

# Lab 5: Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500)

Duván Alberto Gómez Betancur  
Visión Artificial  
Universidad de los Andes  
da.gomez16@uniandes.edu.co

## Abstract

*El presente documento contiene el informe resumen de los resultados obtenidos en los laboratorios 4 y 5 del curso de Visión Artificial. Comprende la implementación de diferentes algoritmos de segmentación estudiados como k-means, Gaussian Mixture Model, segmentación jerárquica y Watershed. Se presentan las pruebas y la respuesta de los algoritmos, evaluando las segmentaciones y los tiempos de cómputo. Además se presentan los resultados obtenidos al aplicar dos de los algoritmos sobre la base de datos Berkeley Segmentation Data Set, comparándolos con los resultados del algoritmo de segmentación conocido como Ultrametric Contour Map (UCM) desarrollado por el grupo de Visión Artificial de la Universidad de Berkeley.*

## 1. Introducción

La segmentación de imágenes es el proceso de dividir una imagen digital en varias partes u objetos, es decir, para efectos técnicos, se trata de dividir una imagen en grupos de píxeles representativos de zonas, regiones u objetos.

El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar tanto por parte de un humano como por parte una máquina.

En la presente práctica de laboratorio se estudiaron cuatro algoritmos de segmentación diferentes, a saber: k-means, gaussian mixture model (gmm), segmentación jerárquica y watershed.

Los algoritmos fueron estudiados en el curso de visión artificial y se implementaron utilizando Matlab R2014a aprovechando las funciones que tiene, como *kmeans*, *fitgmdist* y *linkage* entre otras.

Los métodos fueron validados inicialmente con un conjunto de cuatro imágenes de prueba, posteriormente a partir de esos resultados se seleccionaron los métodos k-means y gmm para probarlos sobre la base de datos Berkeley Segmentation Data Set (BSDS500).

El documento está organizado de la siguiente manera: en

la sección II se presentan los algoritmos de segmentación seleccionados, en la sección III las pruebas y los resultados de las mismas y finalmente en la sección IV está la discusión de los resultados y las posibles mejoras.

## 2. Métodos

Como se mencionó anteriormente se realizó la implementación de los algoritmos k-means, gmm, segmentación jerárquica y watershed en Matlab.

La implementación consistió en el desarrollo de una función denominada *segment\_by\_clustering* cuyo llamado se da de la siguiente manera.

```
segment_by_clustering( rgb_image,  
feature_space, clustering_method,  
number_of_clusters)
```

dónde:

*rgb\_image* es la imagen original en el espacio de color RGB.

*feature\_space* es el espacio de color en el que se desea trabajar, que para este caso puede ser RGB, L\*a\*b o HSV.

*clustering\_method* es el método de segmentación deseado y

*number\_of\_clusters* es el número de clusters o regiones que se quieren identificar, es decir, en las que se desea segmentar la imagen original.

En la sección III se presentan como ejemplo las imágenes resultado al ejecutar la función descrita, con una imagen en el espacio de color RGB y segmentada con cada uno de los algoritmos mencionados.

A partir del resultado obtenido en cuanto al tiempo de cómputo, se seleccionaron los algoritmos k-means y gmm para probar sus funcionamiento sobre la base BSDS500 propiedad de la Universidad de Berkeley pero que se encuentra pública para la comunidad de investigadores en el área de visión artificial.

La base BSDS500 está compuesta de 500 imágenes en ambientes no controlados, divididas en tres conjuntos: entrenamiento, validación y pruebas. Cada imagen cuenta con una segmentación manual que rige como verdad terreno para probar la precisión de los algoritmos evaluados.

A continuación se presentan los algoritmos de k-means y gmm, los cuales arrojaron resultados satisfactorios y fueron los dos algoritmos que tomaron los menores tiempos de cómputo, razón por la cual fueron seleccionados para probar sobre la base de imágenes BSDS500.

## 2.1. K-means

En el método de segmentación K-means se busca agrupar los píxeles de una imagen en k grupos diferentes, los cuales están caracterizados por un punto denominado centroide y a los cuales se asocian las observaciones, en este caso píxeles, considerando aquel grupo al cual la medida de distancia sea la menor, es decir, cada píxel de la imagen pertenece al grupo más cercano a la media.

La función implementada en Matlab para el algoritmo de K-means utiliza la función propia de Matlab llamada *kmeans*, y recibe como parámetros la imagen original que se desea segmentar y el número de regiones en las que se desea segmentar la imagen.

A continuación se presenta un fragmento del código implementado:

```
if strcmp(clustering_method,
'kmeans') [cluster_idx,C] =
kmeans(img,number_of_clusters);
pixel_labels =
reshape(cluster_idx,nrows,ncols);
my_segmentation = pixel_labels;
```

## 2.2. GMM

El método de Gaussian Mixture Model (GMM) implementa el algoritmo *expectationmaximization* para ajustar un modelo de mezcla de Gaussianas a un conjunto de datos o píxeles.

Se trata de un algoritmo de segmentación con asociación suave de los elementos, es decir, se entrega una probabilidad de confianza de la pertenencia de un píxel a una región dada a partir de la medida de la distancia del píxel a la media de la distribución gaussiana representativa de la región y se asocia cada píxel de la imagen al cluster o región al cual esa distancia sea menor.

La función implementada en Matlab para el algoritmo de GMM utiliza la función propia de Matlab llamada *fitgmdist*, a la cual se le envían como parámetros la imagen a segmentar y el número de regiones a identificar.

A continuación se presenta un fragmento del código implementado:

```
if strcmp(clustering_method, 'gmm')
gmm = fitgmdist(img,number_of_clusters);
gmmC = cluster(gmm, img); pixel_labels
```



Figure 1. Imagen original

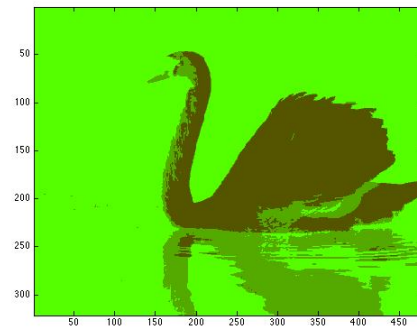


Figure 2. Resultado segmentación con k-means

```
= reshape(gmmC,nrows,ncols);
my_segmentation = pixel_labels;
```

## 3. Pruebas y Resultados

Como se mencionó anteriormente se implementaron los 4 métodos de segmentación y se probó la respuesta de cada uno para la misma imagen, la evaluación de los resultados de cada método se hizo considerando calidad de la segmentación entregada y el tiempo de cómputo.

En la prueba que corresponde a los resultados que se muestran, la imagen original que se utilizó se encontraba en el espacio de color RGB. Dicha imagen se conservó en el espacio de color RGB durante la segmentación y se le definieron 3 regiones de segmentación en cada algoritmo probado.

En la Figura 1 se muestra la imagen original de prueba que se utilizó.

En las Figuras 2 a Figura 5 se presentan los resultados de la implementación de los algoritmos k-means, gmm, segmentación jerárquica y watershed respectivamente.

A continuación se detallan los tiempos de cada uno de los algoritmos

A partir de éstos resultados y considerando el tiempo de cómputo de cada uno de los algoritmos, se decidió entonces

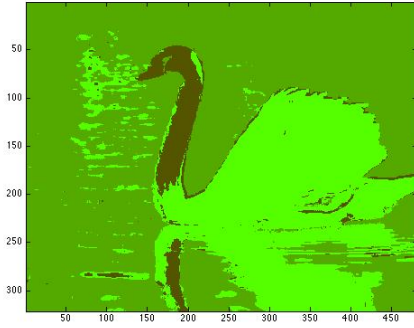


Figure 3. Resultado segmentación con gmm

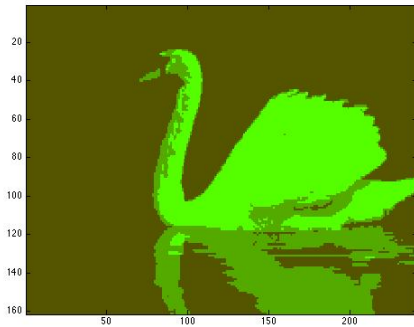


Figure 4. Resultado segmentación jerárquica

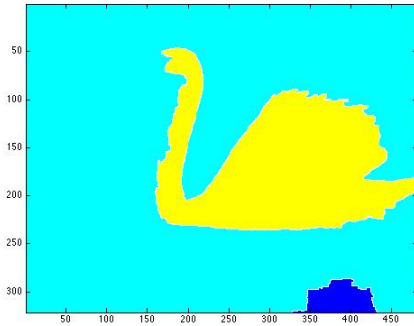


Figure 5. Resultado segmentación con watershed

utilizar los algoritmos k-means y gmm para probarlos sobre la base de imágenes BSDS500 ya que esos dos algoritmos fueron los que menos tiempo tardaron en entregar los resultados de las segmentaciones.

En la Figura 6 se puede observar la gráfica con las curvas Precision Recall al evaluar los métodos K-means (línea negra) y GMM (línea roja) sobre la base de datos BSDS500, comparados con los resultados del algoritmo UCM (línea

Método	Tiempo(Segs)
K-Means	3.23
GMM	5.34
Hierarchical	23.79
Watershed	9.07

Table 1. Resultados Tiempo de Cómputo

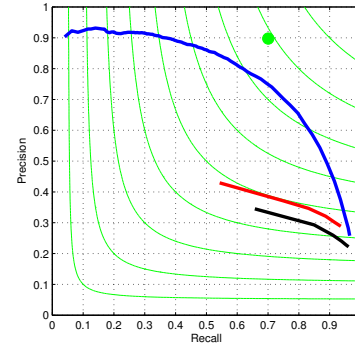


Figure 6. Curva precisionrecall sobre la base BSDS500. K-means (negra), GMM (roja), UCM (azul).

azul).

Para la obtención de las curvas se corrieron los algoritmos de k-means y gmm sobre la base BSDS500 variando en cada caso el número de regiones entre 1 y 7. Además se aprovecharon los códigos disponibles en la página web del proyecto BSDS500.

## 4. Discusión y posibles mejoras

Como se puede observar en las Figuras 2 a la 5 los diferentes algoritmos de segmentación entregan resultados aceptables al correr sobre imágenes en el espacio de color RGB.

Es importante mencionar que cada uno de los algoritmos de segmentación tiene un conjunto de parámetros y limitaciones asociadas, así por ejemplo en el caso de k-means, se trata de un algoritmo completamente dependiente del número de clústeres, es decir, del número  $k$ , ya que al realizar la prueba con un número mayor de  $k$ , los resultados tienden a ser mejores.

Por su parte el algoritmo GMM es limitado ya que se basa en la presunción de que el conjunto de observaciones, en este caso el conjunto de píxeles dentro de la imagen, asociados a cada cluster, sigue el comportamiento de una distribución normal o campana de Gauss y por lo tanto dependerá tanto de eso como del número de distribuciones que se usen, es decir el número de regiones que se definan, de la misma manera que sucede en k-means.

Sin embargo de manera general al analizar los resultados

presentados en las Figuras 2 a 5 se puede observar claramente que para el caso de la imagen estudiada, los algoritmos implementados presentan buenos resultados, logrando en su mayoría segmentar correctamente el objetivo principal de la imagen que en este caso es un cisne.

Por otra parte, analizando los tiempos de respuesta presentados en la Tabla 1, se puede identificar que el algoritmo que más tarda en realizar la segmentación es el jerárquico y los más rápidos son k-means y gmm.

Ahora bien, analizando el comportamiento de los algoritmos de k-means y gmm sobre la base de datos BSDS500 es claro que tienen un comportamiento muy pobre y bastante lejano de los resultados de aproximaciones como UCM.

De los dos algoritmos probados, el que tuvo mejor comportamiento en la base BSDS500 fue gmm logrando valores de precisión de aproximadamente 45% con un recall de 55%, mientras que usando K-means sólo se logra llegar a una precisión de aproximadamente 35% con un recall de 65%.

Sería interesante poder probar los algoritmos sobre la base BSDS500 variando el parámetro de regiones con un número mayor y así poder construir la curva completa de cada método.

Finalmente, como se mencionó anteriormente, se podría pensar que una forma de mejorar el comportamiento de los algoritmos de segmentación, tanto en los casos de la imagen individual como en las pruebas sobre la base BSDS500 es a través de una posible combinación entre las características de los métodos y/o la modificación de los parámetros libres de cada uno, es decir por ejemplo con el incremento del valor de k en el caso de k-means y aumentando el número de distribuciones gaussianas en el caso de GMM.

## References

- [1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.