

## Prácticas de Visión por Computador

### Detección de puntos relevantes y Construcción de panoramas

**FECHA DE ENTREGA: 18 Noviembre**

**Valoración total: (8+4) puntos**

TRABAJO de IMPLEMENTACIÓN :

**1.- (4 puntos) Extracción de regiones relevantes en un espacio de escalas.**

Este punto se centra en detectar KeyPoints sobre cada una de las imágenes de Yosemite.rar (usar versiones en rango de gris) y dibujarlos sobre las imágenes haciendo uso de la función drawKeyPoints(). Para ello, se ha de construir un Espacio de Escalas (pirámide) de Lowe con cuatro octavas en total y tres escalas dentro de cada octava. Suponer que la imagen original está afectada por un alisamiento debido a la captura de  $\sigma = 0.8$  y queremos introducir una primera octava de índice-0 de manera que la imagen semilla ( $v_0^1$ ) en el espacio de escalas tenga un  $\sigma=1.6$ .

- a) ¿Qué operaciones sobre la imagen original de  $\sigma = 0.8$  nos permite fijar una imagen semilla de  $\sigma = 1.6$  ?
- b) Implementar una función que calcule las escalas de cualquier octava de la forma más eficiente posible. Es decir, reusable para cualquier escala.
- c) Usar la función implementada en (b) para calcular las escalas de todas las octavas (0-3). Mostrar las imágenes calculadas agrupadas por octava (mostrar solo las escalas 1,2 y 3 en cada octava).
- d) Calcular el espacio de escalas Laplaciano a partir de la pirámide e identificar los 100 extremos locales con mayor respuesta, presentes en la misma. Extraer en un vector  $(x,y,\sigma)$  para cada uno de ellos en los ejes de la imagen original.
- e) Mostrar la imagen con los extremos locales extraídos usando para ello un círculo de radio  $6x$  sobre la escala de detección  $\sigma$ .
- f) Bonus.- (1punto) Quién realice este punto usando código propio de convolución y submuestreo (pirámide de Lowe).

**2.- (1.5 puntos) (Correspondencias)**

Con cada dos de las imágenes de Yosemite con solapamiento detectar y extraer los descriptores SIFT de OpenCV, usando para ello la función cv2.detectAndCompute() . Establecer los puntos en correspondencias existentes entre las dos imágenes usando el objeto BFMatcher de OpenCV y los criterios de correspondencias “BruteForce+crossCheck y “Lowe-Average-2NN”. Mostrar ambas imágenes en un mismo canvas y pintar líneas de diferentes colores entre las coordenadas de los puntos en correspondencias. Mostrar en cada caso un máximo de 100 elegidas aleatoriamente.

- a) Valorar la calidad de los resultados obtenidos a partir de un par de ejemplos aleatorios de 100 correspondencias. Hacerlo en términos de las correspondencias válidas observadas por inspección ocular y las tendencias de las líneas dibujadas. ¿concluiría que ambos métodos son equivalentes u observa uno mejor que otro?

3.- (2.5 puntos) (**Construcción de un panorama rectangular**) Usar las imágenes del fichero mosaico.rar. Escribir una función que dadas 3 imágenes con solapamiento 2-a-2, a) extraiga los KeyPoints SIFT y establezca un conjunto de puntos en correspondencias entre cada dos imágenes solapadas, c) estime la homografía entre las imágenes a partir de dichas correspondencias y d) muestre un Mosaico bien registrado a partir de dichas imágenes. Estimar las homografías usando la función `cv2.findHomography()` con RANSAC.

(Ayuda: para construir el mosaico será necesario a) definir la imagen en la que pintaremos el mosaico, b) definir la homografía que lleva cada una de las imágenes a la imagen del mosaico usando la función `cv2.warpPerspective()` con el flag `BORDER_TRANSPARENT`).

## **BONUS**

En general los puntos de Bonus solo se tendrán en cuenta si se ha alcanzado en la parte obligatoria el 75% o más de los puntos

B1.- (1.5 puntos) Implementar el refinamiento de localización de puntos extremos en el espacio de escalas Laplaciano. Aplicarlo a los puntos encontrados en el punto.1 y mostrar que la implementación realmente mejora la estimación inicial usando imágenes de los entornos de los puntos.

B2.- (solo una opción) En todos los casos explicar los pasos realizados.

Opción (A) (1.5 puntos) (Seguimiento punto.3) construir un panorama con proyección plana con todas las imágenes del fichero.

Opción (B) (2.5 puntos) Igual que (a) pero añadiendo “Blending” de colores.

Opción (C) (3.5 puntos) construir un panorama con proyección esférica y “Blending” de colores con todas las imágenes del fichero.

**En PRADO se encuentran todas las imágenes citadas.**

**No está permitido usar los recursos del módulo stitching de OpenCV ni otro software que implemente la construcción de Panoramas.**

## **Informe a presentar**

Para este trabajo como para los demás proyectos debe presentar un informe escrito con sus valoraciones y decisiones adoptadas en cada uno de los apartados

de la implementación. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. ( hacer en pdf o en cuaderno colab)

**Normas de la entrega de Prácticas:** EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PERDIDA DIRECTA DE 1 PUNTO CADA VEZ QUE SE DETECTE UN INCUMPLIMIENTO.

1. El código se debe estructurar en funciones, una por cada apartado de la práctica.
2. El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
3. Todos los ficheros juntos se podrán dentro de un fichero zip..
4. SOLO ENTREGAR EL CODIGO FUENTE. NO incluir las “imágenes” dadas.
5. Los path que se usen en la lectura de imágenes o cualquier fichero de entrada debe ser siempre “imagenes/nombre\_fichero”
6. Todos los resultados numéricos serán mostrados por pantalla. No escribir nada en el disco.
7. La práctica deberá poder ser ejecutada de principio a fin sin necesidad de ninguna selección de opciones. Para ellos fijar los parámetros por defecto que se consideren óptimos.
8. Poner puntos de parada para mostrar imágenes o datos por consola

**Forma de entrega:** Subir el zip a PRADO.