

Práctica 3

Visión Artificial

Aarón Negrín y Miriam Bermúdez

3.1 Métodos de “background subtraction”

Para el método de background subtraction hemos tenido que aplicar dos pasos

1. Encontrar una métrica apropiada para la distinción de los shots. Para ello hemos recurrido a la función de Matlab ‘imabsdiff’ que nos permite calcular la diferencia en valor absoluto entre cada uno de los píxeles, tras ello binarizamos la imagen y tomamos como distancia la cantidad de píxeles distintos para un cierto threshold..
2. Tras ello calculamos la mediana de los frames desde que se encontró el shot actual hasta la finalización de este.

Los mayores inconvenientes que nos podemos encontrar sin duda son:

- Extracción de demasiados shots: Es decir, nuestra métrica considera que frames muy parecidos son totalmente distintos.
- Shots no extraídos de forma correcta: Nuestra métrica no es capaz de discernir un cambio brusco entre frames, como resultado la extracción del fondo podría verse distorsionada.

Las imágenes obtenidas finalmente tras aplicar el algoritmo corresponden a una aproximación del fondo de la escena.

Sin embargo este algoritmo presenta fallos, en el caso de la métrica, esta se vería afectada gravemente por cambios de luminosidad, y en el caso de la mediana, ante un shot con pocos frames, la información sobre el background podría no llegar a ser relevante.

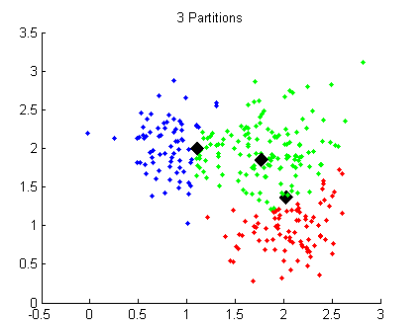
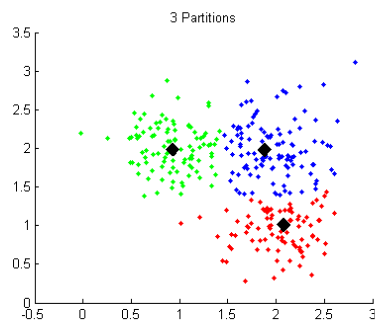
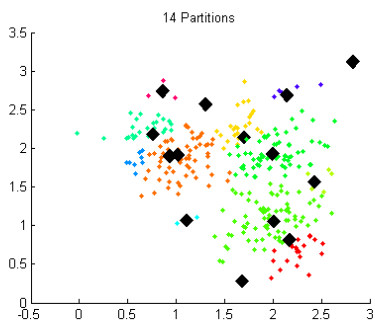
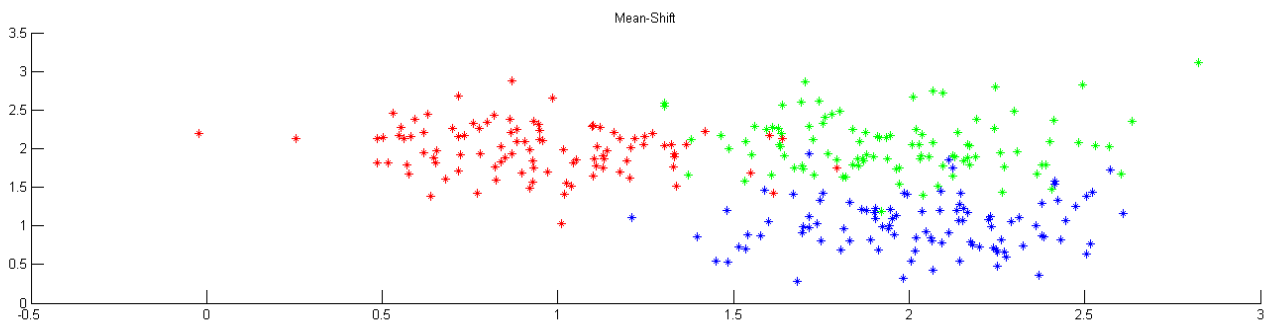
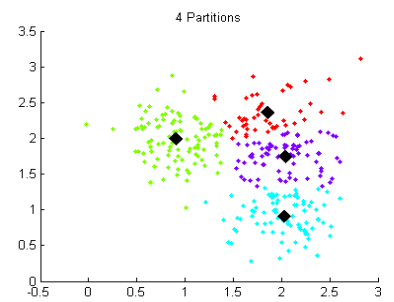
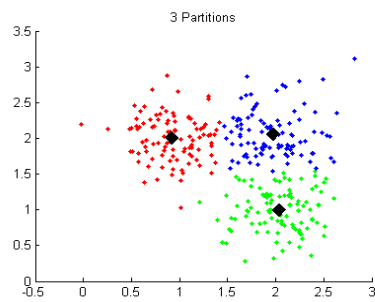
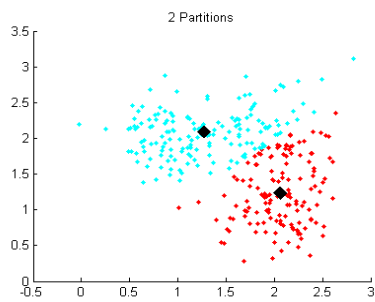
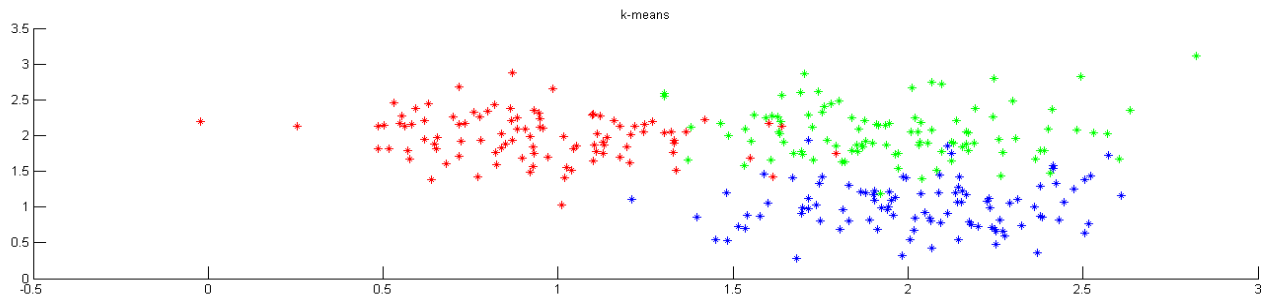
Entre las posibles aplicaciones que se le pueden dar a un extractor del fondo sería por ejemplo el de eliminar objetos molestos que introduzcan ruido en la imagen. Por ejemplo en algunos modelos de cámaras de fotos estas permiten eliminar viandantes o coches de la imagen.

3.2 Métodos de agrupación de datos numéricos

A la hora de realizar clustering sobre datos numéricos hemos recurrido a dos algoritmos distintos: k-means y mean-shift.

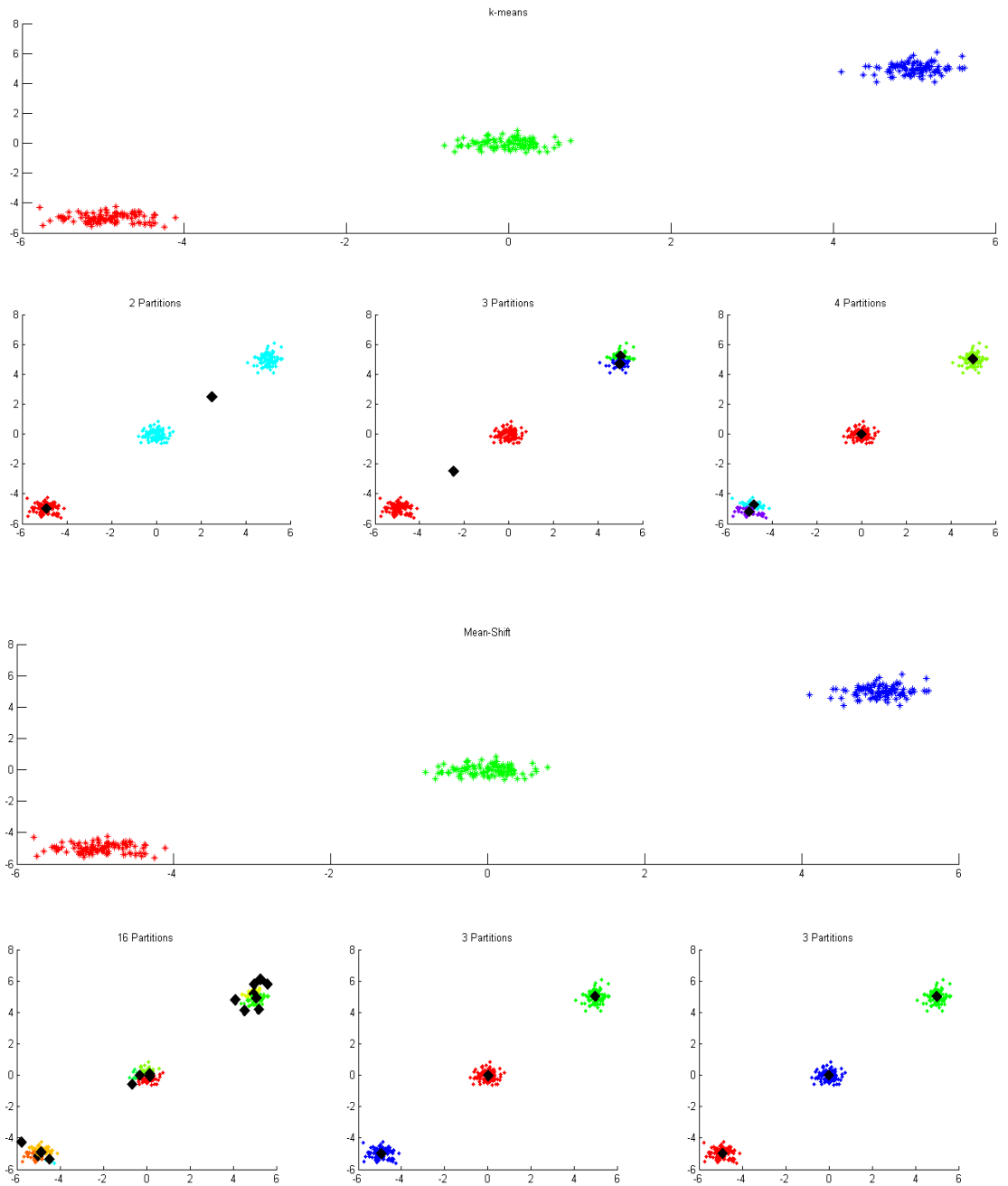
En un caso como este tenemos que tener un factor en cuenta, y es que sabemos de qué forma han sido generados los datos y cuantos ‘paquetes’ de datos tenemos realmente. En una situación así cuando iniciamos k-means con el número exacto de clusters suele dar mejores resultados que mean-shift, algoritmo el cuál en algunas ocasiones puede llegar a generar más clusters de datos de los que realmente existen.

Sin embargo, si no tuviéramos conocimientos de antemano de la cantidad de clusters existentes, ante una nube de datos tan difusa mean-shift, aparte de ser más eficiente podría esclarecer que tipo de relación mantienen los datos entre sí.



Otra de las diferencias de ambos algoritmos es que cuando no nos encontramos ante una nube de puntos tan compacta si no que más bien se presenta de forma no conexas mean-shift funciona mucho mejor que k-means, incluso en el caso de que se introduzca el número exacto de clusters.

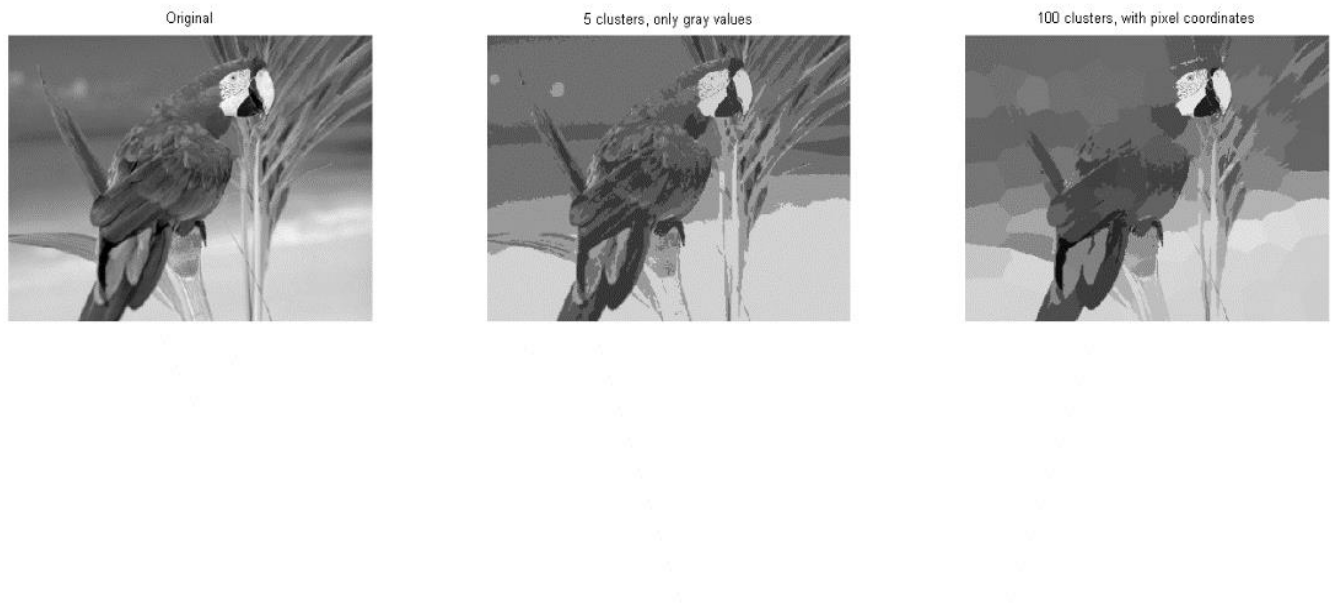
Este comportamiento se debe al hecho de que k-means es perturbable ante extremos locales que pueden alterar el resultado.



3.3 Métodos de agrupación: Segmentación en el espacio RGB

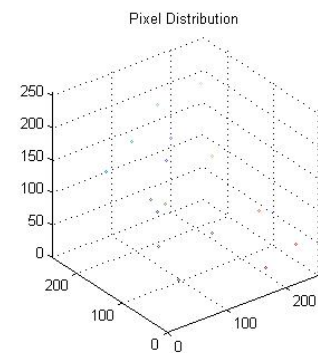
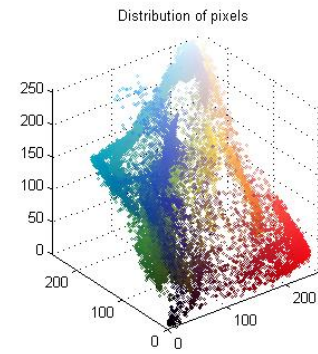
Para los primeros apartados hemos recurrido al clustering haciendo uso del nivel de gris de la imagen.

Tomando únicamente los valores de gris como referencia con una cantidad relativamente baja de clusters, entre 5 y 15, pueden obtenerse buenos resultados que sean capaces de particionar la imagen en conjuntos de una forma razonablemente buena, sin embargo cuando añadimos las coordenadas de los píxeles se requiere de un alto número de clusters



para obtener un resultado coherente.

Esto se debe al hecho de que mientras que al usar únicamente los valores de grises genera clusters dispersos que respeten un cierto valor de gris mientras que al añadir las coordenadas se exige que dichos valores también posean coherencia espacial y que se encuentren cercanos unos de otros.



En el segundo apartado considerando los tres canales de RGB se consiguió un clusterizado de los datos coherentes mucho más efectivo que contando únicamente con la escala de grises, de hecho incluso se obtienen buenos resultados en la clusterización si se añaden datos espaciales como se puede apreciar en la siguiente figura:



Manteniendo además poca cantidad de clusters utilizados

3.4 (Opcional) Métodos de agrupación: Segmentación en el espacio RGB con Mean-shift

3.5 Extracción de descriptores

a. ¿A qué corresponden las variables `f` y `d` que devuelve el método `vl_sift`? ¿Qué tamaño tienen? ¿A qué corresponden sus valores?

Como dicen las propias variables, "`f`" es donde se encuentra el marco de las imágenes, es decir, donde se encuentran nuestros puntos de interés, y "`d`" es el descriptor del frame correspondiente en "`f`". En este caso, "`f`" tiene un tamaño de "4x282" y con valores comprendidos entre [-4.7069,207.07], mientras que "`d`" tiene un tamaño de "128x282" y con valores comprendidos entre [0,180]. Como hemos visto antes, nuestra matriz "`f`" tiene una columna para cada frame, donde `f` (1: 2) corresponde a la posición, `f` (3) a la escala y `f` (4) a la orientación.

b. ¿Si rotas una imagen, se encuentran las correspondencias entre la imagen original y la rotada? ¿Es decir, hay invariancia a rotación? ¿Qué significa la línea que aparece en el interior de los círculos?

Yes, when we executed the code, in the figure 1 we saw the image and the rotated image (90°) and we could appreciate that the characteristics points are conserved in the rotated image in relation with the original image, that means there is no invariance in the rotation. The line that appears inside the circles is the higher orientation value of the descriptor and if they are more than one line inside a circle, it means there are orientations with the same value.

Así es, cuando ejecutamos el código, en la figura 1 vimos la imagen y la imagen girada (90°) y pudimos apreciar que los puntos de características se conservan en la imagen girada en relación con la imagen original, lo cual significa que es invariante frente a rotaciones.

La línea que aparece dentro de los círculos es la orientación con mayor valor del descriptor y si hay más de una línea dentro de un círculo, significa que hay orientaciones con el mismo valor.

c. Si reescalas una imagen, se encuentran las correspondencias entre la imagen original y la reescalada? ¿Es decir, hay invariancia a escala? ¿Qué significa el tamaño de los círculos que se muestran?

Sí, hay correspondencia entre la imagen original y las redimensionadas (como se puede apreciar en la figura 2), lo que significa que no hay invariancia frente al escalado.

Sin embargo la única diferencia que vemos es que hay menos círculos en las imágenes redimensionadas.

Por otra parte, el tamaño de los círculos nos muestran la escala de los puntos característicos.

3.6 Reconocimiento por alineación de puntos característicos

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’?

If we use as scene an image that not contains the Starbucks' logo, we can see that there are a few characteristic points and dispersedly.

Si se hace uso de una imagen que no contenga el logo de Starbucks se puede apreciar que existen unos pocos puntos característicos dispersados.

d. Repite el experimento varias veces cambiando las escalas y orientaciones del modelo. Comenta tus observaciones.

Hemos visto que cuando rotamos la imagen original y hacemos el matching tenemos que los puntos de característicos son prácticamente los mismos que cuando realizamos el matching sin haber realizado la rotación, pero cuando se cambia el tamaño de la imagen original, podemos apreciar diferencias en el matching entre la imagen escalada y la original, con esto se ve que la imagen escalada consigue menos matchings con las otras imágenes que la original, y esto es debido a la pérdida de información sufrida al realizar el escalado.