



Universidad
de Alcalá



Segmentación de Imágenes

Sistemas de Visión Artificial

Universidad de Alcalá

El **objetivo** principal de esta práctica es familiarizarse con las técnicas de segmentación de imágenes:

- desde niveles más sencillos que requieren intervención por parte del usuario (segmentación supervisada) para fijar umbrales, establecer los parámetros de los modelos etc., hasta niveles más complejos que trabajan de forma automática o semi-automática (segmentación no supervisada).
- a partir de técnicas basadas en el histograma, técnicas de agrupamiento (clustering) en regiones y técnicas de bordes.
- basadas en métodos lineales y no lineales de clasificación de cada píxel de la imagen en la región que le corresponda.

A lo largo de la práctica se solicitará que el estudiante:

- aplique, de forma práctica, en el laboratorio, algunas de las técnicas de segmentación estudiadas, estudiando la problemática y limitaciones de las mismas,
- compare unas técnicas de segmentación con otras, analizando el porcentaje de acierto de las mismas,
- distinga la segmentación supervisada frente a la no supervisada,
- diferencie la segmentación basada en técnicas lineales frente a no lineales,
- comprenda los atributos básicos que se suelen analizar para la segmentación de imágenes (luminancia, color, bordes, texturas...)

Al igual que en la práctica anterior, acabadas las actividades propuestas, se entregará una **memoria** sobre las mismas que incluya, para cada ejercicio:

- Imagen original/Imagen procesada/dato solicitado.
- Imágenes intermedias (si existen y son de interés).
- Código.
- Explicación del código y conclusiones sobre el resultado obtenido.

Se valorará positivamente la ampliación de la memoria con las explicaciones que considere oportunas, pruebas adicionales (modificando parámetros, utilizando otras imágenes que resalten algún efecto del procesamiento, etc.), descripción de problemas surgidos en la ejecución de la práctica y solución proporcionada, etc.

Realice los siguientes **ejercicios** en un único fichero .m, con las distintas partes como secciones independientes (separadas por una línea que comience con %%), visualizando los resultados conjuntamente con subplot cuando sea conveniente, e insertando instrucciones para cerrar ventanas (*close all*), eliminar las variables del *workspace* (*clear all*) y pausar (*pause*), donde considere adecuado.

Ejercicios propuestos:

1.

- a) Genere una imagen con objetos con varios niveles de gris diferente,
- b) añada ruido gaussiano,

- c) fíltrelo con un filtro de wiener,
- d) dibuje el histograma,
- e) a la vista del histograma, utilice los umbrales adecuados para segmentar cada uno de los objetos de la imagen por separado,
- f) muestre cada objeto en una imagen diferente.

2.

a) Lea y muestre por pantalla la imagen 'rice.tif', visualice su histograma y, a partir de la información del mismo, fije un umbral global adecuado y segmente la imagen (para distinguir objetos y fondo) basándose en el citado umbral. ¿Es clara la distinción entre dónde acaba un objeto y dónde empieza el siguiente, es decir, es fácil la elección del umbral?

b) Modifique el programa anterior de manera que, en vez de fijar usted el umbral en las líneas de código del propio programa, lo pida al usuario que observa el histograma, visualizando en el prompt de Matlab:

```
umbral = input('Introduce valor del umbral global a la vista de la figura del histograma ')
```

Segmente la imagen a partir del umbral global tecleado por el usuario.

c) Pruebe el programa del apartado anterior con otras imágenes de Matlab (en vez de la 'rice.tif'). Puede utilizar el siguiente código, que convierte a niveles de gris imágenes de diferentes tipos (*truecolor*, *indexed*, *grayscale*):

```
imagen.nombre='rice.tif'; %crea estructura con campo nombre inicializado con 'rice.tif', p. ej.
info_imagen=imfinfo(imagen.nombre); %estructura con campos con inform. de la imagen
```

```
%La imagen se podría pedir al usuario en el prompt de Matlab, haciendo:
% imagen.nombre=input('introduce nombre de fichero entre comillas, p. ej.: "rice.tif"\n');
%Recuerde que las cadenas van entre comillas simples y si en la propia cadena se quiere poner
%una comilla simple, ésta se pondrá como doble. Por eso se pone "rice.tif" (entre comillas
dobles) para que se visualice luego en el prompt 'rice.tif'.
```

```
%Codigo para convertir la imagen (indexed, truecolor, grayscale) a niveles de gris (grayscale)
if strcmp(info_imagen.ColorType,'indexed')
    [X,map]=imread(imagen.nombre);
    I=ind2gray(X,map)
elseif strcmp(info_imagen.ColorType,'truecolor')
    rgb=imread(imagen.nombre);
    I=rgb2gray(rgb);
else
    I=imread(imagen.nombre);
end
```

3.

Varíe el apartado **c)** del ejemplo anterior de modo que umbralice con la función *im2bw*, utilizando un umbral global calculado automáticamente con función *graythresh* de

Matlab (sin requerir la observación del usuario del histograma y elección de umbrales a partir del mismo):

- La función *im2bw* umbraliza cualquier imagen de entrada, sea del tipo que sea (incluso imágenes en color), convirtiéndola a imagen de intensidad y “binarizando” esta última con un umbral en un rango entre [0 1].
- La función *graythresh* utiliza el método Otsu para escoger el umbral global de segmentación minimizando la varianza dentro de los elementos de las clases y maximizándola entre clases.

```
level = graythresh(I);
BW = im2bw(I,level);
figure, imshow(BW)
```

Pruebe con la imagen 'rice.tif' y otras imágenes de Matlab ¿funciona siempre bien *graythresh* (segmentando adecuadamente la imagen)? En caso negativo ¿en qué tipo de imágenes suele funcionar bien?

4. Compare los dos apartados anteriores: segmentación supervisada frente a no supervisada, comentando la problemática de cada una.

5. Implemente un programa que realice la segmentación supervisada por modelos gaussianos (no lineal) para imágenes de cualquier tamaño y tipo (indexadas, color y en niveles de gris) que serán siempre convertidas a niveles de gris en primer lugar.

6. Implemente un programa que realice la segmentación de una imagen con n clases de forma no supervisada (con $n > 2$). Muestre cada clase segmentada en la imagen con un color diferente (para distinguir unas clases de otras en la imagen de resultados de la segmentación).

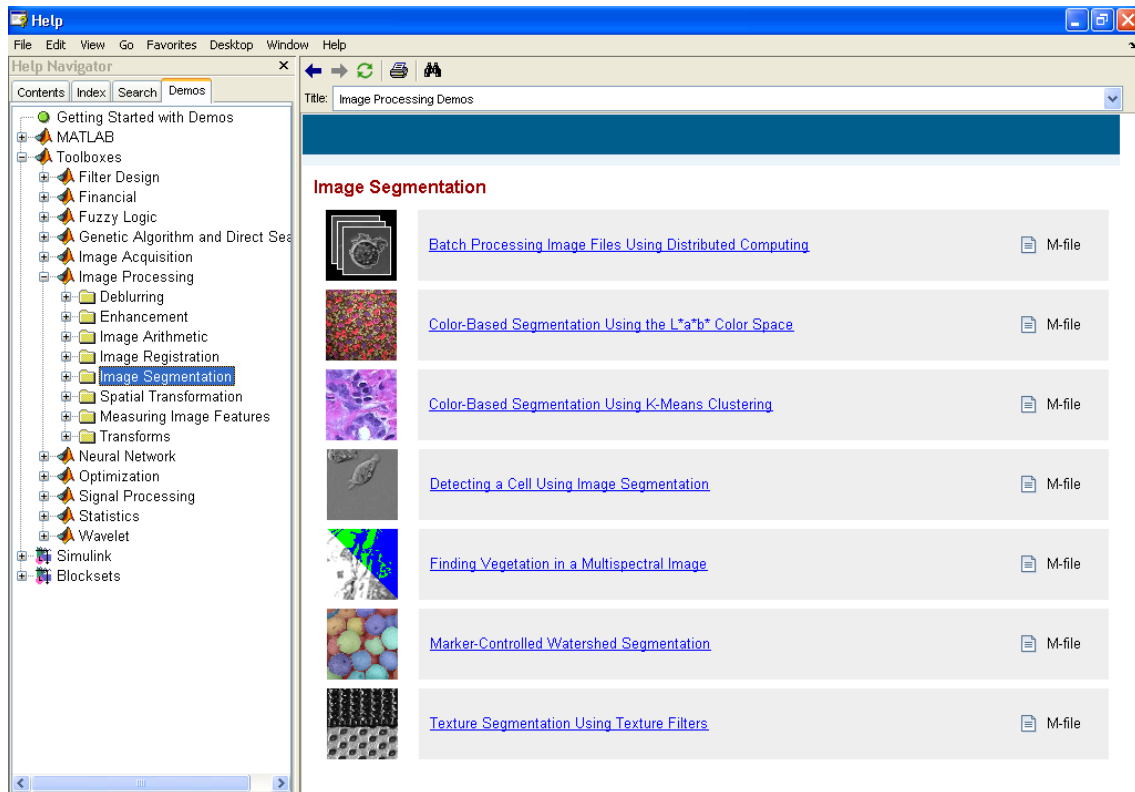
7. Realice un programa que segmente los objetos de la imagen rice.tif mediante bordes (utilizando el filtro de Canny) y análisis local de bordes (si es necesario). Pruebe su programa con otras imágenes.

8. Realice el análisis global con la transformada de Hough.

9. Pruebe los algoritmos de segmentación vistos sobre varias imágenes y extraiga conclusiones sobre los mismos. ¿En qué casos funcionan? ¿Cómo se podrían mejorar?

10. Transformada Watershed

Teclee *demo* en el prompt de *Matlab* y después, métase en la *Toolbox de Image processing* (apartado de *Image Segmentation*):



Ejecute algunas de las demos de Matlab referentes a segmentación de imágenes. Pruebe los ejemplos vistos de la transformada Watershed (incluida la demo de Matlab *Marker-Controlled Watershed Segmentation*) sobre otras imágenes y extraiga conclusiones. Si lo precisa, puede intentar variar el código para obtener mejores resultados.