An Implementation of a Stereo Visual Odometry Algorithm

Avi Singh

Indian Institute of Technology - Kanpur

avisingh@iitk.ac.in

May 25, 2015

Overview

Introduction

Algorithm

Visual Odometry is the estimation of 6-DOF trajectory followed by a moving agent, based on input from a camera rigidly attached to the body of the agent.

Applications

- Mars Rover (no GPS, unreliable wheel odometry)
- Autonomous driving, wearable electronics, augmented reality, and even gaming.

Problem Formulation

Input

Two Stereo Pairs: I_l^t, I_r^t, I_l^{t+1}, I_r^{t+1}

Output

4×4 Transformation Matrix $\mathbf{T}_{t,t+1}$ between the two frames.

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

$\mathbf{R}, \mathbf{t}$ are the rotation matrix and translation vector, respectively.

Preprocessing: Undistortion and Rectification

Since, the stereo rig is calibrated, it is straightforward to undistort and rectify the images.

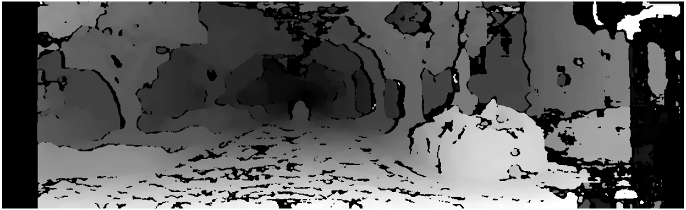
The KITTI dataset used already gives us pre-processed images

Disparity Map

Disparity at a point defined as:

$$d = x_l - x_r \tag{1}$$

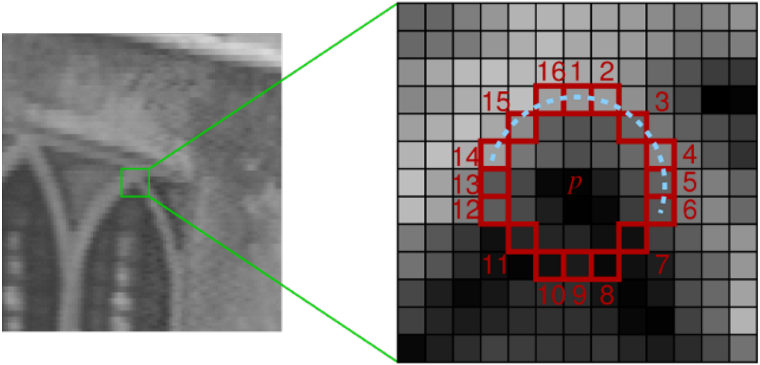
Computed using the Semi-Global Matching Algorithm



Feature Detection

FAST corner detector to detect features in I_l^t, I_l^{t+1}.
Preferred over SIFT, SURF because of its speed.

Figure : Image taken from original FAST paper



Feature Matching

BRISK descriptors are generated on the features detected in the previous step and are then matched by minimizing the sum of absolute differences metric.

Stereo Triangulation

We use the following relation to obtain the 3D coordinates of every feature in $\mathcal{F}_l^t$ and $\mathcal{F}_l^{t+1}$:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{Q} \times \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} \tag{2}$$

where:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & -f \\ 0 & 0 & -1/T_x & 0 \end{bmatrix}$$

The Inlier Detection Step

Scene rigidity is assumed and the consistency matrix **M** is build using the following condition:

$$\mathbf{M}_{i,j} = \begin{cases} 1, & \text{if } || \mathcal{W}_i^t - \mathcal{W}_j^t | - | \mathcal{W}_j^{t+1} \mathcal{W}_i^{t+1} || < \delta. \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The Inlier Detection Step

Selecting a set of matches that are all consistent with each other is same as selecting the maximum clique with **M** as the adjacency matrix.

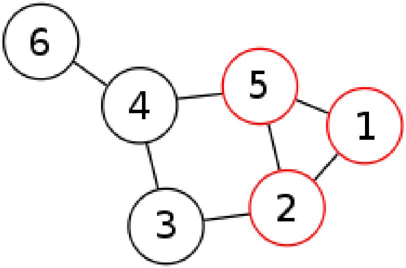


Figure : The largest clique in the above graph contains 1,5,2

The Inlier Detection Step

Since the maximum-clique problem has no known solution in polynomial time, we employ a greedy heuristic to get a sub-optimal solution.

- Select the node with the maximum degree, and initialize the clique to contain this node.
- From the existing clique, determine the subset of nodes v which are connected to all the nodes present in the clique.
- From the set v, select a node which is connected to the maximum number of other nodes in v. Repeat from step 2 till no more nodes can be added to the clique.

Computation of **R** and **t**

In order to determine the rotation matrix **R** and translation vector **t**, we use Levenberg-Marquardt non-linear least squares minimization to minimize the following sum:

$$\epsilon = \sum_{\mathcal{F}^t, \mathcal{F}^{t+1}} (\mathbf{j_t} - \mathbf{P}\mathbf{T}\mathbf{w_{t+1}})^2 + (\mathbf{j_{t+1}} - \mathbf{P}\mathbf{T}^{-1}\mathbf{w_t})^2 \tag{4}$$

Results - To be updated

The End