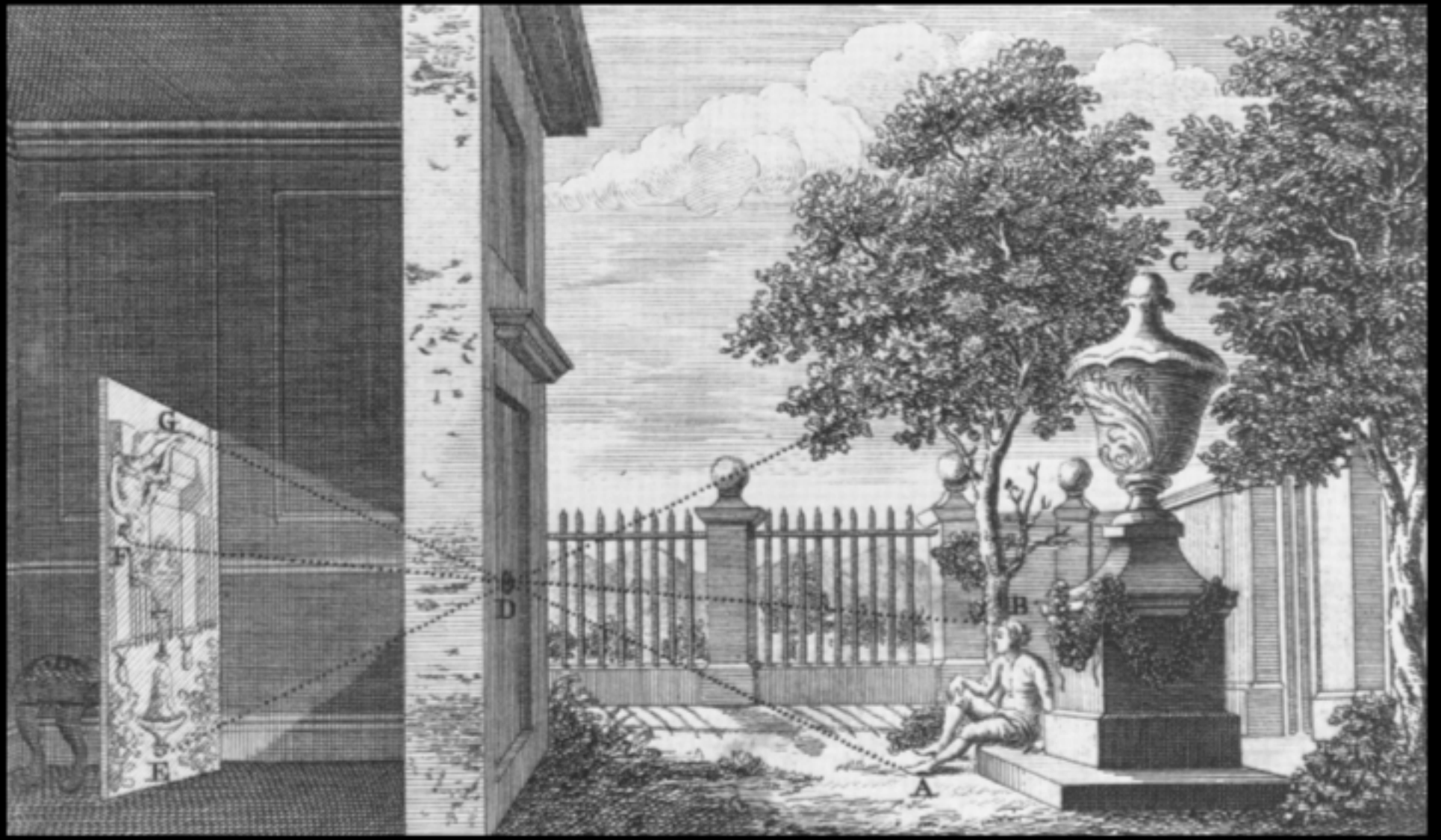
Mobile Cameras and some apps

Kari Pulli
VP Computational Imaging
Light

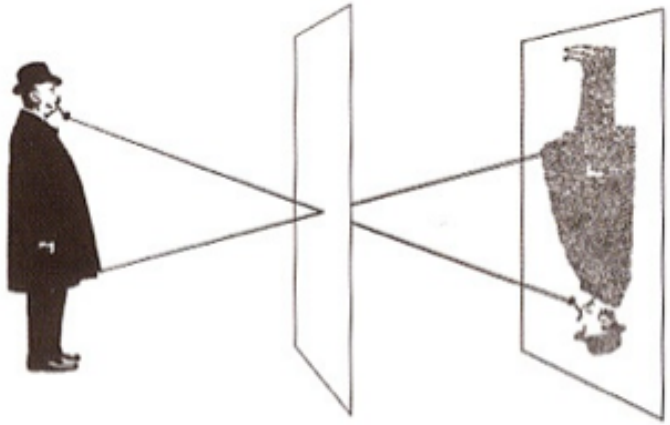
Pinhole camera (a.k.a. camera obscura)



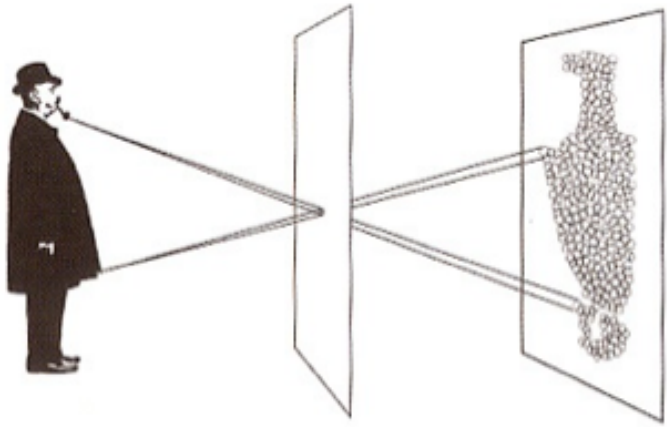
Linear perspective with viewpoint at pinhole

Effect of pinhole size

Photograph made with small pinhole

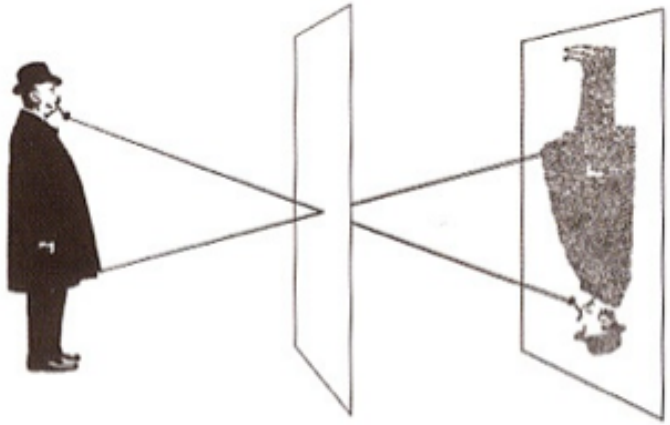


Photograph made with larger pinhole

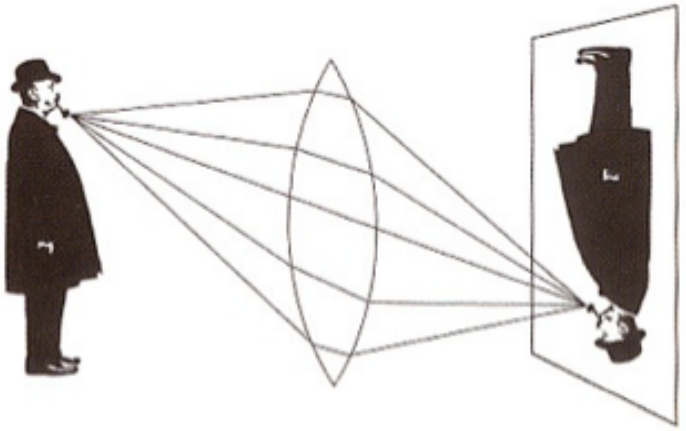


Add a lens to get more light

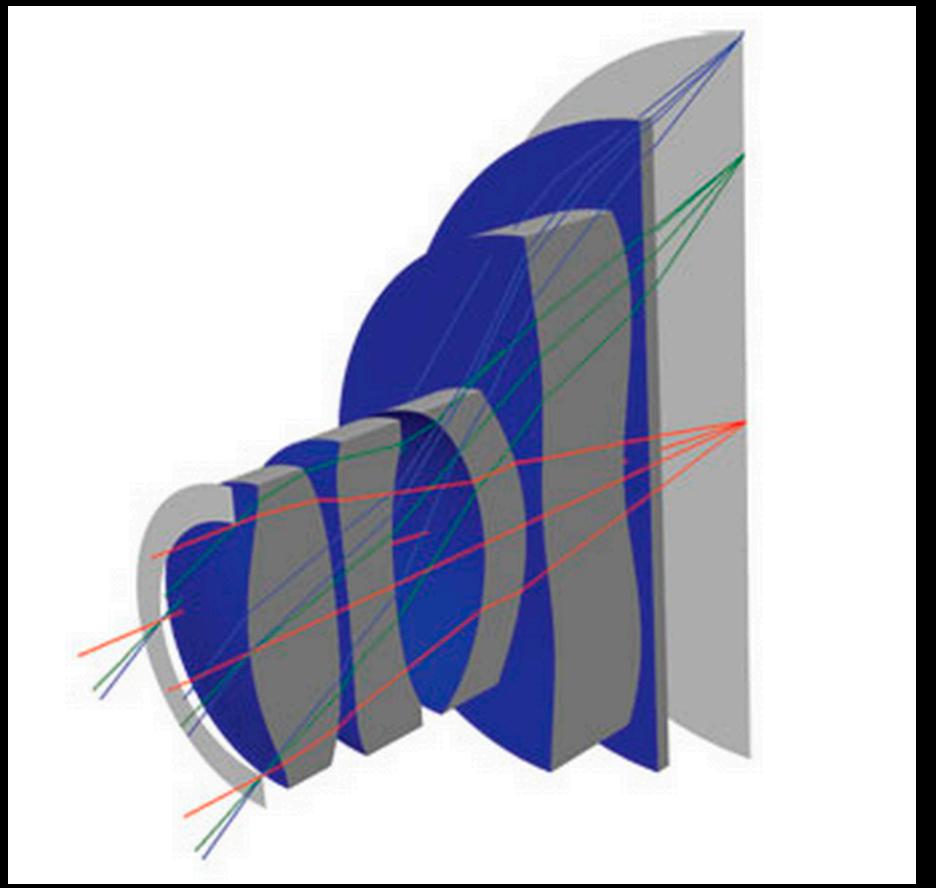
Photograph made with small pinhole

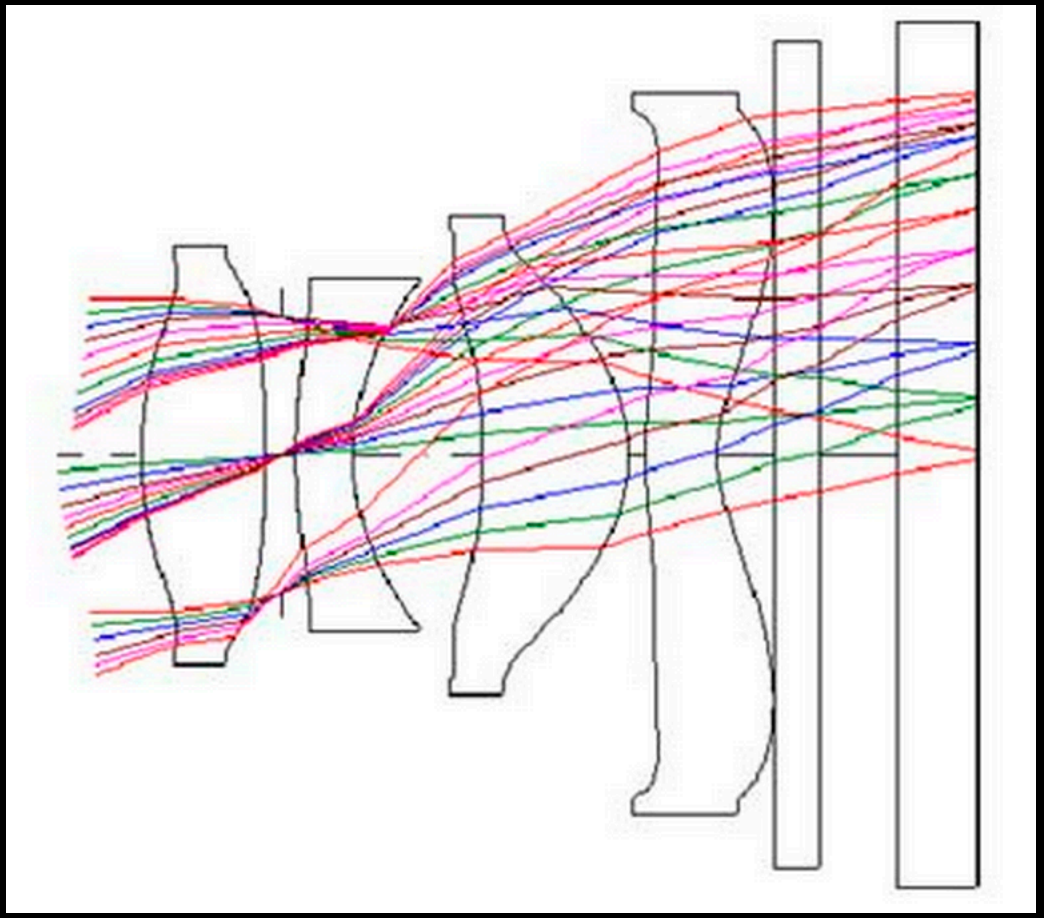


Photograph made with lens

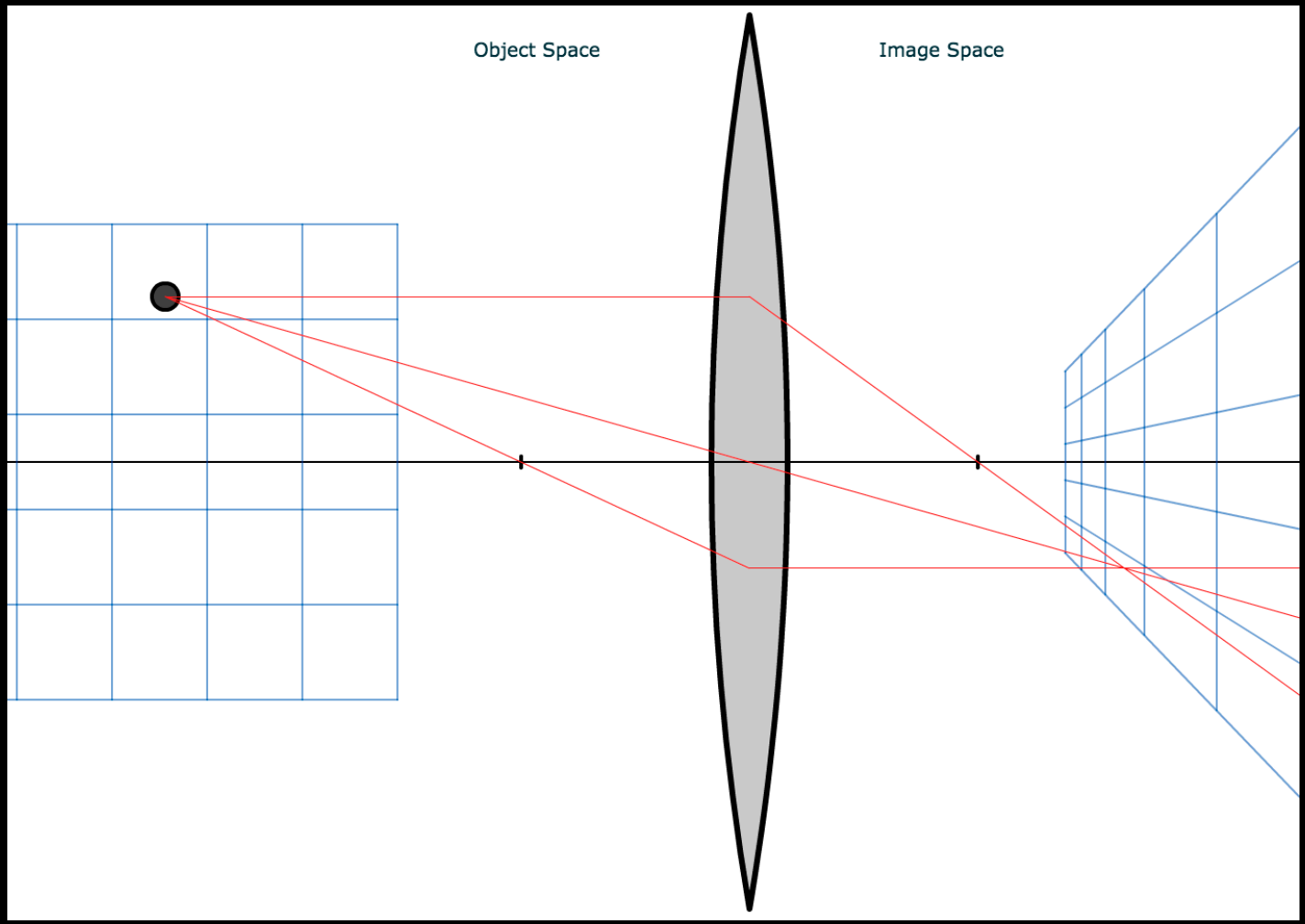


Real lenses are complex





Thin lens approximation: Gauss's ray diagram

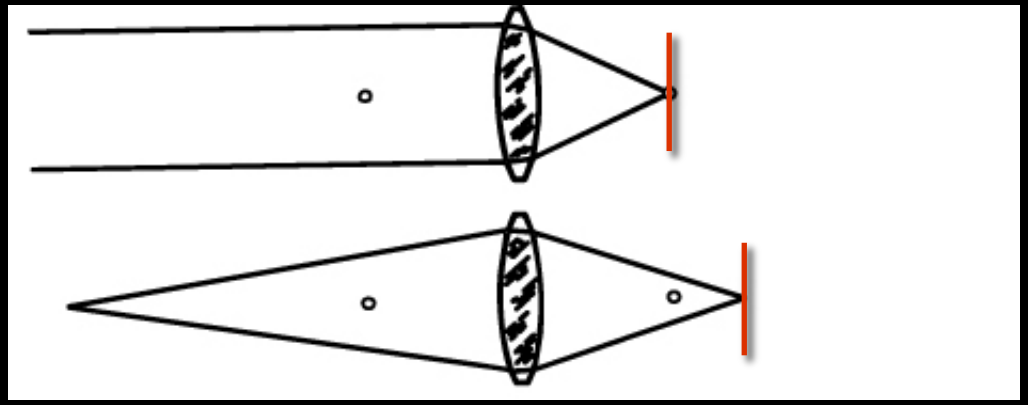


http://graphics.stanford.edu/courses/cs178/applets/thinlens.html

Changing the focus distance

$$\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}$$

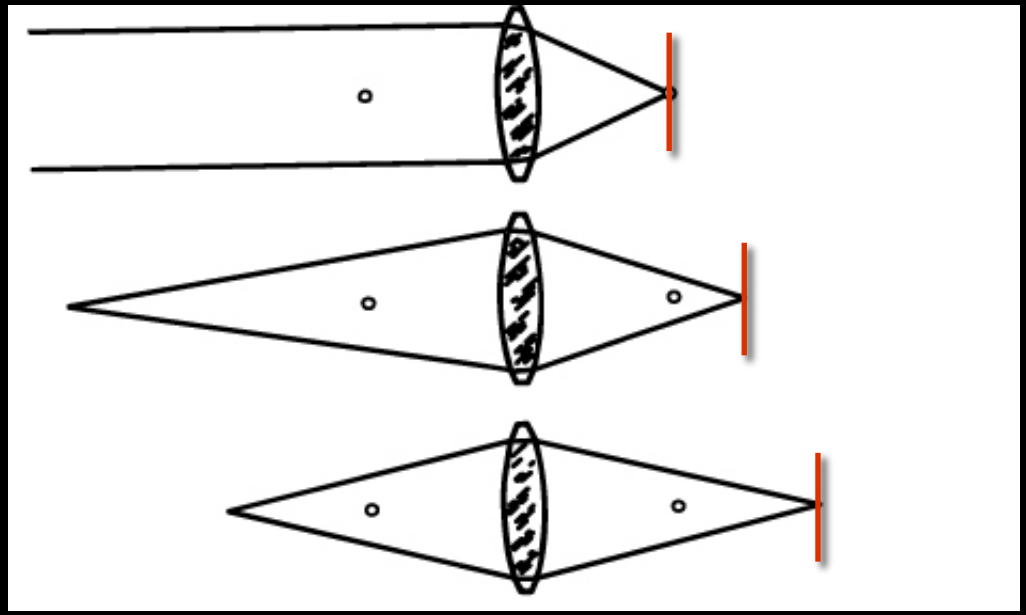
- To focus on objects at different distances
  - move sensor relative to the lens



Changing the focus distance

$$\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}$$

- To focus on objects at different distances
  - move sensor relative to the lens



- At $s_o = s_i = 2f$ we get 1:1 imaging (macro) because

$$\frac{1}{2f} + \frac{1}{2f} = \frac{1}{f}$$

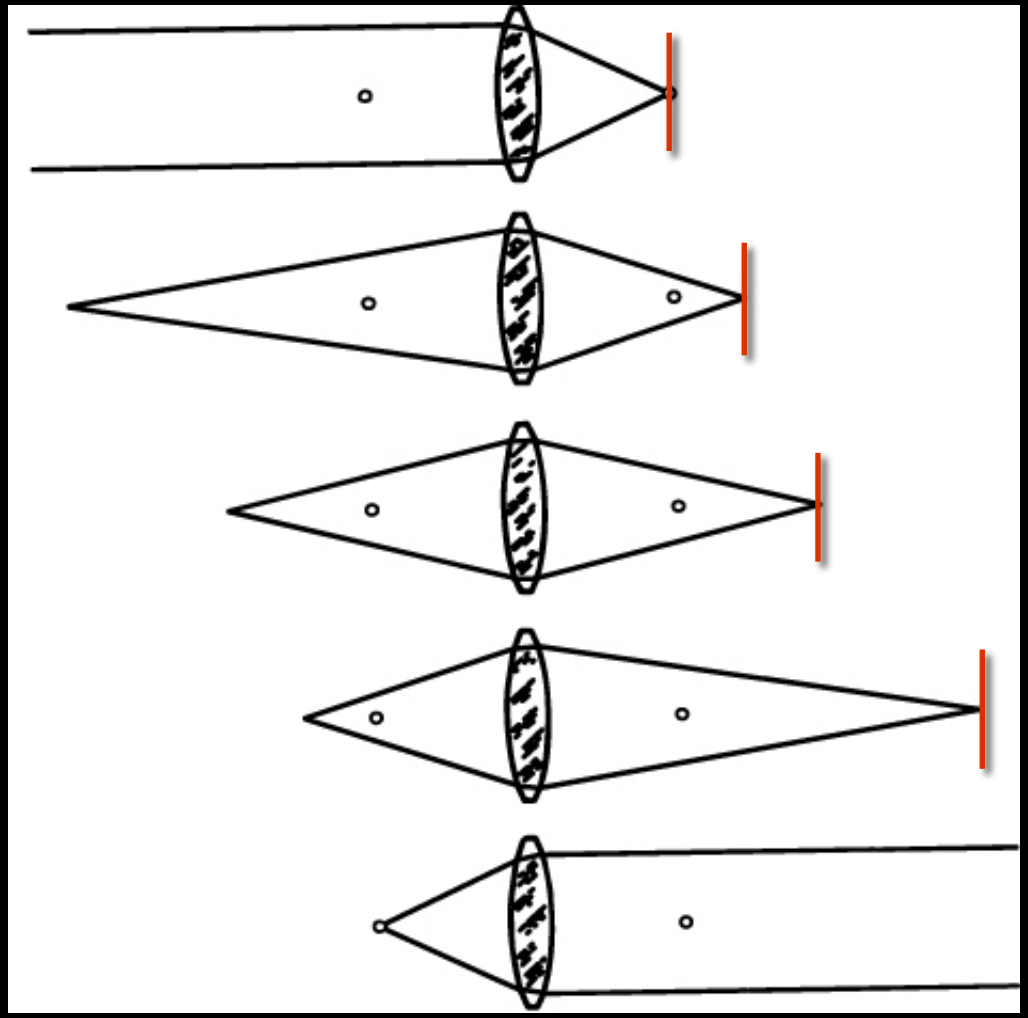
Changing the focus distance

$$\frac{1}{s_o} + \frac{1}{s_i} = \frac{1}{f}$$

- To focus on objects at different distances
  - move sensor relative to the lens
- At $s_o = s_i = 2f$ we get 1:1 imaging (macro) because

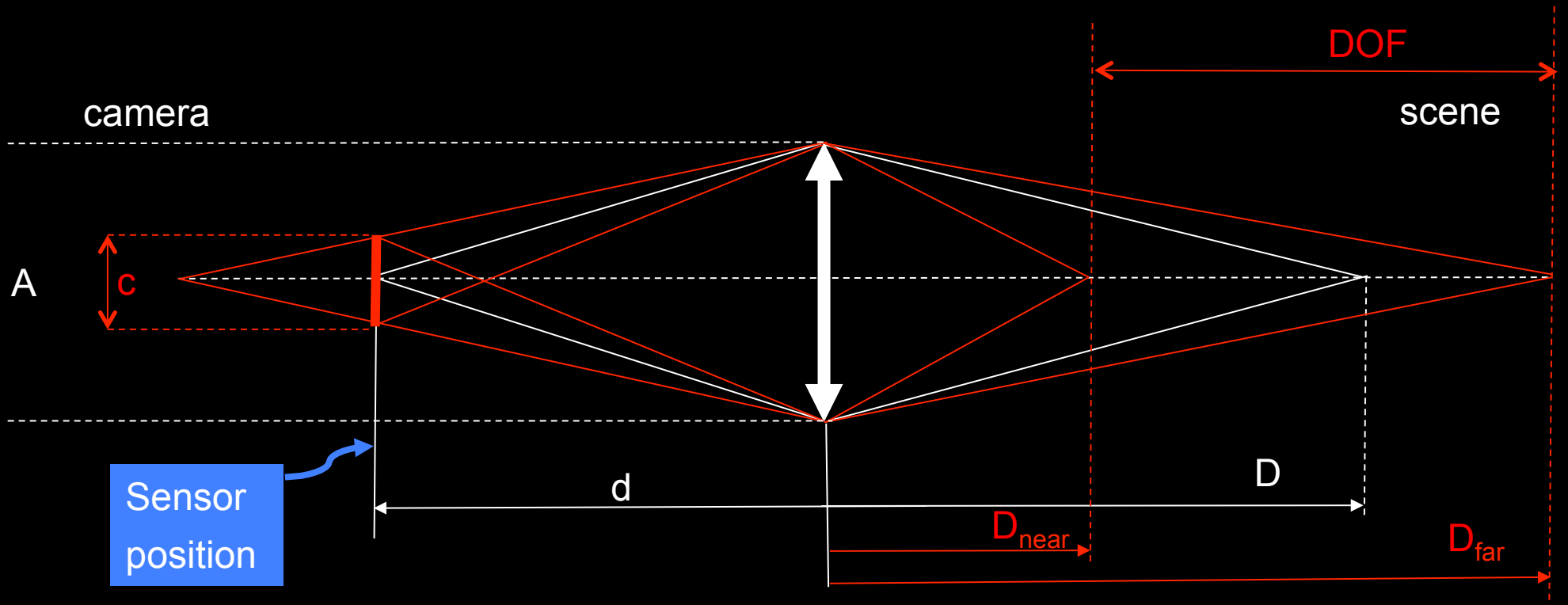
$$\frac{1}{2f} + \frac{1}{2f} = \frac{1}{f}$$

- Can't focus on objects closer to the lens than *f*



http://graphics.stanford.edu/courses/cs178/applets/gaussian.html

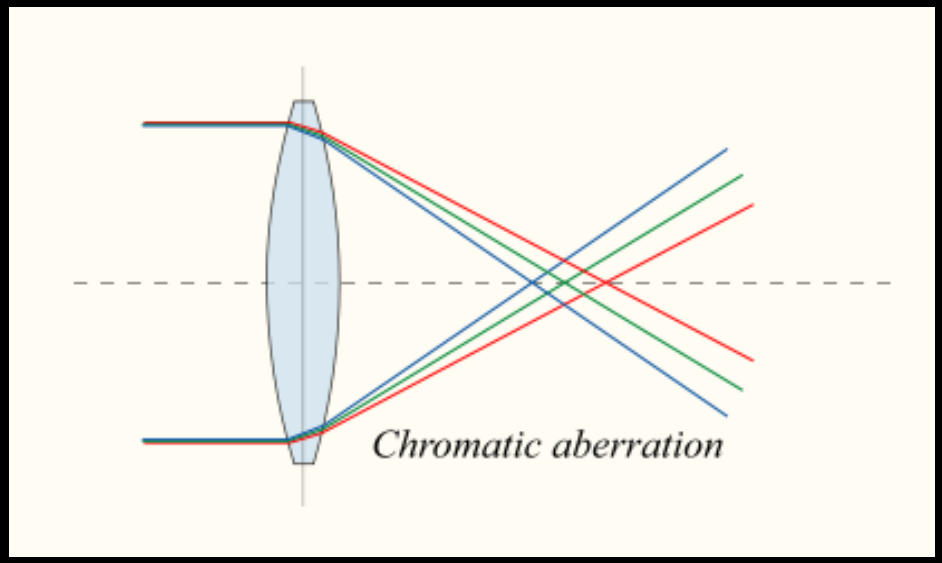
Focusing

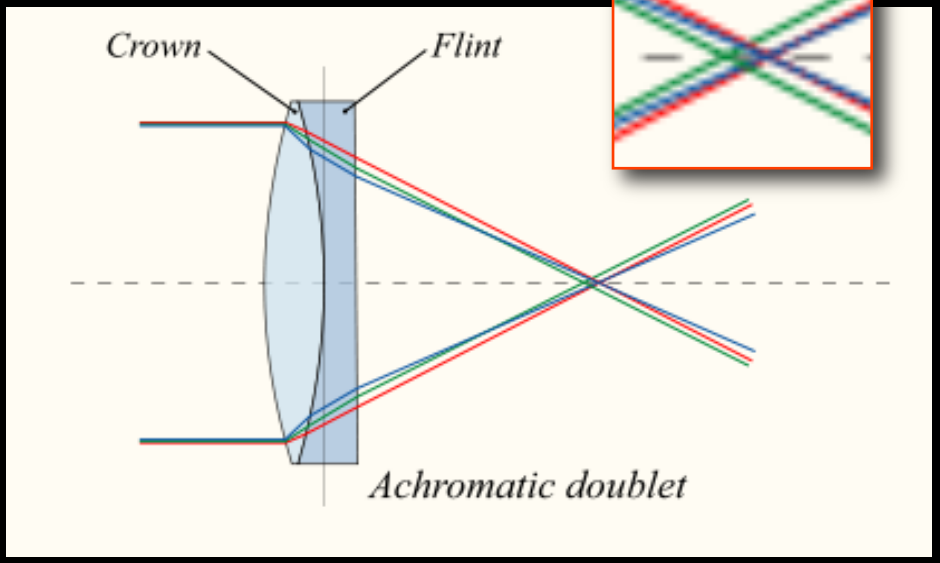


- Depth of field (DOF) = the range of distances that are in focus

http://graphics.stanford.edu/courses/cs178/applets/dof.html

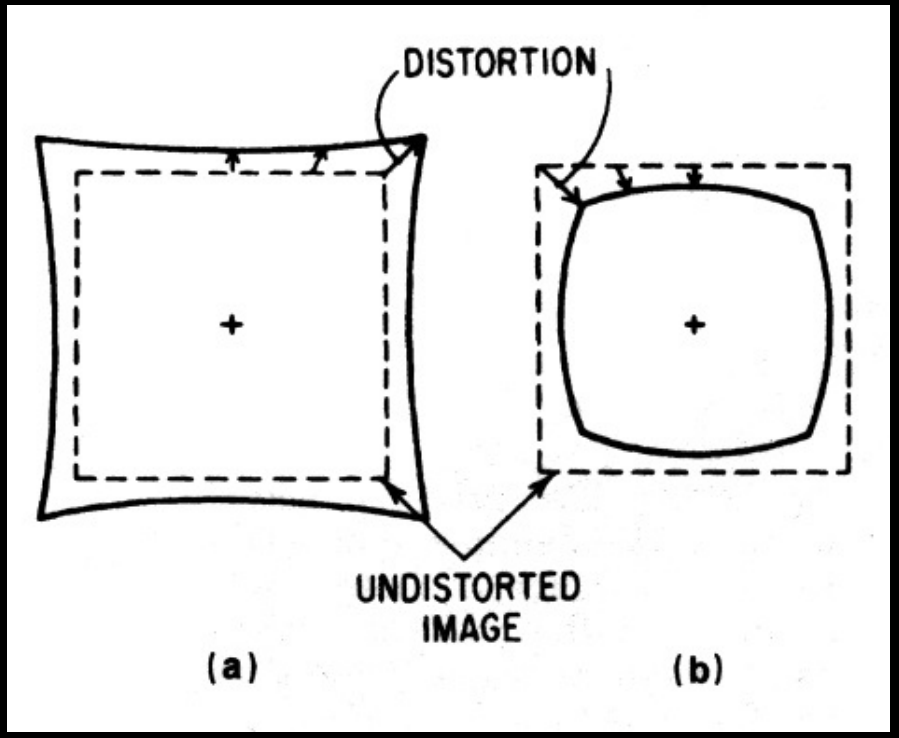
Chromatic aberration





- Different wavelengths refract at different rates
  - so have different focal lengths
- Correct with achromatic doublet
  - strong positive lens + weak negative lens
    = weak positive compound lens
  - align red and blue

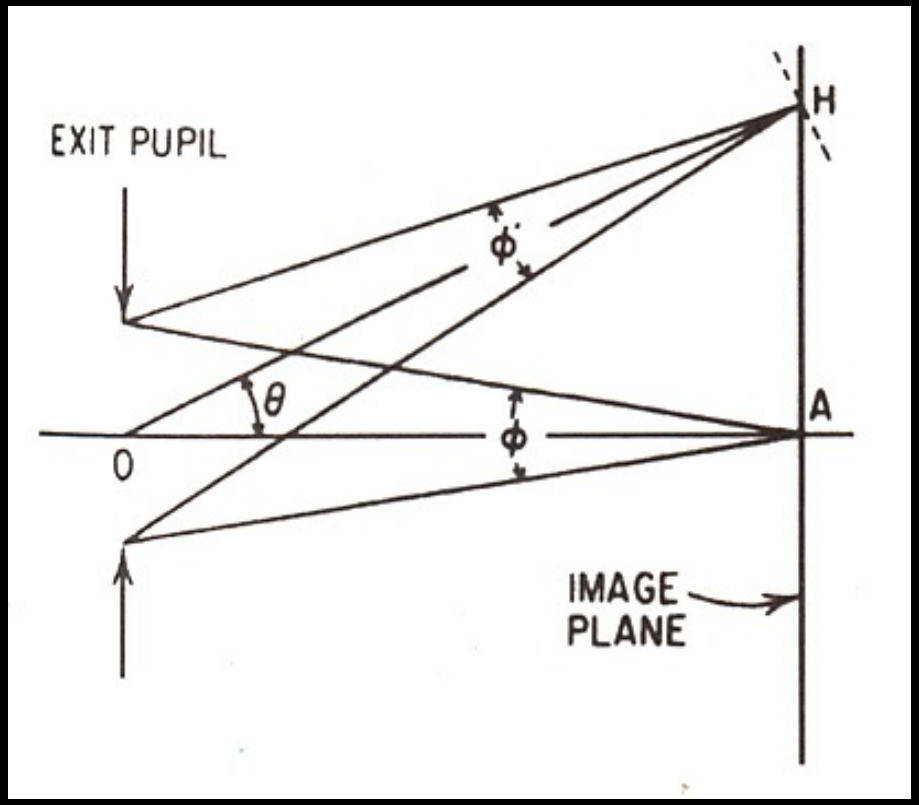
Lens distortion



- Radial change in magnification
  - (a) pincushion
  - (b) barrel distortion

Vignetting

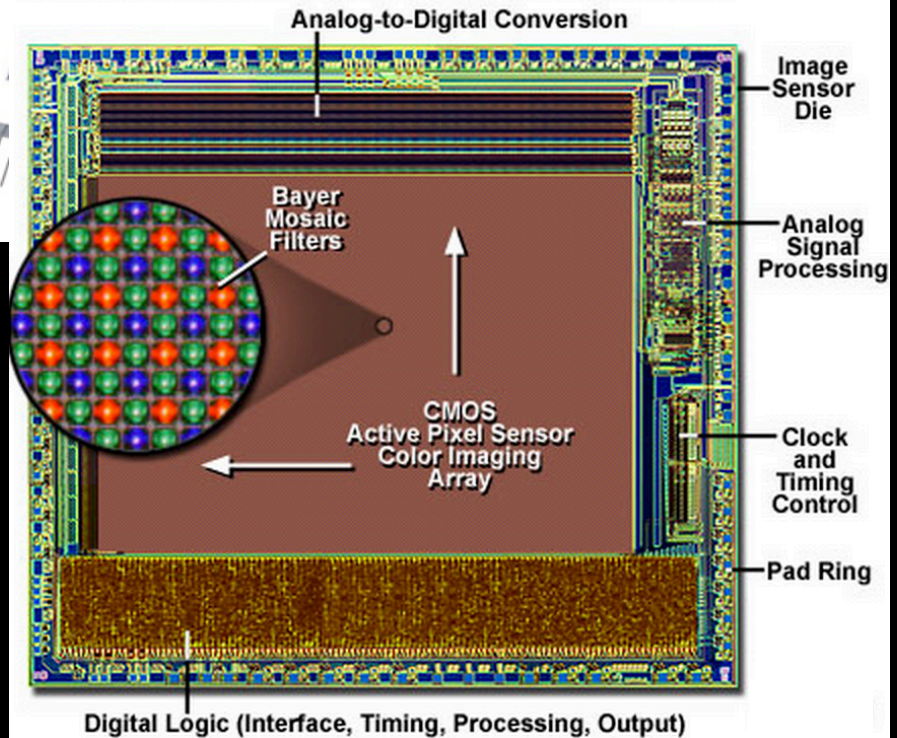
- Irradiance is proportional to
  - projected area of aperture as seen from pixel
  - projected area of pixel as seen from aperture
  - distance^2 from aperture to pixel
- Combining all these
  - each ~ a factor of cos θ
  - light drops as cos^4 θ

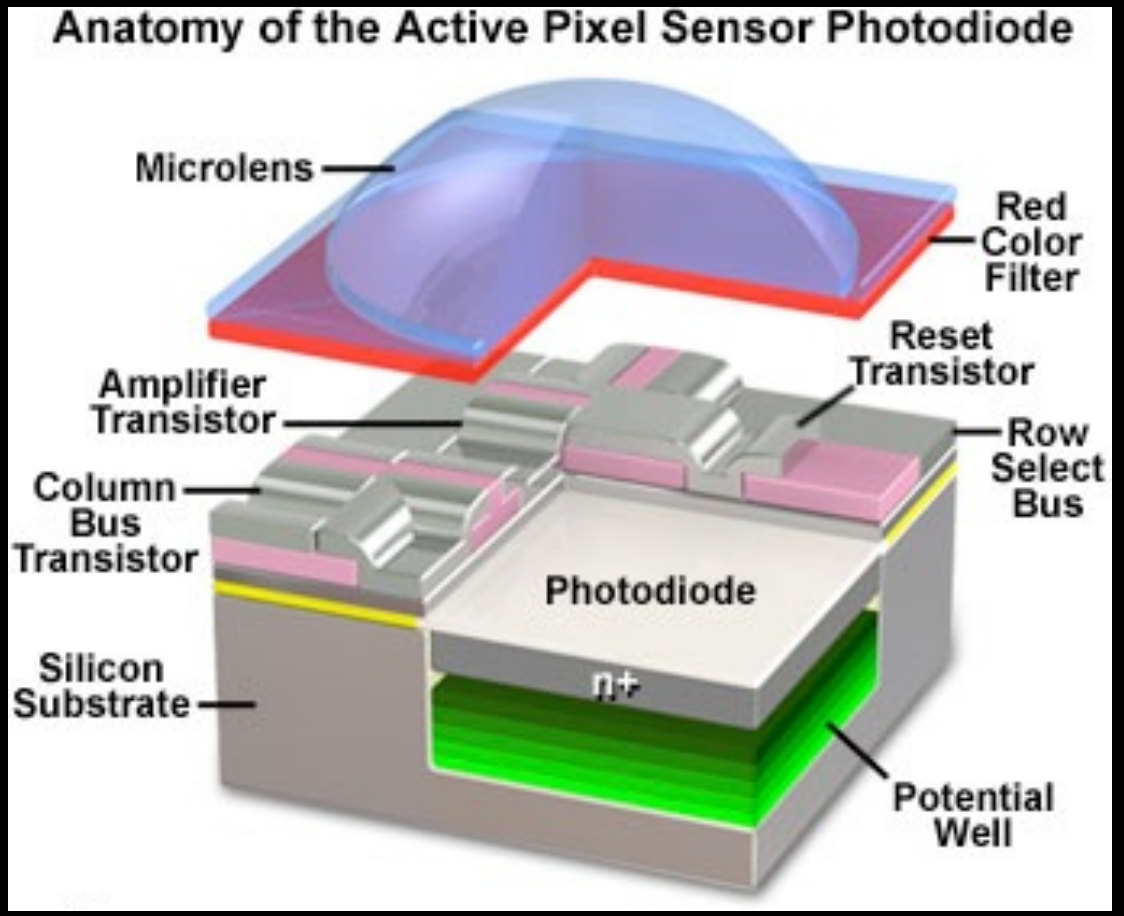


- Fix by calibrating
  - take a photo of a uniformly white object
  - the picture shows the attenuation, divide the pixel values by it

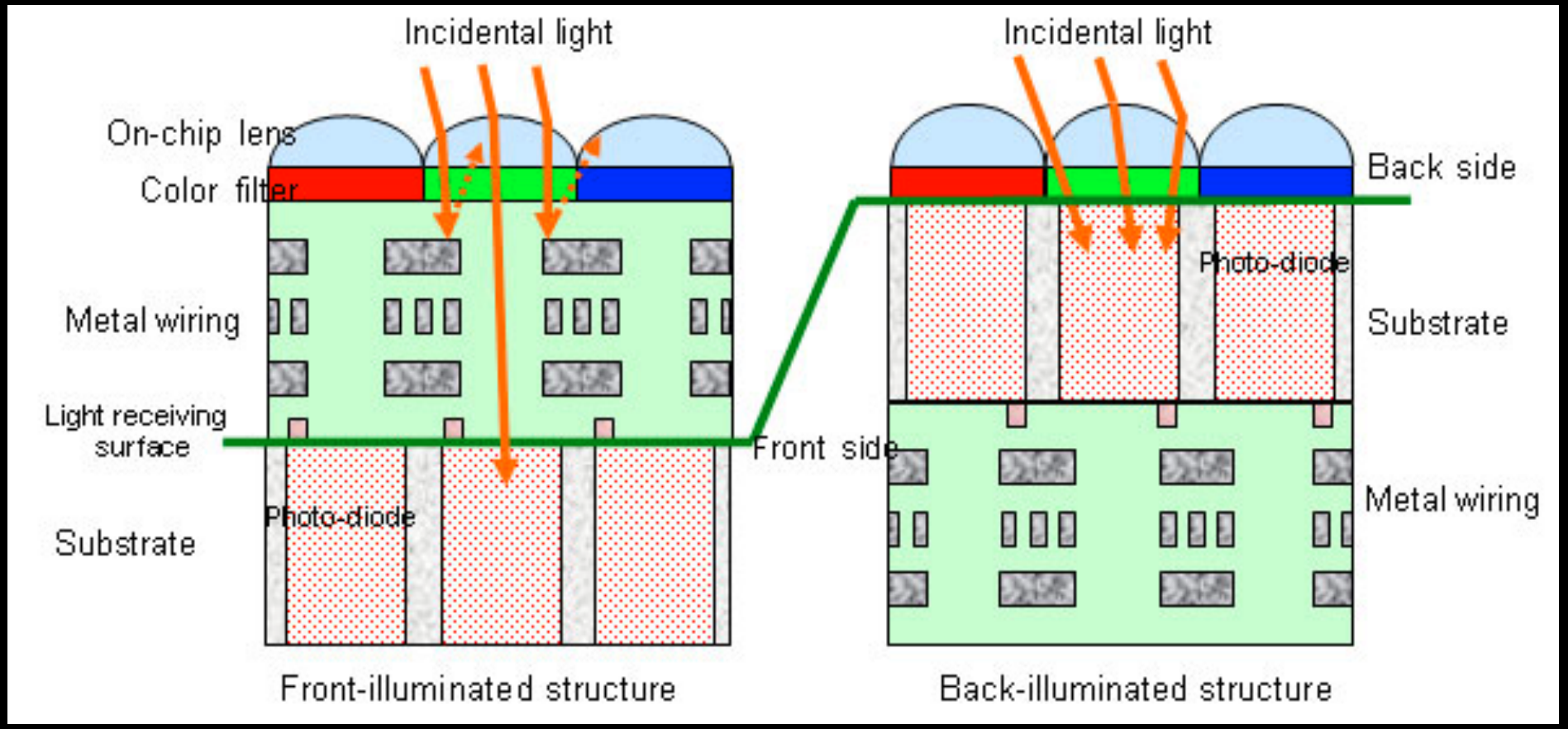
CMOS sensor

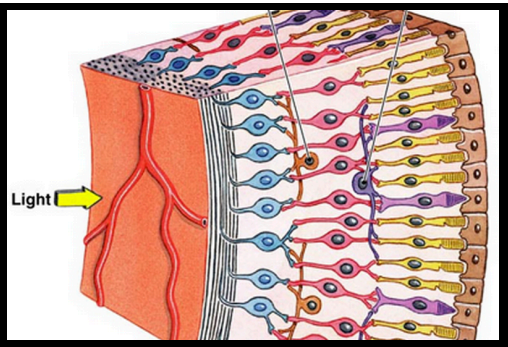
CMOS Image Sensor Integrated Circuit Architecture

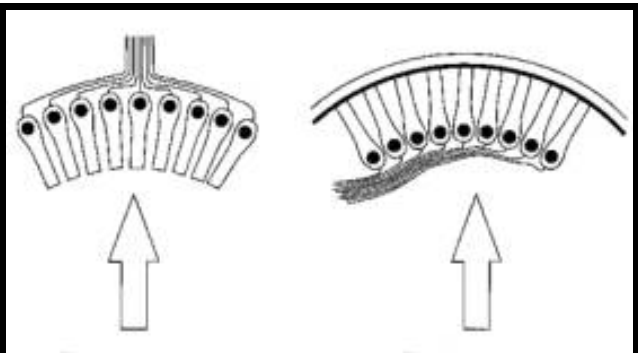




Front- vs. Back-illuminated sensor



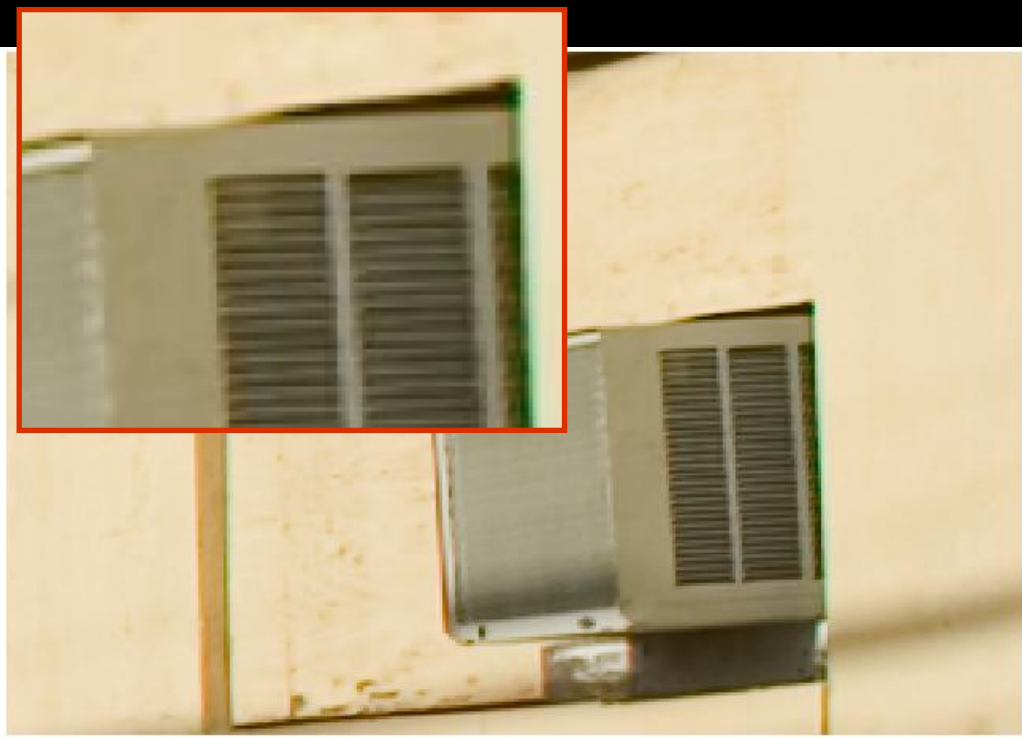




Anti-aliasing filter

- Two layers of birefrigent material
  - splits one ray into 4 rays

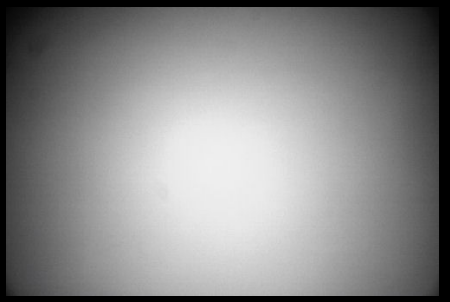
anti-aliasing filter removed



normal

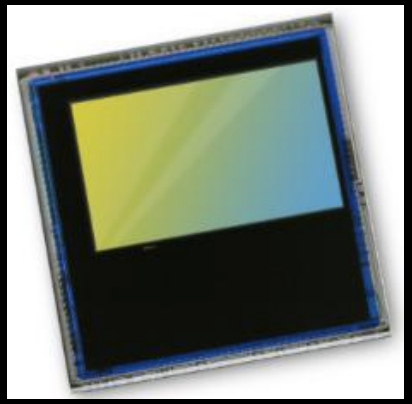
From “raw-raw” to RAW

- Pixel Non-Uniformity
  - each pixel has a slightly different sensitivity to light
    - typically within 1% to 2%
  - reduce by calibrating an image with a flat-field image
    - also eliminate the effects of vignetting and other optical variations



- Stuck pixels
  - some pixels are turned always on or off
  - identify, replace with filtered values

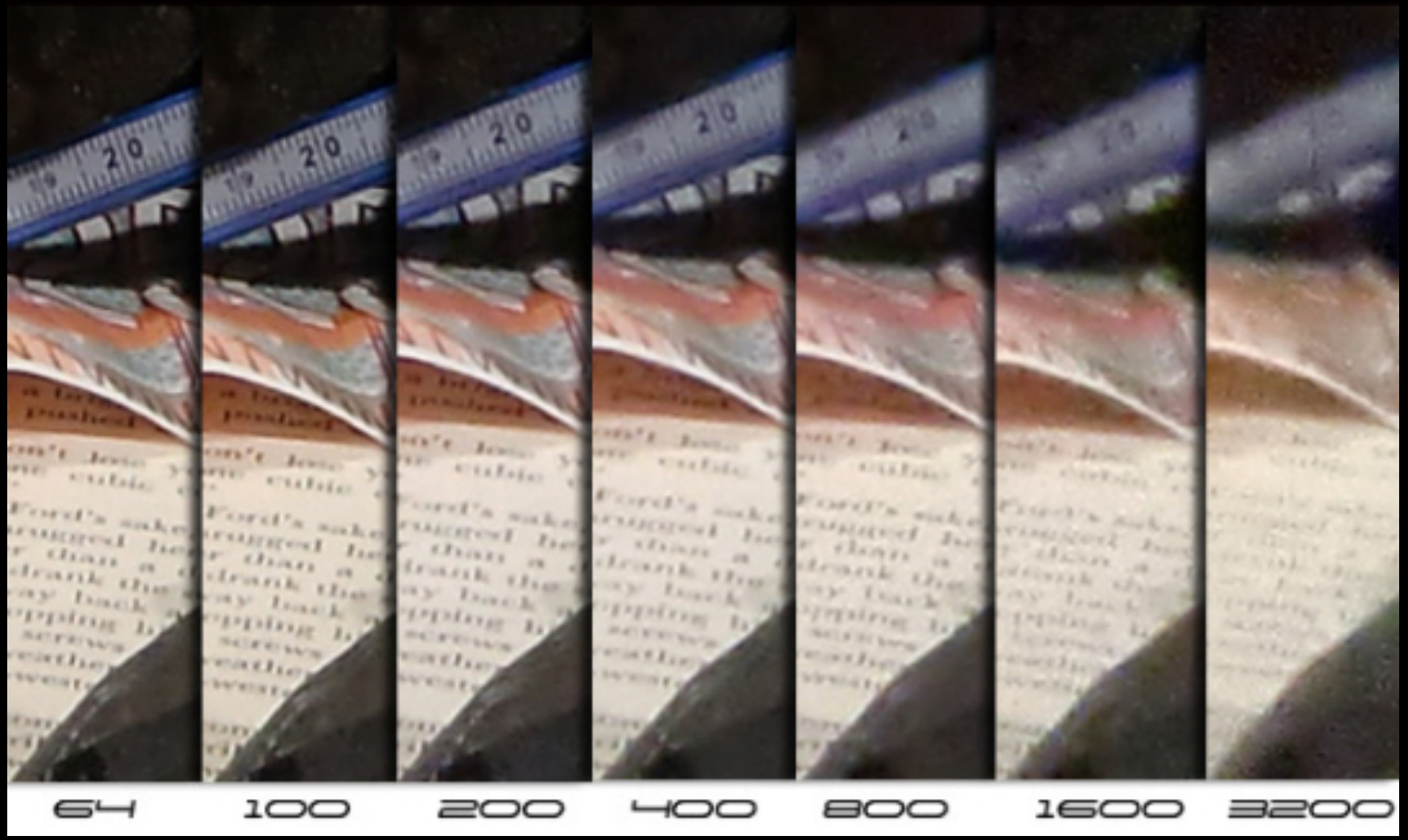
- Dark floor
  - temperature adds noise
  - sensors usually have a ring of covered pixels around the exposed sensor, subtract their signal



ISO = amplification in AD conversion

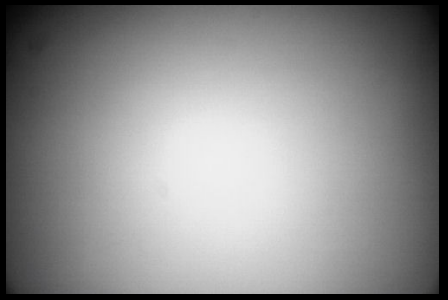
- Before conversion the signal can be amplified
  - ISO 100 means no amplification
  - ISO 1600 means 16x amplification
  - +: can see details in dark areas better
  - -: noise is amplified as well; sensor more likely to saturate

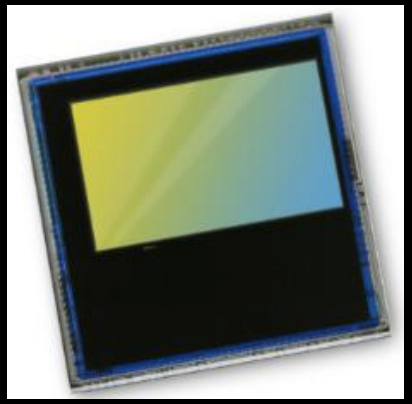
ISO



From “raw-raw” to RAW

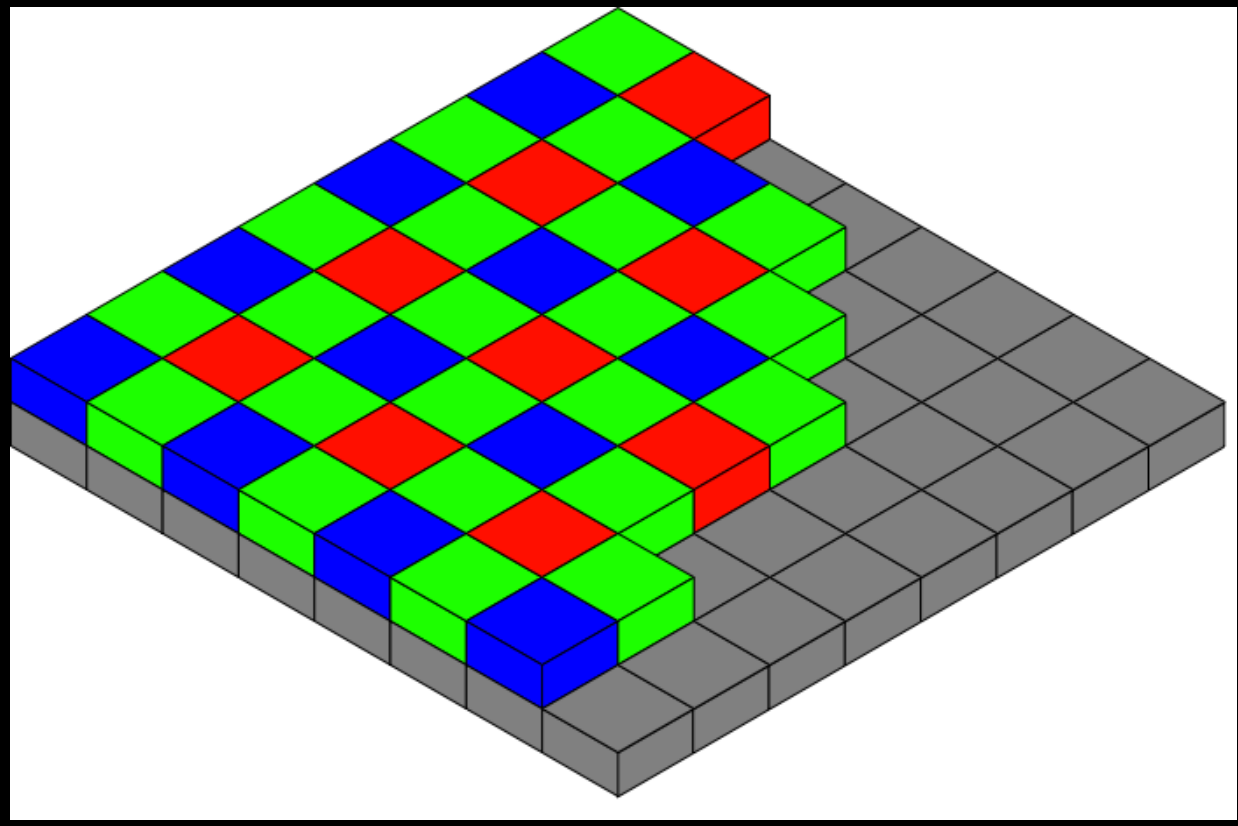
- Pixel Non-Uniformity
  - each pixel in a CCD has a slightly different sensitivity to light, typically within 1% to 2% of the average signal
  - can be reduced by calibrating an image with a flat-field image
  - flat-field images are also used to eliminate the effects of vignetting and other optical 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



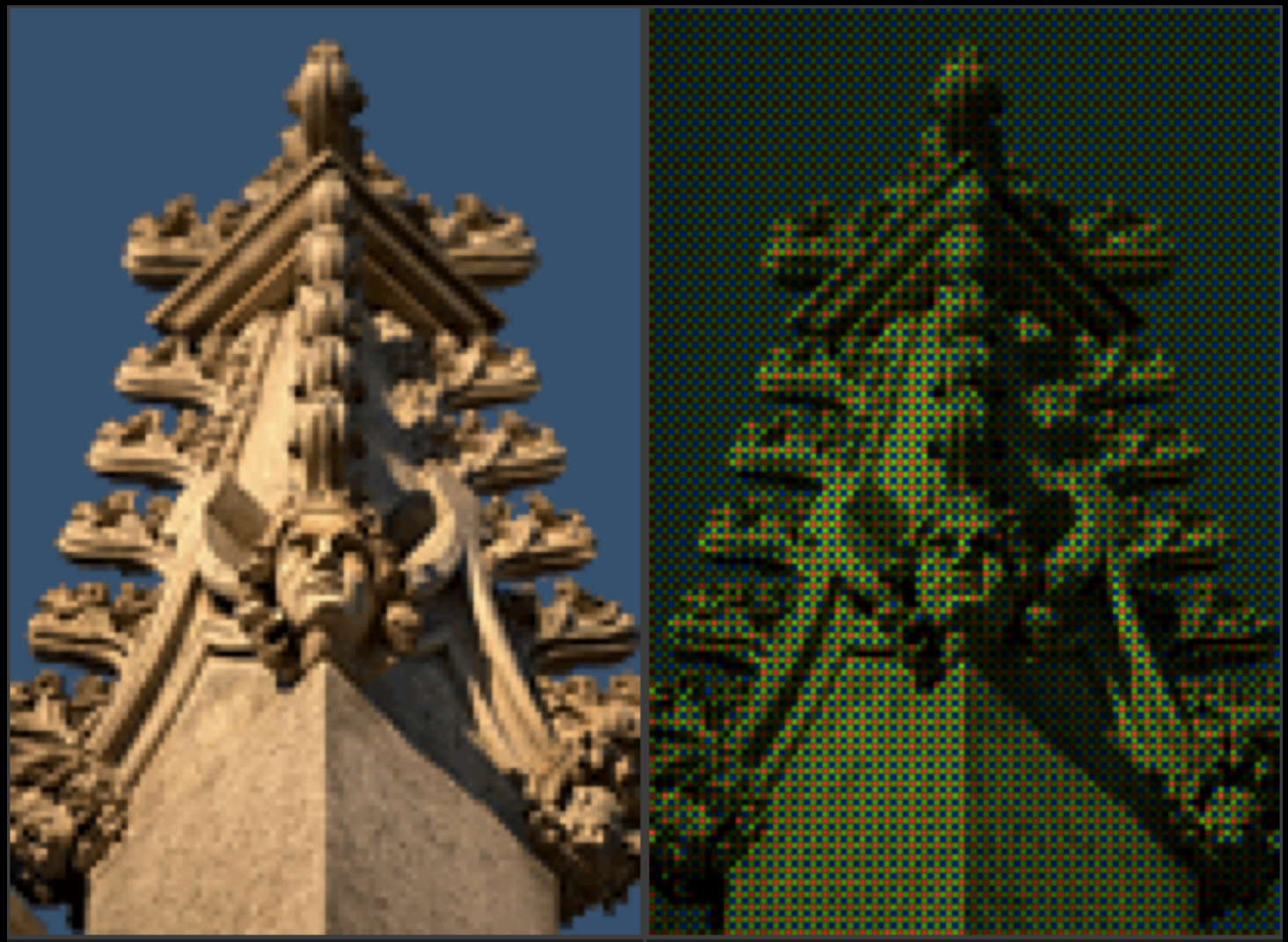


Color filter array

- Bayer pattern



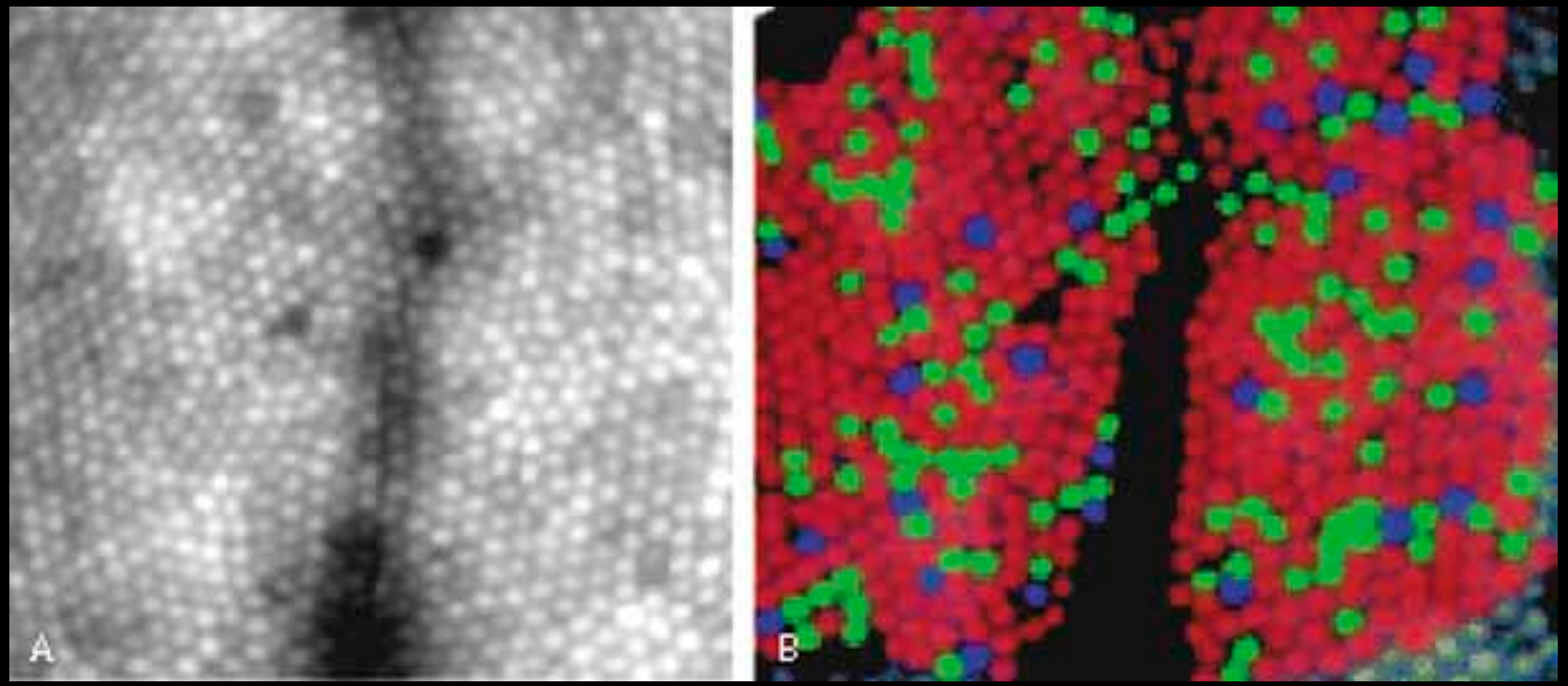
Demosaicking

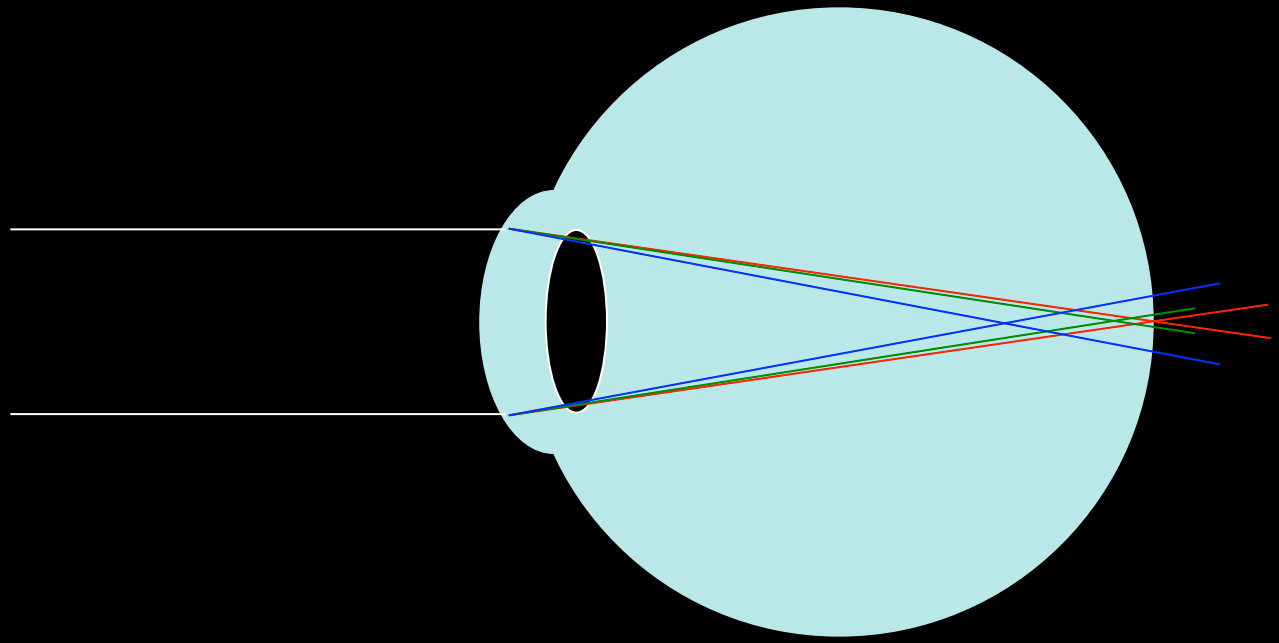


Original Scene
(shown at 200%)

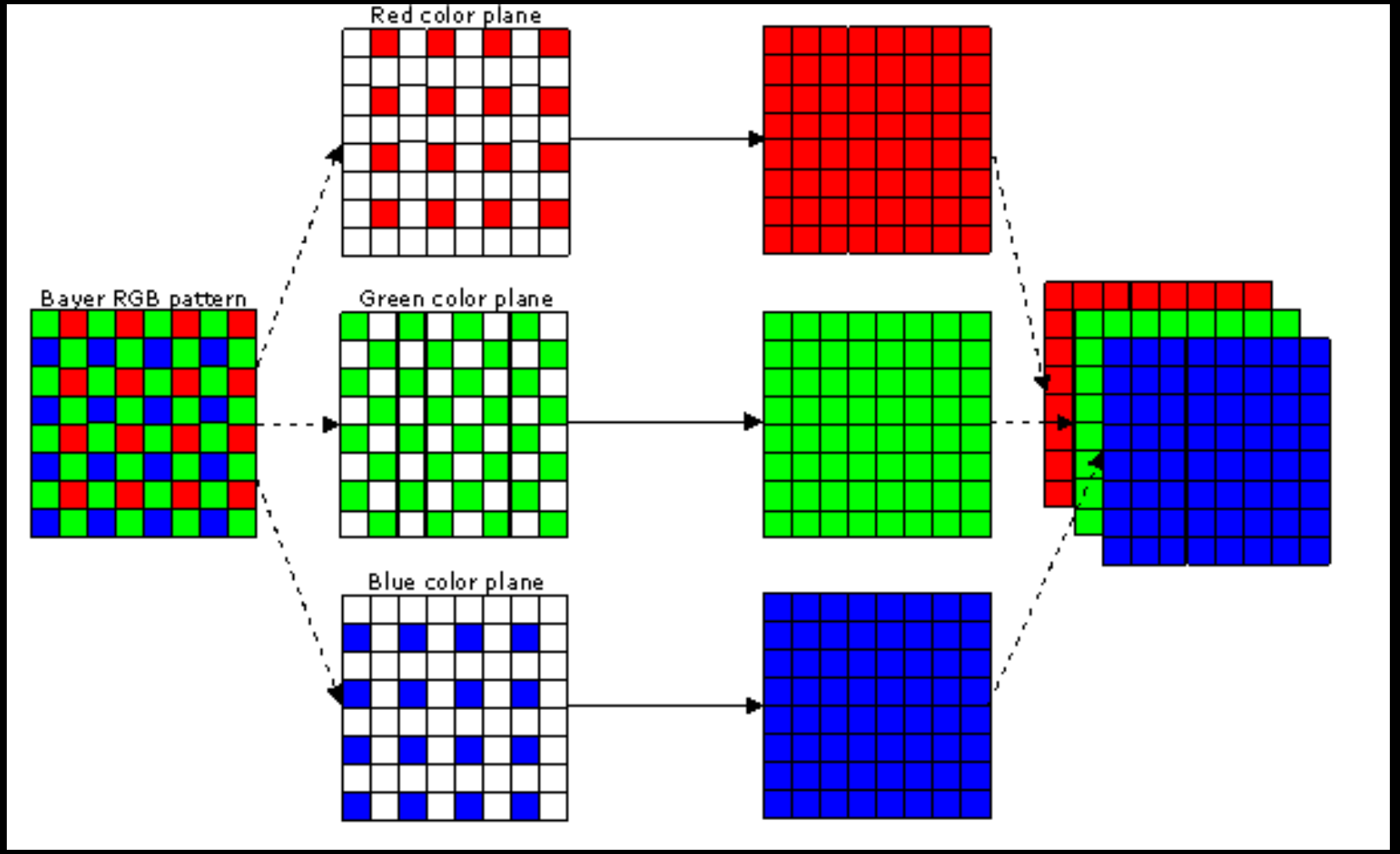
What Your Camera Sees
(through a Bayer array)

Your eyes do it too…



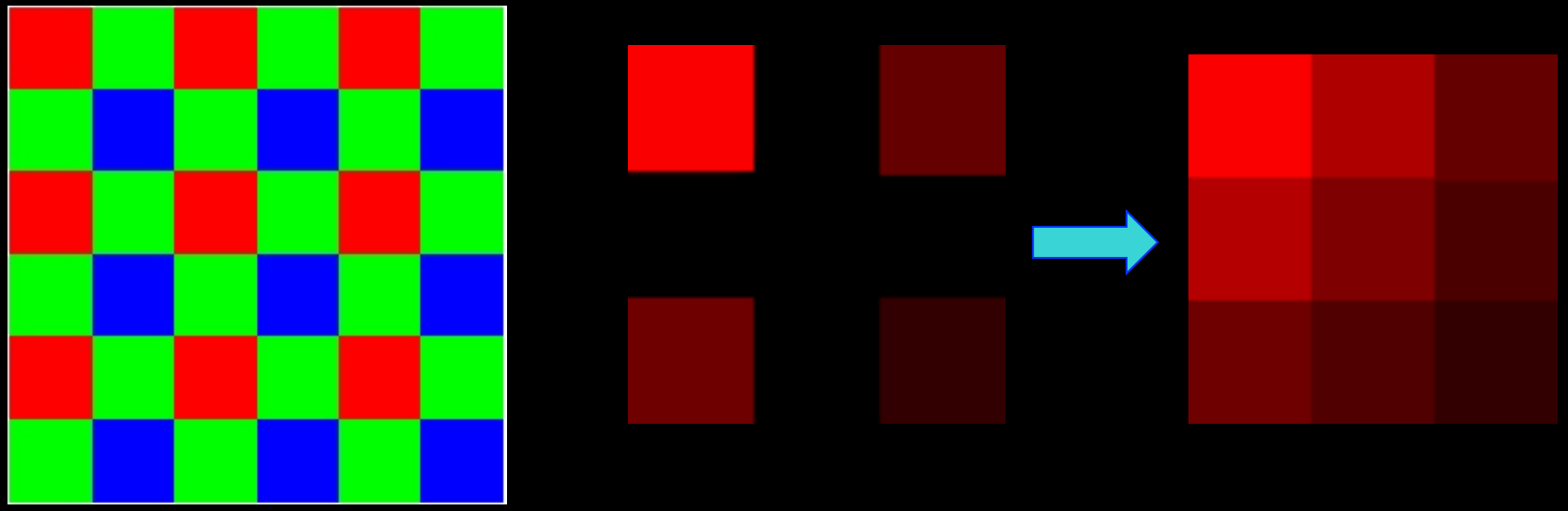


Demosaicking

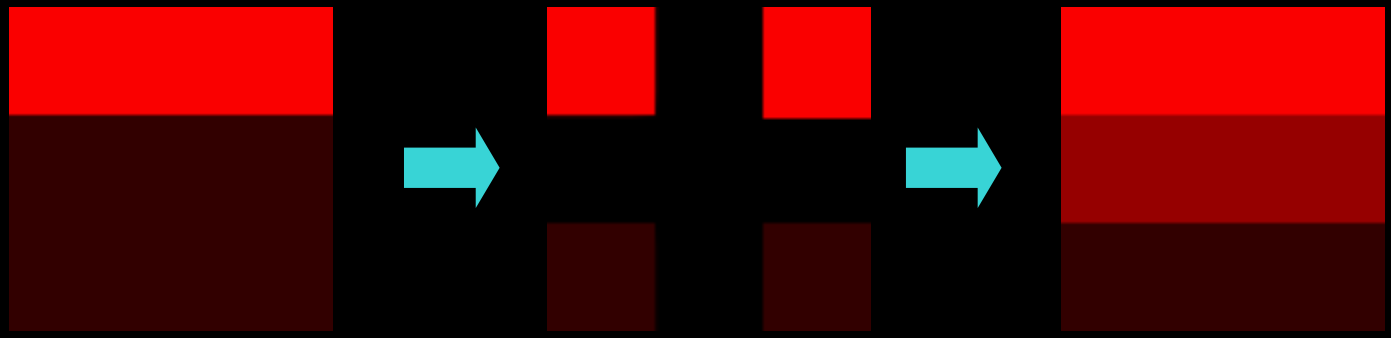


First choice: bilinear interpolation

- Easy to implement



- But fails at sharp edges



Take edges into account

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

ADAPTIVE DEMOSAICKING
Ramanath, Snyder, JEI 2003

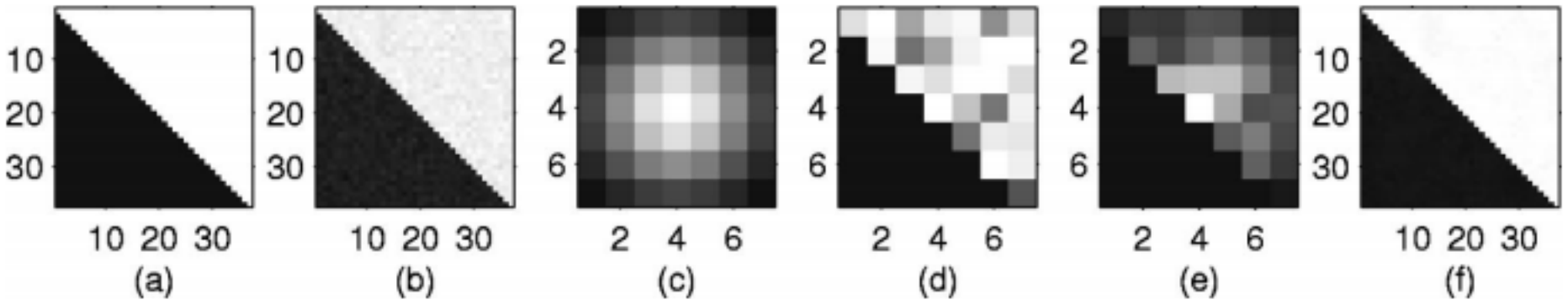


Fig. 3 Bilateral filtering: (a) original image, (b) image corrupted by Gaussian noise, (c) 7×7 blur kernel, (d) 7×7 similarity kernel at row=18, col=18, (e) 7×7 bilateral filter kernel, and (f) resulting image (denoised and sharpened).

Take edges into account

- 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

$$\hat{G}(i,j) = \hat{G}(i,j) + \alpha\Delta_{\hat{R}}(i,j)$$

HIGH-QUALITY LINEAR INTERPOLATION FOR DEMOSAICING OF BAYER-PATTERNED COLOR IMAGES

Malvar, He, Cutler, ICASSP 2004

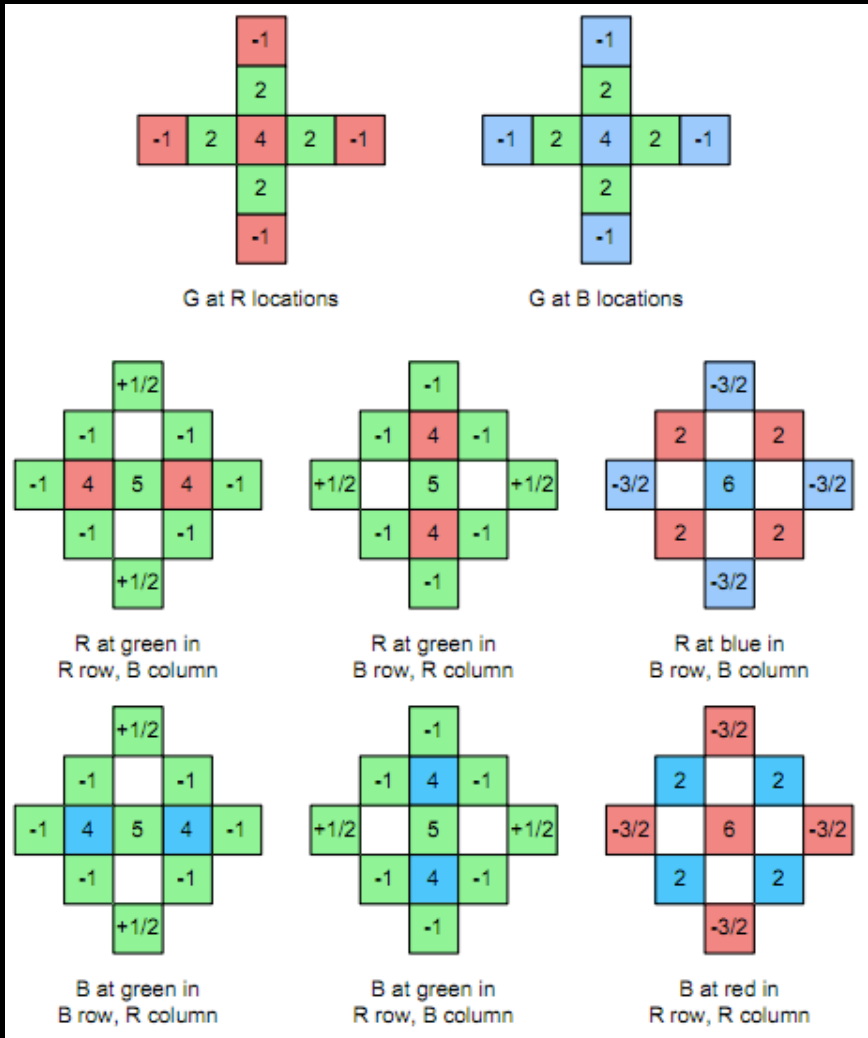


Figure 2. Filter coefficients for our proposed linear

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

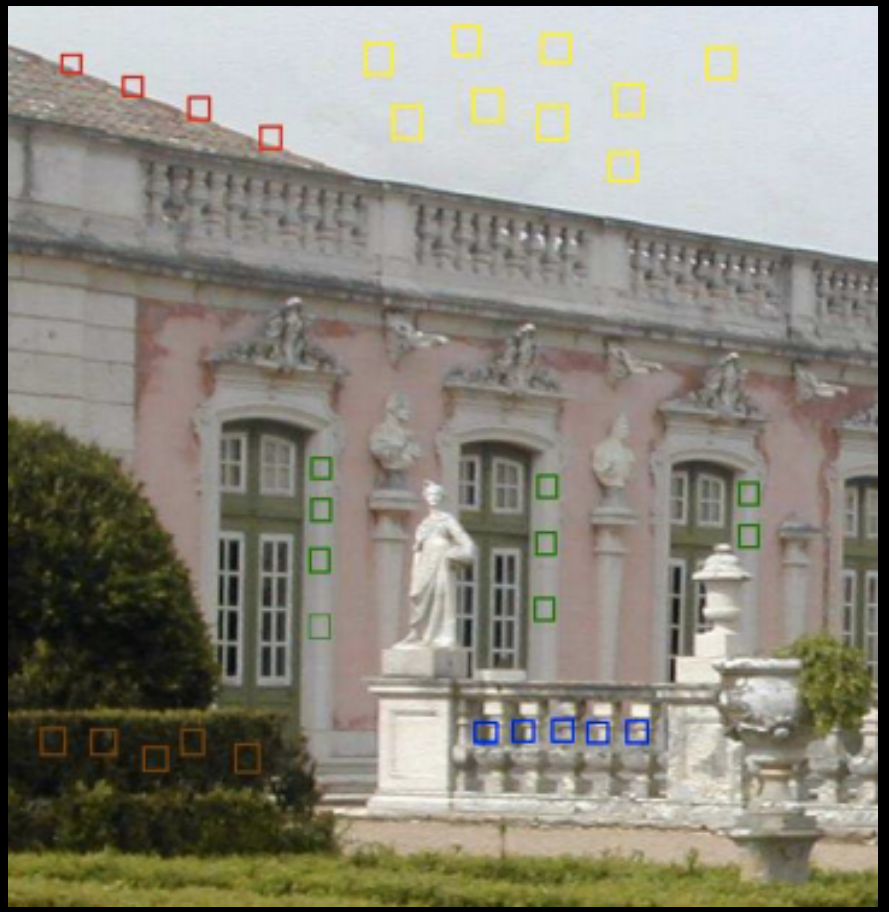


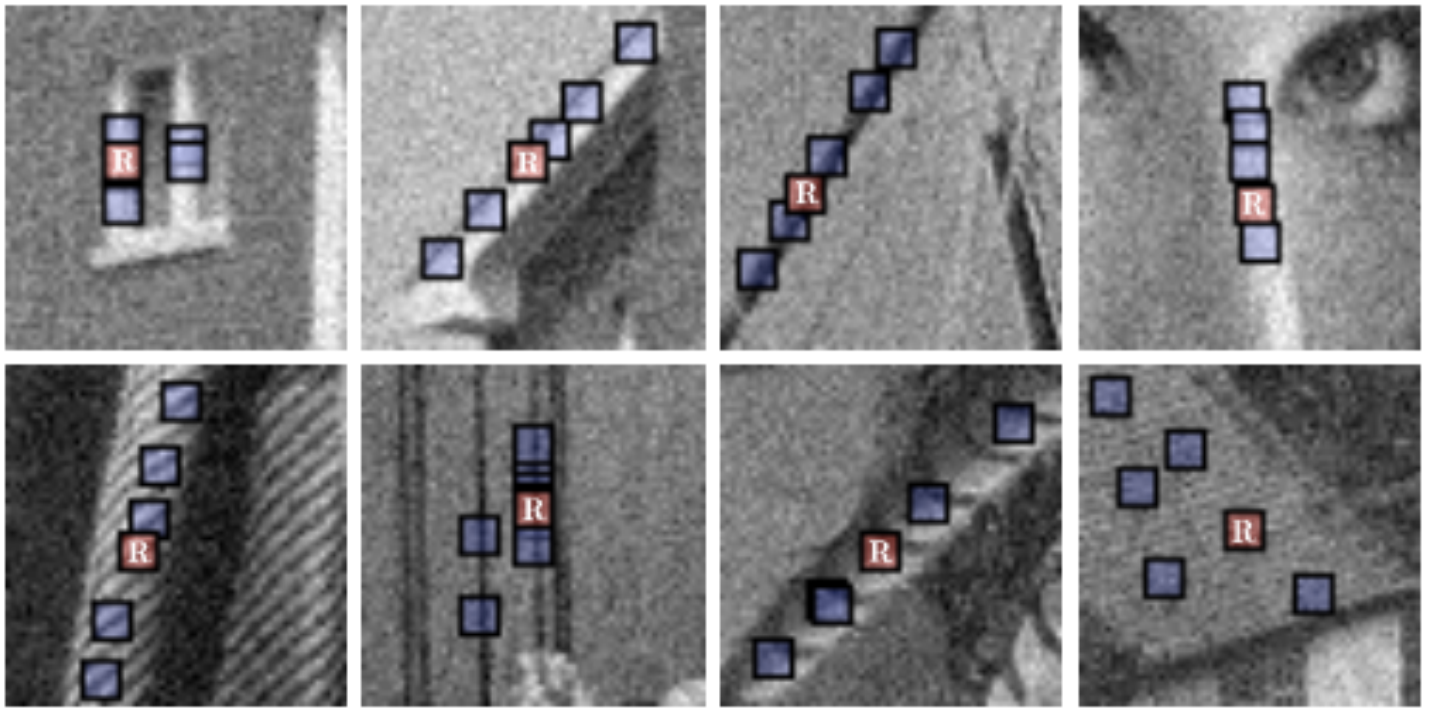
Image and movie denoising by nonlocal means
Buades, Coll, Morel, IJCV 2006

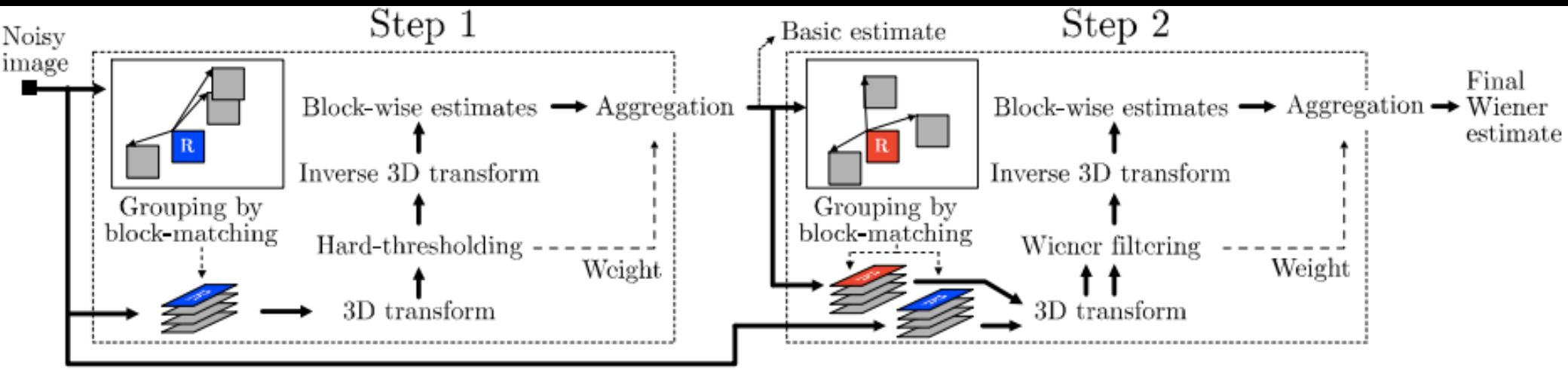
BM3D (Block Matching 3D)

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

Image denoising by sparse 3D transform-domain collaborative filtering

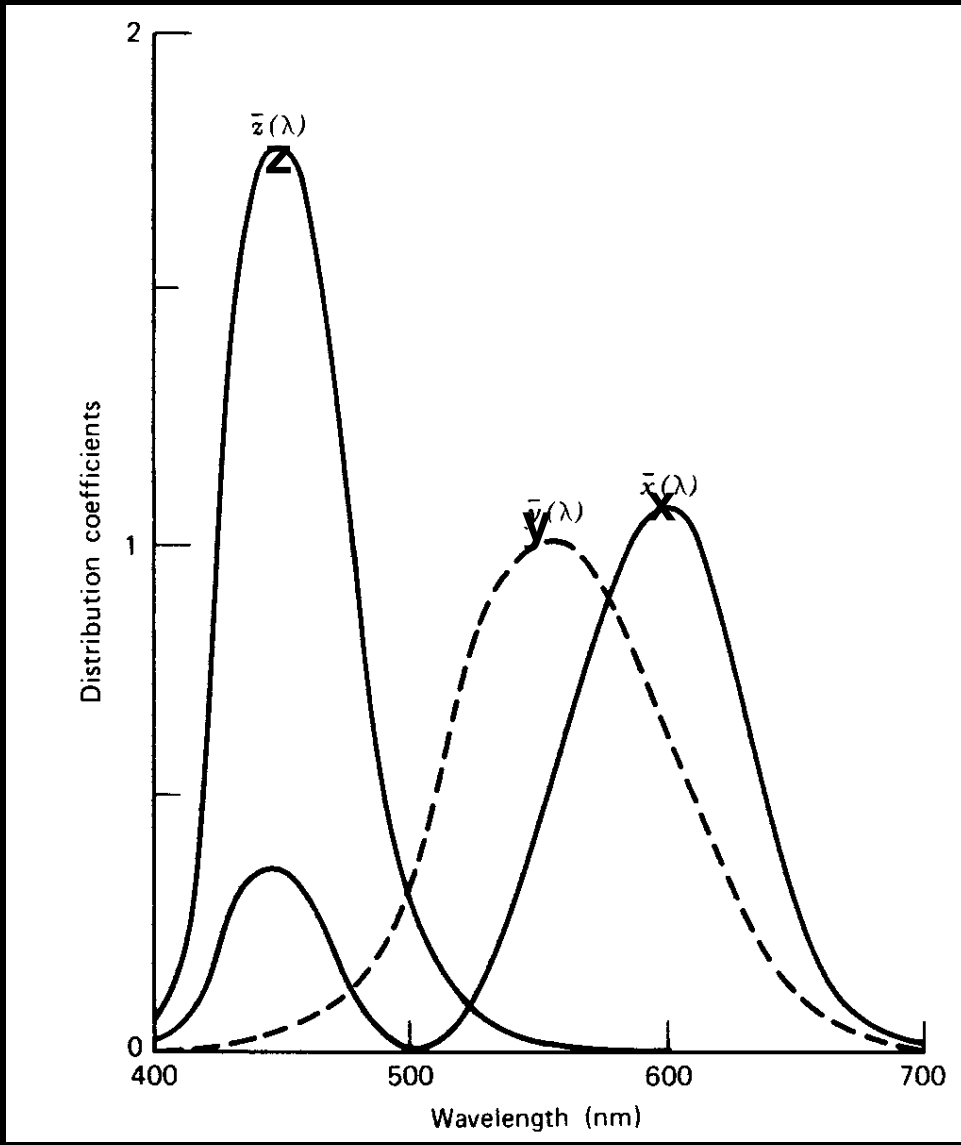
Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, *Senior Member, IEEE*





The CIE *XYZ* System

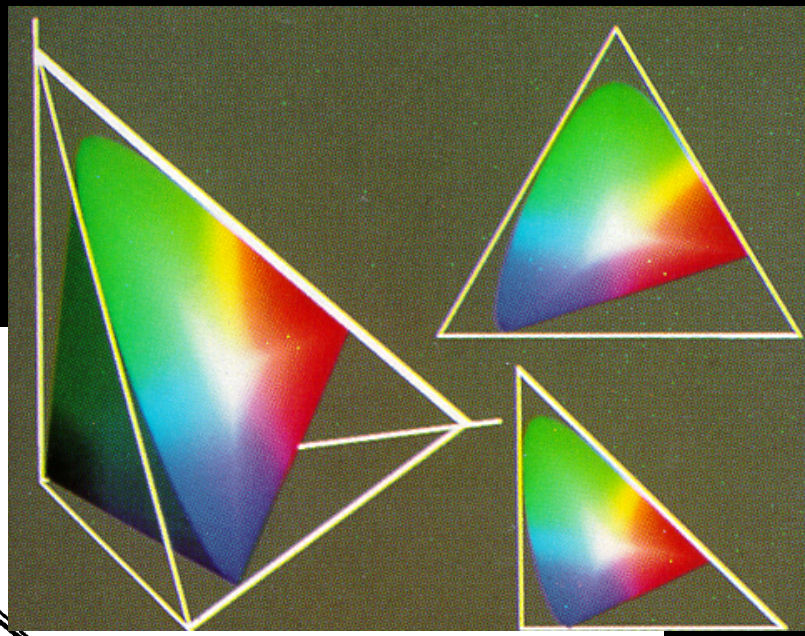
- A standard created in 1931 by CIE
  - Commission Internationale de L'Eclairage
- Defined in terms of three color matching functions
- Given an emission spectrum, we can use the CIE matching functions to obtain the *x, y* and *z* coordinates
  - y corresponds to luminance perception

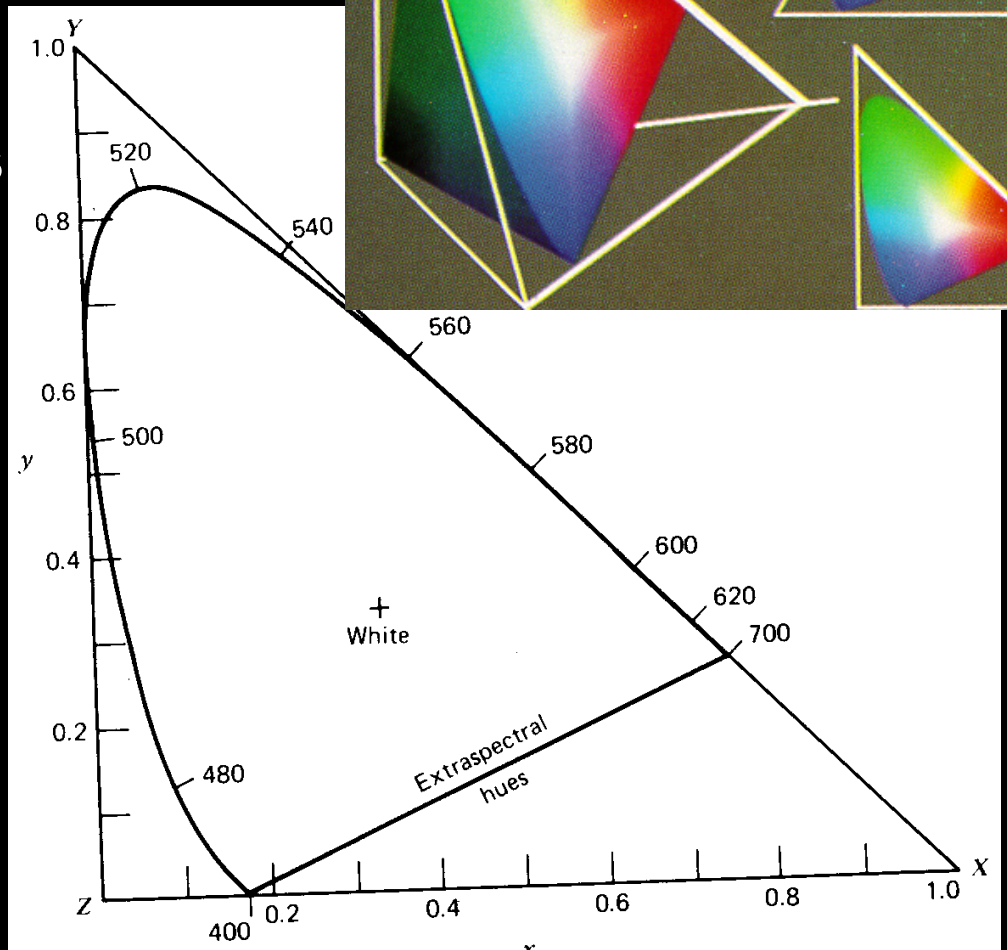


http://graphics.stanford.edu/courses/cs178/applets/threedgamut.html

The CIE Chromaticity Diagram

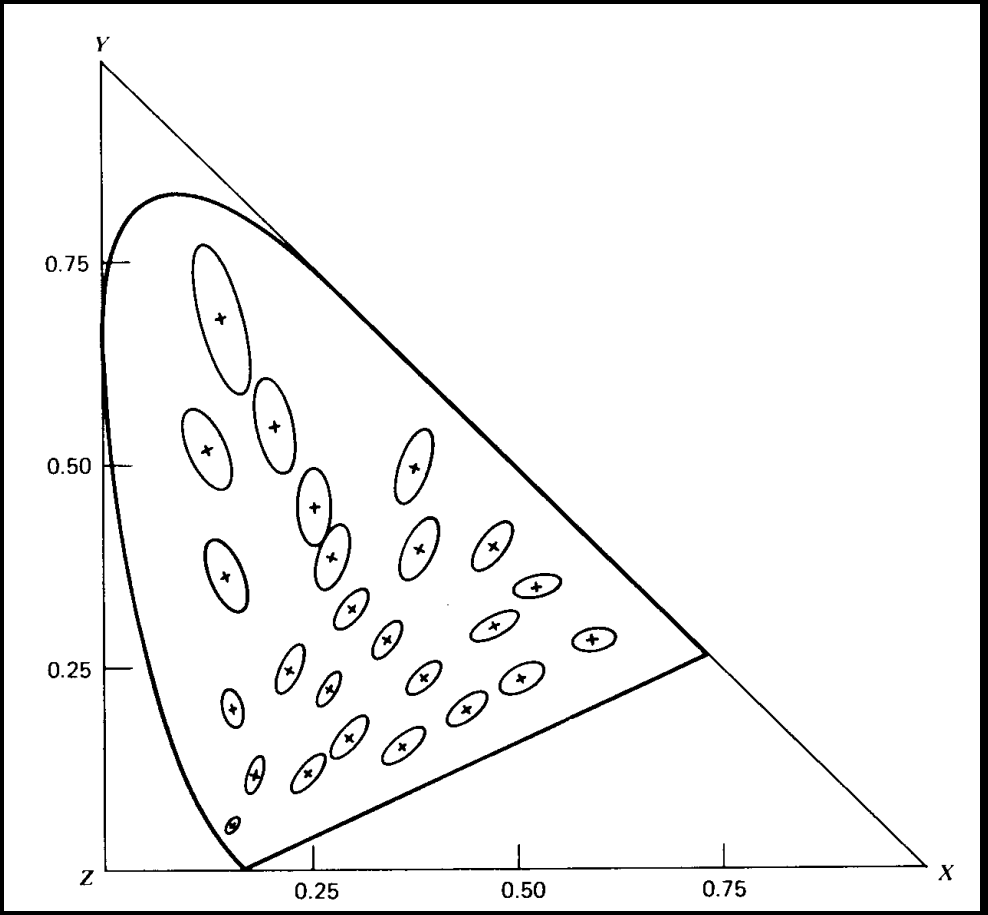
- Intensity is measured as the distance from origin
  - black = (0, 0, 0)
- *Chromaticity coordinates* give a notion of color independent of brightness
- A projection of the plane $x + y + z = 1$ yields a *chromaticity* value dependent on
  - *dominant wavelength (= hue)*, and
  - *excitation purity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





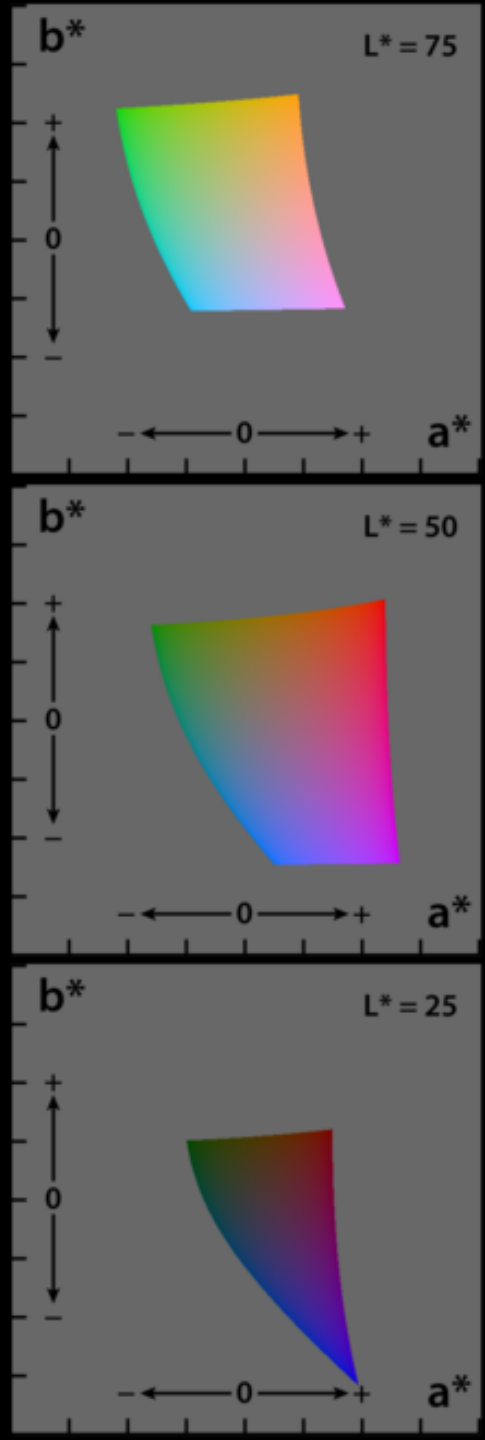
Perceptual (non-)uniformity

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



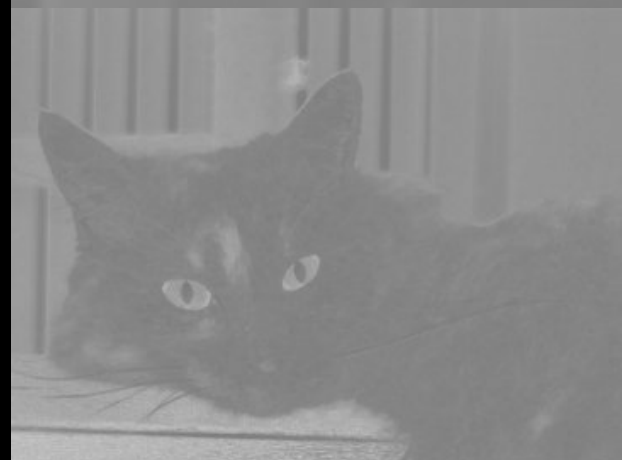
CIE L*a*b*: uniform color space

- 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



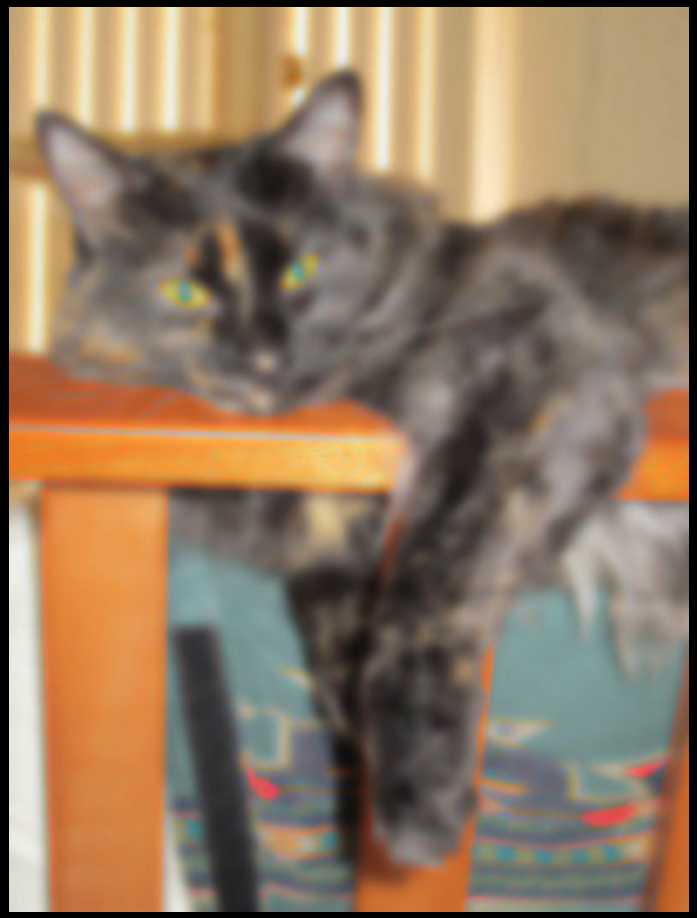
Break RGB to Lab channels

Blur “a” channel (red-green)

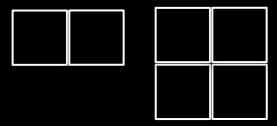


Blur “b” channel (blue-yellow)

Blur “L” channel



YUV, YCbCr, …

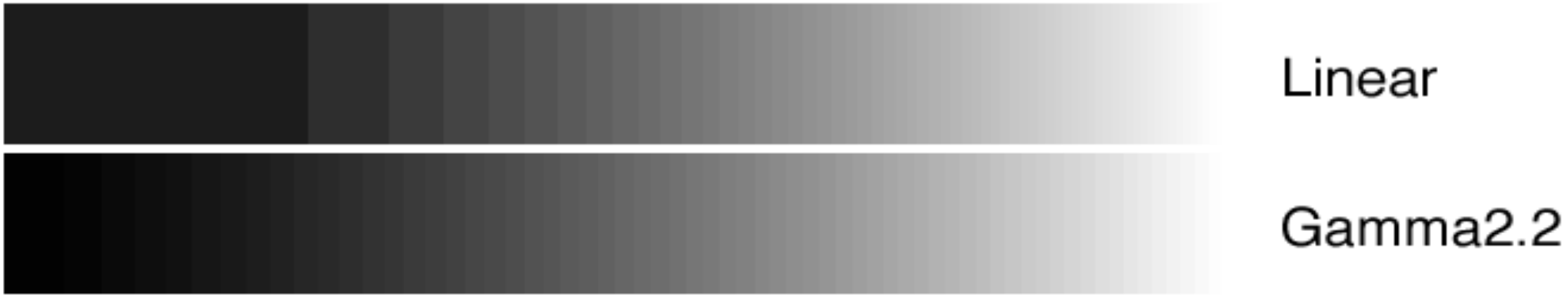
- Family of color spaces for video encoding
  - Uses the fact that eye is sensitive to luminance
  - Filters colors down (2:1, 4:1) 
- Channels
  - Y = luminance [linear]; Y’ = luma [gamma corrected]
  - CbCr / UV = chrominance [always linear]
- Y′CbCr is not an absolute color space
  - it is a way of encoding RGB information
  - the actual color depends on the RGB primaries used

How many bits are needed for smooth shading?

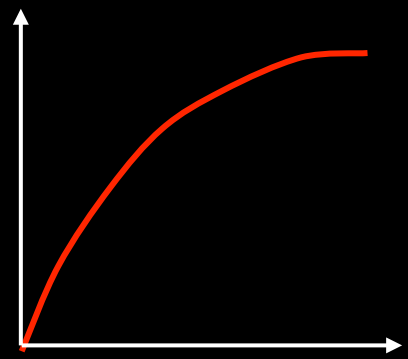
- With a given adaptation, human vision has contrast sensitivity ~1%
  - call black 1, white 100
  - you can see differences
    - 1, 1.01, 1.02, … needed step size ~ 0.01
    - 98, 99, 100 needed step size ~ 1
  - with linear encoding
    - delta 0.01
      - 100 steps between 99 & 100 → wasteful
    - 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 X^{γ} where $\gamma = 1/2.2$
  - then encode



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

http://graphics.stanford.edu/courses/cs178/applets/gamma.html

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

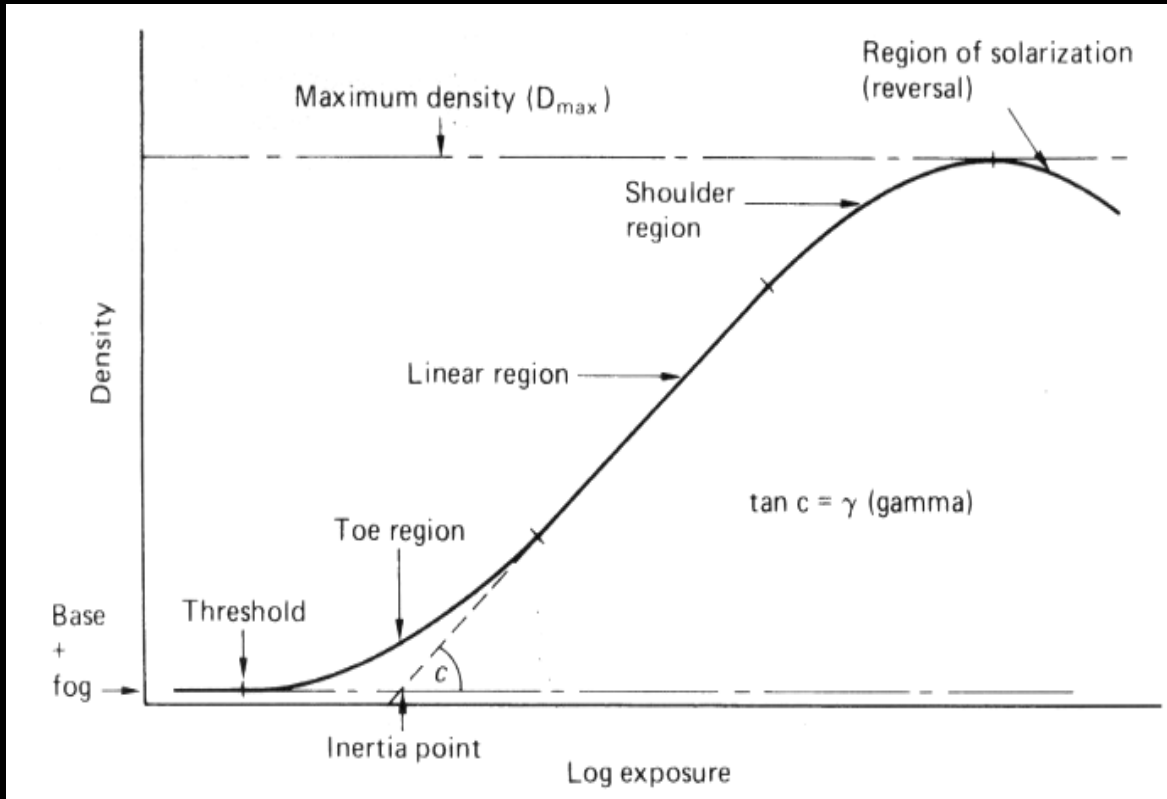
- First need to map from sensor RGB to standard
  - need calibration

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

Film response curve

- Toe region
  - the chemical process is just starting
- Middle
  - follows a power function
  - if a given amount of light turned half of the grain crystals to silver, the same amount more turns half of the rest
- Shoulder region
  - close to saturation

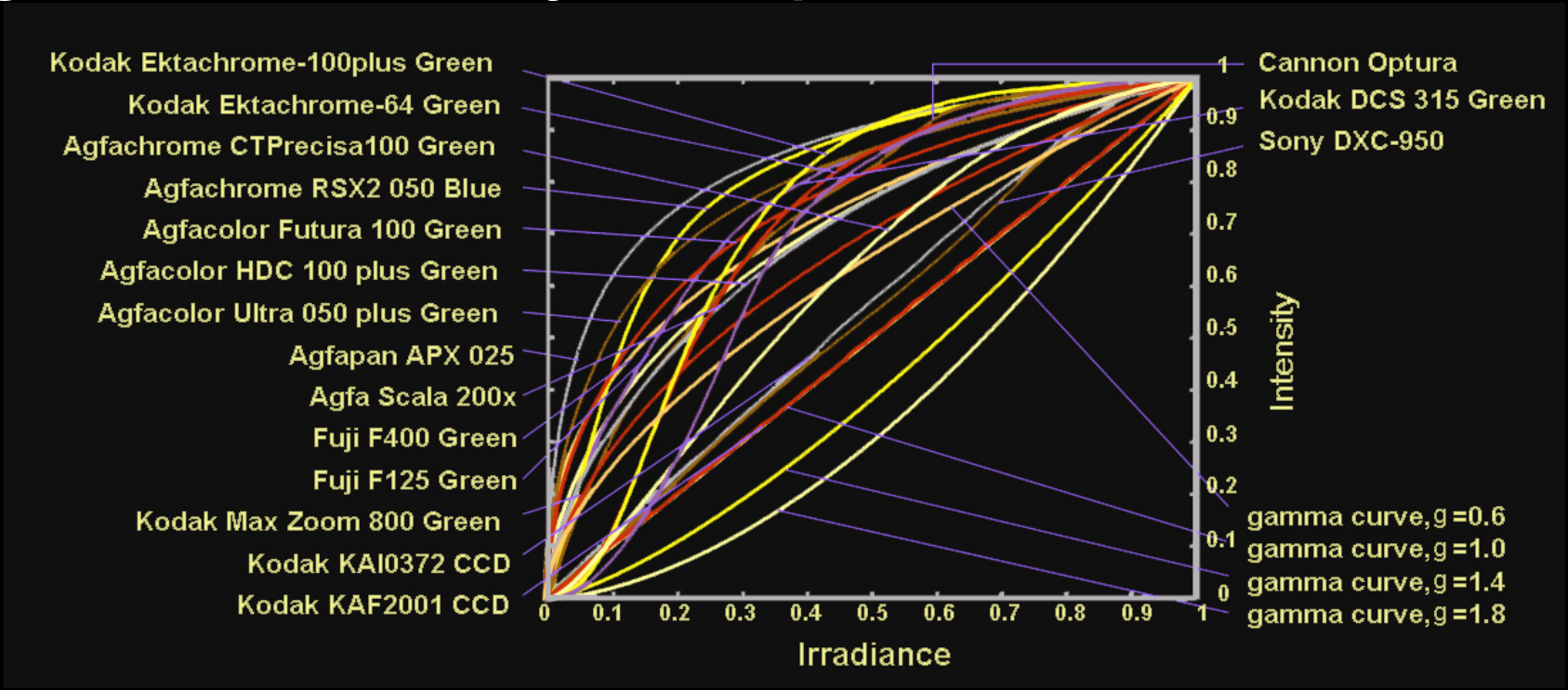


The 'geography' of the characteristic curve of a negative material

- Toe and shoulder preserve more dynamic range at dark and bright areas
  - at the cost of reduced contrast
- Film has more dynamic range than print
  - ~12bits

Digital camera response curve

- Digital cameras modify the response curve



- May use different response curves at different exposures
  - impossible to calibrate and invert!

3A

- Automated selection of key camera control values
  - auto-focus
  - auto-exposure
  - auto-white-balance

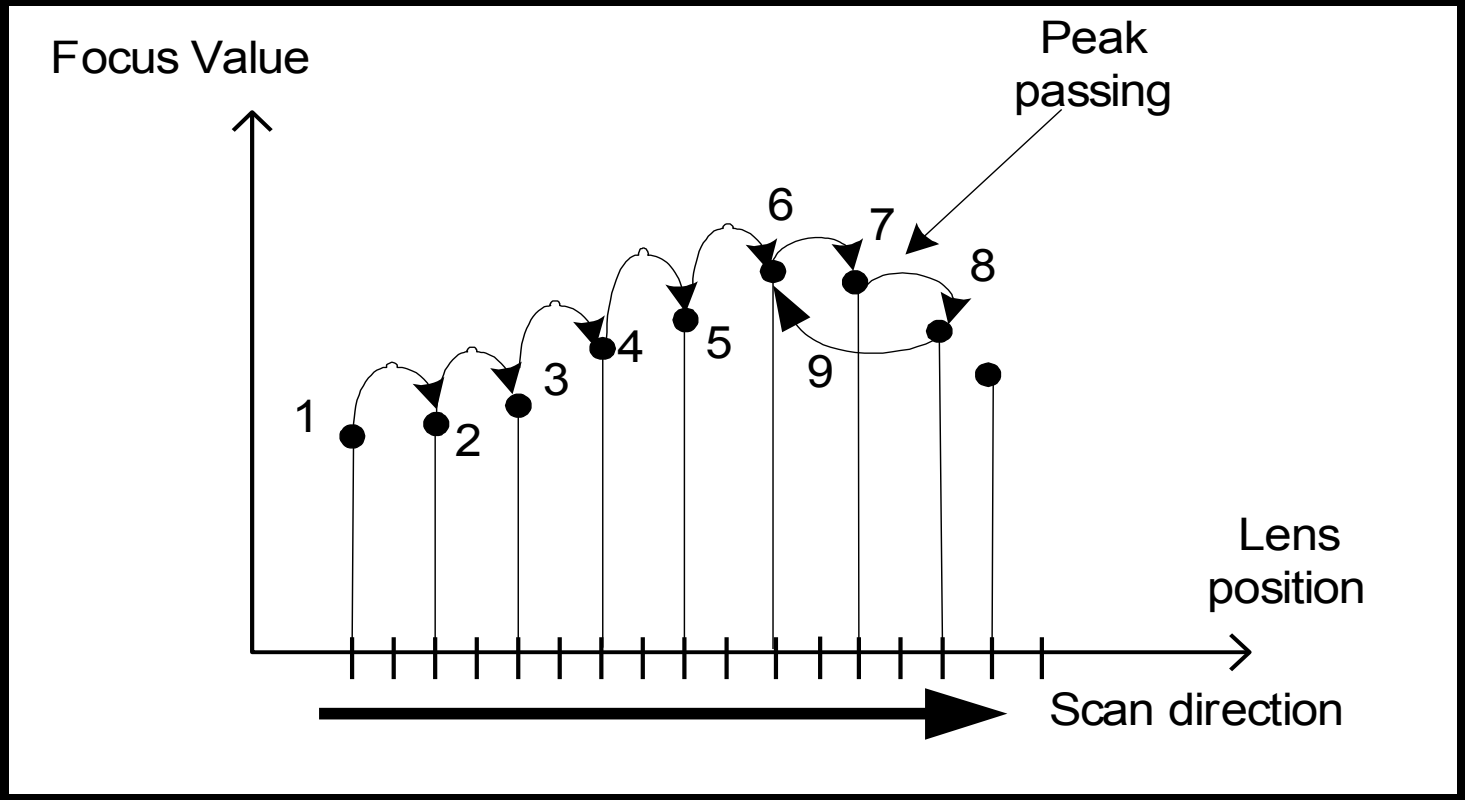
Contrast-based auto-focus

- ISP can filter pixels with configurable IIR filters
  - to produce a low-resolution sharpness map of the image
- The sharpness map helps estimate the best lens position
  - by summing the sharpness values (= Focus Value)
    - either over the entire image
    - or over a rectangular area

http://graphics.stanford.edu/courses/cs178/applets/autofocusCD.html

Hunt for the peak

- First coarse search
  - back up after peak, do a finer search



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)

Best way to do white balance

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

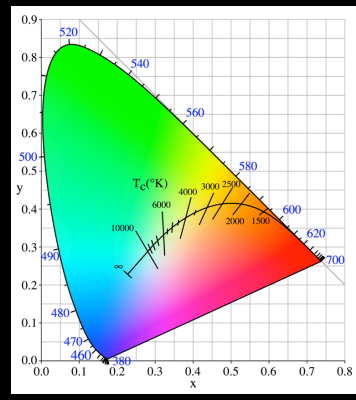
- If the object is recoded as r_w, g_w, b_w
  - use weights k/r_w, k/g_w, k/b_w
    - where k controls the exposure

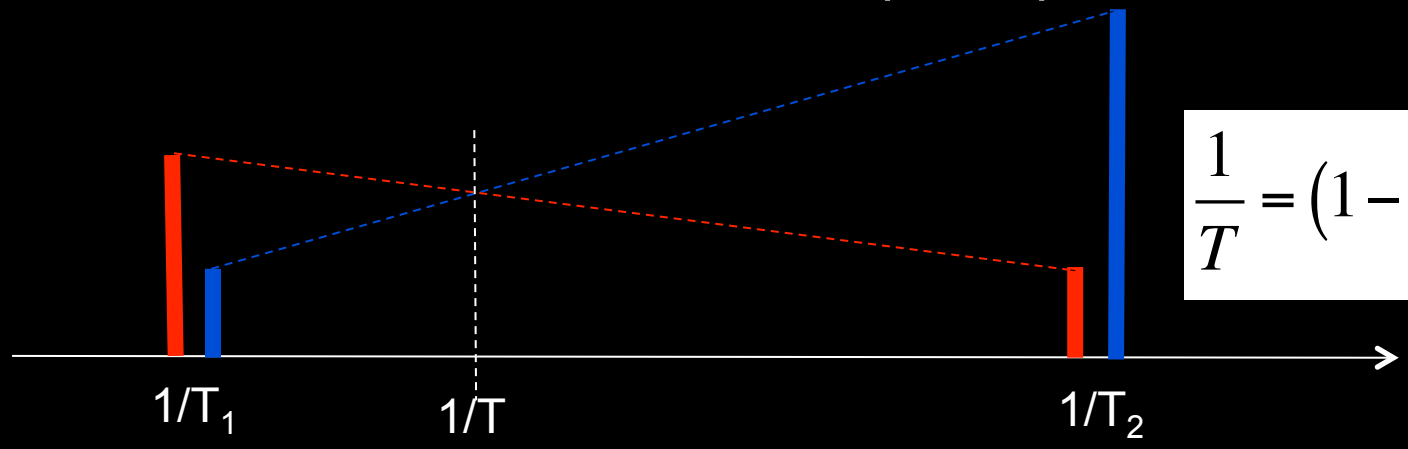
Brightest pixel assumption

- 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

Estimating the color temperature

- Pre-calibrate to “warm” and “cold” lighting
- 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 T_1 and T_2
  - calculate the average values R, B
  - solve α, use to interpolate matrices (or 1/T)





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

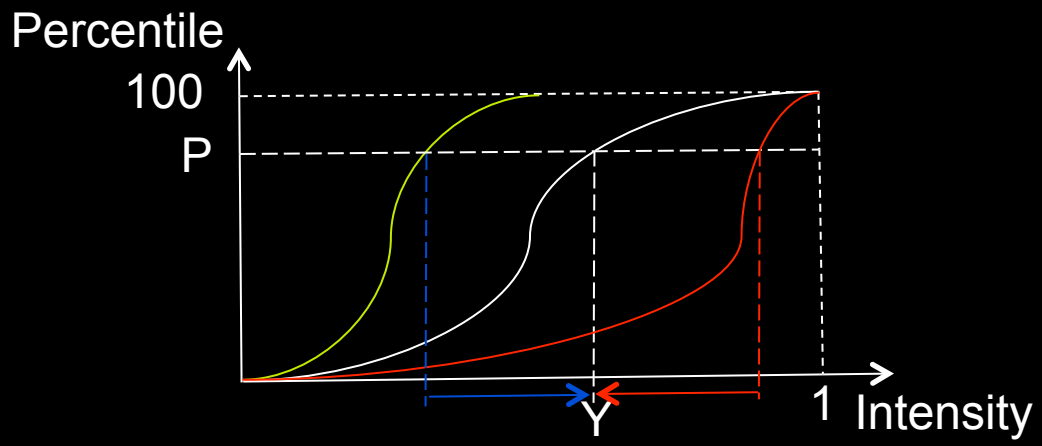
$$R = (1-\alpha)R_1 + \alpha R_2,\ B = (1-\alpha)B_1 + \alpha B_2$$

Auto-exposure

- Goal: well-exposed image (not a very well defined goal!)
- Possible parameters to adjust
  - Exposure time
    - Longer exposure time leads to brighter image, but also motion blur
  - Aperture (f-number)
    - Larger aperture (smaller f-number) lets more light in causing the image to be brighter, also makes depth of field shallower
    - Phone cameras often have fixed aperture
  - Analog and digital gain
    - Higher gain makes image brighter but amplifies noise as well
  - ND filters on some cameras

Exposure metering

- **Cumulative Density Function of image intensity values**
  - ***P* percent of image pixels have an intensity lower than *Y***

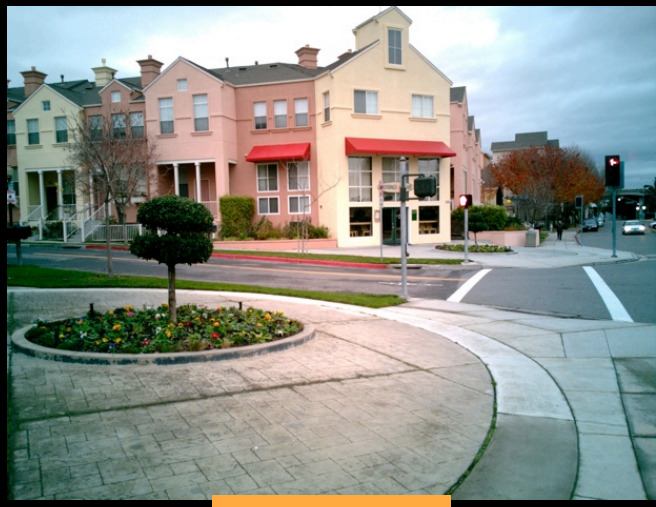


Exposure metering examples

- Adjustment examples
  - P = 0.995, Y = 0.9
    - max 0.5% of pixels are saturated (highlights)
  - P = 0.1, Y = 0.1
    - max 10% of pixels are under-exposed (shadows)
  - Auto-exposure somewhere in between, e.g., P = 0.9, Y = 0.4

Highlights

Auto-exposure



Shadows

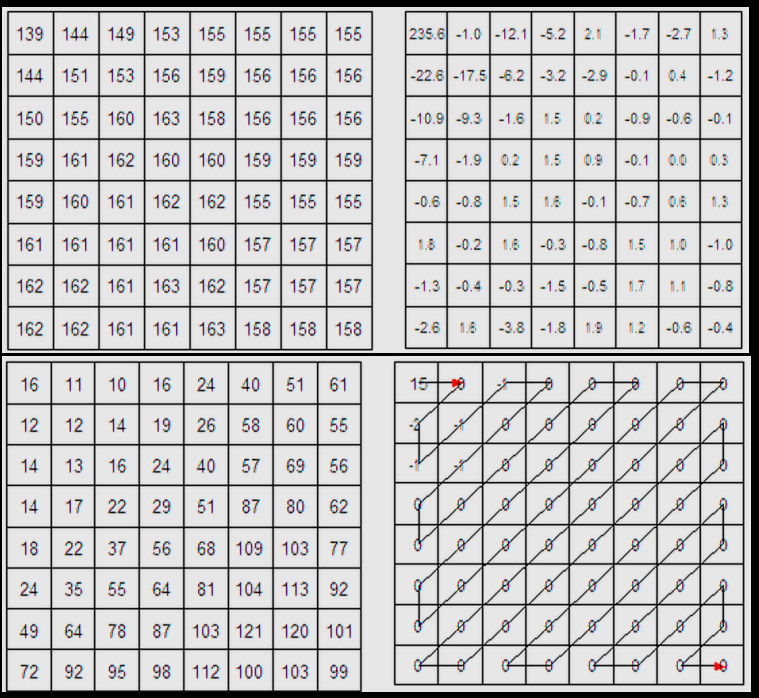
JPEG Encoding

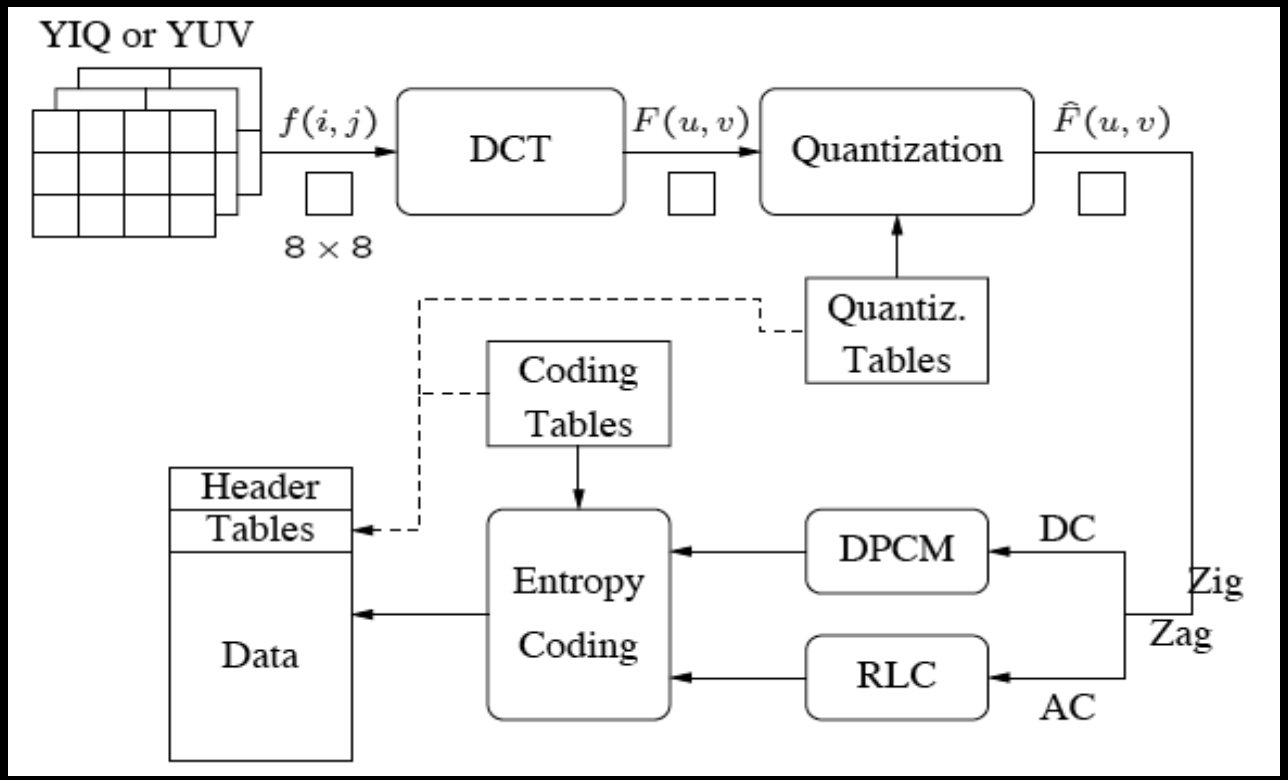
1. Transform RGB to YUV or YIQ and subsample color
2. DCT on 8x8 image blocks
3. Quantization
4. Zig-zag ordering and run-length encoding
5. Entropy coding

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

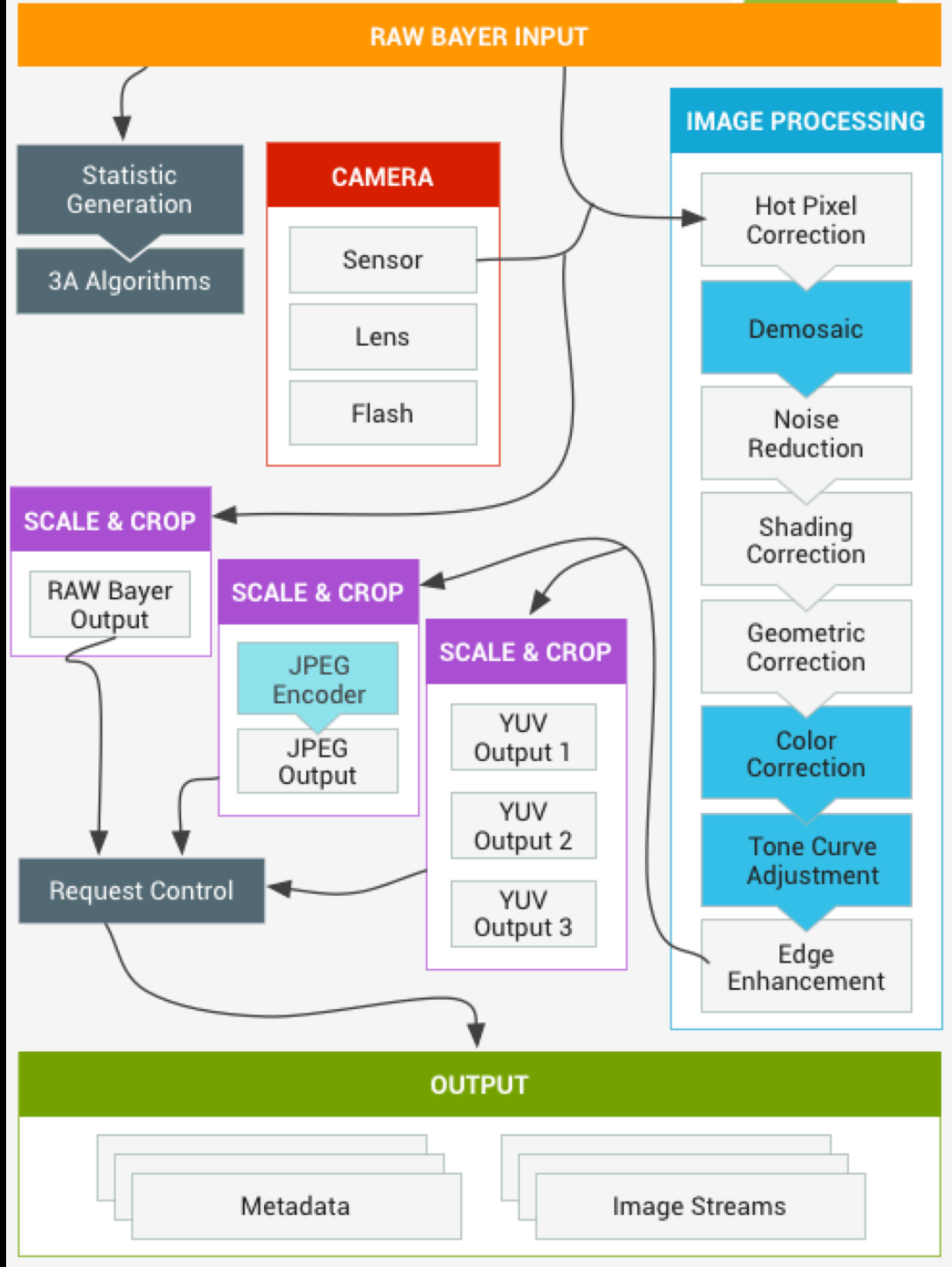
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99



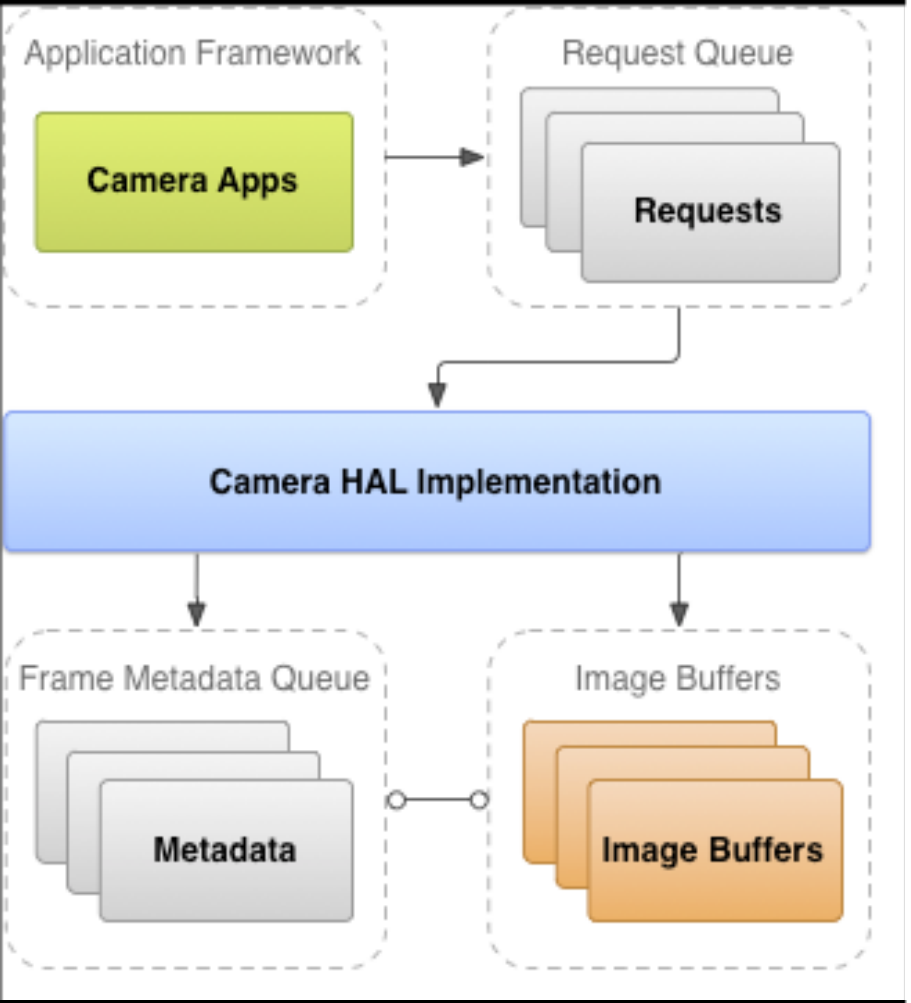


Android Camera Architecture

http://source.android.com/devices/camera/index.html

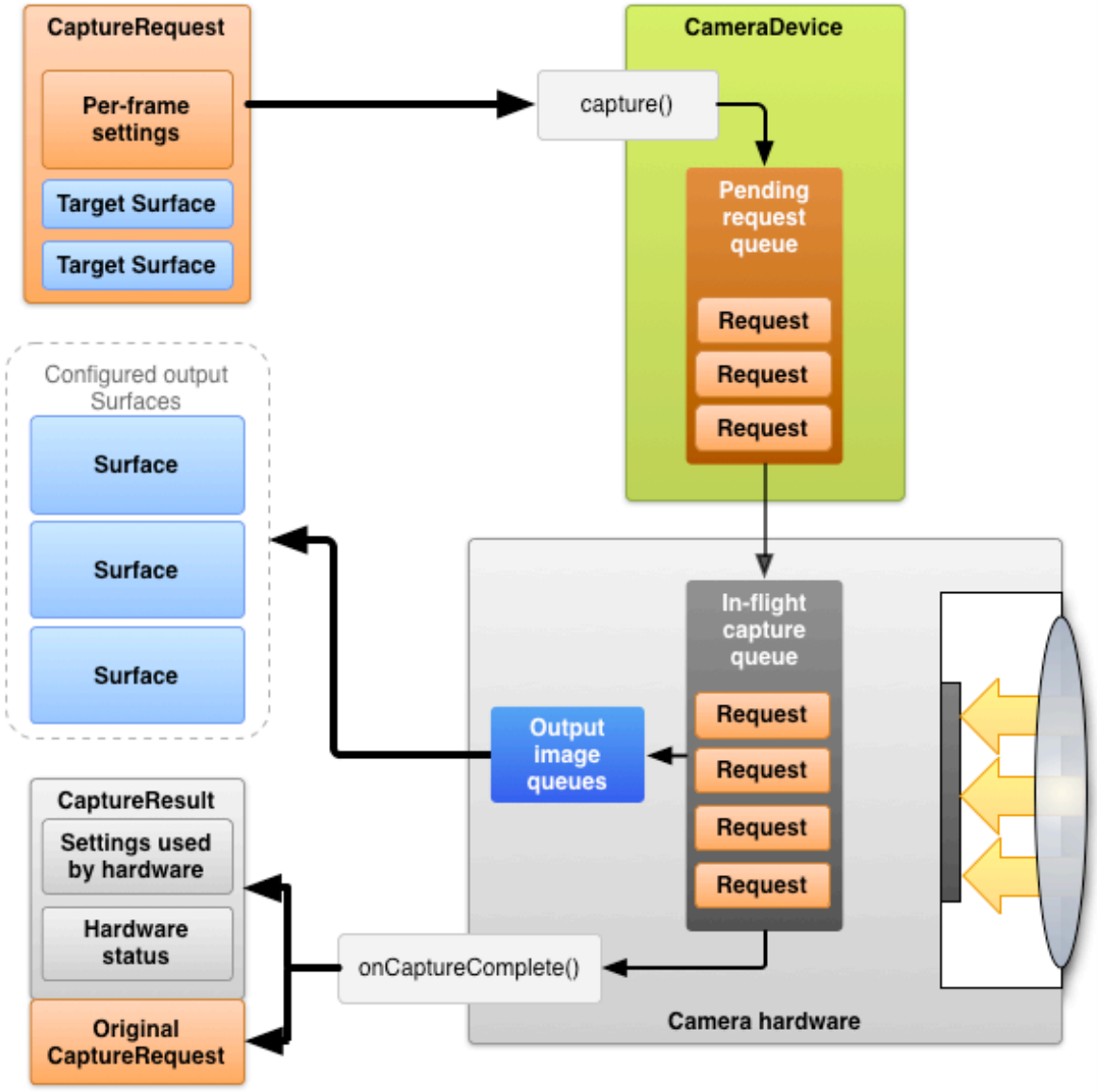


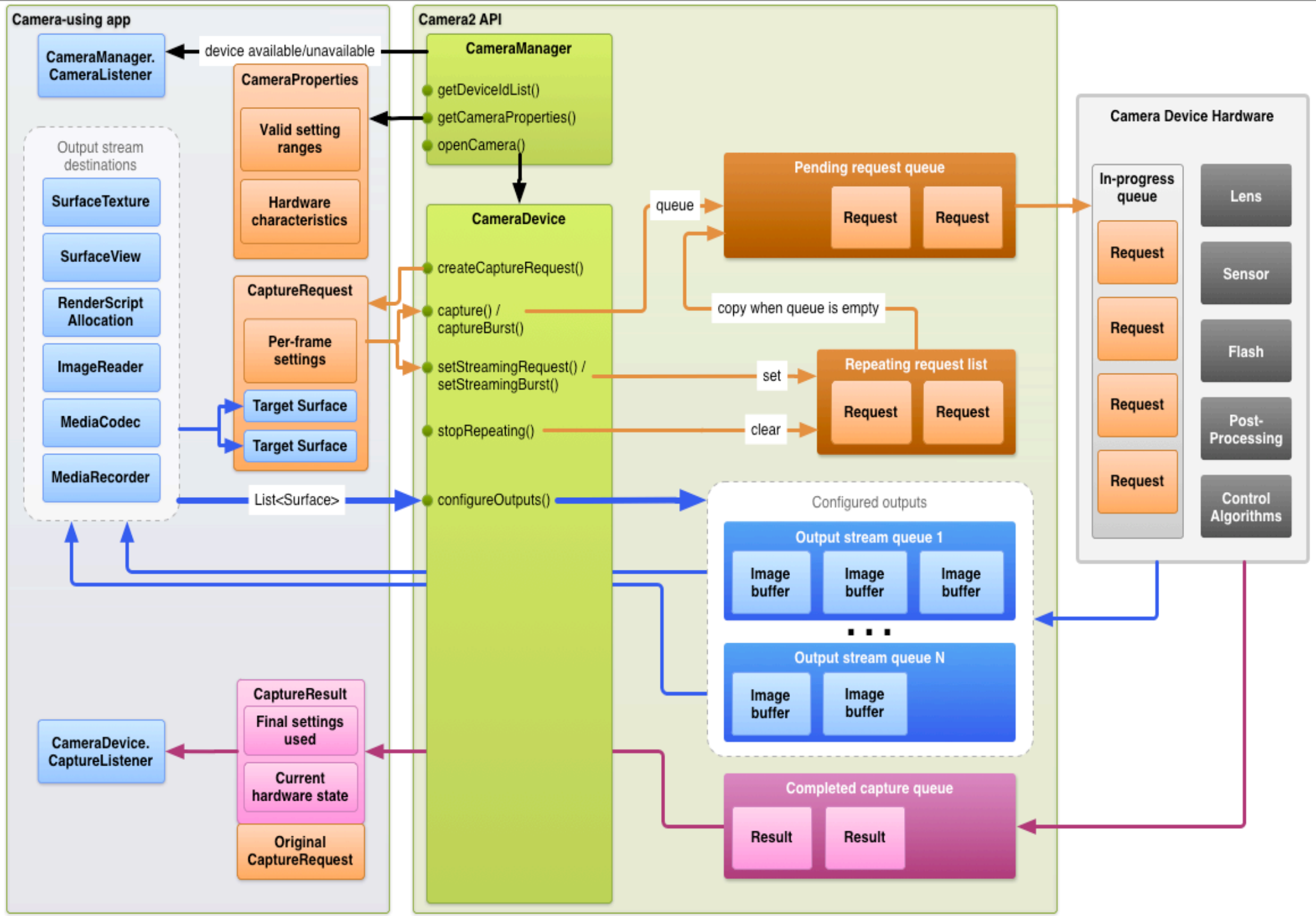
HAL v3

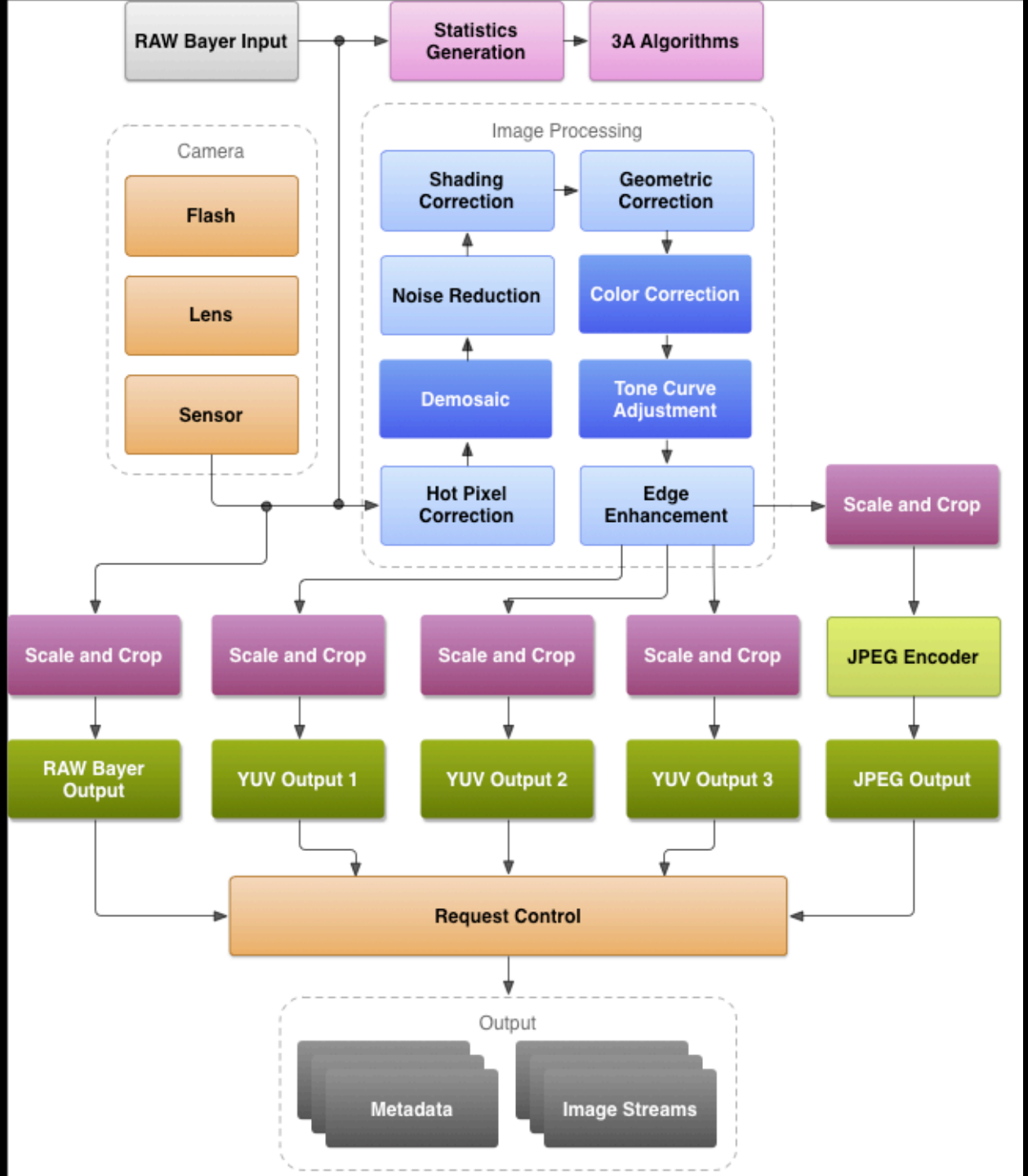


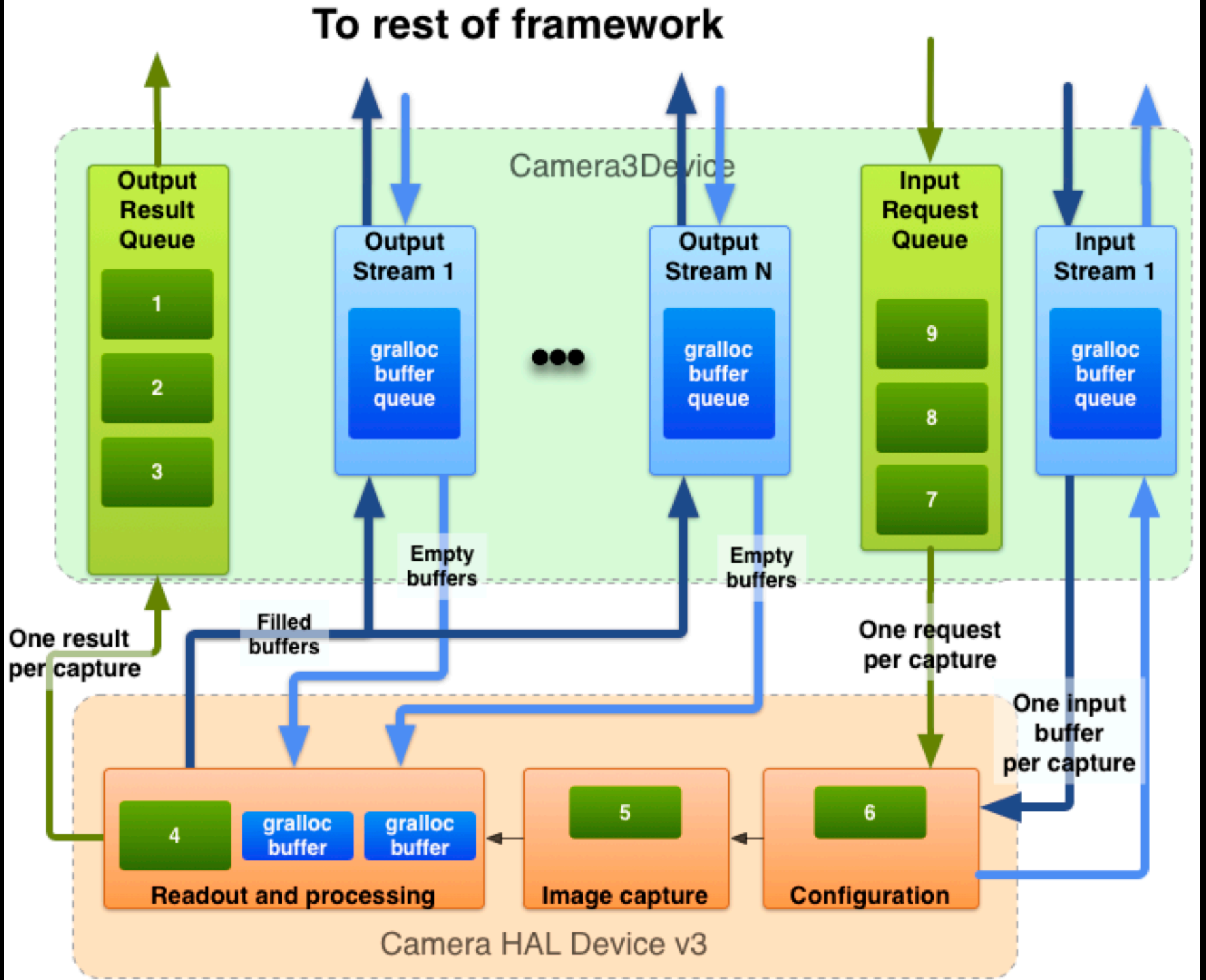
Camera2 API Core Operation Model

1 Request ⟹ 1 image captured ⟹
1 Result metadata + N image buffers




Camera-using app
CameraManager.
CameraListener
device available/unavailable
Camera2 API
CameraManager
getDeviceIdList()
getCameraProperties()
openCamera()
CameraProperties
Valid setting ranges
Hardware characteristics
Output stream destinations
SurfaceTexture
SurfaceView
RenderScript Allocation
ImageReader
MediaCodec
MediaRecorder
CaptureRequest
Per-frame settings
Target Surface
Target Surface
List<Surface>
CameraDevice
createCaptureRequest()
capture() / captureBurst()
setStreamingRequest() / setStreamingBurst()
stopRepeating()
configureOutputs()
queue
Pending request queue
Request
Request
copy when queue is empty
set
clear
Repeating request list
Request
Request
Camera Device Hardware
In-progress queue
Request
Request
Request
Request
Lens
Sensor
Flash
Post-Processing
Control Algorithms
Configured outputs
Output stream queue 1
Image buffer
Image buffer
Image buffer
Output stream queue N
Image buffer
Image buffer
CaptureResult
Final settings used
Current hardware state
Original CaptureRequest
CameraDevice.
CaptureListener
Completed capture queue
Result
Result


RAW Bayer Input
Statistics Generation
3A Algorithms
Camera
Flash
Lens
Sensor
Image Processing
Shading Correction
Geometric Correction
Noise Reduction
Color Correction
Demosaic
Tone Curve Adjustment
Hot Pixel Correction
Edge Enhancement
Scale and Crop
Scale and Crop
Scale and Crop
Scale and Crop
Scale and Crop
JPEG Encoder
RAW Bayer Output
YUV Output 1
YUV Output 2
YUV Output 3
JPEG Output
Request Control
Output
Metadata
Image Streams


To rest of framework
Camera3Device
Output Result Queue
1
2
3
Output Stream 1
gralloc buffer queue
Output Stream N
gralloc buffer queue
Input Request Queue
9
8
7
Input Stream 1
gralloc buffer queue
Empty buffers
Empty buffers
Filled buffers
One result per capture
One request per capture
One input buffer per capture
4
gralloc buffer
gralloc buffer
Readout and processing
5
Image capture
6
Configuration
Camera HAL Device v3

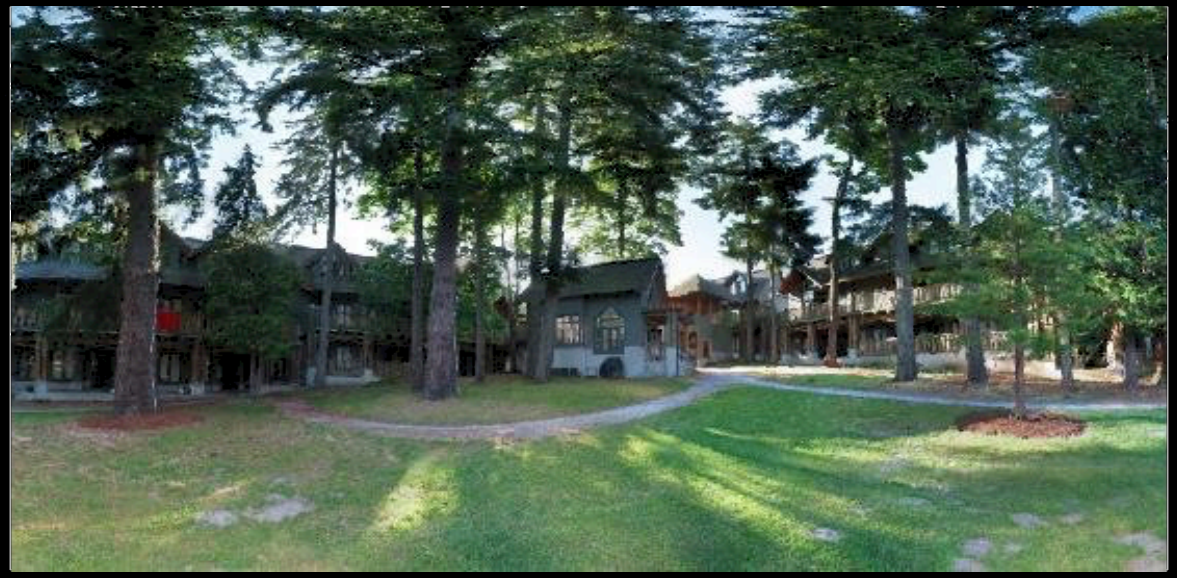
Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°

Slide from Brown & Lowe

Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°



Slide from Brown & Lowe

Are you getting the whole picture?

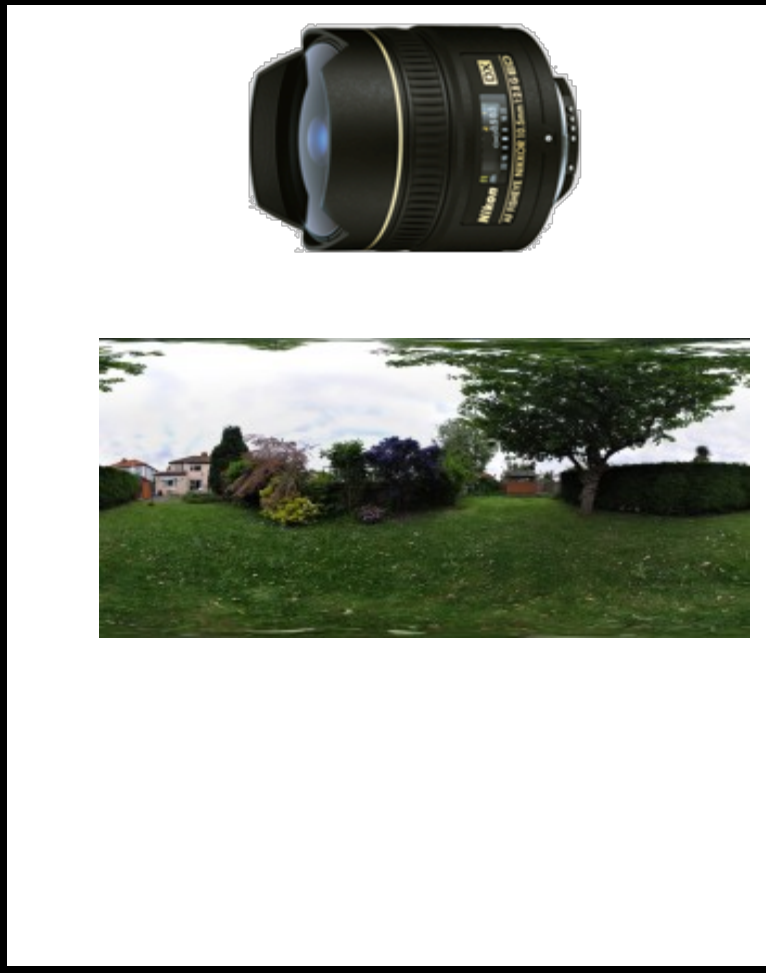
- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°
- Panoramic Mosaic = 360 x 180°

Slide from Brown & Lowe

Panoramas with wide-angle optics

http://www.0-360.com

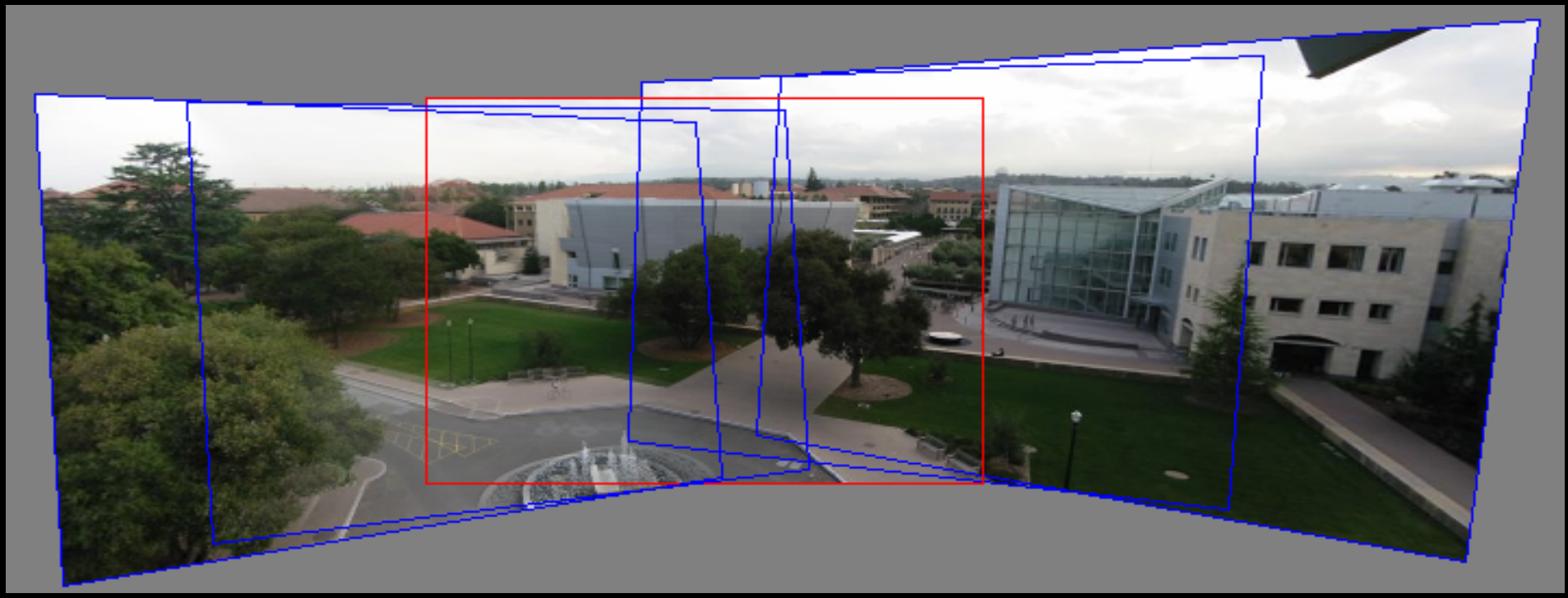
AF DX Fisheye-NIKKOR 10.5mm f/2.8G ED



Building a Panorama

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Perspective stitching



- Pick one image, typically the central view (red outline)
- Warp the others to its plane
- Blend

Stitch images on the same plane



common picture plane of mosaic image

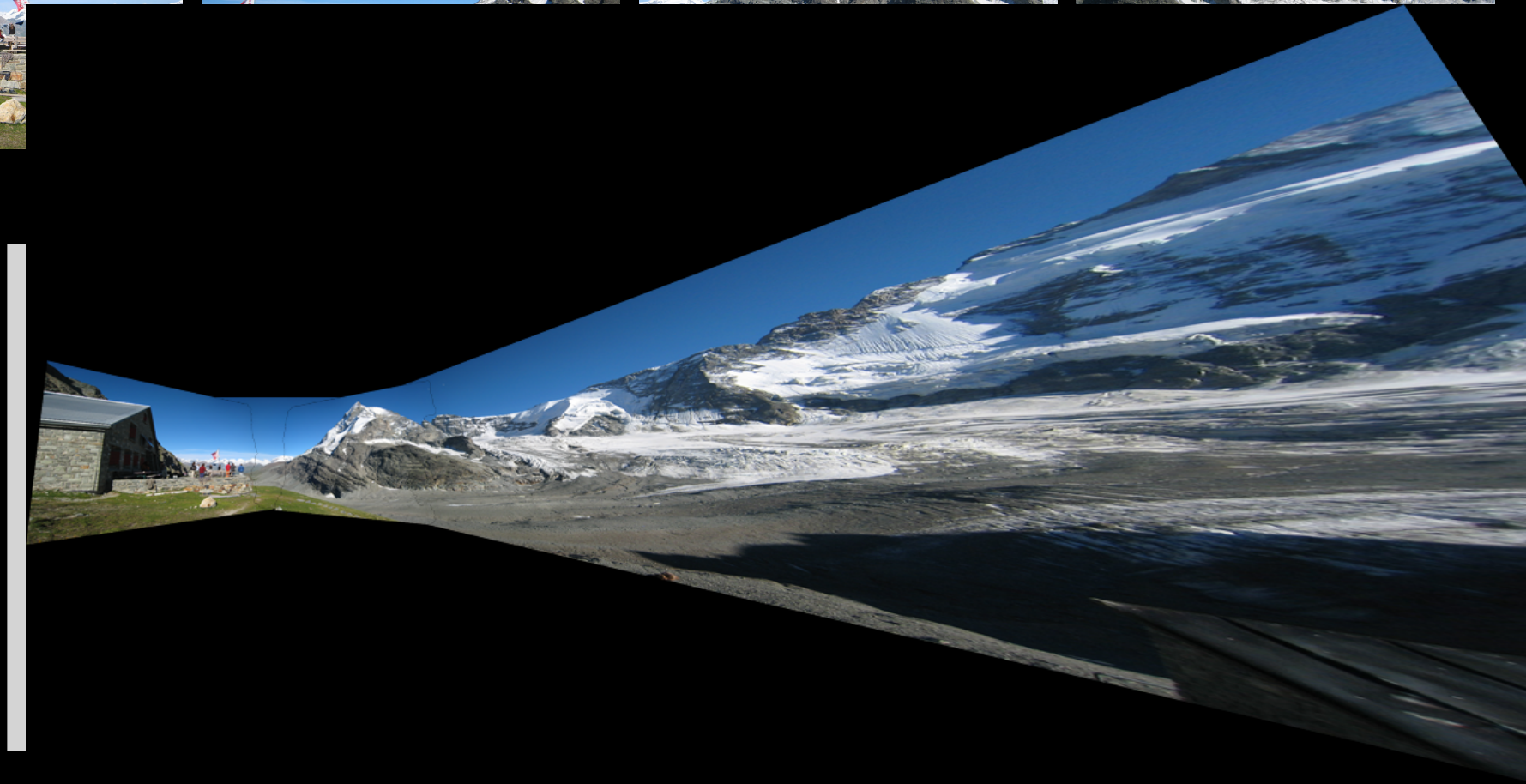
perspective reprojection

Pics: Marc Levoy

Using 4 shots instead of 3







Cylindrical works better

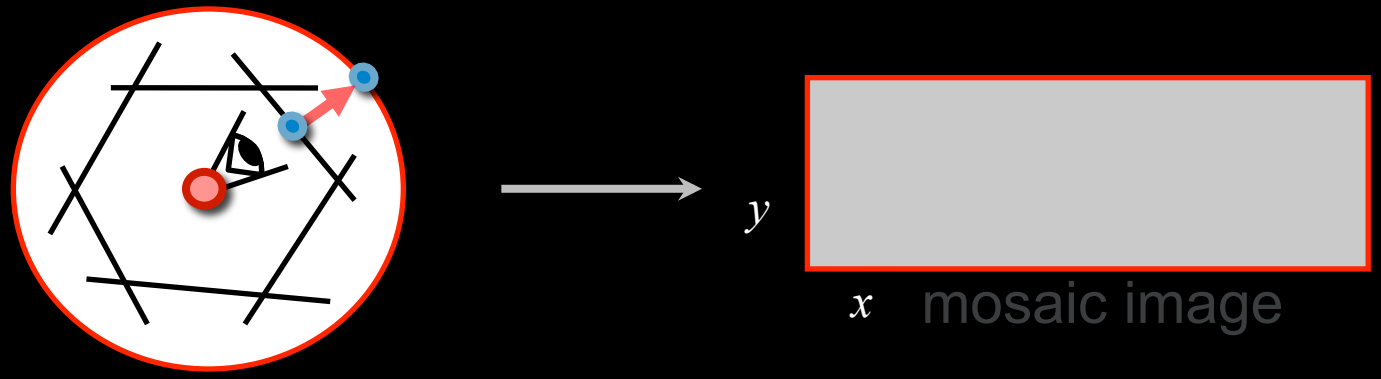


surface of cylinder

cylindrical reprojection

Cylindrical panoramas

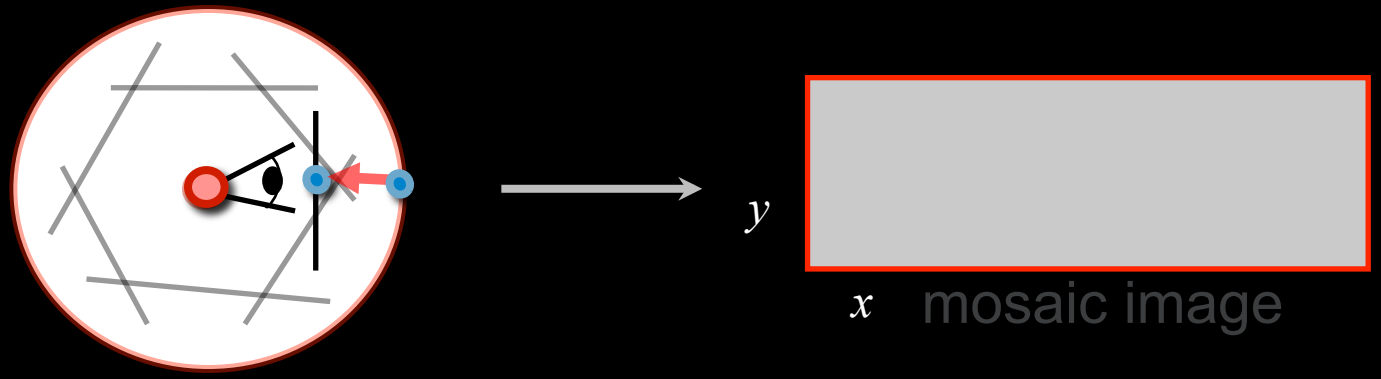
- What if you want a 360° panorama?



- Project each image onto a cylinder
- A cylindrical image is a rectangular array

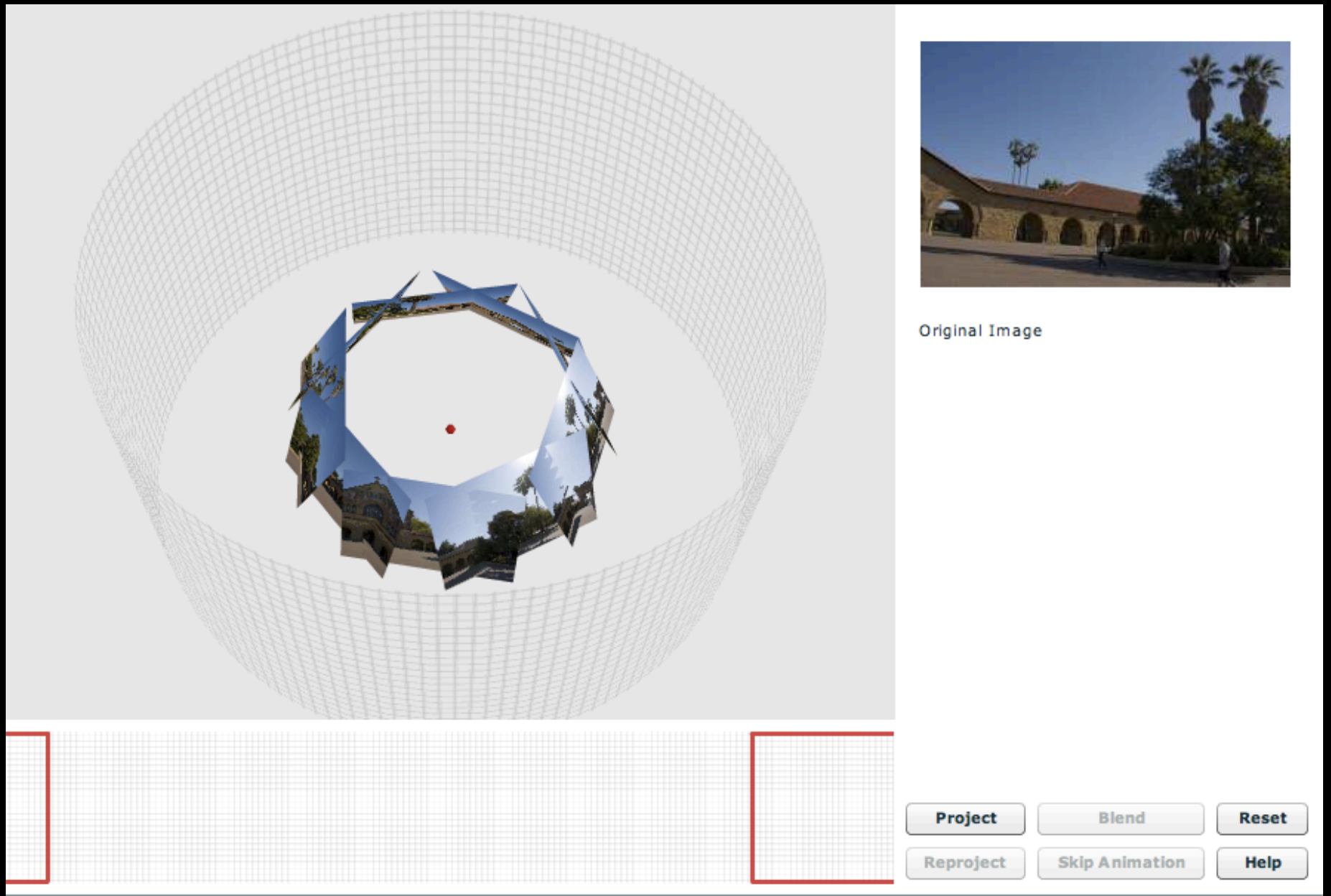
Cylindrical panoramas

- What if you want a 360° panorama?

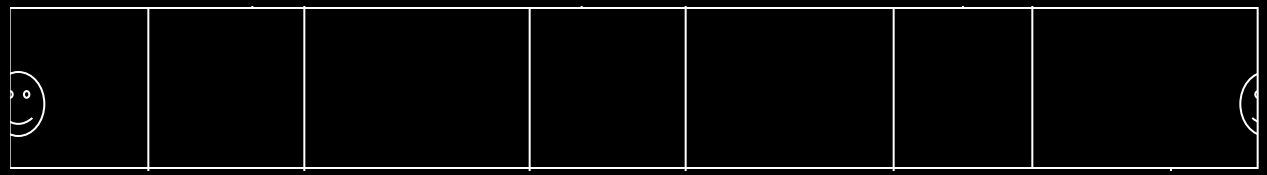


- Project each image onto a cylinder
- A cylindrical image is a rectangular array
- To view without distortion
  - reproject a portion of the cylinder onto a picture plane representing the display screen

Demo http://graphics.stanford.edu/courses/cs178/applets/projection.html

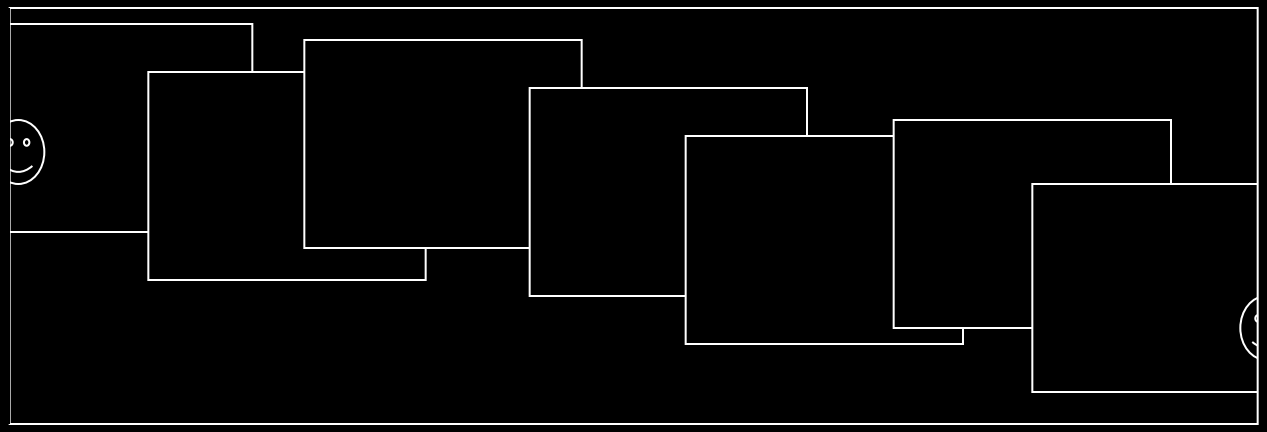


Assembling the panorama



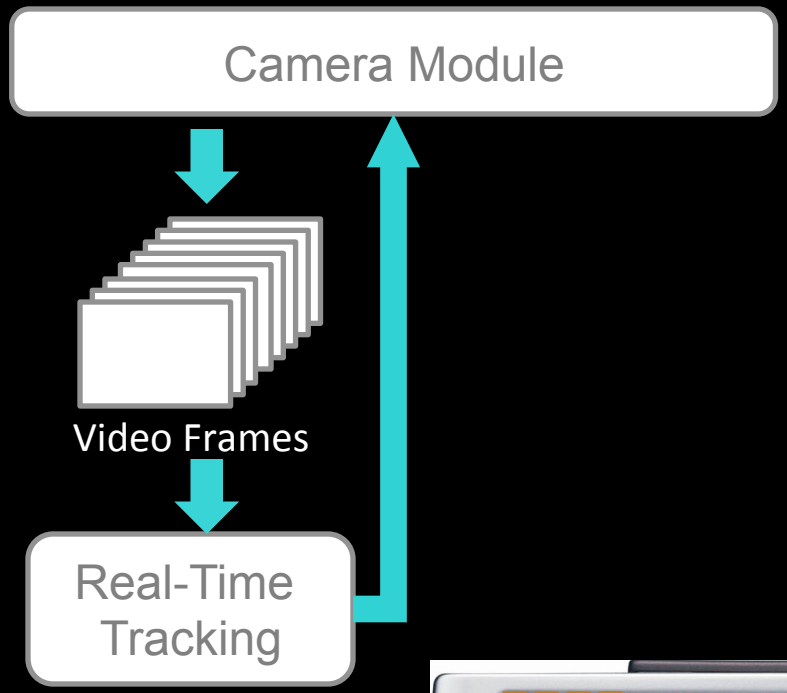
- Stitch pairs together, blend, then crop

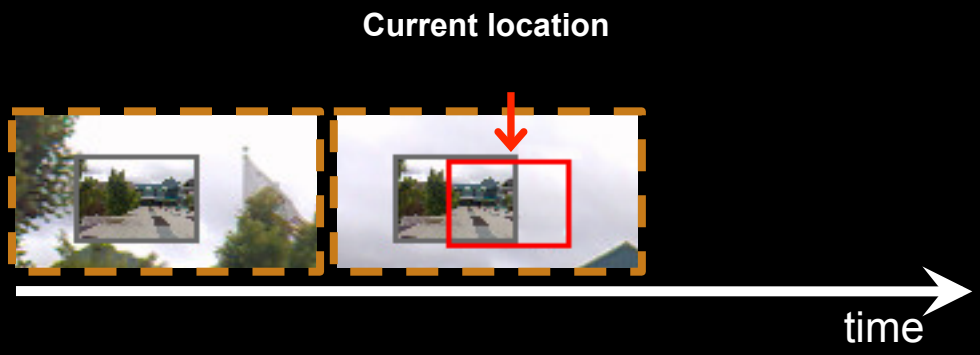
Problem: Drift



- Vertical Error accumulation
  - small (vertical) errors accumulate over time
  - apply correction so that sum = 0 (for 360° panorama)
- Horizontal Error accumulation
  - can reuse first/last image to find the right panorama radius

Registration in practice: tracking





Viewfinder alignment for tracking

Andrew Adams, Natasha Gelfand, Kari Pulli
Viewfinder Alignment
Eurographics 2008

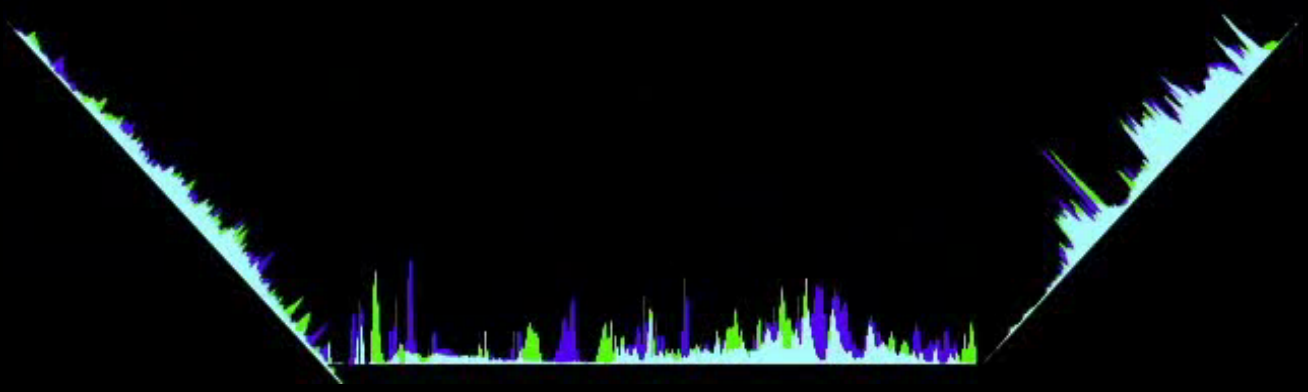
http://graphics.stanford.edu/papers/viewfinderalignment/

Project gradients along columns and rows

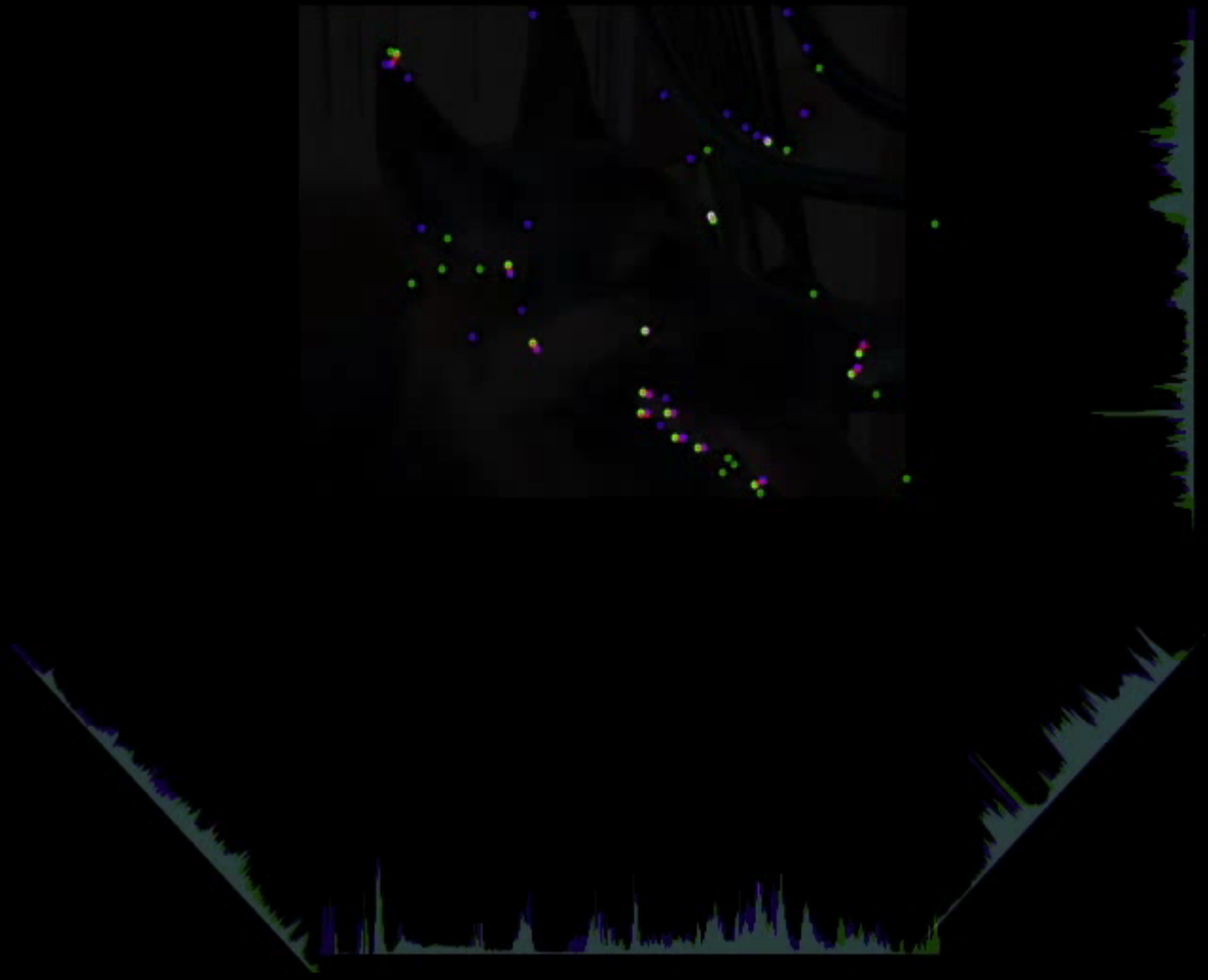
… diagonal gradients along diagonals …

… and find corners

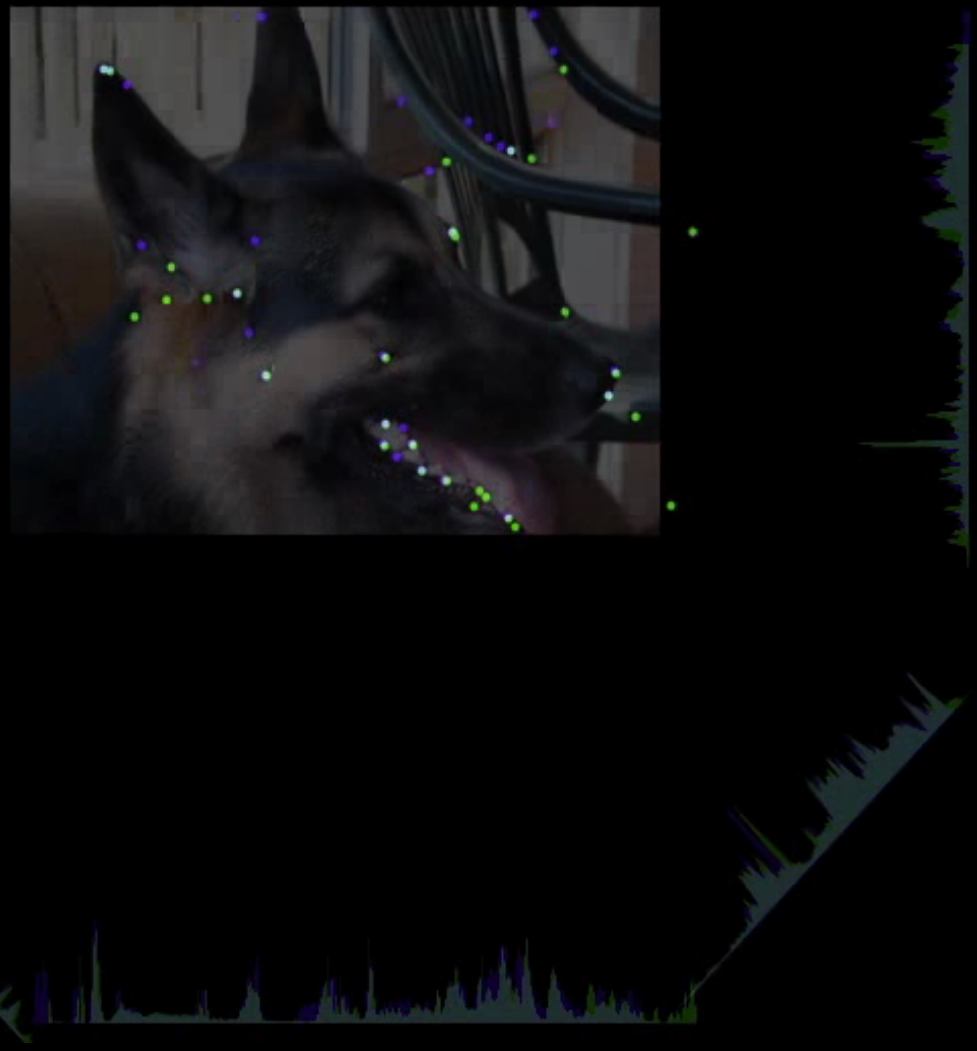
Overlap and match the gradient projections and determine translation



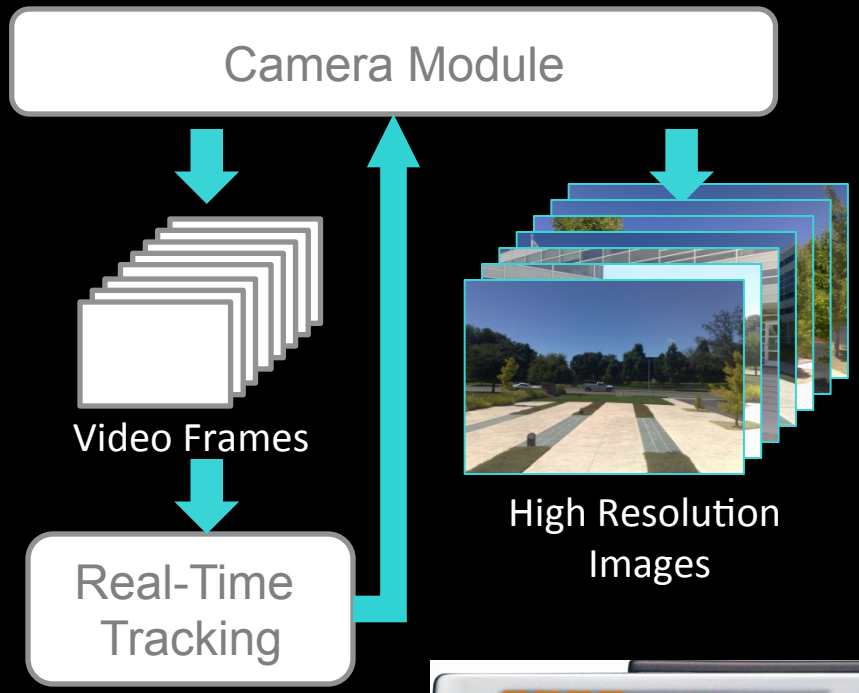
Apply the best translation to corners

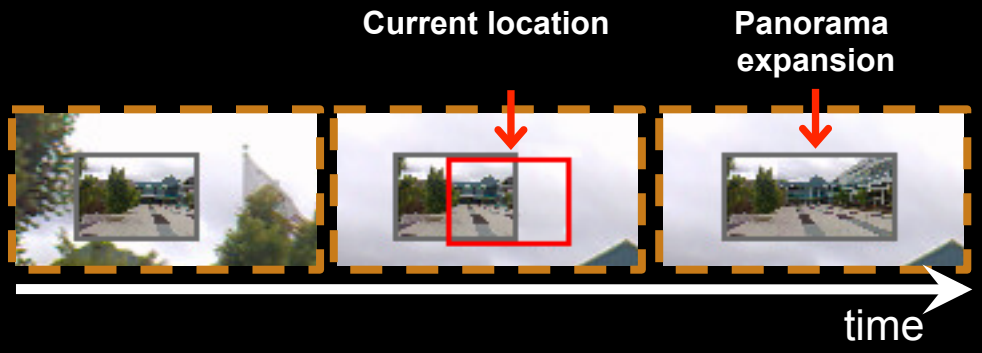


Match corners, refine translation & rotation

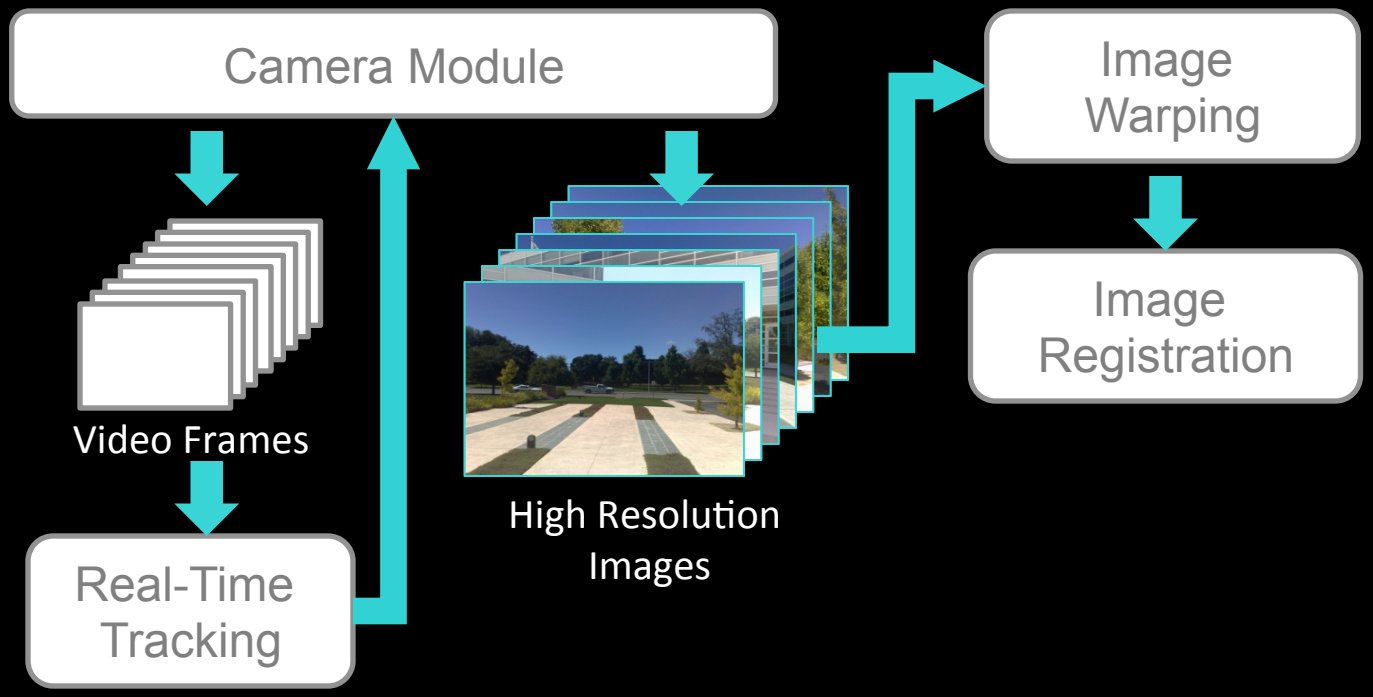


System overview

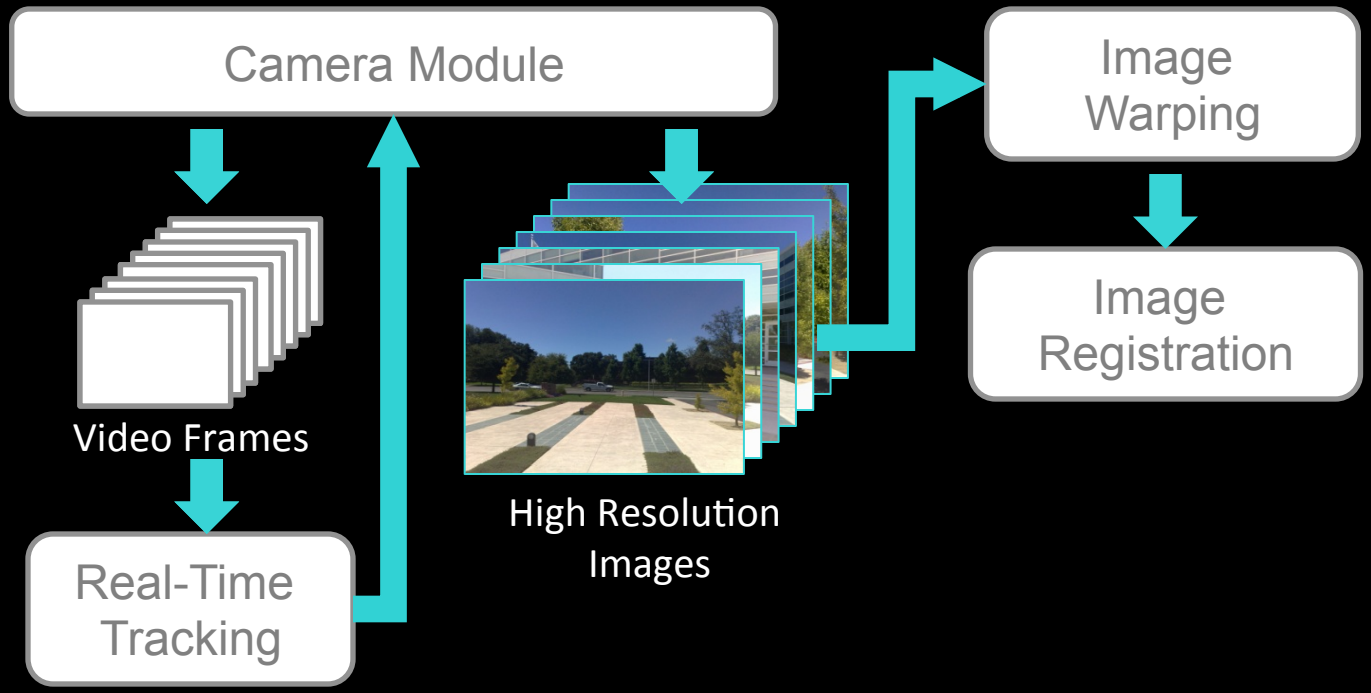




System overview

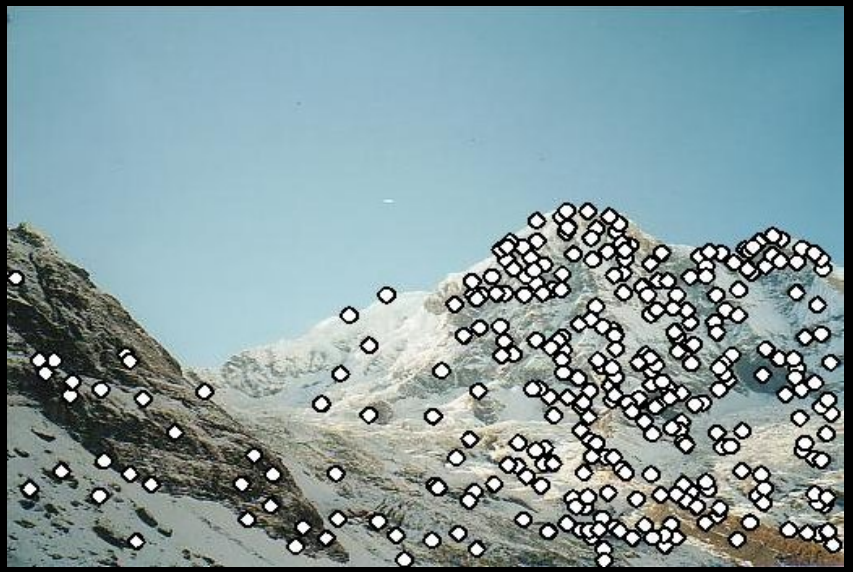


System overview

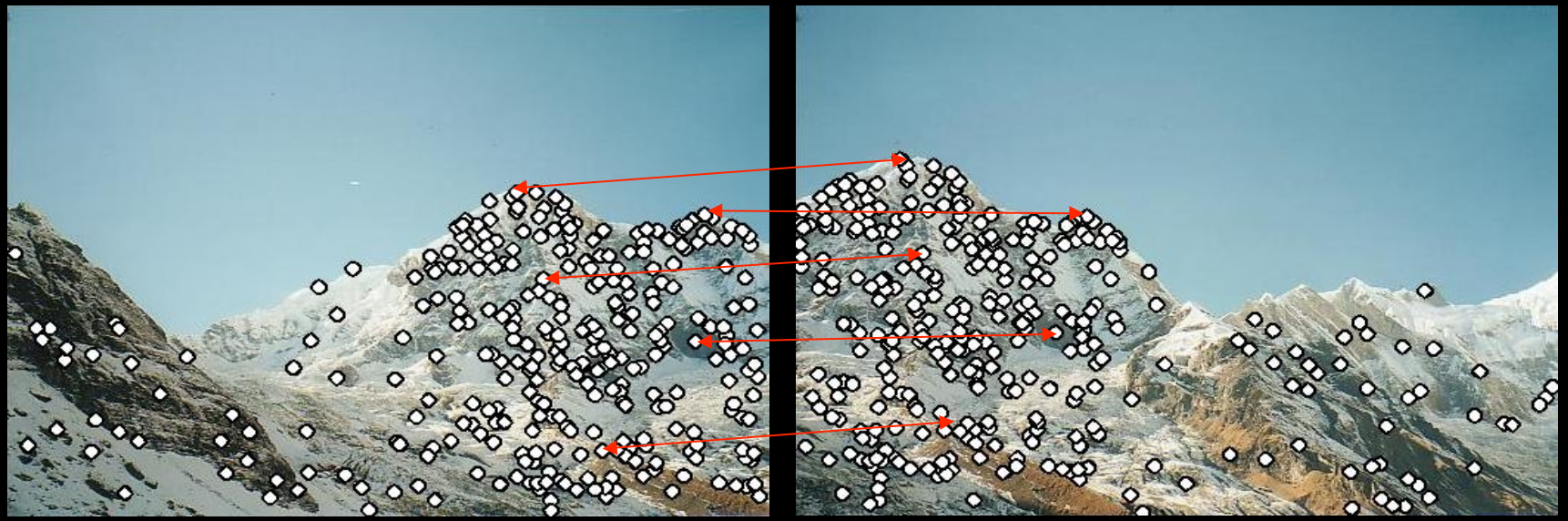


We need to match (align) images

Detect feature points in both images

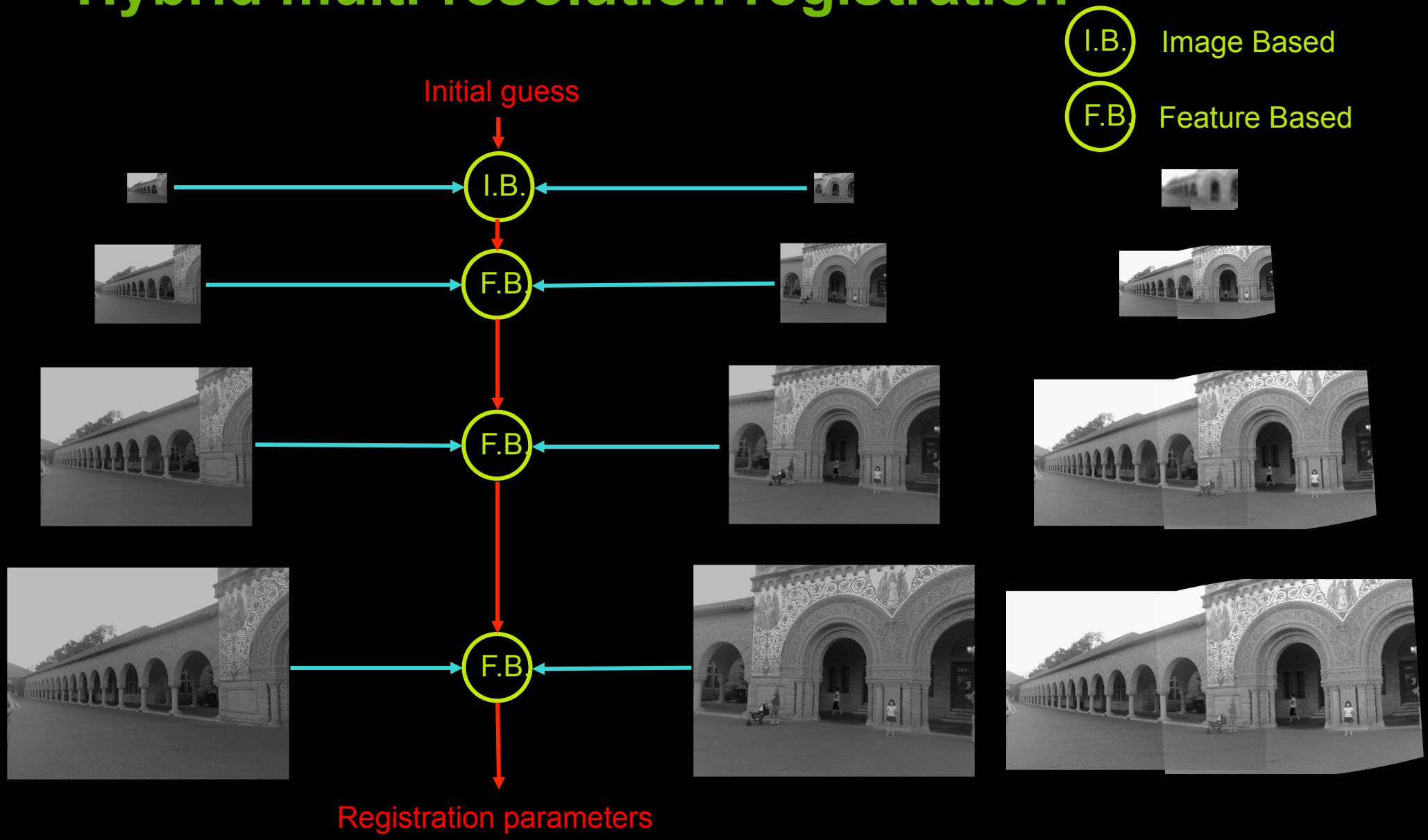


Find corresponding pairs



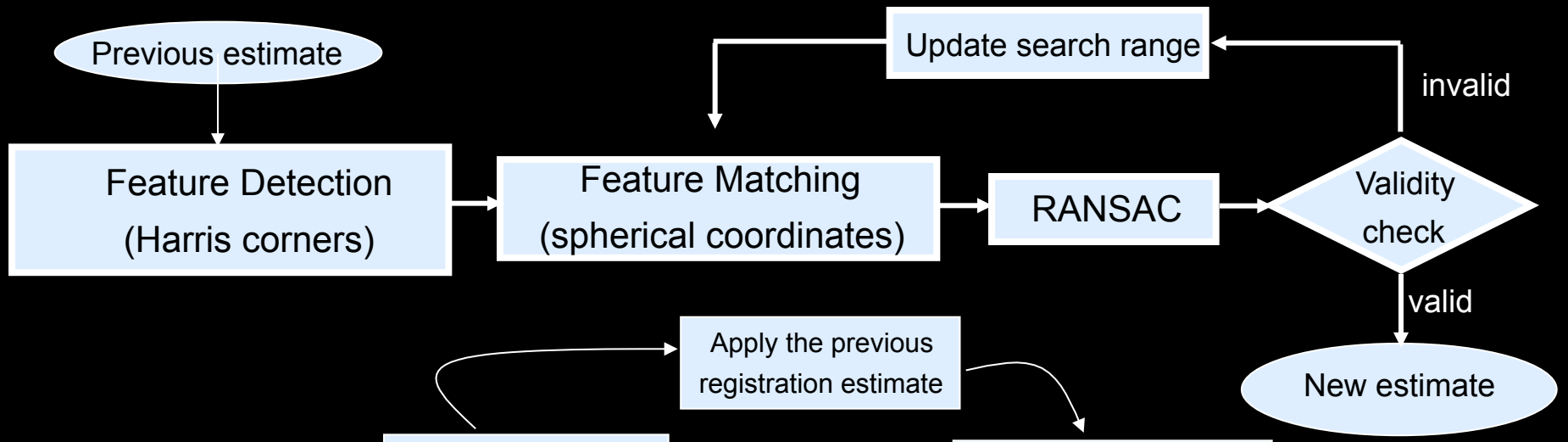
Use these pairs to align images

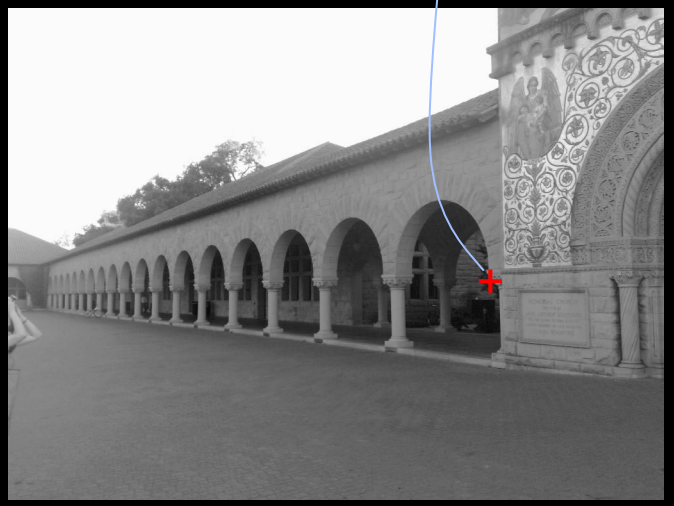
Hybrid multi-resolution registration



K. Pulli, M. Tico, Y. Xiong, X. Wang, C-K. Liang, “Panoramic Imaging System for Camera Phones”, ICCE 2010

Feature-based registration





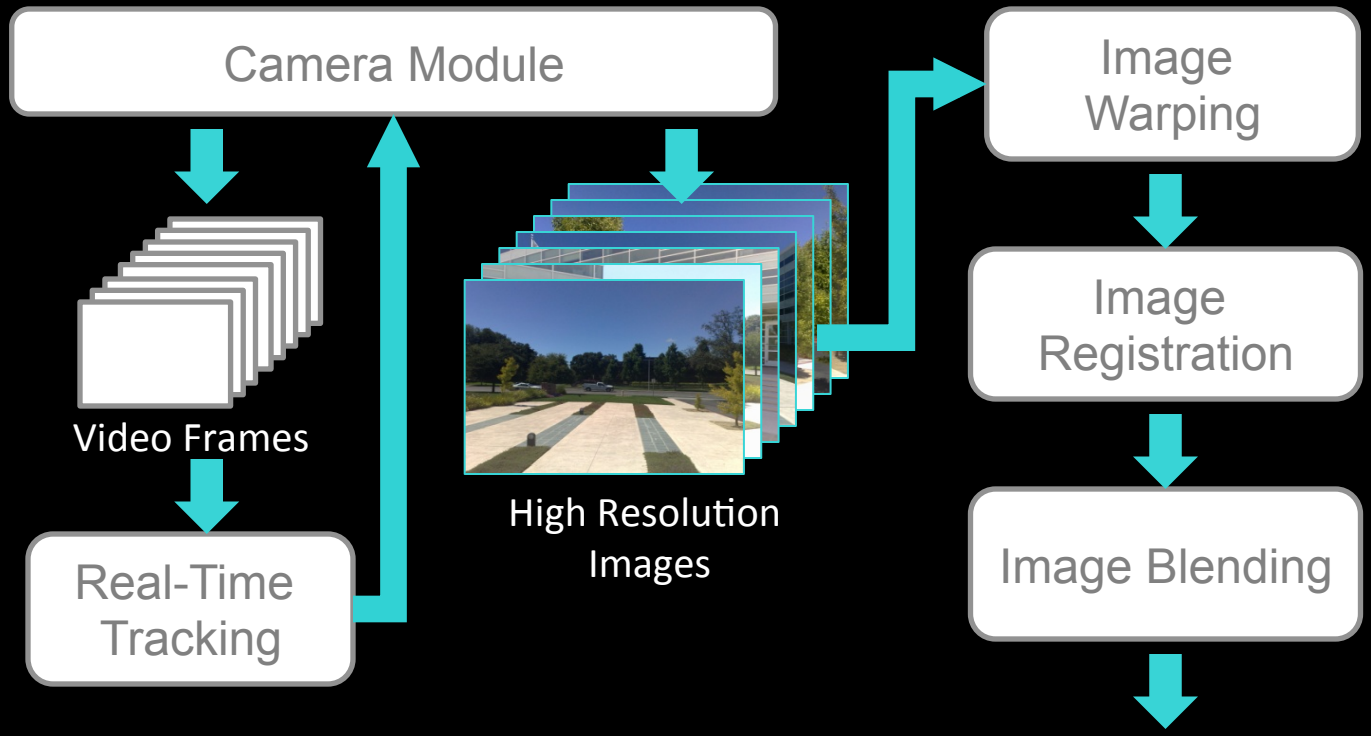
Best block cross-correlation match

Progression of multi-resolution registration

Actual size

Applied to hi-res

System overview



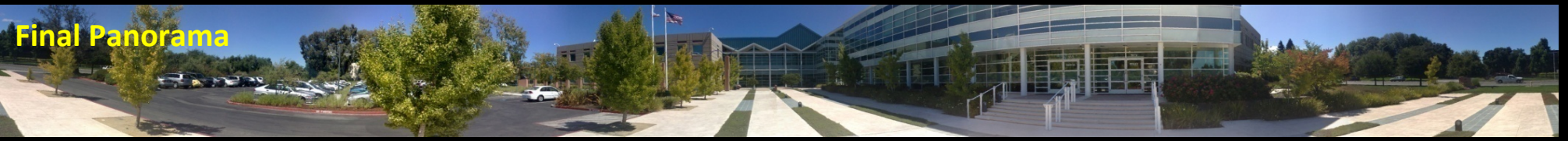


Photo by Marius Tico

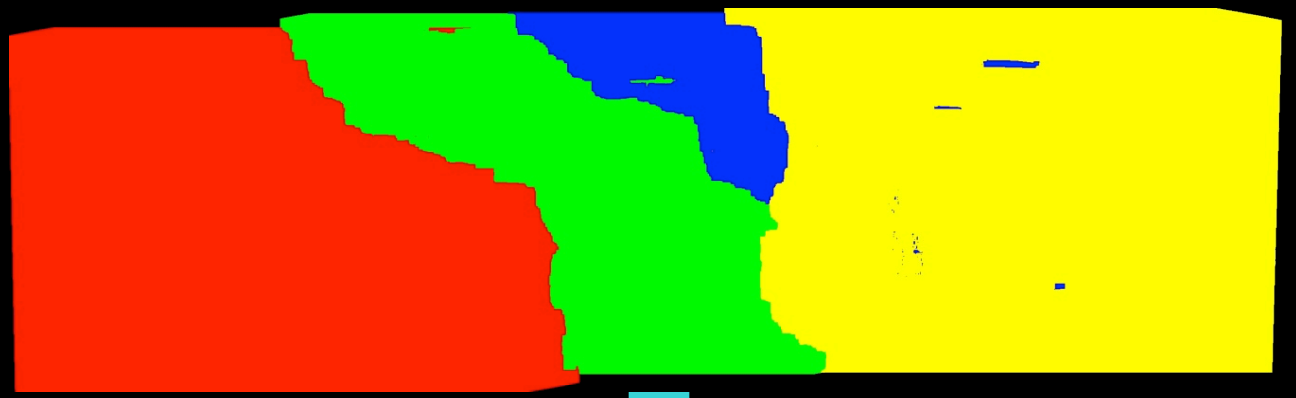
Image blending

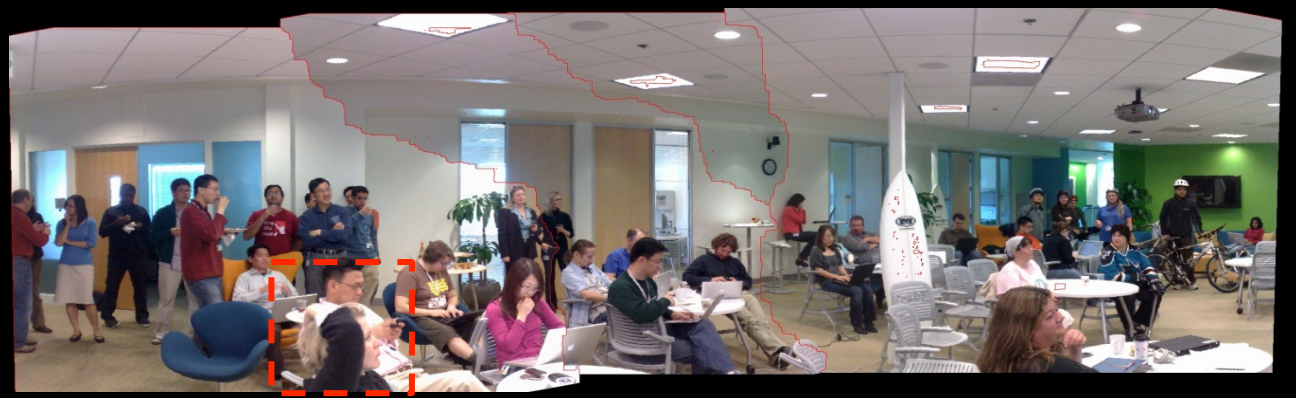
- Directly averaging the overlapped pixels results in ghosting artifacts
  - Moving objects, errors in registration, parallax, etc.

Photo by Chia-Kai Liang

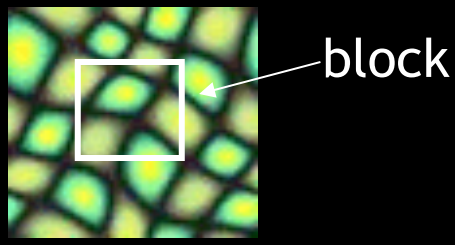
Solution for ghosting: Image labeling

- Assign one input image to each output pixel
  - Optimal assignment can be found by graph cut [Agarwala et al. 2004]

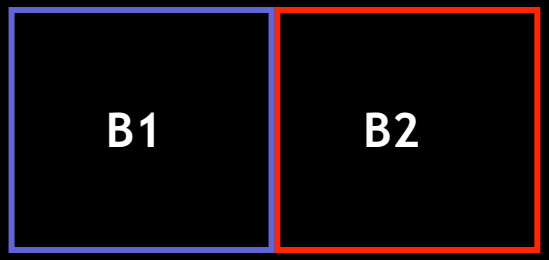




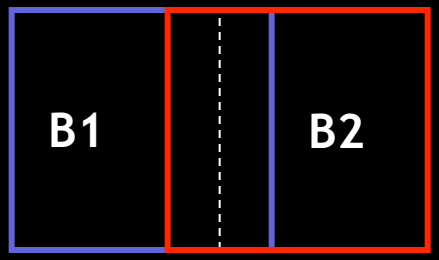
Faster solution with dynamic programming



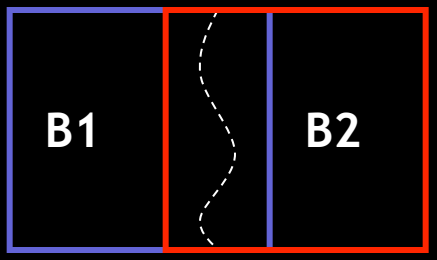
Input texture



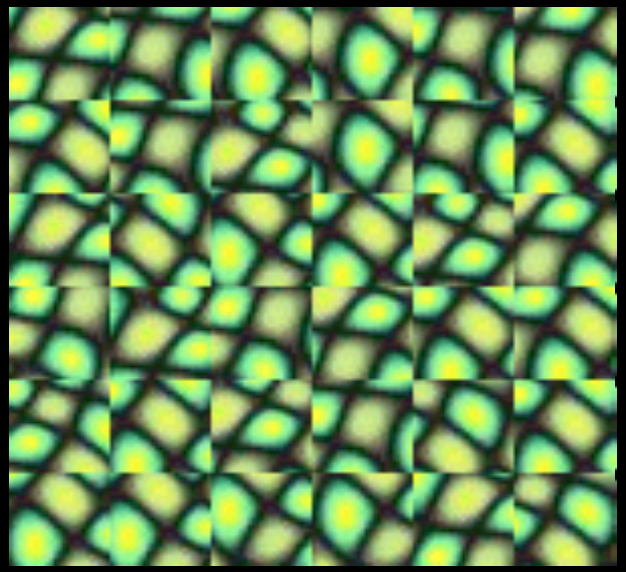
Random placement of blocks

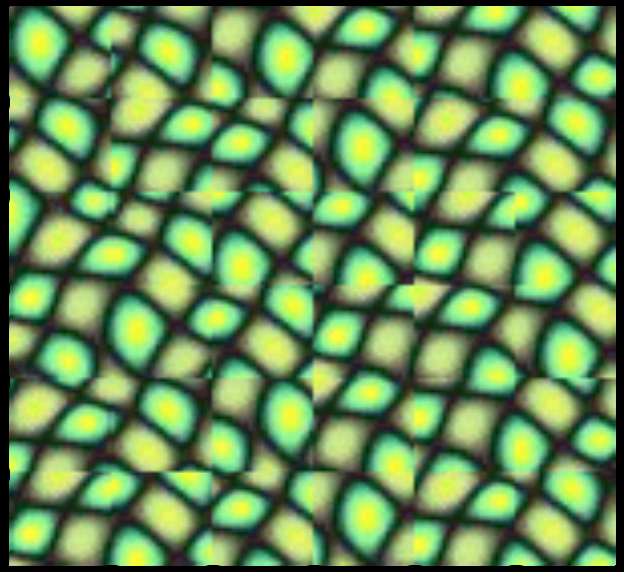


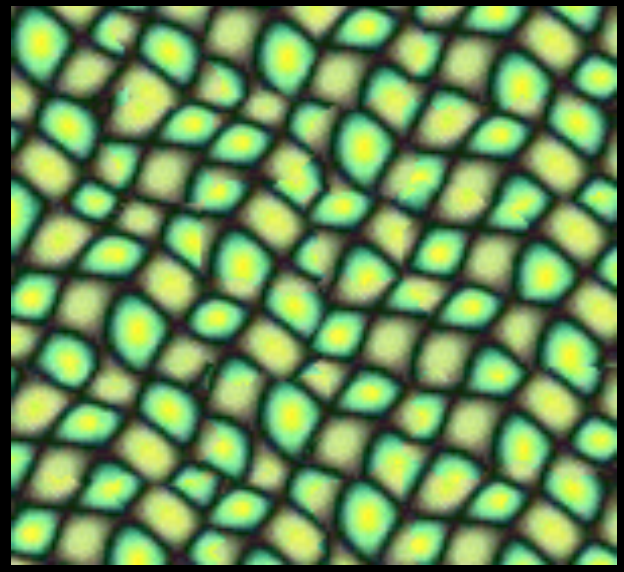
Neighboring blocks constrained by overlap



Minimal error boundary cut

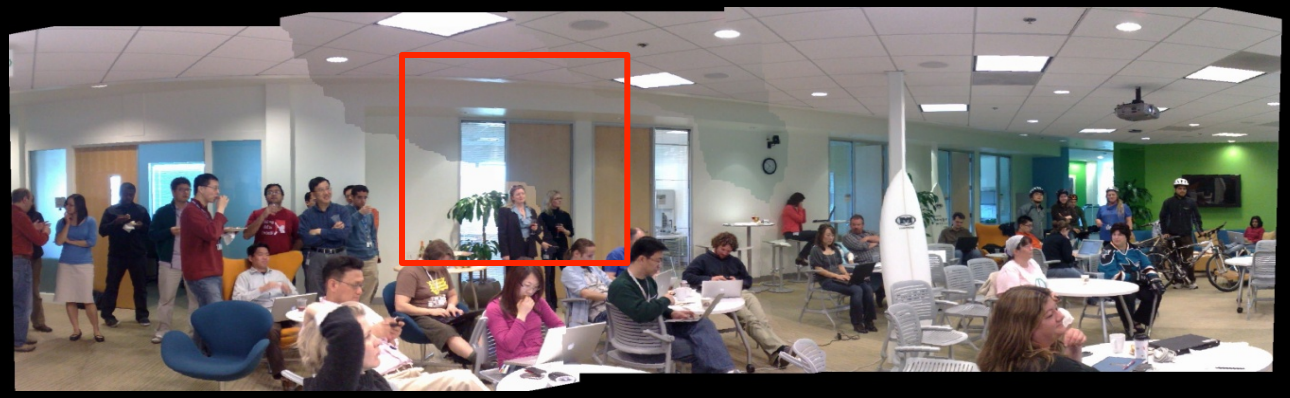




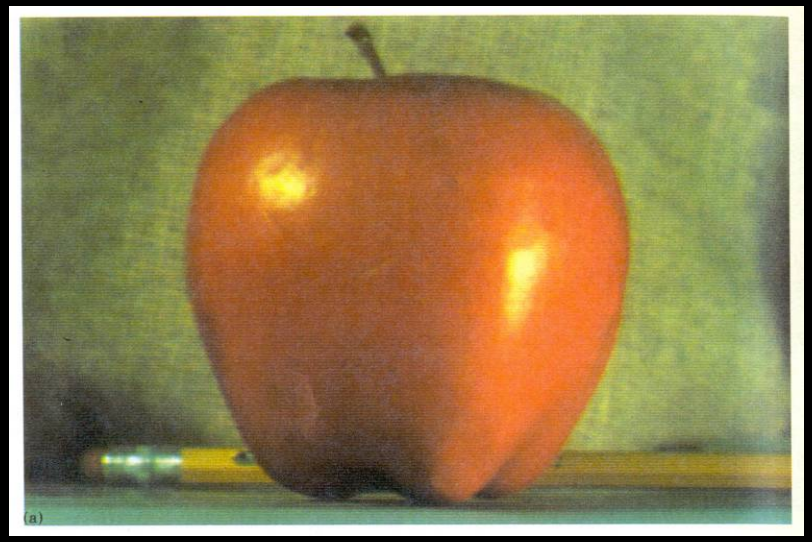


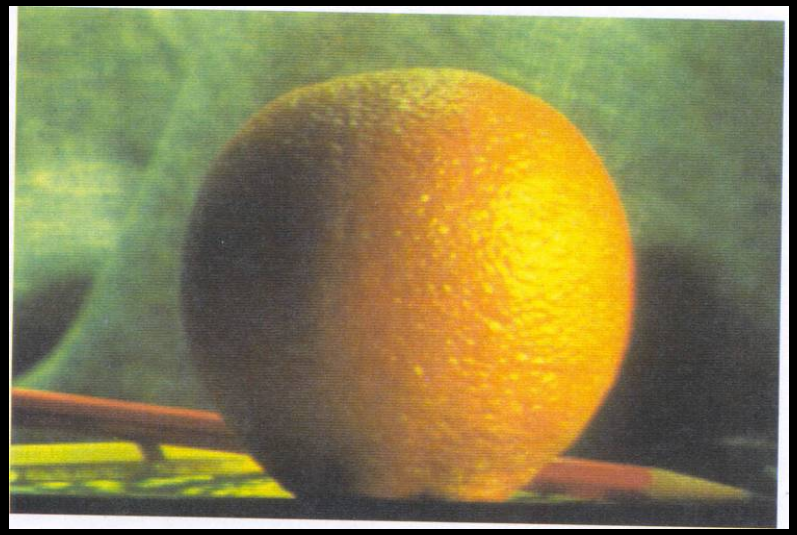
New artifacts

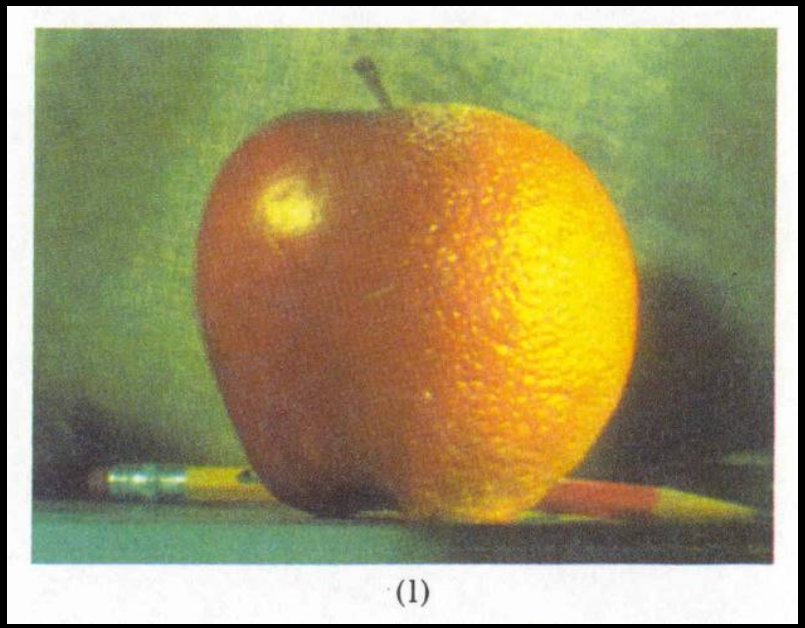
- Inconsistency between pixels from different input images
  - Different exposure/white balance settings
  - Photometric distortions (e.g., vignetting)



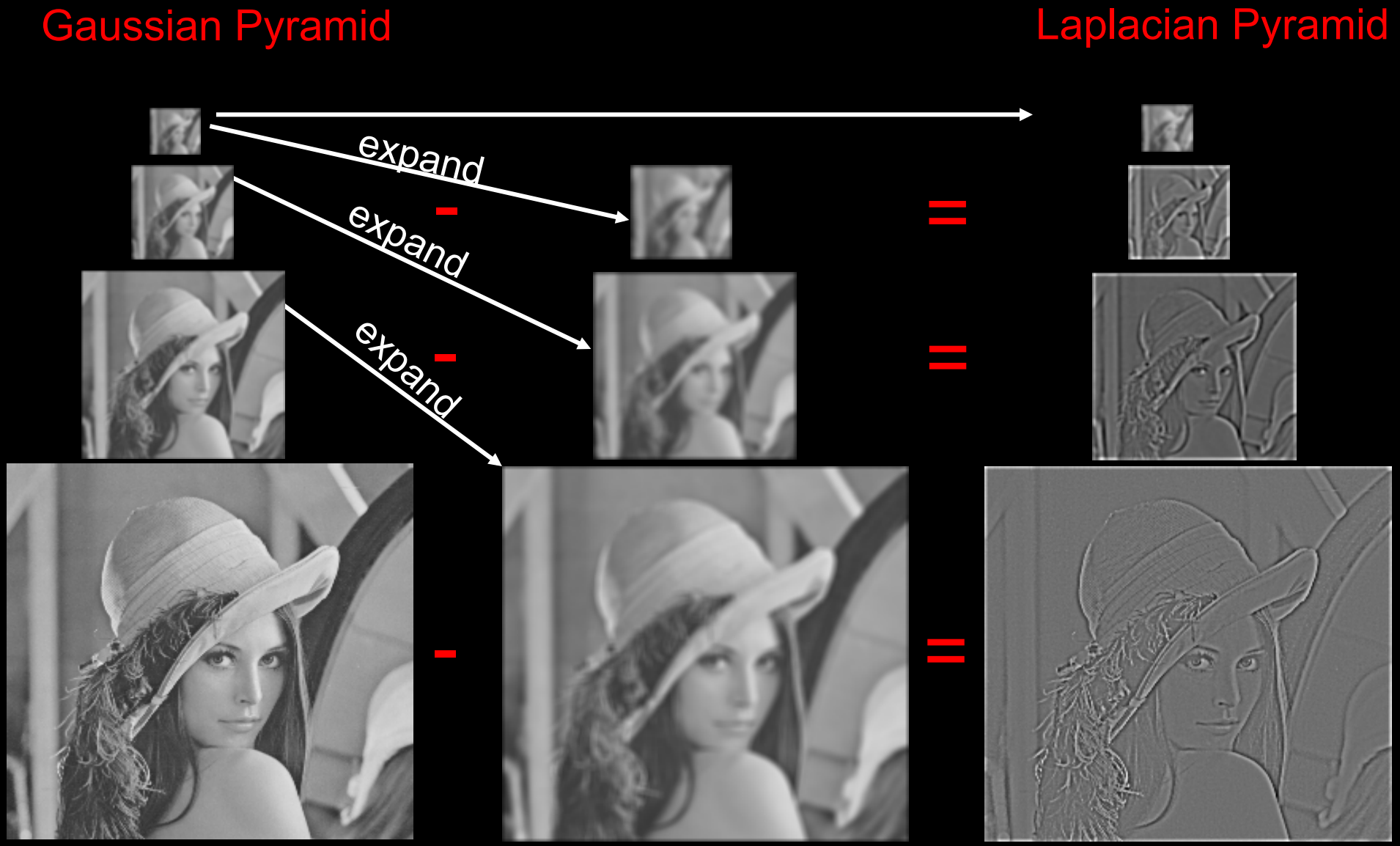
Pyramid Blending







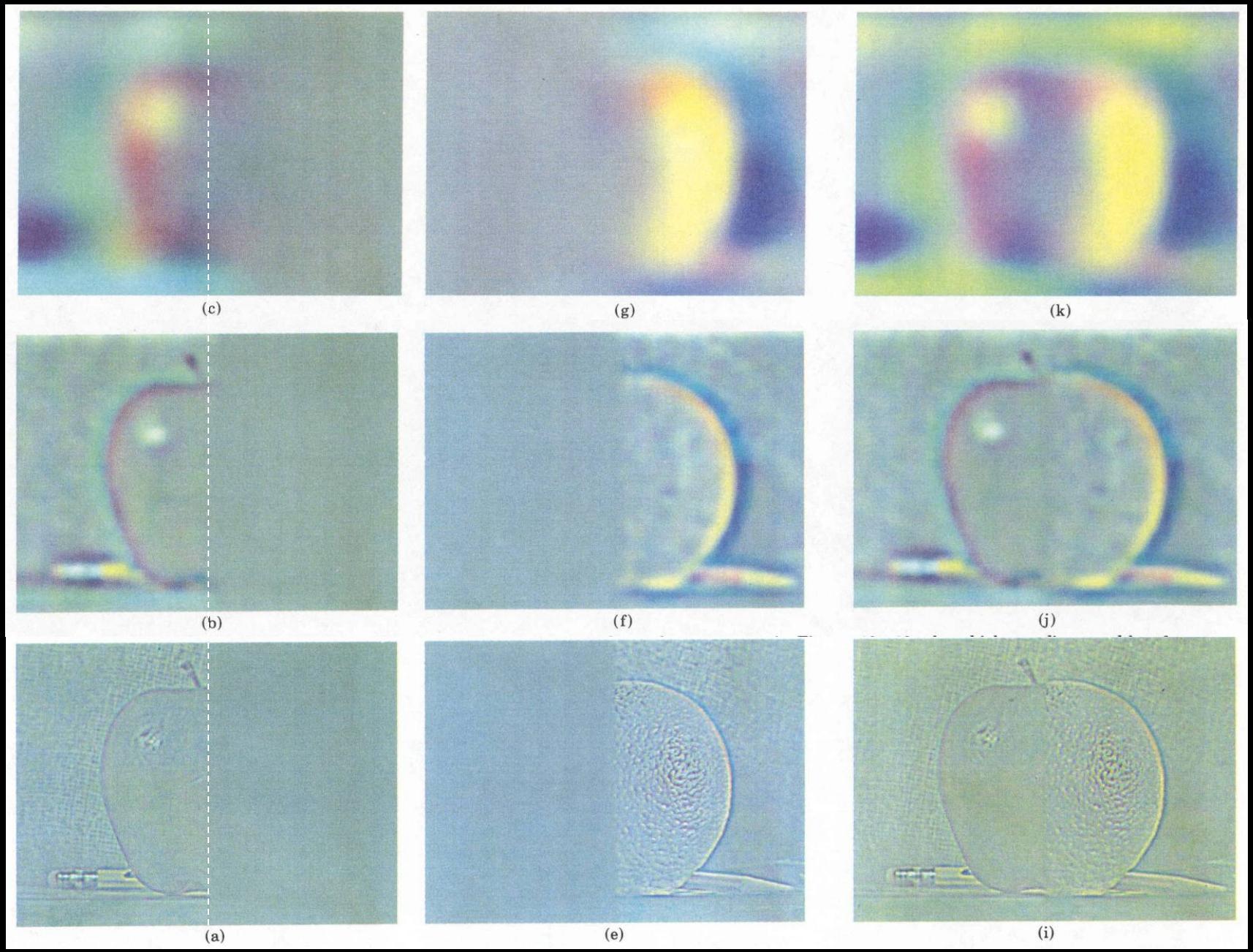
The Laplacian pyramid



Laplacian level 4

Laplacian level 2

Laplacian level 0



left pyramid

right pyramid

blended pyramid

Simplification: Two-band Blending

- Brown & Lowe, 2003
  - Only use two bands: high freq. and low freq.
  - Blends low freq. smoothly

2-band Blending

Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

System overview

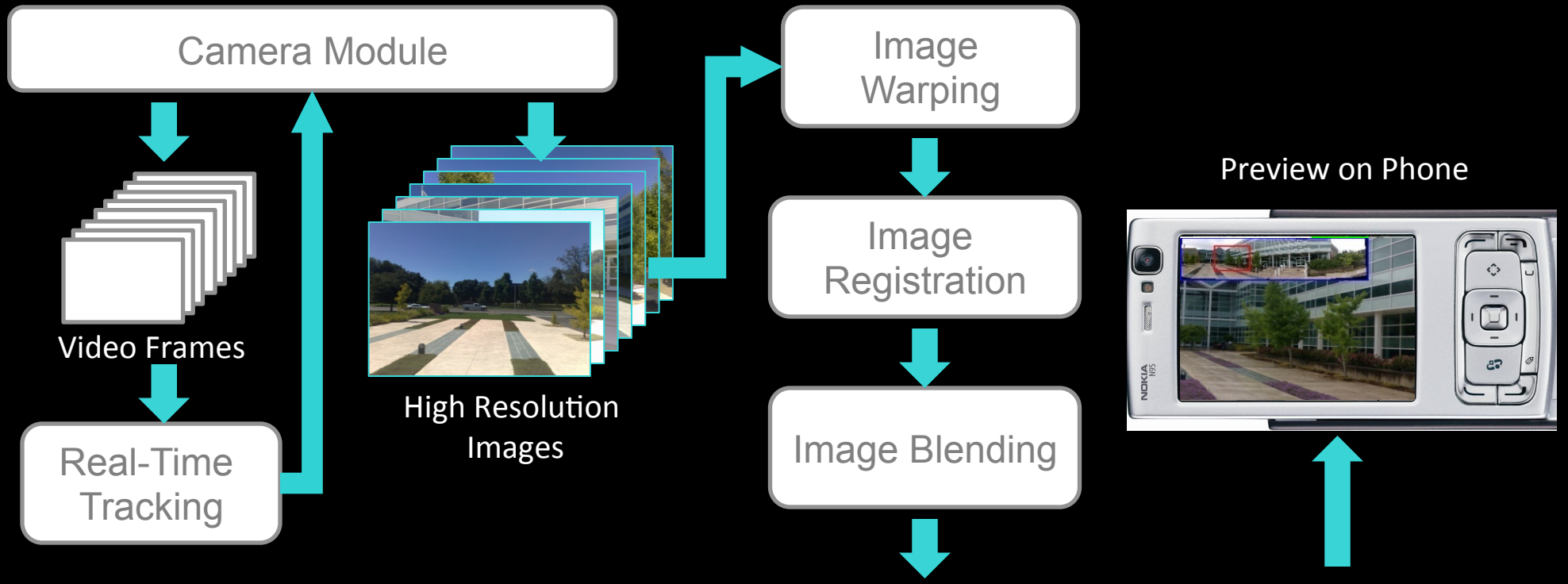
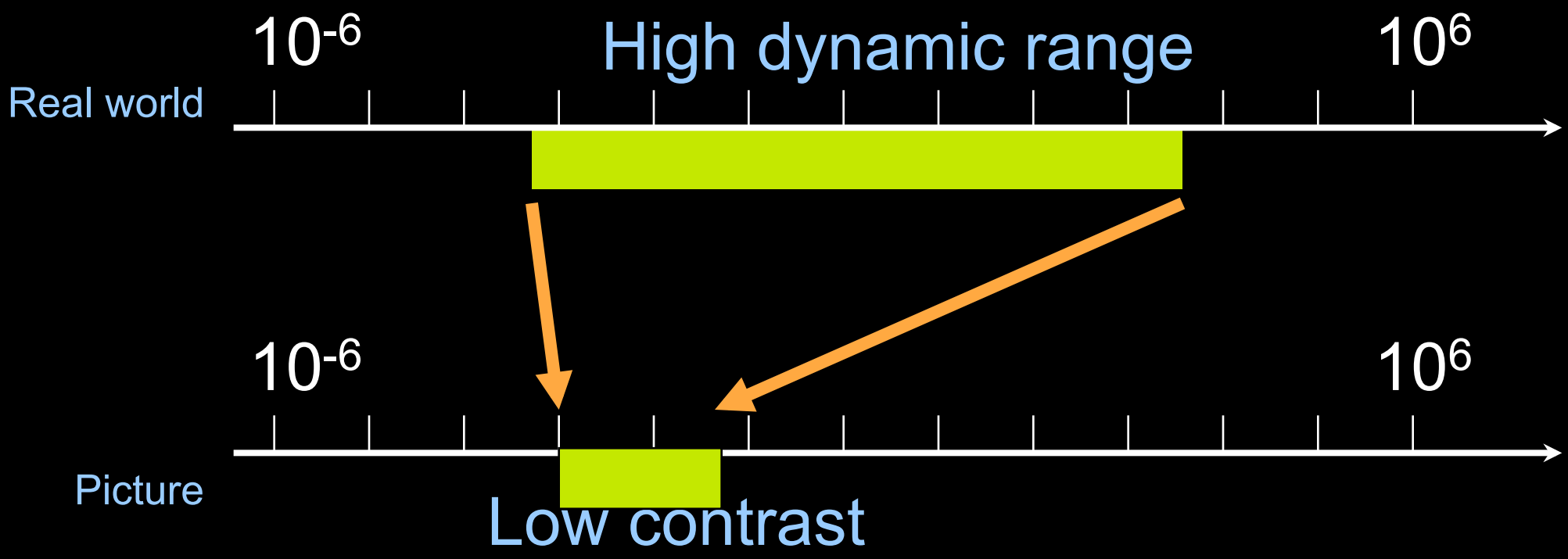


Photo by Marius Tico

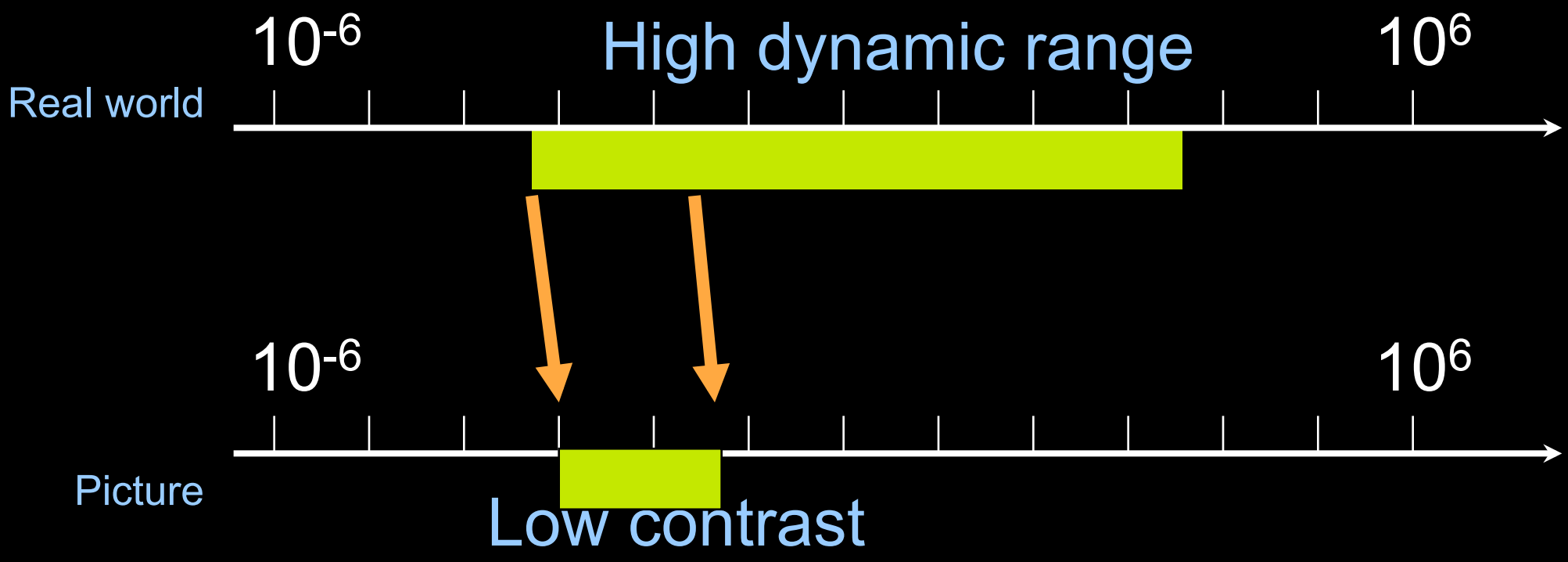
Problems with setting the camera exposure level

- Under-exposed
  - Highlight details captured
  - Shadow details lost
- Over-exposed
  - Highlight details lost
  - Shadow details captured

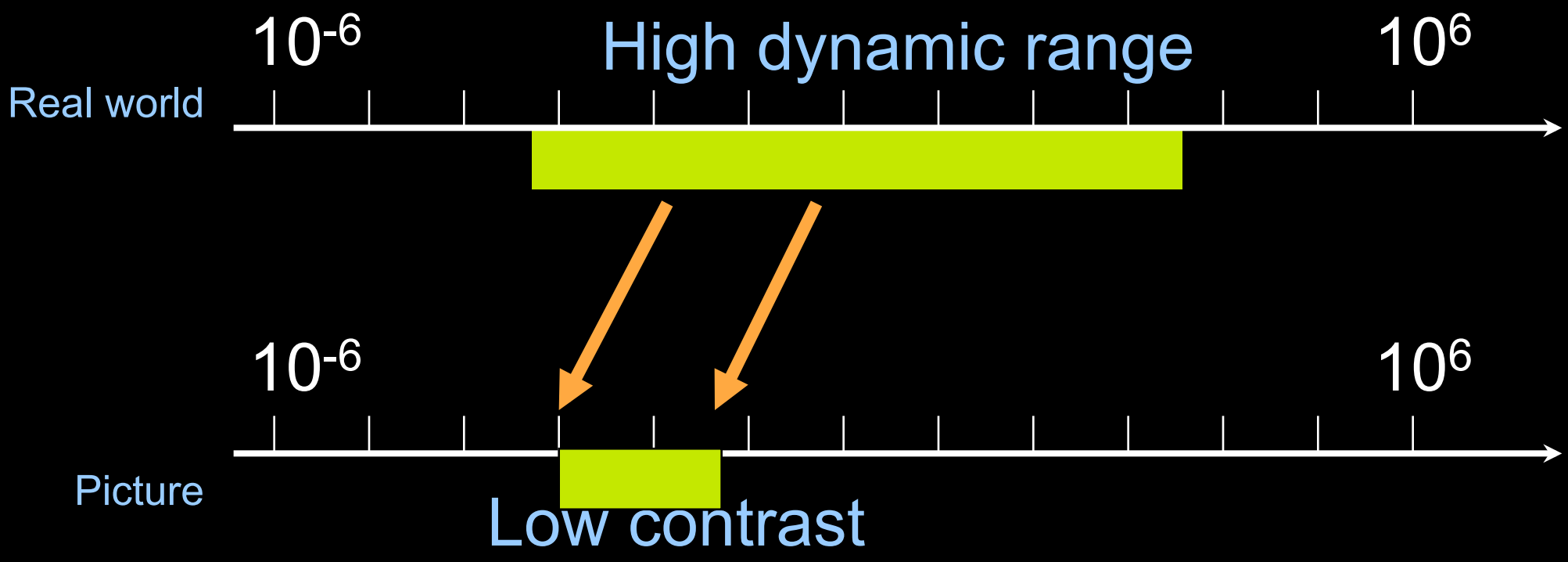
Multiple exposure photography



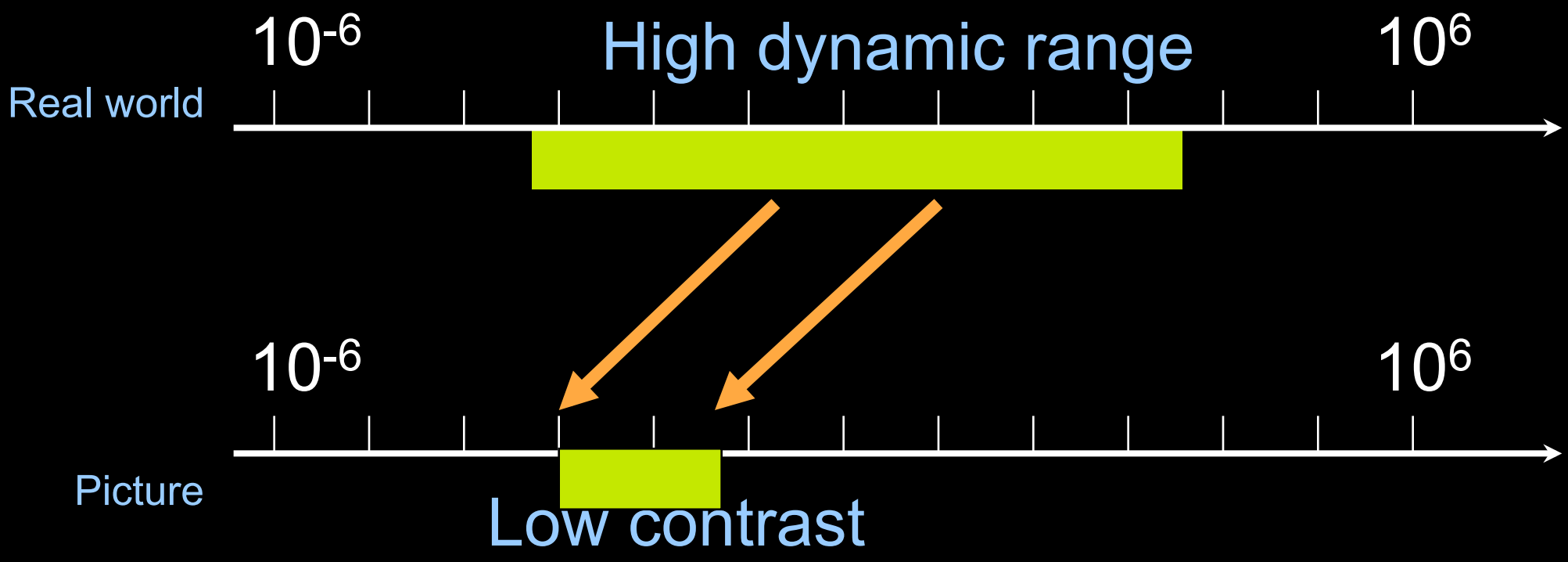
Multiple exposure photography



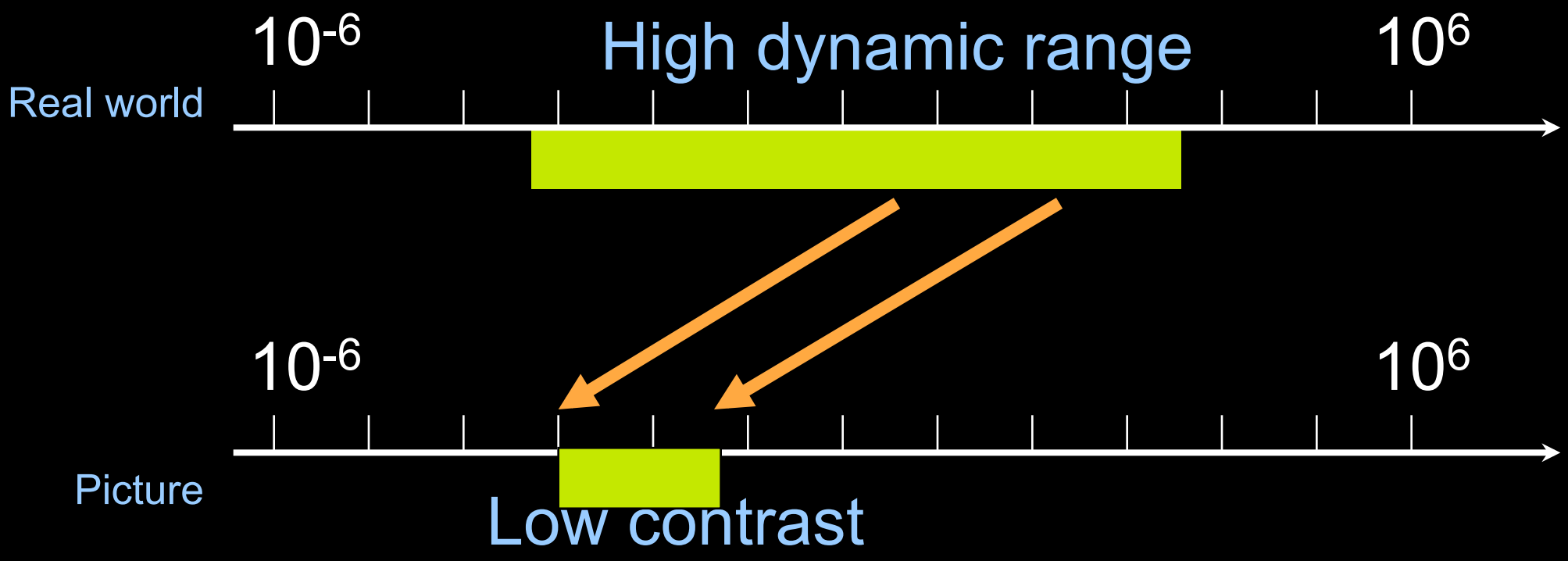
Multiple exposure photography



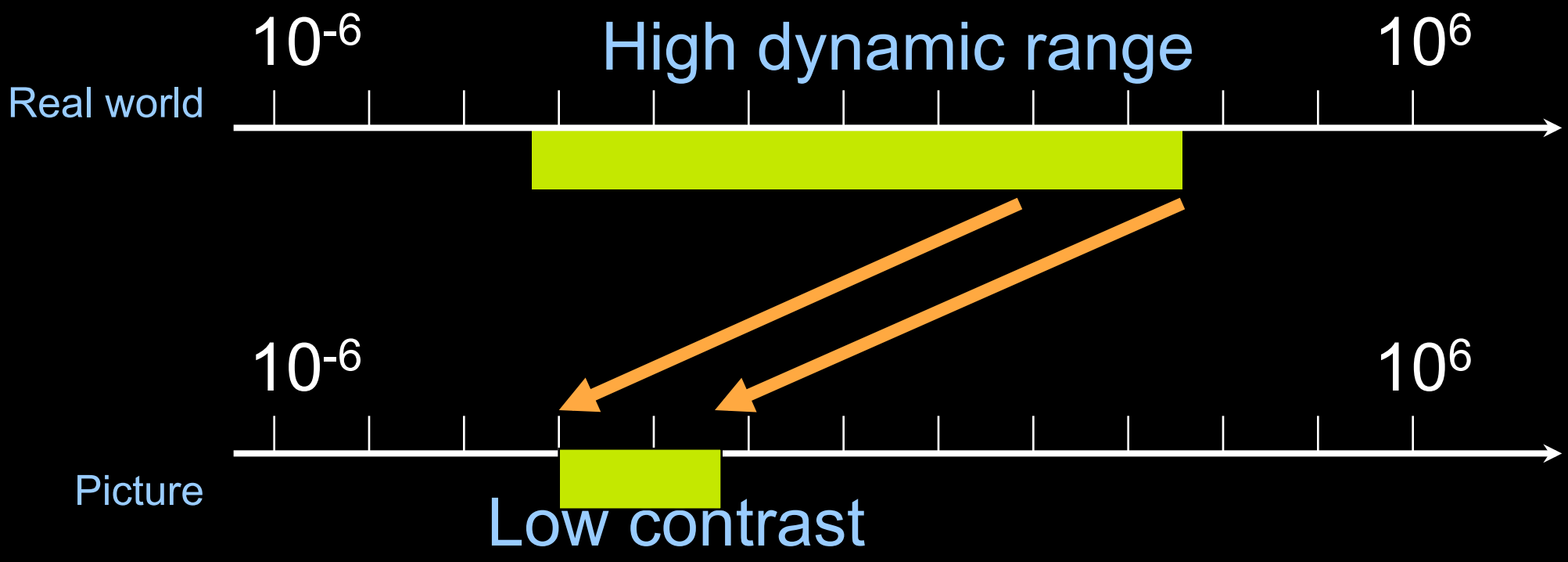
Multiple exposure photography



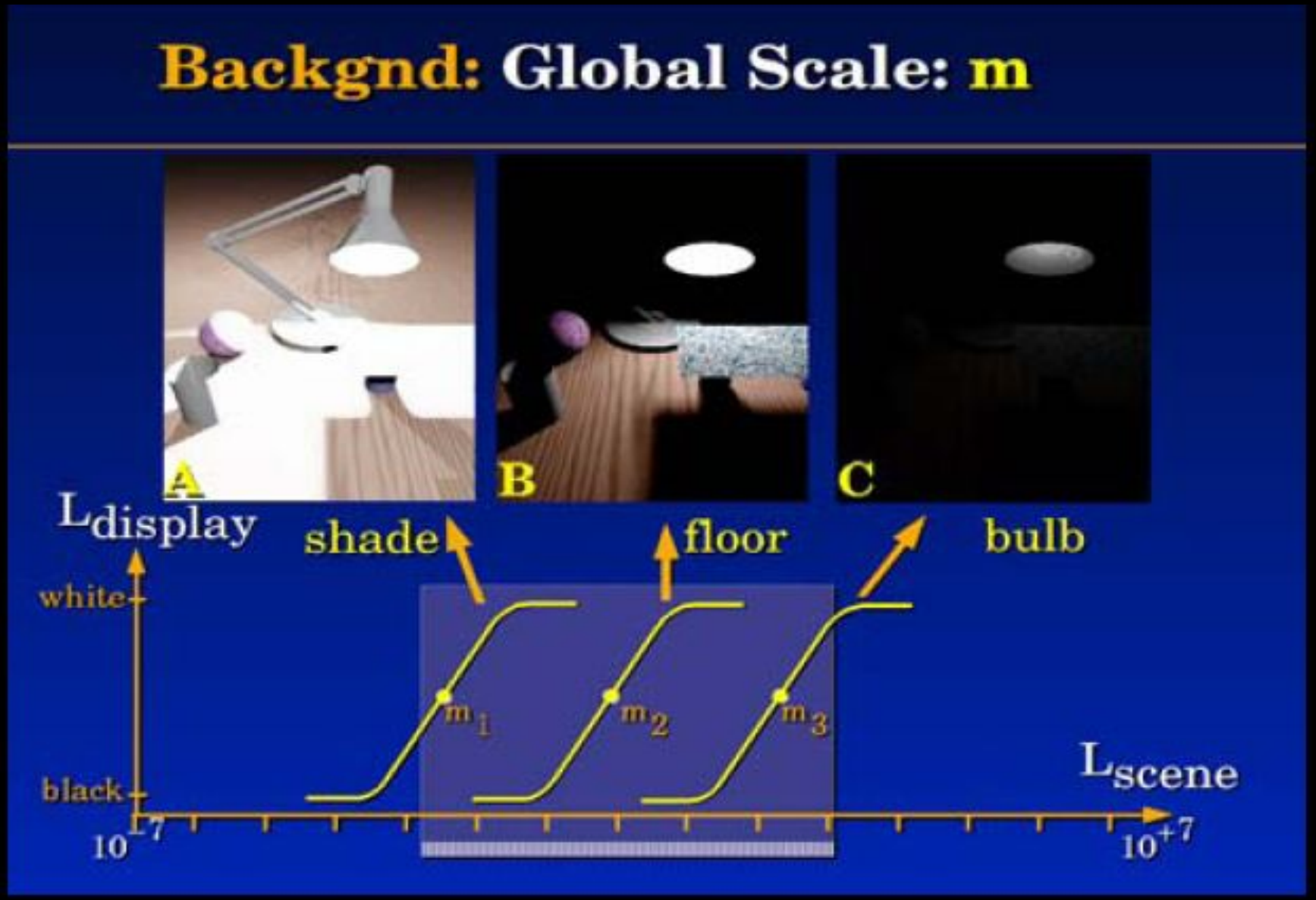
Multiple exposure photography



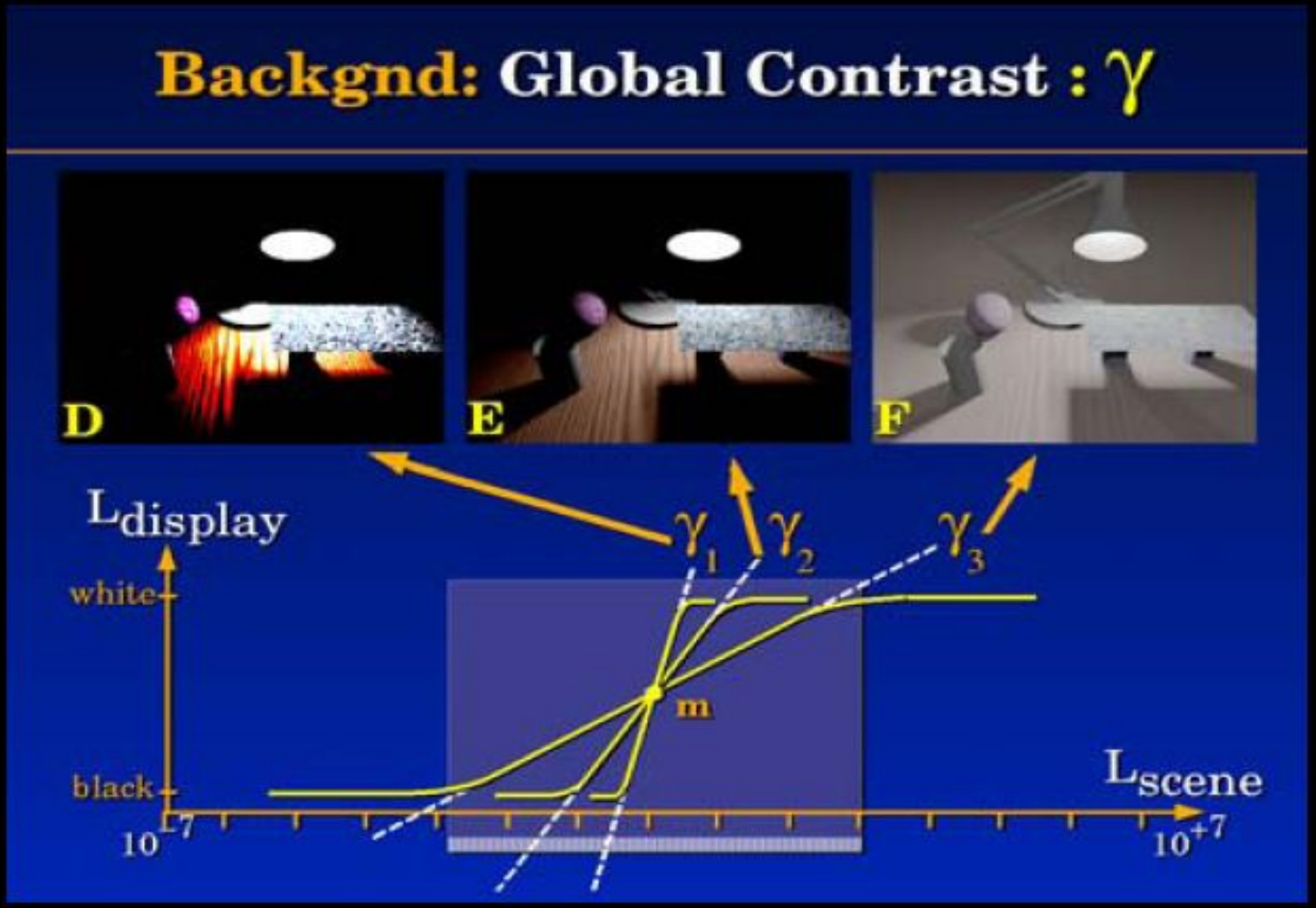
Multiple exposure photography



Tone mapping is not easy



Tone mapping is not easy



Contrast reduction in the digital world

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

Naïve: Gamma compression

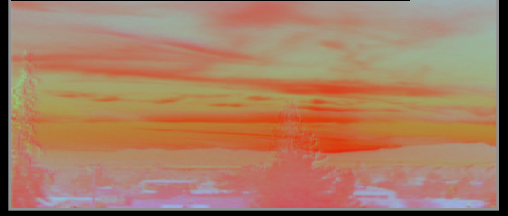
- $X \rightarrow X^{\gamma}$ (where γ=0.5 in our case)
- But… colors are washed-out

Gamma compression on intensity

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

Intensity

Color



Gamma on intensity

Let highlights saturate

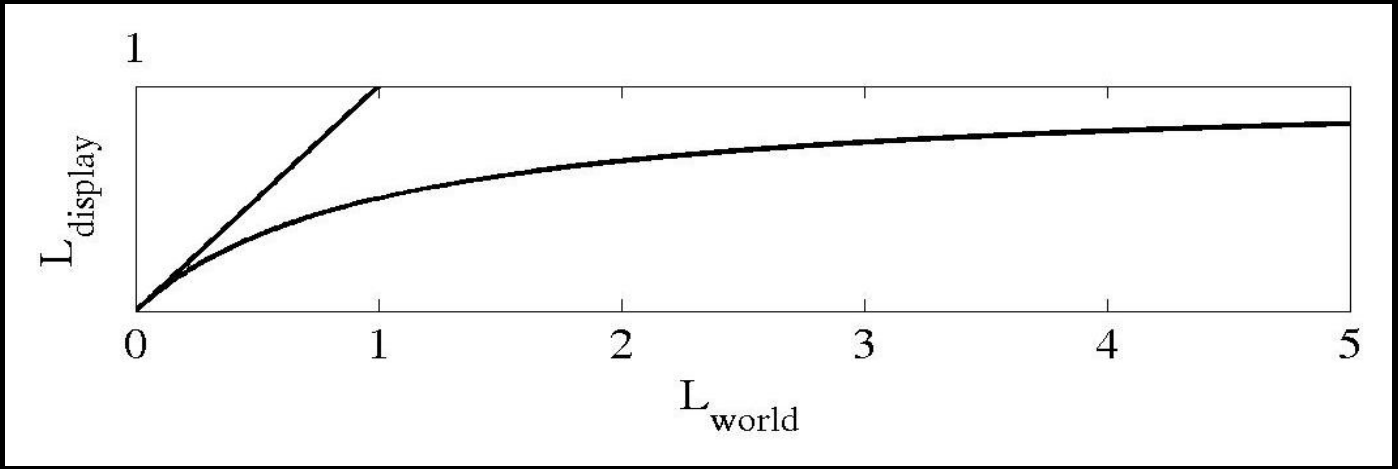
- Darkest 0.1% scaled to display device

Simple global operator (Reinhard et al.)

- Compression curve needs to
  - bring everything within range
  - leave dark areas alone
- In other words
  - asymptote is 1
  - derivative at 0 is 1

http://www.cs.utah.edu/~reinhard/cdrom/tonemap.pdf

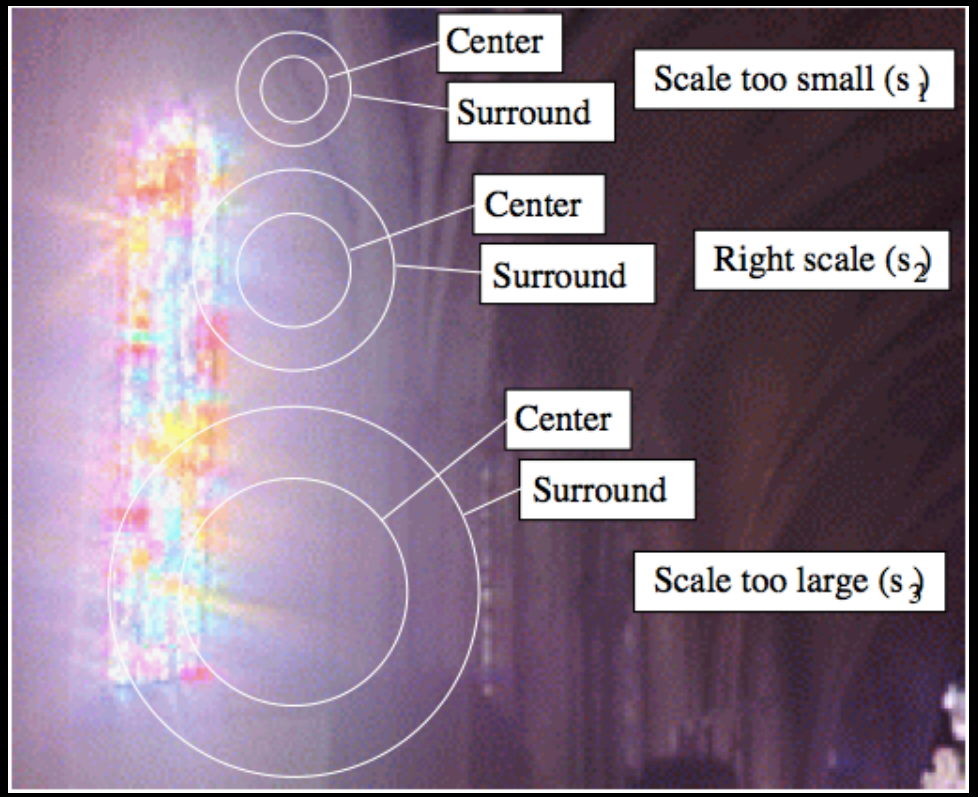
$$L_{display} = \frac{L_{world}}{1 + L_{world}}$$

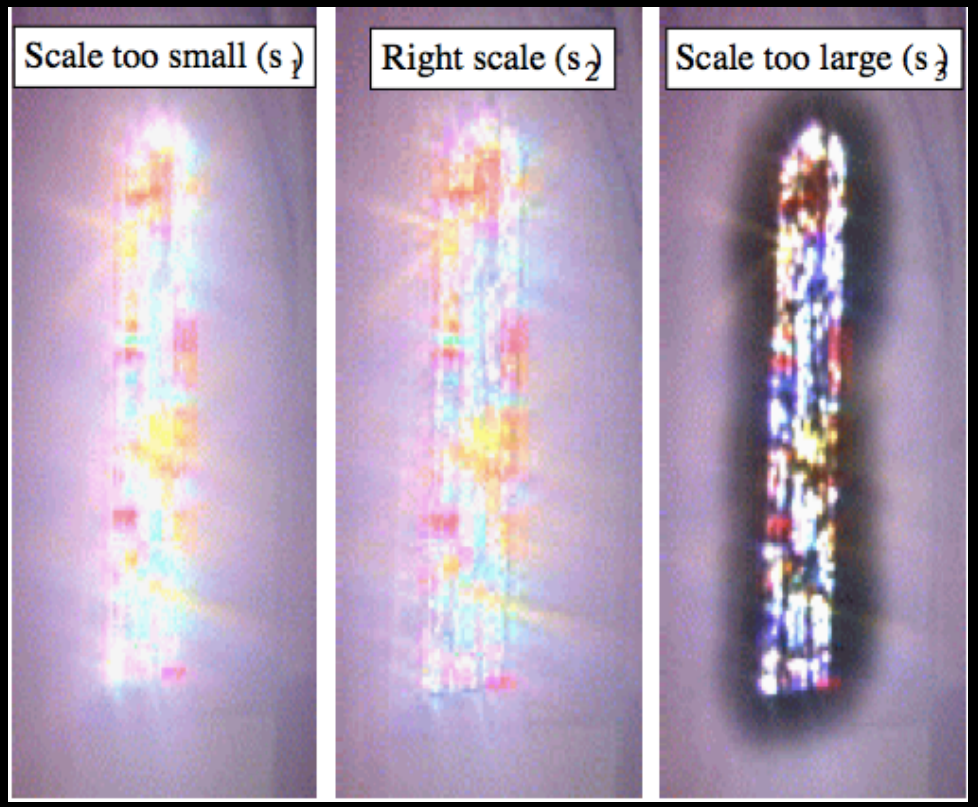


Local tonemapping

$$L_d(x,y) = \frac{L(x,y)}{1 + V_1(x,y,s_m(x,y))}$$

- V_1 = average of the "center"
  - dark pixel (L) on light (V_1)? Lowers L_d more, increased contrast
  - bright pixel (L) on dark (V_1)? Lowers L_d less, increased contrast
- Choose scale right to avoid halos





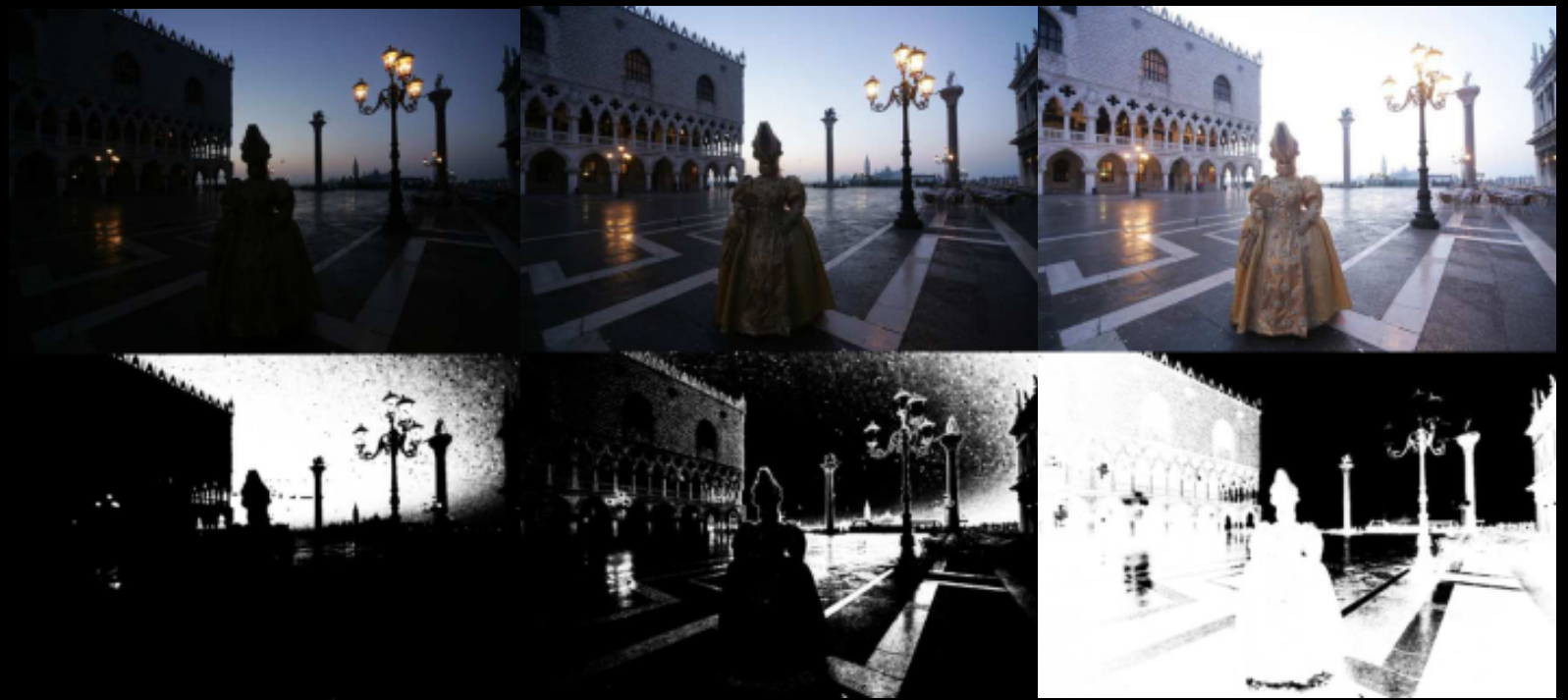
Reinhard operator

Darkest 0.1% scaled
to display device

Exposure Fusion: Simplified HDR

Mertens, Kautz, van Reeth PG 2007

- **Choose the best pixel from one of the images**
  - **Use heuristics for a smooth selection, such as**
    - **Exposure**
    - **Color saturation**
    - **Contrast**



LDR images

Weight maps

Weights from the paper

- **Contrast:** we apply a Laplacian filter to the grayscale version of each image, and take the absolute value of the filter response [16]. This yields a simple indicator C for contrast. It tends to assign a high weight to important elements such as edges and texture. A similar measure was used for multi-focus fusion for extended depth-of-field [19].

- **Saturation:** As a photograph undergoes a longer exposure, the resulting colors become desaturated and eventually clipped. Saturated colors are desirable and make the image look vivid. We include a saturation measure S, which is computed as the standard deviation within the R, G and B channel, at each pixel.

- **Well-exposedness:** Looking at just the raw intensities within a channel, reveals how well a pixel is exposed. We want to keep intensities that are not near zero (underexposed) or one (overexposed). We weight each intensity i based on how close it is to 0.5 using a Gauss curve: $\exp\left(-\frac{(i-0.5)^2}{2\sigma^2}\right)$, where σ equals 0.2 in our implementation. To account for multiple color channels, we apply the Gauss curve to each channel separately, and multiply the results, yielding the measure E.

Similar to weighted terms of a linear combination, we can control the influence of each measure using a power function:

$$W_{ij,k} = (C_{ij,k})^{\omega_C} \times (S_{ij,k})^{\omega_S} \times (E_{ij,k})^{\omega_E}$$

with C, S and E, being contrast, saturation, and well-exposedness, resp., and corresponding "weighting" exponents ω_C, ω_s, and ω_E.

To obtain a consistent result, we normalize the values of the N weight maps such that they sum to one at each pixel (i, j):

$$\hat{W}_{ij,k} = \left[\sum_{k'=1}^{N} W_{ij,k'}\right]^{-1} W_{ij,k}$$

Multi-resolution fusion