



OpenCV Python Tutorials

[Home](#) | [List of Articles](#) | [What's New? \(05-Mar-2013\)](#) | [Links](#)

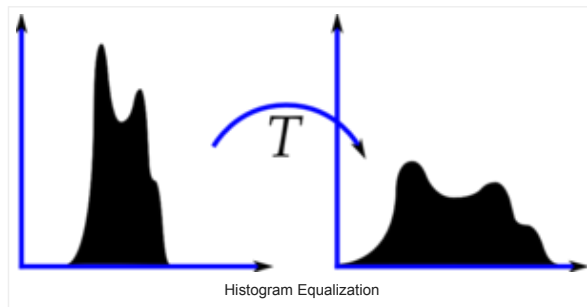
Tuesday, March 12, 2013

## Histograms - 2 : Histogram Equalization

Hi friends,

In last article, we saw what is histogram and how to plot it. This time we can learn a method for image contrast adjustment called **"Histogram Equalization"**.

So what is it ? Consider an image whose pixel values are confined to some specific range of values only. For eg, brighter image will have all pixels confined to high values. But a good image will have pixels from all regions of the image. So you need to stretch this histogram to either ends (as given in below image, from wikipedia) and that is what Histogram Equalization does (in simple words). This normally improves the contrast of the image.



Again, I would recommend you to read the [wikipedia page on Histogram Equalization](#) for more details about it. It has a very good explanation with worked out examples, so that you would understand almost everything after reading that. And make sure you have checked the small example given in "examples" section before going on to next paragraph.

So, assuming you have checked the wiki page, I will demonstrate a simple implementation of Histogram Equalization with Numpy. After that, I will present you OpenCV function. ( If you are not interested in implementation, you can skip this and go to the end of article)

### Numpy Implementation

We start with plotting histogram and its cdf (**cumulative distribution function**) of the image in Wikipedia page. All the functions are known to us except **np.cumsum()**. It is used to find the cumulative sum (cdf) of a numpy array.

```

1 import cv2
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 img = cv2.imread('wiki.jpg',0)
6
7 hist,bins = np.histogram(img.flatten(),256,[0,256])
8
9 cdf = hist.cumsum()
10 cdf_normalized = cdf * hist.max() / cdf.max() # this line not necessary.
11
12 plt.plot(cdf_normalized, color = 'b')
13 plt.hist(img.flatten(),256,[0,256], color = 'r')
14 plt.xlim([0,256])
15 plt.legend(('cdf','histogram'), loc = 'upper left')
16 plt.show()
```

### About Me



Abid Rahman K

G+ Follow 828

[View my complete profile](#)

G+ 837

Total Pageviews

883,450

Search This Blog

Google+ Followers

OpenCV Python

Follow

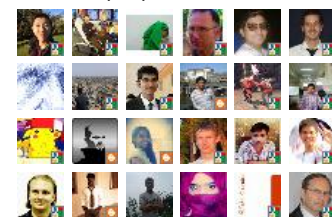


666 have us in circles

[View all](#)

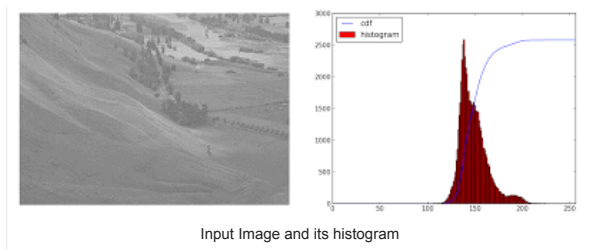
Followers

Join this site  
with Google Friend Connect

Members (100) [More »](#)Already a member? [Sign in](#)

Blog Archive

- 2011 (2)
- 2012 (26)
- ▼ 2013 (10)
  - January (2)
  - ▼ March (4)



You can see histogram lies in brighter region. We need the full spectrum. For that, we need a transformation function which maps the input pixels in brighter region to output pixels in full region. That is what histogram equalization does.

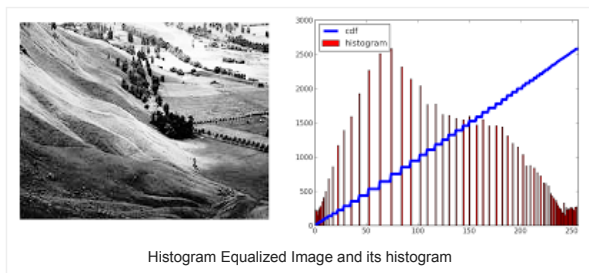
Now we find the minimum histogram value (excluding 0) and apply the histogram equalization equation as given in wiki page. But I have used here, the masked array concept array from Numpy. For masked array, all operations are performed on non-masked elements. You can read more about it from [Numpy docs on masked arrays](#)

```
1 cdf_m = np.ma.masked_equal(cdf,0)
2 cdf_m = (cdf_m - cdf_m.min())*255/(cdf_m.max()-cdf_m.min())
3 cdf = np.ma.filled(cdf_m,0).astype('uint8')
```

Now we have the look-up table that gives us the information on what is the output pixel value for every input pixel value. So we just apply the transform.

```
1 img2 = cdf[img]
```

Now we calculate its histogram and cdf as before ( you do it) and result looks like below :



You can see a better contrast in the new image, and it is clear from the histogram also. Also compare the cdfs of two images. First one has a steep slope, while second one is almost a straight line showing all pixels are equi-probable.

Another important feature is that, even if the image was a darker image (instead of a brighter one we used), after equalization we will get almost the same image as we got. As a result, this is used as a "reference tool" (I don't get a more suitable than this) to make all images with same light conditions. This is useful in many cases, for eg, in face recognition, before training the face data, the images of faces are histogram equalized to make them all with same light conditions. It provides better accuracy.

### OpenCV Implementation

If you are bored of everything I have written above, just leave them. You need to remember only one function to do this, `cv2.calcHist()`. Its input is just grayscale image and output is our image.

Below is a simple code snippet showing its usage for same image we used :

```
1 img = cv2.imread('wiki.jpg',0)
2 equ = cv2.equalizeHist(img)
3 res = np.hstack((img,equ)) #stacking images side-by-side
4 cv2.imwrite('res.png',res)
```

See the result :



[Histograms - 1 : Find, Plot, Analyze !!!](#)

[Histograms - 2 : Histogram Equalization](#)

[Histograms - 3 : 2D Histograms](#)

[Histograms - 4 : Backprojection](#)

- ▶ [May \(1\)](#)
- ▶ [June \(1\)](#)
- ▶ [July \(1\)](#)
- ▶ [November \(1\)](#)

### Popular Posts

[Install OpenCV in Windows for Python](#)



[Contours - 1 : Getting Started](#)



[Thresholding](#)



[Contours - 2 : Brotherhood](#)



[Contours - 3 : Extraction](#)



So now you can take different images with different light conditions, equalize it and check the results.

Histogram equalization is good when histogram of the image is confined to a particular region. It won't work good in places where there is large intensity variations where histogram covers a large region, ie both bright and dark pixels are present. I would like to share to SOF questions with you. Please checkout the images in the questions, analyze their histograms, check resulting images after equalization :

[How can I adjust contrast in OpenCV in C?](#)

[How do I equalize contrast & brightness of images using opencv?](#)

So I would like to wind up this article here. In this article, we learned how to implement Histogram Equalization, how to use OpenCV for that etc. So take images, equalize it and have your own hack arounds.

See you next time !!!

Abid Rahman K.

Posted by [Abid Rahman K](#) at 12:19 PM



+25 Recommend this on Google

Labels: [brightness](#), [contrast](#), [cv2.calcHist](#), [equalization](#), [histogram](#), [matlab](#), [normalize](#)

11 comments



Add a comment as Steven Vo

Top comments



**Abid Rahman K** 2 years ago - [Python \(Tutorials\)](#)  
Histogram Equalization with Python & Numpy

+4

1



**timger timger** 2 years ago  
i like Numpy



**Abid Rahman K** 2 years ago - [OpenCV \(Diskussion\)](#)  
Histogram Equalization in OpenCV-Python

1




**Abid Rahman K** 2 years ago - [Computer Vision \(Discussion\)](#)  
Histogram Equalization in Python


1





**Abid Rahman K** 2 years ago - [Image Processing \(Discussion\)](#)  
Histogram Equalization in Python


1



**Abid Rahman K** 2 years ago - [Scientific Python \(Discussion\)](#)  
 Histogram Equalization in Python & Numpy  
 +3 1


**OpenCV Python** via Google+ 2 years ago - Shared publicly  
 Histogram Equalization  
 +2 1 · Reply


**John Williams** 3 months ago - Shared publicly  
 Found your blog. Its really have good information on computer programming. Really liked it. Thank you for all the information. To get tutorials on Python programming, You can visit at <http://voidmain.in/>  
 1 · Reply


**Valentine Azom** 5 months ago - Shared publicly  
 Nice tutorial. helped to understand this concept. Please keep it up  
 1 · Reply


**Volkan AĞAOĞLU** 1 year ago - Shared publicly  
 Hi Mr Rahman,  
  
 How can i use CLAHE from on python under windows,  
  
 Can you help me ?  
 +1 1 · Reply


**Marcelo Garcia** 1 year ago - Shared publicly  
 Hello  
 I do have this coding for compare histogram but I don't know what it is missing, I am not able to see the results, can anybody help me with this please. thanks  
 Read more (58 lines)  
 1 · Reply

[Newer Post](#)[Home](#)[Older Post](#)Subscribe to: [Post Comments \(Atom\)](#)[Latex](#)

Simple template. Powered by Blogger.