

Computer Vision Lab 05 -Segmentations BSD500

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

Abstract

1. 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

Gaussian Mixture Model Construye un conjunto de K clusters donde se tiene un ajuste óptimo de los elementos a un conjunto de K gaussianas (clusters) parametrizadas con media y matriz de covarianza . Para resolver algunos problemas de convergencia en la implementación de MATLAB se utilizó un regularizador con valor de 0.0001

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. Para la implementación de MATLAB se utilizó una distancia euclidiana normalizada por la desviación estándar.

Watershed Debido a que el metodo base produce una sobre segmentación de las imágenes en la base de datos. Se agregó una paso de inicialización donde se trata de estimar las regiones 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 para expandirlos. Finalmente el inverso de la imagen fue utilizado como máscara para establecer las regiones donde el algoritmo de watershed puede crear clusters.

1.1. 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

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

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

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

1.2. Selección de los métodos

Existen 24 combinaciones iniciales de parámetros si 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 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, con una desviación estándar de 17.47. Esto indica que el 95% de las imágenes tiene entre 0 y 55 anotaciones.

Este análisis inicial indica que 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 imagen que tiene una gran cantidad de clusters (hasta 95 anotaciones distintas) la mayoría de estas anotaciones cubren una región muy pequeña de 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 (No existe una razón formal para elegir el umbral del 2%, sin embargo, una clúster que agrupe 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)

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 reducción que consiste en escoger al azar un sub-set 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 más adecuada.

A continuación presentamos algunas de las imágenes

A partir de la inspección visual se seleccionaron 2 métodos

1. Espacio de color LAB+xy y cluster por modelo gaussiano mixto

2. Espacio de color HSV+xy y cluster por modelo gaussiano mixto

2. Dataset y Metodología de pruebas

La base de datos berkeley BSD500 contiene 503 imágenes separadas en tres grupos: training: (201) test (201) y validación (101) todas las imágenes tienen el mismo tamaño 481x321 píxeles, tiene 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).

2.1. Verdad de terreno BSD500

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

2.2. 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 métricas de:

Precisión (precision) 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 que proporción de los positivos ha sido arcaada como tal. En este



Figure 1. Ecaption



Figure 2. Ecaption

caso nos referimos a cuantos elementos que pertenecen a una region 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 varios resultados en la segmentación, esto con el fin de crear una curva que representa el desempeño del método.

3. 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 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 son los siguientes

4. Analisis de reusltados

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 tiene claras 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)
- Objetos cuyas texturas tienen patrones con baja frecuencia espacial, es decir están compuestas por elementos gran tamaño, tienden a ser divididos según en los patrones que lo conforman (ejemplo las rayas del tigre)
- Cambios fuertes de iluminación en un objeto de un solo color son clasificados como objetos distintos (ejemplo el vidrio del helicóptero)
- Tiende a sobre segmentar objetos de gran tamaño cuya textura tiene una alta frecuencia espacial, es decir está compuesta por pequeños elementos que se repiten, (ejemplo imagen de la mujer con arco)
- Tiende a dividir los elementos grandes que tienen color constante, (por ejemplo el cielo en las imágenes)

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 el algoritmo seleccionado.

4.1. 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 obtener, en particular información en cuanto a 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 y la. Una segunda falencia del método es que utiliza únicamente la información de local del píxel, en el escenario controlado de BSB esto implica que la información regional del píxel (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 implica 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.

4.2. Aplicacion 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.