Monocular Visual Odometry

Fatma Moanes NourEl-Din
Communication and Information Engineering
ZEWAIL CITY OF SCIENCE AND TECHNOLOGY
Cairo, Egypt
Email: s-fatma_moanes@zewailcity.edu.eg

Mahmoud Yasser
Communication and Information Engineering
ZEWAIL CITY OF SCIENCE AND TECHNOLOGY
Cairo, Egypt
Email: s-mahmoudyasser@zewailcity.edu.eg

Mai Elyamani
Communication and Information Engineering
ZEWAIL CITY OF SCIENCE AND TECHNOLOGY
Cairo, Egypt
Email: s-mai.elyamani@zewailcity.edu.eg

Mohamed Elshaffei
Communication and Information Engineering
ZEWAIL CITY OF SCIENCE AND TECHNOLOGY
Cairo, Egypt
Email: s-mohamed_chaffei@zewailcity.edu.eg

I. Introduction

Precise fully autonomous positioning is a critical task in many different applications such as SLAM robotics and self driving cars. Autonomous positioning and pose estimation can be achieved using different sensors such as Inertial Measurement Units, Encoders, and GPS. However, only using sensors may not yield the most accurate pose estimation. For example, IMUs are known to be prone to errors especially in the presence of external vibrations or magnetic fields, which affect the accelerometer and magnetometer inside the IMU. Also, encoders might suffer from slippage of wheels which leads to erroneous results.

Visual Odometry is the process of estimating the camera's motion by observing the relative motion of the environment in consecutive frames from the camera. Moreover, Visual Odometry is backed up by the increasing computational power which allows the processing of high resolution images while maintaining real-time performance. The benefits of using Visual Odometry is that it is not prone to the same errors that affect sensors such as IMU and encoders. Therefore, fusing data from Visual Odometry and different sensors should lead to better results than just using sensors.

II. Optical Flow

Optical flow is the pattern of motion of objects in a pair of consecutive images. The basic assumption of optical flow is that the brightness values of correspondences remain constant in both images, which is a reasonable assumption since the time difference between both images should be very small. Quantifying the problem of optical flow in order to understand how to solve it is needed. Firstly, the brightness constancy is given by:

$$f(x, y, t) = f(x + \Delta x, y + \Delta y, t + \Delta t)$$

The above equation states that for two images separated by a time difference of Δt then the brightness at a point (x, y) in the first image is equal to the same point on the second image after being displaced by $(\Delta x, \Delta y)$. The first order Taylor expansion of the above formula is:

$$f(x, y, t) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t + O(\Delta x, \Delta y, \Delta t)$$

By inserting the above equation in the brightness constancy constraint equation shown before we get:

$$\frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \frac{\partial f}{\partial t}\Delta t + O(\Delta x, \Delta y, \Delta t) = 0$$

Now, we divide the above equation by Δt and take the limits as Δt tends to zero to get:

$$\frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v + \frac{\partial f}{\partial t} = 0$$

where u and v are the velocities of point (x, y) in the direction of x and y respectively. Now, the problem of optical flow can be defined as the problem of finding u and v for points of interest to track such points across consecutive images.

III. Lucas Kanade

The equation containing the velocities u and v can not be solved on its own, since it contains two unknowns. To solve this problem, Lucas Kanade algorithm proposed the use of a window around each pixel where we assume that the motion of all the pixels in that window is the same which means that all the pixels within the window have the same u and v. Therefore, for a $3x3$ window we have nine equations and two unknowns, which is over-determined. Now, we have a problem in the form of:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_9) & I_y(p_9) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_9) \end{bmatrix}$$

which is in the form of

$$\mathbf{Ax} = \mathbf{b}$$

Therefore, we can solve the above relation using least squares to find good estimates for u and v. The least squares solution to u and v is given by:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_x(q_i) \\ \sum_i I_x(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix}$$

After calculating the velocities u and v we can track a point from frame t to frame $t+1$ by setting $\Delta x = u$ and $\Delta y = v$.

IV. Essential Matrix and Pose Estimation

The essential matrix is a 3 × 3 matrix, E that relates corresponding points in stereo images assuming that the cameras satisfy the pinhole camera model. if y and y' are homogeneous normalized image coordinates in image 1 and 2, respectively, then

yEy'=0

The essential matrix is useful for determining both the relative position and orientation between the cameras and the 3D position of corresponding image points.

E=R[t]

So decomposing the essential matrix will retrieve the relative pose and between the two images.

V. System Overview

The method we use to estimate the pose can be summarized in the following steps:

1) Extract features from key frame
2) Track extracted features to the next frame to find features correspondences
3) Estimate the essential matrix from feature correspondences using a RANSAC based scheme for outliers rejection

Figure 3, shows the system pipeline.

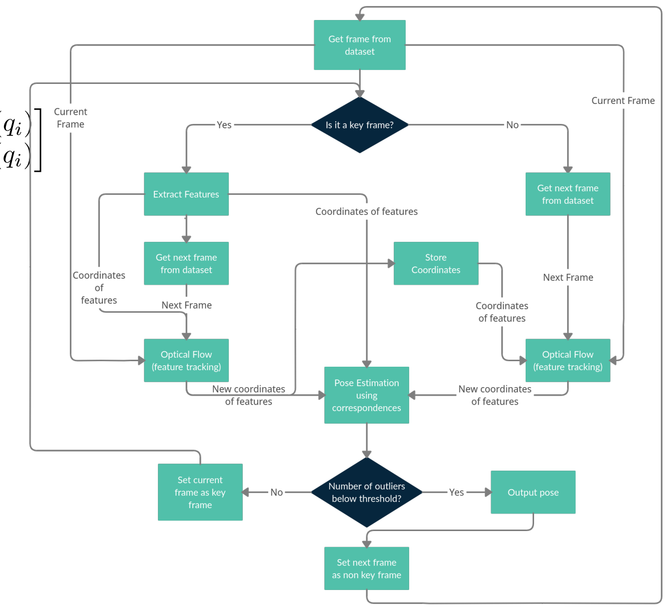

Fig. 1. System Pipeline

The system begins its operation by setting the first frame as a key frame. A key frame is the frame where the system does not perform feature tracking, instead it extracts new set of features to track them in the upcoming frames. The features extracted are then tracked from the key frame to the next frames using Lucas Kanade optical flow. The feature correspondences are used to estimate the essential matrix which is then used to recover the pose. In the process of essential matrix estimation the system uses 5-point algorithm, but it usually has a much larger number of correspondences than five. Therefore, we implemented a RANSAC scheme which keeps iteratively randomly sampling five points from the correspondences and use them to estimate the essential matrix until the best estimate for the essential matrix is obtained. We also set an outlier threshold, so if the number of outliers reported by the RANSAC is above this threshold the system will stop the process of feature tracking, and it will set the current frame as a key frame and restart. This allows the system to maintain the trade-off between performance and accuracy, since the system will not trigger any feature extraction, which is a costly operation, unless it is really needed to maintain accurate results.

VI. Results and Discussion

Figures 2 and 3 shows the estimated and the truth trajectories after running the algorithm for the first 150 frame.

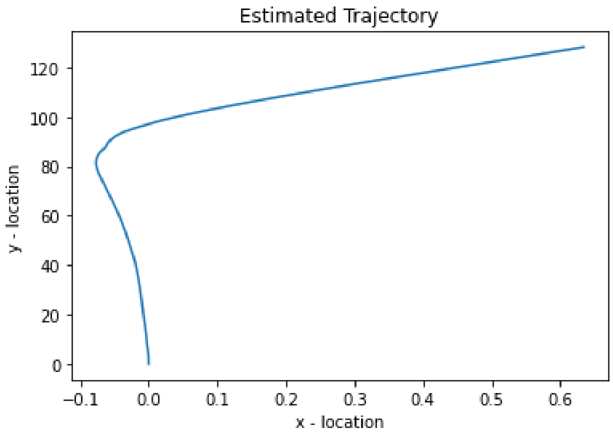

Fig. 2. Estimated trajectory

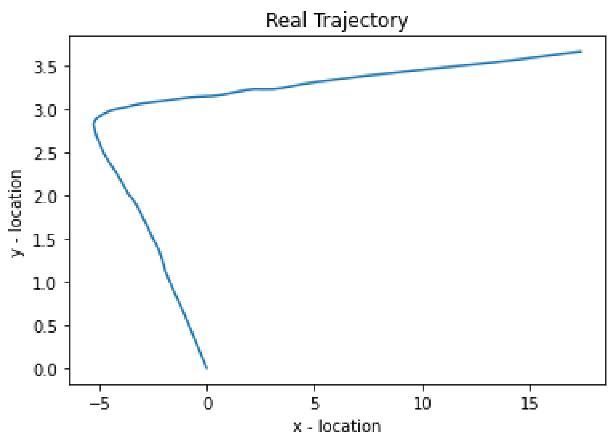

Fig. 3. True trajectory

There are few things to note on the results:

1) The scale of the y-location and x-location is not consistent in both of the trajectories. This is due to the fact that essential matrix only compute the relative normalized displacements between the two cameras. So, the estimated trajectory is calculated only based on the rotation matrix.
2) The noise may affect the optical flow tracking, making an accumulated error, which gives a total error of 20 degrees after the turn.

VII. Limitations

There are some limitations that may affect the efficiency of the algorithm:

1) In the pictures where there is a traffic light and the cars are not moving while there are any moving objects around them, it doesn't give accurate results as the other dataset pictures.
2) Large hyper parameter search space: our approach has tons to hyperparameters that must be jointly optimized:
   a) Feature extraction (block size, minimum distance)
   b) Lucas–Kanade (window size, blurring parameters).
   c) Essential Matrix estimation: Algorithm type (8-points, 5-points) and Ransac hyperparameters (outliers errors threshold).
   d) Over all system parameters (number of inliers threshold).
   e) Adaptive parameters.

VIII. Possible Improvements

Possible Improvements are

1) Using filters as kalman filter to deal with the noise and optimize the motion estimation.
2) Implementing pyramidal method (Lucas–Kanade)
3) Fusion between sensors and visual odometry methods.

References

[1] Chuanqi, C., Xiangyang, H., Zhenjie, Z., amp; Mandan, Z. (2017). Monocular visual odometry based on optical flow and feature matching. 2017 29th Chinese Control And Decision Conference (CCDC). https://doi.org/10.1109/ccdc.2017.7979301