

Informe laboratorio - Procesamiento de Imágenes Digitales

Transformada de Fourier

Ruben Rodriguez

5 de septiembre de 2022

Índice

1. Desarrollo	1
2. Anexo (CODIGO)	11

1. Desarrollo

1. Experimente con las funciones `fft2`, `ifft2` y `fftshift` y reproduzca los resultados de la figura.



Figura 1: Resultados de las respectivas funciones.

2. Reproduzca los resultados de la figura, que muestra las propiedades de traslación y rotación de la transformada de fourier.

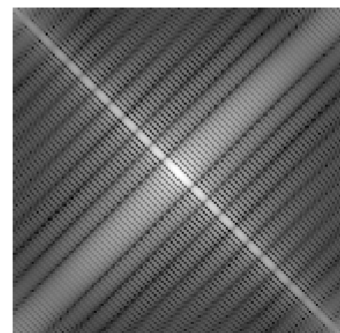
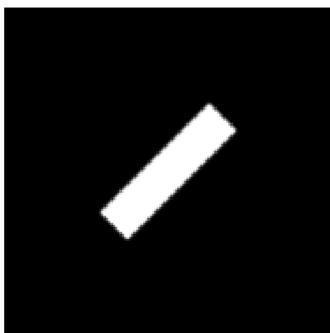
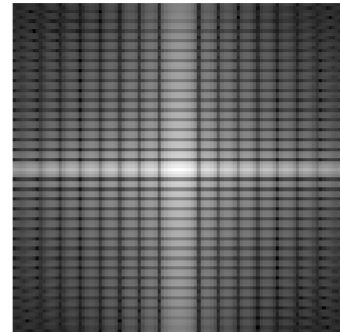
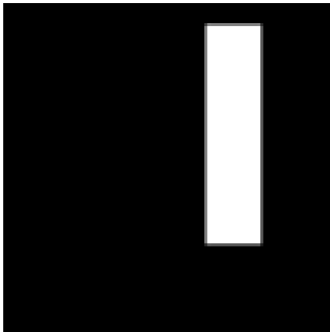


Figura 2: Imagenes con tranformaciones geometricas y respectivas trensformaciones.

3. Reproduzca los resultados de la figura.

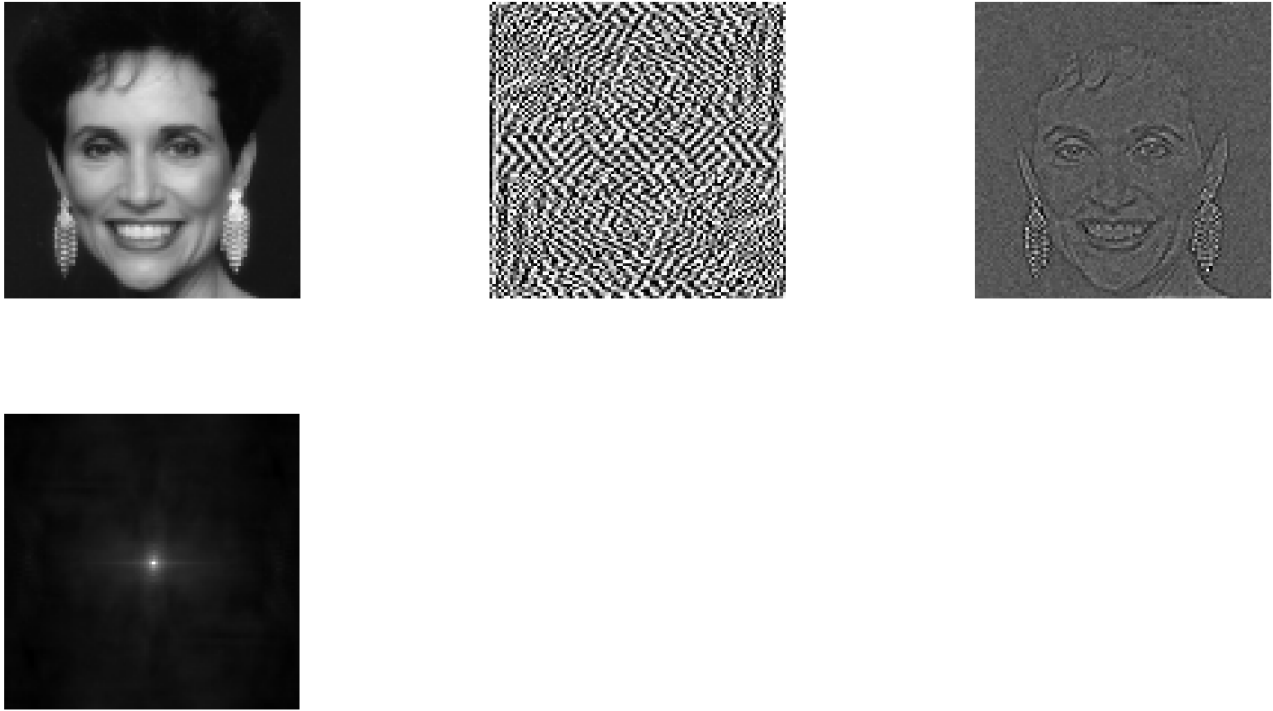


Figura 3: Imagenes generadas(original, phase angle, phase angle reconstruction, reconstruction using only spectrum).

4. Escriba una función en MATLAB para calcular la transformada de Fourier de una imagen, y una para la inversa, use las funciones `fft2` e `ifft2` para comparar los resultados.

```

1 function [fft2] = myfft2(imagen)
2     [N, M] = size(imagen);
3     fft2=zeros(N,M);
4     for u=1:M
5         for v=1:N
6             sum_x=0;
7             for x=1:M
8                 sum_y=0;
9                 for y=1:N
10                    sum_y=sum_y+...
11                    complex(double(imagen(y,x)))*exp(-2i*pi*((u-1)*(x-1)/M +
12                    (v-1)*(y-1)/N));
13                end
14                sum_x=sum_x+sum_y;
15            end
16            fft2(v,u)=sum_x;
17        end
18    end
19    %%
20 function [ifft2] = myifft2(imagen)
21     [N, M] = size(imagen);
22     ifft2=zeros(N,M);
23     for x=1:M
24         for y=1:N
25             sum_u=0;
26             for u=1:M
27                 sum_v=0;
28                 for v=1:N
29                    sum_v=sum_v+...
30                    complex(double(imagen(v,u)))*exp(-2i*pi*((u-1)*(x-1)/M +
31                    (v-1)*(y-1)/N));
32                end
33                sum_u=sum_u+sum_v;
34            end
35            ifft2(y,x)=real(sum_u)/(M*N);
36        end
37    end

```

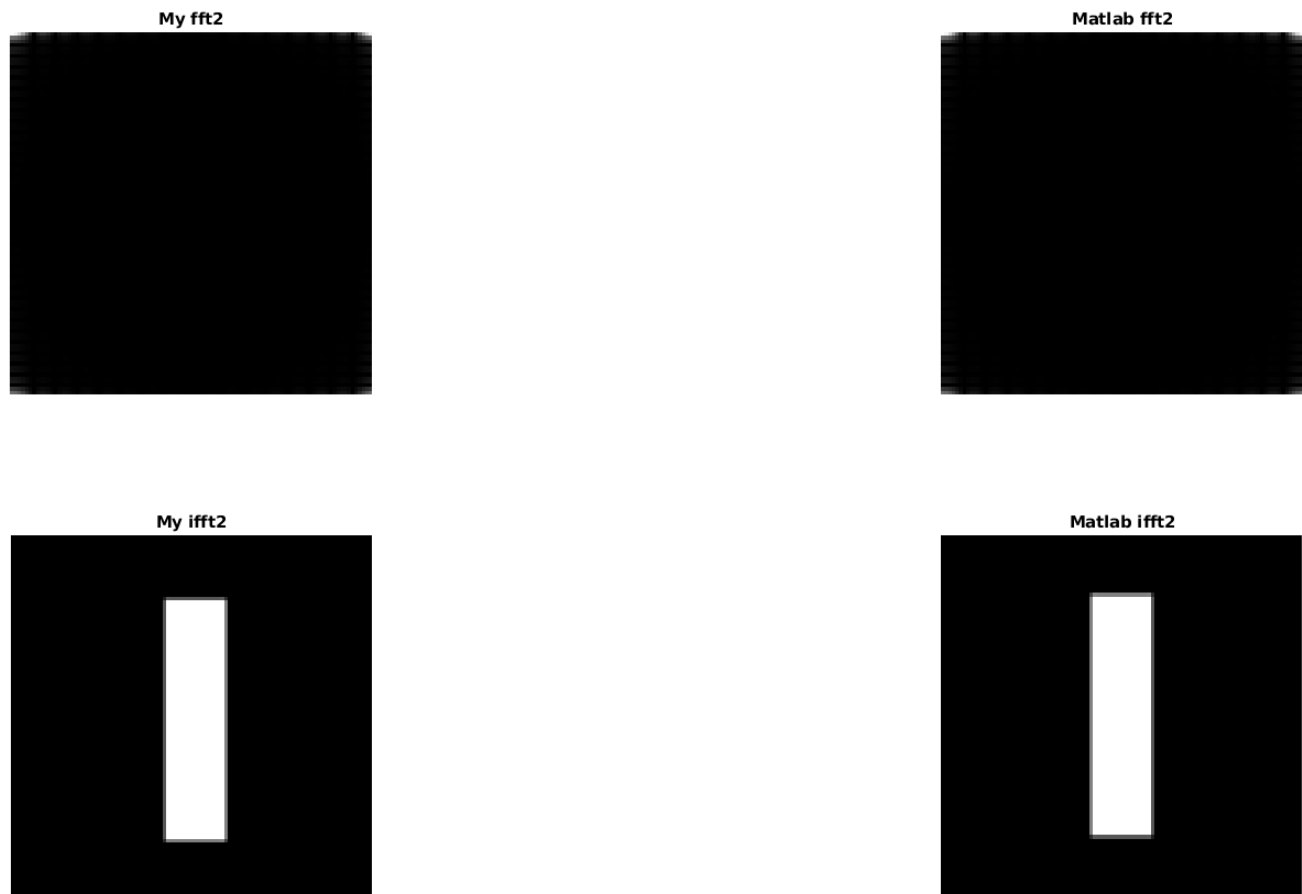


Figura 4: Comparacion de imagenes generadas por nuestra funcion y la funcion de matlab

No se encuentran diferencias.

5. Escriba una rutina que genere los filtros descritos en la sección anterior, la función debería recibir el tamaño del filtro, el tipo de filtro (ideal, Butterworth, Gaussian), la frecuencia de corte, y si el filtro es pasa bajo o pasa altas. Realice pruebas generando varias filtros con frecuencias de corte distintas y muestre los resultados.

```

1  % Punto 5 - 6
2  cortes = [5,10,20,40];
3  filtros = ["ideal","Butterworth","Gaussian"];
4  pasa = ["bajo","alto"];
5  for p=1:2
6      figure;
7      for j=1:3
8          for i=1:4
9              % USANDO NUESTRA FUNCION IMPLEMENTADA
10             subplot(3,4,i+4*(j-1)), imshow(fourierFilters([100
11                 100],filtros(j),pasa(p),cortes(i)),[]),title(strcat("Pasa
12                 ",pasa(p)," ",filtros(j)," ",int2str(cortes(i))), "FontSize",8)
13             end
14         end
15     end
16 % Punto 5 - 6
17 I5 = imresize(imread("coins.jpg"),[100 100]);
18 for p=1:2
19     figure;
20     for j=1:3
21         for i=1:4
22             % USANDO NUESTRA FUNCION IMPLEMENTADA
23             mask = fourierFilters(size(I5),filtros(j),pasa(p),cortes(i));
24             I5F = fft2(I5);
25             I5FS = fftshift(I5F);
26
27             fmask = mask.*I5FS;
28
29             Ifiltered = real(ifft2(ifftshift(fmask)));
30
31             subplot(3,4,i+4*(j-1)), imshow(Ifiltered,[]),title(strcat("Pasa
32                 ", pasa(p)," ",filtros(j),"
33                 ",int2str(cortes(i))), "FontSize",8)
34         end
35     end
36 end

```

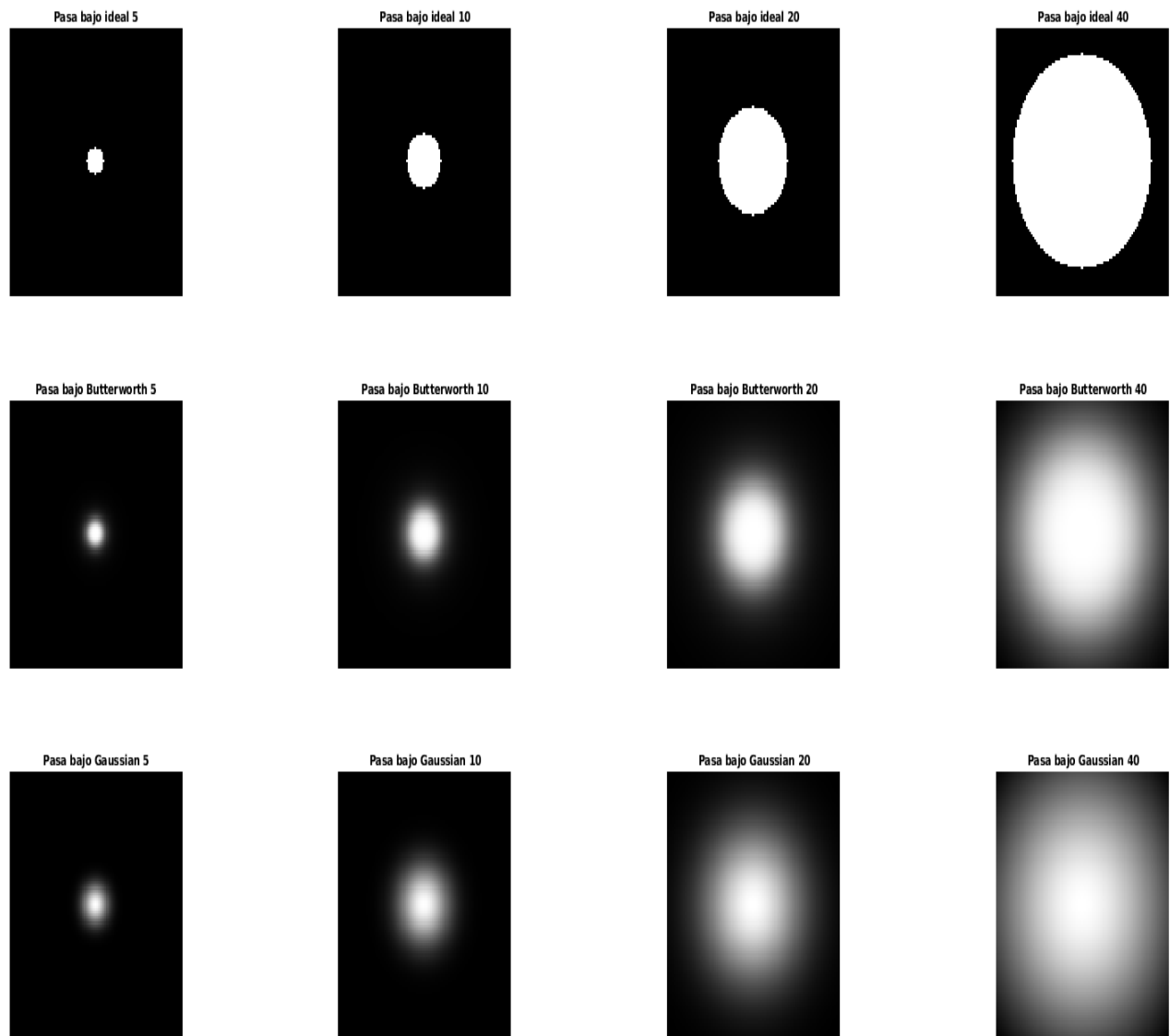


Figura 5: Filtros pasa baja generados.

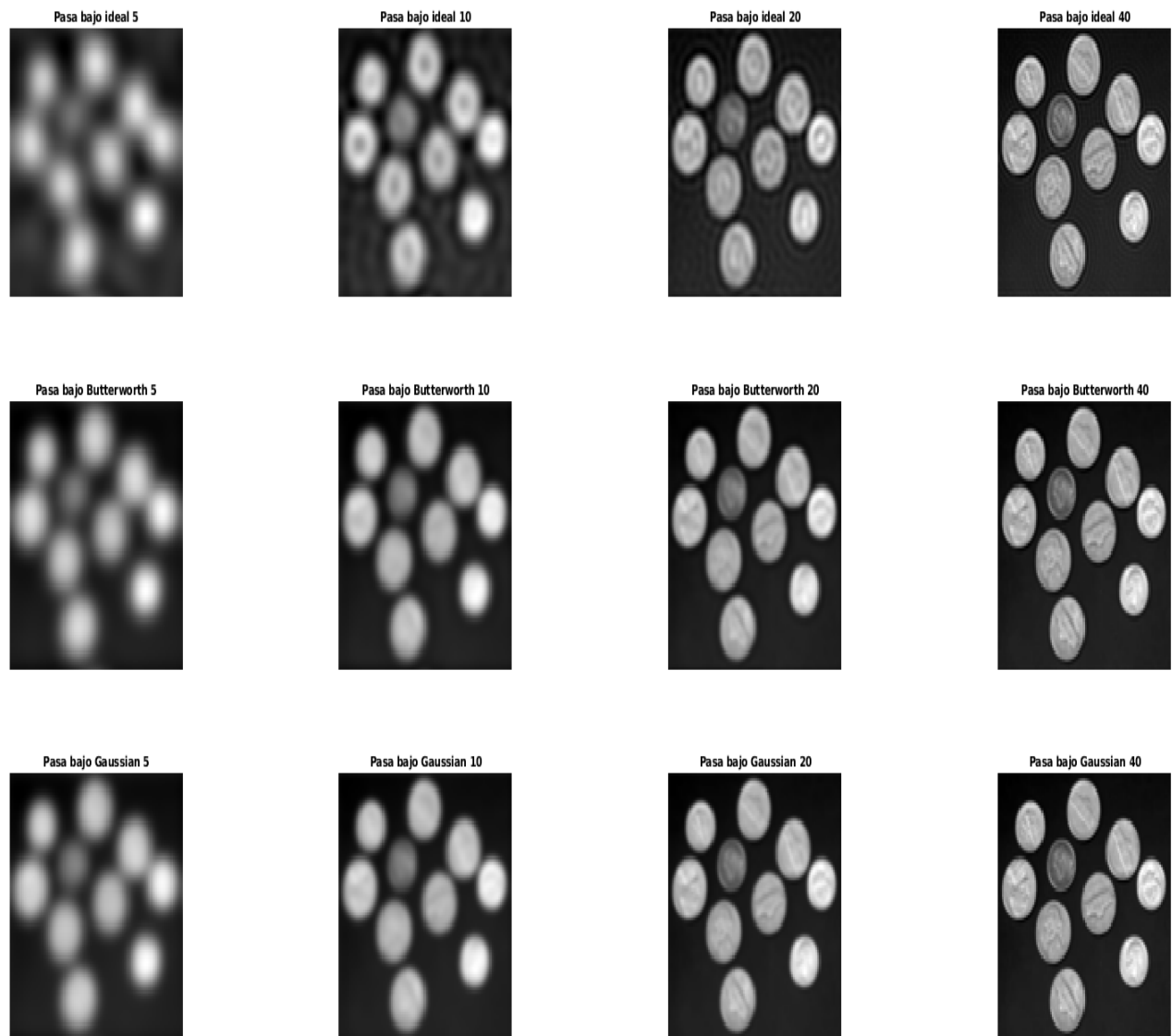


Figura 6: Filtros pasa bajo con varias frecuencias de corte aplicado a la imagen coins.png.

6. Realice el mismo proceso del punto anterior usando filtros pasa alto.

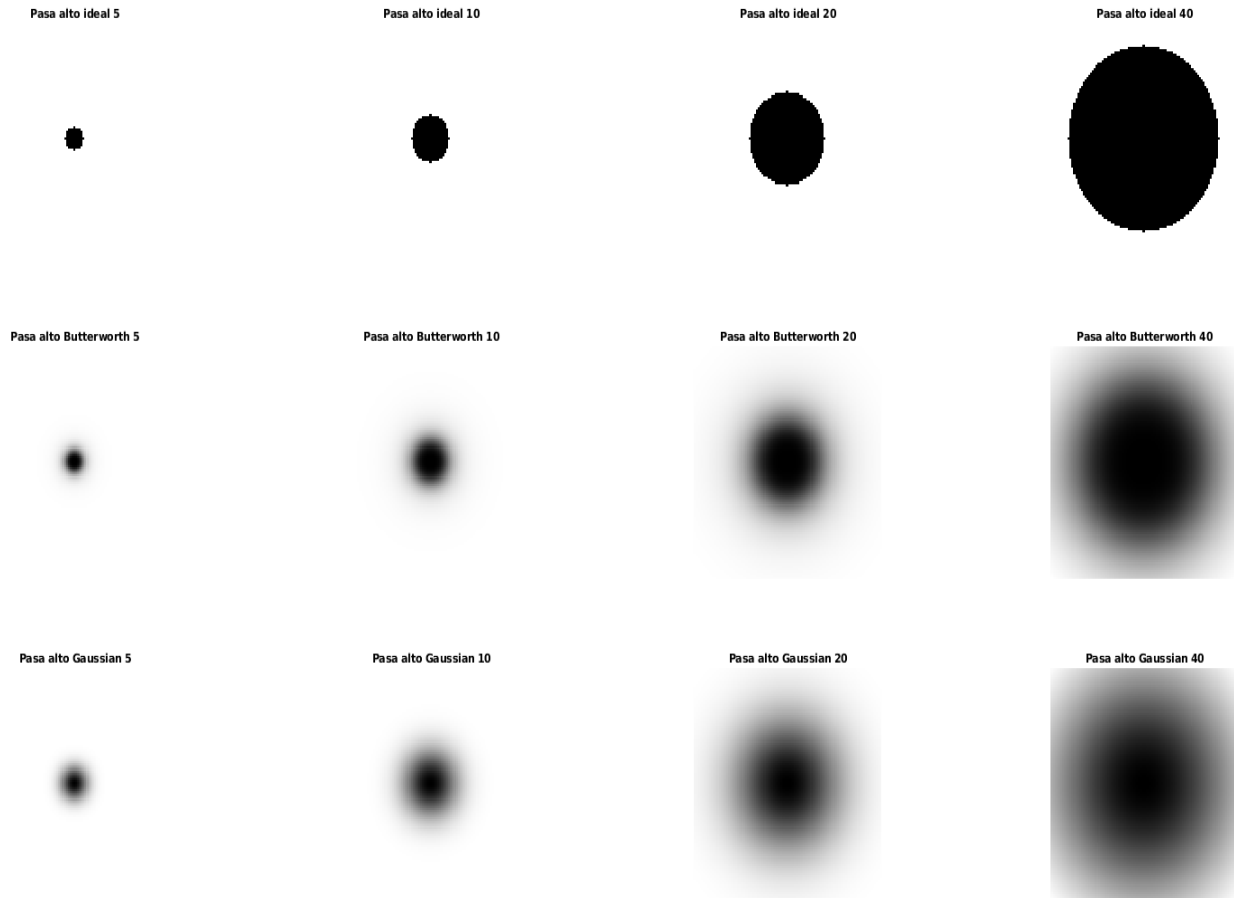


Figura 7: Filtros pasa alto generados.

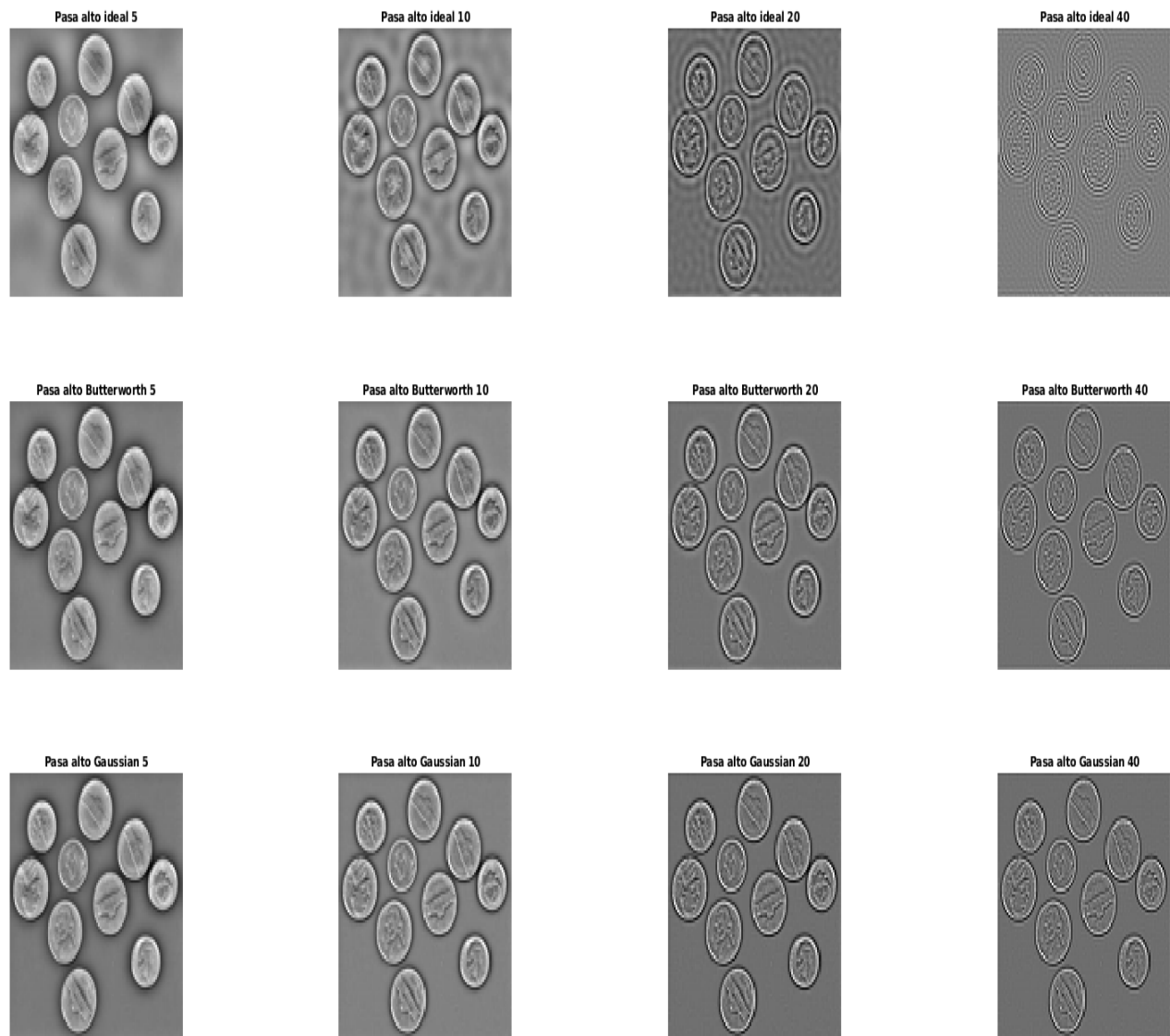


Figura 8: Filtros pasa alto con varias frecuencias de corte aplicado a la imagen coins.png.

2. Anexo (CODIGO)

Código de matlab.

```
1
2 % Punto 1
3 figure
4 [I] = imresize(imread("Fig0419.tif"),[100 100]);
5 subplot(2,2,1),imshow(I,[])
6
7 a = abs(fft2(I));
8 subplot(2,2,2),imshow(a, [])
9
10 b = abs(fftshift(a));
11 subplot(2,2,3),imshow(b, [])
12
13 c = log(1 + fftshift(a));
14 subplot(2,2,4),imshow(c,[])
15
16 % Punto 2
17 figure
18 I2 = imtranslate(I,[20 -10]);
19 subplot(2,2,1),imshow(I2)
20
21 a2 = log(1 + fftshift(abs(fft2(I2))));
22 subplot(2,2,2),imshow(a2, [])
23
24 I3 = imrotate(I,-45,"bicubic");
25 subplot(2,2,3),imshow(I3)
26
27 b2 = log(1 + fftshift(abs(fft2(I3))));
28 subplot(2,2,4),imshow(b2, [])
29
30 % Punto 3
31 figure
32 I4 = imresize(imread("FigP0402(a)(woman).tif"),[100 100]);
33 subplot(2,3,1),imshow(I4)
34
35 %spectrum
36 I5S= log(1 + fftshift(abs(fft2(I4))));
37 % subplot(2,3,4),imshow(a3_3,[])
38
```

```

39 a3 = fft2(I4);
40 % phase angle
41 a3_1 = exp(1i*angle(a3));
42 subplot(2,3,2),imshow(real(a3_1), [])
43 % phase angle reconstruction
44 a3_2 = ifft2(a3_1);
45 subplot(2,3,3),imshow(real(a3_2), [])
46 % reconstruction using only spectrum
47 a3_3 = ifftshift(ifft2(abs(a3)));
48 subplot(2,3,4),imshow(a3_3, [])
49
50 % phase angle woman - rectangle spectrum
51 % a3_4 = a3_1.*c;
52 % a3_4 = real(ifft2(ifftshift(a3_4)));
53 % subplot(2,3,5),imshow(a3_4, [])
54
55 % phase angle rectangle - woman spectrum
56 % a3_5 = exp(1i*angle(fft2(I))).*I5S;
57 % a3_5 = real(ifft2(ifftshift(a3_5)));
58 % subplot(2,3,6),imshow(a3_5, [])
59
60 % Punto 4
61 figure
62 subplot(2,2,1),imshow(abs(myfft2(I)), []), title("My fft2")
63 subplot(2,2,2),imshow(abs(fft2(I)), []), title("Matlab fft2")
64 subplot(2,2,3),imshow(myifft2(fft2(I)), []), title("My ifft2")
65 subplot(2,2,4),imshow((ifft2(fft2(I))), []), title("Matlab ifft2")
66
67 % Punto 5 - 6
68
69 cortes = [5,10,20,40];
70 filtros = ["ideal","Butterworth","Gaussian"];
71 pasa = ["bajo","alto"];
72
73 for p=1:2
74     figure;
75     for j=1:3
76         for i=1:4
77             subplot(3,4,i+4*(j-1)), imshow(fourierFilters([100
78                 100],filtros(j),pasa(p),cortes(i)), [],title(strcat("Pasa
79                 ",pasa(p)," ",filtros(j)," ",int2str(cortes(i))),"FontSize",8)
80         end

```

```

79     end
80 end
81
82 % Punto 5 - 6
83
84 I5 = imresize(imread("coins.jpg"),[100 100]);
85
86 for p=1:2
87     figure;
88     for j=1:3
89         for i=1:4
90             mask = fourierFilters(size(I5),filtros(j),pasa(p),cortes(i));
91             I5F = fft2(I5);
92             I5FS = fftshift(I5F);
93
94             fmask = mask.*I5FS;
95
96             Ifiltered = real(ifft2(ifftshift(fmask)));
97
98             subplot(3,4,i+4*(j-1)), imshow(Ifiltered,[]),title(strcat("Pasa ",
99                                     pasa(p)," ",filtros(j)," ",int2str(cortes(i))), "FontSize",8)
100         end
101     end
102 end
103 function [filtro] = fourierFilters(tamano,tipos,pasa,frecuencia)
104
105     n = 2;
106     filtro = zeros(tamano(1),tamano(2));
107     domain = @(y,x) ((y-floor(tamano(1)/2))^2 + (x-floor(tamano(2)/2))^2)^0.5;
108
109     switch tipos
110     case 'ideal'
111         filtro = @(y,x) domain(y,x)<=frecuencia;
112     case 'Butterworth'
113         filtro = @(y,x) 1/(1 + (domain(y,x)/frecuencia)^(2*n));
114     case 'Gaussian'
115         filtro = @(y,x) exp((-1/(2*(frecuencia^2)))*((domain(y,x))^2));
116     end
117
118     for y=1:tamano(1)
119         for x=1:tamano(2)

```

```

120     filtro(y,x) = filter(y,x);
121     if pasa == "alto"
122         filtro(y,x) = 1 - filtro(y,x);
123     end
124 end
125 end
126 end
127 -
128 function [fft2] = myfft2(imagen)
129
130     [N, M] = size(imagen);
131     fft2=zeros(N,M);
132
133     for u=1:M
134         for v=1:N
135             sum_x=0;
136             for x=1:M
137                 sum_y=0;
138                 for y=1:N
139                     sum_y=sum_y+complex(double(imagen(y,x)))*exp(-2i*pi*((u-1)*(x-1)/M
140                                     + (v-1)*(y-1)/N));
141                 end
142                 sum_x=sum_x+sum_y;
143             end
144             fft2(v,u)=sum_x;
145         end
146     end
147 -
148 function [ifft2] = myifft2(imagen)
149
150     [N, M] = size(imagen);
151     ifft2=zeros(N,M);
152
153     for x=1:M
154         for y=1:N
155             sum_u=0;
156             for u=1:M
157                 sum_v=0;
158                 for v=1:N
159                     sum_v=sum_v +
160                         complex(double(imagen(v,u)))*exp(-2i*pi*((u-1)*(x-1)/M +

```

```
                (v-1)*(y-1)/N));  
160         end  
161         sum_u=sum_u+sum_v;  
162     end  
163     ifft2(y,x)=real(sum_u)/(M*N);  
164 end  
165 end  
166 end
```