Particle Filter Based Mosaicking For Tracking Forest Fires

Justin M. Bradley and Clark N. Taylor
MAGICC Lab, Brigham Young University, Provo, Ut, 84602

Using miniature air vehicles (MAVs) is a cost effective, simple method for collecting data about the size, shape, and location characteristics of a forest fire. However, noise in measurements used to compute pose (location and attitude) of the camera on board the MAV leads to significant errors in the processing of collected video data. Typical methods using MAVs to track fires attempt to find single geolocation estimates and filter that estimate with subsequent observations. While this is an effective method of resolving the noise to achieve a better geolocation estimate, it reduces a fire to a single point or set of points. A georeferenced mosaic is a more effective method for presenting information about a fire to fire fighters. It provides a means of presenting size, shape, and geolocation information simultaneously. We describe a novel technique to account for noise in pose estimation of the camera by converting it to the image domain. We also introduce a new concept, a Georeferenced Uncertainty Mosaic (GUM), in which we utilize a Sequential Monte Carlo method (a particle filter) to resolve the noise in the image domain and construct a georeferenced mosaic that simultaneously shows size, shape, geolocation, and uncertainty information about the fire.

I. Introduction

Forest fire monitoring has become an increasingly important area of research, seeking to provide tools to enable fire fighters to more quickly locate, analyze, and hopefully extinguish fires. Government programs have been set up in the U.S. and Europe to address this challenge. Technological approaches to solving the fire monitoring problem have included the use of satellite imagery, early detection systems with cameras on strategically placed towers, and placement of ground cameras in carefully planned locations.

Another approach being explored is using multiple autonomous miniature air vehicles (MAVs) in cooperation to collect data about a fire or multiple fires using cameras on board the MAVs. This solution is particularly attractive as MAVs can be deployed quickly and easily, minimizing risk to personnel. Much of the prior work in this area has addressed the issues of cooperative control and path planning dynamics to ensure that the MAVs collect the video (infrared, near-infrared, or visible) necessary to monitor a fire.

Once video of a forest fire has been collected, it is necessary to process the video to provide fire fighters with accurate and concise information about the location and attributes of the fire. A fundamental problem with providing accurate geolocation information is that the pose (location and attitude) estimate of the MAV — and therefore the pose estimate of the camera mounted on the MAV — is inaccurate. Merino et al. propose a method for overcoming the error in geolocation estimates by (1) segmenting color images and thresholding infrared images to produce binary images (fire or no fire), and (2) applying an unscented Kalman filter on the fire geolocations to compute a more accurate estimate over time. This approach, however, assumes that all forest fires can be represented by a single point or set of points. Forest fires, however, can be of arbitrary shape and size. Furthermore, by segmenting and thresholding images to create binary images, information about “warm,” as opposed to “hot,” spots may be lost.

We propose a novel approach to the problem of estimating and presenting fire information. We represent the fire as a non-parametric two-dimensional function of real-world location, whose values represent our...
confidence that there is fire at the given location. This function can be visually represented to the user by sampling it at regular 2-D grid points and presenting the samples as an image, which we term a Georeferenced Uncertainty Mosaic (GUM). This approach has two advantages. First, this method makes no assumptions about the shape or extent of the fire, and therefore allows representation of arbitrary fire shapes and sizes. Second, this representation allows uncertainty in geolocation to be presented to the system operator in a way that is intuitive and easily understood.

To create a GUM we need observations (infrared images) showing the probability of a fire at a given location. We construct these observations by applying a blurring function to the image captured from the on-board infrared camera. This blurring function is derived from uncertainty in the pose estimate of the MAV. A particle mosaic is the result of compositing these observations together using a Sequential Monte Carlo method (a particle filter) in a mosaic-like fashion. We then use the particle mosaic to form the GUM, which is presented to the system operator.

This paper is organized as follows: in Section II we describe our method to convert noise in the pose estimate of the camera into the image domain. In Section III we describe a novel approach to mosaicking, namely, a particle mosaic created through the use of a Sequential Monte Carlo method, as well as the creation of the GUM. In Section IV we present results of our algorithm, and Section V concludes the paper.

II. Mapping Uncertainty to the Image Domain

We desire to present fire fighters with high-level information about observed forest fires, including size, shape, and geolocation. What we obtain from the MAV, however, is an infrared image and associated noisy pose estimate. We want to form a filter in which we construct a posterior distribution indicating the information desired. We therefore, need an observation density to use in the filter. To form a suitable observation density we begin by forming a mapping function relating points in the assumed planar scene to points in the observed image. We take the derivative of this mapping function with respect to each pose parameter and combine this information into a spatially varying blurring function. We then apply the constructed blurring function to the observed image forming an uncertainty image, in which pixel values indicate the probability of fire at the corresponding geolocation.

A. Setup and Terminology

We assume that the pose of the MAV has six degrees of freedom representing the location and attitude of the camera on board the MAV. They are:

- \( t_x \): translation in the \( \hat{x} \) direction going out the nose of the MAV.
- \( t_y \): translation in the \( \hat{y} \) direction going out the right wing of the MAV.
- \( t_z \): translation in the \( \hat{z} \) direction going out the bottom of the MAV.
- \( \phi \): MAV roll, or rotation about the \( \hat{x} \) axis.
- \( \theta \): MAV pitch, or rotation about the \( \hat{y} \) axis.
- \( \psi \): MAV heading (yaw), or rotation about the \( \hat{z} \) axis.

Let \( x_{ci} \) be a homogeneous point, \([x_{ci}, y_{ci}, 1]^T\), on the image plane of the camera, whose \( \hat{x}, \hat{y} \) components form the location \( p_{x_{ci}, y_{ci}} \) in the image itself. Let \( x_{mi} \) be a point, \([x_{mi}, y_{mi}, 1]^T\), on the image plane of a virtual mosaic camera, whose image plane is also the ground plane.

B. The Mapping Function and Homography

Every point \( x_{ci} \) on the image plane of the camera (and therefore in the image itself) corresponds to a certain real-world geolocation represented by \( x_{mi} \) on the image plane of the virtual mosaic camera. The “image” of geolocation points corresponding to the pixels in the image is called the pre-image.

---

\(^{c}\)This uncertainty information is tightly coupled with fire intensity information: i.e., a dim spot on the GUM may represent low confidence that this position is on fire or high confidence that the position is a “warm” spot. We discuss this difficulty further in Section V.
If we assume the terrain which we are imaging (the ground) is planar, we can use pose and camera calibration parameters to construct a homography matrix as

$$
\begin{bmatrix}
\lambda x_{ci} \\
\lambda y_{ci} \\
\lambda
\end{bmatrix} = H x_{mi},
$$

(1)

where $\lambda$ is a constant. To then determine the corresponding point, $x_{ci}$, on the image plane of the camera, we compute

$$
\lambda x_{ci} = \lambda \begin{bmatrix} x_{ci} \\ y_{ci} \\ 1 \end{bmatrix}.
$$

(2)

This entire process describes the non-linear mapping function used to compute the point correspondences between captured image points and real-world geolocations. We represent this function as

$$x_{ci} = f(H, x_{mi}).
$$

(3)

C. Vector Fields of Motion in Texture Space

Because we do not know the true MAV pose parameters, the homography matrix we estimate is not correct. Thus, a given world point could correspond to a number of points in the imaging plane. We would like to blur the captured image, so that our uncertainty in the pose estimate is expressed by uncertainty as to the precise pixel location of a given feature. Put another way, by blurring the image, we can make it difficult to determine exactly where a given world point is located in the image, effectively expressing our uncertainty about the pose of the camera in the image domain.

In order for this blurring to express the pose uncertainty in a meaningful way, the blurring needs to express how errors in the MAV pose parameters map to errors in the image location of a terrain feature. We can accomplish this by computing partial derivatives of the mapping function between world locations and image locations. The result is a vector field and represents the motion in texture space of the pre-image. For example, we could compute how errors in $\psi$, the MAV heading angle, translate into errors in image location of a given ground point as

$$\frac{\partial x_{ci}}{\partial \psi} = \frac{\partial f(H, x_{mi})}{\partial \psi},
$$

(4)

for each point $p_{x_{ci}, y_{ci}}$ in the image. Using these partial derivatives, we can compute a vector field representing the total change in the $\hat{x}$ and $\hat{y}$ directions of the image for changes in $\psi$. In Figure 1 we show an image of vectors representing the vector field of motion in texture space for changes in $\psi$. If we do this for each of the six pose parameters, we obtain six separate vector fields describing the motion in texture space for changes in each pose parameter.

D. Constructing the Blurring Function

We now discuss how to combine the information from the vector fields into a spatially varying blurring function we can apply to the observed image. The result of applying the blurring function is an uncertainty image where brighter values represent less uncertainty in the location of the fire. This is done by constructing six zero-mean bivariate Gaussian random variables for each pixel and then combining these six random variables into one bivariate Gaussian representing our uncertainty for that pixel.

We create the individual random variables representing uncertainty from a single pose parameter by constructing its covariance matrix according to information from the vector field. As an example we will examine the creation of the blurring function for $\psi$. Let $\xi$ be the magnitude of the vector in the vector field at image location $p_{x_{ci}, y_{ci}}$. Let $\beta$ be the angle that vector makes with the $\hat{x}$ axis. Also let $\sigma_\psi^2$ be the variance of the white noise process associated with the $\psi$ estimate.

We need to construct a bivariate Gaussian random variable over some pixel $p_{x_{ci}, y_{ci}}$. It should have infinitesimal variance in one direction and $\xi \sigma_\psi^2$ variance in the orthogonal direction, thus representing the
uncertainty indicated by the vector in the vector field. By analyzing the spectral decomposition of a covariance matrix from real-valued random variables, we learn how to construct the covariance for the bivariate Gaussian. We compute the covariance matrix

\[ C_\psi = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} \xi \sigma_\psi^2 & 0 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix}^T, \]  

where \( 0 < \epsilon \ll 1 \) is an infinitesimal number.\(^d\)

This process can be visualized by first, constructing a bivariate Gaussian over some pixel \( p_{x_i, y_i} \). It has infinitesimal variance in the \( \hat{y} \) direction, and \( \xi \sigma_\psi^2 \) variance in the \( \hat{x} \) direction as shown in Figure 2. Second,
Doing this for each pose parameter gives us six bivariate Gaussian random variables for pixel $p_{xc_i, yc_i}$. We can then combine this information into a single bivariate Gaussian random variable. We want to obtain

$$f(x,y) = p_{xc_i, yc_i} + \mathcal{N}(0, C_{tx}) + \mathcal{N}(0, C_{ty}) + \mathcal{N}(0, C_{\psi}) + \mathcal{N}(0, C_{\theta}) + \mathcal{N}(0, C_{\phi}).$$

(6)

If the six bivariate random variables are zero-mean Gaussian and we assume they are independent from each other, then their addition is a new bivariate Gaussian having zero-mean and covariance computed by adding the covariance matrices formed for each pose parameter. Thus Equation (6) becomes

$$f(x,y) = \mathcal{N}\left(p_{xc_i}, \begin{bmatrix} C_{tx} & C_{ty} \\ C_{ty} & C_{\psi} & C_{\theta} & C_{\phi} \end{bmatrix}\right),$$

(7)

representing the blurring function. This corresponds to a convolution of the six bivariate Gaussian random variables and can be seen in Figure 4.

We now compute this for each pixel and use these blurring functions to blur the image to obtain the uncertainty image. The uncertainty image represents our uncertainty in the geolocation of the fire.
III. Georeferenced Uncertainty Mosaic (GUM)

Once we have uncertainty images we can composite them in a meaningful way to reduce the error in geolocation estimates of the fire. We use the uncertainty images as observation densities in a particle filter in a mosaic-like fashion to produce a particle mosaic, from which we form a GUM.

A. Problem Setup

Let $\mathbf{x}_k$ be the state vector at time $k$, where the state is the geolocations of fire. Let $\mathbf{y}_k$ represent the observation at time $k$. We have no prior information about where the fire is before beginning the filter and thus the prior is simply a uniform distribution $p(\mathbf{x}_0)$ over the region of interest. Our model is described by

$$p(\mathbf{x}_0),$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) \text{ for } k \geq 1, \text{ and}$$

$$p(\mathbf{y}_k|\mathbf{x}_k) \text{ for } k \geq 1.$$  

We desire to find the posterior distribution $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, describing the probability of fire at each geolocation. This is the classical Bayesian filtering problem and represents all the information we have about the state $\mathbf{x}$ at any time $k$. The solution to this problem is represented by

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k})}{\int p(\mathbf{y}_{1:k}|\mathbf{x}_{0:k}) p(\mathbf{x}_{0:k}) \, d\mathbf{x}_{0:k}}.$$  

This solution is very difficult — nigh impossible — to sample from. Therefore we seek a method for approximating or otherwise finding this distribution.

B. Filtering Uncertainty Images Using a Particle Filter

We have chosen to use a Sequential Monte Carlo method (particle filter) as such methods are fast, can represent arbitrary distributions, have modest memory and computational requirements, and they are relatively simple to implement. In this work we have specifically implemented the well known Bootstrap filter.\(^{10}\) In such Monte Carlo methods, particles (discrete samples) are used to represent the posterior distribution. This is justified by the well known fact that any distribution can be approximately represented by enough discrete samples of it.\(^{10}\)

Because this is a recursive Bayesian state estimation scheme we are interested in the marginal density $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, also known as the filtering distribution. This posterior represents the current state of the system at time $k$ given all past observations. This posterior can be broken up into a recursive formula to produce this density. The recursive formula has a prediction step:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \, d\mathbf{x}_{k-1},$$  

and an update step:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})},$$

where we assume the measurement $\mathbf{y}_k$ is conditionally independent of earlier measurements $\mathbf{y}_{1:k-1}$ given $\mathbf{x}_k$. This means we can recursively estimate the current state of the system given the dynamical motion model and all past observations. We outline our approach for each step in the two sections that follow.

1. Prediction Step

Let $\mathcal{S}_{k-1}$ represent the set of particles from the previous iteration. Because the fire does not move quickly, or in a predictable fashion, our motion model will consist entirely of noise in the geolocation of the fire. This is an important step as it allows for Brownian motion of the particles, thus preserving the integrity of the filter.

Let $s^i_{k-1}$ represent a particle in the set. For each particle we compute its new $(x, y)$ position as

$$\begin{bmatrix} x^i_{k+1} \\ y^i_{k+1} \end{bmatrix} = \begin{bmatrix} x^i_{k-1} \\ y^i_{k-1} \end{bmatrix} + \begin{bmatrix} \mathcal{N}(0, \sigma^2_x) \\ \mathcal{N}(0, \sigma^2_y) \end{bmatrix}.$$  

6 of 13

American Institute of Aeronautics and Astronautics
This provides the marginal distribution \( p(x_k|y_{1:k-1}) \) and new set of particles \( S'_k \). Note that we have not yet included any observation information.

2. Update Step

In this step we use the Sampling Importance Resampling approach to update the set \( S'_k \) from the prediction step to obtain the posterior at time \( k \). Because of the way in which we constructed the likelihood images, they are a representation of the likelihood of fire at a geolocation point. Therefore, we use the uncertainty image created from blurring as the likelihood function \( p(y_k|x_k) \) for the filter. Because each particle lies on the ground plane we use the homography matrix constructed previously to warp each particle onto the imaging plane of the camera,

\[
\lambda \begin{bmatrix} x_{ci} \\ y_{ci} \\ 1 \end{bmatrix} = H \begin{bmatrix} x'_{si,k} \\ y'_{si,k} \\ 1 \end{bmatrix},
\]

where \( \lambda \) is an unknown scaling factor. We can now easily update the weight of each particle. Let \( p_{x_{ci},y_{ci}} \) represent the value of the pixel in the uncertainty image at location \( x_{ci}, y_{ci} \). Also let \( w_{s'_i,k} \) represent the weight currently assigned to particle \( s'_i,k \). We compute the particle’s new weight

\[
w_{s'_i,k} = p_{x_{ci},y_{ci}} w_{s'_i,k}.
\]

This is done for each particle giving us a new set of weighted particles \( S'_k \).

We now resample to obtain the posterior set of particles \( S_k \). In the resampling step we sample from the cumulative distribution formed by the weights assigned to the particles to move the particles in low probability regions to regions of higher probability and normalize the weight of each particle. Let \( N \) represent the number of particles. To resample:

\[
\text{for } i = 1...N, \\
\quad \text{draw one sample } s'_i,k \text{ from } S'_k \\
\quad \text{Normalize weights of particles.}
\]

This gives us the mosaic of particles \( S_k \) representing the posterior \( p(x_k|y_{1:k}) \).

Intuitively, in the prediction step we are changing each particle’s \((x, y)\) location in the mosaic to introduce random motion and allow the particles to “explore” the region. In the update step we are updating each particle’s probability of a fire at that location and then by resampling allowing particles to move to regions of higher probability.

C. Mosaicking with Particles

Creating a mosaic with particles presents some unique challenges. The first issue deals with adding new particles. With each new image the camera will not in general image the exact same scene. This means with each image we could potentially see more fire. Therefore we need to add new particles accordingly. This is done by warping the image down to the ground plane using the homography created. We then compute the new area seen by the camera and add the appropriate particles.

The second issue is more challenging. If the current observation does not provide any new information to a portion of the mosaic of particles then we shouldn’t update particles from those areas. This requires care in creating the cumulative distribution and subsequently performing the resampling.

The third issue has to do with presenting the mosaic of particles to the user as a GUM. When using a particle filter, particles will congregate to regions of higher probability. To convert the final particle mosaic into a GUM we determine the number of particles within each pixel region, and scale it by the pixel with the maximum number of particles. This produces a GUM in which the brightness of each pixel is a representation of the confidence that a fire is there.

IV. Results

In the Multiple AGent Intelligent Coordination and Control (MAGICC) Lab at Brigham Young University, we have developed an extensive set of tools to simulate as well as fly MAVs for various missions.
To simulate the flight of a MAV, a flight simulator, Aviones, has been developed to accurately describe the physics, noise, jitter, and other characteristics associated with true flight of such an aircraft. Furthermore, a ground station application, Virtual Cockpit has been developed enabling uploading of waypoints, analysis of data, and video capture.

A. Geolocation Accuracy

In these results we show that our method is comparable to work done by Merino et al. in geolocation accuracy. To produce these results we drew a pure white circle on a black background at a known geolocation in Aviones. We then simulated the MAV’s flight in a loiter about the center of the circle and captured both the video and the noisy telemetry (pose information) using Virtual Cockpit. This accurately simulates our true flight characteristics including non-synchronized video frames and telemetry, as well as jitter from atmospheric disturbances.

We then ran our algorithm, blurring each frame with associated telemetry information and then passing the frame to the particle filter. In Figure 5 we show one original frame from the video sequence with its blurred counterpart. In the blurred image in Figure 5(b) the gray boundary near the edges of the image represents the portion of the image that was not blurred since the blurring function required information from pixels outside the boundaries of the image.

![Image before blurring](image1.png) ![Image after blurring](image2.png)

Figure 5. (a) Original image of simulated fire from video. (b) Image after mathematical correlation with blurring function.

In Figure 6 we show two different views of the final GUM. In Figure 6(b) is the East vs. number of particles view showing accuracy of the majority of particles in the east direction. The black lines representing the true geolocation of the circular fire. In Figure 6(a) is the North vs. number of particles view showing accuracy in the north direction. The accuracy is characterized in Table 1.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Error (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>4.1992 m</td>
</tr>
<tr>
<td>North</td>
<td>3.0620 m</td>
</tr>
<tr>
<td>Total</td>
<td>5.1970 m</td>
</tr>
</tbody>
</table>

Table 1. Geolocation Accuracy

In Figure 7 we show the East vs. North view. The black circle is the true size, shape, and geolocation of the fire. The white circle encompasses the region of highest probability and is the estimated size, shape, and geolocation of the fire from the GUM. Note the particles outside the regions of higher probability which show the uncertainty in the geolocation of the fire. These regions of lower probability are conservative enough...
that they cover the true fire. This would enable the system operator to ascertain where the fire is and assess his confidence in that selection.

B. Arbitrary Size and Shape Accuracy

In these results we show that this method accurately reconstructs the shape of two different arbitrarily shaped fires at different geolocations. Once again using Aviones we drew two arbitrary fires and then simulated the MAV’s flight loitering about the fire. An image collected from the simulated flight is shown in Figure 8(a). A properly blurred version of this image using our technique is shown in Figure 8(b).

We then passed the blurred images into the particle filtering portion of the algorithm to obtain the GUM.

C. Real Flight Results

To obtain the real flight results we flew a MAV over a nearby park with a downward facing infrared camera. Due to the time of year it was not possible to construct a controlled fire to test our algorithm. To compensate we were in flight by 5:30 a.m. in order to have a higher contrast of heat sources. In other words, we wanted to have something hot against something cool in order to simulate a real fire.

At the park the grass was still wet from the watering schedule but the trees were emitting a significant amount of heat. Therefore, we flew a path, East to West, following a fence along which several trees were planted. In Figure 10(a) we show one frame of the infrared video taken from this flight. The bright white spots in the image represent the trees. The bright line forming a corner is the fence over which we flew. The regions of darker gray and black represent the colder, freshly watered grass.

We gathered this video along with associated telemetry and then applied our algorithm. In Figure 10(b) we show a blurred version of the original frame. The final GUM is shown in Figure 11. The brightest tree in the video (lower right in Figure 10(a)) was reconstructed (furthest Northeast bright spot in the final GUM) along with several other smaller trees in the infrared video. The tree just coming into view at the top of Figure 10(a) is not represented very well. This is because the fence also has heat associated with it, is larger in area, and therefore demands more particles in its reconstruction. The algorithm has done an excellent job reconstructing the size, shape, and geolocation of the hot spots in the video. It has also preserved distances between objects as well as orientation.
Figure 7. East vs. North view of GUM.

Figure 8. (a) Original image of two fires. (b) Properly blurred version.
Figure 9. East vs. North view of GUM showing reconstruction of two fires of arbitrary shape and size.

Figure 10. A side-by-side comparison of an infrared image and its blurred counterpart. (a) A frame of the original infrared video taken during flight. (b) The same frame from the video after applying the proper blurring.
Figure 11. An East vs. North view of the final GUM formed from processing the infrared video obtained from real flight.
V. Conclusions and Future Work

In this paper we have outlined our approach to process the infrared video collected from tracking a fire using multiple MAVs. We introduced a novel method of converting uncertainty in pose estimation of the camera to the image domain to produce an observation density for a particle filter. We also described a new method of mosaicking using a particle filter and addressed challenges involved in their creation. We introduced the concept of a Georeferenced Uncertainty Mosaic which represents the confidence of the location of a fire and is constructed from a particle mosaic.

Our method provides several key advantages over existing methods, namely, the shape, size, geolocation and uncertainty information can all be seen simultaneously in the GUM. Furthermore, we do not discard useful information through segmentation and thresholding, but rather, we incorporate all levels of “heat” or intensity in the image.

There are however, some problems that need to be addressed. First, the blurring function is especially slow and consumes a lot of system resources. Work could be done to determine if a non-spatially varying blurring function would suffice. Additionally, whereas our method does not discard information by segmenting or thresholding there is a tight coupling of location confidence and heat intensity introduced by the blurring function. Work could be done to filter a luminance value as well as geolocation thereby allowing the GUM to report size, shape, geolocation, geolocation uncertainty, and a heat or luminance value simultaneously.

Finally, there are many improvements which can be made to the Bootstrap filter method of particle filtering to allow it to compute the posterior faster and with fewer observations while still maintaining the integrity of the filter.

VI. Acknowledgments

We would like to thank a few individuals. Evan Andersen for help with obtaining real flight results, and Bryce Ready for lengthy discussions on various related topics. We would also like to acknowledge that the work in this paper was supported in part by a grant from AFRL/SNAR.

References