Vilnius University
Institute of Mathematics and Informatics
L I T H U A N I A

INFORMATICS ENGINEERING (07 T)

VISION-AIDED LOCALIZATION AND NAVIGATION OF AN AUTONOMOUS HELICOPTER

Rokas Jurevičius

October 2017

Technical Report MII-DS-07T-17-10

VU Institute of Mathematics and Informatics, Akademijos str. 4, Vilnius LT-08663, Lithuania
www.mii.lt

Abstract

Conventional UAV (abbr. Unmanned Air Vehicle) auto-pilot systems uses GPS signal for navigation. While the GPS signal is lost, jammed or the UAV is navigating in GPS-denied environment conventional autopilot systems fail to navigate safely. UAV should estimate it's own position without the need of external signals. Localization, the process of pose estimation relatively to known environment, may solve the problem of navigation without GPS signal. Downward looking camera on a UAV may be used to solve pose estimation problem in combination with visual odometry and other sensor data. In this report a vision-based particle filter application is proposed to solve GPS-denied UAV localization. The application uses visual odometry for motion estimation, correlation coefficient for apriori known map image matching with aerial imagery, KLD (abbr. Kueller-Leiblach distance) sampling for particle filtering. Research using data collected during real UAV flight is performed to investigate: UAV heading influence on correlation coefficient values when matching aerial imagery with the map and measure localization accuracy compared to conventional GPS system and state-of-the-art odometry.

Keywords: Particle Filter Localization, GPS-Denied Navigation, Visual Odometry, KLD Sampling, Correlation Coefficient

Contents

1 Introduction

GPS signal used for UAV navigation is vulnerable to signal jamming and spoofing [KSBH14]. Encoded military standard GPS signals are safe against spoofing, although they are still vulnerable to jamming and are not publicly available. Conventional autopilot systems fail to navigate safely since there is no available alternatives to GPS positioning. UAV should estimate it's own position without the need of external signals. Localization, the process of pose estimation relatively to known environment, may solve the problem of navigation without GPS signal. Particle filters have solved localization problem for autonomous robots [Thr02] using laser scanners and panoramic vision [ATD05]. Mixed particle filter algorithm may address robot localization issue with better precision over long distance and long duration flights. Algorithm enables navigation in GPS-denied environment and contributes to UAV safety as a GPS backup system.

2 Vision-based particle filter localization

This paragraph shows the theory behind particle filter localization and algorithms used in particular steps particle filtering. Fig. **??** shows the schema of the proposed algorithm. Each step is described in subsections in details.

2.1 Particles

Particle is a hypothesis for the aircraft's possible position in map. A number of particles is maintained in the algorithm to evaluate more than one possible location of the aircraft and propagate the possibilities over time. Each particle is assigned an image similarity value on time t

$$b_t = b_{t-1}\frac{R_t + 1}{2}$$

that is calculated using UAV image similarity value R_t with the map image on the particle location. Initial particle density value is assigned $b_0 = 1$. Particles are also assigned weight value which is used during sampling. The particle weight w_i is calculated by normalizing all probabilities $w_{i,t} = \frac{b_{i,t}}{\sum_{j=0}^{n} b_{t,j}}$, where n is the number of particles, i is single particle index, t is time of current iteration.

2.2 Particle sampling

Sampling is the stage of the Particle Filter when particles are re-sampled according to their weight. Each iteration re-samples particles to find the most plausible UAV location over time. KLD-sampling technique was selected due it's to ability to dynamically adjust particle count thus reducing computational costs when it is not necessary. Sampling uses Kueller-Leiblach distance [Fox01] to calculate minimal number of particles that keeps particle probability distribution the same. The technique has shown good results against other sampling techniques on simulated flight data - it provides the same localization accuracy, but dynamic particle count allows to decrease computational times up to 1.7 times [JMT16].

2.3 Odometry

Visual odometry is the process of calculating aircraft (or robot) motion from camera images. In this setup monocular SVO [FPS14] (abbr. Semi-direct Visual Odometry) with downward facing camera is used to calculate motion. SVO algorithm was selected due to high accuracy compared with other algorithms and real-time execution on embedded is possible due to semi-dense algorithm implementation.

SVO algorithm was selected because of more accurate positioning compared to other algorithms and real-time execution on embedded platforms. SVO algorithm was shown to run 55 frames per second on an embedded flight computer.

2.4 Motion model

Motion model is used for dead-reckoning of the UAV pose from odometry data (visual and movement speed sensors). The UAV pose may be described using six parameters $\langle x, y, z, \theta_{roll}, \theta_{pitch}, \theta_{yaw} \rangle$ in space relative to the known environment. Parameters x, y and z are the coordinate locations in 3D environment. Parameter z is equivalent to altitude, which can be measured using sensors (barometer, laser) with relatively high precision.
In the case of localization in orthophoto map the altitude is only required for image pixel scaling, so it can be ignored during localization. Roll and pitch angles are required for the calculation of camera relative elevation angle and image center on the map. Those parameters can be ignored by using camera gimbal hardware in the case if camera is configured to always look downward. The search space thus is narrowed down to pseudo-planar movement using only three parameters (see fig. 1), where aircraft pose $P_t = \langle x, y, \theta_{yaw} \rangle$, where θ_{yaw} is UAV heading angle (UAV platform angle to North). Figure 1 shows planar movement of the particle with appriori position $\langle x, y, \theta \rangle$ and posteriori position $\langle x', y', \theta' \rangle$. Particle movement is described as translational movement $\hat{\delta}_{tran} = \delta_{tran} + \epsilon_{tran}$, where

- $\hat{\delta}_{tran}$ is planar movement with an extra noise
- δ_{tran} is measured movement change measured by sensors (usually odometry)
- ϵ_{tran} is additional random noise value

Rotational movement is described as $\hat{\delta}_{rot} = \delta_{rot} + \epsilon_{rot}$, equation explanation is analogues to translational movement.

The pose update can be calculated after new sensor data using these equations [TBF05]:

$$x' = x + \alpha_1 \hat{\delta}_{tran} cos(\theta_{yaw} + \alpha_3 \hat{\delta}_{rot})$$

$$y' = y + \alpha_2 \hat{\delta}_{tran} sin(\theta_{yaw} + \alpha_3 \hat{\delta}_{rot})$$

$$\theta'_{yaw} = \theta_{yaw} + \alpha_3 \hat{\delta}_{rot}$$

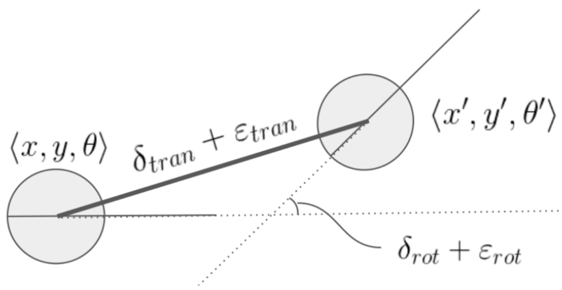

Figure 1: Simplified planar motion model for UAV

, where:

- x', y' and θ'_{yaw} are posterior UAV location relative to the orthophoto map
- α_n - measurement noise scale coefficients, selected manually
- $\hat{\delta}_{tran}$ - translational (movement speed) measurement with measurement noise ϵ_{tran}, obtained:
$$\hat{\delta}_{tran} = \delta_{tran} + sample\_normal(\epsilon_{tran})$$
- $\hat{\delta}_{rot}$ - rotational (heading angle) measurement with measurement noise ϵ_{rot}, obtained:
$$\hat{\delta}_{rot} = \delta_{rot} + sample\_normal(\epsilon_{rot})$$
- $sample\_normal$ is Gaussian distribution sampling function:
$$sample\_normal(\epsilon) = \epsilon \cdot gaussian(0, \tfrac{1}{3})$$

2.5 Particle propagation

This step of the Particle Filter uses sensor data, odometry and motion model to propagate the particles after re-sampling. Propagation moves the old re-sampled particles into their current locations according to movement that happened since last Particle Filter iteration.

2.6 Particle map matching

Template matching technique is used to match image viewed by the camera and cropped image from map on the particle pose. Matching is done using monochrome gray scale images to make the matching faster. The pixel gray scale value y is calculated using formula from OpenCV library [BK08]:

$$y_{x,y} = 0.299r_{x,y} + 0.587g_{x,y} + 0.114b_{x,y}$$

, where $r_{x,y}$, $g_{x,y}$ and $b_{x,y}$ are the respective red, green and blue pixel values on image x and y coordinates. Normalized correlation coefficient (CCOEFF) from OpenCV [BK08] library was used to calculate image similarity R_t on time t between camera image T and cropped map image I:

$$R_t = \frac{\sum_{x=0}^{w}\sum_{y=0}^{h}(T'(x,y)\cdot I'(x,y))}{\sqrt{\sum_{x=0}^{w}\sum_{y=0}^{h}T'(x,y)^2\cdot\sum_{x=0}^{w}\sum_{y=0}^{h}I'(x,y)^2}}$$

, where

- $T'(x,y) = T(x,y) - \frac{\sum_{x'=0}^{w}\sum_{y'=0}^{h}T(x',y')}{w\cdot h}$
- $I'(x,y) = I(x,y) - \frac{\sum_{x'=0}^{w}\sum_{y'=0}^{h}I(x',y')}{w\cdot h}$
- w, h are the image dimensions (width and height).

2.7 UAV Pose Estimation

True location of the UAV is calculated by recursively estimating particle belief density values as described in [TBF05]. Belief is calculated for each particle as conditional probability

$$bel(P_t) = p(P_t|P_{0:t-1}, m_{1:t-1}, b_{1:t-1})$$

, where

- P_t - predicted aircraft pose on time t
- m_t - sensor data (may be IMU, barometer, wind speed and other data used for dead-reckoning) on time t
- b_t - previously described particle probability value on time t

The belief is a probability on location P_t, conditioned on all previous sensor data and all particle probability density values. The particle with highest belief is considered to be the true UAV pose for current iteration.

3 Experimental results

This section describes experimental environment, hardware components used for data collection and obtained results

3.1 Experimental setup

A fixed-wing UAV was used to collect aerial imagery and sensor data during 1 km flight. Basler acA640-120uc industrial camera with global shutter was used to collect aerial imagery alongside with other sensor data provided from UAV flight controller. Images was recorded in 640x480 resolution at 90 FPS. Data was recorded using MPEG2-TS video format and sensor data was recorded as meta-data alongside the video stream. The video playback allows data to be read with the same timing as it was recorded on UAV. Map used for matching was downloaded from Google Maps [Sve10] using highest available zoom level. Initial particle count for the particle filter was set to 500.

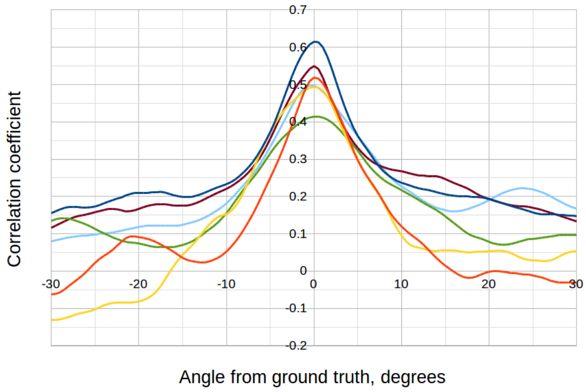

Figure 2: Correlation coefficient values when matching map with aerial imagery on all heading angles.

3.2 UAV Heading impact on correlation coefficient

This section investigates magnetometer error impact on correlation coefficient since it was noticed during localization experiment. Typical magnetometers used in UAV's may contain noise in measuring heading direction. Fig. 2 presents correlation coefficient values for 6 images captured during real flight and matched with according orthophoto map images. Table 1 contains average similarity change values versus heading change. Data from table 1 suggests that +/- 2 degrees of heading angle error can be ignored, because it affects correlation coefficient only up to 10% on average.

Table 1: Heading change impact on similarity coefficient

Heading change, °	Average similarity change, %
+10	67.78
+5	34.59
+2	9.38
-2	10.15
-5	35.49
-10	62.08

3.3 Localization accuracy

In this section we will evaluate localization accuracy compared with conventional GPS positioning system and pure visual odometry positioning. Accumulated odometry error correction is expected when using particle filter in combination with odometry. Flight trajectory reconstruction using odometry and particle filter localization is presented in fig. 3. Trajectory errors in meters are presented in fig. 4, the plot shows that odometry suffers from cumulative errors as it was introduced in the introduction of the report.

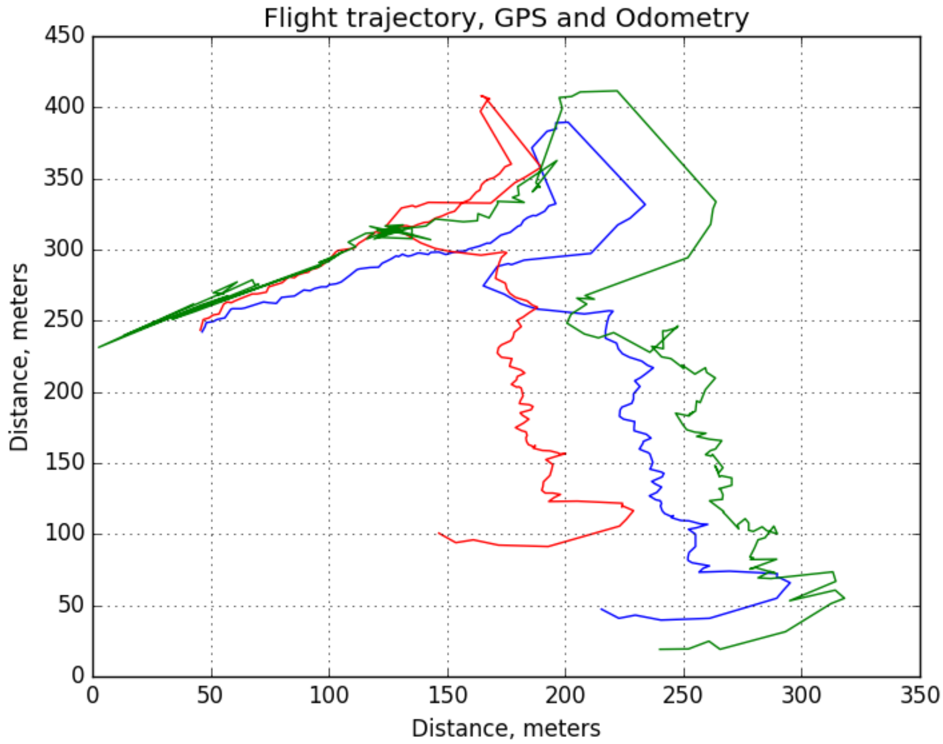

Figure 3: Flight trajectory reconstruction using odometry (red), particle filter (green) and conventional GPS sensor (blue).

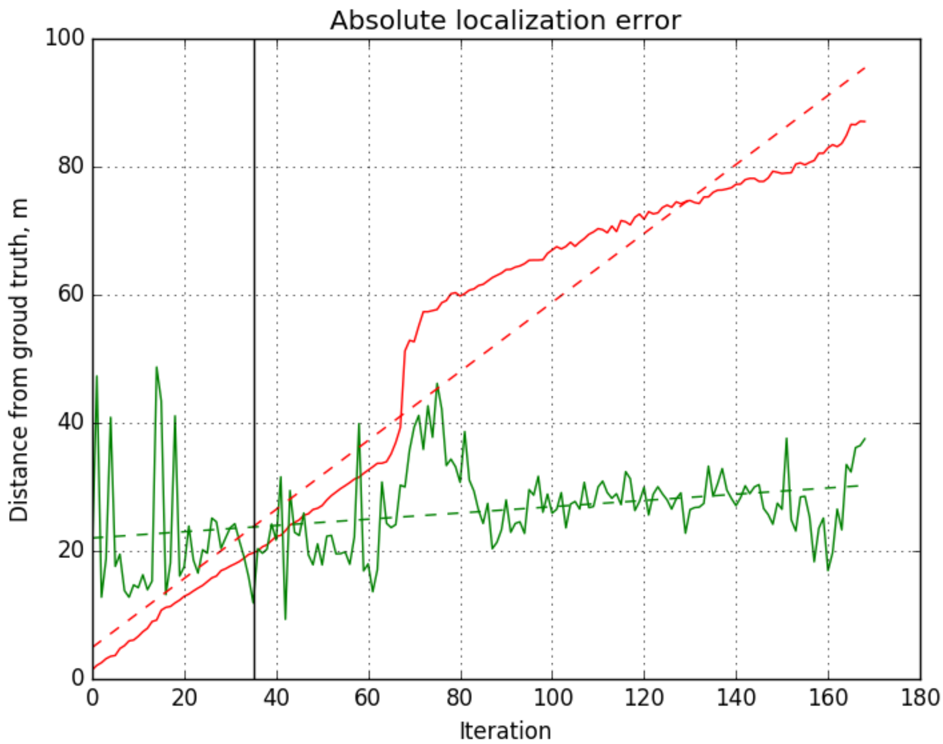

Figure 4: Particle filter localization and visual odometry absolute errors.

The dashed lines are the trend-lines of the errors, the vertical line show the breaking point of the trend-lines at 35 seconds flight time. Since the breaking point of accuracies shows that proposed algorithm adds a lot less errors during long time flights. Additional experiments are required to validate whether errors won't add up after longer flights. Particle filter localization was able to keep error values in around 50 meter range. After the 1 kilometer flight the final error was reduced by a factor of 2 compared to localization from visual odometry only. Proposed algorithm error trend-line slope is reduced by a factor of 11 times compared with visual odometry.

4 Conclusions and future work

This report analyses an application of particle filter for UAV localization in previously known orthophoto map using images from downward facing camera on the UAV platform. The obtained results concludes:

- Flight heading error +/- 2 degrees causes correlation coefficients errors in up to 10% range. Particle filter execution was done with +/- 5 degree uncertainty in UAV heading.
- Experiment on real flight data shows that particle filter is able to reduce the slope of accumulating errors with a factor of 11 times compared to visual odometry.
- At the end of experimental flight, particle filter localization allowed to improve position precision by a factor of 2 compared with position from odometry data only.

Future work in the field of this report would include algorithms that could reduce the impact of errors in sensor measurements. An alternative image similarity coefficient based on deep learning can be proposed in replacement of correlation coefficient to improve localization accuracy of the algorithm.

5 References

References

[ATD05] Henrik Andreasson, André Treptow, and Tom Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3348–3353. IEEE, 2005.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[Fox01] Dieter Fox. Kld-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, pages 713–720, 2001.

[FPS14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.

[JMT16] R Jurevicius, V Marcinkevicius, and V Taujanskas. Comparison of image similarity functions and sampling algorithms in vision-based particle filter for uav localization. 2016.

[KSBH14] Andrew J Kerns, Daniel P Shepard, Jahshan A Bhatti, and Todd E Humphreys. Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, 31(4):617–636, 2014.

[Sve10] Gabriel Svennerberg. *Beginning Google Maps API 3*. Apress, 2010.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. 2005.

[Thr02] Sebastian Thrun. Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 511–518. Morgan Kaufmann Publishers Inc., 2002.