DRAGONFLEYE: A PASSIVE APPROACH TO AERIAL COLLISION SENSING

CYRUS MINWALLA

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO
APRIL 2013

DRAGONFLEYE: A PASSIVE APPROACH TO AERIAL COLLISION SENSING

by **Cyrus Minwalla**

a dissertation submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

DRAGONFLEYE: A PASSIVE APPROACH TO AERIAL COLLISION SENSING

by **Cyrus Minwalla**

By virtue of submitting this document electronically, the author certifies that this is a true electronic equivalent of the copy of the dissertation approved by York University for the award of the degree. No alteration of the content has occurred and if there are any minor variations in formatting, they are as a result of the conversion to Adobe Acrobat format (or similar software application).

Examination Committee Members:

1. Dr. Richard Hornsey
2. Dr. Robert Allison
3. Dr. Mokhtar Aboelaze
4. Dr. Burton Ma
5. Dr. Costas Armenakis
6. Dr. Siu O'Young

Abstract

This dissertation describes the design, development and test of a passive wide-field optical aircraft collision sensing instrument titled 'DragonflEYE'. Such a "sense-and-avoid" instrument is desired for autonomous unmanned aerial systems operating in civilian airspace. The instrument was configured as a network of smart camera nodes and implemented using commercial, off-the-shelf components. An end-to-end imaging train model was developed and important figures of merit were derived. Transfer functions arising from intermediate mediums were discussed and their impact assessed. Multiple prototypes were developed. The expected performance of the instrument was iteratively evaluated on the prototypes, beginning with modeling activities followed by laboratory tests, ground tests and flight tests. A prototype was mounted on a Bell 205 helicopter for flight tests, with a Bell 206 helicopter acting as the target. Raw imagery was recorded alongside ancillary aircraft data, and stored for the offline assessment of performance. The "range at first detection" (R_0), is presented as a robust measure of sensor performance, based on

a suitably defined signal-to-noise ratio. The analysis treats target radiance fluctuations, ground clutter, atmospheric effects, platform motion and random noise elements. Under the measurement conditions, R_0 exceeded flight crew acquisition ranges. Secondary figures of merit are also discussed, including time to impact, target size and growth, and the impact of resolution on detection range. The hardware was structured to facilitate a real-time hierarchical image-processing pipeline, with selected image processing techniques introduced. In particular, the height of an observed event above the horizon compensates for angular motion of the helicopter platform.

Acknowledgements

The hardware implementation of the DragonflEYE sensor developed in this dissertation required the participation of technical staff at the National Research Council of Canada and York University, and from students, summer and otherwise, under my supervision at York University and NRC Canada. I, therefore, gratefully acknowledge the work of the following individuals: Paris Ang (NRC-Carleton), Stanley Lio (VISOR), Kyle Watters (VISOR), Gary Zeng (VISOR), Sepehr Nazari (VISOR), Jeremy Chan-Hao Wang (VISOR), Scott Desjardins (NRC), Michael Mullins (NRC), James Jamison (NRC), Camille Lebrun (NRC), Malcolm Imray (NRC), Scott Desjardins (NRC), Ator Sarkisoff (Petrie) and Paul Woitalla (Petrie). I would like to extend thanks to colleagues Mussie Tekeste, Neela Maharaj and Michael Liscombe for their general contributions to camera characterization.

This thesis would not have been possible without the tireless efforts of Dr. Richard Hornsey and Dr. Paul Thomas at the Vision Sensor Laboratory, whose wisdom, guidance and insight have been invaluable throughout this journey. For

that I thank them.

My supervisors at NRC Canada, Sion Jennings and Kristopher Ellis, deserve thanks for their efforts. In particular, I would like to thank Sion Jennings for being an amazing manager, and Kristopher Ellis, for being one of the smartest people I know. I would like to thank the pilots, Robert Erdos and Stephan Carignan for their expert flying under such dangerous conditions, and Dr. Gregory Craig for being my chaperone in the skies.

Thanks go out to the senior colleagues at VISOR, including newly minted Drs. Edward Shen, Wei Gao, Elliot Tsai and Christopher Thomas, for making the laboratory environment a happy (and fun!) place to be.

Last of all, I would like to thank my wife, Neha, for her unwavering faith in my abilities, and my son, Michael, for being the impetus to pen this dissertation post haste. I love you both.

Table of Contents

List of Tables

List of Figures

Abbreviations

ADS-B Automatic Dependent Surveillance - Broadcast

AGL Above Ground Level

AILS Airborne Information for Lateral Spacing(s)

AFRL Air Force Research Laboratory (US)

APS Active Pixel Sensor

APU Application Processing Unit

ARM Advanced RISC Machines

ATC Air Traffic Control

AWGN Additive White Gaussian Noise

BGA Ball Grid Array (mount)

CCD Charge-Coupled Device

CDS Conflict Detection and Resolution

CMOS Complementary Metal Oxide Semiconductor

COTS Commercial, Off-The-Shelf

CYOW Ottawa International Airport

CPU Central Processing Unit

DMA Direct Memory Access

DMIPS Dhrystone MIPS (see MIPS)

DN (dn) Digital Number

DSP Digital Signal Processor

ECEF Earth-Centered Earth-Fixed

EDSC Embedded Distributed Smart Cameras

EO Electro-Optical

EOP(s) Effective Operation(s) per second

EST Eastern Standard Time

EWGPS Enhanced Ground Proximity Warning System

FAA Federal Aviation Administration (United States)

FFT Fast Fourier Transform

FFTW Fastest Fourier Transform (of the) West

FLIR Forward Looking Infrared (camera)

FOM Figure of Merit

FOV Field-of-View

FPU Floating Point Unit

FPS Frames Per Second

FWHM Full-Width at Half-Maximum

GB/MB/KB Giga-/Mega-/Kilo- Byte

GHz Gigahertz

GMT Greenwich Mean Time

GPIO General Purpose Input Output

GPU Graphics Processing Unit

GPS Global Positioning System

IC Integrated Circuit

i.i.d Independent and Identically Distributed

IMC Instrument Meteorological Conditions

IMU Inertial Measurement Unit

INS Inertial Navigation System

IP Internet Protocol

IR Infrared

IVSS Intelligent Visual Surveillance Systems

LCD Liquid Crystal Display

LIDAR/LADAR Light Detection And Ranging

LoG Laplacian-of-Gaussian

LTI Linear Time-Invariant

LWIR Long-wave Infrared Region

MAV Micro Air Vehicle

METAR Meteorological Terminal Aviation Routine Weather Report

MFLOP Million Floating-Point Operations

MIPS Million Instructions Per Second

MHz Megahertz

MP Mega-Pixel

MSL Mean Sea Level

MTF Modulation Transfer Function

MWIR Medium Wave Infrared Region

NTP Network Time Protocol

NVD Night-Vision Device

OMAP Open Multimedia Applications Processor

OpenGL Open Graphics Library

OpenGL ES Open Graphics Library for Embedded Systems

PCA Principal Component Analysis

PDF Probability Density Function

PRM Precision Runway Monitor

PSD Power Spectral Density

PSF Point Spread Function

RAM Random Access Memory

RISC Reduced Instruction Set Architecture

ROI Region of Interest

SAA Sense-and-Avoid

SBC Single Board Computer

SDK Software Development Kit

SIMD Single Instruction Multiple Data

SSE Streaming SIMD Extensions (see SIMD)

SWIR Short-Wave Infrared Region

TCAS Traffic Alert and Collision Avoidance System

TCP Transport Control Protocol

UAS Unmanned Aerial System(s)

UHF Ultra High Frequency

UI User Interface

URET User Request Evaluation Tool

USB Universal Serial Bus

UTC Coordinated Universal Time

UTM Universal Transverse Mercator

UV Ultraviolet

UVC USB Video Class (see USB)

V4L Video For(4) Linux

VFR Visual Flight Rules

VHF Very High Frequency

VMC Visual Meteorological Conditions

1 Introduction

1.1 Synopsis

This chapter describes the intended goal of the thesis and connects that goal to the general literature. Section 1.2 describes the context of sense-and-avoid and its application, while Section 1.3 provides specifications. Section 1.4 illustrates the motivation for pursuing sense-and-avoid along with the state-of-the-art in current literature. In Sections 1.5 and 1.6, active and passive optical systems respectively, are discussed in the context of sense-and-avoid applications.

1.2 General Discussion

Maintaining separation between aircraft in congested airspace is a difficult problem. Current solutions rely on an infrastructure of pre-established routes and standardized procedures. Humans are essential to the process, as they are excellent at making decisions, can evaluate changing information, and are able to compensate for

deviations from standard operating procedures. Despite their benefits, humans are still prone to operational error and can succumb to physiological stresses, prompting the consideration of automated decision-support mechanisms.

Removal of humans from the aircraft they pilot can simplify the design of such aircraft. Unmanned aerial systems (UAS) are popular due to the potential for longer flights, lower fuel and maintenance costs, and enhanced personnel safety. Life-support systems, cabin spaces and human cargo requirements can be replaced by additional fuel, payload and high(er)-performance engines. Flight envelopes of typical aircraft, normally restricted from human physiological limitations, may be enhanced. In-production UAS are capable of non-stop flights in excess of 20 hours. Eliminating the cockpit area, control inceptors and associated linkages, ejection mechanisms, and life-support systems drastically reduces the complexity of the aircraft, which in turn, reduces cost. UAS are capable of fielding a comprehensive sensor suite due to space savings, and can be outfitted with hazardous payloads (ultra high-power, radioactive) not normally possible with manned flight.

1.3 Definition of sense-and-avoid (SAA)

The "sense-and-avoid" (SAA) paradigm, suggested by the Federal Aviation Administration (FAA), requires that aircraft, manned and unmanned, maintain separation from and avoid collisions with other aircraft. While easily solved in manned sys-

tems, the SAA problem is quite challenging for unmanned aerial systems.

A candidate SAA system can be decomposed into five general functions: (1) Detect conflicting aircraft, (2) Determine right of way, (3) Analyze flight paths, (4) manoeuvre to avoid, and (5) Communicate. The present work is solely concerned with the sensing aspect, which entails the timely detection of conflicting traffic. The fundamental elements of this function include searching for potential threats, rejecting clutter, tracking detected aircraft and the evaluation of closure rates of detected traffic.

This thesis deals explicitly and exclusively with the scenario where two aircraft may end up on a collision course strictly as a function of their respective flight plans. Obstacle collision avoidance, which entails avoidance of static scene elements, either natural or man-made, and avoiding deliberate collisions, when the collision is desired by either aircraft or high-velocity objects (missiles), are not considered. The models developed, and the assumptions that follow, are intended to be consistent with and valid under this proposed scope. The similarity of elements of this scenario with these other situations is outside the scope of this thesis.

1.4 Background and Motivation

North American and European regulatory agencies [ER04] [F2404] [Nor01] stipulate that UAS must demonstrate a record of safety and reliability similar to that for

manned aircraft in order to operate without restriction in civil airspace. Regulations state that regardless of the airspace class or separation services provided by air traffic control, pilots (operators) are required to "see and avoid" other aircraft. U.S. National Transportation Safety Board statistics show that the majority of mid-air collisions occur under the following conditions:

1. Visual flight rule (VFR) meteorology
2. Daytime flights
3. Student or amateur pilots with or without a certified instructor on board
4. Within the traffic pattern
5. Below 3000 ft above-ground-level (AGL)

The highest mid-air collision risk is in unrestricted airspace, where low-flying, inexperienced, "recreational" pilots fly without the aid of air traffic control. Ultra-light aircraft and balloons are particularly risky, owing to their limited visual/radar profiles and lack of transponders.

The importance of the collision avoidance problem is well-described in survey papers on the subject [KY00] [AR09]. Literature labels them as "conflict detection and resolution" (CDR) systems, since their job is to predict an impending conflict, communicate the detected conflict to a human operator and, in some cases, assist in the resolution of the conflict.

Active and passive systems exist for the detection of traffic conflicts, and can be further classified into cooperative and non-cooperative paradigms. A "cooperative"

system relies on transmissions between a host platform and target aircraft, whereas a "non-cooperative" system does not. Table 1.1 presents example technologies for aircraft detection, with discussions presented in subsequent sections.

Table 1.1: Representative collision detection instruments

	Cooperative	**Non-cooperative**
Active	TCAS, ADS-B	Airborne Radar, LIDAR, Sonar
Passive	-	Thermal, Optical, Acoustic

Integrated approaches, based on two or more solutions sampled across active and passive domains, are increasing in popularity. Systems exploring the sensor fusion paradigm include IR/LIDAR, EO/IR [FMAR08] and EO/LIDAR [NP12]. Integrated designs can offer the best of both worlds, or the worst, depending on the implementation. Various engineering challenges specific to the sensing methodology manifest themselves.

1.5 Active systems

Active systems have cooperative and non-cooperative variants, as described in Table 1.1. Cooperative systems monitor a section of sky around the UAS by interrogating transponders on other aircraft. The key advantages are the availability of range

and bearing information, as well as reasonable functionality in both visual and instrument meteorological conditions (VMC and IMC). These instruments provide the ideal level of safety; however every airborne vehicle in vicinity must carry one for the system to be effective. This constraint may not be practical or cost-effective for light air vehicles such as single-engine, fixed-wing hobby aircraft, gliders or balloons.

Active non-cooperative instruments rely on a sensor-actuator principle, where a periodically emitting signal is tracked by a sensor mounted on the UAS, with emission and detection constrained within the same package. The return typically provides range, bearing and closure rate, allowing for avoidance prioritization of oncoming traffic. Depending on the sensor used, these systems may work in both visual and instrument meteorological conditions, but may be cost-effective since an onboard interrogator is not required. Limitations include a much larger power budget due to constant operation, installation costs, mass and lack of stealth.

1.5.1 Identification Friend-or-Foe (IFF)

Most commercial systems in production and use fall under the IFF banner. This is an active, cooperative system, relying on an infrastructure of transponders and towers to coordinate air traffic.

Current systems include the Traffic-alert and Collision Avoidance Systems (TCAS)

[RTC97], Airborne Information for Lateral Spacing (AILS) [HS99][AE01], Enhanced Ground Proximity Warning System (EGPWS) for controlled flight in near-ground situations [Bat99] [BKH+99], Precision Runway Monitor (PRM) for monitoring close parallel flight approaches [RS91], TCAS-based Traffic and Collision Alert Device (TCAD), and the User Request Evaluation Tool (URET) [BLM97]. The Automatic Dependent Surveillance Broadcast (ADS-B) [Ves08], a next-generation analog of TCAS, is certified for commercial use in the United States. The FAA has mandated that by 2020, greater than 80% of the commercial air traffic in the United States must have an operational ADS-B system onboard.

IFF systems are excellent for the collision sensing problem. Principal limitations to the IFF paradigm lie in cost and regulation: Every aircraft requires one for the system to work. For light, general-aviation aircraft, the cost of certification, installation and maintenance can be excessive.

1.5.2 Microwave radar

Microwave radar is a mature technology, and has been employed successfully for many years in the detection of aircraft and obstacles in manned aviation. The microwave designation covers radio frequencies between 300 MHz and 300 GHz, with the higher frequency range resulting in wavelengths on the order of millimetres. Radar systems exploiting the high end of the frequency spectrum are a recent

addition to the microwave family, and are referred to as millimetre wave (mm-wave) radar. Active mm-wave radar features sufficient range, precipitation, cloud and fog penetration, and bearing/range information on potential targets [Wol04].

Current state-of-the-art SAA [BBP05] radar meets visual flight rule specifications and possesses attractive power and mass characteristics. Drawbacks include limited resolution and scan rate, poor noise performance, and power clutter suppression, although the latter can be mitigated by sophisticated signal-processing techniques. The largest barrier to detection is atmospheric absorption, due to the presence of compounds such as water, carbon dioxide and oxygen, whose absorption band is strongest in the millimetre wave region. Effective radar technology employs multiple bands where absorption is limited (35, 95, 140 and 220 GHz are common).

1.5.3 Airborne laser radar

Light Detection and Ranging (LIDAR), also commonly referred to as 'laser radar', [Str94] is an active technology relying on pulsed coherent light. Distance is gauged by measuring the reflected pulse off of a target of interest. Multiple pulses can be utilized to calculate speed. Beam divergence is a critical issue for small targets (1-10 m) at long ranges (5-10 km). Covering the desired angular field requires the use of scanning technology, increasing the instrument's complexity, bulk and potentially reducing precision.

Use of LIDAR was first envisaged in [KLOS06] for SAA applications, but the subject has received precious little attention since. LIDAR systems are difficult to implement in practice, as they suffer from drawbacks, including poor weather penetration, low signal-to-noise ratio, susceptibility to platform stability and potential (temporary) blinding of manned aircraft.

1.6 Passive sensing

Passive systems are only available as a non-cooperative variant. A passive SAA system consists of a sensor which detects impending collisions strictly from differences between the scene and intrinsic target emissions. This scenario is most analogous to human vision. The fundamental advantages of this technique are its low cost, low mass, low power, and the ability to detect non-transponder-equipped traffic. Prime disadvantages include the inability to obtain direct range or closure information and poor performance in IMC. Targets close to the sun position may be difficult to detect. Additionally, depending on the implementation, this method may consume a high amount of data link bandwidth.

1.6.1 Passive optical imaging systems

'Passive' and 'optical' require precise definitions. An optical sensor responds to 'light,' which is taken here to cover electromagnetic radiation in the spectral band

from the ultraviolet (UV) through the infrared (IR), up to, and including the thermal region, at ~ 20 μm in wavelength. Such a sensor uses the photoelectric effect: A photon with sufficient energy colliding with the sensing region in a semiconductor may generate a mobile electron within the substrate [Jan01]. The "passive" nature of system means that the instrument does not source the illumination, but relies on self-emitted or reflected light from the scene. The term 'camera' is defined as a light imaging system, where optics gather light and a semiconductor photo-diode (or equivalent detector) converts the gathered light into electrical charge. Deviations from the standard require a prefix, such as an IR camera, or an image-intensifier camera/night-vision device (NVD).

1.6.2 Visible, near-IR sensing

Imaging in the visible band is typically from $400 - 850$ nm, often referred to as the 'vis-near-IR' range. Typical consumer-grade cameras will have an IR cutoff filter attenuating wavelengths greater than about 700 nm. The sensors are constructed in silicon using standard fabrication processes.

Multi-megapixel image sensors have the following benefits:

- Light, low-power and cost effective, with good responsivity and excellent angular resolution [KLOS06].

- Plastic aspheric optics [McG10] have drastically reduced mass and cost footprints, making image sensors an attractive option for small to medium UAS.
- Acceptable frame rate (> 10 Hz).
- Clusters of sensors can be assembled to selectively sample the desired (large) field of view. These clusters can be structured for selective or over-sampling in spatial, temporal and spectral domains.
- Excellent clutter suppression.

Resolution and noise requirements for an SAA system are exceeded by available technology [Gri05]. Use of a camera system to solve the sense-and-avoid problem is well documented in literature [CDK+02], with trials in [MUDT05] [MUD+07], with a detection range of ~ 6 km demonstrated in [DGSD09]. A prototype optical system for unmanned SAA has been tested by the US Air Force Research Lab (AFRL) [MUD+07].

However, there are shortcomings of visible light sensors, such as:

- Susceptibility to weather and atmospheric effects.
- Orientation with respect to the sun non-trivially impacts the observed signal, since direct sunlight can saturate or damage the sensor.
- Scattered light limits the maximum range available to the sensor and can mask a target.
- Range must be inferred from spatial and temporal computer vision techniques.

- Rolling shutter artifacts from platform vibration are noticeable in consumer-grade CMOS sensors.

Ameliorating these issues determines the feasibility of the instrument, and is the primary focus of this work. Image processing plays a powerful role in this regard, the impact of which is assessed in detail.

1.6.3 Infrared sensing

Infrared (IR) is a term that needs clarification. A useful spectral operating range for 'infrared' sensors starts at about 1 μm and extends to $\sim$ 20 μm in wavelength. In the short-wave infrared region (SWIR) spectral region ($1 - 2$ μm), photo-detection can have properties similar to the more familiar silicon devices used in the visible spectrum. Also of interest is the long-wave infrared region (LWIR), from about $5 - 15$ μm, because much of the radiant intensity self-emitted from objects at room temperature falls within this range. Extensive research on target detection and recognition has been conducted, and is currently on-going, by the military for forward-looking infrared (FLIR) systems [Bha86], mounted on military land, sea and air vehicles.

IR sensors possess unique advantages: They are less susceptible to atmospheric scattering, but generally have strong atmospheric spectral absorption bands. Spectral discrimination of targets is a popular use for IR sensors, as many objects have

strong infrared spectral signatures [WZ89]. In the thermal infrared, ambient illumination is not required for detection, and the atmospheric path-length is halved.

However, IR sensors are not without caveats. Some disadvantages include higher cost relative to visible light sensors, both in sensor fabrication and imaging optics, difficulty in achieving adequate resolution (due to diffraction), and the need for cooling to reduce noise. Although infrared sensors are not considered further in this thesis, some of the methodology discussed in later chapters for visible light is transferable to this domain.

1.7 Contributions

The primary contributions of this dissertation to the state-of-the-art may be summarized by the following points:

- Utilization of a smart camera array to address the aircraft collision-sensing problem. Demonstration, via a custom prototype sensor, that a purely passive multi-camera electro-optical instrument is sufficient, within suitable constraints, to solve the collision detection aspect of SAA.
- Identifying the importance of the horizon for the localization of likely targets. Establishing and confirming the theory that likely targets are localized near the horizon.
- Identifying that the observed signal is dominated by atmosphere, target mo-

tion and platform motion. A methodology for measuring SNR that is robust against target and platform motion was developed. Evaluation against multiple datasets was conducted and sensitivity analysis performed.

- Developing a horizon contour extraction technique from a typical cluttered aerial sequence using machine learning techniques. Useful for measuring or mitigating platform motion in an IMU-denied aircraft/UAS.
- Developing a pre-processing pipeline modeled on fast, low-level algorithms for target detection, which can be employed on low-power embedded systems for real-time target extraction.
- Development of other empirical figures of merit for the evaluation of passive, optical sense-and-avoid instruments. The approach is unique, agnostic to the instrument parameters utilized, and strongly supported by experiments. The measurement, evaluation and sensitivity analysis of the chief figures of merit is also provided.

Other efforts which contribute to the general field of image sensors, unmanned systems and computer vision include:

- Design, architecture and development of both hardware and software for the laboratory, ground and flight-test prototypes. This includes development of a custom in-house, MATLAB-based ray-tracing model to evaluate and constrain the choice of typical parameters and trade-offs.

- Design and system-engineering contributions across the evolution of prototypes, including choice of sensor (for cost and development ease), choice of lens (cost), choice of computing environment (mass, power, volume), choice of development environments (C#, C++, C, ASM) and operating systems (Windows, Linux). Also developed was a framework for the evaluation of image processing options.

1.8 Conclusion

In summary, the Federal Aviation Administration (FAA) has recently approved the use of unmanned aerial systems in commercial airspace, increasing the demand for sense-and-avoid systems necessary to guarantee a level of safety equivalent to manned flight. Numerous options exist to fill the gap, and passive, non-cooperative alternatives have a role by providing acceptable performance in resolution, sensitivity and frame-rate with low mass, low power, low clutter and low cost. Contributions of this thesis include the development and rigorous evaluation of a purely passive collision sensing instrument.

1.9 Organization

The remainder of the thesis is organized as follows: Chapter 2 defines the sense-and-avoid problem in detail, describing the requirements and design constraints. It is followed by Chapter 3, which gives a signal model governing detection in a passive optical sensing instrument. Meanwhile, Chapter 4 describes the instrument design, and its multiple incarnations, in view of the system specifications in Chapter 2 and scientific/engineering challenges faced by similar wide-scale smart camera array designs. Chapter 5 details the laboratory models and ground testing methodology, including the analysis of selected tests. Meanwhile, Chapter 6 presents flight tests conducted at the National Research Council's Flight Test Facility in Ottawa, with a detailed analysis on the raw data collected through these flight trials. In Chapter 7, a pre-processing pipeline is presented and evaluated on captured imagery utilizing representative techniques. Finally, Chapter 8 summarizes the research arc by highlighting key results and sets the tone for future work.

2 Problem Statement

2.1 Synopsis

This chapter defines the type, space and scope of the problem. Topics covered here include the collision detection paradigm, the general scenario and operating environment, core technologies such as smart cameras, cellphone image sensors and embedded computing, constraints on the design based on suggested requirements and system engineering trade-offs, and the test methodology. The chapter culminates with a table of end-point requirements for a commercial instrument, acknowledging that any prototype developed may only represent a sub-set of the identified requirements.

2.2 Introduction

A definition of the problem-space is necessary to frame a solution for the sense-and-avoid (SAA) problem. As stated earlier, we concern ourselves only with the

sensing aspect of the sense-and-avoid paradigm. In this section we will develop the rationale for a passive, optical paradigm, driven primarily by the excellent angular resolution offered in a light, inexpensive package. Development of any sensing instrument must be preceded by a discussion of the scenario and requirements under which that instrument is required to function, culminating in a list of desired specifications. The actual implementation may include compromises due to flight safety considerations and (practical) engineering constraints on, for instance, mass, cost and time.

2.3 Collision detection paradigm

We are interested in a solution to the collision detection aspect of the sense-and-avoid problem. The solution is to be constrained along with the operating environment, sensing and platform degrees of freedom, limiting the possibilities to a tractable sub-set.

The operating environment, or scenario, is restricted to general aviation aircraft operating in visual meteorological conditions (VMC) and along pre-planned flight paths in the absence of inclement weather conditions. Similarly, SAA is constrained to sensing alone, passive sensing in particular. A surrogate is substituted in place of a proper unmanned aerial system (UAS). Implementation constraints of mass, power and volume are discussed, but circumvented by using a large aircraft to act

as the UAS surrogate and restricting the field-of-view appropriately. Note that the design decisions are not limited to the constraints, and the proposed design demonstrates a clear path to miniaturization. Any desired analysis is conducted offline to maximize data collection during flights, as flight tests are a limited, precious resource.

Once the scenario is adequately developed, an instrument concept is introduced. This concept is based on a paradigm that is both adequate for solving the scenario-specific problem, and scalable to the general sense-and-avoid problem.

2.3.1 Rationale

Given the prior general framework, this dissertation attempts to answer the following questions:

- Can a passive optical sensor array deliver adequate performance to solve the sensing aspect of airborne sense-and-avoid? If so, under what limitations must it function?
- Can we identify robust figures of merit for evaluating present and future passive sense-and-avoid systems?

The airborne sense-and-avoid problem entails the following key requirements: large field-of-view, high-resolution, adequate frame-rate, and localized image processing. Such constraints map well to a distributed smart-camera approach. A

production-level instrument may utilize hundreds of cameras to satisfy the resolution and field-of-view requirements inherent to the problem. Therefore, the cameras must be smart, light and low-power, and any processing methodology must be hierarchical and scalable.

2.3.2 General scenario

The passive collision sensing instrument is expected to operate in day-time conditions, at general aviation altitudes, and under VMC. Fig. 2.1 illustrates the scenario of interest. In it are shown a UAS host platform, which houses the sensor, and one or more targets, which may be in a collision course. Both the target and the host platform are expected to be in level, non-evasive flights. Also, it is assumed that a collision is not the desired outcome for either aircraft. Both target and host aircraft are expected to be operating below the cloud ceiling for the duration of their flight paths.

The background scene conditions are assumed to be as follows: visibility is high, in excess of 10 km. The principal source of radiance is a combination of direct and ambient solar illumination, which is typically $\sim 10^2$ W/m^2 as measured on the Earth's surface on a sunny day. Incident radiance on the detector plane from the target is solely a function of the reflected solar radiance. Specular reflections and polarization effects are included, but not explicitly modeled. The cloud ceiling

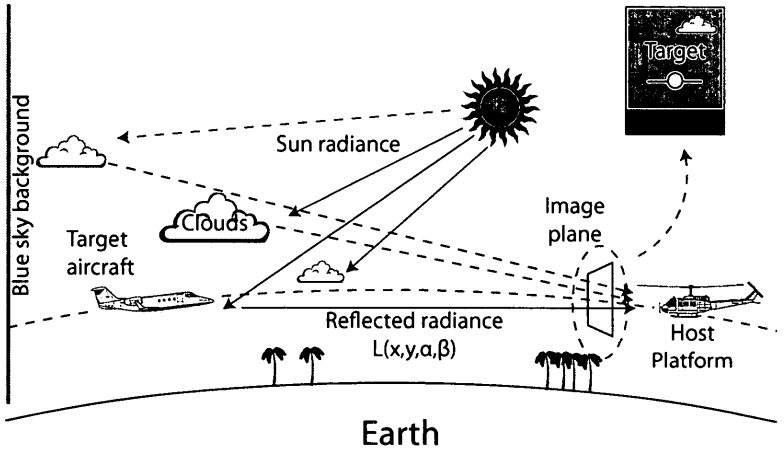


Figure 2.1: General operational scenario for the instrument.

is expected to be high, well in excess of typical general aviation altitudes. Solar or scene obscurations are not considered, while terrain and man-made features are expected to be minimal at the desired platform altitude during level-flight operations.

2.3.3 Instrument concept

The instrument is conceptualized as a distributed, smart camera array assembled from commercial off-the-shelf components. Collision detection via a distributed camera system falls under the broad class of intelligent visual surveillance systems (IVSS) [KCY+10]. Where processing is localized, the scheme can be referred to as the Embedded Distributed Smart Cameras (EDSC) [She12]. The DragonflEYE instrument is designed around this paradigm, and as such, consists of a homogeneous

network of physically identical smart nodes. Sampling configurations may vary in density, uniformity, and selectivity. The configuration may be done in hardware, or given an appropriate instrument topology, dynamically in software.

Such a system could theoretically encompass hundreds of cameras in a small form-factor, such as the AWARE-2/ARGUS-IS by DARPA Research [BGS+12]. Processing and bandwidth constraints become important. However, giving individual cameras local 'intelligence' can drastically simplify bandwidth constraints. Incorporating local discrete processing elements is the next step towards simplifying the data mining problem emerging from such 'gigapixel' imagery.

The use of a network of cameras serves to solve the optical problem of field-of-view coverage. The optics must span an extensive non-spherical field of view while retaining a small, uniform point-spread function (PSF), which is difficult and expensive to achieve with single-element imaging systems. In addition, parts of the viewing frustum that include the sun must be blocked to reduce stray/scattered light and prevent damage to the detectors.

2.4 Constraints

2.4.1 Minimum miss distance and angular coverage

The United States FAA suggests an instrumental field-of-view coverage spanning $\pm 110° \times \pm 15°$ in azimuth and elevation respectively [Fed83]. Furthermore, a "near mid-air collision" is deemed to occur "as a result of proximity of less than 500 feet to another aircraft." To avoid near-collisions, a minimum miss distance of 500 ft (150 m) is the objective of manned and unmanned SAA systems. Note that these near-miss values are distinct from typical horizontal and vertical separation requirements in commercial flight, and as such, do not account for hazardous conditions engendered by the proximity of two aircraft.

Determining the minimum collision distance is subject to the size and type of aircraft, as larger aircraft generate significant wake turbulence [FAA04]. For instance, a small UAS within 152 m of a Boeing 747 ($\sim$ 60 m wing-span [Boe11]) is likely to encounter significant wake turbulence from wingtip vortices and/or jetwash, tasking the flight envelope, communications, reliability of potentially sensitive onboard instruments (gyroscope, magnetometer) and stability of autopilot flight control loops [ER95]. Nonetheless, for small aircraft with wing-spans of 20 m or less, a minimum miss distance of 152 m is acceptable, and can be utilized to drive design decisions.

2.4.2 Time-to-evade metric

The "time to evade" metric (t_{evade}) encapsulates the entire time from detection of a target to the completion of a successful avoidance manoeuvre. The period can be defined as a sum of individual times (Eq. 2.1). The inverse corollary can be referred to as the minimum detection range, R_{ms} for a desired miss distance.

$$t_{evade} = t_{sight} + t_{rec} + t_{coll} + t_{dec} + t_{lagmus} + t_{lagair} + t_{man} \tag{2.1}$$

FAA Advisory Circular 90-48C [Fed83] specifies a t_{evade} of 12.5 seconds, broken down in Table 2.1. Although the requirement exists for manned aircraft, any unmanned aircraft must provide an equivalent level of safety, and is therefore subject to the same.

Table 2.1: Description of individual times contributing to total available evade time [Fed83]

Type	Time [s]	Description
t_{sight}	0.1	Detect the object
t_{rec}	1.0	Recognize it as an aircraft
t_{coll}	5.0	Determine if it is on a collision course
t_{dec}	4.0	Decide on a manoeuvre
t_{lagmus}	0.4	Neuromuscular latency
t_{lagair}	0.2	Aircraft response latency
t_{man}	1.8	Time to complete an avoidance manoeuvre

Although an avoidance instrument has the entire period to perform its operation, a detection instrument must reach a decision as per $t_{det} \simeq t_{sight} + t_{rec} + t_{coll} -$

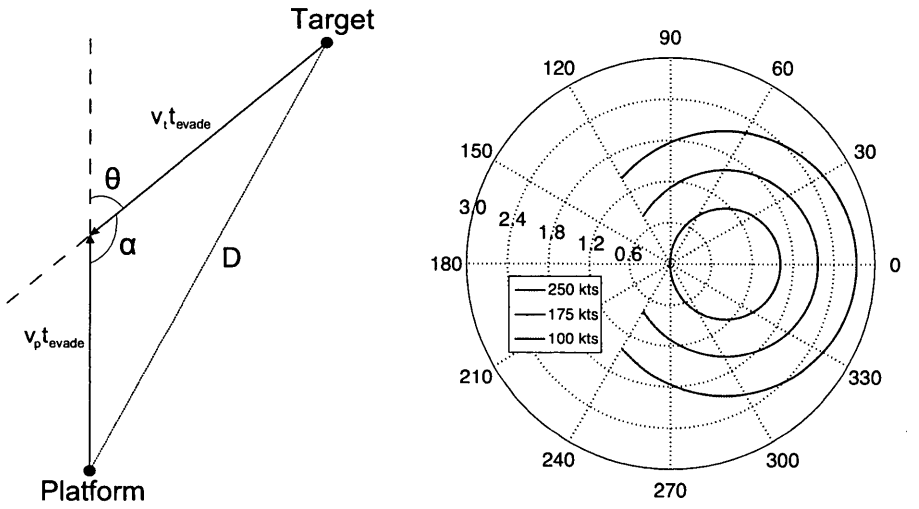
$t_{info} = 5.5$ s, where $t_{info} = 0.5$ s is the amount of time the pilot will require to process the information.

Note that Table 2.1 represents aggregate values. In particular, t_{man} is stated for general aviation aircraft, and can vary dramatically with aircraft size and type. Also implicit in the scenario is the assumption that both target and host platform are similar sized aircraft, traveling at similar airspeeds. Subsequent analysis in the document is conducted against a t_{evade} of 15 s to provide an additional level of safety.

2.4.3 Collision geometry and sensor field-of-view

The definition of a collision can be built from the afore-mentioned FAA guidelines. Fig. 2.2a depicts a vector diagram of two aircraft on a collision course. Here, the target and host ground velocities are indicated by V_i and V_o respectively. Similarly, θ, is the azimuthal offset of the target in the host reference frame. Assuming non-manoeuvring and non-accelerating flight and an absence of winds, the host and target aircraft will collide at a time of t_{evade} seconds. The relative course of the target aircraft is given by the angle α.

The separation distance, D, between the aircraft can be calculated based on the input parameters θ, V_i, V_o, and t_{evade}. Fig. 2.2b presents the locus of solutions for a host with a velocity of 100 knots. Each point represents the minimum



(a) Collision-course vector diagram (b) Distance [km] between aircraft for 15 s time-to-collision for a host velocity of 100 knots [MTE+12]

Figure 2.2: Collision-track geometry

detection distance, where $D = R_m$, as a function of horizontal approach angle θ, with $t_{evade} = 15$ s and constant velocities. The plot is limited to values of approach angle in the range of ± 120 degrees, while each connected curve represents a unique target speed, varying between 100 knots and 250 knots. It can be observed that increasing approach angles result in reduced minimum detection distances, as the effective closing velocity is reduced. It is also apparent that the shortest time to impact is encountered by the head-on collision trajectory. Thereby, the head-on case represents the critical, performance-limiting path for any conceived SAA instrument, and the "time-to-impact" and "range-at-first-detection", derived from

constant closing velocity, are two important figures of merit.

In addition, it is noted that the design or selection of an appropriate SAA for a candidate UAS, the reaction time considered must be a function of the UAS's available performance in the anticipated flight regime, as well as regulatory constraints in the operating airspace.

2.4.4 Heading and angular target size

Typical aircraft exhibit a small frontal cross-section in comparison to the lateral cross-section. The collision track, therefore, becomes an important parameter for establishing any minimum resolution requirement. Fig. 2.3 illustrates a polar plot of angular target size (radius) against angular azimuth, for targets at the minimum detection distance. A frontal area of 6.8 m^2 and lateral area of 20.4 m^2 were assumed for the target. These measurements were considered representative of general-aviation-sized aircraft, to which the Bell 206, used in our measurements, is similar. The plot assumes a host speed of 100 knots and a time-to-collision of 15.0 s.

For a target as large as the Bell 206, the required minimum angular resolution to fully subtend the target within one pixel can be calculated (Eq. 2.5) to be ~ 0.2 mrad. Note that the required distance for a fixed t_{evade} to a potential collision changes with azimuthal offset. Note that the head-on collision case will dominate

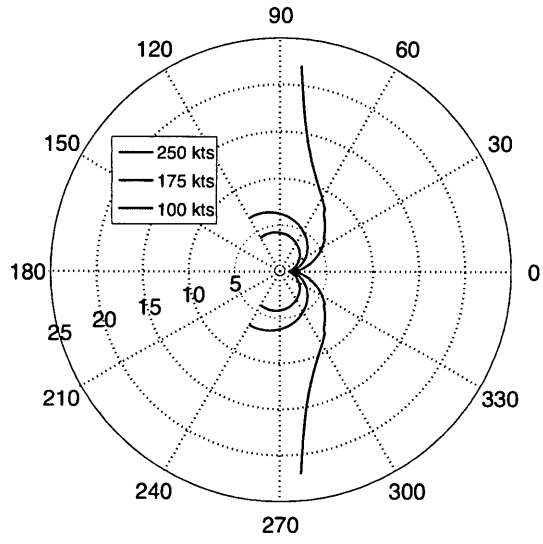


Figure 2.3: Angular target size [mrad] for a 15 s time-to-impact [MTE+12]

the resolution requirements. Furthermore, asymmetries in the frontal and lateral cross-sections imply a discontinuous scaling with increasing azimuthal offset.

2.4.5 Angular resolution and total field coverage

The effective viewing frustum (or field-of-regard) and angular resolution are linked inversely, therefore, optimization of one occurs at the detriment of the other. The total (instrument) field-of-regard can be expressed as per Eq. 2.2,

$$T_{FOV} = \sum_{i=1}^{N} FOV_i - \sum_{i=1}^{N-1} O_{i,i+1} \tag{2.2}$$

where N denotes the number of cameras, FOV_i is the field-of-view for camera

i, while $O_{i,i+1}$ defines the overlap between adjacent cameras i and $i+1$. Let us consider $w_i = FOV_i$ to denote the angular extent of the field-of-view of one camera in one dimension. Then w can be expressed as a function of the lens focal length, f, and sensor orthogonal dimension, d, such that $w_i = 2\arctan\left(\frac{d_i}{2f_i}\right)$, which may be generalized to Eq. 2.3 for identical cameras. Eq. 2.2 similarly generalizes to Eq. 2.4.

$$w = 2\arctan\left(\frac{d}{2f}\right) \tag{2.3}$$

$$T_{FOV} \approx NFOV_i - (N-1)O_{i,i+1} \tag{2.4}$$

Typical targets are expected to have a cross-section on the order of 1–10 metres. Since Z is expected to be > 1 km, which is much larger than most focal length values, the lens can be set to the hyperfocal limit, maximizing the depth of field [Hec02]. Note that typical off-the-shelf webcams have a pixel pitch (d_{pix}) ranging from 6.0 μm to 1.2 μm. The required focal length, f, can be approximated within the thin lens limit as Eq. 2.5,

$$f \approx Z\frac{d_{pix}}{d_{foot}} \tag{2.5}$$

where d_{pix} is the physical size of the pixel in the detector and d_{foot} is the corresponding pixel footprint at a range Z.

Consider the following hypothetical derivation for determining the desired focal length, and subsequently, the number of cameras required to cover a horizontal field of view of 120 degrees. Assuming a target dimension of 1.0 m at a range of

$Z = 10$ km, let the pixel footprint be filled by the target, such that $d_{foot} = 1.0$ m. Furthermore, let us assume a pixel-pitch of $d_{pix} \simeq 2.5$ μm and a sensor resolution of 1600×1200 pixels, both of which are typical values for most commercially available high-definition webcams. Finally, let us assume an overlap factor of 10% between cameras.

Given the values for d_{foot}, Z and d_{pix}, one can calculate a focal length of $f = 25.0$ mm (Eq. 2.5). Similarly, given f and the sensor resolution, one can compute the FOV per camera to be $w = 0.16$ rad or 9.2 degrees. The overlap of 10% now corresponds to an angular overlap of $O = 0.9$ degrees. Utilizing Eq. 2.4, one can calculate $N \simeq 15$, which is the number of cameras required to fill the desired horizontal coverage.

2.5 Smart camera

The addition of image processing to a camera, to create a "smart" camera, adds to the mass, cost and power consumption of the node, while improving the ability of the instrument to detect targets.

Fig. 2.4 illustrates a smart camera module. A smart camera is an integrated unit comprised of three major hardware components: An image sensor equipped with an optical mount, an integrated processor, and a standard communications interface. Modern commercial image sensors support on-board auto focus, auto

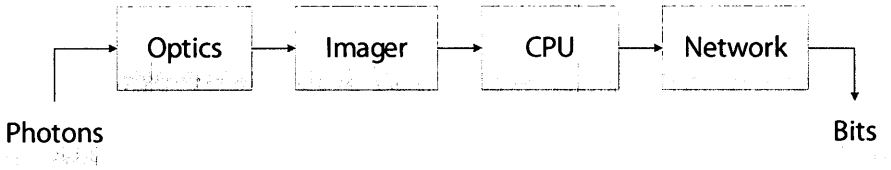


Figure 2.4: Block diagram of a smart camera

exposure and auto white-balance. Optics can be micro-machined plastic aspherical lenses [TC07], or even constructed as diffractive elements from surface metallic layers [Tho11]. For convenience, the optics utilized in the developed prototypes are commercial off-the-shelf C-mount lenses. This choice was made for flexibility in testing multiple focal lengths and ease of availability.

2.5.1 History

Smart cameras have existed for at least a decade [RW08] [RWS+08] [Rat10]. Advances in CMOS technology continually increase transistor density for a given die size, allowing multiple electrically compatible components to be packaged onto the same die. The advent of system-on-chip (SoC) architectures, such as the OMAP utilized on the prototype camera nodes, provides an intermediate step in allowing discrete packaged components (CPU, SSD, RAM) to be sandwiched into a unified package [Tex11]. In the future, smart cameras will come integrated with low-power general purpose processors within the same IC, drastically reducing mass and cost.

Modern designs for smart cameras include versions with high-bandwidth interfaces [NBPL08], embedded DSP architecture [KRR09], custom embedded parallel-processing [KASD07] and low-bandwidth implementations [CJH05] [CAB+08] [SSS+03] for wireless and ad-hoc networks. Applications may include remote sensing, target detection, traffic monitoring, people tracking and fire-fighting.

2.5.2 Cellphone cameras

The explosion of small light-weight, inexpensive image sensors developed on the APS CMOS technology is a primary driver for the timeliness of a passive collision sensing instrument such as DragonflEYE. Multi-megapixel CMOS image sensors are readily available for purchase, where cell-phone cameras, in particular, offer excellent resolution and good responsivity in a compact and power-efficient package.

The industry continues to grow unabated, with micro-lens to improve fill factor [Rho99][ABT03], back-side illumination technology offering increased quantum efficiency [PCN+05][NDF00][MST08] as well as $> 99\%$ fill factor[SBB+05], and plastic aspheres for compact, high-resolution lenses [TC07]. These new advancements render image sensors an attractive option in systems where mass, power and cost are constraints.

In the collision detection paradigm, the use of a network of smart cameras serves to address the optical problem of field-of-view coverage, as the optics must span an

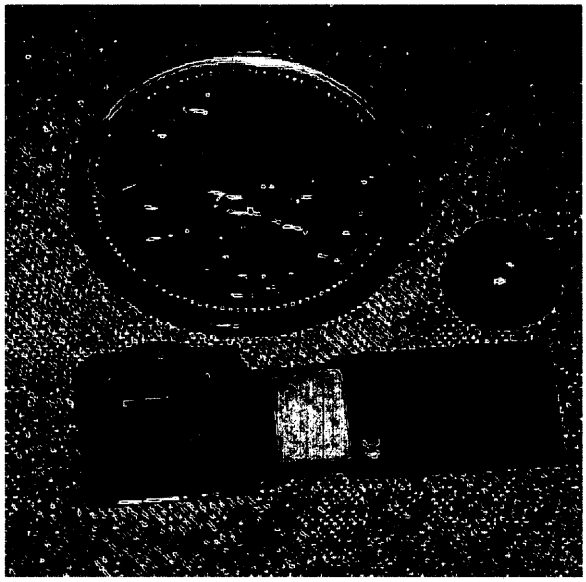


Figure 2.5: Snapshot of a Micron 0.6 MP cellphone camera [She12]

extensive field-of-view while retaining a small, uniform point-spread function (PSF), which is difficult and expensive to achieve with single-element imaging systems. In addition, parts of the viewing frustum that include the sun must be blocked to reduce stray/scattered light and prevent damage to the detectors.

2.5.3 The 'compression ratio' figure of merit

Effectiveness of a processing technique in a constrained bandwidth scenario can be judged by the 'compression ratio', which defines the amount of output data produced for the given input data. In essence, the compression ratio highlights the benefit of a hierarchical, or distributed, processing framework over a more tradi-

tional, centralized paradigm. Note that any raw data stream may be 'compressed' by switching to a more efficient representation, which is typically achievable at the expense of fidelity, or by limiting the storage to 'useful' information. The former refers to traditional compression, while the latter describes processing. Here we limit the discussion to processing, and as such, the remainder of this section focuses on utilizing processing to extract the necessary information from the raw data and transmitting the 'useful' bits.

In a traditional centralized computing model, computational capacity scales linearly with frame rate and quadratically with resolution. Therefore, limits on resolution have a greater impact on performance over reductions in frame-rate or increases in computational capacity. In a distributed architecture, such as a smart camera network, processing scales linearly with the number of nodes, assuming the individual nodes have sufficient computing resources. To illustrate the impact, consider the collision detection scenario:

Let us consider 100 cameras, where each camera has an 8-bit pixel resolution, a total pixel count of 1 megapixel (MP) and a frame-rate of 10 FPS. The data stream from an individual camera is 10 MB/sec, for a total data influx of 1.0×10^9 bytes (1.0 GB/sec) to a central sink. Consider the Fast Fourier Transform (FFT) as the benchmark filter operation, with the output designated as the image coordinate pair with the highest filter response. The FFT has a best-case runtime of $O(Nlog(N))$,

where N is the number of pixels. Assuming a 100% processing efficiency, with no operational overhead, the total computational load is ~ 9 billion FLOPS.

Consider a modern general-purpose CPU containing an equivalent floating-point multiply-accumulate instruction: An dual-core Intel Core i7 operating at a frequency of 2.4 GHz. Assuming optimal configuration and zero memory latency, the CPU can process one scalar add and one scalar multiply operation per cycle per core [Int12], resulting in 4.8×10^9 effective operations per second (EOPs). 100 cameras will fully saturate a 1.0 Gigabit ethernet link (assuming no losses), and the designate CPU will require two seconds to process one second worth of imagery. Such a system scales poorly with N, as doubling the number of cameras ($N = 200$) will require two 1.0 Gigabit ethernet links and additional computing resources to bridge the disparate data streams. Such a system benefits if images are processed locally.

There are restrictions on the types of problem that benefit from local processing. For the SAA scenario, infrequent, small targets are the norm. Initial detection of an "event" takes the preponderance of processing. This task is well suited to distributed processing nodes. Once a detection occurs, image buffering, frame-rate, image resolution and image size can be adjusted to optimize further processing.

2.6 Embedded computing

The low cost and copious availability of smart cameras would not be possible without the advent of embedded computing platforms tailored for multimedia and connectivity. These platforms provide the CPU, memory, multimedia and networking footprint required to handle the bandwidth and processing functionality of a smart camera.

2.6.1 History

Embedded devices are typically low power and possess limited computing resources. In recent years, the exponential growth of consumer-based hand-held devices, including smartphones and tablets, has generated demand for high-performance, compact, power-efficient central processing units (CPUs) with multimedia capabilities. The dominant solution to the problem of high-performance, low-power chips is a heterogeneous processing architecture, or the birth of the Applications Processing Unit (APU).

Relying on new 'Package-on-Package' (PoP) technology [PZ07], the APU architecture consists of a CPU, which may have one or more core, and may or may not include a local floating point unit (FPU). The CPU may be paired with a discrete ARM-based or third-party (PowerVR, S3, Ti) graphics core, as well as a discrete

mixed processing unit (MPU) such as a digital signal processor (DSP) and/or a micro controller. The CPU communicates with the components via a local interconnect bus and a shared memory map. Inter-component communications using direct memory access (DMA) is not commonly supported, and if available, is restricted to high-performance variants. The Texas Instruments Open Multimedia Applications Processor (OMAP), an APU variant envisioned for use in DragonflEYE, is explored in detail.

2.6.2 The OMAP APU Architecture

The OMAP is an APU variant designed by Texas Instruments (Ti). It is a true system-on-chip architecture, combining a CPU, graphics processing unit (GPU), floating point co-processor, volatile and non-volatile storage (random access memory (RAM), Flash) within a single integrated circuit (IC). Popularity for the platform grew exponentially when Ti released a free reference printed circuit board (PCB) design, and the electronics distributor, Digikey, provided a low-cost ($150) development platform labelled the "BeagleBoard" [Bea09] from the reference design. Ti further contributed by releasing their DSP software development kit (SDK) and DSPLink bridge for free on Linux. Variants of the OMAP type architecture power the majority of modern tablets and smartphones in existence, including the venerable iPhone and iPad.

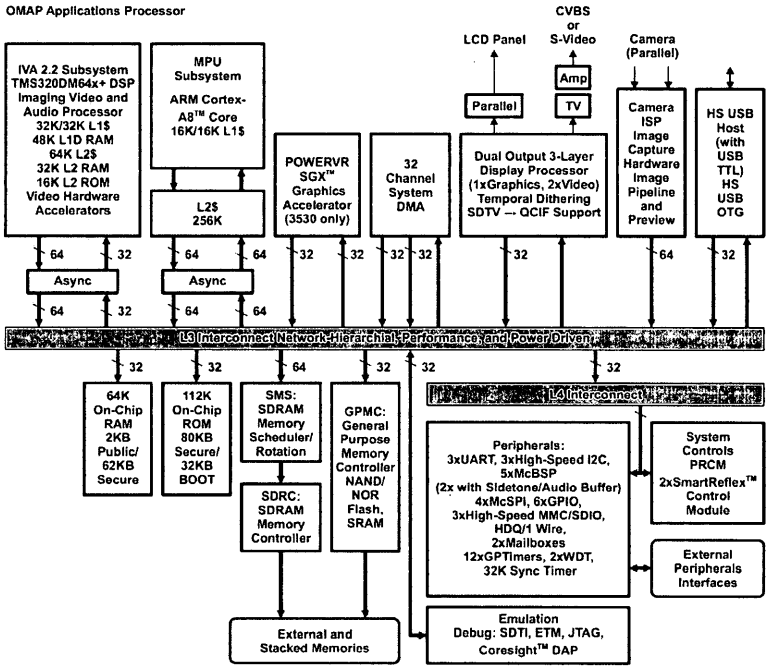


Figure 2.6: System architecture of the Ti OMAP 3530 [Tex11]

The OMAP 3530 comes equipped with a Cortex-A8 CPU, a NEON engine: A 128-bit floating point SIMD (single instruction multiple data) co-processor [ARM13], a TMS320C6400 DSP core and a PowerVR SGX530 GPU. Fig. 2.6 depicts the CPU architecture of the OMAP-3530, with individual processing elements and other peripherals connected across a common interconnect bus. The NEON core supports 16 simultaneous operations, and just like SSE, is limited to single-precision floating-point values, although the ARM implementation is not IEEE-754 compliant. The DSP core, although fixed-point, comes with significant image-processing IP support available for Ti DSPs. Development is simplified as the library handles fixed-point

conversion internally. The PowerVR SGX530, supporting OpenGL ES version 2.0, can be harnessed in a similar fashion via the OpenCL libraries.

The flexibility stemming from common interconnects and PoP architecture allows the OMAP to be tailored to specific applications. This makes the OMAP an extremely versatile and capable processor. However, utilizing maximum functionality typically requires extra programming effort as compared to a general-purpose CPU.

The OMAP utilizes a System-on-chip (SoC) architecture configured in a PoP module format, as illustrated in Fig. 2.7. The lower package is the APU, while the upper package combines the RAM and Flash within one IC. Although the dies in the upper package appear sandwiched together, they are electrically distinct, and interface independently with the APU. The interconnects are BGA (Ball-grid-array), where extra interface pads are provided on the APU package for bus and clock interconnects. Some lines, such as power and ground, may act as direct passthroughs to the PCB (printed circuit board) BGA mount, reducing the power drain on the APU. In summary, the PoP assembly can be considered to be an intermediate paradigm between die integration and discretized PCB connectivity [PZ07].

Table 2.2 highlights the individual processing units sandwiched into a single chip. Preliminary benchmarks indicate that utilization of the SIMD architecture

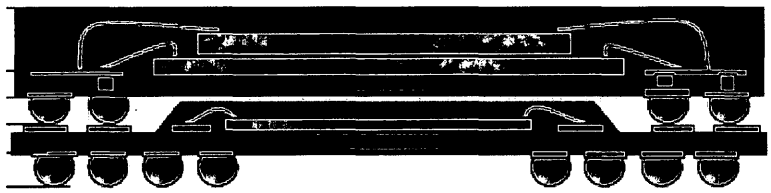


Figure 2.7: Package-on-Package (POP) architecture - BGA mount

results in an order of magnitude improvement in floating-point performance over software emulation. Note that the NEON core is now natively supported in Linux kernel versions beyond 3.6.

Table 2.2: OMAP 3530 Computing Architecture

Core type	Name	Precision	Speed (MHz)	Performance
CPU	ARM Cortex-A8	Fixed-point	600	1200 DMIPS
Media	NEON SIMD	Single	600	N/A
DSP	TMS320C64x+	Fixed-point	430	4000 DMIPS
GPU	PowerVR SGX530	Fixed-point	200	1600 MFLOPS

It is observed that the ARM general purpose CPU only accounts for 17% of the total computing capacity available on the OMAP and 0% of the floating point support. Note that this is considering the best-case scenario, where algorithms are tuned in assembly to near 100% efficiency. The utilization will likely worsen in real scenarios after factoring in high-level support and operating system overhead. Proper utilization of the DSP and NEON cores is therefore essential to derive maximum performance from the architecture.

2.7 Tests

Tests are required to evaluate performance and establish benchmarks for any potential instrument. The tests can be broken up into the following components: modeling, laboratory tests and field tests, which are further decomposed into ground and flight tests. Evaluation of robust figures of merit is a required outcome of the test methodology.

2.7.1 Modeling

A modeling environment is desired to test the efficacy of the basic collision detection paradigm. The model must provide opportunities to test both radiometric and geometric characteristics of the collision detection problem. In particular, behaviour at extreme ranges for likely targets must be estimated. The choice of critical parameters for the instrument, such as lens focal length, sensor resolution and field-of-view coverage, must be tested. Target characteristics such as size, geometry and reflectivity (texture, contrast) may be studied. Tracks generated by combined motion of the target and platform may provide synthetic datasets for the evaluation of image processing algorithms. The background and atmosphere may be simulated independently or in concert, as a static or dynamic scene, with solid colour, or textural components, or mimic realism by utilizing captured imagery for

textural information.

Modeling preceded instrument hardware development, but was also used to analyze the results of laboratory and field tests, as needed.

2.7.2 Laboratory tests

Characterization and calibration of the sensor and the optics is the fundamental purpose of the laboratory tests. For individual cameras, measurements may include the sensor quantum efficiency, spectral responsivity, dark current and the behaviour of digital gain parameters, such as gamma, brightness and contrast. These tests may be used to compare sensors across technologies (CCD vs CMOS for instance) and applications (webcam vs machine vision). Lenses require evaluation for point-spread effects tied to aberrations and distortion. Configurations of sensors and lenses may be considered for measuring mass/cost versus performance ratios. The bulk of these tests were conducted at the Vision Sensor (VISOR) facilities by staff, and are not discussed in detail in this dissertation.

For a prototype sensor array, inter-camera calibration, field-of-view overlap between adjacent cameras and total coverage are parameters whose measured values can be inserted into the system model. Outdoor experiments in a controlled environment can measure atmospheric scattering and absorption effects, which are difficult to simulate accurately and realistically. Chapter 5 details the outdoor

tests conducted for the instrument.

The embedded computing environment local to each smart camera may also be assessed. Key figures of merit include burn-in tests and filter performance evaluation. Chapter 5 presents performance benchmark experiments for the OMAP-3530 architecture, where the convolution of an image with a filter kernel is considered to be the rate-limiting operation. The Fast Fourier Transform (FFT) is an efficient implementation of the convolution operation and was treated as the test operation. The equivalent frames per second (FPS) per power, cost and mass footprints are treated as the figures of merit.

2.7.3 Field tests

The goals of the field tests are threefold: (1) Establish an operating procedure for evaluation, including installation of the instrument, safety considerations, and coordination of the aircraft and the flight crew, (2) Verify sensor capability independently of modeling assumptions (3) Measure important figures of merit for sensor performance in real environments, including, but not limited to, the range at first detection.

The first goal will be handled via ground tests. The sensor is fully instrumented placed onto a stationary, rigid ground (host) platform, while a designated target aircraft performs near-collision fly-overs along pre-selected collision geome-

tries. Given the stationary nature and location of the "host" platform, a collision course implies a descending trajectory for the target aircraft. The in-flight precision of establishing and maintaining a known collision trajectory will also be assessed during these tests, and tests may be added or removed depending on performance. Instrument reliability, validity of acquired imagery and general sensor behaviour under favourable scene conditions would also be assessed.

The second and third goals will be achieved via flight tests. In particular, the performance of the instrument under the specified scenario will be assessed. Collision geometries for the ground and flight tests will encompass head-on collisions and azimuthal offsets, altitude offset head-on flybys, overtakes and descending cases. Ancillary cases, such as perpendicular constant-altitude geometries may also be conducted as time and cost permits. A processing pipeline may be operational and engaged for these tests, if time permits. However, the collection of raw data (imagery and relevant ancillary information) may be prioritized for initial flights, as the processing pipeline can be refined offline.

The detection of potential targets at an adequate range is established as the principal figure of merit, and the desired outcome for these flight tests. The role of the system PSF can be analyzed in this context as it presents the fundamental limit to detectability. Target behaviour, size and growth, reflectivity fluctuations, can also be measured to provide insights into the impact of atmosphere on the sys-

tem PSF. The availability of real targets, the atmosphere, solar variations, ground clutter and other naturally occurring artifacts, should provide a detailed and realistic assessment of DragonflEYE, which is quite difficult to achieve in a modeling or laboratory environment.

2.8 End-point requirements

Prior discussions regarding the scenario, design constraints and test methodology can be condensed to reveal end-point requirements for a final/commercial version of the DragonflEYE instrument. Table 2.3 illustrates these requirements. This table, although comprehensive, represents a wish-list for the final product. Operational, safety and budgetary constraints may result in flight-capable prototypes achieving a subset of the requirements. Nonetheless, any fundamental research and any system engineering activities conducted will utilize a scalable, flexible approach, such that the end-point requirements remain attainable in a relatively straightforward fashion.

2.9 Conclusion

In summary, this chapter defines the requirements for any commercial passive collision-sensing instrument. To that end, a rationale was developed for the passive,

Table 2.3: End-point requirements for a collision detection instrument

Features	Nominal Value	Units
Miss distance	500 (152)	ft (m)
Time-to-evade	15	s
Field-of-regard (azimuth $\times$ elevation)	220×30	degrees
Angular resolution	0.2 ($\simeq 1$)	mrad (pixel)
Overall dimensions	$60 \times 20 \times 10$	cm^3
Mass	< 10	kg
Power	< 100	W
Update rate	10	Hz
Scene Dynamic Range	10^5	-
Cost	10^4	$

optical paradigm, driven primarily by the excellent angular resolution offered by such a solution in a light, inexpensive package. The general scenario is presented, appropriately constrained, followed by a discussion of available technologies. Design constraints were identified and analyzed, culminating in a list of desired specifications. The actual prototypes may include compromises due to flight safety considerations and engineering constraints.

3 Scene Models

3.1 Synopsis

This chapter describes a model for the formation of signals for the collision detection scenario. In particular, the fence geometry is highlighted, along with key figures of merit arising from the fence concept. A practical definition of the 'horizon' is provided, motivated by its importance as an omnipresent scene feature in the defined scenario. Signals at the fence boundary are studied. Target size, position and composition are discussed. An imaging train transfer function model is developed by considering the impact of atmosphere, motion, optics, detector and sun orientation. The system signal-to-noise transfer model is developed from the signal, the transfer function model and all appropriate noise sources. This chapter culminates with a technique for calibrating a large camera array.

3.2 Introduction

Radiometric calculations determine the intensity, direction and composition of light incident to a detector element [BW99]. In the present scenario, the incident irradiance from targets can be represented as a sum of contributions from independent point-like radiant sources, under the following assumptions:

- Sources are assumed incoherent. Intensity at a point is assumed to be a sum of source intensities instead of a sum of amplitudes. Covariant terms, which give rise to interference, are assumed to be zero. In particular, coherent sources such as lasers are excluded.
- Self-emission and fluorescence are ignored.
- An explicit spectral analysis is not presented. Any functions possessing a spectral component are evaluated at a nominal wavelength of 550 nm, unless otherwise noted.

3.3 'Threat' fence

The 'fence' concept represents the effective field-of-view covered by the sensor at the far range of target detection. Fig. 3.1 outlines the fence geometry concept as a large-radius ribbon encircling the platform, where the ribbon-width indicates the elevation coverage. A similar ribbon marks the inner fence boundary. Since target

and signal characteristics are much more prominent at the inner fence, much of the system signal modeling and analysis effort will be directed towards the study of effects at or near the outer fence, with the assumption that stronger signals and larger targets are easier to detect than weaker signals and smaller targets.

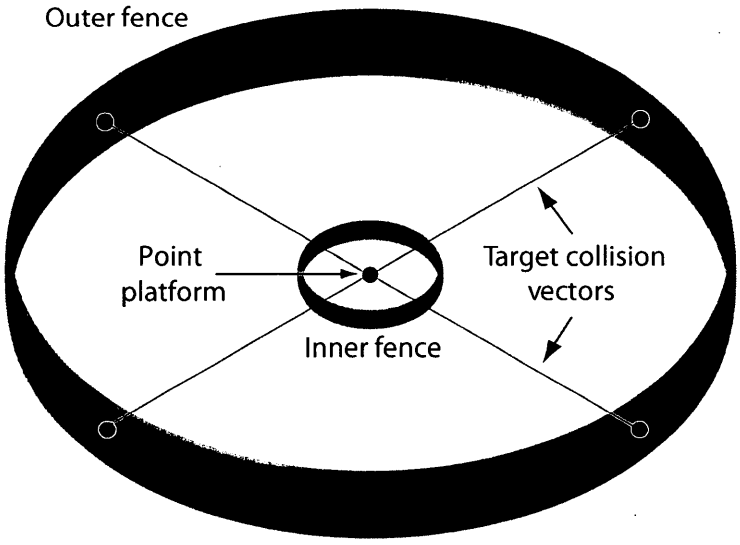


Figure 3.1: Fence penetration by potential targets

Level flight is the normal mode of operation for all general aviation aircraft. Therefore for a collision trajectory to exist, the host and target must be co-altitude. Note that the fence will wander in elevation and azimuth as the orientation of the platform varies about its control settings. This variation is present even though the sensor is rigidly mounted to the platform. Additional vibration-induced variations between sensor and platform can also occur. Likewise, routine course corrections

by the pilot (or autopilot) will affect the absolute fence position, with the position relative to the platform is only affected by vibration.

Similarly-sized targets also appear as quasi-point objects in relation to the platform. By treating the platform as a stationary reference frame, the combined velocity vector can be transferred to the target. Note that the fence is nominally fixed in this reference frame. It can be observed that the angular components of this combined vector are fixed, in both azimuth and elevation. Therefore, one can conclude that any target on a collision course with the platform will maintain a fixed angular position in the platform's field-of-view. In acquired imagery, relative movement for a particular target can be attributed to its current heading. Furthermore, the observed position of the target is highly sensitive to angular perturbations, such as platform rotational motion, whereas the impact of positional deviations is small at the expected fence distance (> 1 km). A direct application of the fence methodology highlights key ranges of detection.

3.3.1 Range at first detection - R_0

3.3.1.1 Motivation

The range at first detection, R_0, is defined as the range at which the target is first acquired. By extension, it indicates the radius (distance) to the outer fence. R_0

is a natural outgrowth of the fence methodology and the principal parameter of interest in this work.

As a figure of merit, R_0 possesses potent benefits. It is intuitive and easy to visualize. Variations in R_0 are sensitive to target, sensor and scene parameters, therefore the behaviour of R_0 can be used to optimize sensor performance. R_0 can also establish limits to useful operations, for example low visibility conditions. Since it is a distance measure, R_0 is independent of host and target velocities, unlike the time-to-evade metric. In addition, R_0 can be derived robustly from the raw image data captured during ground and field tests [MTE+12]. Furthermore, precision in the estimate of R_0 can be greatly enhanced by distributing the error over the whole run. The analysis is described in greater detail in Chapters 5 and 6.

3.3.2 Time to impact - t_0

A direct corollary of the range at first detection is the estimated time-to-collision, t_0. The value for t_0 can be computed directly from R_0 if the host and target velocities are known.

Recall, from Section 2.4.2, that the U.S. FAA suggests a minimum miss (or fly-by) distance of 152 metres, within which the two aircraft in question are considered to have collided. Also recall that the time-to-evade metric, t_{evade} metric, as per suggested regulations, defines the minimum window for detection to occur before

collision is unavoidable. We can, therefore, define t_0 to be the time until the target is within 150 m of the host platform, and compare it to t_{evade}. The comparison reveals two salient characteristics. Firstly, avoidance is possible if $t_0 > t_{evade}$. Second, the difference of t_0 and t_{evade} reveals a "processing time" buffer, such that, the earlier the detection, the larger the time buffer available to the host platform for confirming a potential collision and deciding a corresponding course of action.

The weakness of t_0 lies in the limited knowledge of the target. In the absence of full telemetry, t_0 can only be estimated indirectly from imagery alone. Therefore, R_0 is a much more convenient metric. Nonetheless, estimates of t_0 from secondary metrics, such as target size and growth, can still prove useful as ballpark figures for collision evaluations, and future algorithms may be devised to extract further utility from these parameters.

3.3.3 Human pilot detection range - R_v

An additional metric for comparison and evaluation is the performance of the human pilot under similar conditions. This comparison is motivated by the fact that any system capable of meeting or exceeding human performance technically qualifies as a valid solution, regardless of regulations or stipulations. The mean visual acquisition range, R_v, is the range where a typical pilot is expected to spot an intruder under optimal conditions. Accurate measurements of R_v are challenging

to obtain [GS89], as fluctuations in target irradiance and the cognitive load of the observer dramatically affect performance. It is much harder to measure in flight trials as scene conditions are never identical and each mock-collision run may yield only one sample. Many trials are necessary to obtain a good statistical model for evaluation.

3.4 Horizon

A practical definition of a horizon is the curve of intersection between the visible earth and the sky. Although well-approximated at high altitudes and large distances by an ovoid-Earth model, terrain features can introduce discrepancies at lower altitudes. Assuming flat terrain and small deviations from the level flight scenario, offsets due to terrain features are small compared to the horizon range. In the ovoid earth mode, range to the horizon is directionally dependent, but temporally fixed, for a constant altitude.

3.4.1 Motivation

The horizon is an excellent reference for a network of cameras, since it facilitates calibration across individual fields of view. The extended nature of the horizon suggests that additional scene features are not needed for relative positioning of a target object, therefore a measure of the proximity to the horizon partially ob-

viates the need for platform motion compensation. An arc of the horizon curve is present in all images, and is generally visible under visual flight rules (VFR), although atmospheric scattering can blur the contour. Because the horizon has a simple defining property that is common to all camera nodes, namely a common polar axis, measurement errors can be reduced by comparing the horizon-contour measurements from different camera nodes.

In light of the fence concept, the horizon may appear as a linear projection that bifurcates the fence ribbon. Quantitatively, the ribbon defines a spatial probability density function indexed to the likelihood of a target, and the horizon may be located at its peak value. Combining the fence and the horizon has two distinct advantages: (1) the region of likely targets is now a smoothly varying region-of-interest (ROI) weighting for pixel importance (2) All detected targets may be localized independent of platform motion. Derating, but not eliminating, non-ROI pixels ensures flexibility in detecting strong responses outside the expected region of interest.

3.4.2 Derivation

Fig. 3.2 describes the declination and distance to the horizon as a function of aircraft altitude and pose in an Earth-centered-Earth-fixed (ECEF) coordinate system. The host and target aircraft are in constant-velocity level-flight, at altitudes of A_p and A_t (not shown) respectively, where, given the local radius of the earth, R_e,

$R_p = R_e + A_p$ and $R_t = R_e + A_t$. For convenience, the platform coordinate reference frame is body-centric and the attitude corresponds to Tait-Bryan/Euler angle notation, with the coordinate system (X, Y, Z) defined along the right-wing, nose, and up directions respectively. Therefore, rotation about the X axis corresponds to pitch, Y to roll, and Z to yaw/heading. Note that a similar convention is assumed for the target.

The world coordinates are denoted in a local Cartesian representation, such as the local Universal Transverse Mercator (UTM) projection. Notation between the world coordinate reference frame and the body-axis coordinate reference frame is kept consistent by following the right-hand rule. Furthermore, the body-axis reference frame follows the East, North, Up (ENU) convention, such that east and starboard map to X, north and fore map to Y, and Up and altitude map to Z. The body-axis reference frame is assumed to be rigidly connected to the world reference frame by a Euclidean rotation and translation.

With regards to the horizon, of particular interest are two parameters from the platform's perspective, namely the line-of-sight distance to the horizon, D_p, and θ_p is the angle of the horizon from the platform zenith (not shown). Platform distance to the horizon along the Earth's surface, D_{sp}, may be similarly defined. Under the ovoid Earth assumption, the platform's attitude is the only quantity required to

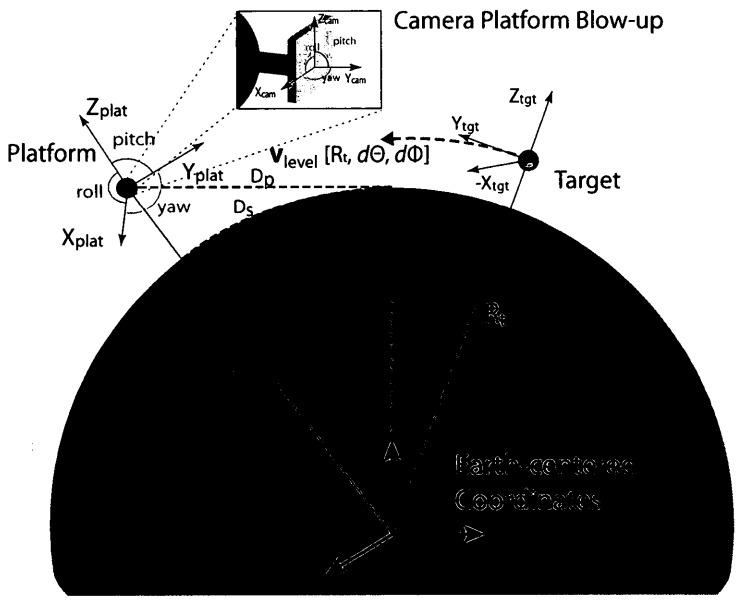


Figure 3.2: Coordinate system for evaluating the horizon

discern the values of D_p, D_{sp} and θ_p, as indicated in Eqs. 3.1, 3.2 and 3.3.

$$D_p \simeq \sqrt{2A_p(R_e + A_p)} \tag{3.1}$$

$$D_{sp} = R_e \arccos\left(\frac{R_e}{R_p}\right) \tag{3.2}$$

$$\theta_p = \arccos\left(\frac{R_e + A_p}{R_e}\right) \tag{3.3}$$

Here, ΔA_p and ΔR_e are uncertainties associated with local measurements of the aircraft's position and the Earth radius, respectively. Additionally, note that R_e is a function of latitude. Applying the computation to a general aviation altitude of 300 m ($\sim$ 1000 ft) reveals a $D_p = 62$ km. Given that horizontal human visibility approaches 30 km in clear sky conditions, resolving the horizon is beyond the range

of human visibility for most flights. At these distances, the horizon appears as a blurry merging of 'sky' and 'ground', rather than a precise boundary.

3.4.3 Projection of the horizon

Given a world point, $\mathbf{X} = [X, Y, Z]^T$, its equivalent projection in image coordinates, $\mathbf{x} = [x, y, 1]^T$, can be computed by the camera matrix $\mathbf{P}$ (Eq. 3.4).

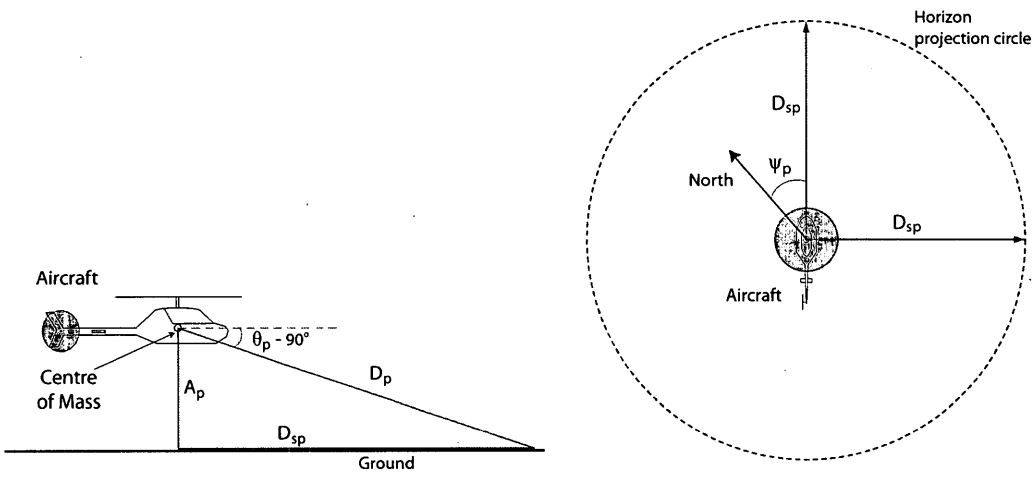
$$\mathbf{x} = \mathbf{PX}, \qquad \text{where} \quad \mathbf{P} = \mathbf{K} \cdot [\mathbf{R}| - \mathbf{Rt}] \tag{3.4}$$

$$\mathbf{K} = \begin{bmatrix} s_x & 0 & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \tag{3.5}$$

Here $\mathbf{K}$ is the intrinsic matrix of the form in Eq. 3.5, while $[\mathbf{R}| - \mathbf{Rt}]$ is a 3×4 augmented matrix representing a Euclidean transformation, which can be derived by calibrating the camera. Given world coordinates of the horizon, the corresponding image points can be computed by multiplying the world point by the extrinsic matrix ($\mathbf{M} = [\mathbf{R}| - \mathbf{Rt}]$), projecting (dividing) by Z, and multiplying the result by the intrinsic matrix. For mathematical convenience, the remainder of the derivation assumes homogeneous coordinates, such that each projection may be represented by a 4×4 matrix.

The critical step to this calculation is the representation of the horizon in world coordinates. Let us consider a world reference frame defined in UTM coordinates (for convenience), and the known aircraft position and orientation, denoted by

$\mathbf{X_p} = [X_p, Y_p, Z_p, 1]$ and Euler angles $[\theta_p, \phi_p, \psi_p]$, converted into a suitable rotation matrix, $\mathbf{R_p}$. If perceived from a top-down configuration, the horizon, under the flat terrain model, projects as a circle of radius D_{sp} on the earth's surface, centered at the aircraft UTM coordinates (Fig. 3.3b). A side view (Fig. 3.3a) depicts the direct linear distance to the horizon as D_p, while D_{sp} depicts the surface distance and A_p is the local ground altitude corrected from the mean sea level (MSL). Note that the aircraft centre of gravity is assumed to reside at the centre of the IMU.



(a) Side view

(b) Birds-eye view

Figure 3.3: Birds-eye and side views of the projected horizon

Since the radius of the circle, D_{sp}, is strictly a function of the platform altitude, coordinates to points on the horizon circle ($\mathbf{X_h}$) are given by the Cartesian representations, as per Eq. 3.6. Note that these points are relative to the aircraft

coordinate system.

$$\mathbf{X_h} = \begin{bmatrix} X_h \\ Y_h \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} D_{sp}\cos(\alpha - \psi_p) \\ D_{sp}\sin(\alpha - \psi_p) \\ 0 \\ 1 \end{bmatrix}, \text{ where } \alpha = (-\pi, \pi) \tag{3.6}$$

Here, ψ_p denotes the heading of the aircraft. The azimuth, α, spans the full circle with the plane of the horizon. A set of $\mathbf{X_h}$ points may be sampled at a resolution determined by the number of cameras and the desired number of horizon points per camera. Note that projecting all sampled points into each camera will reveal a sub-set of points that are valid image coordinates for that particular camera.

For a given camera, i, in the collision detection instrument, the k^{th} horizon point is projected as per Eq. 3.7. One can decompose $\mathbf{P_i}$ to reflect intermediate transformations. Note that since the origin is at the IMU, $\mathbf{t_p} = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}]^{\mathbf{T}}$ and therefore $\mathbf{P_p} = \mathbf{R_p}$. Further note that although the horizon projection is heading-invariant, the aircraft's pitch and roll will dictate the location of the line within the image, therefore R_p must remain part of the derivation.

$$\begin{aligned} \mathbf{x_i} &= \mathbf{P_i X_h} \\ \therefore \mathbf{x_i} &= \mathbf{K_i} \cdot [\mathbf{R_i}| - \mathbf{R_i t_i}] \cdot [\mathbf{R_{cam}}| - \mathbf{R_{cam} t_{cam}}] \cdot [\mathbf{R_p}| - \mathbf{R_p t_p}] \cdot \mathbf{X_h^k} \end{aligned} \tag{3.7}$$

Now, since D_{sp} is much larger than any intermediate translations, we may, without loss of generality, simplify the transformation to just rotations (Eq. 3.8).

$$\mathbf{x_i} \simeq \mathbf{K_i R_i R_{cam} R_p X_h^k} \tag{3.8}$$

Here, R_{cam} represents the rotation offset between the platform and the camera array, while R_i is the rotation offset between the array and one designate camera. Normally the inversion of the projection matrix is numerically approximated, and therefore prone to error. Recalling the symmetry property of a rotation matrix, namely that $\mathbf{R^{-1} = R^T}$, the present derivation of $\mathbf{P_i}$, due to its exclusive reliance on rotation matrices, has a valid non-zero inverse. Eq. 3.9 illustrates the short derivation confirming this fact.

$$\begin{aligned} \mathbf{P_i^{-1}} &= \mathbf{(R_i R_{cam} R_p)^{-1}} \\ &= \mathbf{R_p^{-1} R_{cam}^{-1} R_i^{-1}} \\ \therefore \mathbf{P_i^{-1}} &= \mathbf{R_p^T R_{cam}^T R_i^T} \end{aligned} \tag{3.9}$$

Finally, Eq. 3.10 defines the resultant projective transform and its inverse for a single camera in the array.

$$\mathbf{P_i = R_i R_{cam} R_p}, \quad \mathbf{P_i^{-1} = R_p^T R_{cam}^T R_i^T} \tag{3.10}$$

This derivation highlights two key facts: Firstly, the lever arms offsetting the instrument from the aircraft centre of gravity (IMU) are not required to achieve an explicit calibration. Secondly, the projection of a point onto the focal plane (before the application of the intrinsic matrix) is invertible. Therefore, $\mathbf{P}$ and $\mathbf{P^{-1}}$ can be computed offline, and applied in real-time, in-situ as a single matrix multiplication. Any point of interest on one focal plane may be transformed to others by matrix operations.

3.4.4 Proximity to the horizon - ΔH_{tgt}

The extended nature of the horizon suggests that the horizon alone may be sufficient for extracting the relative vertical positioning of a target object. We can quantify this property as the "horizontal proximity", ΔH_{tgt} and designate it as an important figure of merit. In a level-flight scenario, targets on a collision course with the host must be co-altitude. Recall that, as per Fig. 2.2a in Section 2.4.3, the temporal variations in the target's azimuth are minimal for the duration of the collision trajectory. This observation extends to elevation as well, since temporal variations in elevation, observed as vertical pixel shifts, correspond to changes in the target altitude.

Consider a relative position of a target starting at long-range, beyond the horizon limit. At distances beyond the horizon, the target is obscured by the horizon, and beyond the direct line of sight, irrespective of atmospheric visibility. As the target crosses the horizon threshold limit, it becomes visible. As it progresses along a collision course, its apparent elevation begins to increase towards a local maximum, after which point it stays constant for the remainder of the flight. If atmospheric effects dominate, as expected, the fence radius will lie well within the horizon radius, and the target will appear to "emerge" from the atmosphere at a fixed elevation offset from the horizon. In summary, the target position, in azimuth and elevation,

is fixed relative to the host reference frame for the duration of the collision course.

In the absence of all other effects, a fixed target will appear to be temporally "stuck" in pixel-space. However, the host platform is prone to random motions and vibrations, causing the target to move randomly in pixel-space. Note that since platform motions affect the frame of reference, pixel-shifts of the general scene elements are precisely correlated to pixel-shifts of the target. Therefore, despite random host motions, the target's apparent proximity to the horizon will remain constant for the duration of the period until collision.

3.5 Detector footprint

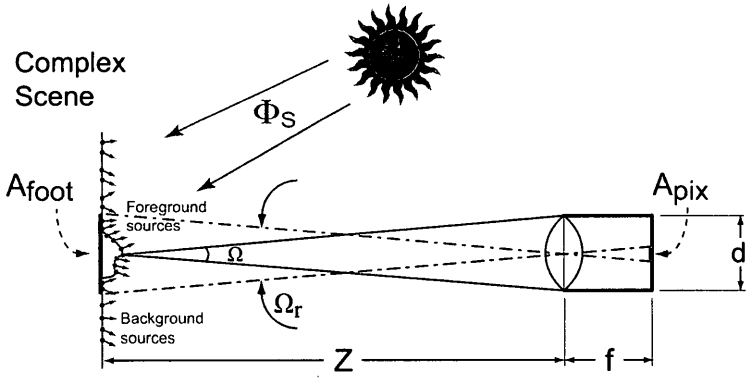


Figure 3.4: Imager irradiance of a target of interest

Consider a detector element, or pixel, subtending a non-trivial footprint area, A_{foot} [m^2], within the scene (Fig. 3.4). This footprint is a function of the pixel

area, A_{pix}, the focal length of the lens (f) and distance to the scene surface (Z). Given that the pixel solid angle (α_{pix}) can be defined as per Eq. 3.11, the pixel footprint can consequently be expressed as Eq. 3.12.

$$\alpha_{pix} = \frac{A_{foot}}{Z^2} = \frac{A_{pix}}{f^2} \qquad [\text{sr}] \tag{3.11}$$

$$\therefore A_{foot} = \frac{A_{pix} Z^2}{f^2} \qquad [\text{m}^2] \tag{3.12}$$

The parameters, P_d and P_s, are the power contributions of single diffuse and specular (differential) elements respectively. The footprint represents an integral over many such elements in the scene. A single footprint can include both background and target elements. Although the scene is shown as a plane in Fig. 3.4, it could be a curved surface. Additionally, obscuration of the line of sight is ignored. Let $P_d = \Phi_{df} \delta A_p$ and $P_s = \Phi_{sp} \delta A_p$, where Φ is the power per unit-area, or irradiance, in units of $[\text{W} \cdot \text{m}^{-2}]$. By treating δA_p as the differential variable, one can take the integral over A_{foot} to determine the total power incident on the pixel. (Eq. 3.14).

$$P_{foot} = \int^{A_{foot}} \Phi_{df} \tau_d(\theta, \phi) \delta A_p + \int^{A_{foot}} \Phi_{sp} \tau_s(\theta, \phi) \delta A_p \tag{3.13}$$

$$P_{foot} = A_{foot} \Phi_{df} \tau_d(\theta, \phi) + A_{foot} \Phi_{sp} \tau_s(\theta, \phi) \tag{3.14}$$

where $\tau_d(\theta, \phi)$ and $\tau_s(\theta, \phi)$ are orientation weighting functions. In the purely geometric sense, the most convenient one as it happens, $\tau_d()$ and $\tau_s()$ can be defined as Heaviside step functions (Eq. 3.15 - 3.16), thereby defining the contributing

elements only as those whose radiance normals fall within the solid angle of the detector element.

$$\tau_d(\theta,\phi) = \begin{cases} 1, & |\ \hat{\mathbf{u}}_{\mathbf{lens}} \cdot \hat{\mathbf{u}}_{\mathbf{diff}}\ |^2 < \Omega \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

$$\tau_s(\theta,\phi) = \begin{cases} 1, & |\ \hat{\mathbf{u}}_{\mathbf{lens}} \cdot \hat{\mathbf{u}}_{\mathbf{spec}}\ |^2 < \Omega \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

Note that each point-like source is assumed to be radiating independently and interference is ignored. The scene area subtended by the pixel field-of-view (A_{foot}) is considered to be planar, constant-ranged and uniformly distributed. The first assumption ensures that the normals of all diffuse sources within the footprint are quasi-parallel, and similarly so for the angles of reflection of all potential specular sources. These assumptions hold as long as the source and sensor are sufficiently far away. Meanwhile, the second assumption allows point-like sources from the target and background scene to be summed together. A more gradual gradient for $\tau_d()$ and $\tau_s()$ functions allow the incorporation of various non-geometrical effects into an otherwise geometrical model. These might include diffraction, and external factors, such as the point-spread function from the atmosphere.

3.6 Target signal

A general target can be approximated by a large number of planar surfaces, sometimes termed 'micro-facets', each of which has a small area, δA_p. For outdoor

targets, solar irradiance (Φ_S [W $\cdot$ m^{-2}]) is the primary source of reflected flux. For reference, on a sunny cloudless day, the typical solar flux in the visible spectrum is approximately 100 W $\cdot$ m^{-2} [WZ89]. At the distances considered, the reflected flux is considered to be the primary source of *overall* output flux. Figure 3.5 illustrates the radiant exitance of a single facet under solar illumination, while Eq. 3.17 expresses the output power as a function of solar flux.

$$P_p \approx \delta P_r = \rho \Phi_S \delta A_p \cos\theta \qquad [\mathrm{W}] \tag{3.17}$$

where ρ is the reflectance coefficient, θ [angle] is the normal between the sun and the point-like surface, δA_p [m^2] is the surface area, and Z is the distance between the source and the imaging system, which is considered to begin at the front element of the lens. If the imaging plane is determined by a lens of diameter d, typically the front light-gathering element, then $A_r = \frac{\pi d^2}{4} \cos\psi$, and the equivalent solid angle is $\Omega = \frac{A_r}{Z^2}$.

Note that the sun acts as a quasi-parallel source in this model. Additionally, scene elements where the line of sight is blocked are ignored. The medium between the target surface and the sensor is assumed to be air, and furthermore, assumed to be homogeneous with a constant ambient temperature. This renders the light unaffected by any effects caused by varying refractive index during its passage through the medium.

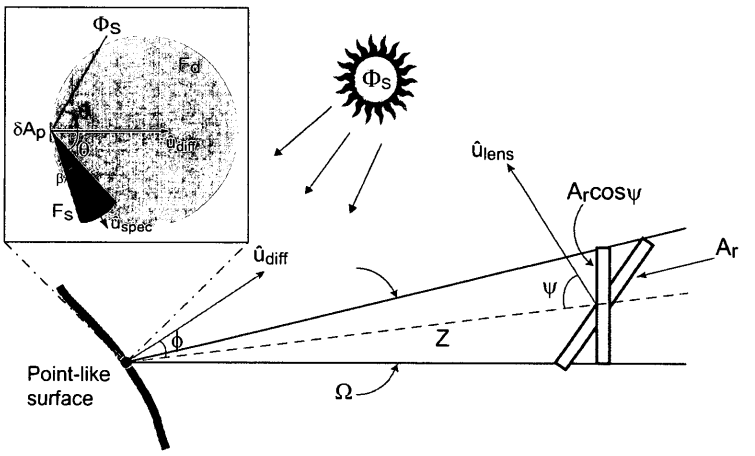


Figure 3.5: Reflected solar radiance from a small planar segment of the target, with inset depicting diffuse and specular reflections [Kop98].

3.6.1 Diffuse and specular reflections

Surfaces can exhibit a superposition of diffuse and specular reflections as per the dichromatic reflection model [UG04]. Diffuse reflections result from light waves penetrating the material surface, multiplying, reflecting and refracting internally, and then refracting back out. This behaviour is well-modeled by Lambert's law, provided the angle of incidence is small.

On the other hand, specular reflections are caused when incident light waves are reflected directly on the object surface, at the medium interface. Specular highlights are observed strongly in the direction of proper reflection, are highly viewpoint dependent, and can be strongly polarized. The Torrance-Sparrow model [TS67] is popular for specular reflections, however a much simpler model, ignoring

polarization, is considered here for convenience.

Consider the inset in Fig. 3.5, which depicts both diffuse and specular reflectance functions, F_d and F_s, respectively. As illustrated, the diffuse reflectance function, F_d, is distributed uniformly over a hemisphere, and attenuated by the cosine of the angle (ϕ) between the line-of-sight vector and the source normal ($\hat{u}_{diff}$), such that $F_d = \dfrac{\cos\phi}{\pi}$.

Meanwhile, a straightforward representation for specular reflectance (F_r) is that of a regular conic (see Inset:Fig. 3.5), depicting a beam originating from the point-like surface with a pointing direction, θ, from the surface normal ($\hat{u}_{diff}$), and a divergence angle (β). With reference to the same figure, if the output diffuse power from the point-like surface is incident upon the imaging surface area, A_r, then H_d (Eq. 3.18) encapsulates the radiant intensity from the diffuse component.

$$H_d = \rho_d \Phi_S \delta A_p \cos\theta \frac{\cos\phi}{\pi} \qquad [\mathrm{W}\cdot\mathrm{sr}^{-1}] \tag{3.18}$$

Here, the only new variable, ρ_d, is the diffuse reflectance coefficient. A similar derivation can be had for specular radiant exitance H_r (Eq. 3.20), assuming Lambertian falloff over the conic base.

$$H_r = \rho_r \Phi_S A_p \cos\theta F_r \qquad [\mathrm{W}\cdot\mathrm{sr}^{-1}] \tag{3.19}$$

$$= \frac{\rho_r \Phi_S \delta A_p \cos^2\theta}{\pi \tan^2\beta} \qquad [\mathrm{W}\cdot\mathrm{sr}^{-1}] \tag{3.20}$$

Here, at the distance Z, the surface area subtended by the base of the cone can be

quantified as $A_s = \pi Z^2 \tan^2 \beta$, and the resulting solid angle as $\Omega_s = \pi \tan^2 \beta$.

Note that the previous equations define the radiant exitance from a single facet. To determine total power received at the imager optics, one must integrate over all incident sources. Although it is assumed that incident power is transferred to a single detector element (pixel) without loss, various sources may restrict or limit the detector irradiance. The presence of specular reflections implies strong signal fluctuations from the target due to host and platform orientation changes, as well as changes in position and intensity of the ambient solar illumination.

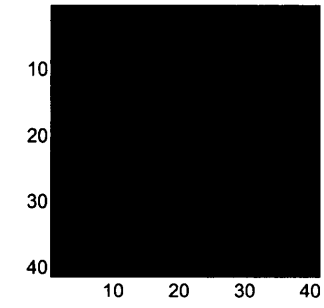
3.6.2 Target non-uniformity

Experimental data indicates that observed targets exhibit high degree of deviation from the ideal model. Modern aircraft are a mixture of alloys, glass, polycarbonate and composite materials, exhibiting strong textural components and large gradients along surface boundaries. The composition and distribution varies with sun position, target orientation and host attitude. Specular highlights are especially challenging, as they appear as bright, strong signal sources, but are temporally sporadic and can look like artifacts. The received signal is strongly influenced by the orientation of the light source (sun), therefore radiance profiles vary with the time of day. Temporal alterations in the viewing orientation from motion in both host and target platforms can dramatically alter the observed signal.

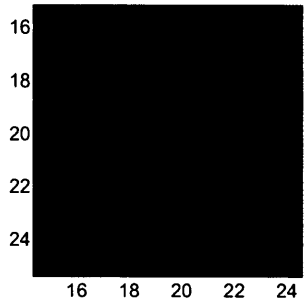
Radiant non-uniformity of realistic targets can be quite difficult to model in simulation environments. Spatial, short-term, and long-term variations in the observed albedo can be drastic due to solar output variations due to solar orientation, obscurations, and platform and target attitudes. In a day-VFR (visual flight rules) scenario, the sun is typically above both host and intruder aircraft, radiating directly onto, or obliquely across, the target's top canopy. Besides the variation of reflectivity due to Lambertian scattering, parts of the aircraft are often obscured from direct illumination. The signal profile can be sensitive to the orientation of the target with respect to the solar direction. This creates a radiance discontinuity readily discernible in target imagery (Fig. 3.6).

Additional target facets and textures further challenge the hemispherical assumption, resulting in more complex radiance profiles. Consider the transverse (Fig. 3.6c) and medial (Fig. 3.6d) radiance profiles of a Bell 206 Jet-ranger helicopter (Fig. 3.6a). The 206 has multiple radiance discontinuities, due to the large rotor, pilot canopy, followed by a semi-circular nose obscuring the underbelly.

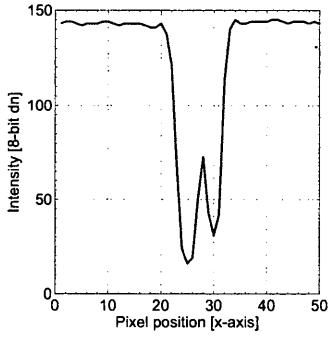
At longer ranges, the system point spread function mixes these target components with the local background. Specular reflections can be present at all ranges and contribute to the dynamic variability of the observed signal. Solar orientation can also affect the perceived profile. Note that the 'radiance discontinuity profile' is likely to be unique to a particular aircraft model and may serve as an important



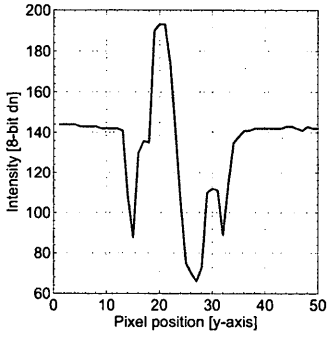
(a) Selected target ROI



(b) Magnified target



(c) Transverse intensity profile



(d) Medial intensity profile

Figure 3.6: Crop of a Bell 206 Jet-ranger observed at 3.5 km with intensity profile cross-sections along the central medial and transverse axes.

parameter in classification.

3.7 Signal-to-Noise (SNR) transfer model

The radiance from a localized target at the range R can be approximated as the sum of incoherent intensity contributions from planar differential elements illuminated

by sunlight. Therefore, the non-saturation photocurrent generated in an imaging pixel by target contributions (i_t), and incremental background contributions (i_b), can be described as per Eqs. 3.21 and 3.22 respectively, adjusted for losses due to absorption (τ).

$$i_t = k \cdot \zeta A_{foot} \cdot \Phi_O \cdot \tau_O \tag{3.21}$$

$$i_b = k \cdot A_{foot} \cdot \Phi_B \cdot \tau_B \tag{3.22}$$

$$\text{where } k = \frac{\eta q G}{E_{ph}},\ \zeta \in [0, 1]$$

In both equations, G denotes the gain, η, the photodiode's quantum efficiency, q, the charge of an electron, and E_{ph}, the energy of a photon. Since wavelength dependence is not considered in the model, E_{ph} can safely assume a constant value, preferably the dominant wavelength of 520 nm in reflected sunlight. Similarly, Φ_O and Φ_B denote the target and background irradiances, respectively. Typical targets at the limit of detection are expected to be smaller than A_{foot}, therefore ζ, as bounded, defines the fraction of A_{foot} subtended by the target. An approximate, theoretical single-pixel signal-to-noise ratio (SNR_t) for a point-like target embedded in the background, in the presence of transfer functions and noise sources, can be defined as per Eq. 3.23 [Kop98],

$$SNR_t \approx \frac{\mid M_t i_t - M_b i_b \mid \cdot t_i}{\sqrt{(i_t + i_b) \cdot t_i + \sigma_r^2 + \sigma_{rs}^2 + \sigma_{tgt}^2}}, \tag{3.23}$$

where M_t and M_b are the modulation transfer functions of the target and background respectively, while t_i is the integration period. Noise is encapsulated in the denominator as a combination in quadrature of shot ($\sigma_s = \sqrt{(i_t + i_b).t_i}$), read (σ_r) and reset (σ_{rs}) noise elements [Jan01]. An unconventional term, σ_{tgt}, has been added to account for Gaussian fluctuations in target radiance. The introduction of the empirical quantity, σ_{tgt}, is an attempt to include the effect of changing target signal on the ability to detect the target.

3.8 Transfer functions

Transfer functions limit the maximum resolution and contrast at the output in imaging systems. In aerial imaging systems, the combined modulation transfer function can be treated as a linear convolution of atmospheric (M_A), platform motion (M_V), optical (M_O) and detector (M_D) contributions (Eq. 3.24).

$$M = M_A(t) * M_V(t) * M_O * M_D \tag{3.24}$$

Note that while M_O and M_D are assumed to be spatio-temporally static, M_V is a function of time and M_A, a function of range. Assuming a nominal wavelength and real-valued signals, the modulation transfer functions (MTFs) are adequate for calculating the received signal magnitude.

3.8.1 Atmosphere

Atmospheric background radiance adversely affects contrast, noticeably limiting target acquisition ranges. Atmospheric attenuation can be summarized by absorption and scattering mechanisms [KB70]. Most environments contain significant amounts of aerosols, causing scattering. Below the 2.0 km altitude boundary, the typical aerosol radius is 10^{-2} microns, and the mean atmospheric density is 15000 particles/cm^3. Note that increased humidity can increase aerosol concentration [SHK$^+$02]. Aerosol scattering results in a blur circle, which may be approximated by a normalized, bivariate-symmetric Gaussian distribution. Eq. 3.25 depicts the generalized scattering magnitude function,

$$M_S = \frac{8\pi^4 N\alpha^2}{\lambda^4 Z^2} \tag{3.25}$$

$$\therefore M_S = k/Z^2, \text{ where } k = \frac{1}{\lambda^4} \text{ (for } \lambda = 550 \text{ nm)} \tag{3.26}$$

Here, N is the number of scatterers, α, the polarization, λ, the incident photon wavelength, while Z is the distance to the observer. Although normally Eq. 3.25 is dominated by wavelength as $M_S \propto \lambda^{-4}$, assuming a fixed 'average' value for λ, the relationship can be simplified to Eq. 3.26.

Assuming a semi-infinite, homogeneous medium with uniform particulate radius and density [SHK$^+$02], scattering can be treated as a source of attenuation, and

approximated by the Beer-Lambert Law (Eq. 3.27),

$$I_{out} = I_0 e^{-(b_r+\tau)Z} + B_0(1 - e^{-(b_r+\tau)}Z) \tag{3.27}$$

$$\therefore I_{out} = I_0 e^{-\sigma_e Z} + B_0, \quad \text{where } Z >> f \tag{3.28}$$

where the input intensity I_0 is exponentially attenuated at range R. Meanwhile, σ_e is the atmospheric extinction coefficient, decomposed into scattering (b_r) and absorption (τ) coefficients. In the vis-near-IR spectral region of primary interest here, τ can be ignored while the scattering coefficient, b_r, is important. Similarly, the background signal B_0 is tied to atmospheric attenuation as a function of range. For long-distance imaging, $Z >> f$, background irradiance is approximately constant for a given attenuation coefficient (Eq. 3.28).

The extinction coefficient can be approximated by Koschmeider's formula (Eq. 3.29) for horizontally viewed targets [Ove76]. This is an appropriate assumption in the present scenario as most co-altitude targets will emerge at or near the horizon. The relationship can be simplified if a nominal wavelength of 550 nm is assumed (Eq. 3.30).

$$\sigma_e = \frac{3.912}{kV}\left(\frac{\lambda}{550}\right)^q \tag{3.29}$$

$$\sigma_e \simeq \frac{3.912}{kV} \quad (\text{for } \lambda = 550 \text{ nm}) \tag{3.30}$$

Here, the extinction coefficient, σ_e, is related to an empirical, measured quantity, namely the visibility V. An initial estimate for local visibility can be obtained

from the local aerodrome meteorological terminal aviation routine weather report (METAR). Note that these estimates are empirical and measure general visibility in the entire region as viewed from the airport. Prevailing local visibility conditions may vary significantly, by as much as a factor of five, from the METAR observations. The reciprocal constant, k, is often heuristically modified to correct for observational deviations, with values ranging between 2.0 and 5.0. The METAR visibility was used for safety considerations and constituted a part of the preflight "go/no-go" decision-making process.

3.8.2 Platform motion

Aerial platforms are prone to periodic shifts from onboard propulsion mechanisms and wind effects (gusting). Flight dynamic modes dominate [ER95], contributing low-frequency, high-amplitude oscillations that are coupled across the control surfaces. Mechanical vibrations are high-frequency, low-amplitude artifacts that are generated at the mount's resonant frequency.

Hence, aircraft-mounted cameras are subjected to linear, sinusoidal and random motion simultaneously [Hol03]. Image artifacts are frequently termed 'shear', 'skew' or 'wobble', however, the net result is always target blurring. Consider Eq. 3.31, a combination of three MTFs: The first term models linear motion as a *sinc*() function, where ω_l is the spatial sampling frequency, v_l, the linear velocity, and t_i,

the integration time.

$$\begin{aligned} M_V &= M_{lin} * M_{sin} * M_{rnd} \\ &= \left| sinc(\frac{\omega_l}{2\pi} v_l t_i) \right| * J_0(a_s \omega_s D) * e^{-\sigma_r {\omega_r}^2} \end{aligned} \tag{3.31}$$

In the second term, the blur radius is twice the amplitude (a_s) of the oscillatory frequency (ω_s). Finally, random motion is best approximated by a Gaussian distribution of standard deviation (σ_r) as the RMS random displacement. Note that all platform MTFs are unity-gain, positive and real-valued.

3.8.3 Optics

The optical transfer functions can be decomposed into diffraction, aberration and detector-sampling components (Eq. 3.32).

$$M_O = M_{diff} * M_{aber} * M_{distort} * M_{defocus} \tag{3.32}$$

Lens diffraction (M_{diff}) is the fundamental lower limit to optical spatial transfer functions, and is commonly defined by the Fraunhofer diffraction pattern, or Airy pattern (Fig. 3.7). The intensity I is defined as a function of the radial distance (Eq. 3.33) for a fixed wavelength, focal ratio and incident irradiance [BW99].

$$I(r) = I_0 \left(\frac{2J_1(x)}{x} \right), \quad \text{where } x = \frac{\pi r}{\lambda F} \tag{3.33}$$

where J_1 is the first-order Bessel function, I_0 is the maximum intensity (usually normalized), λ is the wavelength, F is the focal ratio of the lens, and r specifies

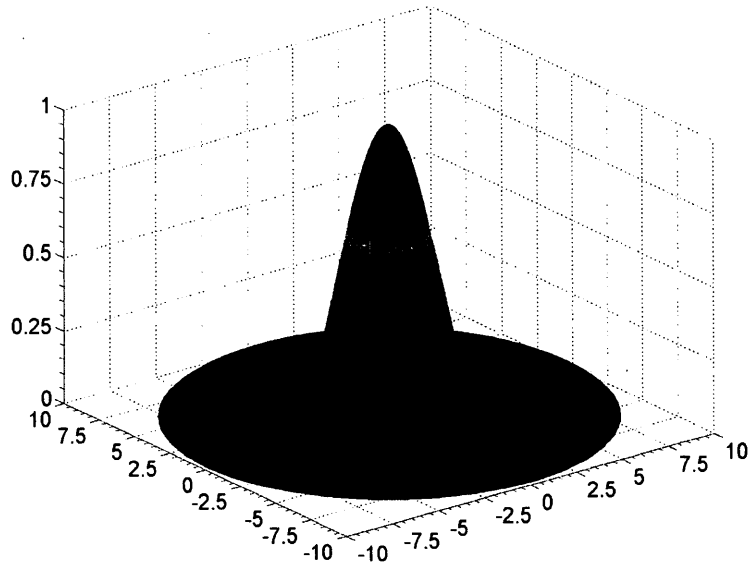


Figure 3.7: The Fraunhofer (Airy) diffraction pattern.

the radial distance from the optic axis. In most applications, the width of the main lobe is of particular interest, and the half-maximum occurs at $x \approx 1.62$. The disc may be approximated by a Gaussian function, where the full-width-at-half-maximum (FWHM) is related to the standard deviation σ by $FWHM = 2\sqrt{2ln(2)}\sigma$. Alternatively, for a given lens of known focal ratio, the half-width of the Airy disc may be approximated since $\sigma \approx 0.45\lambda F$.

Although the diffraction limit is the lower limit, typical lenses exhibit spheroid aberrations (M_{aber}), distortion ($M_{distort}$) and astigmatic artifacts, especially in inexpensive commercial C-mount lenses. These aberrations are well-described by multi-degree polynomial formulations, such as the Zernike equations [Mah94]. Spherical

and chromatic aberrations are not easy to correct without an explicit deconvolution of the point-spread function (PSF), which is not always possible if the PSF is not invertible.

Although distortion is not explicitly a point-broading effect, it is included with the standard optical PSF model for completeness. Distortion is commonly corrected *ex tempore* in image coordinates by Brown's distortion model, which corrects for radial and tangential distortion [Bro71]. This distortion model is represented by an infinite Taylor series (Eq. 3.35).

$$x_u = x' \cdot (1 + k_1 r^2 + k_2 r^4 \dots) + t_1(r^2 + 2(x')^2) + 2t_2(x'y')(1 + t_3 r^2 + \dots) \quad (3.34)$$

$$y_u = y' \cdot (1 + k_1 r^2 + k_2 r^4 \dots) + t_2(r^2 + 2(y')^2) + 2t_1(x'y')(1 + t_3 r^2 + \dots) \quad (3.35)$$

$$\text{where } x' = x_d - x_c, \quad y' = y_d - y_c, \text{ and } \quad r = \sqrt{x'^2 + y'^2}$$

Here, (x_u, y_u) represent the undistorted version of the distorted point (x_d, y_d), while (x_c, y_c) is the point of intersection of the optic axis with the camera plane, which is often referred to as the 'principal point'. Points are denoted in the camera plane, therefore image coordinates must first be transformed (multiplied by $\mathbf{K^{-1}}$) before applying distortion correction.

Since the distortion model is an infinite Taylor series, the application is usually restricted to a finite number of coefficients. The first two radial (k_1, k_2) and tangential (t_1, t_2) coefficients usually provide adequate correction for the majority

of lens configurations. Also note that it is common to assume that the centre of projection in image coordinates is identical to the principal point, although this assumption may contribute additional errors [HK07]. Finally, defocussing artifacts ($M_{defocus}$) are common in long-range applications and attributable to human error. Care must be taken to ensure that the lens is focused to infinity, and the focus ring securely locked in place for flight trials.

3.8.4 Detector

The signal output, $i(t)$, for a single pixel recording a discrete event at time t can be modeled as per Eq. 3.36 [WGS07].

$$i(t) = g^{\gamma}(i_0 + \mu_d + \sigma_s + \sigma_r) + \sigma_q \tag{3.36}$$

$$\therefore i_s(t) = g^{\gamma}(i_0 + \sigma_r) \qquad [\text{dn}] \tag{3.37}$$

Here, i_0 denotes the true scene intensity, while g denotes the camera gain, in units of [dn/photons]. Noise sources present through the chain include: Dark current (μ_d), shot noise (σ_s), read noise (σ_r) and quantization noise (σ_q). and is dominated by the random generation of photoelectrons. γ is the response conversion factor, utilized to extract greater dynamic range near the saturation region, with values between 0.5 and 0.7 being typical. Note that shot noise is significant in outdoor scenes, and expected to swamp all other noise sources, simplifying the

output signal to a shot-noise limited regime $i_s(t)$ (Eq.3.37).

Brightness and contrast parameters add additional variability to the digitized output. Brightness is an offset component while contrast determines the slope for a given digital number. If the camera is configured in auto-exposure mode, brightness and contrast are adaptive and must be denoted as temporally varying parameters.

3.8.5 Solar orientation

A final point of note is with regards to the orientation of the sun with respect to the sensor. Given the wide field-of-view specification, the sun can often be in the direct field of view of one or more camera nodes. Immediate effects include saturation, blooming, retro-reflections and lens flare, while long-term exposure may generate excess heat, resulting in increased dark current or undesired photo-response non-uniformity. Long-term ultraviolet radiation can cause permanent changes in the substrate doping, leading to burn-in (large drain current), increased cross-talk and undesired photo-response non-uniformity [UG04].

Using a multi-camera system is of positive benefit, as the crippled camera module(s) may be turned off. A mechanical shutter may be introduced if damage from long-term exposure is anticipated. Processing with regards to the determination of the sun position, and operation of the mechanical shutter, can be performed locally per node. Location of the sun may help orient the sensor if the time of day is

known.

3.9 Calibration of a large camera array

It is difficult to calibrate a multi-camera array subtending a large field of view without the aid of large targets or precise mounting systems. As such, this section describes a calibration technique for just such an array observing long-range scene elements. Note that for a large camera array, the principal calibration of interest is the rectification or transformation between adjacent image pairs. If the rectification between a camera and its neighbours are known from all cameras, then any scene element generating an image coordinate in one camera may be specified in the coordinate system of any other camera within the calibrated array. The technique described herein represents an adaptation from material presented in Hartley and Zisserman's *Multiple View Geometry* [HZ03].

The principal advantage of this technique is that it obviates the need to fabricate a massive calibration target covering the full instrument field-of-view. Also, it can be applied directly to captured ground-based or aerial imagery, therefore once the instrument is configured for the task (lenses focussed to infinity) and affixed to the aircraft, it need not be modified for calibration.

Consider a point in the world scene, $\mathbf{X}$, which projects a valid image coordinate pair $\mathbf{x_i}$ in camera i, and a corresponding pair $\mathbf{x_j}$ in camera j. Let us now define

a homography, $\mathbf{H_{i,j}}$, that represents a transformation between coordinates in one camera plane to coordinates in the second camera plane (Eq. 3.38).

$$\hat{\mathbf{x}}_\mathbf{j} = \mathbf{H_{i,j}}\hat{\mathbf{x}}_\mathbf{i}, \quad \text{where } \mathbf{H_{i,j}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \quad \hat{\mathbf{x}} = [x, y, 1]^\mathbf{T} \tag{3.38}$$

Note that this homography represents a projective transformation between the two camera planes, and that $h_{33} = 1$, constraining the search to eight parameters. The homography as described can be derived per camera pair, and utilized for correlating point matches, or rectification during a panoramic stitch. The individual parameters of the matrix can be computed by the normalized Direct Linear Translation (DLT) algorithm (Algorithm 4.2, pg 109) in [HZ03], if at least four matching point pairs are provided.

Let us consider a camera pair i and j, with the world origin at camera i such that camera j is defined by a linear rotation and translation relative to camera i (Eq. 3.39).

$$\mathbf{P_i} = \mathbf{K_i}[\mathbf{I}|\mathbf{0}], \quad \mathbf{P_j} = \mathbf{K_j}[\mathbf{R}|-\mathbf{Rt}] \tag{3.39}$$

Now, let us consider a case where cameras i and j are adjacent and have overlapping fields of view. We can choose any two cameras within the instrument. If an overlap exists, some scene points will have valid image coordinate pairs in both cameras. Given correspondences and sufficient constraints, it is possible to recover the homography between the two cameras.

Consider points from the horizon projecting into this overlap. If we assume that the overlap between adjacent cameras to be narrow, on the order of a few degrees, then one can impose a useful constraint on points from the horizon which project into this narrow FOV. Consider the set of horizon points within the FOV, defined by $\mathbf{X_{h,F}}$. Given the Earth's circumference and relative distance of the horizon points from the camera centre, $\mathbf{X_{h,F}}$ can be assumed to project onto a plane $\mathbf{pi_F^T X_h = 0}$, with $\pi_\mathbf{F} = (\mathbf{n^T}, D_{sp})^T$. Note that D_{sp} defines the distance to the plane, while $\mathbf{n}$ defines the normalized plane-normal vector.

Given the world plane, $\pi_\mathbf{F}$, the homography between two cameras, $\mathbf{H_{i,j}}$, is given by Eq. 3.40.

$$\mathbf{H_{i,j}} = \mathbf{K_j}\left(\mathbf{R_{i,j}} - \frac{\mathbf{t_{i,j}} \cdot \mathbf{n}^T}{D_p}\right)\mathbf{K_i^{-1}} \tag{3.40}$$

Let us take the limit where D_p approaches infinity Eq. 3.41. This is a valid approximation since $D_{sp} >> \mathbf{t_{i,j}}$.

$$\mathbf{H_{i,j}^{\infty}} = \lim_{D_{sp}\to\infty} \mathbf{K_j}\left(\mathbf{R_{i,j}} - \frac{\mathbf{t_{i,j}} \cdot \mathbf{n}^T}{D_{sp}}\right)\mathbf{K_i^{-1}} \tag{3.41}$$

Then, the homography at infinity ($\mathbf{H^{\infty}}$) simplifies to Eq. 3.42, with the corresponding projection matrix $\mathbf{P_{i,j}}$ defined accordingly.

$$\therefore \mathbf{H_{i,j}^{\infty}} = \mathbf{K_j R_{i,j} K_i^{-1}}, \quad \mathbf{x_j} = \mathbf{K_j R_{i,j} K_i^{-1} x_i} = \mathbf{P_{i,j} x_i} \tag{3.42}$$

Since $\mathbf{H_{i,j}^{\infty}}$ represents a *planar* homography, it may be computed by the DLT algorithm mentioned previously. It is noted that a known $\mathbf{H_{i,j}^{\infty}}$ is sufficient to map

points between cameras, rendering an explicit decomposition unnecessary. Nonetheless, if the intrinsic parameters are known, such as by performing an offline interior calibration of the cameras, then the relative rotation between cameras $\mathbf{R_{i,j}}$ may be readily uncovered in a straightforward fashion.

Given the matrices $\mathbf{K_i}$ and $\mathbf{K_j}$, image points $\mathbf{x_i}$ and $\mathbf{x_j}$ may be transformed into the focal plane as $\mathbf{x'_i} = \mathbf{K_i^{-1}x_i}$ and $\mathbf{x'_j} = \mathbf{K_j^{-1}x_j}$. Then the infinite homography may be computed on $\mathbf{x'_i}$ and $\mathbf{x'_j}$. The resultant $\mathbf{H^\infty}$ is by definition simplified to $\mathbf{H^\infty_{i,j}} = \mathbf{R_{i,j}}$.

Unfortunately, the use of point correspondences taken solely from the horizon is insufficient to solve for $\mathbf{H^\infty}$, as the points are collinear, and the corresponding constraints are poor. Additional scene features may be utilized, such as landmarks and other notable scene elements, provided that distance to the scene element is much greater than the lever arms, ergo $D_{sp} >> \mathbf{t}$. However, images captured in on the ground configuration highlighting distant landmarks can provide stable images with adequate features for this calibration technique.

3.10 Conclusion

Key ideas at the heart of the thesis have been described in this chapter: The fence paradigm, leading, in particular, to the range at first detection as a key figure of merit. The horizon as an omnipresent scene feature, a target signal model, interme-

diate transfer functions. A complete description of the imaging train is provided, from image formation, to intermediate transfer functions, and culminating in digitization. We end this chapter with a note on how to calibrate a large camera array with minimal effort.

4 Instrument Description

4.1 Synopsis

Described in this chapter are the design elements, evolution and modifications for the DragonflEYE prototype. A design outline is presented first, describing the instrument topology, followed by a functional description of each component and the constraints driving their choices. Laboratory and field prototypes are next, culminating in a description of the prototype utilized for flight trials. A discussion of the critical design changes between prototypes is presented, differentiated into software and hardware sections. The final section discusses a path to the future, in light of the steady evolution of CMOS technology, dynamic control and distributed sensing.

4.2 Introduction

The DragonflEYE prototype is designed as a smart-camera network array. Such a paradigm is flexible and scalable, and can remain relevant as requirements evolve. Considerations for the DragonflEYE instrument were driven by the following factors:

- Requirements and constraints as laid out in Chapter 2.
- Hardware and software hierarchy to adequately manage the high data rate.
- Evaluation of the instrument in a realistic environment.
- Engineering challenges encountered in the design and implementation of large camera arrays.

The design process was iterative, and progressed as a series of prototypes, the highlights of which are presented here. Additional details are available in reports written for the National Research Council of Canada (to which the author heavily contributed) [TML+10][MTW+11][TML+11].

4.3 Architecture

The instrument consists of identical smart camera modules connected to a common networking back-plane (Fig. 4.1). These modules, or 'nodes' interface with a control node, which provides a unified interface for the instrument. A 'test node' was

utilized during evaluation runs to collect raw imagery as well as ancillary data. In the prototype, the local Ethernet had a 100 Mbps link for every five nodes and these groups of nodes were linked by a 1.0 Gbps channel. A description and brief history of the smart camera paradigm is presented in Chapter 2.

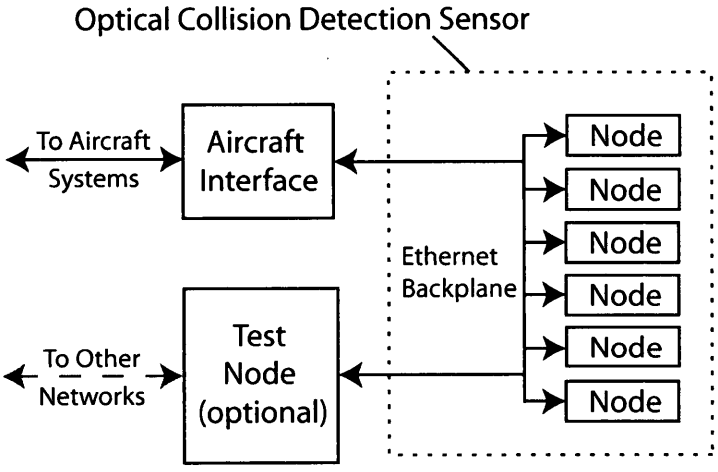


Figure 4.1: Operational configuration of the SAA instrument

Each node had the same hardware components, to within the limits set by manufacturer and machining tolerances. The camera calibration coefficients were stored locally, while the individual (computer) clocks were calibrated at instrument startup against an external clock. The software on each physical node could support a "camera" or a "control" mode of operation. Note that the nodes utilized in the flight test prototype employed a simpler state machine, with activities limited to image acquisition and synchronization. This decision reflected the need to maximize image data throughput while minimizing synchronization latencies and points of

failure.

4.3.1 Camera node software

The ideal camera node is structured as a smart camera with decentralized control and adaptive processing behaviour, whereby any one camera node may choose to act as a control node. Selection of the control node may be fixed, as in predefined, or adaptive. This guarantees that a) the control node utilizes the same hardware as individual camera nodes, and b) in case of failure, functionality of the control node may be transferred to a lower-priority camera, such as a side or rear camera in the event of high-speed forward flight. The software architecture of such a camera node is presented in Fig. 4.2.

Each camera node was equipped with all the software necessary for startup and operation. Upon application of power, each camera node independently loaded its operating system, read default parameters from a local memory card and started the root process.

The root process is responsible for the creation and maintenance of global/shared data structures, such as start-up lists, camera configuration parameters, network parameters, and a shared circular buffer with pointers for each thread. A monitor thread acts as the main thread of the application, spawning image control and sensor control child threads. The monitor thread ensures continued operation via

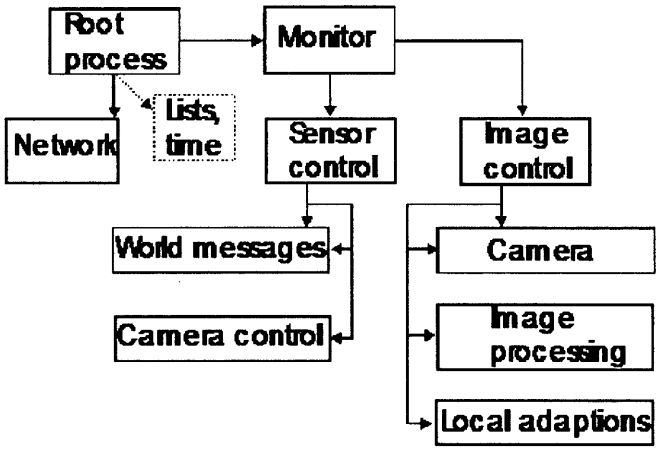


Figure 4.2: Software framework for a decentralized smart camera node [TML+11]

periodic updates of the child thread. In the event of a thread stall or crash, the monitor respawns a fresh thread from the last known state and starts execution on the newly created thread. Note that each camera may be configured as a "camera node", subsequently spawning the image processing thread, or a "control node", which spawns the "camera control" thread.

The two child threads are "sensor control" and "image control". The "sensor control" thread is responsible for communications with the control node, test node, and other camera nodes. It may receive control information from the control node and requests from the test node. If the camera node in question is configured as the control node, the "sensor control" thread may spawn the control node thread,

which becomes responsible for interrogating other camera nodes, disseminating information, and communicating with the world node. Communication with the world node may entail receipt of the aircraft status, position and orientation, and transmission of events generated by individual camera nodes. In debug mode, the smart node may also communicate with one or more test nodes, passing raw image information, camera configuration, statistics and ancillary information.

The "image control" thread is responsible for the general operation of the camera, including acquisition and processing. The acquisition thread is responsible for the configuration of the camera, acquisition of imagery, local storage, synchronization and any logging functions. Images acquired by the acquisition thread are placed in a shared circular buffer. The buffer consists of a data structure (array, linked list) that stores elements (structures or classes) encapsulating the imagery alongside other ancillary information, such as acquisition time-stamp, sensor configuration, processing flag, event list, among others. Each thread (acquisition, sensor control, and processing) maintains its own pointers to the shared circular buffer. The processing thread, launched as a child thread by the acquisition thread, is responsible for the analysis of acquired imagery and the generation of likely events. Updated event lists are tagged for transmission and stored alongside other ancillary data within the shared buffer.

4.3.2 Control node

The instrument has a designated control node, which may be any one of the physical modules. Moreover, control may migrate between nodes. The control node acts as the arbiter between the aircraft and the instrument. It will acquire information from the aircraft and broadcast it to local camera nodes. Similarly, any potential targets spotted by a node will be forwarded to the aircraft system, or to other nodes, via the control node.

The output of the sensor, including detected targets, images, time-stamps and other ancillary data, is made available at the output of the control node. For simplicity during tests, the control node was usually assigned to a specific physical module that did not acquire images, or passed to the "test" node, to minimize synchronization issues. The latter mode of operation was utilized for the field and flight tests.

4.3.3 Aircraft (World) node

The world node provides an interface transition between aircraft systems and the instrument, primarily by encapsulating and isolating the instrument sensor network from other aircraft systems. This computer will be the "aircraft node" of the sensor network. Typically equipped with dual Ethernet cards, it will be a node on both

the instrument network and the aircraft network. Note that despite appearing as a node on the network, the aircraft node is not considered to be a physical part of the collision detection instrument hardware. All information regarding the aircraft, the external world, or operator commands, will flow through the aircraft node. Note that the sensor itself may operate in the absence of an aircraft node, although at reduced functionality.

The aircraft node will only see the control node by design. Software is needed on the aircraft node computer in order to interact with the sensor network. This software may be limited to receiving, parsing and repackaging the aircraft packets for control node consumption. Details of the aircraft packet are proprietary to the Flight Research Laboratory, NRC Canada, and are not provided in this document [Fli].

4.3.4 Test node

The network configuration allocates space for a test node. The test node may interact with any node on the local instrument network, including the control node, and may acquire and record imagery from one or more nodes simultaneously. Unlike the world node, the test node is considered to be a part of the sensor, albeit present only during testing. Hence, like other sensor nodes, the test node can also function as the control node of the camera network.

The test node acted as a convenient debug port in the early prototype variants, allowing raw data to be collected during experiments. A graphical interface may exist on the test node to expedite ease of use and enhance functionality. The test node also acted as a convenient storage point for streamed images during field tests.

4.3.5 Practical considerations

For practical purposes during prototype development, distinct aircraft, test and control nodes were not implemented. Rather, the functionality pertaining to these nodes was merged into a common hardware platform, such as a desktop PC running a commercial operating system. Subsequently, node behaviour was encapsulated within separate threads in a common application framework to maximize performance and minimize latency between "nodes". The application framework was written in a high-level language supporting standard user interface (UI) paradigms. In this particular case, the platform of choice was a Visual Studio C# application running on a Windows XP machine. Implementation details for the individual prototypes are discussed in the remainder of this chapter.

4.4 Laboratory prototype

The design process was iterative, and three different prototype phases are described following: laboratory, ground and field. This particular section discusses the labo-

ratory prototype.

4.4.1 Architecture

The laboratory prototype was configured as a network of five camera nodes in a 5×1 layout, as depicted in Fig. 4.3. Each node comprised of a 2.0 Megapixel Logitech Quickcam Pro for Notebooks sensor module, an OMAP-3530 processor BeagleBoard (Rev. C4) [Bea09], and a Pentax CCTV 25 mm C-mount lens (model C32500KP) [Pen12]. Specifications for the prototype are outlined in Table 4.1, with additional descriptions in [MTW+10]. The use of C-mount lenses required a custom mechanical interface for the detector array and increased the mass and size of each physical node. On the benefit side, the C-mount lenses had high quality and were interchangeable. Focal lengths and focal ratios were readily configurable, with motorized iris versions also available (although not tested). Custom lenses could be utilized provided they conformed to the C-mount mechanical specifications.

The focal length of 25 mm provided a module field-of-view of $\sim$ 8.0 degrees and a geometric pixel field-of-view of 0.2 mrad (larger with the PSF). Shorter focal lengths give more coverage at the expense of resolution. For the tests reported here, high resolution was deemed more important than wider coverage, as it was envisaged that multiple arrays could be utilized to cover the full FAA-mandated field. Although the range performance of the instrument could be improved with

Table 4.1: Prototype Sensor Parameters

Component	Parameter	Nominal Value	Units
Camera group	Number of modules	5	-
	Angular field coverage	45×8	degrees
	Module separation	9	degrees
	Mass	4	kg
	Power	30	W
	Communications	1	Gbps
Single module	Resolution	1600×1200	pixel
	Dynamic Range	8	bit
	Frame rate	5	fps
	Pixel pitch	2.0	μm
	Focal length (f)	25	mm
	F-number ($f/\#$)	1.4–16	-
	Effective FOV	10×8	degrees
Processing	CPU (ARM Cortex 8)	500	MIPS
	Memory (SDRAM)	256	MB
	Storage	8	GB
	Ethernet	100	Mbps

a longer focal length, convenience factors during flight tests, specifically random platform orientation variations, restricted the maximum focal length to 25 mm.

The laboratory prototype was only used in the laboratory. Various hardware and software configurations were evaluated on this platform. Preliminary evaluations and calibration were performed on the in-house laser-scanner dome test facility [WML+11].

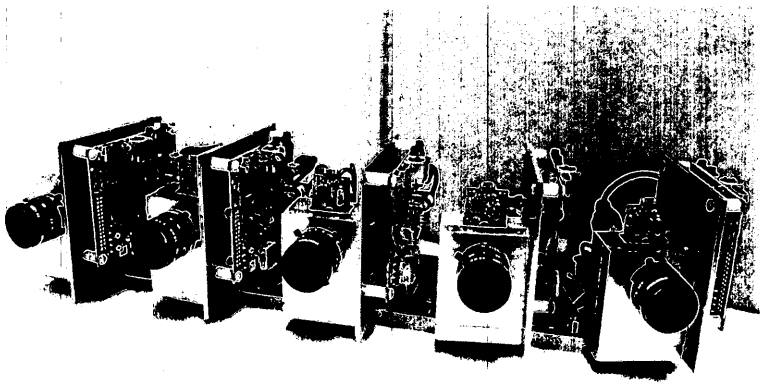


Figure 4.3: First laboratory prototype

4.4.2 Camera node software

The Beagleboard [Bea09] aboard each smart node utilized Angstrom, a light-weight, non real-time, Linux distribution [Nar10]. Angstrom is based on the popular Debian distribution, with packages tailored specifically for the OMAP chipset. The vanilla OS was further streamlined by removing all unnecessary start-up daemons and the X11 graphics interface. The camera capture code was written in C, with the camera itself interfaced via USB, through the UVC (USB Video Class) V4L (Video for Linux) drivers.

The software was identical on each camera node and nodes were activated independently. Each node was assigned a unique IP address that was read from the local memory during startup along with default camera parameters. At the end of the start-up process, each camera node could acquire and store images locally, had

network connectivity, and could respond to control information. The failure of any one node to start or connect to the network did not affect other physical nodes.

4.4.3 Test node software

The test node was a desktop PC running Windows XP. It ran a software graphical interface (Fig. 4.4) written in Visual C# 2010, which was developed at VISOR [MTW+11], and allowed control over resolution and frame-rate.

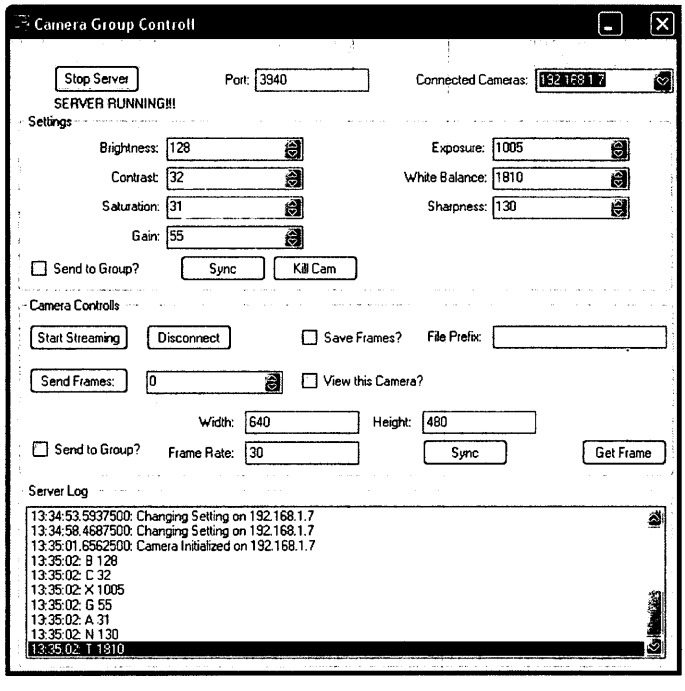


Figure 4.4: User interface of the prototype laboratory test node

Acquisition was fixed to asynchronous user-triggered captures and image streaming (live view) was not implemented. This restriction posed challenges in obtaining

the correct focus, and required trial and error.

4.5 Ground prototype

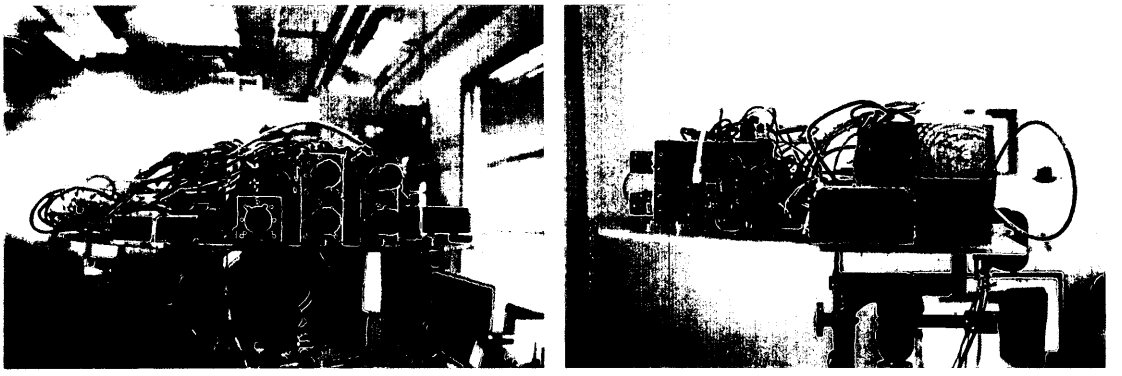
A variant of the laboratory prototype was utilized for ground-based tests at York University. This prototype was taken outside and mounted onto a stable tripod.

4.5.1 Hardware

Five camera nodes were utilized, where each module comprised of a Logitech C300 sensor board, a 25-mm Pentax C-mount lens (model C32500KP) [Pen12], and an Overo Water Gumstix module [Ove10] configured with a Tobi expansion board. The system could be coupled with a mobile inertial navigation system (INS) mounted rigidly to the camera platform. A desktop PC served as the test/control/aircraft node, and interfaced with both the camera nodes and the INS.

Power was supplied by a portable generator daisy-chained to a DC power supply, to simulate the electrical conditions of the host platform. Fig. 4.5 depicts the ground prototype assembled at NRC including the prototype sensor and the repackaged INS. The "piggy-backing" nodes served as spotting scopes, the rationale for which is detailed in Section 5.6.

Preliminary ground tests were conducted at York without a local INS. In this case, white cards served as the target and range was restricted to within 1 km of



(a) Front view (b) Side view

Figure 4.5: Ground prototype instrument at NRC

the sensor platform. Further ground tests were then conducted at the National Research Council with a Bell 206 helicopter serving as the target. Details of the experimental procedure are provided in Section 5.6.

4.5.2 Camera node software

Camera nodes now utilized Ubuntu v10.04 (Maverick Meerkat) specifically tailored for the ARM platform. The transfer to Ubuntu from Angstrom was driven by the availability of a significantly larger software repository and depth of online technical support. Packages such as FFTW, OpenCV and ImageMagick, easily available through the Software Centre, are examples of such convenience.

As noted earlier, the multi-threaded architecture envisioned in the laboratory prototype (Fig. 4.2) was designed to transmit 'hits' or events, rather than raw

imagery. For initial ground and flight trials, however, a much simpler architecture, directed towards maximizing image acquisition, transmission throughput and minimizing latency, was developed. Fig. 4.6 illustrates the software architecture of the simplified camera node. This strategy was aimed at the acquisition of a library of test imagery for the (off-line) development of image processing algorithms. This model was designed as a finite state machine (FSM), ergo it was 'tightly' written, did not utilize threads, and looped endlessly.

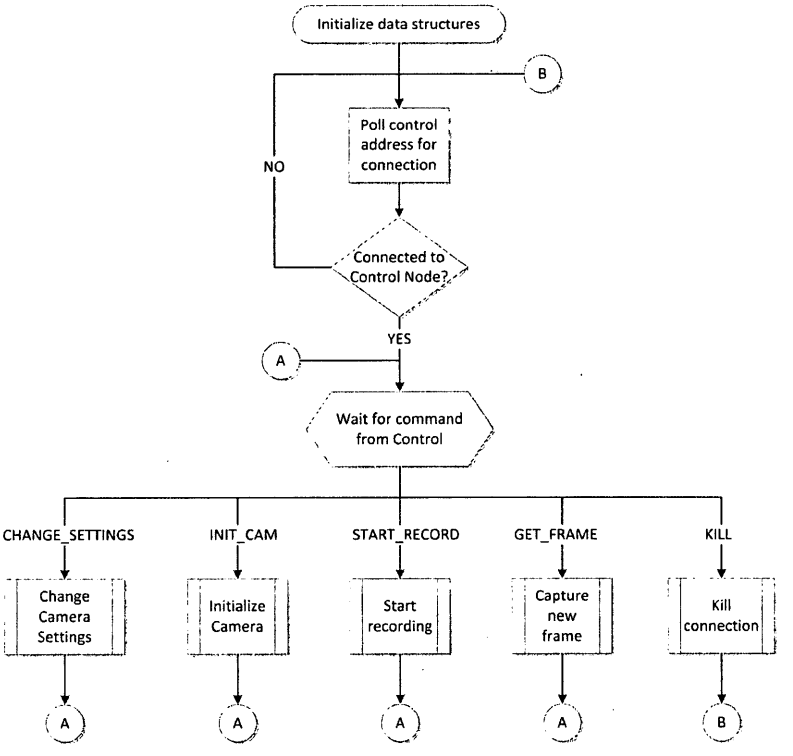


Figure 4.6: Software architecture for camera nodes in ground trials

A flowchart for the acquisition sub-process is presented in the flight prototype

section (Section 4.6.5) to minimize repetition (Fig. 4.9). Others are eschewed for brevity and are available in the technical reports [TML+10] [MTW+11] [TML+11]. A short discussion regarding the legend of the flowchart is deemed necessary for clarity. Although most of the symbols utilized are self-explanatory, the double-boxed rectangles bear an explanation. These rectangles are utilized to interchangeably refer to distinct sub-sections, sub-processes, or threads, depending on the context. An explicit distinction was deemed inutile, as the context is evident or explicitly discussed in every flowchart, and the level of execution did not affect the order of operations performed.

Initial attempts at time-stamping were primitive, with only the data round-trip time, defined explicitly as the interval between the start of a frame request packet and the complete delivery of an image, being recorded. Time-stamp synchronization was performed by setting individual node clocks manually with a stop-watch. This methodology reduced the worst-case precision of the time-stamp to ± 0.5 second, which although adequate for the functionality of the instrument, was time-consuming and prone to error. This was especially true if distinct modules defaulted to different time-zones upon boot, or the RTC battery failed, forcing a clock reset. A simultaneous, multi-node, on-demand synchronization was desirable, and was implemented for the flight test evaluation.

4.5.3 Test node software

The world, test and control node functionality were integrated into one interface, and all were implemented simultaneously on one machine. Two network interface cards (NICs) were utilized, with unique IPs, to distinguish between the aircraft and test interfaces.

The test node interface was rewritten utilizing standard object-oriented paradigms. Development of the test-node was conducted in Visual C#, and utilized version 2.0 of the .NET Framework to maintain compatibility with older systems (Windows 2000). It utilized a multi-threaded architecture, where each camera node was encapsulated by a camera object. Upon starting acquisition, each camera object spawned its own thread and opened its own file handles, with the root test node application monitoring the resource usage between running threads. The software could accommodate hundreds of threads, by extension hundreds of cameras, and was only limited by the processing power and memory available on the underlying hardware.

Highlights of the changes in architecture and functionality are as follows:

- Camera settings for individual nodes provided by test node.
- Complete restructure of the threading model for camera nodes
- Thread balancing and semaphores for access to shared data structures.

- Live streaming of a single camera.
- Merger of test node and control/world node functionality into a common software application.

The primary benefit of a distinct thread-based approach resides in the ability to stop and restart threads on demand. If a node crashes, or otherwise fails to respond, the system may kill the thread, free the data structures, and reopen the port for incoming requests, allowing the camera node to reconnect. Killing the thread will cause the camera node to perform its own recovery and prepare for reconnection. This feature allowed the camera node to be disconnected and reconnected without shutting down the entire camera network, or otherwise changing the connectivity/acquisition status of any other node on the network. The detailed test node software architecture (Fig. 4.12) is presented in Section 4.6.6 in an effort to centralize the description.

4.6 Flight prototype

4.6.1 Instrument hardware architecture

The flight prototype consisted of ten nodes, arranged in a 5×2 configuration, depicted in Fig. 4.7, with parameters detailed in Table 4.2. The sensor board was extracted from a Logitech C600 webcam and integrated with a Pentax 25

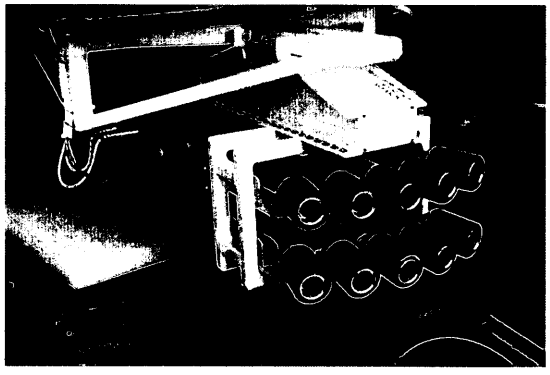
mm C-mount lens (model 32500KP) [Pen12] in a custom mechanical enclosure. Once again, the Overo Gumstix Water module [Ove10] served as the single board computational platform (SBC) on each node. The camera nodes were linked by an Ethernet backplane.

4.6.2 Mechanical

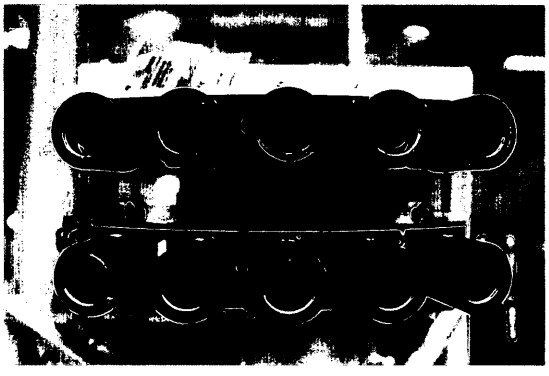
Flight safety is of paramount concern during flight tests. To accommodate safety constraints and reduce costs, the instrument was mechanically partitioned into two sections: a camera grid and a computational box housing an SBC array and a pair of Ethernet switches. This mechanical partitioning did not affect the behaviour of the instrument, which remained electrically equivalent to earlier prototypes. Each SBC functioned as per the flowchart in Fig. 4.8, explained in detail below.

Each "passive camera" enclosure housed the detector array and the lens. Cameras were individually mounted in ruggedized aluminum casings and weather-sealed by UV haze filters. Neutral density and polarizing filters were not utilized in these experiments, although they may be considered for future tests. Individual cameras could be adjusted to ± 10 degrees in azimuth for the desired horizontal overlap. Rows of cameras were adjustable in elevation to ± 10 degrees for optimal vertical overlap. The cameras were fitted to centre the viewing frustum about the apparent horizon when flying at 80 knots. The pitch angles were 0.8 ± 0.1 and -4.3 ± 0.1

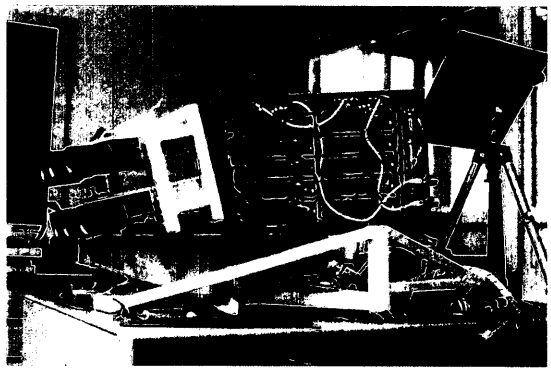
degrees relative to the inertial measurement unit (IMU) for the top and bottom rows respectively. The bulk of the sensor mass was attributed to ruggedization of the frame for safety.



(a) Prototype mounted to the chassis



(b) Front view sans filters



(c) Side view with exposed SBCs

Figure 4.7: The DragonflEYE flight prototype

Table 4.2: **Flight prototype parameters**

Component	Parameter	Value	Units
DragonflEYE Prototype	Topology	5×2	-
	FOV	30×9	deg
	Mass	50	kg
	Power	75	W
	Network	1.0	Gbps
Camera Node	Resolution	1600×1200	pix^2
	Mass	0.5	kg
	Frame-rate	2.5	fps
	Pixel pitch	2.5	μm
	Focal length (f)	25	mm
	F-number	5.6	-

4.6.3 Electrical

Although physically partitioned, the camera configuration was electronically identical to a smart node topology, in that each sensor was mated to its own SBC and could potentially pre-process images. The test, control and world node functionality was encapsulated within one rackmount PC utilizing an Intel Core 2 Quad (Q6600) and Western Digital Velociraptor (10000 rpm) drives. Network interfaces were provided by a quad Gigabit Ethernet interface card by Matrox, of which one link was utilized for the DragonflEYE instrument. Typical bandwidth utilization approached 35% on the Gigabit link during operation.

Two 802.3x (Ethernet) signal interfaces were provided to the instrument via weather-sealed connectors. The first link served as the data link between the in-

strument and the rack-mount (world node) PC, while the second served as a test (and backup) port, and was not utilized during flight. It was possible to bridge the ports if extra bandwidth was required. Communications were handled by daisy-chaining two managed Gigabit switches, allowing the test node to connect at a link speed of 1.0 Gbps, while the smart nodes linked at 100 Mbps. A separate 100 Mbps link connected the rack-mount PC to the aircraft. System throughput was limited to 2.5 fps for synchronization and storage, as the objective of the flight trials was the storage of imagery in real-time synchronized to a GPS time-stamp, and consequently, the precise position and orientation of both host and target aircraft at the time of acquisition. This frame-rate was deemed adequate for the flight tests.

A flight test engineer could interact with the sensor via an in-cockpit LCD display and trackball. Note that the trackball was required as a touch-screen was unavailable. Remote kill switches were installed in the centre console, allowing the flight test personnel to power cycle the test node, instrument camera nodes or intermediate Ethernet switches in case of a software/hardware failure.

4.6.4 Power

A 24 VDC supply, installed into the nose of the aircraft, supplied power to the camera nodes. Each node was equipped with a slow-burn fuse and a local voltage regulator with an input range of 10-40 VDC. Cameras were powered off the

USB port on the SBC. Two Ethernet switches for inter-node communications were powered by a separate aircraft 12.0 VDC rail. Noise isolation was excellent and no effects of aircraft electrical noise were observed.

4.6.5 Camera node software

The camera node software was similar to the version utilized for ground tests (Section 4.5.2). Once again, each node utilized the Ubuntu ARM distribution v10.04 running in console-mode and configured to boot at a custom run-level to eliminate unnecessary startup processes. The OS performs auto-login as the root user at the end of the boot sequence, with the main process configured to automatically launch upon login. There are security concerns associated with this approach, especially if wireless data-links are considered in the future.

Fig. 4.8 illustrates the general state-machine for the camera nodes utilized in the flight tests. Note that it is similar to Fig. 4.6, although there are some distinct changes. The primary change is the introduction of a proper time synchronization scheme, which tightly couples the camera, test and world nodes to the GPS time provided by the aircraft. Other changes include the removal of remote control capability of the camera settings.

Upon launch, the node application (Fig. 4.8) loaded the node IP, control IP, camera settings and calibration data from a text-file. It then initialized basic data

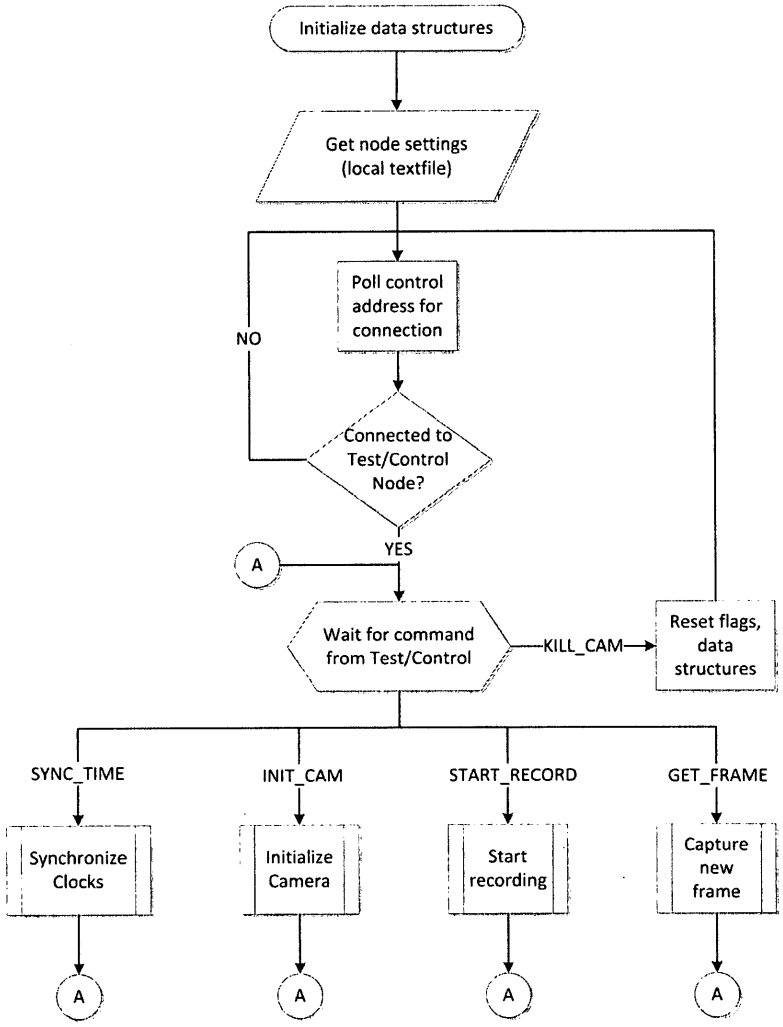


Figure 4.8: Software architecture - Camera node

structures and settled into a loop, periodically checking for a test node at a predefined network address.

Fig. 4.9 flowchart depicts the frame acquisition sequence, while 4.10 flowchart illustrates the time-synchronization procedure. The frame acquisition sequence is linear and straightforward to follow. Two salient points are of note: First, the

camera node acknowledges success by echoing the command packet to the control node. Second, the "additional processing option" in Fig. 4.9 can be a placeholder for image conditioning as well as hit-detection options.

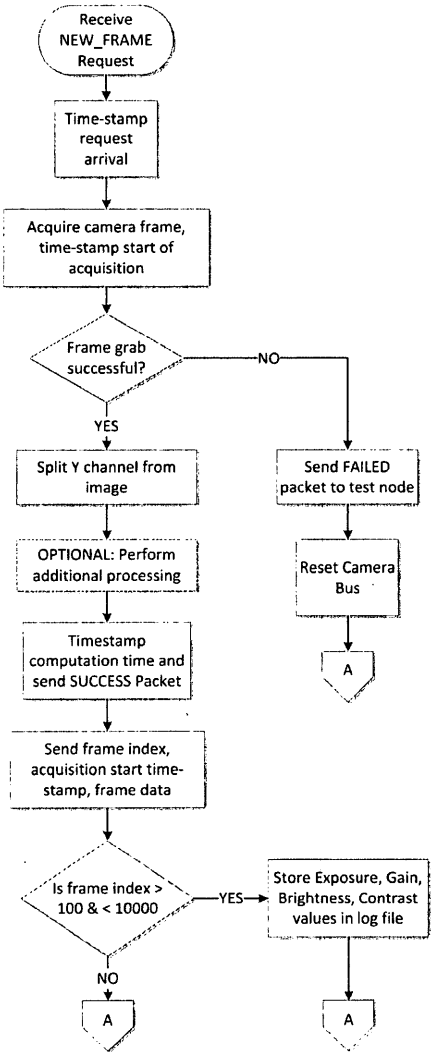


Figure 4.9: Acquiring a frame - Camera node side

The time synchronization scheme in Fig. 4.10 is robust in that it relies on the

offset (in milliseconds) from a common time-base, and not the actual time. In this case, the common time-base is seconds from Epoch, defined in Unix time as January 1st, 1970.

This mechanism of synchronizing time is agnostic of the local system time, time format or time zone configuration. Upon receipt of the test-node offset, each camera node calculates its own offset from Epoch, and then calculates the difference between that offset and the test offset. This difference is stored, and applied as a correction upon each subsequent time-stamp generated locally by the camera node. This new synchronization scheme reduces the worst-case offset from 0.5 s to 10 ms, limited by the single-packet round-trip latency of the system. This error was considered more than adequate for the present flights. However, mechanisms do exist to further reduce this latency, such as averaging multiple packets, repurposing a node as a network time protocol (NTP) server, or providing a hardware timing pulse on the GPIO pin of each SBC supplied by a common source.

4.6.6 Test node software

The test-node software was similar to the framework described in Section 4.5.3. Fig. 4.11 presents the user interface for the test node. The table along the right-side pane displayed the information encapsulated in the inertial navigation system (INS) packet received from the aircraft. Buttons were over-sized to facilitate triggering

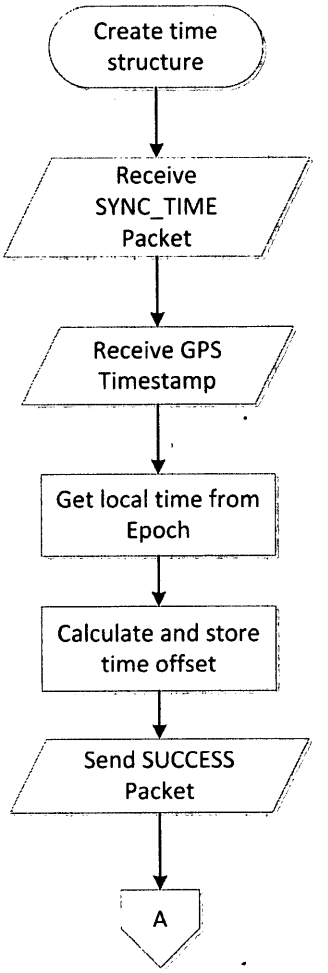


Figure 4.10: Time synchronization - Camera node side

during flights. Note that certain buttons were deliberately under-sized or disabled during in-flight operations (such as "Disconnect" and "View") to prevent accidental disconnects or live-stream generations. All toggle buttons were debounced to 0.5 seconds in software to eliminate accidental multi-clicks.

The application launched upon boot of the rack-mount PC, which could be

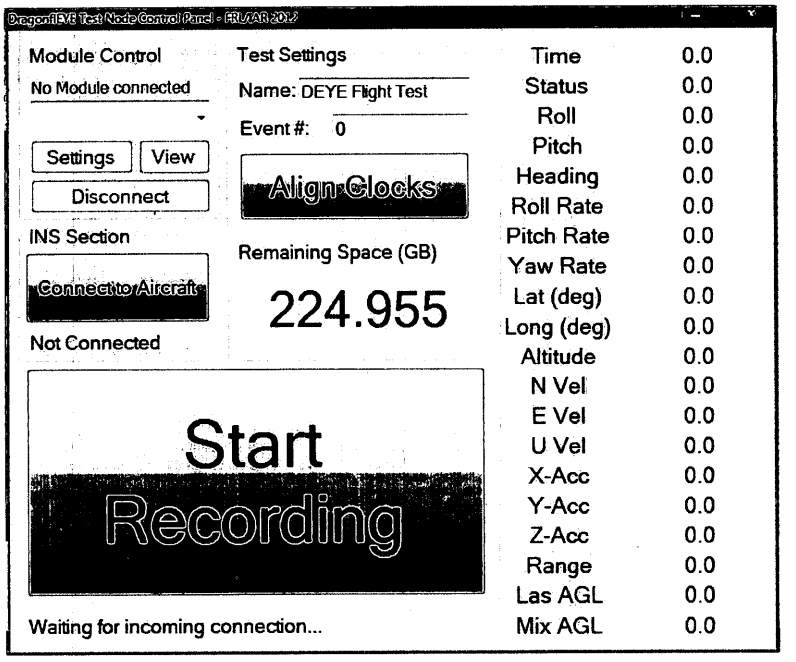


Figure 4.11: User interface (UI) for the flight prototype

rebooted on-demand by the flight engineer, in case of software/OS failure, by a toggle switch embedded in the centre console. Once in flight, steps required to operate the software were minimized to the greatest extent possible, and were as follows:

- Power on test computer. Wait for nodes to connect.
- Connect to aircraft systems.
- Align clocks.
- To record an event, click on "Start Recording". Event number auto-advances. Button changes to display "Stop Recording".

- Upon completion, click on Stop Recording. The button changes to say "Start Recording".

Manual control over image recording was a desired feature, since much of the flight time was utilized to position the aircraft on collision trajectories. Fig. 4.12 depicts the test node architecture. Note that it is considerably more complex than the camera node. Presented following are the specific threads that are launched from the main test-node application.

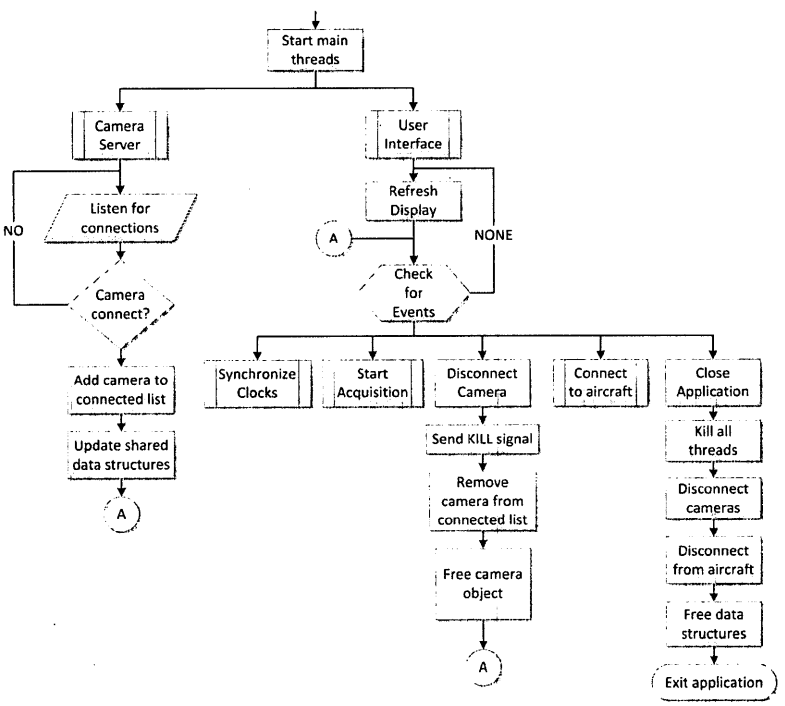


Figure 4.12: Software architecture - Test node

4.6.6.1 Time synchronization

Time synchronization is essential from an experimental perspective to obtain an image registered to a world position for any subsequent analysis. Fig.4.13 illustrates the sequence of steps utilized to sync a camera node. This sub-process is balanced by its counterpart in the camera node (Fig. 4.10). Note that the key feature of the approach employed is the broadcast packet, which scales the system to any number of cameras, as long as they reside on the local sub-network.

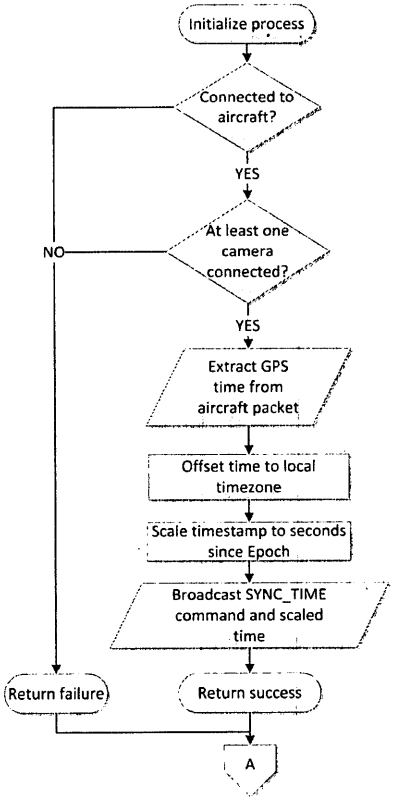


Figure 4.13: Time-stamp synchronization - Test node side

The "Align Clocks" button is always available and enabled in the UI, allowing time to be asynchronously synchronized. It is recommended that time be synchronized between successive collision vectors to minimize clock drift errors, although this error was observed to be minimal compared to other systematic errors.

4.6.6.2 Start and stop recording

Fig.4.14 depicts the sequence of steps followed by the main process in anticipation of a recording session, in particular the generation of threads based on a camera connection list. Note that cameras disconnected by the host will always automatically reconnect as per their finite state machine. The disconnect effectively acts as a software reboot for the camera node application (see 4.8).

Also note that the "Start Recording" button mirrors a "Stop Recording" state that is not depicted in the flowchart in the interest of brevity. Its main task is only one: Set the global recording flag to false. This immediately stops all camera threads as per their state machine.

Fig.4.15 describes the state machine of a camera thread upon the start of a recording. The recording threads perform independently and asynchronously, with the host process keeping track of all threads. This behaviour is desired as it is flexible and scalable to an arbitrary number of cameras, with the only limitation being the computing resources of the test node computing hardware. Although

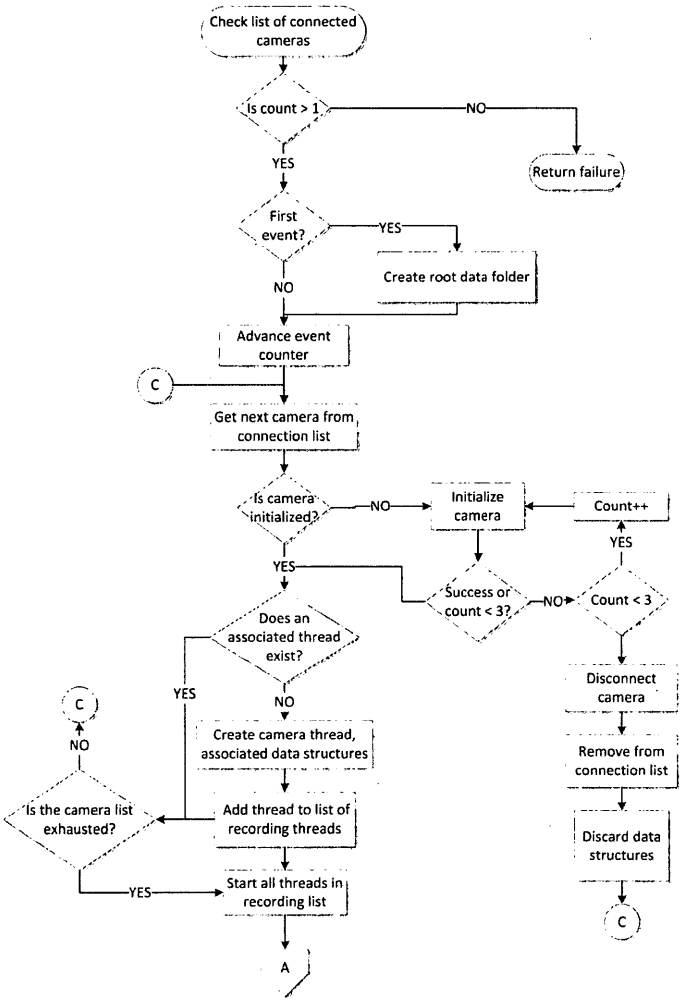


Figure 4.14: Sequence of operations when recording starts - Test node

all threads operated with equal priority under the present scenario, thread prioritization is possible with this framework, such as the emphasis of forward-looking cameras at the expense of side and rear-view modules.

Each thread, once spawned, initializes the necessary data structures and opens a handle to a log file. It then waits for the a global "recording" flag to be set to

commence operations. The global flag ensures that all threads start recording at the same time.

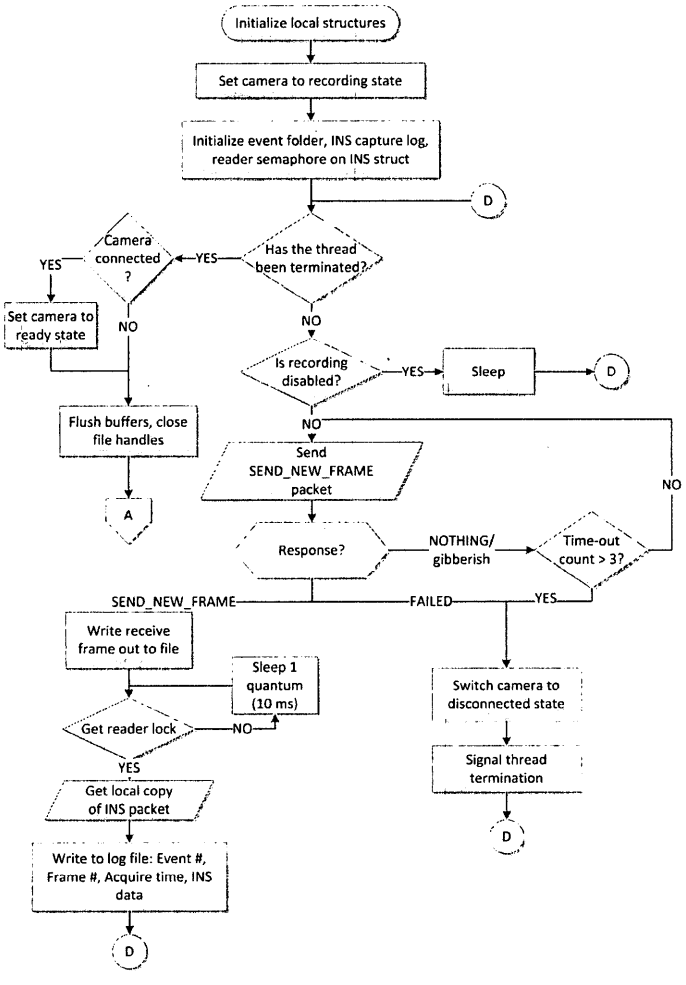


Figure 4.15: Acquiring a frame - Test node side

Note that a camera thread interacts directly with a physical camera node through a `CAMERA` object passed by the test node host process. This `CAMERA` object encapsulates the connection socket, acquisition buffers and other local camera

structures. The size of the frame-buffer is assumed and fixed beforehand, and not communicated by the camera. This is strictly for performance reasons as the buffer copy operation is a key bottleneck and can be heavily optimized for a known image size.

Information regarding the INS packet acquisition and storage is curtailed for proprietary reasons. One notable point is that the INS thread utilizes a `Writer` lock (as a mirror to the recording thread `Reader` locks) to access the packet data structure. This prevents partial copies and maintains general data structure integrity. Note that multiple `Readers` may access a `Reader`-locked resource whereas only one `Writer`, to the exclusion of all other processes, may acquire a `Writer` lock at a time.

Although the aircraft navigation system delivered packets at a rate of 400 Hz, the final packet capture rate was chosen to be 50 Hz, as testing proved it to be a good balance between data rate and camera thread starvation. Note that this choice introduces an additional time-stamp error as the worst-case difference between GPS time arrival and frame acquisition time is 20 ms. This, combined with the worst-case network latency of 10 ms, computes the worst-case error to 30 ms. If assuming a closing rate of 100 m/s, this resolves to an error of 3 m in the position estimate of each aircraft, for a combined worst-case separation error of 6 m. This was considered adequate for the experiments. In our analysis, we assume a larger error

of ± 10 m to maintain a safe buffer, while accounting for the "unknown unknowns". In comparison, the GPS positional accuracy was deemed to be < 1 m.

4.6.7 Limitations

Optimizing thread performance in a multi-threaded .NET application proved to be the most challenging aspect of the test-node software. A camera thread needs to be able to write to sockets it may not be the creator of, write to the filesystem, and allow self-termination as well as remote-termination within the same encapsulation.

The `BackgroundWorker` model provided by the .NET framework was originally utilized for camera threads in the laboratory environment. However, this approach was marred by performance constraints. Threads created by `BackgroundWorker` cannot be terminated by the calling thread and must self-terminate. In addition, `BackgroundWorkers` do not allow explicit thread prioritization and are considered, by default, to be low-priority processes to guarantee UI responsivity. In the test-node application, the threads perform the bulk of the work, and may prioritize themselves at the expense of UI updates.

The .NET `Thread` object was utilized instead. It is fully functional Thread object available in managed and unmanaged versions. It offers prioritization, encapsulation and remote management via delegates, but is more code-intensive to implement and manage, as all interactions require encapsulation and delegation.

This thread model performed significantly better, and scaled well with available hardware.

Other highlights are summarized as follows:

- The local acquisition buffer per node did not always completely flush. New events would sometimes contain a few images from the previous event. This bug is caused by the interaction between dictionary structure and the frame-buffer within the acquisition pipeline. The dictionary structure is a remnant from ground tests as a provider for image cycling functionality. To compensate, record stamps are adjusted after the fact to align properly with acquired imagery. This bug still exists in the code at the time of writing,
- Windows limits capture timer performance to 100 Hz (10 ms). Utilization of the high-performance (multimedia) timer requires a spin-lock, which negatively impacts the performance of a multi-threaded application and can arbitrarily freeze the UI. Therefore, even though the aircraft navigation system delivered packets at a rate of 400 Hz, capture could only occur at a maximum rate of 100 Hz, at the expense of all other threads (UI freeze, for instance). Therefore, the aircraft packet was limited to a 50 Hz acquisition rate.
- All data transfer between the test-node camera thread and the camera node software was conducted over a single TCP/IP socket. Although this approach guaranteed zero lost packets, camera throughput was negatively impacted.

4.7 Conclusion

In summary, this chapter describes the evolution of instrument prototypes, from a laboratory test-bench, scaling up to the flight prototype flown at NRC Ottawa. Hardware features were limited by cost, mass and power constraints, and suffered primarily from cost at the expense of greater integration. Potential designs had to be reconfigured in the light of flight safety requirements. The software ecosystem is demonstrably flexible and scalable, allowing a large number of cameras to connect and operate simultaneously, limited only by the available processing power and network bandwidth.

The path to the future is tied to greater integration of smaller, cheaper cameras, such as integrated cell-phone modules, with lighter lens assemblies. Utilization of hierarchical processing at the node level will minimize bandwidth requirements, while introduction of distributed processing at the node level will allow for decentralized schemes such as the 'virtual camera' paradigm.

5 Performance Evaluation: Simulation and ground tests

5.1 Synopsis

This chapter discusses modeling activities and ground tests in support of the DragonflEYE instrument. Models were tested within an in-house simulation environment, while characterization was done on a custom, in-house, "dome" optical test facility. The simulation environment provides a convenient model of the scenarios of interest, and subsequent ground tests support and validate the simulation results. Ground tests were also conducted at the National Research Council's flight test facility in Ottawa, where instrument functionality and flight trial methodology were confirmed for future flight tests.

Practical, measurable definitions for the key figures of merit (FOM) discussed in 3 are also presented. These definitions apply to the subsequent chapter on flight tests (Chapter 6) as well. Work published during the course of the evaluation is

referenced as appropriate.

5.2 Introduction

The use of simulations to model the test environment can be an important aspect of experimental work, as simulations can be adequate representations of the scenario while retaining the flexibility to evaluate a wide variety of parameter configurations. Comparison of simulation work to real-life field measurements is an important step in determining the feasibility of the instrument. As such, laboratory tests for characterization and ground tests were utilized to observe the behaviour of the instrument in simple but representative environments. The discussion on flight tests is deferred to the following chapter.

5.3 Definition of analysis figures of merit (FOM)

Before beginning the discussion on modeling efforts, it is useful to outline the desired measurements designated as the output of the modeling activities. In this section, various figures of merit (FOM) for data analysis are described. Table 5.1 lists these parameters, along with their salient characteristics.

The explicit quantification of the key metrics outlined in earlier sections (Sections 3.3, 3.7 and 2.4.4) is presented prior to the analysis.

Table 5.1: Experimental Aircraft Flight Specifications

Symbol	Description
SNR_e	Pixel-equivalent signal-to-noise ratio, excluding I_{tgt} fluctuations
R_0	Range at which the target is first observed
T_0	Time to impact
$FWHM_{tgt}$	Target size at R_0
$C_{back}(R)$	Contrast between target and background at R_0
$C_{tgt}(R)$	Intra-target contrast at R_0
$h_{tgt}(R)$	Apparent target proximity to the horizon
χ_{proc}	Multiplicative factor increase in R_0 by image processing

5.3.1 Empirical SNR - SNR_e

An expression for SNR, SNR_t, was presented in Chapter 3 (Eq. 3.23). However, it is not readily measured for complex target objects. Therefore, Eq. 5.1 depicts the formulation of SNR_e, which was developed as an empirical equivalent of SNR_t from acquired test imagery. Even though the target and background had spatial extent, SNR_e is structured to look like a single-pixel event against a uniform background, thus forcing the detection problem into a standard format [Kop98].

$$SNR_e = \frac{|d_s - d_b|}{d_n} \tag{5.1}$$

Here, d_s is the equivalent target pixel intensity value, while d_b is the local background signal estimated at the target pixel. Both d_b and d_n were, to first order, independent of range between target and host for the ranges of interest. The absolute value operator forces SNR_e to be non-negative even though the effective

target signal might be weaker than the background — a common occurrence. This empirical technique did not require prior knowledge of the target, the scene or other imagery, generated reproducible detections and was insensitive to specular reflections or other signal dynamic range constraints.

For each image in which a target was expected, its signal, d_s, was defined by the following procedure. First, target locations were manually selected offline in each frame. Next, a region of interest (ROI) was defined, centered about the selected location. Then, the pixel location corresponding to the "most-visible" target feature was extracted from this ROI, explicitly defined as the pixel with the largest variation from the ROI mean. This operation assumed that the pixel with the greatest contrast was most prominently visible to the human operator and was intended to reduce selection errors. The target signal, d_s, was then computed by taking the mean of a 3×3 window centered at this "most-visible" target feature. This second operation was an empirical procedure intended to smooth the complex, disjoint radiance profile expected for the target aircraft used in flight trials, but came at the expense of location precision.

The local background signal, d_b, was calculated as follows. An annular ROI was selected around the target region, and a mean pixel intensity was computed. In general, the use of median gave the same result as mean. When this annular ROI contained the horizon, only the segment on the same side of the horizon as

the target was used. In the head-on case, the target was usually above the horizon and was seen against a sky background.

An estimate of the empirical random noise, d_n, was computed by differencing temporally adjacent local background annuli. Then, the standard deviation of each difference image was computed. This differencing procedure was intended to eliminate scene structure. Fig. 5.1 is a histogram of noise over a sequence of difference images acquired from one trial. This empirical d_n includes contributions from read noise, shot noise, reset noise, fixed pattern noise and photo-response non-uniformity. Each of the individual contributions was estimated to be < 1 in units of "digital number." Note that σ_{tgt} has been explicitly excluded from d_n by using the background ROI. Because d_n is not expected to change significantly with range, spikes due to noise anomalies were reduced using a moving average.

5.3.2 Validation of the working SNR_e

The rationale behind this practical SNR model is driven by the following observations. The incident irradiance (target or background) and the digital signal are linked by the sensor transfer function, which is linear. Therefore, d_s and d_b are connected to i_t and i_b by proportionality constants k_s and k_t, encapsulating all intermediate transfer mechanisms, such that $d_s = k_s i_t$ and $d_b = k_b i_b$. In addition, although the system transfer function is different for target and background signals,

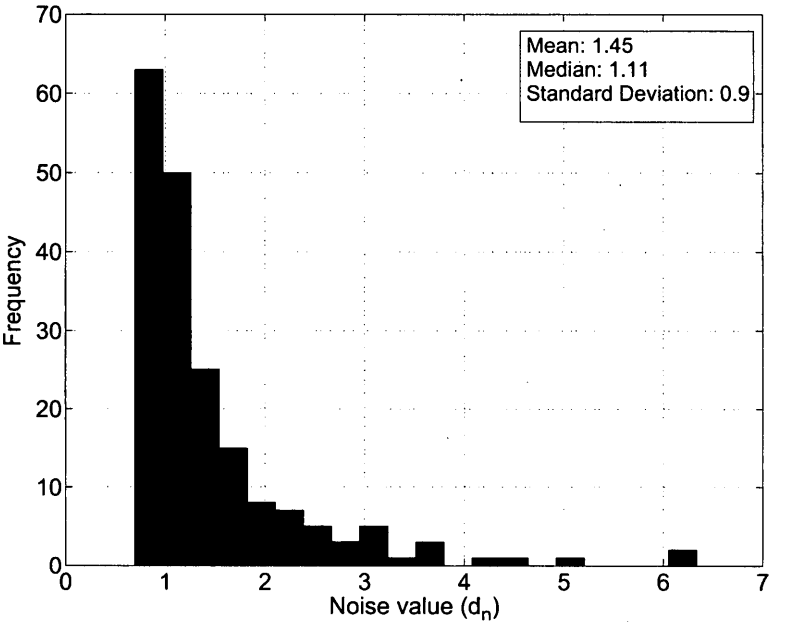


Figure 5.1: Histogram of noise estimates from one flight trial

the *sensor* transfer function is common, as it is a function of the optics and the detector. Furthermore, any noise sources attributed to the sensor are included in the measured digital number.

The functions, SNR_e and SNR_t, exhibit identical behaviour at the limits, with both functions approaching zero at infinite range (Eq. 5.2), and reaching saturation as the target approaches zero range (Eq. 5.3).

$$\lim_{R\to\infty} SNR_t(R) = \lim_{R\to\infty} SNR_e(R) = 0 \tag{5.2}$$

$$\lim_{R\to 0} SNR_t(R) = C, \lim_{R\to\infty} SNR_e(R) = kC \tag{5.3}$$

Saturation, however, is reached much earlier, once the target fills the pixel. Note

that the values may be separated by a proportionality constant k related to the sensor transfer equation. In the limit where $\frac{d_s}{d_b} >> 1$, SNR_e reduces to the conventional interpretation of SNR. Specular reflectance may sporadically cause spikes in i_t, which should be observable in the digital number d_s. Tilts of the host and target platforms affect signal localization, but not signal strength, assuming that inter-pixel errors such as fixed pattern noise (FPN) and photo-response non-uniformity (PRNU) are minimal and/or partially self-corrected within a commercial sensor (correlated double sampling CDS).

5.3.3 Range at first detection - R_0

The range at first detection, R_0, is explicitly defined as the range value at which the empirical SNR crosses a predefined detection threshold, S_{th}. As a figure of merit, R_0, has the benefit of a simple link to measured quantities, and its random error can be precisely determined from the uncertainties in the fit parameters of the fitted SNR_e curve. For flight trials, a complete set of SNR_e values against range values is obtained. The estimate of R_0 can then be determined from the fitted curve of SNR_e, reducing errors.

Let us define a function $SNR_e = f(R|\vec{\mathbf{a}})$ that is a fit to the empirically acquired SNR values. The independent variable R signifies range. The vector $\vec{\mathbf{a}}$ contains n fit parameters, enumerated as $[a_i, a_2, \ldots, a_n]$. Ideally, $n << N$, where N is the number

of samples utilized to perform the fit, allowing a_i to be computed simultaneously in an over-determined, least-squares fashion. Similarly, the excellence of the fit is determined by a commonly used fit quality criterion, such as the R-squared value or others as necessary. If $f(R)$ is a good fit, the parametric uncertainty in $\vec{\mathbf{a}}$ will be small.

Furthermore, let us assume that $f(R)$ derived in this fashion is a constrained continuous, invertible function over the ranges of interest. Then, the range at first detection, R_0, and its associated error, δR_0, can be quantified as per Eqs. 5.4 and 5.5 respectively.

$$R_0 = f^{-1}(S_{th}|\tilde{\mathbf{a}}) \Big|_{SNR=S_{th}} \tag{5.4}$$

$$\delta R_0 = \sqrt{\sum_i^n \left(\frac{\partial f(R)}{\partial a_i}\delta a_i\right)^2} + \delta R_{sys} \tag{5.5}$$

Note that Eq. 5.5 is derived from the generic calculus of errors and calculates the aggregate error in R_0 from the contributions of each fit parameter a_i. Additional systematic errors may be present, and are denoted as δR_{sys}. These systematic errors account for network latencies, time-stamp capture and storage offsets, as well as other quantization errors inherent in any digital system.

5.3.4 Time to impact - t_0

The time-to-impact is a corollary to range at first detection if the closing speed (v_c) is known. Note that t_0 may be more important than R_0 from the perspective of regulations, which are often defined in miss distances and times to collision (Section 3.3). Given the prior assumptions on $f(R)$, t_0 and the associated error can be computed from Eq. 5.6.

$$t_0 = \frac{R_0 - R_m}{\overline{v_c}} \tag{5.6}$$

Here, v_c is the closing rate measured in appropriate units, while R_m represents the minimum "miss" distance. In typical flight nomenclature, two aircraft are considered to have collided if they pass within a minimum threshold distance of each other. This distance, which correlates directly with the "inner fence" concept in Chapter 3, represents an offset from R_0 which must be taken into account before calculating t_0. Note that this analysis only applies if both aircraft travel at a relatively fixed airspeed. The error in t_0, denoted as δt_0 can be calculated in similar fashion to R_0 (Eq. 5.7).

$$\delta t_0 = \frac{R_0 - R_m}{\overline{v_c}} \tag{5.7}$$

$$\therefore \delta t_0 \simeq t_0 \cdot \left.\frac{\delta R_0}{R_0}\right|_{v_c}, \quad \text{where } R_0 >> R_m \tag{5.8}$$

The general formulation in Eq. 5.7 can be simplified to Eq.5.8 by noting the following: Firstly, $\overline{v_c}$ represents an average value of the closing rate that is derived

completely from the integral of range values over a fixed period of time, therefore $\delta v_c = 0$. In addition, R_m is a constant, and for all cases under consideration, $R_m << R_0$, as close-range detections are unimportant. Furthermore, additional systematic errors are already encapsulated within the systematic error component δR_{sys} in δR_0 and do not require inclusion.

5.3.5 Apparent angular target size - (σ_x, σ_y), ($FWHM_x$, $FWHM_y$)

At R_0, the angular size of a typical aircraft is expected to be a few pixels in the image. This constraint, as noted in Chapter 2, was used in the selection of optical hardware. The observed pixel size is further increased by the effective system point-spread function. A Gaussian function well-approximates the primary lobe for many typical PSFs. To measure it experimentally, a 2-D Gaussian distribution function was selected to approximate the apparent angular size of the target, denoted by a a non-degenerate bivariate Gaussian distribution, $\mathcal{N}(x, y)$ in Eq. 5.9, with ρ as the covariance.

$$\mathcal{N}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right)} \tag{5.9}$$

If the x- and y- axes are assumed to be independent and identically distributed (i.i.d), ρ resolves to unity, simplifying the Gaussian distribution to $\hat{\mathcal{N}}(x, y)$ (Eq.

5.10). Note that certain optical aberrations, such as distortion and astigmatism, may limit the validity of the independent axes assumption.

$$\hat{\mathcal{N}}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}\right)} \tag{5.10}$$

A common score for evaluating the PSF is the full-width at half-maximum (FWHM) metric, which is related to the standard deviations (σ_x, σ_y) of the 2D Gaussian fit by a constant (Eq. 5.11). Although multiple factors contribute to the spatial extent of the point-spread function (as established in Chapter 3), asymmetry in the PSF can be attributed primarily to target, optics and detector asymmetries. Note that the ratio, $\frac{\sigma_x}{\sigma_y}$, can be established by detector pixel geometry or image post processing.

$$FWHM_x = 2\sqrt{2ln(2)}\sigma_x \quad \text{and} \quad FWHM_y = 2\sqrt{2ln(2)}\sigma_y \tag{5.11}$$

The calculations are employed in Section 6.11.3, where the approximate target size is measured for raw imagery collected from head-on collision runs.

5.3.6 Inter-target contrast - C_{back}

The "inter-target" contrast ratio (C_{back}) is used as a measure of the target's visibility against the local background. The measured value (Eq. 5.12) assumes a linear transfer between the incident irradiance and the digitized value for both target and

background signals, such that $d_s = ki_t$ and $d_b = k'i_b$, where k and k' are constants determined by the sensor transfer function.

$$C_{back}(R) = \frac{d_s - d_b}{d_s + d_b} \tag{5.12}$$

Here, d_s and d_b refer to target and background signals respectively. Note that, although not explicitly stated, parameters d_s, d_b, i_t and i_b are functions of the range R, whereas k and k' are unaffected by range. When there is a localized target on a uniform background, the relative importance of different spatial frequencies can be affected by MTF and post-processing. Note that the derivation of C_{back} coincides strongly with the operating definition of SNR_e, and therefore, measurement of C_{back} explicitly is expected to contribute little. In particular, if the impact of noise is relatively constant, $C_{back} = SNR_e$ within a scale-factor.

5.3.7 Intra-target contrast - C_{tgt}

Alongside inter-target contrast, the "intra-target" contrast ratio (C_{tgt}) is also useful as a metric for evaluating the degree of non-uniformity in the target signal. Once again, the measured value (Eq. 5.13) assumes a linear transfer between the incident irradiance and the digitized value for target and background signals, except in this case, $k = k'$ for both quantities.

$$C_{tgt}(R) = \frac{\max(d_s) - \min(d_s)}{\max(d_s) + \min(d_s)} \tag{5.13}$$

Here, $\max(d_s)$ and $\min(d_s)$ refer to maximum and minimum detector response within a target footprint, respectively. Recalling the target non-uniformity profile in Fig. 3.6, approximations to the maximum and minimum signals are denoted by the peaks and troughs in the cross-section plots. Note the local maximum and minimum can be spatially disjoint, and therefore must be computed for intensity values over the entire region of interest.

Note that, as mentioned earlier, at long ranges, target details will be mixed, resulting in reduced intra-target contrast. Specular reflections, solar orientation, target motions, and platform motions, all contribute to the observed profile. The shape of the radiance profile and the value of C_{tgt} are expected to provide useful classifiers for detection-based processing activities.

5.3.8 Image processing advantage - χ_{proc}

Given an image processing algorithm, it is useful to have a performance metric to measure its efficacy. Although various metrics are available in literature for the performance determination of algorithms, we choose an empirical "image processing advantage" (χ_{proc}) as a straightforward metric that compares the R_0 achieved with two processing alternatives, as per Eq. 5.14,

$$\chi_{proc}(A, B) = \frac{R_0(A)}{R_0(B)} \tag{5.14}$$

where R_0 is denoted as a function of the processing techniques A and B. This parameter can be computed whenever a set of images as a function of range is available.

5.4 Simulations

5.4.1 Rationale

Simulations are useful for determining the ballpark for various parameters of the instrument design (focal ratio, focal length, integration time). Realistic values can enhance the utility of results derived from the models. Instrument and scenario parameters are defined and discussed in Chapters 2 and 3. Simulations are also a cost-effective technique for providing guidance for detailed configuration of the prototype instrument during the flight tests.

For the present case, modeling was used to justify the general over-design of the sensor modules across prototype iterations, such that small differences between targets and background signals could be detected. Some of the "over-design" parameters included focal ratio, frame-rate and computing power, all of which increased the size, cost and power consumption of the prototype sensor.

In particular, the pixel field-of-view (FOV) was made smaller than the design requirements to allow for pixel sub-sampling, with the desired FOV achieved by

increasing pixel count, at the expense of computation time. Similarly, the frame-rate was reduced to improve computation time and storage. Flexibility in the modelling environment meant that the requirements could easily be reversed (high frame-rate, low pixel count), by simply adjusting the parameters.

5.4.2 Simulation environment

Simulations were conducted in an in-house MATLAB ray-tracing environment. The ray-tracer depicted realistic geometric behaviour of the target, albeit with some limitations. An Earth-centred Earth-fixed (ECEF) coordinate reference frame was utilized. Level flight in an ovoid earth model establishes a curved trajectory, where distance to the target is given by the arc length. At the distances (10 km) and altitudes (1000 ft) considered, corrections due to ovoid Earth were expected to be small, with terrain deformations and man-made structures playing a larger role. Fig. 5.2 illustrates a sample ray-trace with an embedded target, with a zoomed-in highlight depicted in Fig. 5.3.

The output image accurately captures the variations in a reference image, while allowing localized targets to be added. Fig. 5.4 depicts the user interface governing the simulation environment, not all of the parameters were exposed in this interface. Some of the key parameters that could be varied included the sensor resolution, pixel pitch, focal length, gain and quantum efficiency, as well as target size, reflectance

Figure 5.2: Typical output of the in-house MATLAB raytracing environment, with the embedded target highlighted in red.

(specular, diffuse), colour, distance from the camera and the vector of travel.

The light source was modeled as both a point- and an ambient- source, reflecting the dual nature of the sun and atmospheric scattering. However, although the atmosphere could have been modeled as a particle cloud, it was not as the computational expense was too extreme. Targets were modeled by spheres. Standard values for the detector quantum efficiency, gain and pixel pitch were assumed. Pixels in the imaging array were assumed to be unaffected by fixed-pattern and photo-response non-uniformity artifacts.

The simulation back-end is depicted in Fig. 5.5 as an object hierarchy. Key

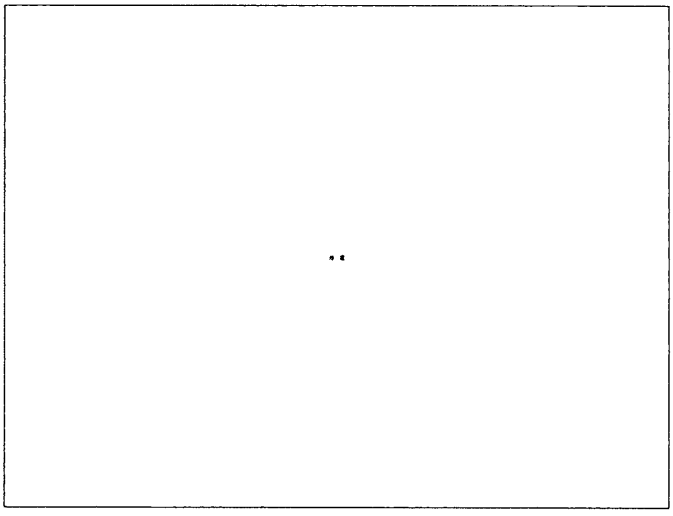


Figure 5.3: Zoomed-in version of the simulated target.

highlights of the simulation back-end include:

- Multiple camera modules facing arbitrary directions.
- Multiple targets flying along arbitrary tracks.
- The inclusion of pre-rendered footage (or live footage) as the background.
- Parallel processing for faster rendering, scalable to an arbitrary number of cores (minimum three required).

The background model was commonly shared among all modules, and consequently all threads. Note that the parallel computational toolbox was utilized to accelerate the rendering of images on available local and network CPUs. Unfortunately, the CUDA and OpenCL libraries were not sufficiently mature at the time

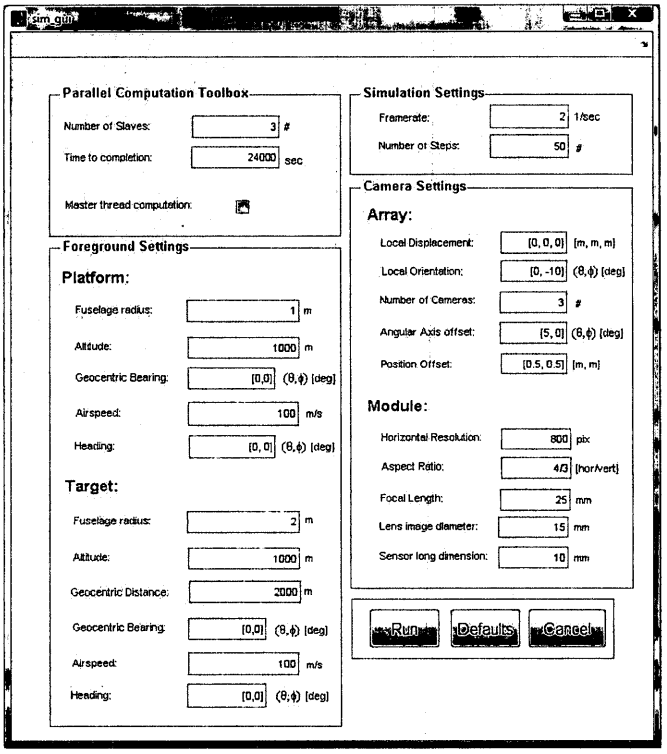


Figure 5.4: Simulation environment console with configurable parameters.

of development (2008), and graphics processing unit (GPU)-based parallelization was not implemented.

The initial framework assigned a thread/process per camera, however, this proved to be a poor utilization of resources if the number of available processors did not match the number of camera modules. Better performance was achieved by block image processing, whereby each image was segmented into a number of disjoint (overlapping) segments, with each segment assigned to an individual pro-

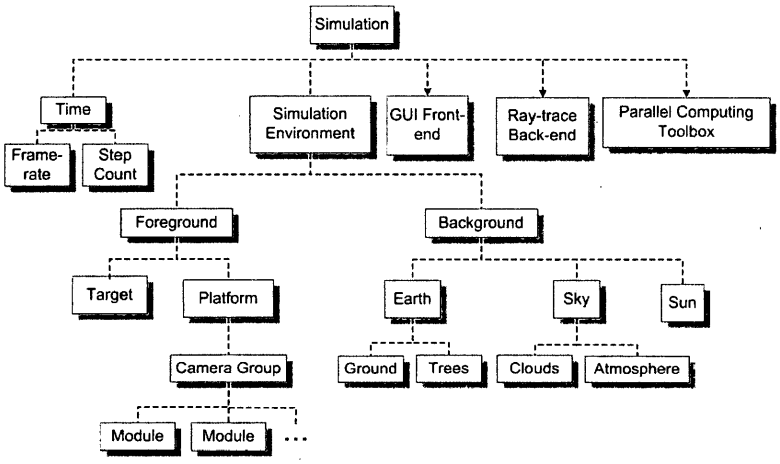


Figure 5.5: Simulation data structures.

cess.

5.4.3 SNR with respect to local contrast

Fig. 5.6 shows the variation of SNR for a single pixel as a function of target range for different background reflectivities as per Eq. 5.1. The images used were generated via ray-tracing synthetic targets at realistic distances embedded into flight imagery (c.f. 5.3). Ambient lighting was controlled to simulate different background intensities.

In Fig. 5.6, the modeled noise is primarily sourced from the ambient illumination shot noise. The SNR values require the selection of numerous instrument parameters, such as integration time, optical throughput and quantum efficiency, and so can best be considered as relative. These parameters usually affect both

signal and background equally. When appropriate, the parameters are germane to a potential prototype. Note that the signal can be brighter or darker than the background. When signal and background have the same reflectivity, the signal disappears. In this situation, the use of spectral or polarization features of the target and background may be important.

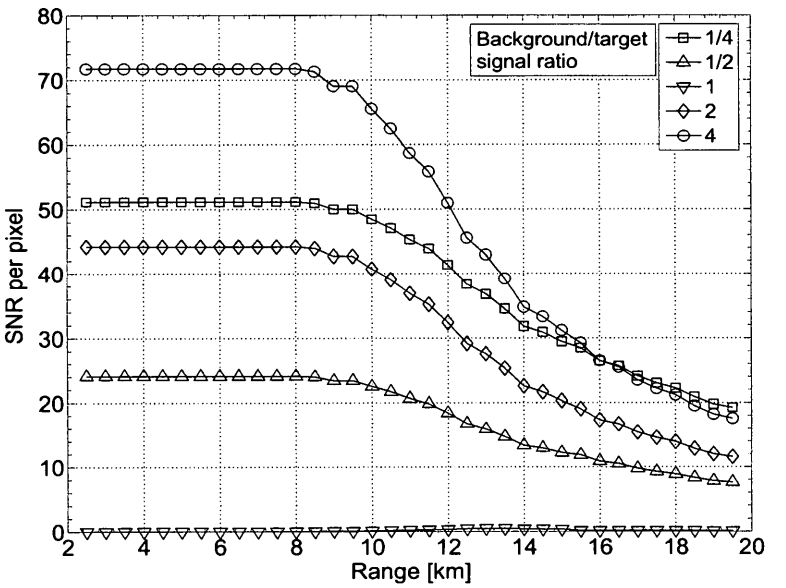


Figure 5.6: Variation of SNR with range for a sequence of background reflectivities, with saturation indicating a pixel filled with target.

An effective target area of 1.0 m^2 has been used for modeling. At sufficiently long range, the target is small compared to the pixel footprint (A_{foot}), and most of the signal (and noise) comes from the background. For nearby targets, the entire pixel is filled with light from the target. At intermediate ranges, the signal depends

on the difference in reflectivity between target and background. The variation of signal with range arises from the increase in occlusion of the background by the target, until A_{foot} is completely filled by the target. Note that complex, large targets may not have smoothly continuous behaviour. In cases where the two signals are sufficiently similar, the target disappears into the background at all ranges. The behaviour of the model was deemed satisfactory at ranges out to at least 10 km for realistic choices of model parameters.

5.4.4 Estimating the point-spread function (PSF)

The curves in Fig. 5.6 are idealized as the system point-spread function is absent. Fig. 5.7 displays the representative point spread functions (PSF), some of which are modeled, while others are acquired from typical representative experimental data. The camera in this case assumed a lens focal length of $f = 12.5$ mm, with an aperture setting of $F\# = 2.8$, and a pixel pitch of 3.5 μm. The optical PSF (black) is modeled from camera and lens parameters. The PSF of the optics/detector has a full width at half maximum (FWHM) of ~ 2.0 pixels, which includes diffraction, aberrations, camera focus and detector array sampling.

The motion broadening in Fig. 5.7 has been determined using the measured jitter in the yaw axis of the surrogate UAV in level flight at an inbound velocity of 90 knots. Displacement of a point in the image for an integration time of 75

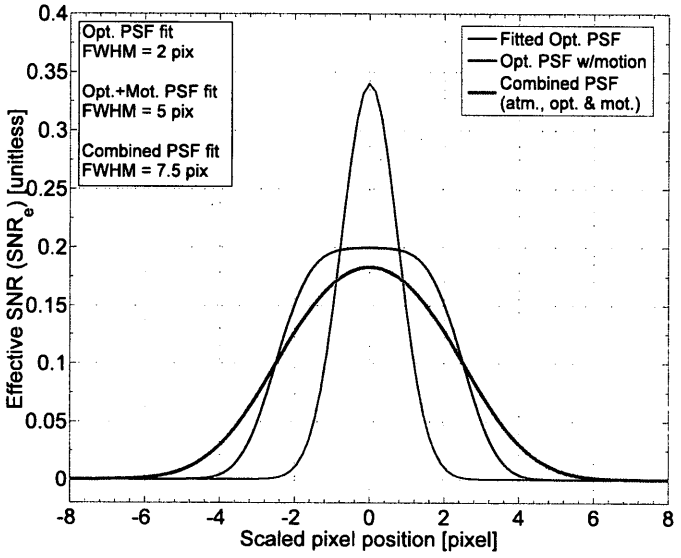


Figure 5.7: Typical optical and motion PSFs of a rotorcraft in level altitude flight.

ms is averaged to reconstruct the new PSF (red). Motion of the sensor platform adversely affects the PSF at long integration times. The time of 75 ms serves as the upper end of expected integration times, and therefore provides a worst-case approximation of platform-motion jitter. The motion PSF was then convolved with an estimate of atmospheric scatter for a 632 nm laser at 6.4 km [KDS98], with the result in orange.

5.4.5 Estimates of R_0 from SNR and PSF models

Once SNR and PSF are adequately modeled, the range at first detection can be studied by computing the SNR of a target pixel under the influence of transfer functions. As such, the effective SNR of a pixel containing the target is shown in Fig. 5.8, with various point-spread functions added. Here, σ represents the standard deviation of the PSF in units of pixels. A detection threshold at SNR = 6.0 was arbitrarily chosen.

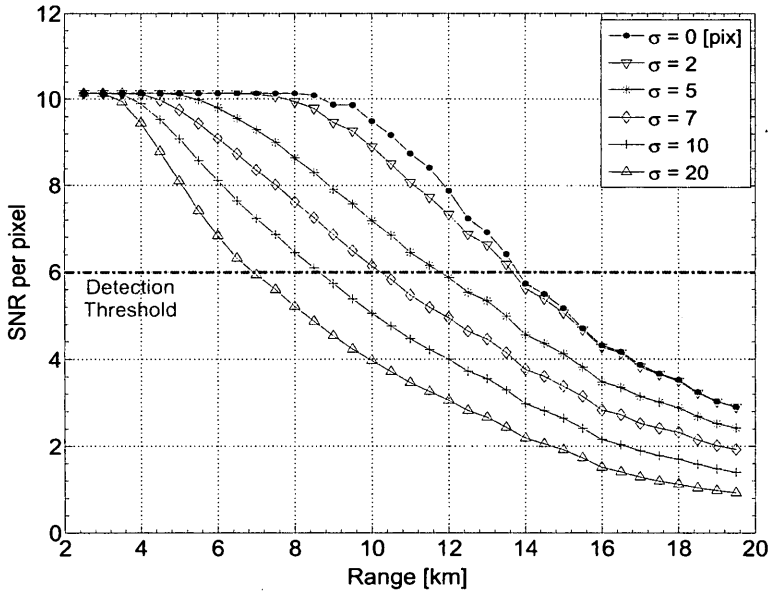


Figure 5.8: Impact of point-spread function on SNR

Fig. 5.9 illustrates the impact of point-spread function on instrument performance, through its effect on R_0. The detection threshold is identical to the one utilized in Fig. 5.8. Reduction of R_0 is commensurate with PSF width and atmo-

spheric visibility is dominant as it is the largest contributor to PSF width. Some over-design of the sensor under normal conditions (i.e. R_0 greater than 10 km) would allow for adequate performance at the low-visibility end of visual flight rules.

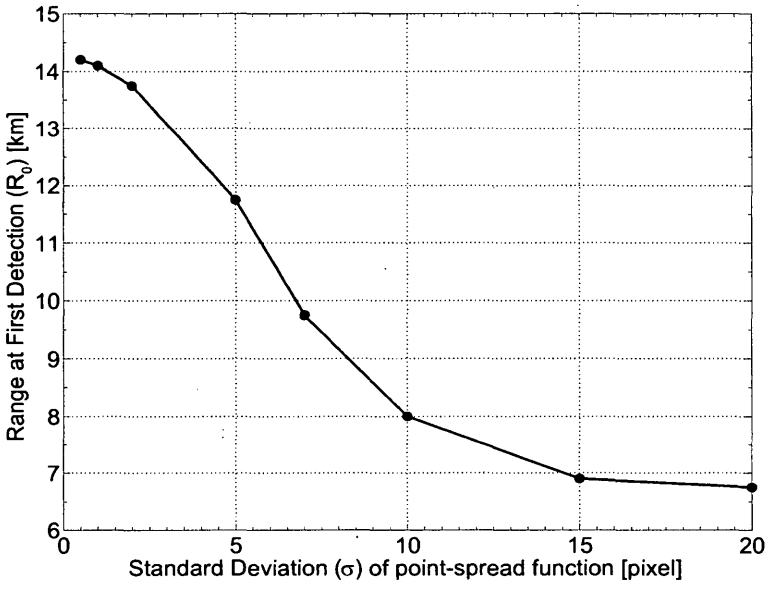


Figure 5.9: Impact of point-spread function on R_0

A critical assumption here is that the point-spread function is independent of range. This is not true in real-life as atmospheric attenuation and scattering make the PSF range-dependent, and may dominate other stages in the image pipeline. Furthermore, complex targets have non-standard behaviour once the pixel-field-of-view is filled by the target, as different parts of the target may saturate the signal at different ranges.

5.4.6 Impact of image processing on SNR and R_0

Image processing is expected to recover some of the performance reduction caused by the system point-spread-function. In Fig. 5.10, the SNR versus range is shown for low atmospheric visibility, for the uncorrected and corrected cases. Here, a PSF $\sigma = 10$ pixels was used, as the worst-case combined PSF as per the full-width at half maximum value of 7.5 pixels estimated in Fig. 5.7. Also shown, for reference, is the curve with negligible PSF, where the transfer function is the Dirac delta function.

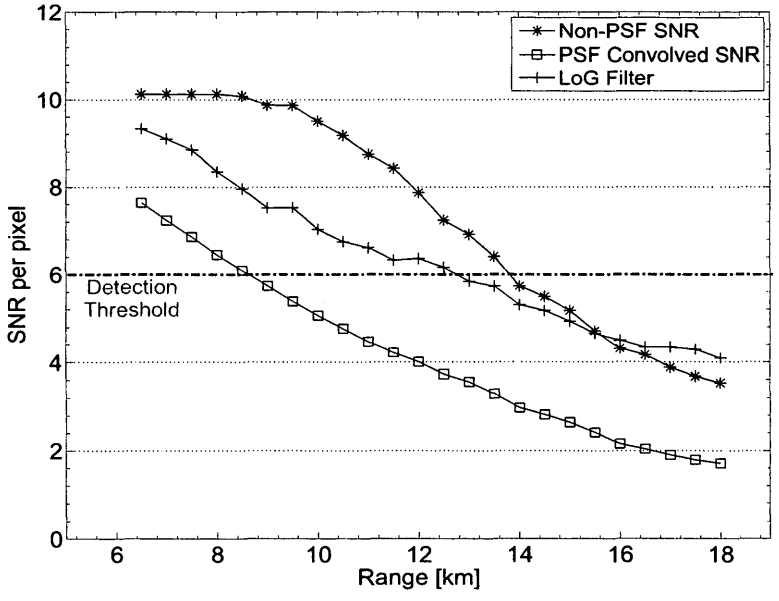


Figure 5.10: Impact of image processing on R_0

The correction was performed by applying a simple Laplacian of Gaussian (LoG) based point-detection algorithm [GW02]. In this case, R_0 has increased by 40%

due to image processing, with the range of detection shifting from 8.0 km out to nearly 12.0 km. Therefore, the "image processing advantage" figure of merit, χ_{proc}, evaluates to $\chi_{proc} = 1.5$. Advanced processing techniques are expected to significantly improve the performance. Trade-offs between performance, size and power come into play here, such as using higher resolution sensors, limiting the framerate, or increasing the computing capacity. It is expected that χ_{proc} would allow quantitative trade-offs that mix these system parameters.

5.4.7 Proximity to the horizon - h_{tgt}

Fig. 5.11 shows a number of simulated, head-on, near-collision trajectories at varying altitudes of the target platform [MTW+10]. Preliminary flights with a Bell 205 platform and a Harvard Trainer aircraft target validate the figure for selected (safe) trajectories. Motion of the platform is common to target and horizon, hence, measurements of Δh_{tgt} can be averaged over many frames.

For targets on a collision course, defined by matching altitude (100 m in the figure) and level flight, the horizon-height profile is relatively constant for the duration of the run. For all other target elevations, Δh_{tgt} grows non-linearly with decreasing closing distance. In order to track h_{tgt}, the angular positioning precision of the sensor needs to be less than ~ 1.0 mrad. For the modeled sensor choices, the probability of a target miss should be discernible out to at least 7.5 km. Although

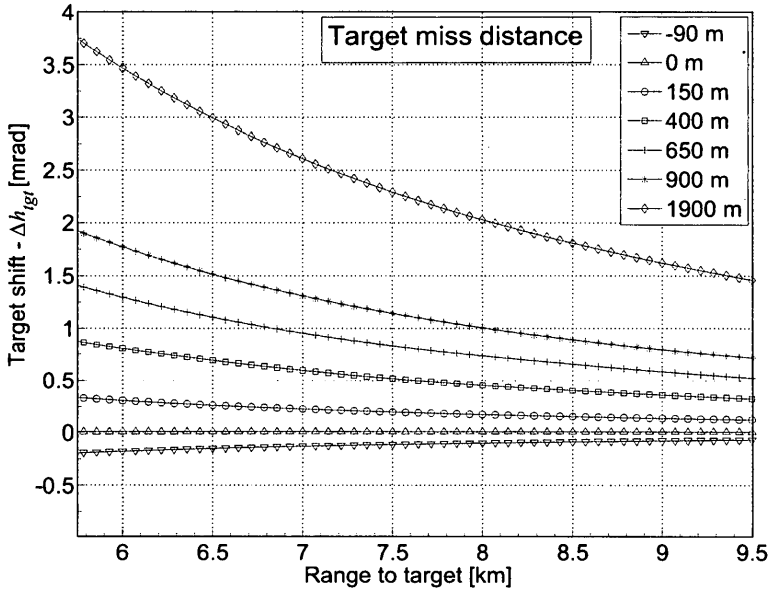


Figure 5.11: Apparent above-horizon displacement in pixels for different target altitudes with both platform and targets on level flight-paths

the variation may be within the platform PSF, appropriate image processing to deconvolve the point-spread is expected to reveal the target at this range.

5.4.8 Limitations of modeling efforts

This particular in-house simulation environment suffers from some limitations; however, these limitations are generic to most simulation environments. Here, targets are represented as purely diffuse spheres. Although, texturing and higher order geometries entail greater complexity, they require more computational power. Reflection models are restricted to diffuse and specular types, where specular shading

utilized the Phong (plastic shading) model. The Phong model is computationally simple, but not altogether representative of targets with complex, multi-textural elements (See Section 3.5).

Furthermore, the ray-tracing operation always occurs on static objects. Each iteration of the simulation time-space represents a geometric transformation for all objects in the scene, followed by a ray-tracing operation. This particular sequence of steps was utilized for computational expedience. Since the transformation and ray-tracing steps are discrete and independent, and scene elements are static during the ray-trace, the simulator is incongruent with the behaviour of actual targets and actual detectors in real environments, where photon collection occurs over a finite period of time, giving rise to shutter artifacts, scene fluctuations, as well as platform and target motion convolution. Such behavioural artifacts are not modeled here.

5.5 Laboratory tests

Laboratory tests were conducted in two parts: A detailed characterization of the sensor with the custom dome facility at the Vision Sensor Laboratory and preliminary outdoor tests in a field. Characterization was performed by various VISOR staff, in order to verify model predictions and specify parameter values. The characterization tests included the measurement of the sensor dynamic range, Rayleigh criterion evaluation, point-spread function of spot targets, noise performance, esti-

mation of the sensor field-of-view for typical lenses and computational performance. In the interest of brevity, laboratory characterizations are curtailed to the evaluation of the computational performance. Additional details regarding the characterization are available in technical reports submitted to the National Research Council (NRC) of Canada [TML+10] [MTW+11], which provide a detailed description of the methodology, evaluation parameters and results. A paper on selected characterizations utilizing the laser-scanner optical dome facility are presented in [WML+11].

5.5.1 Evaluation of computational performance

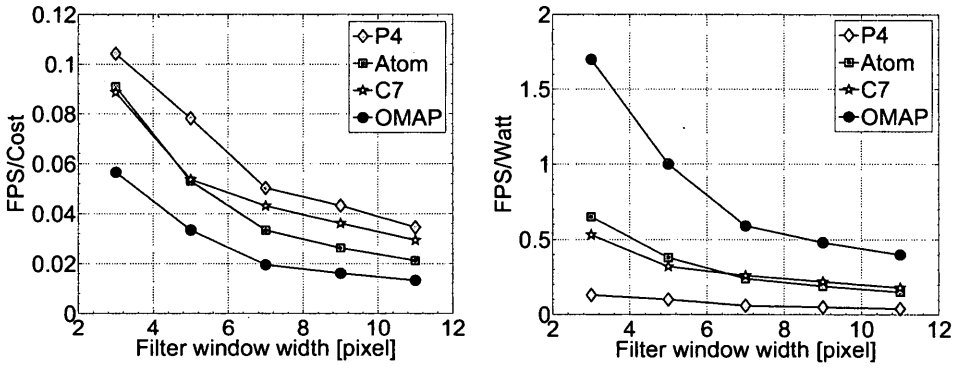
A benchmark is a necessary aid towards the selection of the optimal computing platform. A small ARM-based platform, such as the Beagleboard, was found to be optimal for collision detection, with rationale in support of this conclusion described as follows. Note that the benchmark methodology described herein is generic, and can be applied to any instrument comprised of multiple embedded platform. Although platform homogeneity is assumed, the analysis can be readily extended to heterogeneous interfaces. Fig. 5.12 illustrates a comparison of the different computing platform options, with a curve for each of the four options, and each plot corresponding to a different benchmark. Table 5.2 displays the relative parameters for the platforms tested: TI OMAP (ARM)[Bea09], Via C7 (x86)[Jet07] and Intel

Atom (x86-64)[ZOT09]. A desktop based on the Pentium 4 acted as the reference. All systems were configured with a derivative of Linux.

The benchmark metric was the time required to compute a filter operation on one image. The numerator in the y-axis for all three benchmarks is the acronym for frames per second (FPS), which is chosen as the performance figure-of-merit. Each point on the curve has the kernel size as the abscissa, and the normalized performance as the ordinate. The described metric is representative of a standard image-processing pipeline, where the filter step is typically the rate-limiting operation.

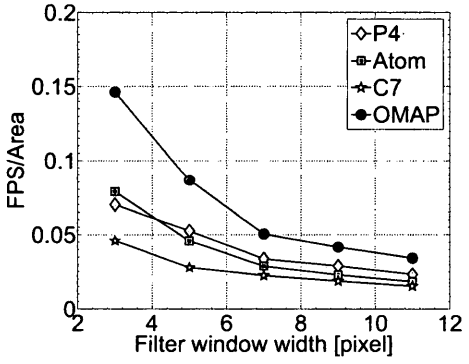
Fig. 5.12a illustrates the performance per platform cost. The reference PC was the top performer (largest value), as expected, since the cost does not vary significantly across platforms. The OMAP platform, consequently, performed the worst due to a high cost penalty associated with miniaturization. Similarly, Fig. 5.12b illustrates the performance per watt for each platform. Here, the OMAP demonstrates the best performance on a constrained power budget. These results are as expected, given that the OMAP power footprint is one order of magnitude smaller than either the Atom or C7, and two orders of magnitude smaller than the reference PC.

Finally, Fig. 5.12c presents the performance relative to the platform area. Volume was not explicitly calculated as height was expected to contribute little to the



(a) Cost

(b) Power



(c) Area

Figure 5.12: Relative performance of computing choices

overall module dimensions. Given surface-mount components, height for a single-board computer is determined predominantly by the CPU heatsink, and is therefore directly proportional to the power consumption. Here, the OMAP out-performs other systems due to its small form factor and low power usage.

Table 5.2: Test platform parameters

Parameter	Unit	C7	Atom	OMAP	PC
Architecture	-	x86	x86	ARM	x86
CPU	[MIPS]	2900	7800	1000	5500
RAM	[MB]	1024	1024	256	1024
PCB Area	[cm^2]	289	289	60	744
Power	[W]	15	30	4	400
Cost	[$]	150	250	150	500

5.5.2 Outdoor field tests

Outdoor tests were conducted locally, in a field north of the Keele Campus of York University. These tests were used to validate model predictions regarding the PSF, differences between target and background signals, as well as variations in sunlight illumination.

5.5.2.1 Experimental methodology

The outdoor test was performed with a prototype sensor comprised of a smart camera group mounted on a tripod, and a planar cardboard target placed some distance away. A diffuse white cardboard sheet of known dimensions, mounted atop a 12 foot stand, acted as the target. The scene and cameras were stationary and the target location was fixed and known. Short sequences of images were acquired of the target at the chosen range.

The cardboard target was approximately 1.0 pixel in angular size (excluding

PSF) at the chosen range of $\sim$ 800 m, and could be seen against a dark local background. Saturation was expected to play a factor, since the scene dynamic range exceeded the pixel bit-depth (8 bits), as is typical on a sunny day. All scene elements were in focus beyond 10 m. Atmospheric visibility was estimated to be greater than 2.0 km, and its impact on the system was considered to be negligible.

5.5.3 Point-spread function for an ideal target

Fig. 5.13 illustrates a typical outdoor capture with inset target crops and histogram highlighted. Saturation on some pixels limited the maximum signal in the area near the target to $dn_{max} \simeq 100$ dn. High-pass filtering extracted the target from the background (Fig. 5.14a), which can also partially compensate for the point spread function.

One-dimensional profiles of the target were fitted by a Gaussian function, of which the x-axis signal plot was illustrated in Fig. 5.14b. A target profile with a full width at half maximum (FWHM) of 2.5 ± 0.5 pixel was observed. At the range considered, the dominant contributer to point-spread effects was expected to be optical in nature.

The empirical signal-to-noise ratio (SNR) for the target was calculated as per Eq. 5.15 at the range of 0.8 km,

$$SNR_e(R = 0.8) = \frac{\mid d_s - d_b \mid}{d_n} \approx 17 \text{ dB} \tag{5.15}$$

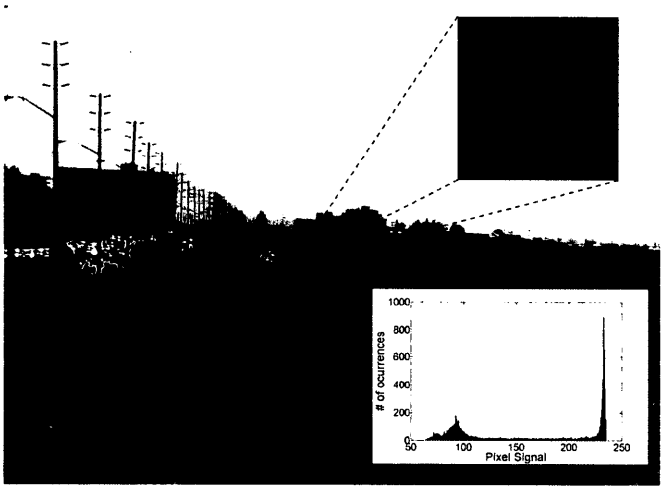
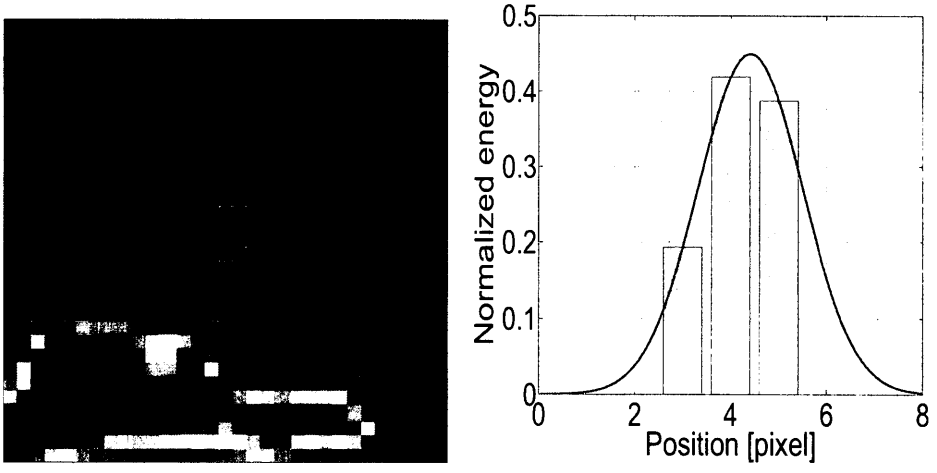


Figure 5.13: A complex outdoor scene with inset histogram and target region-of-interest crop



(a) Target signal after background removal

(b) Target width profile

Figure 5.14: Estimation of the PSF from target features.

Here, d_s, the target signal in digital number (dn), was extracted by using a region of interest (ROI) window about the target. The background signal, d_b, was estimated by averaging ROIs selected from the immediate vicinity of the target. Similarly, the noise, d_n, was calculated from the sensor read noise of $\leq \pm 0.5$ dn and the estimated shot noise (target and background) of $\simeq 0.56$ dn.

It is noted that the observed PSF FWHM was directly in line with the modeled PSF (Fig. 5.8) for an optics-dominated transfer function, and corresponded to PSFs observed for standard laser spot sizes during laboratory characterization [TML+10].

5.6 Ground trials

Ground trials were conducted on-site at the Flight Research Laboratory in Ottawa. These ground tests were essential to the development of a working methodology for subsequent flight tests. In particular, they were necessary to meet (or exceed) the minimum performance requirement, namely detection of the target in excess of 5.0 km, before clearance for further flight tests would be granted.

5.6.1 Rationale

The ground tests established, or confirmed, typical parameters for the instrument, host platform and target platform for the flight tests to follow. Among the necessary aspects for target and host platforms included: Close coordination of the host and

target platforms and execution of the planned flight path. This proved to be fairly challenging. Flights by the target platform also established the validity of typical types of flights that could be conducted given the constraints of time and fuel.

In terms of the instrument, the tests confirmed the correct operation of the instrument, successful aircraft integration and user interface behaviour under test conditions. For the camera, they served as a necessary validation of standard parameters, including flight altitude, time of day, clear sky conditions. and the choice of lens focal ratio. The instrument was also integrated with a portable inertial navigation system (INS) with characteristics similar to the host platform INS.

5.6.2 Experimental Setup

Fig. 5.15 illustrates the field test rig. A prototype sensor, comprised of smart camera nodes, was mounted onto a tripod. The tripod was anchored solidly into the ground to minimize vibrations from wind gusts. A Bell 206 Jet-ranger helicopter (Fig. 6.1a) acted as a real target by conducting multiple fly-bys over the sensor test site.

Power was supplied to the test-site by a local generator, placed downwind approximately 20 feet, to minimize the impact of combustion pollutants on the cameras. An array of five cameras (Table 4.1), recorded the flights at an acquisition

rate of 3.0 Hz. Each lens was set to a focal ratio of F# = 5.6, which offers a good trade-off between lens aberrations (spherical, astigmatic) and diffraction-limited PSF effects. One of the nodes was re-purposed as a spotting scope later during the tests.

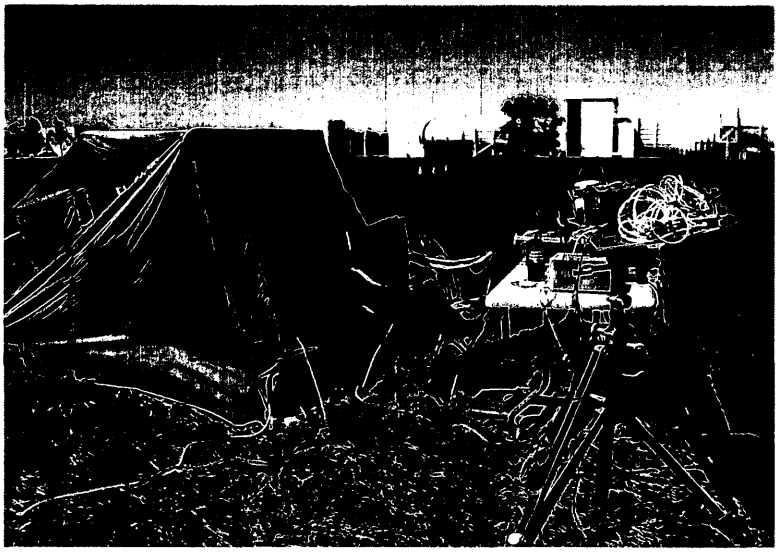


Figure 5.15: Ground prototype test site configured for Bell 206 fly-bys

5.6.3 Procedure

Burn-in tests were performed prior to actual flights, with extended recording sessions of outdoor scenes. The lenses were also focused to infinity by sighting a distant landmark and confirming maximum sharpness in a live-feed. This proved to be a necessary step as the sensor position is prone to machine tolerances, and may not sit at the rated distance from the lens rear flange.

Tests were conducted in clear sky conditions, with minimal clouds and scenery. Figure. 5.16 depicts the planned flights for the ground test platform.

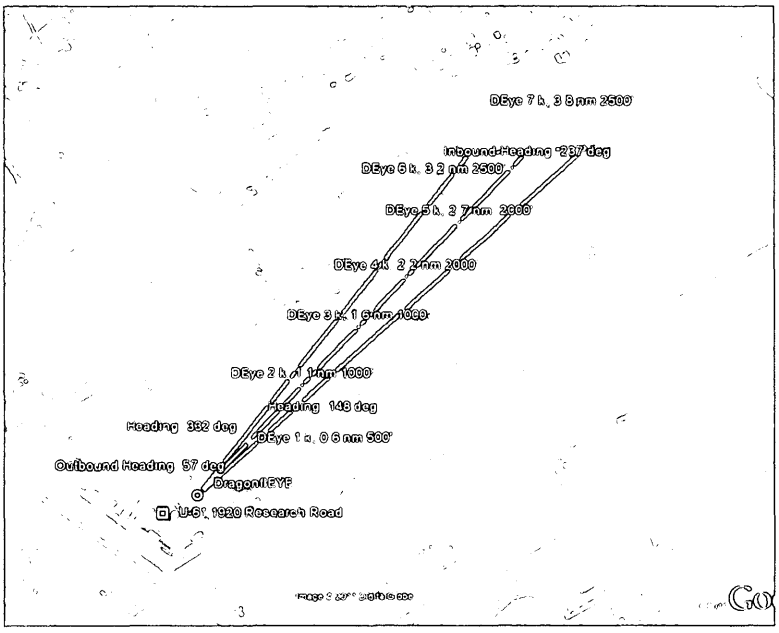


Figure 5.16: Overlay of all flights conducted with the ground prototype

Here white lines depict the desired flight paths as well as the viewing frustum of the camera. The following flight trajectories were planned:

- Head-on and azimuthal collision tracks: 0°, −5°, −10°.
- Perpendicular crossings: 8 km, 6 km, 4 km.

Head-on and azimuthal collision trajectories were achieved by having the 206 fly a nominal glide-slope of -7.0 degrees from an initial distance of 9.0 km and an initial altitude of 2800 ft. The glide-slope matched the platform's pitch, thereby

simulating a 'co-altitude' head-on collision trajectory. Other trajectories included level overhead passes and perpendicular crossings at varying altitudes. Given the strong glide-slope, perpendicular crossings at increasingly closer ranges proved to be quite challenging to fly, given that altitude varied as a function of the distance to the ground-station before commencement of each trial.

Air traffic constraints limited the region of operations to a cone constrained by the approach path of two runways at the Ottawa International Airport, which is indicated here by red lines. This was a safety consideration to ensure minimal interference with inbound and outbound commercial traffic.

5.6.4 Results

Flights were flown in accordance with the planned tests on two separate occasions. Fig. 5.17 depicts the actual trajectories flown in yellow and overlaid onto the planned paths depicted above in Fig. 5.16. The trajectories can be observed to map very closely to the desired paths for the desired instrument performance parameterization, highlighting the expert capabilities of the test pilots at NRC.

Fig. 5.18 illustrates a capture from the ground tests. The target was typically spotted at a range of $\sim$ 5.0 km by the ground crew, although offline analysis of captured imagery offered better performance. Given the stability of the host platform and the absence of clutter the target was easy to spot at detection ranges

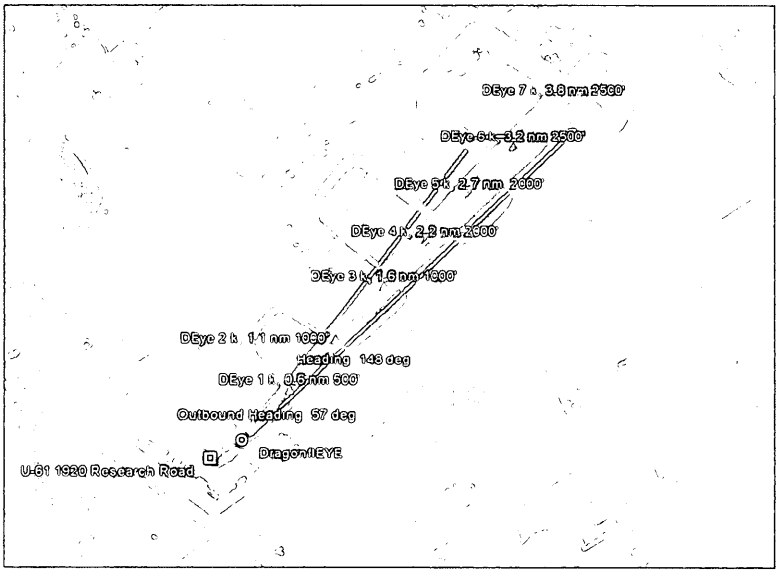


Figure 5.17: Overlay of all flights conducted with the ground prototype.

$>$ 7.0 km. This measurement clearly exceeded the minimum performance threshold of 5.0 km set by NRC Canada, confirming the validity of the instrument design and paving the path for further flight tests.

Other instrument functionality was also confirmed. Bugs related to INS integration and packet capture were identified and removed. Functionality of both the camera node state-machine and the test node multi-threaded user interface was experimentally verified. Synchronization with the aircraft inertial measurement unit (IMU) and global positioning system (GPS) was verified. Initial logging and time-stamping mechanisms were tested and confirmed, although the code-base underwent further refinements for time-stamp synchronization before the actual flight tests.

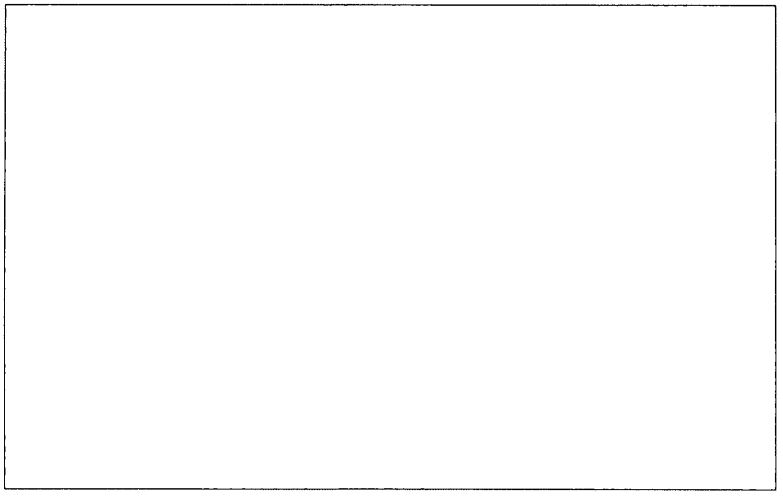


Figure 5.18: A single capture from the ground tests, with the target indicated by a red reticle.

5.6.5 Challenges

Instrument alignment with the inbound target trajectory (heading and pitch) proved to be challenging, given the limited viewing frustum of the sensor and lack of positional information. Field-of-view coverage proved to be a dominant issue, despite the stability of the ground platform. The decision was made to outfit the instrument with higher resolution sensors (2.0 MP vs 1.3 MP) at the expense of frame-rate and an additional row of five cameras to guarantee sufficient coverage of the field of view in the presence of host platform oscillations.

Flight coordination was difficult at best as ADS-B was unavailable at the ground station, primarily for safety reasons and hand-held VHF radios were limited in

range. Therefore, the iPod Intercept Display (Fig. 6.4) could not be utilized for beyond line-of-sight alignment. Usage of the display is discussed in greater detail in the flight test section (Chapter 6).

The extreme glide-slope flown by the target helicopter prevented any analysis of horizontal proximity. Since the incoming glide-slope had to be matched by an equivalent (upward) tilt in pitch of the camera platform, insufficient ground detail was captured in the imagery as a result.

5.7 Conclusion

In summary, this chapter presented modeling activities conducted within a in-house simulation environment as well as ground tests. The simulation environment provides a convenient model of the scenarios of interest, and subsequent ground tests support and validate the simulation results. Field tests conducted at the NRC's flight research facility confirmed instrument operation and functionality, as well as meeting the minimum performance requirements to pave the path for flight trials. As well, the ground targets helped establish a modus operandi for subsequent flight trials.

Practical, measurable definitions for the key figures of merit (FOM) were also discussed, some of which were outlined theoretically in Chapter 3. Specific incarnations of the practical definitions appear in Chapter 6.

6 Performance Evaluation: Flight Tests

6.1 Synopsis

This chapter describes the flight trials conducted for the test and evaluation of the DragonflEYE instrument. Details on the prototype are presented in Chapter 4, while the test strategy is outlined in Chapter 2. This chapter is structured as follows: First, details of the flight tests are presented, including experimental setup, venue, flight operations, data acquisition methodology and system operation. Deviations from the proposed test strategy, due to operational and safety considerations, are also discussed.

The description of flight tests is followed by post-flight analysis of the captured imagery. In particular, the extraction of the range at first detection and time-to-impact from real (noisy) data are presented for multiple head-on cases. Long-range behaviour of the target is studied by measuring target size and growth for the duration of typical trajectories. Perpendicular crossings allow the study of a quasi-stationary target, from which the system point-spread function may be estimated,

while head-on cases can be utilized to study target growth as a function of range.

6.2 Introduction

Flight trials are critical to the proper evaluation of instruments. Aspects of the real scenario, usually not anticipated during modeling or ground tests, can conspire to limit instrument performance. Presented in this chapter are details of the flight trials conducted with the DragonflEYE collision-sensing instrument. Two rotorcraft were used: A Bell 205 as the host platform and a Bell 206 as the target. Note that use of rotorcraft offered additional challenges not normally present in traditional fixed-wing aircraft.

6.2.1 Rationale

The flight tests described herein were designed to verify the performance of the instrument. Performance was measured against pre-defined criteria and modeling results. In the present case, the primary figures of merit included the range at first detection, R_0, and its corollary, the time-to-impact, t_0. Ancillary figures, such as target size and growth, contrast against background, and proximity to the horizon, were also analyzed and discussed. The decision to record raw imagery in the absence of any processing highlights the desire to maximize the value of data from highly expensive trials.

6.3 Venue

Trials were conducted in controlled airspace near the Flight Research Laboratory at Ottawa International Airport (CYOW). Trials described below were flown on Feb 28, 2012, between 4:00 pm and 5:00 pm EST. Two helicopters were flown, a Bell 205, which acted as the host platform and possessed experimental fly-by-wire capability and acted as a surrogate UAS [EG05], and a Bell 206, which acted as the target. The DragonflEYE instrument was rigidly mounted to the host platform. Safety and airworthiness tests were conducted, to verify the safety of the airframe with the instrument attached, before the aircraft was cleared for flight. These tests acted as the gate-keeper for the progression of any further flight tests, as safety of the airframe and its occupants is paramount.

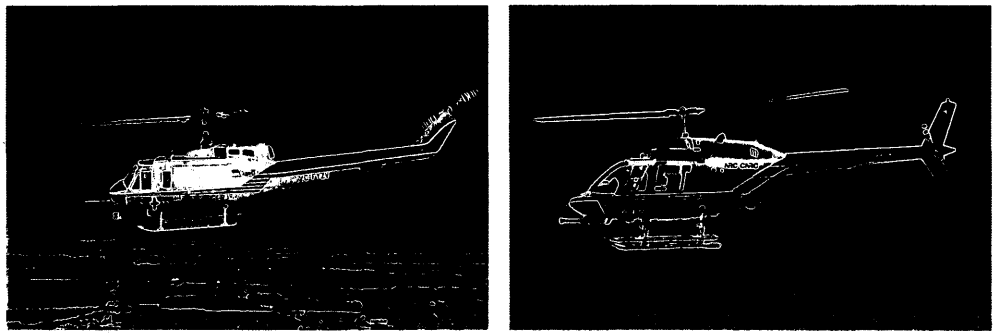
Flights were conducted in relatively clear sky conditions, in areas where the local terrain structure was relatively flat, at a nominal altitude of 2000 ft above ground level (AGL), with the sun at roughly 90 degrees in azimuth to the flight path, unless otherwise specified. The hourly METAR visibility was in excess of 20 km, with typical wind speeds at 20 km/h. The cloud ceiling was reported in excess of 3.0 km, and proximity to the airport limited the altitude ceiling to 3.0 km.

6.4 Aircraft

Fig. 6.1 displays the Bell 205 (Fig. 6.1a) and the Bell 206 (Fig. 6.1b) helicopters, as configured for the flight tests. Each aircraft was instrumented with an automatic dependent surveillance-broadcast (ADS-B) transponder [Ves08] and an inertial navigation system (INS) [LDR03] comprised of an inertial measurement unit (IMU), a global positioning system (GPS), and a single-board computer (SBC). A flight test engineer interacted with the instrument via an in-cockpit LCD display and trackball. Maintaining the desired collision trajectory beyond the range of visual acquisition was an important practical consideration. This task was assisted by an iPod Intercept Display [KEC+11], which provided ADS-B-based bearing and range guidance for the flight crew. The DragonflEYE instrument was calibrated in orientation with respect to the host IMU, such that the pointing direction with respect to the aircraft's heading was explicitly known. Activation and initialization of the instrument occurred at the beginning of the flight sessions.

6.5 Airworthiness tests

Installing any instrument to the chassis of an aircraft is treated as a modification to the aerodynamic profile of the airframe. Airworthiness tests must be performed to ensure that the modified aircraft can operate within rated tolerances during normal



(a) Host - Bell 205 (b) Target - Bell 206

Figure 6.1: Aircraft flown for the flight tests

flight operations.

In rotorcraft, the principal source of structural stress is resonance coupling with the main-rotor fundamental frequency. If the mounted instrument possesses a resonant frequency that is similar to, or a harmonic of, the rotor fundamental frequency, the system may enter a positive feedback loop, culminating in structural failure.

Airworthiness tests were performed by NRC staff at the Flight Research Laboratory in Ottawa [IE12]. A description of the steps that were followed is provided for completeness. Safety verification of the instrument was conducted in multiple parts. Initially, the structure was redesigned to create a self-contained instrument. Therefore, any parts that may be dislodged during flight are contained within the super-structure and the risk of contact with control surfaces is minimal. This design methodology allowed the safety engineers to treat the instrument as a sealed

Table 6.1: Experimental Aircraft Flight Specifications

Aircraft	Parameter	Value	Units
Host Platform	Model	BHT-205A1	-
	Dimensions*	17.4 x 4.5 x 2.75	m^3
	Test Airspeed	70	kts
	Gross weight	4085	kg
Target	Model	BHT-206B	-
	Dimensions*	11.8 x 3.5 x 1.9	m^3
	Cross-section Area	6.8	m^2
	Test Airspeed	90	kts
	Gross weight	1450	kg

*including rotor blades

unit, and subsequently disregard the composition, interconnection and/or rigidity of the internal components. For the DragonflEYE flight prototype, satisfying this paradigm required the SBCs to be placed inside a large box (Fig. 4.7a), and split from the detector/lens assembly. This bifurcation represents a deviation from the physical node methodology envisioned in the laboratory and ground prototypes. Furthermore, each individual node was outfitted with an ultraviolet (UV) haze filter, glued in-place, to act as a weather seal. Note that the impact was primarily on the mechanical packaging. Signal, electrical and software subsystems of the nodes were similar to earlier prototypes. Note that the redesign, fabrication and assembly was performed by internal NRC staff.

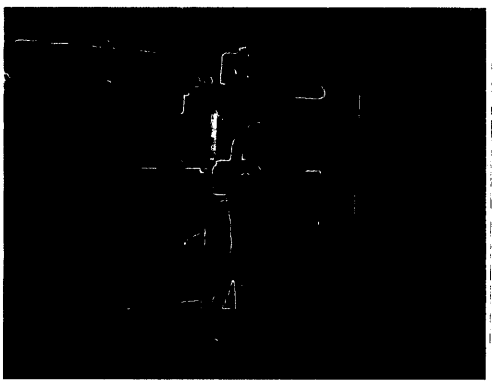
After the completed design was fabricated, assembled, and mounted to the airframe, a ground vibration test was conducted to analyze the resonant frequencies of

the newly installed instrument. The test was conducted by fastening accelerometers at the moment arm nodes within the structure. Fig. 6.2 illustrates the accelerometer placement in vertical (Fig. 6.2a), lateral (Fig. 6.2b) and shear (Fig. 6.2c) configurations.

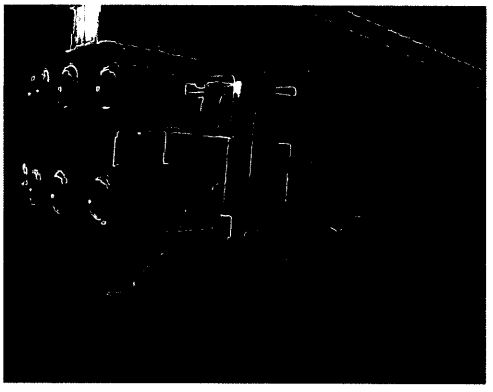
Vibrations were expected to be a non-trivial factor for the cameras, given the cantilever nature of the mount. Fig. 6.3a illustrates the observed power spectral density (PSD) for vertical excitation of the mount. A peak at 19 Hz is observed, indicating the dominant resonant frequency, followed by a secondary peak at 47 Hz. The PSD plot for lateral excitation exhibits similar characteristics, Fig. 6.3b, with a peak observed at 17 Hz. Note that the settling time, although not plotted, was measured to be $\sim$ 3.5 s from the point of excitation. An estimated angular deflection of $\pm$10 pixels ($\pm$0.06 degrees), was measured from DragonflEYE imagery recorded during the tests. Vibrations in the shear direction (along the roll axis) were of lower magnitude and not considered significant.

A two-bladed rotorcraft, such as the Bell 205, exhibits a fundamental main-rotor frequency at $\sim$ 10 Hz, indicating that the resonant frequency of 19 Hz lies close to the first harmonic of the rotor frequency. Since the first harmonic may manifest itself only during emergency manoeuvres, the aircraft was cleared for airworthiness flights, which would assess behaviour under extreme conditions. Typical airworthiness manoeuvres include sudden acceleration and deceleration to and from

(a) Vertical

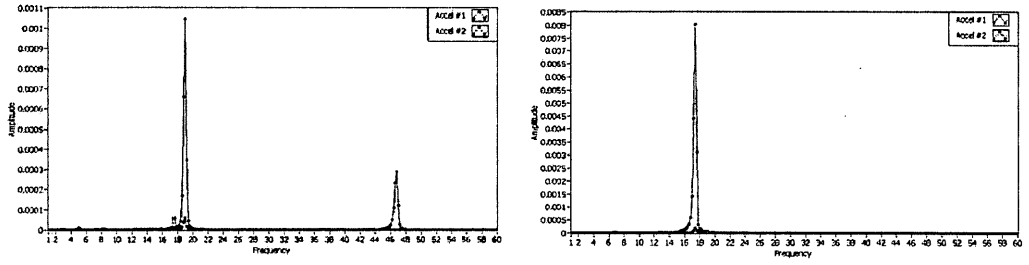


(b) Lateral



(c) Fore-Aft

Figure 6.2: Accelerometer configurations for the ground vibration tests [IE12].



(a) Vertical (b) Lateral

Figure 6.3: Power spectral density for excitation modes [IE12].

maximum airspeeds, sudden course changes, auto-rotation descents, flying through inclement weather (strong winds, precipitation, icing), among others. A sub-set of these tests was performed on DragonflEYE. No stability issues were reported, clearing the instrument for experimental flight tests.

6.6 Flight Evaluation Trials

Evaluation runs, shown schematically in Fig. 6.7, were flown at a constant airspeed along a pre-determined flight path. Flights were generally within acceptable limits of the planned route, subject to operational constraints. The head-on and azimuthal collision runs were conducted with the host in level flight and the target approaching along a well-defined, level collision trajectory. For the descending runs, the target flew in level flight, while the host descended upon the target (from fore and aft) along a pre-determined glideslope. In the overtake trajectories, both aircraft started

in level flight with the host (Bell 205) a predetermined distance behind the target. This run culminated in the host (Bell 205) catching up to the target (Bell 205).

6.6.1 The iPod Intercept Display

Fig. 6.4 illustrates the iPod Intercept Display utilized during flight tests [KEC+11]. Developed at the Flight Research Laboratory, NRC Canada by Kristopher Ellis, this display was mounted to the cockpit dashboard on both the host and target aircraft, and parsed the ADS-B data packet to provide range and heading information of the 'other' aircraft beyond the visual line of sight.

Here, the 'local' aircraft is depicted by a cross, while the 'foreign' aircraft is depicted by an 'airplane' icon. Regions of equivalent range are indicated by concentric circular rings. A beam, offset by slip correction, depicts the intended flight path. The foreign aircraft icon changes colour to indicate the probability of a collision, with yellow indicating likely and red for extremely likely. A call-out box above the aircraft icon depicts the range to the foreign aircraft (target or host) in nautical miles. The flight crew utilizes the call-out box to assign a range measurement upon visual acquisition.

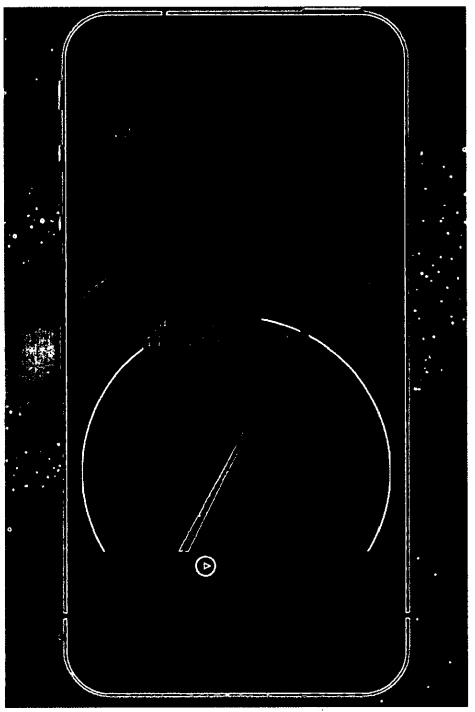


Figure 6.4: The iPod Intercept display depicting a target (in red) on a 25° offset collision course with a time-to-impact of 39 seconds.

6.6.2 Operational procedure

Flights are flown in sorties, where a sortie encompasses the execution of multiple pre-planned flight trajectories, including hand-over, re-alignment and compensation for changes during in-flight conditions. Once initiated, each collision trajectory took between two and three minutes to complete. The flight paths deviated from the ideal profile due to random drop-outs of the ADS-B, with some runs affected more strongly than others. Visual tracking was preferred once positive visual identifica-

tion had been established. Given the dangerous nature of such flights, failure of both ADS-B and visual confirmation typically resulted in a run being aborted via procedural separation coordinated through VHF radio communications. Table 6.2 describes the trajectories covered from the perspective of the host platform:

Table 6.2: Collision vectors flown during flight test evaluation

Type	Description
Head-on	Co-altitude, descend (tough to fly).
Azimuthal offset	Co-altitude, test angles: $\theta = [-5°, -10°, -15°]$.
Perpendicular crossing	Co-altitude, test distance: $D = [6.0, 8.0, 10.0]$ [km].
Overtake	Co-altitude, descend, airspeed differential.
Into the sun	Overtake facing the sun.
Ancillary	Landing, Dash-8 flyby.

The planned closing rate for head-on cases was 296 km/h (160 knots), with the host platform flying at 130 km/h (70 knots) and the target at 166 km/h (90 knots), although post-flight measured velocities were higher. Descending overtake cases employed a slower closing velocity of 75 km/h (40 knots). Each run started at a nominal separation of 12 km and took approximately 3 minutes to complete. Experienced pilots maintained trajectories to within 1 percent of the desired path. Fig. 6.5 highlights a typical head-on run, with the host (205) flight-path in yellow and the intruder (206) flight-path in green. Typical break-off range was 100 m. The iPod Intercept Display aided in quick visual acquisition by dramatically limiting the scan area.

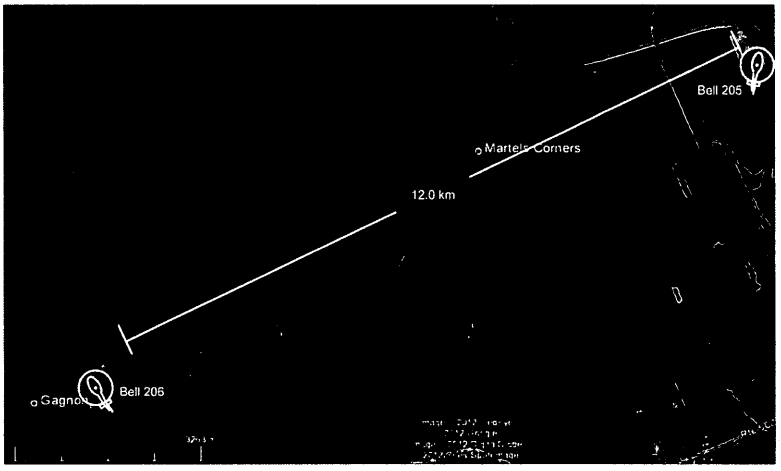


Figure 6.5: A typical head-on run projected onto a satellite terrain map

Head-on and azimuthal-offset collisions define the critical operating path for the instrument, and so are emphasized in the analysis. Perpendicular runs depict a roughly constant-ranged target, although it is noted that the host still moved towards the target. Overtake and descending runs can place the target amid ground clutter. Fig. 6.6 denotes all the collision vectors flown during the Feb. 28 2012 sortie.

Here, yellow denotes the host platform (Bell 205) while red denotes the target (Bell 206 Jet-ranger). The trials were flown in a field 20 km southwest of the airport, away from population-dense areas.

During each evaluation run, image archiving was triggered upon radio confirmation between aircraft. Archiving was stopped after the target aircraft flew past the host. Target locations were localized offline by manual selection of the target

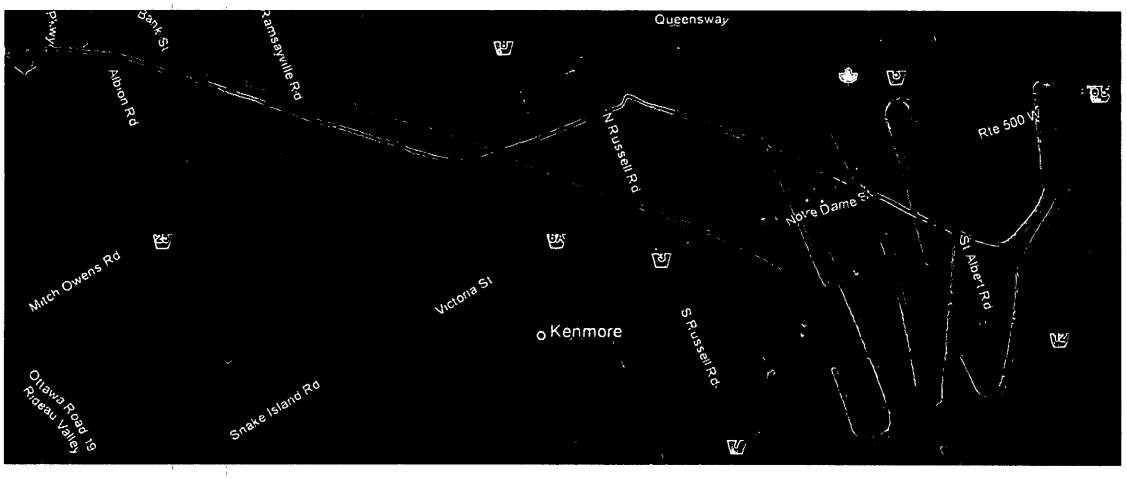
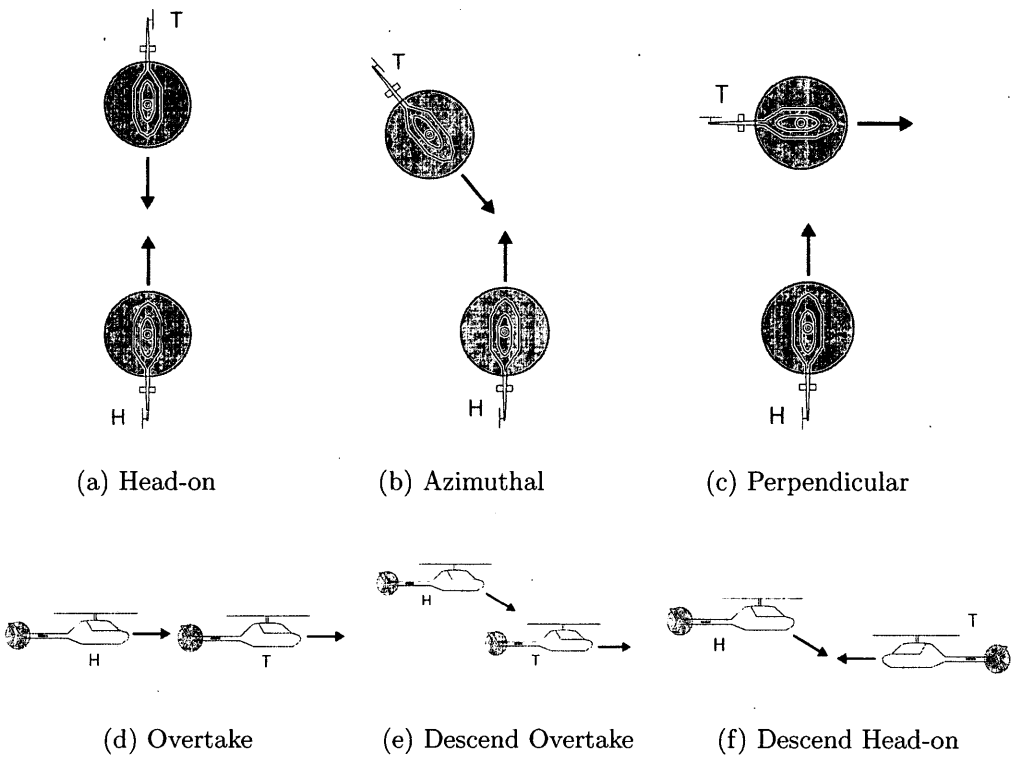


Figure 6.6: All collision vectors flown during the Feb. 28, 2012 collection. Yellow denotes the host (205) path, while red denotes the target (206) path.

in each frame. To aid in proper target identification, the image sequence depicting the run was run in reverse during selection, starting with the easily visible target in close proximity to the host platform.

6.6.3 Safety Considerations

Total flight time for a sortie was restricted by the aircraft's fuel capacity, as transit-time to and from the test site had to be factored in. Flights were conducted in the mid-afternoon, corresponding to a decrease in air traffic around the Ottawa International Airport (CYOW). Weather conditions were actively monitored, given that the flights were conducted in winter, and the sudden onset of precipitation



(a) Head-on (b) Azimuthal (c) Perpendicular

(d) Overtake (e) Descend Overtake (f) Descend Head-on

Figure 6.7: Schematic representation of flight trials. The letters 'H' and 'T' represent the host and target platforms respectively.

could limit visibility, violating the 'clear sky' requirement. Wind gusts proved to be another hazard that required re-alignment. In the advent of strong winds, the flight path would be realigned to minimize cross-wind, therefore reducing wind-induced vibrations in the camera platform.

Pilots maintained continuous radio contact with each other and ATC at CYOW. In the event of communications failure, each aircraft maintained a holding pattern until communications could be resumed. In case of equipment failure, pilots were authorized to land in nearby clearings and power-cycle any components, including avionics. Failure in critical avionics typically meant a return to base.

Atmospheric conditions did not always cooperate. The tests were conducted in winter, which challenged the assumptions behind solar radiance and contributions from the ground. Specular highlights from the snow, ground structures and vehicles were noticeable and solar radiance could fluctuate depending on cloud cover.

6.7 Experimental analysis

For these trials, the principal output is SNR_e as a function of range, where 'range' is the separation between the host and target aircraft, as determined by their onboard GPS instruments. Note that the formulation of SNR_e utilized here is identical to Eq. 5.1 as per Chapter 5. Fig. 6.8 shows the measured SNR_e against range for a representative head-on run. A 'relative' time axis is also shown on the top

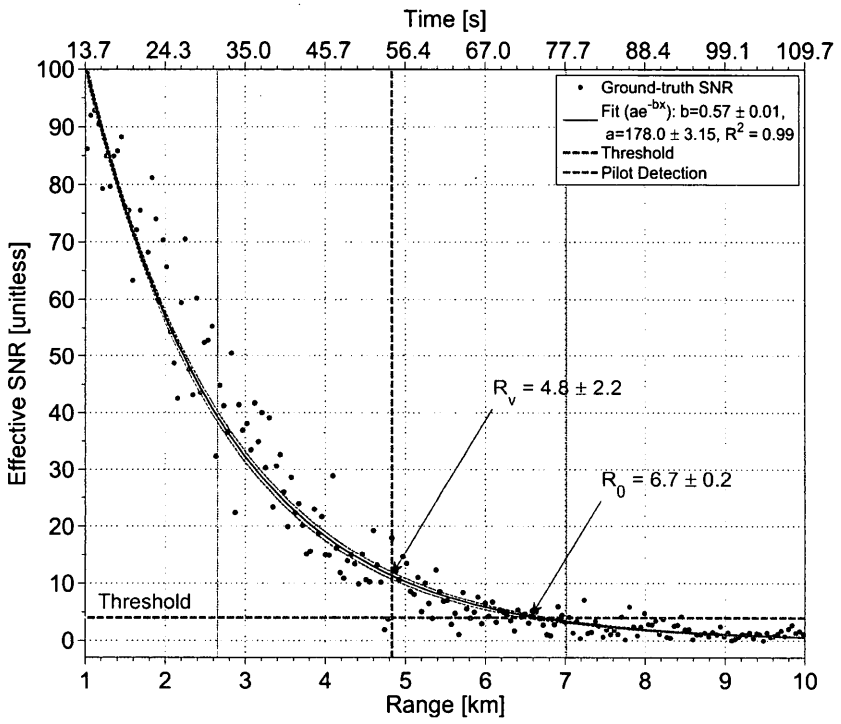


Figure 6.8: SNR as a function of range and time for a typical co-altitude head-on run.

x-axis. The time origin was set to the range at which the target would cross a "near-collision" boundary [Fed83].

At extreme ranges, the SNR_e curve tends to zero, as expected when the target signal vanishes into the background. Each camera node functioned in full auto-exposure mode, placing the most common signal intensity, represented by the average background, at the midpoint of the dynamic range curve. This process halved the dynamic range available to the target. For realistic targets with a spatially disjoint radiance profile, this exposure procedure was deemed appropriate.

An empirical fit of the data was important in order to reduce random error. As mentioned in Chapter 5, random fluctuations from the target, denoted as σ_{tgt} were not accounted for in the measured SNR_e. These variations were an important limit to "manual" observations of the target as well. For an empirical fit, it was important that the number of adjustable parameters be minimal to provide a good approximation for the correct physical model. Furthermore, a good fit can be ensured by utilizing a large observation count compared to the degrees of freedom in the expression. The data was a good fit to an exponential curve (Eq. 6.1), with range as the independent axis.

$$SNR_e(R) = ae^{-bR} \tag{6.1}$$

Here, b is the rate of exponential decay, while a is a scaling factor. An error estimate of two standard deviations is shown for each parameter, with prediction bounds corresponding to these estimates plotted as dotted lines fencing the fit. The fit is precise, with the coefficient of determination (R^2) value observed to be greater than 0.99. Additional parameters did not significantly improve the R^2 value. Parameters a and b have uncertainties of less than 2% over two standard deviations, lending credence to the choice of model. If the exponential variation, b, is due entirely to atmospheric attenuation, σ_e, the equivalent local sensor visibility from the Koschmeider approximation (Eq. 3.30) would be $V_{meas} = 7.0 \pm 1.0$ km. This value is considerably smaller than the local meteorological forecast (V =

24 km), indicating that other factors probably contribute to the slope of SNR_e. A theoretical model of the fit is not important for the determination of R_0.

6.7.1 Variations in target and background signal

The observed variability in the SNR_e curve of Fig. 6.8 is much larger than the computed random noise, d_n. Fig. 6.9 shows the individual components of the SNR_e calculation as a function of range. Both the background and noise signals are relatively constant, while the target signal is responsible for the exponential variation of SNR_e. Rapid temporal variations in the signal are also observable and significant. The non-uniformity decreases with increasing range, until the target is unresolved and blends into the background. These variations in d_s are tentatively attributed to fluctuations due to periodic angular motions of the target, (σ_{tgt}), and may be treated independently from other random noise sources (Eq. 3.23). Note that the host platform wobbles with an angular half-width of ± 22 mrad. However, given the extremely short integration time of 1/5000 s, the observed location is affected, but not the received signal intensity.

6.8 Measurement of R_0

In this context, we explicitly define the range at first detection, R_0, as the range value at which the empirical SNR crosses a predefined detection threshold, S_{th}. As

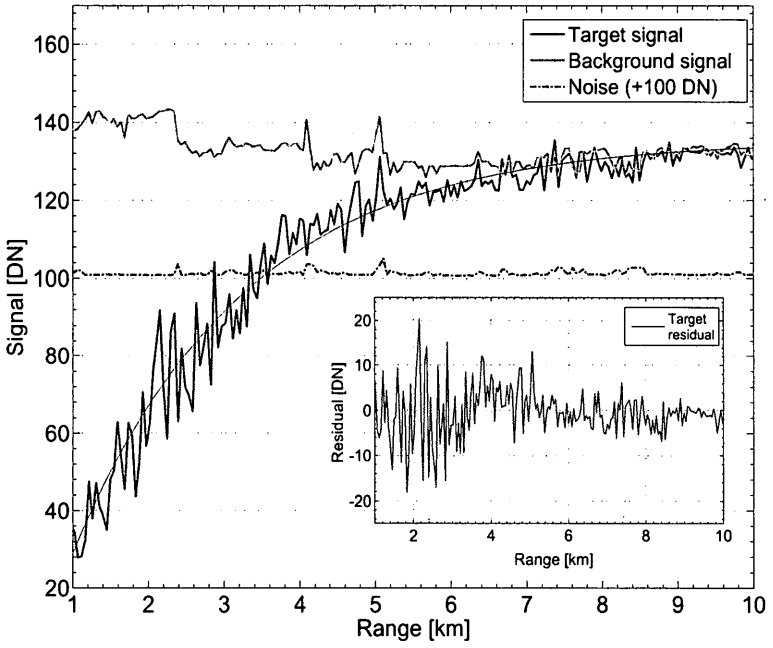


Figure 6.9: Individual signal, background and noise elements from the SNR plot in Fig. 6.8. The inset shows the residual for the target signal from the trend. Noise has been translated by 100 DN for clarity.

a figure of merit, R_0, has the benefit of a simple link to measured quantities (Eq. 6.2), and its random error can be precisely determined from the uncertainties in the fit parameters (Eq. 6.3). Since determination of R_0 uses the imagery from the entire trajectory of a test flight it results in an averaging of errors. In particular, the large fluctuations in SNR_e curve of Fig. 6.8, which strongly affect any visual determination of R_0, are suppressed.

In Fig. 6.8, a horizontal dashed-line indicates S_{th} chosen at $SNR = 4$, which corresponds to a probability of detection (P_d) value of 90% and a false-alarm rate

of 1% under a Gaussian ergodic process assumption [Hol03]. R_0 can be calculated as per Eq. 6.2, with the precision derived as per Eq. 6.3.

$$R_0 = \frac{\ln a}{b} - \frac{\ln S_{th}}{b} \tag{6.2}$$

$$\delta R_0 = \sqrt{\left(\frac{\delta a}{ab}\right)^2 + \left(\frac{\delta b}{b^2}(\ln S_{th} - \ln a)\right)^2} + \delta R_{sys} \tag{6.3}$$

Here, a and b are the fit parameters from Eq. 6.1, while δR_s encapsulates the systematic error in range values. Position values are derived from the INS units installed on both aircraft, with a positioning error of $\delta R_{sys} = \pm 10$ m or less. For this particular head-on run, a value of $R_0 = 6.7 \pm 0.2$ km was observed. Note that δR_{sys} was small compared to the total error in R_0.

Fig. 6.10 shows the fractional error $\frac{\delta R_0}{R_0}$ as a function of threshold (S_{th}). The percentage error is shown to decrease with increasing threshold values, even though the raw precision (δR_0) increases with higher threshold values. The change in fractional error is small over the range of the plot, indicating the relative insensitivity of the metric to the exponential nature of SNR_e or the choice of threshold. In short, the metric indicates a very robust methodology for measuring R_0. It is hypothesized that this benefit is relatively independent of the details of the experiment.

6.8.1 Visual acquisition range

The mean visual acquisition range, R_v, is the range where a typical pilot is expected to spot an target. In the present circumstances, the measurement conditions are assumed to be sufficiently good that the recorded R_v can be treated as optimal. Along with its estimated error, R_v was calculated from aggregated flight crew observations, correlated to ADS-B distance estimates as displayed on the cockpit iPod intercept display. A vertical dashed line in Fig. 6.8 denotes a value of $R_v = 4.8 \pm 2$ km with dashed vertical error bars at two standard deviations.

Despite the known location, the observed variability in R_v is fairly large. This large variability is both a function of target signal fluctuations and the psychovisual characteristics of human agents uncovering small targets in an aerial scene [And91], in particular, cognitive load on the observer in the presence of other flight tasks. Therefore, the effectiveness of R_v as a comparison metric is reduced. A larger sample size would reduce the error due to averaging at the expense of more flights.

6.8.2 Time to collision

A direct corollary of the range at first detection is the estimated time-to-collision metric, t_0. The U.S. FAA defines a "collision distance", R_m, of 150 metres (500

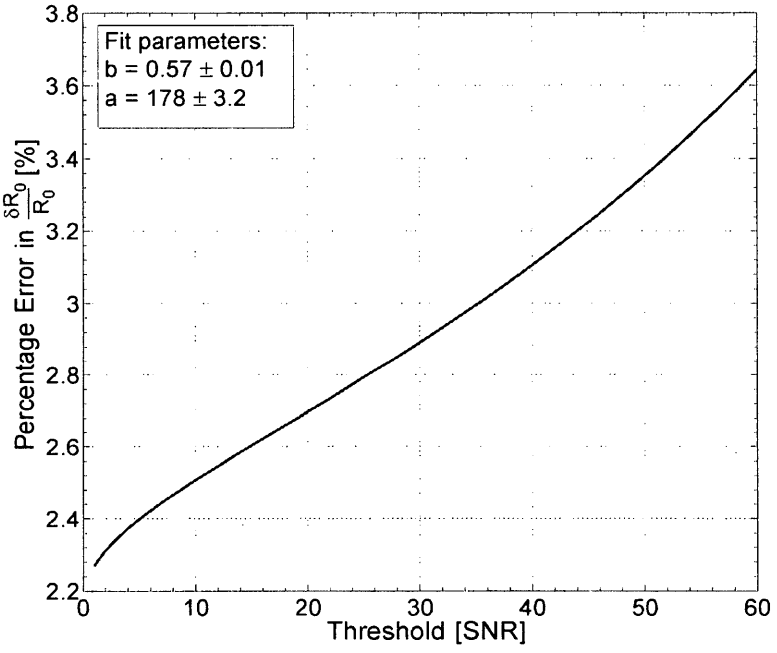


Figure 6.10: Fractional error, $\frac{\delta R_0}{R_0}$, as a function of threshold, calculated for the head-on run depicted in Fig. 6.8.

feet). Similarly, the FAA defines a "minimum time-to-evade", t_m, of 12.5 seconds [Fed83]. We can, therefore, define t_0 to be the time until the target is within 150 m (500 ft) of the host platform, and compare it to the minimum time to evade. Therefore, t_0 and the associated error can be computed from Eq. 6.4 and Eq. 6.5 respectively.

$$t_0 = \frac{R_0 - R_m}{\overline{v_c}} \tag{6.4}$$

$$\delta t_0 \simeq t_0 \cdot \frac{\delta R_0}{R_0} \tag{6.5}$$

Here, R_0 is in units of kilometres, while v_c is the closing rate between the two aircraft. Given the time-stamped GPS range values per acquired image, v_c was estimated from a linear fit to a range against time curve. Since v_c is a quantity calculated from a fit to range values, its error is negligible. Therefore, the error in t_0, including systematic error, is derived from the error in R_0. For the representative run in Fig. 6.8, the time-to-impact computes to $t_0 = 66 \pm 2$ s.

6.9 Evaluation of multiple runs

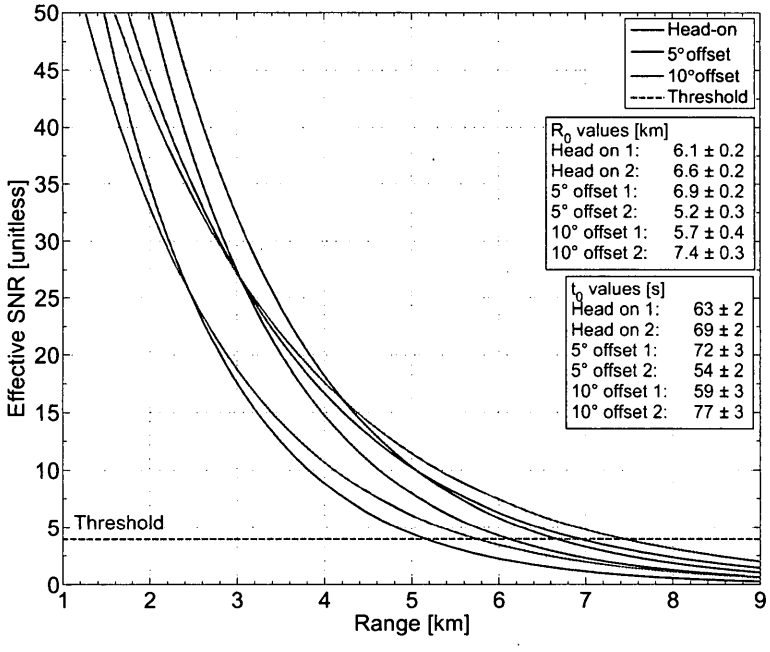


Figure 6.11: Evaluation of SNR curves for multiple events from the February 2012 collection (Table 6.3).

Table 6.3: Statistics on head-on and offset collision geometries for February 28, 2012.

Type	Time [h:m:s]	Heading [deg]	Sun (Az, El) [deg, deg]	$\mathbf{R_0 \pm 2\sigma}$ [km]	$\mathbf{t_0 \pm 2\sigma}$ [s]
Head-on	15:55:18	-25	(-96, 17)	6.1 ± 0.2	63 ± 2
5° offset	16:00:13	160	(80, 17)	6.9 ± 0.2	69 ± 2
10° offset	16:04:47	-14	(-105, 16)	5.8 ± 0.4	72 ± 3
Head-on	16:09:13	162	(80, 15)	6.7 ± 0.2	54 ± 2
5° offset	16:13:57	1	(-119, 15)	5.2 ± 0.3	59 ± 3
10° offset	16:17:41	170	(80, 14)	7.4 ± 0.3	77 ± 3
Mean	-	-	-	$\mathbf{6.3 \pm 1.7}$	$\mathbf{66 \pm 19}$

Fig. 6.11 provides SNR curves plotted against range for multiple head-on and azimuthal-offset runs. The R_0 values were extracted for each run as per the prior established methodology, and enumerated in Table 6.3. Each row in Table 6.3 represents a single run.

The initial column lists the collision geometry flown, as per Fig. 6.7. Timestamps correspond to the start of the image acquisition. Heading values denote the host aircraft's pointing direction relative to the true-north. Since the heading may vary due to course corrections, the value listed for each run represents the median estimated over the trial duration. Sun azimuth values were calculated from GPS data (latitude, longitude, time-of-day) at the start of each trial and offset relative to the host platform. Elevation measures the sun's vertical angular offset from the horizon. All degree-valued measurements (heading, azimuth, elevation) assume a

two-standard-deviation error of ± 1.0. A mean $\overline{R_0} = 6.3 \pm 1.7$ km was calculated across all runs, and indicates an error much larger than the precision of a single run. It is hypothesized that the run-to-run variability in R_0 is not random, and can be attributed to scenario factors such as changing sun angle, trajectory geometry and wind-induced target attitude variations.

6.9.1 Verification of the methodology

A subsequent session of flight trials was utilized to ensure that the range extraction methodology is valid for data collection on an arbitrary day, and robust against the time of day, fluctuations in weather patterns, drop-outs and other ancillary factors.

A second set of flight trials was conducted on Mar 7, 2012, and a new sequence of collision vectors were flown. Flights were conducted in the morning, with twelve trajectories flown over the period of one hour. Atmospheric conditions consisted of clear-sky visibility, in excess of 15 standard miles as per the METAR profile. Ancillary conditions (temperature, pressure, humidity, UV) were similar, if not identical, to the February collection. The noticeable exception was the presence of strong cross-winds, with gusts on occasion exceeding 24 knots, which prevailed for the duration of the collection. Thunder-showers were predicted from noon onwards, limiting the window of collection. In addition, constant course corrections consumed more fuel, which required the aircraft to return to base sooner than expected, forcing

a cancellation of some of the proposed geometries.

The primary focus of these flight trials was the analysis of instrument behaviour in cases where the target was embedded in ground-clutter. Consequently, the bulk of collision vectors flown on that day covered descending approaches, which juxtaposed the target in below-horizon backgrounds. Nonetheless, four trials were available for analysis in the March collection, comprising the sum total of two head-on and two azimuthal offset cases. Analysis of descending trajectories is pending and relegated to future work. It is hypothesized that the current background model is insufficient to encapsulate the impact of ground clutter, and a new background model, combining statistical and structural/textural components may be required.

Fig. 6.12 illustrates the SNR as a function of range, calculated as per the above-mentioned methodology. The analysis pipeline was kept identical in all respects to that applied to the February collection, with the only changes being the path indicating the location of the new imagery and the number of images collected. A preliminary overview indicates that the data is well-behaved and in accordance with the model.

Strong cross-winds challenged the constant-trajectory flight plan, as frequent course corrections were required to maintain the collision trajectory. Some slippage was inevitable despite best efforts. In consequence, variations in the observed signal were larger (see Table 6.4) and target drop-outs were observed in some flight trials.

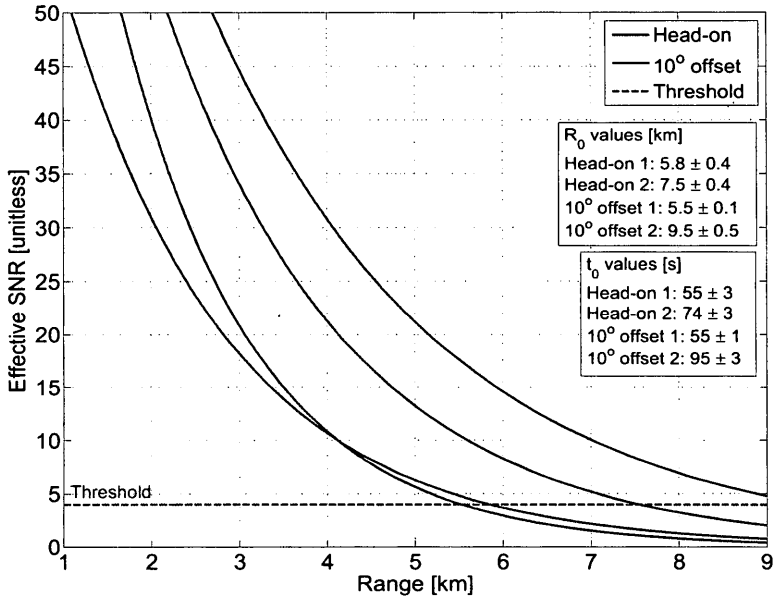


Figure 6.12: Evaluation of SNR curves for multiple events from the March 2012 collection (Table 6.4).

This effect was most felt the 10° azimuthal offset cases, which were at the limit of the field-of-view of the instrument. However, since the methodology employed a robust least-squares fitting technique, degradation was graceful, and R_0 could still be estimated for those runs with a corresponding increase in error. The results are detailed in Table 6.4.

A mean $\overline{R_0} = 7.1 \pm 1.8$ km was calculated from the observations. Both the mean and the standard deviation were comparable to the expected value acquired from the February collection, thereby affirming the validity of the model and the technique.

Table 6.4: Statistics on head-on and offset collision geometries for March 07, 2012.

Type	Time [h:m:s]	Heading $\pm 2\sigma$ [deg]	Sun (Az, El) [deg, deg]	$R_0 \pm 2\sigma$ [km]	$t_0 \pm 2\sigma$ [s]
Head-on	9:35:14	-124 ± 2.8	(68, 25)	5.8 ± 0.4	55 ± 3
10° offset	9:43:59	65 ± 5.0	(-102, 28)	7.5 ± 0.4	74 ± 3
Head-on	9:48:59	-114 ± 2.2	(71, 29)	5.5 ± 0.1	53 ± 1
10° offset	9:54:59	88 ± 3.2	(-107, 30.2)	9.5 ± 0.5	95 ± 3
Mean	-	-	-	$\mathbf{7.1 \pm 1.8}$	$\mathbf{69 \pm 20}$

6.10 Impact of Resolution on R_0

The instrument angular resolution of 0.1 mrad was designed to fill a pixel with a 1.0 m^2 target at 10 km. The present target (Bell 206) cross-section of 1.9×3.5 m has an angular extent that is five times larger, thereby exposing the potential for resolution-based analysis. Fig. 6.13 depicts the observed target SNR when the imagery is sub-sampled.

Curves were generated by a bilinear sub-sampling of the image sequence. Factors of 2 and 4 were used, resulting in equivalent resolutions of 1.0 megapixel and 0.5 megapixel respectively, from the raw 2.0 megapixel image. In the absence of motion-induced MTF effects, it was expected that R_0 would be directly proportional to resolution to the limit of atmospheric attenuation. This hypothesis was validated from the curves. It is noted that averaging increases precision, as expected, but is capped to the limit of the intrinsic error.

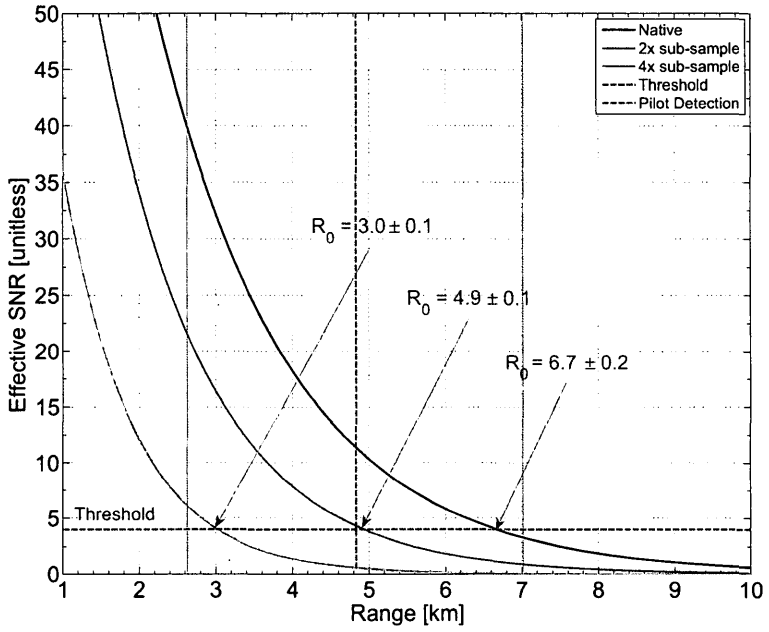


Figure 6.13: The impact of resolution on the range at first detection.

6.11 Target size and growth

The apparent target size within an image is a useful parameter for estimating the system point-spread function (PSF). Section 5.3.5 depicts the Gaussian function approximation to the system PSF. At the hyperfocal limit, blurring and other optical artifacts are minimized for a given lens, therefore knowledge of the pixel pitch, lens focal length and target geometry should provide a useful first-order comparison between the measured PSF and the expected projected target size.

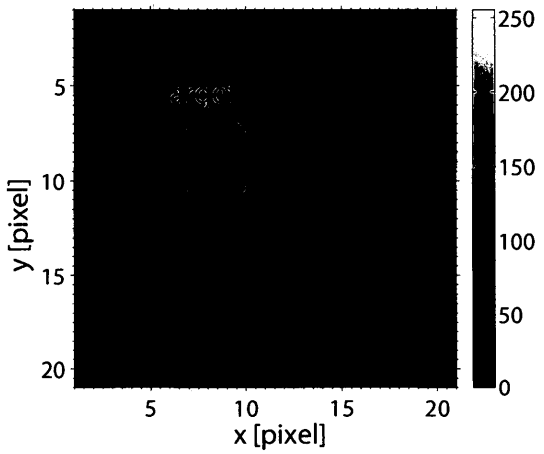
6.11.1 Estimating the target size

Fig. 6.14a shows 21×21 crop of the target at a range in excess of 6.0 km, approximately the R_0 limit. Highlighted is a circular reticle indicating the approximate location of the target. Fig. 6.14b shows a 3-D relief map of the same image with the mean local background subtracted. Note that the target is quite difficult to see in the image, but is readily apparent in the relief map.

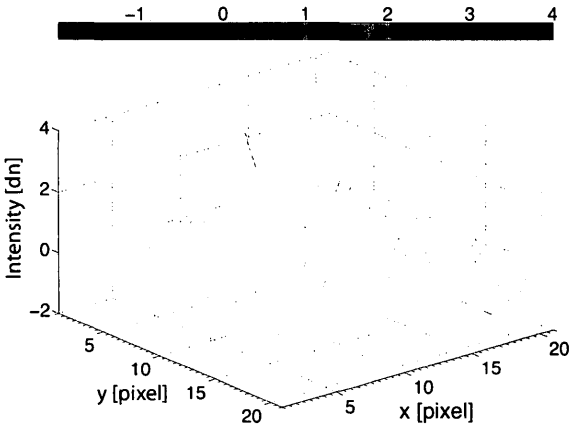
Fig. 6.14c shows the Gaussian fit, $\hat{\mathcal{N}}(x, y)$, to the target profile. Background noise in the image influences accuracy of the fit. Error estimates were procured from the reciprocal diagonal elements of the Hessian matrix, with the covariances ignored. The full-width at half-maximum ($FWHM$) is the key parameter of interest and can be summarized by the formula in Eq. 5.11. The value is determined independently per axes as per the earlier i.i.d assumption. Table 6.14d has the relevant fit parameters listed with 1-sigma error bounds. The FWHM values can be compared to the geometric target angular size in the absence of any point-spread functions.

6.11.2 Perpendicular crossings

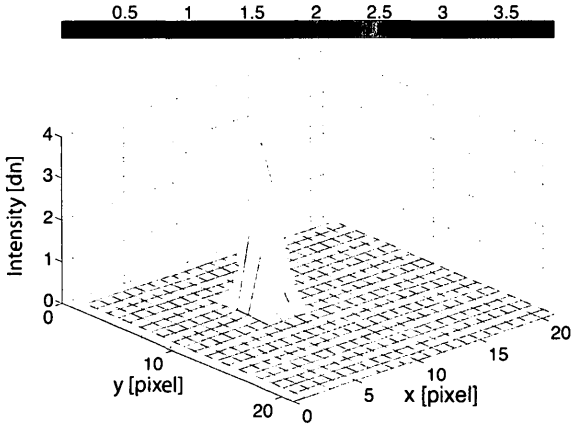
Typical head-on runs, although critical, reveal little about the shape of the target at a given range, since only a few images are available. Perpendicular crossings



(a) Crop of intruder



(b) Background-subtracted raw profile



(c) 2-D Gaussian fitted to intruder signal profile

Parameter	Value	Units
$\hat{\mathcal{N}}(\mu_x, \mu_y)$	4.5 ± 0.3	dn
μ_x	8.32 ± 0.02	pix
μ_y	8.77 ± 0.01	pix
σ_x	0.82 ± 0.01	pix
σ_y	0.58 ± 0.01	pix
$FWHM_x$	1.93 ± 0.03	pix
$FWHM_y$	1.37 ± 0.03	pix

(d) Fit parameters for 2-D Gaussian

Figure 6.14: Estimation of the apparent target size at $R_0 = 6.7$ km

were vital in allowing the measurement of target attributes at quasi-fixed ranges. Fig. 6.15 presents the vertical size of the target in pixels for a perpendicular run, depicted with a trend-line. Two x- axes are presented for comparison, one at range and the other at normalized time. Note that the closing rate is non-zero as the host-

platform is still flying forward in level-flight. A dotted line highlights the geometric shape at a given range.

Large random variations can be observed in the vertical dimensions. It is hypothesized that much of the fluctuation is due to rolling shutter wobble resulting from random vibrations of the nodes. The trend-line indicates a lower-floor to the y- axis profile. It is anticipated that the difference between the trend intercept and the geometric intercept can provide a first-order estimate to the PSF lower-bound, the bulk of which is fixed and therefore optical in causation, on the order of ≈ 2 pix.

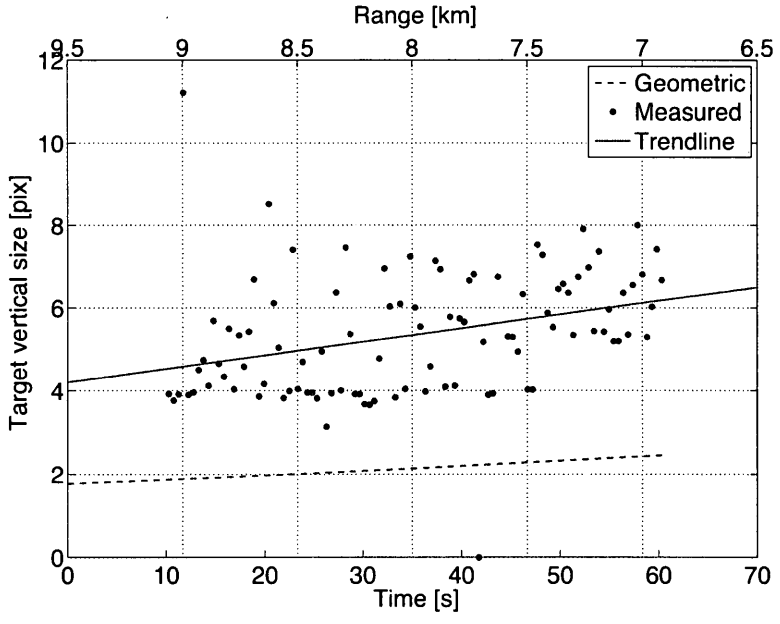


Figure 6.15: Variation of the vertical target size as a function of time for perpendicular crossing runs.

In perpendicular flight, the apparent target profile typically includes the tail

section, which dramatically alters the aspect ratio compared to the head-on case. The Gaussian approximation is a poor fit to the more complex shape, therefore target width estimates based on the Gaussian fit are quite untrustworthy. However, our choice of model treats x- and y- axes independently, and the y-axis is quite precisely fitted, therefore comparing estimates in the vertical axis and applying rotational symmetry should offer a good, first-order, baseline estimate of the PSF.

6.11.3 Target growth in a head-on case

Fig. 6.16 illustrates relative target size measured as per range along the medial and transverse cross-sections, where the size measurements correspond to the 2D Gaussian-fit profile calculated as per the methodology in Section 6.11.1.

Oscillations are noted and are likely due to target orientation fluctuations. Rolling shutter wobble due to the CMOS sensor utilized may also contribute to under-scan/over-scan artifacts, although that was not explicitly studied.

Fig. 6.17 depicts the target area as a function of range, generated by calculating the area empirically from Fig. 6.16. The range-dependent fit function is calculated by performing an inverse square-law fit ($f(x) = \frac{a}{x^b} + c$) to the data points. The fit contrasted to the geometric profile ($1/R^2$) for the target at the range R. Note that at longer ranges, atmospheric scattering dominate and the target signal merges into the background.

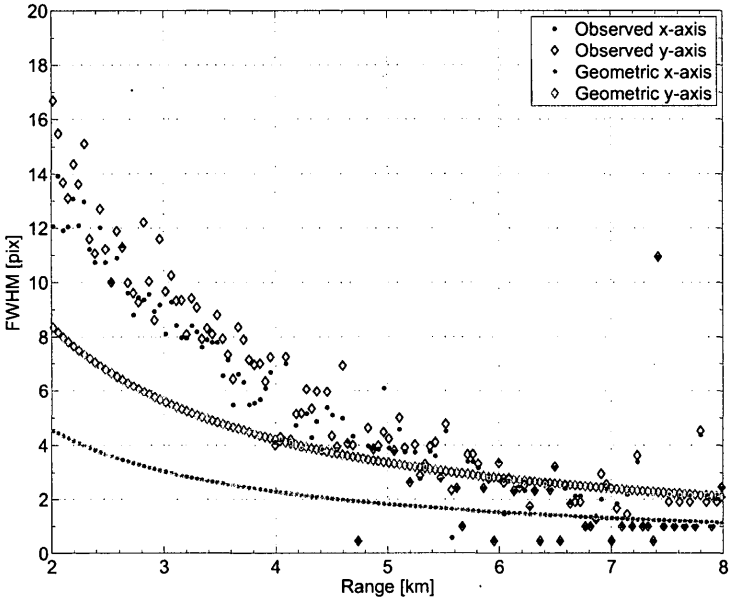


Figure 6.16: Variation of target size as a function of range along the lateral (x) and transverse (y) axes.

At close ranges, the quality of the per-image 2D-Gaussian profile-fit to the target is reduced as the target fills the region-of-interest, thus preventing the signal, and consequently the fit, from reaching the noise floor. Occasional spikes were due to a breakdown in the experimental analysis conditions, in particular the 2D Gaussian fit appeared to be sensitive to noisy images. Note that these spikes never occurred in x- and y- axes simultaneously, were not significant in the area curve and could be safely ignored as outliers by any robust fitting mechanism.

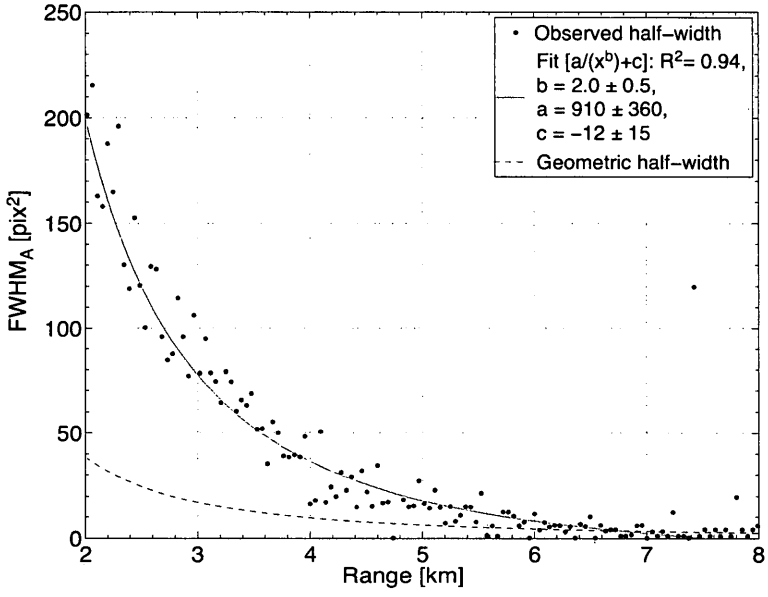


Figure 6.17: Variation of target area as a function of range.

6.12 Conclusion

In summary, flight tests were conducted covering a range of operational scenarios under specified assumptions. A single airborne estimate of $R_0 = 6.7 \pm 0.2$ km under typical visual meteorological conditions can be more precise than the typical pilot measurement under the same conditions (4.8 ± 2.2 km). Comparing R_0 values over multiple runs ($\overline{R}_0 = 6.3 \pm 1.7$ km) reveals larger variations, which can be attributed primarily to changing scene conditions. Such variability is expected, given the abject difficulty in obtaining precisely identical scene conditions for multiple trials. Target size and growth were discussed, and compared to expect PSF values based on modeling. Secondary topics such as the impact of resolution and apparent target

proximity to the horizon were also presented, and confirmed to be in line with modeling specifications. The horizontal proximity metric, which is constrained by the system point-spread function, was expected to aid in the measurement of the target collision probability.

7 Algorithms: Localization and Detection

7.1 Synopsis

This chapter presents the image processing pipeline required to establish the presence of the target in the field of view of a single smart camera. The proposed processing pipeline is hierarchical in nature and decomposes into distinct noise removal, localization and detection stages, which are subsequently discussed. The proposed implementation represents a choice of techniques for each of the individual stages. The techniques utilized arise naturally from the figures of merit presented in the previous chapters, and represent one possible solution in a myriad of choices. The implementation was tested offline on synthetic and in-flight extracted image sequences. A real-time implementation of the full pipeline was not realized, although the choice of techniques guarantees a relatively straightforward path.

7.2 Introduction

Detecting a small target amid noise and clutter is the principal task of any processing pipeline utilized for target detection. This chapter presents the general theory behind hierarchical processing, followed by a pre-processing pipeline to achieve this detection on a single camera node. Figures of merit, outlined in the previous chapter, serve as the driving force for the development of the proposed pipeline. One technique is presented for each stage, followed by a performance comparison of the overall pipeline to measured detection range values. The choice for algorithms is deliberately local, causal and low-computational-cost, to expedite implementation a low-power pre-processing board.

Note that the algorithms/techniques presented here are *representative*, to show the efficacy of the pipeline. Although they were chosen to provide a good trade-off between efficacy and computational performance, they were not explicitly compared to others, and therefore do not represent an optimal choice, either in detection range or in computational performance. The hardware and software design of DragonflEYE were selected, in part, to facilitate empirical experimentation of the image processing options. However, the comprehensive evaluation of algorithms is beyond the scope of this thesis.

7.3 Hierarchical processing pipeline

7.3.1 Rationale

The primary rationale behind the choice of the smart camera is its ability to extract useful information from the scene prior to upload of image data to a central processor. In this manner, bandwidth bottlenecks typical to large camera arrays may be circumvented.

Here we consider a binary partitioning of the detection problem into pre- and post- processing stages. The pre-processing stage is mostly concerned with the detection of likely targets from a single raw image. The post-processing stage is directed towards the sorting and classification of likely targets, culminating in the determination of a probabilistic collision track. Further discussion in this thesis is restricted explicitly to the pre-processing pipeline, where pre-processing is explicitly assumed to refer to activities conducted locally on each smart camera node.

It is noted that, in terms of linear-time-invariant (LTI) system theory, the pre-processing algorithms as described are assumed to be memoryless systems while post-processing algorithms will be causal systems. This distinction is arbitrary and motivated by the computational performance of any cost-effective modern implementation of a smart camera. It does not indicate any inherent limitation in the smart camera paradigm, as the capabilities of a smart camera are expected to scale

with computational advances.

7.3.2 Smart camera processing pipeline

Fig. 7.1 illustrates the pre-processing pipeline envisioned for each smart camera. The input is a raw camera image, while the output is a ranked list of hits.

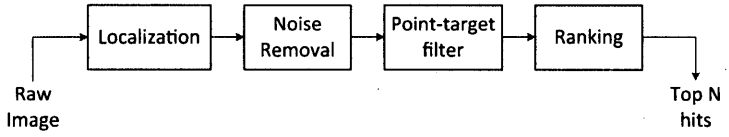


Figure 7.1: Smart camera processing pipeline.

As indicated in the flowchart, the pipeline can be divided into four principal stages, namely localization, noise removal, point target detection, and ranking.

7.4 Target localization

A typical aerial scene can be bifurcated into ground and sky segments, with the horizon emerging as a natural boundary. Emerging targets may be imaged against either of the two backgrounds. Given the differences in solar orientation, textural composition and spectral characteristics, different background models for the ground and sky may be developed. The presence of clutter (non-interesting features similar to targets) is significant in both segments.

The required detection algorithm can be tremendously simplified if the region of interest can be localized to a subset of the acquired scene. In most cases, targets are expected to be sparse and localized to regions near the horizon. Note that the cameras rigidly mounted to the host platform still need a relatively large field of view to compensate for platform orientation changes and shifts.

Figure 7.2: Sample aerial capture with a wide-angle lens.

In real imagery acquired at general aviation altitudes under clear sky assumptions (Fig. 7.2), the horizon is routinely observed to possess a smooth gradient (Fig. 7.3) rather than a sharp discontinuity near the ground-sky region. This 'gray' bounded region is a direct consequence of atmospheric effects and illustrates the complexity of extracting the horizon contour using purely image processing

techniques.

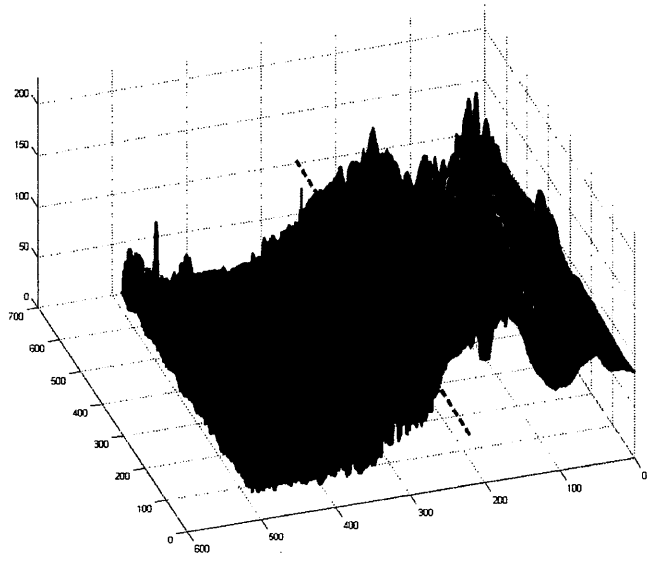


Figure 7.3: The horizon occurring as a smooth gradient in typical aerial images.

A solution, therefore, is to exploit the differences between ground and sky scene components to extract the horizon boundary. In IMU-denied situations, such as micro air vehicles (MAVs), an algorithm may be necessary to achieve a horizon contour. The values from both can be compared to ground-truth values extracted via manual selection from image sequences. This section describes a shape-agnostic machine learning technique as one such solution to the horizon

7.4.1 Comparison to prior work

Linear approximations to the horizon contour exist in literature. The work in [ENIW02] is directed towards colour images, where the absence thereof as a classification metric reduces the effectiveness of the technique for our situation. This is most evident when the images contain smooth intensity gradients or strong textural components. Refinements to ameliorate the effects were proposed in [FSHG06]. An alternate method utilizing the Radon transform is described in [BXZ05]. Simple line-fitting algorithms, such as the Hough transform, can also be employed.

Local implementation of [FSHG06] was tested on synthetic and natural images, with the implementation details omitted for brevity. The first test sample utilized a binary image with a distinct edge. After successfully matching the "horizon" to the trivial case, randomly selected images through a Google search were tested. Fig. 7.4 illustrates a sample, likely acquired at ground-level, with the detected horizon indicated in green. Note the offset of the detection from the true horizon due to vegetation. The algorithm proved to be robust against image discontinuities, as evidenced by correct performance despite the sharp acquisition misalignment.

In Fig. 7.5, the algorithm estimate is a poor fit to the actual horizon line. This image is simulated from the previously-described MATLAB ray-tracing environment, with the background modeled on imagery acquired by the Flight Research

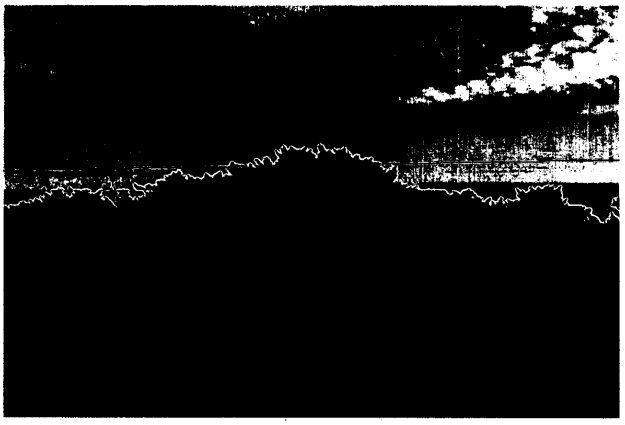


Figure 7.4: Horizon detection on a random image utilizing a linear approximation [FSHG06].

Laboratory at NRC Canada. It is hypothesized that the absence of suitable texture discrimination and the inability to distinguish smooth gradients limited the efficacy of the technique.

7.4.2 Contour extraction via machine learning

Establishing the contour broadens the linear estimation concept, by classifying each pixel, or pixel 'neighbourhoods' (windows), into ground or sky. The output is a horizon contour sensitive to terrain discontinuities, artifacts and structural characteristics. Discrimination between ground and sky pixels can be performed by algorithmic classification or machine learning techniques. Described herein is a classification system based on the principle of a support vector machine, which is

Figure 7.5: Output of the horizon detection algorithm on simulated imagery using a linear approximation [FSHG06].

divided into training and testing sections. The algorithm is entirely autonomous, including during in the training stage, with the exception of parameter weights.

7.4.3 Algorithm template

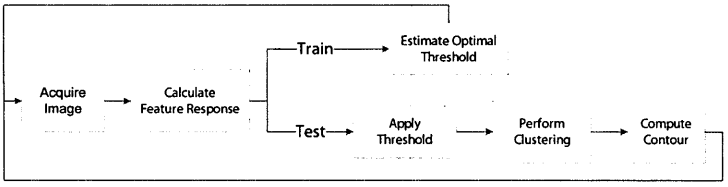


Figure 7.6: Algorithmic template for machine-learning contour extraction.

Fig. 7.6 depicts the data-flow diagram for the contour extraction algorithm. The following set of constraints are imposed on the learning problem:

- The scope of possible aircraft attitudes is restricted such that segments above the contour are always assumed to be sky, while segments below are always assumed to be ground. However, it is noted that determination of the actual contour as per the technique described herein is agnostic to this definition.
- Ambient illumination and scene structure have low temporal variance, and aircraft behaviour is gradual, not dynamic.
- Sufficient ground and sky pixels are available in the training frames for classification to be reliable.
- Adequate time is available to train the system, before transference into test/-operating mode.

These constraints were fully satisfied in the detailed discussion below. Note that training may happen periodically as scene conditions evolve, the frequency of which can be determined heuristically.

7.4.3.1 Training

Given an appropriately selected training image, a large number of random pixel neighbourhood locations can be generated via Monte Carlo sampling. The number

of locations and the size of the pixel neighbourhoods can be adjusted in response to sensor and scene requirements.

7.4.3.2 Feature descriptors

Assuming mutual exclusivity, the segmentation task can be accomplished by a set (Eq. 7.1) of feature descriptors (p_j) and their weightings (w_j), operating on sampled neighbourhoods ($\mathbf{x}_i$) of the image.

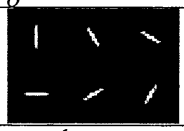
$$\begin{aligned} F(i) &= [w_1(\mathbf{x_i} \star p_1), w_2(\mathbf{x_i} \star p_2), ..., \ w_N(\mathbf{x_i} \star p_N)] \\ &= [f_1, f_2, ..., f_N] \end{aligned} \tag{7.1}$$

This framework is powerful in its ability to incorporate any spatial feature descriptor. Examples of ones not considered here include a Harris corner detector and a median filter. Table 7.1 highlights the ones utilized here, divided into statistical, perceptual, structural and Fourier categories. As noted earlier, this analysis is representative and relative comparisons between classifiers were not performed.

7.4.3.3 Class discrimination

Each pixel neighbourhood contributes one data point to the N-dimensional classification space, with N being the number of classifiers used. Neighbourhoods with similar characteristics are expected to cluster together in parameter space. If the

Table 7.1: Interesting feature descriptors

Type	Name	Formula
Perceptual	Highlight	$max(\mathbf{x})$
	Shadow	$min(\mathbf{x})$
	Contrast	$\frac{max(\mathbf{x}) - min(\mathbf{x})}{max(\mathbf{x}) + min(\mathbf{x})}$
Statistic	Mean	$\frac{1}{m}\sum_{i=1}^{m} x_i$
	Smooth	$1 - \frac{1}{1+\sigma^2(\mathbf{x})}$
	Skew	$\frac{1}{\sigma^3} \cdot E(\mathbf{x} - \mu^3)$
Structure	Gabor	
Fourier	High-pass	$G = \begin{cases} 0; \omega \leq t \\ 1; \ else \end{cases}$

neighbourhood sample space is large, a sum of Gaussian PDFs (probability density functions) can well-approximate the statistical distribution of neighbourhoods, with each PDF denoting a cluster.

A search over a sparsely populated n-dimensional space is computationally prohibitive. Solutions exist to diagonalize the problem, some of which include PCA (principal component analysis), sparse matrix analysis, and network flows. Here, a linear diagonalization is pursued for computational ease at the expense of accuracy. One can simplify the search-space by projecting each n-dimensional feature vector onto a line, defined by its Mahalanobis distance [Mah36]. In the limit where the classifiers are orthogonal, the covariance matrix becomes diagonal, and the Mahalanobis metric simplifies to a scalar Euclidean distance, represented as a function

in Eq. 7.2.

$$E(i) = \sqrt{f_1^2 + f_2^2 + ... + f_N^2} \tag{7.2}$$

The choice of good classifiers is therefore critical, where good means orthogonal, for the Euclidean assumption to hold. Note that the choice for Mahalanobis distance as the figure of merit is driven by convenience. Other mechanisms for space reduction, such as eigenvectors (PCA), sparse matrix analysis, or sub-space projection among others, are considered to be equally valid.

7.4.3.4 Minimum-error thresholding

Fig. 7.7 shows a histogram of the distance function, with classified clusters clearly visible. Note that there is some overlap between the 'ground' peak (lower abscissa value but higher peak) and the 'sky' (higher abscissa values). Here, lower for ground and higher for sky are assigned from external constraints. The degree of overlap gives the misclassification rate. Similar curves can be created by projecting the clusters onto other axes of the classification space, in order to assess the segmentation efficacy of given features.

An optimal threshold [Ots79] minimizes the inter-class variance between the two clusters. Kittler's minimum-error thresholding [KI86] can be used if the clusters are assumed to Gaussian in N-dimensional feature space. Note that this implies that the

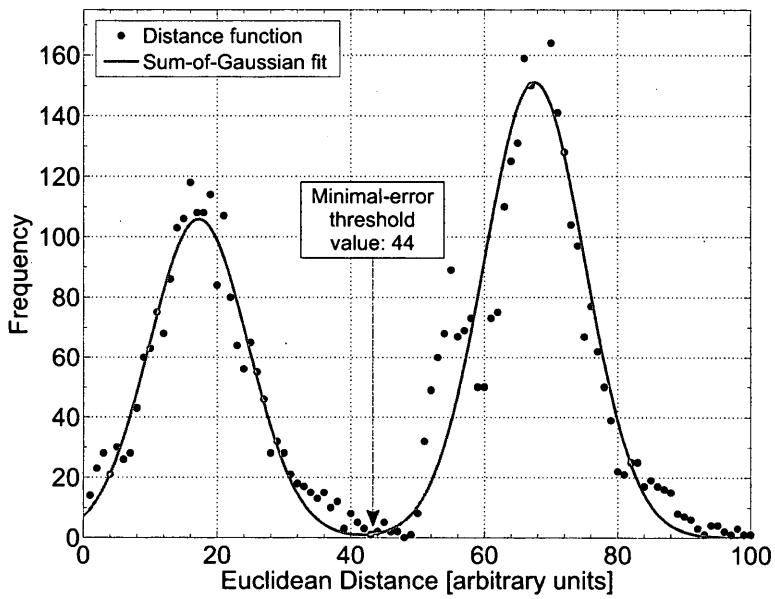


Figure 7.7: Distance vector histogram of the ground test image (Fig. 7.8a)

features are orthogonal and noise is independent and identically distributed (i.i.d). This assumption held true for the images tested, but a more general method may need to be considered in the future, such as the Fischer linear discriminant [DHS00]. The arrow in Fig. 7.7 indicates the optimal threshold value. Note that neither the number of samples used in the learning process nor the number of bins used in the histogram/thresholding stage were optimized. Sudden scene changes (sunlight, emergency manouevres) may require a different threshold. Periodic learning can mitigate the impact of such effects, where the run-time cost of learning is amortized over the time interval.

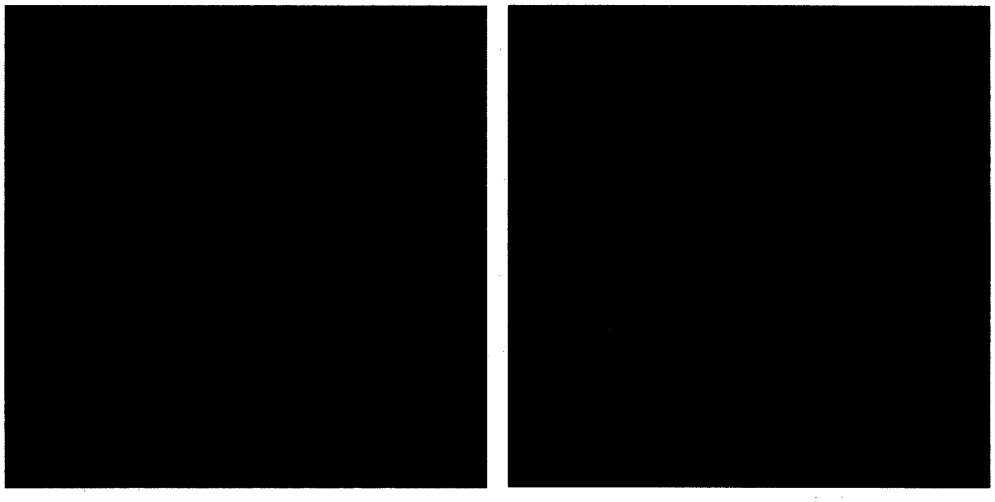
7.4.3.5 Clustering

Once the threshold is learned, further images in the sequence are processed by a sliding window operation, where the feature vector and distance metric are computed per pixel position. Computational run-time can be reduced during implementation by restricting processing to a region-of-interest (ROI) sub-set, selected either by sub-sampling the image, or based on external knowledge from other on-board systems.

Pixel neighbourhoods may be misclassified, causing small 'pockets' of sky in ground and vice versa. Although it is desired to keep these to a minimum, region growing can help eliminate grossly incorrect pockets, while maintaining the location and structure of the horizon. Region-growing was implemented here by a sequence of erosion operations followed by an equal number of dilation steps, often referred to as morphological closing [GW08]. Water-shedding poses a good alternative, although it was not tested. The closing operator, however, is sensitive to window size. Small windows can result in insufficient pocket removal, while large windows may cause loss of contour detail, although few pockets far from the contour were not considered detrimental. A longest-edge gradient extraction gave the resulting contour. Errors can be balanced among the learning and clustering steps to optimize the use of computing resources.

7.4.4 Tests against representative images

Fig. 7.8a shows a 'clear sky' horizon as seen from close to the ground. At higher altitudes (300 m) and in the presence of clouds, the horizon can be more indistinct (Fig. 7.8b). There may also be a 'boundary' region that prevents precise determination of the horizon line.



(a) Low altitude test image. (b) 300 m (1000 ft) altitude test image.

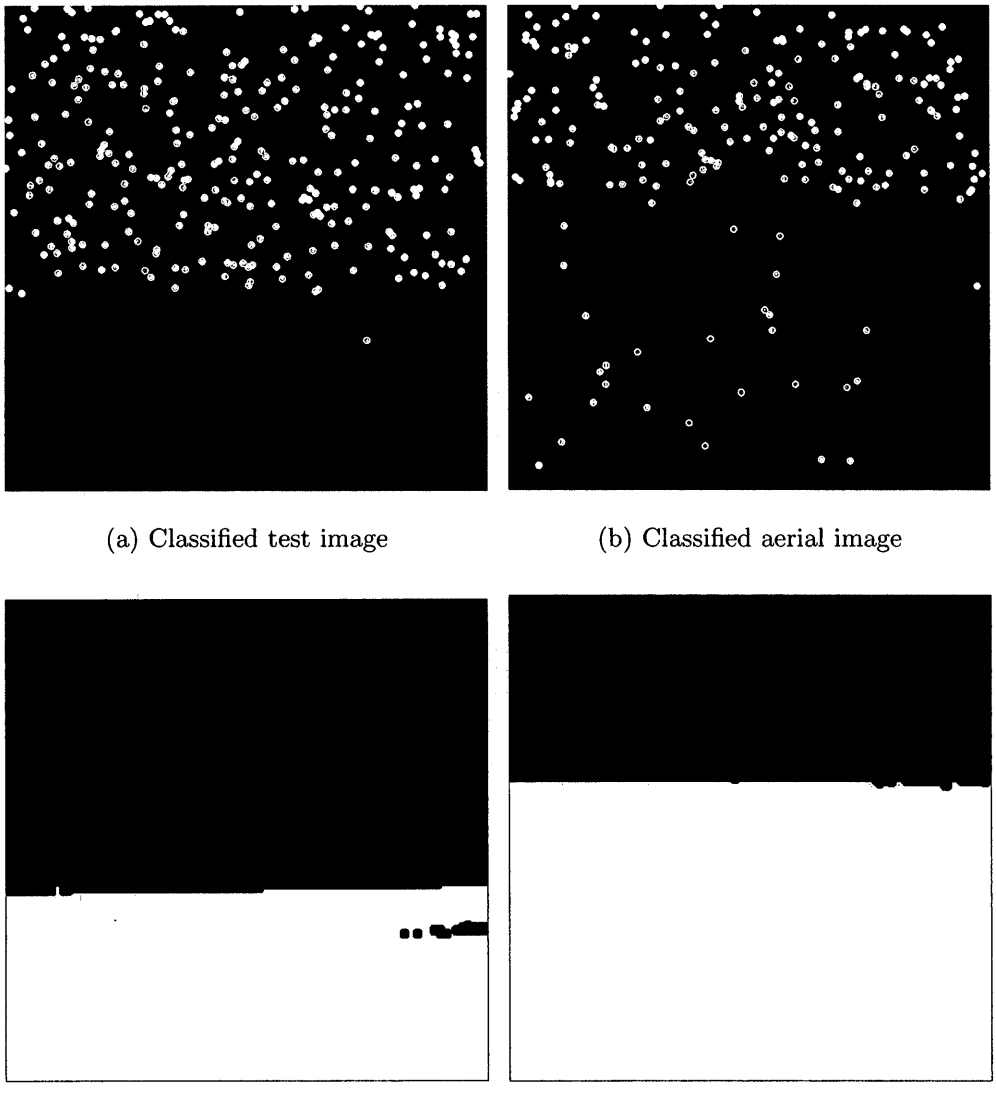
Figure 7.8: Horizon images acquired at different altitudes

Fig. 7.9a illustrates the ideal image from Fig. 7.8a, classified into ground and sky. The dots are randomly selected points that have been classified, with lighter dots indicating sky, and darker dots for ground. Fig. 7.9b, on the other hand, presents the diffuse horizon boundary of Fig. 7.8b. This image is part of a large dataset acquired during the flight of an airborne NRC Bell 205 surrogate UAS.

The set was acquired from three side-mounted cameras, positioned at 0, 45 and 90 degree azimuth offsets respectively. Cameras shared a common mount-plate nominally aligned with the aircraft's pitch axis. Each image was tagged with correlated instrument readouts. Classification was conducted on a set of 5000 pixel neighbourhoods (11×11) extracted from the training image via random Monte Carlo sampling (for clarity, fewer samples are shown in Figs. 7.9a and 7.9b). Parameters and weights were tuned by hand, but are expected to be adaptive in future work. In particular, the skewness measure hindered rather than helped, and was turned off, while the mean response increased misclassification and was reduced.

An 11×11 pixel neighbourhood was used for clustering, with the size kept constant for all classifiers and the sub-sampling set to the single-side dimension ($k = m = 11$), essentially non-overlapping windows. Fig. 7.9c depicts the contour for the straightforward horizon image, while Fig. 7.9d outlines the contour extracted from the sample flight image. In both test cases, the training and test images were kept the same. A black border was added around both images strictly for illustration purposes. Note that mis-classification can occur even when the horizon is distinct (Fig. 7.8a), emphasizing the importance of the clustering (and curve-fitting) steps.

Fig. 7.10 depicts an airborne image of the 205 in level flight. It is observed that the extracted contour is very robust against false positives such as crop lines, hedgerows, cloud boundaries and other strong gradients, and is better than typi-



(a) Classified test image

(b) Classified aerial image

(c) Test contour

(d) Aerial contour

Figure 7.9: Classification and contour estimation of test versus realistic imagery

cal linear estimation algorithms under similar conditions. It behaves well despite readily observable pixel noise. Finally, note that the contour depicts a bow in the

horizon, perhaps indicating the Earth's curvature.

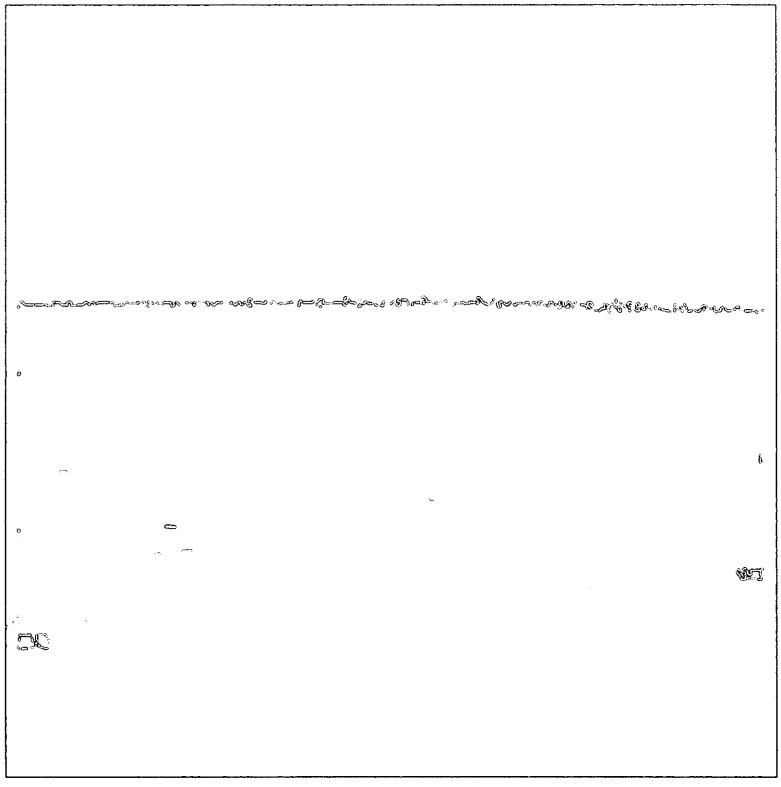


Figure 7.10: Horizon extraction from a forward-looking camera image during level flight. The extracted contour is marked in white. Brightness was boosted to increase readability.

Similarly, Fig. 7.11 depicts an airborne image during a banking manoeuvre. Note that the algorithm performs just as well, signifying that it is agnostic to orientation. This is a key distinction from linear estimation algorithms, such as the Hough transform, which may require a model of the line (position and orientation). Cartesian and polar linear models have their respective trade-offs. Here, the algo-

rithm side-steps the modeling constraint by not requiring one. Also note that the contour is completely connected. However, since it is only a single pixel in width, it may not be visible due to compression and resizing.

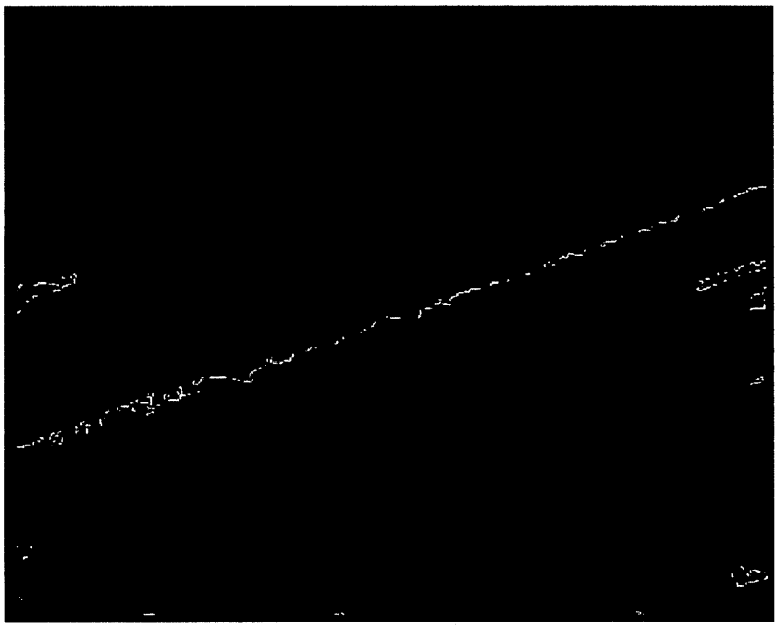


Figure 7.11: Horizon extraction of an image during a banking manoeuvre.

7.4.5 Measurement of aircraft roll

The technique was then applied to a contiguous image sequence extracted from the test data. Typical flight images were at a resolution of 5.0 MP. For convenience, images were restricted to a square, 1.0 megapixel ROI centered in the raw image. The first image in the sequence was chosen for training. Contours were extracted

from subsequent images as described. Initially, the extracted contour on any specific image was fitted to a line via least-squares. The polynomial fit to the sky/earth boundary was dominated ($> 95\%$) by the linear terms and the errors in slope and intercept were within 1%.

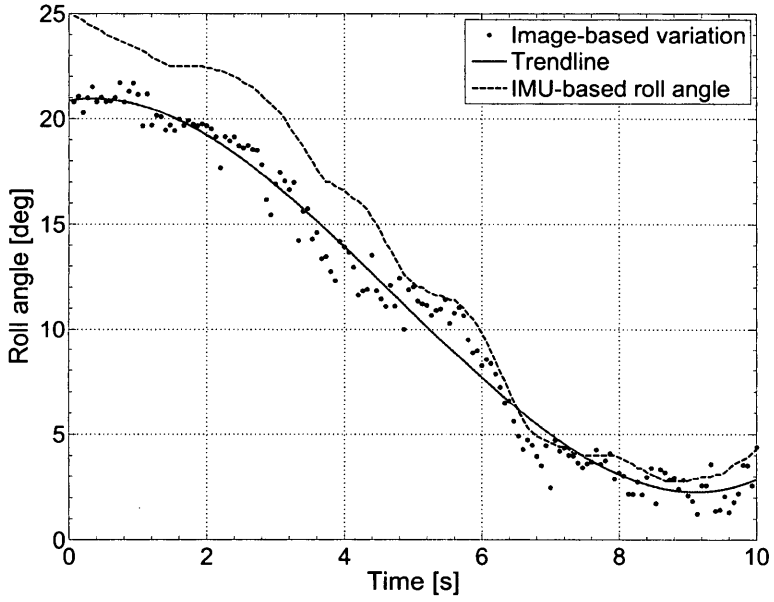


Figure 7.12: Estimated aircraft roll versus IMU measurements

The sequence starts with the aircraft in a banking manoeuvre and ends as it begins to settle, covering a period of 10 seconds. Fig. 7.12 shows the estimated slope of the line as a function of time for successive images, along with correlated IMU measurements. Because the camera module for Fig. 7.12 was facing forward, the slope corresponded approximately to the aircraft roll, as depicted by the dotted line. Image-based estimates of roll are denoted by dots, with a solid fit line showing

the general trend.

A comparison of the horizon with the aircraft attitude, as simultaneously recorded from its attitude sensor, shows a strong correlation. The horizon had both periodic oscillations of ± 2.0 mrad in amplitude and high-frequency components. The amplitude of the high-frequency components is > 2.0 mrad, and is expected to include both measurement uncertainty of the image processing and random jitter in the aircraft attitude [SWP+01]. Significant offsets between the IMU and the described technique estimates may be attributed to misalignment between the instruments, coupling of control axes during the manoeuvre, and aircraft dynamics.

7.4.6 Probabilistic ROI selection

It was noted earlier that likely targets are co-altitude and non-manoeuvring, therefore they may pierce the far fence flying at a constant altitude. Given the absence of manoeuvres on the host platform, such targets will emerge at or near the horizon. Therefore once the horizon position is known in the imager, a region of interest (ROI) centred at the horizon may be extracted.

Alteernatively, ROI selection may be treated as an intensity weighting function. A typical cropped ROI represents the weighting as a step function, where pixels outside the ROI are culled (or set to zero). Although computationally expedient, such ROI selection has poor compensation for occasional outliers that may emerge

far from the expected fence region. A spatial probability density function, as described in Chapter 3 can provide a weighting factor per pixel and arise naturally from the likelihood of a target in the scene. In particular, the density function describes the likelihood that the target may emerge within the fence.

Fig. 7.13 illustrates a sample of normalized spatial probability density functions that pre-weight pixels according to their importance. The abscissa indicates the relative pixel value as a distance from the horizon (horizon is at zero). Included are the step function, demonstrating the typical culling ROI, a Bartlett window for a linear roll-off and a Gaussian window for exponential roll-off with increasing distance from the horizon. A window of 50 pixels was chosen for the step function, with the Gaussian matched accordingly, while the Bartlett was depicted at a half-width of 100 pixels.

The weighting function can be stored in the look-up table as a function of distance. Once the horizon is measured, the distance from each image pixel to the horizon can be computed on the fly, and the corresponding weight pulled from the weighting look-up table indexed by this distance quantity. This reduces the run-time, per-frame computation to a single multiplication. Secondly, this mechanism facilitates the use of an adaptive spatial density function, which may grow and shrink as a function of the host, scene or computational parameters. For instance, the fence may be narrowed in vertical and azimuthal width to restrict the

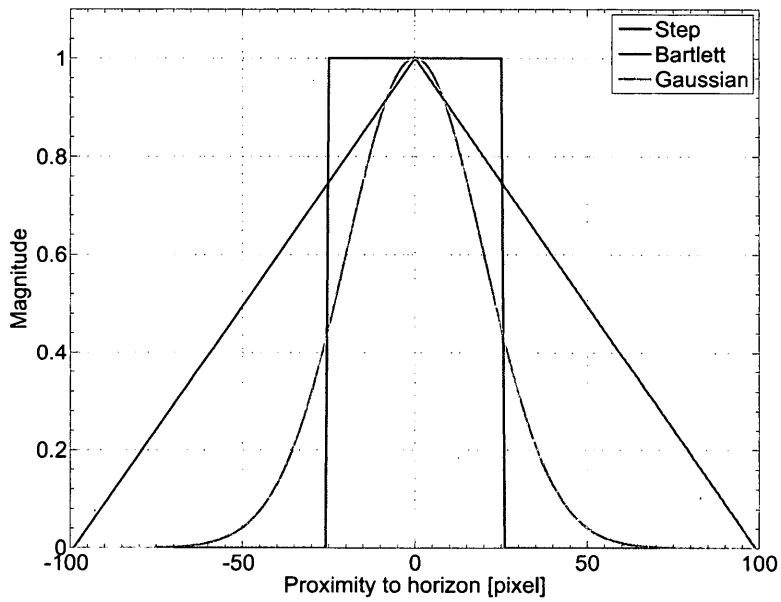


Figure 7.13: Spatial probability density functions for ROI weighting.

search-space as the speed of the host platform increases. The look-up table may be updated periodically by an appropriate machine-learning algorithm operating in a low-priority thread, so as not to adversely affect the critical pipeline.

Fig. 7.14 illustrates a raw capture from one smart camera within DragonflEYE during a head-on collision run. Note that the target is clearly visible within the likely fence region. In Fig. 7.15, an ROI weighting function, centred at the horizon is applied. A Gaussian PDF with a half-width of 150 pixels was used. Although it may be implicitly assumed that the ROI would be centred at the horizon, this is not a requirement of the implementation.

In summary, the ROI selection weighs pixels relative to their proximity to the

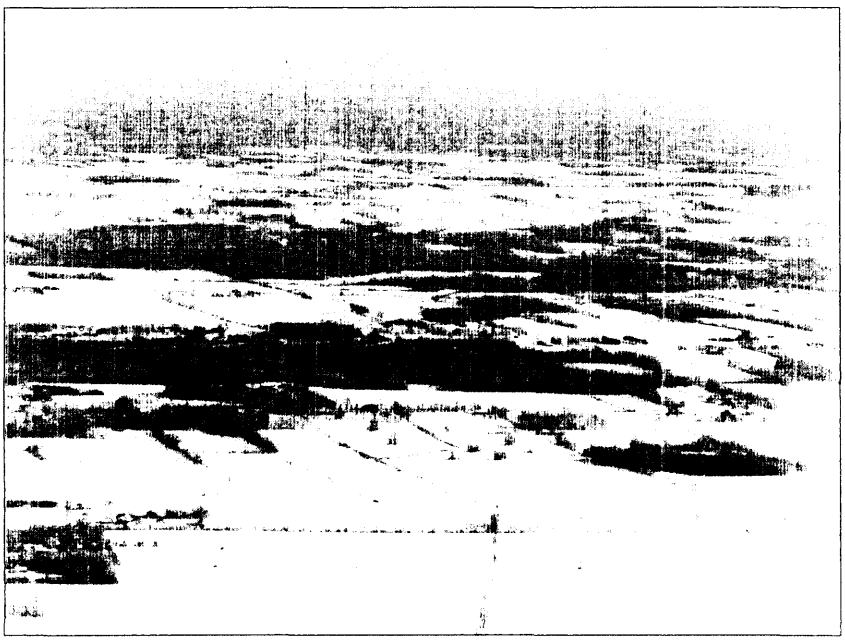


Figure 7.14: Sample image for the bottom-middle camera with a visible target (top-left).

horizon. Pixels far from the horizon are derated by the weighting function, reducing clutter for the subsequent processing steps. Note that the weighting may happen simultaneously as a multiplicative constant during the filter convolution step.

7.5 Noise removal

Noise can both affect the quality of the observed target and generate false targets. A typical forward-looking airborne image during mid-day contains sufficient signal,

Figure 7.15: ROI weighting function applied to the selected image.

such that shot noise is the dominant source of noise. If the sensor is operating in auto-exposure mode and the scene is sufficiently bright, exposure time may decrease and pixel reset noise may also become a factor.

Naive noise reduction algorithms, such as a unity-gain low-pass filter, can result in a loss of intensity and gradient information [GW08]. Since the target usually acts as a point source at the detection limit, the use of a low-pass filter in the pipeline would adversely impact the detection range. Therefore, any noise reduction mechanism under consideration must be edge-preserving to maximize detection

range. The Wiener filter is one such technique, and is utilized here as the representative noise reduction technique. Note that although the classical Wiener filter is not edge-preserving, it is not as destructive as a typical low-pass filter. Furthermore, gradient-sensitive implementations of the classical Wiener filter [KM06] are a relatively straightforward extension.

7.5.1 Wiener filtering

Norbert Wiener, in his seminal paper in 1949 [Wie64], described a filtering technique for ergodic processes with additive white noise. Given spectral information about the signal and noise, an ideal deconvolution filter to remove additive noise can be computed as follows: Consider a received signal, $s(t)$, comprised of the ideal signal $\hat{s}(t)$ that has been corrupted by additive white Gaussian noise (AWGN), $n(t)$ (Eq. 7.3).

$$s(t) = \hat{s}(t) + n(t) \tag{7.3}$$

Then, the optimal Wiener filter, $g(t)$, when convolved with $s(t)$, will return the uncorrupted $\hat{s}(t)$ as the result (Eq. 7.4). The symbol '$\star$' denotes a convolution operation in the following equation. It is noted that since Wiener is an adaptive algorithm, it requires an estimate of the noise model before applying the signal correction.

$$\hat{s}(t) = g(t) \star s(t) = g(t) \star [\hat{s}(t) + n(t)] \tag{7.4}$$

A typical implementation of the Wiener algorithm has the following pseudo-code [JEV11]:

- Estimate the noise covariance coefficient ν_W over the entire image.
- Slide an $n \times n$ window across the image.
- For a window centred at the pixel $im(i,j)$, compute the mean and variance $(\mu_{i,j}, \sigma^2_{i,j})$ over the local neighbourhood.
- Compute the resulting pixel value $\hat{im}(i,j)$ given by Eq. 7.5:

$$\hat{im}(i,j) = \mu_{i,j} \frac{\sigma^2_{i,j} - \nu_W}{\sigma^2_{i,j}} (im(i,j) - \mu_{i,j}) \tag{7.5}$$

Note that the `wiener2` function in MATLAB's image processing library implements this algorithm.

7.5.2 Practical Considerations

The Wiener filter as described is sub-optimal from a computational performance perspective. Algorithmic efficiency may be simplified by two key observations. Note that ν_W represents the noise covariance estimated over the entire image and is computed per image. Depending on the noise source, computational efficiency may be increased by supplying a pre-computed covariance value if the noise is known to be additive and ergodic, such as sensor fixed-pattern noise or dark noise. For noise factors which are a function of the signal, such as shot noise, ν_W may be

estimated in a worst case sense by taking the square root of the average signal for the entire image. One can extend that methodology to an adaptive Wiener filter by considering the square root of the average signal within a window as the locally average noise estimate, i.e. $\nu_W = \sqrt{\mu_{i,j}}$. Therefore, ν_W, can be computed locally and pipelined as part of the windowing operation.

A computational bottleneck also exists in the windowing operation. A naive implementation of the windowing operation will estimate the mean first, since intensity values must be zero-mean before the standard deviation may be computed, resulting in two full parsings over the entire image. Therefore, given a $l \times m$ pixel window containing $k = l \times m$ elements and an n-pixel image, the worst-case run-time for the naive mean and standard deviation windowed computation is given by $O(2nk)$. Simplifying the run-time to just one parsing would increase efficiency by 100%. Such an algorithm does exist, and is known as the Welford algorithm [Knu97].

The Welford algorithm provides a running estimate of the mean and standard deviation along a sequence of observables. It is especially useful when a local, up-to-date estimate of the mean and variance is desired. The algorithm works as follows: Given a stream of incoming values, initialize the running estimators, $\mu_1 = x_1$ and $S_1 = 0$, where x_1 represents the first datum, and the index $j = [0, 1, \ldots k]$. Then,

Eq. 7.7 gives the step in the sequence for an arbitrary value of j.

$$\mu_j = \mu_{j-1} + \frac{x_j - M_{j-1}}{j} \tag{7.6}$$

$$S_k = S_{j-1} + (x_j - \mu_{j-1})(x_j - \mu_j) \tag{7.7}$$

Here, the recursive estimate runs over the whole window, where x_j represents a single pixel intensity value, each. Note that the mean value at the k^{th} sample is given by μ_k, while the variance σ_k^2 can be computed as $\sigma_k^2 = S_k/(k-1)$.

If utilized, the Welford implementation can simplify the Wiener filter implementation to a single-pass algorithm. It would still retain the order of $O(k^2 n)$ as the windowing operation is costly. However, the Welford algorithm now allows the Wiener filter operation to merge with previous and subsequent LTI filters in the processing pipeline, reducing the total run-time.

7.6 Point-target detection

Clutter rejection is a significant challenge in the detection of point targets within a typical aerial scene. Here, clutter explicitly refers to features in the scene that bear similar characteristics to the target, but belong to the background. The presence of ground clutter can complicate the detection of targets, especially in descending head-on and descending overtake cases. Sufficient localization can dramatically reduce clutter, simplifying the detection process. In addition, fewer pixels to analyze

can mean faster computational times, or equivalently, the employment of increasingly sophisticated algorithms.

7.6.1 Foreground and background models

Point-targets may be extracted by 'hunting' the foreground for a suitable signal. Typically relying on a series of spatial filters convolved with the base image, these methods are fast and have a constant computation time, regardless of the complexity of the scene. However, foreground methods require sufficient signal from the target of interest to operate reliably, and often exhibit poor performance in the presence of noise [BBV10].

An alternative technique is to subtract a model of the background from the image. If the model is accurate, the target is uncovered by exclusion. A straightforward implementation consists of acquiring a background image in the absence of targets and subtracting this template background from subsequent images in the sequence. Such a technique is not particularly useful if fluctuations in illumination, random noise, or occlusion are present. A statistical model of the background may be developed, which can adapt over time to account for these fluctuations.

A significant body of literature has been dedicated to background modeling. In a survey article [BBV10], authors Bouwmans, El Baf and Vachon classify background modeling into four distinct categories: Basic Background Modeling [MS95] [LH02],

Statistical Background Modeling [WADP97] [SG99] [EHD00], Fuzzy Background Modeling [BBV08] [SMP08] and Background Estimation [TKBM99] [MMSZ05]. The most frequently used methods are statistical in nature, due to their adaptivity to changes in illumination and robustness to the introduction and/or removal of non-foreground objects.

This dissertation only covers the implementation of foreground detection algorithms. In particular, the Laplacian-of-Gaussian (LoG) filter is presented as the representative algorithm for this pipeline stage.

7.6.2 Laplacian of Gaussian (LoG) filter

Typically utilized in edge-detection and edge-enhancement scenarios, the Laplace operator ($\nabla^2 f(x,y)$) is a second order linearly-separable derivative of the image, as per Eq. 7.8.

$$\nabla^2 f(x,y) = \frac{\delta^2 f(x,,y)}{\delta x^2} + \frac{\delta^2 f(x,y)}{\delta y^2} \tag{7.8}$$

The LoG algorithm takes an image, applies a low-pass (Gaussian) filter to it, then computes the Laplace operator on the low-pass filtered image 7.10.

$$g(u,v) = f(u,v) \star \mathcal{N}(\mu, \sigma^2) \tag{7.9}$$

$$f_{LoG}(u,v) = \nabla^2(\lambda \cdot g(u,v)) \tag{7.10}$$

Here, $g(u,v)$ denotes a Gaussian filtered version of the original image, while

λ denotes a weighting between the smoothed image and the Laplace operator. This operation may be approximated by a kernel convolution, where the approximation accuracy is a function of the kernel size and the degree of connectivity. Eq. 7.11 illustrates a 3x3 4-connected LoG filter kernel ($\mathbf{W_{LoG}}$) and its negative ($\mathbf{Wn_{LoG}} = -\mathbf{W_{LoG}}$).

$$\mathbf{W_{LoG}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \ \mathbf{Wn_{LoG}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{7.11}$$

Similarly, Eq. 7.12 denotes the 3x3 4-connected LoG filter (and its negative) rotated by 45 degrees, in order to encapsulate alternative orientations of targets. Note that the weighting on the diagonal elements is deliberately increased to ensure that the kernel is unity-gain, as typical diagonal values equal $\frac{1}{\sqrt{2}}$.

$$\mathbf{W_{LoG,45}} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \ \mathbf{Wn_{LoG,45}} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \tag{7.12}$$

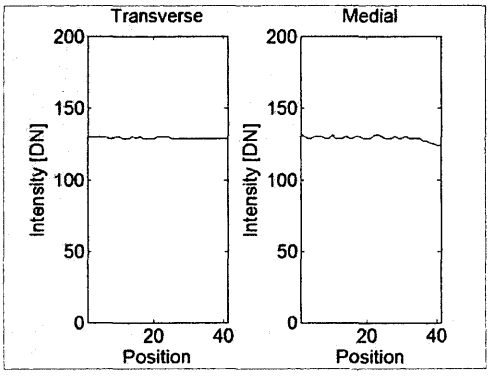
Each 45 degree variant is pre-convolved to its matching 90 degree variant, such that $\hat{\mathbf{W}} = \mathbf{W_{LoG}} \star \mathbf{W_{LoG,45}}$ and $\hat{\mathbf{W}}\mathbf{n} = \mathbf{Wn_{LoG}} \star \mathbf{Wn_{LoG,45}}$. These two composite kernels are convolved separately against the pre-selected region of interest, generating two processed images. This bifurcation is designed to account for cases where the target may be darker than the background. Performance is not affected if the bifurcation is conducted in SIMD architecture as the raw image values are shared.

If a smooth weighting function was applied for the ROI selection (such as the Gaussian function), pixel values below a known threshold are skipped during convolution. If a sufficient number of consecutive pixels are skipped, the kernel window is thought to have moved beyond the region of interest and the convolution loop ends. Note that the sensor can be fully saturated if the sun is directly behind the target, rendering the LoG filter ineffective.

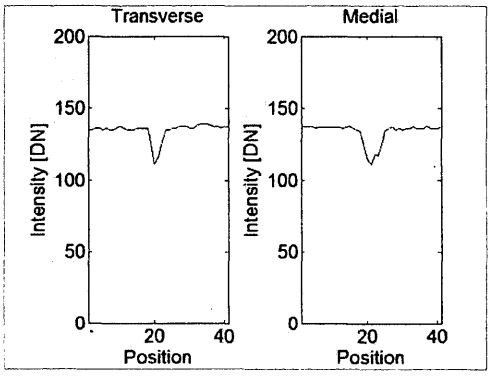
7.6.3 The rationale behind LoG

The Laplacian of Gaussian, as a blob detection filter [LE90], is a defacto choice for point-target detection in a wide variety of scenarios. The use of LoG in target detection here is a natural extension of the target signal cross-section profile as a function of range, as the spatial profile of the LoG filter directly corresponds to the observed target profile. Recalling the discussion on target non-uniformity in Section 3.6.2, consider Fig. 7.16. In this figure, snapshots at intermediate ranges of the target cross-section profiles are depicted, from beyond detectability at 7.5 km to within the inner fence boundary at 1 km.

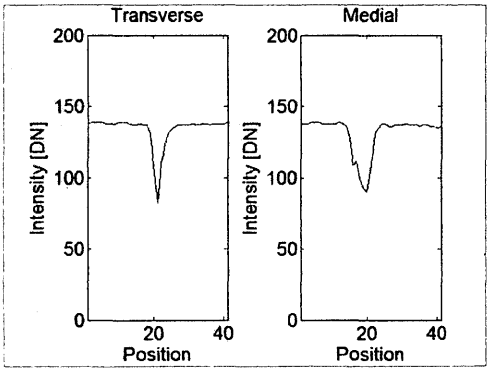
These profiles, in general, reflect the ovoid nature of the target as illuminated by the sun. The sun casts part of a ovoid target into shadow for a wide range of solar orientations, creating a radiance discontinuity. At close ranges, the radiance discontinuity can be greater than the dynamic range of the sensor, causing



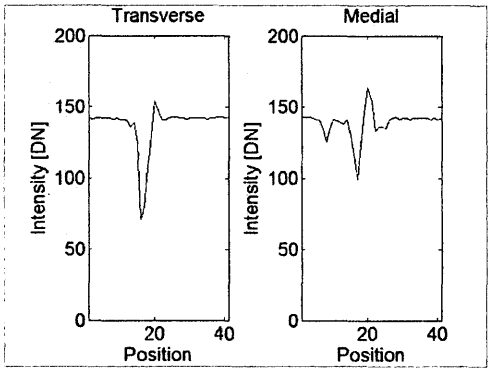
(a) Distance: 7.5 km



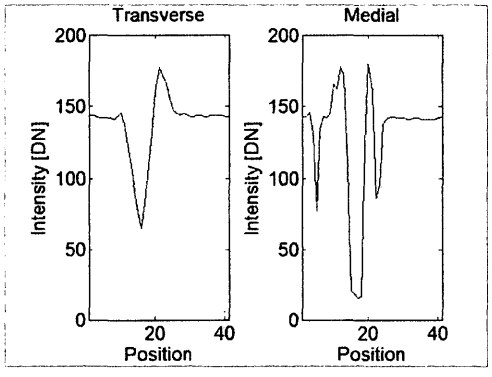
(b) Distance: 5.0 km



(c) Distance: 3.0 km



(d) Distance: 2.0 km



(e) Distance: 1.0 km

Figure 7.16: Selected target cross-section profiles for a head-on run.

saturation and noise-floor artifacts, as evidenced by the flattening of the trough in Sub-fig. 7.16e.

In addition to the matching spatial profile, the LoG filter enjoys computational benefits. As a 2D, symmetric, linear, time-invariant (LTI) filter, the LoG filter is linearly separable into 1D orthogonal components, dramatically reducing the computation time. Since the filter is statically defined, it receives a significant performance boost when implemented on single-instruction multiple-data (SIMD) architectures, as the register value can be carried forward through the computation, reducing the number of load instructions. Furthermore, the LoG filter may be pre-convolved with other 2D LTI kernels to form a composite kernel, reducing the critical run-time to just one convolution parse per image. The range-dependence of both the target profile and the atmospheric PSF suggest that a range-dependent filter kernel may improve performance. However, this issue was not addressed in detail.

7.7 Ranking

The basic output of the processing pipeline under consideration is a 'hit', denoted by pixel location and a response. A filter can be treated as a feature operator or classifier, eliciting responses (or correlations) for individual pixels within the raw image. The ranking operation orders pixels according to their feature response,

where the output is the set of pixels whose response is above a pre-determined threshold. If multiple feature operators are applied to the image, the ranking operation must aggregate these responses, including boosting strong classifiers and rejecting weak ones. Machine learning techniques, such as boosting [FS97], can help in this regard. Here, although two classifiers are utilized to select the pixels of interest, namely horizon proximity and LoG response, we eschew the need for classification by pre-weighting for horizontal proximity in the localization stage. Additional classifiers are relegated to future work.

With the application of the LoG filter, the ranking operation within the generic pipeline (Fig. 7.1) is simplified. The output pixels from each processed image are combined into one list and sorted in descending order by their filter response values. The top N pixels from the sorted list are reported. This *modus operandi* gives the algorithm a very deterministic run-time.

Note that proximity to the horizon and target non-uniformity are explicitly classified by virtue of the ROI selection and the LoG filter stages respectively. It is possible to apply other classifiers at this stage, in a manner similar to the horizon estimation work discussed earlier. Many of the same classifiers may be utilized here, subject to meaningful responses during testing. Note that each additional classifier will incur a computational penalty.

7.8 Tests

The afore-mentioned placeholder techniques can be combined to form a representative pipeline as per Fig. 7.17. Note that this pipeline constitutes a subset of the generic pre-processing pipeline depicted in Fig. 7.1.

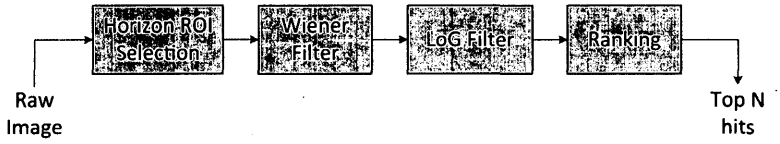


Figure 7.17: Representative processing pipeline for benchmarking.

Although this pipeline provides a good mix of optimal detection range and computationally efficiency, it does not represent the pinnacle in either. The representative pipeline was tested against synthetic targets generated by a MATLAB raytracing model, as well as imagery captured from flight tests.

7.8.1 Definition of $R_{0,proc}$

A figure of merit is useful for quantifying the performance of a fully-automated detection algorithm. Here, $R_{0,proc}$ is chosen to denote the "processed range at first detection", and is explicitly defined as the range at which hits were on target with a high degree of probability and a constant false-alarm rate. The definition of detection probability and constant false-alarm rate are adapted from radar terminology

for images with certain caveats, discussed as follows.

Recalling the definition of a hit from Section 7.7, it is noted that hits are ordered by their response. Therefore, a target spanning multiple pixels may generate multiple hits within the top N hits. Hits that are not on target are defined as false alarms. With these two facts in hand, a target is considered to have been detected if at least one hit out of the top N hits is on target. Therefore, N, as a controllable parameter, provides behaviour similar to the probability of detection in radar theory, but tailored for imagery. Similarly, the probability of false-alarms can be mapped to the remainder of hits in N that are not from the target of interest. One can also observe that increasing N increases the false-alarm rate, as the number of pixels on target are finite in a sparse target scenario.

Of interest is the range at which subsequent images in the sequence (at closer ranges) have consistent hits on target. If the top N hits are observed as a function of range, radiance fluctuations from real targets can cause the target to randomly drop out and re-emerge in the list of hits. This property is especially noticeable at the detection limit. Therefore, the processed range at first detection, $R_{0,proc}$, can be defined as the range within which the target appears in the top N hits for all subsequent images with a defined probability, p_{proc}. In this manner, the performance of the algorithm can be tailored to meet a desired specification.

Note that $R_{0,proc}$, like its counterpart R_0, is empirically defined, and subject to

the choice of 'threshold' parameters, which are selected for the scenario, instrument and operating constraints. Standard choices are provided in the subsequent sections for benchmark considerations, and are not considered to be optimal.

7.8.2 Image processing for synthetic targets

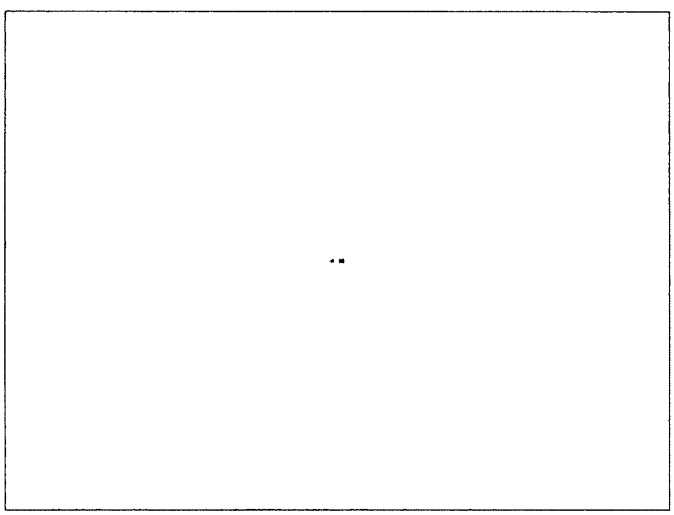


Figure 7.18: Magnified version of the rendered target.

Fig. 5.2 illustrated a sample output of the rendered scene with an embedded spherical target. Fig. 7.18 (re-presented from Fig. 5.3) depicts a close-up of the target. The synthetic target was placed amid an in-flight acquired background image, at or near the horizon, where most targets of interest are expected to arise. Noise was evidently present and explicitly not removed to obtain a realistic result.

The target was a 1.0 m radius sphere placed at a distance of 10.0 km, for an effective angular extent of 2.0 pixels for the modeled camera. Both specular and diffuse reflections were modeled, with the target surface assumed to be marginally specular.

Figure 7.19: Processed synthetic image (from Fig. 5.2. Brightness was enhanced for readability.

The benchmark processing pipeline utilizing the LoG filter produced Fig. 7.19. Note that the image was not pre-weighted to judge the performance against outliers. The target of interest is highlighted in red and a false positive is highlighted in green. The false positive is a strong specular reflection from the roof of a truck, showing the importance of localization.

Fig. 7.20 illustrates a magnitude plot of the filter response for a region of interest (ROI) band localized at the horizon. The target response is strong indicating that a 2.0 m cross-section target should be readily detected at 10.0 km. However, it is noted that the response is in the absence of noise and point-spread effects. In particular, platform and atmospheric effects are not modeled at all. The introduction of atmospheric, platform and optical transfer functions are expected to severely degrade the detection range of the target as well as increase the number of false positives.

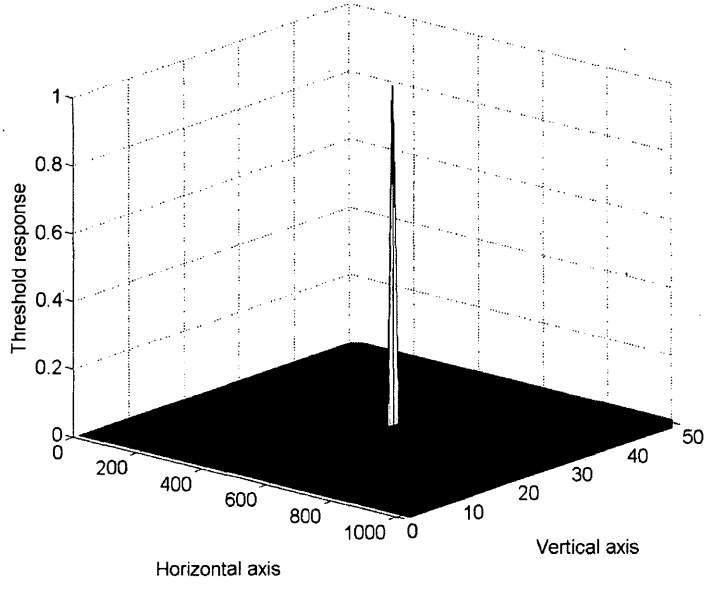


Figure 7.20: Signal magnitude of processed target.

7.8.3 Target detection in head-on Cerberus imagery

Fig. 7.21 shows an image acquired by NRC Canada's Cerberus instrument [MTE+12]. This instrument was a camera array comprising of three cameras, each of which was equipped with a 12 mm lens. The image sensor was a 5.0 MP CCD sensor with a 3×2 aspect ratio and a pixel pitch of 3.5μm. The geometric angular field-of-view per pixel was calculated to be 0.28 mrad, with the total Cerberus field-of-view spanning 120 degrees by 33 degrees. The cameras were rigidly mounted to a plate, which itself was rigidly mounted to the gun-mount attachment on the Bell 205.

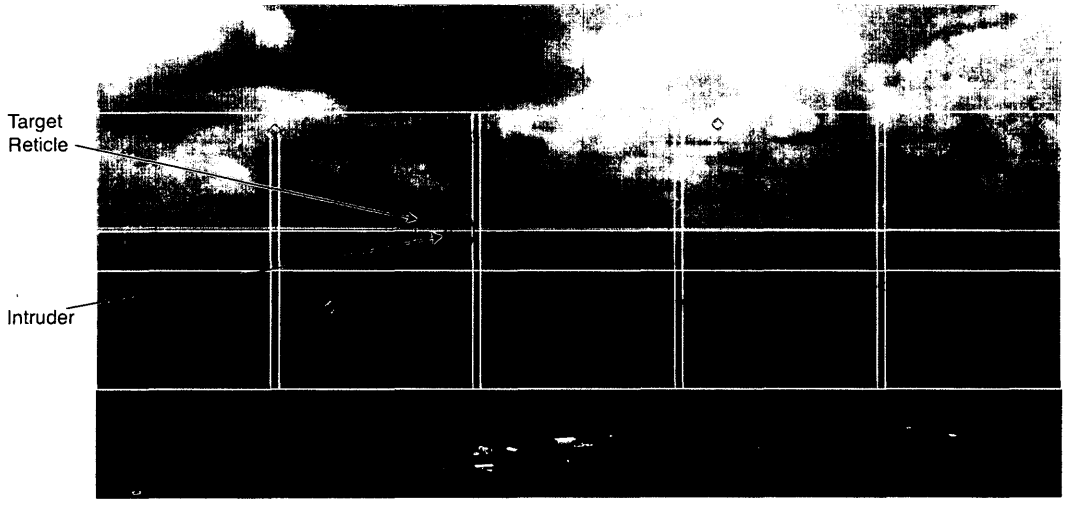


Figure 7.21: A processed frame from the flight trials of NRC's Cerberus array [MTE+12]. Image brightness has been adjusted for readability.

Mapping the field-of-view of Cerberus to DragonflEYE is useful in providing a sense of scale. In Fig. 7.21, the image is the output of just the forward-looking cam-

era from Cerberus. White boxes were drawn to show the approximately matching fields-of-view of individual DragonflEYE smart camera nodes. Note that horizontal and vertical overlap regions are present between adjacent node sub-images, and correspond approximately to the overlap configured in the flight prototype of DragonflEYE. All processing was restricted to the fields-of-view denoted by the boxes.

Since the angular resolution of Cerberus was lower than that of DragonflEYE, the LoG filter was applied locally to each sub-image without horizon ROI pre-weighting. Spurious signals distant from the horizon, such as specular reflections from rooftops and other artifacts, have strong responses but are clutter. Specular reflections, in general, are expected to challenge the performance of the LoG filter. A white line marks the horizon, while a blue annulus indicates the target position, both of which are computed from the ADS-B systems on board the host and target aircraft, and transformed through the camera viewframe. Red diamonds indicate potential targets from an image processing algorithm, with the intensity of red hue indicative of rank. Only the top five hits, ranked by SNR, were displayed per camera view. The target is visible within the annulus and has scored multiple hits.

For the Cerberus instrument, the top five hits were chosen (N = 5) and p_{proc} was set at 90%. Selecting these parameters provided a processing detection range of $R_{0,proc} \simeq 3.9$km. If compared to the angular resolution of the DragonflEYE flight prototype, the Cerebus has an effective down-sample factor of 2.5. Recalling the

image resolution plot in Fig. 6.13, it is reasonable to assume that the expected range at first detection (R_0 value) for Cerberus should lie within the 3.0 - 4.0 km range. The empirically determined $R_{0,proc}$ lies within the expected bounds, suggestive of good performance from the algorithm. However, since the actual relationship between resolution and range is non-linear, it is difficult to precisely measure the expected R_0 value without additional analysis. Furthermore, the comparison does not take into account atmosphere effects, range-dependent PSF and target radiance fluctuations. Accurate modeling is both difficult to achieve and difficult to verify.

7.8.4 Target detection in head-on DragonflEYE imagery

Presented in this section are results from the application of the target detection pipeline on the DragonflEYE image dataset. Since the DragonflEYE instrument is an amalgam of 10 free-running cameras, image synchronization proved to be a challenge during analysis. Furthermore, cameras exhibited non-trivial random, inter-camera motion during flight. These two factors in combination, made it difficult to calculate the rectification between cameras from aerial imagery alone.

Fig. 7.22 depicts a composite panoramic stitch of all ten cameras for one instant in time. Frame labels reflect camera node numbers within the DragonflEYE array. To obtain the closest image in time, one camera was designated as the base and its image time-stamp utilized as the reference time-stamp. For all other cameras, the

image closest in time to that stamp was chosen. Registration is typically relative to one base camera, and the camera node chosen for synchronization was also chosen as the reference camera for relative image registration. For all images presented here, Camera 3, as labelled in Fig. 7.22 served as the reference camera for both synchronization and registration.

Figure 7.22: DragonflEYE panoramic stitch of a time instant within a head-on collision run. Target was at a range of 3.5 km.

Registration was performed by computing a non-reflective similarity transform apriori per camera, relative to the base image. Note that this transform was used to approximate the generalized homography specified in Section 3.9. This transformation was recorded and applied to subsequent frames. Care was taken to select a time instant where platform motion was minimized. The composite horizon projection, denoted by a blue dotted line, was constructed as follows: First, the horizon position was estimated in the reference camera, either by the projection methodology

described in Sec. 3.4.3, or by manual selection. Then, the projected horizon was transformed into local image coordinates per camera via the local pre-computed similarity transform.

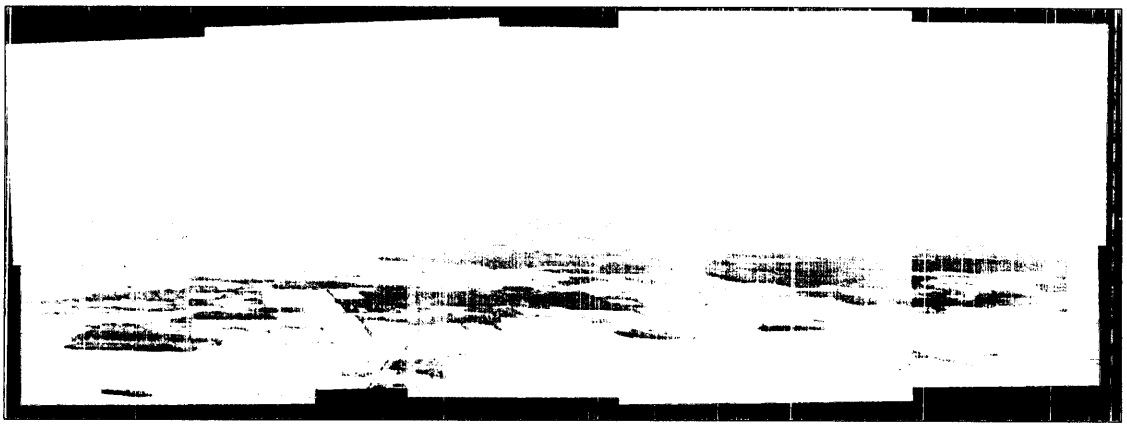


Figure 7.23: Processed panoramic stitch of a head-on target at a range of 3.5 km. The target is highlighted by a green circular reticle with multiple diamond 'hits'.

Fig. 7.23 illustrates a processed version of Fig. 7.22, where the pipeline outlined in Fig. 7.17 was utilized. The target, indicated by the blue circular reticle, is captured at a range of 3.6 km. An intermediate range was chosen for illustration as the target was small compared to the total coverage. Due to random sensor mount jitter, the ADS-B-based reticle experienced significant wander and was not utilized. To compensate, targets were detected manually across frames in selected cameras. Also noticeable is the short distance between the target and the projected horizon, lending support to the horizon proximity hypothesis.

The pre-processing algorithm was implemented with the following settings: ROI selection was performed by a Gaussian window with a half-width of 150 pixels, which was offset from the horizon by 50 pixels to bias the sky portion of the image. Similarly, detections were restricted to the top five hits per camera and a p_{proc} of 90% was utilized. In Fig. 7.23, the target lies within an overlap region and multiple hits are denoted on the target. Note that a tracking operation will further prune spurious detections (false alarms) over a temporal sequence. However, discussion and implementation of a tracking algorithm is beyond the scope of this work.

For the head-on cases tested under the constrained scenario conditions, and given the values of N and p_{proc}, the processed detection range was observed at $R_{0,proc} = 5.8$ km. Note that R_0, for the same sequence (from Chapter 6), was measured at $R_0 = 6.7$ km. More sophisticated algorithms are expected to increase the image processing advantage, pushing sensor performance to the modeling limit.

7.9 Conclusion

In summary, a hierarchical image processing paradigm was proposed, segmenting processing activities into pre- and post- stages. Pre-processing activities were expected to occur locally on individual smart cameras. A representative technique was suggested for each stage with a rationale presented for the choice. The techniques were tested against targets in synthetic and real environments. The proposed

pipeline performed well in the simulation environment with a detection range approaching 10 km. In real scenarios, the range performance was reduced, with typical detection ranges within the limit of the measured R_0. Future processing efforts are expected to improve upon the processing advantage.

8 Conclusion

8.1 Summary

In summary, this dissertation explored the design, development and test of a passive, wide field, optical airborne sense-and-avoid instrument titled 'DragonflEYE'. The instrument was constructed from commercial, off-the-shelf components, and configured as a network of smart camera nodes. An end-to-end imaging train model was developed and important figures of merit were derived. Transfer functions arising from intermediate mediums were discussed and their impact assessed. Development of the instrument proceeded as an evolution of prototypes. Expected performance of the instrument was iteratively evaluated, starting with modeling activities, followed by laboratory tests, ground tests and culminating in flight tests. This prototype was mounted on a Bell 205 helicopter for flight tests, with a Bell 206 helicopter acting as the target. Raw imagery was recorded alongside ancillary aircraft data, and stored for the offline assessment of performance. The image processing alternatives were evaluated offline. The instrument obtained a detection

range exceeding 6.0 km under the proposed conditions, suggesting that the purely passive option is a viable one for certain scenarios.

Multiple figures of merit were developed to quantify the performance of the instrument. Of primary importance is the "range at first detection" (R_0), based on a suitably defined signal-to-noise ratio (SNR). A high degree of fluctuation in the SNR is observed and attributed primarily to changes in the received signal due to variations in target orientation. The presented methodology provides excellent robustness against these fluctuations, ground clutter and other noise elements. The absolute value and precision of R_0 are compared to flight crew acquisition ranges under ideal conditions for multiple trajectories, with typical sensor performance exceeding human performance under identical circumstances. Secondary figures of merit, such as the target size and growth, and proximity to the horizon were also measured, and their impact on the system point-spread function assessed.

A pre-processing pipeline was defined for the smart camera nodes, and implemented with representative techniques for each stage. The performance of this pipeline was measured against both synthetic targets and in-flight captured imagery. An empirical figure of merit was identified for the quantification of the role of image processing. Although typical performance against in-flight imagery was lower than modeling predictions, it well-exceeded the minimum suggested guidelines of the FAA for the scenarios under consideration.

Some challenges were noted along the way. The use of off-the-shelf web-cams limited the calibration and configuration of the image sensors, as access to the camera registers was curtailed. For example, gamma correction was present and strong, likely due to poor sensitivity of the native sensor. The sensor itself was a rolling shutter CMOS, which, during flight, gave rise to rolling shutter wobble artifacts, which are correctable, but at the expense of frame-rate [BBKS10]. The sensor mount exhibited undesired cantilever effects, and inter-camera jitter was noticeable. Registration in the presence of random jitter proved difficult but doable. Physical calibration of the camera mount with respect to the IMU proved unstable for projective measurements, but was compensated for by manual extraction. The use of global shutter (Micron TrueSNAP) CMOS imagers, and a single-body instrument would mitigate these effects. A locally mounted accelerometer chip within the sensor body could help estimate and remove random jitter.

Flight tests were difficult to execute, as atmospheric conditions did not always co-operate. Tests were conducted in winter, which challenged the assumptions behind solar radiance and contributions from the ground. Specular highlights from the snow, ground structures and vehicles were noticeable.

The software architecture of the test and camera nodes was not optimal for raw image capture. The sensor was geared towards optimizing synchronization, not throughput. As such, the test and camera nodes communicated over a single

TCP/IP socket. This choice, while guaranteeing against packet loss, significantly limited the total achievable frame-rate per camera node. The ideal configuration would consist of a TCP link for commands and a UDP link for image data. The strict schedule of flight tests limited the time available to make these improvements.

8.2 Future Work

Despite the copious results presented in the thesis, the analysis has only scratched the surface relative to the amount of data collected. At this stage, many improvements are suggested.

On the instrument hardware end, miniaturization and integration in merging the sensor and the processing onto one die, should allow for more cameras to be mounted within the same cost, mass and power constraints.

Analysis of the sensor under atypical configurations, such as descending head-on and overtake trajectories would be valuable. The data was acquired during the present flight trials, but not analyzed. The target is juxtaposed against ground clutter, making it difficult to spot beyond 2.0 km, even with a trained eye. Detection algorithms based on background subtraction may help in this regard. Further characterization of the instrument is desired in the areas of target shape, growth, and measurement of ancillary artifacts, such as rolling shutter wobble in certain CMOS imagers would be valuable. Comparison between rolling shutter CMOS and

global shutter CCD in images acquired under identical conditions would be useful. The data for such an experiment was already acquired during the current tests.

Longer-term developments may revolve around a decentralized processing approach, such as the 'virtual' camera paradigm. Adaptive processing, where changes in platform speed result in a scaling of the field-of-view in both resolution and coverage, such as rear cameras being turned off, and a rededication of resources to improve the detection range of front cameras. Consensus-based processing approaches, where the detection of an event in one camera FOV, attracts the interest of other cameras, would be especially interesting.

A collision sensing instrument may act as a pilot replacement in a UAS, or an assist in manned aircraft. Although primarily presented in the former role throughout the dissertation, the latter role is equally valid, and may in fact, precede the former for commercial reasons. If acting in an assisting roll, human factors in terms of user interface, frequency of reminders, choice of display, colours, reticles, and attracting pilot attention become important. The introduction of the instrument into the cockpit should reduce the pilot's overall cognitive load, not increase it. Care must be taken with the number and frequency of hits reported and false alarm rates among others, highlighting an entirely new avenue of research.

Bibliography

[ABT03] G. Agranov, V. Berezin, and R. H. Tsai, "Crosstalk and microlens study in a color cmos image sensor," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 4–11, 2003.

[AE01] T. S. Abbott and D. M. Elliott, "Simulator Evaluation of Airborne Information for Lateral Spacing (AILS) Concept," NASA, Langley Research Centre, Hampton, Virginia, Tech. Rep., 2001.

[And91] J. Andrews, "Unalerted Air-to-Air Visual Acquisition - ATC 152," Lincoln Laboratories, MIT, Lexington, MA, Tech. Rep., November 1991.

[AR09] B. M. Albaker and N. Rahim, "A survey of collision avoidance approaches for unmanned aerial vehicles," in *International Conference for Technical Postgraduates*, 2009, pp. 1–7.

[ARM13] *The NEON Engine*, ARM Ltd., 2013. [Online]. Available: http://www.arm.com/products/processors/technologies/neon.php

[Bat99] C. D. Bateman, "The introduction of enhanced ground-proximity warning systems (EGPWS) into civil aviation operations around the world," in *Proc. 11th Annual European Aviation Safety Seminar*, 1999, pp. 259–273.

[BBKS10] S. Baker, E. Bennett, S. B. Kang, and R. Szeliski, "Removing rolling shutter wobble," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2010, pp. 2392 –2399.

[BBP05] R. Bernier, M. Bissonnette, and P. Poitevin, "DSA Radar - Development report," in *AUVSI Conference on Unmanned Vehicle Systems*, 2005.

[BBV08] F. E. Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of Gaussian models: Application to background modeling," in *Proc. International Symposium on Visual Computing*, 2008, pp. 772–781.

[BBV10] T. Bouwmans, F. E. Baf, and B. Vachon, "Statistical background modeling for foreground detection," in *Handbook of Pattern Recognition and Computer Vision*, 4th ed., C. H. Chen, Ed. World Scientific Publishing Co., 2010, pp. 181–197.

[Bea09] *Beagleboard System Reference Manual*, Beagleboard.org, Dec. 2009, Accessed: 2011-Jan-5. [Online]. Available: http://beagleboard.org/static/BBSRM_latest.pdf

[BGS+12] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. Kittle, D.R.Golish, E. M. Vera, and S. Feller, "Multiscale gigapixel photography," *Nature*, vol. 486, pp. 386–389, 2012.

[Bha86] B. Bhanu, "Automatic target recognition: State of the art survey," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-22, no. 4, pp. 364 –379, 1986.

[BKH+99] P. Bateman, L. Kendall, P. Hermann, S. Wright, J. Kistler, G. Ostrom, and N. Paterson, "Product specification for the MK VI and MK VIII enhanced ground proximity warning system," Honeywell Inc., Tech. Rep., 1999.

[BLM97] D. J. Brudnicki, K. Lindsay, and A. McFarland, "Assessment of field trials, algorithmic performance, and benefits of the User Request Evaluation Tool (URET) conflict probe," in *IEEE Conference on Digital Avionics Systems Conference*, vol. 2, 1997, pp. 9.3 –35–9.3–44.

[Boe11] *Type Certificate Data Sheet No. A20WE - Rev. 45*, Boeing Corporation, Aug. 2011. [Online]. Available: http://www.boeing.com/commercial/airports/misc/A20WE-Rev-45.pdf

[Bro71] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering and Remote Sensing*, vol. 37, no. 8, pp. 855–866, 1971.

[BW99] M. Born and E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 7th ed. Cambridge University Press, 1999.

[BXZ05] G.-Q. Bao, S.-S. Xiong, and Z.-Y. Zhou, "Vision-based horizon extraction for micro air vehicle flight control," *IEEE Trans. Instrumentation and Measurement*, vol. 54, no. 3, pp. 1067–1072, 2005.

[CAB+08] P. Chen, P. Ahammad, C. Boyer, S.-I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L.-C. Chang, J. Tygar, and S. Sastry, "CITRIC: A low-bandwidth wireless camera network platform," in *Proc. IEEE/ACM International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[CDK+02] W.-Z. Chen, J. M. DeLuca, J. D. Koeller, W. F. O'Neil, and I. H. Wong, "Autonomous Unmanned Aerial Vehicle (UAV) Airspace Operations Sensing Requirements. Volume 1 - Performance," Northrop Grumman Corporation - Air Combat Systems, El Segundo, CA, Tech. Rep., July 2002.

[CJH05] Z. Cao, Z. Ji, and M. Hu, "An image sensor node for wireless sensor networks," in *Proc. IEEE International Conference on Information Technology: Coding and Computing*, vol. 2, 2005, pp. 740–745.

[DGSD09] D. Dey, C. Geyer, S. Singh, and M. Digioia, "Passive, long-range detection of aircraft: Towards a field deployable Sense and Avoid System," in *Proc. Conference on Field and Service Robotics*, 2009, pp. 1–10.

[DHS00] R. O. Duda, P. E. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2000.

[EG05] K. Ellis and A. W. Gubbels, "The National Research Council of Canada's Surrogate UAV Facility," in *Proc. UVS Canada Conference*, 2005.

[EHD00] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. European Conference on Computer Vision*, 2000, pp. 751–767.

[ENIW02] S. Ettinger, M. Nechyba, P. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," in *Proc. IEEE Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2134–2140.

[ER95] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*, 3rd ed. Wiley, 1995.

[ER04] D. Ebdon and J. Regan, "Sense-and-avoid requirement for Remotely Operated Aircraft (ROA)," Langley Air Force Base, USAF, Tech. Rep., June 2004.

[F2404] A. S. F2411-07, "Standard specification for the design and performance of an airborne sense-and-avoid system," ASTM International, Tech. Rep., 2004.

[FAA04] FAA, *Pilot and Air Traffic Controller Guide to Wake Turbulence*, 2004. [Online]. Available: http://www.faa.gov/training_testing/training/media/wake/04SEC2.PDF

[Fed83] Federal Aviation Administration (FAA), "Advisory Circular 90-48C: Pilots' role in collision avoidance," US DOT, FAA, Tech. Rep., 1983.

[Fli] Flight Research Laboratory - NRC Canada and the Vision Sensor Laboratory, "Private Communication."

[FMAR08] G. Fasano, A. Moccia, D. Accardo, and A. Rispoli, "Development and test of an integrated sensor system for autonomous collision avoidance," in *Proc. International Congress of the Aeronautical Sciences*, 2008.

[FS97] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[FSHG06] S. Fefilatyev, V. Smarodzinava, L. O. Hall, and D. B. Goldgof, "Horizon detection using machine learning techniques," in *Proc. Conference on Machine Learning and Applications*, Dec 2006, pp. 17–21.

[Gri05] D. E. Grilley, "Resolution requirements for passive sense-and-avoid," Alion Science and Technology, Tech. Rep., 2005.

[GS89] D. M. Green and J. A. Swets, *Signal Detection Theory and Psychophysics*, 1st ed. Peninsula Publications, 1989.

[GW02] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, January 2002.

[GW08] ——, *Digital Image Processing*, 3rd ed. Prentice Hall, January 2008.

[Hec02] E. Hecht, *Optics*, 4th ed. Addison-Wesley, 2002.

[HK07] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1309–1321, Aug. 2007.

[Hol03] G. C. Holst, *Electro-optical Imaging System Performance*, 3rd ed. SPIE Press, 2003.

[HS99] R. Hemm and G. Shapiro, "Airborne Information for Lateral Spacing (AILS) Benefit Estimate," NASA, McLean, Virginia, Tech. Rep., 1999.

[HZ03] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 1st ed. Cambridge University Press, 2003.

[IE12] M. Imray and K. Ellis, "Ground vibration test - bell 205 dragonfly eye installation," NRC Aerospace - Flight Research Laboratory, Tech. Rep., 2012.

[Int12] Intel Corporation, "Intel 64 and ia-32 architectures optimization reference manual," Intel Corporation, Tech. Rep., 2012. [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html

[Jan01] J. R. Janesick, *Scientific Charge-coupled Devices*, 1st ed. SPIE Press, 2001.

[Jet07] *User manual of VIA CN700 Chip*, Jetway Information Company, Aug. 2007, Accessed: 2011-Jan-5. [Online]. Available: http://www.jetwaytw.com/manual/J7F2.rar

[JEV11] S. Jayaraman, S. Esakkirajan, and T. Veerakumar, *Digital Image Processing*, 1st ed. Tata McGraw-Hill Education, 2011.

[KASD07] R. Kleihorst, A. Abbo, B. Schueler, and A. Danilin, "Camera mote with a high-performance parallel processor for real-time frame-based video processing," in *Proc. IEEE/ACM International Conference on Distributed Smart Cameras*, 2007, pp. 109–116.

[KB70] N. S. Kopeika and J. Bordogna, "Background noise in optical communications," in *Proc. IEEE Conference*, vol. 58, 1970, pp. 1571–1577.

[KCY+10] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong, "Intelligent visual surveillance - a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 926–939, 2010.

[KDS98] N. S. Kopeika, I. Dror, and D. Sadot, "Causes of atmospheric blur: comment on atmospheric scattering effect on spatial resolution of imaging systems," *J. Optical Society of America*, vol. 15, no. 12, pp. 3097–3106, 1998.

[KEC+11] J. Keillor, K. Ellis, G. Craig, D. Rozovski, and R. Erdos, "Studying collision avoidance by nearly colliding: A flight test evaluation," in *Proc. Human Factors and Ergonomics Annual Meeting*, 2011, pp. 41–45.

[KI86] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[KLOS06] B. C. Karhoff, J. I. Limb, S. W. Oravsky, and A. D. Shephard, "Eyes in the Domestic Sky: An Assessment of Sense and Avoid Technology for the Army's"Warrior" Unmanned Aerial Vehicle," in *Proc. IEEE Conference on Systems and Information Engineering Design Symposium.* IEEE, April 2006.

[KM06] K. Krajsek and R. Mester, "The edge preserving wiener filter for scalar and tensor valued images," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, K. Franke, K.-R. Muller, B. Nickolay, and R. Schafer, Eds. Springer Berlin Heidelberg, 2006, vol. 4174, pp. 91–100.

[Knu97] D. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, 3rd ed. Addison-Wesley, 1997, vol. 2.

[Kop98] N. S. Kopeika, *A System Engineering Approach to Imaging*, 1st ed. SPIE Press, 1998.

[KRR09] A. Kandhalu, A. Rowe, and R. Rajkumar, "DSPCam: A camera sensor system for surveillance networks," in *Proc. IEEE/ACM International Conference on Distributed Smart Cameras*, 2009, pp. 1–7.

[KY00] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, pp. 179–189, 2000.

[LDR03] B. W. Leach, J. Dillon, and R. Rahbari, "Operational experience with optimal integration of low-cost inertial sensors and gps for flight test requirements," *Canadian Aeronautics and Space Journal*, vol. 49, no. 2, pp. 41–54, 2003.

[LE90] T. Lindeberg and J.-O. Eklundh, "Scale detection and region extraction from a scale-space primal sketch," in *Proc. Third International Conference on Computer Vision*, Dec 1990, pp. 416–426.

[LH02] B. Lee and M. Hedley, "Background estimation for video surveillance," *Image and Vision Computing, New Zealand*, pp. 315–320, 2002.

[Mah36] P. Mahalanobis, "On the generalised distance in statistics," in *Proc. National Institute of Science, India*, vol. 2, no. 1, 1936, pp. 49–52.

[Mah94] V. N. Mahajan, "Zernike circle polynomials and optical aberrations of systems with circular pupils," *Applied Optics*, vol. 33, no. 34, pp. 8121–8121, 1994.

[McG10] J. McGuire, "Manufacturable mobile phone optics: Higher order aspheres are not always better," in *Proc. OSA International Conference on Optical Design*, 2010.

[MMSZ05] S. Messelodi, C. Modena, N. Segata, and M. Zanin, "A Kalman filter based background updating algorithm robust to sharp illumination changes," in *Proc. International Conference on Image Analysis and Processing*, 2005, pp. 163–170.

[MS95] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images," in *Proc. British Conference on Machine Vision and Applications*, 1995, pp. 187–193.

[MST08] J. McCarten, J. Summa, and C. Tivarus, "Back-illuminated sensor with low cross-talk," U.S. Patent 7 923 830 B2, Apr., 2008.

[MTE+12] C. Minwalla, P. Thomas, K. Ellis, R. Hornsey, and S. Jennings, "Flight test evaluation of a prototype optical instrument for airborne sense-and-avoid applications," in *Proc. SPIE Conference on Defense, Security and Sensing - Unmanned Systems Technology*, Apr. 2012, pp. 83 870R–1–14.

[MTW+10] C. Minwalla, M. Tekeste, K. Watters, P. Thomas, R. Hornsey, K. Ellis, and S. Jennings, "Modeling a prototype optical collision avoidance sensor for unmanned aerial vehicles," in *Proc. IEEE International Conference on Sensors*, Nov 2010, pp. 1657–1662.

[MTW+11] C. Minwalla, P. Thomas, K. Watters, M. Liscombe, N. Maharaj, W. Gao, E. Shen, and R. Hornsey, "Visor milestone #4 report," VISOR Laboratory, York University, Tech. Rep., Jan. 2011.

[MUD+07] J. McCalmont, J. Utt, M. Deschenes, M. Taylor, R. Sanderson, J. Montgomery, R. S. Johnson, and D. McDermott, "Sense and avoid technology for unmanned aircraft systems," in *Proc. SPIE Conference on Automatic Target Recognition XVII*, vol. 6566, no. 1, 2007.

[MUDT05] J. F. McCalmont, J. Utt, M. Deschenes, and M. J. Taylor, "Sense and avoid technology for global hawk and predator UAVs," in *Proc. SPIE Conference on Infrared Technology and Applications XXXI*, 2005.

[Nar10] *The Angstrom Distribution*, Narcissus, Open Embedded, 2010. [Online]. Available: http://www.angstrom-distribution.org/

[NBPL08] E. Norouznezhad, A. Bigdeli, A. Postula, and B. Lovell, "A high resolution smart camera with Gig-E vision extension for surveillance applications," in *Proc. ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–8.

[NDF00] R. Nixon, N. Doudoumopoulos, and E. R. Fossum, "Backside illumination of CMOS image sensor," U.S. Patent 6 429 036 B1, Jan, 2000.

[Nor01] Norbert Tränapp, IABG, "Preliminary study on integration of unmanned aerial vehicles into future air traffic management," CARE Innovative Action, Tech. Rep., 2001.

[NP12] S. H. H. Nourzad and A. Pradhan, "Binary and multi-class classification of fused lidar-imagery data using an ensemble method," in *In Proc: Construction Research Congress.* ASCE, 2012.

[Ots79] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62 –66, Jan 1979.

[Ove76] I. Overington, *Vision and Acquisition: Fundamentals of human visual performance, environmental influences and applications in instrumentational optics*, 1st ed. Pentech Press Limited, 1976.

[Ove10] *Overo Gumstix - Water*, Overo Corporation, 2010. [Online]. Available: https://www.gumstix.com/store/product_info.php?products_id=228

[PCN+05] B. Pain, T. Cunningham, S. Nikzad, M. Hoenk, T. Jones, B. Hancock, and C. Wrigley, "A back illuminated megapixel cmos image sensor," in *Proc. IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, 2005.

[Pen12] *Pentax CCTV Catalog*, Pentax - A Ricoh Corporation, 2012. [Online]. Available: http://www.pentaxcctvus.com/catalog/index.html

[PZ07] S. M. Pope and R. C. Zeta, "Package-on-package secure module having anti-tamper mesh in the substrate of the upper package," U.S. Patent 7 923 830 B2, Apr. 13, 2007.

[Rat10] T. D. Raty, "Survey on contemporary remote surveillance systems for public safety," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. PP, no. 99, pp. 1–23, Mar. 2010.

[Rho99] H. E. Rhodes, "Microlens array with improved fill factor," U.S. Patent 6 307 243 B1, Jul., 1999.

[RS91] Research and D. Service, "Precision runway monitor program," USDOT, FAA, Washington, D.C., Tech. Rep., 1991, report ID: DOT/FAA/RD-91/5.

[RTC97] RTCA Inc., "Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II) Airborne Equipment," Federal Aviation Administration (FAA), Tech. Rep., 1997.

[RW08] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, Oct 2008.

[RWS+08] B. Rinner, T. Winkler, W. Schriebl, M. Quaritsch, and W. Wolf, "The evolution from single to pervasive smart cameras," in *Proc. ACM/IEEE International Conference on Distributed Smart Cameras*, 2008, pp. 1–10.

[SBB+05] V. Suntharalingam, R. Berger, J. Burns, C. Chen, C. Keast, J. Knecht, R. Lambert, K. Newcomb, D. O'Mara, D. Rathman, D. Shaver, A. Soares, C. Stevenson, B. Tyrrell, K. Warner, B. Wheeler, D.-R. Yost, and D. Young, "Megapixel cmos image sensor fabricated in three-dimensional integrated circuit technology," in *Proc. IEEE International Solid-State Circuits Conference*, vol. 1, Feb. 2005, pp. 356–357.

[SG99] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, no. xxiii, 1999, pp. 637–663).

[She12] E. Shen, "Optimization of resource allocation using self-organization in an embedded distributed smart camera network," Ph.D. dissertation, York University, 2012.

[SHK+02] A. Smirnov, B. N. Holben, Y. J. Kaufman, O. Dubovik, T. F. Eck, I. Slutsker, C. Pietras, and R. N. Halthore, "Optical properties of atmospheric aerosol in maritime environments," *AMS Journal of the Atmospheric Sciences*, vol. 59, pp. 501–523, 2002.

[SMP08] M. Sigari, N. Mozayani, and H. Pourreza, "Fuzzy running average and fuzzy background subtraction: Concepts and applications," *Computer Science and Network Security*, vol. 8, no. 2, pp. 138–143, 2008.

[SSS+03] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications within sensor networks," in *Proc. IEEE*, vol. 91, no. 8, 2003, pp. 1235–1246.

[Str94] C. E. M. Strauss, "Synthetic-array heterodyne detection: a single element detector acts as an array," *Optics Letters*, vol. 19, no. 20, pp. 1609–1611, 1994.

[SWP+01] A. J. Smerlas, D. J. Walker, I. Postlethwaite, M. E. Strange, J. Howitt, and A. W. Gubbels, "Evaluating H_∞ controllers on the NRC Bell 205 Fly-by-wire helicopter," *Control Engineering Practice*, vol. 9, no. 1, pp. 1 – 10, 2001.

[TC07] H. C. Tang and C. S. Chen, "Optical system for taking image," U.S. Patent 7 274 528 B1, Sept. 25, 2007.

[Tex11] *OMAP 3530 Applications Processor*, Texas Instruments, 2011, Accessed: 2011-Jan-10. [Online]. Available: http://www.ti.com/lit/ds/symlink/omap3530.pdf

[Tho11] C. Thomas, "Micrite: Towards a sub-100-micron distributed sensor system," Ph.D. dissertation, York University, 2011.

[TKBM99] K. Toyama, J. Krumm, B. Brummitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. International Conference on Computer Vision*, 1999, pp. 255–261.

[TML+10] P. Thomas, C. Minwalla, M. Liscombe, K. Watters, M. Tekeste, N. Maharaj, W. Gao, E. Shen, and R. Hornsey, "Visor milestone #3 report," VISOR Laboratory, York University, Tech. Rep., Jun 2010.

[TML+11] P. Thomas, C. Minwalla, S. Lio, K. Watters, and R. Hornsey, "Visor milestone #6 report," VISOR Laboratory, York University, Tech. Rep., Jan. 2011.

[TS67] K. E. Torrance and E. M. Sparrow, "Theory for off-specular reflection from roughened surfaces," *J. Optical Society of America*, vol. 57, no. 9, pp. 1104–1114, 1967.

[UG04] S. Umeyama and G. Godin, "Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 639 –647, 2004.

[Ves08] A. Vesel, "ADS-B Broadcast Monitoring System and Method," U.S. Patent 7 383 124 B1, Jun. 3, 2008.

[WADP97] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "PFinder: Real-time tracking of the human body," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[WGS07] P. Withagen, F. C. A. Groen, and K. Schutte, "CCD color camera characterization for image measurements," *IEEE Trans. Instrumentation and Measurement*, vol. 56, no. 1, pp. 199–203, 2007.

[Wie64] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* MIT Press, 1964.

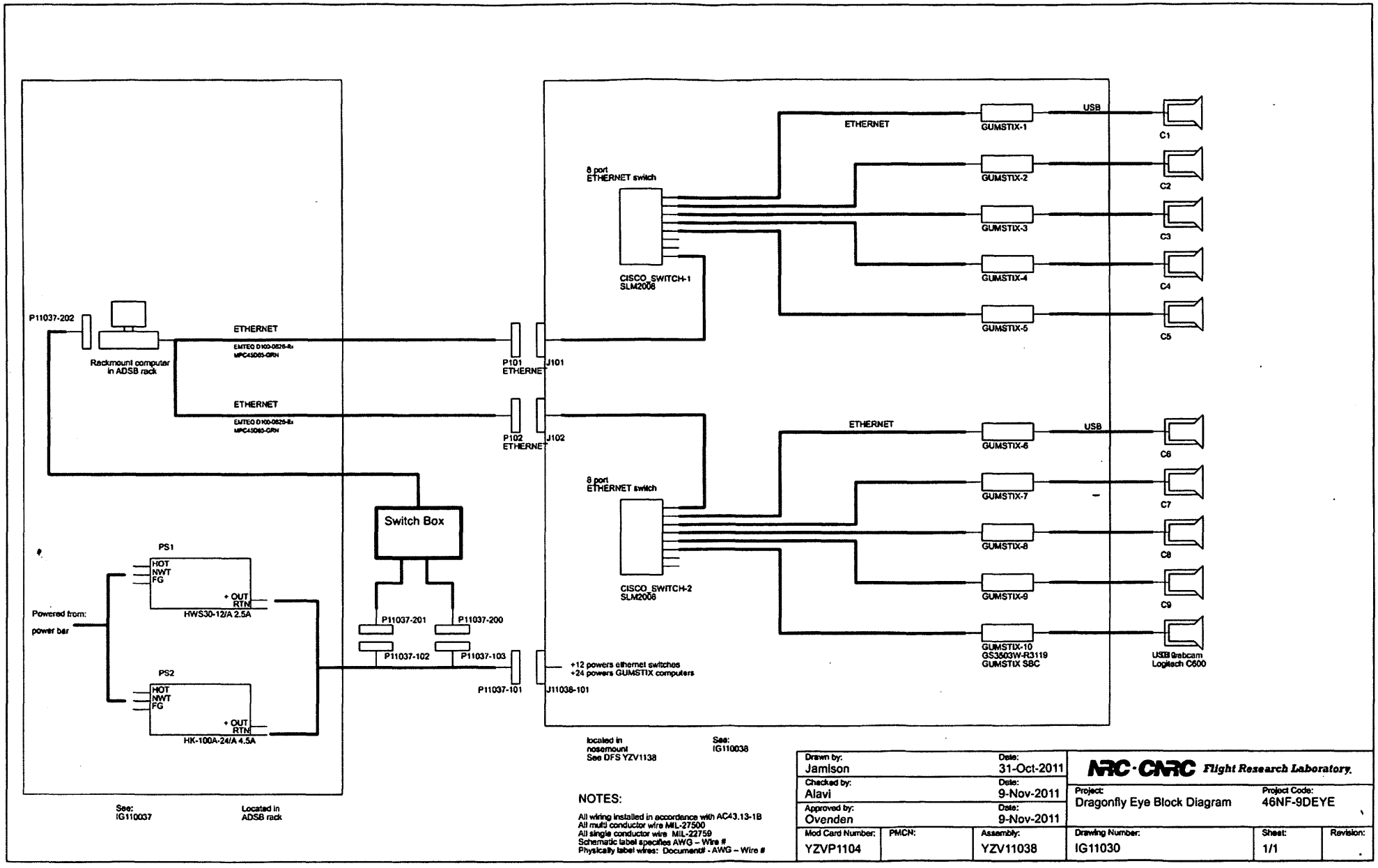
[WML+11] K. Watters, C. Minwalla, M. Liscombe, H. I. Lio, P. Thomas, R. Hornsey, K. Ellis, and S. Jennings, "Characterization of an optical collision avoidance sensor," in *Proc. Canadian Conference on Electrical and Computer Engineering*, May 2011.

[Wol04] R. C. Wolfe, "NASA ERAST Non-cooperative DSA Flight Test," Modern Technology Solutions, Tech. Rep., 2004.

[WZ89] W. L. Wolfe and G. J. Zissis, *The Infrared Handbook*, 3rd ed. United States Government Printing Office, 1989.

[ZOT09] *ZOTAC ION-ITX A Series*, ZOTAC International Limited, Jul. 2009, Accessed: 2011-Jan-5. [Online]. Available: http://www.zotac.com/pdbrochures/mb/ZOTAC-ION-ITX_v3.pdf

Appendix A: Electrical Schematic of the Flight Prototype



NOTES:

All wiring installed in accordance with AC43.13-1B
All multi conductor wire MIL-27500
All single conductor wire MIL-22759
Schematic label specifies AWG – Wire #
Physically label wires: Document# - AWG – Wire #

Drawn by: Jamison		Date: 31-Oct-2011	NRC·CNRC Flight Research Laboratory	
Checked by: Alavi		Date: 9-Nov-2011	Project: Dragonfly Eye Block Diagram	Project Code: 46NF-9DEYE
Approved by: Ovenden		Date: 9-Nov-2011		
Mod Card Number: YZVP1104	PMCN:	Assembly: YZV11038	Drawing Number: IG11030	Sheet: 1/1 — Revision: