Visual SLAM on Android
MAD854

Ambrish Rawat
Supervisor: Dr. Amitabha Tripathi

Indian Institute of Technology Delhi

June 30, 2015

Overview

Introduction

- Autonomous navigation in an unknown environment
- Applications – Driverless cars, Undersea Navigation, Navigation on a Planet's surface
- SLAM – an essential tool for robots when GPS is not available
- Different algorithms used for SLAM - Kalman Filter based, FastSlam
- Sensors usually employed - laser sensors, sonar, one or more cameras

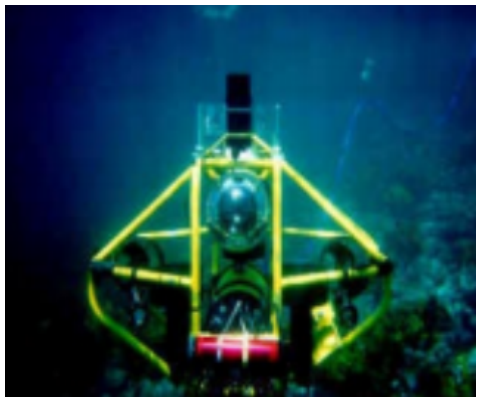


Figure: Undersea Navigation

Figure: Navigation on a Planet's Surface

Motivation

- Develop SLAM on an Android device
- Harnesses the capabilities of various sensors – accelerometer, gravity sensor and magnetic field sensor
- Use the device camera for landmark detection

Problem Description

- The Chicken-and-egg problem (the map or the motion?)
- Correlation between the error in the device pose and the error in the map

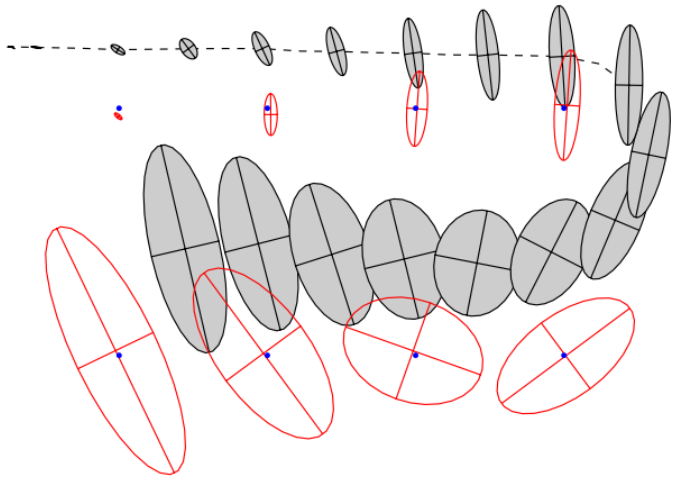


Figure: Before closing the loop

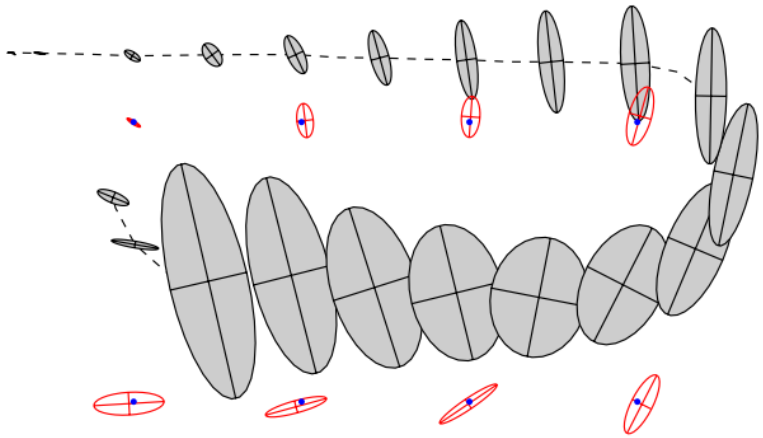


Figure: After closing the loop

Kalman Filter

- State estimation of a linear, discrete-time, finite-dimensional time-varying system

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{1}$$

- The measurement relates to the state as

$$z_k = Hx_k + v_k \tag{2}$$

- w and v are assumed to be white noise

Kalman Filter Solution

- *a priori* estimate (x_{pk}) – estimate at step k given knowledge of the process prior to step k
- *a posteriori* estimates (x_{ok}) – estimate of states after the measurement
- x_{ok} is computed as a linear combination of x_{pk} and a weighted difference between an actual measurement z and a measurement prediction Hx_{pk}

$$x_{ok} = x_{pk} + K(z_k - Hx_{pk}) \tag{3}$$

- K is calculated such that the posteriori ($E(x, x_{ok})$) error covariance is minimised.

Kalman Filter

- a feedback control loop of 'predictor' (or 'time-update') and 'corrector' (or 'measurement-update') equations
- Time-update equations

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_{k-1} \\ P_k &= AP_{k-1}A^T + Q \end{aligned} \tag{4}$$

- Measurement-update equations

$$\begin{aligned} K_k &= P_k H^T (HP_k H^T + R)^{-1} \\ x_k &= x_k + K(z_k - Hx_k) \\ P_k &= (1 - K_k H)P_k \end{aligned} \tag{5}$$

Extended Kalman Filter

- Non-linear assumption (f and h are non-linear functions)

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{6}$$

$$z_k = h(x_k, v_k) \tag{7}$$

- f is referred to as the time-update model
- h is referred to as the measurement model

Extended Kalman Filter

- Time-update equations

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, 0) \\ P_k &= A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \end{aligned} \tag{8}$$

- Measurement-update equations

$$\begin{aligned} K_k &= P_k H_k^T (H_k P_k H_k^T + V_k R_{k-1} V_k^T)^{-1} \\ x_k &= x_k + K_k(z_k - h(x_k, 0) \\ P_k &= (1 - K_k H_k) P_k \end{aligned} \tag{9}$$

- A and W are Jacobians of partial derivatives of f with respect to x and w and respectively
- H and V are Jacobians of partial derivatives of h with respect to x and w and respectively

SLAM

- Use the odometery data to update the current estimate of state (time-update)
- Use re-observation of landmark to improve upon the estimated state (measurement-update)
- New landmarks are added to state as and when they are observed.

Pose Tracking

- Accelerometer based computation of velocity and distance

$$
\begin{aligned}
v[i] &= \frac{1}{2}(a(i) + a(i-1))\Delta t \\
x &= \sum_{i=1}^{n} \frac{1}{2}(v(i) + v(i-1))\Delta t
\end{aligned}
\tag{10}
$$

Coordinate Systems

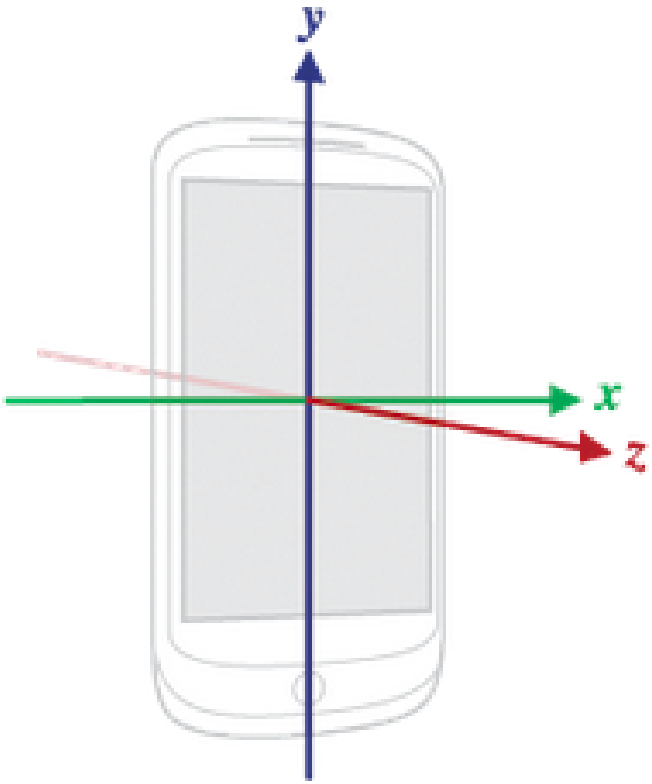


Figure: Device's Coordinate System

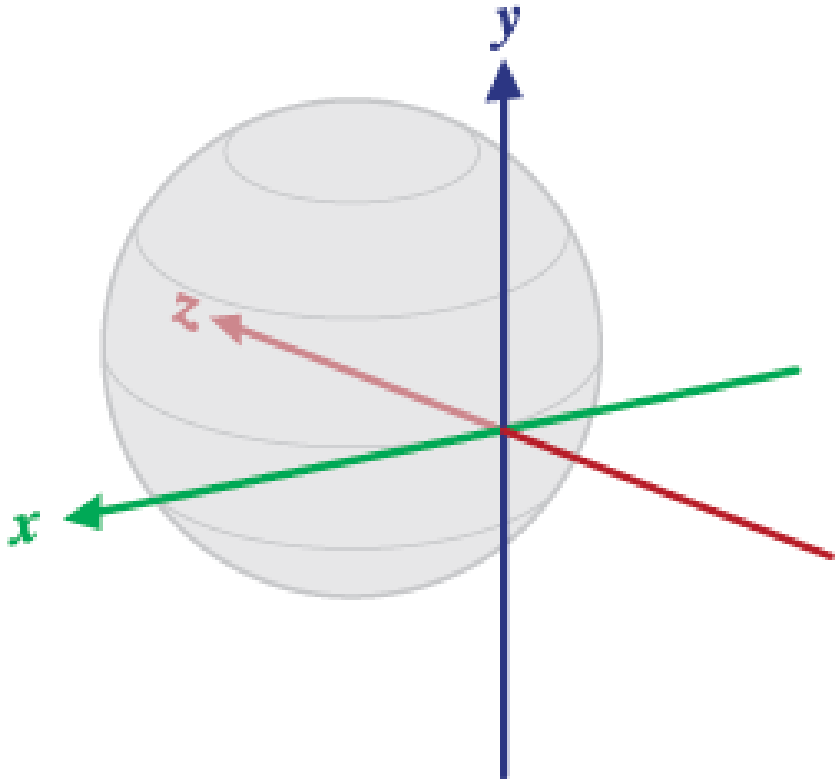


Figure: World Coordinate System

Landmark Detection

- Landmark position required in the Innovation term while updating the state vector
- Innovation -the difference between the estimated device position and the current measurement of the position
- Template matching was used for landmark detection in this work

Landmark Detection

The metric used in this work was,

$$R(x,y) = \sum_{x',y'} (T'(x',y') I(x+x', y+x')) \tag{11}$$

where T' and I' are given by,

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$

$$I'(x+x', y+y') = I(x+x', y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'', y+y'')$$

Figure: Sliding Window

Android Application

- An Android Application **SLAMActivity** was developed using Android SDK 24.0.2
- The app was tested on a Google Nexus 7 (2013) tablet with Android Lollipop 5.1
- QR codes were used as landmarks

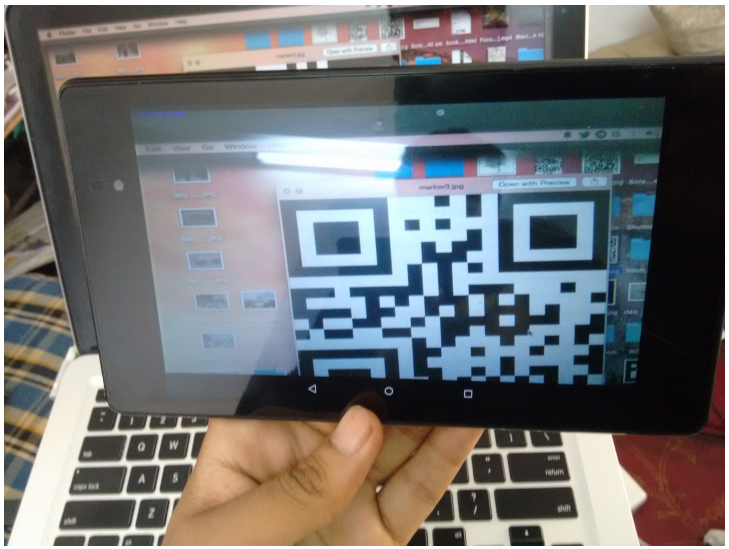


Figure: Google Nexus 7 (2013) displaying the QR code

Prediction and Measurement Model

- Prediction model for Kalman Filter

$$\begin{bmatrix} x + \Delta x \\ y + \Delta y \\ \theta + \Delta\theta \end{bmatrix} \tag{12}$$

- δx δy are obtained by integrating the linear acceleration data
- $\delta\theta$ is obtained from by using the gravity and magnetic field sensor data
- Measurement model for Kalman Filter

$$\begin{bmatrix} r \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} \\ \tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) \end{bmatrix} \tag{13}$$

Algorithm

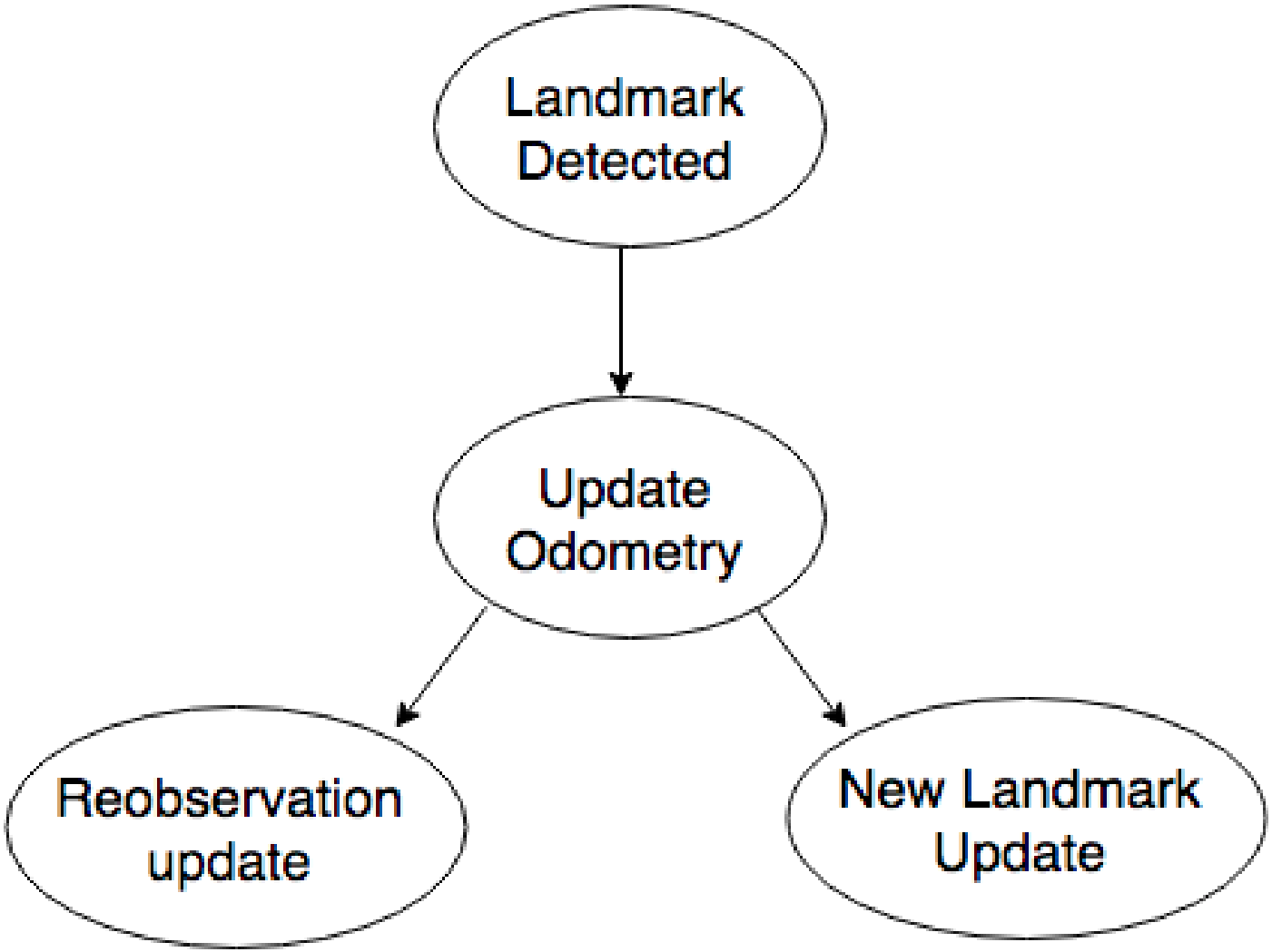
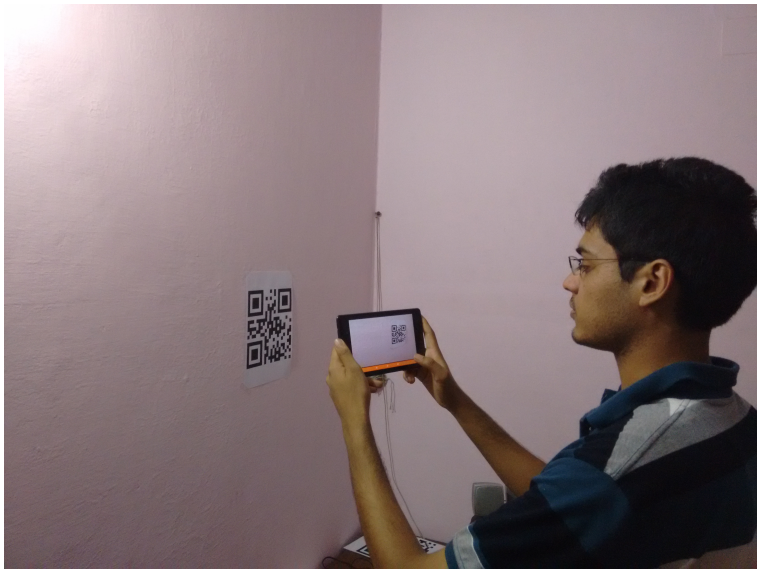


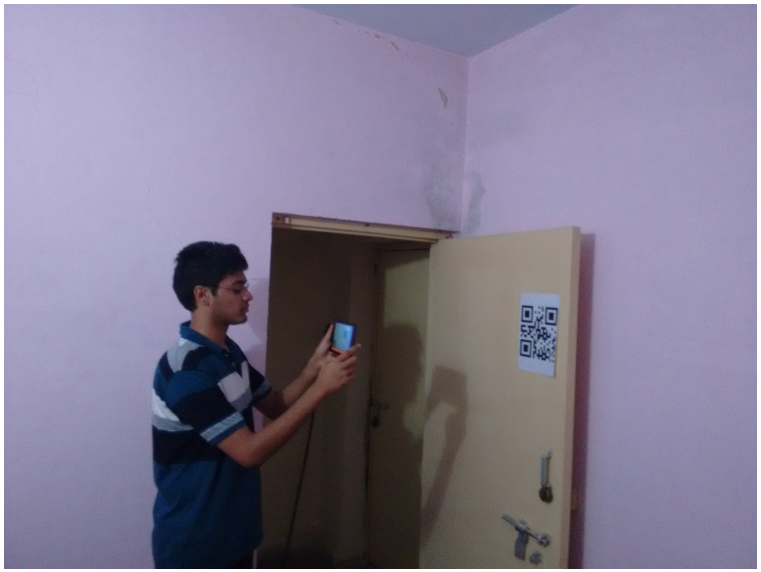
Figure: Processing done for every videoframe

Application Demo

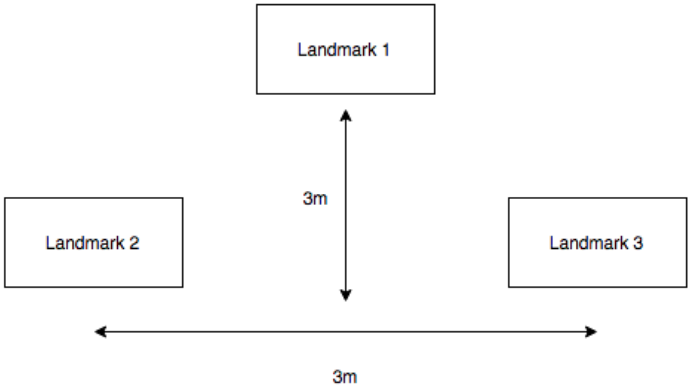
- QR codes were placed at different locations in a room/hall
- The device was moved manually in the room while the application was running on the device
- A csv file of corresponding to the state variables vas updated at each iteration

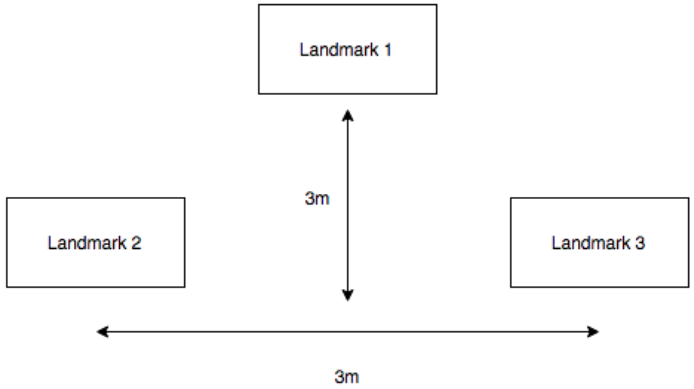
Application Demo





Application Demo





Conclusion

- Device sensors like accelerometers are known to provide highly noisy data. An improper smoothening and integration could have led to erroneous odometery update
- A simplistic landmark detection model could also have influenced the working of the program. In most of the visual SLAM based systems, stereo cameras provide the necessary feature detection and matching for landmarks. Corner detectors are often used as landmarks in these cases which provide a more extensive landmark framework.
- Since there were too few landmarks in this framework the number of SLAM iterations were also less. This could be another source for the deviation observed in the results.

Thank You