Aim: - To implement changes in spatial and tonal resolution

Tools Used: - MatLab

Theory: -

Spatial Resolution:

1. Zooming: Zooming can be done in two ways
   1. Replication: - In replication we simply replicate each pixel and then replicate each row. Hence image of size n x n is zoomed to 2n x 2n. Zooming by replication gives the final image a patchy look since clusters of grey levels are formed. This can be substantially reduced by using a better method of zooming known as interpolation.
   2. Interpolation: - In this method instead of replicating each pixel, average of two adjacent pixels along the rows is taken and placed between two pixels. The same operation is then performed along the columns. The patchiness that was present in the replicated image is much less in the interpolated image.
2. Shrinking: Shrinking of image can be done by omitting alternate rows and then omitting alternate pixels in the selected rows.

Tonal Resolution:

Changing the number of gray levels: - The quality of a gray-level image is significantly affected by its gray-level resolution. In 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.

Algorithm: -

1. Zooming by Replication
   1. Read i/p image.
   2. Replicate each pixel
   3. Replicate each row
   4. Display o/p image
2. Zooming by Interpolation
   1. Read i/p image
   2. Average of two adjacent pixels along the rows is taken and placed between two pixels
   3. Do the same along columns
   4. Display o/p image
3. Shrinking
   1. Read i/p image.
   2. Copy alternate pixel of alternate row to o/p image
   3. Display o/p image
4. Changing the number of gray levels
   1. Read i/p image.
   2. Read each pixel value and convert it to binary 8 bit
   3. Replace a few (as needed) LSBs with Zeros
   4. Convert the new value to decimal
   5. Write the new decimal value to o/p image
   6. Display o/p image

Program:-

%Aim: To perform spatial and tonal gradation on image using Matlab

%Part-I: Image shrinking and Zooming

clc;

close all;

clear all;

a=imread('cameraman.tif');

[m,n]=size(a);

for i=1:m/2

for j=1:n/2

b(i,j)=a(2\*i,2\*j);

end

end

figure;

imshow(a);

title('Original Image');

gtext('Pranay Vora 51');

figure;

imshow(b);

title('Shrinking operation on Image');

gtext('Pranay Vora 51');

%image zooming using replication

for i=1:2\*m

for j=1:2\*n

p=i-floor(i/2);

q=j-floor(j/2);

c(i,j)=a(p,q);

end

end

figure;

imshow(c);

title('Replication Operation on Image');

gtext('Pranay Vora 51');

%image zooming using interpolation

for i=1:2\*m

for j=1:2\*n

p=i-floor(i/2);

q=j-floor(j/2);

d(i,j)=a(p,q);

j=j+1;

end

i=i+1;

j=1;

end

for i=1:2\*m

for j=2:2\*n-1

d(i,j)=[d(i,j-1)+d(i,j+1)]\*0.5;

end

end

for j=1:2\*n

for i=2:2\*m-1

d(i,j)=[d(i-1,j)+d(i+1,j)]\*0.5;

end

end

for i=2:2\*m-1

for j=2:2\*n-1

d(i,j)=[d(i-1,j+1)+d(i+1,j+1)]\*0.5;

end

end

figure;

imshow(d);

title('Zoomed Image using Interpolation method');

gtext('Pranay Vora 51');

%Tonal Gradation of Image.

A=imread('cameraman.tif');

[m,n]=size(A);

for i=1:m

for j=1:n

a=A(i,j);

b=dec2bin(a,8);

c=b(1:5);

d=b(1:2);

e=b(1);

B(i,j)=bin2dec(c);

C(i,j)=bin2dec(d);

D(i,j)=bin2dec(e);

end

end

imshow(A);

title('Original Image 8 bit');

gtext('Pranay Vora 51');

B=B\*16;

B=uint8(B);

figure,imshow(B);

title('Image 4 bits');

gtext('Pranay Vora 51');

C=C\*85;

C=uint8(C);

figure,imshow(C);

title('Image in terms of 2 bits');

gtext('Pranay Vora 51');

D=D\*255;

D=uint8(D);

figure,imshow(D);

title('Binary Image');

gtext('Pranay Vora 51');

**Output:**

|  |  |
| --- | --- |
|  |  |



|  |  |
| --- | --- |
|  |  |
|  |  |