Universidad de Murcia

Facultad de Informática

— Visión Artificial —

Documentación Ejercicios

2016/2017

Autor
José María Sánchez Salas
josemaria.sanchez12@um.es

Profesor
Alberto Ruíz García
aruij@um.es

Índice

1 Introducción

Este documento contiene la documentación de los ejercicios propuestos, así como de algunos expermientos realizados, de la asignatura Visión Artificial perteneciente a la mención de Computación del Grado de Ingeniería en Informática, impartido en la Facultad de Informática de la Universidad de Murcia en el curso académico 2016-2017.

La estructura de este documento es la siguiente: la primera sección contiene las explicaciones de los ejercicios propuestos, mientras que la segunda sección contiene las explicaciones de los experimentos que realizados, haciendo uso de lo aprendido en los ejercicios.

Es importante destacar, que **para todos los ejercicios**, para cerrar el programa hay que pulsar la tecla Escape (`Esc`), por eso mismo, en la resolución de los ejercicios, si es necesaria la pulsación de teclas, no se va a exponer el uso de esta tecla.

2 Ejercicios Propuestos

2.1 Ejercicio 1

Enunciado

Modificación de los canales de color (brillo, saturación, etc.) sobre la secuencia de imágenes tomada con la webcam.

Resolución

Este ejercicio ha sido resuelto de la siguiente manera: una vez capturada la imagen de la cámara, se hace una copia para evitar modificar la imagen obtenida, se cambia el espacio de color a HLS, se modifican los canales L y S, se convierte al espacio de color BGR y se muestra la imagen modificada.

2.2 Ejercicio 2

Enunciado

Amplía `player.py` para seleccionar con el ratón una región de interés (ROI) y almacenar en una lista imágenes, que pueden opcionalmente guardarse en disco. Este programa nos servirá como base para futuros ejercicios.

Resolución

Lo más importante de este ejercicio es cómo seleccionar una región de interés de la webcam. Para ello, hay que introducir un manejador de eventos de ratón; este manejador hace lo siguiente: cuando se pulsa el botón izquierdo, almacena las coordenadas del ratón como coordenadas iniciales del ROI, además se indica que se inicia el dibujo del ROI; mientras no se levante el botón, si se mueve el ratón, se van almacenando como coordenadas finales del ROI las coordenadas donde está el ratón, esto se hace así, para poder ir dibujando el ROI conforme se va creando; una vez que se levante el botón izquierdo, se almacena definitivamente las coordenadas finales y se añade, automáticamente dicho ROI a la lista de ROIs.

El programa principal, lo único que hace es: ver si se ha pulsado alguna tecla (y si es así, hacer la acción correspondiente); si se ha iniciado el dibujo del ROI, dibujarlo; mostrar la imagen de la cámara.

Las teclas que se procesan en este programa son:

- Tecla `Enter`: Para eliminar el dibujo del ROI.
- Tecla `Espacio`: Para parar la webcam en una imagen.
- Tecla `s`: Para almacenar en disco los ROIs almacenados.

2.3 Ejercicio 3

Enunciado

Construye un clasificador de objetos en base a la similitud de los histogramas de color del ROI (de los 3 canales por separado).

Resolución

Haciendo uso del ejercicio anterior, la resolución de este ejercicio se basa simplemente en una vez seleccionado el ROI, se calcula y almacena su histograma, para comparar con el histograma del ROI de la imagen actual y si supera un determinado valor (variable global `maxDiff`), cambiar el color del ROI a azul, indicando es similar al ROI almacenado.

Las teclas que se procesan en este programa son:

- Tecla `Enter`: Para seleccionar el ROI y calcular su histograma.
- Tecla `Espacio`: Para parar la webcam en una imagen.

2.4 Ejercicio 4

Enunciado

Construye un generador que detecte frames estáticos, o frames en movimiento, a partir de una secuencia de imágenes.

Resolución

Para la resolución de este ejercicio se ha hecho uso de la librería `umucv` proporcionada por el profesor, para crear un historial de las 9 últimas imágenes y mostrarlas de manera conjunta para que se vea el paso de los movimientos. Sin embargo, el cálculo para detectar movimiento, se hace únicamente con la imagen actual y la anterior. Una vez obtenidas la imagen actual y la anterior, se hace la diferencia entre ellas y si supera un determinado valor (variable global `maxDiff`), el programa muestra por terminal el mensaje `I caught you!`, indicando que ha habido movimiento.

2.5 Ejercicio 5

Enunciado

Construye un servidor web sencillo usando `flask` que muestre una cierta transformación de las imágenes tomadas con la cámara.

Resolución

Este ejercicio proporciona las siguientes transformaciones de imágenes:

- **Suavizado Gaussiano**. La dirección para realizar esta transformación es: `http://localhost:5000/transforms/gaussianBlur/<blurringFactor>` donde `blurringFactor` es el factor de suavizado con el que realizar la transformación. Este factor debe ser numérico, si no lo es, el servidor da un error.
- **Conversión a blanco y negro**. La dirección para realizar esta transformación es: `http://localhost:5000/transforms/convertToBW`.
- **Umbralización**. La dirección para realizar esta transformación es: `http://localhost:5000/transforms/threshold/<threshold>` donde `threshold` es el umbral con el que realizar la umbralización. Este umbral debe ser numérico, si no lo es, el servidor da un error.
- **Bordes**. La dirección para realizar esta transformación es: `http://localhost:5000/transforms/edges/<thresholds>` donde `thresholds` son los umbrales necesarios para la transformación de bordes `Canny`. El formato de los umbrales es: `thresh1xthresh2`, donde `thresh1 < thresh2`. Ambos umbrales deben ser numéricos y cumplirse la condición anterior, si no es así, el servidor da un error.

2.6 Ejercicio 6

Enunciado

Implementar el efecto chroma con imágenes en vivo de la webcam. Pulsando una tecla se captura el fondo y los objetos que aparezcan se superponen en otra imagen o secuencia de video.

Resolución

La tecla para capturar el fondo es la tecla `Enter`. Tras pulsar esta tecla, el programa captura el fondo, crea el chroma y pone los objetos que aparezcan en la imagen `cube3.png`. Una característica interesante de este programa, es que proporciona un slider para ajustar el umbral con el que hacer el chroma.

Para hacer el chroma, tanto el fondo como la imagen actual se pasan al espacio de color YUV para eliminar la luminosidad y quedarnos con el color y la saturación, para así poder hacer un chroma más efectivo.

2.7 Ejercicio 7

Enunciado

Implementa la segmentación por color usando modelos de histograma en un programa que admite como argumento:

a) una carpeta con trozos de imagen que sirven como muestras de color y

b) otra imagen que deseamos clasificar.

El resultado puede ser un conjunto de máscaras para cada clase, o una "imagen de etiquetas", donde diferentes colores indican cada una de las regiones.

Resolución

La ejecución de este programa debe ser: `./exercise07.py [-h] models image`, donde:

- `[-h]` es una opción para mostrar la ayuda.
- `models` es la carpeta que contiene los modelos de color.
- `image` es la imagen que se quiere segmentar.

Tras ejecutar correctamente el programa, este cargará los modelos (como color del modelo, se obtiene el color medio de todo el modelo), leerá la imagen y segmentará la imagen. Para segmentar la imagen, lo que hace es obtener las diferencias con todos los modelos, obtener para cada diferencia, el modelo más cercano con él, segmentar la imagen. Una vez terminado, el programa mostrará la imagen original y la imagen segmentada en color. No se tratan todos los píxeles de la imagen, pues se filtran aquellos píxeles que no pertenecen a ningún modelo.

Los modelos y las imágenes para la ejecución de este ejercicio están en la carpeta `exercise07`

2.8 Ejercicio 8

Enunciado

Mostrar el efecto de diferentes filtros de imagen sobre la imagen en vivo de la webcam, seleccionando con el teclado el filtro deseado y permitiendo modificar sus posibles parámetros (p.ej. el nivel de suavizado) con las teclas de flecha.

Resolución

La realización de este ejercicio se ha hecho de la siguiente manera: puesto que todos los filtros deben de tener una tecla particular para aplicarse, todos tienen un valor inicial para poder realizar el filtro, un valor actual con el que se aplica el filtro, y para poder modificarse con las teclas de flechas, todos tienen sus valores de "paso" para modificar el valor actual del filtro; he decidido crear diccionarios con cada propiedad anterior: tecla del filtro, valor inicial, valor actual y valor de paso.

De esta manera, lo que consigo, es que en el código para la aplicación del filtro solamente trabaje con esos diccionarios y con una variable que indica el filtro a aplicar. Es decir, esté desacoplado al filtro real a aplicar. El programa principal, lo único de lo que se encarga es de procesar las teclas pulsadas para la aplicación del filtro correspondiente.

Este programa se diferencia de los anteriores en que, una vez que se aplica un filtro, para salir de él hay que pulsar la tecla `Esc`. Esto hace que se deje de aplicar el filtro y aparecerá otra vez la imagen de la cámara original, listo para aplicar otro filtro. Para salir del programa, es necesario pulsar la tecla anterior cuando se encuentre la imagen de la camara original.

2.8.1 Extensión

La extensión del ejercicio 8 consiste en permitir aplicar los filtros a un ROI seleccionado y mostrar la imagen normal y el ROI con el filtro aplicado. De esta manera, se puede comparar el efecto de los filtros con la imagen original.

2.9 Ejercicio 9

Enunciado

Reconocimiento de objetos con la webcam basado en el número de coincidencias de keypoints. Pulsando una tecla se pueden ir guardando modelos (p. ej. carátulas de CD). Cuando detectamos que la imagen está bastante quieta, o cuando se pulse otra tecla, calculamos los puntos de interés de la imagen actual y sus descriptores y vemos si hay suficientes coincidencias con los de algún modelo.

Resolución

Este ejercicio proporciona la siguiente funcionalidad con el teclado:

- Tecla `s`: para almacenar el modelo que hay en el ROI seleccionado.
- Tecla `Enter`: para procesar la imagen actual en busca de los modelos seleccionados.

El programa, conforme se van seleccionando modelos, va mostrando el último modelo almacenado. Una vez que se quiera procesar la imagen actual, entonces, procesa la imagen en busca de los modelos, si alguno supera el mínimo de coincidencias (variable `minMatches` que se puede ajustar en el slider que aparece encima de la imagen principal), entonces se muestra en otra ventana la imagen del modelo con el que ha coincidido.

2.10 Ejercicio 10

Enunciado

Reconocimiento de formas con la webcam (o sobre un conjunto de imágenes) basado en descriptores frecuenciales.

Resolución

Este programa se debe ejecutar de la siguiente manera: `./exercise10.py [-h] [-d DEV] templates labels`, donde:

- `[-h]` es la opción de ayuda.
- `[-d DEV]` es la opción para indicar otro dispositivo de webcam (por defecto es el 0) o un directorio donde se encuentran las imágenes que se quieren procesar.
- `templates` es la imagen que contiene los modelos.
- `labels` son las etiquetas asociadas a cada modelo.

Este programa permite procesar tanto imágenes tomadas de la webcam como de un conjunto de imágenes (dentro de un directorio). Para el procesar las imágenes tomadas por la webcam, es necesario que se seleccione el ROI y se pulse la tecla `Enter` para procesarlo. En cambio, para la secuencia de imágenes, se procesa cada imagen completamente. El procesamiento de cada imagen se hace por separado, pulsando la tecla `n` se permite procesar la siguiente imagen.

En ambos casos, el programa, cuando procesa la imagen, muestra por consola el resultado obtenido. Una cosa importante a tener en cuenta en este ejercicio, es que como se pueden producir falsos positivos, estos se intercalan en la salida, por lo que suelen ensuciar el resultado obtenido.

2.11 Ejercicio 11

Enunciado

Resolución

2.12 Ejercicio 12

Enunciado

Resolución

3 Experimentos

Los experimentos que se han llevado a cabo se encuentran en la carpeta `experiments`.

3.1 Reconocedor de matrículas de coche

Enunciado

Programa de reconocimiento de matrículas con la webcam.

Resolución

Haciendo uso del ejercicio 10, se ha desarrollado este programa para reconocer matrículas de coche con la webcam. Para ello, primero se ha de seleccionar un ROI de la imagen, para después pulsar la tecla `Enter` y procesarlo. Hace uso de los modelos `platetemplates.jpg` que proporcionó el profesor y que se encuentra en la carpeta `images`