

Prácticas de Visión Artificial

Práctica 2: Filtrado y detección de contornos, histogramas

Los temas principales de esta práctica son:

- 1) Suavizado de una imagen
- 2) Detección de contornos en una imagen

Para completar la práctica, es necesario conocer los siguientes conceptos: linear filters, histogram, convolution, edges.

Los siguientes apartados del libro “Computer Vision: Algorithms and Applications” de Richard Szeliski proporcionan información adicional a la vista en las clases de teoría:

Chapter 3: Image Processing: Point Operators & Linear Filtering.

Chapter 4: Feature detection and matching: Edges.

2.1 Procesamiento de imágenes con diferentes escalas y filtros de suavización

Ejecutar la siguiente función:

```
function [] = ej21() % file is the name of the image
%ej21 This function illustrates the effect of resizing and smoothing
with different 1D and 2D masks

close all;
file='face.png';

im=imread(file);

imGray=rgb2gray(im);
resizedIm=imresize(imGray, 2);

h_hor=[1 1 1 1 1]; % 1D mask defined
h_hor=h_hor/sum(h_hor); % normalization

% define a vertical mask and apply it on the gray image.

smoothedImGray1D_hor=imfilter(double(imGray),h_hor);
% Repeat the smoothing several times in order to remove the line in
the middle.

%Look for alternative of the mean filter to remove the line in the
middle.
%smoothedImGray1D_...=

% define the 2D mask, shall we normalize it?

% smoothedImGray2D = ... %convolving with a 2D mask

figure, subplot(3,4,2),imshow(im, []), title('Original image'),...
subplot(3,4,5),imshow(imGray), title('Gray image'),...
```

```
subplot(3,4,6),imshow(resizedIm),title('Resized image'),...
subplot(3,4,7),imshow(uint8(smoothedImGray1D_hor)),title('Smoothed
image'),...
subplot(3,4,9),imhist(imGray), title('Histogram of Original im.'), ...
subplot(3,4,10),imhist(resizedIm),title('Histogram of resized
im.'),...
subplot(3,4,11),imhist(uint8(smoothedImGray1D_hor)),title('Histogram
of smoothed image');
```

- a) Observar cómo desaparecen los detalles de la imagen cuando se re-escala (aumentando o reduciendo) el tamaño de la imagen. ¿Cambia el histograma de las dos imágenes (la original y la re-escalada)? ¿Qué pasa con la reducción del tamaño de la imagen original? ¿Se pierden los detalles de la imagen re-escalada? ¿El histograma cambia significativamente?
Si que cambia el histograma. Que la imagen se oscurece un poco. Algunos detalles se pierden. Si cambia el numero de pixels considerablemente
- b) Repetirla suavización varias veces para eliminar la línea del medio. ¿Cuántas iteraciones hace falta para eliminar la línea?
- c) Suavizar con una máscara vertical. ¿Cuál es la diferencia en la imagen filtrada aplicando máscaras horizontales y verticales? Nota: Para visualizar los diferentes resultados podéis cambiar los parámetros del *subplot*.
- d) Seleccionar una de las imágenes de “images.zip” y repetir a)-c). Aplicar diferentes tamaños de filtros y comentar los resultados. Nota: Para ver mejor el efecto, escoger alguna imagen con textura.
- e) Definir una máscara 2D. Comentar cómo el tamaño de la máscara afecta el resultado final del filtraje.
- f) ¿Se puede aplicar el filtro sobre la imagen en color? ¿Se puede visualizar el histograma de la imagen suavizada en color? ¿Qué tipo debe ser la imagen antes de aplicar la convolución y por qué?
- g) ¿Cuál es la diferencia usando las siguientes máscaras:
 - a. $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
 - b. $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
- h) ¿Qué pasa si no normalizamos la máscara? Aplica varias veces la convolución sobre la imagen con el fin de observar el efecto de suavizado mejor.
- i) **(Opcional)** Aplicar la suavización a diferentes imágenes de “images.zip”.

Nota 1: La mayoría de los filtros necesitan imágenes de tipo *double*. Sin embargo, para mostrar y calcular el histograma hay que convertir las imágenes *double* en *uint8*.

Nota 2: Comprueba cómo puedes utilizar *imshow(imagen,[])* para visualizar maximizando el contraste de la imagen.

2.2 Procesamiento de imágenes con filtros ponderados y filtros no lineales

- a) Generar el kernel (núcleo) de la Gaussiana con el comando de Matlab y aplicar la convolución sobre alguna de las imágenes seleccionadas para el Ejercicio 2.1. Repetir 100 veces la suavización para ver mejor el efecto. Comparar con la suavización con filtro de la media. Utilizar diferentes valores de sigma y comentar su efecto. ¿Qué valor de sigma consideráis más adecuado para suavizar los detalles de esa imagen en concreto, y quedarse con los objetos y estructuras principales?
- b) Proponer un filtro alternativo al filtro de la media que permite eliminar la línea del medio de la imagen 'face.png' aplicando el filtro una única vez. Restar la imagen original de la suavizada con el fin de ilustrar la diferencia entre ellas (ten en cuenta que algún píxel puede quedar en negativo).

2.3 Determinar los contornos óptimos

- a) Leer la imagen 'logo.png' y encontrar sus contornos (¿Cuál es el comando en Matlab?)
- b) Aplicar los diferentes operadores de contornos vistos en clase de teoría y encontrar los parámetros óptimos para cada uno de ellos. Utiliza *subplot* y *title* para visualizar los diferentes resultados.
¿Cuál es el mejor detector de bordes? ¿Cuáles son los parámetros óptimos para esta imagen? ¿Hace falta normalizar la máscara como en el filtraje para la suavización?
- (Opcional)** Sobreponer los contornos sobre la imagen como se muestra en la figura 1 (derecha).
- c) Repite el experimento con otras imágenes de las incluidas en "images.zip". Comenta si los parámetros se deberían cambiar para cada imagen.
 - a. ¿Se mejoran los contornos si la imagen se suaviza antes?
 - b. ¿Cuáles son las limitaciones que ves en la extracción de los contornos en las diferentes imágenes?
- d) **(Opcional)** ¿Cuántas maneras diferentes hay para aplicar el detector de Sobel? Aplicar el detector de Sobel de las diferentes maneras, y visualizar y comparar los resultados obtenidos.



Figura 1. Imagen original (arriba, izquierda), bordes extraídos (fondo negro) y la superposición de los bordes y la imagen original (abajo, derecha).

2.4 Aplicación de la suavización para construir imágenes híbridas

a) Dadas las imágenes 'einstein.jpg' y 'monroe.jpg', construir la imagen híbrida y visualizarla a diferentes escalas para obtener el efecto visual de las imágenes híbridas (Figura 2).

Nota: Aplicando una suavización sobre una imagen I , estamos aplicando el filtro "low-pass". Llamemos a la imagen resultante $L(I)$. Si restamos de la imagen original la imagen filtrada obtenemos las frecuencias altas $H(I)$ de la imagen, es decir aplicamos el filtro "high-pass":

$$H(I) = I - L(I)$$

Una imagen híbrida se construye sumando el filtro que deja pasar las frecuencias bajas ("low-pass") y el que deja pasar las frecuencias altas ("high-pass"):

$$\text{Hybrid}(I_1, I_2) = L(I_1) + H(I_2)$$

Para resaltar el efecto, en la imagen $L(I_2)$ utilizar menor sigma y en $H(I_2)$ utilizar mayor sigma. Nota: Aplica en `fspecial('gaussian', hsize, sigma)` con la fórmula $hsize = 4 * sigma + 1$. ¿A qué corresponde *hsize*?

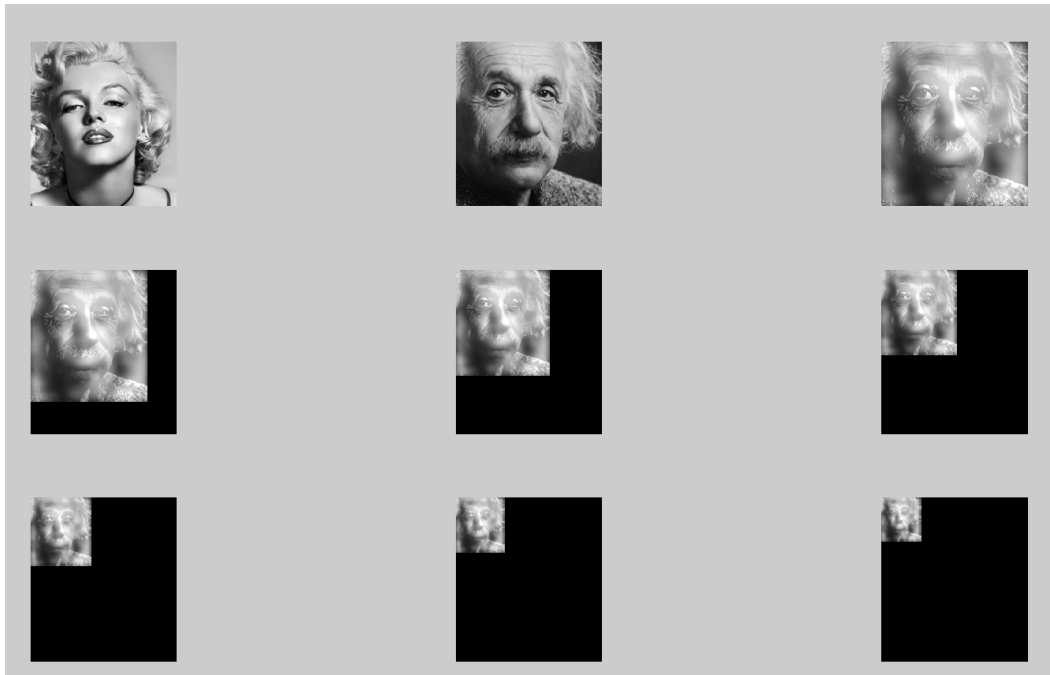


Figura 2. Imágenes híbridas.

b) Visualizar la imagen híbrida sobre la misma figura/región para poder evaluar a partir de qué tamaño se pasa a percibir a Marilyn Monroe.

(Opcional) Construir imágenes híbridas a partir de otras imágenes. Aplicar el efecto a imágenes en color.

2.5 Anonimización de vídeos

Uno de los problemas importantes en Visión Artificial es anonimizar la apariencia de las personas en los vídeos e imágenes.

Dado el vídeo 'BigBang.mp4', filtrar las imágenes (enteras, no sólo la cara) del vídeo con la máscara adecuada (del Ejercicio 2.1) para hacer la cara no identificable. Visualizar y retornar el vídeo en color con las caras no identificables.

Help: Se puede usar *VideoReader* para abrir un vídeo y *read* para leer las imágenes una por una del objeto que retorna *VideoReader*.

Entrega de la práctica

Tiempo máximo de entrega: martes 25 de Octubre **a las 23:00h.**

Material para entregar: archivo "nombres_apellidos_P2.zip" que contenga:

- Archivos .m de cada ejercicio con las funciones creadas. **El código debe estar comentado en inglés.**
- Fichero .pdf que incluya las respuestas planteadas en los distintos ejercicios.