Odometría Visual

Tutor: Gonzalo Olguín

1. Descripción

La Odometría Visual es un problema en visión computacional aplicado a la robótica y vehículos autónomos, donde la meta es estimar el movimiento relativo de una cámara en el espacio a partir de una secuencia de fotogramas. Gracias a esta estimación y métodos más avanzados de SLAM, un vehículo puede realizar tareas de localización en tiempo real, navegación autónoma, mapeo 3D, entre otras. Si bien los métodos más robustos se basan en una fusión sensorial que incluye LiDARs, IMUs o imágenes de profundidad, desarrollar sistemas que funcionen sólo en base a imágenes permite reducir costos considerablemente.

Figura 1: Keypoints obtenidos con FAST.

La odometría visual se puede abordar de diversas maneras, pero para visión únicamente monocular (una sola cámara, sin sensores extra), en general hay dos tipos de métodos:

- **Métodos Clásicos Basados en Keypoints/matching**: Utilizan detectores y descriptores como ORB, SIFT o SURF para identificar puntos de interés en cada imagen y luego realizar matching entre fotogramas consecutivos. la transformación entre fotogramas consecutivos se obtiene de la optimización del error de reproyección entre los matches y, en base a la transformación, se estima la trayectoria.

- **Métodos Basados en Deep Learning**: Se usan redes neuronales para extraer características más complejas y robustas, para el matching entre características o incluso para predecir de forma directa la pose relativa entre dos imágenes consecutivas. Estos métodos tienden a ser más precisos en entornos complejos debido a su robustez ante cambios de iluminación, pero requieren una gran cantidad de datos de entrenamiento y mayor capacidad computacional.

Para este proyecto, se deberá implementar un sistema básico de odometría visual 2D usando el dataset KITTI [1]. Se proponen dos enfoques, aunque se acepta que los estudiantes sugieran un método diferente o combinaciones de éstos:

- Odometría Visual Basada en Características deep: Utilizar extractores de características como SuperPoint [2] para detección de keypoints, y métodos tradicionales o deep (Ej. SuperGlue [3]) para realizar matching entre fotogramas y estimación de transformación relativa.
- Estimación de Pose Completa con redes deep: Entrenar o realizar finetunning de una red neuronal para predecir la transformación entre fotogramas de forma directa.. Ejemplos de esto son PoseNet [4] o DeepVO [5].

2. Objetivos

- Desarrollar un sistema de odometría visual en 2D que aproxime la trayectoria de una cámara en movimiento.
- Implementar y evaluar uno de los enfoques propuestos en al menos dos trayectorias del dataset KITTI, en base a métricas relevantes.
- Analizar los efectos de los parámetros del/los modelo(s) en término de las métricas sobre el *ground truth*.
- Investigar sobre los métodos y desafíos de la odometría visual en robótica.

3. Observaciones

1. El dataset KITTI en escala de grises con ground truth tiene un peso de aproximadamente 12GB y los métodos con entrenamiento de redes pueden llegar a tomar un largo tiempo de entrenamiento. Se recomienda para los últimos dos métodos tener una computador con GPU o acceso a Google Colab.

2. Muchos de los métodos para odometría visual monocular son de código abierto, incluyendo los mencionados en este documento.

3. Debido a la complejidad del problema y el nivel/tiempo del curso, no se espera que se obtengan resultados de alta precisión. Sin embargo, el objetivo es lograr un sistema que pueda seguir la trayectoria de manera aproximada, comparable en precisión básica a métodos clásicos como ORB (para odometría, no SLAM), permitiendo observar un seguimiento general de la trayectoria sin considerar la escala de ésta.

Referencias

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," *CoRR*, vol. abs/1712.07629, 2017.

[3] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," *CoRR*, vol. abs/1911.11763, 2019.

[4] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-dof camera relocalization," *CoRR*, vol. abs/1505.07427, 2015.

[5] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," *CoRR*, vol. abs/1709.08429, 2017.