|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Image Processing Practical #1 | | |
|  | | | |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Cours**  **e:** | Image Processing | | |
| **Topic** | Image Processing Basics | | |
|  | | | |
| **Pixel** | | | |
| Program to calculate number of samples required for an image  figure; m=4; n=6; N=400;  Fs=m\*N\*n\*N;  disp(Fs,'Number of samples required to preserve the information in the image='); | | | |
| **Image Properties** | | | |
| Program to access image properties – Dimension, height, width, number of channels, accessing and modifying any pixel | | | |

|  |
| --- |
|  |
| **Sampling** |
| Program to study the effects of reducing the spatial resolution of a digital image |
| **Quantization** |
| Program to study the effects of varying the number of intensity levels in a digital image  i=imread("("C:\Users\student\Downloads \IPCV\images\.png"); subplot(2,2,1);  imshow(i); title('original image') i=double(i); k1=(i\*255)/64;  subplot(2,2,2); k1=uint8(k1);  imshow(uint8(k1)); |

|  |
| --- |
| title('Quantization 64'); k2=(i\*255)/32;  subplot(2,2,3); k2=uint8(k2); imshow(uint8(k2)); title('Quantization 32'); k3=(i\*255)/16;  subplot(2,2,4); k3=uint8(k3); imshow(uint8(k3)); |
| **Image Addition** |
| Program to perform image addition for noise reduction.  figure;  i=imread("("C:\Users\student\Downloads \IPCV\images\.png"); i=imnoise(i,'salt & pepper',0.001);  subplot(1,2,1); imshow(i); title('noisy Image');  k=imadd(i,50); *//I +30 subplot*(1,2,2); imshow(k);  title('Smooth Image after performing image addition'); |

|  |
| --- |
|  |
| **Image Subtraction** |
| Program to compare images using subtraction for enhancing the difference between images  figure;  i=imread('C:\Program Files\scilab-6.0.2\IPCV\images\tool1.png'); j=imread('C:\Program Files\scilab-6.0.2\IPCV\images\tool2.png'); subplot(2,2,1);  imshow(i); title('Image 1');  subplot(2,2,2); imshow(j); title('Image 2'); k=imabsdiff(i,j); subplot(2,2,3); imshow(k);  title('Image 3 = Image 1 - Image 2'); |

|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Image Processing Practical #2 | | |
|  | | | |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21906A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Enhancement | | |
|  | | | |
| **Image Negative** | | | |
| **Code:**  figure();  i=imread("("C:\Users\student\Downloads \IPCV\images\balloons.png"); a=255-double(i);  a=uint8(a); subplot(1,2,1); imshow(i); title("Original Image"); subplot(1,2,2); imshow(a); title("Negative Image");  Output: | | | |
| **Thresholding** | | | |
| **Code:**  figure();  r=imread("("C:\Users\student\Downloads \IPCV\images\coins.png"); a=r;  [row col]=size(a); | | | |

|  |
| --- |
| T=input('Enter value of threshold:');  // here we took 100 for i=1:1:row  for j=1:1:col if(r(i,j)>=T)  a(i,j)=r(i,j); else  a(i,j)=0; end  end end  subplot(1,2,1); imshow(r); title('Original Image'); subplot(1,2,2); imshow(a);  title('Image obtained using threshold');  Output: |
| **Log Transformation** |
| **Code:**  r=imread("("C:\Users\student\Downloads \IPCV\images\baboon.png") c=25;  S=c\*log(1+double(r)); O=uint8(S); subplot(2,2,1); imshow(r) title("Original Image"); subplot(2,2,2); imshow(O);  title("Log Transformation") |

|  |
| --- |
| Output: |
| **Piece-Wise Linear Transformation** |
| **Code:**  a=imread("("C:\Users\student\Downloads \IPCV\images\peppers.png") subplot(2,2,1);  imshow(a); title("Original Image") r=double(a)/255;  c=1;  gamma=8.3; s=c\*(r).^gamma; subplot(2,2,2); imshow(s); title("Gamma Image") **Output:** |
| **Gray Level Slicing** |

|  |
| --- |
| **Code:**  Without Background  a=imread("("C:\Users\student\Downloads \IPCV\images\peppers.png"); [r,c]=size(a);  for i=1:r for j=1:c  if (a(i,j)>50 & a(i,j)<100) x(i,j)=255;  else x(i,j)=0; end end end  x=uint8(x); subplot(2,2,1) imshow(a); title('Original Image'); subplot(2,2,2); imshow(x);  Output:  Inserting image...  With Background  a=imread(("C:\Users\student\Downloads \IPCV\images\coins.png"); [r,c]=size(a);  for i=1:r for j=1:c  if (a(i,j)>60 & a(i,j)<110) x(i,j)=255;  else  x(i,j)=a(i,j); *// if without background use x(i,j)=0 end*  *end end*  *x*=uint8(x); subplot(2,2,1) imshow(a);  title('Original Image'); |

|  |
| --- |
| subplot(2,2,2); imshow(x);  title('Grey level slicing without background');  Output: |
| **Bit Plane Slicing** |
| **Code:**  f=imread("C:\Users\student\Downloads \IPCV\images\lena.png"); f=double(f);  [r,c]=size(f);m=1; for i=1:r  for j=1:c MSB(i,j)=bitand(f(i,j),128);  seven(i,j)=bitand(f(i,j),64);  sex(i,j)=bitand(f(i,j),32);  five(i,j)=bitand(f(i,j),16);  four(i,j)=bitand(f(i,j),8);  three(i,j)=bitand(f(i,j),4);  two(i,j)=bitand(f(i,j),2);  LSB(i,j)=bitand(f(i,j),1); end  end  subplot(2,2,1);imshow(MSB);title(" MSB");  subplot(2,2,2);imshow(seven);title("7th");  subplot(2,2,3);imshow(six);title("6th");  subplot(2,2,4);imshow(five);title("5th");  subplot(2,2,5);imshow(four);title("4th");  subplot(2,2,6);imshow(three);title("3rd");  subplot(2,2,7);imshow(two);title("2nd");  subplot(2,2,8);imshow(LSB);title("1st");  subplot(2,2,9);imshow(f);title("original");  Output: |

|  |
| --- |
| Original: |

|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Image Processing Practical #3 | | |
|  | | | |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Cours**  **e:** | Image Processing | | |
| **Topic** | Histogram Processing | | |
|  | | | |
| **Plot Histogram of an Image** | | | |

|  |
| --- |
| clear; a=imread("C:\Users\student\Downloads\seed.tif"); subplot(2,2,1);  imshow(a); [r,c]=size(a); h=zeros(1,256); s=0:255  for i=1:r 3  for j=1:c 3 if (a(i,j)==0)  a(i,j)=1; end k=a(i,j);  h(k)=h(k)+1; end  end subplot(2,2,2); plot(h);  title('Histogram of the Image')  **Output:** |
| **Plot Histogram of Low Contrast, Bright , dark and High Contrast Images** |
| clear; a=imread("C:\Users\student\Downloads\seed.tif"); subplot(4,2,1);  title("Original Image") imshow(a); subplot(4,2,2); h1=imhist(a); plot(h1); subplot(4,2,3); a=a+100;  title("hight Contrast Image") imshow(a);  subplot(4,2,4) |

|  |
| --- |
| h2=imhist(a); plot(h2); subplot(4,2,5); a=a-180;  title("low Contrast Image") imshow(a);  subplot(4,2,6) h3=imhist(a); plot(h3);  mmin =min(a(:));  mmax = max(a(:));  lmin=0;lmax=255; *//p1=(p-mmin)\*(255/(mmax-mmin));* a=(a-mmin)\*((lmax-lmin)/(mmax-mmin))+lmin; subplot(4,2,8);  h4=imhist(a); plot(h4); subplot(4,2,7);  title(""); imshow(a);  **Output:** |

|  |
| --- |
| **Histogram Equalization** |
| clc; clear;  a=imread("C:\Program Files\scilab-6.0.2\IPCV\images\seed.tif"); [r,c]=size(a);  subplot(2,2,1); imshow(a) title("Original Image") subplot(2,2,2); h=imhist(a); plot(h);  title("Histogram of Original Image") a=double(a);  big=max(max(a)); // largest intensity present in the image tot = r\*c; // total no of pixels in the image z=zeros(1,256); //empty array for hostogram pdf=h/tot; // probabilty of occurance of intensity cdf(1)=pdf(1); //first cumulative addition  for i=2:1:big cdf(i)=pdf(i)+cdf(i-1); end new=round(cdf\*big); for i=1:1:r  for j=1:1:c temp=a(i,j); if(temp==0) temp=1; end b(i,j)=new(temp); end  end b=uint8(b); subplot(2,2,4); h9 =imhist(b) plot(h9);  title('Equalized Histogram'); subplot(2,2,3) imshow(uint8(b)) title('Equalized mage');  **OUTPUT:** |

|  |
| --- |
|  |

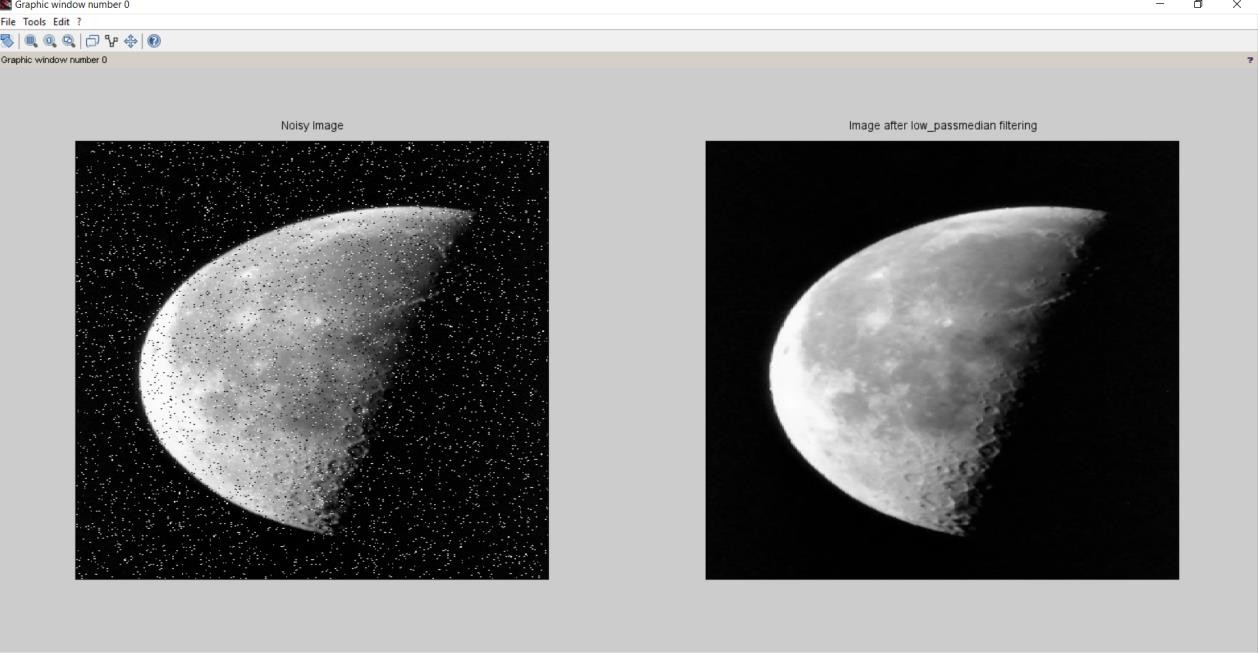
|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Image Processing Practical #4 | | |
|  | | | |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Smoothing in Spatial Domain | | |
|  | | | |
| **Averaging** | | | |

|  |
| --- |
| clc; clear;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); subplot(1,3,1);  imshow(a);  c=imnoise(a,'gaussian',0.3); d=double(c);  b=d; *//backup*  m=(1/9)\*(ones(3,3)); *//[1,1,1; 1,1,1; 1,1,1;]*  [r1,c1]=size(a); for i=2:r1-1  for j=2:c1-1  a1=d(i-1,j-1)+d(i-1,j)+d(i-1,j+1)+d(i,j-  1)+d(i,j)+d(i,j+1)+d(i+1,j)+d(i+1,j)+d(i+1,j+1); b(i,j)=a1\*(1/9);  end end  subplot(1,3,1); imshow(a); title('Original Image'); subplot(1,3,2) imshow(uint8(b)); title ('Filtered Image'); subplot(1,3,3); imshow(uint8(c)); title('noisy image') |

|  |
| --- |
|  |
| **Min Filter** |
| clc; clear;  p1=imread("C:\Program Files\scilab-6.1.1\IPCV\images\moon.tif"); pmin=imnoise(p1,'salt & pepper');  *//d=double(c);*  *//b=d; //backup*  *//m=(1/9)\*(ones(3,3)); //[1,1,1; 1,1,1; 1,1,1;]*  figure; subplot(1,2,1); imshow(pmin); title('Noisy Image'); [r1,c1]=size(pmin); for i=2:1:r1-1  for j=2:1:c1-1  a1=[pmin(i-1,j-1) pmin(i-1,j) pmin(i-1,j+1) pmin(i,j-1) pmin(i,j) pmin(i,j+1) pmin(i+1,j-1) pmin(i+1,j) pmin(i+1,j+1)]; a2=min(min(a1));  newmin(i,j)=a2;  end end  subplot(1,2,2); imshow(newmin); |

|  |
| --- |
| title('Image after min filtering'); |
| **Max Filter** |
| clc; clear; figure;  p1=imread("C:\Program Files\scilab-6.1.1\IPCV\images\moon.tif"); pmax=imnoise(p1,'salt & pepper');  *//d=double(c);*  *//b=d; //backup*  *//m=(1/9)\*(ones(3,3)); //[1,1,1; 1,1,1; 1,1,1;]*  figure; subplot(1,2,1); imshow(pmax); title('Noisy Image'); [r1,c1]=size(pmax); for i=2:1:r1-1  for j=2:1:c1-1  a1=[pmax(i-1,j-1) pmax(i-1,j) pmax(i-1,j+1) pmax(i,j-1) pmax(i+j) pmax(i,j+1) pmax(i+1,j-1) pmax(i+1,j) pmax(i+1,j+1)]; a2=max(max(a1));  newmax(i,j)=a2; end |

|  |
| --- |
| end subplot(1,2,2); imshow(newmax);  title('Image after max filtering'); |
| **Median Filter** |
| clc; clear;  p1=imread("C:\Program Files\scilab-6.1.1\IPCV\images\moon.tif"); pmed=imnoise(p1,'salt & pepper');  *//d=double(c);*  *//b=d; //backup*  *//m=(1/9)\*(ones(3,3)); //[1,1,1; 1,1,1; 1,1,1;]*  figure; subplot(1,2,1); imshow(pmed); title('Noisy Image'); [r1,c1]=size(pmed);  for i=2:1:r1-1 for j=2:1:c1-1  a1=[pmed(i-1,j-1) pmed(i-1,j) pmed(i-1,j+1) pmed(i,j-1) pmed(i,j) pmed(i,j+1) pmed(i+1,j-1) pmed(i+1,j) pmed(i+1,j+1)]; a2=gsort(a1);  med=a2(5); |



newmed(i,j)=med;

end end

subplot(1,2,2); imshow(uint8(newmed));

title('Image after low\_passmedian filtering');

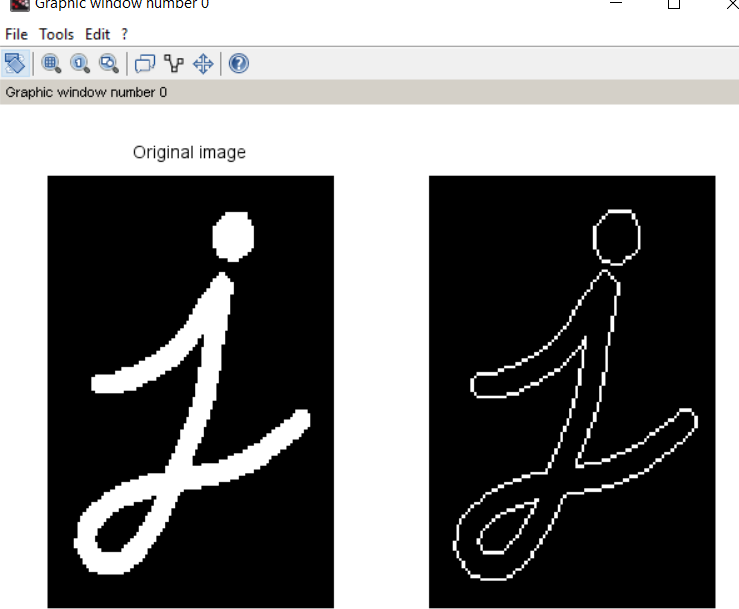
Image Processing Practical #5

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Sharpening in Spatial Domain | | |

**First Order Derivative Filter**

|  |
| --- |
| clear;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\morpex.png"); subplot(5,2,1)  imshow(a); d=double(a);  v=[1 0 -1;2 0 -2;1 0 -1];  h=[-1 -2 -1;0 0 0;1 2 1];  [r1,c1]=size(a); for i=2:(r1-1)  for j=2:(c1-1)  news(i,j) =(v(1)\*d(i-1,j-1))+(v(2)\*d(i-1,j))+(v(3)\*d(i-1,j+1))+(v(4)\*d(i,j-  1))+(v(5)\*d(i,j))+(v(6)\*d(i,j+1))+(v(7)\*d(i+1,j-1))+(v(8)\*d(i+1,j))+(v(9)\*d(i+1,j+1));  end end  subplot(5,2,2) imshow(uint8(news)) for i=2: (r1-1)  for j=2: (c1-1)  newh(i,j) =(h(1)\*d(i-1,j-1))+(h(2)\*d(i-1,j))+(h(3)\*d(i-1,j+1))+(h(4)\*d(i,j-  1))+(h(5)\*d(i,j))+(h(6)\*d(i,j+1))+(h(7)\*d(i+1,j-1))+(h(8)\*d(i+1,j))+(h(9)\*d(i+1,j+1));  end end  subplot(5,2,3) imshow(uint8(newh)) for i=2:(r1-1)  for j=2:(c1-1)  newvr(i,j) =(v(1)\*-1\*d(i-1,j-1))+(v(2)\*-1\*d(i-1,j))+(v(3)\*-1\*d(i-1,j+1))+(v(4)\*-  1\*d(i,j-1))+(v(5)\*-1\*d(i,j))+(v(6)\*-1\*d(i,j+1))+(v(7)\*-1\*d(i+1,j-1))+(v(8)\*-  1\*d(i+1,j))+(v(9)\*-1\*d(i+1,j+1)); end  end subplot(5,2,4)  imshow(uint8(newvr)) for i=2: (r1-1)  for j=2: (c1-1)  newrh(i,j) =(h(1)\*-1\*d(i-1,j-1))+(h(2)\*-1\*d(i-1,j))+(h(3)\*-1\*d(i-1,j+1))+(h(4)\*-  1\*d(i,j-1))+(h(5)\*-1\*d(i,j))+(h(6)\*-1\*d(i,j+1))+(h(7)\*-1\*d(i+1,j-1))+(h(8)\*-  1\*d(i+1,j))+(h(9)\*-1\*d(i+1,j+1)); end  end  subplot(5,2,5) |

|  |
| --- |
| imshow(uint8(newrh)); final=uint8(news)|uint8(newh)|uint8(newvr)|uint8(newrh); subplot(5,2,6)  imshow(uint8(final))  **Output:** |
| **Second Order Derivative – Laplacian Filter** |
| clear;  p=imread(' C:\Program Files\scilab-6.0.1\IPCV\images \morpex.png') subplot(1,2,1)  title('Original image') imshow(p) d=double(p)  v=[0 1 0;1 -4 1;0 1 0];  [r1,c1]=size(p);  for i=2:1:r1-1 for j=2:1:c1-1  newv(i,j)=(v(1)\*d(i-1,j-1))+(v(2)\*d(i-1,j))+(v(3)\*d(i-1,j+1))+(v(4)\*d(i,j-  1))+(v(5)\*d(i,j))+(v(6)\*d(i,j+1))+(v(7)\*d(i+1,j-1))+(v(8)\*d(i+1,j))+(v(9)\*d(i+1,j+1))  end end |



subplot(1,2,2)

imshow(newv)

**Output:**

Image Processing Practical #6

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Smoothing in Frequency Domain | | |

**Ideal Low Pass Filter**

|  |
| --- |
| clc;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);  [r,c]=size(a1);  d=input("Enter the cut off frequency"); for u=1:r  for v= 1:c  d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5  if d1<=d  h(u,v)=1; else  h(u,v)=0;  end end end  b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1));  Output: |
| **Butterworth Low Pass Filter** |
| clc;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);  [r,c]=size(a1);  d=input("Enter the cut off frequency"); n=input("Enter exponential");  for u=1:r  for v= 1:c  d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5;  h(u,v)=1/(1+(d1/d)^(2\*n));  end end |

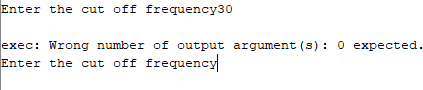


|  |
| --- |
| b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1)); Output: |
| **Gaussian Low Pass Filter** |
| clc;  a=imread(" C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);  [r,c]=size(a1);  d=input("Enter the cut off frequency"); for u=1:r  for v= 1:c  d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5  h(u,v)=exp(-(d1\*d1)/(2\*d\*d));  end end  b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1));  Output: |



Image Processing Practical #7

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Sharpening in Frequency Domain | | |



clc;

a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);

[r,c]=size(a1);

d=input("Enter the cut off frequency"); for u=1:r

for v= 1:c

d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5

if d1<=d

h(u,v)=0; else

h(u,v)=1;

end end end

b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1));

Output:

**Ideal High Pass Filter**

|  |
| --- |
|  |
| **Butterworth High Pass Filter** |
| clc;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);  [r,c]=size(a1);  d=input("Enter the cut off frequency"); n=input("Enter exponential");  for u=1:r  for v= 1:c  d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5;  h(u,v)=1/(1+(d/d1)^(2\*n));  end end  b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1)); Output: |

|  |
| --- |
|  |
| **Gaussian High Pass Filter** |
| clc;  a=imread(" C:\Program Files\scilab-6.0.1\IPCV\images\camera man.png"); a1=double(a);  [r,c]=size(a1);  d=input("Enter the cut off frequency"); for u=1:r  for v= 1:c  d1=(((u-(r/2))^2)+ ((v-(c/2))^2))^0.5  h(u,v)=1- exp(-(d1\*d1)/(2\*d\*d));  end end  b=fft2(a1); c=fftshift(b); new=h.\*c; new1=abs(ifft(new)); subplot(2,2,1); imshow(uint8(a1)); subplot(2,2,2); imshow(uint8(new1));  Output: |

# A picture containing drawing, stop, room Description automatically generated

Image Processing Practical #8

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Restoration | | |

**Arithmetic Mean Filter to Remove Gaussian Noise**

Code:

clc;

clear all;

a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); figure;

c=imnoise(a,'gaussian'); d=double(c);

b=d; *//backup [*r1,c1]=size(a); for i=2:r1-1

for j=2:c1-1

a1=d(i-1,j-1)+d(i-1,j)+d(i-1,j+1)+d(i,j-1)+d(i,j)+d(i,j+1)+d(i+1,j-1)+d(i+1,j)+d(i+1,j+1); p=a1\*(1/9);

b(i,j)=p; end

end subplot(1,3,1); imshow(a);

title('Original Image'); subplot(1,3,2); imshow(c);

title('Noisy Image'); imshow(uint8(c)); subplot(1,3,3)

title("Arithmethic Mean Filter") imshow(uint8(b));

Output:

|  |
| --- |
|  |
| **Geometric Mean Filter to Remove Gaussian Noise** |
| Code:  clc;  clear all;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); figure;  c=imnoise(a,'gaussian'); d=double(c);  b=d; *//backup [*r1,c1]=size(a); for i=2:r1-1  for j=2:c1-1  a1=d(i-1,j-1)\*d(i-1,j)\*d(i-1,j+1)\*d(i,j-1)\*d(i,j)\*d(i,j+1)\*d(i+1,j-1)\*d(i+1,j)\*d(i+1,j+1); p=a1.^(1/9);  b(i,j)=p; end  end subplot(1,3,1); imshow(a);  title('Original Image'); subplot(1,3,2); imshow(c);  title('Noisy Image'); imshow(uint8(c)); subplot(1,3,3) title("Geometric Mean Filter") imshow(uint8(b));  Output: |

|  |
| --- |
|  |
| **Harmonic Mean Filter to remove Salt & Pepper Noise** |
| Code:  clc;  clear all;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); figure;  c=imnoise(a,'gaussian'); d=double(c);  b=d; *//backup [*r1,c1]=size(a); for i=2:r1-1  for j=2:c1-1  a1=1/d(i-1,j-1)+1/d(i-1,j)+1/d(i-1,j+1)+1/d(i,j-1)+1/d(i,j)+1/d(i,j+1)+1/d(i+1,j- 1)+1/d(i+1,j)+1/d(i+1,j+1);  b(i,j)=9/a1; end  end subplot(1,3,1); imshow(a);  title('Original Image'); subplot(1,3,2); imshow(c);  title('Noisy Image'); imshow(uint8(c)); subplot(1,3,3) title('Harmonic Mean Filter') imshow(uint8(b));  Output: |

|  |
| --- |
|  |
| **Contra Harmonic Mean Filter to Remove Salt & Pepper Noise** |
| Code:  clc;  clear all;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); figure;  q=input("enter the q"); c=imnoise(a,'salt & pepper',0.007); d=double(c);  b=d; *//backup [*r1,c1]=size(a); for i=2:r1-1  for j=2:c1-1  a1=d(i-1,j-1)^(q+1)+d(i-1,j)^(q+1)+d(i-1,j+1)^(q+1)+d(i,j-  1)^(q+1)+d(i,j)^(q+1)+d(i,j+1)^(q+1)+d(i+1,j-1)^(q+1)+d(i+1,j)^(q+1)+d(i+1,j+1)^(q+1);  a2=d(i-1,j-1)^(q)+d(i-1,j)^(q)+d(i-1,j+1)^(q)+d(i,j-1)^(q)+d(i,j)^(q)+d(i,j+1)^(q)+d(i+1,j- 1)^(q)+d(i+1,j)^(q)+d(i+1,j+1)^(q);  b(i,j)=a1/a2; end  end subplot(1,3,1); imshow(a);  title('Original Image'); subplot(1,3,2); imshow(c);  title('Noisy Image'); imshow(uint8(c)); subplot(1,3,3) title('Harmonic Mean Filter') imshow(uint8(b));  Output: |

|  |
| --- |
|  |
| **Mid Point Filter to Remove Salt & Pepper Noise** |
| clc;  clear all;  a=imread("C:\Program Files\scilab-6.0.1\IPCV\images\moon.tif"); figure;  c=imnoise(a,'salt & pepper',0.008); d=double(c);  b=d; *//backup [*r1,c1]=size(a); for i=2:r1-1  for j=2:c1-1  a1=[d(i-1,j-1) d(i-1,j) d(i-1,j+1) d(i,j-1) d(i,j) d(i,j+1) d(i+1,j-1) d(i+1,j) d(i+1,j+1)]; a2=gsort(a1);  maxi=a2(9); mini=a2(1); mid=(1/2)\*(mini+maxi); b(i,j)=mid;  end end  b=uint8(b); subplot(1,3,1); imshow(a); title('Original Image'); subplot(1,3,2); imshow(c);  title('Noisy Image');  imshow(uint8(c)); subplot(1,3,3) |

|  |
| --- |
| title('MidPoint Filter') imshow(uint8(b));  Output: |



Image Processing Practical #9

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Color Image Processing | | |

**Display different color planes and plot histogram of color image**

figure;

a=imread("("C:\Users\student\Downloads \IPCV\images\peppers.png"); R=a(:,:,1);

G=a(:,:,2);

B=a(:,:,3);

subplot(3,2,1); imshow(a); title("Origninal Image"); subplot(3,2,2); imshow(R);

title("Red Component");

|  |
| --- |
| subplot(3,2,3); imshow(G);  title("Green Component"); subplot(3,2,4); imshow(B);  title("Blue component");  [yR,x]=imhist(R);  [yG,x]=imhist(G);  [yB,x]=imhist(B);  subplot(3,2,5); plot(x,yR,x,yG,x,yB,"Linewidth",2);  //subplot(3,2,5); xlabel("RGB Intensity"); ylabel("No.of Pixels"); set(gca(),"grid",[1,1])  Output: |
| **Convert RGB image to HSV and CMY model** |
| figure;  a=imread("("C:\Users\student\Downloads \IPCV\images\peppers.png");  HSV=rgb2hsv(a); subplot(2,1,1 title("RGB2HSV") imshow(HSV);  CMY=imcomplement(a); subplot(2,1,2); |

|  |
| --- |
| title("RGB2CMY") imshow(CMY);  Output: |
| **Color Image Smoothing** |
| clc;clear all;  I=imread("C:\Program Files\scilab-6.0.1\IPCV\images\peppers.png"); F = fspecial('average',11);  pic(:,:,1) = imfilter(I(:,:,1),F);  pic(:,:,2) = imfilter(I(:,:,2),F);  pic(:,:,3) = imfilter(I(:,:,3),F); Average\_Image = uint8(pic); subplot(2,2,1);  imshow(I); title('Original Image'); subplot(2,2,2); imshow(Average\_Image);  title('Average Filtered Image'); I=imnoise(I,'gaussian',0.07);  F1 = fspecial('gaussian',3);  pic1(:,:,1) = imfilter(I(:,:,1),F1);pic1(:,:,2) = imfilter(I(:,:,2),F1); pic1(:,:,3) = imfilter(I(:,:,3),F1);  GLPF\_Image = uint8(pic1); subplot(2,2,3); imshow(I);  title('Noisy Image(Gaussian Noise)'); subplot(2,2,4);  imshow(GLPF\_Image); title('GLPF Filtered Image')  Output:- |

|  |
| --- |
|  |
| **Color Image Sharpening** |
| clc;clear all;  I=imread("C:\Program Files\scilab-6.0.1\IPCV\images\peppers.png"); F=fspecial('prewitt');  pic(:,:,1) = imfilter(I(:,:,1),F);  pic(:,:,2) = imfilter(I(:,:,2),F);  pic(:,:,3) = imfilter(I(:,:,3),F); Sharp\_Image = uint8(pic);s ubplot(1,2,1);  imshow(I); title('Original Image'); subplot(1,2,2); imshow(Sharp\_Image);  title('Sharp Filtered Image')  **Output:-** |

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Image Processing Practical #10 | | |
|  | | | |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Morphological Image Processing | | |
|  | | | |
| **Erosion** | | | |

|  |
| --- |
| Code:-  *//Program to apply erosion clc*;  clear all; figure;  a=imread('C:\Program Files\scilab-6.0.1\IPCV\images\morpex.png'); d=a;  [r,c]=size(d); m=ones(3,3);  for i=2:1:r-1  for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1))  (m(5)\*d(i,j)) (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A1(i,j)=min(new);  end end  subplot(1,2,1); imshow(a); title('Original Image');  subplot(1,2,2); imshow(A1); title('Erosion Image');  Output:- |

|  |
| --- |
|  |
| **dilation** |
| Code:-  *//Program to apply dilation clc*;  clear all; figure;  a=imread('C:\Program Files\scilab-6.0.1\IPCV\images\morpex.png'); d=a;  [r,c]=size(d); m=ones(3,3);  for i=2:1:r-1  for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1))  (m(5)\*d(i,j)) (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A1(i,j)=max(new);  end end  subplot(1,2,1); imshow(a); title('Original Image');  subplot(1,2,2); |

|  |
| --- |
| imshow(A1); title('Dilation Image');  **Output:** |
| **Opening** |
| figure;  a=imread('C:\Program Files\scilab-6.0.1\IPCV\images\line.tif'); d=a;  [r,c]=size(d); m=ones(3,3);  for i=2:1:r-1 for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1)) (m(5)\*d(i,j))  (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A1(i,j)=min(new);  end end d=A1;  [r,c]=size(d);  for i=2:1:r-1 for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1)) (m(5)\*d(i,j))  (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A2(i,j)=max(new); |

|  |
| --- |
| end end  subplot(1,2,1); imshow(a); title('Original Image'); subplot(1,2,2); imshow(A2);  title('processed image - o'); Output:- |
| **Closing** |
| *clc*;  clear all; figure;  a=imread('C:\Program Files\scilab-6.0.1\IPCV\images\circbw.png'); d=a;  [r,c]=size(d); m=ones(3,3);  for i=2:1:r-1 for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1)) (m(5)\*d(i,j))  (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A1(i,j)=max(new);  end end d=A1;  [r,c]=size(d);  for i=2:1:r-1 |

|  |
| --- |
| for j=2:1:c-1  new=[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1)) (m(5)\*d(i,j))  (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A2(i,j)=min(new);  end end  subplot(1,2,1); imshow(a); title('Original Image'); subplot(1,2,2); imshow(A2);  title('processed image - o');  Output:- |
| **Boundary Extraction** |
| clc; clear  a= imread('C:\Program Files\scilab-6.0.1\IPCV\images\boundary.tif'); d=a;  [r,c]=size(d); m= ones(3,3)  for i=2:1:r-1 for j=2:1:c-1 |

|  |
| --- |
| new =[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1))  (m(5)\*d(i,j)) (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))]; A1(i,j)=min(new);  aa(i,j)= d(i,j) - A1(i,j) end  end  subplot(1,2,1) imshow(a); title("Original Image")  subplot(1,2,2) imshow(aa)  title("Processed Image Boundary Extraction") |
| **Morphological Gradient** |
| clc;  Clear  a= imread('C:\Program Files\scilab-6.0.1\IPCV\images\boundary.tif'); d=a;  [r,c]=size(d); m= ones(3,3) for i=2:1:r-1  for j=2:1:c-1 |

|  |
| --- |
| new =[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1)) (m(5)\*d(i,j))  (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))] A1(i,j)=max(new);  End end d=A1;  [r,c]= size(d) for i=2:1:r-1 for j=2:1:c-1  new =[(m(1)\*d(i-1,j-1)) (m(2)\*d(i-1,j)) (m(3)\*d(i-1,j+1)) (m(4)\*d(i,j-1))  (m(5)\*d(i,j)) (m(6)\*d(i,j+1)) (m(7)\*d(i+1,j-1)) (m(8)\*d(i+1,j)) (m(9)\*d(i+1,j+1))] A2(i,j)=min(new);  aa(i,j)= A1(i,j)-A2(i,j) end  end subplot(1,2,1) imshow(a);  title("Original Image") subplot(1,2,2) imshow(aa)  title("Processed Image Gradient") |



Image Processing Practical #11

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Harshal Kadam | **Roll**  **Number** | 21306A1024 |
| **Subject/Course:** | Image Processing | | |
| **Topic** | Image Segmentation | | |

**Edge Detection – Sobel operator**

clc;

clear all;

p=imread('C:\Program Files\scilab-6.0.1\IPCV\images\edge2.tif'); figure;

subplot(3,2,1); imshow(p); title('Original Image');

v1=[1,0,-1;2, 0, -2;1, 0, -1]; *//x-direction*

*h1*=[-1, -2, -1; 0, 0 ,0; 1, 2, 1 ]; *//y-direction*

*v2*=[-1, 0, 1; -2, 0, 2; -1, 0, 1];

h2=[1, 2, 1; 0, 0, 0; -1, -2 ,-1];

gv1=abs(imfilter(double(p),v1)); subplot(3,2,2);

imshow(gv1); title('Verical Edges');

gh2=abs(imfilter(double(p),h2)); gh1=abs(imfilter(double(p),h1)); subplot(3,2,3);

imshow(gh1); subplot(3,2,4); imshow(gh2); title('horizontal Edges');

gv2=abs(imfilter(double(p),v2)); subplot(3,2,5);

imshow(gv2); title('vertical2 Edges'); finaledge=gh1+gv2+gh2+gv1; subplot(3,2,6); imshow(finaledge); title('Final Image');

**Output:**

|  |
| --- |
|  |
| **Edge Detection – Canny Algorithm** |
| clc;  clear all;  p=imread('C:\Program Files\scilab-6.0.1\IPCV\images\edge2.tif'); figure;  subplot(2,2,1); imshow(p); title('Original Image'); d=double(p); thresh=0.02;  sigma=3; *// 3, 5, 7*  *E*=edge(d, 'canny', thresh, sigma); subplot(2,2,2);  imshow(E);  title('Canny Edge Image');  Output: |