**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech**

**Sem: VII ACAY: 2020-21**

**EXPERIMENT NO. 1**

**Aim:**

To write a program in Python to reduce grey level resolution of the given image and observe its effect.

**Software:**  PYTHON.

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Spatial resolution and grey level resolution |

**Outcome:**

After successful completion of this experiment students will be able to:

1. Understand the effect of varying grey level resolution

**Theory:**

**Grey level resolution:**

Grey level resolution of an image depends on the number of bits required to represent each pixel of the image. In a regular 8 bit image, each pixel is represented by 8 bits. So this image can have grey levels ranging from [0,255]. Grey level resolution of an image can be reduced by reducing the number of bits required to represent each pixel.

**Algorithm:**

*Grey level Resolution:*

* Read the original image.
* Letnumberof bits needed torepresent each pixel in the original image be *k*. So maximum grey level in the original image is *2k*
* Let number of bits needed to represent each pixel in the new image be *b.*  So maximum grey level in the new image is *2b*
* Let pixel in original image be referred to as *s*
* Then the pixel in the new image corresponding to pixel in the original image will be mapped as follows:

|  |
| --- |
| Name of the Experiment: To vary gray level resolution of the given image |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance:10/07/2020 Date of Submission: 10/07/2020 |

**Code:**

from skimage import data

from skimage import io

from skimage.color import rgb2gray

import matplotlib.pyplot as plt

image = data.astronaut()

#print (image)

#io.imshow(image)

img\_gray = rgb2gray (image)

img\_gray = img\_gray \* 255

img\_reso = img\_gray.copy()

b= 1

num\_levels = 2\*\*b

step\_size = 256//num\_levels

print (step\_size)

sh = img\_reso.shape

print (sh)

for r in range (0,sh[0]):

for c in range (0,sh[1]):

temp = img\_reso[r][c]

for i in range (0,num\_levels):

if temp >= i\*step\_size and temp < (i+1)\*step\_size:

img\_reso [r][c]=i\*step\_size

break

plt.figure()

plt.figure(figsize=(7,7))

plt.subplot (2,1,1)

io.imshow (img\_gray, cmap='gray')

plt.title ('Original Image')

plt.subplot (2,1,2)

io.imshow (img\_reso, cmap='gray')

plt.title ('Low Reso Image')

**Output:**

b=1



b=2



b=3



**Conclusions:**

* Larger the number of bits the greater the resolution.
* The given image can be converted to binary image if the number of bits equals to 1.
* Quality of image improves with increases with number of bits.