

Informe laboratorio - Procesamiento de Imágenes Digitales

Filtrado espacial lineal y no lineal

Ruben Rodriguez

30 de julio de 2022

Índice

1. Desarrollo	1
2. Anexo (CODIGO)	11

1. Desarrollo

1. Crear una función en Matlab que reciba dos parámetros: una imagen, una máscara. Esta será una función genérica y deberá ser usada para crear todos los algoritmos de filtrado de este laboratorio, en algunos casos con las adiciones de parámetros necesarias.

```
1  function [filteredImage] = filter(image,filterType,filterSize,paddingType)
2
3      if isa(filterType,'char')
4          switch filterType
5              case 'min'
6                  myfilter = @(x) min(x,[],"all");
7              case 'max'
8                  myfilter = @(x) max(x,[],"all");
9              case 'mean'
10                 myfilter = @(x) round(mean2(x),0);
11             case 'median'
12                 myfilter = @(x) median(x,"all");
13         end
14     else
```

```
1     den = sum(filterType,"all");
2     if den == 0
3         den = 1;
4     end
5     myfilter = @(x) round(sum(filterType.*double(x),"all")/den,0);
6 end
7
8 filtersize = filterSize;
9 imageWithPadding = padding(image,filtersize,paddingType);
10 filteredImage = image;
11
12 [sizeY,sizeX,sizeZ] = size(image);
13
14 for z=1:sizeZ
15     for y=1:sizeY
16         for x=1:sizeX
17             Y=y+(filtersize-1)/2;
18             X=x+(filtersize-1)/2;
19             Y1=Y-(filtersize-1)/2;
20             X1=X-(filtersize-1)/2;
21             Y2=Y+(filtersize-1)/2;
22             X2=X+(filtersize-1)/2;
23
24             area = imageWithPadding(Y1:Y2,X1:X2,z);
25
26             newValue =myfilter(area);
27             filteredImage(y,x,z) = newValue;
28         end
29     end
30 end
31 end
```

2. Crear una función que complete los bordes de la imagen con ceros, replicando el último píxel del borde.

```
1 function [imageWithPadding] = padding(image, filterSize, ceros)
2
3     sizeYfilter = filterSize;
4     [sizeY,sizeX,~] = size(image); % image size
5
6     paddingSize = (sizeYfilter-1)/2; % calculated padding size
7     imageWithPadding = padarray(image,[paddingSize,paddingSize],0,'both');
8     % image with padding of zeros
9     [sizeYpadding, sizeXpadding, ~] = size(imageWithPadding); % image size
10    % with padding
11
12    if ~ceros
13        % Y
14        imageWithPadding(paddingSize+1:sizeYpadding-paddingSize ...
15            ,1:paddingSize,:) = repmat(image(1:sizeY,1,:),1,paddingSize);
16
17        imageWithPadding( ...
18            paddingSize+1:(sizeYpadding-paddingSize), ...
19            (sizeX+paddingSize+1):sizeXpadding,:) = repmat( ...
20                image(1:sizeY,sizeX,:),1,paddingSize);
21
22        % X
23        imageWithPadding(1:paddingSize,1:sizeXpadding,:) = repmat( ...
24            imageWithPadding((paddingSize+1),1:sizeXpadding,:),paddingSize,1);
25
26        imageWithPadding((sizeY+paddingSize+1):sizeYpadding, ...
27            1:sizeXpadding,:) = repmat( ...
28                imageWithPadding(sizeY+1,1:sizeXpadding,:),paddingSize,1);
29    end
30 end
```

3. Aplicar el filtro de media a la imagen a.png, probar con diferentes tamaños de máscara 3, 5, 9, 15 y 35.

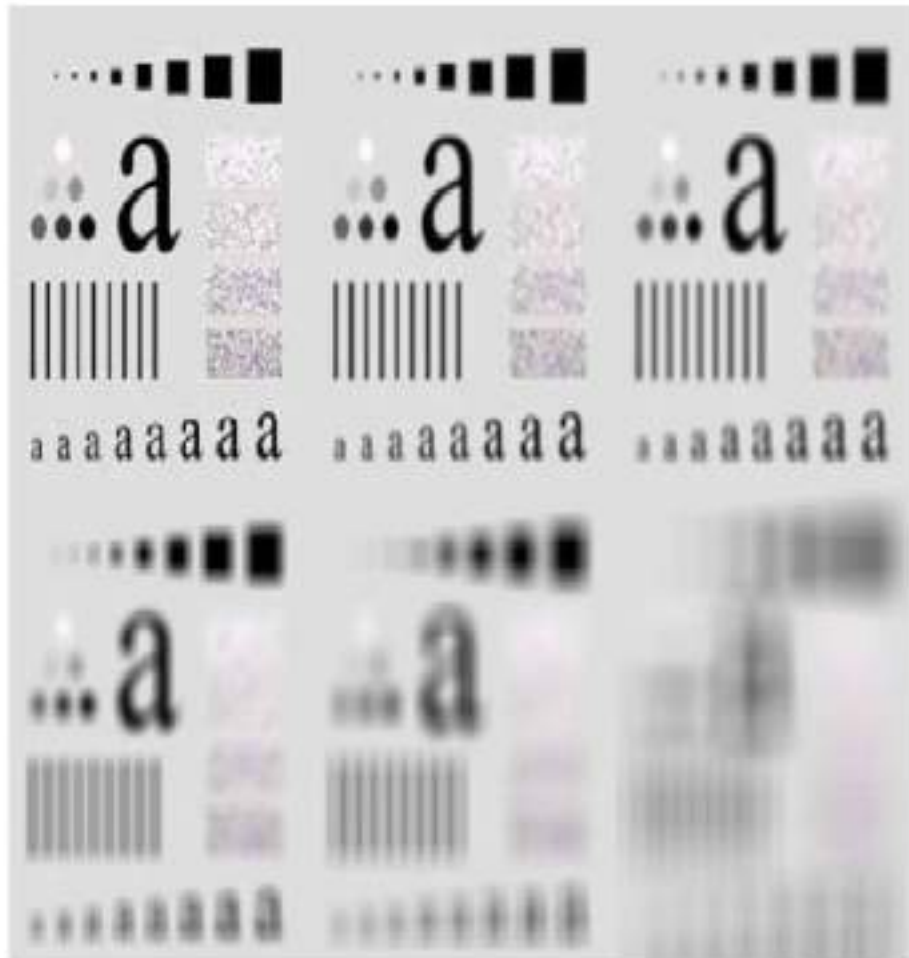


Figura 1: a.png original y con filtros de la media:3,5,9,15,35.

La imagen se difumina a medida que el tamaño del filtro aumenta.

4. Eliminar el ruido de sal y pimienta en la imagen circuit.png utilizando el filtro no lineal de la mediana (comparar el filtrado de mediana y el filtrado de media).

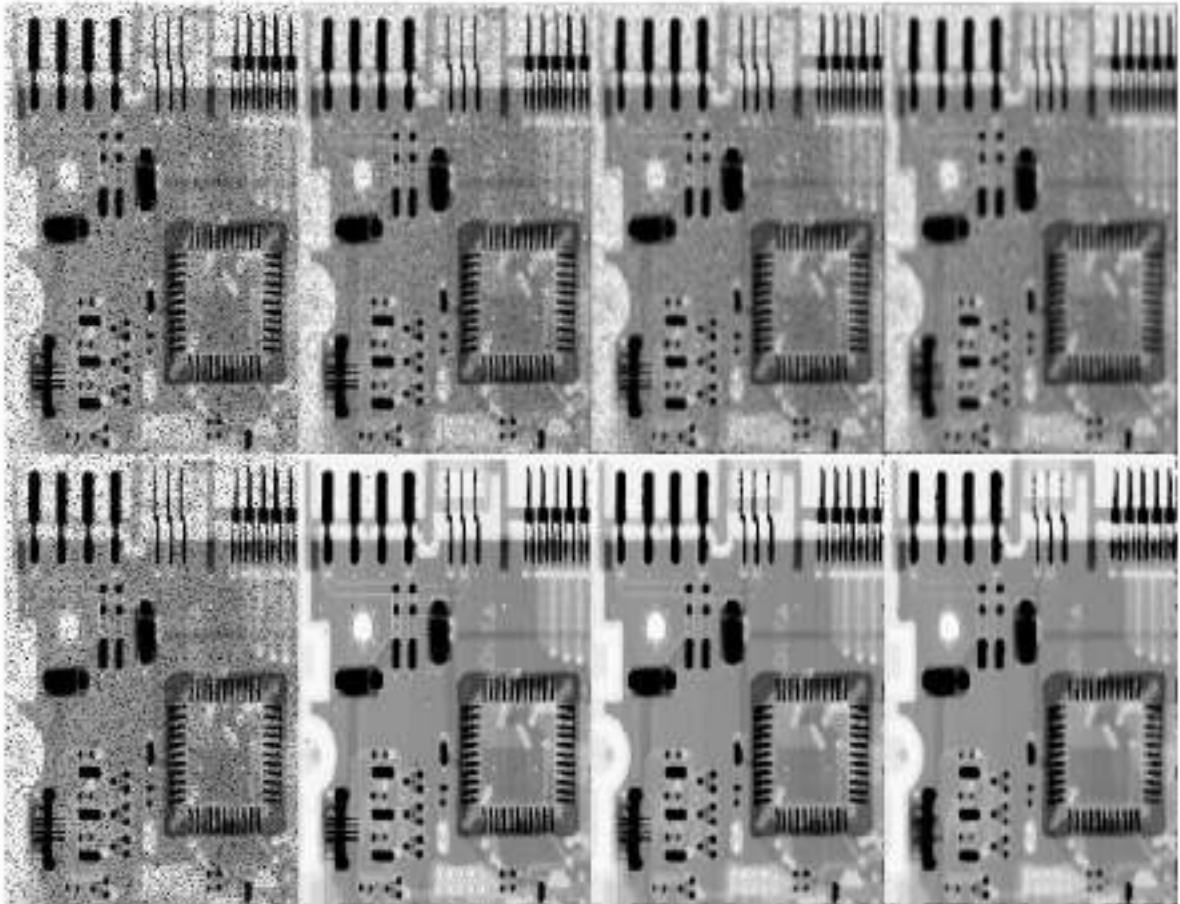


Figura 2: imagen original, media, mediana. Tamaño de filtro:3,5,7.

Ambos filtros eliminan el ruido, pero a diferencia del filtro de la media, el filtro de la mediana muestra claridad y calidad en la imagen.

5. A partir de la función generica del punto 1 aplicar el filtrado mediano, maximo y minimo a la imagen green.png.
- a) Comparar filtrado de mediana con el filtrado de media. ¿Cuál obtiene mejor resultado? y ¿por qué?



Figura 3: media y mediana. Tamaño de filtro:3,5,7.

En el filtro de tamaño 7x7 la mediana logra eliminar de mejor manera el hilo sobre las hojas. Debido a que la mediana casi siempre se va a encontrar mas hacia los verdes.

- b) Comparar filtrado de mediana con el filtrado máximo y mínimo. ¿Cuál obtiene mejor resultado? y ¿por qué?

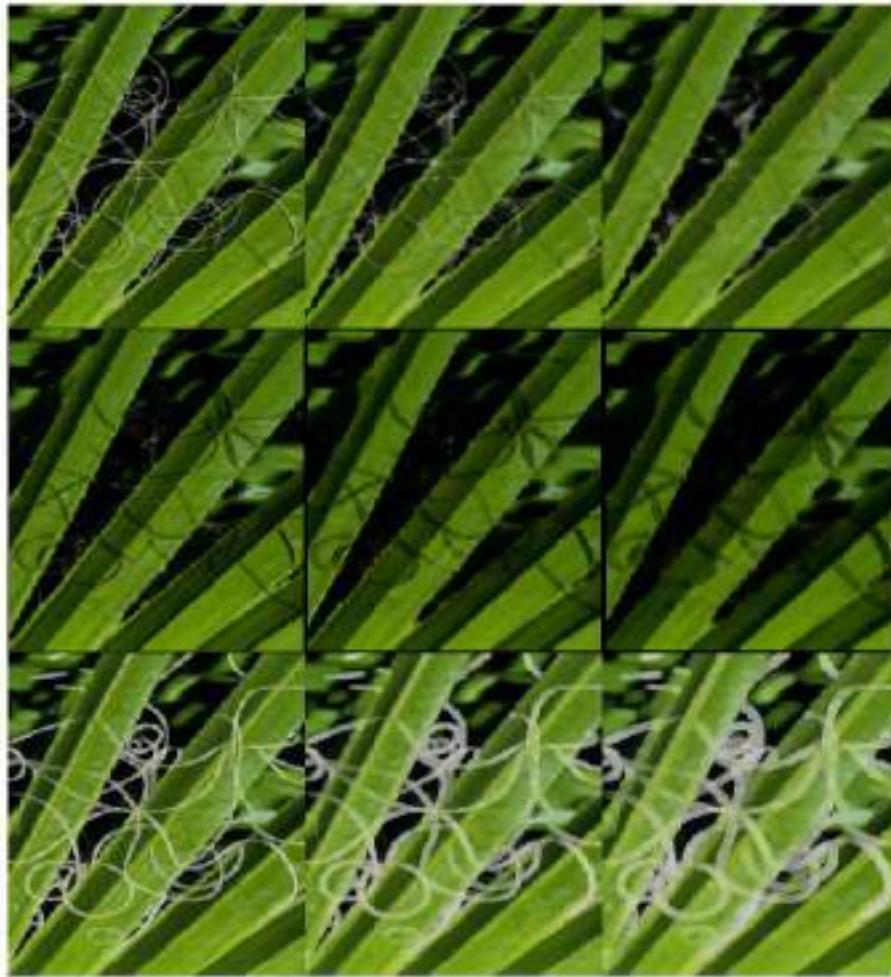


Figura 4: mediana, min, max. Tamaño de filtro:3,5,7.

EL filtro del minimo de 3x3 obtiene buenos resultado que a pesar de dejar sombras en las hojas no difumina tanto la imagen.

6. Usando su implementación de filtrado, intente comparar los siguientes filtros de detección de bordes.

a) Sobel para detección de bordes: tamaño 3×3 , Gx vertical y Gy Horizontal.

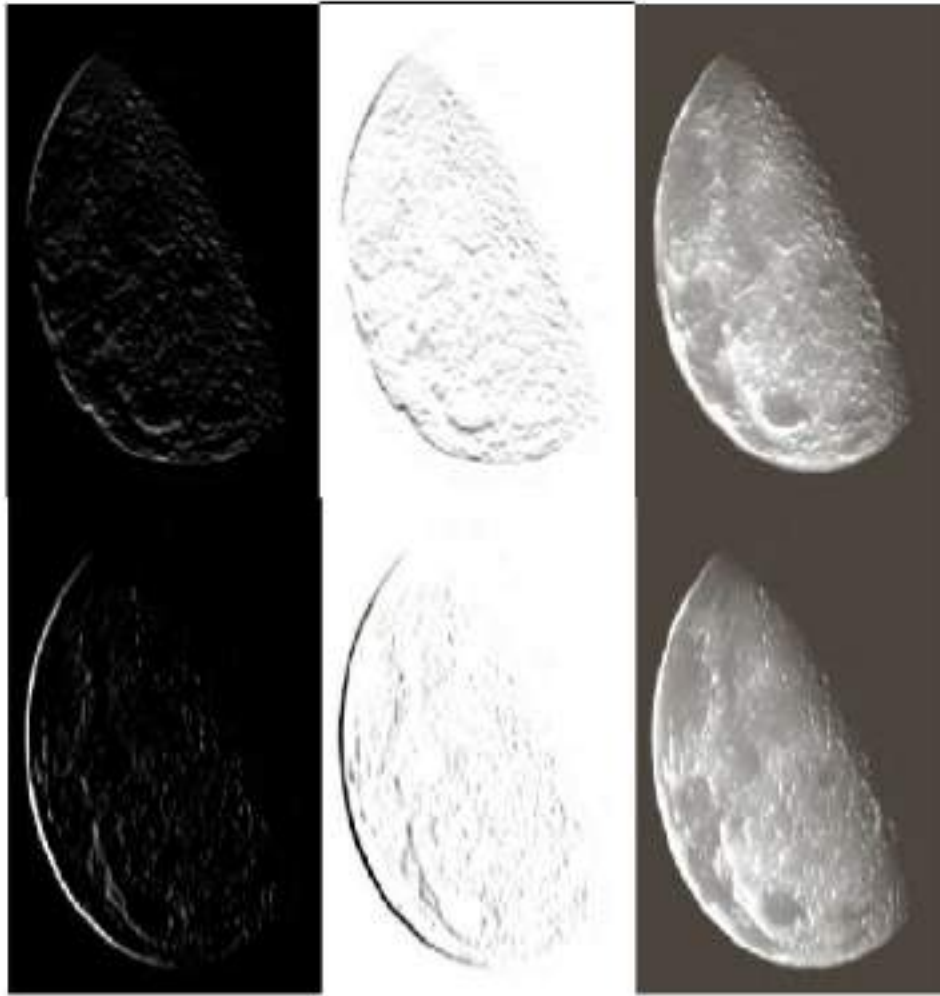


Figura 5: Sobel X y Y. Tamaño de filtro:3.

b) Laplaciano para detección de bordes: tamaño 3×3 .

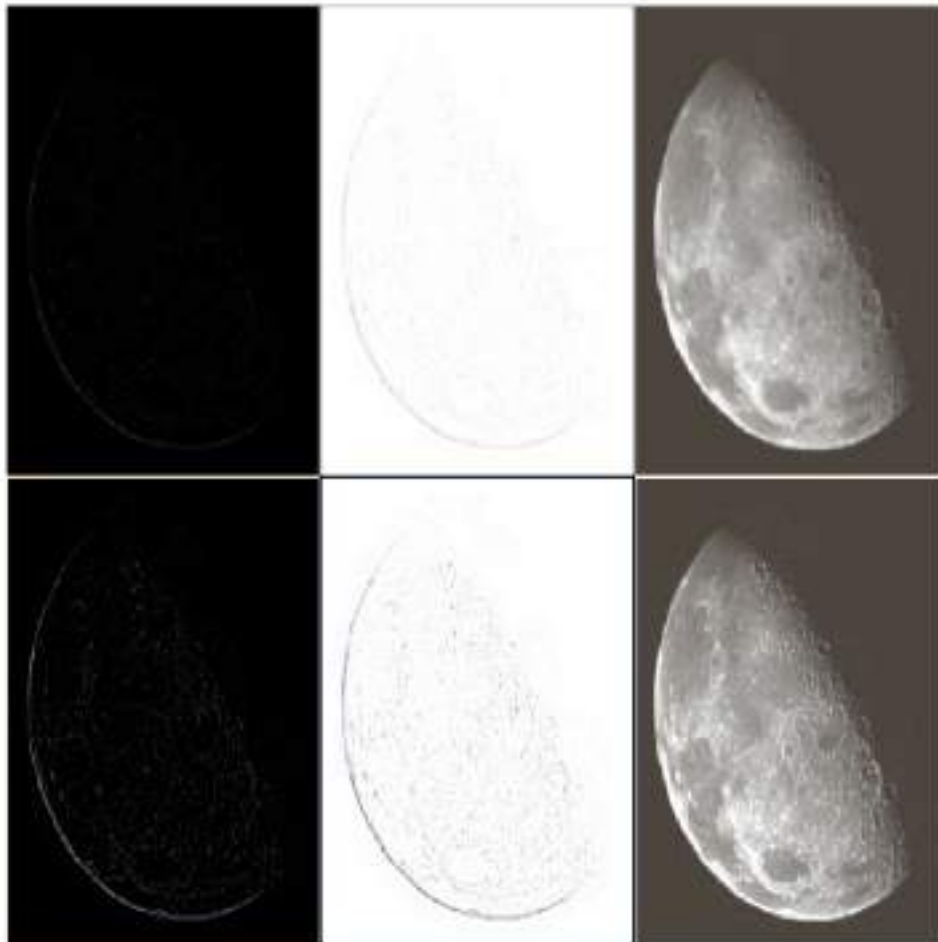


Figura 6: Laplace, centro de 4 y 8. Tamaño de filtro:3

EL filtro laplaciano con centro 8, obtiene mejores detalles.

7. Escriba una función en Matlab para aplicar la mascara de enfoque (unsharp masking).



Figura 7: original, unsharp y highboost. Tamaño de filtro:3.

```
1 function [mask] = unsharp(imagen,filterSize,k)
2     [blur] = filter(imagen,'mean',filterSize,1);
3     mask = imagen + k*(imagen - blur);
4 end
```

2. Anexo (CODIGO)

Código de matlab.

```
1
2 % imagen = imread("circuit.jpg");
3 % imagen = imread("moon.png");
4 % imagen = imread("a.png");
5 % imagen = imread("green.png");
6
7 % padding Test
8 % filtro = ones(17,17);
9 % imgpad1 = padding(imagen, 3, 1);
10 % imshow(imgpad1)
11
12 % PUNTO 3
13 I = imread("a.png");
14
15 figure();
16
17 montage({I,filter(I,'mean',3,false),filter(I,'mean',5,false), ...
18         filter(I,'mean',9,false),filter(I,'mean',15,false), ...
19         filter(I,'mean',35,false)})
20 % subplot(2,3,1); imshow(I); title("Original");
21 % subplot(2,3,2), imshow(filter(I,'mean',3,false)), title("mean: Filter size 3");
22 % subplot(2,3,3), imshow(filter(I,'mean',5,false)), title("mean: Filter size 5");
23 % subplot(2,3,4), imshow(filter(I,'mean',9,false)), title("mean: Filter size 9");
24 % subplot(2,3,5), imshow(filter(I,'mean',15,false)), title("mean: Filter size
25         15");
26 % subplot(2,3,6), imshow(filter(I,'mean',35,false)), title("mean: Filter size
27         35");
28
29 % Punto 4
30 I = imread("circuit.jpg");
31
32 figure();
33 a = filter(I,'mean',3,true);
34 b = filter(I,'mean',5,true);
35 c = filter(I,'mean',7,true);
36 d = filter(I,'median',3,true);
37 e = filter(I,'median',5,true);
38 f = filter(I,'median',7,true);
```

```
37 montage({I,a,b,c,I,d,e,f},'Size',[2 4])
38 % subplot(2,3,1); imshow(I); title("Original");
39 % subplot(2,3,2), imshow(filter(I,'mean',3,true)), title("mean: Filter size 3");
40 % subplot(2,3,3), imshow(filter(I,'mean',5,true)), title("mean: Filter size 5");
41 % subplot(2,3,4), imshow(filter(I,'mean',9,true)), title("mean: Filter size 9");
42 % subplot(2,3,5), imshow(filter(I,'mean',11,true)), title("mean: Filter size
    11");
43 % subplot(2,3,6), imshow(filter(I,'mean',13,true)), title("mean: Filter size
    13");
44
45 % figure();
46 % a = filter(I,'median',3,true);
47 % b = filter(I,'median',5,true);
48 % c = filter(I,'median',9,true);
49 % d = filter(I,'median',11,true);
50 % e = filter(I,'median',13,true);
51 % montage({I,a,b,c,d,e})
52 % subplot(2,3,1); imshow(I), title("Original");
53 % subplot(2,3,2), imshow(filter(I,'median',3,true)), title("median: Filter size
    3");
54 % subplot(2,3,3), imshow(filter(I,'median',5,true)), title("median: Filter size
    5");
55 % subplot(2,3,4), imshow(filter(I,'median',9,true)), title("median: Filter size
    9");
56 % subplot(2,3,5), imshow(filter(I,'median',11,true)), title("median: Filter size
    11");
57 % subplot(2,3,6), imshow(filter(I,'median',13,true)), title("median: Filter size
    13");
58
59 % PUNTO 5 a
60 I = imread("green.png");
61
62 figure();
63 a = filter(I,'median',3,true);
64 b = filter(I,'median',5,true);
65 c = filter(I,'median',7,true);
66 d = filter(I,'mean',3,true);
67 e = filter(I,'mean',5,true);
68 f = filter(I,'mean',7,true);
69 montage({a,b,c,d,e,f},'Size',[2 3])
70 % subplot(2,3,1), imshow(filter(I,'median',3,false)), title("median: Filter size
    3");
```

```
71 % subplot(2,3,2), imshow(filter(I,'median',5,false)), title("median: Filter size
    5");
72 % subplot(2,3,3), imshow(filter(I,'median',7,false)), title("median: Filter size
    7");
73
74 % subplot(2,3,4), imshow(filter(I,'mean',3,false)), title("mean: Filter size 3");
75 % subplot(2,3,5), imshow(filter(I,'mean',5,false)), title("mean: Filter size 5");
76 % subplot(2,3,6), imshow(filter(I,'mean',7,false)), title("mean: Filter size 7");
77
78 % Punto 5 b
79 I = imread("green.png");
80 figure();
81 a = filter(I,'median',3,true);
82 b = filter(I,'median',5,true);
83 c = filter(I,'median',7,true);
84 d = filter(I,'min',3,true);
85 e = filter(I,'min',5,true);
86 f = filter(I,'min',7,true);
87 g = filter(I,'max',3,true);
88 h = filter(I,'max',5,true);
89 i = filter(I,'max',7,true);
90 montage({a,b,c,d,e,f,g,h,i}, 'Size', [3 3])
91 % subplot(1,3,1), imshow(filter(I,'median',3,false)), title("median: Filter size
    3");
92 % subplot(1,3,2), imshow(filter(I,'median',5,false)), title("median: Filter size
    5");
93 % subplot(1,3,3), imshow(filter(I,'median',7,false)), title("median: Filter size
    7");
94 %
95 % figure();
96 % subplot(1,3,1), imshow(filter(I,'min',3,false)), title("min: Filter size 3");
97 % subplot(1,3,2), imshow(filter(I,'min',5,false)), title("min: Filter size 5");
98 % subplot(1,3,3), imshow(filter(I,'min',7,false)), title("min: Filter size 7");
99 %
100 % figure();
101 % subplot(1,3,1), imshow(filter(I,'max',3,false)), title("max: Filter size 3");
102 % subplot(1,3,2), imshow(filter(I,'max',5,false)), title("max: Filter size 5");
103 % subplot(1,3,3), imshow(filter(I,'max',7,false)), title("max: Filter size 7");
104
105 % PUNTO 6
106 imagen = imread("moon.png");
107
```

```
108 sobelX = [-1,-2,-1;0,0,0;1,2,1];
109 sobelY = sobelX';
110
111 [sobel_X] = filter(imagen,sobelX,3,true);
112
113 figure();
114 a = sobel_X;
115 b = imcomplement(sobel_X);
116 c = imagen + sobel_X;
117 d = sobel_Y;
118 e = imcomplement(sobel_Y);
119 f = imagen + sobel_Y;
120 montage({a,b,c,d,e,f},'Size',[2 3])
121
122 % subplot(2,3,1); imshow(sobel_X); title("Sobel X")
123 % subplot(2,3,2); imshow(imcomplement(sobel_X)), title("complement Sobel X")
124 % subplot(2,3,3); imshow(imagen + sobel_X); title("image + Sobel X")
125 %
126 % [sobel_Y] = filter(imagen,sobelY,3,1);
127 % subplot(2,3,4);imshow(sobel_Y); title("Sobel Y")
128 % subplot(2,3,5); imshow(imcomplement(sobel_Y)), title("complement Sobel Y")
129 % subplot(2,3,6);imshow(imagen + sobel_Y); title("image + Sobel Y")
130
131 imagen = imread("moon.png");
132
133 laplace1 = [0,-1,0;-1,4,-1;0,-1,0];
134 laplace2 = [-1,-1,-1;-1,8,-1;-1,-1,-1];
135
136 [imgLaplace1] = filter(imagen,laplace1,3,1);
137
138 figure();
139 a = imgLaplace1;
140 b = imcomplement(imgLaplace1);
141 c = imagen + imgLaplace1;
142 d = imgLaplace2;
143 e = imcomplement(imgLaplace2);
144 f = imagen + imgLaplace2;
145 montage({a,b,c,d,e,f},'Size',[2 3])
146
147 % subplot(2,3,1); imshow(imgLaplace1), title("Laplace: 4")
148 % subplot(2,3,2); imshow(imcomplement(imgLaplace1)), title("complement Laplace:
    4")
```

```
149 % subplot(2,3,3); imshow(imagen + imgLaplace1), title("image + laplace:4")
150 %
151 % [imgLaplace2] = filter(imagen,laplace2,3,1);
152 % subplot(2,3,4);imshow(imgLaplace2), title("Laplace: 8")
153 % subplot(2,3,5);imshow(imcomplement(imgLaplace2)), title("complement Laplace:
    8")
154 % subplot(2,3,6);imshow(imagen + imgLaplace2), title("image + laplace:8")
155
156 % PUNTO 7
157 imagen = imread("moon.png");
158
159 mask = unsharp(imagen,5,1);
160 mask2 = unsharp(imagen,5,10);
161
162 figure();
163 a = imagen;
164 b = mask;
165 c = mask2;
166 montage({a,b,c},'Size',[1 3])
167
168 % subplot(1,3,1); imshow(imagen); title "original";
169 % subplot(1,3,2); imshow(mask); title "unSharp";
170 % subplot(1,3,3); imshow(mask2); title "highBoost";
171
172 FILTER
173 function [filteredImage] = filter(image, filterType,filterSize,paddingType)
174
175     if isa(filterType,'char')
176         switch filterType
177             case 'min'
178                 myfilter = @(x) min(x,[],"all");
179             case 'max'
180                 myfilter = @(x) max(x,[],"all");
181             case 'mean'
182                 myfilter = @(x) round(mean2(x),0);
183             case 'median'
184                 myfilter = @(x) median(x,"all");
185         end
186     else
187         den = sum(filterType,"all");
188         if den == 0
189             den = 1;
```



```

190     end
191     myfilter = @(x) round(sum(filterType.*double(x),"all")/den,0);
192 end
193
194 filtersize = filterSize;
195 imageWithPadding = padding(image,filtersize,paddingType);
196 filteredImage = image;
197
198 [sizeY,sizeX,sizeZ] = size(image);
199
200 for z=1:sizeZ
201     for y=1:sizeY
202         for x=1:sizeX
203             Y=y+(filtersize-1)/2;
204             X=x+(filtersize-1)/2;
205             Y1=Y-(filtersize-1)/2;
206             X1=X-(filtersize-1)/2;
207             Y2=Y+(filtersize-1)/2;
208             X2=X+(filtersize-1)/2;
209
210             area = imageWithPadding(Y1:Y2,X1:X2,z);
211
212             newValue =myfilter(area);
213             filteredImage(y,x,z) = newValue;
214         end
215     end
216 end
217 end
218 PADDING
219 function [imageWithPadding] = padding(image, filterSize, ceros)
220
221     sizeYfilter = filterSize;
222     [sizeY,sizeX,~] = size(image); % image size
223
224     paddingSize = (sizeYfilter-1)/2; % calculated padding size
225     imageWithPadding = padarray(image,[paddingSize,paddingSize],0,'both'); %
226         image with padding of zeros
227     [sizeYpadding, sizeXpadding, ~] = size(imageWithPadding); % image size with
228         padding
229
230     if ~ceros
231         % Y

```

```
230     imageWithPadding(paddingSize+1:sizeYpadding-paddingSize ...
231         ,1:paddingSize,:) = repmat(image(1:sizeY,1,:),1,paddingSize);
232     imageWithPadding( ...
233         paddingSize+1:(sizeYpadding-paddingSize), ...
234         (sizeX+paddingSize+1):sizeXpadding,:) = repmat( ...
235         image(1:sizeY,sizeX,:),1,paddingSize);
236     % x
237     imageWithPadding(1:paddingSize,1:sizeXpadding,:) = repmat( ...
238         imageWithPadding((paddingSize+1),1:sizeXpadding,:),paddingSize,1);
239     imageWithPadding((sizeY+paddingSize+1):sizeYpadding, ...
240         1:sizeXpadding,:) = repmat( ...
241         imageWithPadding(sizeY+1,1:sizeXpadding,:),paddingSize,1);
242     end
243 end
244 UNSHARP
245 function [mask] = unsharp(imagen,filterSize,k)
246     [blur] = filter(imagen,'mean',filterSize,1);
247     mask = imagen + k*(imagen - blur);
248 end
```