

Trabajos de Visión por Computador

TRABAJO-2

Detección de puntos relevantes y Construcción de panoramas

FECHA DE ENTREGA: 21 de noviembre

Valoración total: 10/6

TRABAJO de IMPLEMENTACIÓN :

1.- Detección de puntos Harris multiescala (MOPS, Brown, Szeliski, Winder, 2004). En este apartado se trata de implementar el detector de regiones descrito en el artículo mencionado. En este algoritmo se suponen que todas las imágenes están tomadas en una escala semejante (zoom y distancia iguales) por lo que el objetivo es extraer todas las regiones relevantes que haya a distintas escalas enteras de las imágenes. Por cada región detectada necesitaremos guardar la siguiente información: (las coordenadas x, y de su centro, su orientación, su escala)). Usar para ello la estructura que resulte más fácil. Presentar los resultados con las imágenes Yosemite.rar.

- a) (OPCION: 10 puntos) Escribir una función que extraiga la lista potencial de puntos Harris en una imagen a distintas escalas. Para ello construiremos una pirámide Gaussiana de la imagen con 3 escalas usando $\sigma=1$. Sobre cada nivel de la pirámide usar la función OpenCV `cornerEigenValsAndVecs` para extraer los mapas de auto-valores de la matriz Harris en cada píxel ((fijar los valores de `blockSize` en el rango [3-13] y `ksize` en el rango [3-9]), usar la versión de nivel de gris de las imágenes). Crear una matriz con el valor del criterio selección Harris asociado a cada píxel usando ($k=0.04$). Implementar la fase de supresión de valores non-máximos sobre dicha matriz (ver descripción al final). Ordenar de mayor-a-menor(ver `sortIdx()`) en cada escala los puntos resultantes de acuerdo a su valor Harris. Seleccionar al menos 1500 puntos de entre los de mayor valor distribuidos entre las distintas escalas (las escalas más bajas tendrán más puntos: 70-20-10). Mostrar el resultado dibujando un círculo sobre la imagen original de radio proporcional a la escala y centro las coordenadas de los puntos (ver `circle()`). (3 puntos)
- b) Extraer los valores (x, y , escala) de los puntos resultantes en el apartado anterior en un vector y refinar su posición espacial a nivel sub-píxel usando la función OpenCV usando `cornerSubPix()` con la imagen del nivel de pirámide correspondiente. Actualizar los datos (x, y , escala) de cada uno de los puntos encontrados. (0.5 puntos)
- c) Calcular la orientación relevante de cada punto Harris, usando un alisamiento fuerte de las derivadas en x e y de las imágenes, en la escala correspondiente, como propone el paper MOPS de Brown&Szeliski&Winder. (Apartado 2.5) y añadir la información del ángulo al vector de información del punto. Pintar sobre la imagen de

círculos anterior un radio en cada círculo indicando la orientación estimada en ese punto (1 punto)

(OPCION: 6 puntos) Escribir una función que extraiga la lista potencial de puntos Harris en una imagen a distintas escalas. Para ello construiremos una pirámide Gaussiana de la imagen con 4 escalas usando $\sigma=1$. Sobre cada nivel de la pirámide usar la función OpenCV `cornerHarris` que devuelve el mapa de valores del criterio Harris en cada píxel. Seleccionar al menos los 1500 puntos de entre los de mayor valor distribuidos entre las distintas escalas (las escalas más bajas tendrán más puntos: 70-20-10). Mostrar el resultado identificando con un círculo o una cruz sobre la imagen original las coordenadas de los puntos seleccionados (ver `circle()`). (1.5 puntos)

- a. Calcular la orientación relevante de cada punto Harris, usando un alisamiento fuerte de las derivadas en x e y de las imágenes, en la escala correspondiente, como propone el paper MOPS de Brown&Szeliski&Winder. (Apartado 2.5) y añadir la información del ángulo al vector de información del punto. Pintar sobre la imagen de círculos anterior un radio en cada círculo indicando la orientación estimada en ese punto (1 punto)

2.- Usar los detectores OpenCV (KAZE/AKAZE) sobre las imágenes de Yosemite.rar. Extraer extraer sus listas de keyPoints y establecer las correspondencias existentes entre ellas (Ayuda: usar la clase `DescriptorMatcher` y `BFMatcher` de OpenCV). Valorar la calidad de los resultados obtenidos bajo el criterio de correspondencias OpenCV “`BruteForce+crossCheck`”. (1.5 puntos)

3.- (OPCION: 10 puntos) Escribir una función que forme un Mosaico de calidad a partir de $N > 3$ imágenes relacionadas por homografías, sus listas de keyPoints calculados de acuerdo al punto anterior y las correspondencias encontradas entre dichas listas. Estimar las homografías entre ellas usando la función `findHomography(p1,p2,CV_RANSAC,1)`. Para el mosaico será necesario. a) definir una imagen en la que pintaremos el mosaico; b) definir la homografía que lleva cada una de las imágenes a la imagen del mosaico; c) usar la función `warpPerspective()` para trasladar cada imagen al mosaico (ayuda: mirar el flag `BORDER_TRANSPARENT` de `warpPerspective` para comenzar). (4 puntos)

(OPCION: 6 puntos) Escribir una función que forme un Mosaico de imágenes a partir de dos imágenes relacionadas por una homografía, sus listas de keyPoints calculados de acuerdo al punto anterior y las correspondencias encontradas entre dichas listas. Estimar la homografía entre ellas usando la función `findHomography(p1,p2,CV_RANSAC,1)` y crear un mosaico con ambas imágenes. Para el mosaico será necesario. a) definir una imagen en la que pintaremos el mosaico; b) definir la homografía que lleva cada una de las imágenes a la imagen del mosaico; c) usar la función `warpPerspective()` para trasladar cada imagen al mosaico (ayuda: mirar el flag `BORDER_TRANSPARENT` de `warpPerspective` para comenzar). (2 puntos)

Ayuda para la Supresión de No-máximos: Esta fase del algoritmo elimina como candidatos aquellos píxeles que teniendo un valor alto de criterio Harris dicho valor no es el máximo local de un entorno centrado en dicho punto de tamaño prefijado (parámetro de entrada). En consecuencia solo estamos interesados en píxeles cuyo valor Harris represente un máximo local. La selección máximos locales puede implementarse de distintas maneras, pero una que es razonablemente eficiente es la siguiente: a) escribir una función que tomando como entrada los valores de un entorno nos diga si el valor del centro es máximo local; b) escribir una función que sobre una imagen binaria inicializada a 255, sea capaz de modificar a 0 todos los píxeles de un rectángulo dado; c) Fijar un tamaño de entorno/ventana y recorrer la imagen de valores Harris con dicha ventana preguntando, en cada posición de valor 1 de la imagen binaria, si el valor Harris es máximo local; d) en caso negativo poner a cero dicho píxel en la imagen binaria; e) en caso afirmativo, poner a cero en la imagen binaria todos los píxeles del entorno (menos el central); f) los píxeles que queden con valor 255 son los píxeles seleccionados.

En el fichero imagenes.rar se encuentran todas las imágenes citadas y varios grupos de imágenes con los que poder componer mosaicos en distintas proyecciones

No está permitido usar los recursos del módulo stitching de OpenCV.

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 texto plano)

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, cuyo nombre debe ser Apellido1_P[1-3].zip.
4. En C++ SOLO ENTREGAR EL CODIGO FUENTE. En python incluir el directorio "imágenes" con las imágenes usadas.
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. Solo poner puntos de parada para mostrar imágenes o datos por consola
9. NO ESTA PERMITIDO EL USO DE CODIGO C/C++ que no sea absolutamente estándar. Por ejemplo no se permite dimensionar matrices a partir de variables o usar tipos de datos específicos de un compilador.

Forma de entrega:. Subir el zip al Tablón docente de CCIA.