Student Name: ______________________________Roll No: _________________ Section: _______________

Changing gray levels and spatial resolution and Intensity Transformations

Lab Objectives

The objective of this lab is to understand and work with:

1. The effect of changing the gradient of gray levels in the input image.
2. The effect of changing spatial resolution in the input image using
   a) Nearest neighbor interpolation b) Bilinear interpolation
3. Image enhancement in spatial domain using the following gray level transformation function

- Linear Transformation
  - Image Negation function
  - Identity function
- Logarithmic Transformation
- Power Law Transformation

Changing the number of gray levels

The quality of a gray-level image is significantly affected by its gray-level resolution. Other words, increasing the number of bits per pixel has a great effect in improving the quality of gray-level images. This is because that a higher number of gray levels would give a smooth transition along the details of the image and hence improving its quality to the human eye.

Example: Changing the number of gray levels

```
% Changing the Gray Resolution From 256 to 1
I = imread('cameraman.tif');
K= imfinfo('cameraman.tif');
if(K.BitDepth ==24)
I=rgb2gray(I);
end
[r,c] = size(I);
I2= uint8(zeros(r,c));
for i = 1:r
for j=1:c
if (I(i,j)>128)
I2(i,j) =256;
else
I2(i,j) =1;
end
end
end
figure,
subplot(121),imshow(I);
subplot(122),imshow(I2);
```

Student Name: ____________________________Roll No: ________________ Section: ______________

Figure 1 Changing the number of gray Levels

Reducing the Spatial Resolution

Changing the spatial resolution of a digital image, by zooming or shrinking, is an operation of great importance in a wide range of applications (i.e. in digital cameras, biomedical image processing and astronomical images). Simply, zooming and shrinking are the operations of oversampling and under sampling a digital image, respectively. Zooming a digital image requires two steps: the creation of new pixel locations, and assignment of gray levels to those new locations. The assignment of gray levels to the new pixel locations is an operation of great challenge. It can be performed using two approaches:

Nearest Neighbor Interpolation: each pixel in the zoomed image is assigned the gray level value of its closest pixel in the original image.

Bilinear Interpolation: the value of each pixel in the zoomed image is a weighted average of the gray level values of the pixels in the nearest 2-by-2 neighborhood, in the original image.

Example : Reducing the Spatial Resolution

```
% Shrinking the image to 1/2
I = imread('cameraman.tif');
K= imfinfo('cameraman.tif');
if(K.BitDepth ==24)
I=rgb2gray(I);
end
[r,c] = size(I);
I2(1:r/2, 1:c/2) = I(1:2:r, 1:2:c);
figure,
subplot(121),imshow(I);
subplot(122),imshow(I2);
```

Student Name: ____________________________Roll No: ________________ Section: ______________

Figure 2 Reducing the Spatial Resolution

Image Enhancement in Spatial Domain -Basic Grey Level Transformations

Image enhancement is a very basic image processing task that defines us to have a better subjective judgment over the images. And Image Enhancement in spatial domain (that is, performing operations directly on pixel values) is the very simplistic approach. Enhanced images provide better contrast of the details that images contain. Image enhancement is applied in every field where images are ought to be understood and analyzed. For example, Medical Image Analysis, Analysis of images from satellites, etc.

Image enhancement simply means, transforming an image *f* into image *g* using *T*. Where *T* is the transformation. The values of pixels in images *f* and *g* are denoted by *r* and *s*, respectively. As said, the pixel values *r* and *s* are related by the expression,

```
s = T(r)
```

where *T* is a transformation that maps a pixel value *r* into a pixel value *s*. The results of this transformation are mapped into the grey scale range as we are dealing here only with grey scale digital images. So, the results are mapped back into the range [0, L-1], where $L=2^k$, k being the number of bits in the image being considered. So, for instance, for an 8-bit image the range of pixel values will be [0, 255].

There are three basic types of functions (transformations) that are used frequently in image enhancement. They are,

- Linear,
- Logarithmic,
- Power-Law.

Student Name: ___________________________Roll No: ________________ Section: ______________

The transformation map plot shown below depicts various curves that fall into the above three types of enhancement techniques.

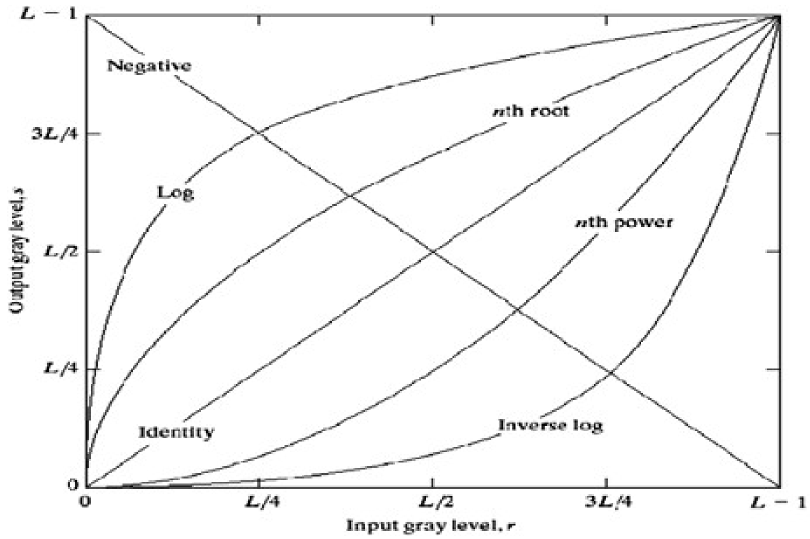


Figure 3: Plot of various transformation functions

The Identity and Negative curves fall under the category of linear functions. Identity curve simply indicates that input image is equal to the output image. The Log and Inverse-Log curves fall under the category of Logarithmic functions and *n*th root and *n*th power transformations fall under the category of Power-Law functions.

Image Negation

The negative of an image with grey levels in the range [0, L-1] is obtained by the negative transformation shown in figure above, which is given by the expression,

```
s = L - 1 - r
```

This expression results in reversing of the grey level intensities of the image thereby producing a negative like image. The output of this function can be directly mapped into the grey scale lookup table consisting values from 0 to L-1.

Log Transformations

The log transformation curve shown in fig. 3, is given by the expression,

```
s = c log(1 + r)
```

where *c* is a constant and it is assumed that $r \geq 0$. The shape of the log curve in fig. A tells that this transformation maps a narrow range of low-level grey scale intensities into a wider range of output values. And similarly maps the wide range of high-level grey scale intensities into a narrow range of high level output values. The opposite of this applies for inverse-log transform. This transform is used to expand values of dark pixels and compress values of bright pixels.

Student Name: ___________________________Roll No: ________________ Section: ______________

Power-Law Transformations

The *n*th power and *n*th root curves shown in fig. 3 can be given by the expression,

$$s = cr^{\gamma}$$

This transformation function is also called as *gamma* correction. For various values of γ different levels of enhancements can be obtained. This technique is quite commonly called as *Gamma Correction*. If you notice, different display monitors display images at different intensities and clarity. That means, every monitor has built-in gamma correction in it with certain gamma ranges and so a good monitor automatically corrects all the images displayed on it for the best contrast to give user the best experience.

The difference between the log-transformation function and the power-law functions is that using the power-law function a family of possible transformation curves can be obtained just by varying the λ.

These are the three basic image enhancement functions for grey scale images that can be applied easily for any type of image for better contrast and highlighting. Using the image negation formula

given above, it is not necessary for the results to be mapped into the grey scale range [0,L-1]. Output of L-1-r automatically falls in the range of [0, L-1]. But for the Log and Power-Law transformations resulting values are often quite distinctive, depending upon control parameters like λ and logarithmic scales. So the results of these values should be mapped back to the grey scale range to get a meaningful output image. For example, Log function $s = c \log(1 + r)$ results in 0 and 2.41 for r varying between 0 and 255, keeping c=1. So, the range [0, 2.41] should be mapped to [0, L-1] for getting a meaningful image.

Power Law Transform

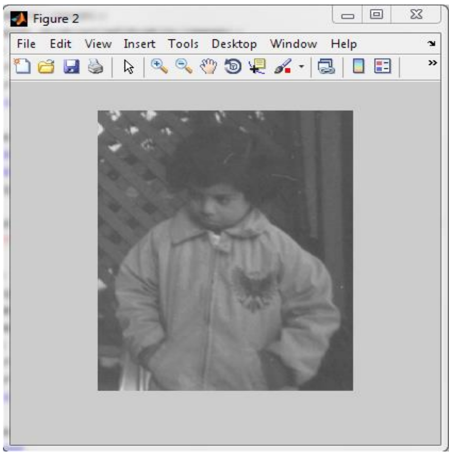
MATLAB CODE

```
image=imread('pout.tif');
figure;
imshow(image);
image_double=im2double(image);
[r c]=size(image_double);
cc=input('Enter the value for c==>');
ep=input('Enter the value for gamma==>');
for i=1:r
for j=1:c
imout(i,j)=cc*power(image_double(i,j),ep);
```

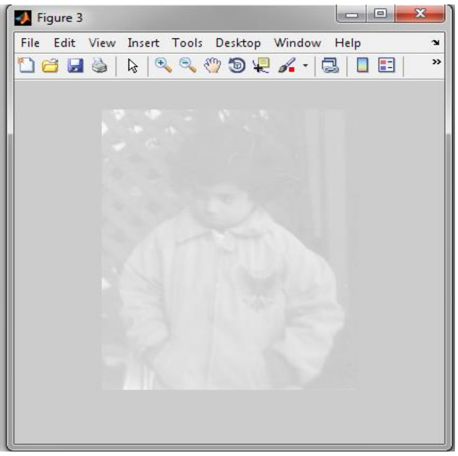
```
end
end
figure,imshow(imout);
```

OUTPUT

Enter the value for c==>1 Enter the value for gamma==>.2 % for gamma value less than 1 u gets Bright image

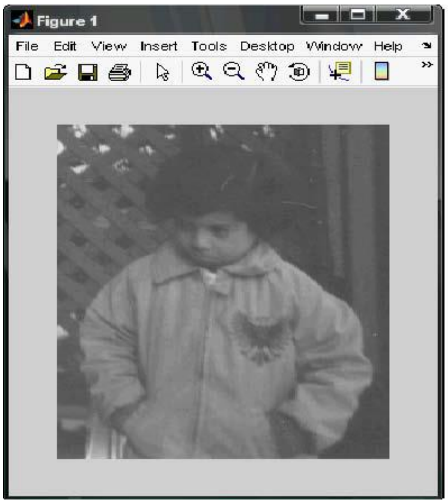


Original Image

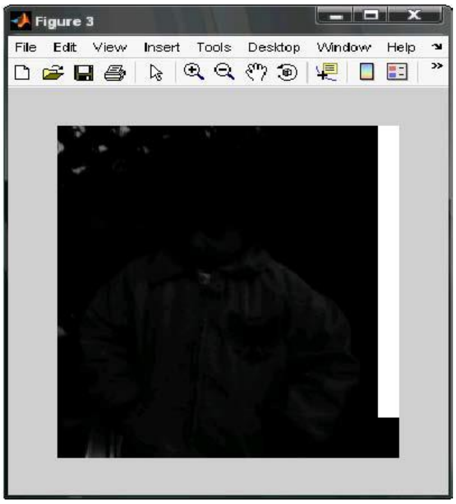


Result after Applying Power Law Transformation

Enter the value for c==>1 Enter the value for gamma==>5 % for gamma value GREATER THAN 1 u gets dark image



Original Image



Result after Applying Power Law Transformation

Student Name: ______________________________Roll No: _________________ Section: _______________

Student Exercise:

Task 1
Reducing the Number of Gray Levels in an Image
Write a computer program capable of reducing the number of gray levels in an image from 256 to 2, in integer powers of 2. The desired number of gray levels needs to be a variable input to your program.

Task 2
Zooming and Shrinking Images by Nearest Neighbor
Write a computer program capable of zooming and shrinking an image by nearest neighbor algorithm. Assume that the desired zoom/shrink factors are integers. You may ignore aliasing effects.

Task 3
Zooming and Shrinking Images by Bilinear Interpolation
Write a computer program capable of zooming and shrinking an image by bilinear interpolation. The input to your program is the desired size of the resulting image in the horizontal and vertical direction. You may ignore aliasing effects.

Task 4
Implement the following transformation on your selected image and comment your observation:

- Negation transform.
- Logarithmic transform.
- Power transform.