

# Prácticas de Visión Artificial

---

## Práctica 3: Segmentación, descriptores de imágenes & SIFT.

---

Los temas principales de esta práctica son:

- 1) Agrupación.
- 2) Segmentación de imágenes.
- 3) Extracción de descriptores/características de una imagen usando SIFT.
- 4) Reconocimiento por correspondencia.

Para completar la práctica es necesario conocer los siguientes conceptos de teoría: agrupación por K-Means, HOG y el método SIFT. 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 4: Computer Vision: Algorithms and Applications.

Chapter 5: Segmentation.

### 3.1 Métodos de “background subtraction”

Dado el vídeo ‘Barcelona.mp4’, que contiene imágenes adquiridas con cámaras estáticas, eliminar todos los artefactos relacionados con el movimiento extrayendo las imágenes el fondo.

Nota: Una de la aplicaciones de esta opción se puede relacionar con el “botón eliminar turistas” inventado para las cámaras de fotos. Por ejemplo, Adobe usa el “Monument mode” que suprime automáticamente la gente que pasa delante de la cámara. Para realizar el algoritmo se necesita:

- a) Encontrar donde se acaba una escena y comienza otra (estos frames se denominan *shots* del vídeo). ¿Qué medida de las imágenes se puede utilizar para distinguir las escenas?
- b) Aplicar un algoritmo de background subtraction (consultar el material de teoría).

Visualizar, para cada segmento delimitado por dos shots del vídeo, la imagen estática extraída y las imágenes a partir de las cuales se ha obtenido (Figura 1).

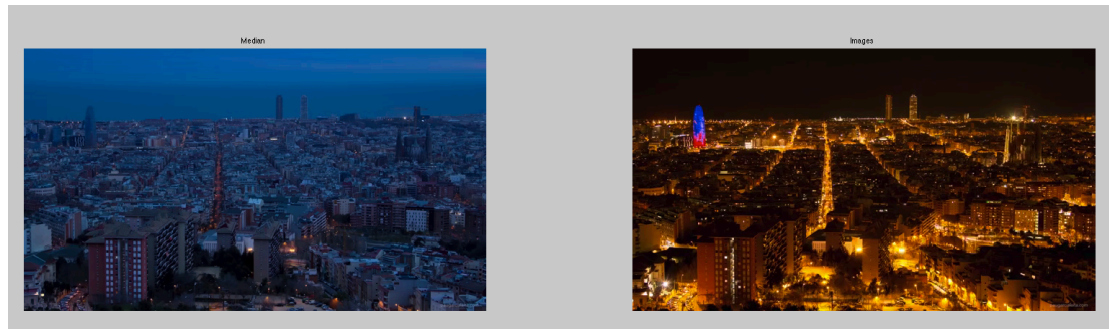


Figura 1.

Comenta en detalle la implementación. ¿Qué sucede si los shots no están correctamente extraídos? ¿Qué sucede si encuentras demasiados shots en el vídeo? Comenta qué representan las imágenes estáticas obtenidas. ¿En qué situaciones el algoritmo funciona y en cuáles no? ¿Qué sucede si restas de la imagen original la imagen del fondo? Visualízalo.

Ves otras aplicaciones que se pueden sacar de este algoritmo?!

**[OPCIONAL]** Aplicar el algoritmo a alguno de los siguientes vídeos, o a algún vídeo que hayas encontrado donde tiene sentido aplicarlo.

- [Hokkey](#)
- [Red light running](#)
- [Waynedale Memorial Parade](#)
- [Bilbao Timelapse](#)
- [Barcelona airport](#)
- [Boqueria \(25:00\), Colon \(27:00\), Sagrada familia \(41:10\)](#)
- [Paseo de Gracia \(2:59\), Museo de Historia \(3:12\)](#)
- [Removing tourists in Barcelona pictures \(4:11\)](#)

### 3.2 Métodos de agrupación de datos numéricos

Crea un fichero ej32.m que implemente los siguientes pasos:

- a) La función *gaussRandom* (*mu*, *sigma*, *numSamples*), proporcionada con el enunciado, permite generar nubes de puntos con una distribución gaussiana con matriz de covarianza diagonal, utilizando como centro las coordenadas de *mu*. Genera tres nubes de 100 puntos con centros [1 2], [2 2] y [2 1]. En los tres casos utilizar una desviación estándar de 0.1 en todos los ejes. Visualiza los puntos generados (help: *plot*).
- b) Utilizar el método *kmeans* para agrupar los datos anteriores. Visualizar en un mismo plot (help: *subplot*) una primera fila con los datos originales, y el resultado (incluyendo los centros) de las agrupaciones con 2, 3 y 4 centros respectivamente en la segunda fila del *subplot*, utilizando diferentes colores (Figura 2).

- c) Comenta los resultados que has encontrado, valorando el número de agrupaciones que has encontrado en cada caso y la similitud con los datos iniciales.

Nota: En estos ejercicios, las características por las que se aplica el *kmeans* (y que definen el espacio de características) son las posición de los puntos, es decir, sus coordenadas.

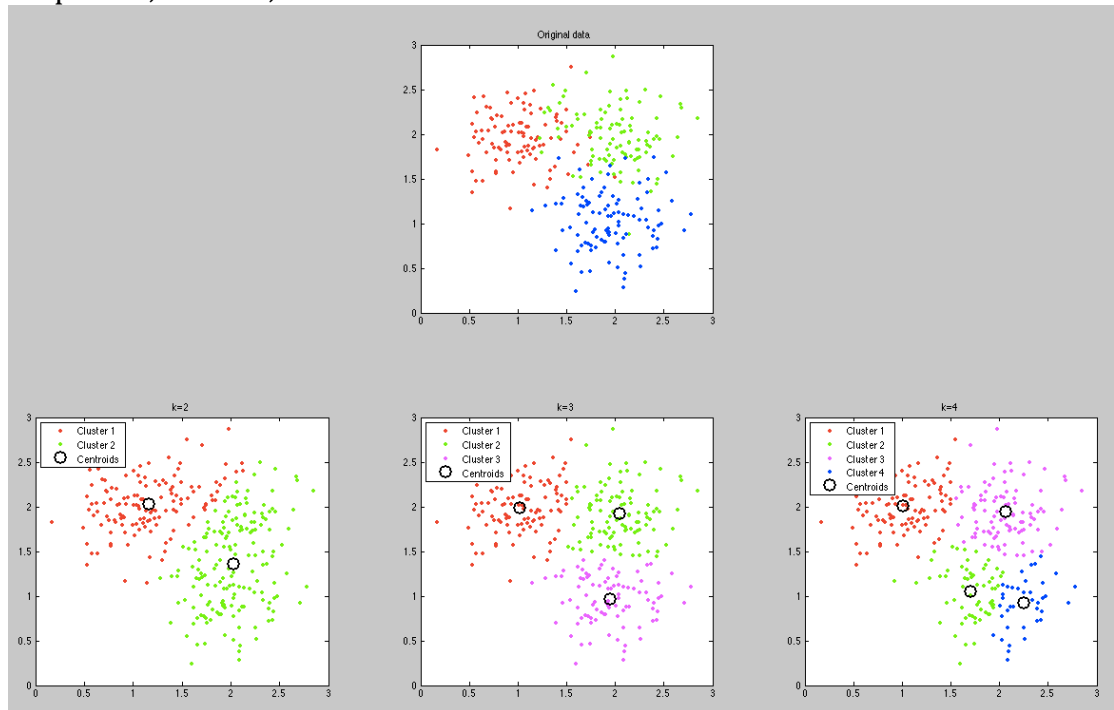


Figura 2. Datos originales y el resultado del k-means con  $k=2,3$ , y 4.

### 3.3 Métodos de agrupación: segmentación en el espacio RGB

- a) Lee la imagen 'loro.png'. Conviértela a escala de grises y aplica la segmentación con el *kmeans*. Prueba diferentes valores de  $k$  para encontrar la mejor segmentación.

Nota: Fíjate cómo se han convertido los píxeles a características y en qué formato se han de pasar al *kmeans*.

Nota: El *kmeans* funciona mejor cuando las características son como mínimo 2D. Por eso, en el caso 1D crea una matriz de dos columnas en las que ambas coinciden.

- b) Visualiza la imagen segmentada utilizando el nivel de gris promedio encontrado con el método de segmentación. ¿A qué corresponde?
- c) Añade como características las coordenadas de los píxeles y comprueba si mejora el resultado de la segmentación.
- d) Visualiza el resultado anterior en una figura junto con la distribución de sus colores (utiliza la función *plotPixelDistribution* facilitada con el enunciado).

- e) A partir de la imagen de entrada, crea una matriz que contenga en cada fila la tripleta RGB de un píxel de la imagen. Tendrá tantas filas como píxeles haya en la imagen.
- f) Utilizar el método *kmeans* para reducir el número de colores de la imagen a 16 colores diferentes.
- g) Visualizar en una misma figura las imágenes del primer apartado, y la imagen con 16 colores junto con su distribución de colores (Figura 3).

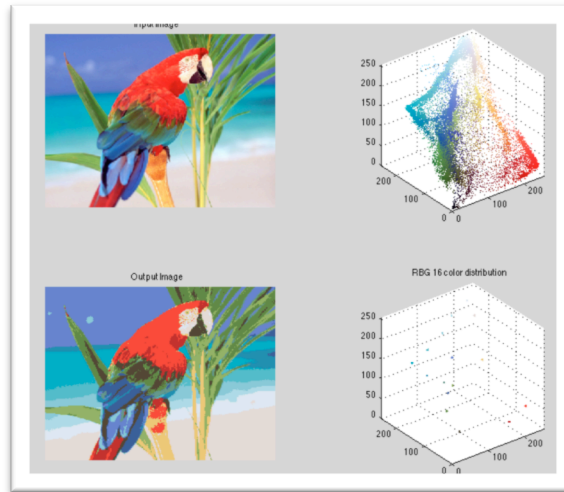


Figura 3.

### 3.4 Extracción de descriptores

Para recordar los conceptos, conocer la librería que implementa el SIFT: <http://www.vlfeat.org/index.html>. Para instalarla en Matlab, necesitamos ejecutar:

```
% Install the VLFeat toolbox
srcPath=cd('toolbox');
vl_setup;
cd(srcPath);
close all;
```

Revisar el uso del método SIFT en el tutorial: <http://www.vlfeat.org/overview/sift.html>.

A continuación, utilizar el fichero ej34.m, que se encuentra en el material de la práctica. En el código hay comentarios indicando el apartado correspondiente. Analizar el código y contestar a las preguntas:

- a) ¿A qué corresponden las variables *f* y *d* que devuelve el método *vl\_sift*? ¿Qué tamaño tienen? ¿A qué corresponden sus valores?
- b) En este apartado se muestran los puntos característicos detectados con SIFT en una misma imagen con rotación. Compara los resultados obtenidos antes y después de hacer la rotación y comenta lo que ves. ¿Hay invariancia a rotación? ¿Qué significa la línea que aparece en el interior de los círculos?

- c) En este caso, se comparan los resultados con una misma imagen a distintas escalas. Compara los resultados obtenidos antes y después de hacer el reescalado y comenta lo que ves. ¿Hay invariancia a escala? ¿Qué significa el tamaño de los círculos que se muestran?
- d) En este apartado se genera una imagen sintética y se calcula el descriptor en dos puntos distintos de la imagen. Verás que se generan dos figuras con el mismo formato, pero con una pequeña diferencia en el cálculo del descriptor. Fíjate en los descriptores de la fila inferior del *subplot*. ¿Qué diferencia encuentras entre los mostrados en la primera figura y la segunda?
- e) **[OPCIONAL]** Repite el mismo experimento que en el apartado anterior, pero analizando el punto del vértice (50,50). ¿Qué pasa en este caso?

### 3.5 Reconocimiento por alineación de puntos característicos

Crea un fichero ej35.m y, utilizando la función *showMatches* proporcionada con el enunciado, implementa los siguientes experimentos:

- a) Utiliza el método *showMatches* usando como modelo la imagen 'starbucks.jpg' y como (nueva) escena la imagen 'starbucks6.jpg'.
- b) Repite el experimento usando otras imágenes como modelo y/o escena. Para cada imagen modelo, enseña el resto de imágenes de Starbucks en orden de su semejanza con el modelo.
- c) ¿Qué pasa si se le pasa una imagen que no contiene el logo de Starbucks? ¿Qué método propondrías (sin implementarlo) para definir la probabilidad que la imagen "escena" corresponde a la imagen "modelo"?
- d) Repite el experimento 4 veces cambiando las escalas y orientaciones del modelo. Comenta tus observaciones.
- e) **[OPCIONAL]** Repite el experimento con un conjunto de imágenes nuevo. Por ejemplo, con anuncios de Coca-Cola u otra marca que fácilmente puedes encontrar en internet.

### Entrega de la práctica

Tiempo máximo de entrega: martes 22 de Noviembre a las 23:00h.

Material para entregar: archivo "nombres\_apellidos\_P3.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.