

Trabajo 4. Panorama

El objetivo es desarrollar un programa capaz de crear panoramas a partir de un conjunto de imágenes. Para ello se utilizarán los módulos `calib3D` y `features2D` de OpenCV. Como mínimo se deberán presentar resultados con:

- A. Imágenes de un objeto plano tomadas desde distintos puntos de vista
- B. Imágenes de una escena 3D de exterior tomadas girando la cámara

Se recomienda consultar:

- OpenCV User guide, cap. 2, con el ejemplo:
 - ✓ `opencv/sources/samples/cpp/matcher_simple.cpp`
- OpenCV Tutorials, Cap. 5 (`calib3d`) y Cap. 6 (`feature2d`), ejemplos:
 - ✓ `opencv/sources/samples/cpp/descriptor_extractor_matcher.cpp`
 - ✓ `opencv/sources/samples/cpp/tutorial_code/features2D/`
 - ✓ `opencv/sources/samples/cpp/tutorial_code/calib3d/camera_calibration/`
- OpenCV Reference Manual

Se recomienda seguir estos pasos:

1. Toma de imágenes y paso a niveles de gris, extracción de puntos de interés y búsqueda de emparejamientos iniciales por fuerza bruta (todos con todos) buscando el vecino más próximo, y comprobando el ratio al segundo vecino. El programa deberá poder mostrar los emparejamientos obtenidos.
2. **(opcional)** Calibración de la cámara y corrección de la distorsión radial.
3. Búsqueda de la homografía que alinea las imágenes con RANSAC. El programa deberá poder mostrar los *inliers* obtenidos. En este punto se proponen 3 alternativas de menor a mayor dificultad (y puntuación):
 - a) OpenCV permite utilizar RANSAC automáticamente cuando se calcula la transformación entre imágenes.
 - b) Implementar una función propia que lleve a cabo el RANSAC calculando de antemano el número de intentos que se llevarán a cabo.
 - c) Implementar una función propia que lleve a cabo el RANSAC y que calcule de forma adaptativa el número de intentos necesario (nota: se puede hacer una estimación sencilla de la probabilidad p de que un punto elegido al azar pertenezca al modelo a medida que se van haciendo intentos).

-
4. Creación del panorama. A medida que se vayan introduciendo imágenes se calculará la alineación y se actualizará el panorama. El programa deberá mostrar el tiempo de cómputo total necesario para añadir cada imagen. Aquí también se proponen distintos niveles de dificultad y puntuación:
 - a) Trabajar con imágenes guardadas en disco
 - b) Construir el panorama en vivo a partir de las imágenes de la cámara, pulsando en el teclado cada vez que se quiere añadir una imagen nueva
 - c) Idem, con toma de imágenes automática (sin pulsar el teclado), manteniendo un solape suficiente
 5. **Opcional:** Cualquier mejora que se os ocurra. Algunos ejemplos:
 - ✓ Comparar resultados con SURF, SIFT y ORB
 - ✓ Comparar resultados con FLANN
 - ✓ Eliminar fantasmas
 - ✓

Deberéis entregar en moodle:

1. Memoria en pdf explicando cómo habéis implementado el trabajo, discutiendo las decisiones de diseño que hayáis tomado, e incluyendo ejemplos de resultados.
2. Código completo desarrollado.

Fecha de entrega y defensa: 29-Abril-2015