Computational Photography

Siavash Bigdeli
University of Bern
Fall 2014

Deadline

Deadline:

October 16

before class (14:00)

Project 2

- Topics
  - High dynamic range images
  - Bilateral filter
- Assignments
  - Capturing HDR images
  - Tone mapping HDR images using the bilateral filter
  - Tone adjustment using the bilateral filter

Capturing HDR images

- Use HDRShop (version 1 is free) http://www.hdrshop.com/
- Camera can be borrowed

1. Measure response curve (HDRShop)
2. Capture images at different exposures
3. Assemble HDR image (HDRShop)

Tone mapping HDR images

- Implement bilateral filter in Matlab
- Only two nested loops, not four!

$$\frac{1}{k(x)} \sum_{\xi} \left[h(\xi - x) \, d(f(\xi) - f(x)) \, f(\xi) \right]$$

Gaussian filter

- 1D filter for the range weight d
  - Note normalization to unit integral

$$d(t) = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{t^2}{2\sigma_r^2}}$$

Gaussian filter

- 2D Gaussian for distance weight h
  - Standard deviation σ_s^2, variance σ_s
  - Note normalization to unit integral

$$h(t_1, t_2) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{t_1^2+t_2^2}{2\sigma_s^2}}$$

- In our case t_1, t_2 are the coordinates of the vector $\xi - x$. As a formula:

$$\xi - x = \vec{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

Gaussian filter

- Matlab evaluates exponential per matrix element:

```
>> A = [0 1; 2 3]
A =
    0     1
    2     3

>> exp(A)
ans =
    1.0000    2.7183
    7.3891   20.0855
```

Gaussian filter

$$h(t_1, t_2) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{t_1^2+t_2^2}{2\sigma_s^2}}$$

```
>> t1 = repmat([-2:2], 5, 1)

t1 =

    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2

>> t2 = repmat([-2:2]', 1, 5)

t2 =

    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1
     0     0     0     0     0
     1     1     1     1     1
     2     2     2     2     2
```

Gaussian filter

$$h(t_1, t_2) = \frac{1}{2\pi\sigma_s^2} \boxed{e^{-\frac{t_1^2+t_2^2}{2\sigma_s^2}}} \qquad \text{with } \sigma_s = 1$$

```
>> exp(-0.5*(t1.^2+t2.^2))

ans =

    0.0183    0.0821    0.1353    0.0821    0.0183
    0.0821    0.3679    0.6065    0.3679    0.0821
    0.1353    0.6065    1.0000    0.6065    0.1353
    0.0821    0.3679    0.6065    0.3679    0.0821
    0.0183    0.0821    0.1353    0.0821    0.0183
```

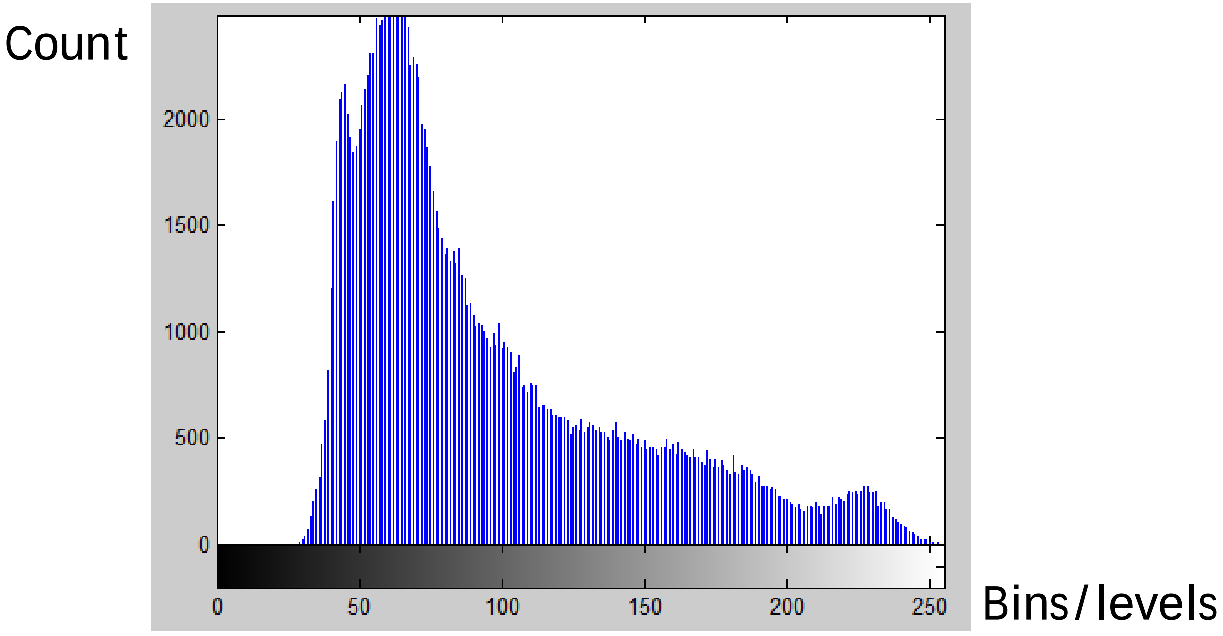
No loop required!

Tone adjustment

- Histograms
- Histogram matching
- Two-scale tone adjustment procedure

Image histogram

- Count number of pixels for each intensity level
  - Discrete number of levels (bins)
  - Separate histograms for color channels
- Normalized histogram
  - Divide value of each bin by total number of pixels
  - Discrete probability distribution for pixel values



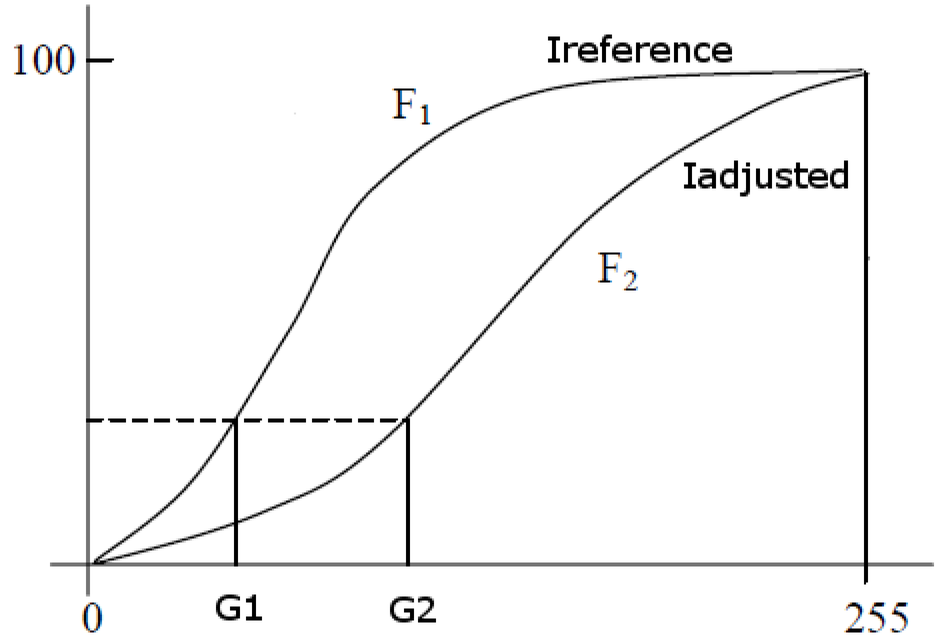
Histogram of red channel

Histogram matching & equalization

- Histogram matching
  - Given a desired histogram
  - Map each value of an image channel to a new value, such that the new histogram matches the desired histogram
- Histogram equalization
  - The desired histogram is constant

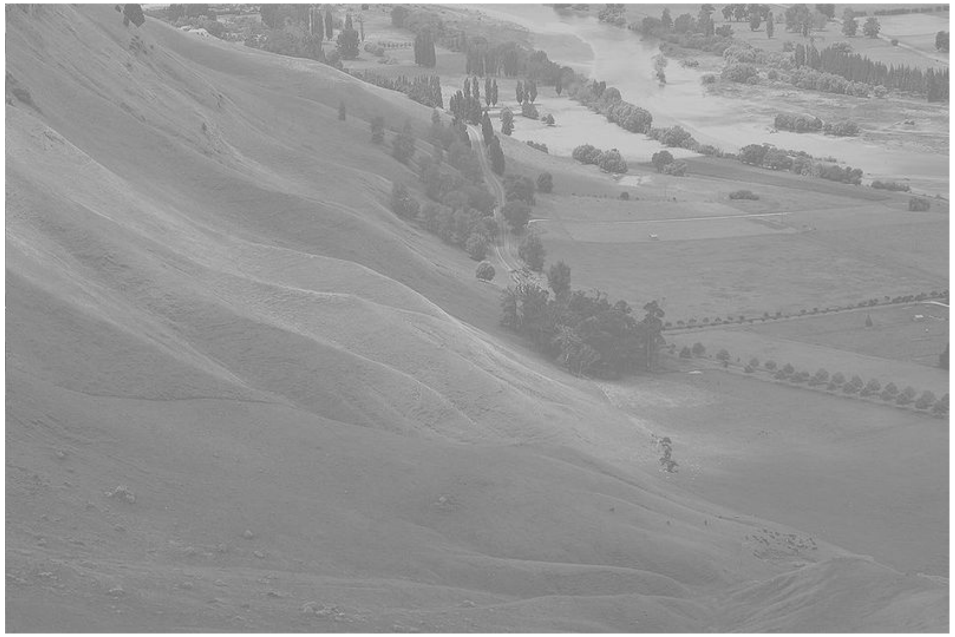
Histogram matching

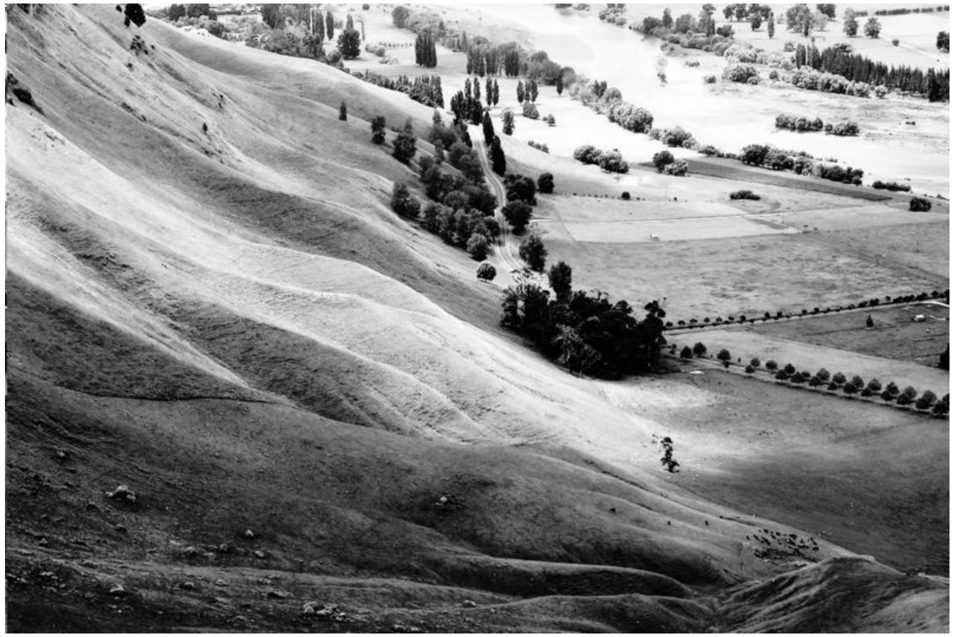
- Histogram matching is done by adjusting the cumulative distribution function (cdf)

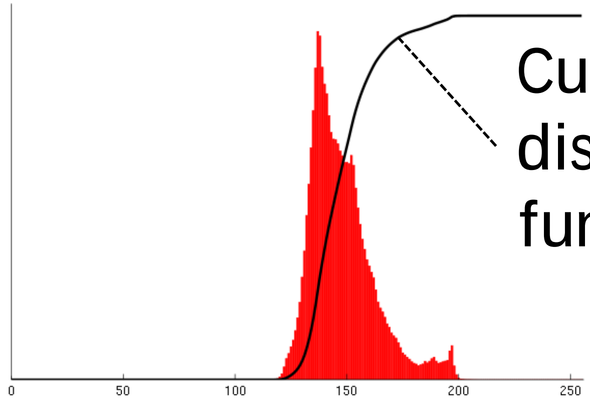


- Pixels G2 get intensitiy of Pixels G1

Histogram equalization

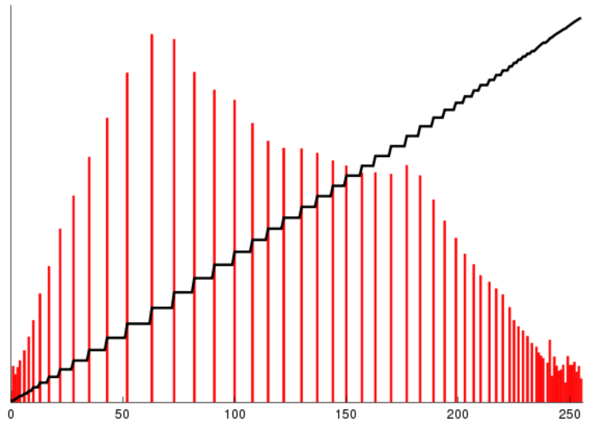






Cumulative distribution function (CDF)

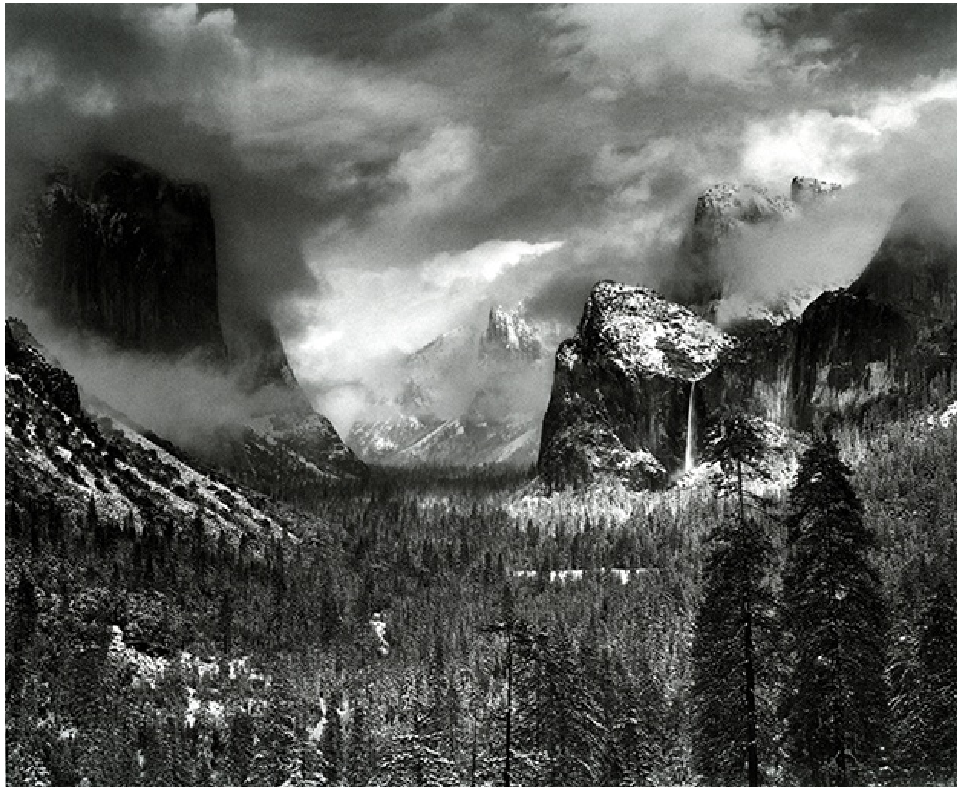
Input

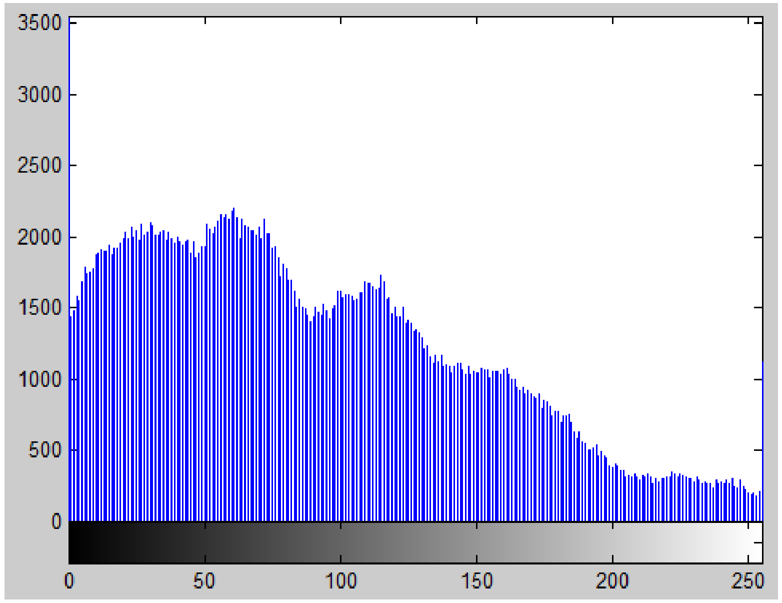


Equalized

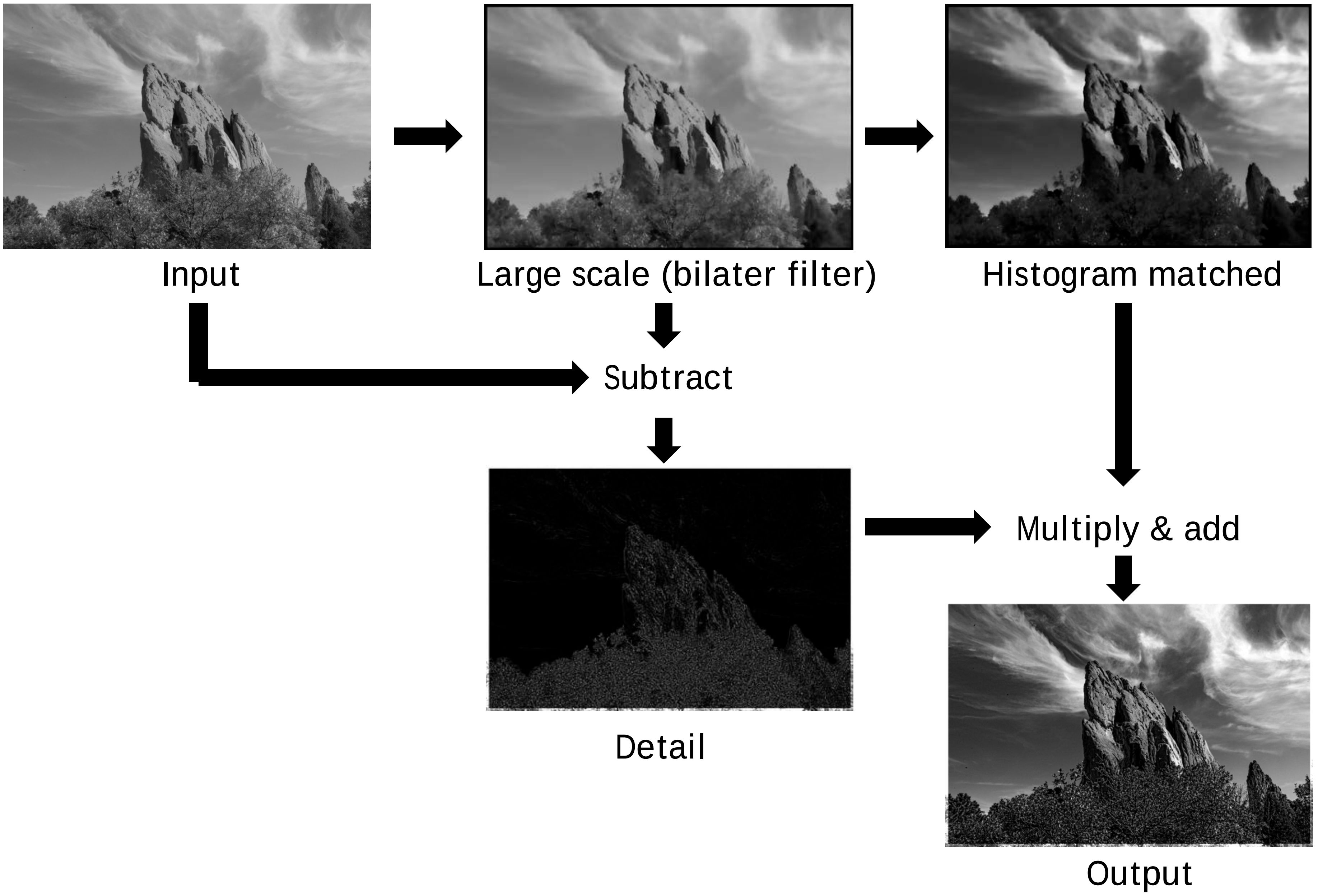
Histogram matching

- Match histogram to an „interesting“ model image


Winterstorm (Ansel Adams)

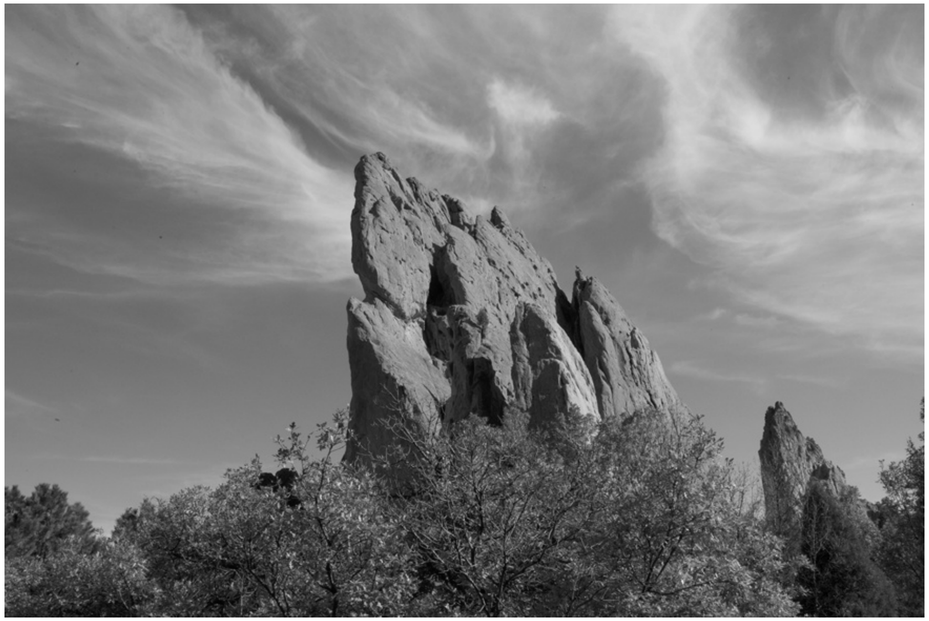


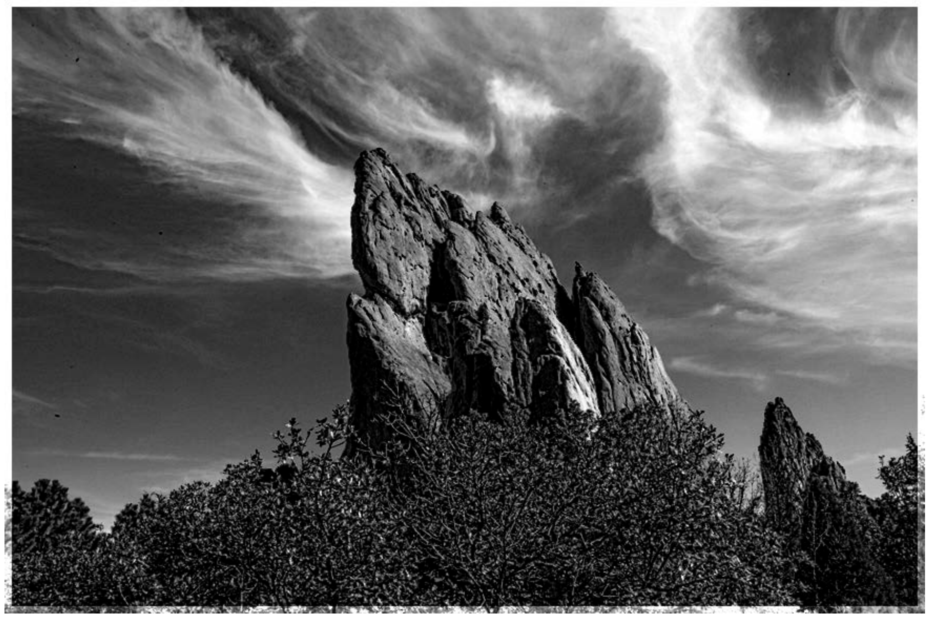
Two-scale tone adjustment



Two-scale tone adjustment

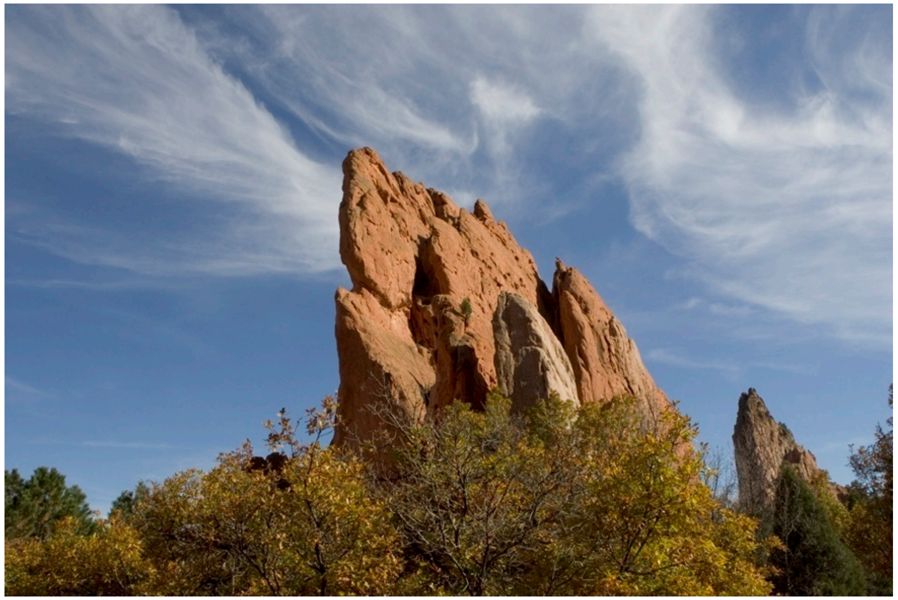
- Get desired large scale contrast
- Emphasize detail


Input

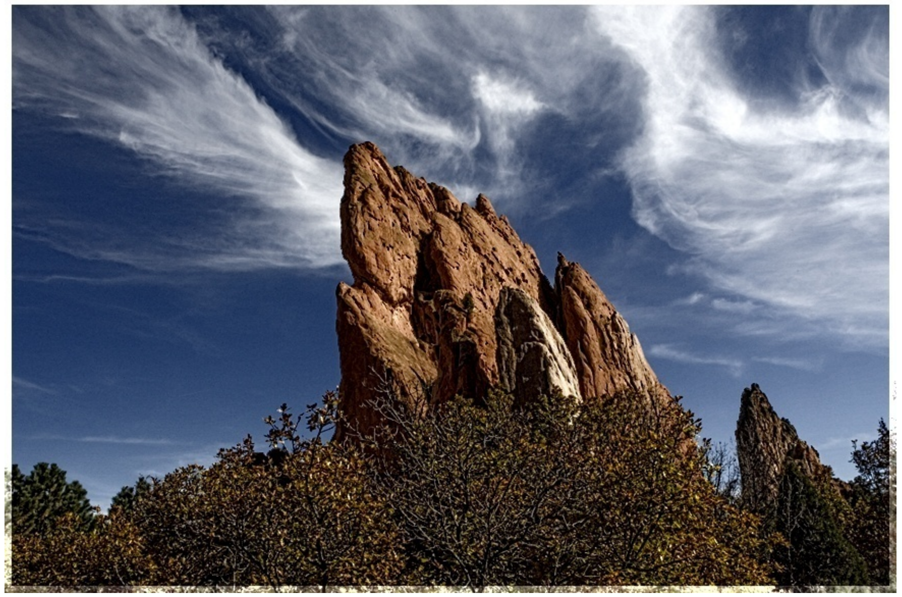

Output

Color images

- Transform to YUV, work on Y, transform back



Input



Output