# Part 1

## To create a digital image we need to convert the continuously sensed data into a digital form. The digital image is represented by:

## The points (x,y) on the grid on which we sample

## The intensity of that specific point

## Quantization is the process that usually comes after the sampling action and takes the value (intensity) in a specific point and converts it to a district value according to the number of bits that represent the limited digital image pixels. Mainly, the quantization will be used to create a digital image.

## Uniform quantization means that the quantization step will be equal for all the steps. In the case of 8-bit representation q will be 1/256. Hence we will implement the function as shown below. def quant\_img(img, N=256):

## f\_max = img.max()

## f\_min = img.min()

## dynamic = f\_max - f\_min

## return np.floor((img/dynamic)\*N)/N

| N = 50 | N = 3 |
| --- | --- |
|  |  |

## 

## 

## Once we are rounding color \ grayscale level values, we actually make bigger level steps between two pixels, therefore some of the image’s colors or shades will be lost so the smoothie color transition between two pixels will disappear and the big steps will cause false contours.

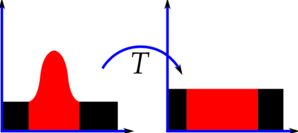
1. K-Means is a clustering algorithm, which clusters together data points based on the number of clusters you want to identify in your data. In K-Means Clustering Algorithms, K is the number of clusters. K-Means randomly selects the initial clusters and then based on the distance, assigns the data points to the nearest clusters.

# Part 2

## Contrast stretching is an image processing technique that is used for increasing the dynamic range of an image intensity. Sometimes we can use it to emphasize objects in a specific range of color or grayscale values. Mostly the transformation will be linear and monotonic increasing. The image below illustrates an example of contrast stretching of an image that uses all the dynamic range after the stretching transformation.

## 

1. Histogram equalization is a widely used contrast-enhancement technique in image processing because of its high eﬃciency and simplicity. The purpose of this method is to divide the intensities in uniform distribution on the spectrum.



1. my\_hist(img) - takes a uint8 image and returns a 1 × 256 vector with the corresponding histogram count.

Math formula:

Implementation:

def my\_hist(img):

hist\_vect = [0]\*256

img\_vect = img.reshape(-1, 1)

for pix in img\_vect:

hist\_vect[int(pix)] += 1

return hist\_vect

Results:

## 

| Image | Histogram by my\_his() |
| --- | --- |
|  |  |

## 

## my\_hist\_eq(img) - takes a uint8 image and returns the uint8 equalized image.

## Math form:

Implementation:

## from numpy.ma.core import zeros\_like

## 

## def my\_hist\_eq(img):

## N, M = img.shape

## eq\_img = zeros\_like(img)

## 

## hist\_vect = np.asarray(my\_hist(img))

## 

## for f in range(256):

## eq\_img = eq\_img + np.floor( img \* ( 255 / (M\*N) ) \* hist\_vect[f]).astype(np.uint8)

## 

## return eq\_img

## 

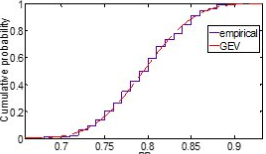
## Results:

| Beforemy\_his\_eq() | Aftermy\_his\_eq() |
| --- | --- |
|  |  |
|  |  |

## 

## The intensity distribution of an image is not uniform because a digital image is a district signal, which means that if we will implement the same algorithm as continuous signals, we will loss significant data.

1. In the figures above, in questions 1 and 2, we can see that contrast stretching is stretch the histogram to be on the whole dynamic range, while the (ideal \ continuous) histogram equalization will change the histogram distribution to be uniform.
2. Exist in our git.
3. Exist in our git.
4. Given the following continuous histogram specification:

1. Monotonic transformation means that for all the dynamic range transformed pixels will be consistent. If and will not be monotonic, then the pixels transformation will not be consistent because some of the low intensity pixels will be transformed to lower values and some of them for higher values. In the standard the transformation functions should be monotonic increased functions, so the image’s colors will be saved, which means that low intensity pixels will get low values and hight intensity pixels wil alsol remain high.
2. TBD  
   

# Part 3

## This is the objec we want to work on.

