

# 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, comparados con los resultados del algoritmo de segmentación conocido como Ultrametric Contour Map (UCM).*

## 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 y, como se verá en la siguiente sección, aprovechando las funciones que tiene.

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 presenta un resultado ejemplo al ejecutar la función descrita.

A partir del resultado obtenido en el tiempo de cómputo, se seleccionaron los algoritmos k-means y gmm para probar sus respuestas 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 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 al cual la medida de distancia sea la menor, es decir, que el píxel 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 se le envían la imagen a segmentar y el número de regiones a identificar.

### 2.2. GMM

El método GMM implementa el algoritmo *expectation-maximization* para ajustar un modelo de mezcla de Gausianas a un conjunto de datos o píxeles.

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.

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.

## 3. Pruebas y Resultados

Como se mencionó anteriormente se implementaron 4 métodos de segmentación y se probó la respuesta de cada uno para la misma imagen, considerando calidad de la segmentación y tiempo de cómputo.

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.

Para todos los resultados que se muestran se utilizó el espacio de color RGB y 3 regiones de segmentación.

A partir de éstos resultados y considerando el tiempo de cómputo de estos algoritmos, se decidió entonces utilizar los algoritmos k-means y gmm para probarlos sobre la base de imágenes BSDS500.

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 azul).

Para la obtención de las curvas se corrieron los algoritmos de k-means y gmm sobre la base BSDS500 variando



Figure 1. Imagen original

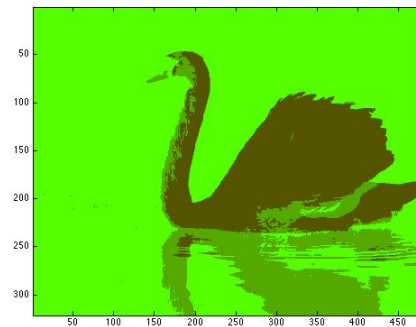


Figure 2. Resultado segmentación con k-means

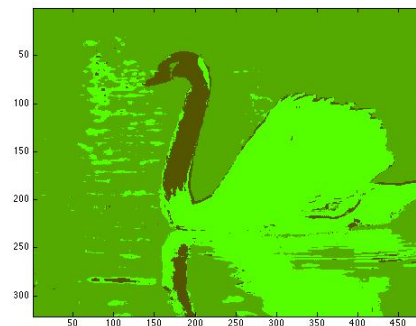


Figure 3. Resultado segmentación con gmm

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

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.

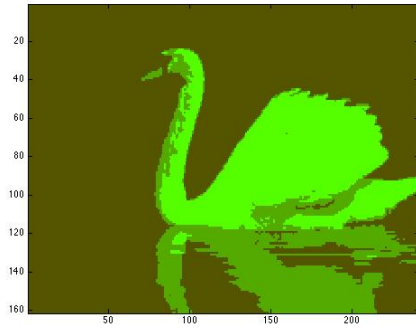


Figure 4. Resultado segmentación jerárquica

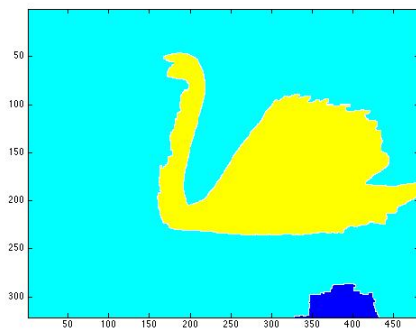


Figure 5. Resultado segmentación con watershed

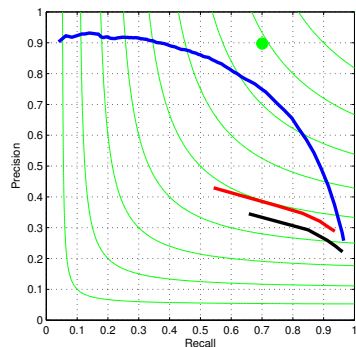


Figure 6. Curva precisionrecall sobre la base BSDS500

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 parte de la presunción de que el conjunto de observaciones, en

este caso píxeles, 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 regiones que se definan.

Sin embargo de manera general al analizar las figuras se puede observar claramente el buen comportamiento de los algoritmos de segmentación.

Como se mencionó anteriormente una forma de mejorar el comportamiento de los algoritmos de segmentación es a través de sus parámetros libres, es decir por ejemplo con el incremento del valor de  $k$  en el caso de k-means.

## 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.