Dynamic range

The standard method to get maximum dynamic range:

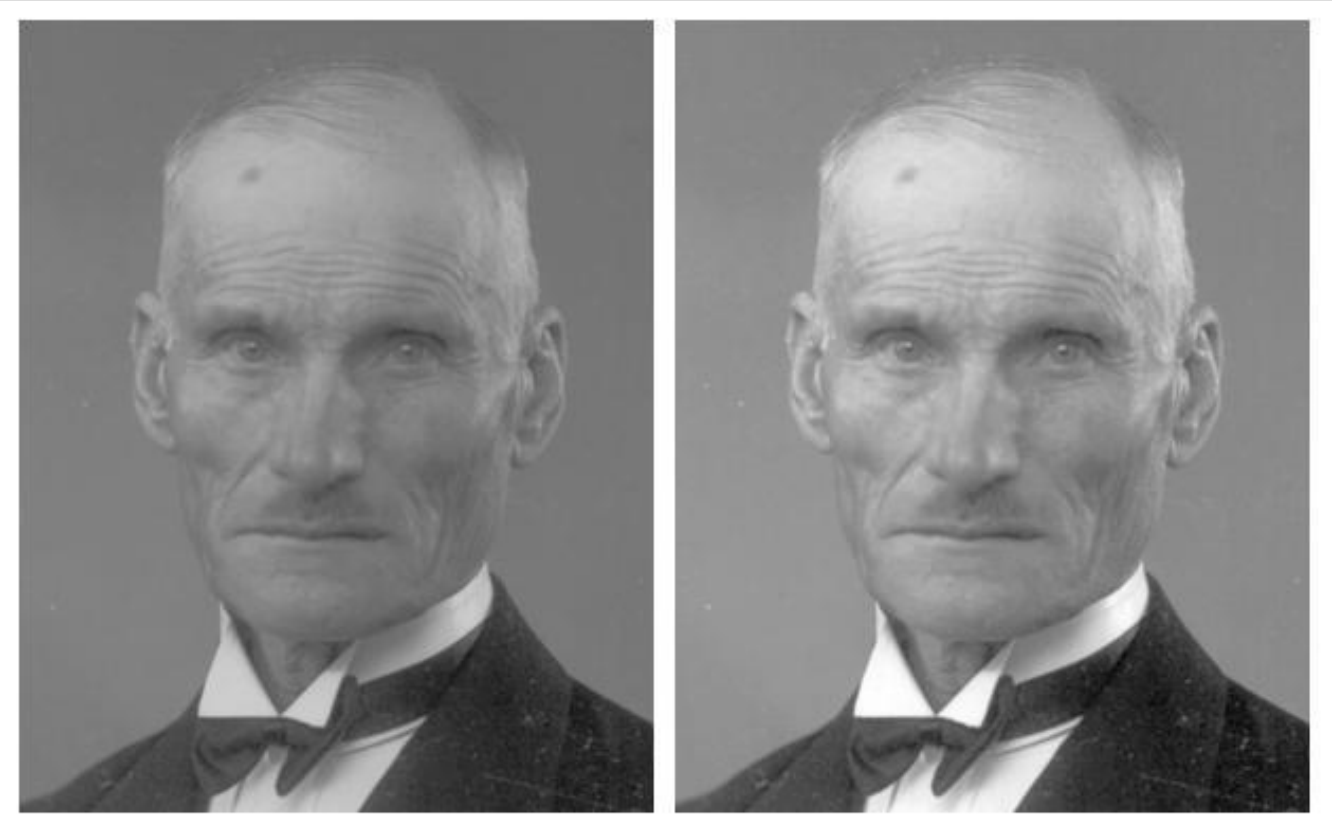
$$Q = \frac{P - P_{\min}}{P_{\max} - P_{\min}}$$

```
function Q=Autolevel(P)
%*********************************
Q=(P-min(P(:)))/(max(P(:))-min(P(:)));
return
end
```

AutoLevel.html

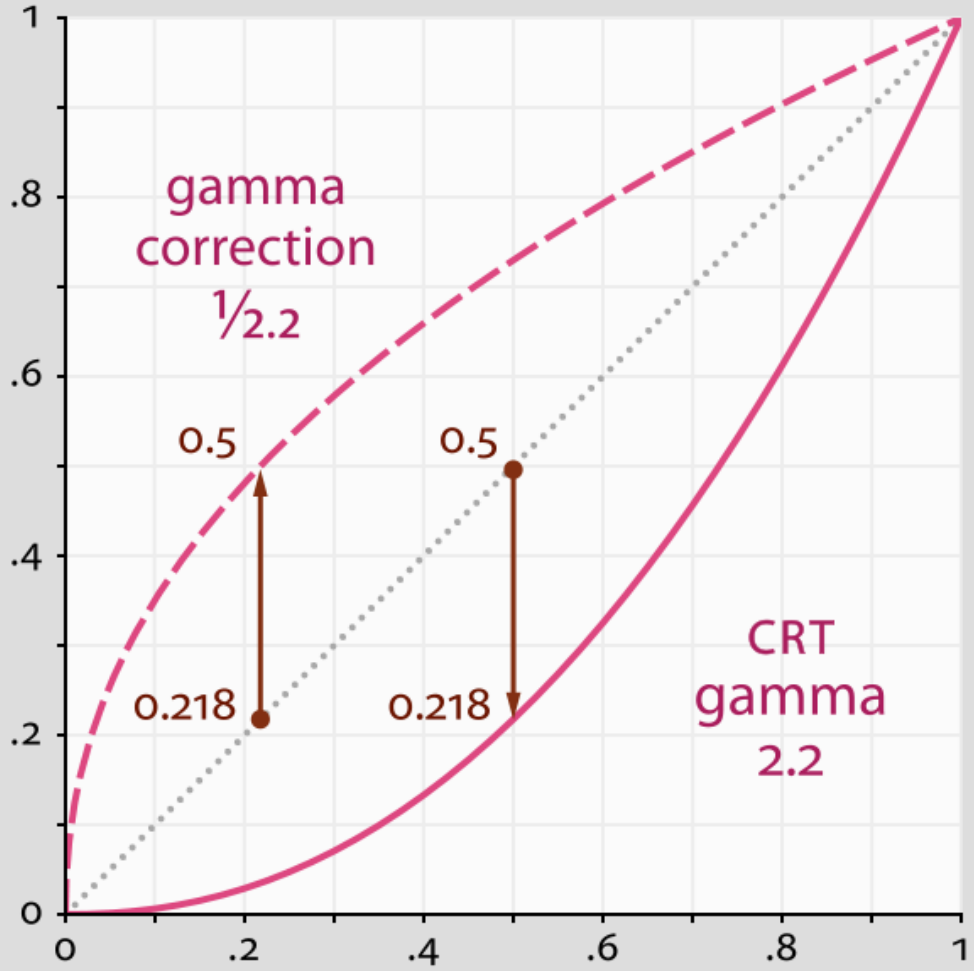
Dynamic range

AutoLevel example:



Matlab\html\AutoLevel.html

Gamma Correction



$$V_{out} = A\, V_{in}^{\gamma}$$

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

The Matlab command is imadjust: imadjust.m

Post-Processing

If you don’t use a raw-format as output then you apply

Demosaicing: Mapping 4 single sensor measurements to RGB

White-balancing: Combining different color channels

Gamma mapping

Compression: .jpg

Quality increasing left to right

White Point Correction

The pixel value at a point depends on

- The illumination
- The object
- The camera

We usually compensate the effects of illumination changes

A camera measures incoming light:

Different illuminations – different images

Von Kries

Assume I1 is the RGB image of a scene under illuminant 1
rewrite the image as a matrix of size (number of pixels x 3)

Assume I2 is the RGB image of the same scene under illuminant 2
rewrite the image as a matrix of size (number of pixels x 3)

What is the transformation

$$T: I1 \rightarrow I2?$$

A very good and simple model is a diagonal 3x3 matrix

$$T = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$$

This model is the von-Kries model

How can we find T?

Regression, backslash, pinv!

using the relation as a linear equation

$$I2 = I1*T$$

with T as unknown

In photo editing software like Photoshop
one can declare that a certain point should be white
the new pixel value is then given by the RGB vector (1,1,1)

In computer vision one often uses the assumption that the brightest point
(with the highest intensity) is white in the scene. If it has the RGB vector (r,g,b)
then the normalization is T = inv(diag(r,g,b))

Another assumption is that the average RGB vector
is in general a gray vector (c, c, c).
If the average vector in a given image is not gray one can normalize accordingly

T = inv(diag(mean(I1)))

Histograms, PDF, CDF in Matlab

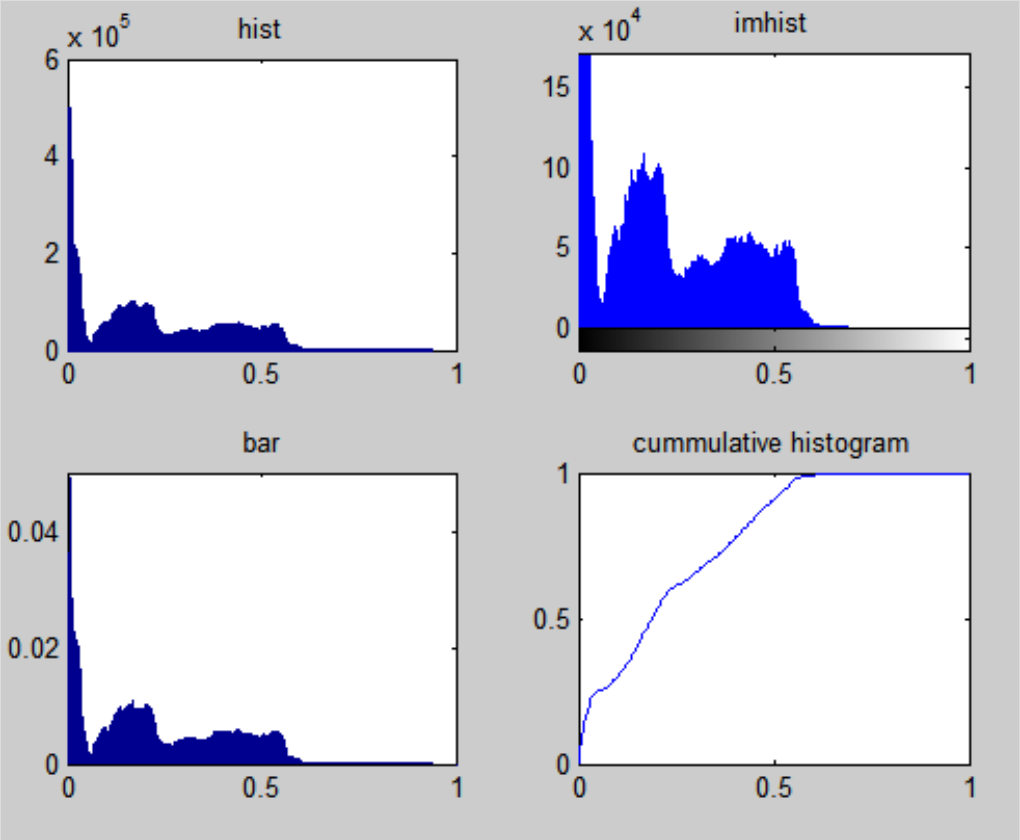
```
%% Read in HDR image

ima = hdrread('../images/kansai.hdr');
numel(unique(ima(:)))
2406
whos ima
ima       2602x3908x3          122023392  single
imshow(ima)
```

histpdfcdf.html

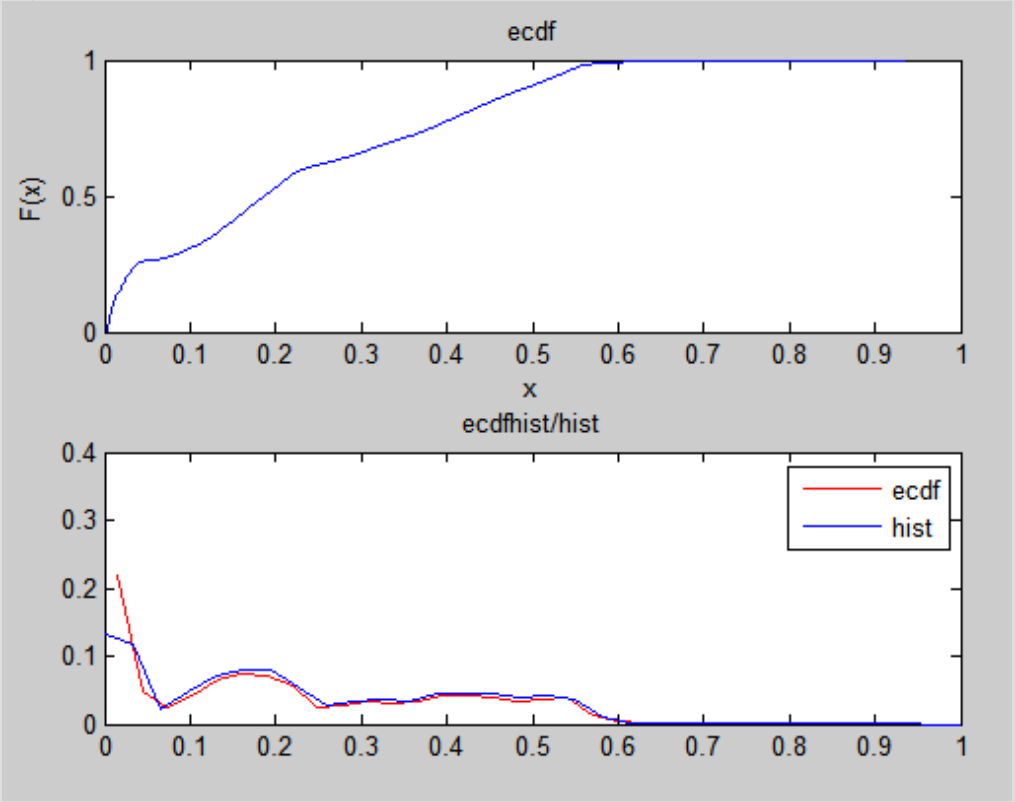
hist and imhist

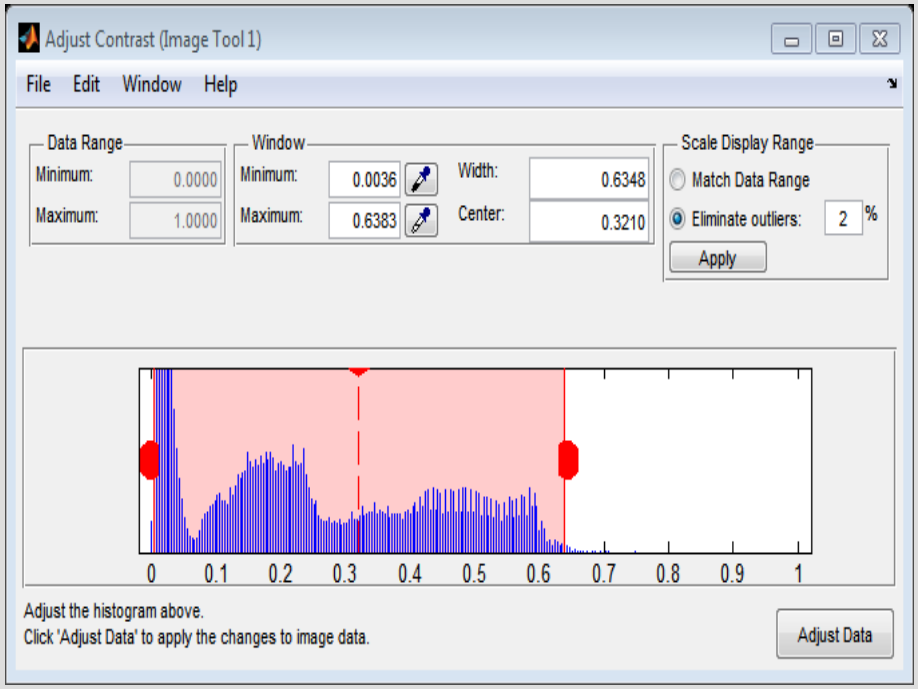
```
%% RGB 2 Gray - Normalization - imtool
gim = sum(double(ima),3);
gimn = gim/max(gim(:));
imtool(gimn)
grayim = rgb2gray(ima);
numel(unique(grayim(:)))
   1699181
imshow(grayim)
subplot(2,2,1),hist(grayim(:),256);
title('hist')
subplot(2,2,2),imhist(grayim,256)
title('imhist')
[c,x] = imhist(grayim,256);
subplot(2,2,3),bar(x,c/sum(c))
ax = axis;
ax(1)=x(1);ax(2)=x(end); axis(ax)
title('bar')
subplot(2,2,4),plot(x,cumsum(c)/sum(c))
title('cummulative histogram')
```

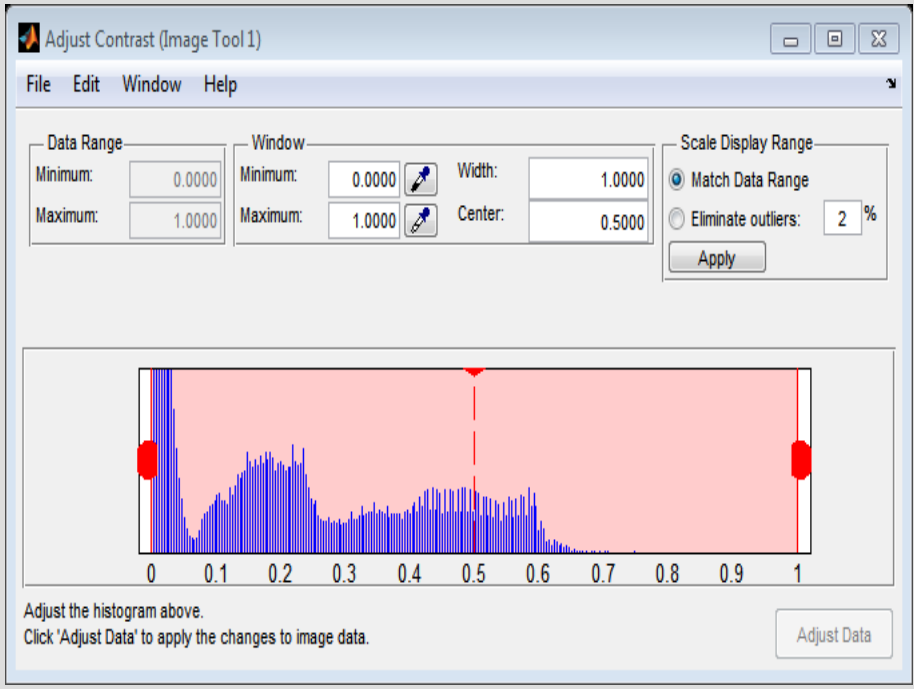


ecdf and ecdfhist

```
subplot(2,1,1),ecdf(grayim(:))
title('ecdf')
[f,x] = ecdf(grayim(:));
[c1,x1] = ecdfhist(f,x,32);
[c2,x2] = imhist(grayim,32);
subplot(2,1,2),plot(x1,c1/sum(c1),'r')
title('ecdfhist/hist')
hold on;subplot(2,1,2),plot(x2,c2/sum(c2),'b');
legend('ecdf','hist')
```



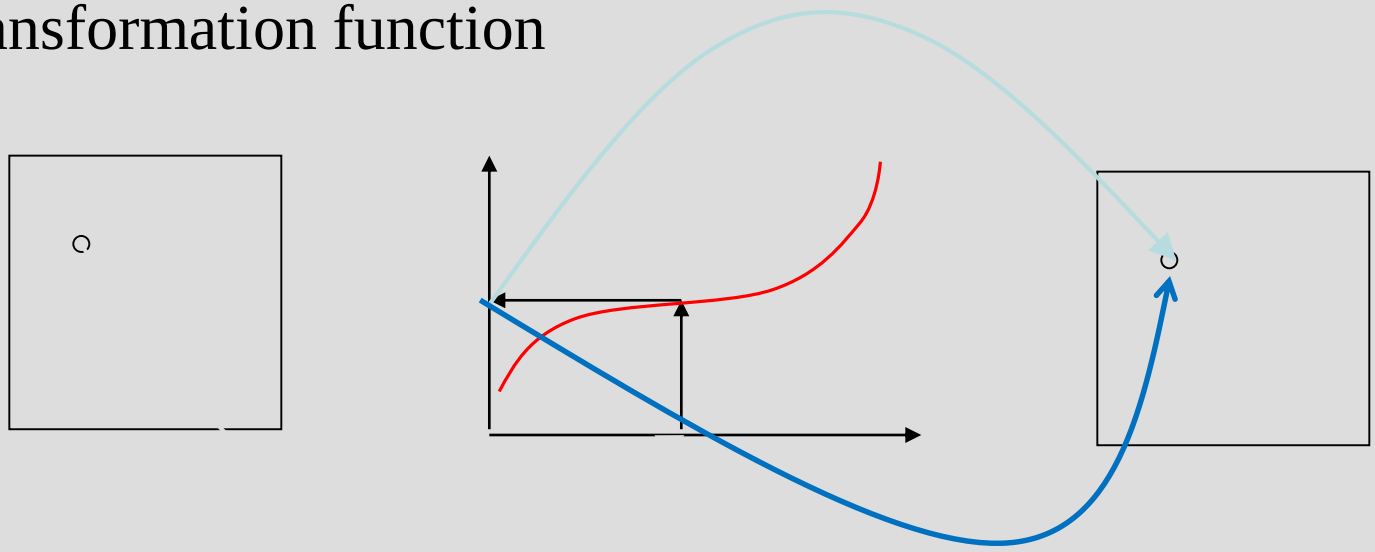

Adjust Contrast (Image Tool 1)
File Edit Window Help
Data Range
Minimum: 0.0000
Maximum: 1.0000
Window
Minimum: 0.0036
Maximum: 0.6383
Width: 0.6348
Center: 0.3210
Scale Display Range
Match Data Range
Eliminate outliers: 2 %
Apply
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Adjust the histogram above.
Click 'Adjust Data' to apply the changes to image data.
Adjust Data


Adjust Contrast (Image Tool 1)
File Edit Window Help
Data Range
Minimum: 0.0000
Maximum: 1.0000
Window
Minimum: 0.0000
Maximum: 1.0000
Width: 1.0000
Center: 0.5000
Scale Display Range
Match Data Range
Eliminate outliers: 2 %
Apply
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Adjust the histogram above.
Click 'Adjust Data' to apply the changes to image data.
Adjust Data

Histogram Equalization

One way of modifying the intensity values is *equalizing*. It aims at an image with an even histogram - each intensity value should ideally have the same number of pixels.

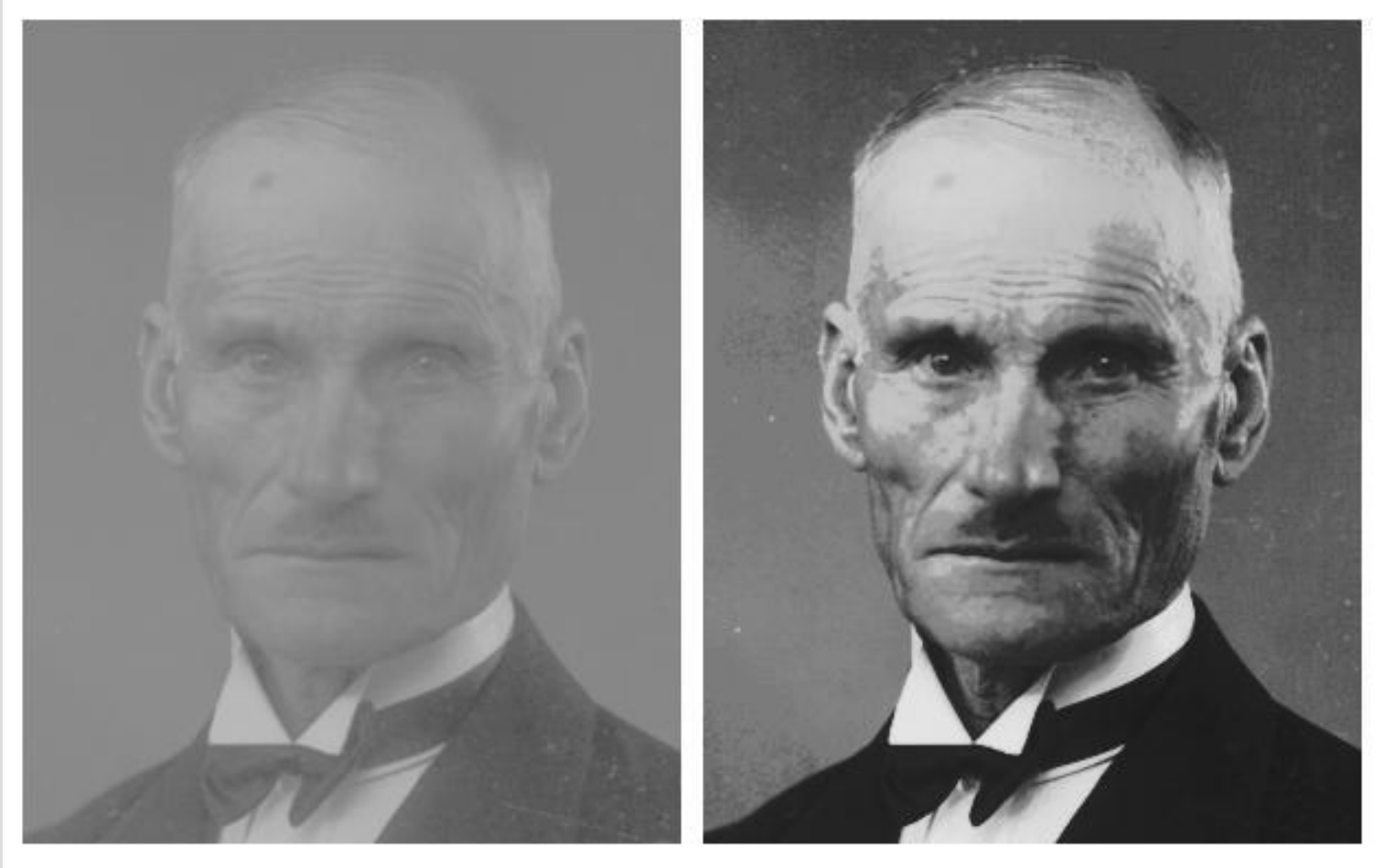
Calculate the histogram, create the cumulative histogram normalized with the total number of pixels and use it as a transformation function



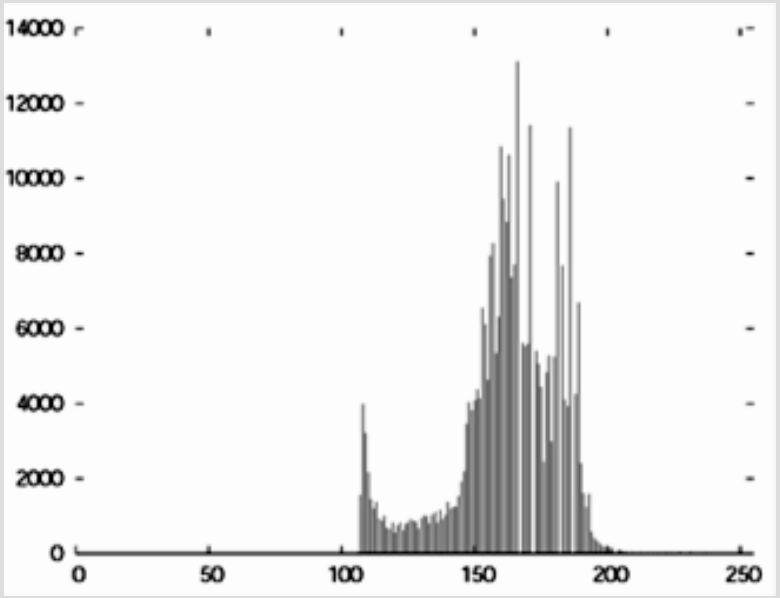
Original f(x,y) Cum. hist. P(f) Result g(x,y)

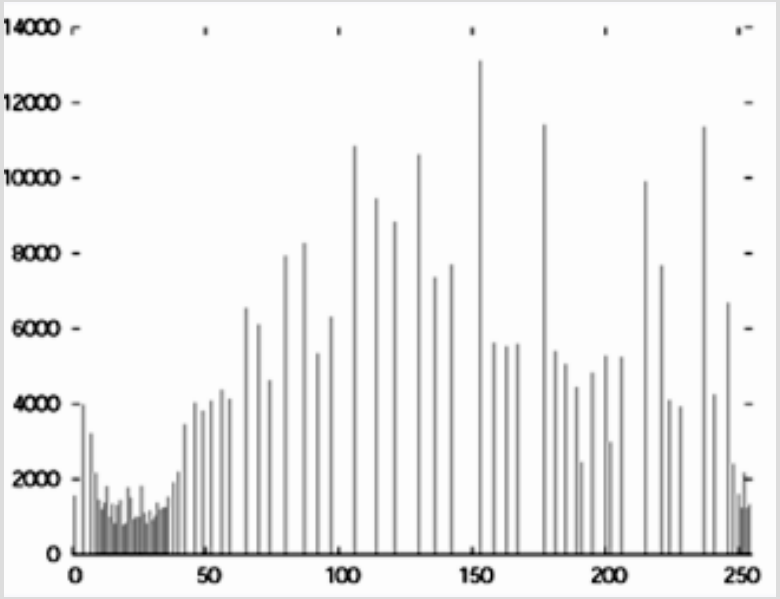
Histogram Equalization

Example:



Histogram equalization





Histogram equalization and truncation

```
ima = hdrread('../Images/kansai.hdr');
grayim = double(rgb2gray(ima));
ngray = min(double(grayim),quantile(grayim(:),0.95));
ngray=ngray/max(ngray(:));
imshow(ngray)
figure(2);imshow(histeq(ngray,256),[])
```

without histeq

with histeq

truncandhisteq.html

Pointwise operations

It is easy to add/subtract/multiply/divide/compare images pixelwise

But don’t forget the types

Gamma mapping is done by imadjust

and

several images can be viewed side-by-side
with montage and imshowpair

Demo

```
dima=double(ima)/max(double(ima(:)));
mim=dima;
mim = cat(3,mim,histeq(dima,16));
mim=cat(3,mim,dima./histeq(dima,16));
mm = dima;
mm(mm<0.35)=255;
mim = cat(3,mim,mm);
mim = cat(3,mim,imadjust(dima,[0;1],[0;1],2));
mim = cat(3,mim,imadjust(dima,[0;1],[0;1],0.5));
mim = reshape(mim,size(mim,1),size(mim,2),1,size(mim,3));
montage(mim,'Size',[2,3])
```

PointwiseMappings.html

Summary

1) Basic difference between global and rolling shutter
2) Resizing and interpolation methods
3) (In)homogeneous coordinates and geometrical transformations
4) Distance measures and distancemap
5) Intensity transforms and tonemapping
6) Whitepoint correction
7) PDF, CDF, histogram equalization
8) MATLAB programming

HDR – High Dynamic Range

Reiner Lenz

2015

Illumination Levels

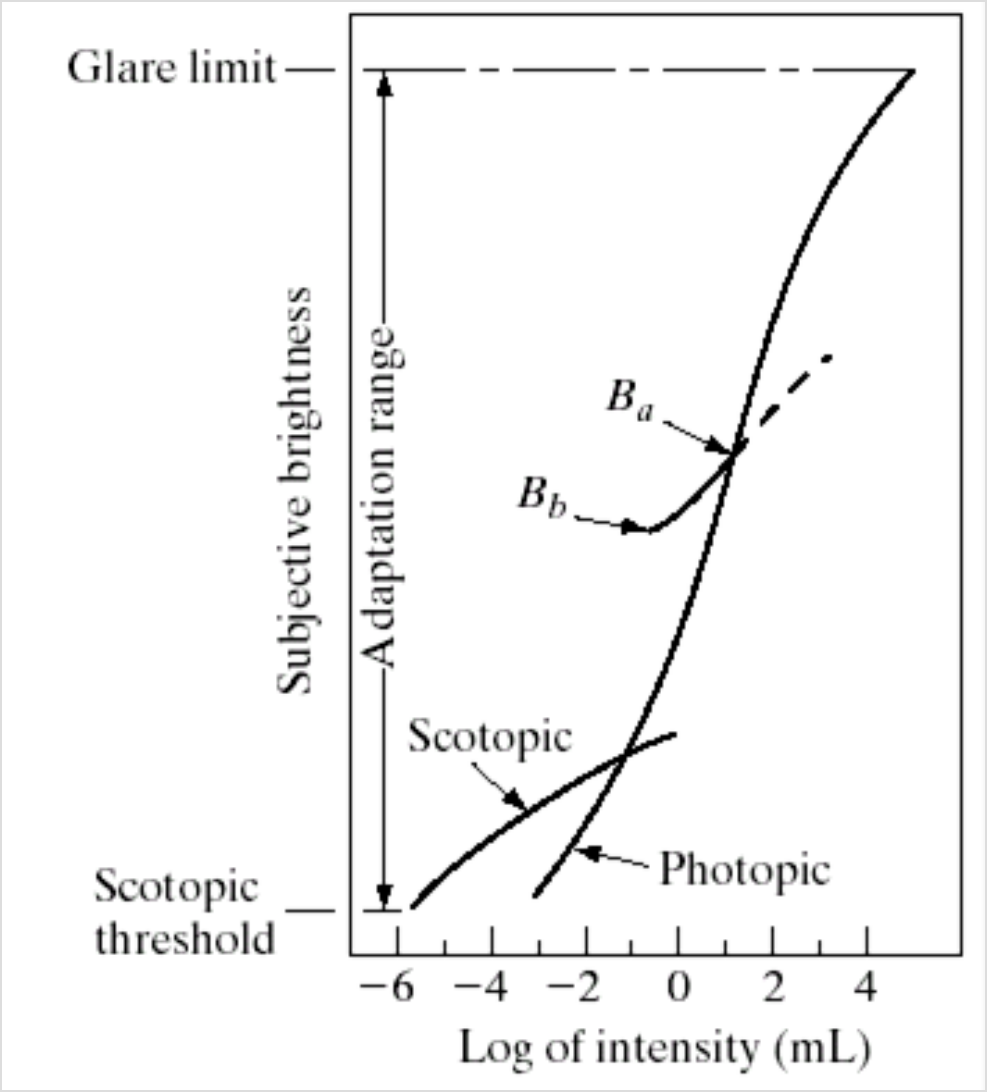
Starlight	10^{-3}
Moonlight	10^{-1}
Indoor Lighting	10^{2}
Sunlight	10^{5}

Wandell – cd/m^2

Dynamic Range: $10^5/10^{-3} = 10^8 = 100\ 000\ 000 = 2^{23}$

Jpg-Image 0 ... 255

Sensitivity of the Eye



Scotopic: rods (dim light vision)

Photopic: cones (color vision)

Range of intensity levels: 10^{10} with adaptation

Only a few at given Adaptation level

Examples

Terminology

Illumination:

Irradiance is the power of electromagnetic radiation per unit area incident on a surface.

Reflection:

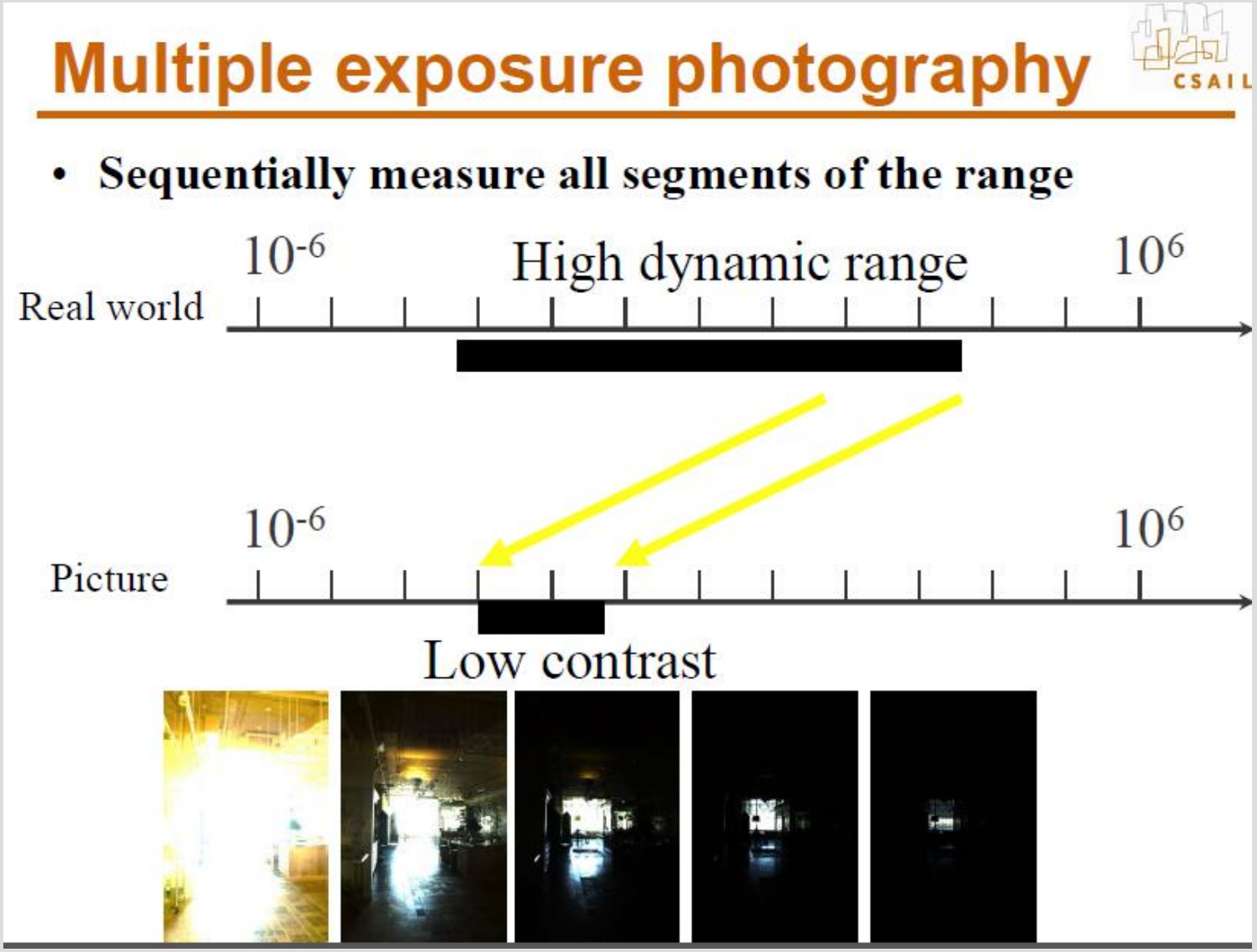
Radiance are measures of the quantity of radiation that is emitted from a surface and falls within a given solid angle in a specified direction

Exposure:

Exposure is the amount of light allowed to fall on each area unit of a photographic

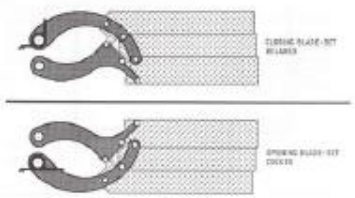
sensor input = reflection * illumination

Sequential Collection of Image



How do we vary exposure?

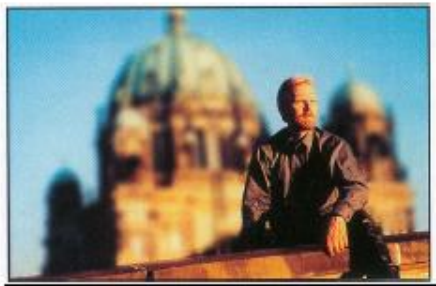
- **Options:**
  - Shutter speed
  - Aperture
  - ISO
  - Neutral density filter

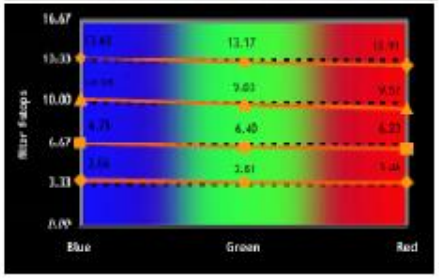


Slide inspired by Siggraph 2005 course on HDR

Tradeoffs

- **Shutter speed**
  - Range: ~30 sec to 1/4000sec (6 orders of magnitude)
  - Pros: reliable, linear
  - Cons: sometimes noise for long exposure
- **Aperture**
  - Range: ~f/1.4 to f/22 (2.5 orders of magnitude)
  - Cons: changes depth of field
  - Useful when desperate
- **ISO**
  - Range: ~100 to 1600 (1.5 orders of magnitude)
  - Cons: noise
  - Useful when desperate
- **Neutral density filter**
  - Range: up to 4 densities (4 orders of magnitude) & can be stacked
  - Cons: not perfectly neutral (color shift), not very precise, need to touch camera (shake)
  - Pros: works with strobe/flash, good complement when desperate



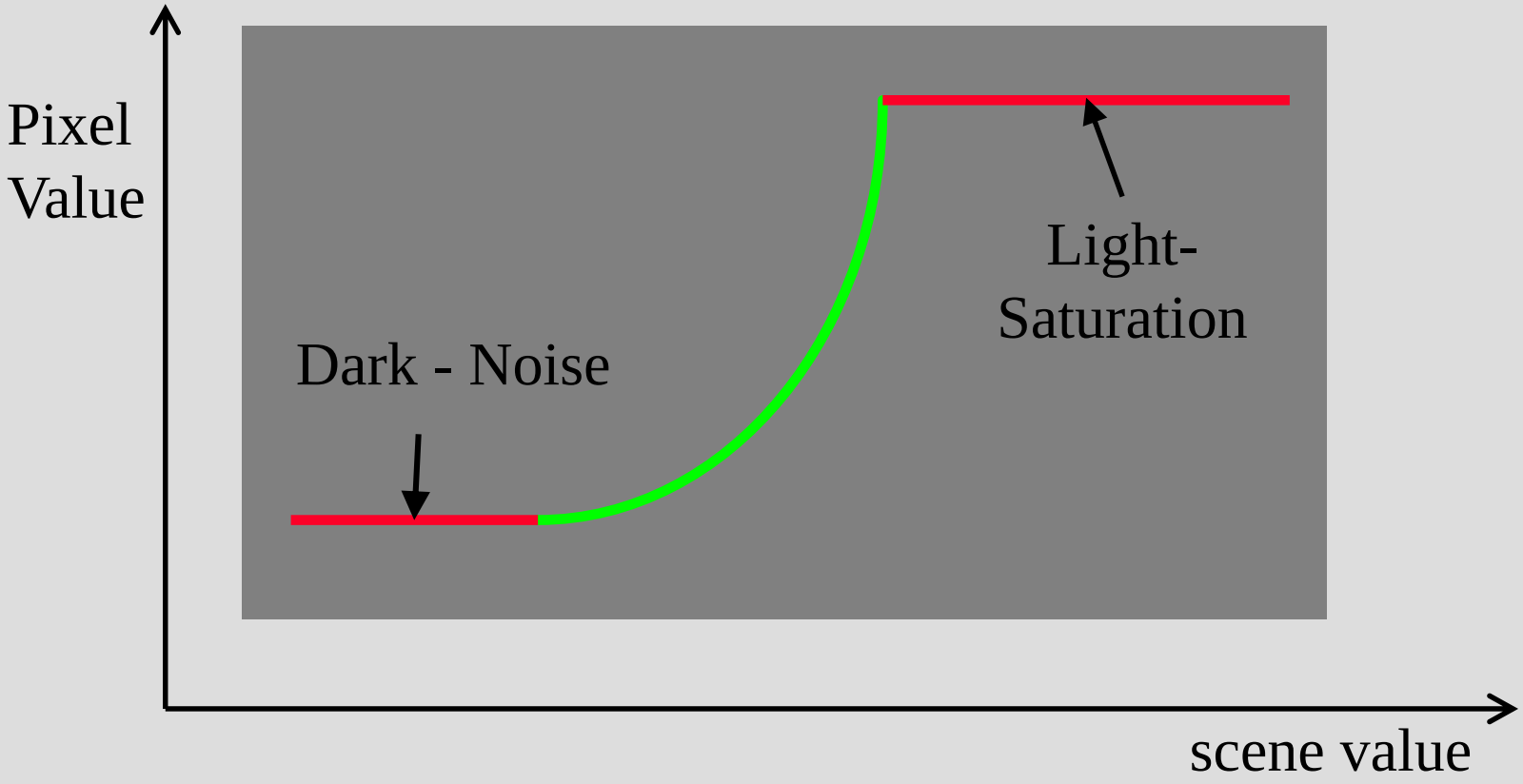


Slide after Siggraph 2005 course on HDR

Response Curve

The relation between the radiance and the pixel values is

- not linear and
- not completely known



Calibrating the response curve

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

Basic Strategy

1) Estimate the radiometric response function from aligned images

2) Estimate a radiance map by selecting or blending pixels from different expositions

3) Tone map the resulting HDR image back into the displayable gamut

Estimate Camera Response Curve

$$Z_{ij} = f(E_i \Delta t_j) \tag{1}$$

Since we assume f is monotonic, it is invertible, and we can rewrite (1) as:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

Taking the natural logarithm of both sides, we have:

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

To simplify notation, let us define function $g = \ln f^{-1}$. We then have the set of equations:

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \tag{2}$$

where i ranges over pixels and j ranges over exposure durations. In this set of equations, the Z_{ij} are known, as are the Δt_j. The unknowns are the irradiances E_i, as well as the function g, although we assume that g is smooth and monotonic.

Debevec – Malik Siggraph 97

Response Curve

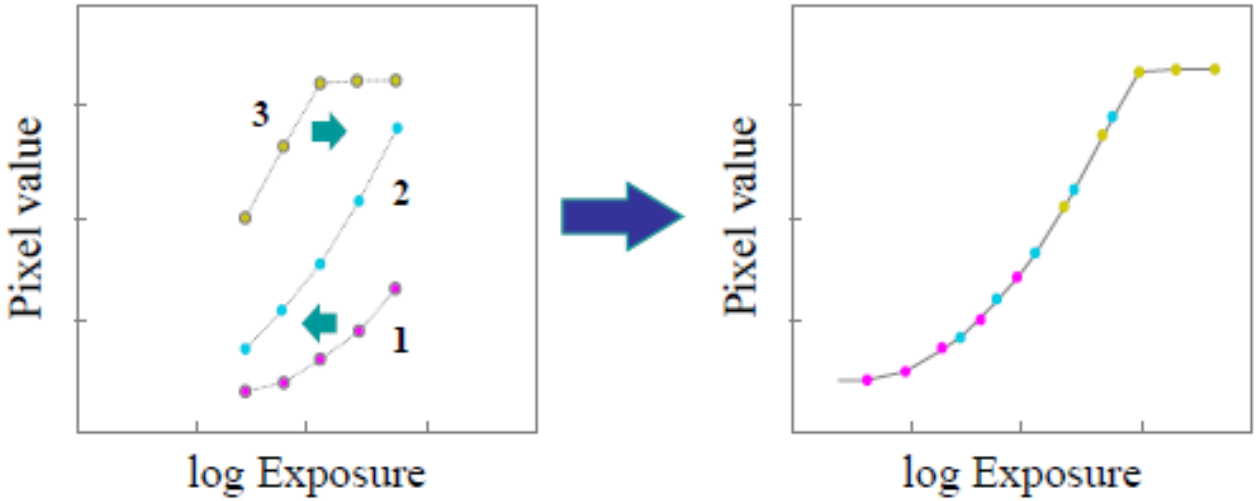


Figure 10.13 Radiometric calibration using multiple exposures (Debevec and Malik 1997). Corresponding pixel values are plotted as functions of log exposures (irradiance). The curves on the left are shifted to account for each pixel's unknown radiance until they all line up into a single smooth curve.