Point Operations

- ARITHMETIC OPERATIONS
- HISTOGRAM PROCESSING

Image Processing in Spatial Domain

Spatial domain refers to the image plane itself.

Image processing methods in spatial domain may be divided into 2 principle categories

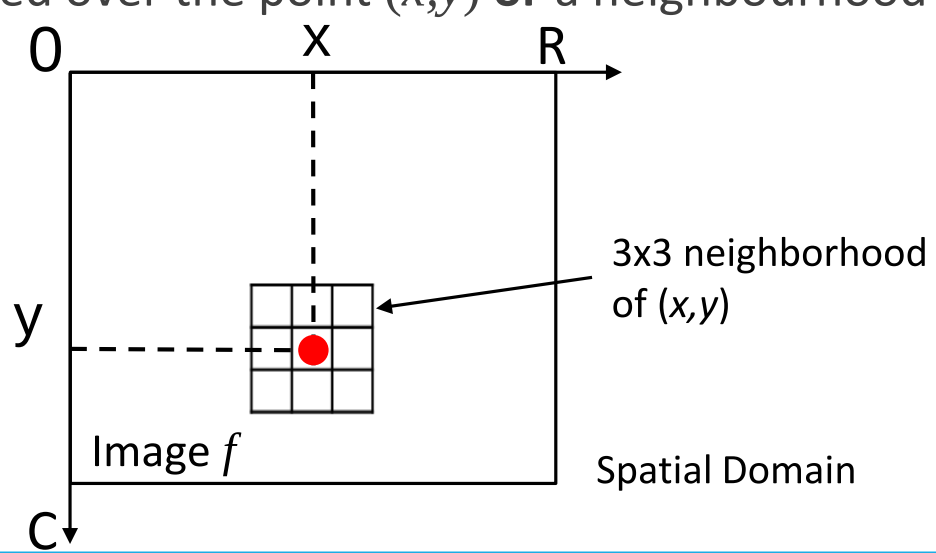
1. **Point operations/Intensity transformation**
   - operate on *single pixels* of an image
   - principally for the purpose of contrast manipulation and image thresholding
2. **Spatial filtering**
   - process the pixel in a small *neighborhood of pixels* around the given pixel
   - deals with performing operations, such as image sharpening

Image Processing Methods in Spatial Domain

The spatial domain processes can be denoted by the expression,

$$g(x,y)=T\left[f(x,y)\right]$$

Where $f(x,y)$ is the input image, $g(x,y)$ is the output image and T is the operator on f defined over the point (x,y) **or** a neighbourhood of the point (x,y).

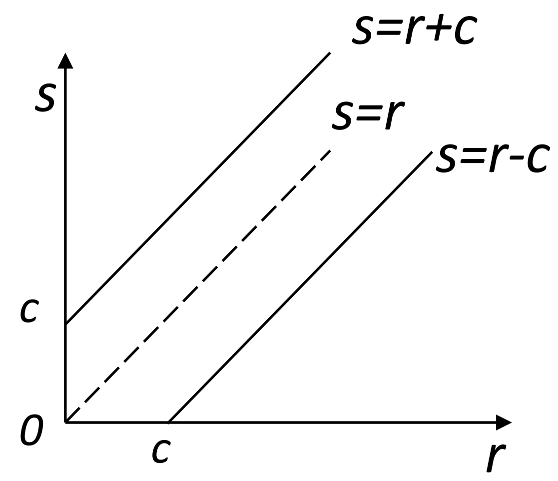


Arithmetic Operations

Let *r* is the old grey level value of a pixel, *s* is the new grey level value of a pixel, *c* is a positive constant.

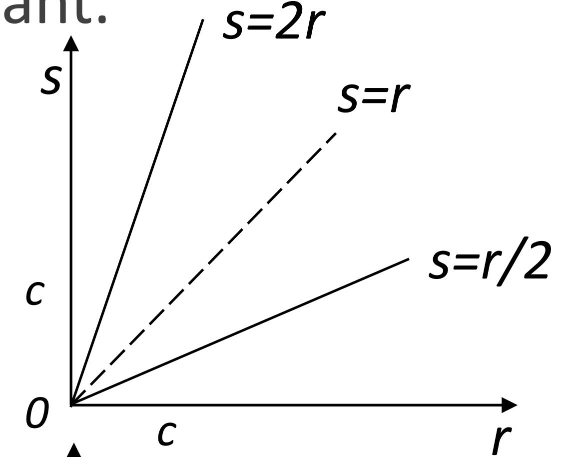
Addition: $s = r + c$

Subtraction: $s = r - c$



Arithmetic Operations

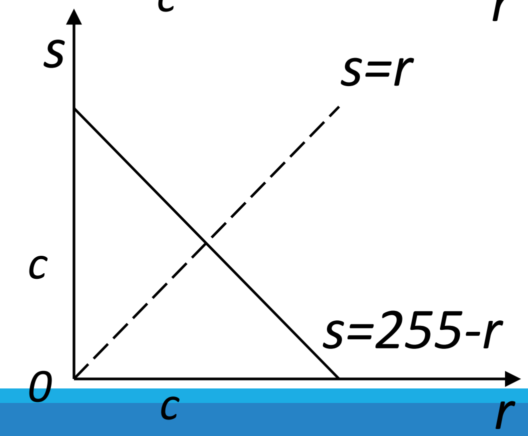
Let r is the old grey level value of a pixel, s is the new grey level value of a pixel, c is a positive constant.



Multiplication: $s = cr$

Division: $s = r/c$

Complement: $s = 255 - r$



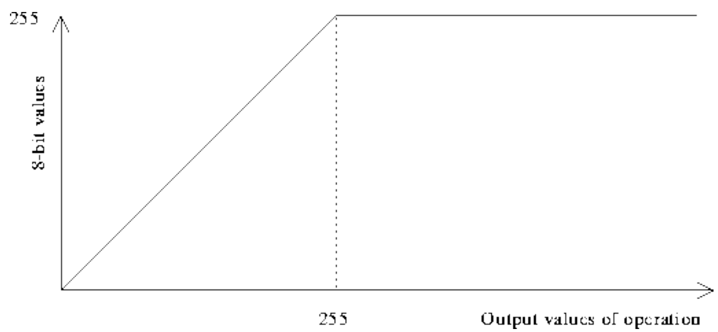
Arithmetic Operations (cont)

To ensure that the results are integers in the range [0, 255], the following operations should be performed.

- Rounding the result to obtain an integer

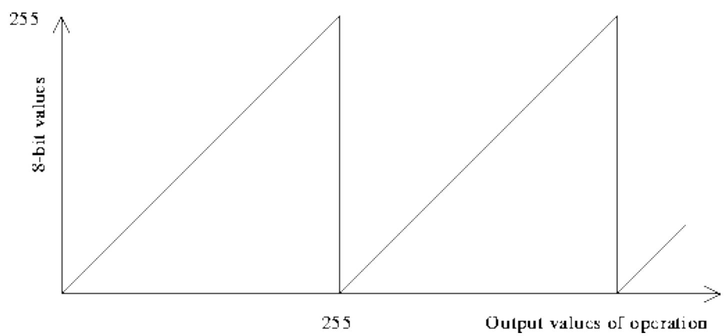
and

- Clipping the result by
  - setting s = 255, if s > 255
  - setting s = 0, if s < 0



or

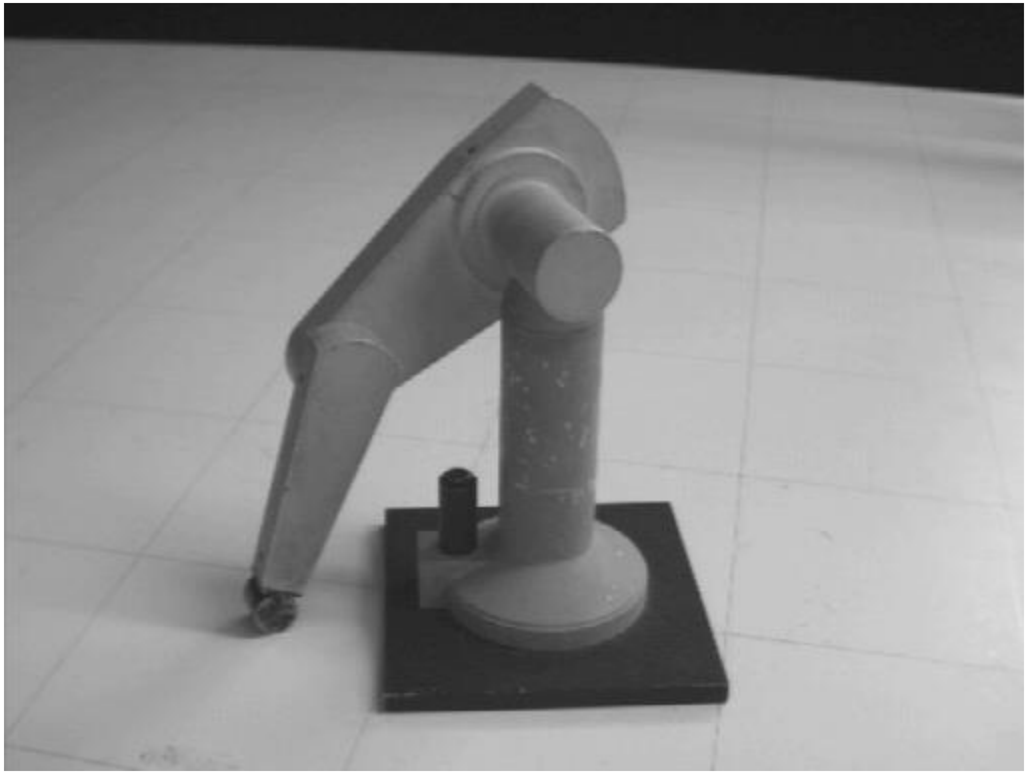
- Wrap-around the results by
  - set $s = s$-255*n, if s > 255
  - set s = 0, if s < 0



Example

grey level： 7-76

x3



grey level： 21-228

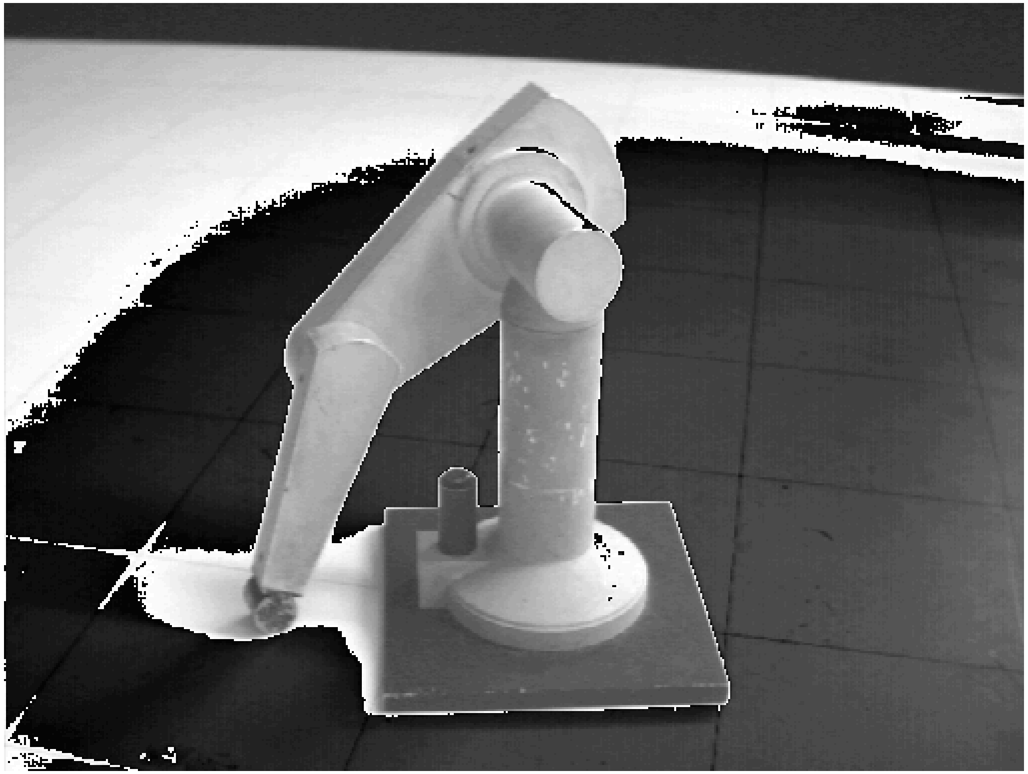
x5:clipping

Old pixel value > 51 ➔ new pixel value > 255

Those pixel values are clipped to be 255.

It is often safest to change to an image format with a large range, *e.g.* floating point, before multiplication.

x5:Wrap around



Old pixel value > 51 ➔ new pixel value > 255

Those pixel values are wrapped around from 255 back to 0.

Addition & Subtraction

- Lighten/darken the input image
- Some details may be lost and those are not retrievable. (because of the rounding and clipping)

- MATLAB code

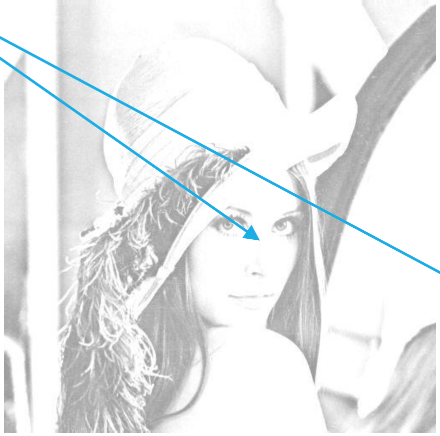
```
commands:
r = imread('filename.ext');
s1 = uint8(double(r) + c);
s2 = uint8(double(r) - c);
```

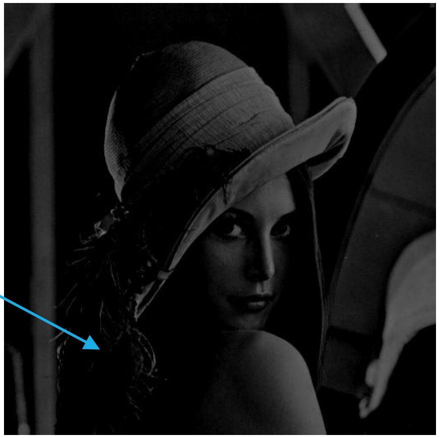
Example: Addition & Subtraction

Added by 128

Subtracted by 128

Some details are lost!





Multiplication & Division

- Lighten/darken the image
- Some details may be lost and those are not retrievable. (but less than addition/subtraction)
- MATLAB code

```
r = imread('filename.ext');
s1 = uint8(double(r)*c); or
s2 = uint8(double(r)/c);
```

Example: Multiplication & Division

Multiplied by 2

Divided by 2

Some details are lost!

Comparison: Addition VS Multiplication

Addition

Multiplication

Comparison: Subtraction VS Division

Subtraction

Division

Complement

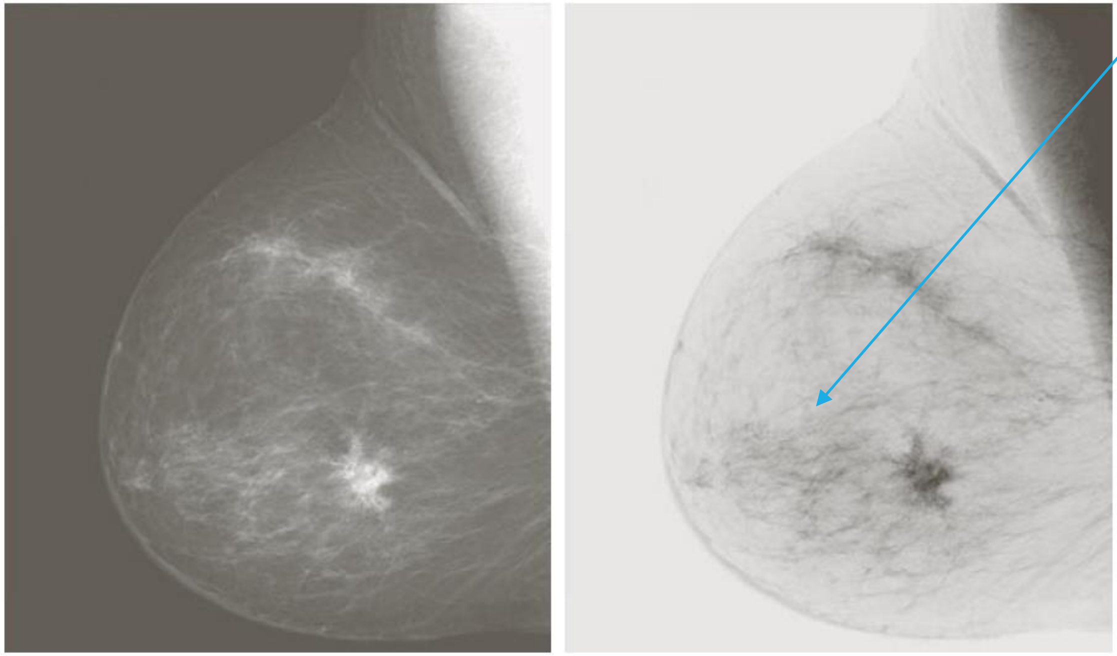
- Create the negative image.
- Suited for enhancing white or grey details embedded in dark regions of an image, especially when the black areas are dominant in size.
- MATLAB Code:

```
x = imread('filename.ext');
y = uint8(255 - double(x));
```

Example: Complement

Example: Complement

Mammogram using negative transformation

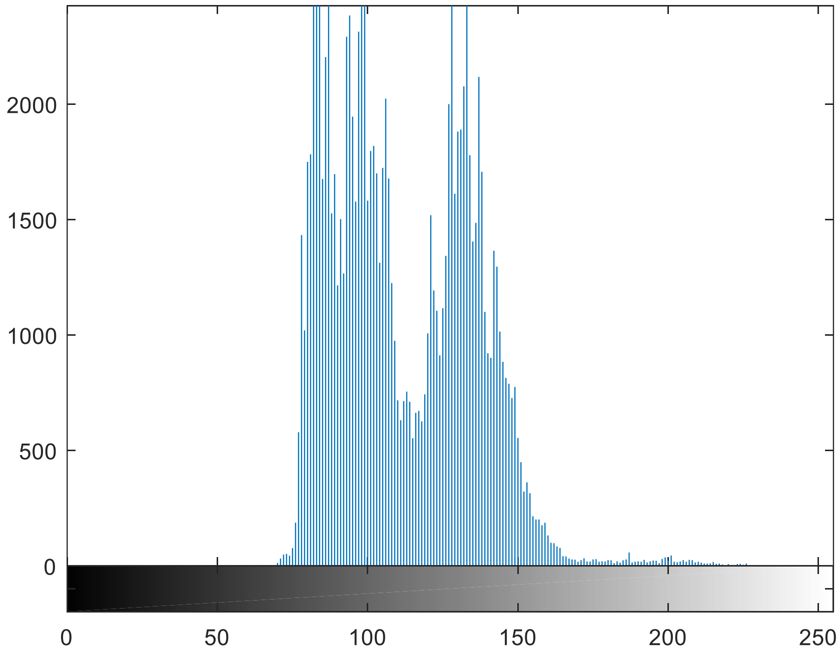


Easy to see tissues in negative image

Recap: Histogram

- Histogram:
  - A graph showing the number of pixels at each intensity level or color.
- Normalized histogram:
  - A histogram where the number of pixel is divided by the total number of pixels, so the range is [0,1]
  - corresponding to the *pdf* function.
- Cumulative histogram:
  - A histogram which shows the number of pixels whose intensity is *less or equal* to each intensity divided by the total number of pixels.
  - corresponding to the *CDF* function.

Recap: Histogram




```
>> p = imread('pout.tif')
>> imshow(p)
```

```
>> figure; histogram(p)
```

What does Histogram describe?

Brightness

- Dark image has grey levels (histogram) cluttered at the lower end.
- Bright image has grey levels (histogram) cluttered at the higher end.

Contrast

- Well-contrasted image has grey levels (histogram) *spread out* over much of the range.
- Low-contrasted image has grey levels (histogram) cluttered in the center.

Contrast Enhancement

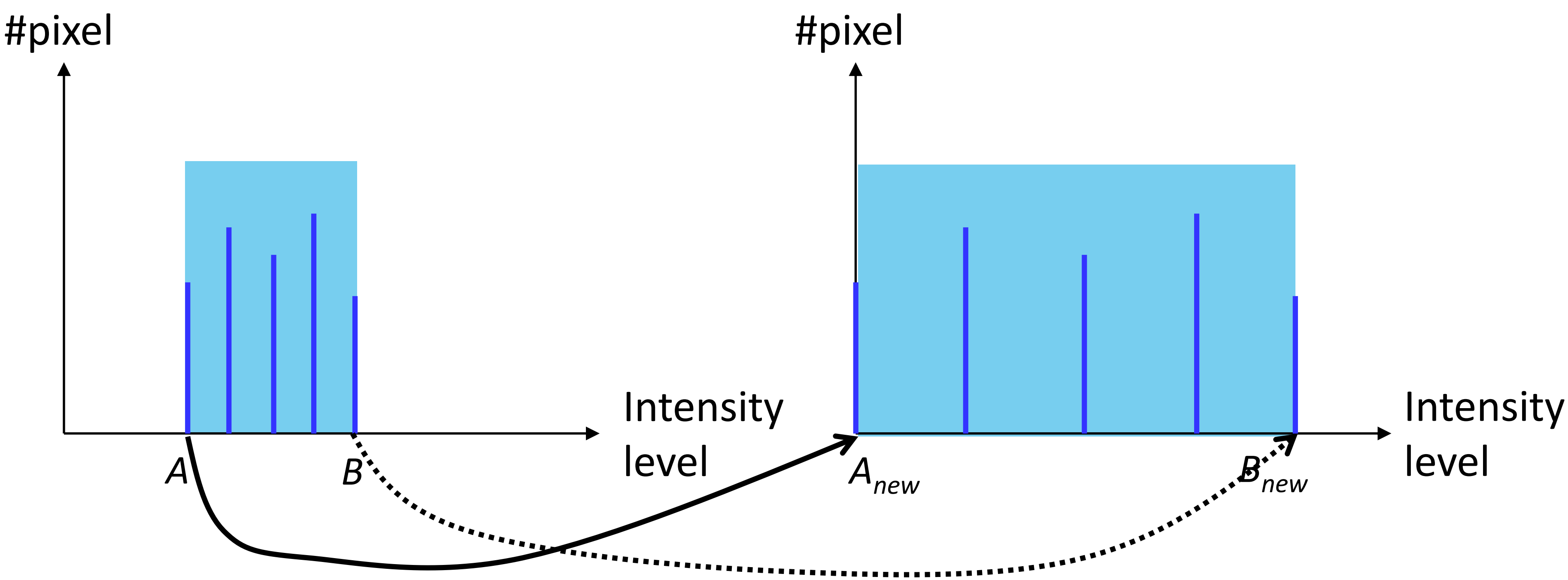
Contrast enhancement by spreading out Histogram

1. Histogram Stretching/Contrast Stretching

2. Histogram Equalization

Histogram/Contrast Stretching

- A process that expanding the range of the intensity levels in an image so that it expands the full intensity range of the recording medium or display device.

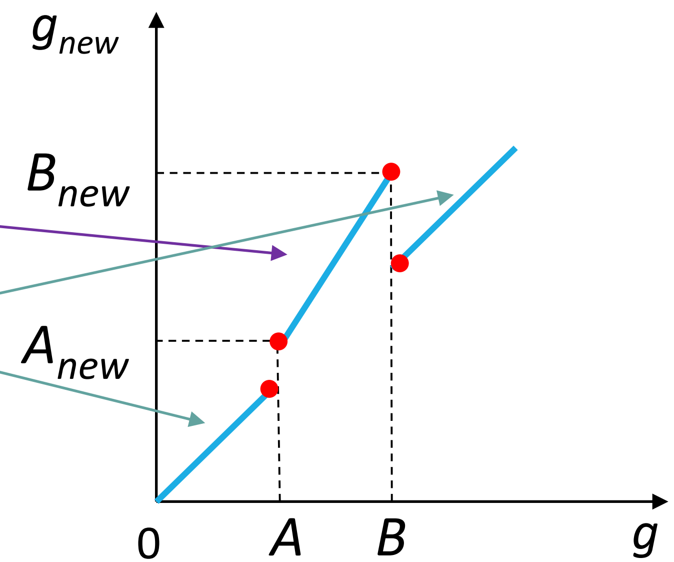


Histogram/Contrast Stretching

Stretch out the grey levels in the center of the range by applying the piecewise linear function:

$$[A, B] \rightarrow [A_{new}, B_{new}]$$

$$g_{new} = \begin{cases} \left(\dfrac{B_{new} - A_{new}}{B - A}\right)(g - A) + A_{new} & A \le g \le B \\ g & g < A \text{ or } g > B \end{cases}$$



Transfer function

Where g is the old grey level,

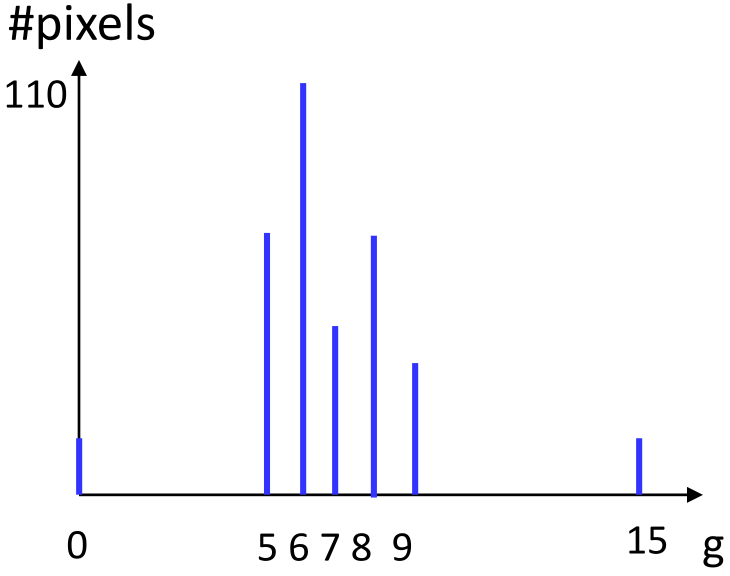
g_{new} is the stretched grey level,

A_{new} can be 0 and B_{new} can be 255.

Example: Histogram/Contrast Stretching

Given an image with histogram shown as below:

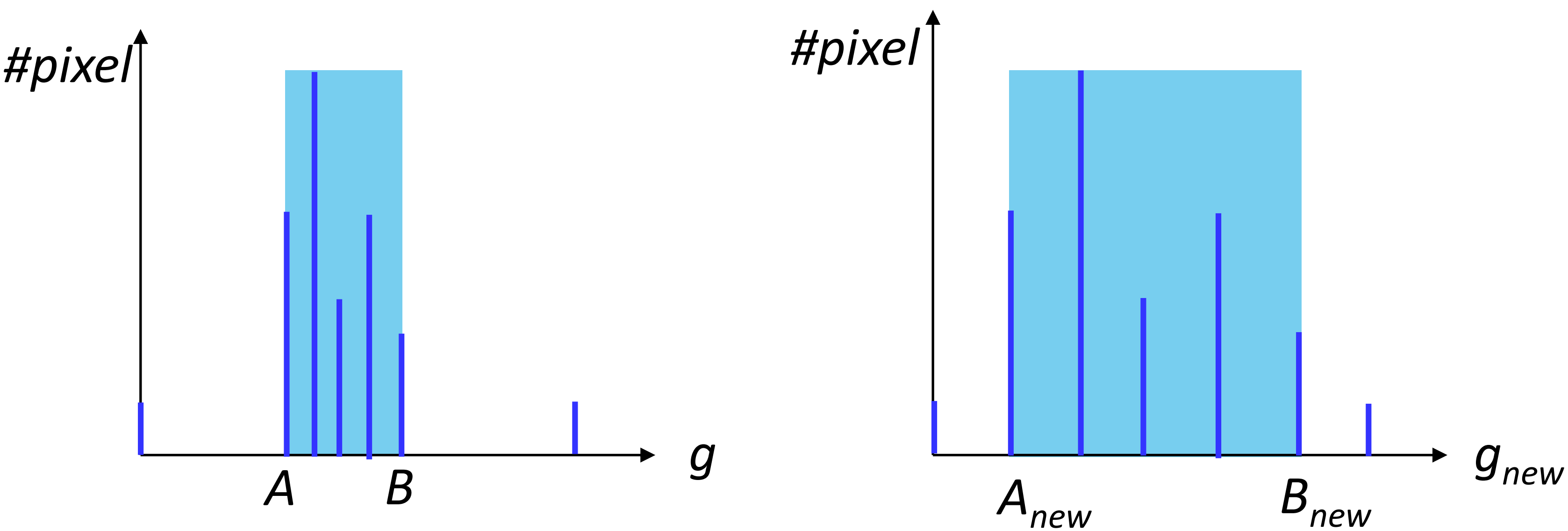
grey level-*g*	# of pixels has *g*-H_I(g)
0	15
1	0
2	0
3	0
4	0
5	**70**
6	**110**
7	**45**
8	**70**
9	**35**
10	0
11	0
12	0
13	0
14	0
15	15



Example: Histogram/Contrast Stretching

$$[A,B] \rightarrow [A_{new},B_{new}]$$

$$g_{new} = \begin{cases} \left(\dfrac{B_{new}-A_{new}}{B-A}\right)(g-A)+A_{new} & A \le g \le B \\ g & g < A \text{ or } g > B \end{cases}$$



Example: Histogram/Contrast Stretching

- Stretch grey levels from [5, 9] to [2, 14]

$$g_{new} = \left(\frac{14-2}{9-5}\right)(g-5)+2 \quad 5 \le g \le 9$$

$$g_{new} = g \quad g < 5 \text{ or } g > 9$$

Old grey level g	New grey level g_{new}
5	2
6	5
7	8
8	11
9	14

- grey levels outside this range are left as original values.

Example: Histogram/Contrast Stretching

New grey level-g_{new}	# of pixels has g-$H_I(g_{new})$
0	15
1	0
2	70
3	0
4	0
5	110
6	0
7	0
8	45
9	0
10	0
11	70
12	0
13	0
14	35
15	15

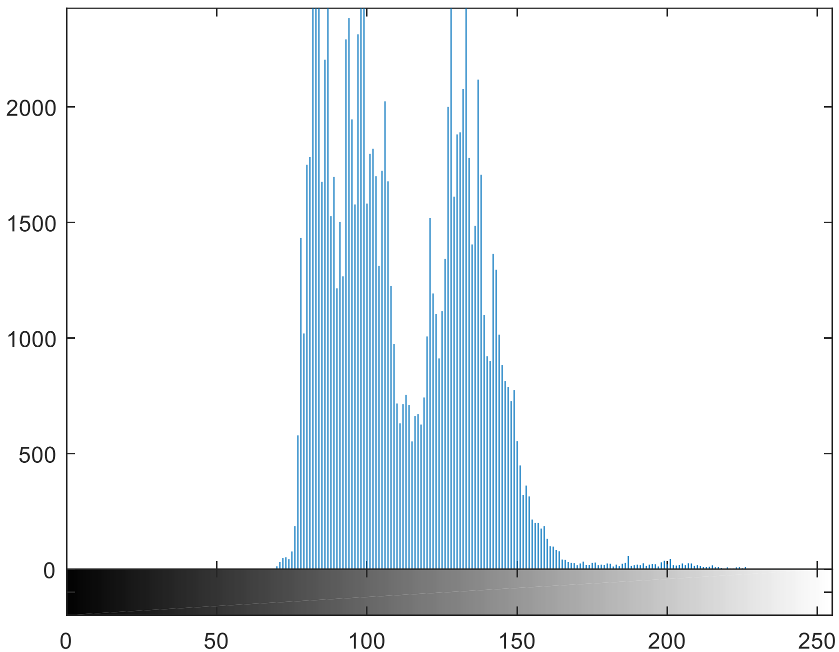
The histogram after stretching

Example: Histogram/Contrast Stretching

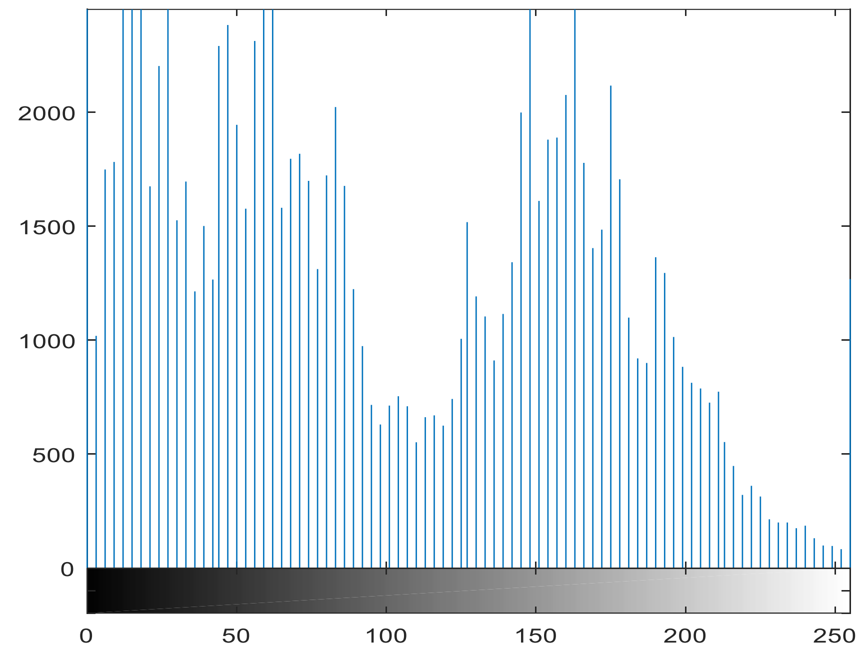
original

output

Example: Histogram/Contrast Stretching



[66,238]



[0,255]

Matlab/Scilab: Histogram/Contrast Stretching

From image processing toolbox

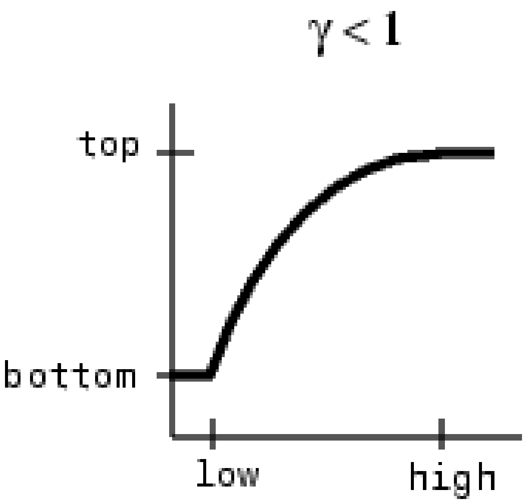
Command: **`imadjust`**

Syntax:

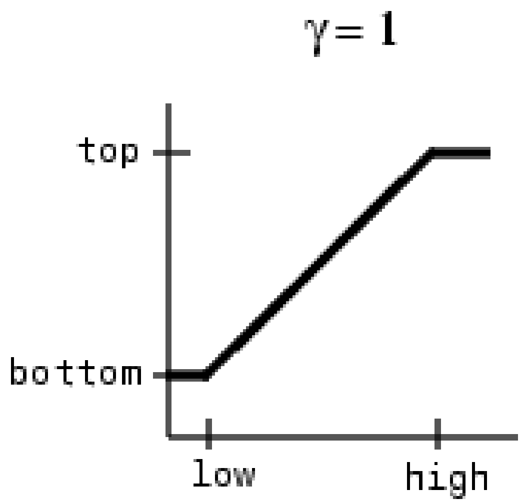
```
J=imadjust(I,[A,B],[C,D],gamma);
```

- convert intensity I $\leq$ A to B
- convert intensity I $\geq$ C to D
- values of A,B,C and D must be between 0 and 1
- Gamma (γ): specifies the shape of the curve describing the relationship between the values in I and J. (positive constant; < 1 concave downward, > 1 concave upward)

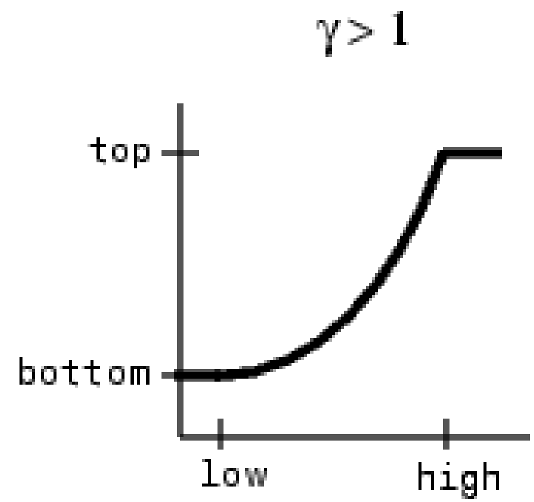
Gamma value (Power–Law Transformation)



Brighten image



Linear mapping



Darken image

Gamma value (Power–Law Transformation)

Image Courtesy of Susan Cohen

Original

Adjust by using Gamma = 0.5

Matlab/Scilab: Piecewise Linear

A function for applying a piecewise linear-stretching function

Command: **`find`**

Syntax: `find(`*`condition`*`)`

Example:

```
pix = find(I >= A & I < B);
```

`pix` holds the index for members in I having intensity between `A` and `B` include `A`.

Similar syntax:

```
pix = I >= A & I < B;
```

Histogram Equalization

- The trouble with the methods of histogram/contrast stretching is that they require user input.
- Histogram equalization is an entirely automatic procedure.
- Idea: Each grey level in the image occurs with the same frequency (the same number of pixel counts).
- To give the output image with uniform intensity distribution.
- To maximize the contrast evenly across the entire image.

Histogram Equalization

- Intensity level g_{new} of the output image I_{new}

$$g_{new} = T(g) = \sum_{\alpha=0}^{g} p_{\mathrm{I}}(\alpha),$$

where g is the intensity level of input image I. $p_{\mathrm{I}}(\alpha)$ is the probability density function (pdf) for intensity level α . $\sum_{\alpha=0}^{g} p_{\mathrm{I}}(\alpha)$ is the cumulative distribution function (CDF) of intensity level g.

- The pdf of output image I_{new} is uniform.

$$p_{\mathrm{I}_{new}}(g_{new}) = \begin{cases} 1 & \text{for } 0 < g_{new} < 1 \\ 0 & \text{otherwise} \end{cases}$$

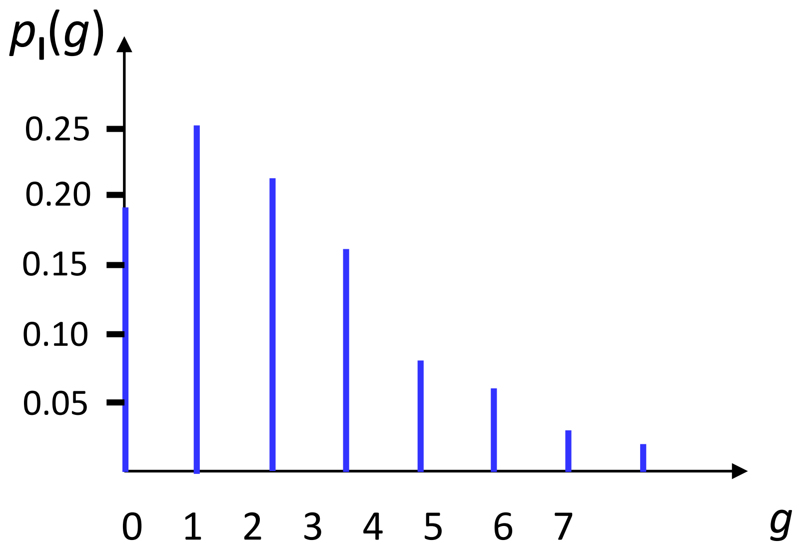
The normalized g_{new} (the range of g_{new} is 0-1)

Example: Histogram Equalization

Suppose that a 3-bit grayscale image of size 64x64 pixels has the intensity distribution as below:

Intensity level *g*	Number of pixels having *g*
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

Example: Histogram Equalization



pdf before histogram equalization

Example: Histogram Equalization

The image has 2^3=8 levels from g=0-7 and the total number of pixels is 64x64=4096.

1. work out $p_I(g)$

Intensity level g	Number of pixels having g-h(g)	$p_I(g) = N(g)/4096$
0	790	0.19
1	1023	0.25
2	850	0.21
3	656	0.16
4	329	0.08
5	245	0.06
6	122	0.03
7	81	0.02

Example: Histogram Equalization

2. use $p_I(g)$ to work out g_{new}.

Intensity level *g*	Number of pixels having g-*h(g)*	$p_I(g)$=*h(g)* */4096*	$g_{new} = \sum_{i=0}^{g} p_I(g)$
0	790	0.19	0.19
1	1023	0.25	0.19+0.25=0.44
2	850	0.21	0.19+0.25+0.21 =0.65
3	656	0.16	0.81
4	329	0.08	0.89
5	245	0.06	0.95
6	122	0.03	0.98
7	81	0.02	1

Example: Histogram Equalization

3. convert the normalized g_{new} to g_{new} with range 0-7.

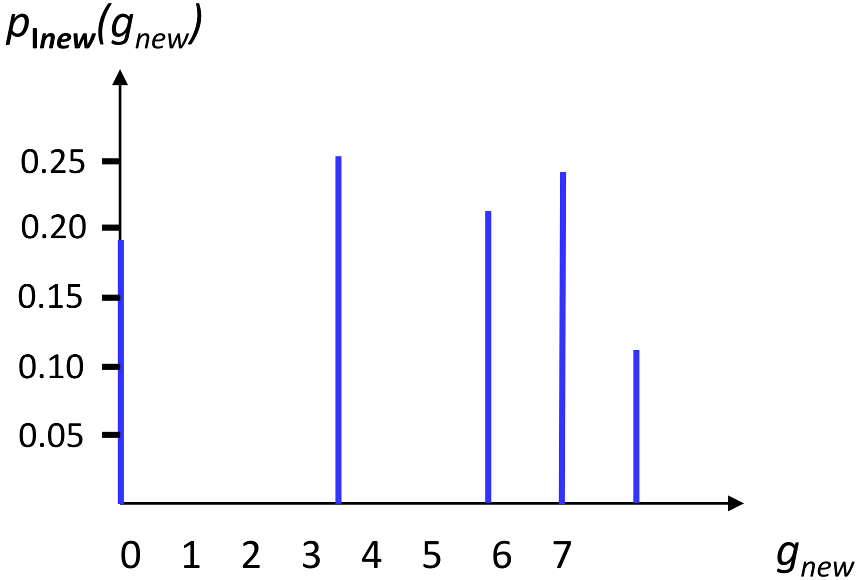
4. round g_{new} to nearest integer.

Intensity level g	Number of pixels having g-h(g)	$p_{\mathsf{I}}(g)$=h(g) /4096	$g_{new} = \sum_{g} p_{\mathsf{I}}(g)$	g_{new} with range 0-7	Rounded g_{new}
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1	7	7

Example: Histogram Equalization

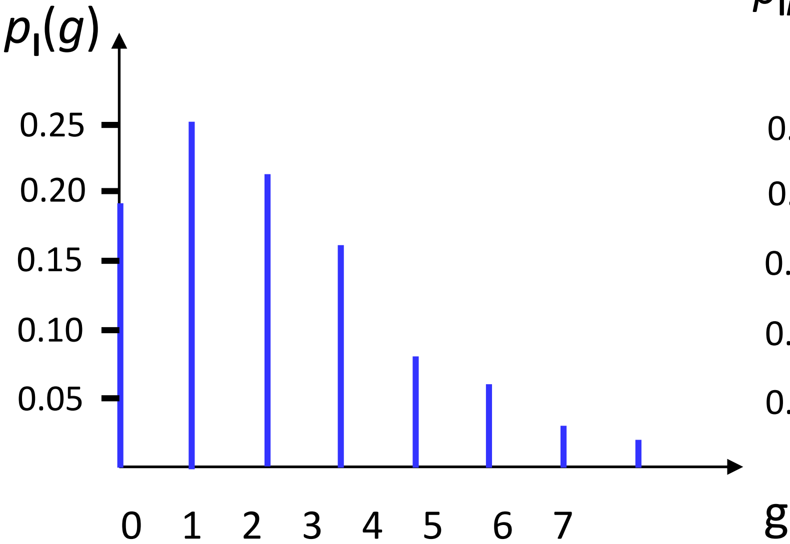
5. generate the new histogram.

g_{new}	$p_{Inew}(g_{new})$
1	0.19
3	0.25
5	0.21
6	0.16
6	0.08
7	0.06
7	0.03
7	0.02

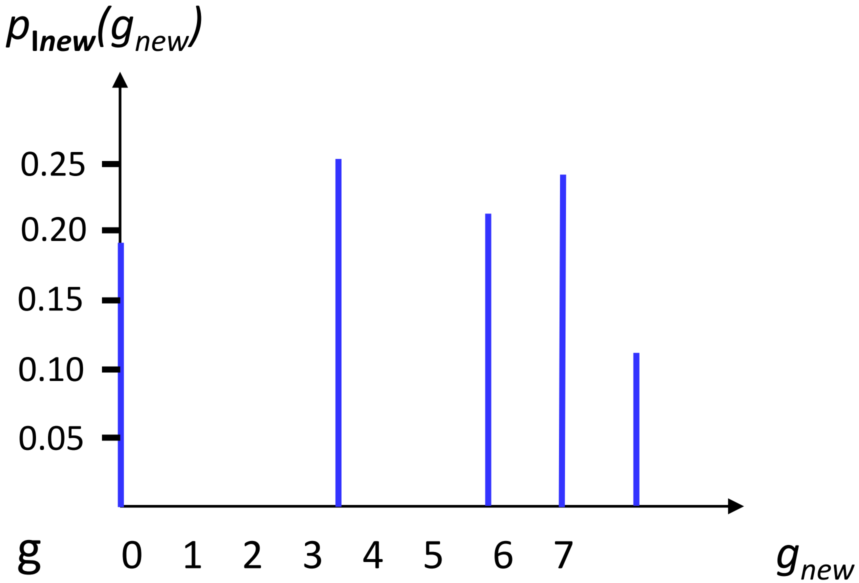


pdf after histogram equalization

Example: Histogram Equalization



pdf of input image *I*



pdf of output image I_{new}

Example: Histogram Equalization

Before

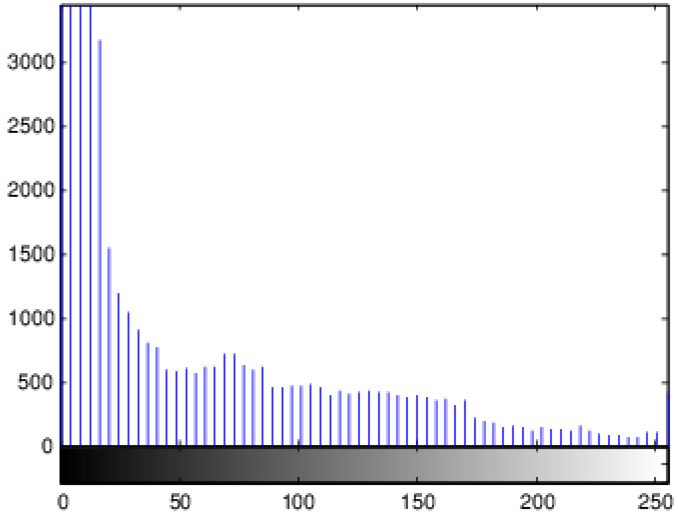
After

http://www.mathworks.com/access/helpdesk/help/toolbox/images/histeq.html

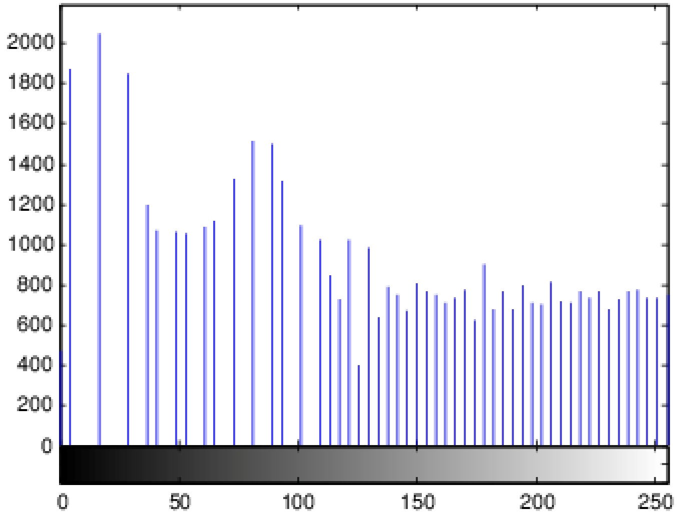
Matlab: Histogram Equalization

Command: **`histeq (from SIVP toolbox)`**

Syntax: `histeq(image, out_graylev)`
`histeq(indexed_im, map, out_greylev)`



Before



After

Lookup Tables

- Used to improve the performance of point operations
- Why?
  - one intensity is always mapped to the same value.
  - reduce the computing time
- Lookup table: array

  Input intensity: index in the array

  Output intensity: value of the number

Example: Lookup Table

Function: `output = input/2;`

MATLAB

```
T = uint8(floor(0:255)/2);
output = T(input);
```

Example: Lookup Table

Function:

$$output = \begin{cases} 0.6667 \times input; & input < 96 \\ 2 \times input - 128; & 96 \le input < 161 \\ 0.6632 \times input + 85.8947; & 161 \le input \end{cases}$$

MATLAB

```
>> T1 = 0.6667*[0:95];
>> T2 = 2*[96:160] - 128;
>> T3 = 0.6632*[161:255] + 85.8947;
>> T = uint8(floor([T1 T2 T3]));
```