

Procesado de imagen y visión por computador

Curso 2018/2019

Segmentación de imágenes

0. Introducción

La segmentación en las imágenes pretende extraer los objetos de las mismas para simplificar el contenido pero perdiendo la menos información posible. Cada técnica utiliza alguna característica dentro de la imagen para realizar esta extracción. Hay diversas técnicas de segmentación entre la que podemos destacar:

- **Detección de bordes.** Mediante esta técnica se definen los bordes de los objetos o formas dentro de la imagen para el posterior análisis. Normalmente tienen una fase de filtrado u obtención del gradiente y un posterior marcado de los píxeles, normalmente mediante umbralizado. Entre las técnicas más destacadas podemos nombrar la de Canny (prácticamente un estándar) u otras como las de Sobel que utiliza máscaras de convolución.
- **Umbralizado.** Sobre imágenes en escala de grises o en la componente de iluminación de una imagen en color se puede fijar un umbral que permita extraer los objetos buscados. Ese umbral puede ser fijo o adaptarse a los cambios de iluminación mediante técnicas como la de Otsu.
- **Segmentación en color.** El color es una fuente de información importante dentro de las imágenes y muchas técnicas lo utilizan para extraer objetos de un determinado color. Las técnicas para la segmentación en color pueden ir desde el más sencillo umbralizado a técnicas más avanzadas basadas en clustering, redes neuronales o máquinas de vectores soporte.

Esta sesión pretende ser eminentemente práctica buscando por un lado el seguir practicando con la librería OpenCV y sus comandos y por otro manejar algunos de los algoritmos más comunes en segmentación de imagen. El fichero *Presentación_Segmentación.pdf* incluye algunas de las técnicas empleadas y puede usarse como referencia.

1. Detección de bordes

En este apartado vamos a probar el método de detección de bordes de Canny. Para ello dispone del programa *Apartado1.py* que realiza un ejemplo de aplicación del mismo. Pruebe el programa y observe como funciona sobre la imagen *Senal.jpg*.

Añada dos barras de desplazamiento que permitan ajustar los umbrales del algoritmo de Canny para obtener los mejores resultados.

2. Umbralización

En este caso queremos probar la umbralización directa de imágenes en escala de grises y comprobar los problemas que suponen los cambios de iluminación. En el fichero *Apartado2.py* hay un ejemplo de umbralización sobre una imagen fija que puede tomarse como base (Use la imagen *Fotograma.png*). Se pide:

1. **Realizar** un programa de permita umbralizar un vídeo. Las pruebas se realizarán sobre el vídeo *movil.avi*.

Procesado de imagen y visión por computador

Curso 2018/2019

Segmentación de imágenes

2. **Modificar** el programa anterior para que el umbral se adapte a los cambios de iluminación. Para ello se utilizará la técnica de Otsu¹ que ya viene implementada en la función *threshold*. Ejecutar el programa obtenido sobre el vídeo *movil.avi* y obtener un vídeo de resultado así como el umbral fijado por la función *threshold* para cada fotograma.

3. Segmentación en color

Finalmente utilizaremos la información de color para segmentar. En el fichero *Apartado3.py* tiene un ejemplo de como podría funcionar. En este caso usando el espacio de color RGB. Se pide:

1. **Ejecutarlo** sobre las imágenes *balls.png* y *pokemon.png*. Probar su funcionamiento hasta entender cómo funciona.
2. **Añadir** el código necesario para cambiar el espacio de color de RGB a HSV² y realizar la segmentación en ese espacio. Tenga en cuenta que OpenCV usa los siguientes rangos:

H: 0 – 180

S: 0 – 255

V: 0 – 255

Pruebe la modificación introducida y observe si los resultados mejoran sobre RGB o no.

¹ http://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0

² http://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html#gsc.tab=0