Kalman Filter for SVO Scale Estimation

Group D

Tampere University of Technology

joonas.melin@tut.fi

November 14, 2014

Overview

Motivation

- SVO is performing adequately on the copter with small tuning and minor fixes
- The scale and orientation of the visual odometry is still unknown
- Kalman Filter can be applied to estimate scale and orientation, how ever, pose of the copter needs to be estimated first

Premises

- Normal EKF equations for the update, we will list the models used
- State $S = \begin{bmatrix} Q_{copter}^{1x4} & P_{position}^{1x3} & V_{velocity}^{1x3} & s_{svo} & Q_{svo}^{1x4} \end{bmatrix}^T$
- $Q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}$
- $P = \begin{bmatrix} x & y & z \end{bmatrix}$
- $V = \begin{bmatrix} x & y & z \end{bmatrix}$

Prediction

- $S^{t+1} = FS^t + G$
- $F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

- $G = \Delta t B U$
- $U = \begin{bmatrix} \Delta\alpha_{euler}^{1x3} & A_{acceleration}^{1x3} & g_{gravitation} \end{bmatrix}$
- $$B = \begin{bmatrix} -q_x/2 & -q_y/2 & -q_z/2 & 0 & 0 & 0 & 0 \\ q_w/2 & -q_z/2 & q_y/2 & 0 & 0 & 0 & 0 \\ q_z/2 & q_w/2 & -q_x/2 & 0 & 0 & 0 & 0 \\ -q_y/2 & q_x/2 & q_w/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2*q_x*q_y - 2*q_w*q_z & 2*q_w*q_y + 2*q_x*q_z & 0 \\ 0 & 0 & 0 & 2*q_w*q_z + 2*q_x*q_y & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2*q_y*q_z - 2*q_w*q_x & 0 \\ 0 & 0 & 0 & 2*q_x*q_z - 2*q_w*q_y & 2*q_w*q_x + 2*q_y*q_z & q_w^2 - q_x^2 - q_y^2 + q_z^2 & -1 \end{bmatrix}$$

Acceleration update

- Measurement is the accelerations measured by the IMU
- Measurement model $A_{imu} = R(Q_{copter})A^{3x1}_{gravitation}$
- R is an quaternion rotation matrix derived from the state
- Jacobian $J = \frac{\partial A_{imu}}{\partial \begin{bmatrix} Q^{1x4}_{copter} & P^{1x3}_{position} & V^{1x3}_{velocity} \end{bmatrix}}$

GPS update

- Measurement is the position measured by the GPS
- Measurement model $P_{gps} = P_{position}$
- Jacobian $J = \frac{\partial P_{gps}}{\partial \begin{bmatrix} Q_{copter}^{1x4} & P_{position}^{1x3} & V_{velocity}^{1x3} \end{bmatrix}}$

Velocity update

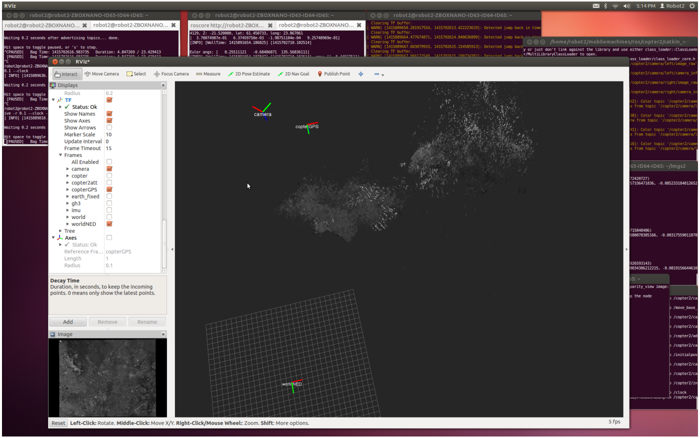
- Measurement is the speed derived from the GPS
- Measurement model $V_{gps} = V_{velocity}$
- Jacobian $J = \frac{\partial P_{gps}}{\partial \begin{bmatrix} Q_{copter}^{1x4} & P_{position}^{1x3} & V_{velocity}^{1x3} \end{bmatrix}}$

SVO scale update

- Measurement is the velocity measured by the SVO
- Measurement model $V_{svo} = R(Q_{svo})(V_{velocity}/s_{svo})$
- Jacobian $J = \dfrac{\partial V_{svo}}{\partial \begin{bmatrix} Q_{copter}^{1x4} & P_{position}^{1x3} & V_{velocity}^{1x3} & s_{svo} & Q_{svo}^{1x4} \end{bmatrix}}$

SVO orientation update

- Measurement is the orientation quaternion measured by the SVO
- Measurement model $Q_{measured} = Q_{svo} Q_{copter}$
- Jacobian $J = \frac{\partial Q_{measured}}{\partial \begin{bmatrix} Q^{1x4}_{copter} & P^{1x3}_{position} & V^{1x3}_{velocity} & s_{svo} & Q^{1x4}_{svo} \end{bmatrix}}$


RViz*
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale
Update Interval
Frame Timeout
Frames
All Enabled
camera
copter
copter2att
copterGPS
earth_fixed
gh3
imu
world
worldNED
Tree
Axes
Reference Fra...
Length
Radius
Decay Time
Duration, in seconds, to keep the incoming points. 0 means only show the latest points.
Add
Remove
Rename
Image
camera
copterGPS
worldNED
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
5 fps

- Modeling the uncertainties for the SVO
- Implementation on ROS
- Connecting the filter and SVO (rudimentary work started)
- Delay compensation for the GPS measurement
- Isolating update steps to smaller submatrices

- Antti: Making the SVO work on copter data
- Tuomas: Making the SVO work on ground vehicle
- Joonas: Implementing the scale and orientation estimation