Digital Negative

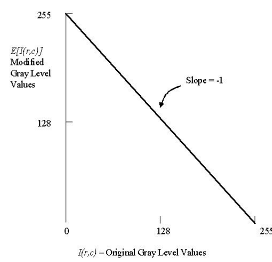
- A **digital negative** can be created with mapping equation (MAX: Maximum grayscale value) as :

$$M[I(r,c)] = MAX - I(r,c)$$

•Equivalent of performing a logical NOT on the input image

•Enables to see details characterized by small brightness changes in the bright regions which may not be otherwise visible

Original image



Inverse Mapping Equation

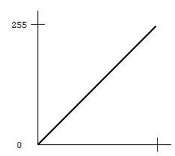
Inverse mapping equation

Caiani 23/03/2012

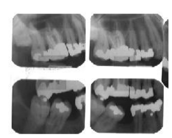
Intensity level slicing

- This is a particular kind of gray level mapping where specific gray level values of interest are mapped to a specified (typically high/bright) value
- Used for feature extraction

Original image

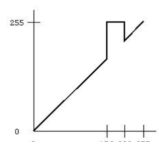


Mapping Function



Mapped image

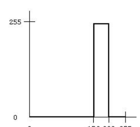
Original image



Mapping Function

Mapped image

Original image



Mapping Function

Mapped image

Caiani 23/03/2012

Non linear mapping equations

Range compression

- Logarithmic function that we use to display spectral images
- Useful when the dynamic range of the input data is very large

Power law

- Mapping equation: $$E(r,c) = M[I(r,c)] = K_1[I(r,c)]^{\gamma}$$ $$\text{where } K_1 \text{ and } \gamma \text{ are positive constants}$$

  Imaging equipment, such as cameras, displays and printers typically react according to the above equation
- A device with a response of the power-law transform, can be compensated for by application of a ***gamma-correction equation*** of the following form (important for proper display of images, whether on a computer monitor or on a printed page):

$$E(r,c) = M[I(r,c)] = K_2[I(r,c)]^{1/\gamma}$$

$$\text{where } K_2 \text{ and } \gamma \text{ are positive constants}$$

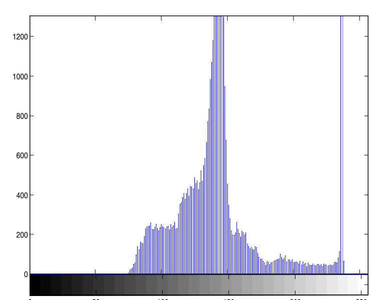
Caiani 23/03/2012

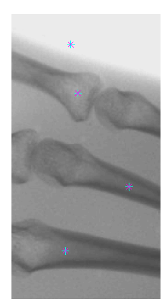
Intensity Transformation Functions in Matlab

First step is to compute the image histogram, then to observe the level distribution.

In this example, there are no pixels below 72 and over 238.

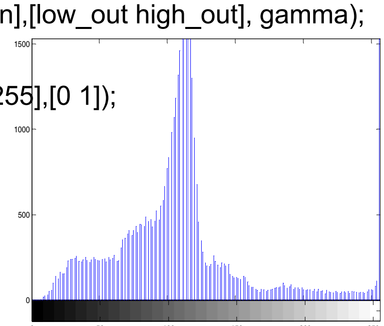
If we remap the image in order to utilize all the 256 levels, contrast would be enhanced.

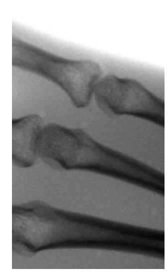




```
>>B=imadjust(I,[low_in high_in],[low_out high_out], gamma);
```

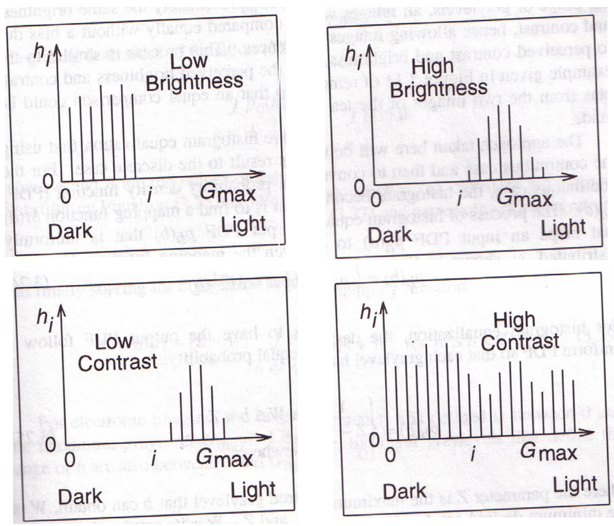
```
>>B=imadjust(I,[72/255 238/255],[0 1]);
```



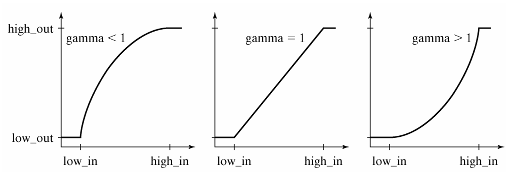


Caiani 23/03/2012

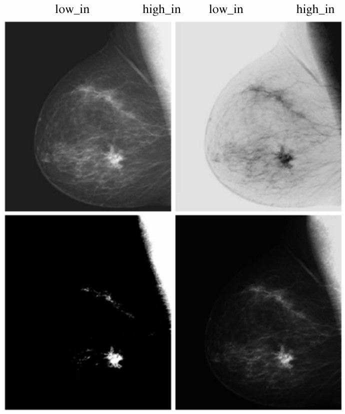
IMAGE HISTOGRAM



Caiani 23/03/2012



Histogram transformation can be non linear if gamma is not 1 (default).



a b
c d

FIGURE 3.3 (a) Original digital mammogram. (b) Negative image. (c) Result of expanding the intensity range [0.5, 0.75]. (d) Result of enhancing the image with gamma = 2. (Original image courtesy of G. E. Medical Systems.)

Apply a transformation to obtain the negative of the image.

Try to apply the described commands in one image of choice.
Then, load 'disco1.jpg' or 'disco2.jpg', and define the better transformation to enhance image visualization.

Caiani 23/03/2012

It can be useful to use **imadjust** automatically, without setting the parameters. It is possible to use **stretchlim** to determine them, optimized for contrast stretching:

Low_High = **stretchlim** (f)

>>B=**imadjust**(f, **stretchlim** (f),[]);

A more complex sintax is:

Low_High = **stretchlim** (f,tol) where tol=[low_frac high_frac]

that user-defines the fraction of the image to saturate at low and high pixel values (default [.01 .99], tol=0 to avoid saturation).

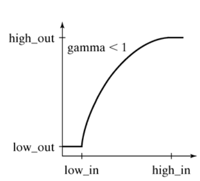
Try to apply the described commands in 'disco1.jpg', entire image and selected ROI.

Logarithmic transformations

They are basic tools for dynamic range manipulation.

Logarithmic transformations are used to compress dynamic range, and are implemented by:

>>g=c***log**(1+double(f)); where c is a constant and f is the image



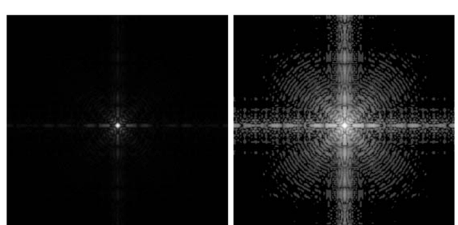
The shape is similar to that,with low=0 and high=1

but fixed and not depending on gamma value.

It is desirable to bring the resulting compressed values back to full range of the display:

>>gs=**im2uint8**(**mat2gray**(g));

Try these commands on an image, comparing with the results obtained using gamma<1.

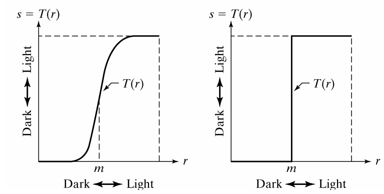


a b

FIGURE 3.5 (a) A Fourier spectrum. (b) Result obtained by performing a log transformation.

Contrast-stretching transformations

It expands a narrow range of input levels into a wide range of output levels, resulting in an image of higher contrast (limiting case, binary image).



a b

FIGURE 3.4 (a) Contrast-stretching transformation. (b) Thresholding transformation.

$$s = T(r) = \frac{1}{1 + \left(\frac{m}{r}\right)^E}$$

where E controls the slope

```
>>g=1./(1+(m./f).^E);
```

Try these commands on an image, for different values of m and E.

Caiani 23/03/2012

Gray Scale Modification by Histogram Modification

- Histogram modification performs a function similar to gray level mapping, but works by **considering histogram's shape and spread**
- Gray level histogram of an image is the distribution of the gray levels in an image
- The histogram can be modified by a mapping function, which will stretch, shrink (compress), or slide the histogram
- Histogram stretching and histogram shrinking are forms of gray scale modification, sometimes referred to as **histogram scaling**

Caiani 23/03/2012

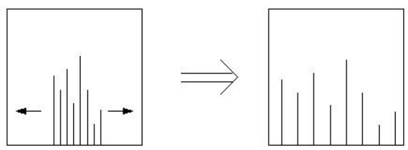
Histogram Modification

- The mapping function equation:

$$Stretching(I(r,c)) = MIN + \left[\frac{I(r,c) - I(r,c)_{MIN}}{I(r,c)_{MAX} - I(r,c)_{MIN}} \right] [MAX - MIN]$$

  - $I(r,c)_{MAX}$ is the largest gray level value in the image I(r,c)
  - $I(r,c)_{MIN}$ is the smallest gray level value in I(r,c)
  - MAX and MIN correspond to the maximum and minimum gray level values possible (for an 8-bit image these are 0 and 255)

Low contrast



High contrast

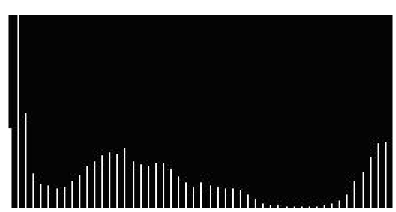
Caiani 23/03/2012

Histogram Stretching

Low-contrast image

Histogram

Image after histogram stretching



Stretched Histogram

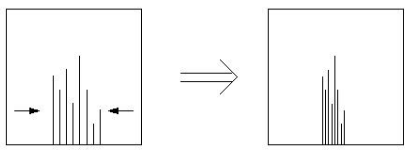
Caiani 23/03/2012

Histogram Shrinking

- The mapping function equation:

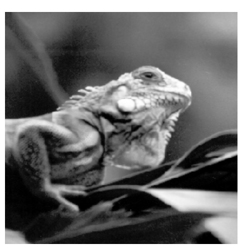
$$Shrinking(I(r,c)) = Sh_{MIN} + \left[\frac{Sh_{MAX} - Sh_{MIN}}{I(r,c)_{MAX} - I(r,c)_{MIN}}\right][I(r,c) - I(r,c)_{MIN}]$$

- $I(r,c)_{MAX}$ is the largest gray level value in the image I(r,c)
- $I(r,c)_{MIN}$ is the smallest gray level value in I(r,c)
- Sh_{MAX} and Sh_{MIN} correspond to the maximum and minimum desired in the compressed histogram
- Decreases image contrast by compressing the gray levels

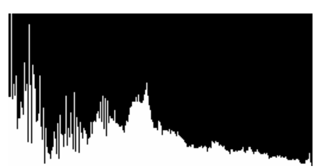


Caiani 23/03/2012

Histogram Shrinking



Original image



Histogram of image

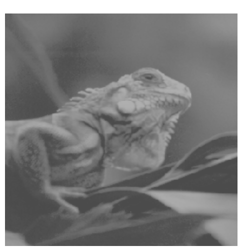
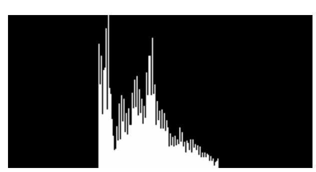


Image after shrinking the histogram to the range [75,175]



Histogram after Shrinking

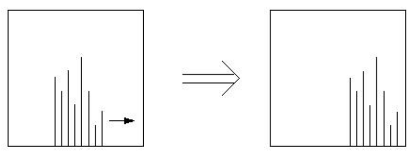
Caiani 23/03/2012

Histogram Sliding

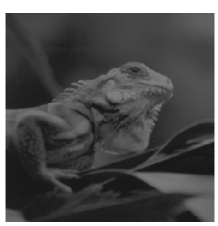
- Used to make an image either darker or lighter, but retain the relationship between gray level values
- Accomplished by simply adding or subtracting a fixed number from all of the gray level values, as follows:

$$Sliding(I(r,c)) = I(r,c) + OFFSET$$

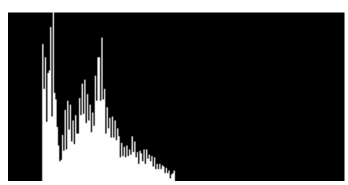
- OFFSET value is the amount to slide the histogram.
- In this equation we assume that any values slid past the minimum and maximum values will be clipped to the respective minimum or maximum
- A positive OFFSET value will increase the overall brightness, while a negative OFFSET will create a darker image



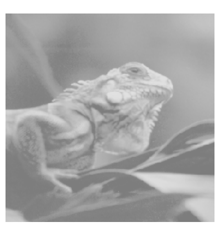
Histogram Sliding



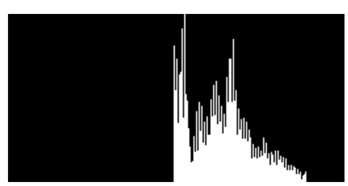
Original Image



Histogram of image



Resultant image from sliding the histogram up by 50



Histogram after sliding

Histogram equalization

- A technique where the histogram of the resultant image is as flat as possible
- The theoretical basis for histogram equalization involves probability theory, where we treat the histogram as the probability distribution of the gray levels (normalized histogram)
- Its function is similar to that of a histogram stretch but often provides more visually pleasing results across a wider range of images

Caiani 23/03/2012

HISTOGRAM EQUALIZATION

The goal of histogram equalization is to distribute the greylevels within an image so that every greylevel is equally likely to occur.

$$s = T(r) = \int_0^r p_r(w)dw$$

$p_r(r)$ is the probability density function (PDF) of the input image

The PDF of the output image is: $p_s(s)$ = 1 for 0<=s<=1 and 0 otherwise

Working with discrete quantities, the histogram of the processed image will not be uniform.

Being $p(r_j)$ for j=0,1,2,…,L-1 the histogram associated with the intensity levels of an image, and recalling that the values in the normalized histograms are approximations to the probability of occurrence of each intensity level in the image:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

Caiani 23/03/2012

Example

- 3-bits per pixel image – range is 0 to 7.

Given the following histogram:

Gray Level Value	Number of Pixels (Histogram values)
0	10
1	8
2	9
3	2
4	14
5	1
6	5
7	2

Caiani 23/03/2012

Example

1) Create a running sum of the histogram values.
   - This means the first value is 10, the second is 10+8=18, next 10+8+9=27, and so on.
   - Here we get 10, 18, 27, 29, 43, 44, 49, 51

2) Normalize by dividing by the total number of pixels. The total number of pixels is: 10+8+9+2+14+1+5+0 = 51
   - (note this is the last number from step 1)
   - So we get: 10/51, 18/51, 27/51, 29/51, 43/51, 44/51, 49/51, 51/51

3) Multiply these values by the maximum gray level values, in this case 7, and then round the result to the closest integer.
   - After this is done we obtain: 1, 2, 4, 4, 6, 6, 7, 7

Caiani 23/03/2012

Example

4) Map the original values to the results from step 3 by a one-to-one correspondence. This is done as follows:

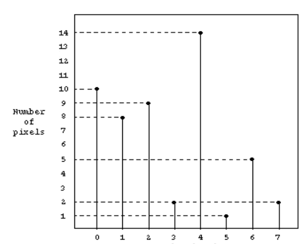
Original Gray Level Value	Histogram Equalized Values
0	1
1	2
2	4
3	4
4	6
5	6
6	7
7	7

Caiani 23/03/2012

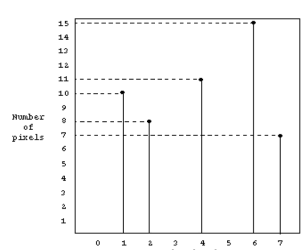
Example

- All pixels in the original image can now be mapped using this lookup table
  - Gray level 0 are set to 1,
  - Gray values of 1 are set to 2
  - Gray values of 2 set to 4,
  - Gray values of 3 set to 4,
  - And so on.

We can see the original histogram and the resulting equalized histogram. Although the result is not flat, it is closer to being flat than the original histogram



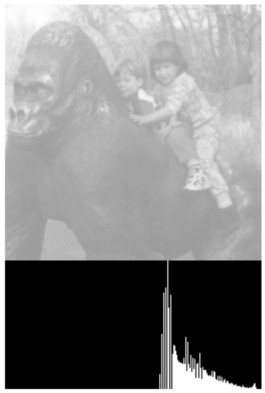
Original Histogram



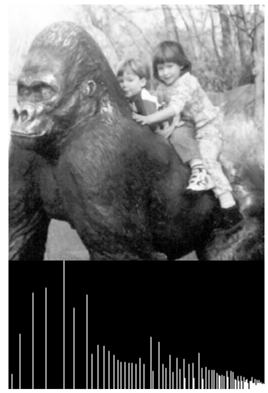
Equalized Histogram

Caiani 23/03/2012

Histogram Equalization Examples



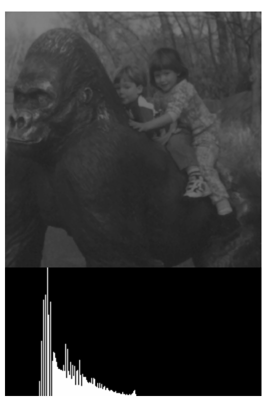
Input image



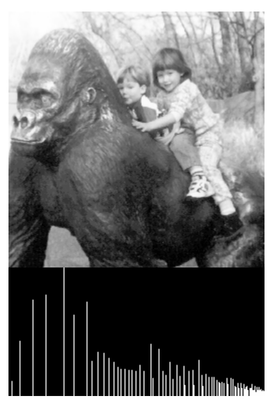
Histogram equalized image

Caiani 23/03/2012

Histogram Equalization Examples



Input image



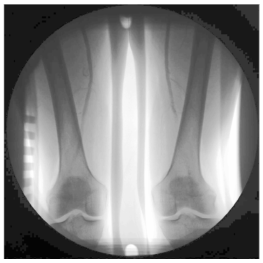
Histogram equalized image

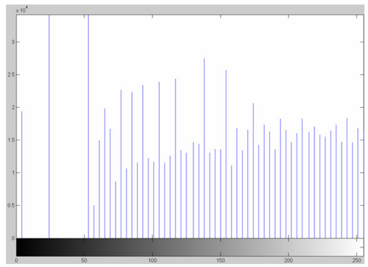
Histogram equalization provides similar results regardless of the input image

Caiani 23/03/2012

Histogram equalization with Matlab

The command C=**histeq**(A,nlev) operates the histogram equalization of A , where nlev is the number of intensity levels for the output image (by default 64 values).

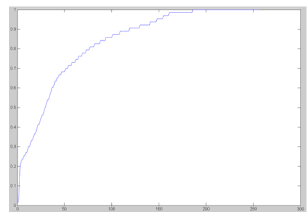




Try it on one of the radiological images, changing nlev.

To be able to use the appied transformation function:

>>[C,T]=**histeq**(A)
>>**plot**(T)



imadjdemo.

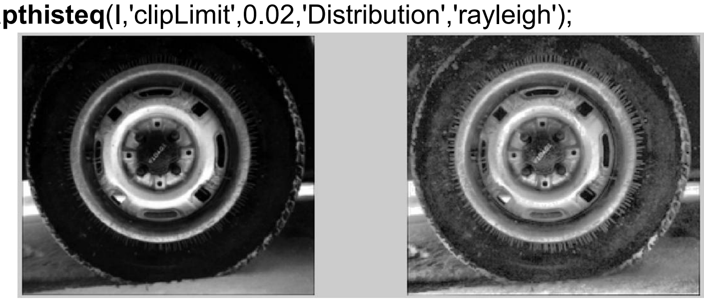
Caiani 23/03/2012

The command D=**adapthisteq**(A) processes small regions of the image (tiles) using histogram specification for each tile individually.

Neighbouring tiles are then combined using bilinear interpolation to eliminate artificially induced boundaries.

A = **adapthisteq**(I,’NumTiles’,[8 8], ‘ClipLimit',0.01, ’NBins’,256, ‘Range’, ‘full’, 'Distribution',‘uniform');

A = **adapthisteq**(I,'clipLimit',0.02,'Distribution','rayleigh');



Select an image, and apply first histeq and adapthisteq for equalization, changing some parameters and observing the differences.
Try then with a different probability distribution.

Caiani 23/03/2012

Histogram matching (specification)

It is useful in some applications to be able to specify the shape of the histogram that we wish the processed image to have.

$$s = T(r) = \int_0^r p_r(w)\,dw$$

$p_s(s)$ has a uniform PDF

Suppose we define a variable z with the property:

$$H(z) = \int_0^z p_z(w)\,dw = s$$

From these two equations we have: $z = H^{-1}(s) = H^{-1}(T(r))$

Then, we can find T(r) from the input image, so we can use it to find the transformed levels z, whose density is the specified $p_z(z)$, provided that we can find H^{-1}.

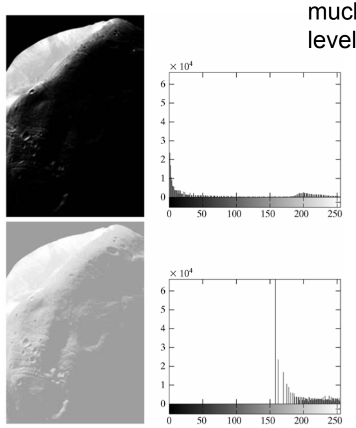
Working with discrete variables, H^{-1} exists if $p(z_k)$ is a valid histogram (unit area and nonnegative values), and none of its components is zero (no bin of $p(z_k)$ is empty)

Caiani 23/03/2012

Matlab implements histogram matching with:

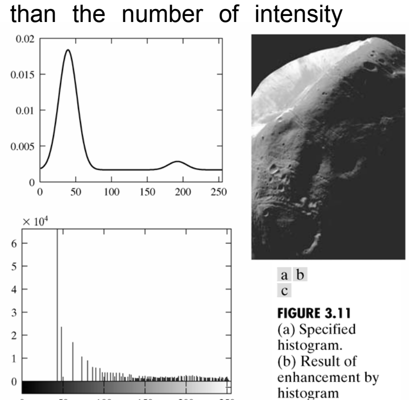
>>g=**histeq**(f,hspec)

where hspec is the specified histogram, containing integer counts corresponding to equally spaced bins. The histogram of g will better match hspec when **length**(hspec) is much smaller than the number of intensity levels in f.



a b
c d

FIGURE 3.10 (a) Image of the Mars moon Phobos. (b) Histogram. (c) Histogram-equalized image. (d) Histogram of (c). (Original image courtesy of NASA).



a b
c

FIGURE 3.11 (a) Specified histogram. (b) Result of enhancement by histogram matching. (c) Histogram of (b).

Select an image, and apply histeq, extracting the transformation applied.
Compute the negative of the original image and apply the found transformation.

Caiani 23/03/2012