

## Estructura tarea 2

### Introducción:

- Describir brevemente lo que se realizará en la tarea
- Enumeración y explicación de las secciones que siguen

### Marco teórico (incluir figuras / ecuaciones cuando corresponda, referencias):

- Describir en qué consiste un punto de interés y los tipos existentes (blob, esquina)
- Describir en qué consiste un descriptor
- Describir cómo se puede generar correspondencias usando descriptores

### Desarrollo parte 1:

- Detectar puntos de interés en tres imágenes, usando tres detectores
- Programar código para dibujar los puntos de interés, mostrar y guardar la imagen resultante
- Programar función de escalamiento `do_downsample( )`, reportar en el informe
- Evaluar visualmente invarianza a escala de los 3 detectores sobre las 3 imágenes
- Programar función de rotación `do_rotate( )`
- Evaluar visualmente invarianza a rotación de los 3 detectores sobre las 3 imágenes

### Desarrollo parte 2:

- Detectar y graficar puntos de interés sobre 5 pares de imágenes entregadas
- Programar la función `do_match( )`
- Calcular y graficar las correspondencias en 5 pares de imágenes entregadas
- Analizar visualmente la calidad de las correspondencias obtenidas

### Conclusiones

- Señale cuáles fueron los aprendizajes obtenidos al realizar la tarea y las dificultades que encontró durante su desarrollo. Señale si los resultados obtenidos son los que esperaba en función de lo que indica la teoría.

Nota extra: Los códigos deben ser implementados en C++, usando OpenCV. En esta tarea, se pueden usar funciones libremente salvo en la función `do_match( )`, la que debe ser programada a bajo nivel usando ciclos `for` y acceso a pixeles usando `at<float>( )`.

Nota extra 2: Todos los códigos implementados deben ser incluidos en la sección correspondiente

Nota extra 3: Los resultados (imágenes generadas) deben ser incluidos en la sección correspondiente