
Actividad de Aprendizaje: **Análisis de Imagen - Convolución**

Asignatura: Procesamiento de Señales

Universidad del Rosario - Escuela de Ingeniería, Ciencia y Tecnología

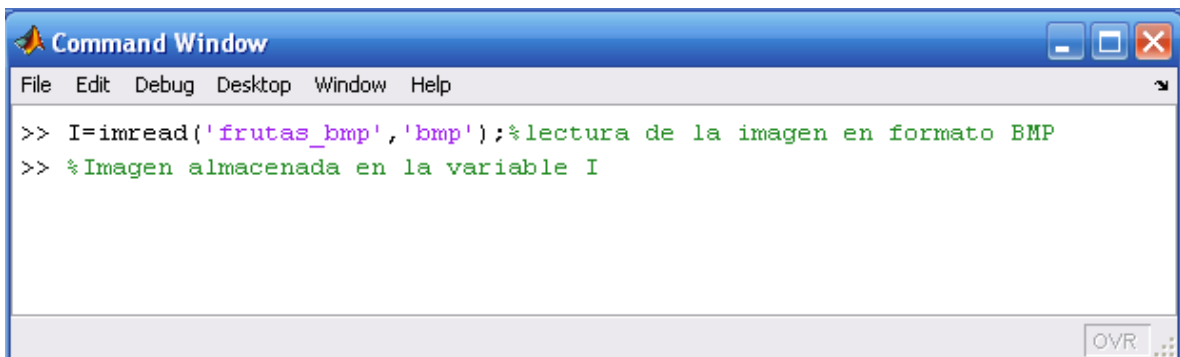
Objetivo:

- Convolución en dos dimensiones.

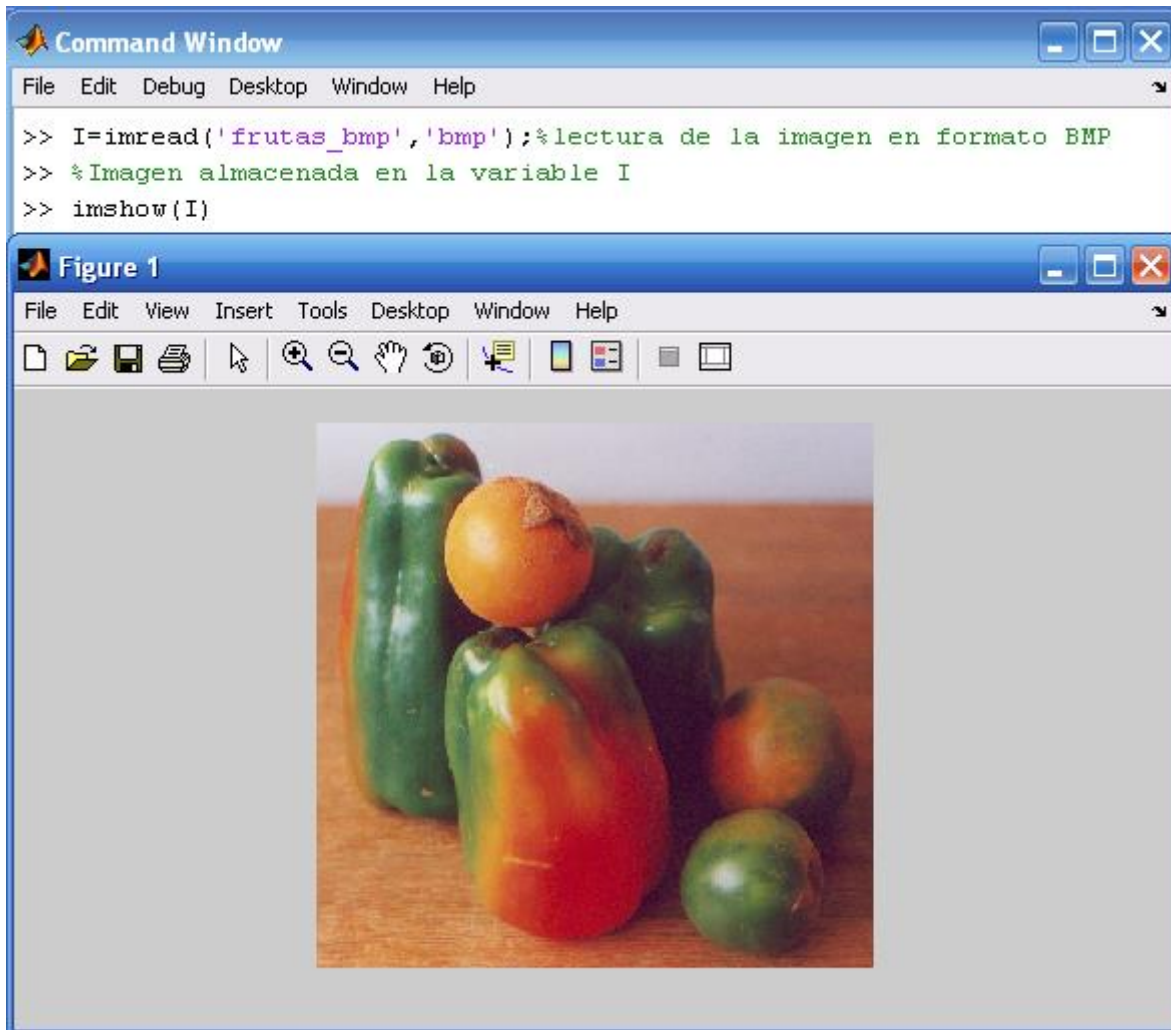
Procedimiento:

1. Se utilizarán las imágenes entregadas la semana pasada. Copie las imágenes presentes en el CD a la carpeta “work” de Matlab, esto con el fin de ubicarlas en el directorio raíz.
2. Lectura de imagen.
Leer imagen “frutas_bmp.bmp” utilizando “imread.m”

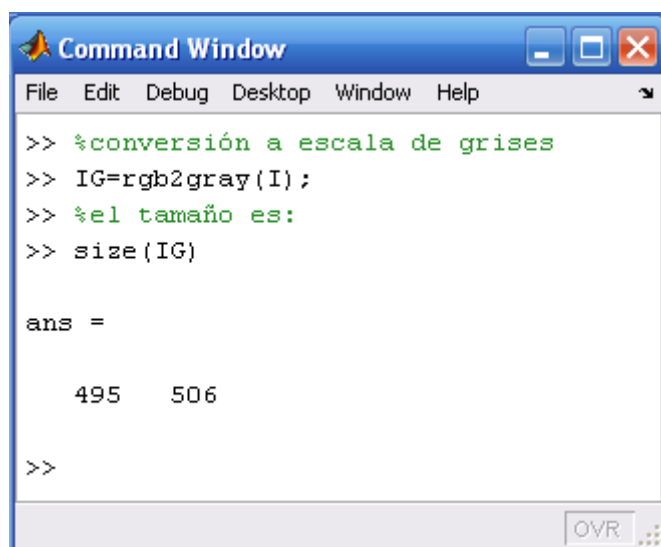
Para leer una imagen desde Matlab, se debe utilizar el comando “imread.m” que es una función.

A screenshot of the MATLAB Command Window. The window has a blue title bar with the text "Command Window" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Debug", "Desktop", "Window", and "Help". The main area of the window contains two lines of MATLAB code: ">> I=imread('frutas_bmp','bmp');%lectura de la imagen en formato BMP" and ">> %Imagen almacenada en la variable I". The code is color-coded: the prompt ">>" is green, the variable "I" is blue, the string 'frutas_bmp' is red, the string 'bmp' is blue, and the comments are green. At the bottom right of the window, there is a button labeled "OVR" and a small icon.

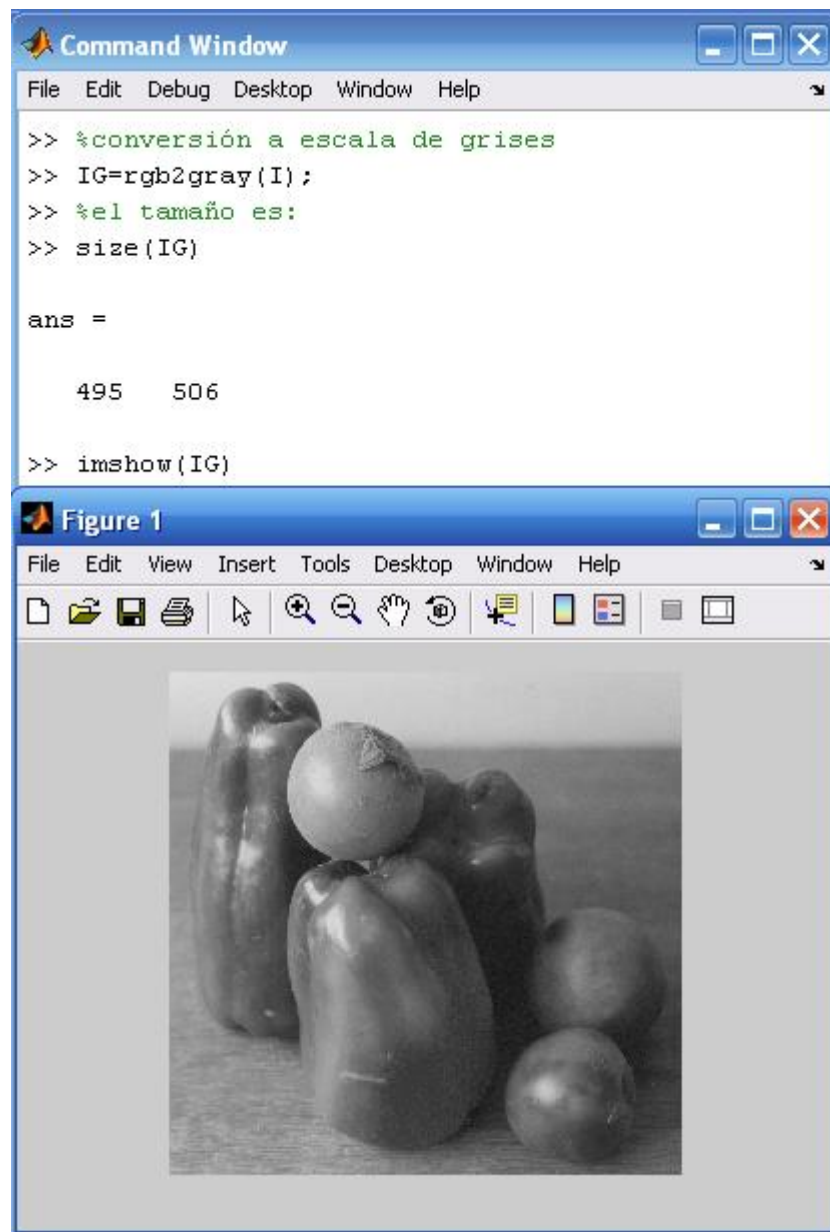
Se visualiza la imagen,



Ahora, se convierte esta imagen a color en una imagen a escala de grises utilizando el comando “*rgb2gray.m*”.



Se puede visualizar la imagen es escala de grises,



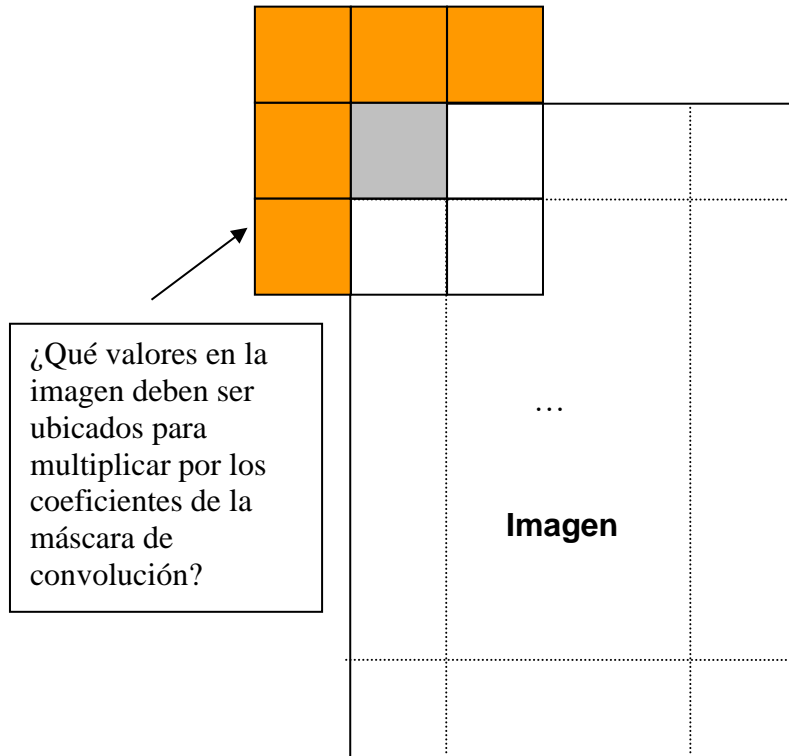
3. Convolución.

La convolución discreta es un **proceso de área** ampliamente utilizado en procesamiento de imágenes para suavizado de imágenes, resalte de bordes, detección de bordes y otros efectos.

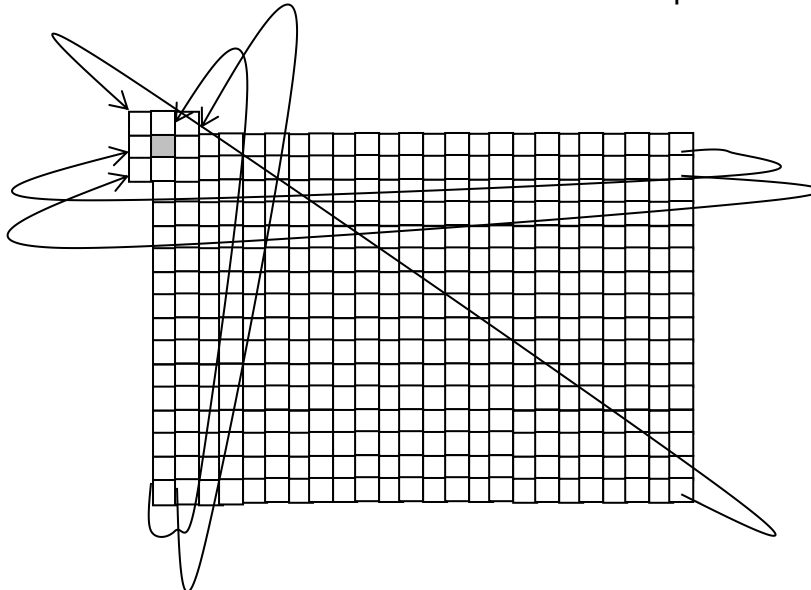
Es bueno tener en cuenta los siguientes aspectos:

- Una convolución es una suma con diferentes pesos de píxeles en la vecindad del píxel fuente(principal).
- Los pesos son determinados por una pequeña matriz llamada la **máscara de convolución** o el **kernel de convolución**.
- Generalmente el tamaño de la máscara de convolución es impar (para manejar un centro). La localización del centro corresponde a la localización de la salida.
- Esta máscara de convolución se desliza y se centra sobre cada uno de los píxeles de la imagen, generando nuevos píxeles de salida. Este nuevo píxel es calculado multiplicando cada uno de los valores en la vecindad con el correspondiente peso(máscara de convolución) y sumando esos productos.
- El nuevo píxel debe estar situado en una nueva imagen. Si este nuevo valor de píxel reemplaza al nuevo píxel, éste valor será utilizado para calcular el valor del siguiente nuevo píxel.
- La suma de los pesos en la máscara de convolución afectan la intensidad total de la máscara resultante. Muchas máscaras de convolución tienen coeficientes que suman 1, con el fin que la imagen convolucionada tendrá la misma intensidad promedio que la imagen original. Algunas máscaras de convolución tienen coeficientes negativos y suman 0, resultando valores de píxel negativos.

- ¿Como manejar los límites de una imagen? Cuando la máscara de convolución se centra en el primer píxel de la imagen (fila 1, columna 1) la ventana se sobrepone en los bordes superior e izquierdo de la imagen.



- Primera solución:* Se manejan las celdas vacías como ceros (zero-padding). Esta es una idea fácil de realizar pero no es buena si los píxeles de los límites de la imagen son igual de importantes que los demás valores de píxel.
- Segunda solución:* se comienza a convolucionar la imagen, teniendo en cuenta que la ventana de convolución no sobrepase a la imagen.
- Tercera solución:* Aumentar de tamaño la imagen antes de convolucionar. Una forma es duplicar los límites de la imagen. Para una máscara de 3x3 se deben copiar la fila superior, la columna izquierda, la columna derecha y la fila inferior.
- Cuarta solución:* se conoce como “wrap”.



- A medida que la máscara de convolución se incrementa, la carga computacional se incrementa.

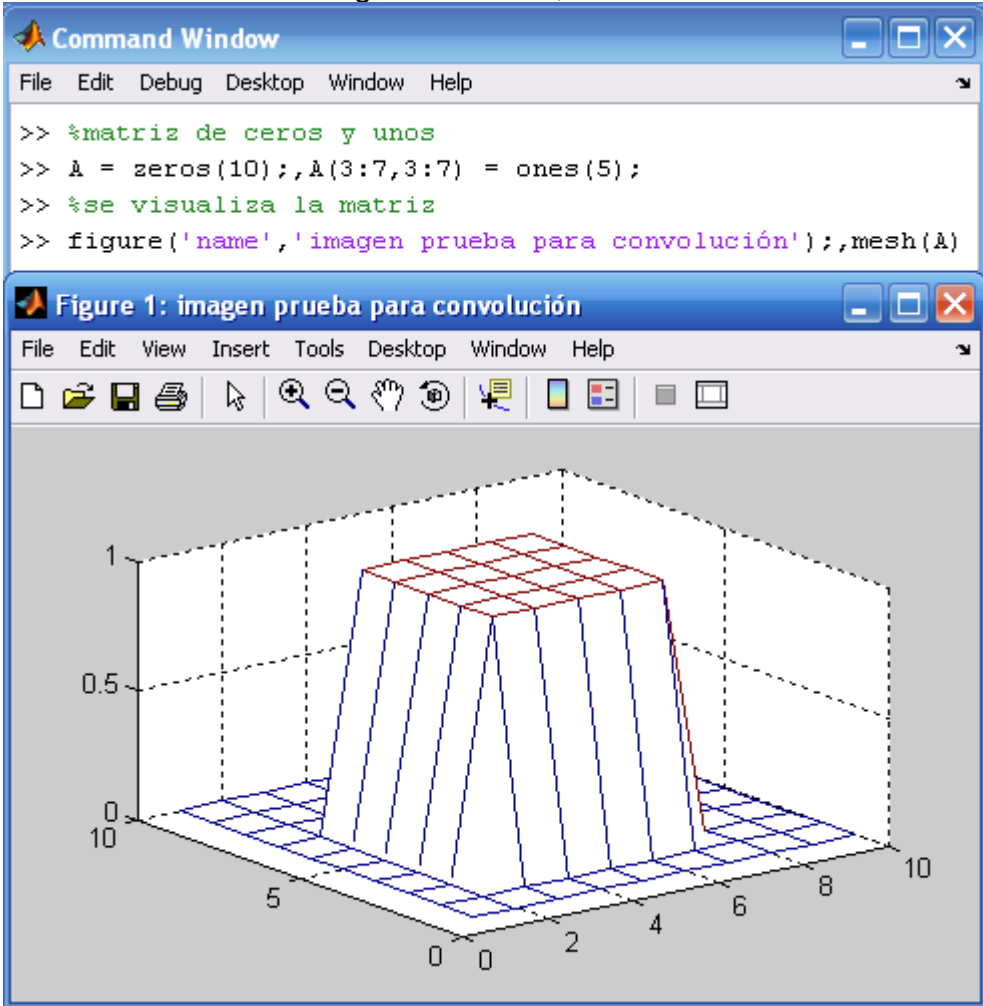
```

Editor - C:\Program Files\MATLAB71\work\opera_convolucion.m
File Edit Text Cell Tools Debug Desktop Window Help
1 - clear
2 - n=512;%numero de filas de la máscara
3 - m=512;%número de columnas de la máscara
4 - c=1;%contador
5
6 - %loop para valores desde tamaño de máscara
7 - %desde 3x3 hasta 15x15
8 - figure('name','Crecimiento exponencial con el aumento del tamaño de la máscara',...
9 - 'units','normalized','position',[0.2054 0.35 0.6953 0.44875]);
10 - for mc=3:2:15;
11 - op=m*n*2*mc*mc;%numero de operaciones suma y multiplicación
12 - plot(c,op,'o--')%dibujo del numero de operaciones
13 - hold on, drawnow%se actualiza la figura y se dibuja
14 - datos(c,:)=[c op];%almacenar los datos
15 - c=c+1;%se incrementa el contador
16
17 - end
18 - plot(datos(:,1),datos(:,2),'r--')
19 - grid on
20 - ylabel('Número de calculos [Operaciones]')
21 - xlabel('Tamaño de la máscara')
22 - title(['Grafica de operaciones para una imagen de ' num2str(n) ' x ' num2str(n)])

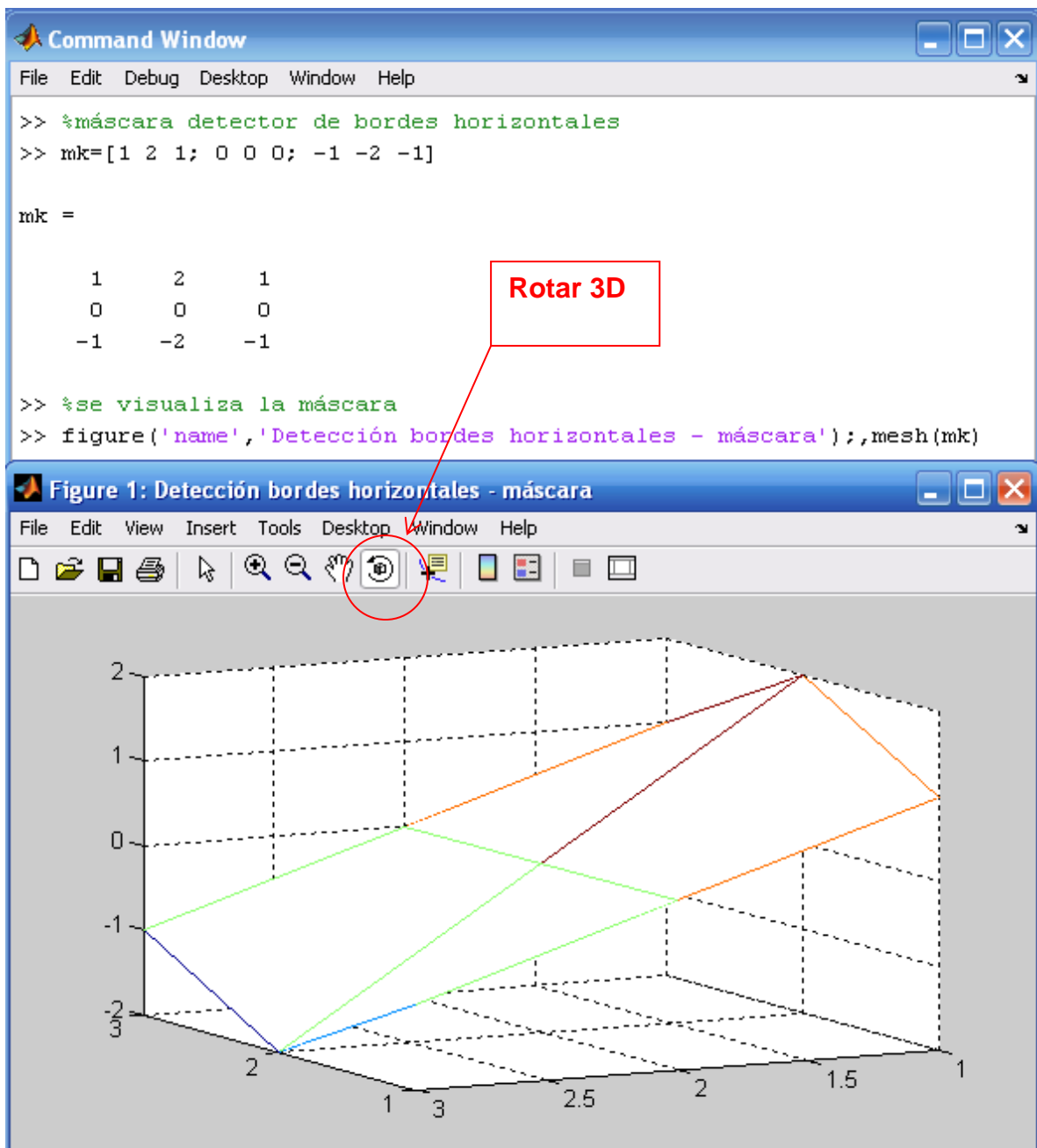
```

Ejecute el script y observe el resultado.

- Se realiza la convolución, utilizando la primera, segunda y tercera solución.
 - i. Primera Solución. Se adicionan ceros.
- Se toma la siguiente matriz,



Se aplicará una máscara que detecta bordes verticales,

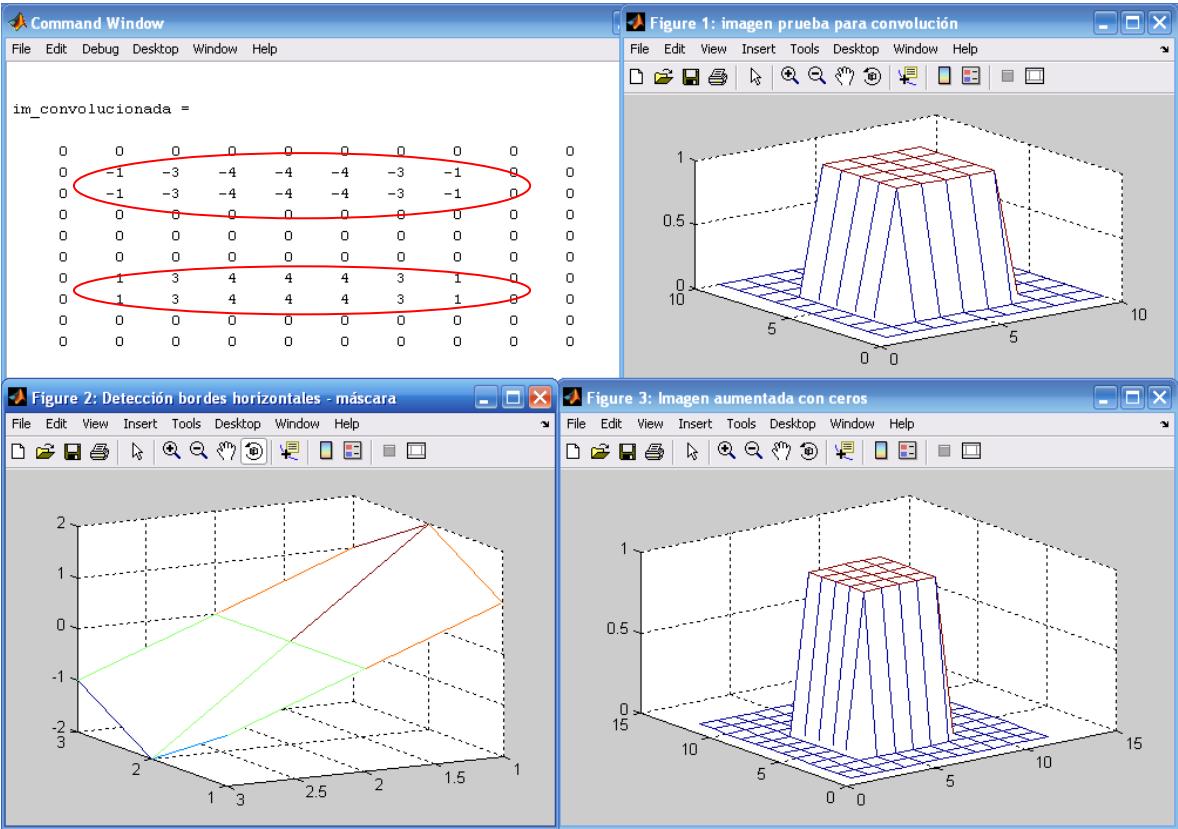


Se plantea el siguiente código de programación que realiza la convolución

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ejemplo_conv1.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base

1 - clear%borra variables en el workspace
2 - %Función pulso rectangular en 2D
3 - %representa a una imagen
4 - xi = zeros(10);xi(3:7,3:7) = ones(5);
5 - %se dibuja
6 - figure('name','imagen prueba para convolución');mesh(xi)
7
8
9 - %Determinación de imagen: color o escala de grises
10 - tam=size(xi);
11 - if length(tam)==2,
12 -     clase='imagen en escala de grises'
13 -     caso=1;
14 - elseif length(tam)==3,
15 -     clase='imagen a color'
16 -     caso=2;
17 - end
18
19 - %construcción de la máscara
20 - %Detector de bordes horizontales
21 - mk=[1 2 1; 0 0 0; -1 -2 -1];%KERNEL DE CONVOLUCIÓN
22 - %Dibujo de la máscara detectora de bordes
23 - figure('name','Detección bordes horizontales - máscara');mesh(mk)
24
25 - %se toma el tamaño de la máscara
26 - tam_mask=size(mk);
27 - n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
28 - %se deben aumentar a la imagen - ceros(primer solución)
29
30 - switch caso
31 -     case 1
32 -         %tamaño de la imagen
33 -         filas=tam(1,1);
34 -         columnas=tam(1,2);
35 -         %se aumenta la primera columna y la última columna
36 -         %se adicionan ceros
37 -         im_aumentada=[zeros(filas,n_f_c) xi zeros(filas,n_f_c)];
38
39 -         %se aumenta la primera fila y la última fila
40 -         %se adicionan ceros
41 -         tam_im_aumentada=size(im_aumentada);
42 -         filas_im_aumentada=tam_im_aumentada(1,1);%filas
43 -         columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
44
45 -         %imagen aumentada en ceros
46 -         im_aumentada_final=[zeros(n_f_c,columnas_im_aumentada);im_aumentada;zeros(n_f_c,columnas_im_aumentada)];
47 -         figure('name','Imagen aumentada con ceros');mesh(xi)%dibujo imagen aumentada
48
49 -         %se realiza un ciclo para pasar la ventana de convolución
50 -         tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
51 -         cont_filas=1;%contador para filas
52 -         for filas_im=n_f_c+1:tam_im_aumentada_final(1,1)-n_f_c,
53 -             cont_columnas=1;%contador para columnas
54 -             for columnas_im=n_f_c+1:tam_im_aumentada_final(1,2)-n_f_c,
55 -                 %productos entre imagen y máscara
56 -                 res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
57 -                 %se suman los productos
58 -                 im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
59 -                 cont_columnas=cont_columnas+1;
60 -             end
61 -             cont_filas=cont_filas+1;
62 -         end
63 -     end
64 -
65 -
66 -
67 -     case 2
68 -
69 -
70 - end
71
72 - im_convolucionada
```

Ejecute el programa,



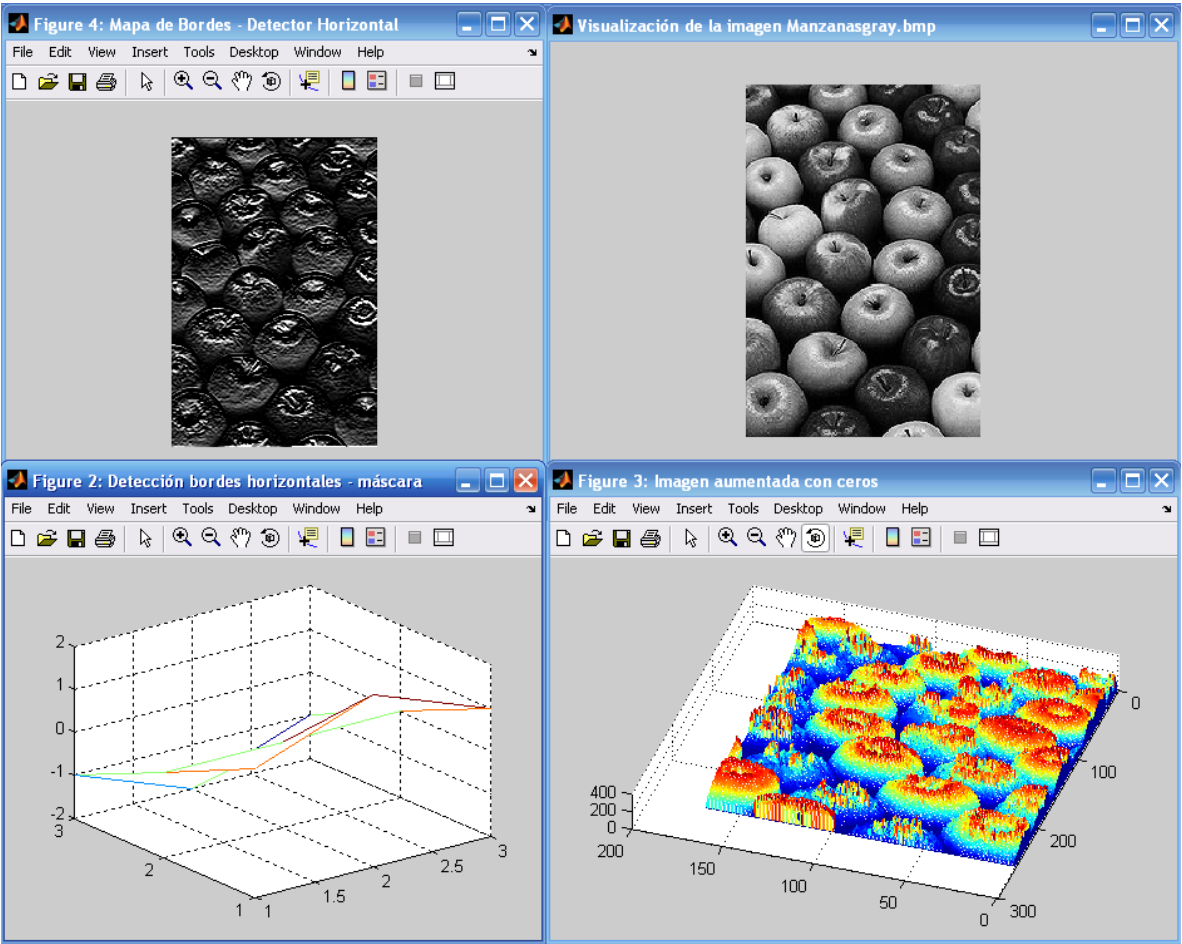
Con este ejemplo se comprueba el funcionamiento del algoritmo de convolución, utilizando una imagen de prueba. Se puede dibujar la matriz resultante (`im_convolucionada`).

Ahora, se utiliza una imagen en escala de grises.


```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ejemplo_conv2.m*
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base

1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9
10 xi=double(xi);
11
12
13 %Determinación de imagen: color o escala de grises
14 tam=size(xi);
15 if length(tam)==2,
16     clase='imagen en escala de grises'
17     caso=1;
18 elseif length(tam)==3,
19     clase='imagen a color'
20     caso=2;
21 end
22
23
24 %construcción de la máscara
25 %Detector de bordes horizontales
26 mk=[1 2 1; 0 0 0; -1 -2 -1];%KERNEL DE CONVOLUCIÓN
27 %Dibujo de la máscara detectora de bordes
28 figure('name','Detección bordes horizontales - máscara');mesh(mk)
29
30 %se toma el tamaño de la máscara
31 tam_mask=size(mk);
32 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
33 %se deben aumentar a la imagen - ceros(primer solución)
34
35 switch caso
36 case 1
37     %tamaño de la imagen
38     filas=tam(1,1);
39     columnas=tam(1,2);
40     %se aumenta la primera columna y la última columna
41     %se adicionan ceros
42     im_aumentada=zeros(filas,n_f_c) xi zeros(filas,n_f_c)];
43
44     %se aumenta la primera fila y la última fila
45     %se adicionan ceros
46     tam_im_aumentada=size(im_aumentada);
47     filas_im_aumentada=tam_im_aumentada(1,1);%filas
48     columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
49
50     %imagen aumentada en ceros
51     im_aumentada_final=zeros(n_f_c,columnas_im_aumentada);im_aumentada;zeros(n_f_c,columnas_im_aumentada)];
52     figure('name','Imagen aumentada con ceros');mesh(im_aumentada_final)%dibujo imagen aumentada
53
54     %se realiza un ciclo para pasar la ventana de convolución
55     tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
56     cont_filas=1;%contador para filas
57     for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
58         cont_columnas=1;%contador para columnas
59         for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
60             %productos entre imagen y máscara
61             res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
62             %se suman los productos
63             im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
64             cont_columnas=cont_columnas+1;
65
66         end
67         cont_filas=cont_filas+1;
68     end
69 end
70
71 case 2
72
73
74 end
75
76
77 figure('name','Mapa de Bordos - Detector Horizontal');imshow(uint8(im_convolucionada))
78
79
```

Se ejecuta el código,

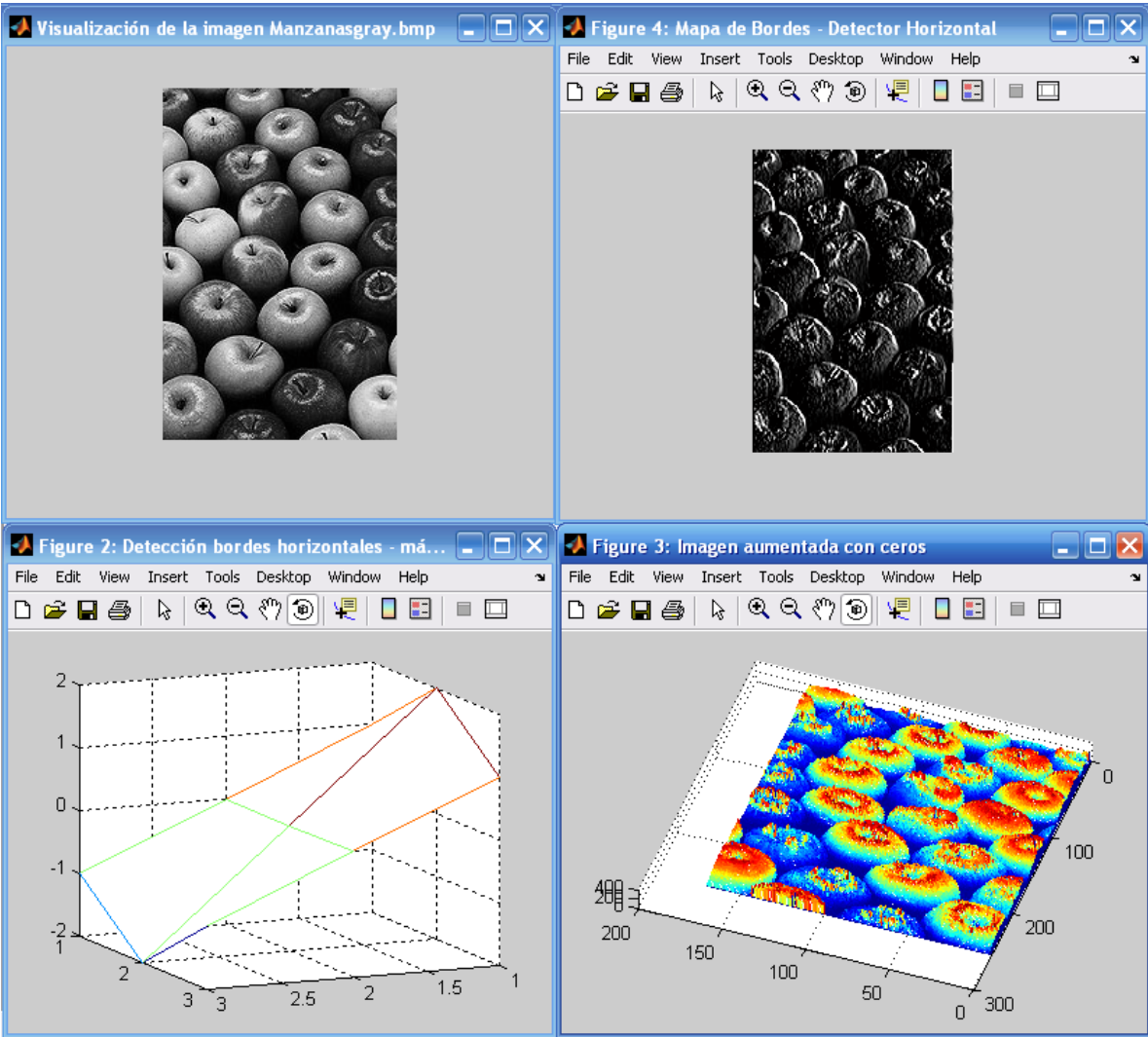


Cambie la ventana de convolución (kernel) por una máscara detectora de bordes verticales,

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ejemplo_conv3.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base

19 -     clase='imagen a color'
20 -     caso=2;
21 - end
22
23
24 %construcción de la máscara
25 %Detector de bordes verticales
26 mk=[1 0 -1;2 0 -2;1 0 -1];%KERNEL DE CONVOLUCIÓN
27 %Dibujo de la máscara detectora de bordes
28 figure('name','Detección bordes horizontales - máscara');,mesh(mk)
29
30 %se toma el tamaño de la máscara
31 tam_mask=size(mk);
32 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
33 %se deben aumentar a la imagen - ceros(primer solución)
34
35 switch caso
36     case 1
37         %tamaño de la imagen
38         filas=tam(1,1);
39         columnas=tam(1,2);
40         %se aumenta la primera columna y la última columna
41         %se adicionan ceros
42         im_aumentada=zeros(filas,n_f_c) x1 zeros(filas,n_f_c)];
43
44         %se aumenta la primera fila y la última fila
45         %se adicionan ceros
46         tam_im_aumentada=size(im_aumentada);
47         filas_im_aumentada=tam_im_aumentada(1,1).%filas
```

El resultado de ejecutar el código es,



Se observa una detección de los bordes horizontales en la imagen. El procedimiento anterior se utiliza en una imagen a color, con un detector de bordes verticales. El código se muestra a continuación,

```

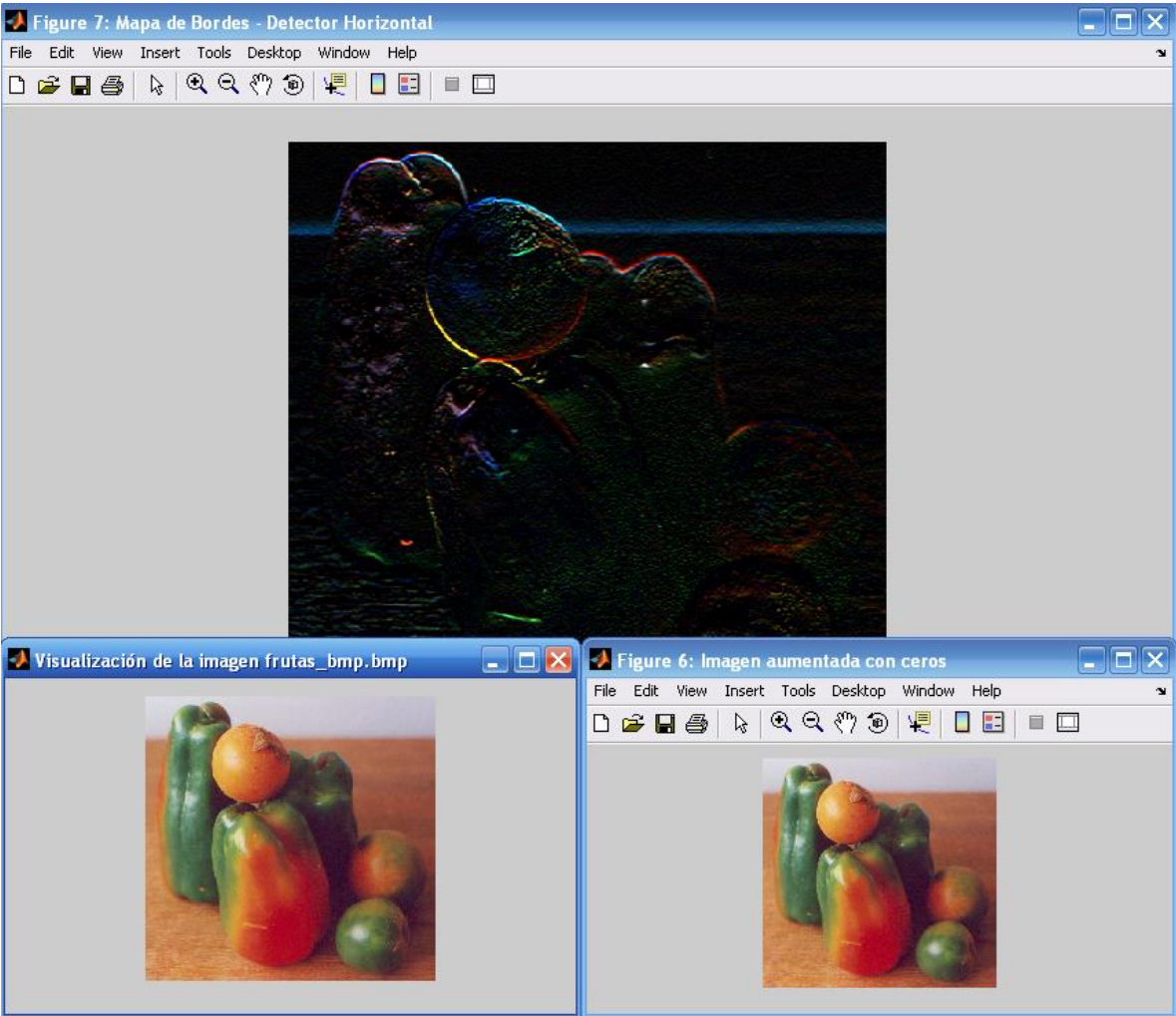
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numberTitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54675]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);%conversión a doble precisión
10
11
12 %Determinación de imagen: color o escala de grises
13 tam=size(xi);
14 if length(tam)==2,
15     clase='imagen en escala de grises'
16     caso=1;
17 elseif length(tam)==3,
18     clase='imagen a color'
19     caso=2;
20 end
21
22 %construcción de la máscara
23 %Detector de bordes verticales
24 mk=[1 0 -1;2 0 -2;1 0 -1];%KERNEL DE CONVOLUCIÓN
25 %Dibujo de la máscara detectora de bordes
26 figure('name','Detección bordes horizontales - máscara');mesh(mk)
27
28 %se toma el tamaño de la máscara
29 tam_mask=size(mk);
30 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
31 %se deben aumentar a la imagen - ceros(primer solución)
32
33 switch caso
34 case 1%caso de imagen a escala de grises
35     %tamaño de la imagen
36     filas=tam(1,1);
37     columnas=tam(1,2);
38     %se aumenta la primera columna y la última columna
39     %se adicionan ceros
40     im_aumentada=[zeros(filas,n_f_c) xi zeros(filas,n_f_c)];
41
42     %se aumenta la primera fila y la última fila
43     %se adicionan ceros
44     tam_im_aumentada=size(im_aumentada);
45     filas_im_aumentada=tam_im_aumentada(1,1);%filas
46     columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
47
48     %imagen aumentada en ceros
49     im_aumentada_final=[zeros(n_f_c,columnas_im_aumentada);im_aumentada;zeros(n_f_c,columnas_im_aumentada)];
50     figure('name','Imagen aumentada con ceros');mesh(im_aumentada_final)%dibujo imagen aumentada
51
52     %se realiza un ciclo para pasar la ventana de convolución
53     tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
54     cont_filas=1;%contador para filas
55     for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
56         cont_columnas=1;%contador para columnas
57         for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
58             %productos entre imagen y máscara
59             res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
60             %se suman los productos
61             im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
62             cont_columnas=cont_columnas+1;
63
64         end
65         cont_filas=cont_filas+1;
66     end
67
68 case 2%caso de imagen a color
69     %tamaño de la imagen
70     filas=tam(1,1);
71     columnas=tam(1,2);
72     %se aumenta la primera columna y la última columna
73     %se adicionan ceros
74     im_aumentada_R=[zeros(filas,n_f_c) xi(:,1) zeros(filas,n_f_c)];%Para la componente rojo
75     im_aumentada_G=[zeros(filas,n_f_c) xi(:,2) zeros(filas,n_f_c)];%Para la componente verde
76     im_aumentada_B=[zeros(filas,n_f_c) xi(:,3) zeros(filas,n_f_c)];%Para la componente azul
77
78     %se aumenta la primera fila y la última fila
79     %se adicionan ceros
80     tam_im_aumentada=size(im_aumentada_R);%para el rojo como referencia
81     filas_im_aumentada=tam_im_aumentada(1,1);%filas
82     columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
83
84     %imagen aumentada en ceros(Rojo, Verde y Azul)
85     im_aumentada_final_R=[zeros(n_f_c,columnas_im_aumentada);im_aumentada_R;zeros(n_f_c,columnas_im_aumentada)];
86     im_aumentada_final_G=[zeros(n_f_c,columnas_im_aumentada);im_aumentada_G;zeros(n_f_c,columnas_im_aumentada)];
87     im_aumentada_final_B=[zeros(n_f_c,columnas_im_aumentada);im_aumentada_B;zeros(n_f_c,columnas_im_aumentada)];
88     figure('name','Imagen aumentada con ceros comp. roja');mesh(im_aumentada_final_R)%dibujo imagen aumentada comp. rojo
89     figure('name','Imagen aumentada con ceros comp. verde');mesh(im_aumentada_final_G)%dibujo imagen aumentada comp. verd
90     figure('name','Imagen aumentada con ceros comp. azul');mesh(im_aumentada_final_B)%dibujo imagen aumentada comp. azul
91
92     im_aumentada_final_RGB(:,1)=im_aumentada_final_R;
93     im_aumentada_final_RGB(:,2)=im_aumentada_final_G;
94     im_aumentada_final_RGB(:,3)=im_aumentada_final_B;
95     figure('name','Imagen aumentada con ceros');imshow(uint8(im_aumentada_final_RGB))%visualización imagen aumentada
96
97     %se realiza un ciclo para pasar la ventana de convolución
98     tam_im_aumentada_final=size(im_aumentada_final_R);%tamaño imagen aumentada final- comp. rojo como ref.
99     cont_filas=1;%contador para filas
100    for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
101        cont_columnas=1;%contador para columnas
102        for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
103            %productos entre imagen y máscara
104            res_mult_R=im_aumentada_final_R(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
105            res_mult_G=im_aumentada_final_G(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
106            res_mult_B=im_aumentada_final_B(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
107            %se suman los productos
108            im_convolucionada_R(cont_filas,cont_columnas)=(sum(res_mult_R(:)));
109            im_convolucionada_G(cont_filas,cont_columnas)=(sum(res_mult_G(:)));
110            im_convolucionada_B(cont_filas,cont_columnas)=(sum(res_mult_B(:)));
111            cont_columnas=cont_columnas+1;
112
113        end
114        cont_filas=cont_filas+1;
115    end
116
117 end
118
119 %concatenan las matrices para formar la
120 %imagen a color
121 im_convolucionada_RGB(:,1)=im_convolucionada_R;
122 im_convolucionada_RGB(:,2)=im_convolucionada_G;
123 im_convolucionada_RGB(:,3)=im_convolucionada_B;
124 %visualización de la imagen de bordes (mapa de bordes a color)
125 figure('name','Mapa de Bordes - Detector Horizontal');imshow(uint8(im_convolucionada_RGB))

```

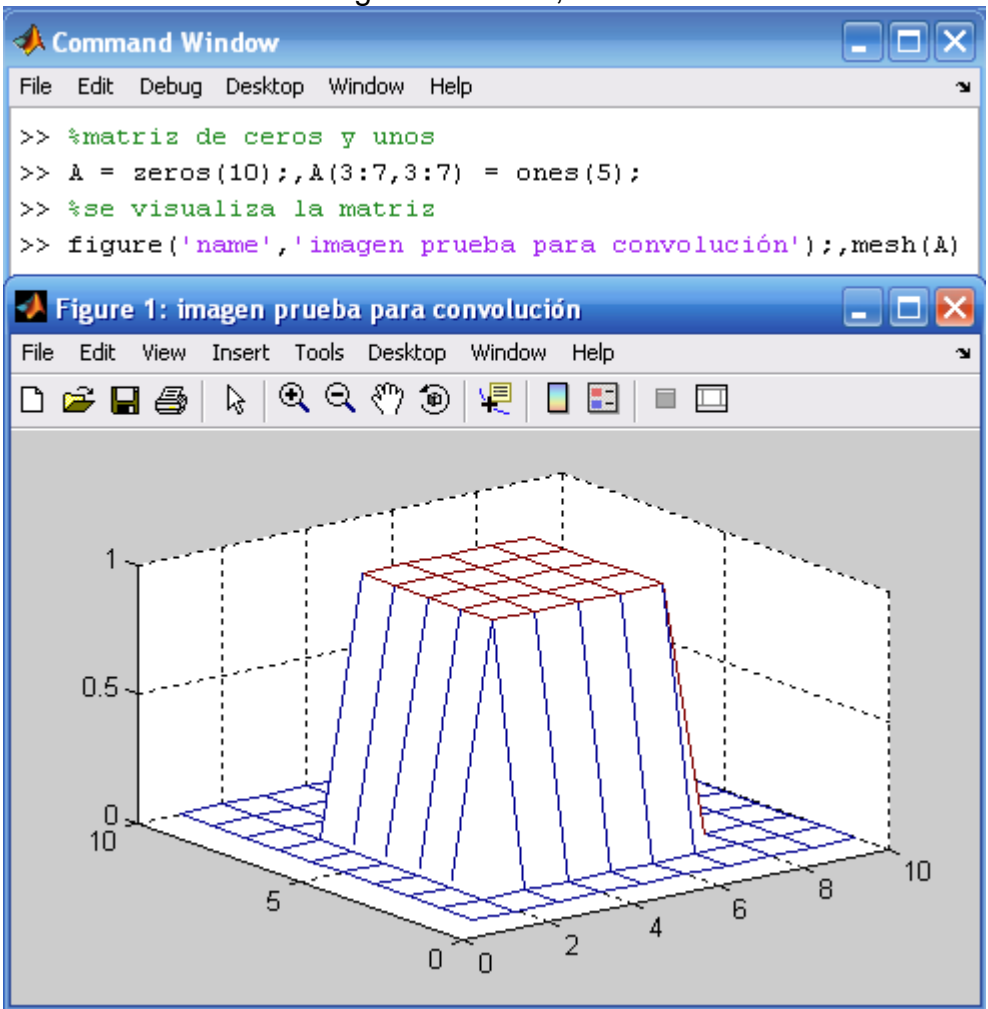
Figure 3: Imagen aumentada con ceros comp. roja

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ejemplo_conv3.m
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
11
12 %Determinación de imagen: color o escala de grises
13 tam=size(xi);
14 if length(tam)==2,
15     clase='imagen en escala de grises'
16     caso=1;
17 elseif length(tam)==3,
18     clase='imagen a color'
19     caso=2;
20 end
21
22 %construcción de la máscara
23 %Detector de bordes horizontales
24 mk=[1 2 1;0 0 0;-1 -2 -1];%KERNEL DE CONVOLUCIÓN
25 %Dibujo de la máscara detectora de bordes
26 figure('name','Detección bordes horizontales - máscara');,mesh(mk)
27
28 %se toma el tamaño de la máscara
29 tam_mask=size(mk);
30 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
31 %se deben aumentar a la imagen - ceros(primer solución)
32
33 switch caso
34 case 1%caso de imagen a escala de grises
35     %tamaño de la imagen
36     filas=tam(1,1);
37     columnas=tam(1,2);
38     %se aumenta la primera columna y la última columna
```

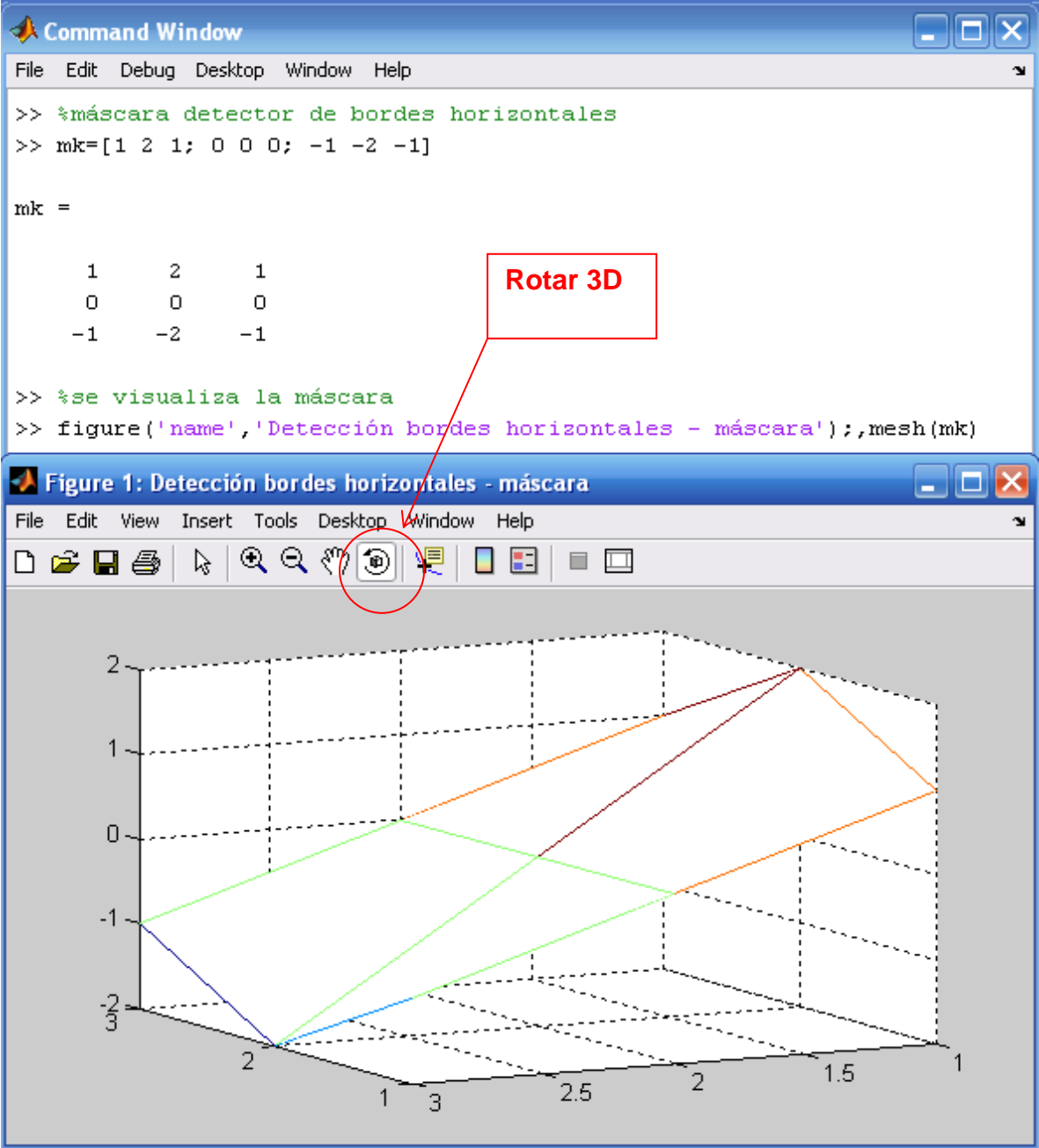
13



ii. Segunda Solución. Se convoluciona la imagen sin sobrepasar los límites de la imagen.
Se toma la siguiente matriz,



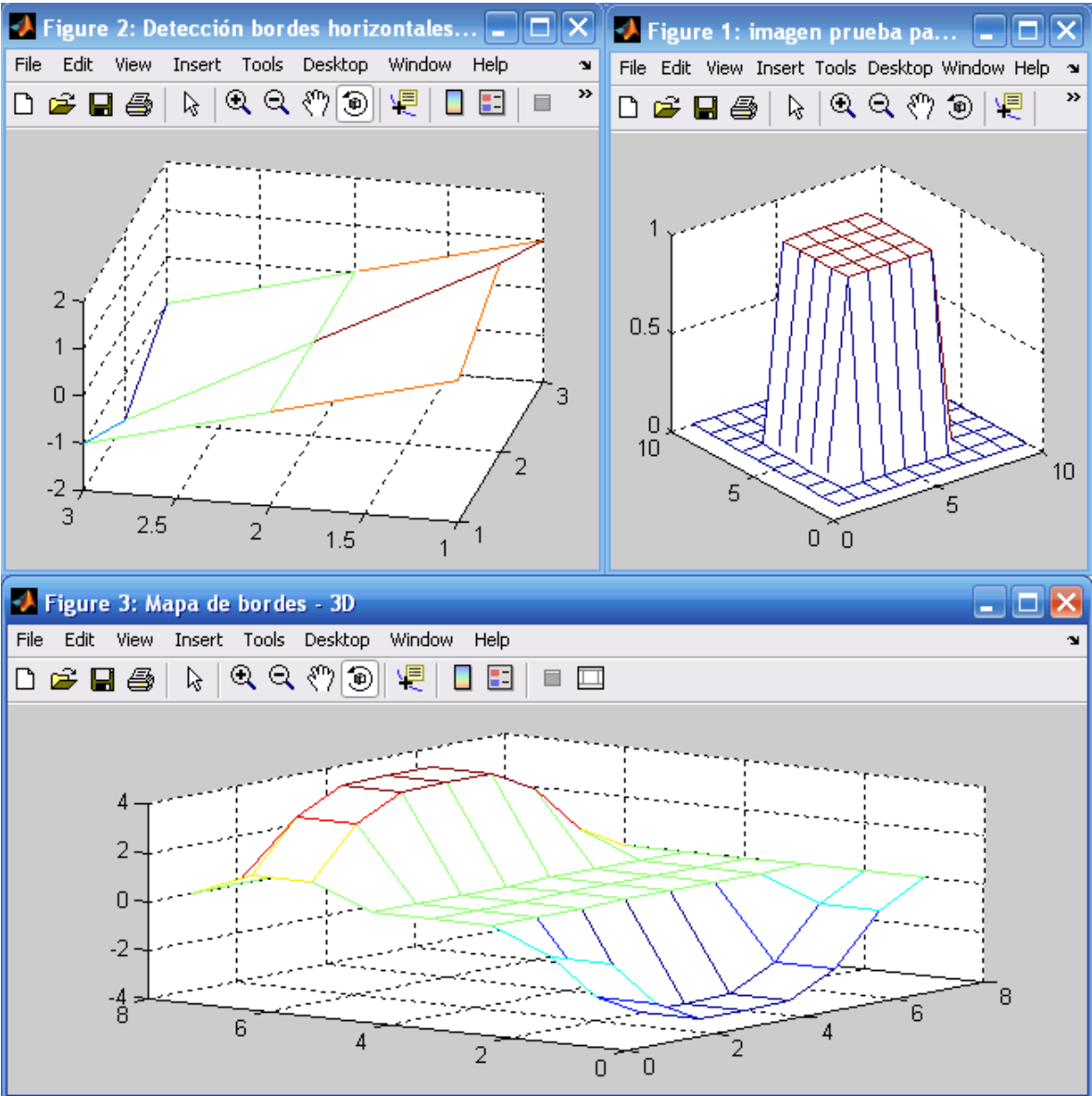
Se aplicará una máscara que detecta bordes verticales,



Se plantea el siguiente código de programación que realiza la convolución

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv1_segunda_solucion.m
File Edit Text Cell Tools Debug Desktop Window Help
1 - clear%borra variables en el workspace
2 - %Función pulso rectangular en 2D
3 - %representa a una imagen
4 - xi = zeros(10);,xi(3:7,3:7) = ones(5);
5 - %se dibuja
6 - figure('name','imagen prueba para convolución');,mesh(xi)
7
8 - %Determinación de imagen: color o escala de grises
9 - tam=size(xi);
10 - if length(tam)==2,
11 -     clase='imagen en escala de grises'
12 -     caso=1;
13 - elseif length(tam)==3,
14 -     clase='imagen a color'
15 -     caso=2;
16 - end
17
18 - %construcción de la máscara
19 - %Detectora de bordes horizontales
20 - mk=[1 2 1; 0 0 0; -1 -2 -1];%KERNEL DE CONVOLUCIÓN
21 - %Dibujo de la máscara detectora de bordes
22 - figure('name','Detección bordes horizontales - máscara');,mesh(mk)
23
24 - %se toma el tamaño de la máscara
25 - tam_mask=size(mk);
26 - n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
27 - %se deben aumentar a la imagen - ceros(primer solución)
28
29 - switch caso
30 -     case 1%caso imagen en escala de grises
31 -         %tamaño de la imagen
32 -         filas=tam(1,1);
33 -         columnas=tam(1,2);
34
35 -         cont_filas=1;%contador para filas
36 -         %Segunda Solución - No se aumenta de tamaño la imagen
37 -         for filas_im=n_f_c+1:1:filas-n_f_c,
38 -             cont_columnas=1;%contador para columnas
39 -             for columnas_im=n_f_c+1:1:columnas-n_f_c,
40 -                 %productos entre imagen y máscara
41 -                 res_mult=xi(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
42 -                 %se suman los productos
43 -                 im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
44 -                 cont_columnas=cont_columnas+1;
45 -             end
46 -             cont_filas=cont_filas+1;
47 -         end
48 -     end
49
50 -     case 2%caso imagen a color
51 -
52 - end
53
54 - figure('name','Mapa de bordes - 3D');,mesh(im_convolucionada)
```

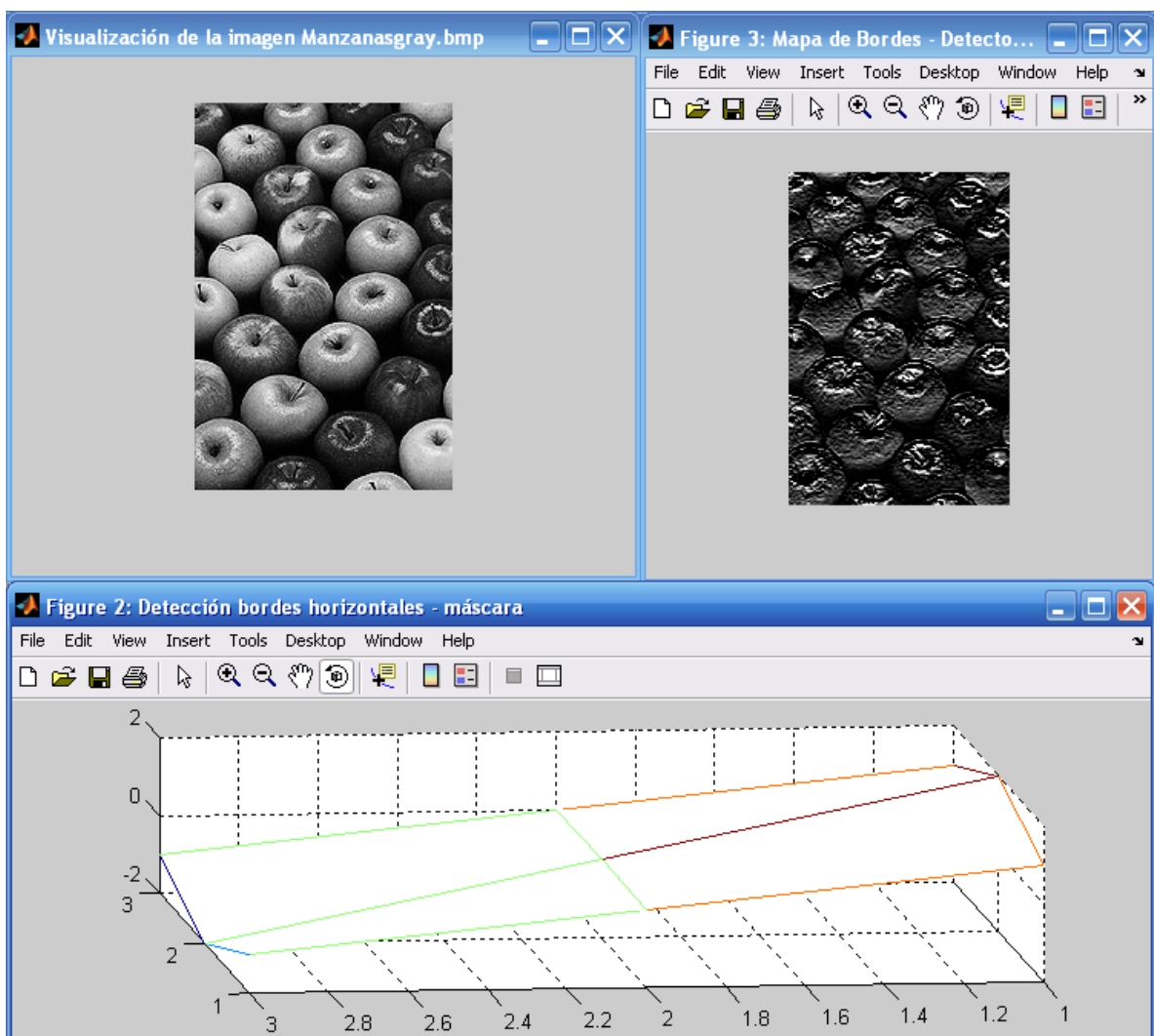
El resultado es,



Con base en el anterior resultado, se prueba con una imagen en escala de grises.

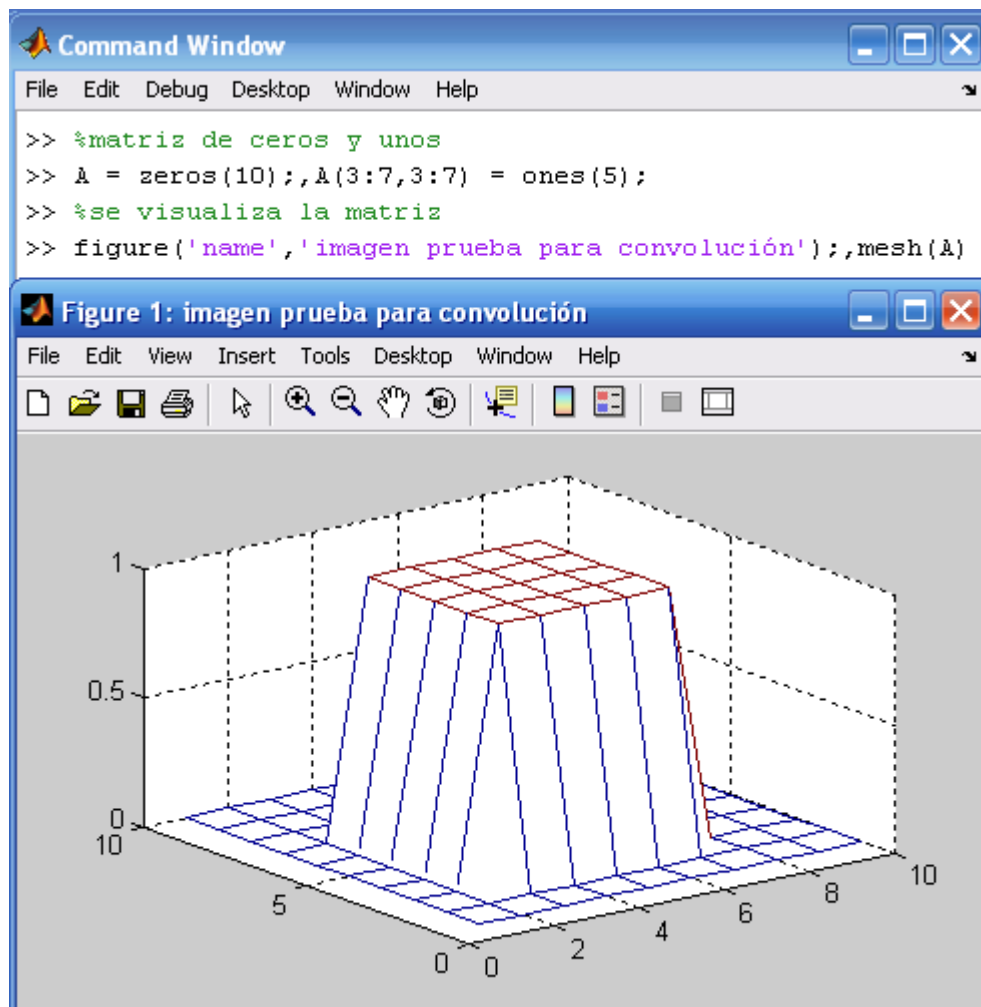
```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_segunda_solucion.m
File Edit Text Cell Tools Debug Desktop Window Help
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 fi=figure('numbertitle','off','name',[ 'Visualización de la imagen ' FILENAME], 'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Detector de bordes horizontales
23 mk=[1 2 1; 0 0 0; -1 -2 -1];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara detectora de bordes
25 figure('name','Detección bordes horizontales - máscara');,mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - ceros(primer solución)
31
32 switch caso
33     case 1%caso imagen a escala de grises
34         %tamaño de la imagen
35         filas=tam(1,1);
36         columnas=tam(1,2);
37
38         cont_filas=1;%contador para filas
39         %Segunda Solución - No se aumenta de tamaño la imagen
40         for filas_im=n_f_c+1:filas-n_f_c,
41             cont_columnas=1;%contador para columnas
42             for columnas_im=n_f_c+1:columnas-n_f_c,
43                 %productos entre imagen y máscara
44                 res_mult=xi(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
45                 %se suman los productos
46                 im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
47                 cont_columnas=cont_columnas+1;
48             end
49             cont_filas=cont_filas+1;
50         end
51     case 2%caso imagen a color
52
53
54 end
55
56 figure('name','Mapa de Bordes - Detector Horizontal');,imshow(uint8(im_convolucionada))
```

El resultado es el siguiente,

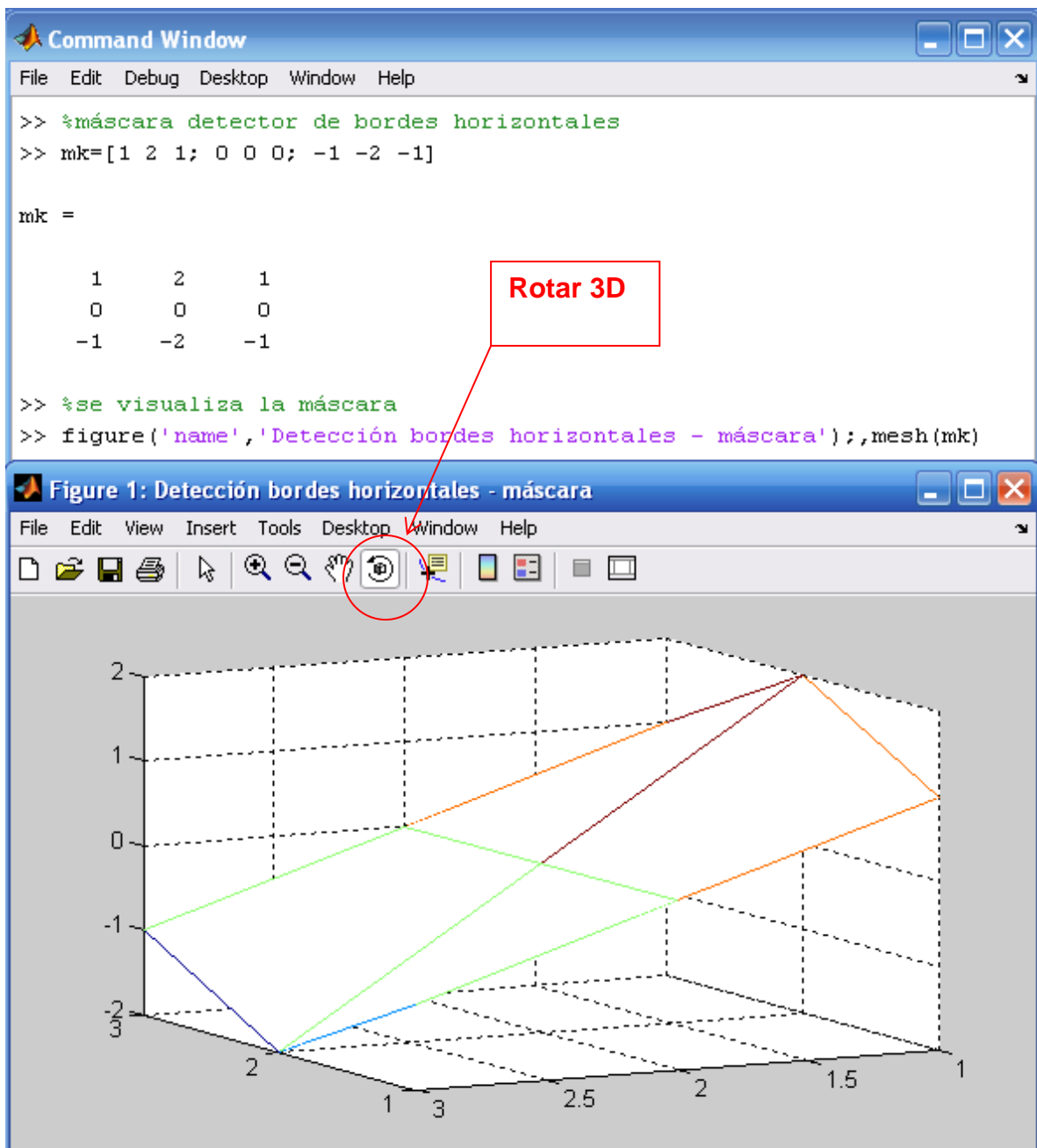


Realizar el código para *convolucionar* una imagen a color utilizando el concepto de segunda solución.

- iii. Tercera Solución. Se aumenta el tamaño de la imagen duplicando sus filas y columnas.
Se toma la siguiente matriz,



Se aplicará una máscara que detecta bordes verticales,

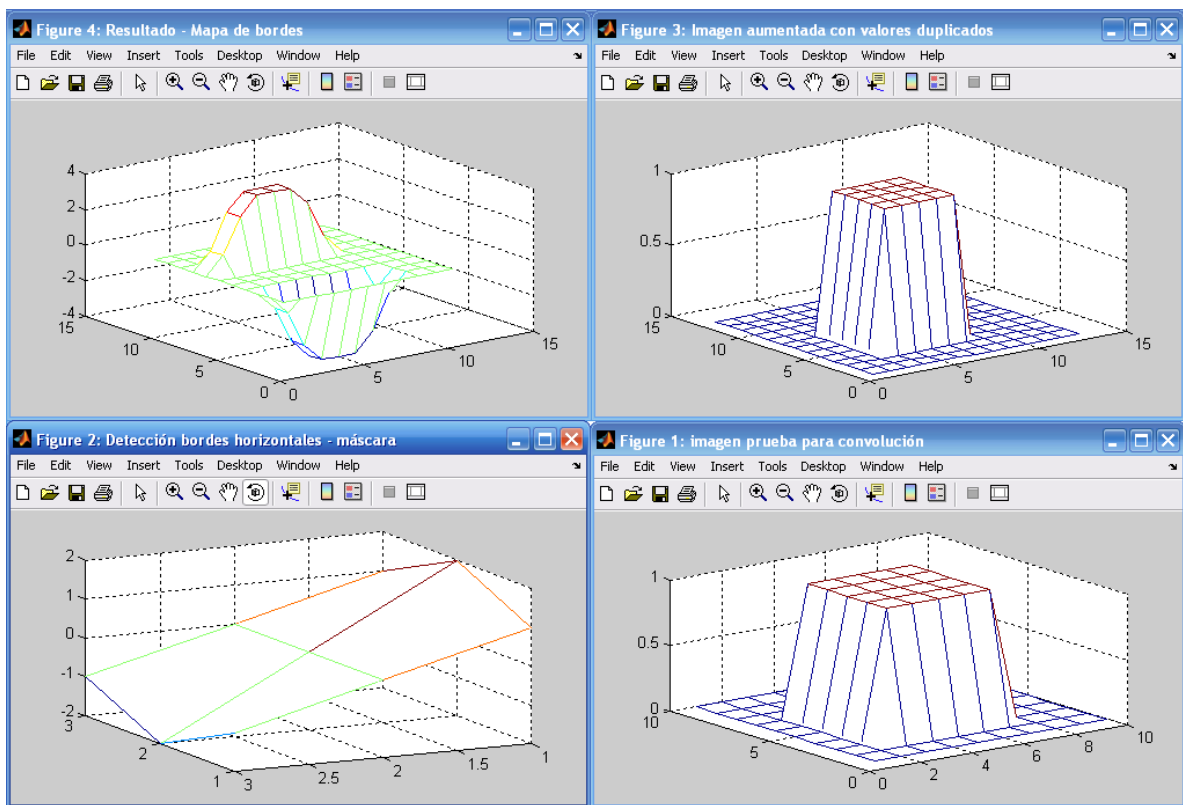


Se plantea el siguiente código de programación que realiza la convolución, con base en la tercera solución.

```
Editor - C:\Program Files\MATLAB71\work\especializacion_teleco_9_Febrero_2008\ej_conv1_tercera_solucion.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base

1 - clear%borra variables en el workspace
2 - %Función pulso rectangular en 2D
3 - %representa a una imagen
4 - xi = zeros(10);xi(3:7,3:7)=ones(5);
5 - %se dibuja
6 - figure('name','imagen prueba para convolución');mesh(xi)
7
8
9 - %Determinación de imagen: color o escala de grises
10 - tam=size(xi);
11 - if length(tam)==2,
12 -     clase='imagen en escala de grises'
13 -     caso=1;
14 - elseif length(tam)==3,
15 -     clase='imagen a color'
16 -     caso=2;
17 - end
18
19 - %construcción de la máscara
20 - %Detectora de bordes horizontales
21 - mk=[1 2 1; 0 0 0; -1 -2 -1];%KERNEL DE CONVOLUCIÓN
22 - %Dibujo de la máscara detectora de bordes
23 - figure('name','Detección bordes horizontales - máscara');mesh(mk)
24
25 - %se toma el tamaño de la máscara
26 - tam_mask=size(mk);
27 - n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
28 - %se deben aumentar a la imagen - ceros(primer solución)
29
30 - switch caso
31 -     case 1%caso imagen escala de grises
32 -         %tamaño de la imagen
33 -         filas=tam(1,1);
34 -         columnas=tam(1,2);
35 -         %se aumenta la primera columna y la última columna
36 -         %se duplican valores de imagen
37 -         im_aumentada=[xi(1:filas,1:n_f_c) xi xi(1:filas,columnas-n_f_c:columnas)];
38
39 -         %se aumenta la primera fila y la última fila
40 -         %se duplican valores de imagen
41 -         tam_im_aumentada=size(im_aumentada);
42 -         filas_im_aumentada=tam_im_aumentada(1,1);%filas
43 -         columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
44
45 -         %imagen aumentada con valores duplicados
46 -         im_aumentada_final=[im_aumentada(1:n_f_c,:);im_aumentada;im_aumentada(filas_im_aumentada-n_f_c:filas_im_aumentada,:)];
47 -         figure('name','Imagen aumentada con valores duplicados');mesh(im_aumentada_final)%dibujo imagen aumentada
48
49 -         %se realiza un ciclo para pasar la ventana de convolución
50 -         tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
51 -         cont_filas=1;%contador para filas
52 -         for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
53 -             cont_columnas=1;%contador para columnas
54 -             for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
55 -                 %productos entre imagen y máscara
56 -                 res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
57 -                 %se suman los productos
58 -                 im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
59 -                 cont_columnas=cont_columnas+1;
60 -             end
61 -             cont_filas=cont_filas+1;
62
63 -         end
64
65 -     case 2%caso imagen a color
66
67 - end
68
69 - figure('name','Resultado - Mapa de bordes');mesh(im_convolucionada)
70
```

El resultado se muestra a continuación,



Comparar todos los resultados anteriores con la función “conv2” de Matlab.

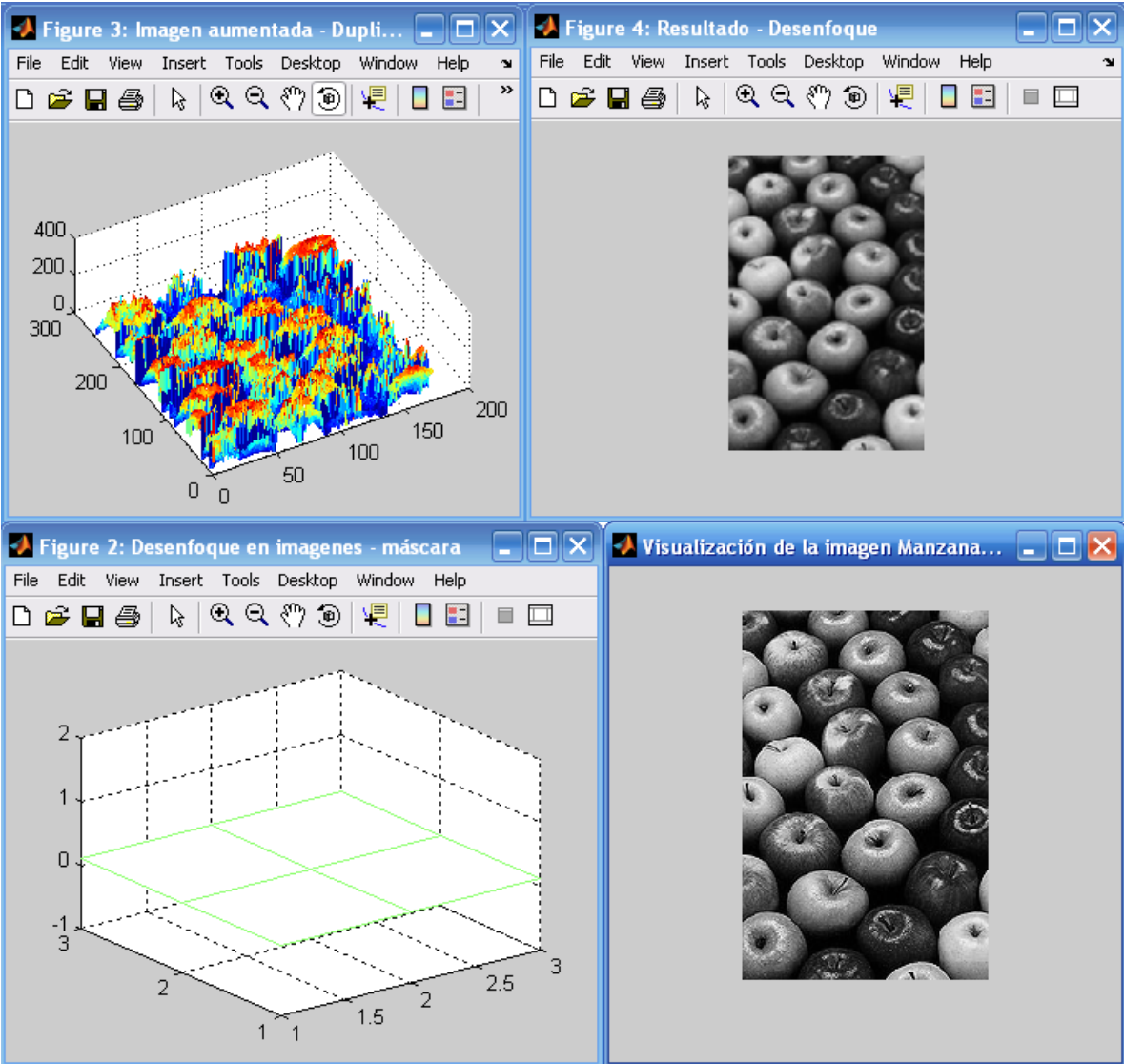
- Efecto de desenfoque.
 - i. Su expresión en inglés es “blurring” o **filtrado pasa-bajo espacial**, remueve detalles finos de una imagen.
 - ii. Se utiliza para simular el efecto de cámara llamado fuera de foco.
 - iii. El desenfoque se realiza a través del proceso de área llamado convolución.
 - iv. En una máscara de desenfoque común todos los coeficientes son iguales. Por ejemplo, para una máscara de 3x3 se tiene en cada coeficiente es $1/9$ y para una máscara de 5x5 cada coeficiente es $1/25$. Esto tiene el efecto de promediado de vecinos.
 - v. El promediado de pixeles tiende a remover valores extremos de un grupo. Entre mas grande sea la máscara, mayor será el efecto de desenfoque.
 - vi. El promediado es una forma efectiva de reducir el ruido Gaussiano en una imagen. Además, no es muy efectivo con el ruido impulsivo.
 - vii. El promediado en una imagen tiende a bajar el contraste de una imagen. Esto implica que valores altos y bajos son cambiados hacia valores medios.

Se toma una imagen en escala de grises y se observa el efecto de cada máscara de desenfoque de acuerdo a su tamaño. Para ello, se utiliza el código de convolución con base en la tercera solución, es decir, se duplican las filas y columnas de la imagen.

Para una máscara de 3x3

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_grises_desenfoque.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Para Desenfoque (Filtro Espacial Pasa-Bajos)
23 mk=[1/9 1/9 1/9;1/9 1/9 1/9 1/9 1/9];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara detectora de bordes
25 figure('name','Detección bordes horizontales - máscara');mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - ceros(primer solución)
```

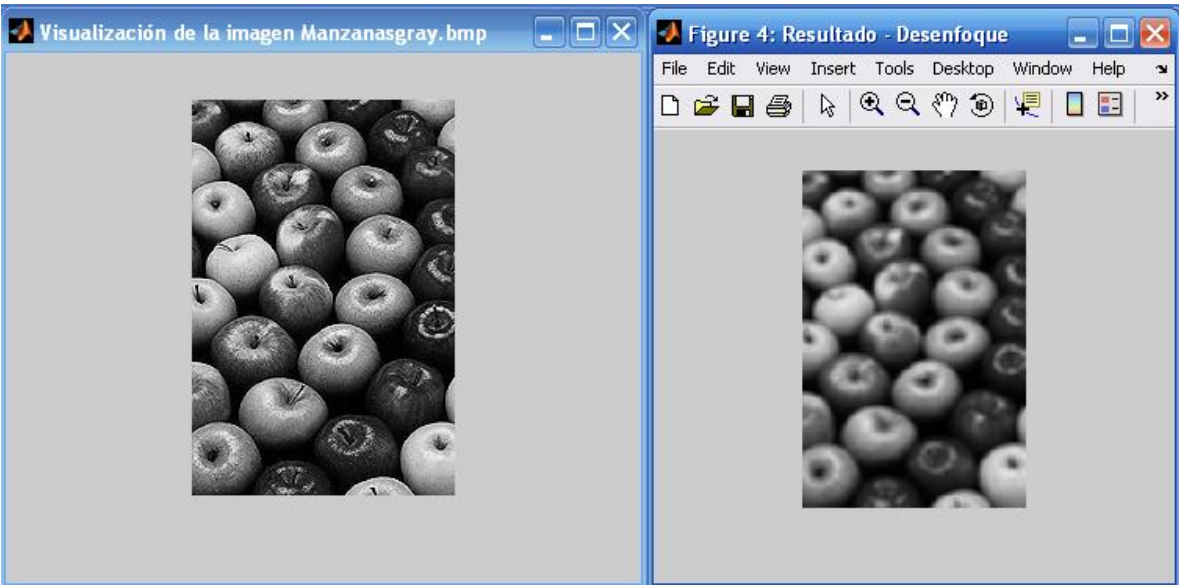
Con el siguiente resultado,



Para una máscara de 5x5,


```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_grises_desenfoque...
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base

1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Para Desenfoque (Filtro Espacial Pasa-Bajos)
23 mk=[1/25 1/25 1/25 1/25 1/25;1/25 1/25 1/25 1/25 1/25;1/25 1/25 1/25 1/25 1/25;...
24     1/25 1/25 1/25 1/25 1/25;1/25 1/25 1/25 1/25 1/25];%KERNEL DE CONVOLUCIÓN
25 %Dibujo de la máscara para desenfoque
26 figure('name','Desenfoque en imágenes - máscara');mesh(mk)
27
28 %se toma el tamaño de la máscara
```



Existe otra máscara de desenfoque,

```
Command Window
File Edit Debug Desktop Window Help

>> %mascara comun para desenfoque de imagenes
>> mk=[1/16 1/8 1/16
        1/8 1/4 1/8
        1/16 1/8 1/16]

mk =

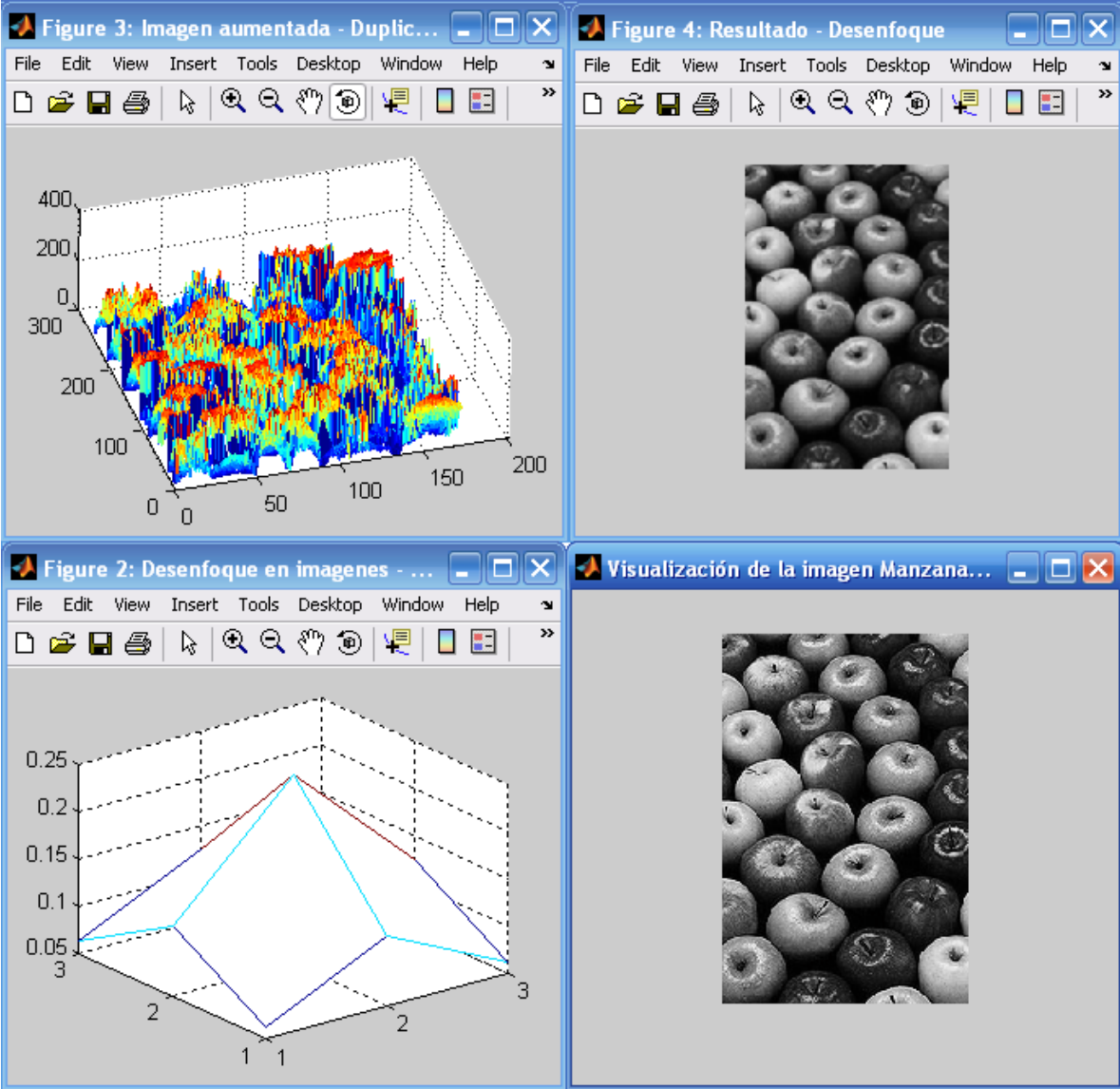
    0.0625    0.1250    0.0625
    0.1250    0.2500    0.1250
    0.0625    0.1250    0.0625

>>
```

Se emplea en el código,

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_grises_desenfoque...
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Para Desenfoque (Filtro Espacial Pasa-Bajos)
23 mk=[1/16 1/8 1/16;1/8 1/4 1/8;1/16 1/8 1/16];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara para desenfoque
25 figure('name','Desenfoque en imágenes - máscara');mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
```

El resultado obtenido es,



Ejercicio: aplicar el desenfoque en imágenes a color.

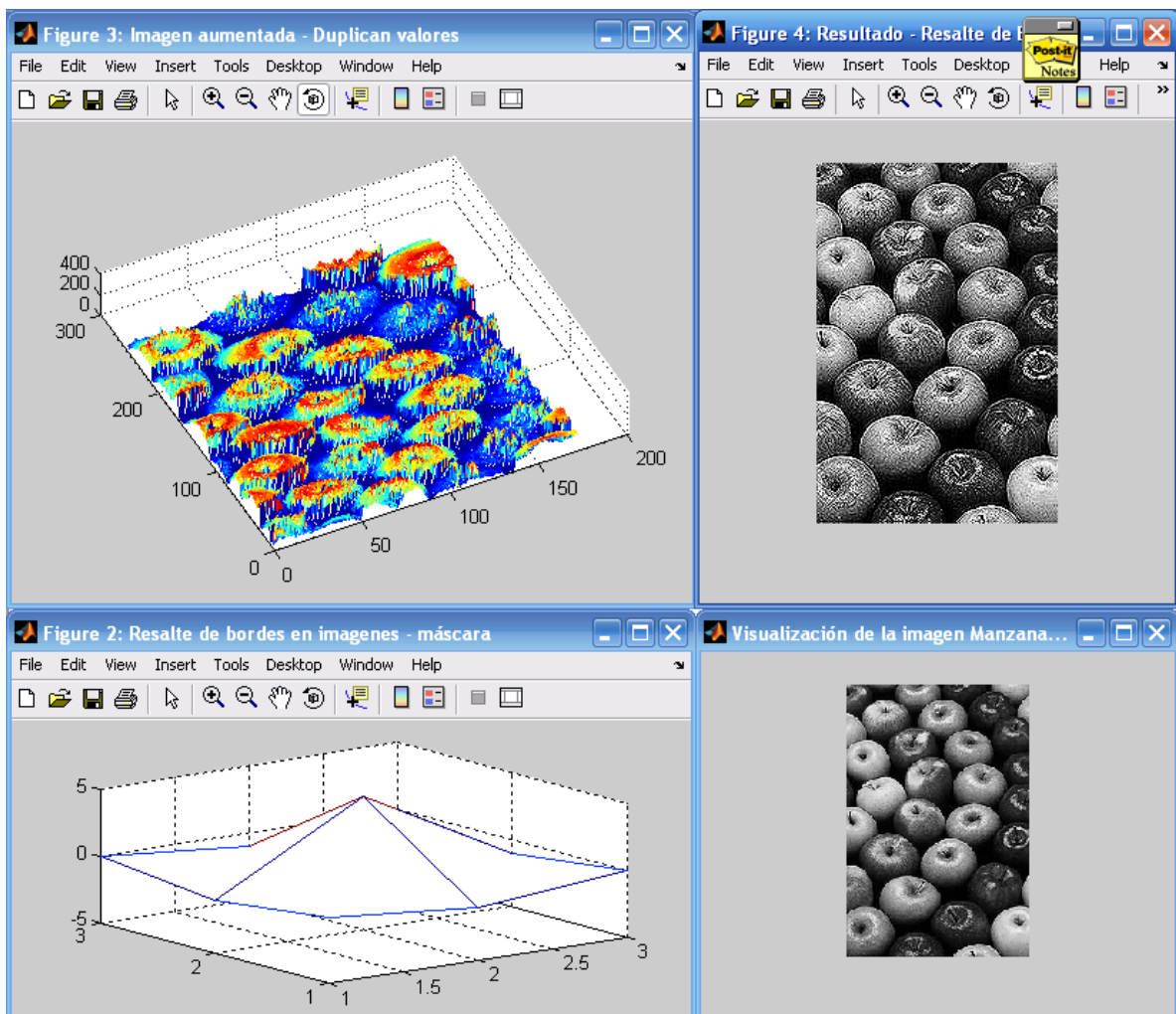
- Resalte de Bordes.
 - i. Tiene el efecto opuesto al desenfoque, ya que enfatiza los detalles de una imagen. Se basa en el filtro pasa-alto espacial. Este removerá las componentes de baja frecuencia.
 - ii. Cuando la imagen está difusa, ya sea antes o durante el proceso de adquisición, es posible restaurarla por medio de un resalte de bordes. El resalte incrementará el contraste de una imagen.
 - iii. La máscara de convolución mas utilizada tiene un coeficiente positivo en su centro y coeficientes negativos alrededor de éste.

Se toma una imagen en escala de grises, se aplica el proceso de area convolución y se observan los resultados todo con base en el planteamiento de la tercera solución (duplicar filas y columnas de la imagen).

```

1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 fl=figure('numberTitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','non
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Resalte de Bordes (Filtro Espacial Pasa-Altas)
23 mk=[0 -1 0;-1 5 -1;0 -1 0];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara para Resalte de Bordes
25 figure('name','Resalte de bordes en imagenes - máscara');,mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - duplicar(tercera solución)
31

```

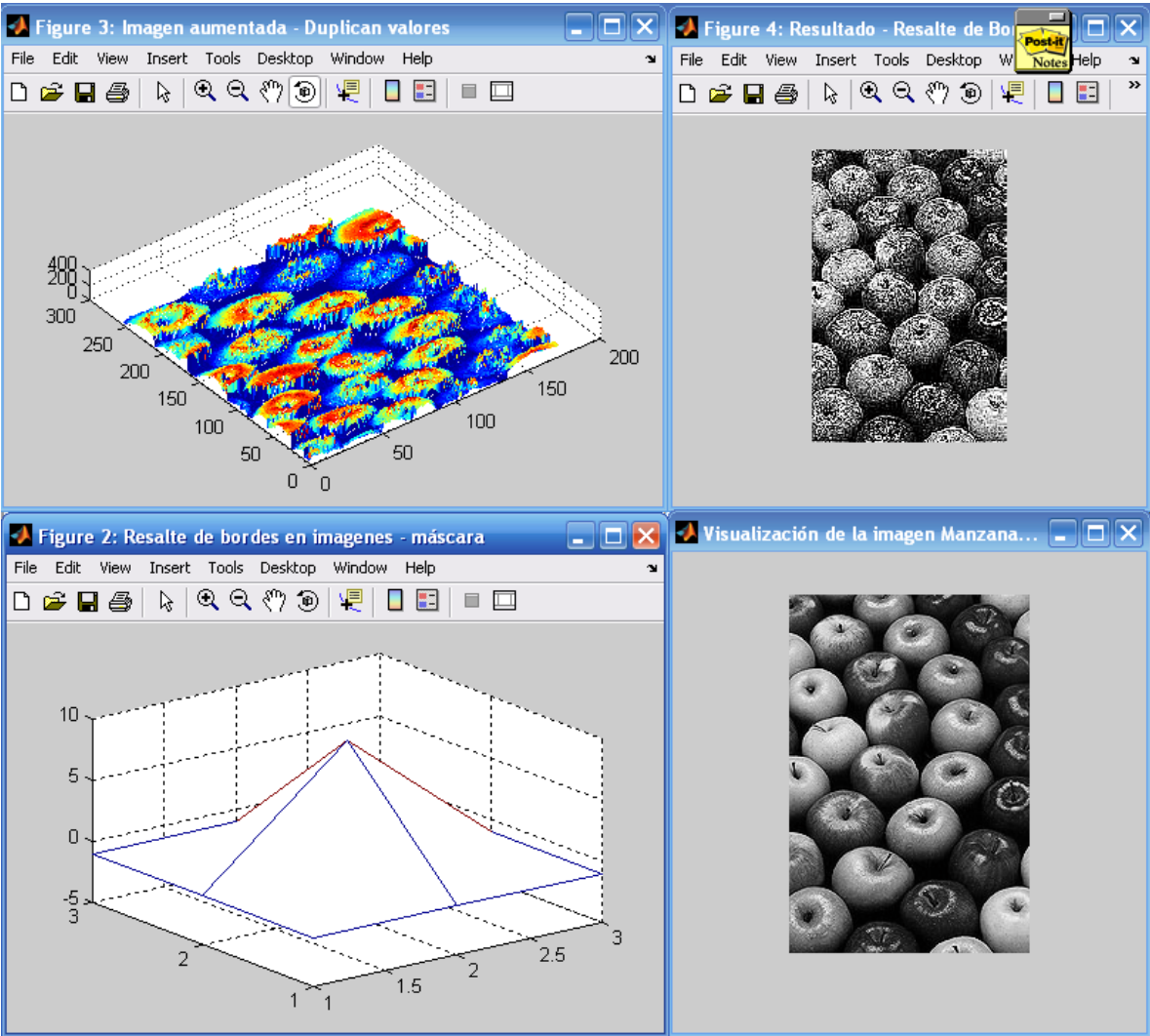


Se utiliza otra máscara para resalte de bordes,

```

1  %se lee la imagen sobre la cual se desea
2  %Aplicar el proceso de convolución
3  [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4  xi=imread([PATHNAME FILENAME]);
5  f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','non
6  'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7  %visualización de la imagen
8  imshow(xi),drawnow
9  xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Resalte de Bordes (Filtro Espacial Pasa-Altas)
23 mk=[-1 -1 -1;-1 9 -1;-1 -1 -1];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara para Resalte de Bordes
25 figure('name','Resalte de bordes en imágenes - máscara');,mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - duplicar(tercera solución)
31

```

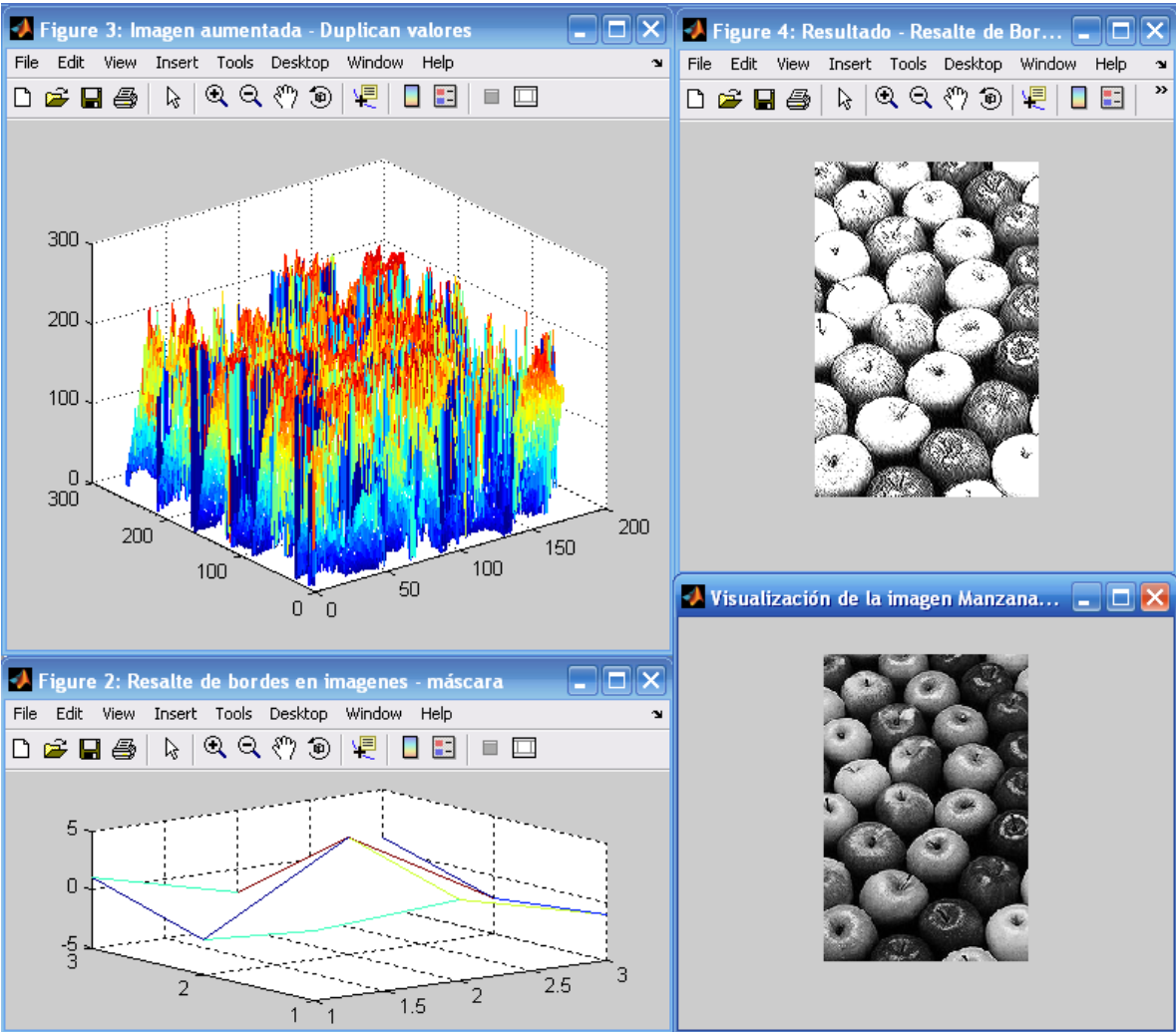


Se utiliza otra máscara para resalte de bordes,

```

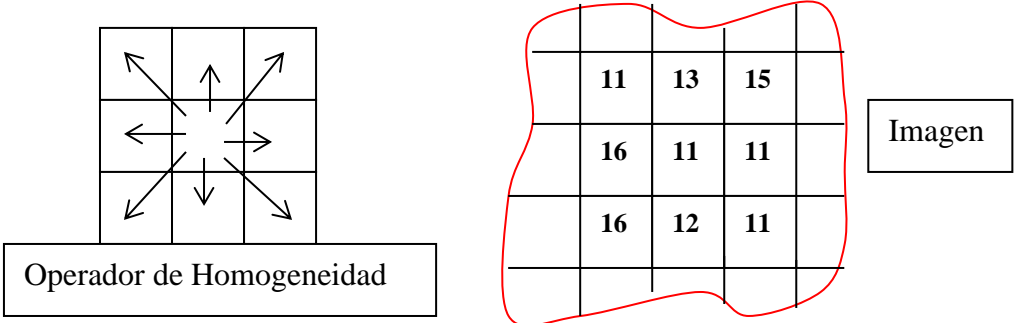
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Resalte de Bordes (Filtro Espacial Pasa-Altas)
23 mk=[1 2 -1;-2 5 -2;1 -2 1];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara para Resalte de Bordes
25 figure('name','Resalte de bordes en imagenes - máscara');mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - duplicar(tercera solución)
31

```

Ejercicio: realizar el resalte de bordes para una imagen a color.

- Detección de bordes.
 - i. Los bordes en una imagen mantienen gran parte de la información de esa imagen. Los bordes dicen donde se encuentran los objetos, sus formas y tamaños.
 - ii. Un borde es donde la intensidad en una imagen se mueve desde un valor alto a un valor bajo o viceversa.
 - iii. La detección de bordes es generalmente el primer paso en la segmentación en imágenes.
 - iv. Existe un conjunto infinito de orientaciones de borde, anchos y formas(perfiles), lo cual genera que algunas veces consuma tiempo de experimentación determinar cual es la mejor técnica para una aplicación.
 - v. El más sencillo y fácil detector de borde determina el máximo valor de una serie de diferencias de pixel. El operador de homogeneidad hace la diferencia de los 8 pixeles vecinos del pixel central de una máscara de 3x3. La salida del operador es el máximo del valor absoluto de cada uno de los pixeles.

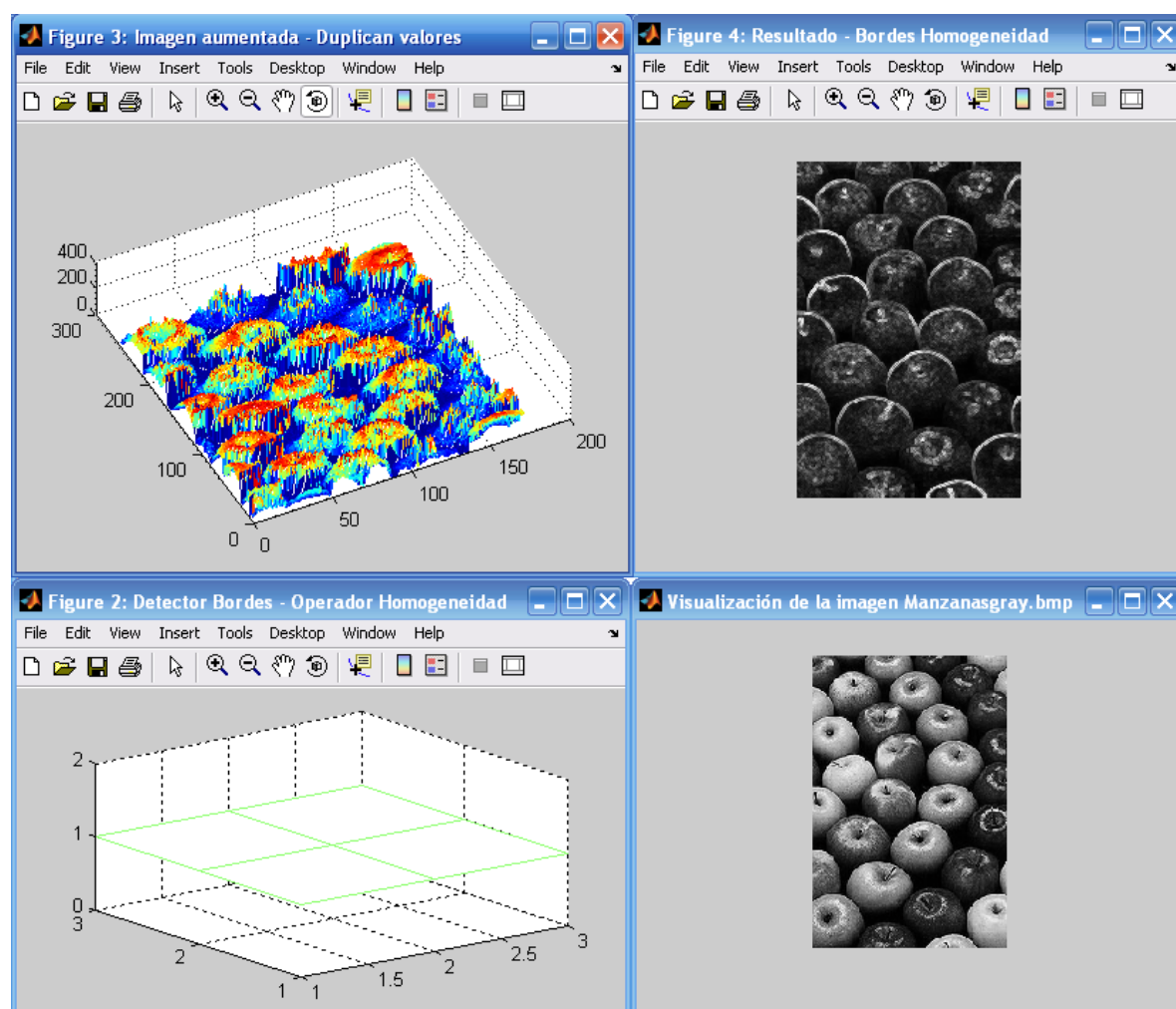


$$new\ pixel = \max\ de\ \{ |11-11| \ |11-13| \ |11-15| \ |11-16| \ |11-11| \ |11-16| \ |11-12| \ |11-11| \} = 5$$

Se toma la imagen en escala de grises, considerando la tercera solución (duplicado de valores de filas y columnas de la imagen).

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_grises_operador_homog.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base

1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     clase='imagen en escala de grises'
15     caso=1;
16 elseif length(tam)==3,
17     clase='imagen a color'
18     caso=2;
19 end
20
21 %construcción de la máscara
22 %Operador de Homogeneidad (Filtro Detector de Bordes)
23 mk=[1 1 1;1 1 1;1 1 1];%KERNEL DE CONVOLUCIÓN
24 %Dibujo de la máscara Bordes Op. Homogeneidad
25 figure('name','Detector Bordes - Operador Homogeneidad');mesh(mk)
26
27 %se toma el tamaño de la máscara
28 tam_mask=size(mk);
29 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
30 %se deben aumentar a la imagen - duplicar (tercera solución)
31
32 switch caso
33     case 1%caso imagen escala de grises
34         %tamaño de la imagen
35         filas=tam(1,1);
36         columnas=tam(1,2);
37         %se aumenta la primera columna y la última columna
38         %se adicionan valores de imagen - duplican
39         im_aumentada=[xi(1:filas,1:n_f_c) xi(1:filas,columnas-n_f_c:columnas)];
40
41         %se aumenta la primera fila y la última fila
42         %se adicionan los valores de imagen (duplican)
43         tam_im_aumentada=size(im_aumentada);
44         filas_im_aumentada=tam_im_aumentada(1,1);%filas
45         columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
46
47         %imagen aumentada - duplican valores
48         im_aumentada_final=[im_aumentada(1:n_f_c,:);im_aumentada;im_aumentada(filas_im_aumentada-n_f_c:filas_im_aumentada,:)];
49         figure('name','Imagen aumentada - Duplican valores');mesh(im_aumentada_final)%dibujo imagen aumentada
50
51         %se realiza un ciclo para pasar la ventana de convolución
52         tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
53         cont_filas=1;%contador para filas
54         for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
55             cont_columnas=1;%contador para columnas
56             for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
57                 %productos entre imagen y máscara
58                 res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
59                 res_rest=abs([res_mult(2,2) res_mult(2,2) res_mult(2,2);...
60                     res_mult(2,2) res_mult(2,2) res_mult(2,2);...
61                     res_mult(2,2) res_mult(2,2) res_mult(2,2)]-...
62                     im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c));
63                 %maximo tomado
64                 im_convolucionada(cont_filas,cont_columnas)=(max(res_rest{:}));
65                 cont_columnas=cont_columnas+1;
66             end
67             cont_filas=cont_filas+1;
68         end
69     end
70
71     case 2%caso imagen a color
72
73 end
74
75 figure('name','Resultado - Bordes Homogeneidad');
76 imshow(uint8(im_convolucionada))
77
ej_conv2_tercera_solucion_grises_operador_homog... ejemplo_conv3.m
script Ln 75 Col 47 OVR
```



Ejercicio propuesto: utilizar el operador de homogeneidad para una imagen en escala de grises.

- Detector de borde como derivada de primer orden.
 - i. Operadores tipo gradiente (gradiente ortogonal) encuentran bordes verticales y horizontales, además, estos operadores trabajan con el concepto de convolución. Algunas máscaras de convolución comunes son,

	H_f	H_c																		
ROBERTS	<table> <tr><td>0</td><td>0</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	-1	0	1	0	0	0	0	<table> <tr><td>-1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	-1	0	0	0	1	0	0	0	0
0	0	-1																		
0	1	0																		
0	0	0																		
-1	0	0																		
0	1	0																		
0	0	0																		
PREWITT	<table> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	1	0	-1	1	0	-1	<table> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	-1	-1	-1	0	0	0	1	1	1
1	0	-1																		
1	0	-1																		
1	0	-1																		
-1	-1	-1																		
0	0	0																		
1	1	1																		
SOBEL	<table> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>2</td><td>0</td><td>-2</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	2	0	-2	1	0	-1	<table> <tr><td>-1</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>1</td></tr> </table>	-1	-2	-1	0	0	0	1	2	1
1	0	-1																		
2	0	-2																		
1	0	-1																		
-1	-2	-1																		
0	0	0																		
1	2	1																		
FREI-CHEN	<table> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>2^{0.5}</td><td>0</td><td>-2^{0.5}</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	2 ^{0.5}	0	-2 ^{0.5}	1	0	-1	<table> <tr><td>-1</td><td>-2^{0.5}</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2^{0.5}</td><td>1</td></tr> </table>	-1	-2 ^{0.5}	-1	0	0	0	1	2 ^{0.5}	1
1	0	-1																		
2 ^{0.5}	0	-2 ^{0.5}																		
1	0	-1																		
-1	-2 ^{0.5}	-1																		
0	0	0																		
1	2 ^{0.5}	1																		

- ii. El detector de filas se llama H_f y el detector de columnas se llama H_c. Los coeficientes de la máscara suma cero, lo cual hace que al convolucionar la máscara con una imagen constante debería mantener una salida de cero. Si la salida es diferente de cero implica que existe un borde.
- iii. La amplitud resultante se puede obtener calculando la contribución de los detectores fila y columna.

$$H(x,y)=\sqrt{H_f^2(x,y)+H_c^2(x,y)}$$

En algunas ocasiones, se puede trabajar con la magnitud,

$$H(x,y)=|H_f(x,y)|+|H_c(x,y)|$$

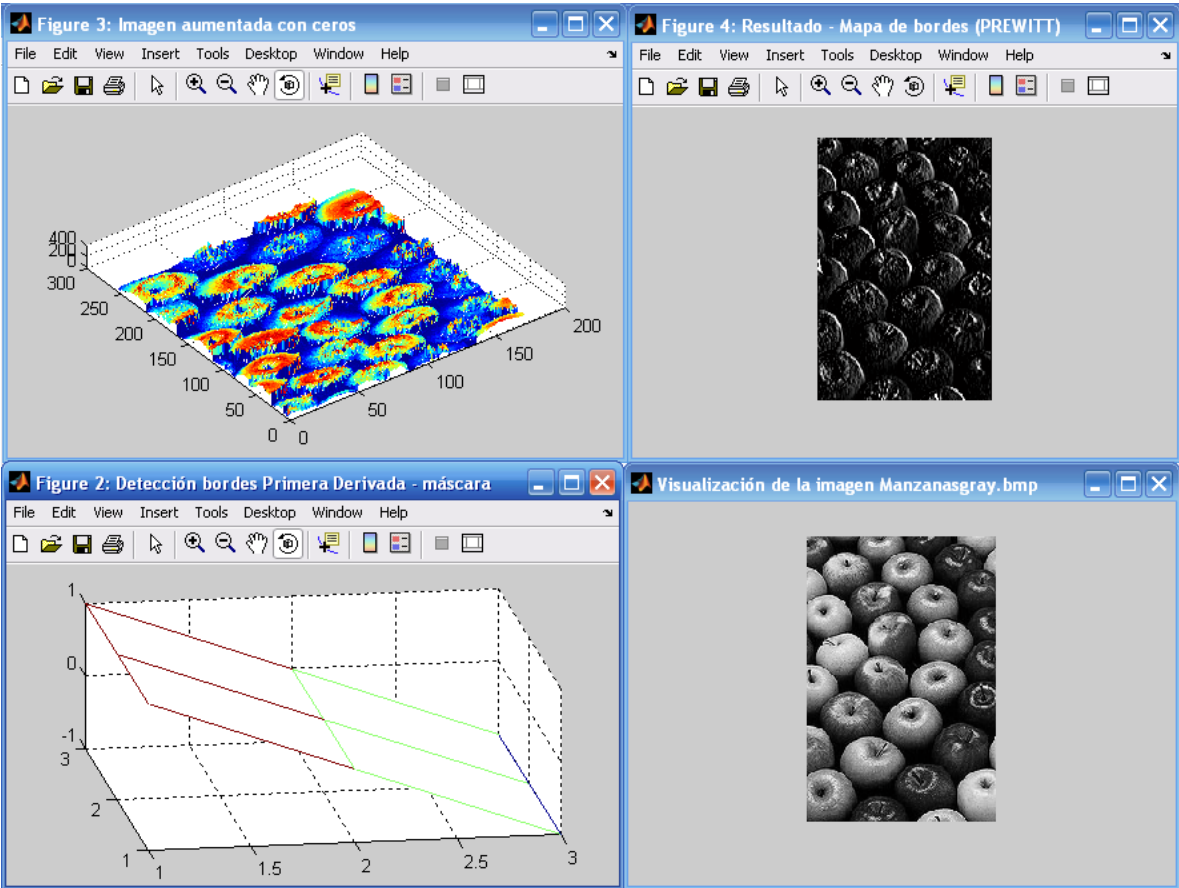
La orientación del borde se maneja de la siguiente forma,

$$\theta = \tan^{-1} \frac{H_c(x,y)}{H_f(x,y)}$$

- iv. El código de programación para el operador fila prewitt es el siguiente,

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_prewitt.m
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack Base
1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14 caso=1;
15 elseif length(tam)==3,
16 caso=2;
17 end
18
19 %construcción de la máscara
20 %Detector de bordes - prewitt Hf
21 mk=[1 0 -1;1 0 -1;1 0 -1];%KERNEL DE CONVOLUCIÓN - prewitt
22 %Dibujo de la máscara detectora de bordes
23 figure('name','Detección bordes Primera Derivada - máscara');mesh(mk)
24
25 %se toma el tamaño de la máscara
26 tam_mask=size(mk);
27 n_f_c=(tam_mask(1)-1)/2;%numero de filas y columnas que
28 %se deben aumentar a la imagen - duplican valores(tercera solución)
29
30 switch caso
31 case 1%caso imagen escala de grises
32 %tamaño de la imagen
33 filas=tam(1,1);
34 columnas=tam(1,2);
35 %se aumenta la primera columna y la última columna
36 %se duplican filas
37 im_aumentada=[xi(1:filas,1:n_f_c) xi(1:filas,columnas-n_f_c:columnas)];
38
39 %se aumenta la primera fila y la última fila
40 %se duplican columnas
41 tam_im_aumentada=size(im_aumentada);
42 filas_im_aumentada=tam_im_aumentada(1,1);%filas
43 columnas_im_aumentada=tam_im_aumentada(1,2);%columnas
44
45 %imagen aumentada - duplican valores
46 im_aumentada_final=[im_aumentada(1:n_f_c,:);im_aumentada;im_aumentada(filas_im_aumentada-n_f_c:filas_im_aumentada,:)];
47 figure('name','Imagen aumentada - duplicada');mesh(im_aumentada_final)%dibujo imagen aumentada
48
49 %se realiza un ciclo para pasar la ventana de convolución
50 tam_im_aumentada_final=size(im_aumentada_final);%tamaño imagen aumentada final
51 cont_filas=1;%contador para filas
52 for filas_im=n_f_c+1:1:tam_im_aumentada_final(1,1)-n_f_c,
53 cont_columnas=1;%contador para columnas
54 for columnas_im=n_f_c+1:1:tam_im_aumentada_final(1,2)-n_f_c,
55 %productos entre imagen y máscara
56 res_mult=im_aumentada_final(filas_im-n_f_c:filas_im+n_f_c,columnas_im-n_f_c:columnas_im+n_f_c).*mk;
57 %se suman los productos
58 im_convolucionada(cont_filas,cont_columnas)=(sum(res_mult(:)));
59 cont_columnas=cont_columnas+1;
60 end
61 cont_filas=cont_filas+1;
62
63 end
64
65 case 2%caso imagen a color
66
67 end
68
69 figure('name','Resultado - Mapa de bordes (PREWITT) Hf');
70 imshow(uint8(im_convolucionada))
71
```

Se tiene el siguiente resultado,

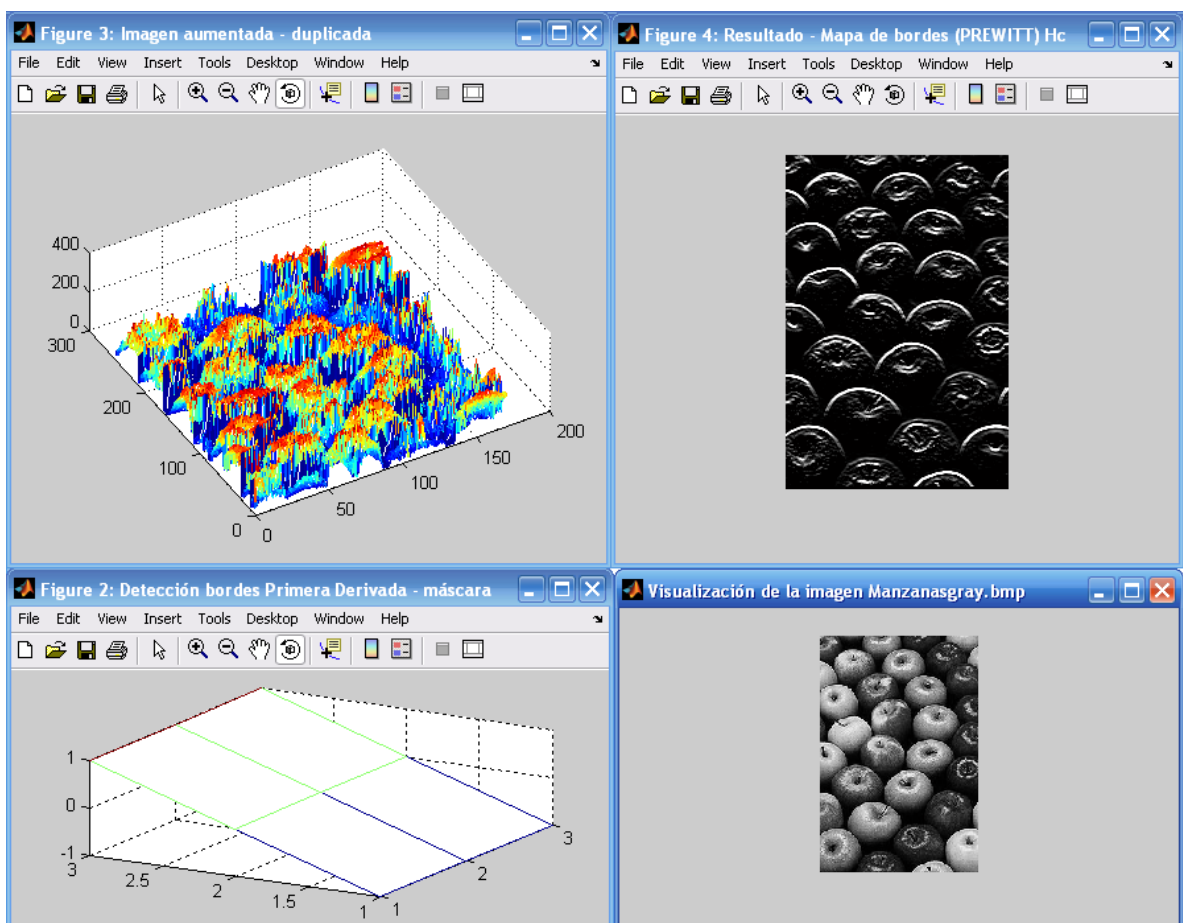


Se observa un resultado con un resalte de bordes verticales.
Se puede ahora observar el resultado de la máscara Prewitt detectora de bordes horizontales.

```
Editor - C:\Program Files\MATLAB71\work\Especializacion_teleco_9_Febrero_2008\ej_conv2_tercera_solucion_prewitt.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base

1 %se lee la imagen sobre la cual se desea
2 %Aplicar el proceso de convolución
3 [FILENAME, PATHNAME] = uigetfile('*.jpg;*.tif;*.bmp', 'LECTURA DE IMAGEN');
4 xi=imread([PATHNAME FILENAME]);
5 f1=figure('numbertitle','off','name',['Visualización de la imagen ' FILENAME],'menubar','none',...
6 'units','normalized','position',[0.2054 0.25 0.6953 0.54875]);
7 %visualización de la imagen
8 imshow(xi),drawnow
9 xi=double(xi);
10
11 %Determinación de imagen: color o escala de grises
12 tam=size(xi);
13 if length(tam)==2,
14     caso=1;
15 elseif length(tam)==3,
16     caso=2;
17 end
18
19 %construcción de la máscara
20 %Detectora de bordes - prewitt Hc
21 mk=[-1 -1 -1;0 0 1;1 1 1];%KERNEL DE CONVOLUCIÓN - prewitt
22 %Dibujo de la máscara detectora de bordes
23 figure('name','Detección bordes Primera Derivada - máscara');,mesh(mk)
24
25 %se toma el tamaño de la máscara
26 tam mask=size(mk);
```

El resultado es una detección de bordes horizontales,



Ejercicio: determine la respuesta de aplicar la máscara de convolución utilizando las otras máscaras (detectores de primera derivada). Utilice para esto una imagen a color y una imagen a escala de grises.

Ejercicio: ¿en qué consiste la umbralización del mapa de bordes y como lo afecta? Mostrar ejemplo.