



Pontificia Universidad
JAVERIANA
Bogotá

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Visión Artificial
Taller 3: Segmentación de imágenes, 2022-30

Objetivo

Identificar el nivel de apropiación de concepto logrado por el estudiante con respecto al proceso de segmentación de imágenes basado en regiones. En particular, se busca que el estudiante demuestre el manejo técnico de una librería (OpenCV o ITK) para proponer alternativas configurables de modelos para segmentar imágenes.

Desarrollo del taller

Ya que se conocen los conceptos básicos para la segmentación de imágenes utilizando el método de crecimiento de regiones, es hora de aplicarlos en el desarrollo de un programa utilizando los conocimientos técnicos básicos adquiridos con el manejo de las librerías ITK y/o OpenCV.

De esta forma, se requiere un programa que, a partir de una imagen y una semilla (posición dentro de la imagen), genere una segmentación aplicando el método de crecimiento de regiones. Este método debe implementarse desde cero, es decir, el objetivo no es utilizar alguna función disponible en las librerías para realizar el crecimiento de regiones. Para lograr este objetivo, el programa:

- Debe realizar la lectura de la imagen, al menos en escala de grises. Se darán puntos adicionales por proveer la opción adicional de segmentar imágenes a color, en cuyo caso la información debe compararse simultáneamente en los tres canales.
- Debe proveer la funcionalidad para indicar la semilla, al menos de forma manual. Se darán puntos adicionales por:
 - Proveer la opción adicional para generar la semilla de forma automática (puede ser con una posición aleatoria dentro de la imagen, o usando algún criterio sobre el valor de intensidad).
 - Proveer la opción para indicar varias semillas, manual o automáticamente. El proceso de crecimiento de regiones debe llevarse “en paralelo” desde cada semilla. Es decir, no vale llamar el proceso varias veces, con una semilla diferente cada vez, sino que todas las semillas hacen parte del frente de crecimiento de forma simultánea.
- Debe proveer una función para evaluar el criterio de homogeneidad (medida de similaridad) en el vecindario de un pixel dado. Para identificar si un pixel vecino se agrega

al frente activo de la región en crecimiento, se compara la medida en su vecindario con la medida en el vecindario del pixel ya etiquetado, y la diferencia entre las dos debe ser menor que un límite de tolerancia dado. Esto implica:

- Definir el vecindario a evaluar, al menos los 4-vecinos (superior, inferior, izquierdo, derecho). Se darán puntos extra por proveer la opción adicional de evaluarlo sobre los 8-vecinos (4-vecinos más superior-izquierdo, superior-derecho, inferior-izquierdo, inferior-derecho).
 - Calcular la medida en el vecindario del pixel, al menos el promedio del valor de intensidad de los vecinos. Se darán puntos extra por proveer opciones adicionales para calcular otras medidas en el vecindario, como la varianza (o desviación estándar) o el contraste (diferencia entre el mayor y el menor valor de intensidad).
 - Calcular la diferencia entre la medida en el pixel de la región y la calculada en el pixel vecino. Esta diferencia debe ser menor a un límite de tolerancia dado por el usuario para que el pixel pueda agregarse a la región en crecimiento.
- Debe realizar el almacenamiento en disco de la imagen segmentada, al menos de la imagen etiquetada (o binarizada: fondo negro, objeto blanco). Se darán puntos adicionales por generar además la imagen segmentada (o umbralizada: fondo negro, objeto con sus intensidades originales). A manera de ejemplo, a continuación se muestra la imagen original, y los dos posibles resultados esperados: la imagen etiquetada (binarizada) y la imagen segmentada (umbralizada).



Imagen original



Imagen etiquetada

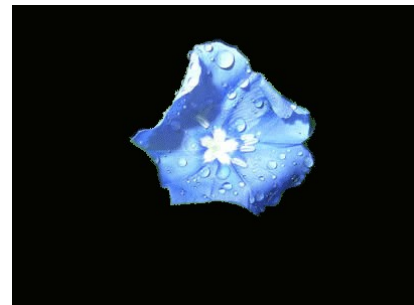


Imagen segmentada

Evaluación

La entrega del taller consistirá en un único archivo comprimido (único formato aceptado: .zip) que contendrá el ejercicio de programación propuesto. Debe incluir el código fuente (en ITK o en OpenCV, correcto y funcional), el archivo `CMakeLists.txt` (correcto y funcional) que permite compilar el programa (si se utiliza C++), y las imágenes de ejemplo utilizadas. El código fuente debe compilar con gcc (g++) o interpretarse con python, con las versiones más recientes de las librerías de Insight Toolkit (ITK) u OpenCV.

Este archivo deberá enviarse a través de la correspondiente asignación en BrightSpace antes de las 2:00p.m. del martes 1 de noviembre de 2022. El envío del archivo comprimido en otro formato diferente a los especificados resultará en una calificación de (0.0/5.0) para el taller.

A nivel informativo y de resumen, a continuación se especifica la funcionalidad mínima esperada y los puntos adicionales propuestos:

- Funcionalidad mínima:
 - Imagen de entrada en escala de grises.
 - Indicación manual de una sola semilla.
 - Promedio de las intensidades como medida de similaridad.
 - Vecindario de 4-vecinos.
 - Límite de tolerancia de la diferencia dado por el usuario.
 - Resultado como imagen etiquetada (binarizada).
- Puntos adicionales:
 - Imagen de entrada en color.
 - Una semilla seleccionada automáticamente.
 - Múltiples semillas indicadas manualmente.
 - Múltiples semillas seleccionadas automáticamente.
 - Varianza (o desviación estándar) de las intensidades como medida de similaridad.
 - Contraste entre las intensidades como medida de similaridad.
 - Vecindario de 8-vecinos.
 - Resultado como imagen segmentada (umbralizada).

La escala de evaluación es la siguiente:

- **Excelente (5.0/5.0):** El estudiante propone un código que provee la funcionalidad completa del crecimiento de regiones (hasta 8 de los puntos adicionales propuestos).
- **Bueno (4.0/5.0):** El estudiante propone un código que provee la funcionalidad estándar del crecimiento de regiones (hasta 4 de los puntos adicionales propuestos).
- **Aceptable (3.0/5.0):** El estudiante propone un código que provee la funcionalidad mínima del crecimiento de regiones.
- **Regular (2.0/5.0):** El estudiante propone un código que solo realiza algunos de los pasos del método de crecimiento de regiones, generando un resultado incompleto o incorrecto.
- **Malo (1.0/5.0):** El código propuesto por el estudiante no compila.
- **No entregó (0.0/5.0):** El estudiante no entrega los archivos solicitados.