Prototype Sense-and-Avoid System for UAVs

Christopher Geyer
Debadeepta Dey
Sanjiv Singh

CMU-RI-TR-09-09

February 2009

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

In this technical report we describe our efforts towards a field deployable Sense and Avoid system for Unmanned Aerial Vehicles (UAVs) which uses passive vision as the main sensing modality. The ability of UAVs to operate safely in the presence of other air traffic is critical towards acceptance of UAVs in civilian and military airspace. This will allow UAVs to be used to their fullest potential. In order to operate freely in the presence of manned airborne traffic UAVs must demonstrate a Sense and Avoid capability that meets or exceeds that of an equivalent human pilot. Furthermore this capability should be achieved without the use of cooperative communication with other aircraft or prior knowledge of other aircrafts' flight plans. We describe our collision avoidance algorithm and software-in-the-loop testing, vision based detection method with a 98% detection rate out to a range of 4.5 miles which exceeds the FAA regulation of 3 statute miles. A field deployable Sense and Avoid system must be able to operate with consistent performance across a variety of atmospheric conditions including cloud, fog and haze of various degrees that can occur under conditions commonly described as Visual Meteorological Conditions (VMC). In order to model the effect of all these conditions on the performance of the detection system we developed an atmospheric image formation modelling system that takes as inputs weather conditions and can predict the appearance of the image of an aircraft of given geometry at various ranges. This predictive model also determines the minimum resolution needed to guarantee the required detection performance. Passive vision provides the bearing to the intruding aircraft. Range estimation is only possible by executing additional maneuvers which causes significant mission interference. We investigate the feasibilty of a flash lidar system that can be used as a confirming sensor to further reduce the false positive rate and provide range of the intruding aircraft.

Contents

List of Figures

List of Tables

1 Introduction

In this project we considered the ability of UAVs to operate safely in civilian airspace in the presence of other air traffic. This ability is increasingly important as UAVs become more widespread and as non-military applications are discovered. "Sense and Avoid" and "See and Avoid" are terms used to describe the capability of a UAV to detect airborne traffic and respond with appropriate avoidance maneuvers so as to maintain minimum separation distances. "See and Avoid" specifically refers to image-based methods for detecting other aircraft. Though development of requirements for sense and avoid systems is ongoing, there is generally agreement that this capability must demonstrate a level of performance that meets or exceeds that of an equivalent human pilot. Furthermore this capability should be achieved without the use of cooperative communication with other aircraft or prior knowledge of other aircrafts' flight plans. Until this capability is provided, the widespread usage of UAVs will not be possible.

In part because of the developments in Iraq and Afghanistan, the market for UASs has grown significantly in the last several years. In 2002, the Department of Defense had in its inventories only 200 UASs. Today it has over $6{,}000$ [6], and its budget for unmanned systems (including ground) is $\$15$ billion. As of 2005, the DoD was operating 19 different types of UASs in Iraq [1]. They range in size and capability, from Aerovironment's $4.2\,\mathrm{lb}$ RQ-11 Raven with cruising speed of $50\,\mathrm{knots}$; to AAI's $200\,\mathrm{lb}$ RQ-7 Shadow; to General Atomics' approximately $1\,\mathrm{ton}$ MQ-1 Predator. One notable and highly publicized success has been the use of Predator/Shadow teams to identify insurgents placing IEDs and track mortar teams [5]. There has been, moreover, significant enthusiasm for UASs outside of the DoD. The Department of Homeland security has been using Predators to patrol the U.S.-Mexico and U.S.-Canada borders [8]. In addition, local police departments across the country are eager to use UASs. Studies have shown that law enforcement pursuits by helicopter have helped improve apprehension rates by 15% [4] these pursuits could eventually be done by UASs more cheaply. Furthermore, there are surely many potential non-security applications and markets, ranging from using UASs instead of traffic helicopters to report on traffic jams, to using UASs to monitor hazardous weather. For example, in 2007, NASA used a Predator to help fight wildfires in California [3].

There are, however, several obstacles to greater proliferation of UASs. Though there is demand for UASs in local law enforcement agencies, the FAA is obligated to maintain the safety of the U.S. National Air Space. The FAA turned down an application by the Palm Beach Police Department to fly a Cyberbug UAV because it did not have a waiver [10]. Though the FAA cleared the Predator to fly over California wildfires, it rejected a request to use a Predator for search and rescue missions in the aftermath of Hurricane Katrina. Even in Iraq, there are obstacles to UAS use. According to one report, a commander gave up trying to use Raven and Dragon Eye UAVs because of delays in obtaining flight clearances [18]. The clearances aimed to prevent collisions with helicopters. Besides bureaucratic obstacles, back in the U.S., private pilots are concerned by the thought of sharing the same airspace with UASs, and AOPA, the Aircraft Owners and Pilots Association has testified to Congress objecting to the temporary flight restrictions (TFRs) that have been used to dedicate to airspace for UASs performing border patrol [2].

The FAA and private pilots share the concern that UASs are not yet safe enough for the U.S. National Air Space (NAS). They need to first demonstrate the ability to see and avoid other air traffic, and then make a strong case with hard numbers that UASs will not increase risk of mid-air collisions in NAS. Until a sense and avoid capability can be provided, we believe the potential for growth in the UAS market will be hampered. It has been our aim in this effort to significantly advance the state-of-the-art in sense and avoid technologies.

In this project, begun in November 2007, we considered both the sense and avoid problems. In the case of an avoid system, we aimed to address some of the complex issues in designing a safe collision avoidance system. For example, one difficulty in developing a safe collision avoidance is how to produce approaches that appropriately decide on an evasion maneuver despite the unknown future actions of the intruder aircraft. Furthermore, the sense and avoid problems are intertwined, in that the maximum performance of the collision avoidance system in part determines the minimum requirements of the sense or aircraft detection system.

The sensing problem presents significant problems because of required minimum ranges, as well as required field-of-regard. We therefore spent a large percentage of effort to better understand performance as a function of lens and imager choices. However, we made significant progress in aircraft detection, and reducing the number of false positives. We believe that with these results that we have come closer to realizing the goal of making UASs safe users of both civilian and military air space.

In the rest of this report we present our findings in collision avoidance and aircraft detection. The next section summarizes our accomplishments during the project. Then, because the collision avoidance performance determines

in part range requirements for aircraft detection, we first discuss results on and analysis of a collision avoidance prototype. We then present some of the results in aircraft detection, and proposals for the design of an aircraft detection system.

2 Summary of Accomplishments

Here is a summary of our accomplishments in the primary tasks:

- **Task 1: Prototype Aircraft Detection and Collision Warning System.** The primary goal of this task was to develop a method to detect aircraft using primarily passive imagers, and decide whether a detected aircraft was on a collision course. We planned to test multiple sensors and algorithms, and choose among them the best performer. Using regulation proposed for UASs, we aimed for a detection range of $3\,\mathrm{miles}$.
  - **Acquired large corpus of aircraft imagery.** We worked with Penn State EOC to collect data from five different EO imagers (Lumenera Lu125, Lumenera Lw625, Imperx IPX-4MP15, Imperx IPX-16MP3, and Imperx IPX-2MP30) combined with a number of different lenses (Zeiss 50mm and 85mm, Nikon 60mm and 105mm, Computar 25-36mm and 25mm, and Fujinon 16mm and 25mm), as well as three IR imagers (Thermoteknix Miricle LWIR, DIOP, and Lumitron). We collected this data both on-board EOC's Cessna aircraft, as well as on the ground using a pan-tilt unit.
  - **Evaluated the sensors by comparing them with the results of an image formation model.** We developed a model of the formation of images of the aircraft that takes into account the relevant optical and atmospheric effects. We found that this model reflected the trends in the data, though the fits to the measured data revealed systematic biases. However, we found that biases from multiple data sets corroborated each other, thereby exposing real performance differences in the lenses and cameras. We used the measured performances to arrive at a minimum resolution of $0.2\,\mathrm{mrad}/\,\mathrm{pixel}$ to detect the Piper Archer II at $3\,\mathrm{miles}$.
  - **Developed a prototype aircraft detection system.** We tested a number of different approaches to aircraft detection and for each we evaluated their performance characteristics. We settled on an approach based on morphological image processing and machine learning techniques to reduce false positives. The code base now totals over $17,000$ source lines of code (SLOC) and by one estimate could have required an equivalent 4.4 man years of code. We are providing the source code and other relevant material to EOC.
  - **Evaluated the performance of the aircraft detection system.** We evaluated the performance of the algorithm in terms of true positive percentage and false positive rate measured in number of false positives per frame. We also broke down performance by range. As of this writing the approach yields a detection rates of 99.7% up to $3.5\,\mathrm{miles}$, and 96.1% up to $4.5\,\mathrm{miles}$, with a false positive rate of 0.02 per frame (using an IPX-4MP15 and Zeiss 85mm lens, thereby yielding 0.005 false positives per megapixel per frame). Furthermore almost all of the false positives that occur are due to objects that may be relevant to collision avoidance: birds, radio towers. Section 4.7 of this report provides further details on these results.
- **Task 2: Prototype a Collision Avoidance System.** In this task we aimed to develop and test approaches to collision avoidance. Keeping in mind the Viking 100 UAV, we approached the problem of how to avoid a collision with another aircraft even though we would not know where in the future it plans to turn. In this task we have accomplished the following:
  - **Developed a prototype collision avoidance system.** We developed a collision system that could avoid another aircraft traveling up to $250\,\mathrm{knots}$ and making lateral maneuvers no more than $1\,\mathrm{g}$, as long as the separation between ownship and intruder aircraft is at least $2.1\,\mathrm{km}$ (as long as the ownship's speed is greater than $40\,\mathrm{knots}$). The system is based on approach called *rapidly-exploring random tree* and *maneuver automata*, which we describe in section 3.1. We also characterize this system's performance in this report (Section 3.2;

  - **Interfaced with the Piccolo autopilot.** We integrated the collision avoidance algorithms with a Piccolo autopilot, and demonstrated a software-in-the-loop (SIL) simulation of the collision avoidance system.

- **Task 2: Investigate Alternative Technologies.** We also pursued other possible technologies for detection, including ladar and radar.

  - **Tested a flash ladar by Advanced Scientific Concepts.** In a separate project with Advanced Scientific Concepts (ASC), and with EOC's help, we collected flash ladar imagery of an aircraft making passes in front of Jimmy Stewart Airport in Indiana, PA. We were able to pick up the aircraft as far as $3{,}000\,\mathrm{ft}$ away. We demonstrated some limited ability to detect aircraft from this imagery.

2.1 Highlights

Some highlights from the report follow here:

- The best detection rate and false positive amount for $3\,\mathrm{miles}$ was 100% and 0.05 false positive per frame, found with the IPX-4M15 camera and Zeiss $85\,\mathrm{mm}$ lens.
- With the same camera and false positive rate the detection rate is 98.1% at $5\,\mathrm{miles}$.
- The lowest false positive rate is 0.014 false positives per frame with an overall detection rate of 92%.
- Holding a 95% detection rate constant, we achieved a nearly $3{,}400$ factor improvement in false positive rate over the initial detector.
- We found that the minimum angular resolution need not be any less than $0.2\,\mathrm{mrad}/\,\mathrm{pixel}$.
- We found that if the signal-to-background ratio can be guaranteed to be at least $1\,\mathrm{dB}$, or equivalently the contrast is at least 26%, then the detection rate would be at least 95% with 1.7 false positives per frame.
- We found that most of the false positives detected after tracking were birds or ground landmarks, such as an antenna.
- Though our implementation of the algorithm runs at $0.8\,\mathrm{sec}$ on $4\,\mathrm{megapixel}$ frames, 90% of the computation is taken up by image processing, which could be computed with special-purpose hardware (e.g., GPU, DSP or FPGA).
- As long as the intruder aircraft is at least $2.1\,\mathrm{km}$ away, is flying less than $250\,\mathrm{knots}$, makes maneuvers not exceeding $1\,\mathrm{g}$, and our ownship's speed is at least $40\,\mathrm{knots}$, then we can come up with an avoidance maneuver within $500\,\mathrm{ms}$.

3 Collision Avoidance

The aim of the second task was to develop a collision avoidance system that given the state of the intruder aircraft would be able to come up with an avoidance trajectory in finite computational time. Any system that performs satisfactorily for this purpose must satisfy a number of performance criteria.It must be able to take into account the uncertainty in the state estimation of the intruder aircraft, utilize the dynamic extents of the ownship as completely as possible in evading the intruder while at the same time come up with a plan that is feasible to execute for the ownship within some finite time. Conventional approaches have the drawback of being extremely slow as a consequence of trying to solve the search problem in a high dimensional state space. The resulting trajectory has the least possible execution cost but takes a long time to calculate. Unmanned Aerial Vehicles (UAVs) must be able to aggresively maneuver to avoid possible threats and obstacles if required while not breaching the flight envelope which can result in possible instability and uncontrollability.

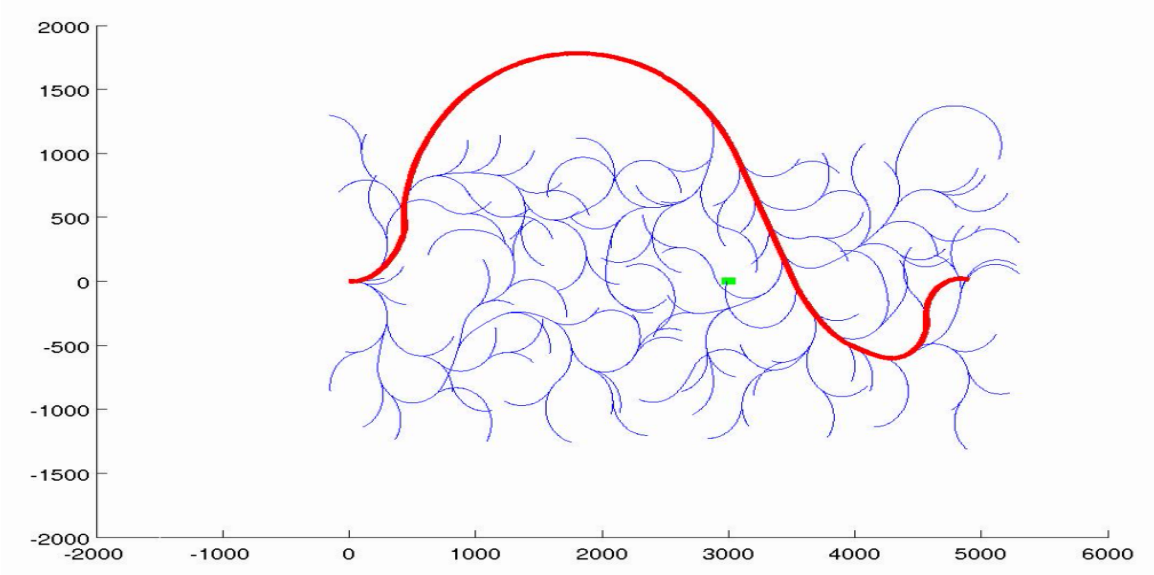


Figure 1: Shows the concatenation of maneuver primitives for generating feasible trajectories

3.1 Algorithm

The concept of discretizing the high dimensional state space of the UAV in a manner which enables us to solve the optimal control problem much more efficiently in finite time was developed in [9]. Traditionally control problems have been broken up into some arbitrary hierarchy with some major types of hierarchy as identified by [19]. The problem of these traditional approaches is that performance and stability of such systems are difficult to prove generally.In highly uncertain scenarios such heirarchical systems may not be able to respond quickly enough. Our system is essentially a hybrid control system with the states of the hybrid system representing feasible trajectory primitives. Given a set of feasible trajectory primitives any solution can be quickly arrived at by concatenating a suitable subset of the trajectory library to answer any query to the system. This library of trajectories can be generated by recording real life flight data for an aircraft piloted either by remote control or by a human inside the aircraft. This scheme of generating offline, a library of 'maneuvers' (Figure.1) is feasible because it has been found that pilots generally use a few control primitives to maneuver and these primitives are reused combinatorially to generate more complex trajectories. Also pilots generally stay most of the time in trajectories which can be described as relative equilibria trajectories or 'trim' trajectories where the velocities along the body axis of the aircraft remain constant. Example of a trim trajectory is straight line level flight or a turn with constant curvature. Pilots generally maneuver by transitioning from a trim trajectory, executing the maneuver and coming back to the same trim trajectory or some other trim trajectory. This can be thought of as a number of discrete mode switches like landing, takeoff, barrel roll, etc. This problem formulation lends itself very naturally to a hybrid control system.

To plan in the presence of moving or fixed obstacles we combined the maneuver automaton with a Rapidly Exploring Random Tree (RRT)[15] which is a single query planner which has been shown to be probabilistically complete for a given state space. The Rapidly Exploring Random Tree is grown from both the start and the goal states and at each iteration a random configuration is sampled from the configuration space. The nearest configuration to this sample is found and using a local planner an attempt is made to connect to either of the trees. If successful then this node is added to the tree. In this manner the trees are expanded. Both the trees grow towards each other and the local planner attempts to merge the two trees by attempting to find a path between the two closest nodes on both the trees. If successful the planner terminates else this process is repeated every few iterations of the expansion of both the trees. Once the two trees have been connected together it is easy to find the least cost path from the start point to the goal point by running A* [13] on the tree.

It must be noted that mutiple query planners e.g. Probabilistic Roadmaps (PRM) [14] build up a tree covering

the entire configuraton space in an offline tree building process. Any query for a given start and goal position can be simply answered by connecting the start and the goal to the nearest respective nodes in the tree and running A* on the tree to find the least cost path. The RRT is a single query planner that builds two trees from the start and the goal positions in real time and tries to connect both as soon as possible and terminates the moment it is successful.

We assume that the intruder aircrafts lateral acceleration is bounded. For then, the evolution of the intruder aircrafts state could be approximated by the linear evolution of a uncertainty ellipse. With the RRT we randomly generate a concatenation of constant bank angle turns, and transitions between bank angles, subject to the constraint that none of the generated paths intersect the avoidance set. The concatenation is not entirely random, in that we choose a random goal state within the state space, and then choose the a leaf of the tree closest to the random state. Then we generate the shortest path from the chosen leaf to the random goal state and add it to the tree as long as it does not intersect the avoidance set. This is an effective and efficient method for finding collision avoidance paths.

3.2 Performance

We present the summary of the performance capabilities of the collision avoidance system. We ran the planner with intruder velocities ranging from $50\,\mathrm{knots}$ to $250\,\mathrm{knots}$ and intruder lateral acceleration of $0.5\,\mathrm{g}$ and longitudinal acceleration of 0.1g for ownship speeds of 40, 50 and $60\,\mathrm{knots}$ and maximum lateral acceleration of 0.3g. For each scenario we generate a binary plot (Figure.4 5 6) where a red dot indicates that for that particular position of the intruder relative to the ownship at the origin, the planner failed to come up wth a feasible solution in $500\,\mathrm{ms}$ and a black dot indicates otherwise. So the boundaries of the red region in the plots centered around the ownship show the minimum distance in every direction at which an intruder must be detected in order to guarantee collision avoidance. A place was also marked as red if the system failed to generate a plan within in $500\,\mathrm{ms}$ of the query.

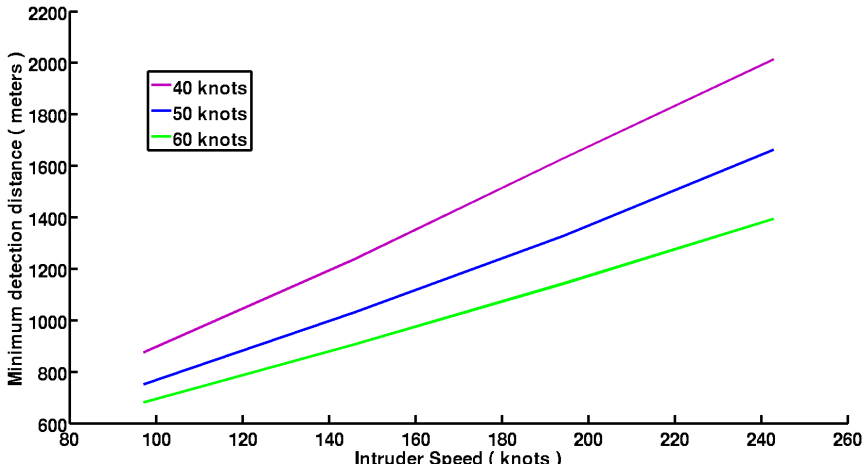


Figure 2: Plot of minimum distance (meters) for detection vs. intruder speed in $knots$ for different ownship speeds

Figure.2 shows the summary plot of minimum distance at which the intruder must be detected in order to guarantee a feasible plan to avoid the intruder even under the uncertainty in future state of the intruder. From the figure it can be seen that under the worst case scenario of intruder speed of $250\,\mathrm{knots}$ and ownship speeds of $40\,\mathrm{knots}$ the minimum distance of detection is approximately $2100\,\mathrm{m}$ and under the best case scenario it needs $700\,\mathrm{m}$ for an intruder speed of $80\,\mathrm{knots}$ and ownship speed of $60\,\mathrm{knots}$. It is to be noted that the minimum distance of detection decreases as the ownship speed increases because the ability of the ownship to evade an intruder increases in proportion.

Figure.3 shows the summary plot of minimum distance at which the intruder must be detected in order to guarantee a feasible plan to avoid the intruder under various values of lateral acceleration bounds for the ownship. It is evident from the plot that the minimum detection distance does not change as a function of lateral acceleration bounds of the intruder for a constant intruder ship speed. The reason for this is the fact that the shape of the red cone Figures(. 4 5 6) is changed as the lateral acceleration bound changes but there is no change in the maximum distance within the cone.

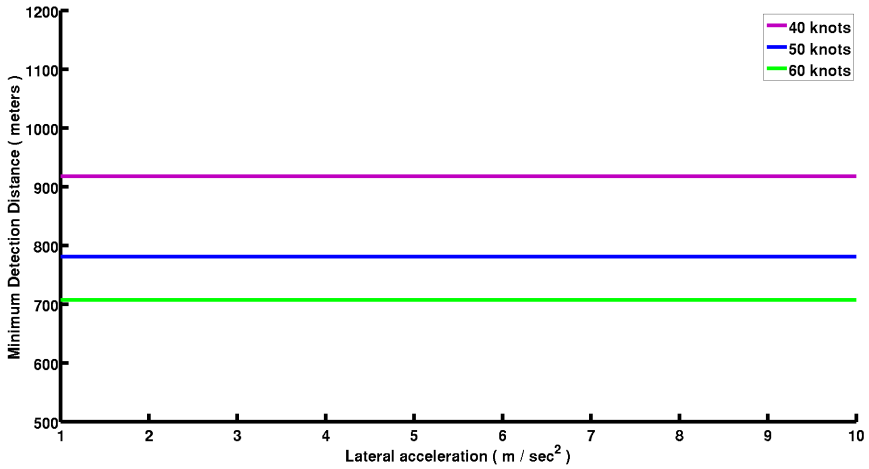


Figure 3: Plot of minimum distance (meters) for detection vs. lateral acceleration in m/sec^2 for different ownship speeds and constant intruder speed of $100\,\mathrm{knots}$

The minimum detection distance is the worst case scenario for detection to occur reliably in order to safely avoid the intruder aircraft. It must be emphasized that the minimum distance reported in these figures is the minimum distance for the planner to be able to avoid the possible threat reliably. Usually first detection must occur at a distance a bit more than the minimum distance required for the planner in order to allow time to track and establish the identity and state of the intruder.

3.3 Summary

We have implemented and characterized a collision avoidance system that is capable of returning trajectory plans in finite computational time while handling the future action uncertainty of the intruder. We have also characterized the limits of performance of the system and related the performance of the planner with the performance criteria imposed on the detection system.

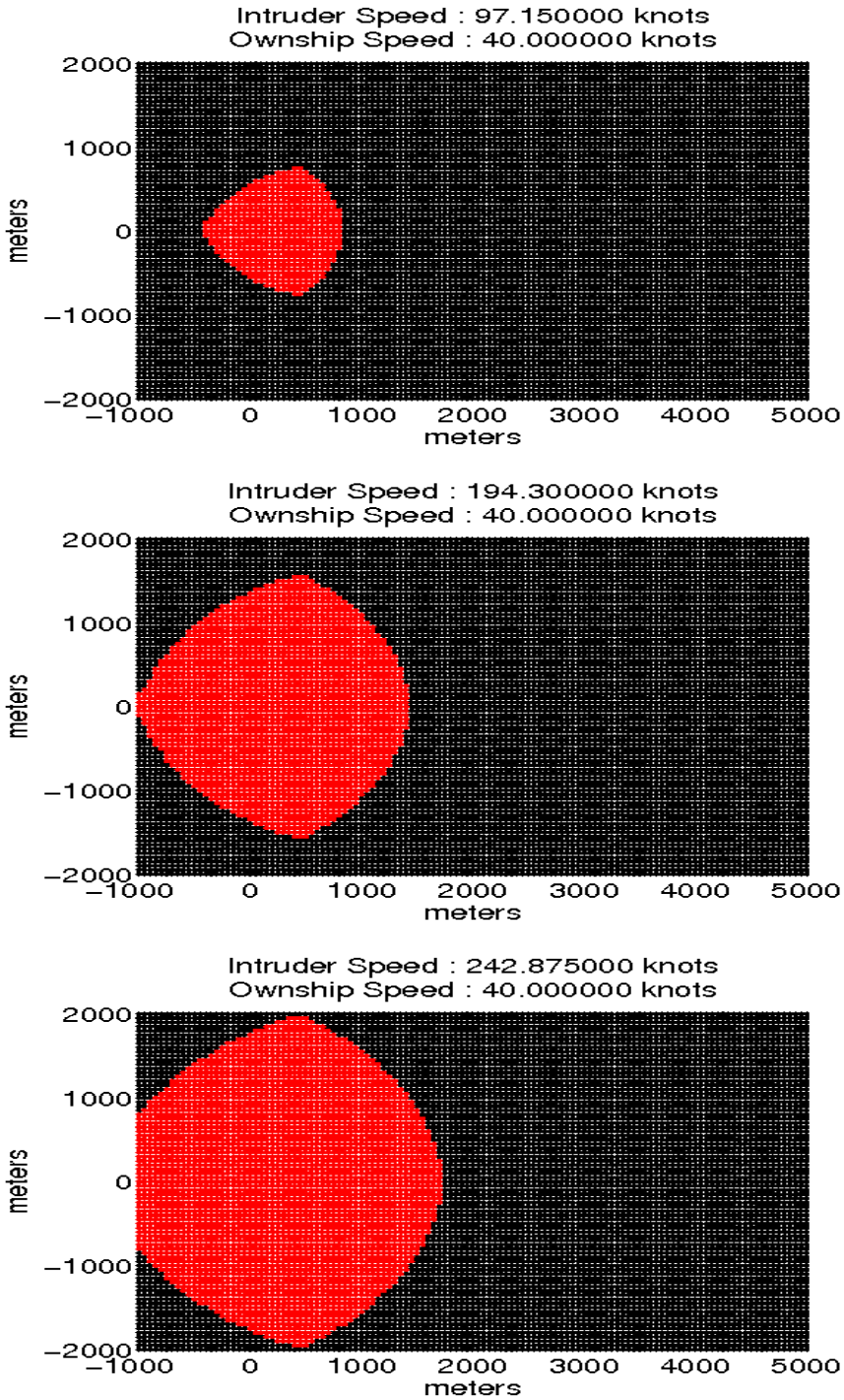


Figure 4: Binary plots showing minimum distance (meters) for various dynamic parameters of the intruder and ownship speed of 40 knots

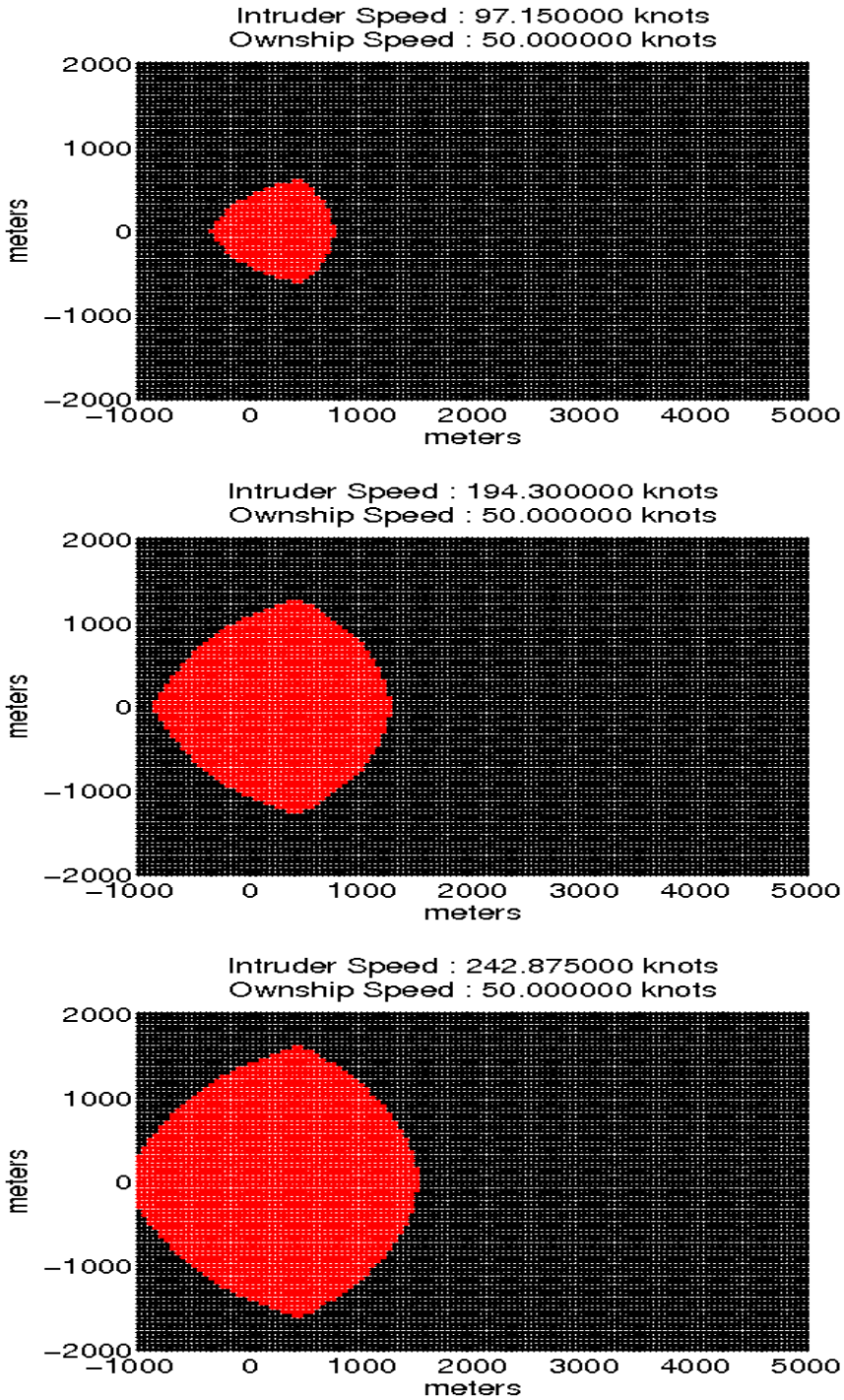


Figure 5: Binary plots showing minimum distance (meters) for various dynamic parameters of the intruder and ownship speed of 50 knots

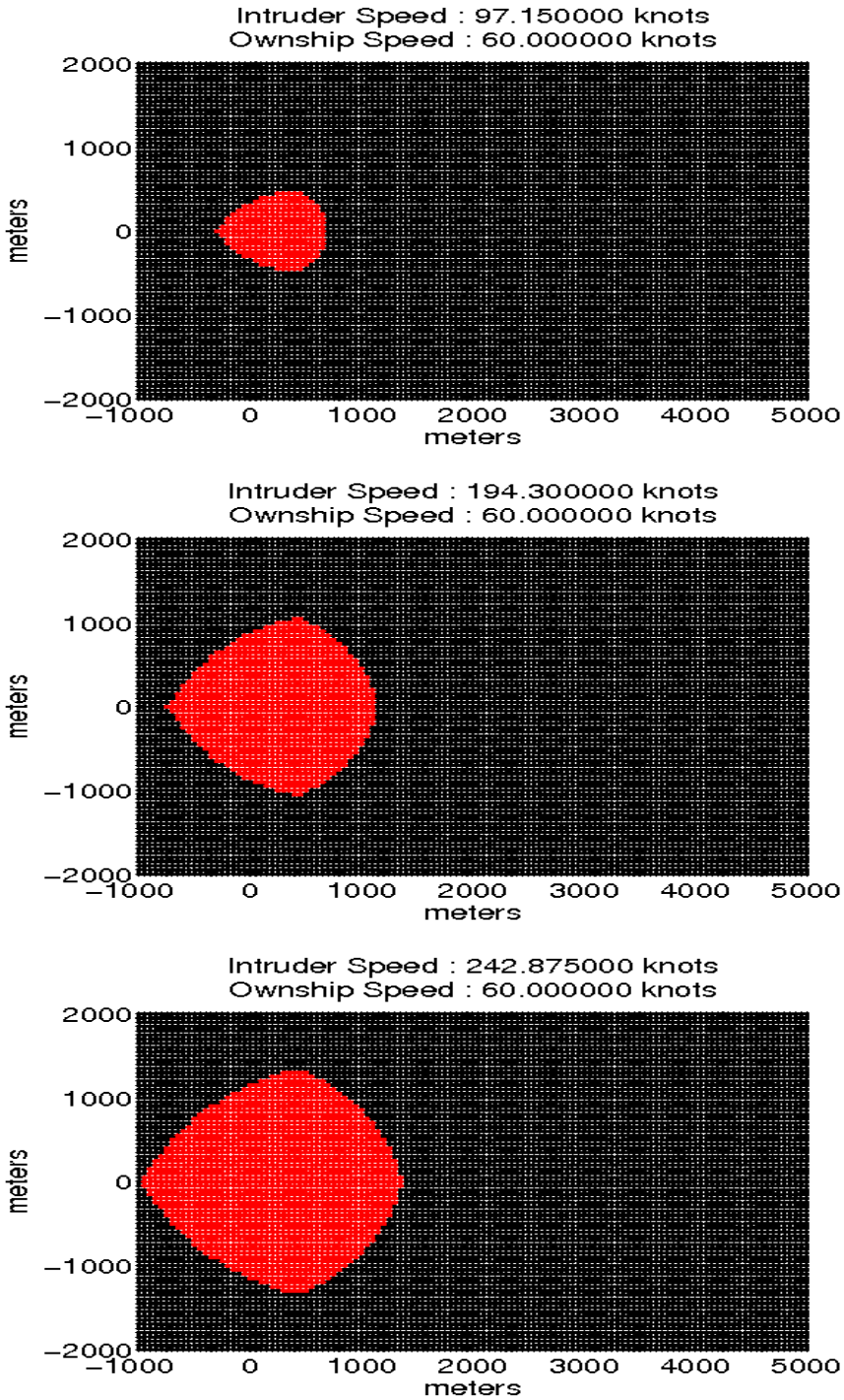


Figure 6: Binary plots showing minimum distance (meters) for various dynamic parameters of the intruder and ownship speed of 60 knots

4 Aircraft Detection

In this section we report on the development of the primary component of a collision detection system, in particular, an aircraft detection and tracking system. In the first section we describe some of the factors that affect performance, and that we considered before embarking on developing the detection and tracking system. We then consider some of the results that we found in early experiments, and make some observations about the characteristics of aircraft targets viewed from a distance. These observations lead to a model of aircraft image formation, explained in section 4.3, which we validate in section 4.4. These lead to some conclusions with regard to what the minimum resolution should be in section 4.5. In section 4.6 we describe the detection and tracking algorithms that we implemented, and in section 4.7, we present an analysis of their performance.

4.1 Performance Factors

We seek in this section to better understand those factors that will affect the performance of an aircraft detection and tracking system. The purpose of this is to decouple those factors that are controllable, from those that are not. Atmospheric factors, for example, are not controllable. The choice of imager, optics, and detection algorithms are controllable.

We argue that the effects on the performance of an aircraft detection system can be decoupled into controllable system factors and uncontrollable environmental factors; these dependencies are shown in **Figure 7**. Environmental factors include atmospheric properties, in particular visibility due to weather conditions and attenuation in the atmosphere; lighting conditions; and intruder aircraft properties. For all of the uncontrollable factors, we have to be able to accommodate any range of these factors that are within the necessary or desired operating range for the system. For example, if the visibility do not meet visual meteorological conditions (VMC), then we cannot plan to fly under visual flight rules (VFR). However, if the conditions are just above VMC, i.e., the visibility is $3.1\,\mathrm{miles}$, then we the sense and avoid system still must be able to detect aircraft out to at least $3\,\mathrm{miles}$. Similarly, if there is sufficient lighting that the UAV is flying VFR, then no matter how dark it is, we still ought to be able to detect out to $3\,\mathrm{miles}$.

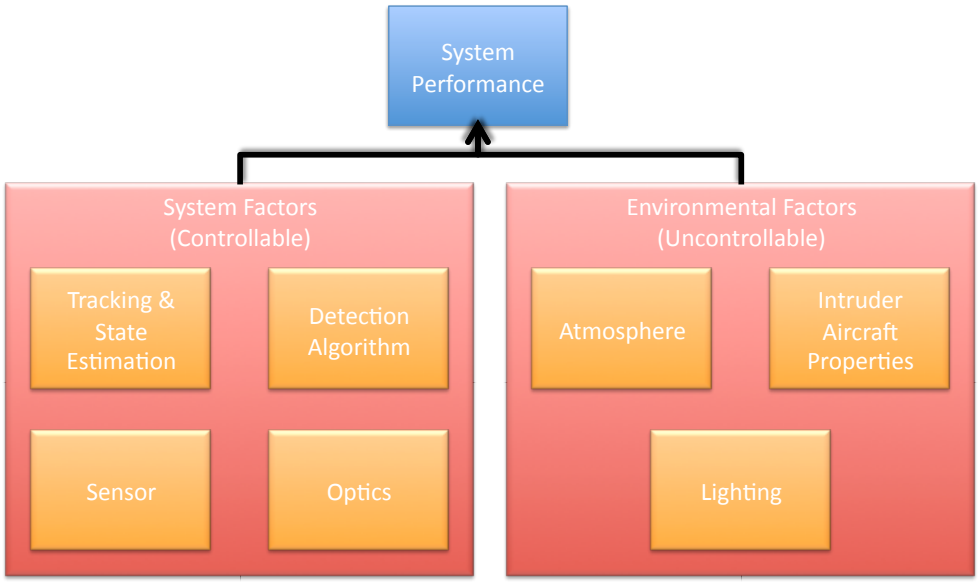


Figure 7: A diagram of factors determining the performance of an aircraft detection system. The factors that we have control over are system factors, in particular the design and selection of algorithms, sensors, optics. We have no control over atmosphere, lighting, and intruder aircraft properties, other than that we can assume that atmospheric and lighting conditions meet visual meteorological conditions (VMC).

Our first observation is that since the threshold objective is to enable a system to be equivalent in reliability and safety to a pilot operating under visual flight rules (VFR), we can assume that the environmental factors are no worse than the limits of visual meteorological conditions (VMC). In other words, though these factors may be uncontrollable, at least to meet a threshold objective, we can assume that these effects meet visual meteorological conditions.

As for the controllable system factors, for an aircraft detection approach based on imagery (though we expect later to possibly integrate other sensing modalities), the primary factors that affect detection performance in a single image are: signal and image processing, and the detection algorithms; the optics including lens focal length, aperture, etc.; and, the imaging device, including its pixel pitch, the exposure used, noise characteristics, and dynamic range, to name a few. In a complete system, numerous other factors affect performance some of which are dependent on the platform, the aircraft dynamics, for example, and where the resulting performance is then determined by the capabilities of tracking and state estimation algorithms.

Our second observation is that the controllable factors are either correlated or are decoupled. In particular, if the modulation transfer function (MTF) of the lens is appropriate for the resolution of the imager, then the resolution of the imager and optics determine a number of radians per pixel, which can then be varied. Furthermore, within a sensing modality (EO or IR), we expect the performance of the detector to be independent of the imaging device. The purpose is to reduce a combinatorial explosion of possibilities, and to characterize the detectors performance in quantities that are independent of the imager. Hence, in section 4.5 we examine the performance of the top-performing detector as a function of the signal-to-background ratio. Then, the signal-to-background ratio depends on the imaging device (optics and sensor), as well as the environmental factors. As long as the signal-to-background ratio generated under the worst conditions by the chosen imager is above threshold, then the combination of the detector with the imaging device is expected to perform above threshold. We take this argument at least as far as informing testing. It is certainly possible that coupling effects—atypical noise characteristics of an imager having affects on detection, for example—may cause deviations from predicted performance; if there are significant deviations from expected performance, then we ought to be able to understand what effects that have not been modeled could be contributing to inconsistencies in predictions.

4.2 Observed Target Characteristics

In an effort to decouple the performance of the algorithms from the environmental characteristics, in this section we present some of the characteristics of the data taken from the Lu125 camera during flights on April 3rd, 2008. We analyzed the images of the target in those images where it is above the horizon, and extracted from the images the 11×11 sub-windows containing the images of the aircraft when it was imaged above the horizon. **Figure 8** shows a random selection of the sub-windows containing the targets. We then performed a number of measurements of the target, and evaluated their trends as function of range.

- First, we observed that more distant targets have an intensity profile that is a small, smooth depression, like an inverted lump of sand. This depression resembles a two-dimensional Gaussian. Therefore, in our first step we fitted to each target an axis-aligned Gaussian, which is a function of the following form:

$$b + \frac{s}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right\} \tag{1}$$

  where b is the background intensity of the sky; s is the intensity or amplitude of the target in intensity; and σ_x and σ_y are the scales in the x- and y-directions, respectively. To find the best b, s, σ_x and σ_y, we choose initial values for them using heuristics, and then we use these initial values to initialize a gradient descent optimization of the sum of squared differences between the intensities of the synthesized image and the real image. In **Figure 9** we show an example of one such fit. The plot on the left is a graph of the intensity of each pixel in an 11×11 sub-window containing the image of the target. On the right we show the corresponding fit to the target. In this case we have inverted intensities to make the hump of the Gaussian visible. We did these types of fits for every target image, and computed the values that follow here from the fitted parameters.

- We calculate the *signal-to-background ratio* from the ratio of the target's peak to the background, in particular, $1 + |s|/(2\pi b\sigma_x\sigma_y)$, which we convert to decibels (by taking $10\log_{10}\cdot$). We plot the signal-to-background ratio as a function of range in **Figure 10**.

- We plot an estimate of the *target size in the image* as a function of range in **Figure 12** (left). We choose the target size to be the average of σ_x and σ_y.
- We plot an estimate of the *amplitude of the target signal* as a function of range in **Figure 12** (right). In this case the estimated amplitude is $|s|/b$.

Note that these trends are dependent on the conditions under which the data was taken; they should not be construed as an invariant over all possible weather conditions. For example, during the April 3rd flights, there was significant haze at flight altitude. This induced an "airlight" effect, i.e., where water particles in the atmosphere reflect light from the sun towards the observer, and thus act as a source of light [16]. Consequently the amount of incident energy in the direction of the target actually gets brighter the further away the target is, which is what we observed in the target data. This is an effect separate from atmospheric attenuation.

We take away two points from this data. First, **the target is generally dark**. Only rarely is the target lighter than its background, and this is especially true of distant targets. This suggests that the contrast in the image of the target is due not to the light being reflected by the target, but by its shadow. Especially in overcast conditions, the amount of light being scattered in the the background sky far exceeds the amount of light reflected by the target. So the target we see in the image is just the shadow cast by the aircraft. Since the light reflected by the target is not very significant, this further suggests that **atmospheric attenuation does not significantly affect the image**. Second, in the hazy conditions the image of the target is lighter than when not hazy the further away it is. This is probably because of the accumulated airlight in between the target and the sensor.

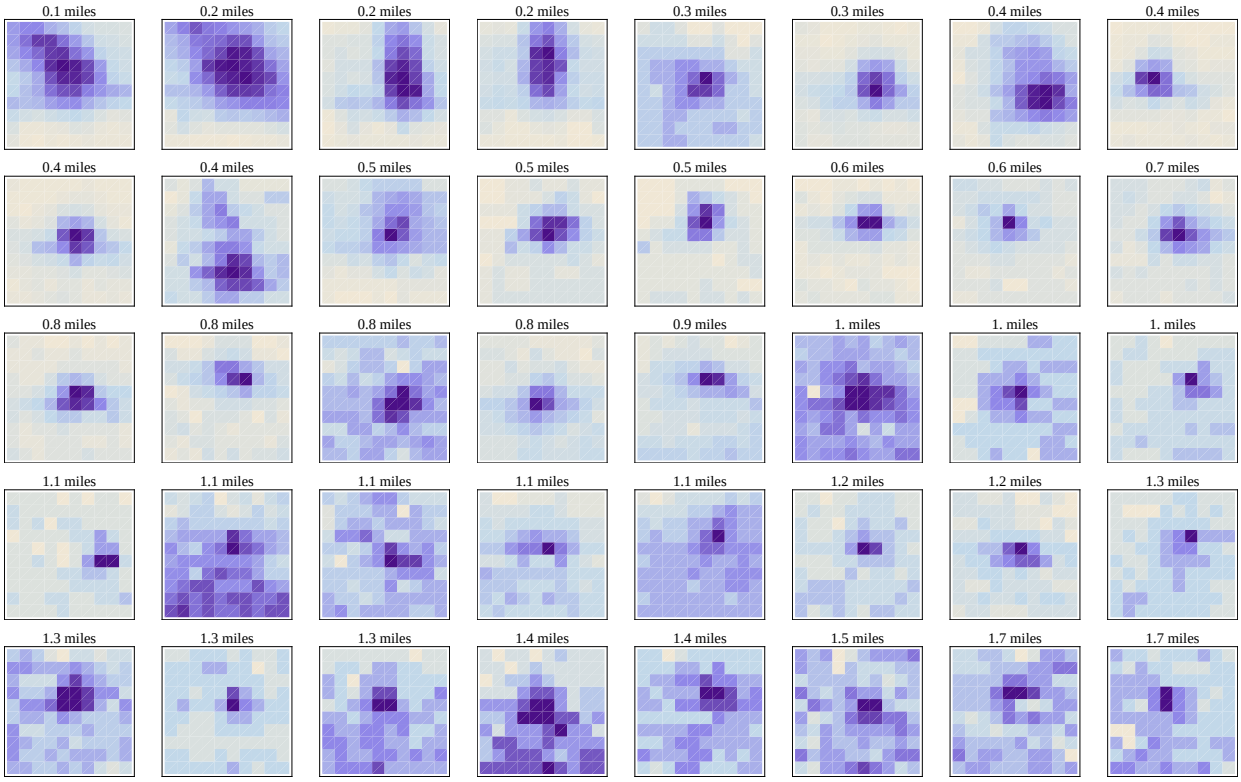


Figure 8: A random selection of 11×11 sub-windows containing the image of the aircraft when it is above the horizon during flights on April 3, 2008. The extracted images are sorted by range to the aircraft, and the corresponding ranges are given above the images.

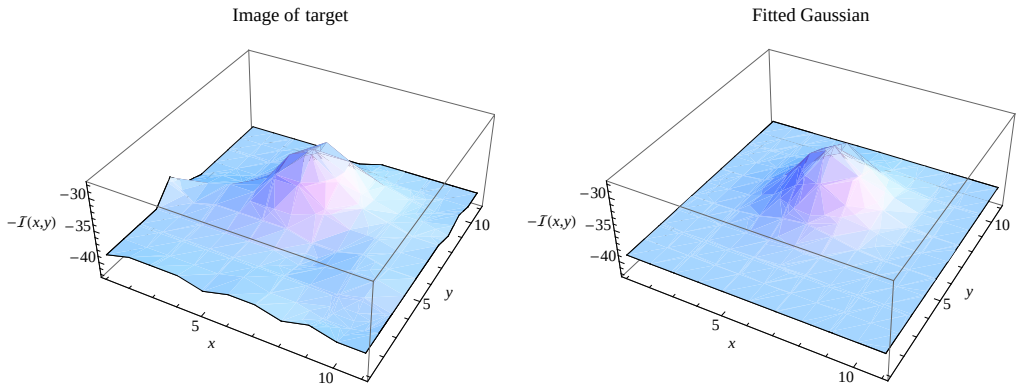


Figure 9: An example of an image of a target shown with inverted intensity. The target is the lump in the middle of the height field. This is an 11×11 image.

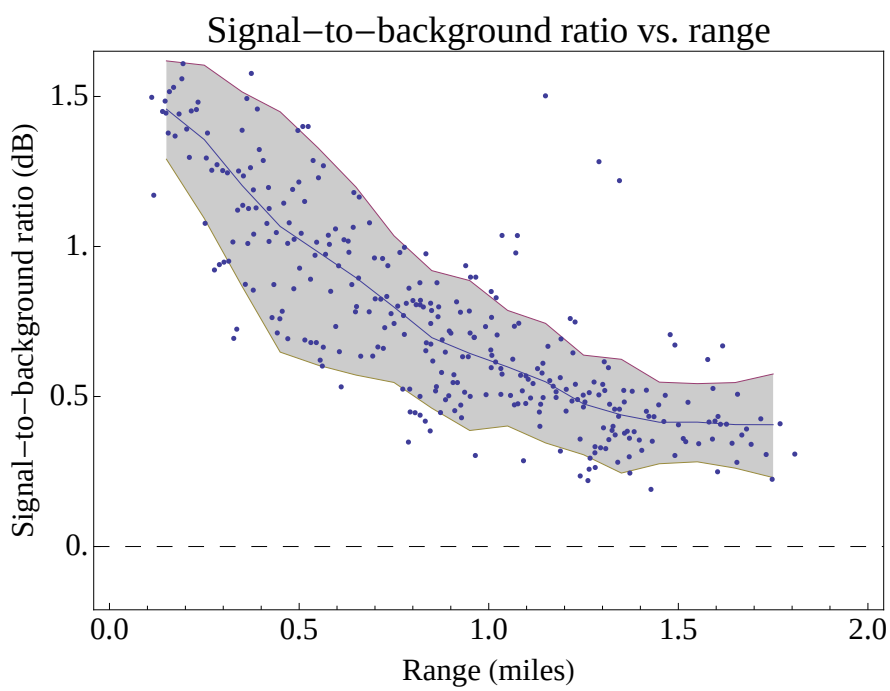


Figure 10: Estimates of signal-to-background ratio of the image of the target versus the range of the target. We also fit a curve to the data, and the gray area shows three times the median deviation from the curve in each direction.

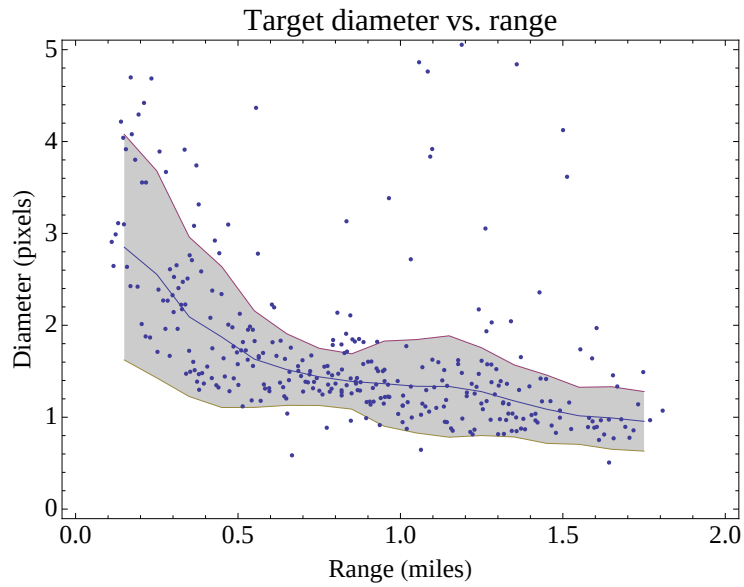


Figure 11: Target diameter estimated from the fitted Gaussians versus range. We use $(\sigma_x + \sigma_y)/2$ as a measure of target diameter. The median trend is given by the curve and the gray region shows thrice the median deviation in each direction.

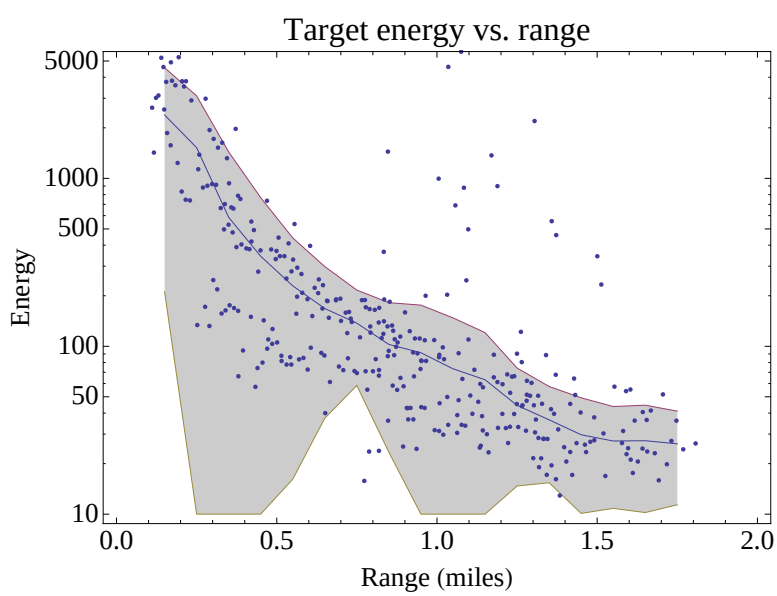


Figure 12: Target amplitude, also measured from the fitted Gaussians, versus range. We use $|s|/b$ to be a measure of target amplitude, i.e., just the relative height of the Gaussian with respect to the background. The median trend is given by the curve and the gray region shows thrice the median deviation in each direction.

4.3 Aircraft Image Formation Model

In an effort to explain the characteristics observed in the previous section, we developed a model of image formation for aircraft imaged above the horizon, and have compared the model's predicted characteristics to the observed image characteristics. We constructed a head-on silhouette of a Piper Archer III, which has the same geometry as the Piper Archer II, from a head-on image of a Piper Archer III obtained from Piper's website. We then used a model of image formation that takes into account atmospheric effects that was proposed by Nayar in [16]. In this model we have aimed to taken into account the effects of haze and atmospheric scattering and potential defocus blur due to poor or mismatched optics.

In the model we construct a synthetic image that predicts the appearance of the target as it would be seen from a camera looking at a target at equivalent range. We assume that we either have access to a three-dimensional model, or an image of the target taken from the desired viewpoint, and that we can obtain a silhouette of the target as seen from the desired direction. We start with a much larger image, which we will downsample and blur image by amounts that will simulate the appropriate range. In particular the predicted appearance of a target imaged at range r is given by the following formula:

$$\mathcal{I} \quad = \quad \downarrow_{f/r} G_{(1+\sigma)r/f} \star \left(c_\infty \, \mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}} + c_\infty \, (1 - e^{-\beta\, r}) \, \mathcal{I}_{\substack{\text{target}\\ \text{mask}}} + \frac{e^{-\beta\, r}}{r^2} \, \mathcal{I}_{\text{target}} \right) .$$

The terms originate from the following assumptions, with reference to **Figure 13**:

- The first term, $c_\infty \, \mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}}$, represents the contribution of the sky. In daylight the sky is so much brighter than the target, so most of the contrast in the image is due to the fact that the aircraft casts a shadow on the observer, as shown in **Figure 14**. So c_∞ is the intensity of the sky, which is just the accumulation of all scattering of light in the sky out to infinity. $\mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}}$ is an image of the silhouette of the target, where pixels on the target are 0, and all others are 1.

- The second term, $c_\infty \, (1 - e^{-\beta\, r}) \, \mathcal{I}_{\substack{\text{target}\\ \text{mask}}}$, represents the contribution of the accumulation of airlight in between the target and the sensor, as depicted in **Figure 15**. The intensity of the airlight is c_∞, since the scattering particles are scattering the ambient lighting. β is the scattering coefficient, which we are assuming to be wavelength independent in this case. The larger β is, the more atmospheric scattering there is. For example in hazy conditions β would be near 10^{-3}; on clear days closer to 10^{-6}. $\mathcal{I}_{\substack{\text{target}\\ \text{mask}}}$ is the exact opposite of $\mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}}$, that is, $\mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}} = 1 - \mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}}$.

- The third term, $\frac{e^{-\beta\, r}}{r^2} \, \mathcal{I}_{\text{target}}$, can model the effects of a glinting target. In this report, we assume this term is zero for the purposes of modeling and comparison. However, this term could be used to model the effects of direct sunlight reflected towards the observer by the aircraft. With a three-dimensional model and knowing the

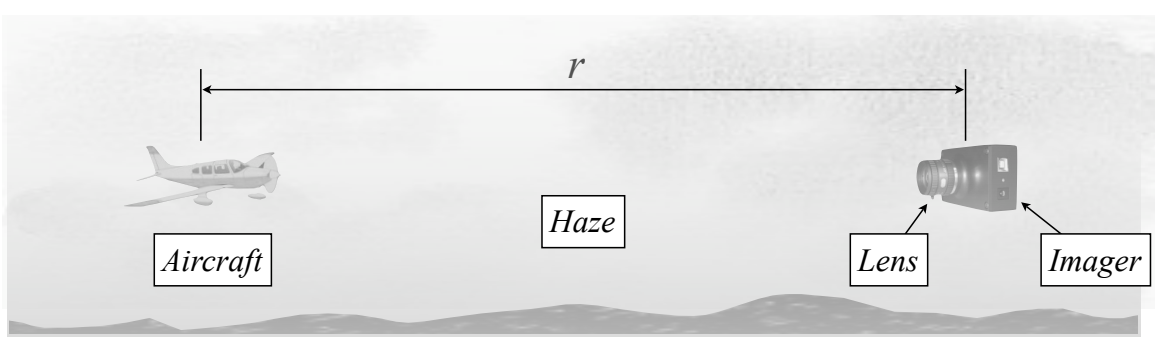


Figure 13: A depiction of a typical scene in which an aircraft is imaged by a sensor some distance r away. The sensor is comprised of a lens and imager, whose optical and sampling properties respectively affect image formation. In addition, haze in between observer and aircraft can affect the image.

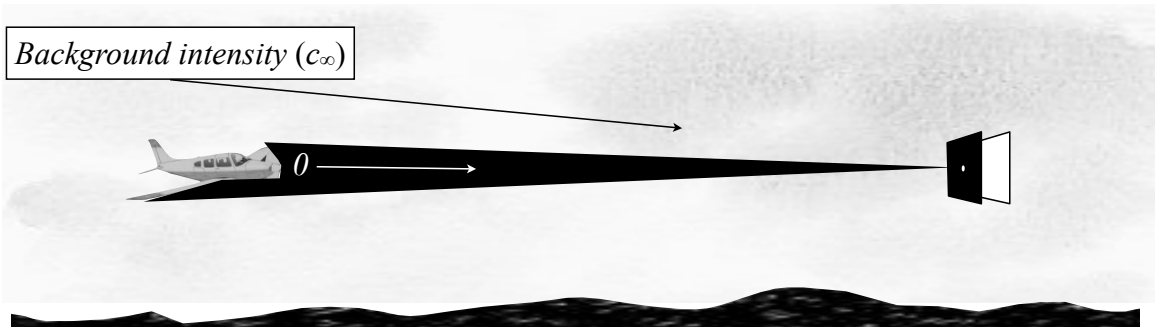


Figure 14: When haze is negligible, and when sunlight is not directly shining on the aircraft, such as would be the case in overcast conditions, then the background lighting is generally significantly brighter than the target. Then, the contrast one sees in the image is essentially just the cast shadow of the aircraft.

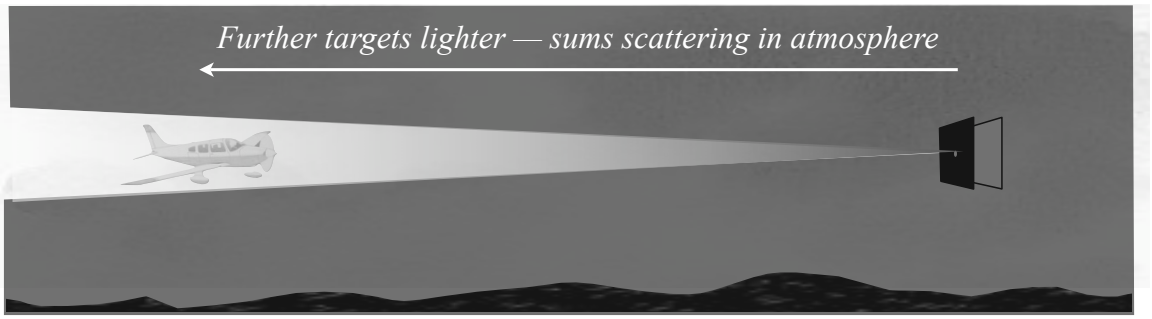


Figure 15: When there is haze, it makes the aircraft look lighter the further away it is, since there is more "airlight" in between the observer and aircraft that is scattering light.

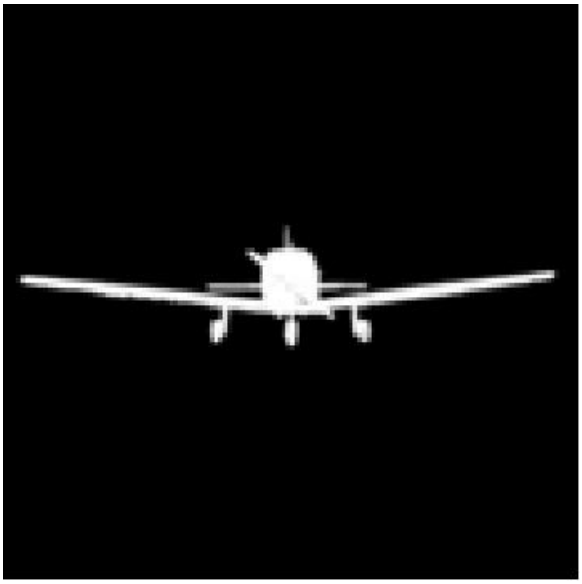
direction of sunlight, one can calculate the amount of light reflected by the target in the direction of the observer. The exponential term models attenuation.

- The image constructed by the sum is the image as it would be seen from a standard distance, which when using an image to generate the silhouette, depends on the distance from which the original picture was taken. The next step is to model both the change in size, as well as optical blurring due to different ranges. The operator $\downarrow_{f/r} G_{(1+\sigma)r/f} \star (\cdot)$ models these effects. $G_w \star (\cdot)$ represents convolution with a two-dimensional Gaussian kernel, of width w; here w is proportional to range and inversely proportional to the focal length of the lens (f). We allow for defocus blur by adding σ. We are therefore assuming that the modulation transfer function of the lens is Gaussian, and that any potential mismatches or defocus can be modeled by the addition of σ. The final operator $\downarrow_s$ represents downsampling with sampling interval s, where here s is proportional to r/f.

To compute the result of the image formation model, we took the image in **Figure 16**, and constructed the image masks $\mathcal{I}_{\substack{\text{bkgd}\\ \text{mask}}}$ and $\mathcal{I}_{\substack{\text{target}\\ \text{mask}}}$. **Figure 17** shows some of the predicted images for various realistic values of β, and at different ranges. As β increases, the contribution of atmospheric scattering increases, and the aircraft becomes less distinguishable. As we simulate seeing the target from further and further distances, the target's wings become less distinguishable, and it degenerates to an elliptical, smooth depression in intensity. Though the wings become less distinguishable, their presence is still affecting the strength of the target, in particular, the amount of depression in intensity.

In **Figure 18** we show the predicted signal-to-background ratio as a function of range, and for various values of β. The left and right of **Figure 19** show the target diameter and target amplitude vs. range. We found that the addition of

(a)



(b)

Figure 16: (a) A head-on view of a Piper Archer III (same geometry as a Piper Archer II). (b) A head-on silhouette of the Piper Archer III constructed from the image on the left.

σ affected the target diameter, but did not affect target amplitude. In other words, as we would expect, extra blurring smears the energy over a wider area, but preserves it. Increasing β, on the other hand, decreases the target energy, but does not affect the target diameter.

Finally, note that with a three-dimensional model of the aircraft, it is possible to evaluate the image formation model with arbitrary intruder aircraft attitude, and arbitrary viewpoint. This would allow us to better compare head-on and passing situations, and with a three-dimensional model it is also possible to predict the effects of direct sunlight and glinting.

In **Figure 20** we show the predicted signal-to-background ratio as a function of range, with different curves for different resolution. We measure resolution in milliradians per pixel ($\mathrm{mrad/pixel}$). The signal-to-background ratio is a key measure of identifiability in the image, therefore this plot can give some information about what resolution would be sufficient for target detection out to the desired range. For example, later we show that an aircraft detector has a detection rate at least better than 90% when the signal-to-background ratio of the targets is at least $1\,\mathrm{dB}$. Thus, if it is coupled with a sensor that has a resolution of $0.2\,\mathrm{mrad/pixel}$, then according to this plot the detector should be able to detect aircraft with at least 90% detection rate out to $3\,\mathrm{miles}$.

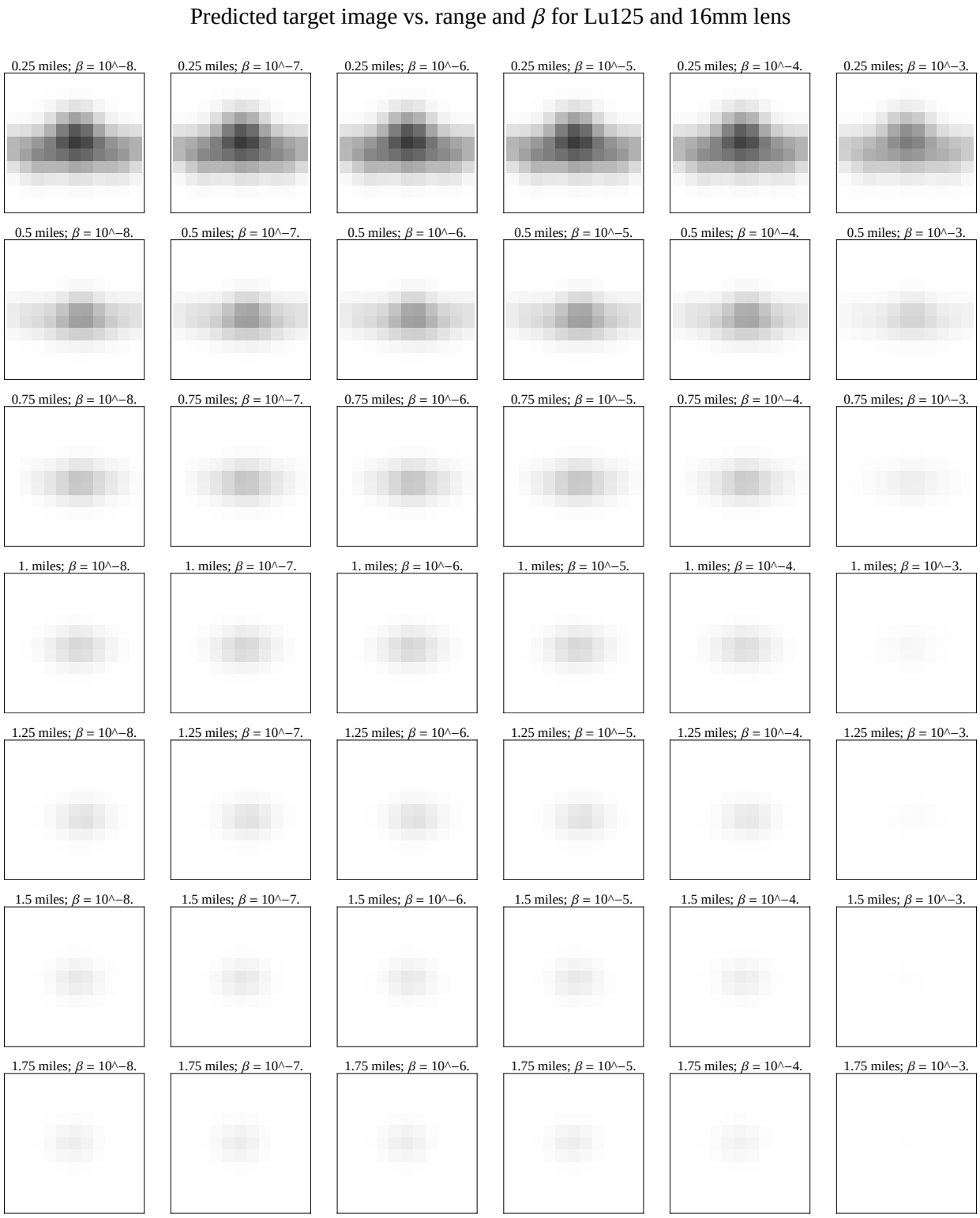


Figure 17: Predictions of how an 11×11 pixel image of a Piper Archer III would look in the Lu125 camera with a $16\,\mathrm{mm}$ lens for various ranges and values of β.

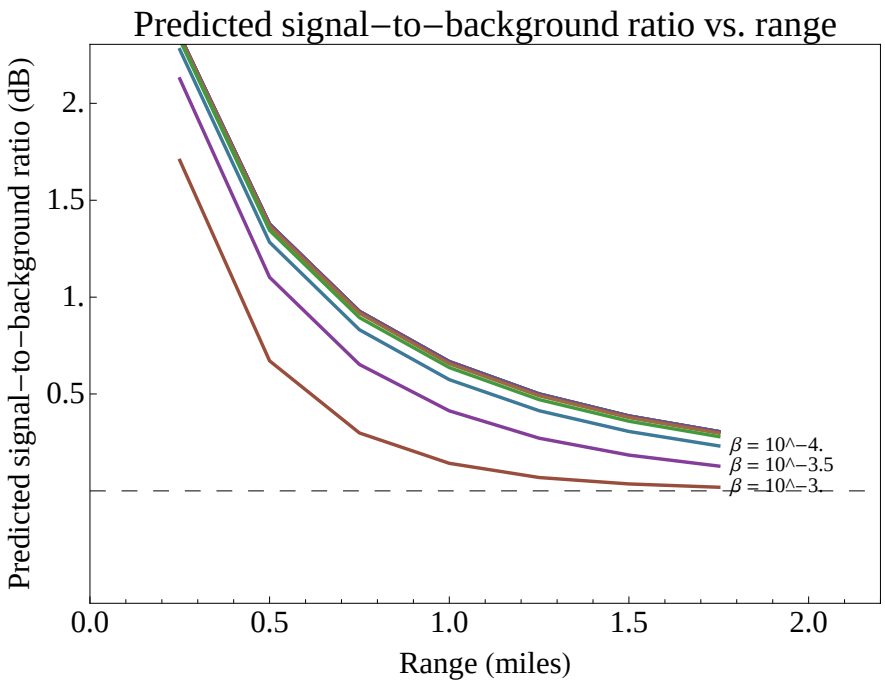


Figure 18: Plot of the prediction of the signal-to-background ratio as a function of range for various values of β, assuming Lu125 camera and $16\,\mathrm{mm}$ lens. β equal to 10^{-3} approximates hazy conditions; a smaller β, e.g., 10^{-6}, is clear sky.

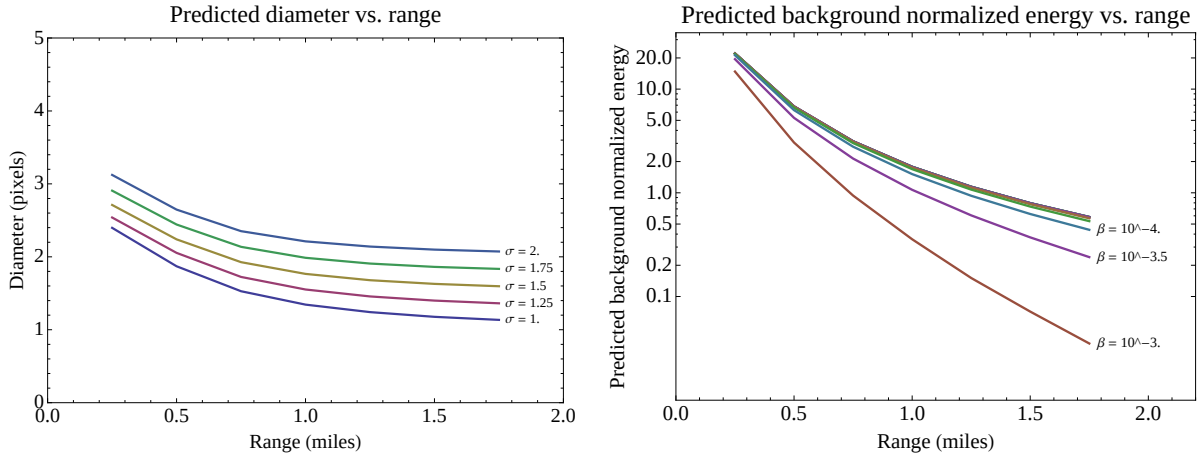


Figure 19: **Left:** Prediction of target diameter as a function of range for different σ; non-zero σ models defocus blur and increases effective size of target; assuming a Lu125 camera and $16\,\mathrm{mm}$ lens. **Right:** Prediction of background-normalized energy of the target, i.e., the integral of the Gaussian, shown for different values of β.

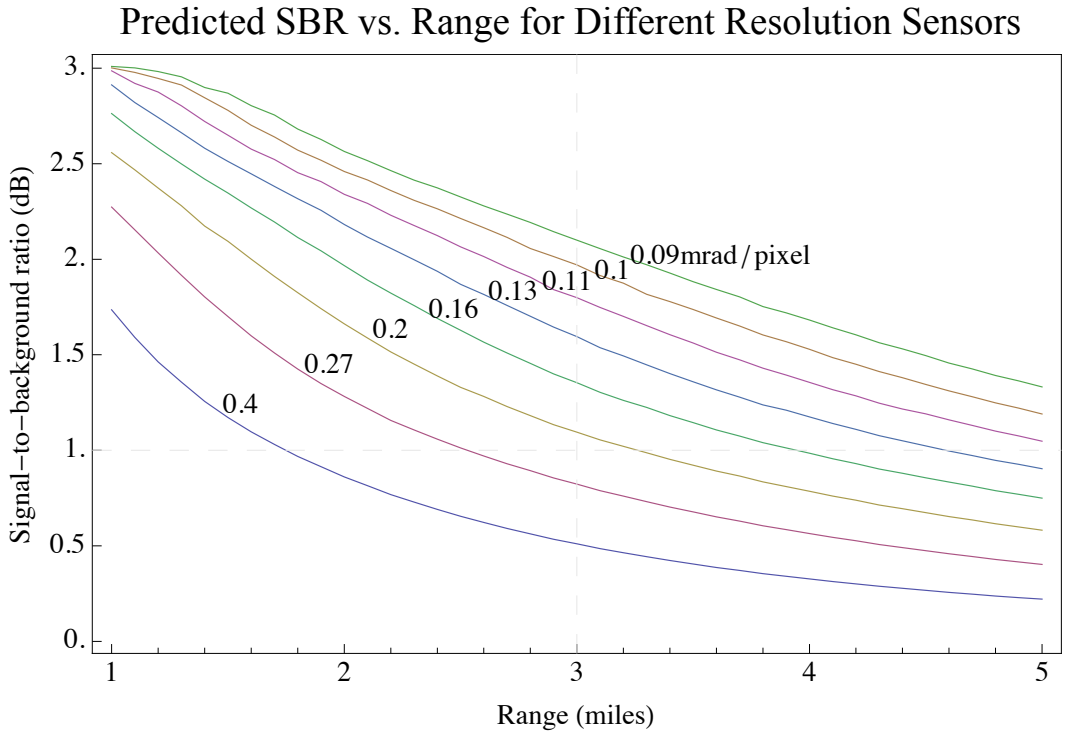


Figure 20: Here we show the predicted signal-to-background ratio as a function of range for sensors with different resolutions (measured in milliradians per pixel). The inflections at close range and high resolution are likely due to the fact that when the target is close enough, its wings are visible and its Gaussian shape has a horizontal-to-vertical size ratio of $6:1$, whereas at closer ranges the shape of the target is closer to a Gaussian with a horizontal-to-vertical size ratio of $3:1$. The larger width causes the peak of the Gaussian to flatten out, thereby decreasing the signal-to-background ratio.

4.4 Validating the Aircraft Image Formation Model

Our next step was to verify the validity of the aircraft image formation model. To accomplish this we acquired images from several different cameras, paired with different lenses, to test the model. In particular, we wanted to verify that the signal-to-background ratio, which encapsulates both size of the target and signal strength, was faithfully predicted by the model. In the past we had tested four different cameras (Lumerena Lu125 and Lw625, and two IR cameras, a Thermoteknix Miricle 307k, and Indigo Systems' Omega), and not having been satisfied with the latter three cameras' performance, we then kept the Lu125 and tested three other cameras. In all, these were:

- *Lumenera's Lu125.* The Lu125 has a $8.58\,\mathrm{mm} \times 6.86\,\mathrm{mm}$ $2/3$" FillFactory IBIS5A-1300 imager with a resolution of 1280×1024. It is a CMOS imager with a $6.7\,\mu\mathrm{m}$ pixel pitch. In its high dynamic range mode ($> 100\,\mathrm{dB}$) it is capable of $7.5\,\mathrm{fps}$. This camera is compatible with a C-mount lens. We used a monochrome model that has a USB 2.0 interface.

- *Imperx IPX-2M30.* The IPX-4M15 has an $11.84\,\mathrm{mm} \times 8.88\,\mathrm{mm}$ Kodak KAI-2020 imager with a resolution of 1600×1200. It is a CCD imager with a $7.4\,\mu\mathrm{m}$ pixel pitch. It is capable of $33\,\mathrm{fps}$. This camera is compatible with a C-mount lens. We used a monochrome model that had a CameraLink interface.

- *Imperx IPX-4M15.* The IPX-4M15 has a $15.15\,\mathrm{mm} \times 15.15\,\mathrm{mm}$ Kodak KAI-4021 imager with a resolution of 2048×2048. It is a CCD imager with a $7.4\,\mu\mathrm{m}$ pixel pitch. It is capable of $15\,\mathrm{fps}$. This camera is compatible with a F-mount lens. We used a monochrome model that had a GigE interface.

- *Imperx IPX-16M3.* The IPX-16M3 has a $36.07\,\mathrm{mm} \times 24.05\,\mathrm{mm}$ Kodak KAI-16000 imager with a resolution of 4872×3248. It is a CCD imager with a $7.4\,\mu\mathrm{m}$ pixel pitch. It is capable of $3\,\mathrm{fps}$. This camera is compatible with a F-mount lens. We used a monochrome model that had a GigE interface.

We paired these cameras with a number of different fixed-focus lenses, and one varifocal lens. We had access to two Zeiss F-mount lenses ($50\,\mathrm{mm}$ and $85\,\mathrm{mm}$), one Nikon F-mount lens ($105\,\mathrm{mm}$), one $25\,\mathrm{mm}$ Computar C-mount lens, one varifocal Computar C-mount lens set at $36\,\mathrm{mm}$, and one $25\,\mathrm{mm}$ Fujinon C-mount lens. For each of the pairs we estimated the focal length using a calibration target and the Camera Calibration Toolbox for Matlab[1], and then using the camera specifications we obtained a number for resolution measured in milliradians per pixel. The total list of combinations we tested is listed below in **Table 1**.

To perform the validation we collected data from all these combinations of cameras and lenses in July, 2008, at the Jimmy Stewart Airport in Indiana, PA. To perform the data acquisition, we integrated a number of components with a pan-tilt unit so that the cameras would always be pointing at the airplane. We used a PTU-D100 pan-tilt unit from Directed Perception, which was driven by a Geo-Pointing Module (GPM) also made by Directed Perception. By using a combination of a Piccolo autopilot on-board the airplane, and a Piccolo Command Center on the ground (both by Cloud Cap Technology), we were able to constantly monitor the latitude, longitude and altitude of the airplane. A laptop processed the telemetry from the Piccolo ground station, and sent it to the GPM, which in turn drove the PTU-D100. We mounted four cameras at a time on the PTU-D100's mounting plate; see **Figure 21**. We recorded the lat./long. and altitude during the flight so that we could provide ground truth range. We show the trajectory flown on July 22, 2008 in **Figure 22**.

This setup was extremely useful in being able to keep the target within the field-of-view of the cameras, however, we did encounter some problems using this setup. First, the ability to receive telemetry from the Piccolo autopilot seemed to depend on the attitude of the airplane. We were unable to use an external antenna on the airplane, and so the autopilot's antenna was placed as near as possible to the rear-right passenger's window. Often, when the right side of the airplane faced away from the airport, we lost communication with the autopilot. Thus, when flying towards the airport, we tried to keep the airport slightly to our right, so as to try to minimize the potential for losing communication.

Having collected data from all the cameras, we then "ground-truthed" the imagery, meaning that we went through every image, found the target and recorded its position. We did this process by hand, and often started by looking at the target when it was large enough and had enough contrast to be clearly visible, and then followed it frame to

[1]The Camera Calibration Toolbox by Jean-Yves Bouguet is available at `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

Camera	Lens	Res.	mpix	FOV	#cams	mrad (predicted)
IPX- 4M15	Nk105mm	2048×2048	4.2	11°×11°	67	**0.09**
IPX- 16M3	Nk105mm	4872×3248	15.8	25°×17°	20	**0.09**
IPX- 4M15	Zs85mm	2048×2048	4.2	13°×13°	48	**0.11**
IPX- 16M3	Zs85mm	4872×3248	15.8	31°×21°	13	**0.11**
IPX- 4M15	Zs50mm	2048×2048	4.2	22°×22°	17	**0.19**
IPX- 16M3	Zs50mm	4872×3248	15.8	50°×34°	5	**0.19**
Lu125	CmpVF@36mm	1280×1024	1.3	18°×14°	33	**0.24**
IPX- 2M30	CmpVF@36mm	1600×1200	1.9	24°×18°	19	**0.27**
Lu125	Cmp25mm	1280×1024	1.3	25°×20°	17	**0.34**
IPX- 2M30	Fujinon 25mm	1600×1200	1.9	33°×25°	10	**0.37**

Table 1: This table lists the camera/lens pairs that we tried during testing. In order, this table lists in each column: (i) the camera; (ii) the lens; (iii) the resolution of the camera; (iv) the number of megapixels ($\mathrm{megapixel}$); (v) the field-of-view of the lens; (vi) an approximate number of cameras that it would take to cover a $270^\circ \times 40^\circ$ field-of-regard; (vii) an estimate of the resolution of the camera/lens pair measured in $\mathrm{mrad/pixel}$.

frame, amplifying the contrast and enlarging the image when necessary, until it was no longer distinguishable from noise. Having ground-truthed all of the images, we then fit Gaussians to windows around the images of the aircraft and computed the signal-to-background ratio. With the GPS position we could then compute the signal-to-background ratio versus range. We show these figures for all of the pairs that we tested in **Figures 23 – 27**. In each case we show an estimate of the SBR-vs.-range using the median SBR in $0.1\,\mathrm{mile}$ windows, as well as a confidence window obtained from three times the median deviation from the median. Since we estimated the resolution for the pair, in each case we superimpose the SBR-vs.-range curve that is predicted for that resolution. We also find in each case the resolution that yields the SBR-vs.-range curve that is closest to the measured SBR-vs.-range curve, and then superimpose the best-fitting curve and the equivalent resolution in these figures, as well.

Figure 21: A picture of the setup that we used to capture images. We used a Directed Perception pan-tilt unit to point a set of four cameras at a time at the target. The target's position was communicated to the ground using a Piccolo autopilot and ground station; the aircraft's position was then communicated to a Directed Perception pointing module that drove the pan-tilt unit based on the latitude, longitude and altitude of the target.

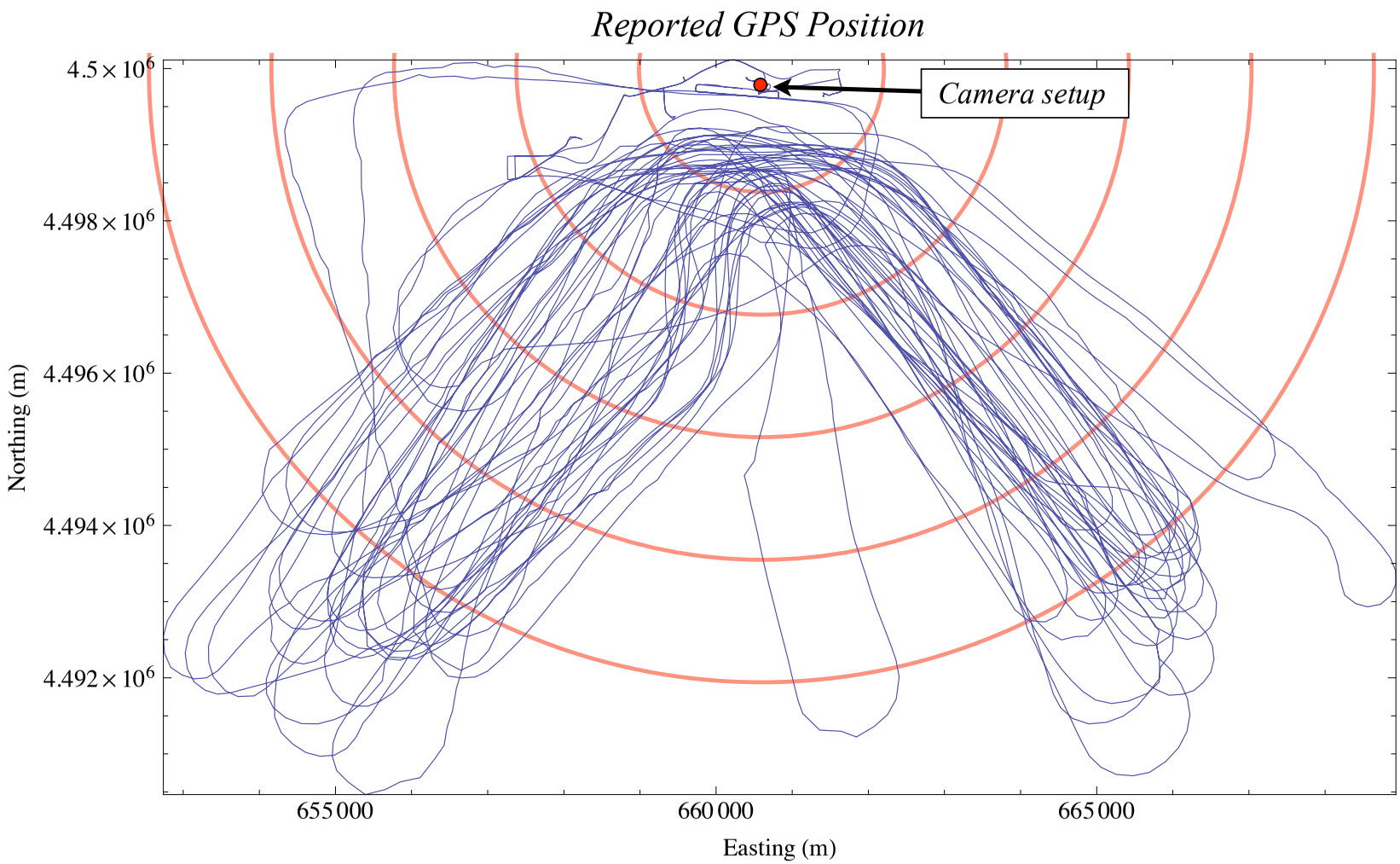


Figure 22: A plot of the trajectory of the aircraft on July 22, 2008. We flew a number of NW/SE and SW/NE passes in front of the hangar where the tripod and cameras were set up. The circles show distance to the hangar in miles. Fort the most part we turned around shortly after reaching 5 miles from the hangar. The axes give UTM northing and easting coordinates in meters.

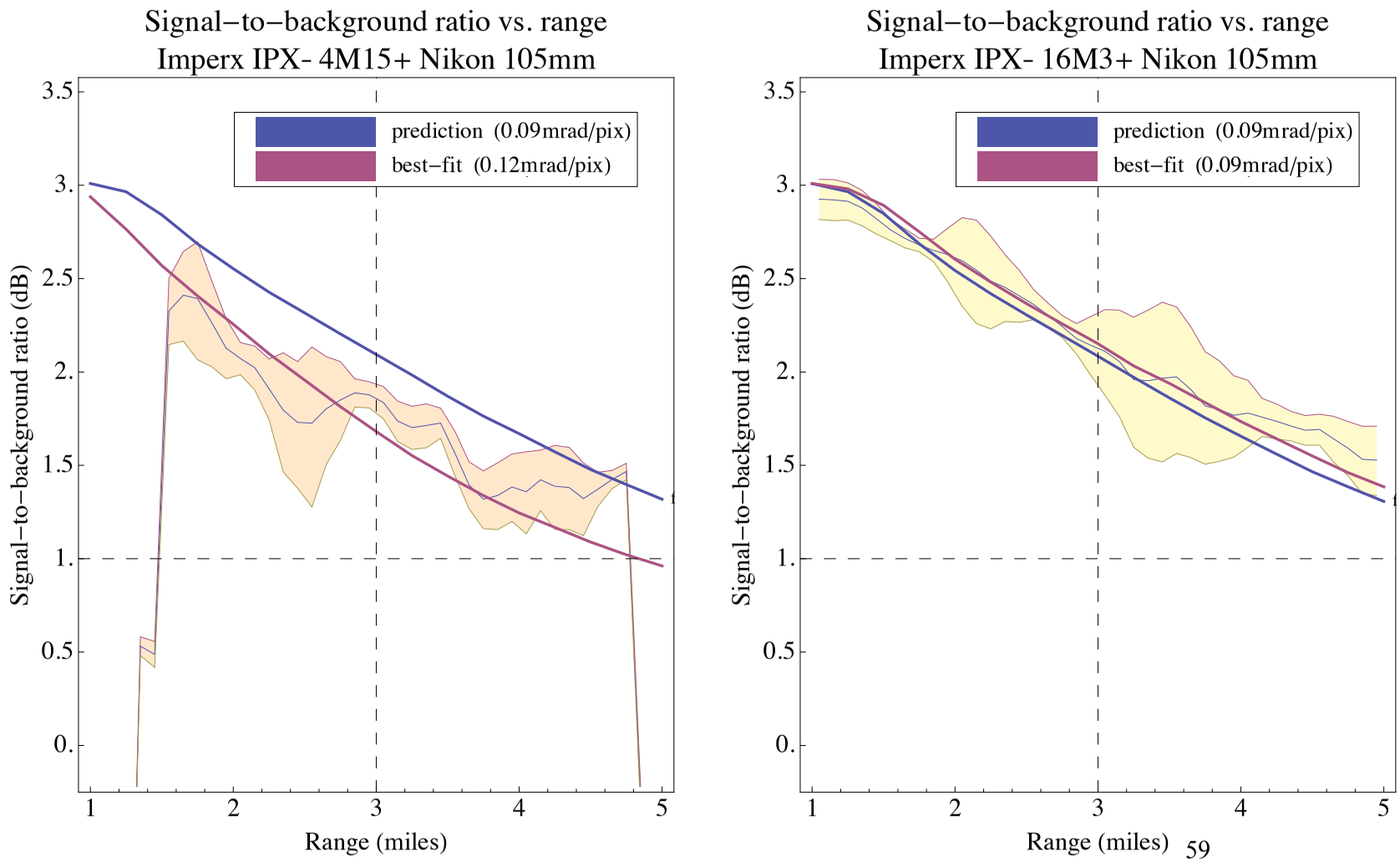


Figure 23: Signal-to-background ratio (SBR) vs. range for the Imperx IPX-4M15 and Nikon 105 mm lens (*left*) and the Imperx IPX-16M3 and Nikon 105 mm lens (*right*). In both cases we show the predicted SBR-vs.-range curve (computed from the camera and lens' specifications) as well as the best fitting curve (computed by finding the resolution that gives the best fitting equivalent curve).

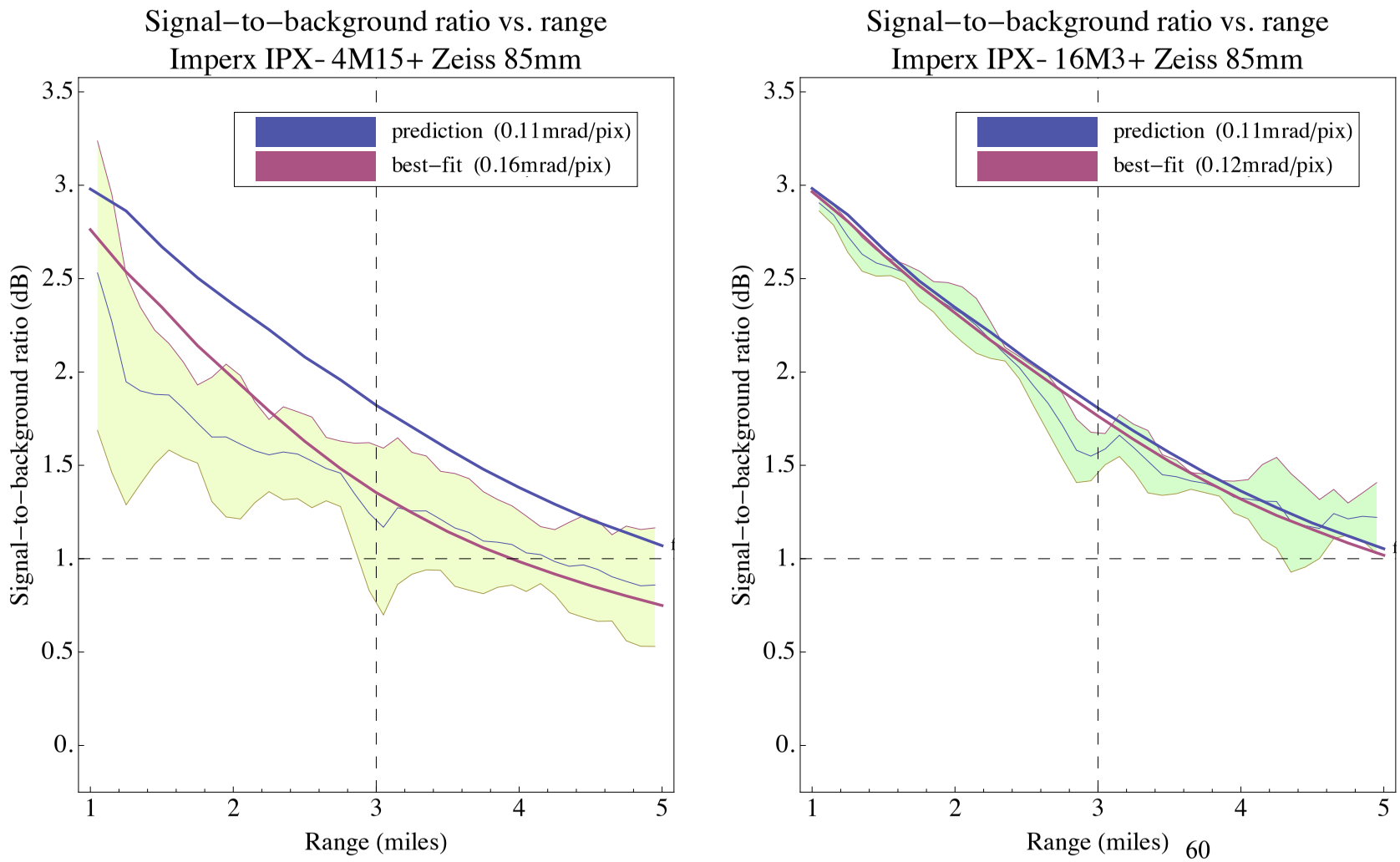


Figure 24: Signal-to-background ratio (SBR) vs. range for the Imperx IPX-4M15 and Zeiss 85 mm lens (*left*) and the Imperx IPX-16M3 and Zeiss 85 mm lens (*right*). In both cases we show the predicted SBR-vs.-range curve (computed from the camera and lens' specifications) as well as the best fitting curve (computed by finding the resolution that gives the best fitting equivalent curve).

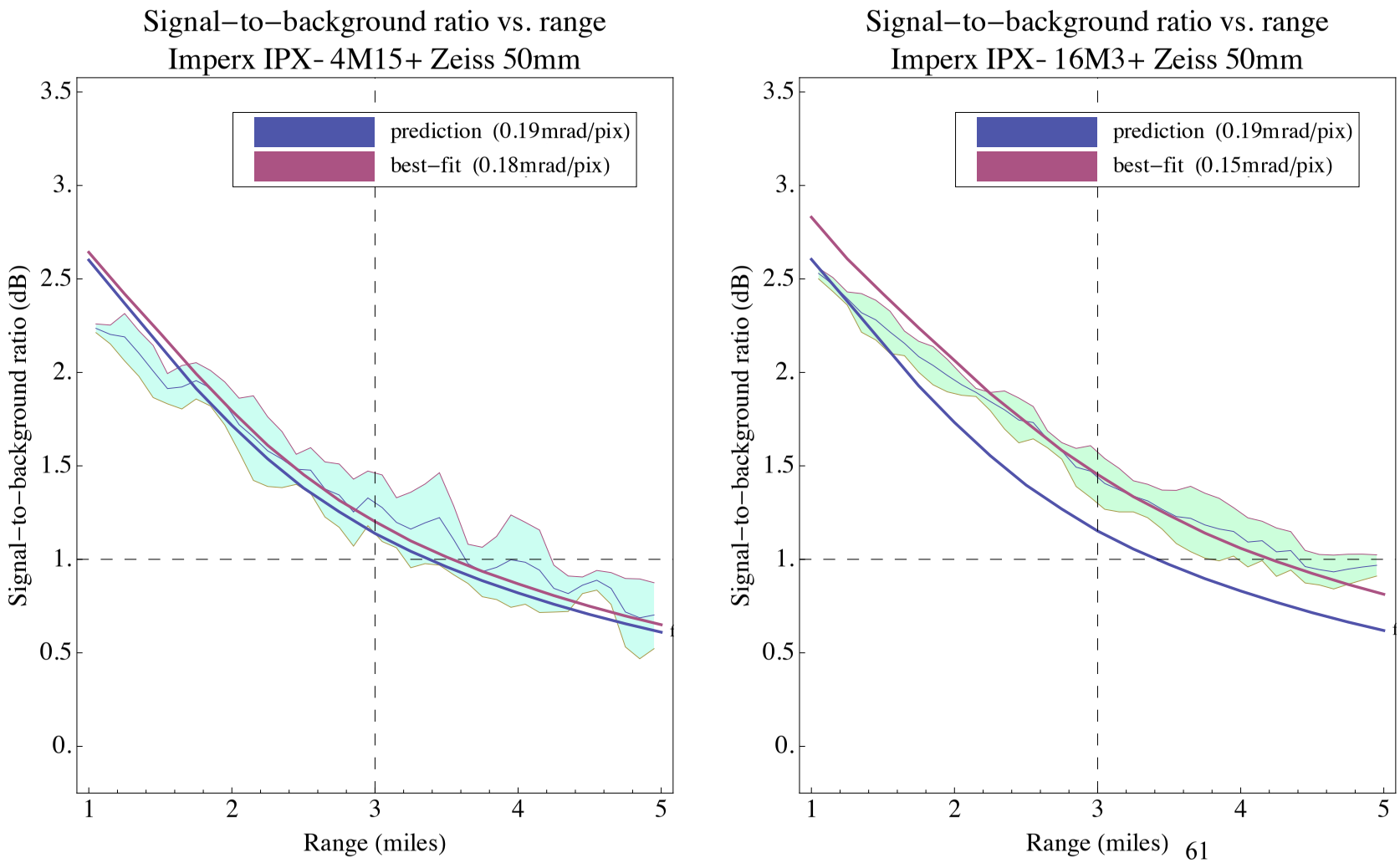


Figure 25: Signal-to-background ratio (SBR) vs. range for the Imperx IPX-4M15 and Zeiss 50 mm lens (*left*) and the Imperx IPX-16M3 and Zeiss 50 mm lens (*right*). In both cases we show the predicted SBR-vs.-range curve (computed from the camera and lens' specifications) as well as the best fitting curve (computed by finding the resolution that gives the best fitting equivalent curve).

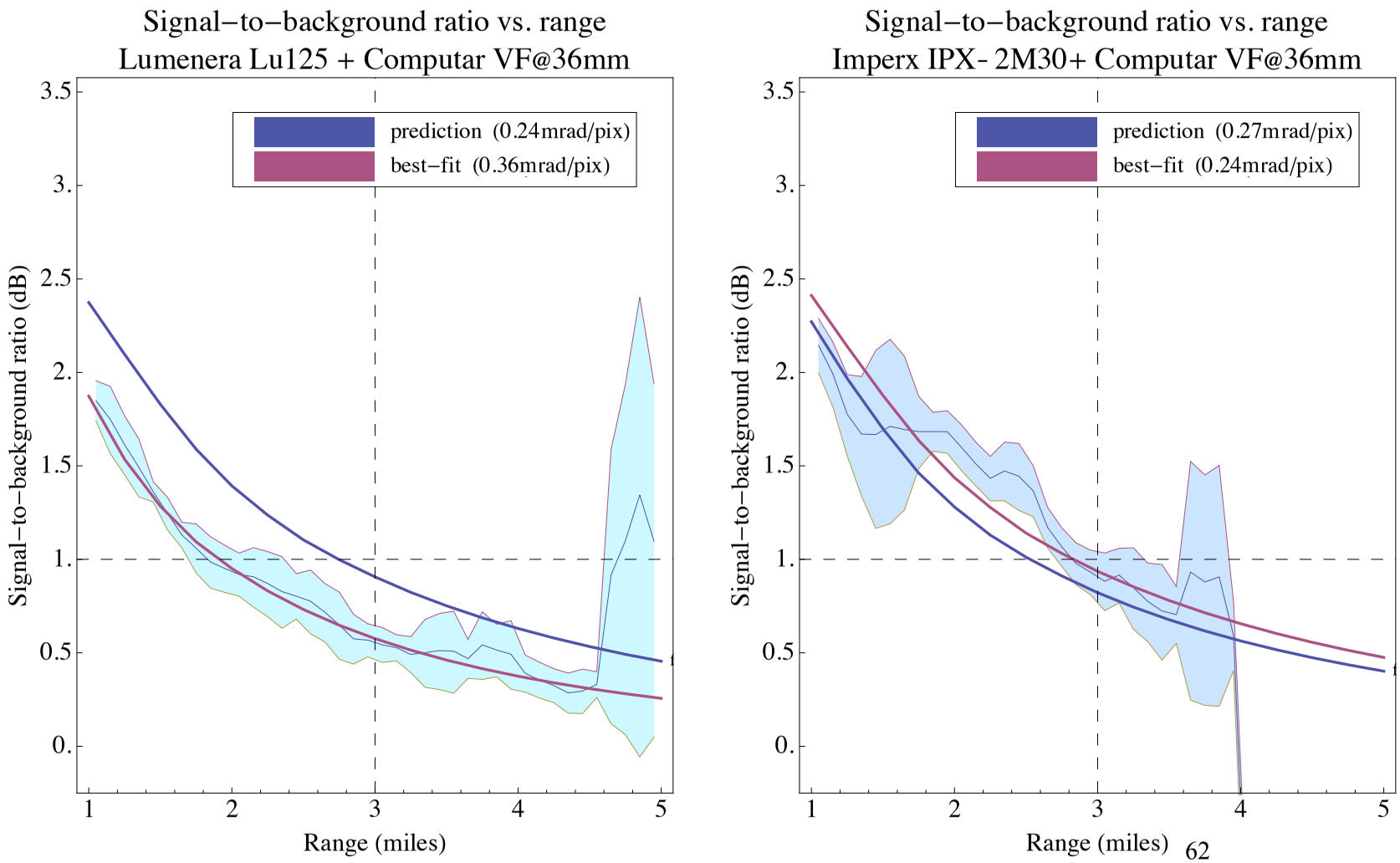


Figure 26: Signal-to-background ratio (SBR) vs. range for the Lumenera Lu125 and varifocal Computar lens set to 36 mm (*left*) and Imperx IPX-16M3 and varifocal Computar lens set to 36 mm (*right*). In both cases we show the predicted SBR-vs.-range curve (computed from the camera and lens' specifications) as well as the best fitting curve (computed by finding the resolution that gives the best fitting equivalent curve).

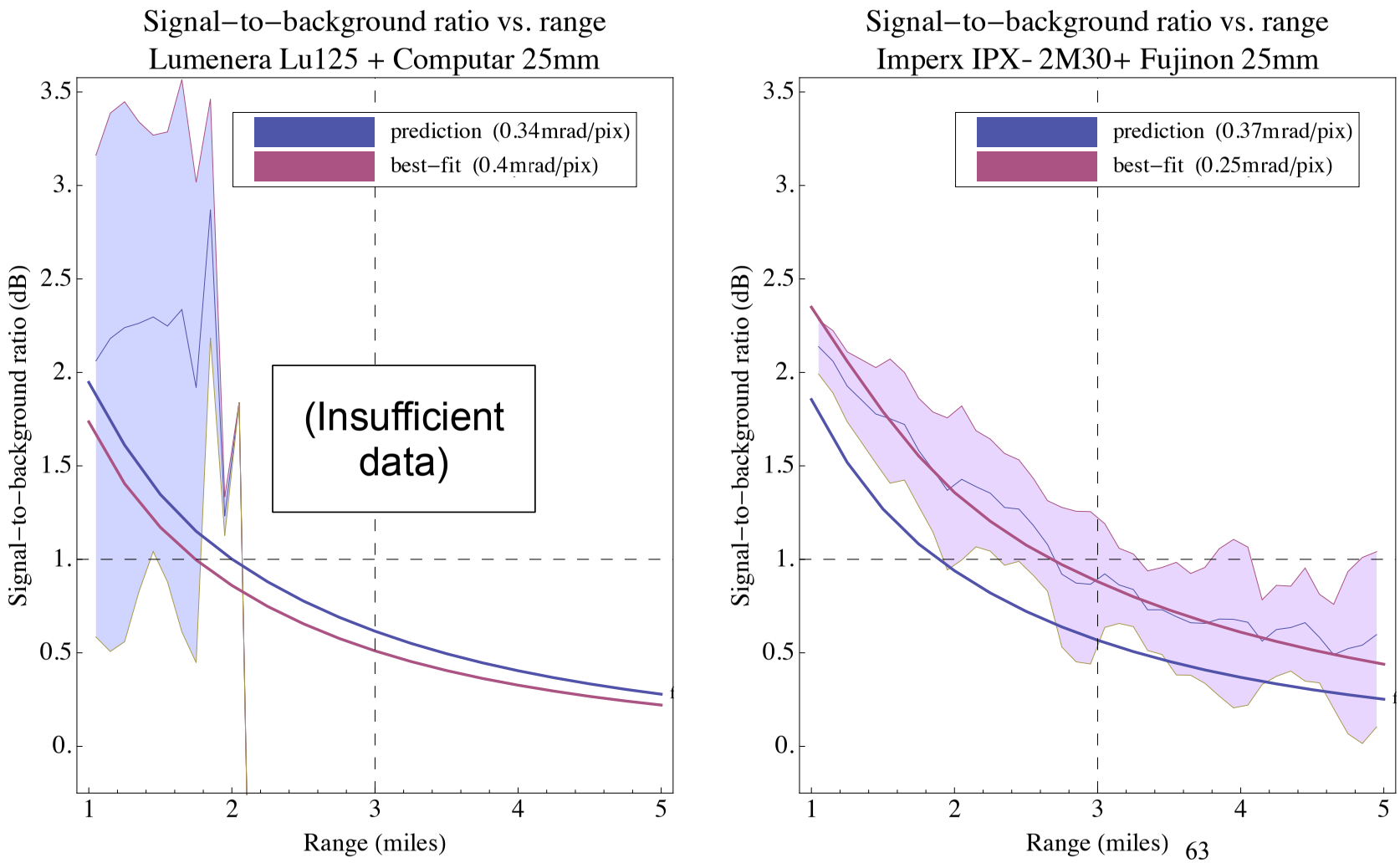


Figure 27: Signal-to-background ratio (SBR) vs. range for the Lumenera Lu125 and Computar 25 mm lens (*left*) and the Imperx IPX-16M3 and Fujinon 25 mm lens (*right*). In both cases we show the predicted SBR-vs.-range curve (computed from the camera and lens' specifications) as well as the best fitting curve (computed by finding the resolution that gives the best fitting equivalent curve).

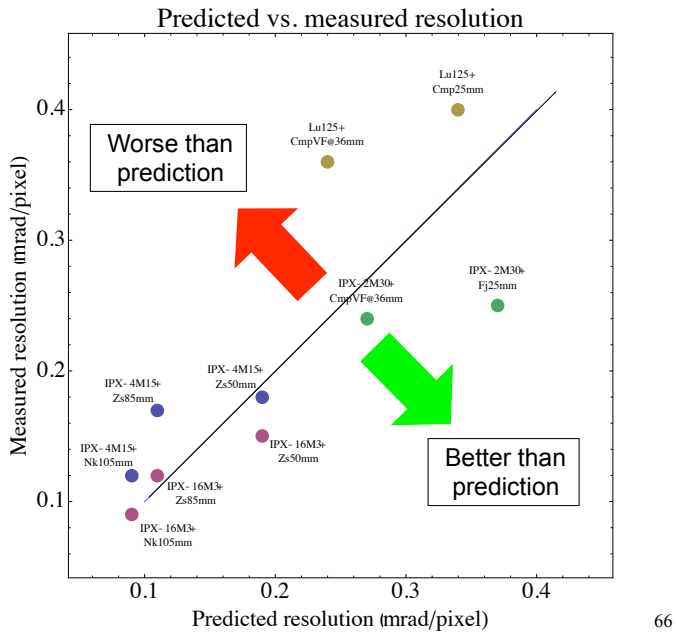


Figure 28: A plot of the predicted resolutions versus the measured resolutions for each camera and lens pair tested. If the point is above the line $y = x$, then the pair did worse than expected; if it is below, it did better than expected. See text for more analysis.

In general, looking at the predicted and measured SBR-vs.-range curves, we see that though the trends seem to have similar characteristics, there is not always agreement. Are these differences due to problems with the model or are they due to inherent performance differences in the cameras? To try to answer this, we plot the predicted vs. measured resolutions in **Figure 28**. In particular, for each camera we have the predicted resolution, as well as the resolution that yields the SBR-vs.-range curve closest to the one measured for the camera (the measured resolution). We plot the positions of these pairs in the figure, where the x-axis is the predicted resolution, and the y-axis is the measured resolution. If the point is above the line, then the camera did worse than predicted; if it is below the line, then it did better than predicted. We notice several trends that indicate the possibility of systematic biases, though their causes are not known:

- In general, the IPX-4M15 seems to perform worse than the IPX-16M3. Because these sensors have the same pixel pitch, for each lens tried, the points for the IPX-4M15 and IPX-16M3 lie on a vertical line. For each lens, the IPX-4M15 is consistently higher (therefore worse) than the IPX-16M3.
- The Zeiss $50\,\mathrm{mm}$ lens outperforms both the Zeiss $85\,\mathrm{mm}$ and the Nikon $105\,\mathrm{mm}$ lenses. For both the IPX-4M15 and the IPX-16M3, the measured resolution is better than predicted only for the Zeiss $50\,\mathrm{mm}$ lens; for the the Zeiss $85\,\mathrm{mm}$ and Nikon $105\,\mathrm{mm}$ the measured resolutions are worse than predicted for both cameras.
- The Lu125 performs worse than the IPX-2M30. The Lu125 consistently performed worse than predicted, and the IPX-2M30 consistently performed better than expected.

That these trends seem to be consistent across cameras and lenses seem to indicate that the differences between measured and predicted SBR-vs.-range curves is due to inherent performance differences in the cameras and lenses, instead of problems with the model.

The significance of this analysis is that we can take a camera and lens combination and measure its performance in a way that we can predict performance; in particular, in the next section we will measure the effect of SBR on detection rates. Though we have not been able to completely decoupled lens and camera performance here, we believe that with additional measurements that this methodology could be used to measure the performance hits that are due to the camera and decouple effects on performance due to the lens.

4.5 Determining the Required Resolution

Our next goal was to determine bounds on the resolution that would be needed to detect aircraft at desired ranges. Visual flight rules insist upon a visibility of $3\,\mathrm{miles}$, and so it is expected that future requirements on sense and avoid for UASs will have to be able to detect aircraft out to at least $3\,\mathrm{miles}$. We therefore consider this requirement in the following analysis; however, other minimum ranges could be considered and the same analysis with slightly different numbers would follow.

To start with we considered earlier detection results obtained with the Lumenera Lu125 camera. We used the morphological filter (described in more detail in the next section) to detect targets in imagery acquired on April 22. We set the threshold on the output of the morphological filter so as to obtain a low false positive rate of 1.7 false positives per frame. The output gave us a list of all the true positives and false negatives (misses). The targets that were detected and those that were missed are shown in **Figure 29**. This plot shows the signal-to-background ratio and range for each positive example in the sequence. Empty circles represent positive examples of the target that the detector missed (the false negatives); filled circles represent examples that the detector detected (the true positives).

In **Figure 30** we show a bar plot of the proportion of targets at given signal-to-background ratios that were detected by the detector with the given false positive rate. The red (upper) area in each bar indicates the proportion of true positives, and the blue (lower) area indicates the proportion of false negatives. The normalized detection rate, or the percentage of positive examples detected,[2] is the ratio of the red (upper-right) area to the total area (sum of upper-right and lower-left areas).

We then perform a hypothetical experiment in which we consider what would happen if we could guarantee that all positive examples had a signal-to-background ratio above a certain level. That would mean cutting off the computation of detection rate to an area to the right of some point on the x-axis. For example, if we could guarantee if all targets were less than $1\,\mathrm{mile}$, then according to **Figure 29**, targets' signal-to-background ratios would with high likelihood be above $0.5\,\mathrm{dB}$. Then we eliminate from the normalized detection rate computation targets with signal-to-background ratio less than $0.5\,\mathrm{dB}$, yielding in this case about a 80% detection rate. The enveloped curve in **Figure 31** shows the estimated detection rate as a function of minimum signal-to-background ratio. If we desire to have a detection rate of at least 95%, then according to this plot, we ought to have to be able to guarantee that targets of interest (within minimum range) have at least a signal-to-background ratio of $1\,\mathrm{dB}$. We now return to **Figure 20**, reproduced in **Figure 32**, which gave us the predicted signal-to-background ratio versus range for different resolution sensors. If we insist that targets within the minimum range produce a signal-to-background ratio of at least $1\,\mathrm{dB}$, then this indicates that the minimum resolution is $0.2\,\mathrm{mrad}/\,\mathrm{pixel}$. **Table 2** shows the list of cameras with predicted and estimated ranges based on a $1\,\mathrm{dB}$ minimum, and using the estimated resolution.

Camera	Lens	Res.	mpix	FOV	#cams	mrad	Predicted range	Estimated range
IPX- 4M15	Nk105mm	2048×2048	4.2	11°×11°	67	**0.09**	**>5.0**	**5.0**
IPX- 16M3	Nk105mm	4872×3248	15.8	25°×17°	20	**0.09**	**>5.0**	**>5.0**
IPX- 4M15	Zs85mm	2048×2048	4.2	13°×13°	48	**0.11**	**>5.0**	**3.9**
IPX- 16M3	Zs85mm	4872×3248	15.8	31°×21°	13	**0.11**	**>5.0**	**>5.0**
IPX- 4M15	Zs50mm	2048×2048	4.2	22°×22°	17	**0.19**	**3.4**	**3.6**
IPX- 16M3	Zs50mm	4872×3248	15.8	50°×34°	5	**0.19**	**3.5**	**4.1**
Lu125	CmpVF@36mm	1280×1024	1.3	18°×14°	33	**0.24**	**2.8**	**2.0**
IPX- 2M30	CmpVF@36mm	1600×1200	1.9	24°×18°	19	**0.27**	**2.6**	**2.9**
Lu125	Cmp25mm	1280×1024	1.3	25°×20°	17	**0.34**	**2.1**	**1.8**
IPX- 2M30	Fujinon 25mm	1600×1200	1.9	33°×25°	10	**0.37**	**1.9**	**2.7**

Table 2: List of camera/lens pairs and their estimated detection ranges, assuming a $1\,\mathrm{dB}$ minimum signal-to-background ratio.

[2] Adjusted by a weight so as to have a uniform distribution over signal-to-background ratio.

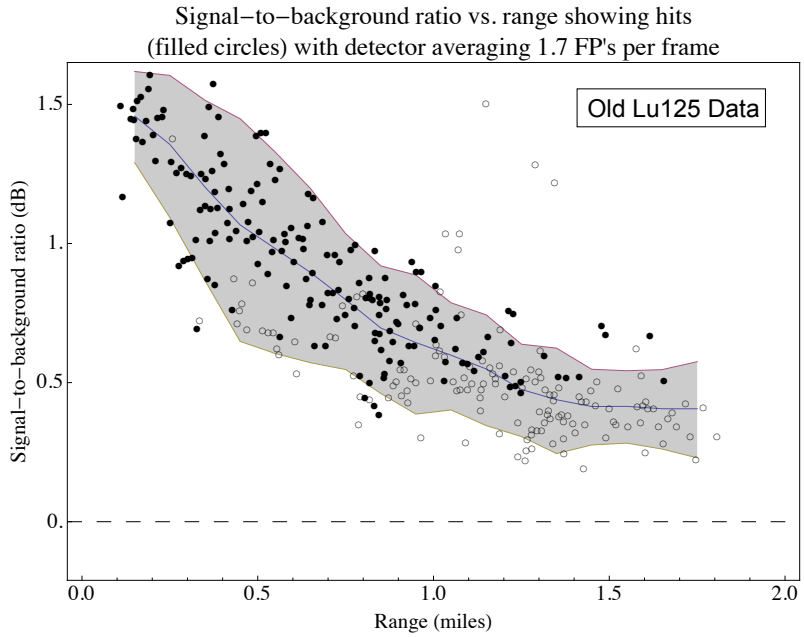


Figure 29: This plot shows the true positives (filled circles) and false negatives (empty circles) made by the Lu125 when morphological filter algorithm was applied to April 22, 2008 data, and shows the points at the corresponding signal-to-background ratio and range at the ground truth position.

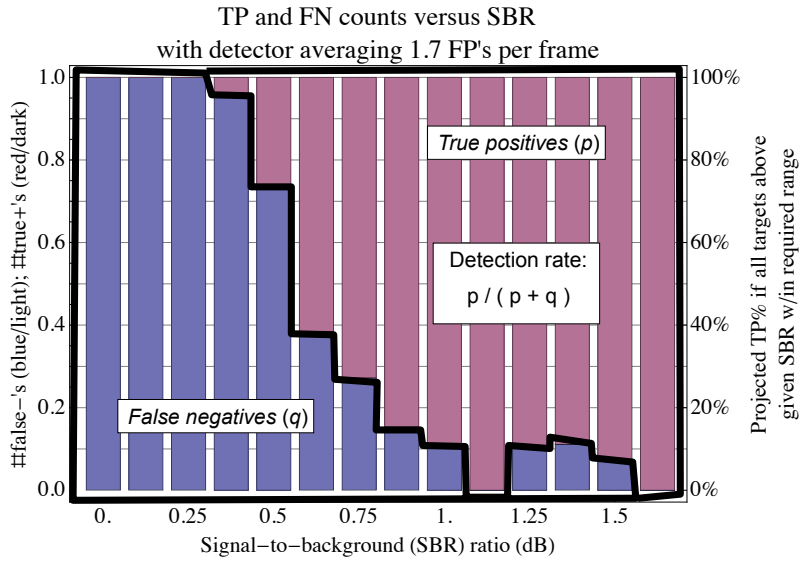


Figure 30: This plot shows the proportion of positive examples of targets that were detected and that had a given signal-to-background ratio. The normalized detection rate (normalizing the number of targets within each signal-to-background ratio bin) is the ratio of the area of the red (upper-right) to total area. Plot computed from April 22 Lu125 data.

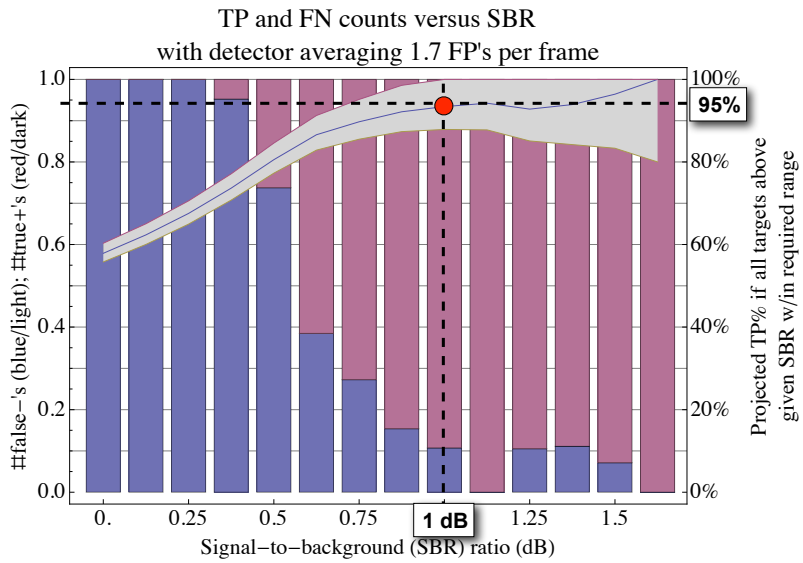


Figure 31: The enveloped curve gives the detection rate as a function of minimum signal-to-background ratio; that is, if we could guarantee that all targets of interest were above a certain signal-to-background ratio, this would be the predicted detection rate. For example, if we stipulated that the detection rate had to be at least 95%, then targets within the minimum desired range would have to have signal-to-background ratios of at least $1\,\mathrm{dB}$.

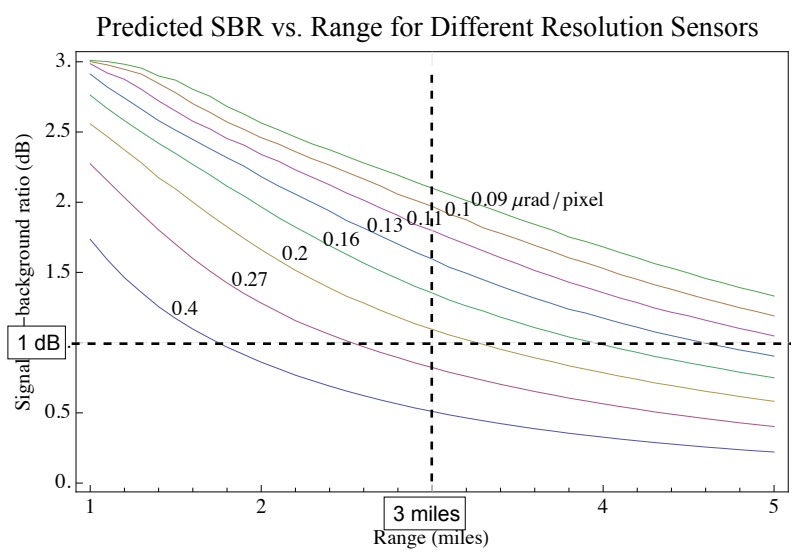


Figure 32: This is a reproduction of **Figure 20**. If we required that targets have a minimum signal-to-background ratio of $1\,\mathrm{dB}$ out to the desired range of $3\,\mathrm{miles}$, then we would have to use sensors having a resolution of at least $0.2\,\mathrm{mrad}/\,\mathrm{pixel}$.

4.6 Detection and Tracking Algorithms

In the first half of the program we examined a number of algorithms for detecting small targets. The focus was on methods that would yield both high detection rates while not having significant computation costs. Matched filters, for example, were not considered because of the computational cost of convolution and the variability in target appearance when viewed from different viewpoints. We tested an algorithm proposed by Gandhi et al. [11, 12], an approach used by Scott Foster in his Paravise system, and a machine learning approach called AdaBoost. We also tried some approaches specifically on the infrared imagery, in particular an approach called track before detect. Here we will focus on the results of improving the most successful approach, morphological filtering.

In the morphological filtering approach, a non-linear filter is applied to the image, which has the property that it peaks when there is a large difference between the minimum and maximum intensities in a neighborhood of the pixel. Our initial results with this filter were not adequate in that either the detection rate was too low, or the number of false positives was too high. In an effort to reduce the number of false positives, we considered methods in machine learning that use training data to learn a classifier to find likely targets. We extracted properties from the window around a detection, such as the parameters of a fitted Gaussian, to evaluate the probability that the given detection is a true target.

In more detail, the processing in the final implementation proceeds as follows on each incoming frame:

1. **Segment out the horizon.** The first step is separating the area below the horizon from the area above. For now the performance is best above the horizon, and we run a horizon detector to find the horizon, and for our ground experiments we blackened out the area below the horizon. To detect the horizon, we construct a cost function that is a function of vertical position; the cost is the sum of the variance above the position and the variance below the position. The sum of variances is generally lower when the chosen position is at the horizon. Our estimate of the vertical position of the horizon is therefore the minimum of this function.

2. **Apply a morphological filter to find possible detections.** We first apply a negative and a positive morphological filter to the image as suggested in [11]. The positive morphological filter, for example, has the following form:

$$\mathcal{M}^{+}(x,y) \quad = \quad \mathcal{I}(x,y) - \max\left\{\max_{|i|\leq w}\min_{|j|\leq w}\mathcal{I}(x+i+j,y), \max_{|i|\leq w}\min_{|j|\leq w}\mathcal{I}(x,y+i+j)\right\}.$$

   What the $\max$ operator is doing is getting a worst-case or conservative estimate of the background intensity; that is, it takes the maximum of all minimas within windows of size $2w+1$, though separately in vertical and horizontal directions. The resulting difference, then, is a worst-case estimate of the difference between the intensity of the target and its background. The positive morphological filter peaks where the target is lighter than the background; the corresponding negative morphological filter peaks where targets are darker than the background. We apply a negative and a positive morphological filter to the image, resulting in two in $\mathcal{M}_{+}$ and $\mathcal{M}_{-}$.

3. **Threshold and choose the top n detections.** We have a threshold T_{+} for T_{-} for positive and negative targets, and we select from $\mathcal{M}_{+}$ and $\mathcal{M}_{-}$ those pixels that exceed the respective thresholds. Since a single target is likely to generate multiple pixels that exceed the threshold, we perform non-maxima and -minima suppression, which picks only the largest or smallest value pixel within a window. We then take the top n pixels of positive type and top n pixels of negative type.

4. **Compute a feature characterizing the shape of each target.** For each of the targets we fit a Gaussian to a local window around the detection. We use a Gaussian parameterized as in equation (1), where b is the background intensity, s is the amplitude, and σ_x and σ_y are the dimensions in the x- and y-directions. In order to avoid re-computing the intensity image for Gaussians we store a library of images of Gaussians for a fixed range of σ_x and σ_y. We perform the fitting over b, s, σ_x and σ_y using gradient descent on the sum of square differences of intensities. To speed computation of the gradient, we pre-compute derivatives of the Gaussian images with respect to σ_x and σ_y. Having computed the best fitting Gaussian, we also compute the difference between the Gaussian image and the input image. We take the sum of positive and negative parts of the residual in a middle sub-window, and upper left, upper right, lower left, and lower right sub-windows.

5. **Use an SVM classifier to estimate the probability that a given detection is a target.** Then, we construct a feature vector consisting of the parameters of the best-fitting Gaussian parameters, and the sums of positive and negative residuals in the sub-windows. We pass this feature vector through a support vector machine (SVM) [7], which is a classifier that we train beforehand on positive and negative examples. During training we construct a interpolated function that, given the output of the support vector machine, yields the log-likelihood ratio of the hypothesis that the input feature vector represents a feature in the target class versus an a feature in the outlier class. We convert this log-likelihood to a probability. The closer this value is 1, the more likely is it that the detection is a true target. We keep only those detections whose probability exceeds a minimum value of p_{min}.

6. **Estimate the motion between frames so as to stabilize target tracks.** Next, we stabilize the motion of targets by estimating the change in the direction of the camera. In practice we will have state information from aircraft inertial systems to help in this regards, however, during our testing we had no such information, so we estimated the translation between frames off-line. The goal is to find a shift in the image that minimizes the sum of square difference between intensities. We use an image pyramid where at the top of the pyramid is an n-times downsampled image (scaled down by 2^n). The purpose of the pyramid is to reduce the magnitude of the shift, making it easier to estimate. We progressively go down the pyramid, doubling in size each time, until we get to the original size of the image. Every time we re-double the size of the image, our estimate of the translation doubles. This allows us to estimate large motions between frames. In practice, this computation could be supplanted by motion estimates from an on-board inertial system.

7. **Match new detections to a list of targets being tracked.** Despite using the SVM classifier to reject outliers, there are still false positives; however, they are often intermittent. In the last step we perform tracking between frames. We implemented a robust tracking scheme that again used an SVM. When running, we assume we have a list of targets, and a list of new detections. The job is to find a detection for each target, and for any remaining detections make new targets from them.

   For every existing target, we consider all possible detections that it could match to within a large window. We then construct from the feature (Gaussian parameters, etc.) associated with the detection and the feature associated with the last detection of the target, a third feature derived from the two. This derived feature joins into a single vector the following: (i) the difference of the target and detection positions; (ii) the detection's feature; and (iii) the difference between the detection's feature and the target's feature. Using an SVM previously trained on negative and positive associations, we compute the probability that the detection matches the target.

   Having computed the probabilities of matching detections to targets, we try to find the matching which maximizes the sum of probabilities. We construct a matrix whose rows correspond to targets, and columns to detections. For any possible matches that are not considered possible, we set these entries to 0. The goal is to choose entries from the matrix, no more than one from any one column, and no more than one from any one row, such that the sum of probabilities is a maximum. We use the Hungarian algorithm [17] to find this matching. For the number of targets we typically have, usually less than 200, this computation can be computed fairly quickly.

In terms of run-time, though the current implementation is not fast enough for real-time, almost 90% of the computation consists of image processing operations and could be implemented without significant difficulty on a GPU, DSP, or FPGA; the remaining computation requires more general computation, but could easily be parallelized. Currently the algorithm runs at $0.8\,\text{sec}$ per $4\,\text{megapixel}$ frame. The computational breakdown, roughly in terms of the stages above, is shown in **Figure 33**.

4.7 Performance

Using the data taken in July with an Imperx IPX-4M15 camera and Zeiss $85\,\text{mm}$ lens, and having ground truthed the images by hand, we set out to evaluate the performance of the developed algorithm. We tested the various steps of the algorithm, and in particular we broke the algorithm into three steps as follows:

- ***Morph*** stands for the algorithm resulting in just applying the morphological filter, and constructing a list of possible targets. In other words, this algorithm corresponds to stopping at step 3. No SVM classifier is applied, and no tracking is done.

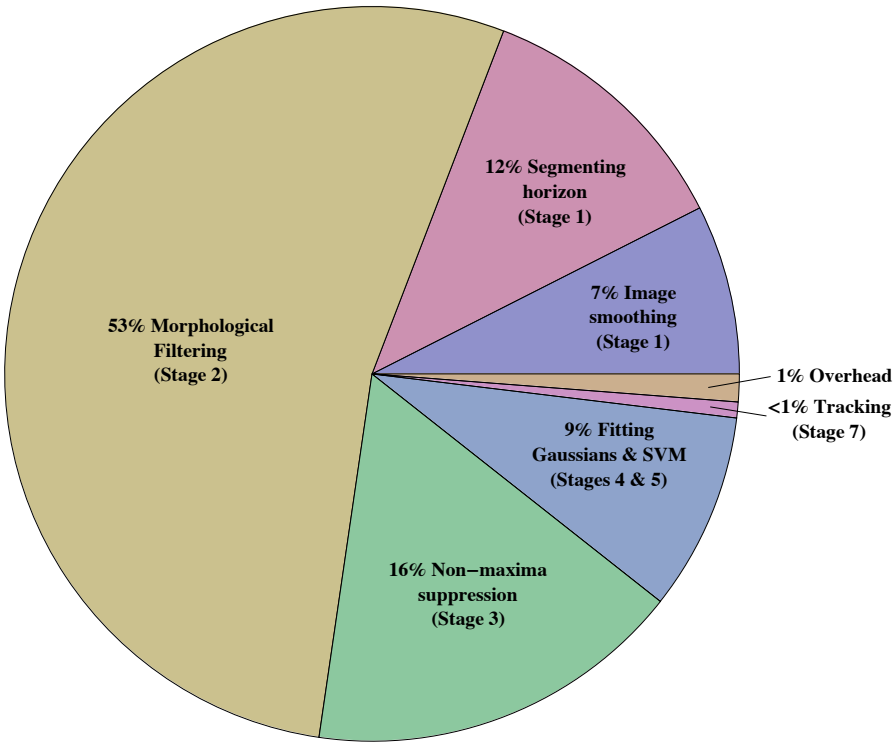


Figure 33: This pie chart shows a break down of timing for each step of the algorithm. All but the 11% for overhead, fitting Gaussians, SVM and tracking are image processing operations and could easily be implemented in hardware.

- ***MorphSVM*** refers to stopping at step 5, i.e. after possible detections have been classified by the SVM. No tracking is done.
- ***MorphiseSVM*** refers to the complete algorithm, all the way to tracking in step 7.

We evaluated the performance of each stage using *receiver operating characteristic* (ROC) curves. These curves measure the specificity (ability to reject outliers) and sensitivity (ability to detect the true target) of a detector. **Figure 34** shows an example ROC curve for a hypothetical detector. The x-axis coordinate gives the false positive rate measured in number of false positives per frame, and the y-axis gives the detection rate, or the percentage of images in which the true target's position was found. A hypothetical perfect detector would have 0 false positives per frame and have a 100% detection rate, thus lying in the upper-left corner of the plot. In general, a detector will have performance dependent on some threshold parameter. When this threshold is lowered, the percentage of true detections may go up, but so will the number of false positives.

We first consider *Morph* and *MorphSVM*, for which the corresponding ROC curves are shown in figure 35. We have also plotted the detection rate for *MorphSVM* as a function of range, and for different false positive values, and we show this in **Figure 36**. In both the case of *Morph* and *MorphSVM*, the variable affecting rates is a threshold. For *Morph*, the threshold is the value returned by the morphological filter at the detection. For *MorphSVM*, the threshold is the probability according the SVM classifier, that the detection is a target. Some comments on their performance and their differences follow:

- First, **Figure 36** shows the contribution to the total detection rate by the detection rates at different ranges. So, even though the detection rate for *MorphSVM* is no higher than 95% overall, the detection rate between 2.5 and 3.5 miles reaches 100%.

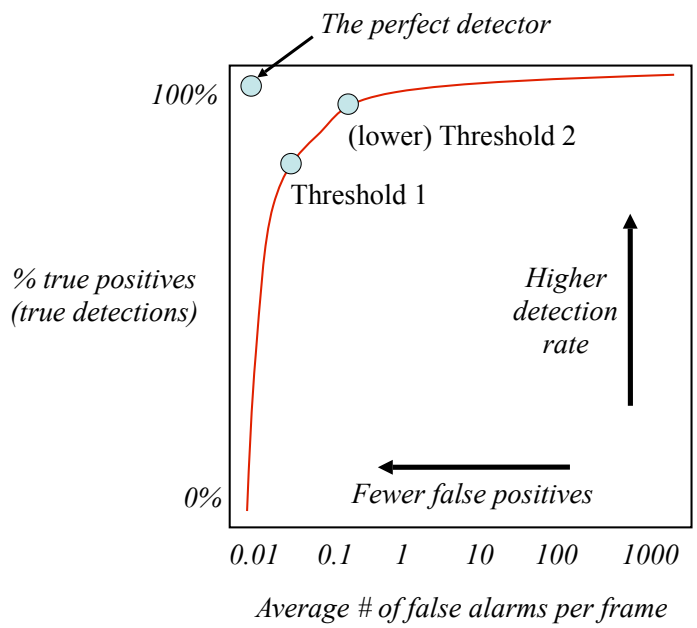


Figure 34: This plot is a hypothetical example of a ROC curve. In a ROC curve one plots false positive rate versus true positive rate. The false positive rate is on the x-axis, and the true positive rate is on the y-axis. A perfect detector is one that does not return false positives, but which detects 100% of the true positives, and is represented by the point in the upper left-hand corner. Normally, though, a detector will reject some true positives, and accept some outliers (negatives). Some internal parameters, such as a threshold will affect the true and false positive rates. For example, if a threshold is used to classify responses, then as we increase the threshold, for example, the false positives may go down, but so will the detection rate.

- **Figure 36** also shows that the detection rate actually *decreases* at closer ranges. This is due to the fact that the algorithms were not optimized for close ranges; it would have added extra complexity to the software architecture as it was being built up. It should not be difficult to improve the numbers at these ranges; for example, a first approach would be to downsample the image and apply the same algorithms.
- Some specific numbers from *Morph* and *MorphSVM* are:

TP %	***Morph* FP/frame**	***MorphSVM* FP/frame**	**FP Reduction Factor**
95%	120	20	5.9×
90%	66	3.9	17×
80%	14	1.0	14×
70%	8	0.66	12×
60%	6.2	0.56	11×

  So we find that the *MorphSVM* stage improves the false positive rate by a factor between about 6 and 17, depending on the detection rate, over *Morph*.

There is quite an improvement in false positive rates of *MorphSVM* over *Morph*, however, the improvements are not sufficiently low when we are aiming for false positives measured in false positives per hour, not frame. At this point, even with tracking, we are not able to reduce false positives to only a few per hour, however, we have a vast

improvement when using tracking. **Figure 37** shows the ROC curve for all three of the stages, and **Figure 38** shows the detection rates versus range for different false positive rates. Whereas before the value affecting performance was a threshold on the output of a filter or classifier, in this case the threshold is the number of frames for which a target has been tracked. Several comments follow:

- First, note that the best overall detection rate of *MorphiseSVM* is higher than the best overall detection rate of *MorphSVM*, even though it is based on the output of *MorphSVM*. We believe this is a temporal effect, in that detections that are intermittently below threshold, are picked up by the tracker. This seems to be corroborated by **Figure 38**, which shows that the detection rate improves noticeably at longer rages, where the signal-to-background ratio would probably fluctuate near threshold values.

- Again, the detection rate decreases slightly at closer ranges, though to a lesser degree in *MorphiseSVM*. is slightly that would have been rejected had they been below threshold are maintained because of the history of the target.

- **Figure 39** better shows the affects of minimum frame number on performance; the points along the curve are the number of frames that a target has to have been tracked for, for it to be declared a possible target; all others are rejected. In our experiments we let this threshold go up to 30 frames, at which point the false positive rate was 0.014 FP/frame, and detection rate 92%. Notice though that this curve is very flat. Most of the true positives have long tracks, and almost all outliers have short tracks. However, what this means is that if targets have to pass a 30 frame test, then we have to wait 30 frames after we first see a potential target to decide that it is indeed a target.

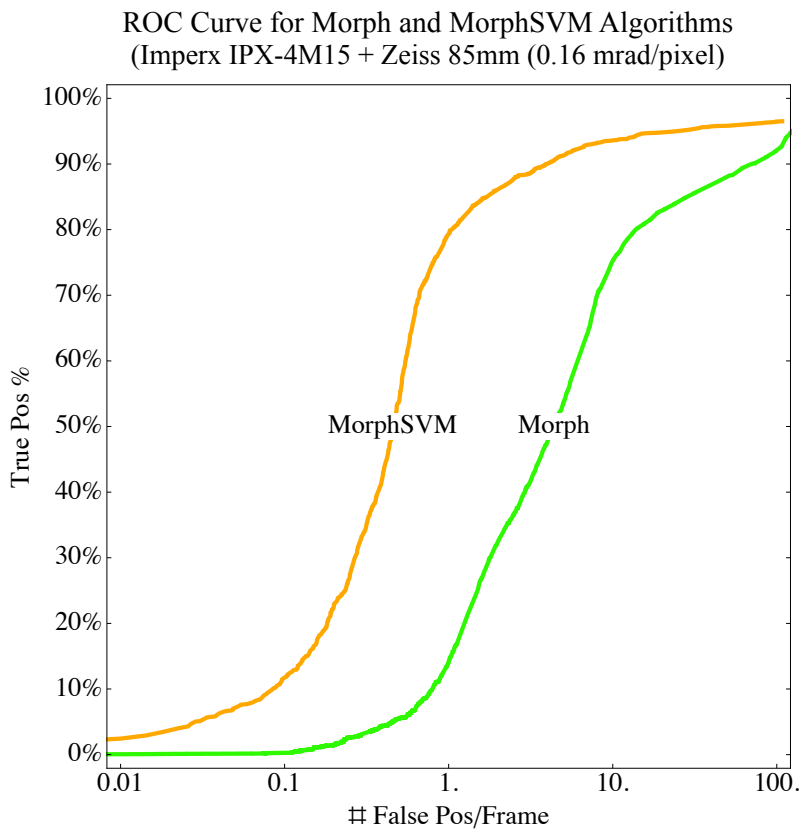


Figure 35: ROC curves for the Morph and MorphSVM algorithms after non-maxima suppression.

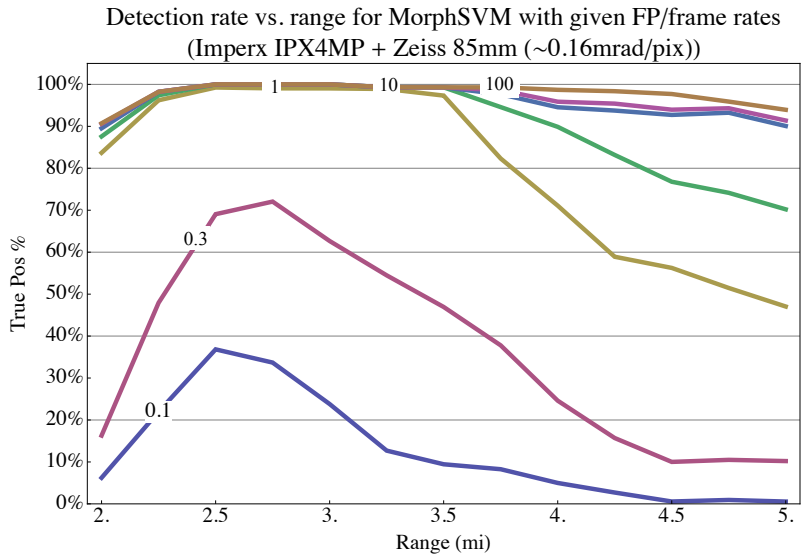


Figure 36: This plot shows for the MorphSVM algorithm the percentage of true positives detected as a function of range. Each of the curves correspond to a different false positive rate, the value of which is the curve's label.

- Overall there is a significant decrease in the number of false positives per frame. We add to the previous table some specific numbers:

TP %	***Morph* FP/frame**	***MorphSVM* FP/frame**	***MorphiseSVM* FP/frame**	**FP Reduction Factor**
97%	-	-	7.3	-
95%	120	20	0.035	571×
90%	66	3.9	-	-
80%	14	1.0	-	-
70%	8	0.66	-	-
60%	6.2	0.56	-	-

  The entries for detection rates below 95% are not filled in because we did not evaluate choose to increase the threshold frames beyond 30. If we had, the detection rate would have eventually fallen. We find that *MorphiseSVM* improves the false positive rate by a factor of over 500 times over *MorphSVM*.

- We found a reasonable compromise in false positive and true positive rate when we insisted that targets be tracked for at least 10 frames. We used this value when generating movies. Then the overall detection rate was 95%, the false positive rate was 0.05 false positives per frame, and the detection rates at different ranges are given in the table below:

Range:	2.0	2.25	2.5	2.75	3.0	3.25	3.5	3.75	4.0	4.25	4.5	4.75	5.0 miles
TP %:	88.5	98.8	99.7	100.0	100.0	99.8	99.7	99.5	96.6	96.4	96.2	98.1	97.0%

  So the detection rate is 100% out to 3 miles.

- Almost of the all of the false positives that made it through tracking at least 10 frames were items that are of interest to collision avoidance. In other words, the detection rate only accounts for the simulated intruder aircraft. Most of the false positives were seagulls or landmarks on the ground that were not segmented out

by the horizon detector (e.g., an antenna in the distance). **Figure 40** shows a histogram of the types of false positives found in one sequence. Though these targets are of interest, and thus could be considered useful, they were not detected 100% of the time. There were instances, for example, where a seagull would be tracked for tens of frames, then lost for several frames, and then tracked again; and the antenna was not always detected.

Overall we have demonstrated a significantly better false positive rate, and we believe that with little additional processing would it be feasible to further reduce the false positive rate. The next challenge is how to do either rough state estimation from target size and its looming, or whether to integrate and fuse it with other active or passive sensors.

Detection bias due to manually ground truthing images. We point out that the ground truth was obtained by going through the images by hand, and labeling the position of the target manually. We did our best to find targets by going backwards through sequences, so that even if the signal-to-background ratio was low, we could track the target by hand, manually adjusting contrast until it was too indistinguishable from noise. Hence, if we were unable to label a target by hand, we could not include it in the sequence, since we could not verifiably say that the target was not there, and if it was we did not have its position. So there is a detection bias, in that the detection rate is a percentage of all targets that we were able to label by hand. Thus, if an image was saturated, we could not locate the target by hand; had we included this frame, the detection rate would have been lower.

It would have been preferable to eliminate this hand-labeling bias by doing the ground truthing automatically. However, there were obstacles to doing ground truth labeling completely automatically: (1) We were unable to obtain state information from the pan-tilt unit that would have enabled us to determine where the pan-tilt unit was pointing. The unit used the HTTP protocol to process commands, and it appeared that requests for state information were handled synchronously, so that state information requests induced an extra latency and actually hampered the target

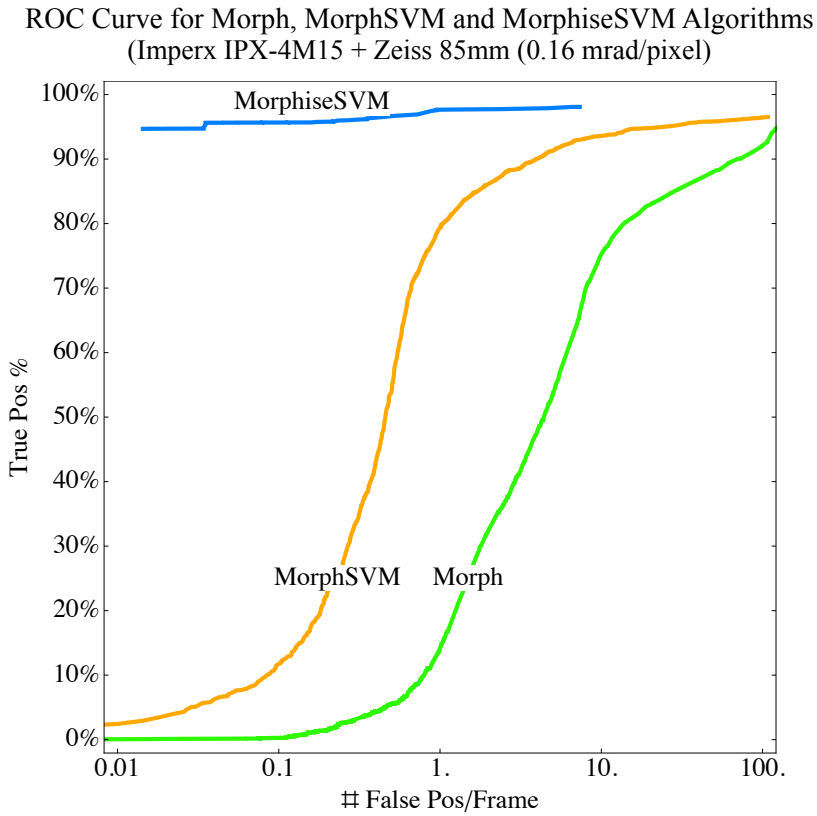


Figure 37: ROC curves for the Morph, MorphSVM and MorphiseSVM algorithms.

tracking performance by the pan-tilt unit. (2) We were only able to get pan-tilt calibration accurate to about 1°, resulting in errors that could exceed 100 pixels (this figure depends on the camera). The calibration process for the pan-tilt unit was cumbersome, had to be performed manually, and was highly dependent on the distance and variability of landmarks used as calibration points.

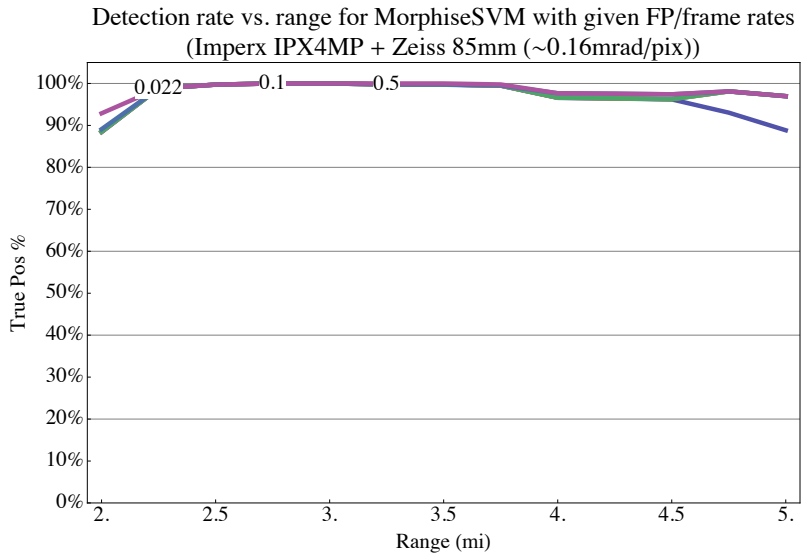


Figure 38: This plot shows for the MorphiseSVM algorithm the percentage of true positives detected as a function of range. Each of the curves correspond to a different false positive rate, the value of which is the curve's label.

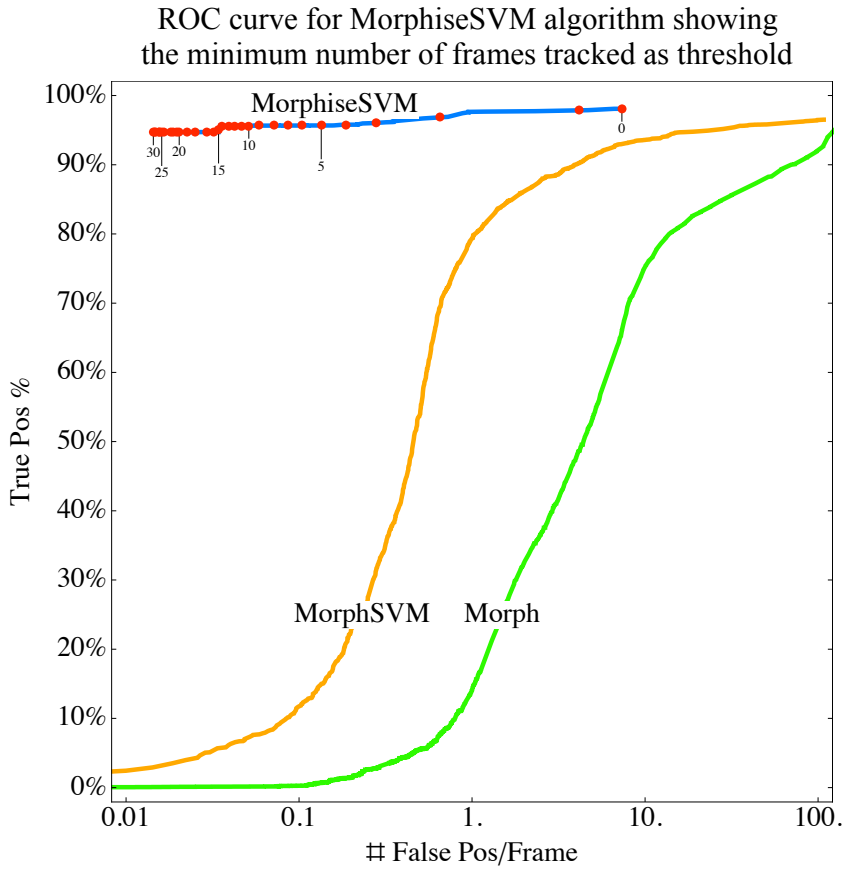


Figure 39: The threshold used to generate the ROC curve for *MorphiseSVM* is the number of frames for which a target is tracked. If a target is not tracked for at least that number of frames, then it is rejected. This plot shows the true positive and false positive rates as a function of this threshold. The number labeling a point is the threshold number of frames that were used to get that true positive and false positive rate.

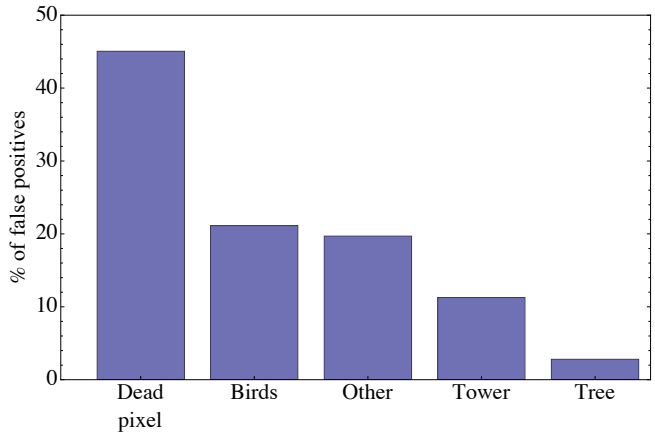


Figure 40: Breakdown of false positives. Almost all false positives that occurred were either dead pixels, birds, or ground features such as towers or trees above the horizon.

5 Conclusion

In this project we have aimed to advance sense and avoid technology. Toward this goal we have developed prototype aircraft detection algorithms that have high detection rates even when the aircraft is more than $3\,\mathrm{miles}$ away. We showed in this report the ability to detect aircraft out to $3\,\mathrm{miles}$ with over 99% detection rate, and false positive rate as low as 0.05 false positives per frame, or one false positive every 20 frames. In addition to algorithm development, we analyzed a number of the issues that affect detection rates, and worked toward determining the minimum resolution required to detect out to a given range for a sense and avoid system. We found that for to reliably detect out to $3\,\mathrm{miles}$ would require sensors with $0.2\,\mathrm{mrad}/\,\mathrm{pixel}$ resolution.

In addition to our work on aircraft detection, we developed a prototpye collision avoidance system. We integrated this with a software simulation of a Piccolo autopilot system, and successfully demonstrated software-in-the-loop simulation of the collision avoidance planner with the autopilot. We found that the developed collision avoidance planner was able to find safe avoidance maneuvers as long as the other aircraft is more than $2\,\mathrm{km}$ away, it is flying less than $250\,\mathrm{knots}$, and our aircraft is flying at least $40\,\mathrm{knots}$.

5.1 Future Work

There are a number of issues that need to be looked at in future work on both collision detection and collision avoidance, these include:

- **State estimation and collision detection.** In this work the aim has been to develop a collision detection and warning system. Though, we have progressed significantly in the area of aircraft detection from passive sensors, we still need to address how to decide whether a given target is in danger of colliding with us. Several issues need to be resolved in this area, in particular:
  - Should we combine a passive aircraft detection system with an active sensor that can be used to obtain range?
  - Is it possible to infer enough information from a passive system alone? Human pilots do not have precise range information, yet they have to obey VFR; is it possible to use target looming and/or other measures to make some intelligent guesses about range?
- **Aircraft detection below the horizon and detection on-board the aircraft.** Another top priority should be detection below the horizon, as well as other issues related to . Detection below the horizon will be especially challenging. A good start may be to consider modeling targets below the horizon, in the same way we have modeled them above, so as to better understand the inherent signal-to-background issues that can arise.
- **Computation.** Computation is another challenge that needs to be resolved. Though the current algorithm takes about $0.8\,\mathrm{sec}$ per $4\,\mathrm{megapixel}$ frame—far too slow for real-time—we saw that most of the computation is image processing. Specialized hardware is likely to help in this regard, and there are a number of possibilities, including a DSP, FPGA or GPU. Graphical processing units (GPUs) are promising—NVIDIA's latest offering, for example, has 240 processors—but there are few embedded options for GPUs. However, before finalizing any decisions, the remaining issues in state estimation and detection below the horizon ought to be decided on before fully considering the best computational platform(s).
- **Complete system design.** All of the above issues affect how a collision detection and warning system should be designed so as to cover the desired field-of-regard. A design needs to be formulated that satisfies the requirements.
- **Right-of-way.** In collision avoidance, we have not incorporated the constraints on right-of-way. One question is how to incorporate the right-of-way rules in the current or any other algorithm, but also how to weigh obeying right-of-way rules with . For example, if a situation can be made significantly safer by not obeying right-of-way rules, when should you take the alternative? How do you define "significantly"?

- **More comprehensive SIL testing.** More comprehensive software-in-the-loop testing needs to be performed on the collision avoidance algorithms before flying them. We may need an automated method for testing the safety of any collision avoidance algorithm.

This covers many of the items that we believe need to be addressed in future work. There is much to do, but all of the items either have paths forward, or the open problems are relatively clearly defined.

References

[1] Nineteen UAV types currently operating in Iraq. http://www.livingroom.org.au/uavblog/archives/nineteen_uav_types_currently_operating_in_iraq.php, Nov. 2005.

[2] AOPA testifies on UAV threat to general aviation. http://aero-defense.ihs.com/news/2006/aopa-uav-certify.htm, Apr. 2006.

[3] Nasa predator drone is flying missions on california fires. http://calfire.blogspot.com/2007/08/nasa-predator-drone-is-flying-missions.html, Aug. 2007.

[4] Geoffrey Alpert. Helicopters in pursuit operations, Aug. 1998.

[5] Rich Bonin. How technology won sadr city battle. http://www.cbsnews.com/stories/2008/10/09/60minutes/main4511800_page4.shtml, Oct. 2008.

[6] Anne Broache. Army official: UAVs are 'unsung heroes' in Iraq. http://news.cnet.com/8301-10784_3-9883300-7.html, Feb. 2008.

[7] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[8] Philip Finnegan. Predator showing promise in homeland security UAV market. http://www.hstoday.us/content/view/1245/27/, Jan. 2006.

[9] E. Frazzoli, MA Dahleh, and E. Feron. Robust hybrid control for autonomous vehicle motion planning. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, 2000.

[10] J. D. Gallop. Police drone plan draws fire. http://www.infowars.com/articles/bb/drone_plane_plan_draws_fire.htm, Feb. 2007.

[11] Tarak Gandhi, Mau-Tsuen Yang, Rangachar Kasturi, Octavia I. Camps, Lee D. Coraor, and Jeffrey McCandless. Detection of obstacles in the flight path of an aircraft. In *CVPR*, pages 2304–2311. IEEE Computer Society, 2000.

[12] Tarak Gandhi, Mau-Tsuen Yang, Rangachar Kasturi, Octavia I. Camps, Lee D. Coraor, and Jeffrey McCandless. Performance characterization of the dynamic programming obstacle detection algorithm. *IEEE Transactions on Image Processing*, 15(5):1202–1214, 2006.

[13] P.E. Hart, N.J. Nilsson, and B. Raphael. Search. *ACM SIGART Bulletin*, pages 28–29, 1972.

[14] LE Kavraki, P. Svestka, J.C. Latombe, and MH Overmars. Probabilistic roadmaps for path planning in high-dimensionalconfiguration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[15] JJ Kuffner Jr and SM LaValle. RRT-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, 2000.

[16] Shree K. Nayar and Srinivasa G. Narasimhan. Vision in bad weather. In *Proc. IEEE International Conference on Computer Vision*, pages 820–827, 1999.

[17] Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Courier Dover Publications, 1998.

[18] Noah Shachtman. Red tape grounds Iraq's robo-planes. `http://blog.wired.com/defense/2007/09/the-american-mi.html`, Sept. 2007.

[19] RF Stengel. Toward intelligent flight control. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(6):1699–1717, 1993.