

Tarea 2 EL7008 – Primavera 2020

Cálculo de puntos de interés y alineamiento de imágenes

Profesor: Javier Ruiz del Solar

Auxiliar: Patricio Loncomilla

Ayudantes: Juan Pablo Cáceres, Hans Starke, Javier Smith, José Villagrán

Fecha enunciado: 01/10/2020

Fecha entrega: 14/10/2020

Esta tarea tiene dos objetivos: (i) calcular puntos de interés usando un detector de Harris y usando un detector de Shi-Tomasi, analizando su robustez ante rotaciones y (ii) alineamiento de imágenes usando descriptores SIFT.

Un par punto de interés – descriptor se define como:

$$(x, y, \vec{d})$$

donde (x, y) corresponde a la posición asociada al punto de interés. El descriptor \vec{d} corresponde a un vector de características que describe la vecindad del punto de interés.

Se puede generar correspondencias entre dos imágenes pareando cada descriptor de una imagen con el descriptor más parecido (menor distancia) en una segunda imagen. Como ejemplo, se muestran múltiples correspondencias entre dos imágenes. Las correspondencias incorrectas pueden ser eliminadas con algoritmos que encuentren transformaciones consistentes con la mayoría de las correspondencias.



Parte 1: Marco teórico

- a) Explique en que consiste el detector de Harris
- b) Explique en que consiste el detector Shi-Tomasi
- c) Describa cómo se puede generar correspondencias usando descriptores
- d) Describa en qué consiste una transformación de homografía

Parte 2: Cálculo de puntos de interés

En esta parte de la tarea hay que programar tanto el detector de Harris como el de Shi-Tomasi usando Cython y Python, para luego analizar visualmente su robustez ante rotaciones.

- a) Completar las funciones para calcular gradientes en x e y: `grad_x()` y `grad_y()`.
- b) Completar la función que calcula el filtro de Harris: `harris()`.
- c) Completar la función que calcula el filtro de Shi-Tomasi: `shiTomasi()`.
- d) Completar la función que selecciona los máximos locales: `getMaxima()`.
- e) Implementar el detector de Harris: `harrisDetector()`.
- f) Implementar el detector de Shi-Tomasi: `shiTomasiDetector()`.
- g) Probar el nivel de invarianza/robustez de las imágenes resultantes al aplicar el filtro de Harris y el de Shi-Tomasi cuando la imagen es rotada en 30 grados, sobre las 3 imágenes entregadas. Evaluar el nivel de invarianza visualmente.
- h) Probar el nivel de invarianza/robustez tanto de los puntos de interés detectados mediante Harris y mediante Shi-Tomasi cuando la imagen es rotada en 30 grados, Sobre las 3 imágenes entregadas. Evaluar el nivel de invarianza visualmente.

Parte 3: Alineamiento de imágenes

En esta parte de la tarea se debe generar una imagen fusionada, alineando un par de imágenes (un par: imagen izquierda, imagen derecha). En esta parte de la tarea, ya se provee el código que calcula y grafica las correspondencias

- a) Generar el código que calcula la homografía, usando la función `cv2.findHomography()`
- b) Generar un código para obtener una imagen transformada usando la homografía, aplicada sobre la imagen derecha. La imagen resultante debe tener un tamaño suficiente para contener la imagen completa
- c) Generar un código para copiar la imagen izquierda sobre la imagen transformada
- d) Procesar los 4 pares de imágenes entregados para la parte 2, y presentar los resultados (las correspondencias, y las imágenes fusionadas)

El código entregado corresponde a una implementación incompleta del sistema, en la cual se indican los pasos que se deben seguir para poder completar la tarea.

El informe debe contener como mínimo: introducción, marco teórico (descripción de los algoritmos), partes relevantes del código dentro de cada sección del informe, resultados (mostrando las imágenes resultantes), análisis de los resultados y conclusiones generales.

Los informes y los códigos deben ser entregados a más tardar el día miércoles 14 de octubre a las 23:59, mediante u-cursos. Cada día de retraso (incluyendo fines de semana) será castigado con un punto de descuento en la nota.

Importante: La evaluación de esta tarea considerará el correcto funcionamiento del sistema, la inclusión de los resultados de los pasos pedidos en el informe, la

calidad de los experimentos realizados y de su análisis, la inclusión de las partes importantes del código en las secciones del informe correspondientes, así como la prolijidad y calidad del mismo.

Se agrega una pauta que indica la estructura esperada del informe.

Dentro de las funciones de Cython, las variables a ser usadas deben ser declaradas al inicio de la función usando `cdef int` o bien `cdef float`

Es posible que haya que ajustar el umbral usado en Harris y Shi-Tomasi dependiendo de la imagen que se procese.

Se recuerda a los alumnos que el informe se debe subir a la plataforma turnitin.

Anexo: cálculo del filtro de Harris y Shi-Tomasi

La respuesta de los filtros de Shi-Tomasi y Harris se pueden calcular en cada píxel es:

$$cornerness_{harris} = \lambda_1 * \lambda_2 - 0.04 * (\lambda_1 + \lambda_2)^2$$

$$cornerness_{shi-tomasi} = \lambda_1$$

donde λ_1 es el menor valor propio de la matriz m , y λ_2 es el mayor valor propio de la matriz m :

$$m = \begin{pmatrix} m_{xx} & m_{xy} \\ m_{xy} & m_{yy} \end{pmatrix} = \sum_{(x,y)} w(x,y) \begin{pmatrix} I_x * I_x & I_x * I_y \\ I_x * I_y & I_y * I_y \end{pmatrix}$$

En la formula anterior, m es el tensor de estructura, I_x es el gradiente de la imagen en x , I_y es el gradiente de la imagen en y , y $w(x,y)$ es una función gaussiana.

Los valores propios se pueden escribir en función del determinante $det = \lambda_1 * \lambda_2$ y la traza $tr = \lambda_1 + \lambda_2$. En consecuencia, la respuesta de los filtros se puede calcular en la práctica del siguiente modo:

$$cornerness_{harris} = det - 0.04 * tr^2$$

$$cornerness_{shi-tomasi} = \frac{tr}{2} - \frac{\sqrt{tr^2 - 4 * det}}{2}$$

con:

$$det = m_{xx} * m_{yy} - m_{xy}^2$$

$$tr = m_{xx} + m_{yy}$$