

# Computer Vision Lab 05 -Segmentations BSD500

Juan Carlos Leon Alcazar  
Universidad de los Andes  
jc.leon@uniandes.edu.co

## 1. Introducción

El presente laboratorio sirve como introducción al benchmark conocido como 'The Berkeley Segmentation Dataset and Benchmark' (BSD500), un dataset y benchmark público, que abordó por primera vez de forma sistemática y reproducible el problema de segmentación de objetos en imágenes naturales. El objetivo principal es evaluar un conjunto de métodos de segmentación construidos en el laboratorio previo y compararlos con el método 'ultrametric contour map' (UCM) [1].

## 2. Métodos de segmentación:

Para el presente laboratorio se seleccionaron 4 métodos de segmentación:

**KMeans** Construye un conjunto de K clusters según K centroides que minimicen la suma total de las distancias de los puntos a los centroides <sup>1</sup> [5]

**Mezcla de Gaussianas (GMM)** Construye un conjunto de K clusters donde se tiene un ajuste óptimo de los elementos a un conjunto de K gaussianas (clusters) parametrizadas por un conjunto de K medias y matrices de covarianza. <sup>2</sup> [3]

**Hierarchical Cluster** Para cierta distancia D, se crea una jerarquía de elementos que son agrupados de manera jerárquica según la similaridad que existe entre ellos. <sup>3</sup> [2]

**Watershed** Debido a que el método base produce una sobre segmentación de las imágenes en la base de datos. <sup>4</sup> [6]

<sup>1</sup>La implementación se encuentra en KmeansSegment.m

<sup>2</sup>Para resolver algunos problemas de convergencia en la implementación de MATLAB se utilizó un regularizador con valor de 0.0001. La implementación se encuentra en GMMSegment.m

<sup>3</sup>Para la implementación de MATLAB se utilizó una distancia euclidiana normalizada por la desviación estándar. La implementación se encuentra en HierarchicalSegment.m

<sup>4</sup>Se agregó una paso de inicialización donde se trata de estimar las re-

## 2.1. Espacios de segmentación

Las alternativas para el espacio de color fueron:

**RGB** es el espacio de color utilizado por múltiples dispositivos electrónicos, los tres canales (rojo, verde y azul) están altamente correlacionados [4]

**LAB** Es una extensión del espacio de color CIE XYZ, que además de evitar la correlación entre canales es perceptualmente uniforme.[4]

**HSV** Al igual que LAB elimina la correlación entre los tres canales, contiene información sobre el matiz, saturación y valor del color[4]

**Coordenadas** Adicionalmente se utilizaron las coordenadas del pixel para agregar información espacial al proceso de segmentación. Se tuvo cuidado de normalizar los valores de color de los pixels (en cualquier espacio) y las coordenadas de tal manera que la magnitud de los dos fuera similar. Esto con el fin de evitar algún sesgo en el proceso de cluster hacia las coordenadas puesto que su rango de valores es mucho más grande que rango de valores de los espacios de color.

De esta manera se cuenta con 6 espacios de color:

- RGB
- LAB
- HSV
- RGB+xy
- LAB+xy
- HSV+xy

giones de la imagen donde existen objetos grandes y continuos a fin de evitar que el algoritmo de watershed genere clusters en esas regiones. Para este fin se aplicó un filtro mediano a la imagen (eliminación de bordes espurios) sobre la imagen filtrada se aplicó el detector de bordes de canny y a los bordes se les realizó la operación morfológica de dilatación afin de para expandirlos. Finalmente el inverso de la imagen fue utilizado como máscara para establecer las regiones donde el. La implementación se encuentra en WatershedSegment.m

## 2.2. Selección de los métodos

Existen 24 combinaciones iniciales de parámetros, se tiene en cuenta los espacios de color y métodos de segmentación (6 espacios de color, 4 métodos de segmentación).

La cantidad de combinaciones aumenta si se considera el valor de K que a priori puede tomar cualquier valor mayor 2 (1 es un valor válido para cualquier método, pero produciría una segmentación trivial). A fin de estimar un rango de valores adecuado para K, analizamos el número de clusters en el set de 'train'. Inicialmente calculamos el promedio de clusters de todas las imágenes en set de entrenamiento. Encontramos un promedio de 19.64 clusters, con una desviación estándar de 17.47. Esto indica que el 95% de las imágenes tiene entre 0 y 55 anotaciones. Con este análisis inicial, una búsqueda exhaustiva de los mejores parámetros posibles implicaría segmentar cada imagen 1320 veces. En la práctica no se cuenta con tanto tiempo disponible así que utilizamos dos estrategias para reducir el número de pruebas.

Primero, estimamos un rango menor para K, si bien hay imágenes que tienen una gran cantidad de clusters (hasta 95 regiones anotadas distintas) la mayoría de estas regiones cubren un área muy pequeña en la imagen. Con esto en mente se decidió incluir en el promedio solo aquellos clusters que ocupen más del 2% de los pixels de la imagen <sup>5</sup>

Después de este cambio, se obtiene un promedio de 6.42 clusters por imagen, con una desviación estándar de 3.28. Esto indica que el 95% de las imágenes tiene entre 2 y 13 clusters. Después de esta reducción se tiene todavía que segmentar cada imagen 244 veces, sumado al tiempo computacional de la evaluación del método, el tiempo de ejecución todavía es prohibitivo.

Para solucionar esto, se aplica una segunda estrategia para evitar hacer pruebas exhaustivas sobre todos los parámetros. Se escoge al azar un subconjunto de 10 imágenes del dataset de 'test', aplicar los 4 métodos de segmentación, en 6 espacios de color y para un valor de K que corresponde a la media (6). A fin de inspeccionar visualmente algunas de las 240 imágenes generadas. El objetivo es tratar de reconocer que combinación de parámetros produce una segmentación que visualmente parece ser más adecuada.

Algunas de las imágenes obtenidas se presentan en las figuras 1, 2 y 3

Se puede observar que, en general, los resultados de K-Means y Watershed tienden a estar sobre segmentados, (especialmente el caso del watershed). Esta sobre segmentación fue el principal argumento para descartar los

métodos de K-Means y watershed aun cuando se utilicen los espacios de colores aumentados con las coordenadas xy.

Los resultados del clustering jerárquico y el GMM son muy similares, sin embargo el cluster jerárquico incluye una operación de submuestreo de la imagen (para evitar consumir toda la memoria del computador). Al incrementar el tamaño de la imagen se pierden detalles especialmente en las zonas donde hay bordes o patrones de alta frecuencia espacial. Es posible que esta pérdida de información reduzca el desempeño durante la de evaluación. De esta manera se selecciona como mejor método el GMM.

En cuanto a los espacios de color, la segmentación en el espacio de color RGB es bastante más ruidosa que los espacios de LAB y HSV (ver figura 1 y figura 6), esto se debe, posiblemente, a que los canales de RGB están bastante correlacionados y es poca la información adicional que se puede extraer si se utilizan 2 o 3 canales.

De esta manera se prefieren los espacios de color LAB y HSV sobre RGB. Finalmente la diferencia entre el uso o el no uso de las coordenadas no es evidente en ninguna de las imágenes de prueba, sin embargo, se prefiere el uso de las coordenadas a fin proveer información espacial al método de clustering.

De esta manera la inspección visual concluye con la selección de 2 métodos de clustering

1. Espacio de color LAB+xy y cluster por GMM
2. Espacio de color HSV+xy y cluster por GMM

## 3. Dataset y Metodología de pruebas

La base de datos berkeley BSD500 contiene 500 imágenes separadas en tres grupos: training: (200) test (200) y validación (100) todas las imágenes tienen el mismo tamaño 481x321 píxeles, tienen formato jpg, son imágenes a color y se encuentran numeradas para poder ser identificadas dentro del dataset. La subdivisión del dataset tiene el siguiente objetivo: Sobre el conjunto de 'train' se prueba inicialmente los métodos de segmentación y se busca un conjunto de hiperparámetros que permita obtener la mejor segmentación posible. El conjunto de test existe para validar el modelo de aprendizaje y los parámetros encontrados en set de 'train' de tal manera que se tengan resultados aceptables dentro de un conjunto de imágenes no visto antes. Cuando se encuentra un modelo e hiperparámetros que presentan un buen desempeño tanto en el conjunto de train como el de test, se usa el conjunto de validación para establecer la calidad del método (precisión, exhaustividad y F-Medida en este caso).

### 3.1. Verdad de terreno BSD500

Cada uno de las subdivisiones del dataset cuenta con una 'verdad de terreno' (Ground Truth) que se encuentra

<sup>5</sup>No existe una razón formal para elegir el umbral del 2%, sin embargo, un clúster que agrupa menos del 2% de los pixels imagen, agruparía menos de 60 pixels; Es poco probable que los métodos utilizados tengan la capacidad de identificar y segmentar una región tan pequeña



Figure 1. Segmentación usando RGB de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6.



Figure 2. Segmentación usando LAB de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6.



Figure 4. Visualización del ground Truth de una de las imágenes en el dataset

dentro del directorio ‘.../BSDS500/data/groundTruth’ esta almacenada dentro de archivos binarios con el formato de MATLAB. y está indexada según el número de imagen en la base de datos. La verdad de terreno de una imagen cualquiera se puede cargar con el comando:

```
‘gt=load(‘.../BSR/BSDDS500/data/groundTruth/2018.mat’)
```

La anotación tiene el mismo tamaño de la imagen original y consta de un único número entero para designar el cluster (el objeto) al que pertenece el píxel. Con ayuda de la función ‘imagesc’ de MATLAB se puede visualizar algunas de las verdades de terreno (ver figura 4).

### 3.2. Benchmark BSD500

El objetivo principal del benchmark construido sobre BSD500 es ofrecer una metodología estandarizada para la evaluación de métodos de segmentación en imágenes naturales. La metodología está basada en la evaluación de las



Figure 8. Segmentación usando HSV+xy de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6. Objetos muy grandes como el cielo tienden a ser divididos en varios elementos

métricas de:

**Precisión** (precisión) esta métrica evalúa la fidelidad de los verdaderos positivos, mientras más cercana sea 1, indica que existe un número bajo de falsos positivos. En este caso nos referimos a elementos clasificados dentro de una región cuando en realidad no pertenece a ella)

**Exhaustividad** (recall) esta métrica evalúa qué proporción de los positivos ha sido marcada como tal. En este caso nos referimos a cuántos elementos que pertenecen a una región no fueron marcados dentro de ella.

**F-Medida** (F-MEasure) es una métrica global que integra la precisión y la exhaustividad. Entregando un puntaje entre 0 y 1. 1 es un puntaje perfecto. Finalmente el benchmark de BSD500 requiere que se utilicen varias configuración de hiperparametros para así mismo construir varias resultados en la segmentación, esto con el fin de crear una curva que representa el desempeño del método.

## 4. Resultados

El benchmark de BSD500 contiene algunos resultados generados con el método ‘ultrametric contour map’ (UCM), para demostrar el uso de la base de datos. Una vez se arreglan los paths para que coincida con la localización de los



Figure 3. Segmentación usando LAB+xy de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6.



Figure 5. Segmentación usando HSV+xy de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6. Se observa que el método no es capaz de considerar el rostro de las personas y su ropa como un único objeto

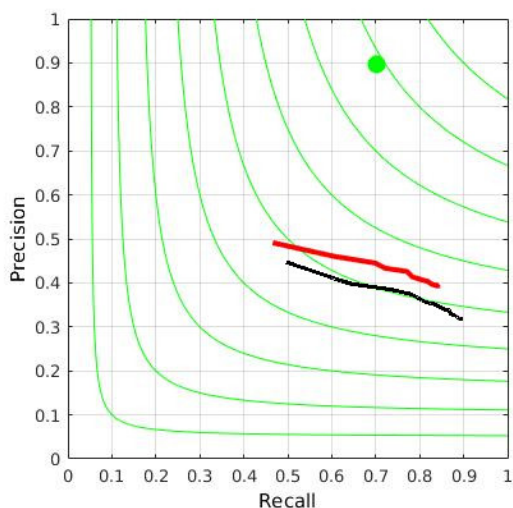


Figure 10. Curvas de precision y recall para los metodos usados en el laboratorio

directorios en el disco local se puede correr sin problema el benchmark 3 que lleva el nombre ‘ 3. boundary benchmark for results stored as a cell of segmentations’ . El resultado es el siguiente:

Los resultados obtenidos al correr el benchmark con los métodos seleccionados se presnetan en la tabla 4 y la figura 10 .

Las curvas obtenidas para los métodos propuestos en este laboratorio comparten la misma deficiencia: no pueden obtener en ningún caso la precisión de 1.0 ni el exhaustividad de 1.0. Esto indica que en ninguna de las combinaciones probadas dentro de este laboratorio (espacio de color y K) se logró evitar falsos positivos o falsos negativos.

#### 4.1. Análisis de resultados

Existe una clara diferencia en desempeño entre los algoritmos creados para el laboratorio y el UCM. La diferencia se puede explicar en gran medida en que los algoritmos utilizan únicamente la información local de color del píxel y la localización absoluta en la imagen. Este tipo de información permite únicamente aproximarse a los verdaderos contornos y tiene varias condiciones de falla entre otras:

- Si dos regiones del mismo objeto tienen colores distintos casi siempre son etiquetadas como dos objetos distintos. (ejemplo el rostro y la ropa de una persona, ver figura 5)
- Objetos cuyas texturas tienen patrones con baja frecuencia espacial, tienden a ser divididos según en los patrones que lo conforman (ejemplo las rayas del tigre, ver figura 6)
- Cambios fuertes de iluminación en un objeto de un solo color son clasificados como objetos distintos (ejemplo el vidrio del helicóptero, ver figura 7)



Figure 6. Segmentación usando HSV+xy de izquierda a derecha: original, Kmeans, GMM, Cluster jerárquico, y watershed.  $k=6$ . Se observa que el método clasifica las rayas del tigre y su sombra como objetos aparte



Figure 7. Segmentación usando HSV+xy de izquierda a derecha: original, Kmeans, GMM, Cluster jerárquico, y watershed.  $k=6$ . Cambios abruptos de iluminación (como en el vidrio del helicóptero) genera objetos inexistentes

- Tiende a sobre segmentar objetos de gran tamaño cuya textura tiene una alta frecuencia espacial (ver imagen de la mujer con arco 9)
- Tiende a dividir los objetos grandes de color aproximadamente constante, (por ejemplo el cielo, ver figura 8)

Todas estas son fallas muy bien conocidas en los algoritmos de segmentación y, en particular, son fallas que se tuvieron en cuenta al momento de diseñar el algoritmo de UCM. En la medida que la mayoría de las condiciones de falla están relacionadas con la asignación de un elemento en el cluster equivocado. La precisión del UCM está bastante por encima de lo obtenido por los algoritmos seleccionados.

## 4.2. Posibles mejoras sobre el método seleccionado

En general es muy difícil mejorar el método propuesto sin incluir información extra sobre las regiones que se espera segmentar. En particular información de la continuidad de las regiones (magnitud de gradiente o estimación de bordes) podría ser incorporada dentro del esquema de clustering a fin de evitar la división y unión incorrecta de regiones.

Una segunda falencia del método es que utiliza únicamente la información de local del pixel, en el escenario controlado de BSB esto implica que la información regional del pixel (textura, bordes, esquinas, etc.) es ignorada completamente. En un caso no controlado implica que el ruido de captura y posibles oclusiones van a afectar de manera más severa los resultados.

Finalmente, y para poder superar el desempeño de UCM, sería necesario incorporar información semántica dentro del método, esto permitiría, por ejemplo, que el rostro y la ropa de una persona se agrupen dentro de un mismo objeto a pesar de no compartir ninguna característica visual. Esto im-

plica que el clustering es solo uno de los pasos dentro de un esquema de clasificación donde se modelan explícitamente las categorías semánticas presentes en la imagen.

## 4.3. Aplicación en la práctica

En un escenario real en el que se requiera aplicar un método de segmentación, no hay razón para preferir alguno de los métodos propuestos por encima del UCM. Existe una muy importante diferencia en el desempeño (especialmente en cuanto a precisión) de los métodos propuestos y UCM. El único caso en el que podría resultar útil el método propuesto es si se requiere una aproximación rápida los posibles objetos de la escena, por ejemplo en la inicialización de un algoritmo de sustracción de fondo.

## References

- [1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2294–2301. IEEE, 2009.
- [2] A. Z. Arifin and A. Asano. Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recognition Letters*, 27(13):1515–1521, 2006.
- [3] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.
- [4] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [5] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [6] L. Shafarenko, M. Petrou, and J. Kittler. Automatic watershed segmentation of randomly textured color images. *Image Processing, IEEE Transactions on*, 6(11):1530–1544, 1997.

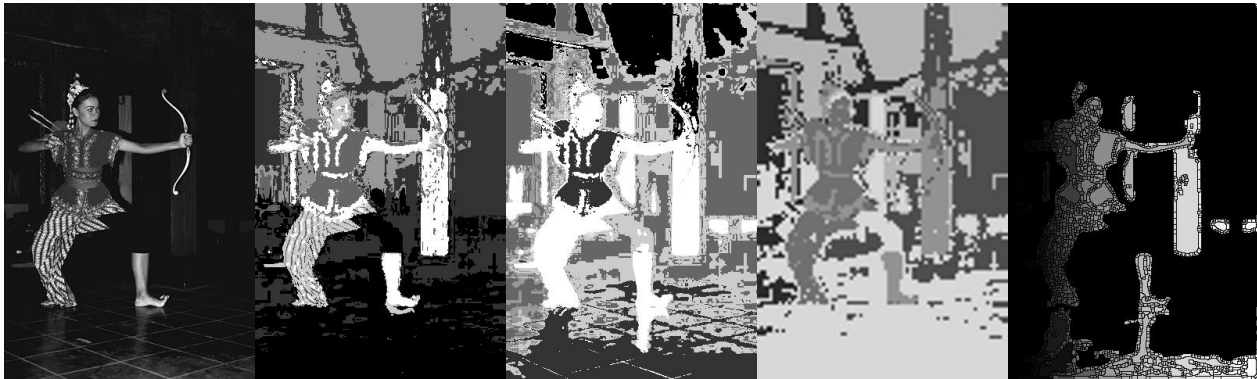


Figure 9. Segmentación usando HSV+xy de izquierda a derecha: original,Kmeans,GMM,Cluster jerárquico, y watershed. k=6. Objetos grandes con texturas complejas tienden a ser sobre segmentados

Metodo	ODS	OID	AreaPR
LAB+xy GMM	$F(0.78, 0.38) = 0.51$	$F(0.80, 0.45) = 0.58$	0.27
HSB+xy GMM	7	8	9
UCM	7	8	9