

Tarea 2 EL7008 – Primavera 2019

Análisis de puntos de interés y generación de correspondencias

Profesor: Javier Ruiz del Solar

Auxiliar: Patricio Loncomilla

Ayudantes: Gabriel Azócar, Giovanni País, Nicolás Cruz

Fecha enunciado: 22 de Agosto de 2019

Fecha entrega: 4 de Septiembre de 2019

Esta tarea tiene dos objetivos: (i) un análisis visual de la robustez ante transformaciones geométricas de tres tipos de detectores de puntos de interés, y (ii) programación de la generación de calces (o correspondencias) entre pares de imágenes, usando puntos de interés DoG y descriptores SIFT. Para esto, se obtendrán los puntos de interés y descriptores (no es necesario programar esta parte), y se generarán calces entre descriptores parecidos entre dos imágenes.

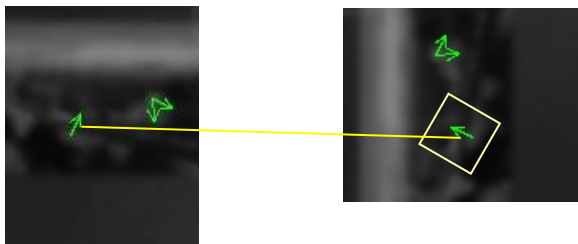
Un par punto de interés – descriptor se define como:

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

Donde (x, y, σ, θ) 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.

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 muestra la imagen de una correspondencia, y de múltiples correspondencias entre dos imágenes. Las correspondencias incorrectas pueden ser eliminadas con algoritmos que no serán abordados en esta tarea.



Parte 1: Invarianza de puntos de interés

- a) Detectar y graficar los puntos de interés en las 3 imágenes entregadas junto con el enunciado de la tarea para la parte 1, usando puntos de interés ORB, DoG (usado en SIFT) y *Good Features to Track* (similar a Harris). Analizar la cantidad de puntos de interés generado por cada método.
- b) Probar el nivel de invarianza de los tres detectores de puntos de interés cuando la imagen es escalada a tamaño $\frac{1}{2}$, evaluar el nivel de invarianza visualmente. El escalamiento de la imagen se debe implementar en la función `do_downsample()`, usando cualquier función de OpenCV disponible, evitando que se produzca aliasing. Analizar y concluir sobre el nivel de invarianza de los detectores.
- c) Probar el nivel de invarianza de los tres detectores de puntos de interés cuando la imagen es rotada en 30 grados, evaluar el nivel de invarianza visualmente. La rotación de la imagen se debe implementar en la función `do_rotate()`, usando cualquier función/funciones de OpenCV disponible. En este caso, la imagen original debe estar completamente contenida en la imagen rotada. Analizar y concluir sobre el nivel de invarianza de los detectores.

Parte 2: Generación de correspondencias

- a) Detectar y graficar los puntos de interés en los 5 pares de imágenes entregadas junto con el enunciado de la tarea para la parte 2, usando puntos de interés DoG (los puntos DoG y descriptores SIFT son calculados a la vez).
- b) Programar la función `do_match()`. Esta función debe calcular las correspondencias entre descriptores de dos imágenes entregadas. Se debe programar sin usar funciones de OpenCV, procesando los descriptores elemento a elemento, usando `at<float>(i,j)`, donde *i* representa el índice del número de descriptor y *j* representa el índice de la dimensionalidad (128 en el caso de SIFT). Se entrega un esqueleto de esta función en el código.
- c) Calcular y graficar las correspondencias en 5 pares de imágenes entregadas.
- d) Analizar la calidad de las correspondencias obtenidas.

Se entregan dos proyectos que proveen una funcionalidad básica, uno para cada parte de la tarea. De este modo, el alumno debe enfocarse en programar los algoritmos pedidos y hacer las pruebas solicitadas. Además, los códigos entregados contienen indicaciones para facilitar la programación. El código a implementar debe basarse en el código base entregado, modificando sólo los archivos `main_parte1.cpp` y `main_parte2.cpp`.

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 4 de sept. 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.