

PRACTICA 2: IMÁGENES HÍBRIDAS

ÍNDICE

1-INTRODUCCIÓN(2)

2-RESPUESTA A LAS PREGUNTAS (2)

3-COMENTARIOS SOBRE ALGUNOS RESULTADOS Y PROBLEMAS(3)

4-MEJORAS(4)



1-INTRODUCCIÓN

El objetivo de la práctica es mezclar dos imágenes, de manera que la imagen final se encuentre dividida en la combinación de las altas frecuencias de una imagen y las bajas de la otra, con el objetivo de que se identifique la imagen final como la de altas frecuencias cuando es de tamaño considerable y, por el contrario, como la de bajas frecuencias cuando es pequeña.

2-RESPUESTA A LAS PREGUNTAS

1-¿Por qué es conveniente o necesario convertirlas a reales cuando se hace filtrado?

Porque si se utilizasen variables int8, el truncamiento de los decimales y los desbordamientos producirían errores.

2-¿Fíjate en el tamaño del filtro. ¿Qué tamaño es el adecuado?

El tamaño del filtro debe ser $2 \cdot (3 \text{ o } 2)$ la frecuencia de corte del filtro, si se escoge 2 se estará teniendo en cuenta el 95,44% de la distribución, mientras que si se escoge el 3, se tendrá en cuenta el 99,74%

3-¿Por qué sumamos 0,5 a la imagen para visualizarla? ¿Qué ocurre si no se le suma?

Si no se le suma 0,5 a la imagen de altas frecuencias al visualizarla queda muy oscura, esto es porque se le ha restado una imagen a otra, de manera que queda la imagen diferencia, que resulta muy oscura, lo que se hace es añadir brillo a todos los valores.

4-Modifica el programa para que cada una de las dos imágenes que se mezclan tenga su propia frecuencia de corte

Esto se encuentra en el código, se realizan dos filtros gaussianos con diferente frecuencia y se repite el proceso sin mayor complicación.

5- ¿Cuáles crees que son las ventajas e inconvenientes de usarla frente a `fspecial` e `imfilter`? Modifica el programa para que use `imgaussfilt` en lugar de `fspecial` e `imfilter`.

Las ventajas son que reduces el número de líneas de código, lo cual lo simplifica y lo hace más comprensible, que solo se introduce un parámetro para el filtro, lo demás lo ajusta el programa de manera normalmente más óptima que un ser humano y que reducirá tiempo de procesamiento puesto que se trata de una función de matlab única y la implementación será la manera de realizar los cálculos internamente será más eficiente que si se combinan varias. Quizás la única desventaja que encuentro es que visualizar el gráfico del filtro que se aplica no es sencillo.

```
prompt="Elija el filtro según el número de la figura"
nfiltrob=input(prompt);
close all hidden;
sigmabf=3*nfiltrob;
bajas_frecuenciasfb = imgaussfilt(imagen1, sigmabf);
figure(1)
subplot(1,3,1); imshow(bajas_frecuenciasfb); title(['Imagen de bajas frecuencias final']);
true_size
```

Fig1-Utilización de `imgaussfilt`

3-COMENTARIOS SOBRE ALGUNOS RESULTADOS Y PROBLEMAS

No he tenido grandes problemas, algunos resultados son los siguientes:

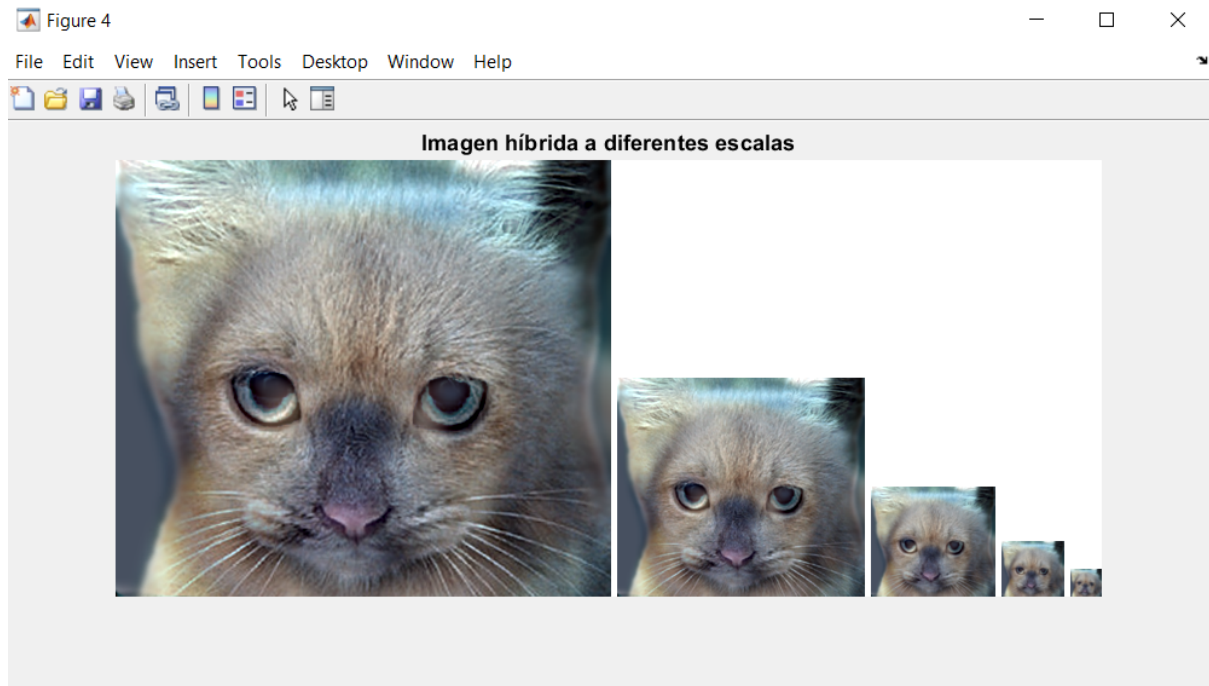


Fig2-Imagen gato perro híbrida transferencia de color del perro al gato.

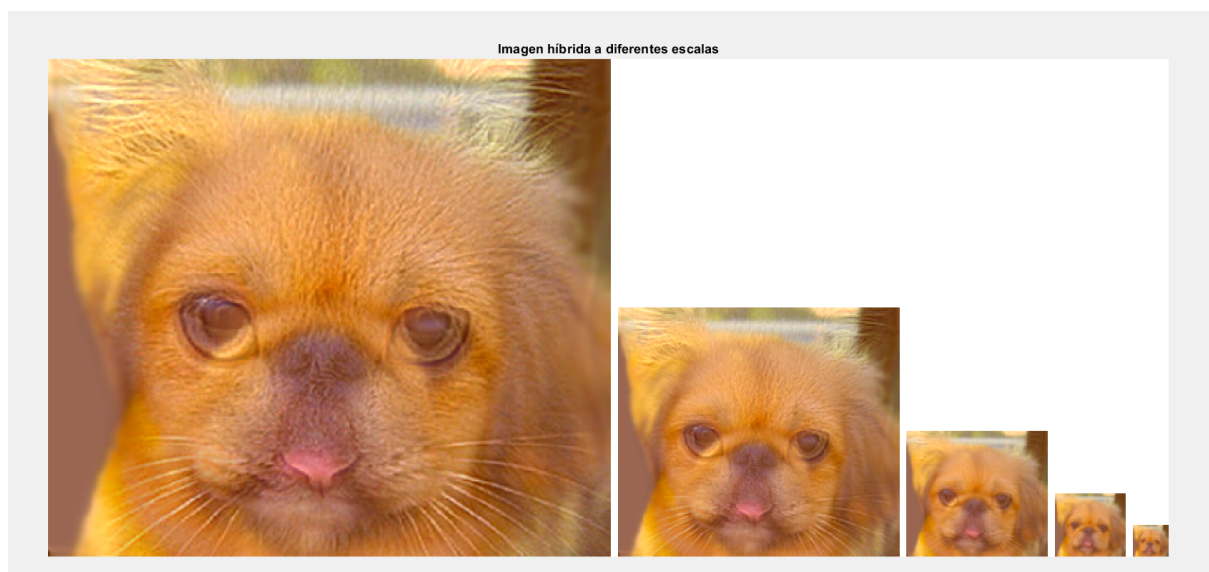


Fig3-Imagen gato perro híbrida transferencia de color del perro al gato

4-MEJORAS

He creado una función que utiliza la función de Reinhart para producir una imagen de salida con los colores de la imagen destino y la forma de la origen, de manera que la mezcla de imágenes tenga más correlación visual y pueda 'confundirse' una imagen con otra. También he hecho un bucle que presenta las imágenes filtradas con filtros con varias frecuencias de corte o sigmas para que se elija para la imagen de frecuencias altas y bajas la frecuencia o sigma deseada.

```

%% Bucle de filtros bajas frecuencias
fcorte= 0;
tamano = 0;
x=0;
fig=1;
for i=3:3:15
    prompt="Pulse tecla para siguiente";
    fcorte=i;
    tamano=(3*fcorte*2)+1;
    filtro=fspecial('gaussian',tamano, fcorte);

    bajas_frecuencias = imfilter(imagen1, filtro);
    figure(fig)
    subplot(1,3,1); imshow(bajas_frecuencias); title(['Imagen de bajas frecuencias ',num2str(fig),' '])
    truesize

    fig=fig+1;
    x=input(prompt);
end
prompt="Elija el filtro según el número de la figura"
nfilterb=input(prompt);
close all hidden;
fcortefb=3*nfilterb;
tamanofb=(3*fcortefb*2)+1;
filtrofb=fspecial('gaussian',tamanofb, fcortefb);
bajas_frecuenciasfb = imfilter(imagen1, filtrofb);
figure(1)
subplot(1,3,1); imshow(bajas_frecuenciasfb); title(['Imagen de bajas frecuencias final']);
truesize
    
```

Fig4-Elección frecuencia imagen de frecuencia baja

```

%% Bucle de filtros altas frecuencias
fcortea= 0;
tamanoa = 0;
y=0;
feg=1;
for i=3:3:15
    prompt="Pulse tecla para siguiente";
    fcortea=i;
    tamanoa=(3*fcortea*2)+1;
    filtroa=fspecial('gaussian',tamanoa, fcortea);

    bajas_frecuenciasalt= imfilter(imagen2, filtroa);
    altas_frecuencias = imagen2-bajas_frecuenciasalt;
    figure(feg)
    subplot(1,3,1); imshow(altas_frecuencias+0.5); title(['Imagen de altas frecuencias ',num2str(feg),' '])
    truesize

    feg=feg+1;
    y=input(prompt);
end
prompt="Elija el filtro según el número de la figura"
nfiltera=input(prompt);
close all hidden;
fcortefa=3*nfiltera;
tamanofa=(3*fcortefa*2)+1;
filtrofa=fspecial('gaussian',tamanofa, fcortefa);
bajas_frecuenciasfa = imfilter(imagen2, filtrofa);
altas_frecuenciasfa = imagen2-bajas_frecuenciasfa;
figure(2)
subplot(1,3,1); imshow(altas_frecuenciasfa+0.5); title(['Imagen de altas frecuencias final']);
truesize
    
```

Fig5-Elección frecuencia imagen de frecuencia alta

```
elija="Que imagen marcará los resultados de transferencia de color";
%La de destino
elige=input(elija);
%elige='1';
if strcmp(elige,'uno')
    disp("opcion1")
    transferida=transferenciacolor(imagen2,imagen1);
    %1 origen 2 destino
    %Si la imagen 1 es la que marca el color será la de destino
    imagen2= transferida;
elseif strcmp(elige,'dos')
    disp("opcion2")
    transferida=transferenciacolor(imagen1,imagen2);
    imagen1=transferida;
    %1 origen 2 destino
    %Si la imagen 2 es la que marca el color será la de destino
    disp(transferida);
else
    disp("No habrá transferencia de color")
```

Fig6-Elección transferencia color