

Trabajos de Visión por Computador

TRABAJO-2

Estimación de Homografías y Construcción de panoramas

FECHA DE ENTREGA: 18 de noviembre

Valoración total: 20 puntos (11 Cuestionario, 9 implementación)

Cuestionario de Teoría: (estará disponible en la web más adelante)

TRABAJO de IMPLEMENTACIÓN :

1.- Estimación de una homografía: Usar las imágenes tablero1 y tablero2 incluidas en el fichero de datos para estimar la homografía que relaciona ambas imágenes. (2.5 puntos)

Para ello realizar lo siguiente:

1.- Seleccionar a mano un conjunto de 10 puntos en correspondencias en ambas imágenes (distribuir dichos puntos de la forma más uniforme posible entre los puntos esquina del tablero).

2.- Estimar la homografía (implementar código propio, **SIN** usar findHomography()) definida entre las imágenes por las parejas de puntos seleccionados

3.-Usar la función warpPerspective() para generar a partir de una de las imágenes y la homografía estimada, la otra imagen. Comentar el resultado obtenido en comparación con la imagen original.

4.- Repetir los puntos anteriores pero seleccionando nuevos 10 puntos, en este caso todos extraídos de 3 cuadrados contiguos de una misma esquina. Comparar el resultado obtenidos con el anterior y valorar las diferencias encontradas.

2.- Usar los detectores SIFT y SURF de OpenCV sobre las imágenes de Yosemite para extraer y pintar sobre las mismas las regiones relevantes encontradas por los mismos (mirar el contenido de la estructura keyPoint asociada a cada punto detectado). Comenzar usando los parámetros por defecto de los constructores de SIFT y SURF e ir modificando los umbrales de detección hasta que obtengamos resultados razonables). Comparar los resultados obtenidos por ambos detectores y justificar los parámetros usados (1.5 punto)

3.- Con los resultados de detección obtenidos en el punto anterior extraer el descriptor asociado a cada punto y establecer la lista de puntos en correspondencias existentes en las imágenes (Ayuda: usar la clase

DescriptorMatcher y BFMatcher). Valorar la calidad de los resultados obtenidos bajo los criterios BruteForce+crossCheck y FlannBasedMatcher(). (1 punto)

4.- Escribir una función que tome como entrada dos imágenes relacionadas por una homografía, sus listas de keyPoints en correspondencia calculados de acuerdo al punto anterior y estime la homografía entre ellas usando el criterio RANSAC con la función findHomography(p1,p2,CV_RANSAC,1). Crear un mosaico con ambas imágenes. (Ayuda: 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: usar `borderMode=BORDER_TRANSPARENT`)). (2 puntos)

5.- Generar un mosaico de proyección plana a partir de múltiples imágenes. Estimar las homografías entre imágenes consecutivas y construir el mosaico a partir de ellas. La calidad del registrado geométrico y la menor deformación posible son los objetivos a alcanzar. (No está permitido usar los recursos del módulo stitching de OpenCV). (2 puntos)

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

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.