Visual Odometry on a Smartphone

Jai Prakash
Carnegie Mellon University
Master of Science in Computer Vision
jprakash@andrew.cmu.edu

Utkarsh Sinha
Carnegie Mellon University
Master of Science in Computer Vision
usinha@andrew.cmu.edu

Abstract

In this project, we are trying to localize the camera using visual odometry. The major component of the project is to generate keyframes according to pre-defined heuristics and triangulate the points to create 3D reconstruction of the scene. The intermediate frames can be found using Perspective-n-point algorithm. In addition, we also perform local bundle adjustment over last few frames so that the localization is locally consistent. We also plan to exploit the onboard inertial sensors to get prior for the localization.

1. Introduction

Augmented reality has been around for years, yet not all problems are solved in the domain. One of the challenges is precise localization of the device in world coordinates. Several augmented reality applications on smartphones are based on markers. One good example of marker-based AR is Vuforia. On the other hand, there are standalone devices like Hololens, which has number of sensors to understand the scene and localize the head mounted display in the scene.

In many such applications, understanding the scene is not important. Localizing the camera in the world is enough to solve certain problems. In this project we focus on localizing the phone camera using the camera and the inertial sensors.

2. Background

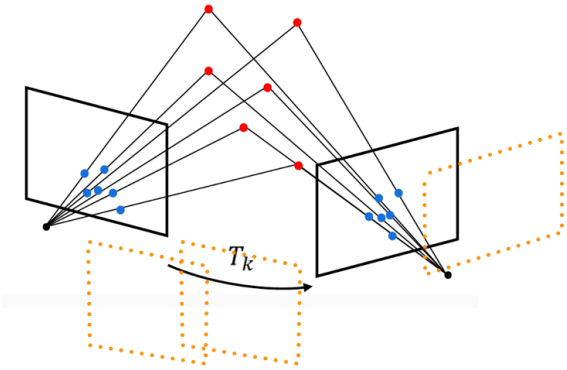
Visual SLAM vs. Visual Odometry: The focus in the visual SLAM techniques is both in reconstruting the scene and also localizing the camera in the scene. However, our main focus is just in localization of the camera. For the scope of the project, we focus only to be locally consistent. So, our system's camera position might drift over time - however, we are only interested in accurate localization in a short timespan. We do not explore ideas like loop closure in this project.

3. Method

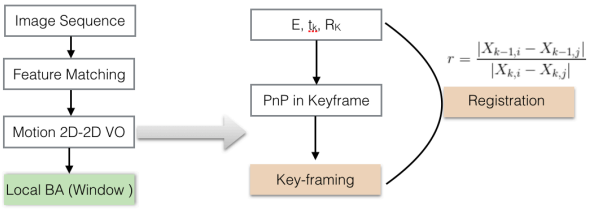
We are using feature based sparse reconstruction to localize the camera.

3.1. Feature Extraction and Matching

We have experimented with OpenCV KLT features, AKAZE features and ORB features. The KLT features can also be used for tracking the features in the subsequent frames using optical flow. For AKAZE[3] and ORB features, the correspondences are found using feature matching. The outliers are removed using the epipolar constraints and finding unique matches i.e. feature from first image matches to a unique feature in second image and vice-versa.



(a) A visualization of the Visual Odometry system. The black frames are the key-frames. The orange frames are intermediate frames whose pose is found using PnP.



(b) Block diagram of the system. The red blocks are implemented as a part of Geometry based vision project. For this project we are focusing on implementation of local bundle adjustment shown in green

3.2. 3D reconstruction

Using the feature matching, we can triagulate the points. The fundamental matrix gives the relationship between the feature points. We used RANSAC based 8-point algorithm to find the fundamental matrix.

$$\mathbf{p}_2^T \mathbf{F} \mathbf{p}_1 = 0$$

The Essential matrix can be found using the equation (assuming the same camera intrinsics in both cameras)

$$\mathbf{E} = \mathbf{K}^T \mathbf{F} \mathbf{K}$$

The essential matrix can be decomposed to rotation and translation component.

$$\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T$$

$$\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T$$

$$[\mathbf{t}]_\times = \mathbf{U}\Sigma\mathbf{U}^T$$

where $\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

This gives rise to four possible camera configurations (with W/W^T and $\pm t$). The correct location can be found using the camera configuration in which all the points are in front of both the cameras.

3.3. Camera pose recovery

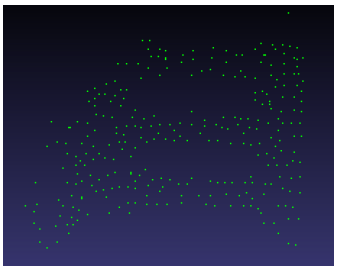
Once the scene is reconstructed using the keyframes, the camera pose can be recovered using Perspective-n-point (PnP) algorithms. By knowing the 3D points from the reconstruction, and its corresponding feature location in any image the camera pose can be recovered using PnP.

3.4. Bundle Adjustment

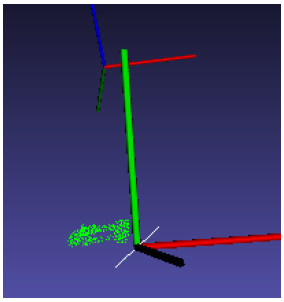
In visual odometry, the current camera pose is obtained by adding the last observed motion to the current detection change. This leads to a superlinear increase in pose error over time. In this section, we look at the techniques we intend to use to correct this pose drift.

One solution is to use bundle adjustment to impose geometrical constraints over multiple frames. The computational cost increases with the cube of the number of frames used for computation. Thus, we limit the number of frames to a small window from the previously captured frames. This approach is called local bundle adjustment.

(a) Temple correspondence using KLT features



(b) Temple reconstruction using keyframes



(c) Recovered camera poses using PnP. The 3D points are obtained using keyframes. The camera pose is found using its corresponding feature in the image

4. Results

So far, we have been working with the datasets available online. The results are illustrated on the Middlebury Temple dataset [2]. We first find the feature correspondences and then remove the outliers using Epipolar constraints. The outliers can be found by using a threshold on distance from epipolar line and along the epipolar line.

We are able to recosntruct the temple structure using the two keyframes (handpicked for now). The results are shown in figure 2b. Once the reconstruction is done, we are able to recover the camera poses using PnP algorithm as illustrated in the figure 2c. We are using OpenCV for feature matching and visualization.

5. Future Work

Until now, we have evaluated the various landmark tracking techniques on the Temple dataset. We intend to test the different techniques across multiple datasets and pick an algorithm that works best.

We will integrate Ceres solver by Google for local bundle adjustment. This would improve the localization of the camera and generate a better tracking of the scene.

If time permits, we also intend to integrate intertial sensors readily available on mobile devices to improve the estimate[8].

References

[1] Szeliski, Richard. *Computer Vision: Algorithms and Applications*. 1st ed. London: Springer-Verlag, 2010. Print.

[2] Temple dataset `http://vision.middlebury.edu/mview/data/`

[3] P. Alcantarilla, J. Nuevo and A. Bartoli, *Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces*, BMVC 2013

[4] F. Fraundorfer, P. Tanskanen and M. Pollefeys *A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles*, ECCV 2010.

[5] D. Nister, O. Naroditsky and J. Bergen *Visual Odometry*, CVPR 2004

[6] O. D. Faugeras and F. Lustman, *Motion and structure from motion in a piecewise planar environment*, IJPRAI, 1988

[7] T. Schops, J. Engel and D. Cremers *Semi-Dense visual odometry for AR on a smartphone*

[8] P. Tiefenbacher, T. Schulze and G. Rigoll *Off-the-shelf sensor integration for mono-SLAM on Smart Devices*, CVPR 2015