Particle Filtering based optimization applied to 3D model-based estimation for UAV pose estimation

Nuno Pessanha Santos & Victor Lobo
Portuguese Navy Research Center (CINAV)
Portuguese Navy (MGP)
2810-001, Almada, Portugal
nuno.pessanha.santos@marinha.pt, vlobo@novaims.unl.pt

Alexandre Bernardino
Institute for Systems and Robotics (ISR)
Instituto Superior Técnico (IST)
1049-001, Lisboa, Portugal
alex@isr.ist.utl.pt

***Abstract*—To estimate the pose of an unmanned aerial vehicle (UAV) during the landing process aboard a ship is used a vision system based on a standard RGB camera using one workstation for processing data. It is used a ground-based vision system to allow the use of a small size and weight UAV, due to the low computer requirements onboard. The resampling step in the particle filter takes a decisive role in the obtained pose estimation, and ten traditional resampling steps are tested and compared with a developed resampling strategy inspired by the genetic algorithms. The obtained results show that a classical resampling step is not sufficient in this problem and gets easily stuck in local minima. Those local minima are originated by the high dimensional search space and from the employed observation likelihood metric, which is very dependent on the UAV geometry which generates ambiguity in the pose estimate for complementary poses (poses with large wing pixel overlap, where the majority of the UAV area is concentrated). Results show that the errors obtained are lower and much more compatible with the requirements for this kind of problems when compared to the common existing resampling strategies.**

***Keywords*—Model-Based Pose Estimation, Particle Filters, Autonomous Vehicles, Computer Vision.**

I. Introduction

Portugal has to control approximately 41335 km^2 of territorial waters, and the fast patrol boats (FPB) have an important role in this mission. Their efficiency can be significantly improved by the support of unmanned aerial vehicles (UAVs), extending the range and capability of operation (e.g. able to transmit georeferenced video in real-time to the FBP). Having less human intervention in the most difficult maneuvers (e.g. landing and take-off) increases the system reliability and is no longer needed to use trained and certified UAV pilots. The available landing site in this kind of ships is usually a small area of 5x6m (stern section), and this limits the used UAV payload [1], [2]. The used UAV model has 5 kg maximum take-off weight (MTOW), 180 cm of wingspan, 150 cm of length and one autonomy of approximately 2 hours.

The vast majority of the developed systems in this field are based on the UAV [3], [4], not being usually considered land-based systems [5]. A ground-based vision system [1], [2] makes it possible to use more processing power, allowing the use of more computationally intensive methods. This choice permits the use of simpler UAVs with commercial autopilots not designed exclusively for this task and with low processing requirements. The only requirement is the ability to execute the trajectories given by the ground control station (GCS) for command and control (C2) after the relative position estimate between the camera and the UAV is obtained. The use of computer vision makes the system tolerant to GPS jamming and able to operate at locations where the signal can be lost or imprecise.

The vision system localization brings, however, several challenges since the moving platform is at sea and is affected by the meteorological conditions. The objective is to obtain in real time, using a ground-based vision system, the airplane 3D position and orientation allowing a successful landing maneuver. In this kind of problems, we can apply a particle filtering framework that represents the distribution of an object's pose as a set of weighted hypotheses (particles) that provides us a confidence measure.

The particle filter is a computationally expensive method, in particular, the particle generation (in this case the object rendering) and the particle evaluation. We will focus on applying a particle filtering framework to a 3D model-based pose estimation problem, trying to retrieve as much information as possible from existing resampling schemes. We compare the performance of ten traditional schemes with a resampling step inspired by the evolution strategies present in genetic algorithms [6], [7] (Genetic Algorithm based Framework – GAbF), to avoid sample impoverishment.

This paper is organized as follows. In section 2 the overall system description is made. In section 3 the adopted particle filter formulation is explained. In section 4 we present some experimental results, comparing the performance obtained in the tested resampling strategies. Finally, in section 5 we present the conclusions and provide directions for further research work.

II. Overall System Description

Landing on a ship is a tough task, especially in small ships that are very sensitive to the weather conditions (e.g. wave and the wind) and have a small area available for landing purposes (as seen in Figure 1). We are using a net-based retention system (Figure 2) that guarantees the safe landing of the UAV. The retention system must be adapted to each

978-1-5090-5278-3/17/$31.00 ©2017 IEEE

Fig. 1. UAV landing illustration.

Fig. 2. Landing area – Experimental tests.

case ensuring the minimal mechanical injury to the UAV and the platform.

The proposed method (Figure 3) is divided into three major parts [1], [2]:

- **Target Detection** – Consists of searching in the image for regions of interest (ROI). This ROI represents an image area that may contain an object classified as the UAV to land;
- **Pose estimation** – Consists of particle initialization (using a pre-trained database of the UAV in multiple poses), particle evaluation (rank each particle by likelihood) and a local optimization (particles are resampled and optimized to best fit the object appearance in the image);
- **Propagation** – The best particle on each iteration is considered the UAV pose estimation for that time instant, and the particle propagation between frames is made adding noise to each one of them to describe the particle uncertainty.

The target detection stage is critical since we are operating in an outdoor environment (Figure 4) and we need to have illumination invariance. The presence of other objects in

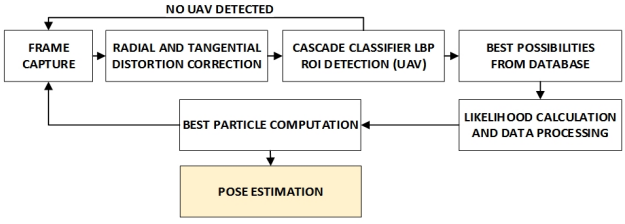

Fig. 3. Simplified system description.

Fig. 4. Outdoor real UAV images (*examples*).

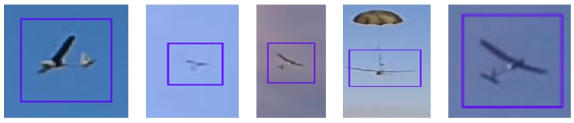

Fig. 5. Obtained ROI using a cascade classifier (*LBP*).

the image can affect the system performance and reliability. Currently, this initial UAV detection is made using a trained local binary pattern (LBP) cascade classifier [1], [8], ensuring that we have a region of interest (ROI) given by the classifier (Figure 5) with small error percentage [1], [2].

In the pose estimation stage, the particle initialization is achieved using the detected ROI in the previous target detection stage, obtaining the oriented bounding box (OBB) that contains the airplane corner points applying the FAST feature detector [9]. Then the OBB is compared with a pre-trained database of UAV OBB in multiple poses that was generated using multiple UAV synthetic images in different poses. The difference between the angle and the aspect ratio of the observation and database is calculated online using the Euclidean distance, and all the poses with a satisfactory score will be the initial dataset of the particle filter. We assume that the object's points are all at the same depth (Z coordinate), projected in a plane parallel to the image. The Z coordinate can be computed by the relationship between the OBB areas and depth. The X and Y coordinates are calculated by the relation between the coordinate of the center of the observation OBB, the obtained Z coordinate and the intrinsic camera parameters focal length f and camera center coordinates C (obtained by calibration). A correct camera calibration step is essential to ensure precision in system performance.

The working environment is critical and will affect many system operation choices namely the selection of the likelihood metric. If we are operating in an outdoor environment, we need to select a metric invariant to illumination changes. Towards this goal, the particle evaluation, in this case, is achieved using one color likelihood metric [1], [2].

In the color likelihood the histograms (as seen in Figure 6) are obtained in the RGB color space (12 bins for each color — B = 12), and the distance between them are calculated using the *Bhattacharyya* similarity metric:

$$L_{color} = 1 - \Sigma_{b=1}^{B} \sqrt{\mathbf{h}^{inner}(b) \cdot \mathbf{h}^{outer}(b)} \qquad (1)$$

where $\mathbf{h}^{inner}$ is the inner histogram, $\mathbf{h}^{outer}$ is the outer histogram and b is the respective histogram bin. Since we are not using any information on the likelihood metric that is directly affected by the illumination changes, we can guarantee robustness to this factor but not for example for the existence of another similar object in the captured frame. For this, we need to adopt other methods to be able to confirm that we are detecting the UAV and for this purpose, we use a LBP classifier to obtain a ROI to be analyzed minimizing the obtained error, as described before. More tests need to be made to characterize the classifier performance accurately.

Fig. 6. An example of inner (*black*) and outer (*blue*) histograms.

In this article, we will focus on the local optimization (inside the Pose estimation stage) that is based on a particle filtering framework. The adopted local optimization framework will be the primary focus of section 3, and the obtained experimental results will be shown in section 4.

III. Implementation Description

The developed approach is based on the evolution strategies present on the genetic algorithms [6], [7], avoiding sample impoverishment. Crossover and mutation operators are adopted, increasing the filter performance and decreasing the number of needed particles for object pose estimation. Particle filters use multiple hypotheses for estimation, which usually works well for non-linear dynamics and observation models.

A. Resampling

The resampling consists in selecting new particle positions and weights such that the discrepancy between the resampled weights is reduced, i.e. eliminating particles with low importance weights and by multiplying particles having high importance weights. The particle filters have several sources of error, that can be summarized in the following points:

- The number of particles is finite, and the resampling leads to approximation errors;
- The resampling step can result in loss of solution diversity (the variance of the particles decrease), principally if we do not have sensor reads;
- Divergence between the system distribution and the proposal distribution (that generates the particles) that can lead to particle degeneracy (only a few of the particles will have significant weight);
- When the number of particles is low compared with the state space dimension, we can have particle deprivation (accidentally discard all particles near the correct state during the resampling step).

If we just introduce a resampling step between successive iterations the parameter space is explored only in the initial iteration, reducing the obtained solution diversity. After t iterations the posterior density will contain one single value, being a poor approach to the problem. To deal with this, we can add some artificial dynamic noise between iterations [10]:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \varphi \tag{2}$$

where φ is artificial noise that can have constant variance, a decreasing variance in time or have another kind of rules developed for the specific implementation. The amount of added noise must be tuned to decrease the needed particle filter convergence time. We adopt three different approaches in our experimental results:

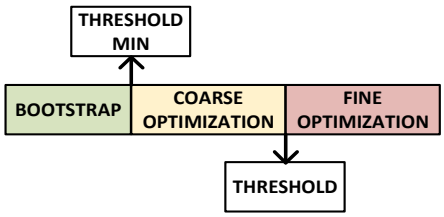

Fig. 7. Particle filter optimization phases.

- **Constant variance noise (Noise)** — Between successive iterations, after the resampling step, is added noise with constant variance;
- **Three discrete phases noise (3Phase)** — Each different phase has a constant variance (decreasing from phase 1 to 3) and is executed n times;
- **Decreasing variance noise (Iterative)** — The variance decreases after each iteration, and is performed until it reaches a minimum value.

B. Genetic Algorithm based Framework

The GAbF operates in three phases (as seen in Figure 7): Bootstrap, Coarse Optimization and Fine Optimization.

In the Bootstrap phase the best N possibilities obtained by comparison with the database are collected in a list (Top N particles). The likelihood of each particle is evaluated and stored in the list. The best M particles are stored in an auxiliary buffer (Top M). The particles with weight very close to zero (below δ = 0.01) are eliminated and replaced with a random particle selected from the Top M buffer, added to Gaussian noise of covariance Σ_B. At this point, all particles have likelihood above δ. In the experimental results using the other tested resampling strategies, the same Bootstrap phase was used to make the particle filter initialization.

Then, we run up to 10 improvement steps. In each step, all particles are evaluated and compared to those in the Top M. If the obtained weight is higher, the Top M is updated. If there are at least two particles in the Top M with likelihood bigger than "Threshold min", the bootstrap phase ends. Otherwise, each particle is perturbed with Gaussian noise with covariance Σ_B. If after 10 of these improvement steps we do not have two particles above "Threshold min", the bootstrap process is restarted up to a maximum of 3 restarts. In our experiments, we have noticed that the occurrence of restarts is very rare.

The coarse optimization phase begins when at least two particles have a weight higher than "Threshold min". At any stage of the coarse and fine optimization steps, the best two particles have a significant role in the optimization process because they will provide the *chromosomes* for an approach inspired by genetic algorithms.

Each particle in the Top 100 list coming from the bootstrap process is analyzed. If the particle is the best one, it is perturbed with some Gaussian noise. If the particle weight is smaller, the best two particles are combined using a crossover operation to create a new particle. The crossover operation consists in the random selection of attributes from the state

vector of the original particles (3D position and the object orientation). To half of the particles generated by crossover is applied a soft mutation by adding Gaussian noise to the result. Together these rules allow a focused particle diversity, simultaneously converging to the best solution and avoiding possible local minima. The process stops when at least two of the particles are above value "Threshold". If this does not happen in 10 iterations, the pose optimization filter returns to the bootstrap phase automatically.

The fine optimization phase is analogous to the coarse optimization phase but the Gaussian noise variance applied in mutation is lower, to make a fine-tuning of the estimated pose. The fine optimization phase ends after five iterations.

We tested ten traditional resampling schemes namely the [11]–[22]: stratified, systematic, residual, residual systematic, optimal, reallocation, metropolis, minimum sampling, multinomial and branching. In the next section, the GAbF algorithm will be compared with these traditional resampling methods.

IV. Results

In this section, we show results from the implemented framework. With real images we do not have ground truth information, results are quantitatively evaluated using synthetically generated frames with ground truth. The method was implemented in C++ on a 2.40 GHz Intel i7 CPU and NVIDIA GeForce GT 750M.

A. Quantitative Performance Evaluation – Traditional Methods

1) Translation Error: Figure 8 to Figure 17 show the obtained translation error from the traditional resampling strategies. The landing area is an irregular area of 5x6 meters, so we need to guarantee a minimum translation error of 1 meter to ensure a safe landing.

As we can see from analysis of the obtained Figures, the translation error decreases with proximity, obtaining a mean value at 5 meters between 0.5 and 0.6 meters for all the tested resampling and noise add strategies. This resolution in translation is clearly achieved, guaranteeing a UAV good estimation in 3D position across the range of distances covered in this test. The accuracy of the system can be increased reducing the number of outliers (outliers are currently less than 5% of the cases) by using the UAV dynamic model in a temporal filtering and data association framework.

2) Rotation Error: Figure 18 to Figure 27 show the obtained rotation error, one combined histogram was obtained for each resampling and noise add strategies. As we can see from the analysis of the Figures, the results are almost all the same independently of the resampling and noise add strategy adopted. This is, we have the vast majority of the estimation error near zero and 180 degrees (the complementary pose). This happens mainly due to the UAV design where the complementary poses achieve a higher correspondence weight and become easily trapped in a local minimum. Since we are operating in a high dimensional space, there is ambiguity between complementary poses, and the pose estimation is achieved with a significant error. In the iterations of the particle filter with added artificial noise, the particles in the previous iteration are moved to a different state even if they are the best ones. Therefore, the highest likelihood of the sample set may decrease along the iterations. On the contrary, the GAbF based method keeps the best particles intact between consecutive iterations, so the highest likelihood is assuredly never decreasing.

As we can see from the obtained Figures, the mean value and the median will be near the center of the obtained histograms since the majority of the data are located near 0 and 180 degrees. The reallocation resampling has the highest performance in the obtained results, but still far from the desired for the system.

B. Quantitative Performance Evaluation – GAbF

Figure 28 show the obtained error using the proposed GAbF algorithm. The translation and rotation error, in this case, decreases with the proximity but less markedly for the rotation. For the translation, we obtain a mean value at 5 meters of 0.3 meters. As referred before, the maximum translation error allowed is 1 meter to ensure a safe landing. Moreover, this accomplishes the needed requirement, with lower error than the obtained for the tested resampling schemes. The rotation error has a median value of 9.4 degrees, which is much lower than the obtained before.

The main difference here are the threshold based phases (Figure 7), and the ability to store information between iterations (Top M). This information is very useful when the algorithm is searching in a wrong direction and diverging from the expected result, having the ability to use the crossover and the mutation operators to get back and converge to the best solution.

V. Conclusions and Future Work

Using this UAV model, and because of its geometry, makes it easy to obtain high scores on the likelihood function for complementary poses since the vast majority of its pixels are located on the wings resulting in a large correspondence score. The working environment limits our likelihood choice since we cannot use likelihood formulas that are affected by illumination changes and can lead to poor results.

The particle filter resampling step was the main focus of our analysis, and ten existing resampling methods were applied with three added noise strategies between iterations. For an effective comparison, the particle number was kept constant between iterations and one resampling step was applied on each iteration (the computation time is equivalent between methods). From the obtained results we can see that the translation error is between 0.5 and 0.6 meters, a value bellow the predefined 1 meter of translation error limit. The rotation error obtained is mainly located near 0 and 180 degrees primarily due to the complementary poses similarity with the real ones, which makes the algorithms stuck in local minima. This happens mainly because between iterations, all the particles, including the best ones, are disturbed with noise

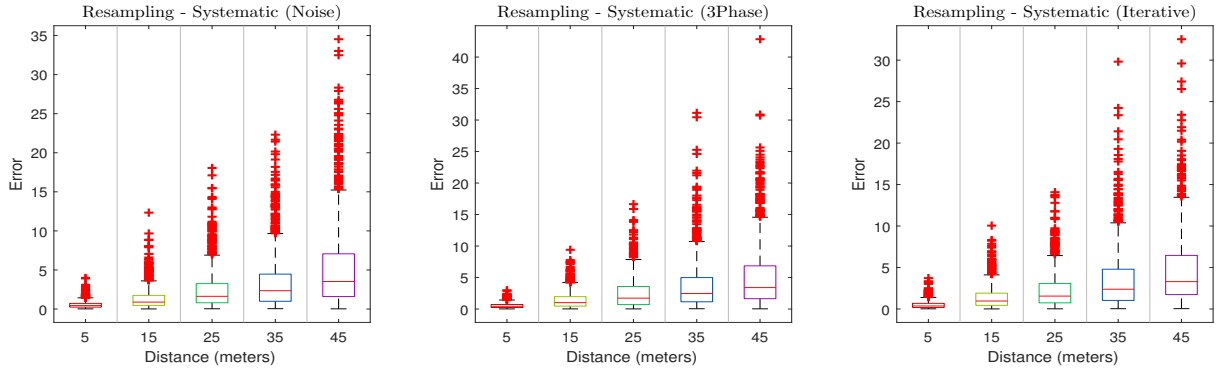

Fig. 8. Translation error – Resampling Systematic.

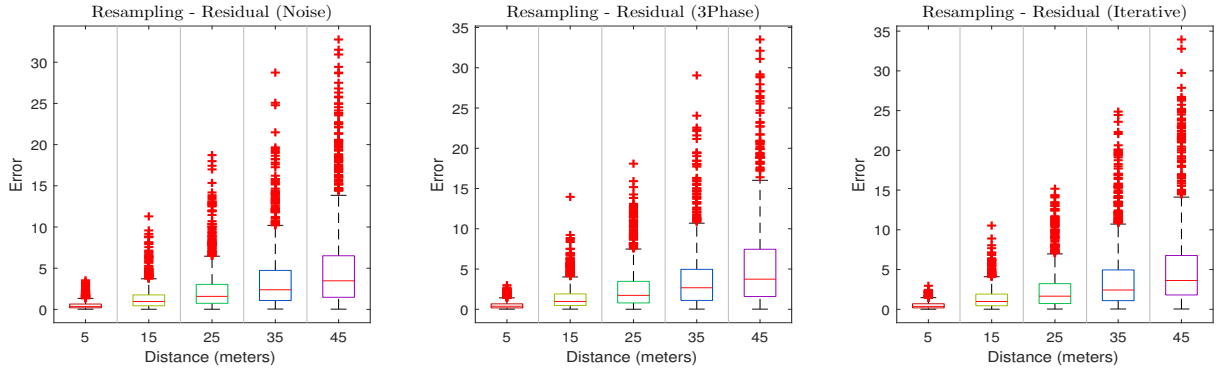

Fig. 9. Translation error – Resampling Residual.

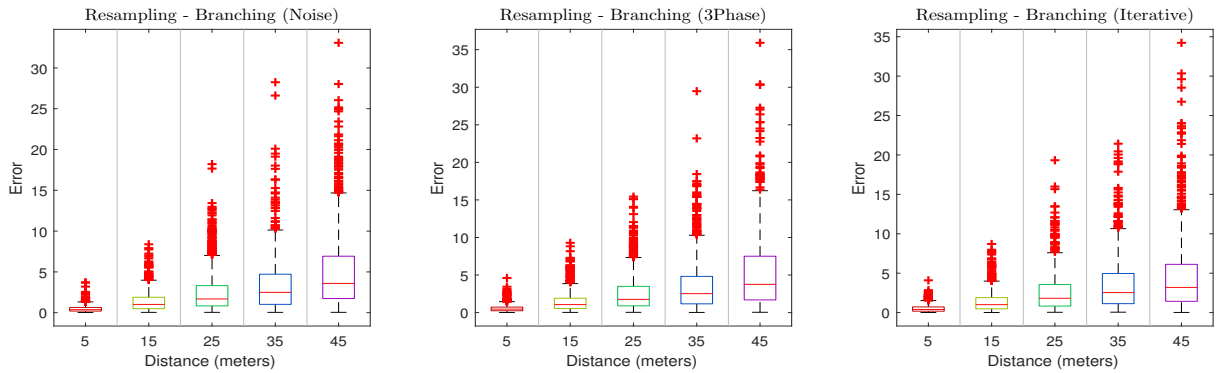

Fig. 10. Translation error – Resampling Branching.

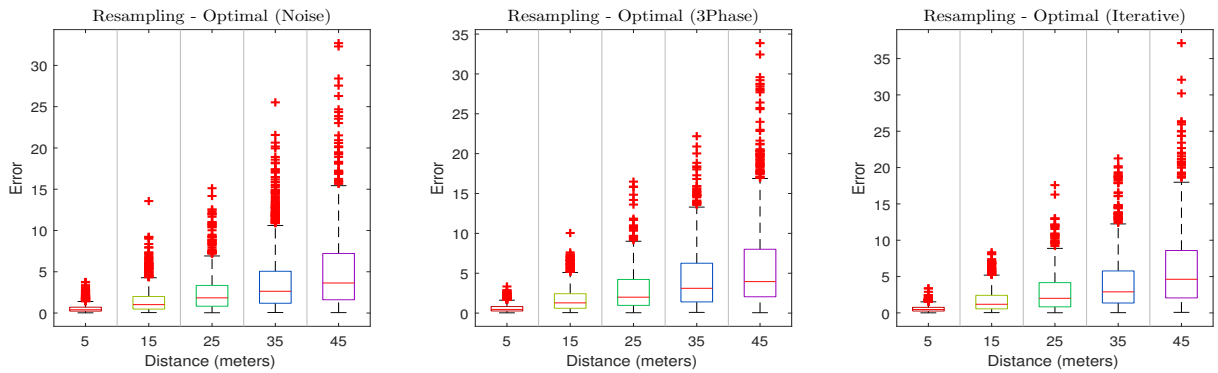

Fig. 11. Translation error – Resampling Optimal.

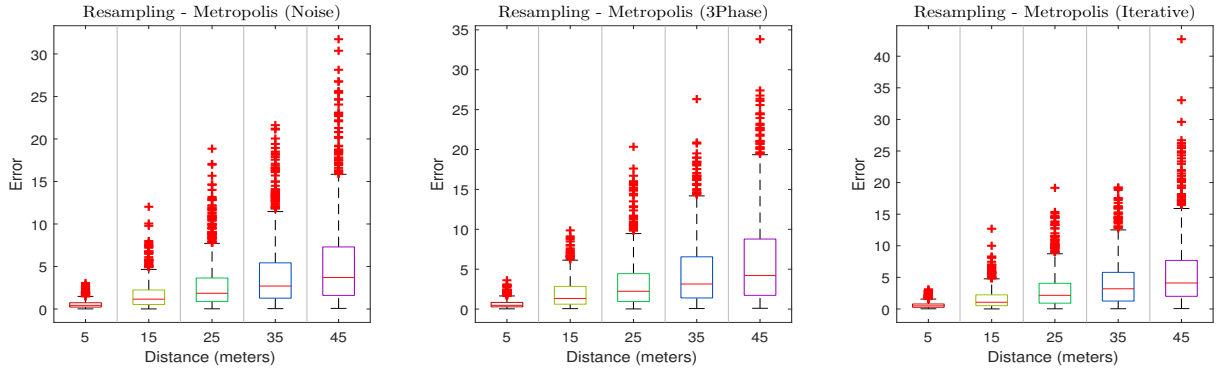

Fig. 12. Translation error – Resampling Metropolis.

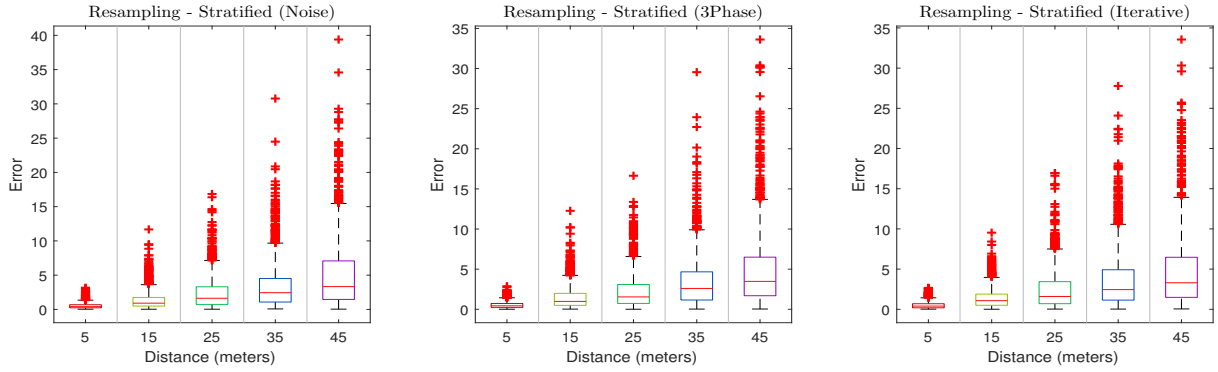

Fig. 13. Translation error – Resampling Stratified.

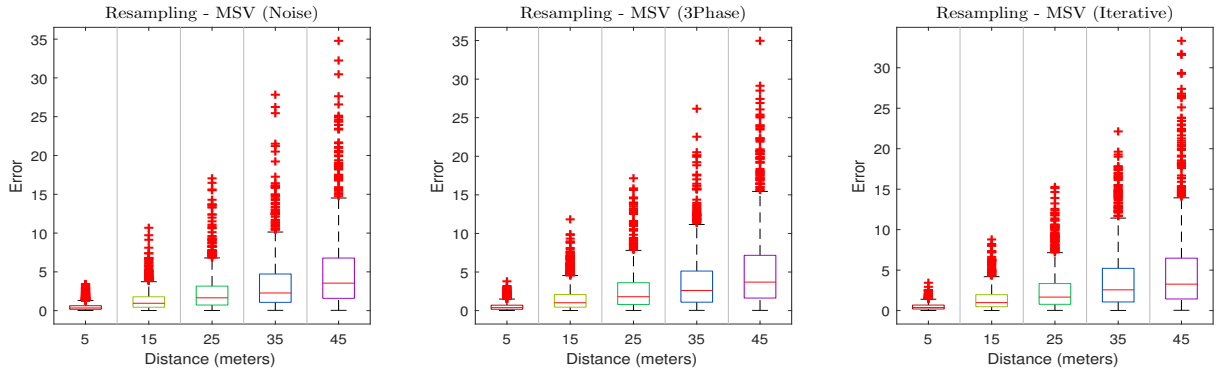

Fig. 14. Translation error – Resampling Minimum Sampling.

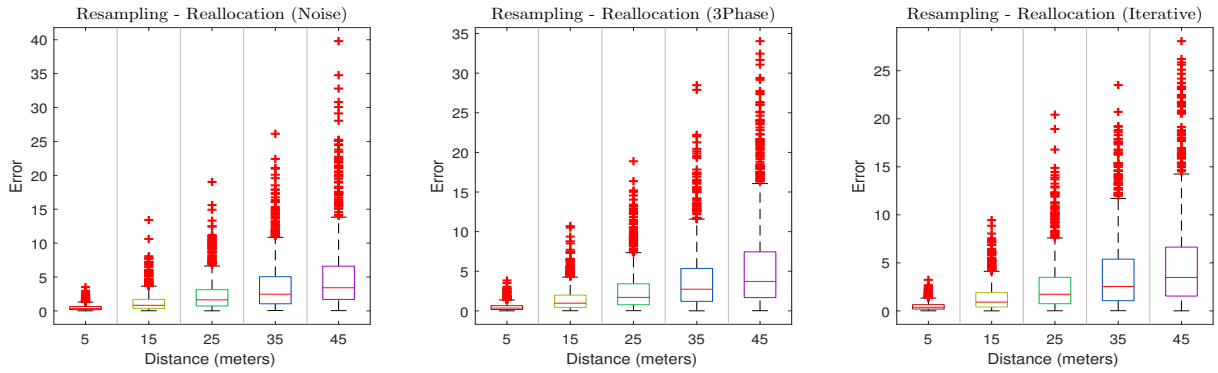

Fig. 15. Translation error – Resampling Rallocation.

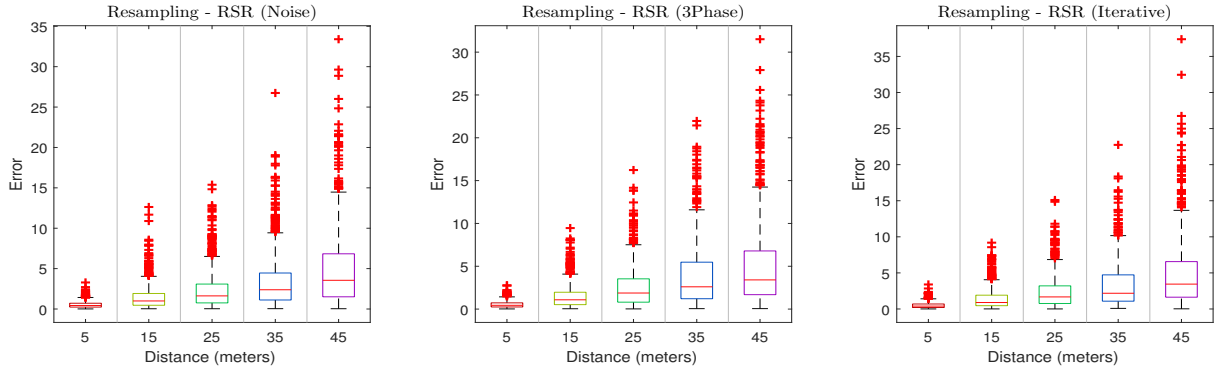

Fig. 16. Translation error – Resampling Residual Systematic.

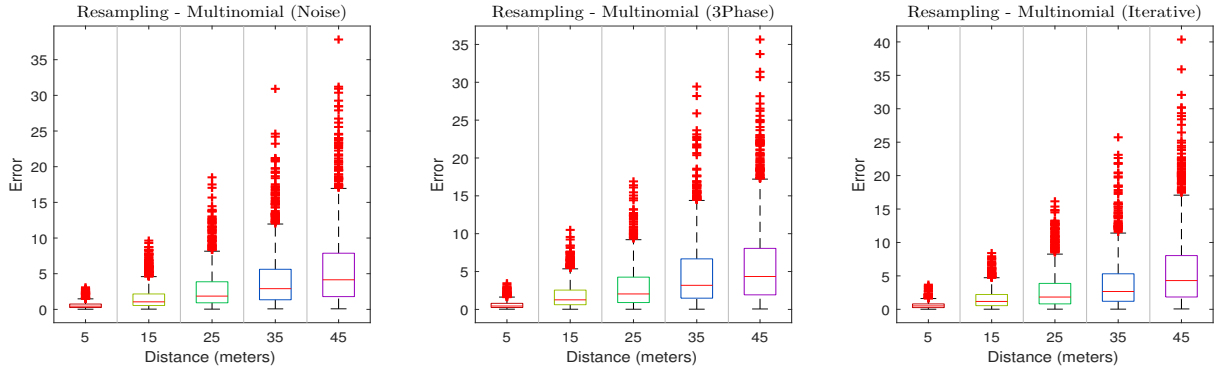

Fig. 17. Translation error – Resampling Multinomial.

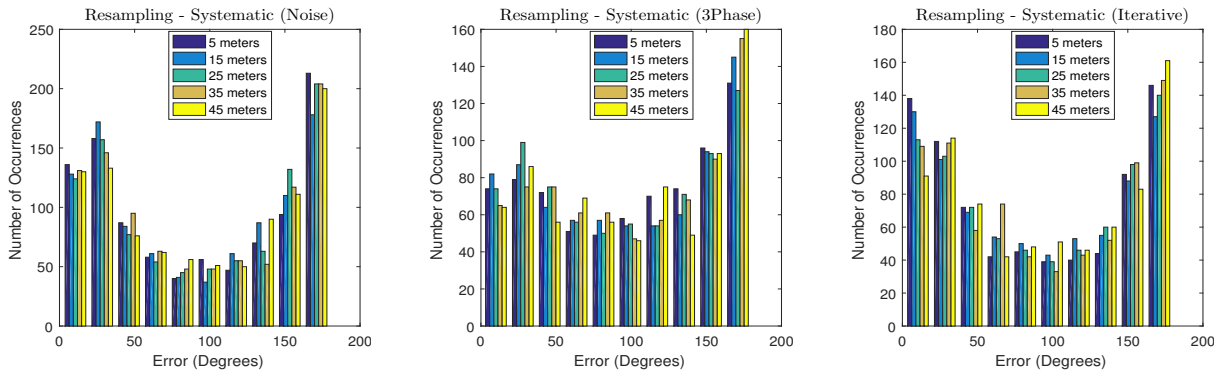

Fig. 18. Rotation error – Resampling Systematic.

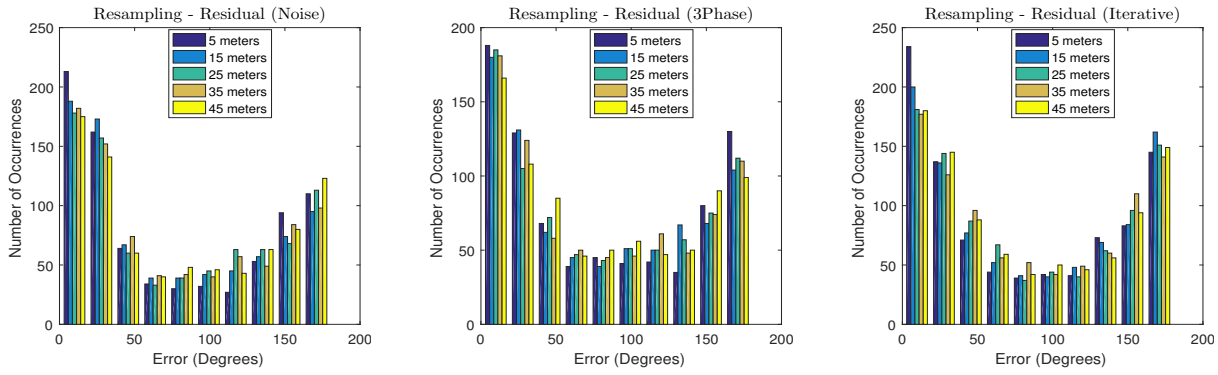

Fig. 19. Rotation error – Resampling Residual.

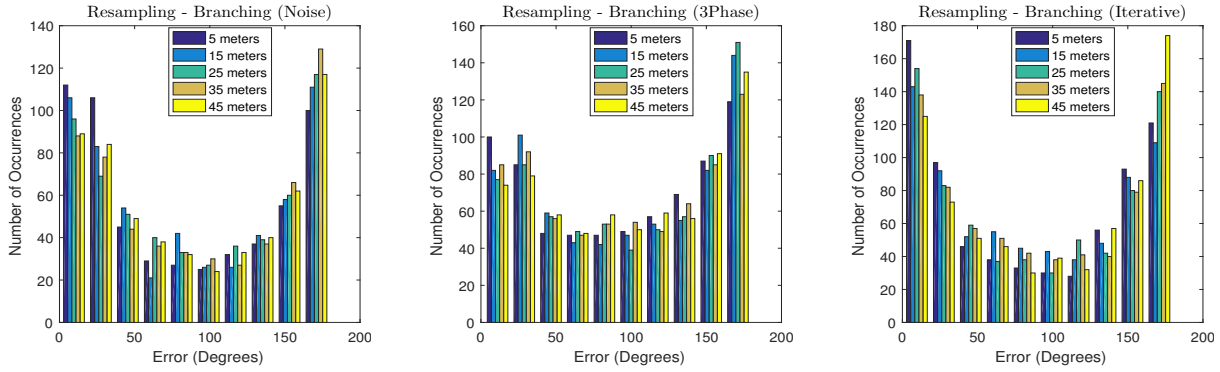

Fig. 20. Rotation error – Resampling Branching.

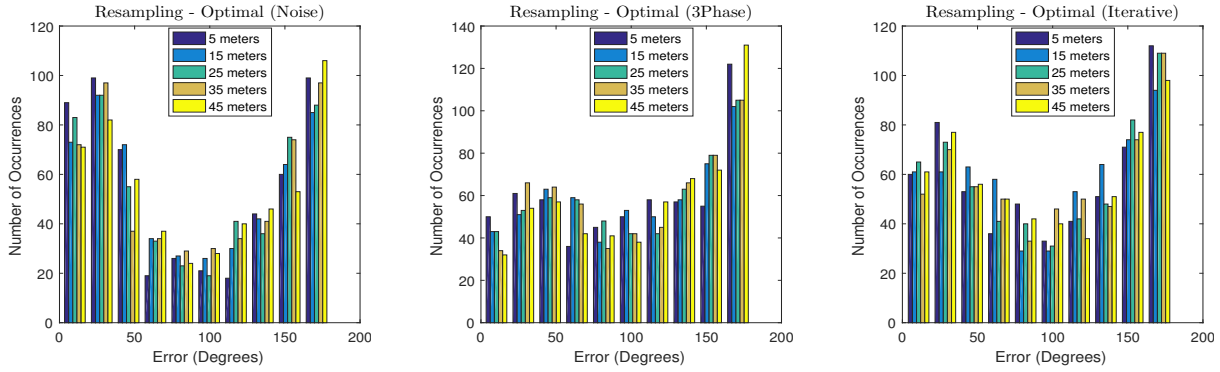

Fig. 21. Rotation error – Resampling Optimal.

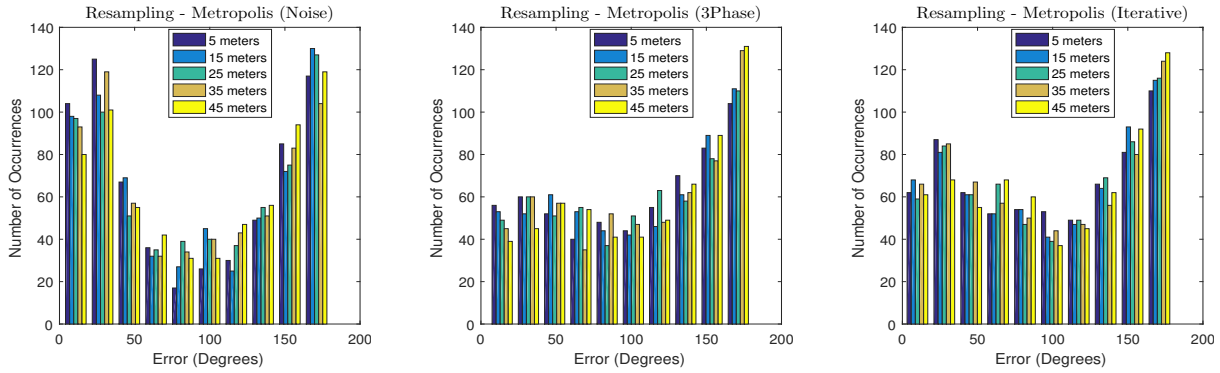

Fig. 22. Rotation error – Resampling Metropolis.

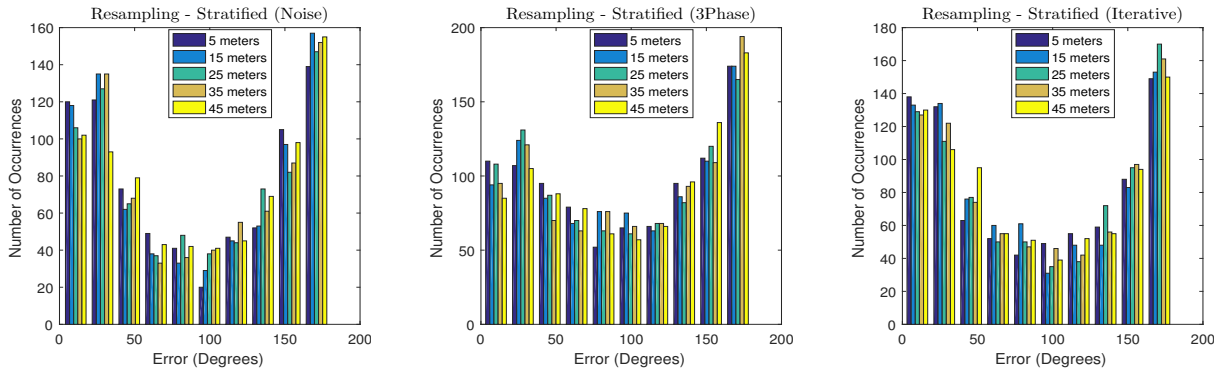

Fig. 23. Rotation error – Resampling Stratified.

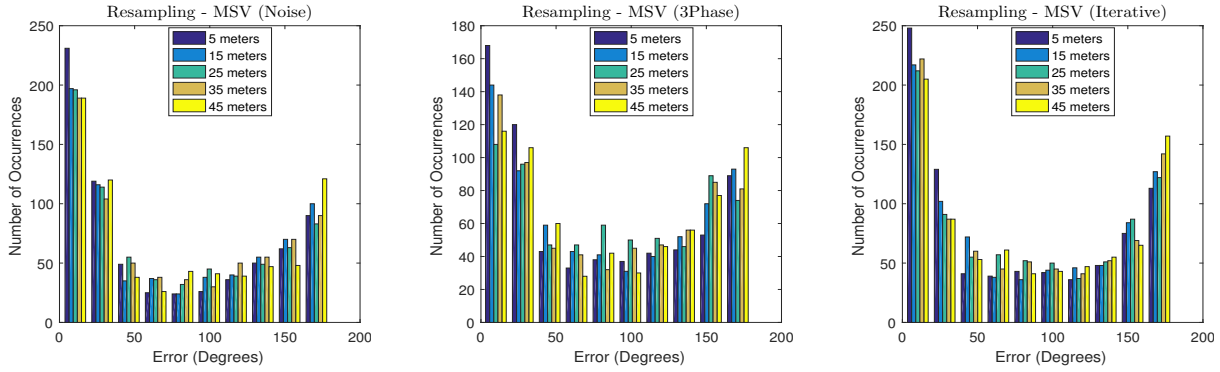

Fig. 24. Rotation error – Resampling Minimum Sampling.

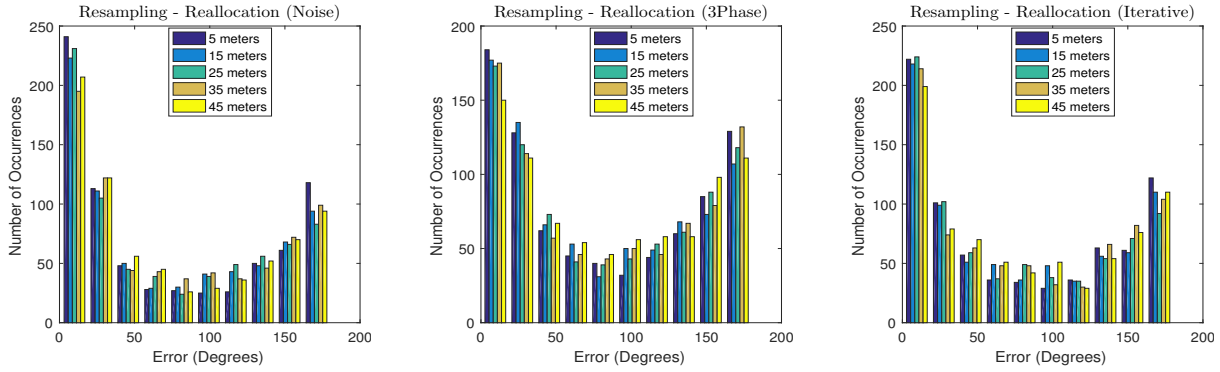

Fig. 25. Rotation error – Resampling Reallocation.

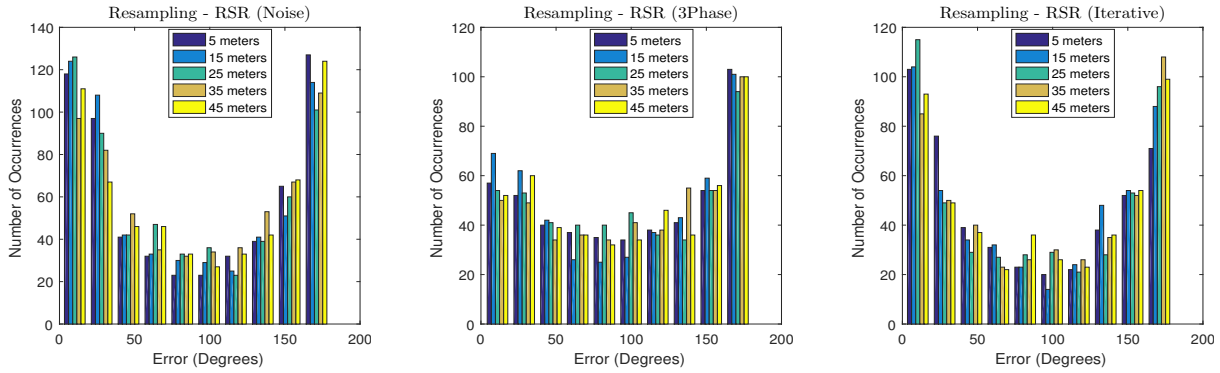

Fig. 26. Rotation error – Resampling Residual Systematic.

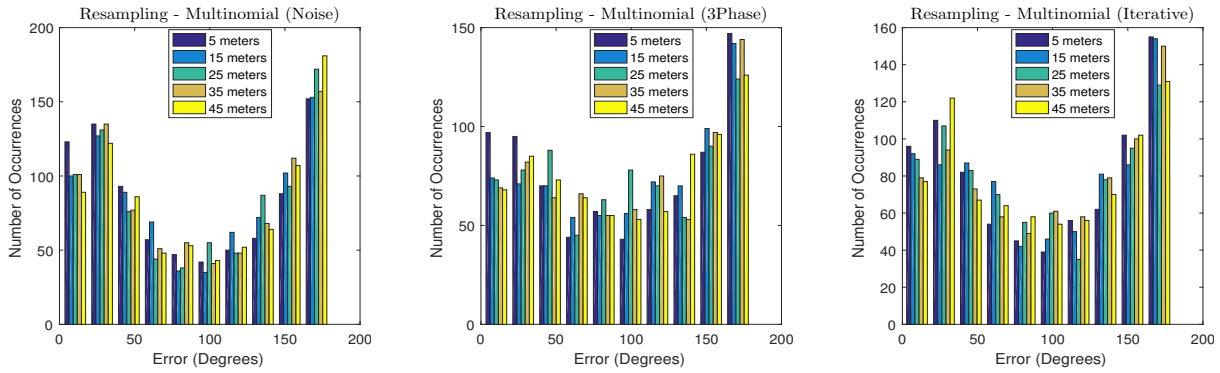

Fig. 27. Rotation error – Resampling Multinomial.

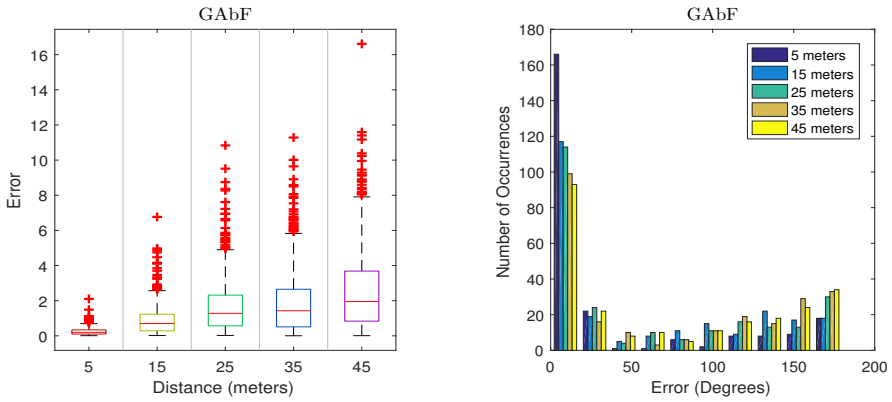

Fig. 28. Translation error (*left*) and rotation error (*right*) — GAbF.

(using one of the defined three strategies), and we can diverge easily from the result since the likelihood metric is very narrow.

A method named GAbF was developed and implemented, this architecture allows a very accurate position estimation (about 4% median error at 5 m) and a reasonable attitude error (about 10 median error at 5m). This happens mainly because we are using a three phase threshold value noise and saving the best particles in a buffer. Dependently of the obtained weight, we combine this particles with crossover and mutation operators. The mutation operator has the objective of giving particle diversity to the solution. This approach allows us to keep the best particles saved in a buffer, and if we diverge from the result, we can ease recovery the right path by using this information. We are using a finite set of particles, and in a high dimensional space if we do not maintain some information that is important we can easily get stuck in a complementary pose or searching in one area that is far from the real pose.

The obtained results are suitable for the following stages of the work, which we will focus on temporal filtering to add robustness to the pose estimation since we can discard some poses that are too different from the previous since it is physically impossible between successive frames to rotate e.g. 180 degrees.

References

[1] N. P. Santos, V. Lobo, and A. Bernardino, "A Ground-Based Vision System for UAV Tracking" presented at the OCEANS15 MTS/IEEE GENOVA, Genova, Italy, 2015.

[2] N. P. Santos, F. Melcio, V. Lobo, and A. Bernardino, "A Ground-Based Vision System for UAV Pose Estimation" International Journal of Mechatronics and Robotics (IJMR) - UNSYSdigital International Journals, vol. 1, p. 7, 2014.

[3] T. Merz, S. Duranti, and G. Conte, "Autonomous landing of an unmanned helicopter based on vision and inertial sensing" in Experimental Robotics IX, ed: Springer, 2006, pp. 343-352.

[4] A. Chavez, D. L'Heureux, N. Prabhakar, M. Clark, W.-L. Law, and R. J. Prazenica, "Homography-Based State Estimation for Autonomous UAV Landing" in AIAA Information Systems-AIAA Infotech@ Aerospace, ed, 2017, p. 0673.

[5] Y. Zhang, L. Shen, Y. Cong, D. Zhou, and D. Zhang, "Ground-based visual guidance in autonomous UAV landing" 2013, pp. 90671W-90671W-6.

[6] S. Park, J. P. Hwang, E. Kim, and H.-J. Kang, "A new evolutionary particle filter for the prevention of sample impoverishment" Trans. Evol. Comp, vol. 13, pp. 801-809, 2009.

[7] A. Carmi, S. J. Godsill, and F. Septier, "Evolutionary MCMC particle filtering for target cluster tracking" in Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th, 2009, pp. 262-267.

[8] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection" in Computer Vision ECCV 2006. vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 430-443.

[9] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection" IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pp. 105-119, 2010.

[10] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models" Statistical science, vol. 30, pp. 328-351, 2015.

[11] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation" in Radar and Signal Processing, IEE Proceedings F, 1993, pp. 107-113.

[12] M. K. So, "Posterior mode estimation for nonlinear and non-Gaussian state space models" Statistica Sinica, pp. 255-274, 2003.

[13] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems" in Radar, Sonar and Navigation, IEE Proceedings-, 1999, pp. 2-7.

[14] T. Li, M. Bolic, and P. M. Djuric, "Resampling Methods for Particle Filtering: Classification, implementation, and strategies" Signal Processing Magazine, IEEE, vol. 32, pp. 70-86, 2015.

[15] E. R. Beadle and P. M. Djuric, "A fast-weighted Bayesian bootstrap filter for nonlinear model state estimation" Aerospace and Electronic Systems, IEEE Transactions on, vol. 33, pp. 338-343, 1997.

[16] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems" Journal of the American statistical association, vol. 93, pp. 1032-1044, 1998.

[17] Boli, P. M. Djuri, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective" EURASIP Journal on Applied Signal Processing, vol. 2004, pp. 2267-2277, 2004.

[18] J. S. Liu, R. Chen, and W. H. Wong, "Rejection control and sequential importance sampling" Journal of the American Statistical Association, vol. 93, pp. 1022-1031, 1998.

[19] Z. Jianping, B. Baoming, and W. Xinmei, "Increased-diversity systematic resampling in particle filtering for BLAST" Systems Engineering and Electronics, Journal of, vol. 20, pp. 493-498, 2009.

[20] M. Boli, P. M. Djuri, and S. Hong, "New resampling algorithms for particle filters" in Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, 2003, pp. II-589-92 vol. 2.

[21] A. Budhiraja, L. Chen, and C. Lee, "A survey of numerical methods for nonlinear filtering problems" Physica D: Nonlinear Phenomena, vol. 230, pp. 27-36, 2007.

[22] D. Crisan and T. Lyons, "A particle approximation of the solution of the KushnerStratonovitch equation" Probability Theory and Related Fields, vol. 115, pp. 549-578, 1999.