

Stereo Visual Odometry using Kitty Dataset

Use stereo images to find 3D locations of keypoints in those images

Dataset: Kitty dataset, gray odometry. Contains left and right stereo images of a scene and camera intrinsics

Frontend:

1. Input is image frame from the sensor
2. Frontend is responsible for extracting the features in the image
3. Performs optical flow tracking or feature matching with the previous frame and calculates the frame's position based on the optical flow result
4. New feature points should be added and triangulated
5. The result of the frontend processing will be used as the initial value of the backend optimization.

Backend:

1. The backend is a slower thread
2. Input is processed keyframes and landmark points
3. Optimizes input and then returns the optimized results
4. Should control the optimization problem's scale within a certain range and cannot keep growing over time.

Steps:

1. Read dataset
2. Get next frame
3. Add the frame
 - a. Initialization: Establishing initial map by triangulating between left and right images
 - b. Tracking: From previous to current frame. Only used for left eye image
 - i. Extract new feature points if tracked features are less. Find corresponding points on right and create new landmarks.

Img -> extract features -> feature matching with prev frame -> pose estimation -> keyframe -> add points in map if needed -> stereo matching -> triangulation -> map update -> backend -> map update

Gained major experience in:

1. C++
2. Libraries: ceres, openCV
3. Major available open source pipelines like ORB SLAM

Point of improvements:

1. C++
2. I used GFTT as feature detector. This is not fast. Can use ORB, SIFT etc
3. Taking care of edge cases, like changing brightness, lost tracking,
4. Used direct method for estimating pose and features for faster computation. Can compare with other techniques like OpticalFlow
5. Can use pose graph instead of BA
6. Add ways to quantify results
7. Include features in major pipeline