

Tarea 2 EL7008 – Primavera 2022

Cálculo de puntos de interés Harris y reconocimiento de objetos particulares usando SIFT y RANSAC

Profesor: Javier Ruiz del Solar
Auxiliar: Patricio Loncomilla

Fecha enunciado: 1 de Septiembre de 2022
Fecha entrega: 21 de Septiembre de 2022

Esta tarea tiene dos partes: (1) implementación del cálculo de puntos de interés mediante el método de Harris, e (2) implementación de detección de objetos usando puntos de interés y descriptores SIFT, usando RANSAC.

Para la parte (1), se debe programar un detector de Harris usando funciones implementadas con Cython y funciones para recuperar los puntos de interés a partir de las imágenes filtradas usando Python.

Para la parte (2), se obtendrán los puntos de interés y descriptores SIFT (no es necesario programar esta parte), se generarán calces entre descriptores parecidos entre dos imágenes y se usará RANSAC para encontrar un grupo de calces que sean compatibles con una misma transformación de semejanza entre una imagen de prueba (*pru*) y otra de referencia (*ref*).

Antecedentes Generales

1.- Un par punto de interés – descriptor se define como:

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

donde (x, y, σ, φ) corresponde a la posición, escala y orientación asociada al punto de interés (se puede ver como la posición, largo y orientación de una flecha). El descriptor \vec{d} corresponde a un vector de características que describe la vecindad del punto de interés.

2.- Una transformación de semejanza se define como:

$$\begin{pmatrix} x_{PRU} \\ y_{PRU} \end{pmatrix} = e \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_{REF} \\ y_{REF} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

donde e corresponde al factor de escala entre las imágenes, θ es una rotación y (t_x, t_y) es la traslación.

3.- Una transformación afín se define como:

$$\begin{pmatrix} x_{PRU} \\ y_{PRU} \end{pmatrix} = \begin{pmatrix} m_{XX} & m_{XY} \\ m_{YX} & m_{YY} \end{pmatrix} \begin{pmatrix} x_{REF} \\ y_{REF} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Alternativamente, dejando los parámetros de la transformación afín separados en un vector, se puede describir como:

$$\begin{pmatrix} x_{REF} & y_{REF} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_{REF} & y_{REF} & 0 & 1 \end{pmatrix} (m_{XX} \ m_{XY} \ m_{YX} \ m_{YY} \ t_x \ t_y)^T = \begin{pmatrix} x_{PRU} \\ y_{PRU} \end{pmatrix}$$

O en forma equivalente como:

$$Ax = b$$

De este modo, si se conoce A y b , y se quiere resolver el siguiente problema:

$$\operatorname{argmin}_x \|Ax - b\|^2$$

la solución se puede obtener como:

$$x^* = (A^T A)^{-1} A^T b$$

4. La transformación de semejanza que proyecta un punto de interés $(x_{REF}, y_{REF}, \sigma_{REF}, \varphi_{REF})$ de la imagen de referencia en un punto de interés $(x_{PRU}, y_{PRU}, \sigma_{PRU}, \varphi_{PRU})$ en la imagen de prueba se define como:

$$e = \frac{\sigma_{PRU}}{\sigma_{REF}}$$

$$\theta = \varphi_{PRU} - \varphi_{REF}$$

$$t_X = x_{PRU} - e(x_{REF} \cos(\theta) - y_{REF} \sin(\theta))$$

$$t_Y = y_{PRU} - e(x_{REF} \sin(\theta) + y_{REF} \cos(\theta))$$

5. El error de proyección, dada una transformación (e, θ, t_X, t_Y) , se define como:

$$e_{PROY} = \left\| e \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_{REF} \\ y_{REF} \end{pmatrix} + \begin{pmatrix} t_X \\ t_Y \end{pmatrix} - \begin{pmatrix} x_{PRU} \\ y_{PRU} \end{pmatrix} \right\|$$

6. Para RANSAC: Dado un conjunto de calces, el consenso asociado a una transformación corresponde a un subconjunto de los calces cuyo error de proyección es menor a un umbral, es decir, los que son compatibles con la transformación. RANSAC intenta encontrar una transformación con el mayor consenso posible.

Se pide realizar los siguientes pasos, cuyos resultados deben ser incluidos en el informe:

Parte 1: Cálculo de puntos de interés Harris

En esta parte de la tarea hay que programar un detector de Harris usando Cython y Python, para luego analizar visualmente su robustez ante rotaciones.

- Completar las funciones para calcular gradientes en x e y: `grad_x()` y `grad_y()`.
- Completar la función que calcula el filtro de Harris: `harris()`.
- Completar la función que selecciona los máximos locales: `getMaxima()`.
- Implementar el detector de Harris: `harrisDetector()`.
- Probar el nivel de invarianza/robustez de los puntos de interés detectados mediante Harris cuando la imagen es rotada en 30 grados, sobre 3 de las imágenes entregadas. Evaluar visualmente la repetibilidad de los puntos y la exactitud de su localización.

Parte 2: Reconocimiento de objetos particulares

- Implementar la función que permite generar una transformación de semejanza a partir de un calce. Se debe notar que en OpenCV las orientaciones de los descriptores son entregadas en grados.
- Obtener la transformación de semejanza con mayor consenso y los calces compatibles usando RANSAC:
 - Programar la selección de calces al azar para generar la hipótesis
 - Programar el cálculo del consenso (calces compatibles con la hipótesis)
 - Programar la selección de la hipótesis con el mayor consenso
 - Probar con los 4 pares de imágenes entregados, graficando los calces
- A partir de un conjunto de calces, calcular y graficar la transformación afín asociada usando mínimos cuadrados:

- a. Implementar la función que encuentra una transformación afín usando mínimos cuadrados a partir de un conjunto de calces
 - b. A partir de los calces encontrados en el punto 2c, calcular una transformación afín usando mínimos cuadrados y graficar el romboide asociado (el código para graficar el romboide se entrega). Hacer esto para los cuatro pares de imágenes.
- 4) Evaluar el efecto de variar los tres parámetros indicados de RANSAC (error máximo para *inliers*, consenso mínimo y número máximo de intentos) sobre las detecciones obtenidas, usando un par de imágenes en que haya una detección inicial correcta.

Se entrega un código base en Python con la estructura esperada del código en la parte (1), y que en la parte (2) incluye un código que ya calcula los puntos de interés, los descriptores y los calces iniciales. Las funciones que calculan la transformación a partir de un calce, la que realiza RANSAC, y la que calcula una transformación afín por mínimos cuadrados deben ser implementadas para lograr que el sistema funcione adecuadamente.

El código además incluye indicaciones que pueden ayudar a implementar la tarea.

El informe debe contener como mínimo: introducción, partes relevantes del código dentro de cada sección del informe, resultados (mostrando las imágenes resultantes), análisis de los resultados, conclusiones generales y un anexo con el código.

Los informes y los códigos deben ser entregados a más tardar el día miércoles 21 de septiembre 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.

El informe debe ser subido a turnitin en formato pdf. El código en el informe debe ser agregado como texto (no como pantallazo).

Parámetros recomendados para RANSAC: 100 intentos, umbral de error de posición 100, umbral de consenso 30

Anexo: cálculo del filtro de Harris

La respuesta del filtro de Harris se puede calcular en cada píxel es:

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

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:

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

con:

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

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