
  - can be reduced by calibrating an image with a flat-field image
  - flat-field images are also used to eliminate the effects of vignetting and other optical variations
- Stuck pixels
  - some pixels are turned always on or off
  - identify, replace with filtered values
- Dark floor
  - temperature adds noise
  - sensors usually have a ring of covered pixels around the exposed sensor, subtract their signal

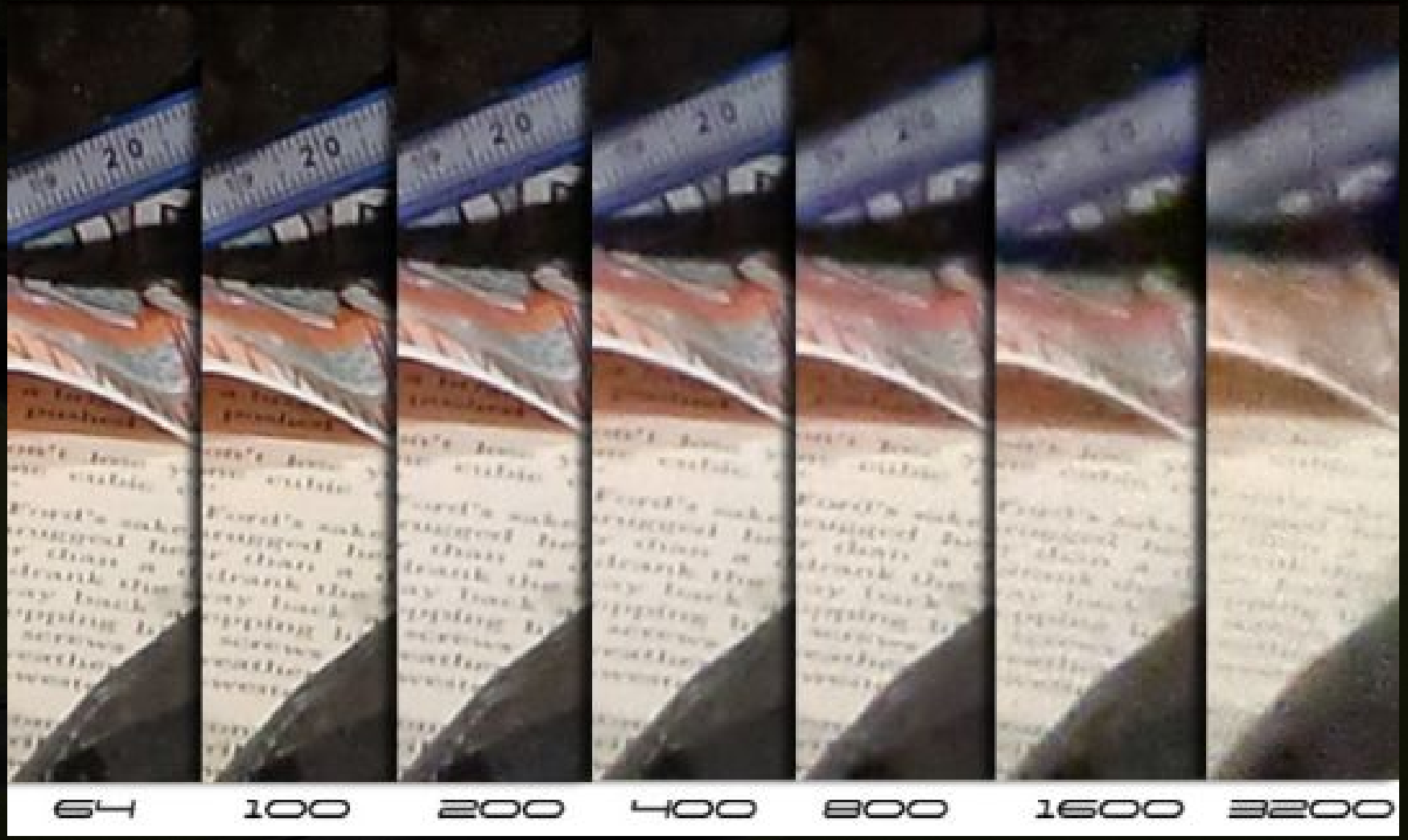




ISO = amplification in AD conversion

- Sensor converts the continuous light signal to a continuous electrical signal
- The analog signal is converted to a digital signal
  - at least 10 bits (even on cell phones), often 12 or more
  - (roughly) linear sensor response
- Before conversion the signal can be amplified
  - ISO 100 means no amplification
  - ISO 1600 means 16x amplification
  - +: can see details in dark areas better
  - -: noise is amplified as well; sensor more likely to saturate

ISO



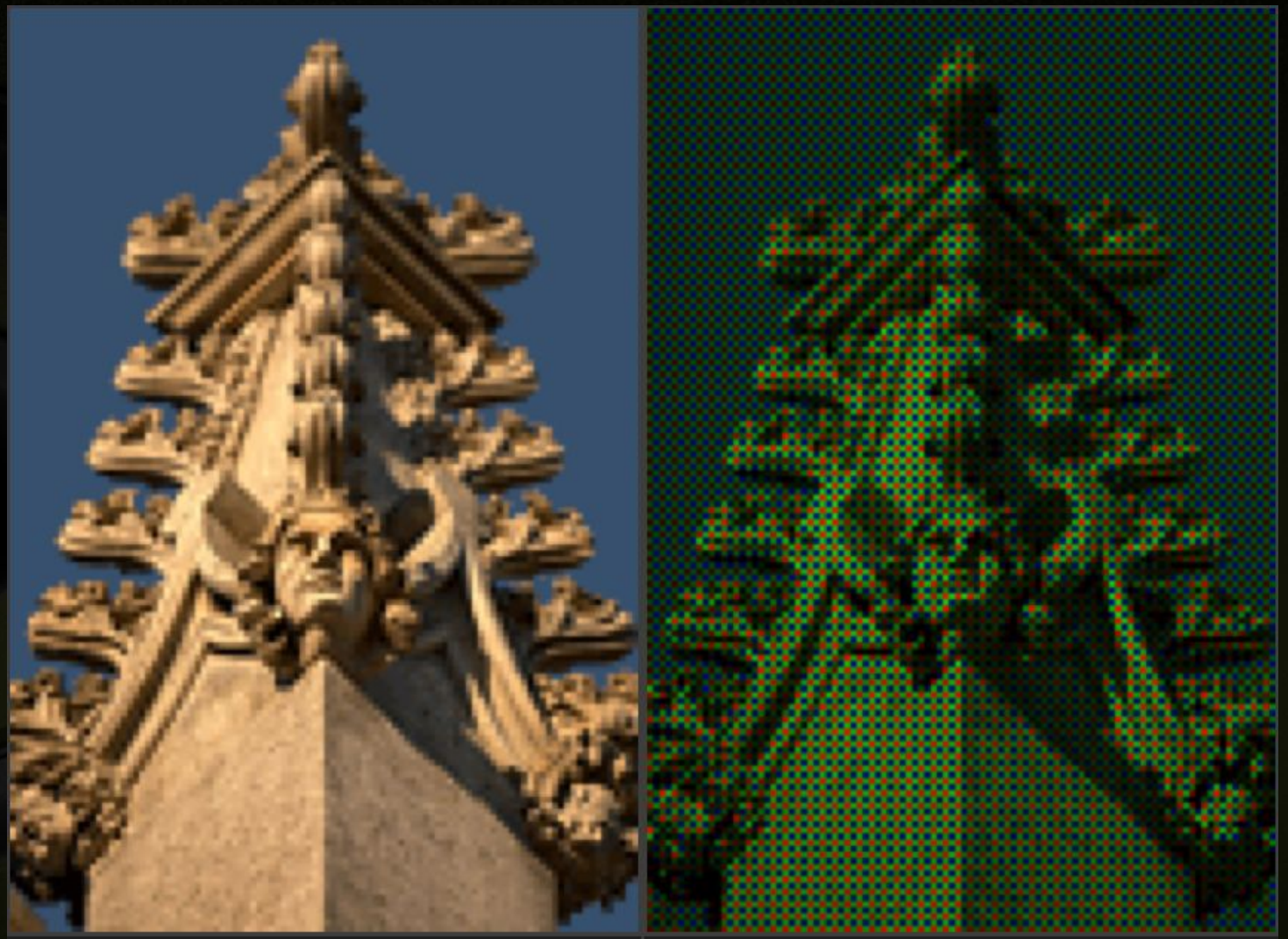
But too dark signal needs amplification

- **With too little light,**
  and a given desired image brightness,
  two alternatives
  - use low ISO and brighten digitally
    - amplifies also post-gain noise (e.g., read noise)
  - use high ISO to get brightness directly
    - a bit less noise

- **But both amplification choices amplify noise**
  - **ideally, you make sure the signal is high by using**
    - **a slower exposure**
      or
    - **larger aperture**

Demosaicking or Demosaicing?

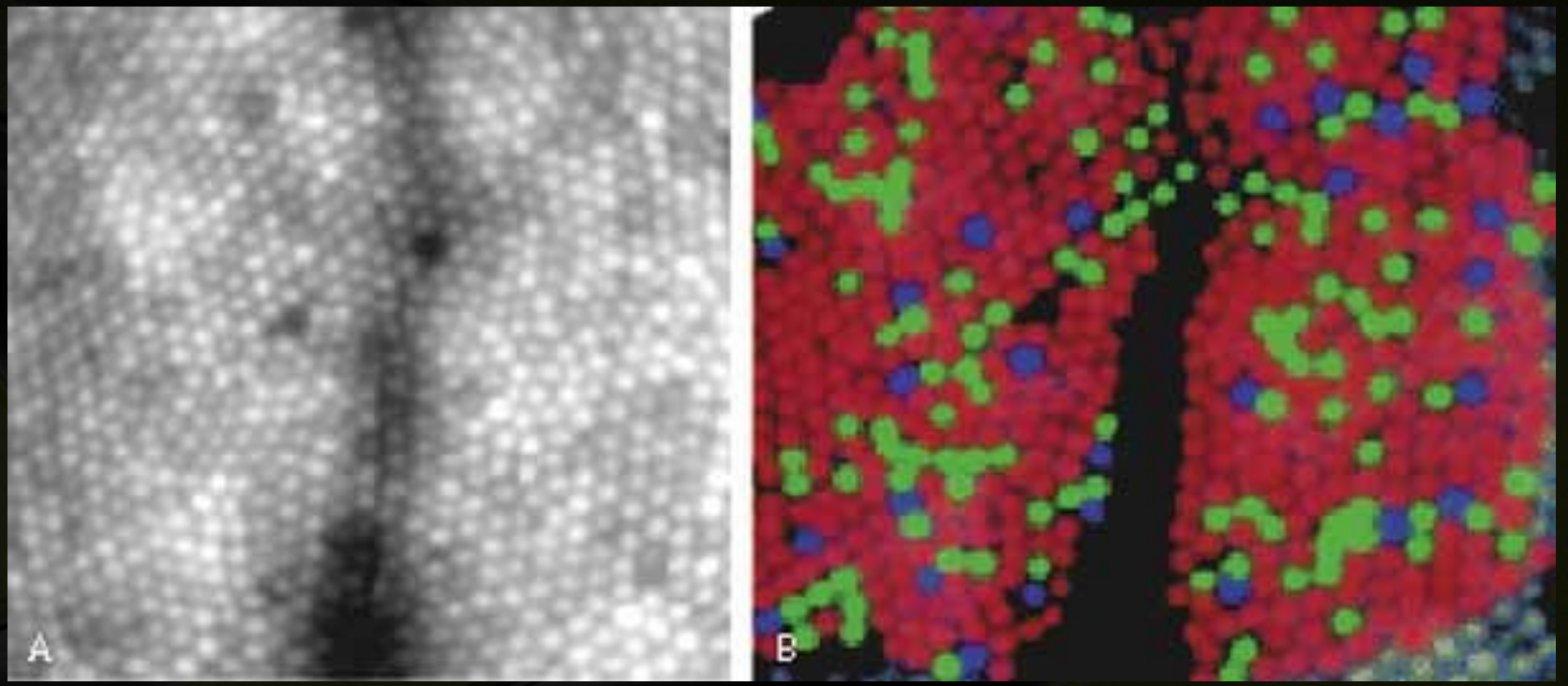
Demosaicking



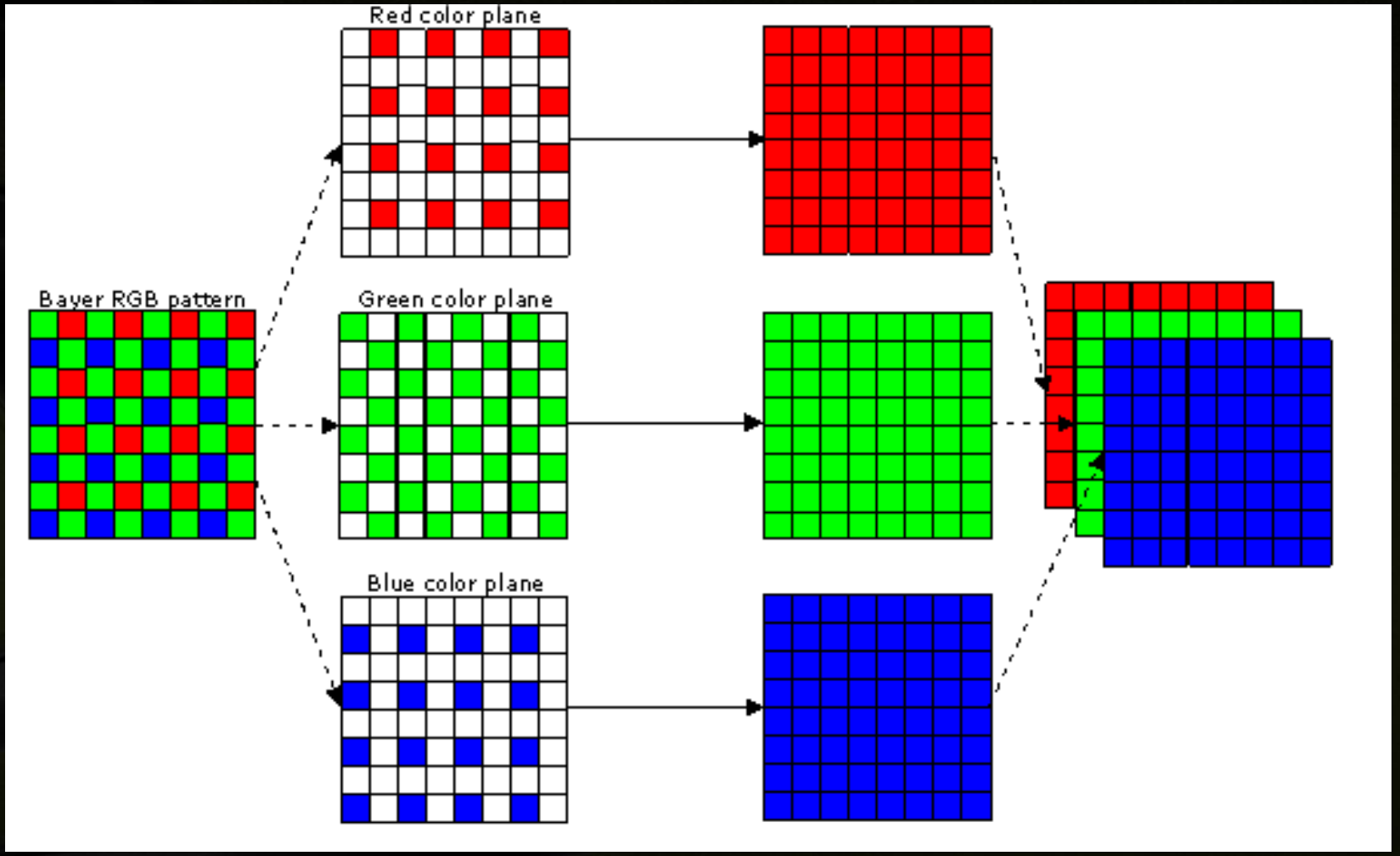
Original Scene
(shown at 200%)

What Your Camera Sees
(through a Bayer array)

Your eyes does it too…

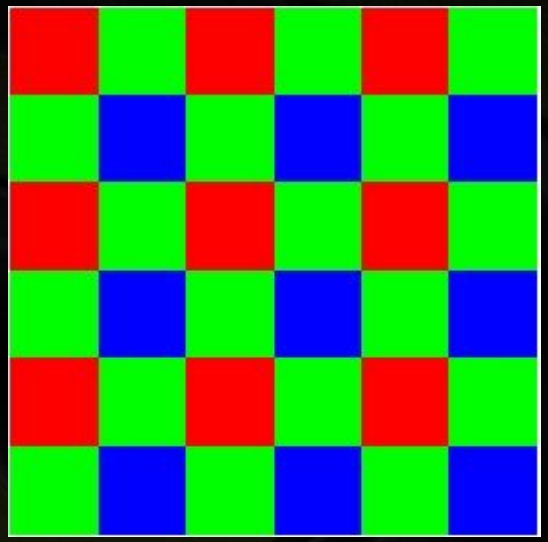


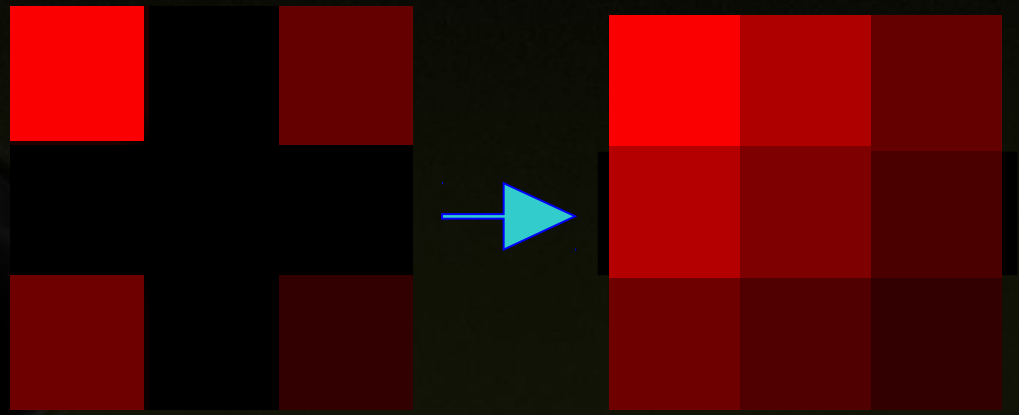
Demosaicking



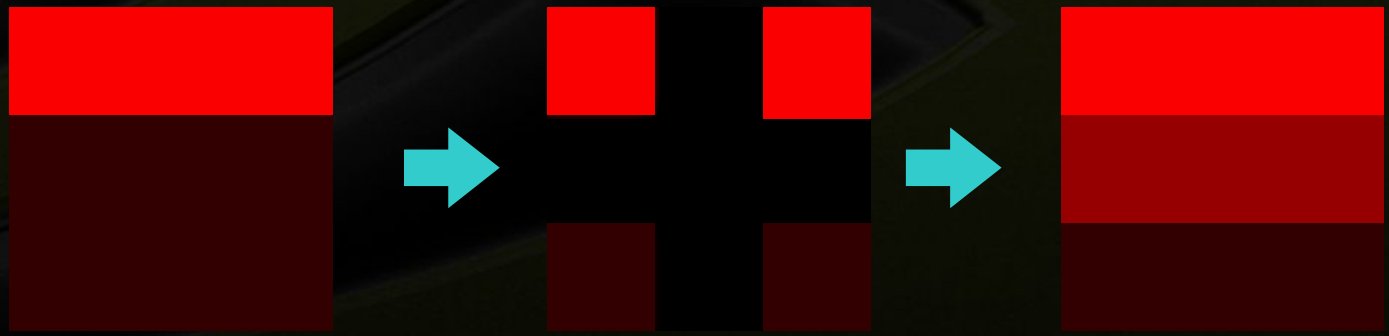
First choice: bilinear interpolation

- Easy to implement

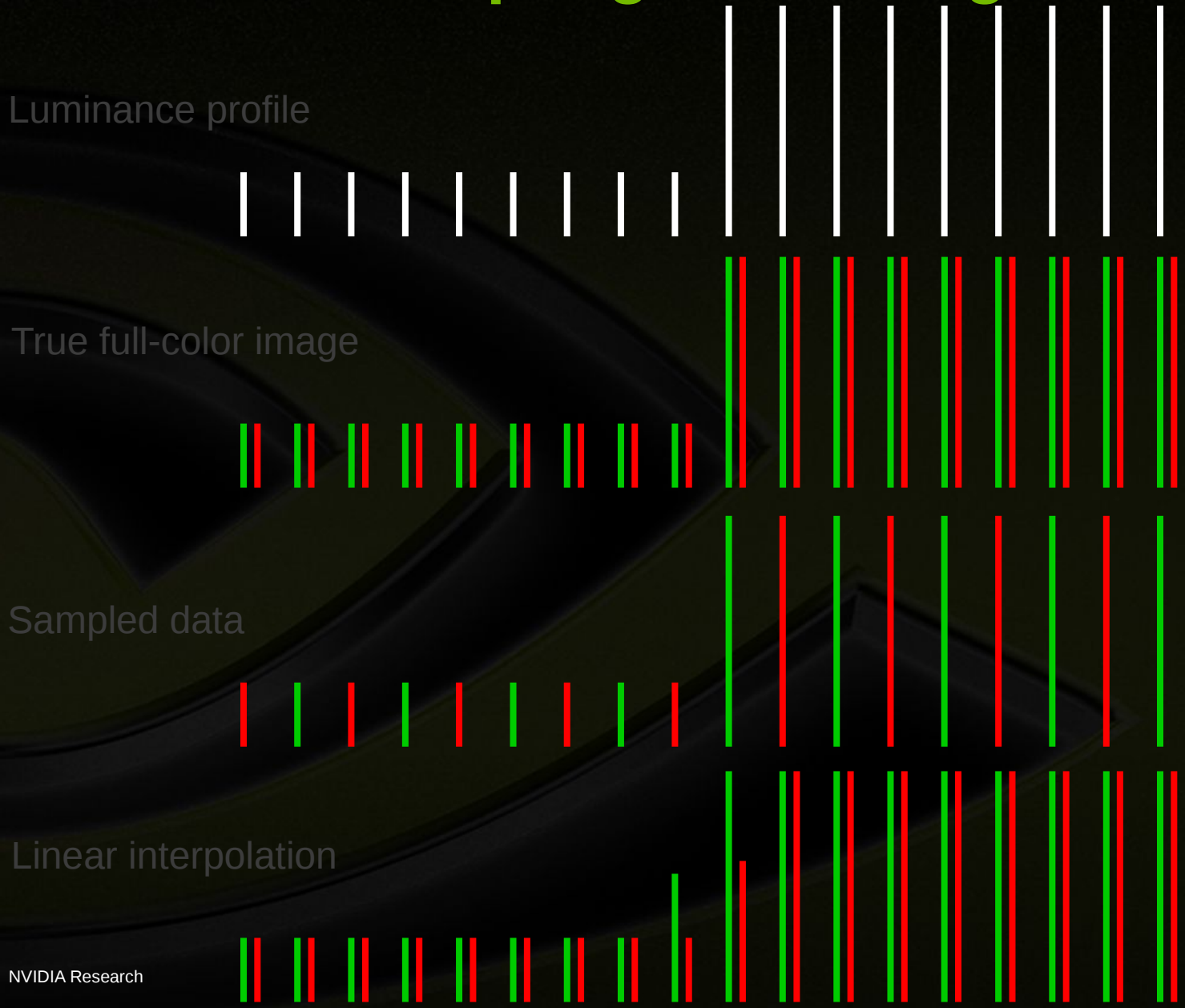




- But fails at sharp edges



Two-color sampling of BW edge

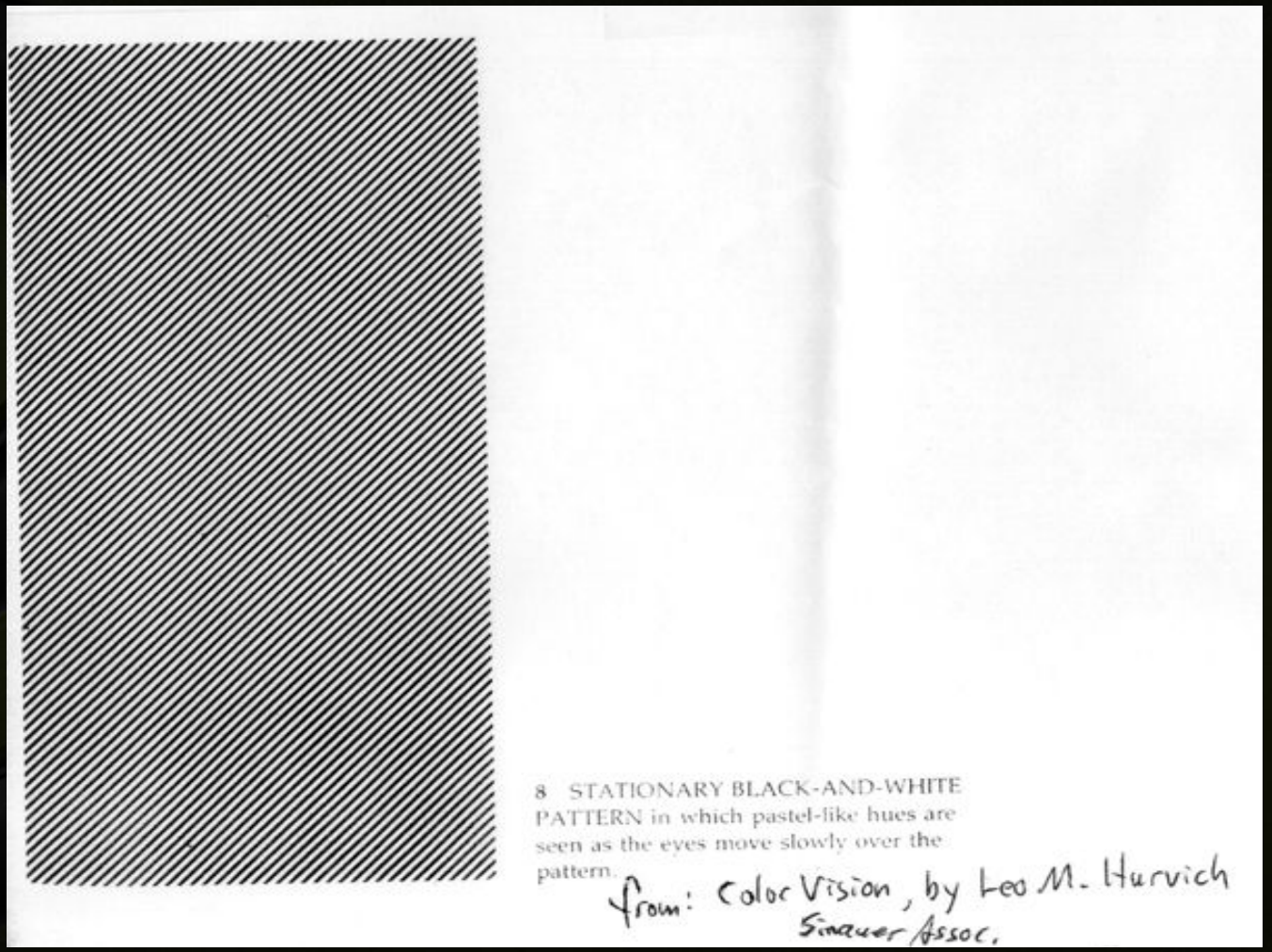


Typical color Moire patterns

low-up of electronic camera
mage.
lotice spurious colors in the
egions of fine detail in the plant

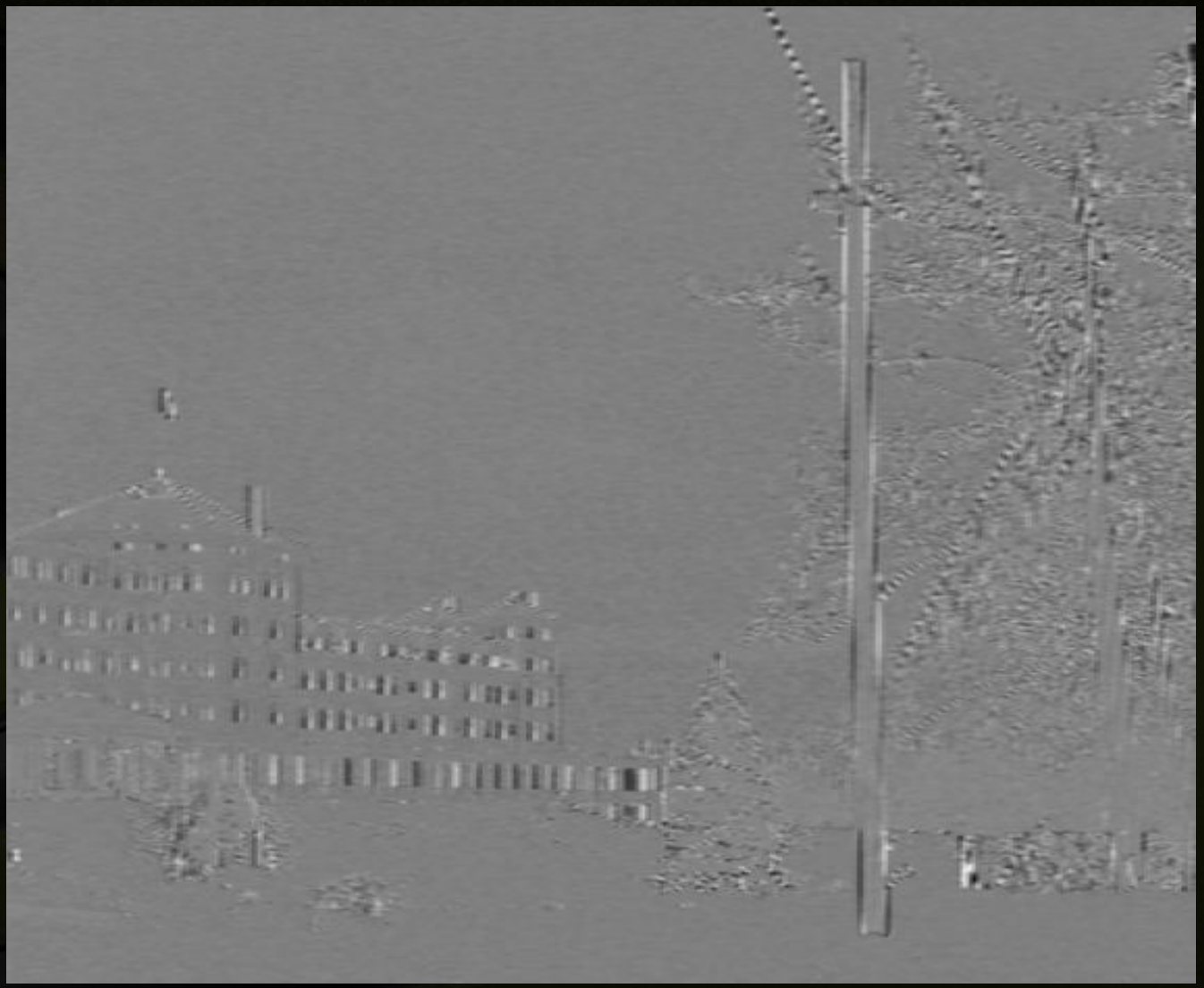
Brewster's colors — evidence of interpolation from spatially offset color samples

Scale relative to human photoreceptor size: each line covers about 7 photoreceptors.



Color sampling artifacts

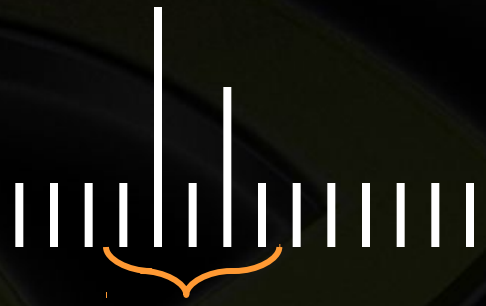
R-G, after linear interpolation



Median filter

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

In:

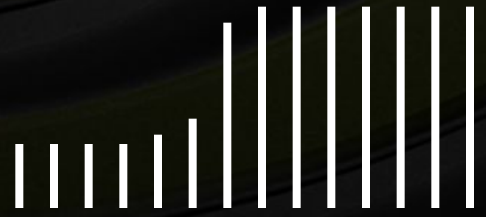


5-pixel neighborhood

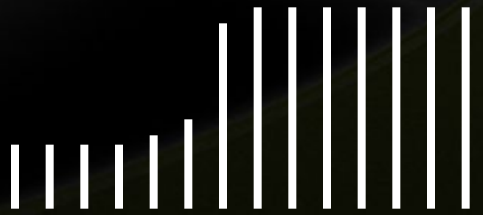
Out:

Spike noise is removed

In:



Out:



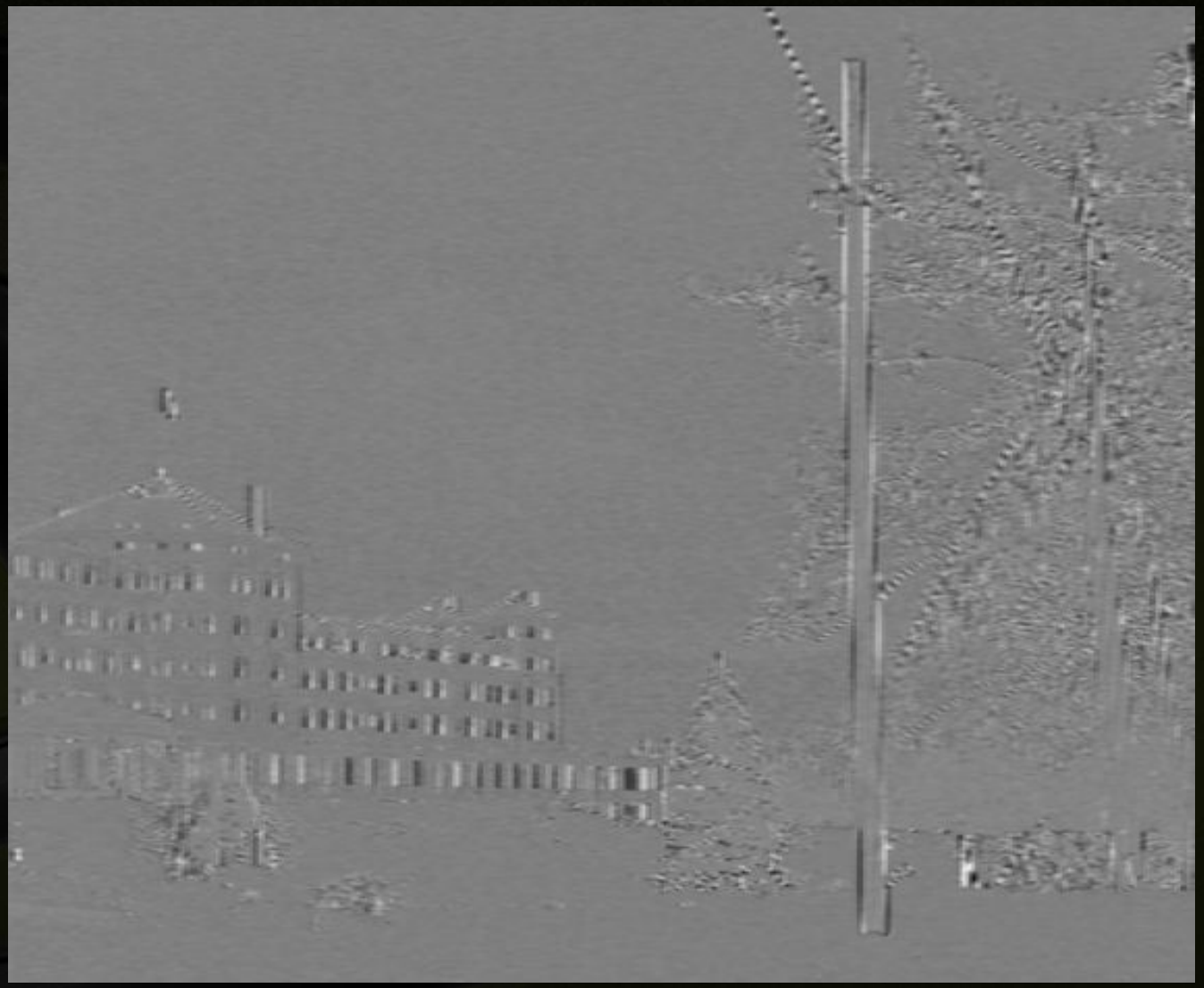
Monotonic edges remain unchanged

Degraded image

Radius 1 median filter

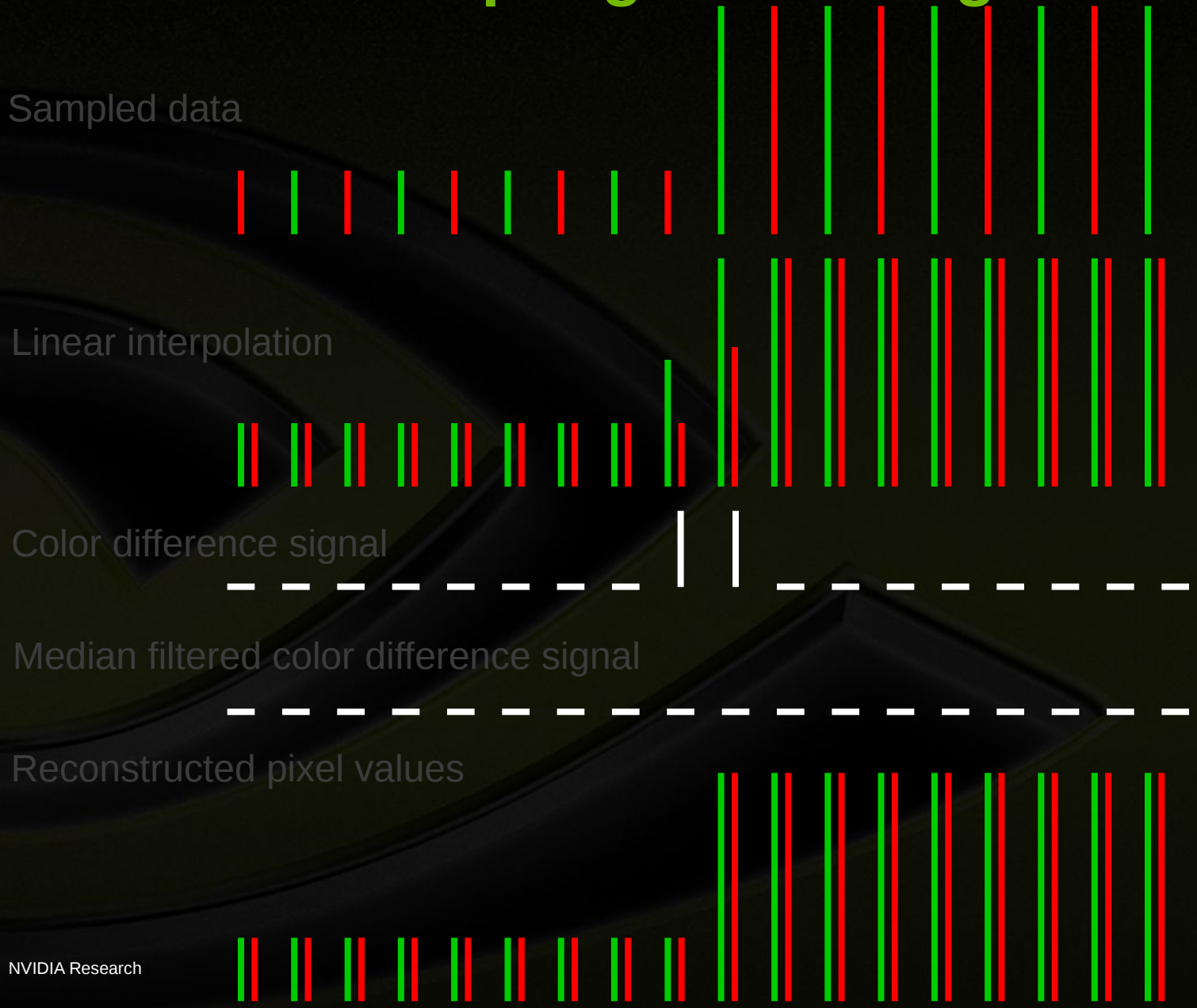
Radius 2 median filter

R – G



R – G, median filtered (5x5)

Two-color sampling of BW edge



Recombining the median filtered colors

Linear interpolation

Median filter interpolation

Take edges into account

- **Use bilateral filtering**
  - **avoid interpolating across edges**

ADAPTIVE DEMOSAICKING
Ramanath, Snyder, JEI 2003

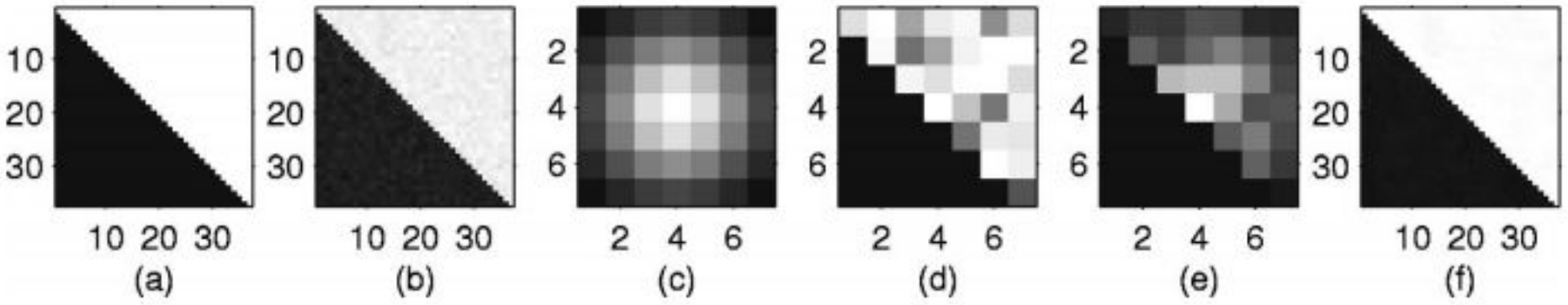


Fig. 3 Bilateral filtering: (a) original image, (b) image corrupted by Gaussian noise, (c) 7×7 blur kernel, (d) 7×7 similarity kernel at row=18, col=18, (e) 7×7 bilateral filter kernel, and (f) resulting image (denoised and sharpened).

Take edges into account

HIGH-QUALITY LINEAR INTERPOLATION FOR DEMOSAICING OF BAYER-PATTERNED COLOR IMAGES
Malvar, He, Cutler, ICASSP 2004

- Predict edges and adjust
  - assumptions
    - luminance correlates with RGB
    - edges = luminance change
- When estimating G at R
  - if the R differs from bilinearly estimated R
    - ⇒ luminance changes
- Correct the bilinear estimate
  - by the difference between the estimate and real value

$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha \Delta_R(i,j)$$

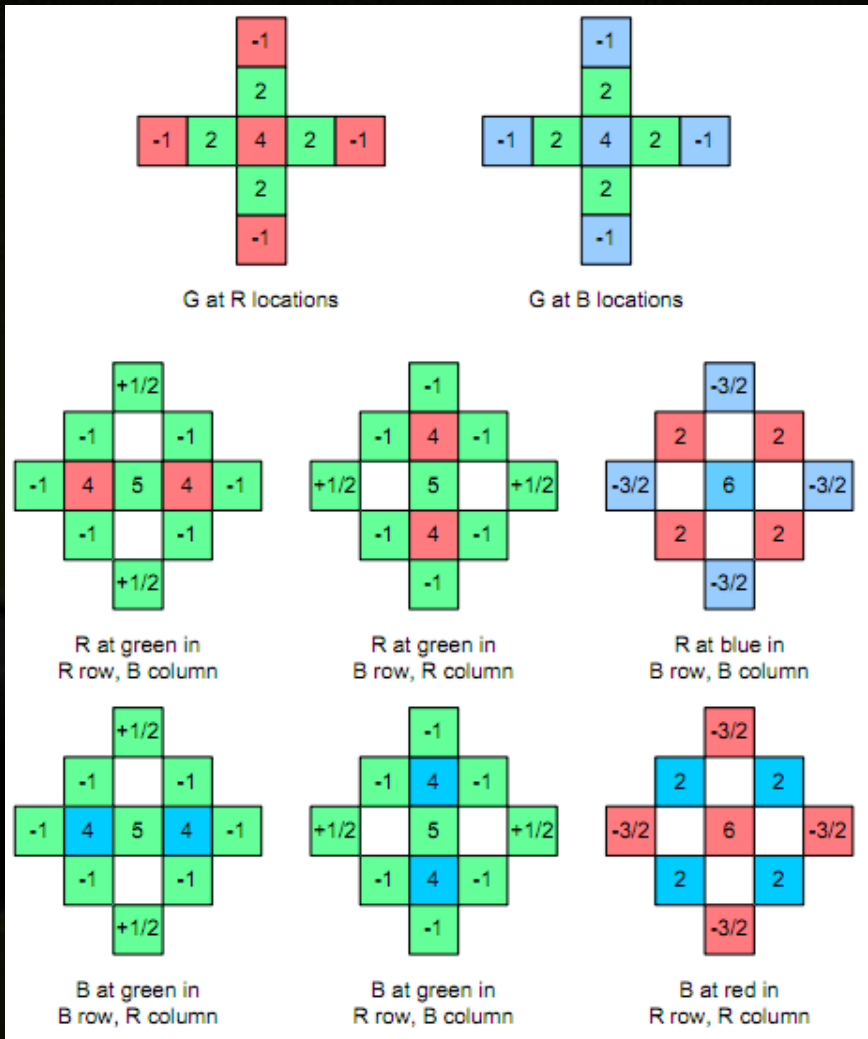


Figure 2. Filter coefficients for our proposed linear

Denoising in ISP?

Joint Demosaicing and Denoising
Hirakawa, Parks, IEEE TIP 2006

- **The experimental results confirm that the proposed method suppresses noise (CMOS/CCD image sensor noise model) while effectively interpolating the missing pixel components, demonstrating a significant improvement in image quality when compared to treating demosaicking and denoising problems independently.**

Denoising using non-local means

- Most image details occur repeatedly
- Each color indicates a group of squares in the image which are almost indistinguishable
- Image self-similarity can be used to eliminate noise
  - it suffices to average the squares which resemble each other

Image and movie denoising by nonlocal means
Buades, Coll, Morel, IJCV 2006

Restoring a highly compressed image

- NL-means is able to remove block artifacts due to compression
  - but at the cost of removing some details, as the difference between the compressed and restored image shows

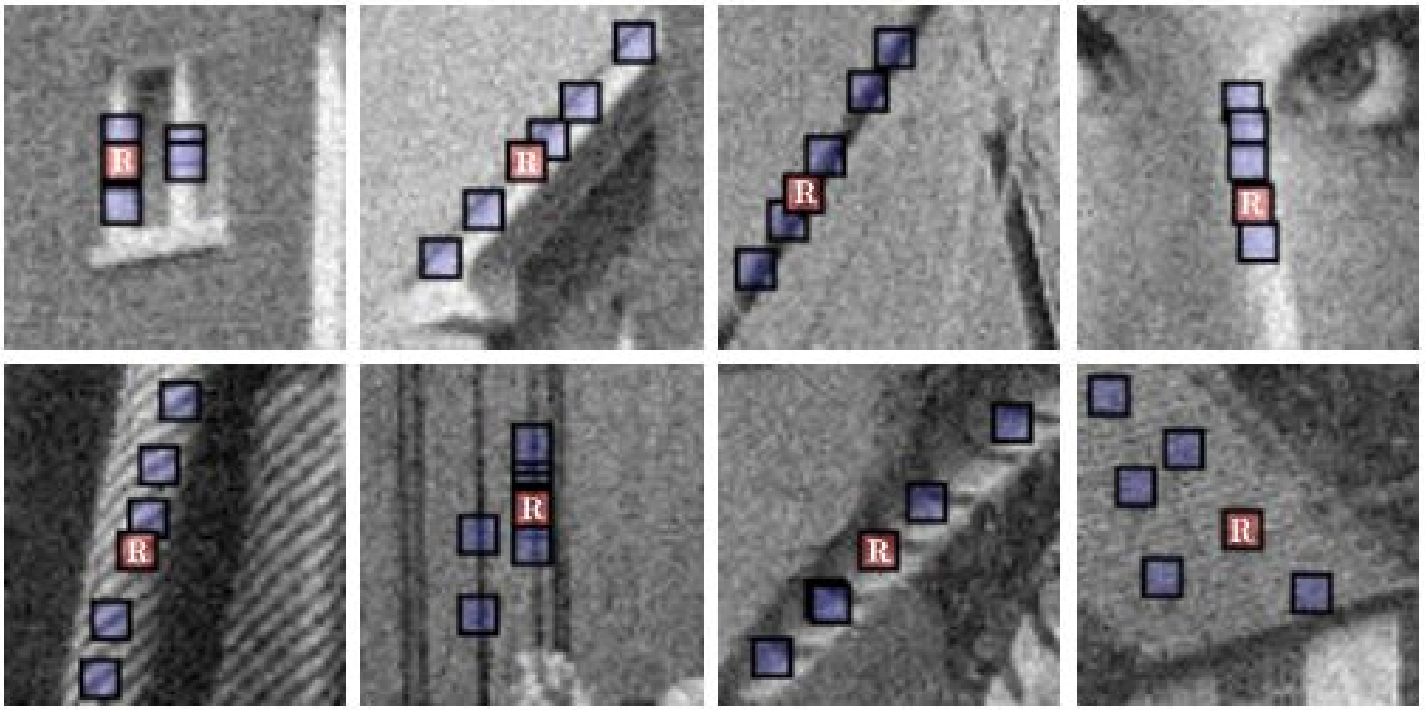


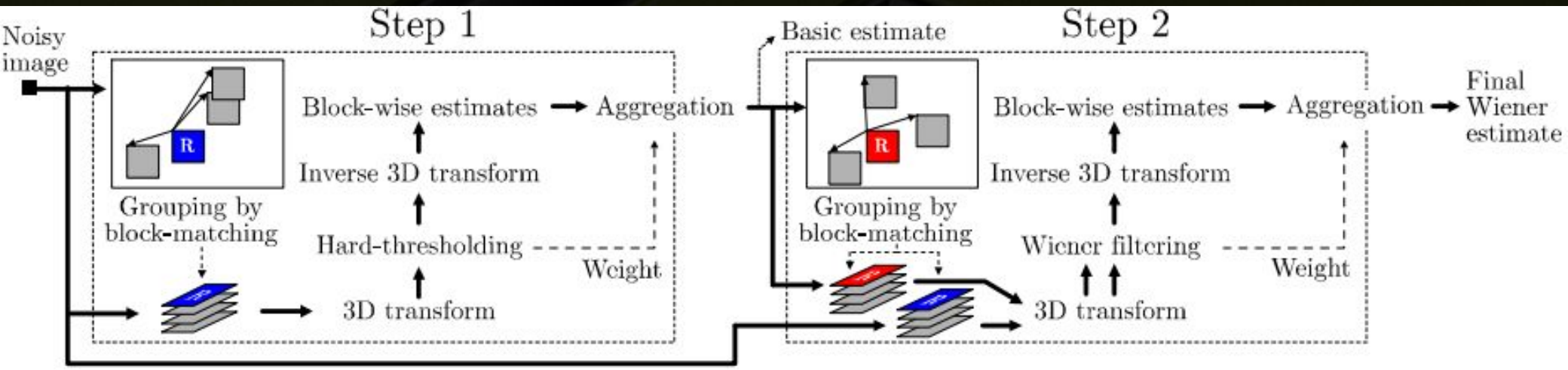
BM3D (Block Matching 3D)

ACCEPTED. TO APPEAR IN IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 16, NO. 8, AUGUST 2007.

Image denoising by sparse 3D transform-domain collaborative filtering

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, *Senior Member, IEEE*



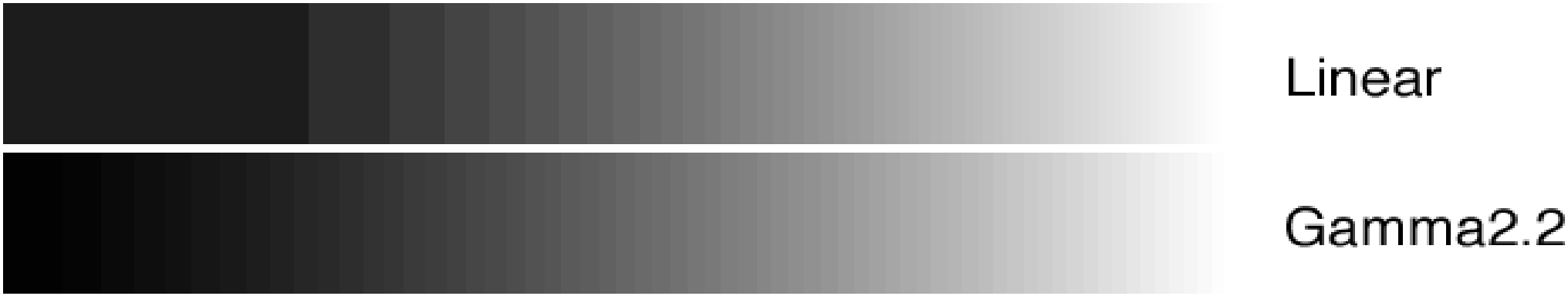


How many bits are needed for smooth shading?

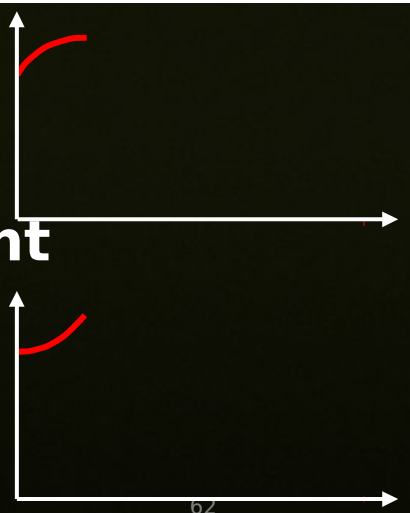
- With a given adaptation, human vision has contrast sensitivity ~1%
  - call black 1, white 100
  - you can see differences
    - 1, 1.01, 1.02, … needed step size ~ 0.01
    - 98, 99, 100 needed step size ~ 1
  - with linear encoding
    - delta 0.01
      - 100 steps between 99 & 100 wasteful
    - delta 1
      - only 1 step between 1 & 2 lose detail in shadows
  - instead, apply a non-linear power function, gamma
    - provides adaptive step size

Gamma encoding

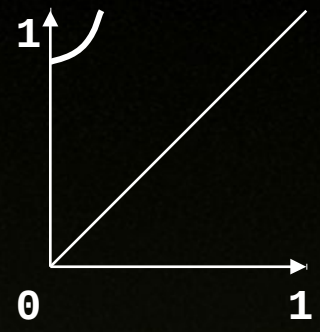
- With 6 bits available (for illustration below) for encoding
  - linear loses detail in the dark end



- Raise intensity X to power Xγ where γ = 1/2.2
  - then encode
- Display applies γ = 2.5 to get back to linear light
  - the difference boosts colors
    to compensate for a dark viewing environment



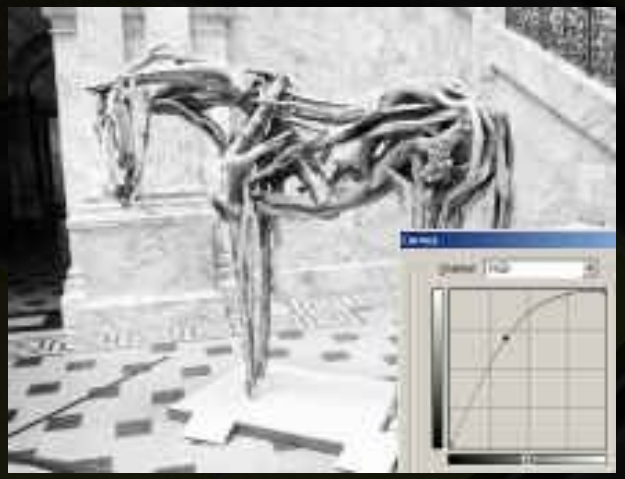
Gamma on displays



- CRT
  - inherent gamma of ~2.35 – 2.55
  - due to electrostatics of the cathode and electron gun
  - just a coincidence that has an almost perfect inverse match to human vision system non-linearity!
- LCD
  - no inherent gamma
  - applied as a look-up table to match with conventions
- If gamma is not right, both colors and intensities shift
  - example: if `(0, 255, 127)` is not gamma corrected,
    - red channel remains 0,
    - green remains 255,
    - blue is decreased by the display

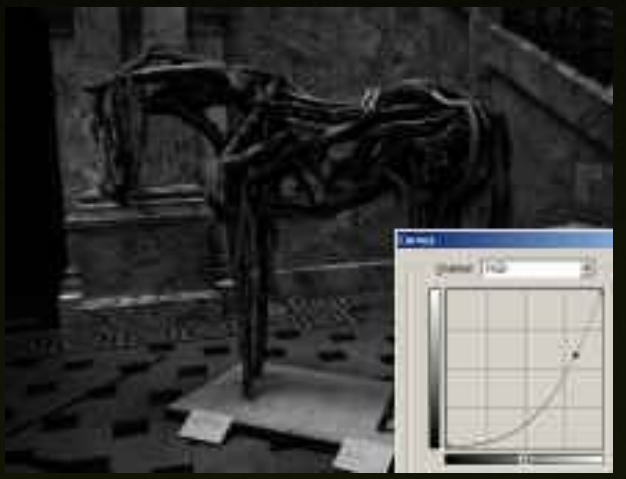
Display brightness and contrast

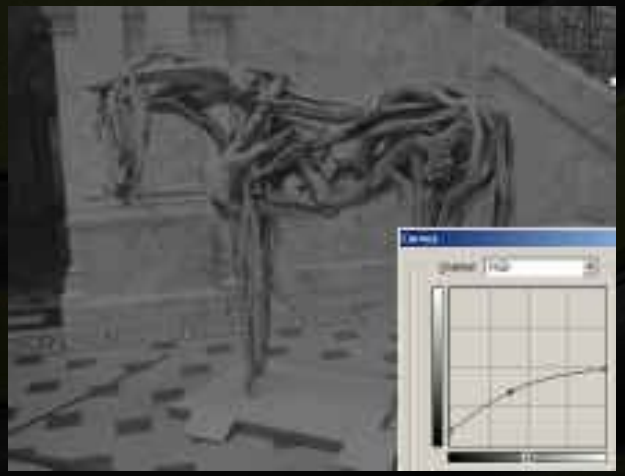
- Brightness and contrast knobs control α and γ
  Which one controls which?

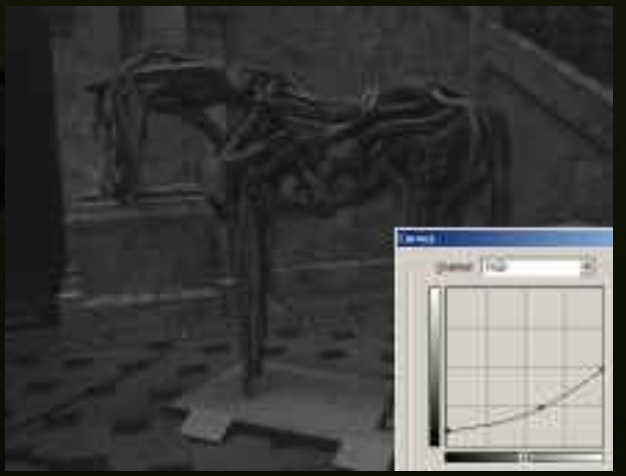


Brightness controls γ,

contrast controls α







Gamma encoding

- With the “delta” ratio of 1.01
  - need about 480 steps to reach 100
  - takes almost 9 bits
- 8 bits, nonlinearly encoded
  - sufficient for broadcast quality digital TV
  - contrast ratio ~ 50 : 1
- With poor viewing conditions or display quality
  - fewer bits needed

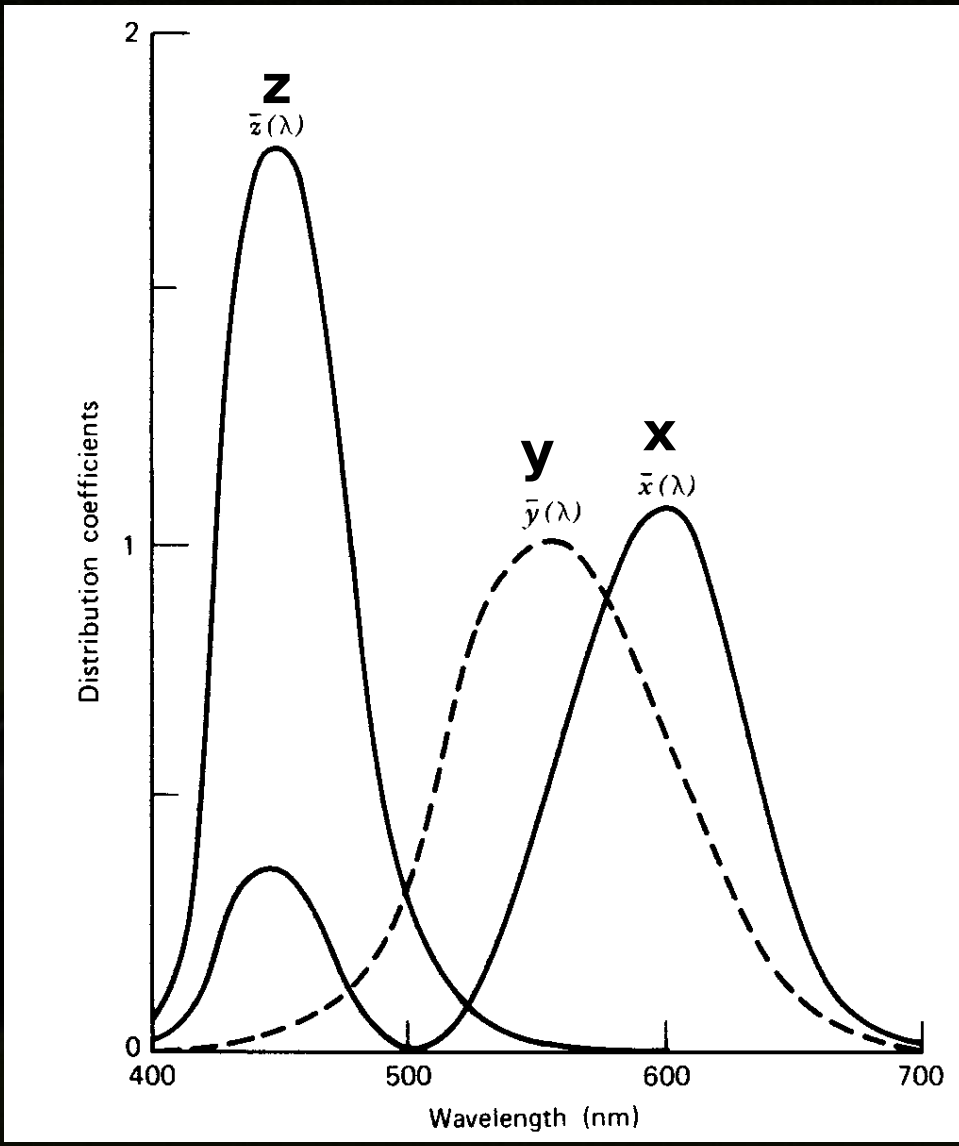
Gamma summary

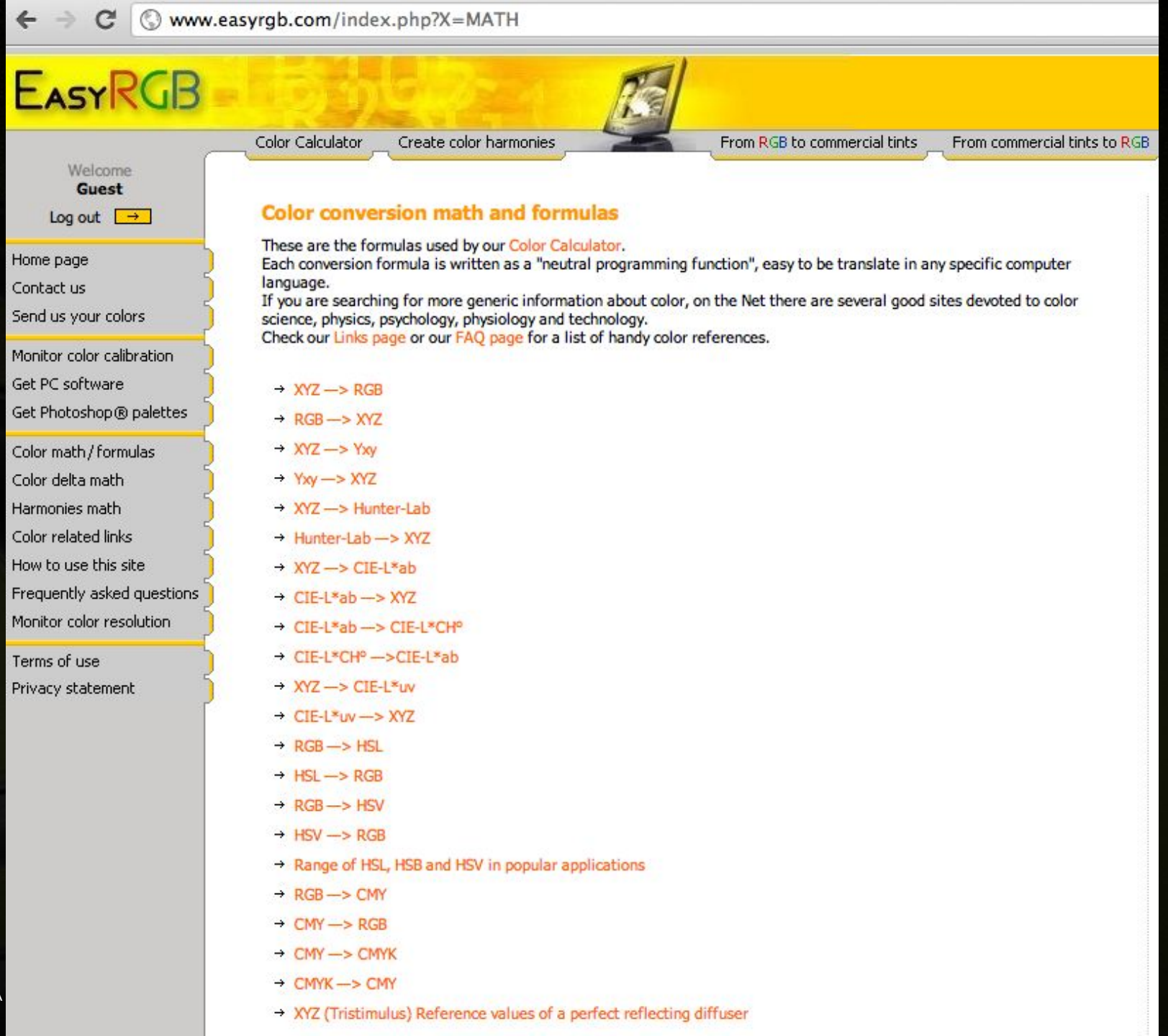
- At the camera or encoding level
  - apply a gamma of around 1 / 2.2
- The CRT applies a gamma of 2.5
- The residual exponent 2.2 / 2.5
  - boosts the colors to compensate for the dark environment

- See
  http://www.poynton.com/GammaFAQ.html
  http://www.poynton.com/notes/color/GammaFAQ.html
  http://www.poynton.com/PDFs/Rehabilitation_of_gamma.pdf

The CIE *XYZ* System

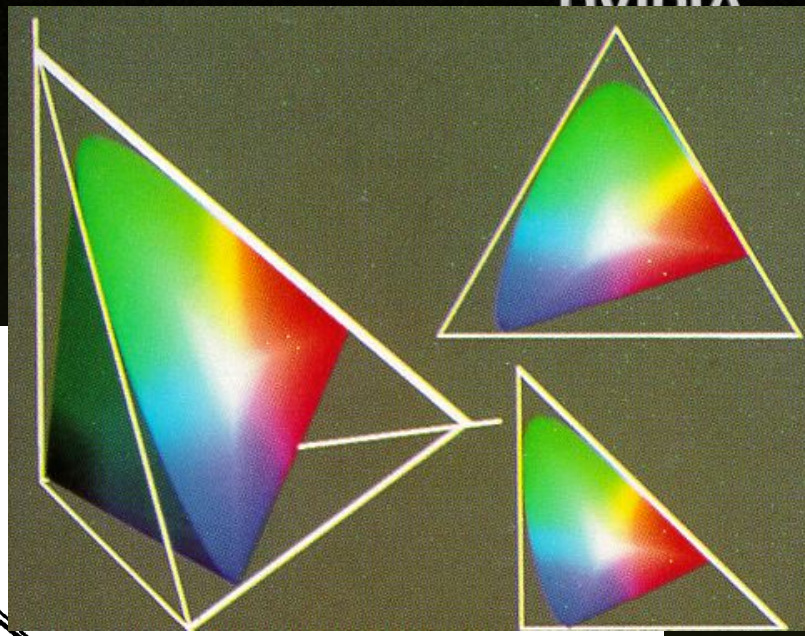
- A standard created in 1931 by CIE
  - Commission Internationale de L'Eclairage
- Defined in terms of three color matching functions
- Given an emission spectrum, we can use the CIE matching functions to obtain the *x, y* and *z* coordinates
  - y corresponds to luminance perception

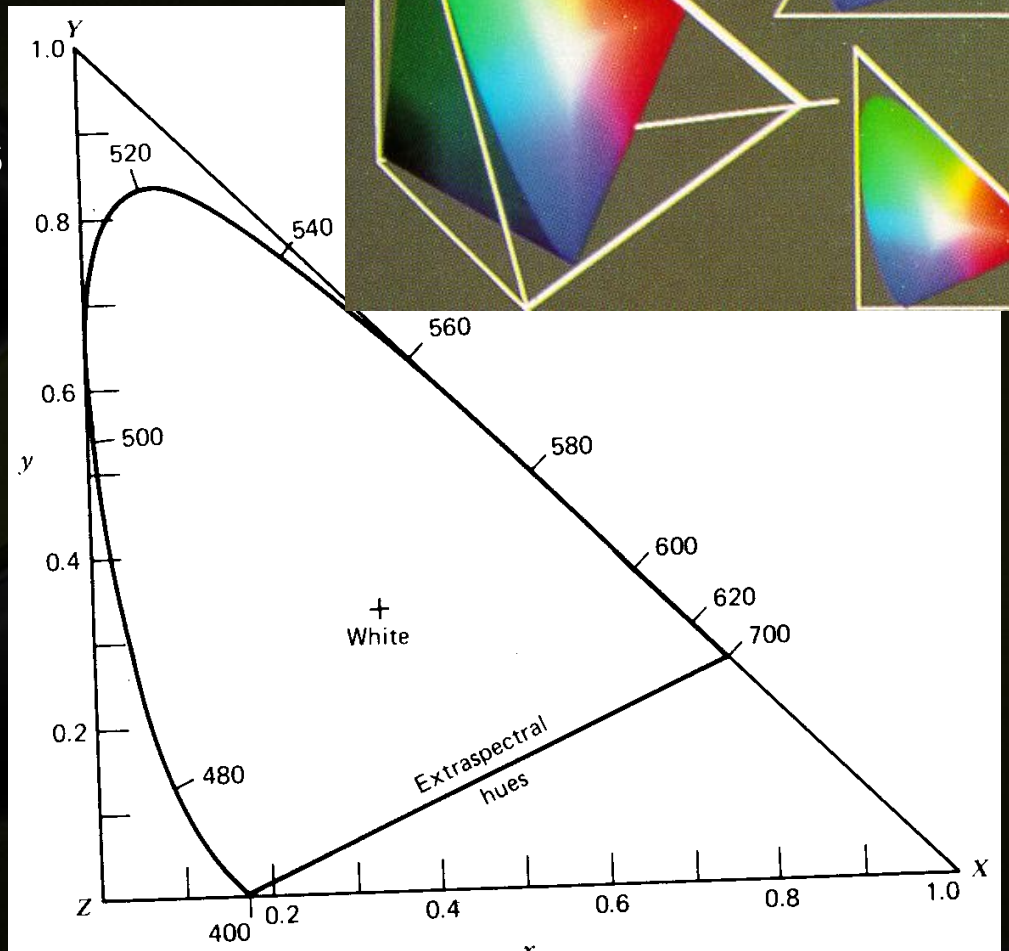



www.easyrgb.com/index.php?X=MATH
EasyRGB
Color Calculator
Create color harmonies
From RGB to commercial tints
From commercial tints to RGB
Welcome
Guest
Log out
Home page
Contact us
Send us your colors
Monitor color calibration
Get PC software
Get Photoshop® palettes
Color math / formulas
Color delta math
Harmonies math
Color related links
How to use this site
Frequently asked questions
Monitor color resolution
Terms of use
Privacy statement
Color conversion math and formulas
These are the formulas used by our Color Calculator.
Each conversion formula is written as a "neutral programming function", easy to be translate in any specific computer language.
If you are searching for more generic information about color, on the Net there are several good sites devoted to color science, physics, psychology, physiology and technology.
Check our Links page or our FAQ page for a list of handy color references.
→ XYZ —> RGB
→ RGB —> XYZ
→ XYZ —> Yxy
→ Yxy —> XYZ
→ XYZ —> Hunter-Lab
→ Hunter-Lab —> XYZ
→ XYZ —> CIE-L*ab
→ CIE-L*ab —> XYZ
→ CIE-L*ab —> CIE-L*CH°
→ CIE-L*CH° —>CIE-L*ab
→ XYZ —> CIE-L*uv
→ CIE-L*uv —> XYZ
→ RGB —> HSL
→ HSL —> RGB
→ RGB —> HSV
→ HSV —> RGB
→ Range of HSL, HSB and HSV in popular applications
→ RGB —> CMY
→ CMY —> RGB
→ CMY —> CMYK
→ CMYK —> CMY
→ XYZ (Tristimulus) Reference values of a perfect reflecting diffuser

The CIE Chromaticity Diagram

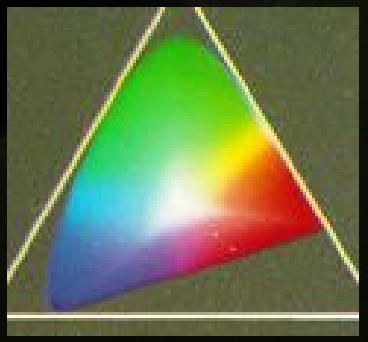
- Intensity is measured as the distance from origin
  - black = (0, 0, 0)
- *Chromaticity coordinates* give a notion of color independent of brightness
- A projection of the plane $x + y + z = 1$ yields a *chromaticity* value dependent on
  - *dominant wavelength (= hue)*, and
  - *excitation purity (= saturation)*
    - the distance from the white at (1/3, 1/3, 1/33)





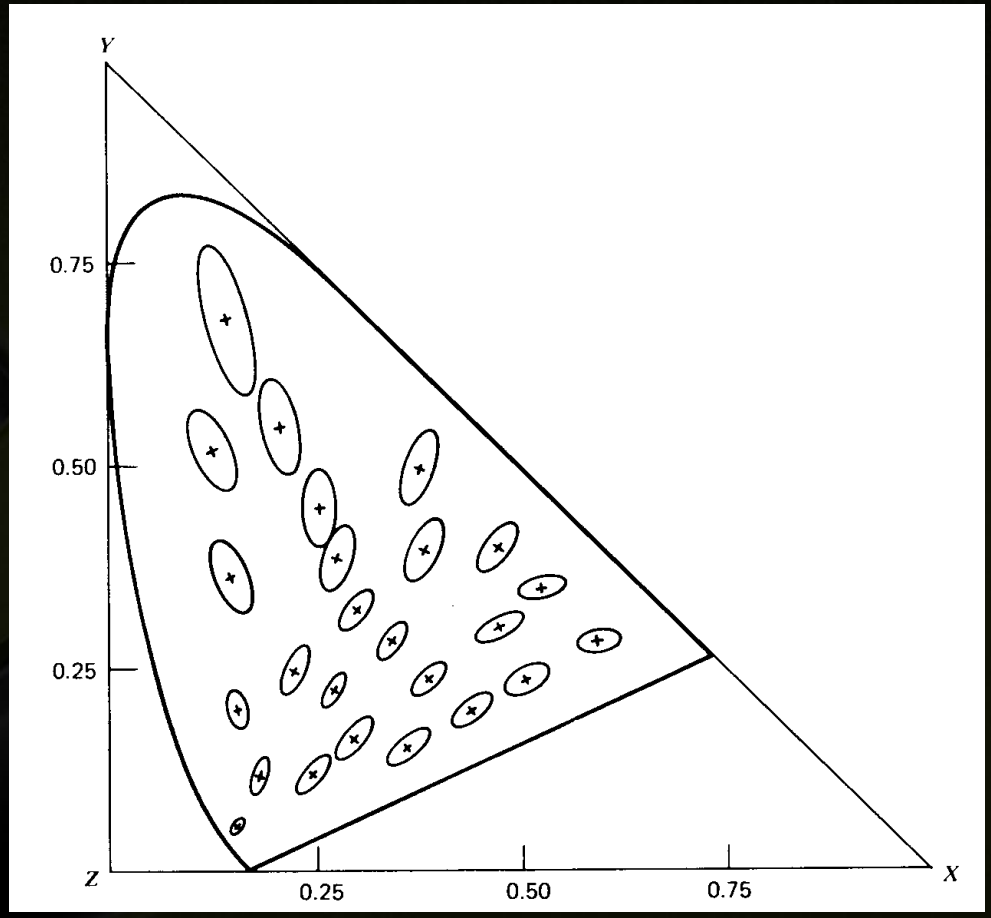
More About Chromaticity

- **Dominant wavelengths go around the perimeter of the chromaticity blob**
  - **a color's dominant wavelength is where a line from white through that color intersects the perimeter**
  - **some colors, called *nonspectral* colors, don't have a dominant wavelength**
    (which ones? colors that mix red and blue**)**
- **Excitation purity is measured in terms of a color's position on the line to its dominant wavelength**
- **Complementary colors lie on opposite sides of white, and can be mixed to get white**
  - **complement of blue is** yellow
  - **complement of red is** cyan



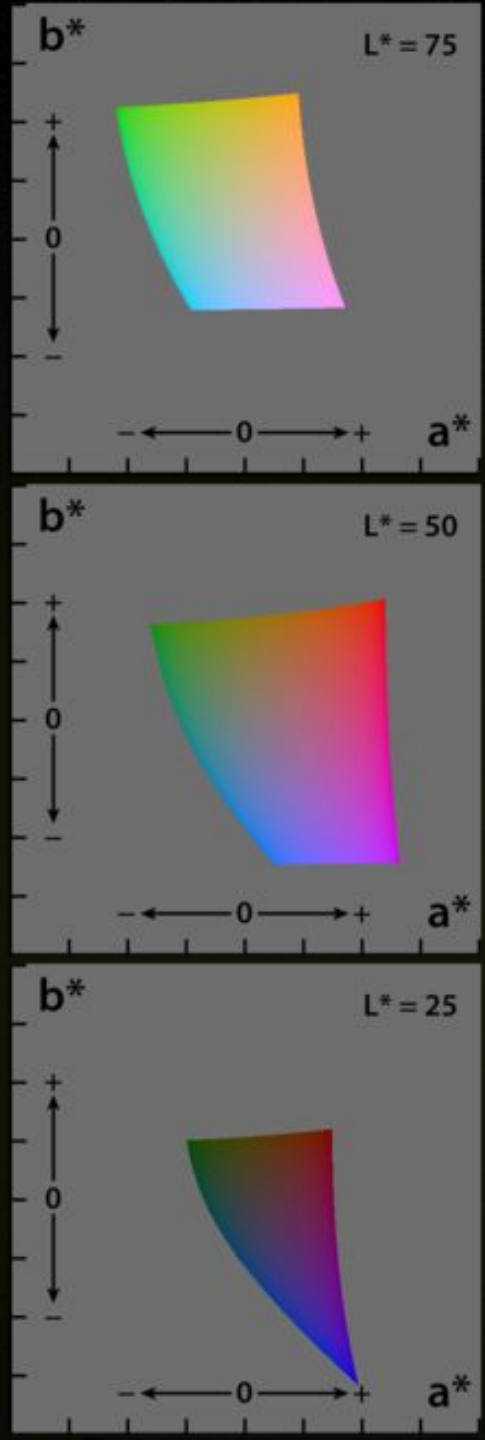
Perceptual (non-)uniformity

- The *XYZ* color space is not perceptually uniform!
- Enlarged ellipses of constant color in XYZ space



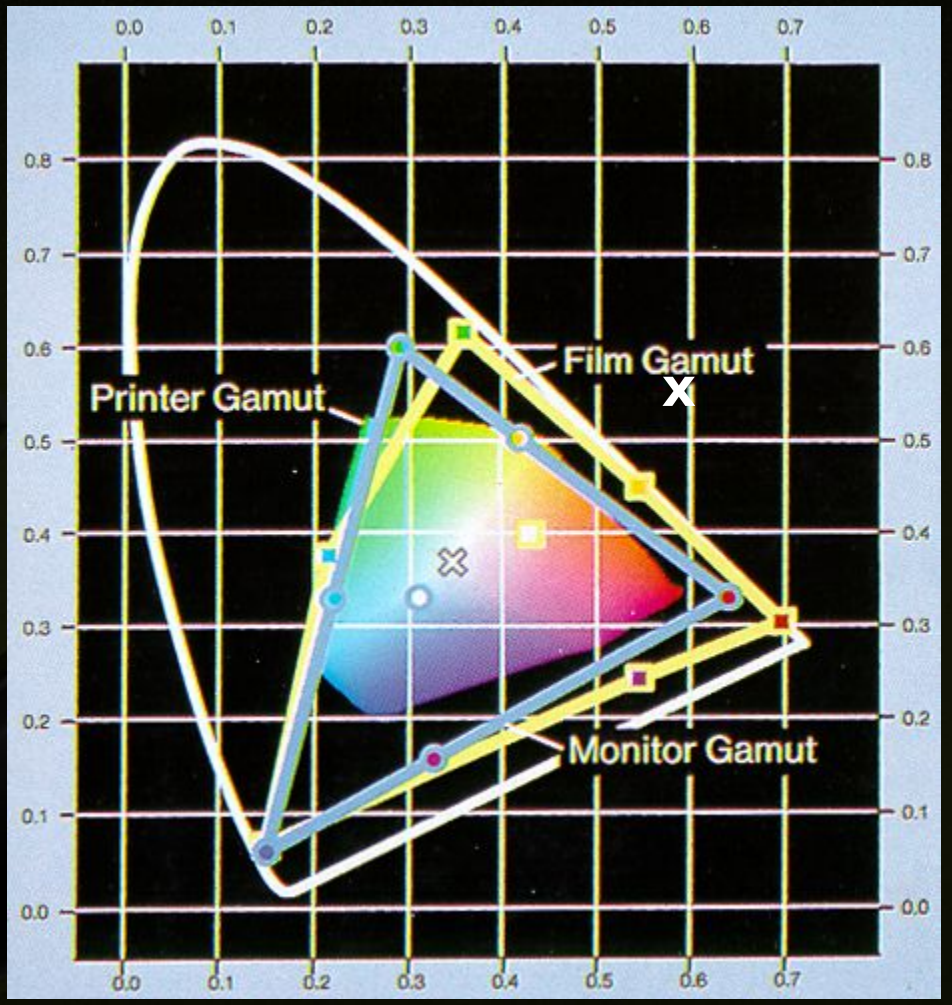
CIE L*a*b*: uniform color space

- Lab is designed to approximate human vision
  - it aspires to perceptual uniformity
  - L component closely matches human perception of lightness
- A good color space for image processing



Gamuts

- Not every device can reproduce every color
- A device's range of reproducible colors is called its *gamut*



YUV, YCbCr, …

- Family of color spaces for video encoding
  - including in FCam, video and viewfinder usually YUV
- Channels
  - Y = luminance [linear]; Y’ = luma [gamma corrected]
  - CbCr / UV = chrominance [always linear]
- Y′CbCr is not an absolute color space
  - it is a way of encoding RGB information
  - the actual color depends on the RGB primaries used
- Colors are often filtered down
  - 2:1, 4:1

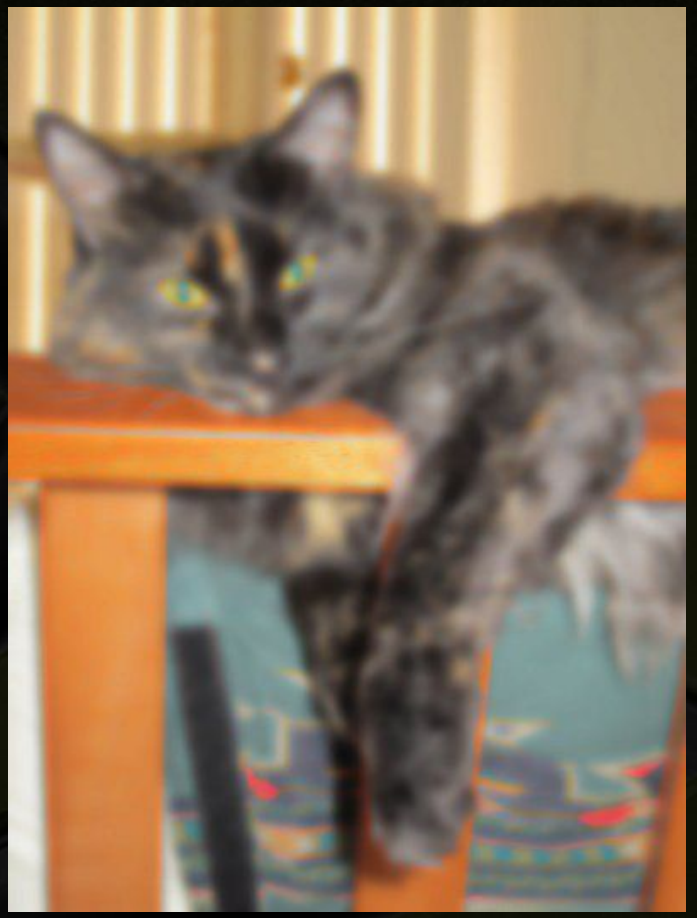
- Many formulas!

Break RGB to Lab channels

Blur “a” channel (red-green)

Blur “b” channel (blue-yellow)

Blur “L” channel



Luminance from RGB

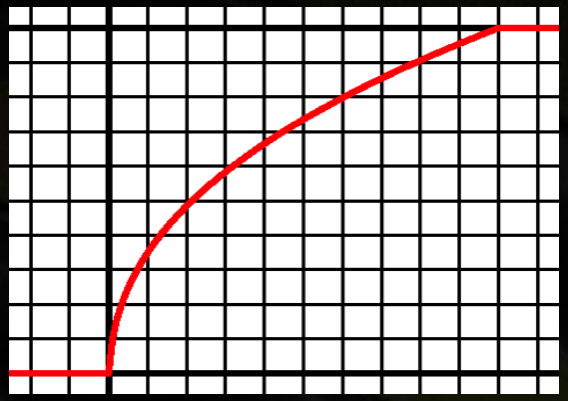
- If three sources of same radiance appear R, G, B:
  - green will appear the brightest, it has the luminous efficiency
  - red will appear less bright
  - blue will be the darkest
- Luminance by NTSC: 0.2990 R + 0.5870 G + 0.1140 B
  - based on phosphors in use in 1953
- Luminance by CIE: 0.2126 R + 0.7152 G + 0.0722 B
  - based on contemporary phosphors
- Luminance by ITU: 0.2125 R + 0.7154 G + 0.0721 B
- 1/4 R + 5/8 G + 1/8 B works fine
  - quick to compute: R>>2 + G>>1 + G>>3 + B>>3
    - range is [0, 252]

Cameras use sRGB

- sRGB is a standard RGB color space (since 1996)
  - uses the same primaries as used in studio monitors and HDTV
  - and a gamma curve typical of CRTs
  - allows direct display
- The sRGB gamma
  - cannot be expressed as a single numerical value
  - the overall gamma is approximately 2.2, consisting of
    - a linear (gamma 1.0) section near black,
    - and a non-linear section elsewhere involving a 2.4 exponent
- First need to map from sensor RGB to standard
  - need calibration

sRGB from XYZ

XYZ → matrix(3x3) → RGBsRGB → RGB´sRGB → RGB8Bit



RsRGB < 0.0031308

R´sRGB = 12.92 RsRGB

RsRGB > 0.0031308

R´sRGB = 1.055 RsRGB(1/2.4) - 0.055

R8Bit = round[255 R'sRGB]

linear relation between XYZ und sRGB:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{pmatrix} \begin{pmatrix} RsRGB \\ GsRGB \\ BsRGB \end{pmatrix}$$

red green blue

Primaries according to ITU-R BT.709.3

Image processing in linear or non-linear space?

- Simulating physical world
  - use linear light
  - a weighted average of gamma-corrected pixel values is not a linear convolution!
    - Bad for antialiasing
  - want to numerically simulate lens?
    - Undo gamma first
- Dealing with human perception
  - using non-linear coding allows minimizing perceptual errors due to quantization

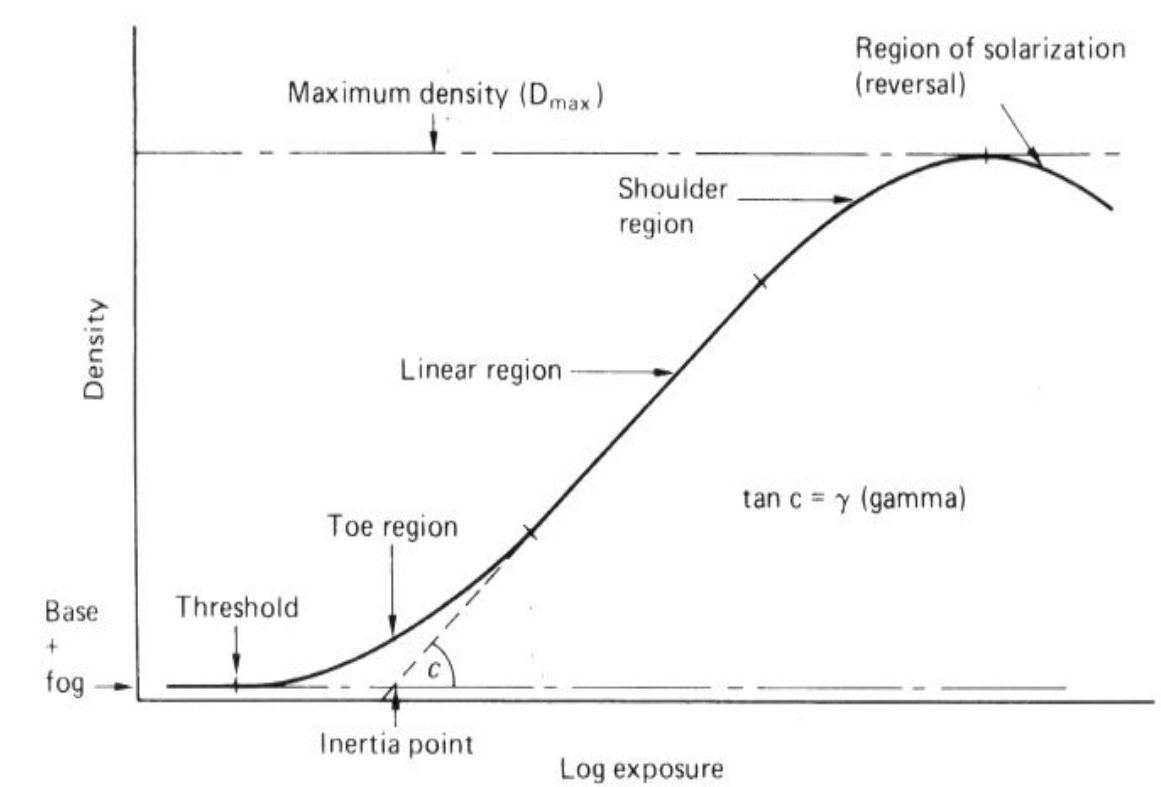
Linearizing from sRGB

$$C_{srgb} = \{R, G, B\} / 255$$

$$C_{linear} = \begin{cases} \dfrac{C_{srgb}}{12.92}, & C_{srgb} < 0.04045 \\ \left(\dfrac{C_{srgb} + 0.055}{0.055} \right)^{2.4}, & C_{srgb} > 0.04045 \end{cases}$$

Film response curve

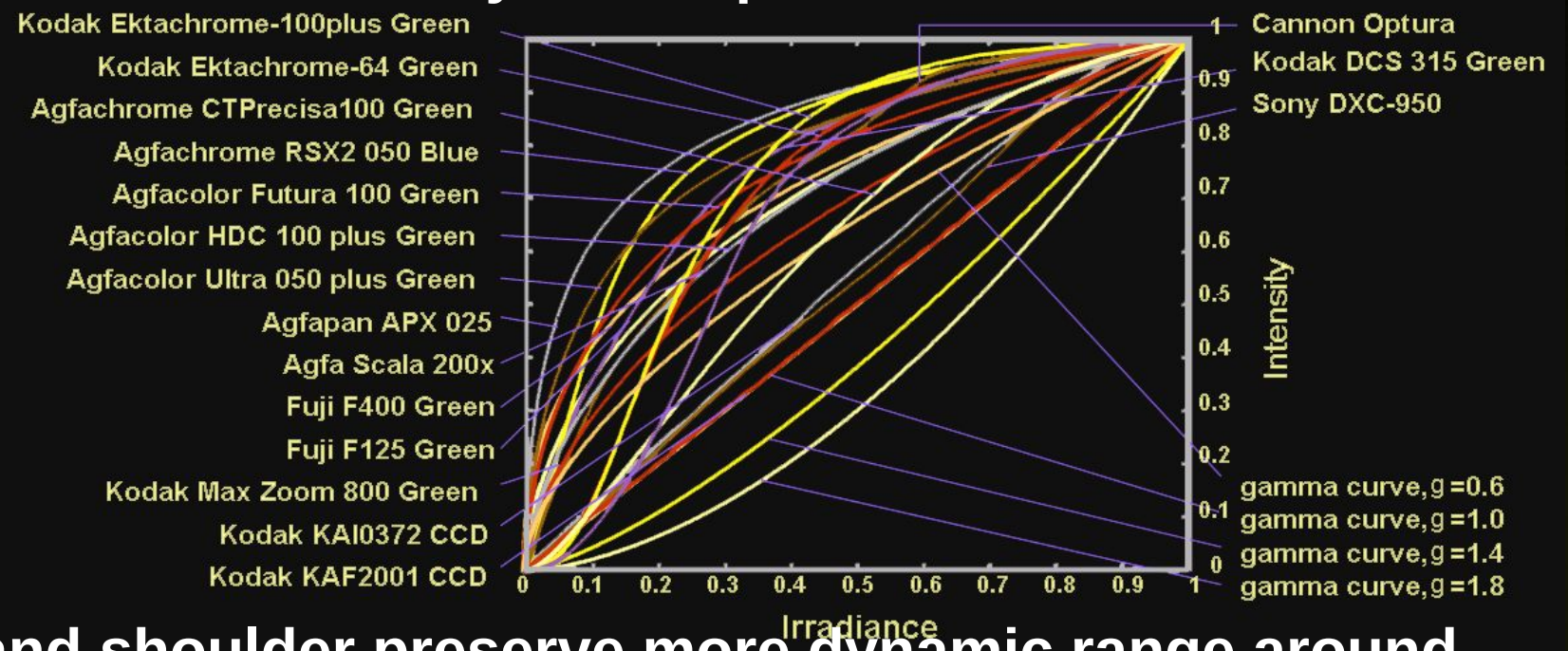
- Middle
  - follows a power function
  - if a given amount of light turned half of a grain crystals to silver, the same amount turns again half of the rest
- Toe region
  - the chemical process is just starting
- Shoulder region
  - close to saturation
- Film has more dynamic range than print
  - ~12bits



The 'geography' of the characteristic curve of a negative material

Digital camera response curve

- Digital cameras modify the response curve



- Toe and shoulder preserve more dynamic range around dark and bright areas, at the cost of reduced contrast
- May use different response curves at different exposures
  - impossible to calibrate and invert!

Auto White Balance

- The dominant light source (*illuminant*) produces a color cast that affects the appearance of the scene objects
- The color of the illuminant determines the color normally associated with white by the human visual system
- Auto White Balance
  - Identify the illuminant color
  - Neutralize the color of the illuminant

(source: www.cambridgeincolour.com)

Identify the color of the illuminant

- Prior knowledge about the ambient light
  - Candle flame light (18500K)
  - Sunset light (20000K)
  - Summer sunlight at noon (54000K)
  - …
- Known reference object in the picture
  - best: find something that is white or gray
- Assumptions about the scene
  - Gray world assumption (gray in sRGB space!)

Best way to do white balance

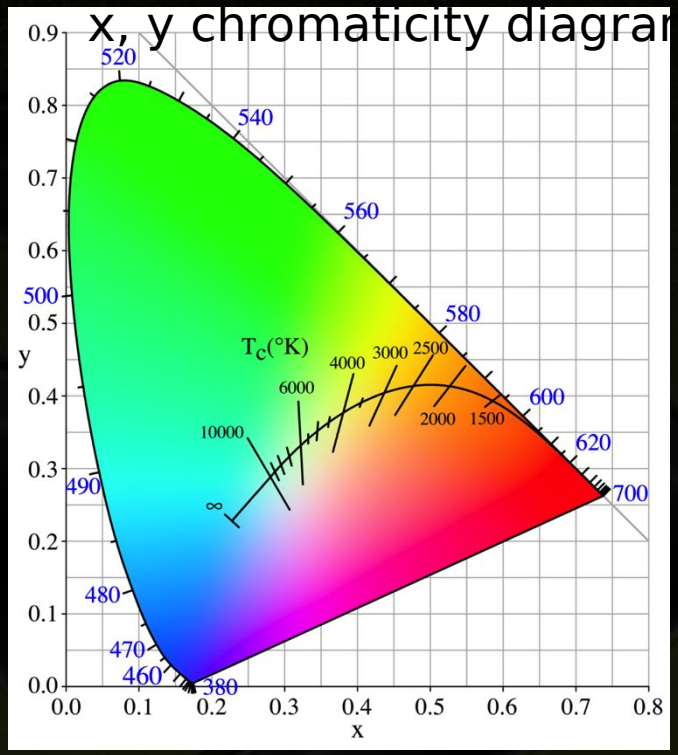
- Grey card
  - take a picture of a neutral object (white or gray)
  - deduce the weight of each channel

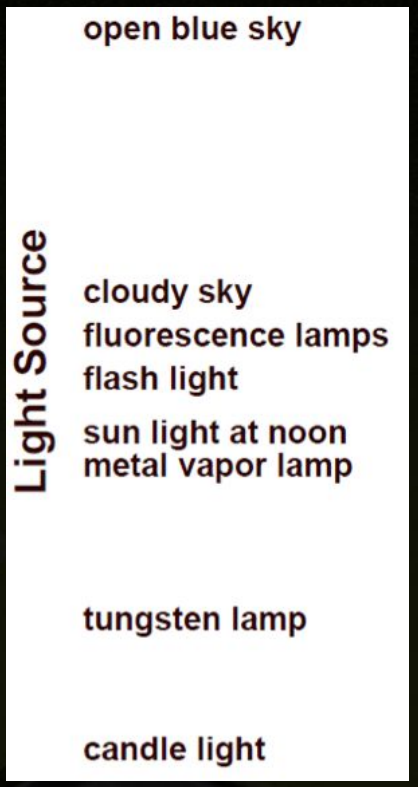
- If the object is recoded as r_w, g_w, b_w
  - use weights k/r_w, k/g_w, k/b_w
    - where k controls the exposure

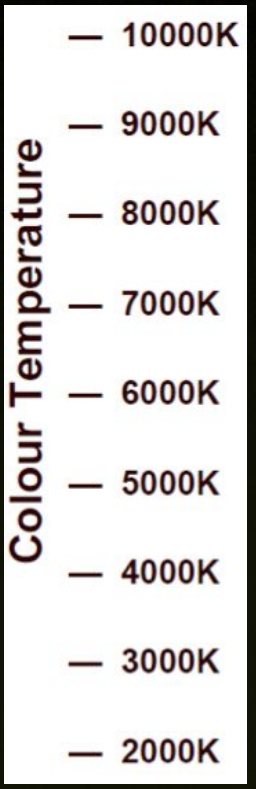
Brightest pixel assumption

- Highlights usually have the color of the light source
  - at least for dielectric materials
- White balance by using the brightest pixels
  - plus potentially a bunch of heuristics
  - in particular use a pixel that is not saturated / clipped

Color temperature







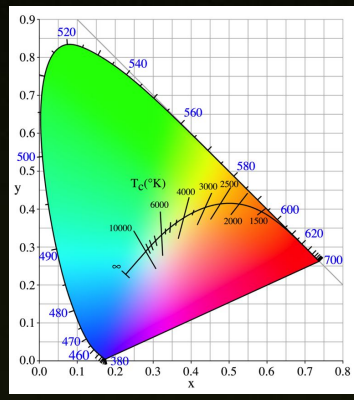
- Colors of a black-body heated at different temperatures fall on a curve (Planckian locus)
- Colors change non-linearly with temperature
  - but almost linearly with reciprocal temperatures 1/T

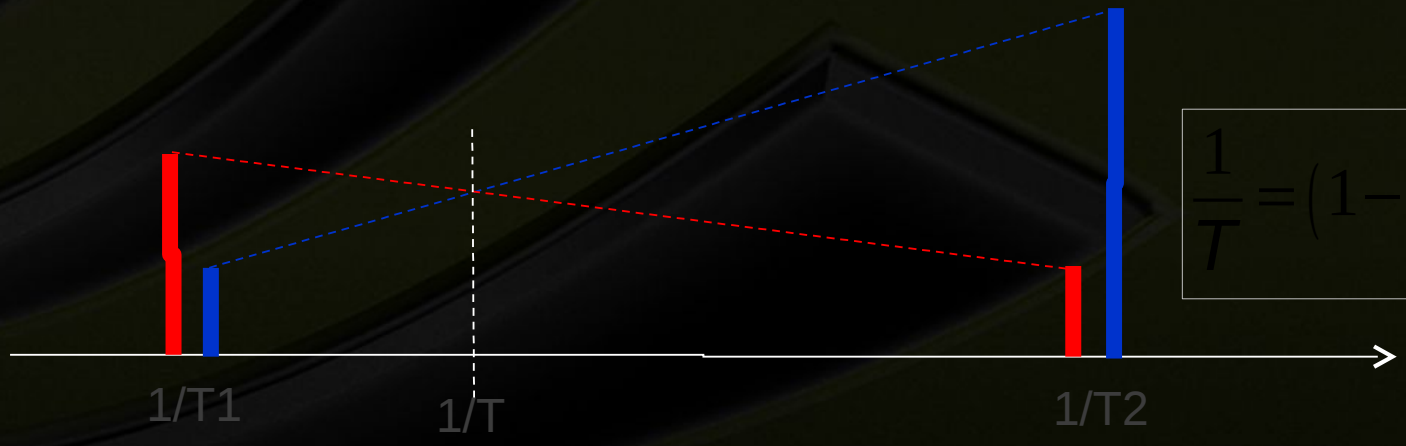
Mapping the colors

- For a given sensor
  - pre-compute the transformation matrices between the sensor color space and sRGB at different temperatures
  - FCam provides two precomputed transformations
    - for 3200oK and 7000oK
- Estimate a new transformation by interpolating between pre-computed matrices
  - ISP can apply the linear transformation

Estimating the color temperature

- Use scene mode
- Use gray world assumption (R = G = B) in sRGB space
  - really, just R = B, ignore G
- Estimate color temperature in a given image
  - apply pre-computed matrix to get sRGB for T1 and T2
  - calculate the average values R, B
  - solve α, use to interpolate matrices (or 1/T)



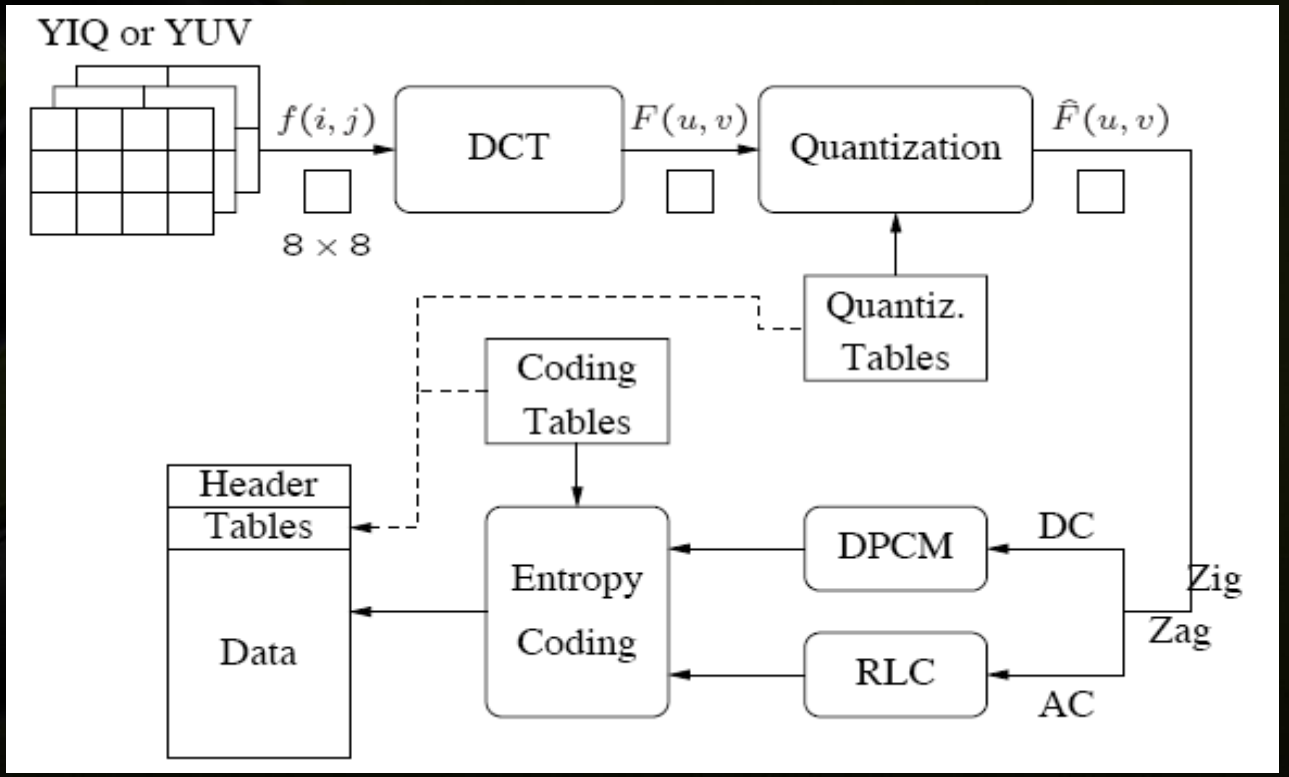


$$\frac{1}{T} = (1-\alpha)\frac{1}{T_1} + \alpha\frac{1}{T_2}$$

$$R = (1-\alpha)R_1 + \alpha R_2 \qquad B = (1-\alpha)B_1 + \alpha B_2$$

JPEG Encoding

1. Transform RGB to YUV or YIQ and subsample color
2. DCT on 8x8 image blocks
3. Quantization
4. Zig-zag ordering and run-length encoding
5. Entropy coding



Converting RGB to YUV

- YUV is not required for JPEG compression
  - but it gives a better compression rate

```
Y =   0.299 * R + 0.587  * G + 0.114 * B
U = -0.1687 * R – 0.3313 * G + 0.5   * B + 128
V =     0.5 * R – 0.4187 * G – 0.813 * B + 128
```

DCT

- The frequency domain is a better representation
  - it makes it possible to separate out information that isn’t very important to human perception
  - the human eye is not very sensitive to high frequency changes

DCT on Image Blocks

- The frequency domain is a better representation
  - it makes it possible to separate out information that isn't very important to human perception
  - the human eye is not very sensitive to high frequency changes
- The image is divided up into 8x8 blocks
  - 2D DCT is performed independently on each block
  - This is why, when a high degree of compression is requested, JPEG gives a "blocky" image result

Quantization

- Quantization in JPEG aims at reducing the total number of bits in the compressed image
  - Divide each entry in the frequency space block by an integer, then round
  - Use a quantization matrix $Q(u, v)$

$$\hat{F}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

Quantization

- Use larger entries in *Q* for the higher spatial frequencies
  - the lower right part of the matrix
- Based on psycho-physical studies
  - maximize compression while minimizing perceptual distortion
- After division the entries are smaller
  - we can use fewer bits to encode them

Table 9.1 The Luminance Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Table 9.2 The Chrominance Quantization Table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Quantization

- Different quantization matrices allow the user to choose how much compression to use
  - trades off quality vs. compression ratio
  - more compression means larger entries in *Q*

Original and DCT coded block

An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i,j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u,v)$

Fig. 9.2: JPEG compression for a smooth image block.

Quantized and Reconstructed Blocks

$$\begin{bmatrix} 32 & 6 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$\tilde{F}(u,v)$

»

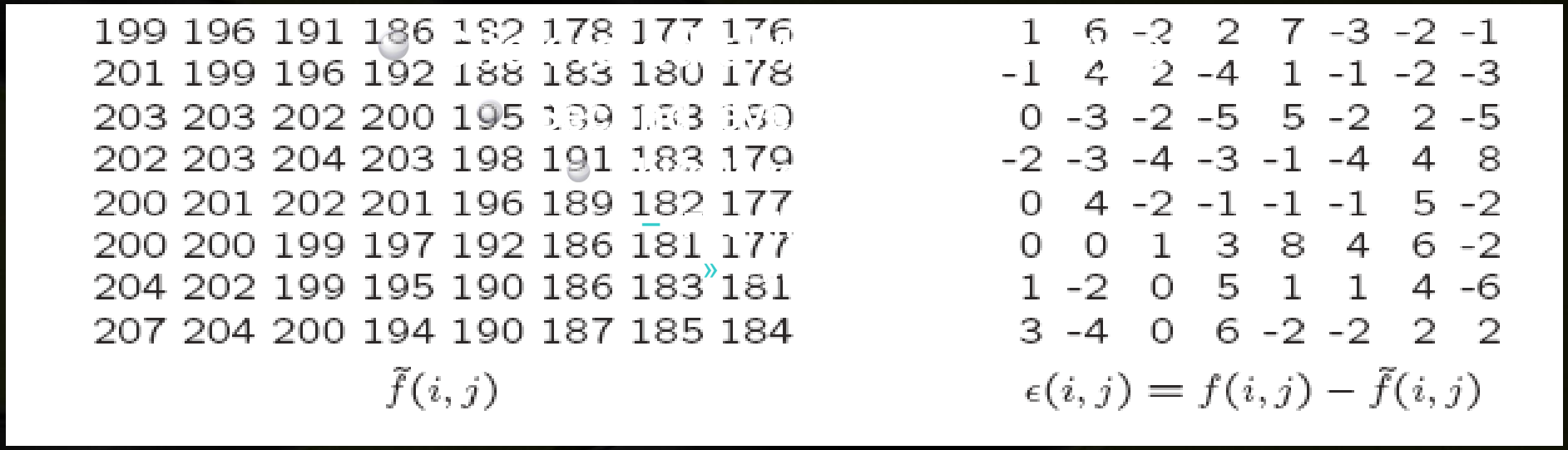
$$\begin{bmatrix} 512 & 66 & -10 & 0 & 0 & 0 & 0 & 0 \\ -12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\ -14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$\tilde{F}(u,v)$

Table 9.1 The Luminance Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

After IDCT and Difference from Original



$$\tilde{f}(i,j) = \begin{bmatrix} 199 & 196 & 191 & 186 & 182 & 178 & 177 & 176 \\ 201 & 199 & 196 & 192 & 188 & 183 & 180 & 178 \\ 203 & 203 & 202 & 200 & 195 & 189 & 183 & 180 \\ 202 & 203 & 204 & 203 & 198 & 191 & 183 & 179 \\ 200 & 201 & 202 & 201 & 196 & 189 & 182 & 177 \\ 200 & 200 & 199 & 197 & 192 & 186 & 181 & 177 \\ 204 & 202 & 199 & 195 & 190 & 186 & 183 & 181 \\ 207 & 204 & 200 & 194 & 190 & 187 & 185 & 184 \end{bmatrix}$$

$$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j) = \begin{bmatrix} 1 & 6 & -2 & 2 & 7 & -3 & -2 & -1 \\ -1 & 4 & 2 & -4 & 1 & -1 & -2 & -3 \\ 0 & -3 & -2 & -5 & 5 & -2 & 2 & -5 \\ -2 & -3 & -4 & -3 & -1 & -4 & 4 & 8 \\ 0 & 4 & -2 & -1 & -1 & -1 & 5 & -2 \\ 0 & 0 & 1 & 3 & 8 & 4 & 6 & -2 \\ 1 & -2 & 0 & 5 & 1 & 1 & 4 & -6 \\ 3 & -4 & 0 & 6 & -2 & -2 & 2 & 2 \end{bmatrix}$$

Same steps on a less homogeneous block

Another 8×8 block from the Y image of 'Lena'

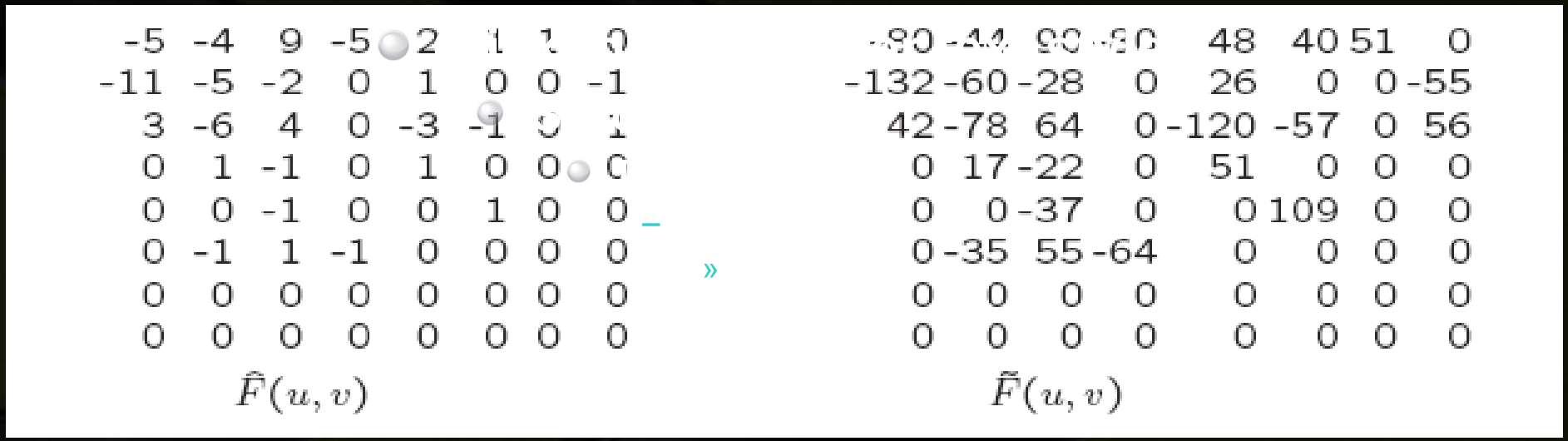
70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

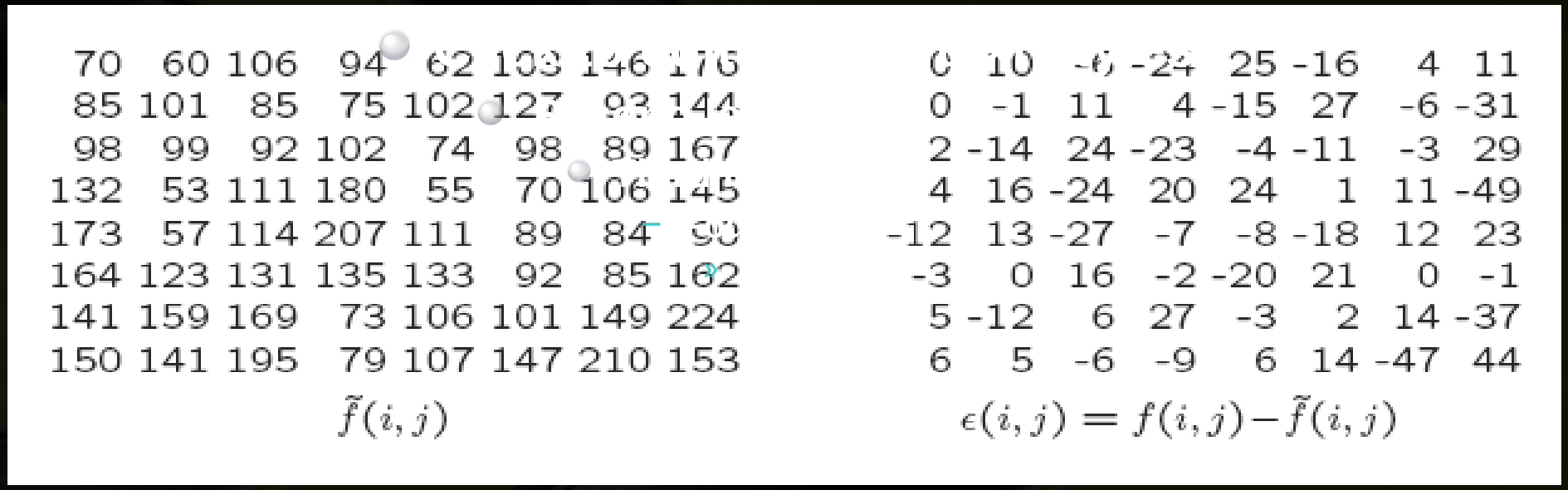
-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

Steps 2 and 3

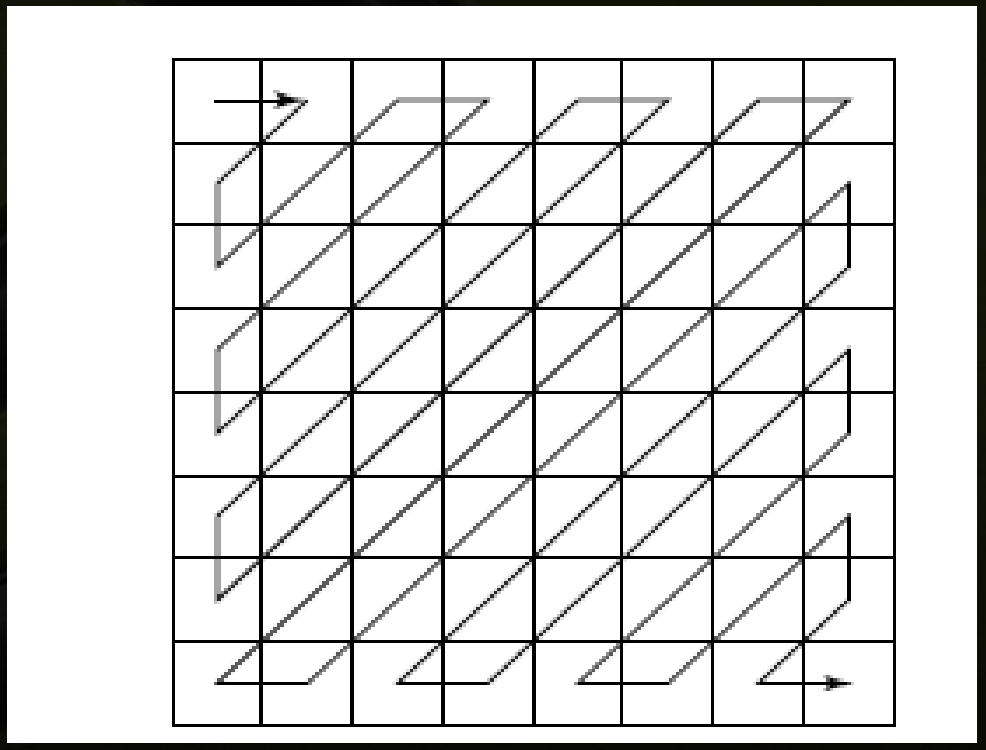


IDCT and Difference



Run-Length Coding

- The AC and DC components are treated differently
- After quantization we have many 0 AC components
  - and most of the zero components are towards the lower right corner (high spatial frequencies)
- To take advantage of this, use zigzag scanning to create a 64-vector



Run-Length Coding

- Replace values in a 64-vector (previously an 8x8 block) by a pair (RUNLENGTH, VALUE)
  - where RUNLENGTH is the number of zeroes in the run
  - and VALUE is the next non-zero value

- From the first example we have
  (32, 6, -1, -1, 0, -1, 0, 0, 0, -1, 0, 0, 1, 0, 0, …, 0)

- This becomes
  (0,6) (0,-1) (0,-1) (1,-1) (3,-1) (2,1) (0,0)

Entropy Coding

- The coefficients are then entropy coded
  - mostly using Huffman coding
  - for DC coefficients, assumed to vary slowly, additionally Differential Pulse Code Modulation (DPCM) is used:
    if the first five DC coefficients are
    150, 155, 149, 152, 144,
    we come up with DPCM code
    150, 5, -6, 3, -8
- These additional data compression steps are lossless
  - most of the lossiness is in the quantization step

Alternatives?

- JPEG 2000
  - ISO, 2000
  - better compression, inherently hierarchical, random access, …
  - but much more complex than JPEG
- JPEG XR
  - Microsoft, 2006; ISO / ITU-T, 2010
  - good compression, supports tiling (random access without having to decode whole image), better color accuracy (incl. HDR), transparency, compressed domain editing
- But JPEG stays
  - too large an install base