Vision-Based Precision Pose Estimation For Autonomous Formation Flying

Rohan Punnoose
Aeronautics and Astronautics
Stanford University
Stanford, CA
rohanjp@stanford.edu

I. Introduction

Autonomous formation flying of aircraft is a challenging problem, and is a key component of various important capabilities for autonomous aircraft, from air-to-air refueling, to formation-based traffic management, to increased aerodynamic efficiency due to reduced induced drag. Tight formation flying requires precise navigation of relative position and orientation, beyond what state-of-the-art inertial and GPS-based sensor suites are capable of. Using vision (in particular, monocular cameras) is a promising way to achieve the precise relative navigation of required precision such that tight formation flight is possible. Indeed, tight formation flying by human pilots relies entirely on human vision (direct line of sight at all times). This research aims to develop accurate and efficient pose estimation algorithms which utilize Bayesian filterning and modern computer vision developments (specifically, deep convolutional neural networks, or CNNs) as a core part of their operation.

II. Prior Research

Prior research in pose estimation for autonomous aircraft formation flying [1-3] mostly came before the widespread rise in popularity/awareness of deep learning algorithms for computer vision. As such, they leveraged classic computer vision techniques and relied on detecting pre-specified markers (LEDs) using hand-designed computer vision algorithms, then running point matching optimization algorithms to correspond the detected markers to their known orientation on the aircraft structure, thereby recovering pose. More recently, [4] leverages CNNs for pose estimation of spacecraft, by discretizing the relative pose space and using a CNN to generate a belief distribution across each of the discrete poses. Spacecraft pose estimation shares many similarities with the aircraft focus of this paper, however there are also salient differences. In particular, the background imagery/lighting, solar effect, and dynamic range are highly different between space and air vision applications.

Particle filters are a classical and well-established Bayesian filtering approach [5, 6], which use a weighted Monte Carlo sampling technique to effectively construct a posterior distribution over the state of a particular Markov process. They are appealing for their ability to represent complicated, multimodal distributions (such as those that may arise in this relative pose estimation problem setting), and they ability to handle nonlinear dynamics. The approach of [7] uses a particle filter for pose estimation of a UAV landing on an aircraft carrier. Each particle represents an onboard-generated rendering which is matched against the camera-generated image. They note particular challenges in initializing the filter, in order to get the filter to converge reliably.

III. Problem Statement

The research problem of this paper can be stated as follows. Aircraft L (the leader) and aircraft F (the follower) have orientations R_L, R_F, respectively, w.r.t. the inertial frame, and positions r_L, r_F, respectively, w.r.t the inertial frame. Aircraft F has camera c_F, with known orientation relative to the F body frame. Camera c_F produces image I_F. The relative pose estimation task is then using I_F to recover estimates of relative pose $\hat{R}_{L/F}$, $\hat{r}_{L/F}$, where $R_{L/F} = R_L R_F^T$ and $r_{L/F} = r_L - r_F$.

IV. Approach

A. Overview

The general approach for this research is to use classical Bayesian filtering techniques and augment them with neural networks, inspired by the success of CNNs in other computer vision applications [10].

Given the incorporation of neural networks into the design, this begs the question of eschewing Bayesian filtering entitely, and simply using deep learning for the entire system. One such approach might be that a CNN inputs the image directly and outputs an estimate of the pose. However, this approach has significant drawbacks. Using a pure, end-to-end neural network does not incorporate any prior knowledge about the dynamics of the system, which might limit the attainable performance. Since the output says nothing about uncertainty, it is difficult to trust the pose label directly, and is also difficult to fit into the larger aircraft navigation/state estimation framework. Additionally, it is non-trivial to parameterize the pose (specifically the attitude) in such a way that the CNN can easily output that parameterization without singularities or additional constraints.

Another option is to have discrete pose labels and have the CNN output corresponding probabilities (effectively a belief distribution), as is done in [4]. While the results in [4] are promising, the pose estimation accuracy is limited by the pose

label discretization coarseness. This is likely unnacceptable for tight formation flight, where the fast dynamics and close distances require more precision than can be attained with a discretized pose label space, since the curse of dimensionality applies to increasing the discretization density in the pose label space.

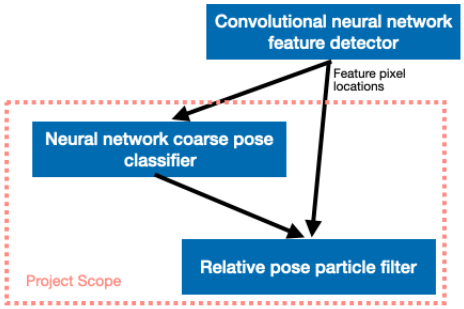

Fig. 1. Architecture diagram

Instead, the approach taken in this work contains the following components:

1. A convolutional neural network is trained to detect prespecified features on the aircraft, with known position in the aircraft structural frame. The development of this subsystem is outside of the scope of this project, instead we assume that a system exists which can accuracy detect known structural features from images. Some features may be occluded and therefore not detected by the CNN, this is handled in the remaining parts of the algorithm.

2. Similar to [4], using the detected feature locations in pixel space, an artificial neural network classifies the detected structural features in the image into a discrete pose label.

3. In parallel, a particle filter (similar to [7]) is run, where each particle generates synthetic feature detection measurements based on particle state. The likelihood weight of the particle is determined by a loss function between the particle's feature locations and the actual feature locations detected by the CNN. If a feature is not detected by the CNN, it is assumed to be occluded and ignored. The sampling of the particle filter is informed by the pose label output by the ANN. This is intended to facilitate the particles in convering to the true distribution, by leveraging probabalistic knowledge contained within the trained ANN.

B. System Dynamics

The aircraft relative motion dynamics, as relevant to formation flight, are a simplified and discretized form of the true, nonlinear continuous dynamics. The state vector is $x = \begin{bmatrix} r_{F/L} & R_{L/I} & R_{F/I} \end{bmatrix}^T$, which are the relative position and inertial-relative orientations. The control inputs are, for each the leader and follower aircraft, $u = \begin{bmatrix} v_x & \theta & \phi \end{bmatrix}^T$, or the forward relative velocity, pitch, and roll. The pitch and roll angles induce forward and lateral velocities, respectively. In the context of the pose estimator on the follower aircraft, the control input to the leader is not available to the estimation algorithm. The camera on the follower aircraft is assumed to gimbal such that it always points directly at the leader aircraft.

C. Synthetic Data and Simulation

A simple ray tracing and camera projection algorithm is used to take structural features and generate synthetic images, both for the purpose of simulation, and for the synthetic image likelihood component of the particle filter.c

Figures 2 and 3 show examples of the simulation, along with pixel locations for the structural features (shown in red). Features which are occluded by other parts of the structure are not detected, which is intended to reflect the fact that a CNN also cannot detect structural features that are occluded by rest of the aircraft structure. There are 14 total features that may be detected on the aircraft.

Fig. 2. Example synthetic rendering and detected features (red)

Fig. 3. Example synthetic rendering and detected features (red)

D. Relative Pose Label Learning

	min	max	n
x [m] (forward)	-100	50	10
y [m] (lateral)	-50	50	10
z [m] (vertical)	-20	20	8
phi [deg] (roll)	-45	45	6

TABLE I
RELATIVE POSE DISCRETIZATION

The relative pose space is discretized into 4800 labels in 4 dimensions (3 positions, and a remaining rotation about the camera boresight), as detailed in Table 1 above. This discretization essentially ignores relative pitch and yaw, which in the context of formation flight is reasonable (also note that the particle filter uses the dynamics in the full relative pose space).

A neural network is trained to predict the correct discrete pose label given input features. The neural network has 42 inputs (x pixel position, y pixel position, and occlusion flag, for each of the 14 structural features), 2 100-neuron hidden layers, and a 4800-way softmax output layer. ReLU is used as the activation function. This network was trained using the Adam optimizer.

The trained network achieved 54% accuracy on a validation test set. In practice, this is quite accurate, since incorrect labels are commonly still relatively "close" in the relative pose space. Figures 4 and 5 show, for the validation set, the error between the the true pose and the classified pose, and the error between the classified pose and the correct label's pose, respectively.

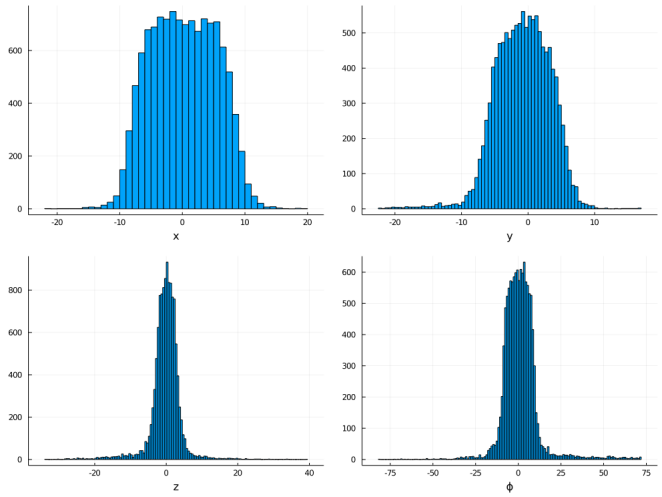

Fig. 4. Error between classified pose and true pose

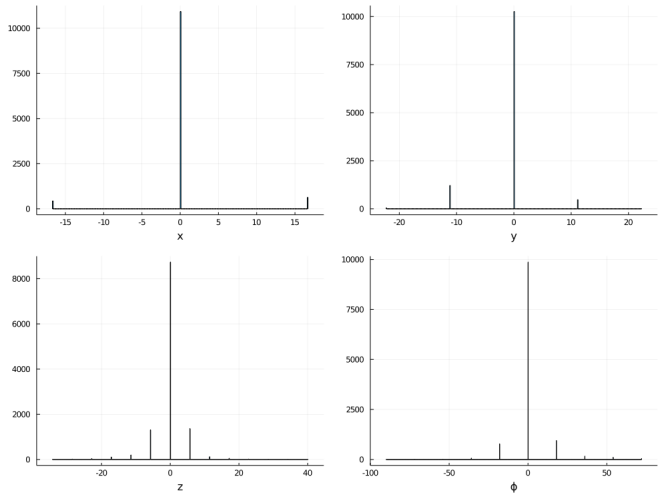

Fig. 5. Error between classified pose and the correct pose label

From Figure 4, it is clear the the majority of true pose error comes from the discretization, though there is a tail due to misclassified poses. Figure 5 shows the frequency of these misclassified poses, and that they tend to be adjacent to the correct pose label.

E. Particle Filter

The particle filter takes in the detected structural features, and the classified pose label from the ANN, in order to compute the final pose estimate. The particle filter functions as follows, at each timestep:

1. At filter initialization, N particles x_i are sampled using the pose label and the error distribution derived from the validation set. Specifically, the algorithm takes samples from the true pose to classified pose error distribution (shown in Figure 5) and applies this delta to the actual classified pose.

2. The particles are propagated through the nonlinear dynamics. Since the forward velocity, pitch, and roll of the leader aircraft are not known to the follower, these quantities are treated as a disturbance and are sampled uniformly.

3. For each particle, a synthetic image and detected feature locations are produced. The weights for each particle w_i are updated based on the measurement likelihood, which is implemented as the inverse of the mean squared error between the actual image's detected features, and the particle's synthetic image's detected features. Occluded features are ignored, since it is computationally expensive to compute the occlusions of features for each particle.

4. The particles are then resampled. floor(αN) samples are drawn from the current set of particles, weighted by likelihood (the standard particle filter resample), and $N -$ floor(αN) samples are draw from the classified pose label + error distribution. α is a tunable paramter of the algorithm. α may also be set to 1 by default, and only drop lower at specific times.

V. Results

The algorithm was implemented and tested on a formation flight approach scenario, where the follower aircraft closes in to the leader over the course of a 10 s trajectory. Figure 6 shows the true relative position of this trajectory.

Figures 7 and 8 show the image as seen be the follower aircraft on the trajectory.

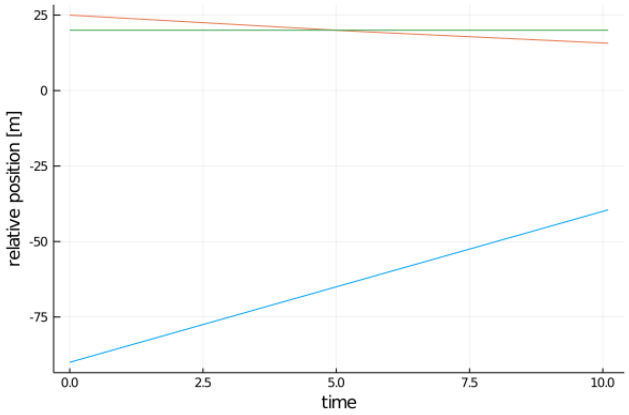

Fig. 6. Relative position trajectory

Figures 9 and 10 show the position and attitude error results of 2 particle filter configurations.

Fig. 7. Image during trajectory

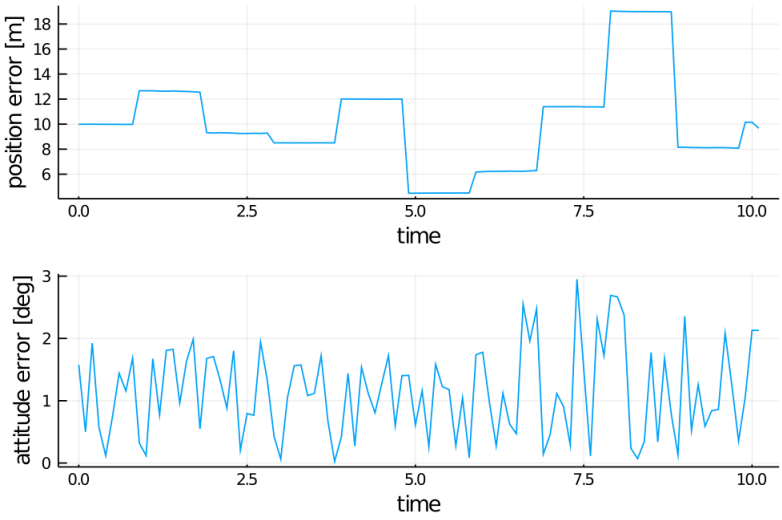

Fig. 9. $N = 1000$, $\alpha = .9$ every 10 cycles

Fig. 8. Image during trajectory

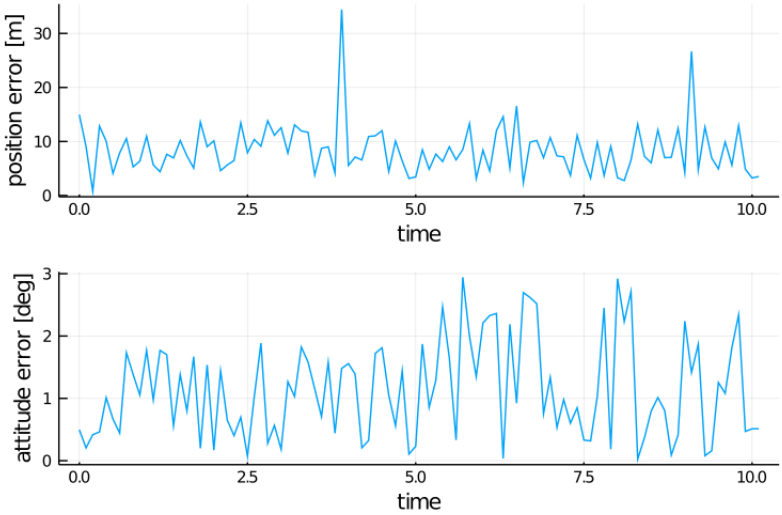

Fig. 10. $N = 5000$, $\alpha = .9$ every cycle

VI. Discussion and Conclusions

These results show that the ANN segment of the pose estimation system is effective at producing a good classified pose. However, this pose label is coarse, inherent to the discretization. While the particle filter is intended to address this, by having the particle distribution converge on the true pose distribution, these results do not indicate that was successful. It appears that the particle filter does not result in much more convergence to the true pose - pose error appears to be dominated by the error from the pose label classification. It is possible that better particle filter performance is attainable through better selection of filter parameters. It is also possible that the number of particles is simply insufficient given the high-dimensional state space, however adding more particles quickly becomes computationally prohibitive.

The success of the ANN at classifying pose based on detected features is notable, however. Future research might build off of its effectiveness at producing a coarse pose label, and use a different method to refine the estimate to higher precision, such as bundle adjustment. Elements of the particle filter might also be retained, as a way of incorporating prior knowledege about the dynamics of the system into the estimator.

References

[1] Mahboubi, Z., Kolter, Z., Wang, T., & Bower, G. (2011). Camera based localization for autonomous UAV formation flight. In Infotech@ Aerospace 2011 (p. 1658).

[2] Wilson, D. B., Goktogan, A. H., & Sukkarieh, S. (2014, May). A vision based relative navigation framework for formation flight. In 2014

IEEE International Conference on Robotics and Automation (ICRA) (pp. 4988-4995). IEEE.

[3] Mammarella, M., Campa, G., Napolitano, M. R., & Fravolini, M. L. (2010). Comparison of point matching algorithms for the UAV aerial refueling problem. Machine Vision and Applications, 21(3), 241-251.

[4] Sharma, S., Beierle, C., & DAmico, S. (2018, March). Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In 2018 IEEE Aerospace Conference (pp. 1-12). IEEE.

[5] Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. IEEE signal processing magazine, 20(5), 19-38.

[6] Doucet, A., De Freitas, N., Murphy, K., & Russell, S. (2013). Rao-Blackwellised particle filtering for dynamic Bayesian networks. arXiv preprint arXiv:1301.3853.

[7] Santos, N. P., Melcio, F., Lobo, V., & Bernardino, A. (2014). A ground-based vision system for UAV pose estimation. International Journal of Mechatronics and Robotics (IJMR)-UNSYSdigital International Journals, 1(4), 7.

[8] Campa, G., Napolitano, M. R., Perhinschi, M., Fravolini, M. L., Pollini, L., & Mammarella, M. (2007). Addressing pose estimation issues for machine vision based UAV autonomous serial refuelling. The Aeronautical Journal, 111(1120), 389-396.

[9] Thrun, S. (2002). Probabilistic robotics. Communications of the ACM, 45(3), 52-57.

[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).