

# Práctica 5

## Segmentación de Imagen (II)

### Curso 2019 – 2020

---

El objetivo de esta práctica es familiarizar al alumno con las técnicas de extracción de características a fin de abordar un problema de segmentación utilizando aprendizaje no supervisado, en concreto el algoritmo  $k$ -medias. Para ello se trabajará con la imagen en color ‘cormoran.jpg’, que acompaña al material de esta práctica.

Utilice la ayuda de MATLAB (`help+` comando) para conocer el funcionamiento de los comandos utilizados en este guión. Tenga en cuenta que a lo largo de la práctica puede utilizar las instrucciones `clear all` y `close all` para evitar posibles interferencias con otras variables o ventanas. La instrucción “`imtool close all`” permite cerrar todas las ventanas generadas con la función `imtool`.

#### ***I. Análisis visual de la imagen***

Lea en MATLAB la imagen ‘cormoran.jpg’ proporcionada como material adjunto en la práctica y almacénela en la variable `I`. Examine visualmente la imagen (representada en la Figura 1) y responda a las siguientes preguntas:

- ¿qué objetos considera claramente discriminativos en la imagen?
- a simple vista, ¿qué características considera discriminativas?

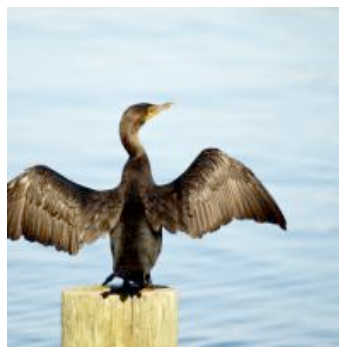


Figura 1. Imagen a segmentar en esta práctica.

Convierta la imagen a escala de grises (variable `I_gris`) y analice visualmente su histograma. ¿Cree que podría realizar la segmentación mediante umbralización múltiple? Justifique su respuesta.

## II. Características RGB y algoritmo *k*-medias

Inicialmente abordaremos el problema de segmentación considerando como características los niveles de intensidad de las componentes R, G y B de la imagen original  $I$ . Para ello:

- Extraiga cada componente de color de la imagen  $I$ .
- Convierta cada componente en un vector columna utilizando el comando `reshape`. Así, por ejemplo, para almacenar como vector columna los niveles de intensidad de los píxeles de la componente R de  $I$ , utilizaremos la siguiente secuencia de instrucciones

```
I_R = I(:, :, 1);
[nrows, ncols] = size(I_R);
I_R_res = reshape(I_R, nrows*ncols, 1);
```
- Represente el *scatter plot* de los datos utilizando la función `plot3` (haga uso del comando `help` para conocer su sintaxis). ¿Observa grupos de puntos claramente diferenciados?

Sobre el espacio de tres dimensiones representado en la figura anterior aplicaremos el algoritmo de agrupamiento *k*-medias con  $k=3$ . Utilizaremos para ello la función `kmeans` del siguiente modo

```
ngrupos = 3;
rgb_res = double([I_R_res I_G_res I_B_res]);
[cluster_idx cluster_center] =
kmeans(rgb_res, ngrupos, 'distance', 'sqEuclidean', 'Replicates', 10);
```

Las **entradas** a la función `kmeans` son: (1) conjunto de ejemplos/observaciones (matriz `rgb_res`, donde cada ejemplo es una fila y cada columna representa una característica), (2) medida de similitud (cuadrado de la distancia Euclídea en nuestro caso, indicado a través del parámetro `'distance'`), y (3) número de inicializaciones (utilice el valor 10 en esta práctica, indicado con el parámetro `'Replicates'`). El algoritmo `kmeans` devuelve la posición de los centroides (variable `cluster_center`) y una etiqueta identificativa del *cluster* al que pertenece cada punto de entrada (variable `cluster_idx`). Puesto que el algoritmo se realiza 10 veces con distintas inicializaciones, la salida proporcionada corresponde a la realización para la que se obtiene la mínima suma de distancias intra-cluster.

**Nota importante:** la matriz de ejemplos/observaciones debe ser de tipo `double`; cada fila de la matriz es una observación (píxel de la imagen, en este caso) y cada columna es una característica.

Las **salidas** de la función son: (1) identificador del *cluster* al que pertenece cada punto (variable `cluster_idx`), y (2) centroide de cada *cluster* (variable `cluster_center`).

Para observar el resultado del algoritmo *k*-medias, puede representar sobre el *scatter plot* anterior los centroides resultantes del siguiente modo:

```
plot3(cluster_center(:,1), cluster_center(:,2), cluster_center(:,3), 'sr');
```

El resultado de la segmentación se puede observar en el espacio imagen generando una imagen en falso color a partir del identificador de *cluster* asociado a cada observación (píxel). Para ello, debe transformar el vector con la identificación del cluster al que pertenece cada píxel en una matriz de las mismas dimensiones que la imagen original. Puede hacer uso del siguiente código.

```
pixel_labels_rgb = reshape(cluster_idx,nrows,ncols);  
I_segm = label2rgb(pixel_labels_rgb);  
figure, imshow(I_segm)
```

¿Considera que el resultado de la segmentación es el deseado?, ¿es posible afirmar que se produce sobresegmentación? Justifique razonadamente sus respuestas.

### III. Características cromáticas ab

Como sabe, en el espacio RGB la información cromática está distribuida en las tres componentes (R, G y B). En esta sección realizaremos una transformación del espacio de representación para separar las componentes cromáticas y acromáticas de la imagen, de modo que aplicaremos el algoritmo *k*-medias únicamente sobre el espacio de componentes cromáticas. La transformación considerada es la transformación *Lab*, transformación que implementa la función `rgb2lab` proporcionada como material adjunto a la práctica.

Realice la transformación de la imagen original (espacio RGB) al espacio *Lab* y extraiga únicamente las componentes cromáticas (componentes *ab*). Represente el *scatter plot* correspondiente e intente identificar visualmente a qué color corresponden las nubes de puntos.

```
[lab_imL, l_L, a_L, b_L] = rgb2lab(I);  
a_res = reshape(a_L,nrows*ncols,1);  
b_res = reshape(b_L,nrows*ncols,1);  
  
figure, plot(a_res, b_res, '.')  
xlabel('a'), ylabel('b')
```

Aplice ahora el algoritmo *k*-medias sobre el nuevo espacio de características y represente la capa de etiquetas de la imagen segmentada. Comente las diferencias con la segmentación obtenida en el Apartado II.

Represente la posición de los centroides en el *scatter plot* correspondiente y responda a las siguientes preguntas:

- ¿cuál es la desviación típica (comando `std` de MATLAB) asociada a la componente *a* obtenida a partir de *I*?
- ¿cuál es la desviación típica asociada a la componente *b* obtenida a partir de *I*?
- ¿cree que alguna de las dos componentes tiene más influencia al determinar la posición de los centroides con el algoritmo *k*-medias?, ¿por qué?

Para que no haya una componente que domine en el cálculo de distancias del algoritmo  $k$ -medias únicamente porque su rango dinámico es mayor, se propone normalizar cada componente del espacio `ab` para que tenga media nula y desviación típica uno. La matriz de ejemplos con características normalizadas se asignará a la variable `ab_norm` (cada fila es un ejemplo y cada columna es una característica). Explique cómo se realiza la normalización si la secuencia de instrucciones es la siguiente:

```
ab_res = [a_res b_res];  
ndim = size(ab_res,2);  
ab_norm = ab_res;  
for ind_dim=1:ndim  
    datos = ab_res(:,ind_dim);  
    datos_norm = (datos-mean(datos))/std(datos);  
    ab_norm(:,ind_dim)=datos_norm;  
end
```

Aplique el algoritmo  $k$ -medias sobre el espacio de características normalizado y visualice en el espacio imagen el resultado de la segmentación. ¿Observa alguna diferencia respecto a la segmentación obtenida sin normalizar las características?, ¿qué conclusión puede extraer sobre la normalización y la aplicación del algoritmo  $k$ -medias considerando distancia Euclídea?

#### ***IV. Características de textura***

MATLAB proporciona una serie de filtros para extraer características relacionadas con la textura. La salida de cada filtro es una imagen del mismo tamaño que la imagen original donde cada píxel contiene la característica de textura extraída en un entorno local del mismo. Consulte la ayuda de las funciones `stdfilt`, `entropyfilt` y `rangefilt` de MATLAB e indique qué descriptor estadístico se obtiene con cada una de las tres funciones.

Considere un entorno de 7x7 píxeles y represente en el espacio imagen el resultado obtenido al aplicar cada uno de los descriptores de textura anteriores sobre la imagen de grises original (almacenada en la variable `I_gris`). Las instrucciones para representar el descriptor asociado al filtro `stdfilt` son las siguientes:

```
S = stdfilt(I_gris,ones(7,7));  
imshow(S,[]), title('S')
```

Interprete cada una de las imágenes obtenidas conforme al descriptor utilizado. Elija dos de estos descriptores como características y realice la segmentación considerando el espacio de características correspondiente.

A la vista de los resultados, y si tuviera que elegir únicamente dos características, ¿qué características elegiría? ¿las obtenidas en el Apartado III o las extraídas en este apartado?

### ***V. Utilización de características de distinta naturaleza***

Para mejorar el resultado de la segmentación obtenida en los apartados anteriores, justifique razonadamente la elección de tres características si se desea utilizar descriptores de distinta naturaleza (color y textura).

Realice la segmentación de I considerando las tres características seleccionadas (recuerde utilizar características normalizadas). A la vista de los resultados, indique qué tipo de características (color, textura), aporta más información para realizar la segmentación de esta imagen.

Como observará, el resultado adolece de sobresegmentación, aunque ésta corresponde a regiones de tamaño muy reducido. Para resolver este problema, una alternativa es recurrir al análisis de cada región y realizar la fusión de regiones contiguas. No obstante, en esta práctica intentaremos resolver el problema de sobresegmentación preprocesando las componentes de color, que son las que más información parece que aportan para segmentar la imagen de esta práctica. Para ello, se propone aplicar un filtro de suavizado de tamaño 7x7 píxeles sobre cada componente cromática del espacio Lab.

Aplique un filtro de suavizado sobre cada componente de color y construya el nuevo espacio de características normalizado. Aplique el algoritmo de  $k$ -medias y comente el resultado.

Como comprobará, parte del pico del cormorán se asocia a la categoría “madera”, por tener un color similar. ¿Cree que la aplicación de una técnica de aprendizaje supervisado podría mejorar el resultado obtenido en esta práctica? Justifique razonadamente su respuesta.

Explique detalladamente (apoyándose en un diagrama de bloques) qué procedimiento seguiría si hubiera abordado la segmentación de esta imagen con un esquema de aprendizaje supervisado. Explique al menos una técnica de las que podría aplicar.