

Herramientas Software para Tratamiento de Imágenes

Máster Oficial en Visión Artificial

Hoja de Problemas evaluable Octave/Matlab

Instrucciones para la entrega

Una vez resueltos los problemas enunciados a continuación, el alumno utilizará el formulario habilitado en el moodle de la asignatura para subir los fuentes en un fichero zip. El nombre del fichero deberá formarse siguiendo el siguiente esquema: Apellido1_Apellido2_Nombre.zip”.

Nota: la hoja de problemas irá creciendo a medida que avancemos temario en el Bloque I. Se entiende que no está completa todavía.

Fecha límite de entrega: -----

Enunciados

1. Javier se ha comprado un disco NAS para que su familia pueda subir todas las fotos que van haciendo de su hija Ana con sus móviles. Javier quiere tener todas las fotos posibles de su hija, pero no con cualquier calidad. Le gustaría poder eliminar todas aquellas fotos que estén desenfocadas de su biblioteca. Nuestro objetivo en este primer problema es ayudar a Javier a automatizar esta tarea.

Lo primero que haremos será generar una colección de fotos enfocadas y desenfocadas que nos permitan probar el método que vamos a desarrollar. Para ello, en este primer problema se pide crear una función (`blurrer`) en Matlab que reciba como argumentos un directorio de entrada y un directorio de salida. El directorio de entrada deberá contener una colección de imágenes inicial. Tras llamar a nuestra función, el directorio de salida contendrá esas mismas imágenes pero desenfocadas. Pasos a seguir por desenfocar cada imagen:

1. leer imagen
2. suavizar la imagen haciendo uso de la la función `imsmooth` utilizando diferentes tamaños de kernel 5x5, 15x15, 21x21
3. cada operación de suavizado deberá generar una imagen en el directorio de salida

Una vez generada la colección de prueba podemos desarrollar nuestro detector de imágenes desenfocadas. Para ello crearemos una segunda función en Matlab que llamaremos `blurDetection`, y recibirá como argumento de entrada un directorio. Esta función irá presentando en una ventana las imágenes que contiene el directorio (una por una). Sobre la imagen deberá aparecer un texto, indicando si la imagen está o no desenfocada y que “cantidad” de desenfoque presenta (`figure, text, waitforbuttonpress`). Pero...¿cómo podemos cuantificar el desenfoque de una foto?.

Una imagen desenfocada presenta una bajo porcentaje de altas frecuencias. Existen diferentes métodos en la literatura que permiten medir el nivel de desenfoque de una imagen. Nosotros utilizaremos un método basado en la varianza de la Laplaciana. Para ello seguiremos los siguientes pasos:

1. Convolucionar la imagen con un kernel laplaciano (`fspecial`)
2. Calcular la varianza de la imagen resultante
 1. La varianza cuantifica el nivel de desenfoque de la imagen (menor varianza implica mayor desenfoque)
 2. En principio será el usuario el encargado de determinar utilizando este valor qué es para él una imagen enfocada y qué es para él una imagen desenfocada. Por defecto fijaremos el umbral en 90.

2. Desarrollar un sistema de visión artificial que permita hacer seguimiento visual por color de un objeto. El sistema debe ser capaz de detectar el objeto (en el caso de que se encuentre presente) en cada fotograma. El objeto puede moverse libremente, pudiendo cambiar su escala y la perspectiva con la que se ve el mismo. Para ello utilizaremos algunas de las rutinas del paquete de procesamiento de imagen de Octave.

Lo primero que necesitamos es ser capaces de segmentar el objeto a seguir en cada fotograma. Para ello convertiremos cada fotograma al espacio de color HSV (`rgb2hsv`). Dado el rango de color en el que se mueve nuestro objeto, segmentaremos el fotograma generando una imagen binaria.

Para determinar el rango de color en el que se mueve nuestro objeto desarrollaremos una pequeña aplicación octave que cargue un fotograma del vídeo y muestre el resultado de la segmentación para ese fotograma utilizando unos rangos por defecto [`hmin..hmax`], [`smin..smax`], [`vmin..vmax`]. El usuario podrá aumentar o disminuir dichos valores de forma interactiva (con eventos de teclado: `waitforbuttonpress`, `kbhit`, `CurrentCharacter`), y ver el resultado de la segmentación para el fotograma cargado. Además, se podrá ver el efecto de la segmentación en el resto de fotogramas. Si el usuario pulsa la tecla 'n' (de next), automáticamente se cargará el siguiente fotograma y podrá verse el resultado de la segmentación en el mismo. La función que permite calcular el rango de valores HSV en el que se mueve nuestro objeto se debe llamar `rangecolorselector`.

Una vez determinado el rango HSV de nuestro objeto, definiremos una función llamada `color_tracking` que se llevará el peso del seguimiento. Por cada fotograma del vídeo se llevarán a cabo las siguientes operaciones:

1. Aplicar un filtro gaussiano al fotograma para eliminar ruido (`imsmooth`)
2. Convertir el fotograma al espacio de color HSV (`rgb2hsv`)
3. Segmentar el fotograma utilizando el rango de color seleccionado
4. Eliminar pequeños focos de ruido en la imagen resultante:
 1. Aplicar dos operaciones de erosión consecutivas (`strel`, `imerode`), utilizando un elemento estructurante cuadrado de tamaño 3x3.
 2. Aplicar dos operaciones de dilatación consecutivas, (`strel`, `imdilate`), utilizando un elemento estructurante cuadrado de tamaño 3x3.
5. Detectar los distintos contornos que aparecen en la imagen y quedarse con aquel que presente mayor área (`regionprops`).
6. Presentar en pantalla el la imagen con el contorno sobreimpreso en color verde. Pintar también el centroide del contorno en color rojo.

7. Esperar 33ms (`pause`) y continuar con el siguiente fotograma.

NOTA: adjuntar junto con el código de esta práctica un fichero `readme.txt` donde se explique brevemente los pasos a seguir para ejecutar las dos funciones desarrolladas. El vídeo/secuencia utilizado no deberá adjuntarse con la práctica. El alumno deberá incluir en el fichero `readme.txt` un enlace a dropbox/drive (o cualquier otro sistema) donde el profesor pueda descargarlo. El enlace deberá estar activo hasta el día en el que se publique la nota de esta hoja de problemas.