Precision Multi-Sensor Optical Navigation Test-bed Utilizing Ground-Truthed Data Set

Jimmy Touma
Air Force Research Laboratory
Email: jimmy.touma@eglin.af.mil

Timothy Klausutis
Air Force Research Laboratory
Email: timothy.klausutis@eglin.af.mil

Michelle Fessenden
Air Force Research Laboratory
Email: Michelle.Fessenden@eglin.af.mil

Carolyn New
Air Force Research Laboratory
Email: carolyn.new@eglin.af.mil

David D. Diel
Scientific Systems Company
Email: ddiel@alum.mit.edu

Abstract

This work describes a rich set of navigation data that has been collected by AFRL and an object-oriented framework that makes both the data and relevant processing algorithms available to the research community. Both aerial and ground vehicle platforms were employed, exposing a single sensor suite to a variety of conditions relevant to civil and military applications, including urban and rural terrain and varied time-of-day. The sensor package includes four visible cameras, two LWIR cameras, inertial measurement units ranging from navigation grade, tactical grade and industrial grade. A GINS is utilized to establish platform attitude and position truth. The raw GPS signals are also collected. All this data is synchronously time-stamped with reference to a unified reference time.

Of particular interest to the research community is the use of visual sensors to navigate in GNSS-limited environments, especially in cases where the navigation solution must exhibit dynamic continuity. Our data delivery framework goes beyond standardizing data formats by providing a means to standardize the application of relevant processing algorithms. This plays a critical role in integrating data from multiple sensors without violating dynamic constraints. In addition, the framework can be viewed as a testbed for user contributed algorithms, which will help system integrators to explore the trade space of both sensors and algorithms.

1 Alternate Navigation Test Bed

The Alternative Navigation group at AFRL/Eglin AFB is engaged in a variety of high precision ground and aerial data collection efforts. We've established a data collection testbed called Alternate Navigation TestBED (ANTBED) that encompasses data collection, reduction, and analysis as well as the evaluation of various multi-sensor data and vision algorithms. This allows us to perform trade studies on different sensor combinations and properties, fuse the gathered data, and obtain an ego-state estimate and global position fix, as shown in Figure 1.

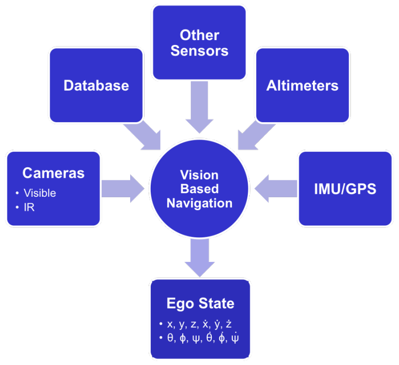
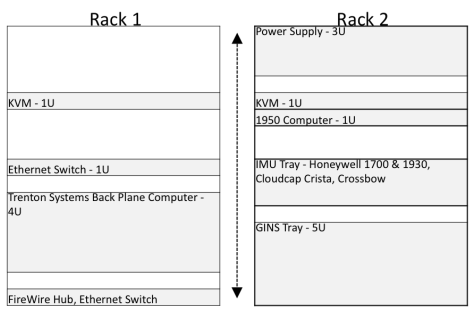


Figure 1: Multi-sensor Fusion Framework

This is accomplished by using specialized hardware that collects data synchronized to a 1 pulse per second signal. With the current sensors suite the data bandwidth is about 60MB per second. A schematic of the interfaces is shown in Figure 3. To avoid buffer over-flow and dropping frames, the visible camera data is streamed to one hard drive, the IR camera data to a second, and the data from the rest of the sensors to a third.

1.1 Data-Collection Computer and Software

Data-collection is handled by a 4U backplane with special interface cards that connect to all the sensors in the system. The computer is a dual-quad core Xeon, 4GB RAM, Windows XP board. The software is a customized multi-threaded C++ application that interfaces with the sensors at a very low level. It also provides a GUI interface that monitors the status of the system and sensors and allows the use to display live data and images.



Rack 1
KVM - 1U
Ethernet Switch - 1U
Trenton Systems Back Plane Computer - 4U
FireWire Hub, Ethernet Switch
Rack 2
Power Supply - 3U
KVM - 1U
1950 Computer - 1U
IMU Tray - Honeywell 1700 & 1930, Cloudcap Crista, Crossbow
GINS Tray - 5U

Figure 2: Rack setup common to both ground and aerial testing. The system collects data at 60 MB per second

1.2 Objectives

The following are the objectives of ANTBED:

- Produce multi-sensor database for alternative navigation research to evaluate the performance and further the development of vision aided and alternative navigation algorithms.
- Perform trade study on sensor suites and vision algorithms by using data from multiple tests under multiple operating conditions including routes, backgrounds, maneuvers, urban canyons, EO/IR sensors, etc…
- Validate hardware/software architecture for long term data collection efforts
- Capture/validate operating procedures
  - Mission planning procedures established/followed
  - Post data analysis procedures established/followed
  - Flight reports and lessons learned documented

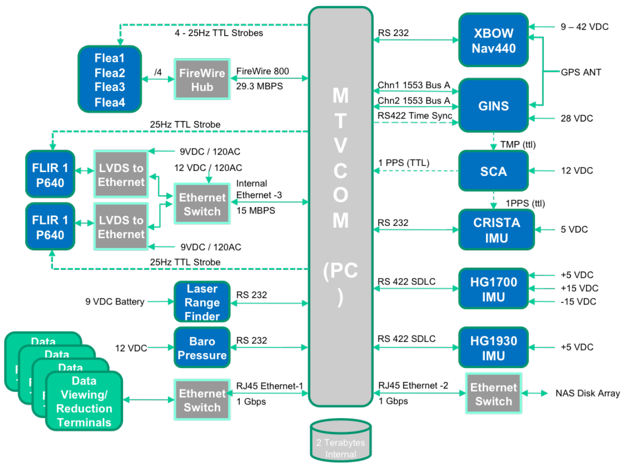


Figure 3: ANTBED Interfaces

2 Test Procedure

ANTBED follows a test procedure as shown in Figure 4. After going through the approval process, a fully functional system in setup on the mobile test vehicle (see 3.1) and is thoroughly tested right before the flight test. ANTBED also has a go/no-go check list that is continuously checked, as described in Appendix A. The plane is surveyed prior to the first fight test in order to obtain the position of cameras, IMUs, GINS, and GPS antenna relative to a common origin. The visible cameras are calibrated prior to each flight test to ensure intrinsic and extrinsic parameter integrity in the case where cameras shift position during a flight test. We use the Matlab Calibration Toolbox [1] to evaluate intrinsic parameters and follow the approach of Kanade, et al [3] andKumar, et all [4] to compute the extrinsic parameters.

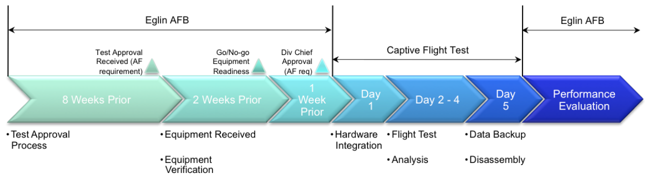


Figure 4: Notional Captive Flight Test Schedule

3 Test Vehicles, Hardware, and Software

One advantage that ANTBED has is that the hardware and software used for ground testing is also the same that is used for captive flight tests. Moreover, the pre-flight ground test mentioned in Section 2 can be designed to collect data for ground-test analysis.

3.1 Mobile Test Vehicle (MTV)

The Mobile Test Vehicle (MTV) is a retro-fitted bus that provides low speed dynamic test environments. It houses MTV-COM, the data collection system, that provides real-time graphical outputs over distributed viewing station and automated real-time/post mission data analysis. Currently ground testing is limited to Eglin AFB where the terrain consists of moderately tall buildings, trees and shrubs on the road-side, and some open space.

3.2 Beech BE-18

ANTBED is currently hosted on a Beech BE-18 craft and is operated by a local contractor Sunshine Aero in Crestview, FL. Figure 7 shows the placement of the racks in relation to the cokpit. Figure 6 shows the placement of the camera mount on a plate at the bottom of the plane. The craft is retrofitted to host the same two racks that are used on MTV, whith a maximum allowed weight of 250lb per rack. The plane has room for 4

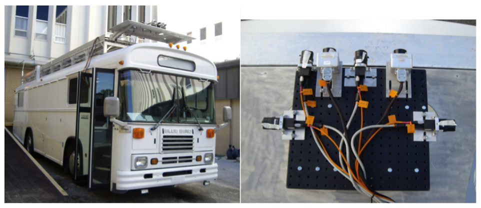

(a) Bus (b) Camera Suite

Figure 5: Mobile Test Vehicle

operators but at least two shoue be present to operate the equipment on the two racks. The plane can fly for approximately 6 hours without refueling so that gives us the ability to cover many different terrain during one flight.

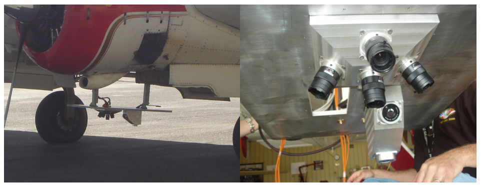

(a) Beech Craft (b) Camera Suite

Figure 6: Captive Flight Test Plane

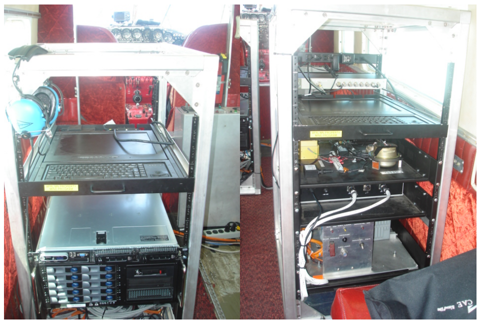

(a) Camea Interfaces and Computer Rack (b) GINS/IMUs/Baro and Monitoring Rack

Figure 7: Captive Flight Test Plane

3.3 Sensor Suite

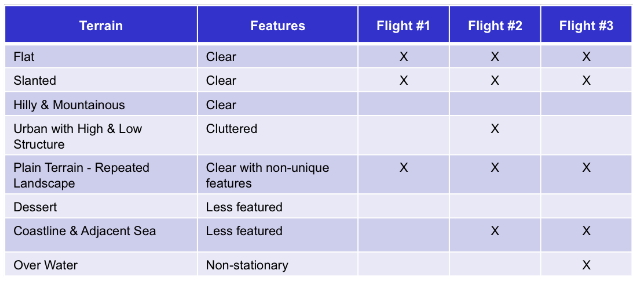
ANTBED's sensor suite consists of the following sensors. A schematic of the layout is given in Figure 2. *Cameras*: The camera suite consists of 4 Point Grey Flea2 Visible Cameras with 1394b interface, Computar Lens H2Z0414C-MP, F1.4, 4-8mm focal length, and manual iris. Two FLIR Photon 640 IR Cameras, F1.4, 25mm lens, and Gigabit Ethernet interface.

GPS/INS Sensors: The GPS/INS sensor suite consists of the GINS provides integrated navigation solution and is used as thruthing system; two low-cost tactical-grade IMUs: Honeywell 1700(ring laser gyro) and 1930 (MEMS gyro) for hight accuracy and reliability; CloudCap Crista IMU MEMS grade IMU that provides high resolution digital rate and acceleration data; Crossbow 440 INS that provides full inertial data, GPS position and inertially derived velocity.

Other Sensors: Honeywell Precision Barometer, Newcon Optik laser range finder that provides distance to features (was not operational for the first captive flight test due to a faulty power connection).

4 Captive Flight Test Data

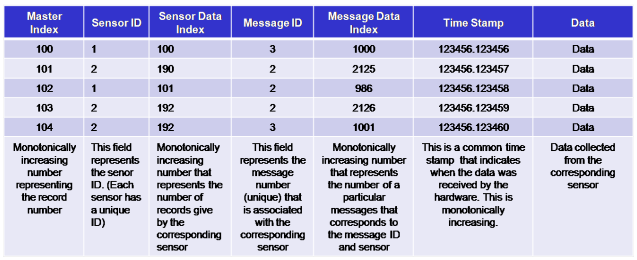
ANTBED conducted it's first captive fligh test during the first weeks of April 2010 and currently we are in the process of data reduction, analysis, and reformatting. We performaed three data-collection flights that covered a large set of different terrain, as shown in Figure 8.



Terrain	Features	Flight #1	Flight #2	Flight #3
Flat	Clear	X	X	X
Slanted	Clear	X	X	X
Hilly & Mountainous	Clear			
Urban with High & Low Structure	Cluttered		X	
Plain Terrain - Repeated Landscape	Clear with non-unique features	X	X	X
Dessert	Less featured			
Coastline & Adjacent Sea	Less featured		X	X
Over Water	Non-stationary			X

Figure 8: Terrain covered by the first ANTBED fligh tesst in April 2010

Due to the large size of the data and data-files, we are in the process of reformatting the data into a few set of self-contained "archives". Each archive will have a header that contains information about the data in that archive, which will have a format outlined in Figure 9. This format will be made available to the community along with sample acess codes writen is Python, C++, and Matlab. Also in dvelopment is a GUI tool that will make it easy to view the data and extract a interesting chunck of it for analysis. This will also be made available to the community upon completion.



Master Index	Sensor ID	Sensor Data Index	Message ID	Message Data Index	Time Stamp	Data
100	1	100	3	1000	123456.123456	Data
101	2	190	2	2125	123456.123457	Data
102	1	101	2	986	123456.123458	Data
103	2	192	2	2126	123456.123459	Data
104	2	192	3	1001	123456.123460	Data
Monotonically increasing number representing the record number	This field represents the senor ID. (Each sensor has a unique ID)	Monotonically increasing number that represents the number of records give by the corresponding sensor	This field represents the message number (unique) that is associated with the corresponding sensor	Monotonically increasing number that represents the number of a particular messages that corresponds to the message ID and sensor	This is a common time stamp that indicates when the data was received by the hardware. This is monotonically increasing.	Data collected from the corresponding sensor

Figure 9: ANTBED "archives" data format

5 Trajectory Optimization Framework

In addition to providing standardized multi-sensor data, we have created a framework for developing and testing algorithms that aid navigation, where "navigation" is defined as continuous self-localization relative to a local or global frame. This framework supports our own datasets as well as user-supplied data. It serves two major purposes: i) To facilitate the development of new trajectory optimization algorithms with a focus on alternative sensors other than GPS; and ii) To assist system integrators who must test a variety of sensors and algorithms on a level playing field and combine them to acheive desired accuracy. Our approach is to define the broad structure of the navigation problem while leaving the details to be implemented differently for each application.

The framework consists of four interfaces (abstract classes) that divide the navigation problem into manageable components, as illustrated in Figure 10. A `DynamicModel` is an algorithm that relates a set of generic parameters to the rigid-body trajectory of a physical system such as a car or airplane. It also generates "costs", which are discussed later in this section. Sensor data comes into the system through a customizable hardware abstraction class labeled `DataContainer`. Each `Measure` relates sensor data to a trajectory hypothesis and generates additional costs. Finally, at the core of the framework is the `Optimizer`, which is an algorithm that minimizes a combination of costs by intelligently generating and modifying the parameters of one or more `DynamicModel` instances and passing them through zero or more `Measure` instances.

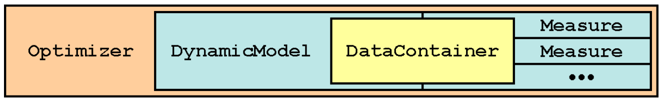


Figure 10: Block diagram of framework components.

A user interacts with the framework by specifying a set of components that derive from the framework classes. These components represent the physical system and define "optimal" for a specific application. The framework then sets up a scalar objective function to be evaluated and optimized. Further details of this approach are provided below, and framework code is available in both MATLAB and C++ online [2].

5.1 Dynamic Model

Consider the following canonical state transition models for continuous and discrete nonlinear systems:

$$\underbrace{\dot{\boldsymbol{x}}_t = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{v}_t\right)}_{\text{continuous}} \qquad \underbrace{\boldsymbol{x}_{n+1} = \boldsymbol{f}\left(\boldsymbol{x}_n, \boldsymbol{u}_n, \boldsymbol{v}_n\right)}_{\text{discrete}} \tag{1}$$

These equations have been established to model the dynamics of numerous physical systems, from Brownian particles to holonomic ground vehicles, fighter jets, and even animals. They are typically placed in an integration loop, such that the state $\boldsymbol{x}$ is computed incrementally at increasing time instants t, given an initial condition. The symbols $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ all represent vector-valued functions indexed by time. The input $\boldsymbol{u}$ represents data that is known, and the input $\boldsymbol{v}$ represents parameters whose probability distributions are known. If an interpolation method is specified, then both the continuous and discrete models can be written in *functional* form as follows:

$$\boldsymbol{x} = \boldsymbol{F}\left(\boldsymbol{v}; \boldsymbol{u}\right) \tag{2}$$

Our `DynamicModel` class standardizes the interface to $\boldsymbol{F}$. It requires that $\boldsymbol{x}$ contain a C^1 continuous 6-DOF rigid-body trajectory relative to an Earth-Centered Earth-Fixed (ECEF) frame. It accesses the data $\boldsymbol{u}$ implicitly, a detail that we notate using a semicolon (;) in the argument list. It requires that each parameter vector $\boldsymbol{v}_n$ consist of boolean, integer, and/or bounded real parameters. And, it defines an indexing system that allows the domain of $\boldsymbol{x}$ to grow with the domain of $\boldsymbol{v}$ in a consistent manner as time moves forward. Finally, assuming that each parameter vector $\boldsymbol{v}_n$ is independently distributed, the `DynamicModel` associates a cost function r with the negative log likelihood of the normalized parameter distribution.

$$c_n = r\left(\boldsymbol{v}_n; \boldsymbol{u}\right) = -\ln\left(\frac{p\left(\boldsymbol{v}_n|\boldsymbol{u}\right)}{\left\|p\left(\boldsymbol{v}_n|\boldsymbol{u}\right)\right\|_\infty}\right) \tag{3}$$

5.2 Measure

Many sensors can be modeled by the following canonical form:

$$\boldsymbol{y}_n = \boldsymbol{g}\left(\boldsymbol{x}_n, \boldsymbol{u}_n, \boldsymbol{w}_n\right) \tag{4}$$

where each measurement $\boldsymbol{y}_n$ arises from the instantaneous state $\boldsymbol{x}_n$, the known data $\boldsymbol{u}_n$, and the stochastic vector-valued function $\boldsymbol{w}_n$. However, some measurements of interest cannot be accurately modeled in this form, because they are not inherently instantaneous. A prime example is a feature match between two images that were acquired at different times. Regardless of whether one uses Optical Flow [7], SIFT [6], normalized cross-correlation [5], or another technique, the computed feature displacement will depend on the position and orientation of the sensor at each time. Another sensor example is a gyroscope, which can be modeled as if it measures instantaneous rotation rates, but is most accurately modeled as measuring changes in orientation over discrete time periods.

One approach to dealing with sensors of the type described above is to create a combining function $\boldsymbol{\psi}_{ab} = \boldsymbol{\gamma}\left(\boldsymbol{y}_a, \boldsymbol{y}_b\right)$, where the pair of indices labeled ab indicate a measurement arising from two times, $n = a$ and $n = b$, sorted such that $a \leq b$. This implies an indexing system that has a graph structure instead of a simple linear index. Each discrete time is associated with a node (or vertex), and each node pair forms an edge. In general, not all of the nodes are connected, making it an incomplete graph. This idea can be incorporated into Equation 4 by rewriting it in *functional* form:

$$\boldsymbol{y}_{ab} = \boldsymbol{g}\left(\boldsymbol{x}, ab, \boldsymbol{u}, \boldsymbol{w}\right) \tag{5}$$

where the sensor accesses its arguments as functions. The functional $\boldsymbol{g}$ can evaluate the body trajectory at any time in its domain, which we assume includes the span $t \in [t_a, t_b]$. Likewise, it can evaluate its other arguments within their valid domains. As a generalization of Equation 4, it can also be used to simulate measurements. However, for the purpose of trajectory optimization, simulated measurements may not be necessary as long as we have a function that can test a hypothetical trajectory against a set of sensor data. Therefore, we define a cost function s for each algorithm or sensor m as follows:

$$c_{m,ab} = s_m\left(\boldsymbol{x}, ab; \boldsymbol{u}\right) = -\ln\left(\frac{p_m\left(\boldsymbol{u}|\boldsymbol{x}, ab\right)}{\|p_m\left(\boldsymbol{u}|\boldsymbol{x}, ab\right)\|_\infty}\right) \tag{6}$$

Our `Measure` class standardizes the interface to s without requiring explicit computation of $\boldsymbol{y}$ or $\boldsymbol{w}$. This not only has the potential to reduce processor burden, but it also makes it possible to wrap a wide variety of sensors and algorithms with a uniform interface. For example, suppose s_1 represents a "smart camera" running a sparse feature tracker, and s_2 represents a GPS unit. Since they both posess the same interface, they can be tested and their performance can be compared objectiely, a property that is especially helpful to system integrators.

5.3 Optimizer

Once cost functions have been defined, putting together the overall objective is straightforward. All costs are additive by definition, which implies the assumption of independent distributions for each parameter vector in the `DynamicModel` and for each edge contributed by each `Measure`. Therefore, the optimization problem boils down to the following composition:

$$\begin{aligned}\boldsymbol{v}^* &= \underset{\boldsymbol{v}}{\operatorname{argmin}}\left\{\sum_n c_n + \sum_m \sum_{ab} c_{m,ab}\right\} \qquad (7)\\ &= \underset{\boldsymbol{v}}{\operatorname{argmin}}\left\{\sum_n r\left(\boldsymbol{v}_n; \boldsymbol{u}\right) + \sum_m \sum_{ab} s_m\left(\boldsymbol{F}\left(\boldsymbol{v}; \boldsymbol{u}\right), ab; \boldsymbol{u}\right)\right\}\end{aligned}$$

where the solution $\boldsymbol{v}^*$ is an optimal parameter hypothesis. To find the optimal trajectory $\boldsymbol{x}^*$, this needs to be inserted back through the dynamic model as follows:

$$\boldsymbol{x}^* = \boldsymbol{F}\left(\boldsymbol{v}^*; \boldsymbol{u}\right) \tag{8}$$

Our `Optimizer` class wraps algorithms that query r and s in order to search for global minima in Equation 7. In general, neither convexity nor uniqueness of solution are guaranteed; however, specific sets of components can be designed to offer these guarantees. To facilitate efficient optimization, the costs can be computed individually before the summation takes place. This formulation suggests automated learning of the relationship between time indexed parameters and time indexed graphs of costs, a subject beyond the scope of this paper. In future work, we plan to demonstrate that our framework generalizes a wide range of trajectory optimization techniques including EKF, UKF, gradient descent, genetic algorithm, simplex, and the pose graph method [8].

6 Conclusion and Way Forward

We have developed time synchronous multi-sensor data collection asset, ANTBED, to support sensor aided navigation techniques. ANTBED

- provides "truth" as a refernce to compare to
- supports multi-sensor configurations for sensor and algorithm trade studies
- operate in multiple sensing conditions such as varying terrain, manuevers, ...

The sensor test bed was demonstrated with initial flight test. Future data production flights to occur may include other sensors and cover differnet terrain. The Air force will be developing a standard challenge problem to facilitate research from industry and academia. Truth will be provided for some flights with other truth being held back for independent evaluation. We've also developed vision-adided, multi-sensor framework for trajectory optimization. The framework seamlessly uses ANTBED data and is open and modular enough to use user-supplied data. Researchers are encourage to download, use, and if provide any feedback ragarding the framework inorder to better the product.

A Go/No-Go Criteria

ANTBED adheres to a strict go/no-go criteria for the purposes of safety and to insure data integrity. Here is a list of the criteria:

Equipment

- All software is installed and communicating and gatherting data from all hardware
- GINS is tacking
- All primary sensors operational and properly recording data
- We have Visual Flight Rules (VFR) conditions & clear line of sight to ground
- No more than 20% cloud cover below flight altitude (our flight altitde varies from 1500 ft to 3000 ft)

Weather

- The ultimate go/no-go authority resides with the pilot in command provided by Sunshine Aero. VFR/IFR conditions will be determined by pilot in command.

- Adhere to Visual Flight Rules (VFR): Will fly VFR if conditions are at least 3mi visibility, 1500ft ceiling up to & 1 hour after scheduled completion of flight. Pilot in command will be responsible for see & avoid clearance of the airspace. Additionally, flight following via radar contact from Eglin ATC will be requested. IFR flight clearance will be recommended if weather conditions are less than 4mi/2500ft and degrading.

- Adgere to Instrument Flight Rules (IFR): If VFR conditions not present, but current/forecasted weather permit safe IFR flight, an IFR flight plan will be filed via Sunshine Aero

- Sever Weather in the region like cyclonic activity, convective sigment, lighting within 10 miles, high winds, or sudden sthunderstorm formation with result in flight delay or cancellation. Wet and high-humidity weather (with visible moisture on the plane and camera plate) with also result in flight delay or cancellation .

References

[1] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[2] Scientific Systems Company. Project “functionalnavigation” on google code. `http://code.google.com/p/functionalnavigation/`.

[3] Jun-Sik Kim, Myung Hwangbo, and T. Kanade. Motion estimation using multiple non-overlapping cameras for small unmanned aerial vehicles. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3076 –3081, 19-23 2008.

[4] R.K. Kumar, A. Ilie, J-M. Frahm, and M. Pollefeys. Simple calibration of non-overlapping cameras with a mirror. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –7, 23-28 2008.

[5] J. P. Lewis. Fast normalized cross-correlation. *Vision Interface*, 1995.

[6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[7] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th International Joint Conference on Artificial Intelligence*, 1981.

[8] Edwin Olson, John Leonard, and Seth Teller. Fast iterative optimization of pose graphs with poor initial estimates. pages 2262–2269, 2006.