NORTHWESTERN UNIVERSITY

Image Super-Resolution Enhancements for Airborne Sensors

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PATIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical Engineering and Computer Science

By

Matthew Woods

EVANSTON, ILLINOIS

December 2016

ABSTRACT

Image Super-Resolution Enhancements for Airborne Sensors

Matthew Woods

This thesis discusses the application of advanced digital signal and image processing techniques, particularly the technique known as super-resolution (SR), to enhance the imagery produced by cameras mounted on an airborne platform such as an unmanned aircraft system (UAS). SR is an image processing technology applicable to any digital, pixilated camera that is physically limited by construction to sample a scene with a discrete, $\boldsymbol{m\,x\,n}$ pixel array. The straightforward objective of SR is to utilize mathematics and signal processing to overcome this physical limitation of the $\boldsymbol{m\,x\,n}$ array and emulate the "capabilities" of a camera with a higher-density, $\boldsymbol{km\,x\,kn}$ ($\boldsymbol{k > 1}$) pixel array. The exact meaning of "capabilities", in the preceding sentence, is application dependent.

SR is a well-studied field starting with the seminal 1984 paper by Huang and Tsai. Since that time, a multitude of papers, books, and software solutions have been written and published on the subject. However, although sharing many common aspects, the application to imaging systems on airborne platforms brings forth a number of unique challenges as well as opportunities that are neither currently addressed nor currently exploited by the state-of-the-art. These include wide field-of-view imagery, optical distortion, oblique viewing geometries, spectral variety from the visible band through the infrared, constant ego-motion, and availability of supplementary

information from inertial measurement sensors. Our primary objective in this thesis is to extend the field of SR by addressing these areas. In our research experiments, we make significant use of both simulated imagery as well as real video collected from a number of flying platforms.

LIST OF ABBREVIATIONS AND ACRONYMS

3D	3-Dimensional
3 d.o.f.	3 degree-of-freedom
6 d.o.f.	6 degree-of-freedom
ADVI	Automatic Derivative Variational Inference
ATR	Automated Target Recognition
BAE	British Aerospace
BTV	Bi-lateral Total Variation
COTS	Commercial Off the Shelf
dB	decibel
DCM	Direction Cosine Matrix
DFT	Discrete Fourier Transform
DIRSIG	Digital Imaging and Remote Sensing Image Generation Model
DNG	Digital Negative
DTED	Digital Terrain Elevation Database
EM	Expectation-Maximization
EROS	Earth Resources Observation and Science Center
ESE	EnSquared Energy
ESF	Edge Spread Function
FOR	Field-of-Regard
FOV	Field-of-View
FPA	Focal Plane Array
FPA	Focal Plane Array
GMRF	Gaussian Markov Random Field
GPS	Global Positioning System
gsd	Ground Sample Distance
HMRF	Huber Markov Random Field
HR	High Resolution
IF	Information Filter
IFEM	Information Filter / Expectation-Maximization
IFOV	Instantaneous Field of View
IPD	Inter Pupil Distance
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infrared
ISO	International Standards Organization
JPG	Joint Photographic Experts Group

KL	Kullback-Leibler
lp	line-pair
LR	Low Resolution
LSF	Line Spread Function
LWIR	Long-Wave Infrared
MAP	Maximum-A-Posteriori
MAV	Micro Air Vehicle
MCMC	Markov Chain Monte-Carlo
MEMS	Micro Electro-Mechanical Systems
MODTRAN	MODerate resolution atmospheric TRANsmission
MTF	Modulation Transfer Function
MTF	Modulation Transfer Function
MWIR	Mid-Wave Infrared
NED	North-East-Down
NEDT	Noise Equivalent Delta Temperature
NFOV	Narrow Field-of-View
NVESD	Night Vision and Electronic Sensors Directorate
NVTHERM	Night Vision Thermal Imaging Systems Performance Model
OpenCV	Open Computer Vision
PCA	Principal Component Analysis
P_m	Probability of Measurement
psf	Point Spread Function
PSNR	Peal Signal to Noise Ratio
RGB	Red-Green-Blue
RISS	Real-Time Infrared/Electro-Optic Scene Simulator
RMS	Root-Mean-Square
ROI	Region of Interest
RPM	Revolutions per Minuit
SAR	Simultaneous Auto-Regressive
SNR	Signal to Noise Ratio
SFM	Structure From Motion
SFR	Spatial Frequency Response
SLAM	Simultaneous Localization and Mapping
SR	Super-Resolution
SRTM	Shuttle Radar Topography Mission
SWaP	Size, Weight, and Power
SWIR	Short-Wave Infrared
TOD	Triangle Orientation Discrimination
TIFF	Tagged Image File Format

TV	Total Variation
UAS	Unmanned Aircraft System
UAV	Unmanned Air Vehicle
μm	micron
USAF	United States Air Force
VBI	Variational Bayesian Inference
WFOV	Wide Field-of-View

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

This thesis discusses the application of advanced digital signal processing techniques, particularly the technique known as super-resolution (SR), to enhance the imagery produced by cameras mounted on an airborne platform such as an unmanned aircraft system (UAS). SR is an image processing technology applicable to any digital, pixilated camera that is physically limited by construction to sample a scene with a discrete, $\boldsymbol{m\ x\ n}$ pixel array. The straightforward objective of SR is to utilize mathematics and signal processing to overcome the physical limitation of the $\boldsymbol{m\ x\ n}$ array and emulate the "capabilities" of a camera with a higher-density, $\boldsymbol{km\ x\ kn}$ ($\boldsymbol{k > 1}$) pixel array. The exact meaning of "capabilities", in the preceding sentence, is application dependent.

SR is a well-studied field starting with the seminal 1984 paper by Huang and Tsai [1]. Since that time, a multitude of papers, books, and software solutions have been written and published on the subject. However, although sharing many common aspects, the application to imaging systems on airborne platforms brings forth a number of unique challenges as well as opportunities that are neither currently addressed nor exploited by the state-of-the-art. Our primary objective in this thesis is to extend the field by addressing these areas.

The first unique challenge of airborne imagery is geometric variety. Depending on the application, airborne cameras vary from having a narrow field-of-view (NFOV) of a few degrees to a wide field-of-view (WFOV) of over 100 degrees. In the latter case, the camera's optics may present a great deal of distortion. Additionally, the orientation, or pose, of the camera with respect to the world can often create a significantly oblique viewing geometry, containing objects and

landscape at a diverse set of ranges and even undergoing transformations in appearance (e.g., a circle appears to be an ellipse when viewed at an oblique angle).

A second consideration for airborne applications is that the imagers frequently operate in different areas of the spectrum including visible (0.3 to 0.7 μm), short wave infrared (SWIR, 0.9 to 1.7 μm), mid wave infrared (MWIR, 3.0 to 5.0 μm), and long wave infrared (LWIR, 7.0 to 13.5 μm). Enhancement of visible band imagery is, as expected, covered the most in existing literature. Infrared imagery brings forth both new challenges as well as opportunities. In contrast to visible band sensors, high manufacturing cost, fabrication complexity, and quantum efficiency makes it impractical to reduce the size of individual pixels on an IR sensor to match the diffraction-limited resolution scale of the optics [2]. Consequently, the fundamental resolution of IR sensors is limited by sampling as opposed to diffraction, that is, IR sensors are Nyquist as opposed to Rayleigh limited [3]. As we will find, this fact can make IR imagery more amenable to SR enhancement than visible imagery which is typically resolution limited by the camera's optics.

The third consideration is relevant for UAS applications. A UAS application (see Figure 1-1 for examples of a variety of UAS applications), as it does not need to carry the weight of human passengers, often has stringent size, weight, and power (SWaP) requirements that drive the use of sensors which are small-size, light-weight, and low-power. These constraints often preclude the use of hardware enhancements, such as large and complex optics, to maximize the image quality for downstream applications. Instead, it is necessary to utilize digital signal processing techniques, such as SR, to compensate for the simplified hardware.

Figure 1-1: Example of a variety of UAS applications ranging from the very large to the very small

1.1 Resolution Critical Applications

The purpose of using SR enhancement is that, for many applications of airborne imagery, resolution is a critical flow-down parameter necessary for the application to meet its specification performance requirements. The most historically relevant and common application class of captured imagery is for human consumption and enjoyment. In this case, resolution corresponds to our perception of the quality and, specifically, "sharpness" of an image. As a consequence, the majority of existing SR literature focuses on human perception as the basis both for objective optimization in algorithm development as well as for metrics of success. In some cases, optimization over some metric of human perception can even be done at the expense of scene recovery fidelity [4].

In this thesis, we alternately focus on the photogrammetric application class for captured imagery. This includes human, autonomous, or hybrid processing and interpretation of imagery. Photogrammetry has been defined by the American Society for Photogrammetry and Remote Sensing (ASPRS) as "the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring and interpreting photographic images and patterns of recorded radiant electromagnetic energy and other phenomena." [5] Many airborne imagery applications are photogrammetric in nature and are continuously moving towards fully automated, vs. human-in-the-loop, processing.

During the course of our research, we identified that standard metrics for SR performance were human perceptive centric and not photogrammetric or task centric. Indeed, existing metrics for SR did not directly evaluate the ability of super-resolution algorithms to super-resolve; that is, to increase the effective resolution of captured imagery. As part of our work, we defined a new performance metric, introduced in Chapter 4, to fill this omission in the literature. Here, we provide

a brief list of resolution critical applications, which utilize both the visual and infrared domains, for which both SR and this new metric are particularly relevant.

- **Small target detection and tracking in a cluttered environment** [6,7]
- **Remote sensing, detection, tracking, and classification of objects of interest in the environment.** This includes human faces, text, vehicles, aircraft, etc. [7-19]
- **Automatic Target Recognition (ATR)**. This is the ability for an algorithm or device to recognize targets or objects based on data obtained from sensors [6,20,21].
- **Simultaneous Localization and Mapping (SLAM) for robotics and Micro Air Vehicles (MAVs)** operating in an unstructured environment [22-27]
- **Thermography for product, building, and agriculture quality and fault inspection** [28,29]
- **Structure from motion (SFM), landscape mapping** [30-33]
- **3D digital model building for computer graphics** [34]
- **Anomalous or abnormal behavior detection** based on surveillance imagery and video [35]
- **Geo-registered landscape imagery** formed by mosaicking video frames [36-38]
- **Visual Feedback and Servo Control** [39,40]

1.2 Outline and Organization of the Thesis

In Chapter 2, we provide an overview of technical background material critical to the remainder of the thesis as well as the current state-of-the-art of super-resolution solutions. In Chapter 3, we develop and describe our first contribution to the airborne imaging domain. In this chapter, we capitalize on the fact that an airborne platform, particularly an autonomous one, will have an onboard inertial measurement unit (IMU) or inertial navigation system (INS) to supplement its flight control and navigation sub-systems. We show that this additional information may be used to either simplify or to replace existing algorithms that solve the general *image correspondence* problem, which has use for both SR as well as other, standard image processing task. In Chapter 4, we explain and demonstrate our new spatial frequency metric for evaluating SR performance. As described in 1.1, this new metric evaluates SR performance explicitly based on its ability to perform its fundamental task of increasing the image resolution. In Chapter 5, we

extend the exiting capabilities of state-of-the-art SR algorithms to include unique factors frequently found in airborne camera applications. These include wide field-of-view, lens distortion, oblique viewing geometries, natural ego-motion, and the presence of a supplementary information from an IMU or INS. In Chapter 6, we examine the utility of SR to improve the statistical performance of remote sensing and object classification. For this evaluation, we focus on the problem of remote text classification as a surrogate, with the expectation that the observations and results extend to other general remote classification problems such as vehicle or face recognition. Chapter 7 concludes the thesis.

CHAPTER 2

TECHNICAL BACKGROUND

This chapter provides technical background on a number of topics critical to the development, analysis, and assessment of super-resolution image enhancement for airborne sensor applications. Section 2.1 presents a mathematical model for digital 2D image formation that will be used throughout the remainder of the thesis. The model includes effects of both optics and pixel integration on the camera's focal plane array (FPA). In section 2.2, we take a more detailed look at the concept of resolution, which has an often erroneous interpretation. We also look at historical and current attempts, such as the Johnson criteria [41,42] and Task Performance Metric [43], which establish a theoretical link and means for requirement flow-down from high-level task performance objectives (e.g., probability of properly classifying an object through remote sensing) to the lower-level resolution specification of the imaging system (camera plus signal processing). In section 2.3, we provide background on the similarities and differences between visible band and infrared band cameras. In section 2.4, we show the history, alternative formulations, and state-of-the-art algorithms and software to solve the SR problem. In section 2.5, we provide background on inertial navigation systems which are one of the supplementary information sources, common in airborne platforms, that we are able to capitalize upon. Finally, in section 2.6, we introduce the high-fidelity Digital Imaging and Remote Sensing Image Generation (DIRSIG) simulation tool that we use as part of our algorithm performance evaluations.

2.1 Image Formation Model

Reference [2] provides a basic model, sufficient for developing and analyzing SR, of the process by which an analog scene in the environment is converted into a digital image. The four

critical components are the analog blur, the analog integration of the pixel detector, digital sampling, and noise. The process is depicted in Figure 2-1.

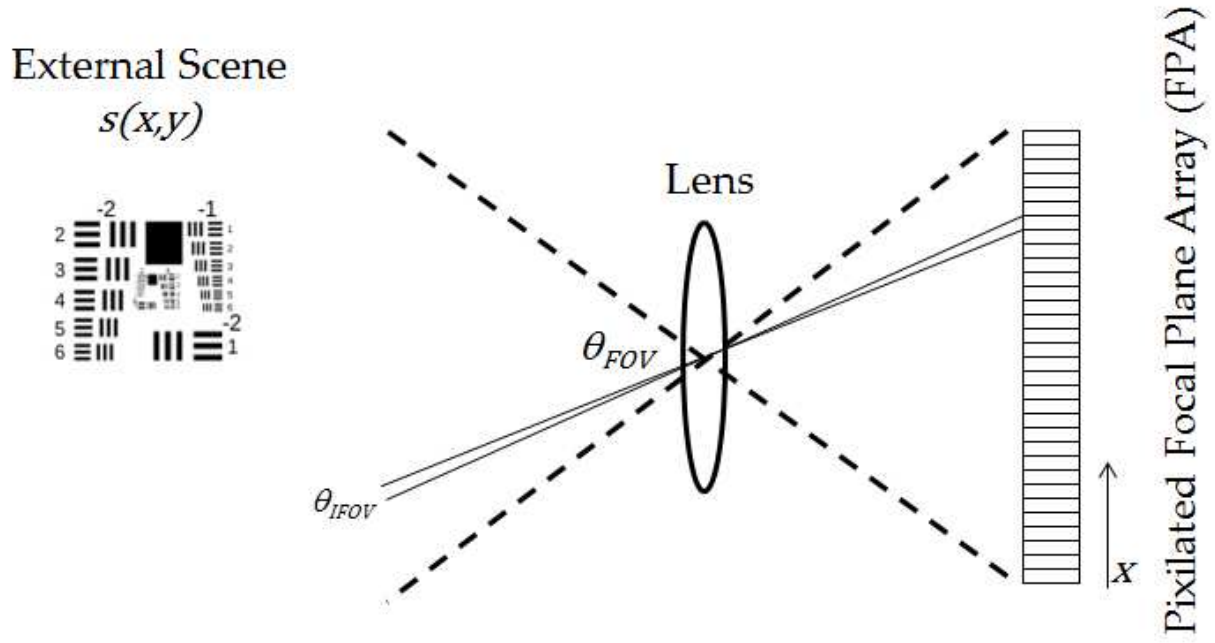


Figure 2-1: Model of the imaging process

Note that the spatial geometric properties of a camera are fixed in angular, not linear space. That is, properties such as the total field of view (FOV) is a fixed angle as is the angular sub-tense, instantaneous FOV ($IFOV$), of each individual pixel. Therefore, when discussing spatial and spatial-frequency properties of the camera and its output images, we will use angular as opposed to linear units (i.e., milli-radians vs. milli-meters). The choice of angular units is particularly relevant in the discussion of remote sensing. Angular units are also prevalent when quantitatively representing the capability of the human eye [43].

The analog imaging process is given, for incoherent light systems, by

$$\boldsymbol{i(x,y) = s(x,y) * h(x,y) * \frac{1}{ab} rect\left(\frac{x}{a}, \frac{y}{b}\right)}, \tag{2.1.1}$$

where $(\boldsymbol{x}, \boldsymbol{y})$ are spatial coordinates (a typical unit would be milli-radians), $\boldsymbol{s}(\boldsymbol{x}, \boldsymbol{y})$ is the external scene, $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y})$ is the blur expressed as a point-spread function (psf), and "$*$" denotes the convolution operator. The function $\frac{\mathbf{1}}{\boldsymbol{ab}}\boldsymbol{rect}\left(\frac{\boldsymbol{x}}{\boldsymbol{a}}, \frac{\boldsymbol{y}}{\boldsymbol{b}}\right)$ represents the integration over the area of a pixel detector with spatial dimensions $\boldsymbol{a}$ x $\boldsymbol{b}$. It is generated from the rectangle function

$$\boldsymbol{rect}(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} \mathbf{1}, \mathbf{0} \leq \boldsymbol{u} \leq \mathbf{1}, \mathbf{0} \leq \boldsymbol{v} \leq \mathbf{1} \\ \\ \mathbf{0}, \boldsymbol{othwewise} \end{cases} . \tag{2.1.2}$$

The actual output of the camera is discrete with

$$\boldsymbol{i'}(\boldsymbol{m}, \boldsymbol{n}) = \boldsymbol{i}(\boldsymbol{m\Delta x}, \boldsymbol{n\Delta y}), \tag{2.1.3}$$

where $(\boldsymbol{m}, \boldsymbol{n})$ are integer pixel indices and $\boldsymbol{\Delta x}$, $\boldsymbol{\Delta y}$ are the spatial separation between individual pixels in the sampling lattice. Note,

$$\boldsymbol{a} \leq \boldsymbol{\Delta x}, \boldsymbol{and} \boldsymbol{b} \leq \boldsymbol{\Delta y}, \tag{2.1.4}$$

with the equality holding in the case of a non-reticulated focal plane array; i.e., one with a 100% fill-factor.

One of the key limitations of the resolution of the imaging system, which is completely independent of pixel density, is the lens blur, quantified by $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y})$. The blur is defined as the impulse response of the imaging system to a single directional ray of incoming light. Although this is a property of the specific lens design, all cameras have a minimum, theoretical blur limit, known as the Rayleigh diffraction limit. The functional form of the lens blur, corresponding to a diffraction limited system, is given as a Bessel function. However, due to the inevitable contribution of effects such as residual lens aberration, jitter, atmospheric turbulence, optical

bandwidth, and electrical cross-talk between pixels, the blur is, in general, adequately represented by a Gaussian as

$$\boldsymbol{h(q) = exp\left(^{-q^2}/_{2\sigma^2}\right) and \sigma = 0.42\lambda/D,} \quad (2.1.5)$$

where $\boldsymbol{q}$ is the angular distance from the center of the point-spread, $\boldsymbol{\lambda}$ is the center wave-length of the camera's spectral bandpass, $\boldsymbol{D}$ is the aperture diameter, and $\boldsymbol{\sigma}$ is the standard deviation of the resulting Gaussian (expressed in milli-radians). It is also useful to consider the blur in the frequency domain, where it is expressed as the modulation transfer function (MTF), given by

$$\boldsymbol{H(\omega) = F\{h(q)\} = exp(-2\pi^2\sigma^2\omega^2),} \quad (2.1.6)$$

where $\boldsymbol{\omega}$ is spatial frequency and $\boldsymbol{F\{h(q)\}}$ represents the Fourier transform of $\boldsymbol{h(q)}$.

To be complete, the model for $\boldsymbol{H(\omega)}$ should also contain factors such as atmospheric contributions to overall blur. This is important for astronomical applications. However, for most CCD type cameras, the blur is dominated by the optics [44].

In some cases, for complex optics, the simplified blur model in (2.1.5) and (2.1.6) may be inadequate as the blur function is not stationary but is also a function of the specific position of the light ray on the FPA. In this case, the image formation model in (2.1.1) is still valid, but there is no direct and global conversion to the frequency domain representation. As many analytic relationships and results associated with the of the image formation model are most naturally expressed in the frequency domain, systems with a spatially varying blur are more difficult to analyze.

For a digital camera, once the blurred scene image is presented to the Focal Plane Array (FPA), it is further influenced by pixelization and the sampling process. Analogous to sampling in

the time domain, the sampling period of a digital camera is equal to the angular sub-tense, or $\boldsymbol{IFOV}$ of each individual pixel. For a narrow $\boldsymbol{FOV}$ camera, the $\boldsymbol{IFOV}$ is near constant and given by

$$\boldsymbol{IFOV = FOV/\#pixels.} \tag{2.1.7}$$

This corresponds to a sampling frequency, $\boldsymbol{\omega_s}$, of

$$\boldsymbol{\omega_s = 1/IFOV.} \tag{2.1.8}$$

Per the Nyquist-Shannon sampling theorem, the largest spatial frequency that can be detected by the camera without aliasing, $\boldsymbol{\omega_{max}}$, is given by

$$\boldsymbol{\omega_{max} = \omega_s/2.} \tag{2.1.9}$$

Any spatial frequencies above $\boldsymbol{\omega_{max}}$, from the external scene, that pass through the optics will, therefore, be aliased. The concept of aliasing is best represented in the frequency domain based on the aliasing property of the discrete Fourier transform (DFT) [45]. If we let $\boldsymbol{I(\omega_x, \omega_x)}$ be the continuous Fourier transform of the analog image $\boldsymbol{i(x, y)}$ given by (2.1.1) and $\boldsymbol{I'\left(\frac{r_x}{M}\omega_s, \frac{r_y}{N}\omega_s\right)}$ be the DFT of the sampled image $\boldsymbol{i'(m, n)}$ given by (2.1.3), then the two spectrums are related by the aliasing property

$$\boldsymbol{I'\left(\frac{r_x}{M}\omega_s, \frac{r_y}{N}\omega_s\right) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} I\left(\frac{r_x}{M}\omega_s - k\omega_s, \frac{r_y}{N}\omega_s - l\omega_s\right),} \tag{2.1.10}$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ are are the width and height of the discrete spectrum. The DFT only exist at integer values of the indicies $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$ which range between $\pm\boldsymbol{M/2}$ and $\pm\boldsymbol{N/2}$ respectively ($\boldsymbol{M}$ and $\boldsymbol{N}$ even). Each element, therefore, of the discrete transform is a linear superposition of a base analog frequency components between $\pm\,\boldsymbol{\omega_s/2}$ and all of the aliased analog frequency components displaced by integer multiples of $\boldsymbol{\omega_s}$.

Geometric Effects

For purposes of modeling the geometric characteristics of imaging sensors, it is convenient to utilize a normalized perspective projection model of the image sensor as discussed in [46] (see Figure-2). In such a model, the image plane is considered to be located at a unit distance from the focal point such that a 3D object located at space vector $\boldsymbol{R^S} = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z}]^{\boldsymbol{T}}$, in a coordinate system $\boldsymbol{S}$ attached to the sensor, will be projected to a normalized pixel location

$$[\boldsymbol{x'} \quad \boldsymbol{y'} \quad \boldsymbol{1}]^{\boldsymbol{T}} = [\boldsymbol{x/z} \quad \boldsymbol{y/z} \quad \boldsymbol{1}]^{\boldsymbol{T}}. \tag{2.1.11}$$

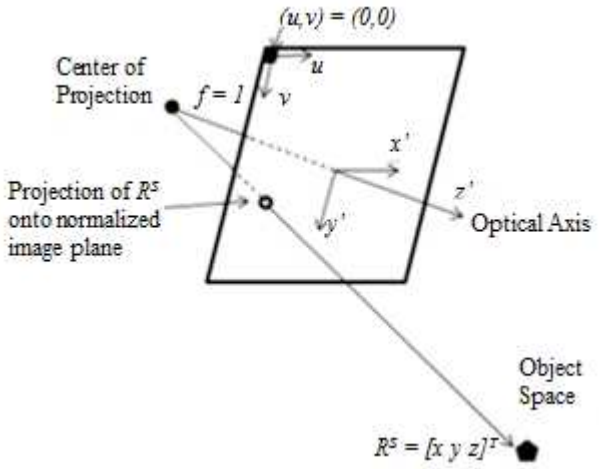


Figure 2-2: Normalized perspective projection model

For an idealized "pin-hole" sensor model, the pixel $(\boldsymbol{u}, \boldsymbol{v})$, illuminated due to a ray of light emanating from the direction of $\boldsymbol{R^S}$, will be related to the normalized pixel location $(\boldsymbol{x'}, \boldsymbol{y'})$ by a simple offset and scale factor; i.e., $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u_0}, \boldsymbol{v_0}) + (\boldsymbol{\gamma_x x'}, \boldsymbol{\gamma_y y'})$ where $(\boldsymbol{u_0}, \boldsymbol{v_0})$ represents the pixel location where the optical axis intersects the image plane and the factors $\boldsymbol{\gamma_x}$, $\boldsymbol{\gamma_y}$ represent the conversion from spatial dimensions to pixel dimensions. However, a true pin-hole camera is an unachievable idealization because the infinitesimal aperture would not permit the imaging sensor to collect any light energy. In reality, therefore, imaging sensors contain a finite diameter aperture

and a focusing lens. The aperture permits sufficient light to pass to the sensing elements while the lens focuses the light rays originating from a single point in space to a small region on the sensors sensing elements (the blur region). As discussed in [46], there are multiple models for the projection of a lens such as thin lens approximation, thick lens approximation, etc. However, in general, for any camera, through modeling and/or calibration, the mapping between the projection $(\boldsymbol{x'}, \boldsymbol{y'})$ of $\boldsymbol{R^S}$ onto the normalized image plane and the observed pixel location $(\boldsymbol{u}, \boldsymbol{v})$ on the actual focal plane is given by a distortion mapping function, $\boldsymbol{f}(\cdot)$, such that

$$(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{f}(\boldsymbol{x'}, \boldsymbol{y'}), \text{ and its inverse} \tag{2.1.12}$$

$$(\boldsymbol{x'}, \boldsymbol{y'}) = \boldsymbol{f^{-1}}(\boldsymbol{u}, \boldsymbol{v}). \tag{2.1.13}$$

The function $\boldsymbol{f}(\cdot)$ is an intrinsic characteristic of the imaging sensor. It is intrinsic because it is a property only of the sensor itself and is independent of how it is installed on a platform. For a given lens design, $\boldsymbol{f}(\cdot)$ may be modeled by an optical design ray tracing software package such as Zemax, which is sold by Zemax Development Corporation of Bellevue, Washington. However, for inexpensive, off the shelf imaging systems, there may be a large sensor to sensor variation due to manufacturing tolerances. As such, $\boldsymbol{f}(\cdot)$ for each specific unit may need to be measured either at the factory or by the user prior to installation on the aircraft. A number of techniques for performing the calibration of the intrinsic parameters are discussed in [46-48]. All techniques, in general, involve imaging a calibration target and fitting the measurements to a lens distortion model via a linear least-squares or other appropriate parameter estimation technique. For the remainder of this thesis, it is assumed that the intrinsic calibration function $\boldsymbol{f}(\cdot)$ has been determined by one of these methods and is available as an input to any image processing algorithms.

Once installed, the orientation of the imaging sensor relative to the platform is represented as a 3x3, orthonormal direction cosine matrix (DCM) mapping vectors from the platform coordinate system $\boldsymbol{P}$ to the sensor coordinate system $\boldsymbol{S}$; i.e. for an arbitrary vector $\boldsymbol{a}$,

$$\boldsymbol{a^S} = \left[\boldsymbol{T_P^S}\right]\boldsymbol{a^P}. \tag{2.1.14}$$

2.2 Resolution: Definition and Requirements

The resolution of a camera corresponds to our perception of "image sharpness" or, equivalently, our ability to discern fine detail within the image. In the case of digital cameras, the term resolution is often, and incorrectly, interpreted as the total number of pixels [49]. While pixel density is an important factor, resolution is also a function of the analog MTF of the camera's optics and electronics. As spatial frequencies in the scene increase, the contrast in the camera's output decreases. The relationship between input spatial frequency and output contrast is the spatial frequency response (SFR) of the camera. At a certain maximum spatial frequency, the cutoff frequency, the contrast will have decreased to a point that no useful information may be extracted from the image at or above that frequency. Quantitatively, resolution is defined as this maximum cutoff frequency of the spatial frequency response (SFR) [49]. The ISO 12233 standard [49] defines a set of methodologies to measure SFR and resolution which is further discussed in Appendix B.

Even with a quantitative definition of resolution, it is difficult to flow-down a minimum resolution requirement for a camera in order to meet the high-level performance requirements of resolution critical task, such as those listed in 1.1. In the case of remote detection and classification;

however, a relationship was first discovered in the 1958 work by Johnson [41,42] and then expanded in recent work [43].

Given the large variety of potential classification task, it would appear particularly difficult to establish a general and quantitative metric for the degree of "image sharpness" or resolution required to perform each tasks. That is, for example, should the task of classifying an aircraft be examined separately from that of classifying a human face or classifying a land vehicle? Fortunately, the 1958 work by Johnson [41-43] found empirically that, in fact, the resolution requirements for a very general class of object detection and classification tasks could be directly tied to the "detectibility" of a minimum, critical spatial frequency within the image.

Johnson broke the problem into three discrete task of increasing difficulty. The first task is detection which corresponds to the observer's ability to determine that something of interest is present within the scene. The second task is recognition which corresponds to the observer's ability to determine the type of object; e.g., separate a person from a vehicle, a vehicle from an aircraft, etc. The third task is classification which corresponds to the observer's ability to select the object from a class of related objects; e.g., a specific human face, a Cesna-172 vs. a Beechcraft, etc. Johnson expressed the associated minimum critical frequencies in terms of detecting line-pairs (lp), as presented in a calibration bar target such as the 1951 USAF resolution target [50] shown in Figure 2-3 or other targets discussed in Appendix B, across the object. The general requirement is 1.0, 4.0, and 6.4 lp for the three task (detection, recognition, and classification) respectively. These thresholds were derived by grading the performance of human subjects attempting to perform the tasks using images of models of typical objects, noting where they achieved a 50%

probability of success, and comparing that result to the maximum density of line-pairs they could resolve with equivalent accuracy on a bar target of equivalent contrast [43].

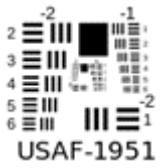


Figure 2-3: USAF 1951 bar resolution target

The concept is exemplified in Figure 2-4 for a Cesna-172 aircraft. Given a required range, $\boldsymbol{R}$, at which to perform the task, we convert the Johnson criteria into a critical spatial frequency requirement for the camera as

$$\boldsymbol{\omega_c = (\#lp)R/d_c,} \tag{2.2.1}$$

where $\boldsymbol{d_c}$ is the critical dimension of the object, $\boldsymbol{\#lp}$ is the number of line-pairs required to perform the task, and $\boldsymbol{\omega_c}$ is the resulting, critical spatial frequency.

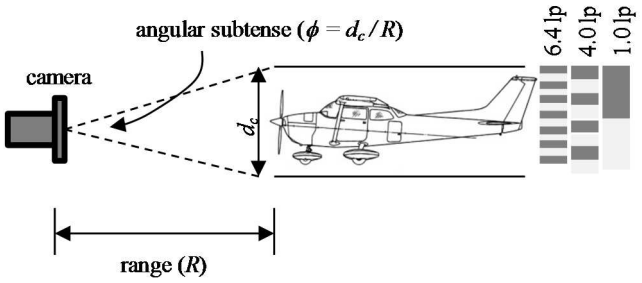


Figure 2-4: Example of line-pairs across an object

There are multiple ways of determining the critical dimension [42,51], the most conservative is to select the minimum dimension across the object. The critical dimension of the Cesna-172 is 4.75 m; so, for the example scenario of Figure 2-4, at a range of 3 km, a camera would have to be able to detect spatial frequencies of 0.63, 2.5, and 4.0 cycles / milli-radian respectively for the task of detection, recognition, and classification of the aircraft. Therefore, if

an imaging system (camera plus signal conditioning and display) would allow an operator to discern, with high probability, line-pairs of density 0.63, 2.5, and 4.0 cycles / milli-radian on a bar target, they would be able to perform the detection, recognition, and classification of aircraft of similar dimension to a Cesna-172 with equivalently high probability.

This same procedure may be used to determine critical frequency requirements for other related task such as face recognition at a distance, etc. Figure 2-5 shows an example of the Johnson criteria applied to the three tasks for the case of a human face from the Yale B face database [15,52-54].

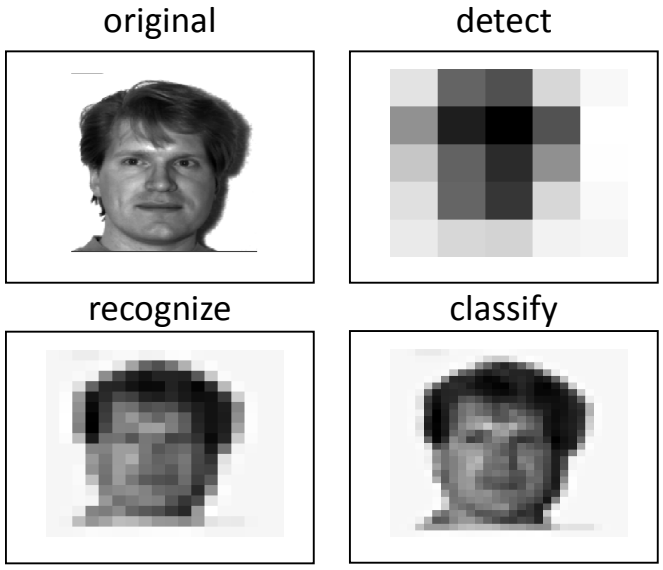


Figure 2-5: Example of the Johnson criteria applied to detection, recognition, and classification of a human face [52]

As an additional example, [15] shows that probability of correct face classification falls below 50% when there are fewer than 12 pixels between the eyes (the inner pupil distance (IPD)). This is quite consistent with the Johnson criteria prediction that 50% classification probability should require the ability to resolve 6.4 line-pairs (12.8 pixels) across the critical dimension.

Over the years, the basic Johnson criteria has been superseded by more sophisticated models, such as the "targeting task performance metric" [43]; however, it still remains as a simple and useful rule-of-thumb for approximating camera requirements. Also, the two underlying principles, first that the ability of an observer to perform a complex object detection and classification task is directly tied to the sensor's ability to resolve high spatial frequencies, and, second, that this resolving power may be evaluated in a subject independent way using calibration bar targets have remained standard. In Chapter 4, we will capitalize on these principles as we derive a metric for evaluating the success of SR algorithms. In a typical engineering proposal or projects, fundamental decisions, such as camera pixel density, must often be made up front prior to detailed development specific to the problem domain. At this point in the design process, there will likely be many alternative concepts (some which may involve SR and some which do not) that must be quickly pruned based on feasibility and risk. In order to support this, simple results such as the Johnson criteria, although imperfect and not tailored to the specific problem domain, provide a method to trade off different camera options early in the development process.

An alternate perspective, on the basis of the Johnson criteria, is to consider a classification problem in a different orthogonal frequency basis than the standard Fourier basis. To illustrate, consider the common principle component analysis (PCA) approach to object classification. Each observation is decomposed into its projections onto a set of orthogonal eigenvectors. Just as a Fourier transform decomposes an observation into its projections upon the orthogonal basis set of sinusoids, the eigenvectors from PCA allow us to represent the observation in an alternate, more task specific, projection. The first 36 eigenvectors (also called eigenfaces) from the Yale B face

dataset [15,52] are shown in Figure 2-6. The use of eigenface projection for face classification is described in [54].

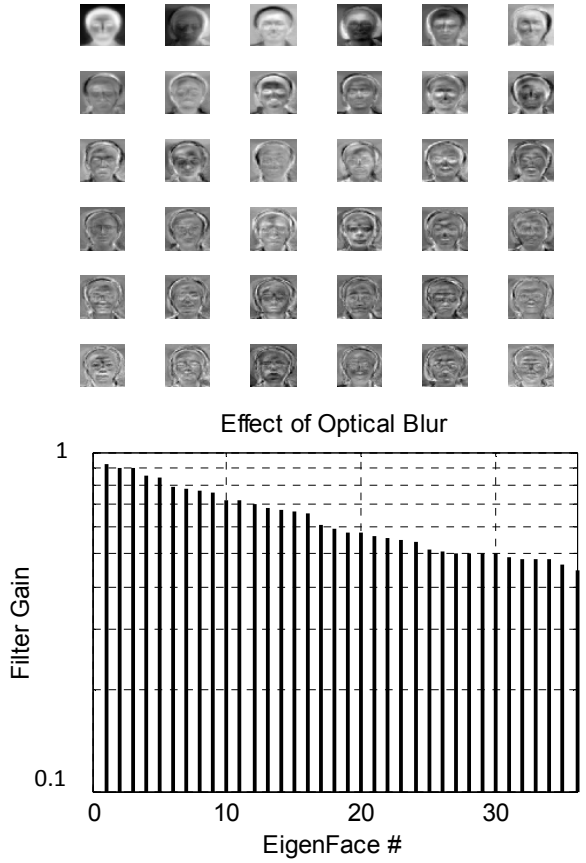


Figure 2-6: Alternate frequency perspective. Eigenfaces (top). Camera's blur expressed as attenuation of each orthogonal eigenface (bottom).

The figure also shows the effect of the camera's optical MTF on the individual eigenfaces. The filter gain for eigenface $\boldsymbol{k}$, shown in Figure 2-6, is given by the relative magnitude of the image of the eigenface after being subjected to the optical blur; i.e. $\|\boldsymbol{He_k}\|_2/\|\boldsymbol{e_k}\|_2$ where $\boldsymbol{H}$ is the optical blur matrix and $\boldsymbol{e_k}$ is the image of eigenface $\boldsymbol{k}$ represented as a vector. As expected, the larger number index eigenfaces are attenuated more significantly by the blur, reinforcing the intuition that the higher indexed eigenfaces represent higher spatial frequencies. Note, however,

that the broadband noise of the camera will be constant across frequencies. Therefore, the higher frequency eigenfaces will possess a lower signal to noise ratio (SNR) and, consequently, be less useful for identification as they are washed out by noise. The camera resolution, therefore, limits the number of eigenface projections that can be reliably applied towards the classification task. As the probability of successful classification depends upon the number of eigenface projections used, the spatial resolution of the camera directly impacts the task performance.

Additionally, the limited pixel density of the camera can cause aliasing of the higher index eigenfaces, leading to corruption of the projections onto the lower index eigenfaces. Unlike in the Fourier domain, it is not possible to specify a simple, specific eigenface index cutoff, above which aliasing will occur and below which it will not. Instead, the degree of aliasing will increase with higher eigenface indicies. This aliasing will further degrade any classification task performance.

2.3 Infrared Spectrum and Cameras

In many ways, 2D digital imagery from infrared and other non-visible band cameras is identical to that from visible band (0.390 – 0.750 μm) cameras. Figure 2-7 shows a side by side comparison of a scene captured by a visible red-green-blue (RGB) camera and a long wave infrared (LWIR) camera.

Figure 2-7: Side-by-side comparison of a visible RGB image (left) and LWIR image (right) of a common scene.

The classical infrared bands are dictated by the, somewhat rare, spectral regions where the atmosphere is transmissive. These are the short wave infrared (SWIR) from 0.9 to 1.7 μm, the mid wave infrared (MWIR) from 3.0 to 5.0 μm, and the long wave infrared (LWIR) from 7.0 to 13.5 μm. One of the most important differences between images taken in the different spectral bands isn't the imager itself but the physics determining the source of in-band photons from the scene. In all scenes, the light, or irradiance, reaching the camera is a combination of reflected and emitted photons. At one extreme, in the visible band, the irradiance is dominated by reflection and successful imaging is dependent upon either a natural or artificial illumination source. At the other extreme, in the LWIR band, irradiance is dominated by self-emission and images are largely unaffected by illumination. The SWIR and MWIR bands share characteristics of both.

Spectral, in-band self-emission of a material is given by Plank's law [55,56] as

$$\boldsymbol{L(T) = \int_{\lambda_1}^{\lambda_2} \varepsilon(\lambda) \frac{2hc^2}{\lambda^2\left(exp\left({}^{hc}/_{\lambda kT}\right)-1\right)} d\lambda,} \qquad (2.3.1)$$

where $\boldsymbol{L(T)}$ is the emitted radiance at temperature $\boldsymbol{T}$, $\boldsymbol{(\lambda_1, \lambda_2)}$ define the lower and upper wavelength limits of the spectral band, $\boldsymbol{\varepsilon(\lambda)}$ is the materials spectral emissivity, $\boldsymbol{h} = 6.626176\, x\, 10^{-34}$ joule-seconds is Plank's constant, $\boldsymbol{c} = 2.9979246\, x\, 10^{8}$ m/s is the speed of light, and $\boldsymbol{k} = 1.380662\, x\, 10^{-23}$ joules/kelvin is Boltzmann's constant. Using these units, and expressing $\boldsymbol{\lambda}$ in meters and $\boldsymbol{T}$ in Kelvins, results in $\boldsymbol{L(T)}$ having radiance units of Watts / (meter^2-steradian), where the steradian is a measure of solid angle. Setting the derivative of $\frac{2hc^2}{\lambda^2\left(exp\left(hc/\lambda kT\right)-1\right)}$ to zero and solving for $\boldsymbol{\lambda}$ at a given temperature provides the Wien displacement law which determines the wavelength of peak energy emission of a material as a function of temperature. It is given by

$$\boldsymbol{\lambda_{max}} = \frac{\boldsymbol{2{,}897.8\ \mu m \cdot Kelvin}}{\boldsymbol{T}}\ , \tag{2.3.2}$$

which is plotted in Figure 2-8.

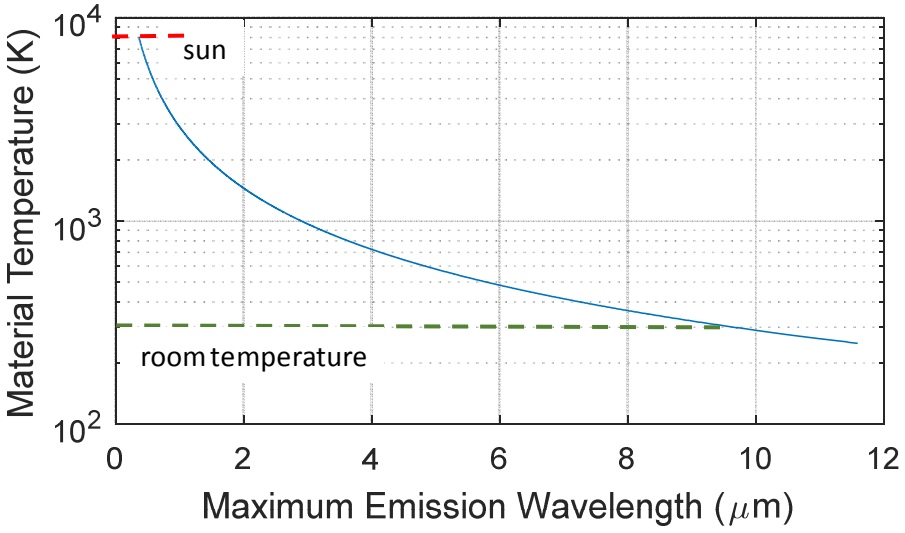


Figure 2-8: Relationship between temperature and peak spectral emission from Wien's displacement law

From the figure, we can see that objects near room temperature, which we are typically going to see in the natural environment, have a peak emission near the center of the LWIR band (7.5 to 13.5 μm). That explains why imaging in this band is dominated by emission. In contrast, energy from the sun, which is typically modeled at 8,000 K, peaks near the visible band (0.390 – 0.750 μm).

Although the physics of photon reflection and emission play a significant role in differentiating visible and infrared imagery, there are also important differences in the construction of the cameras. Principally, the high manufacturing cost of infrared sensors makes it impractical to reduce the size of individual pixels to match the diffraction-limited resolution scale of the optics. Both fabrication complexity as well as quantum efficiency limit the practical minimum pixel size [2]. Instead, the limiting factor for the resolution of most IR systems is the detector pixel size and density on the focal plane array (FPA) [57]. This means that IR sensors are Nyquist as opposed to Rayleigh limited [58]. As we will see in Chapter 4, this physical limitation of the infrared sensors means they will typically have a tighter blur circle relative to the pixel size (equation 2.1.3) which makes them particularly suited to resolution enhancement via SR.

2.4 Super-Resolution

Super-Resolution (SR) refers to a class of image processing algorithms which effectively increase the sampling density of a digital imaging system from $\boldsymbol{M\ x\ N}$ pixels to $\boldsymbol{kM\ x\ kN}$ pixels where $\boldsymbol{k}$ is some value greater than 1. The beginning of the field is typically credited to the 1984 work by Huang and Tsai [1] who, working in the frequency domain, provided a mathematical justification for the feasibility of multi-frame SR based on the aliasing property of (2.1.10). They also provided a set of frequency domain based recovery algorithms that utilized multiple, translationally-shifted image frames.

As with many image processing applications, there is complementary benefit to considering them in both the frequency and spatial domains. As mentioned above, much of the original work in super-resolution consisted of deriving algorithms in the frequency domain [45] directly from the perspective of unrolling aliased frequencies. However, these methods were very limited in the complexity of the image observation model they could handle; e.g., spatially varying blur. Therefore, most state-of-the-art SR algorithms address the problem in the spatial domain. Although preferable for generating algorithms, the spatial domain perspective can obscure understanding the basic phenomenology and, more importantly, the limitations of SR. For example, successful SR is reliant upon the presence of aliased scene content in the image. Otherwise, SR will be unable to recover higher frequency scene content for the simple reason that the high frequencies are removed prior to pixilation [58,59]. This fundamental limitation is not obvious when the image formation model is considered only in the spatial domain. Consequently, for fundamental analysis, the frequency domain perspective is still preferred.

Fortunately, aliasing is very common in image system design since other optical constraints, such as the desire for large field of view and small f-number, often outweighs concerns over aliasing [59]. An interesting extension of that concept is that, if it is known that SR processing will be a fundamental component of the aggregate imaging system, it is beneficial to design optics with as much aliasing as possible. As with other related computational imaging applications such as coded-aperture [57], this will make the system completely reliant on the signal processing as the raw imagery from the camera will be undesirable.

One of the simplest spatial domain approaches is that of registration and resampling of shifted low-resolution (LR) images onto a high-resolution (HR) lattice [2,45,61]. Other artifacts

such as noise, image blur, etc may then be removed by further processing of the HR image with other, non-SR, image enhancement algorithm. This method, as well as many other SR methods, assume the relative motion, or correspondence, between LR image frames is already known and their ultimate performance is highly dependent upon the accuracy of that information. External methods to generate known LR image correspondence include mechanical means (e.g., micro-scanner with known motion profile [3,61]) or estimation methods such as Lucas-Kanade or Horn-Shunk optical flow [62,63]. Even a micro-scanner approach, if the camera is on a moving platform, will require supplementing the known motion of the micro-scanner with the platform's motion. We also introduce a method in Chapter 3, suited to airborne applications, to provide the external correspondence information using inertial sensor data.

Although the resampling methods are simple, most state-of-the-art SR algorithms are model based. A typical, spatial domain model for SR is described in [64-66]. This model assumes that the imaging process has captured $\boldsymbol{L}$ LR images $\boldsymbol{y_k}$ from an unknown HR image $\boldsymbol{x}$. Note, in this formulation, both the LR and HR images are already in the discrete, pixelated domain. The LR images $\boldsymbol{y_k}$ and the HR image $\boldsymbol{x}$ consist of a total of $\boldsymbol{N}$ and $\boldsymbol{PN}$ pixels, respectively, where the integer $\boldsymbol{P} > 1$ is the factor of increase in resolution. In order to represent the problem compactly in matrix-vector notation, the images $\boldsymbol{y_k}$ and x are arranged in lexicographical order as $\boldsymbol{N\ x\ 1}$ and $\boldsymbol{PN\ x\ 1}$ vectors, respectively. The imaging process model includes warping, blurring (MTF), noise, and down-sampling as

$$\boldsymbol{y_k = AH_kC(s_k)x + n_k,} \tag{2.4.1}$$

where $\boldsymbol{A}$ is the $\boldsymbol{N\ x\ PN}$ downsampling matrix, $\boldsymbol{H_k}$ is the $\boldsymbol{PN\ x\ PN}$ blurring matrix, $\boldsymbol{C(s_k)}$ is the $\boldsymbol{PN\ x\ PN}$ warping matrix generated by the image motion vector $\boldsymbol{s_k}$, and $\boldsymbol{n_k}$ is the $\boldsymbol{N\ x\ 1}$ acquisition

noise. Given (2.4.1), the SR problem is to find the best estimate of the HR image $\boldsymbol{x}$ from the set of LR images $\boldsymbol{y_k}$ using prior knowledge about $\boldsymbol{C(s_k)}$, $\boldsymbol{n_k}$, and $\boldsymbol{x}$.

Most of the model-based SR literature utilizes some form of (2.4.1) with, potentially, tailoring to capture unique or extended characteristics of the imaging process. As an example, [70] augments the model by introducing an additional set of unknown parameters to allow for a global photometric correction from multiplication and addition across all pixels by a pair of scalars. This correction can account for effects such as non-uniform illumination.

The direct inversion of the model (2.4.1), or its variants, is well-known to be ill-posed [44,66,67] and, in practice, to produce numerous artifacts, particularly high-frequency oscillations. Many algorithms, such as the method of Farsiu [44], introduce regularization terms to constrain the output by encoding *apriori* assumptions about the image, noise, or motion models. These methods will invert (2.4.1) by numerically solving the unconstrained minimization problem

$$\boldsymbol{x = arg\,\min_{x}\left[\sum_{k=1}^{L}\|y_k - AH_kC(s_k)x\|^2 + \lambda\Upsilon(x)\right],} \qquad (2.4.2)$$

where $\boldsymbol{\Upsilon(x)}$ is a regularizing function that penalizes unnatural artifacts in the HR image $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ is tuning parameter (set by the user) which controls the trade-off between minimizing the data error given by $\boldsymbol{\sum_{k=1}^{L}\|y_k - AH_kC(s_k)x\|^2}$ and the regularization constraints. Equation (2.4.2) may be minimized using any number of numerical methods including gradient descent, gradient descent with variable step size, symmetric conjugant gradient, Levenberg-Marquardt, etc. It is also possible to minimize (2.4.2) over not just the unknown HR image $\boldsymbol{x}$ but to also to consider axillary parameters such as $\boldsymbol{s_k}$ as unknowns and jointly minimize over them as well.

For the regularization penalty function $\boldsymbol{\Upsilon(x)}$, the Farsiu algorithms uses a Tikhonov cost function of the form $\boldsymbol{\Upsilon(x) = \|\Gamma x\|^2}$ where the matrix $\boldsymbol{\Gamma}$ typically implements a high-pass operator

such as a derivative or Laplacian in order to eliminate noise and other high-frequency artifacts. Other popular regularization cost functions include the Gaussian Markov random field (GMRF), the Huber Markov random field (HMRF), the total variation (TV), the bilateral total variation (BTV), and the L_1 norm [45].

Hierarchical Bayesian Solutions

The regularization approach above is largely heuristic. The most powerful set of modern SR algorithms are based a probabilistic approach which embraces the fact that SR is truly a matter of inferring an HR image based on a combination of measured LR data as well as prior information [66]. Furthermore, a hierarchical Bayesian formulation provides a natural, powerful, and extensible method for properly incorporating all of the relevant information [67]. There are also well-established methods in place to solve problems cast within a Bayesian context. Retaining the image formation model of (2.4.1), we start by defining the joint posterior distribution of all parameters

$$\boldsymbol{Pr}\left[\boldsymbol{x}, \{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha} | \{\boldsymbol{y_k}\}, \{\boldsymbol{\Omega_{sk}}\}\right] \propto \prod_{\boldsymbol{k=1}}^{\boldsymbol{L}} \left(\boldsymbol{\beta_k^{mn/2}} \boldsymbol{exp}\left(-\frac{\boldsymbol{\beta_k}}{\boldsymbol{2}} \|\boldsymbol{y_k} - \boldsymbol{B_k}(\boldsymbol{s_k})\boldsymbol{x}\|^{\boldsymbol{2}}\right) \boldsymbol{exp}\left(-\frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{s_k^T \Omega_{sk} s_k}\right) \boldsymbol{Pr}[\boldsymbol{\beta_k}]\right) \boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}] \boldsymbol{Pr}[\boldsymbol{\alpha}], \tag{2.4.3}$$

where $\boldsymbol{\Omega_{sk}}$ is the measurement precision matrix of the warping parameters for LR frame $\boldsymbol{k}$, $\boldsymbol{\beta_k}$ is a hyper-parameter for the likelihood of measured image $\boldsymbol{k}$, $\boldsymbol{\alpha}$ is a hyper-parameter for high-resolution image prior model, $\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}]$ represents the HR image prior model, and the terms $\boldsymbol{Pr}[\boldsymbol{\beta_k}]$ and $\boldsymbol{Pr}[\boldsymbol{\alpha}]$ represent hyper-priors on the hyper-parameters. For convenience, we define $\boldsymbol{B_k} = \boldsymbol{AH_kC}(\boldsymbol{s_k})$ and the other symbols have been defined previously. The image prior model, $\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}]$, is related to the regularization cost functions introduced above. A common one is the

total variation (TV) prior, which is used for image reconstruction problems due to its inherent ability to retain sharp gradients at image edges [65,67], given by

$$\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}] = \boldsymbol{\alpha}^{\boldsymbol{MN/2}}\boldsymbol{exp}\left(-\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{MN}}\sqrt{\boldsymbol{\Delta h_i}^{\boldsymbol{2}} + \boldsymbol{\Delta v_i}^{\boldsymbol{2}}}\right), \tag{2.4.4}$$

where $\boldsymbol{\Delta h_i}$ and $\boldsymbol{\Delta v_i}$ are the horizontal and vertical gradients, respectively, for element $\boldsymbol{i}$ in the $(\boldsymbol{M\ x\ N}\ )$ HR image $\boldsymbol{x}$. $\boldsymbol{\Delta h_i}$ and $\boldsymbol{\Delta v_i}$ may be found using any reasonable gradient estimator. Another is the Gaussian, Simultaneous Auto-Regressive (SAR) prior given by

$$\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}] = \boldsymbol{\alpha}^{\boldsymbol{MN/2}}\boldsymbol{exp}\left(-\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}\|\boldsymbol{Cx}\|^{\boldsymbol{2}}\right), \tag{2.4.5}$$

where $\boldsymbol{C}$ is the Laplacian matrix. The SAR prior is also commonly used in the image recovery literature due to its simplicity. However, it is known to not preserve image edges as well as the TV prior.

The hyper-priors, $\boldsymbol{Pr}[\boldsymbol{\beta_k}]$ and $\boldsymbol{Pr}[\boldsymbol{\alpha}]$, are commonly modeled using either the uninformative distribution, in which case they have to be complete estimated from the data itself, or as Gamma distributions [67]. The Gamma distribution is convenient from an analytic tractability perspective in that it is conjugate to the normal distribution. A conjugate prior distribution is one that results in a posterior distribution having the same functional form as the prior. The ability of the algorithm to automatically learn the hyper-parameters from the data, either with an uninformative prior or some guidance via a Gamma prior, is a powerful capability. Other popular, non-Bayesian, SR methods leave the hyper-parameter estimation to the user which requires a long parameter-tuning process and can limit the applicability of the solution [67].

Note that the Bayesian formulation of (2.4.1) is flexible enough to handle the case where it may be desirable to specify parameters such as $\{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$ as fixed inputs as opposed to

unknowns to be jointly estimated along with the HR image. Such a case may exist if the registration parameters $\{\boldsymbol{s_k}\}$ have already been determined externally or if the user has already determined the desired settings of the hyper-parameters $\{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$. These cases are accommodated in (2.4.1) by simply setting the corresponding prior probability distribution to a delta function at the known input value. More likely, specifically in the case of $\{\boldsymbol{s_k}\}$, external estimates, such as those obtained by optical-flow or the method of Chapter 3, will come with some known bounded error. This later case is also handled naturally by (2.4.1) by setting the prior probability distribution to reflect the known error distribution of the input.

At this point, (2.4.3) is frequently converted into a numerical optimization problem by finding the parameters that minimize the negative log-likelihood of the posterior. This amounts to finding the Maximum-A-Posteriori (MAP) solution and will result in a cost function formulation nearly identical to that of (2.4.2). Although straightforward, this approach can not exploit the full potential offered by probabilistic modeling, as only the posterior mode is sought [68]. As the forward model becomes more complex, either due to more complex image priors or simultaneous estimation of other parameters such as camera motion, finding the mode can become very sensitive to local minima. It also doesn't account for the variation of the HR image due to uncertainty in the other latent parameters. Ideally, we would like to find the expected value of the HR image by marginalizing out the nuisance parameters [67,69].

The ideal objective is to find the expected value of the marginal distribution of x; that is,

$$\boldsymbol{E[x] = \int_{x,\{s_k\},\{\beta_k\},\alpha} x Pr\left[x, \{s_k\}, \{\beta_k\}, \alpha | \{y_k\}, \{\Omega_{sk}\}\right] dx d\{s_k\} d\{\beta_k\} d\alpha.} \quad (2.4.6)$$

By taking the expected value as opposed to the MAP estimate, we properly account for the, possibly large, variation in the auxiliary parameters $\{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$. Unfortunately, solving (2.4.6)

analytically for the marginal posterior is, in general, not tractable. This leaves a set of options. The first is to employ sampling methods such as Markov Chain Monte Carlo (MCMC) [68]. These work in principle but are computationally expensive. They do, however, lend themselves to mass parallelization. So, it is possible that with the increase in computing power, particularly using devices such as GPUs that exploit mass parallelization, these methods may see a reemergence. The second category is to use approximation methods such as Variational Bayesian Inference (VBI) which is well described in [69]. The VBI approach is used in state-of-the-art solutions [65-67].

In short, the VBI method postulates that the intractable posterior distribution in (2.4.1) may be approximated by the product of tractable distributions $\boldsymbol{q}$ such that

$$\boldsymbol{Pr}\left[\boldsymbol{x}, \{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha} | \{\boldsymbol{y_k}\}, \{\boldsymbol{\Omega_{sk}}\}\right] \cong \boldsymbol{q_x}(\boldsymbol{x})\boldsymbol{q_s}(\{\boldsymbol{s_k}\})\boldsymbol{q_\beta}(\{\boldsymbol{\beta_k}\})\boldsymbol{q}(\boldsymbol{\alpha}). \tag{2.4.7}$$

The functional form of each approximating distribution $\boldsymbol{q}$ is then found analytically by determining a form that minimizes the Kullback-Leibler (KL) divergence between the approximating distribution and the true posterior. Once the functional form of the approximating distribution is found, an algorithm finds the parameters of the distribution based on the data. At this point, the algorithm has found a fully Bayesian solution to the problem in that it has an analytic posterior and the marginal expectation in (2.4.6) is reduced to a tractable form $\boldsymbol{E_{q_x}}[\boldsymbol{x}]$.

An important result stemming from the VBI solutions is that, without prior assumption, the functional form of the distributions that minimize the KL divergence are, indeed, Gaussian for the latent image and registration parameters, $\boldsymbol{q_x}(\boldsymbol{x})$ and $\boldsymbol{q_s}(\{\boldsymbol{s_k}\})$. Also, the functional form of the hyper-parameters is Gamma, $\boldsymbol{q_\beta}(\{\boldsymbol{\beta_k}\})$ and $\boldsymbol{q_\alpha}(\boldsymbol{\alpha})$ [67].

It is noteworthy at this point that, although VBI methods are powerful, they have been criticized as the derivation is advanced, tedious, and model specific [71,72]. It is, consequently,

difficult to quickly experiment with different forward models as the VBI solution must be carefully re-derived each time. This has led to recent work in the area of "Black Box" Variational Inference [71] and an associated software package from Columbia University that implements automatic differentiation variational inference (ADVI) [72]. For the ADVI software, a user only needs to provide the posterior model; e.g. (2.4.1) and the input dataset. Conveniently, the posterior only has to be defined to within a scale factor which avoids some of the complex normalization factors that may show up in properly scaled expressions for conditional probabilities. The software uses the probabilistic modeling language STAN [73,74]. During our research, we experimented with using the ADVI software to solve the general SR problem but, at present, have not generated a successful outcome.

Other SR Approaches

In recent years, a new class of SR algorithms, not considered in this thesis, has emerged based upon the sparsity on natural imagery and machine learning techniques that use a dictionary approach to map patches between aliased, low-resolution imagery and the corresponding high-resolution imagery [4,75-77]. One potential advantage of these techniques is that they do not require multiple images.

2.5 Inertial Navigation System

Advances in both inertial navigation technology, specifically using micro electromechanical systems (MEMS), as well as in global positioning system (GPS) receivers has led to low SWaP and cost integrated GPS and INS. The MEMS inertial sensors are small, inexpensive, and consist of an orthogonal triad of linear acceleration measurement devices and an orthogonal triad of angle rate measurement devices. The MEMS inertial sensors are able to capture

rapid changes in the acceleration and angular velocity of the platform vehicle; however, they suffer from the phenomenon of drift. Drift occurs due to the presence of small bias errors in both the acceleration and angular velocity measurements which, over time, can integrate into large position errors. Reference [78] contains a general means for analyzing the error propagation of a typical inertial sensor. The bias error specification of an inertial sensor is proportional to its cost. For instance, an aircraft navigation grade inertial measurement unit, such as the Northrop Grumman LN-200, uses fiber optic gyros as opposed to MEMS and has bias errors low enough to operate for hours with acceptable performance; however, it is also weight and cost prohibitive for a small UAS.

In contrast to inertial sensors, GPS provides very good position accuracy. However, it provides no angular attitude information and its update rate is low. Attitude, for an aircraft, refers to the orientation relative to the local vertical reference frame and is typically expressed as a roll, pitch, yaw Euler sequence. Therefore, in order to provide a light-weight, inexpensive, yet accurate inertial sensor solution, MEMS data is blended with GPS measurements in a Kalman filter in order to capitalize on the relative benefits of both. The optimal blending of GPS and INS is able to maintain accurate position, velocity, as well as angular information. References [79-81] describe the design, observability considerations, and performance of such an integrated GPS/INS solution. The components used in [79] for testing the algorithm and establishing performance results are the inexpensive Crossbow AHRS-DMU_HDX inertial measurement unit, which has a mass of 50 g and fits in a 58 x 58 x 22 mm package, and the off-the-shelf Ashtech Z-XI1 GPS. Both of these components are practical for small UAS integration.

The output of the inertial navigation sensor is the current position, velocity, and attitude of the air vehicle relative to an earth fixed reference frame. Typically, the reference frame is aligned to the local north, east, and down directions ($\boldsymbol{NED}$). The position output is the geodetic latitude, longitude, and altitude of the platform ($\boldsymbol{\phi, \theta, h}$) using the world geodetic survey of 1984 (WGS-84) standard. The velocity output is provided in the $\boldsymbol{NED}$ coordinate system, V^{NED}. The attitude output is represented as a 3x3 orthonormal direction-cosine-matrix (DCM) mapping vectors in the $\boldsymbol{NED}$ coordinate system to the platform coordinate system, $\boldsymbol{P}$; i.e., for an arbitrary vector $\boldsymbol{a}$,

$$\boldsymbol{a^P = \left[T^P_{NED}\right] a^{NED}.} \tag{2.5.1}$$

2.6 Simulation for Image Enhancement Evaluation

We believe that, for testing and characterizing the performance of image processing algorithms and concepts, input data collected from real cameras and input data generated through simulation are complementary methods. Each provides unique and valuable capabilities. The obvious advantage of real cameras is that they, by definition, contain all of the complexity and interactions of the real world. However, data collections and experiments are limited to only using cameras that that have been fabricated and procured as well as to using scene scenarios that are practical to setup in the field or lab using available resources. Our ability to extend beyond available cameras and scene scenarios may be prohibited due to cost (particularly relevant for IR cameras), fabrication technology, procurement lead time, need for hardware customization, or a myriad of other factors. In addition, we never know the complete ground truth of either the scene or the camera's degradations (although we can partially mitigate by placing known targets within the scene). In contrast, simulation is model based and its fidelity is limited to the fidelity of the

model (this is mitigated through model validation against real data). However, it is able to produce synthetic data from an unlimited variety of cameras and scene scenarios. Simulated image data also has the advantage that the ground truth for both the scene and the camera is always known.

In this thesis, we use real data collections, from the devices discussed in Appendix A, and complement it with simulated imagery. At times, particularly when examining a specific theoretical concept, we use very simple simulated imagery; e.g., synthetic generation of bar targets in Chapter 4. However, when we require a synthetic scene to be as close as possible to real imagery, we use the validated Digital Imaging and Remote Sensing Image Generation (DIRSIG) model from Rochester Institute of Technology [82-88].

DIRSIG is an image generation tool that renders the radiance image presented at the aperture of a virtual camera placed at a specific position and pose within a virtual 3D world. In many respects, DIRSIG is conceptually similar to other 3D world rendering engines, such as OpenGL [89,90], as used in the computer graphics field and computer gaming industry. DIRSIG has two key differences. First, it covers the spectral wavelength range of 0.28 to 20.0 μm which includes the visible, SWIR, MWIR, and LWIR bands. Second, it greatly emphasizes fidelity over execution speed such that its output is suitable for algorithm development and performance testing [87]. The core DIRSIG software is supplemented by two other validated, high-fidelity physical models. The first is THERM, which was written by DCS Corporation in the late 1980s, and predicts the temperatures of all the objects in the scene using material thermodynamic properties (thermal conductivity, heat capacity, etc.) and environmental conditions (air temperature, relative humidity, wind speed, etc.) in order to properly compute photon emission. The use of THERM allows the user to simulate the scene at a multitude of weather and thermal loading conditions. It

also uses the MODerate resolution atmospheric TRANsmission (MODTRAN) atmosphere model to properly capture atmospheric effects such as attenuation, scatter, path radiance, lunar illumination, and solar illumination. In order to support these additional models and provide high-fidelity scenes, DIRSIG requires more information in the 3D world model, such as material properties, weather conditions, etc. than other, lower-fidelity rendering engines.

As with all rendering engines, DIRSIG requires the user to supply a high-fidelity 3D world model. The fidelity of the final image will be fundamentally limited by the fidelity of the input model. Fortunately, the stock DIRSIG release supplies a set of pre-generated 3D models. In our work, we use the stock "urban" scene which is an approximately 1.25 km square area constructed to match a region in Rochester, NY centered around the Genesee River Gorge and containing housing and commercial facilities in the surrounding area. We also use the stock airport scene which contains a number of aircraft and buildings. We modified the airport scene to place a couple resolution star targets, similar to that shown in Appendix B, on the ground so that we could make resolution measurements on the processed imagery.

A final, yet important, point about DIRSIG, is that the latest version 4 does not support a high-fidelity camera model. The output from the DIRSIG model is the pixelated, in-band radiance image presented at the aperture of the camera. DIRSIG does not model the camera's MTF, optical distortion, or noise characteristics. We add these effects in all of our simulations by using DIRSIG to render a radiance image that is 4x oversampled relative to the pixel density of the camera being simulated. We then apply the camera's blur to the oversampled image, down-sample, and add Gaussian noise to provide a targeted SNR defined as $\mathbf{20}\boldsymbol{log}_{\mathbf{10}}\frac{img_{max}-img_{min}}{\sigma_{noise}}$.

CHAPTER 3

EFFICIENT IMAGE CORRESPONDENCE MEASUREMENT USING INERTIAL SENSORS

In this chapter, we discuss a capability, unique to airborne observers, which is to derive image correspondence directly from the inertial measurement unit data as opposed to estimating it from the image alone. With respect to SR, as we saw in 2.4, many of the state-of-the-art solutions, particularly those based on Bayesian modeling, don't expect correspondence as an independent input but rather solve for it jointly with the unknown HR image. However, many of these algorithms still need to be seeded by an initial correspondence estimate. Other SR solutions, such as the algorithm of Farsiu [44], explicitly require the correspondence problem to be solved prior to calling the SR algorithm. Additionally, there are other image processing task, other than SR, where we want to directly and efficiently measure the dense correspondence field.

Traditional methods for generating dense correspondence maps, such as Lucas-Kanade and Horn-Schunk [63], continue to be a challenge in the image processing community due to both their computational complexity as well as their inherent reliance on sufficient image texture (i.e., the aperture problem). In addition, if the raw, LR imagery contains aliasing, the presence of aliasing can degrade the optical flow based estimates due to the fact that aliased frequency content doesn't shift in a consistent manner with the rest of the scene (this phenomenon is capitalized upon for SR algorithms but a problem for standard, image-based correspondence methods).

3.1 Using Aircraft Inertial Sensors for Correspondence Estimation

For conditions in which the image flow is dominated by the motion of the sensor platform as opposed to that of individual objects in the scene, an alternative method for determining frame to frame correspondence is to directly calculate it based upon data from an inertial navigation sensor (INS). Landscape video taken from an airborne platform, such as an unmanned aircraft system (UAS), is well suited to the above conditions. The landscape itself is essentially static in an earth fixed reference frame; so, all of the observed image motion is due to the combination of linear and angular motion of the sensor platform. Additionally, airborne platforms have the characteristics 1) they already have an embedded inertial navigation sensor as part of their avionics package and 2) SWaP restrictions may be prohibitive for the high performance computing power needed to estimate motion fields in real-time using image based algorithms.

One challenge in utilizing an inertial navigation sensor is that, in order to generate the sub-pixel accuracies required by algorithms such as SR, the inertial navigation sensor and the imaging sensor must be well aligned and calibrated. In general, this precision alignment will require specialized equipment which may not be practical for small platforms. Therefore, as part of the overall solution, we propose an autonomous, online calibration procedure.

Given the image formation model covered in 2.1 and characteristics of an INS covered in 2.4, it is possible to explicitly calculate the correspondence field between video frames. Let the 3D vector $\boldsymbol{R_G^{NED}}(\boldsymbol{x'}, \boldsymbol{y'}, \boldsymbol{k})$ represent the projection of the normalized image pixel, see Figure 2-2, $(\boldsymbol{x'}, \boldsymbol{y'})$ on frame $\boldsymbol{k}$ from the platform to the ground, represented in the $\boldsymbol{NED}$ coordinate system. Call this ground projection point $\boldsymbol{G}$. Let $(\boldsymbol{x'} + \boldsymbol{\Delta x}, \boldsymbol{y'} + \boldsymbol{\Delta y})$ represent the projection of the same ground point $\boldsymbol{G}$ back onto the normalized sensor image on frame $\boldsymbol{k} + \boldsymbol{1}$ (see Figure 3-1). Then,

$(\boldsymbol{\Delta x}, \boldsymbol{\Delta y})$ is the correspondence vector for pixel $(\boldsymbol{x'}, \boldsymbol{y'})$ between frame $\boldsymbol{k}$ and frame $\boldsymbol{k+1}$. The observed motion is a consequence of the combined linear and angular motion of the platform relative to the ground.

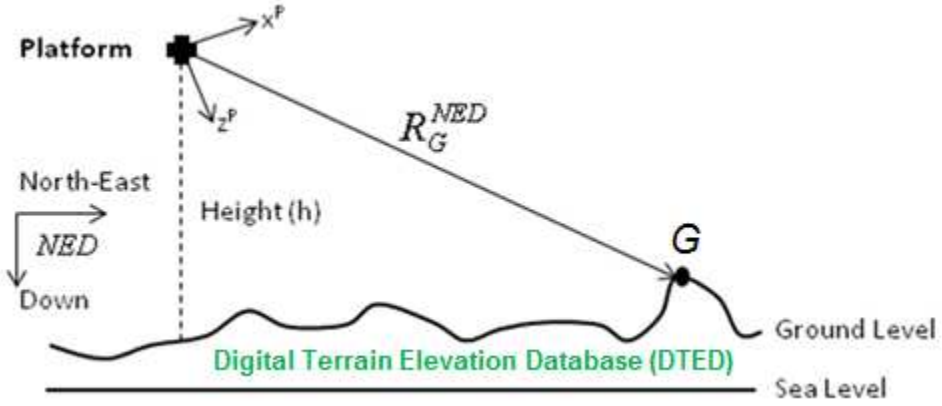


Figure 3-1: Projection to ground

In order to calculate $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y})$ for each pixel in the image, it is necessary to first compute the vector $\boldsymbol{R_G^{NED}(x', y', k)}$ corresponding to the pixel with normalized coordinates $(\boldsymbol{x'}, \boldsymbol{y'})$ according to

$$\boldsymbol{R_G^{NED}(x', y', k)} = \frac{\|\boldsymbol{R_G^{NED}(x', y', k)}\|}{\|[\boldsymbol{x'} \quad \boldsymbol{y'} \quad \boldsymbol{1}]^T\|} \left[\boldsymbol{T_{NED}^P}\right]\left[\boldsymbol{T_P^S}\right]^T \begin{pmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{1} \end{pmatrix}, \tag{3.1.1}$$

where the subscript "$\boldsymbol{k}$" on the $\boldsymbol{NED}$ to platform DCM, $\left[\boldsymbol{T_{NED}^P}\right]_{\boldsymbol{k}}^{\boldsymbol{T}}$, indicates that it represents the orientation of the platform at a time coincident with video frame $\boldsymbol{k}$. The "T" superscript indicates the matrix transpose. Because the DCM matrices are orthonormal, the matrix transpose is equivalent to the matrix inverse. The magnitude of the projected line from the platform to the ground point $\boldsymbol{G}$, $\left\|\boldsymbol{R_G^{NED}(x', y', k)}\right\|$ is calculated based on the altitude and position of the platform. This calculation is discussed in more detail below. The second step is to calculate the translational motion of the platform, $\boldsymbol{\Delta R}$, based on the average velocity. That is,

$$\boldsymbol{\Delta R = \Delta t \frac{V_{k+1}^{NED} + V_k^{NED}}{2}},\tag{3.1.2}$$

where, again, the subscript "$\boldsymbol{k}$" on the velocity denotes the velocity indicated by the inertial navigation sensor at a time coincident with the video frame $\boldsymbol{k}$. The time period $\boldsymbol{\Delta t}$ is the time interval between frame $\boldsymbol{k}$ and frame $\boldsymbol{k+1}$. ΔR is expressed in the NED coordinate system. The third step is to adjust the position of the fixed ground point $\boldsymbol{G}$ relative to the platform using the position change, $\boldsymbol{\Delta R}$. That is,

$$\boldsymbol{R_G^{NED}(x' + \Delta x, y' + \Delta y, k+1) = R_G^{NED}(x', y', k) - \Delta R.}\tag{3.1.3}$$

The final step is to map the $\boldsymbol{NED}$ vector back into the sensor coordinate system $\boldsymbol{S}$. That is,

$$\begin{pmatrix} \boldsymbol{x' + \Delta x} \\ \boldsymbol{y' + \Delta y} \\ \boldsymbol{1} \end{pmatrix} = \boldsymbol{\frac{1}{\alpha}}\left(\left[\boldsymbol{T_P^S}\right]\left[\boldsymbol{T_{NED}^P}\right]_{\boldsymbol{k+1}} \boldsymbol{R_G^{NED}(x' + \Delta x, y' + \Delta y, k+1)} \right),\tag{3.1.4}$$

where α is a normalizing scale factor such that the third element on the left-hand side of equation (3.1.4) is equal to unity. Equation (3.1.4) is rearranged into a form suitable for computer implementation by substituting in equation (3.1.3) and subtracting the vector $[x' \quad y' \quad 0]^T$ from both sides. This yields,

$$\begin{pmatrix} \boldsymbol{\Delta x} \\ \boldsymbol{\Delta y} \\ \boldsymbol{1} \end{pmatrix} = \boldsymbol{\frac{1}{\alpha}}\left(\left[\boldsymbol{T_P^S}\right]\left[\boldsymbol{T_{NED}^P}\right]_{\boldsymbol{k+1}} \left(\boldsymbol{R_G^{NED}(x', y', k) - \Delta R}\right) \right) - \begin{bmatrix} \boldsymbol{x'} \\ \boldsymbol{y'} \\ \boldsymbol{0} \end{bmatrix}.\tag{3.1.5}$$

Equation (3.1.5) represents the final, closed-form solution to the image correspondence vector field. For every pixel location $(\boldsymbol{x', y'})$ on frame $\boldsymbol{k}$, it computes the correspondence vector $(\boldsymbol{\Delta x, \Delta y})$.

Range to Ground

The range to the ground point, $\left\|\mathrm{R}_{\mathrm{G}}^{\mathbf{NED}}(\mathrm{x}', \mathrm{y}', \mathrm{k})\right\|$, is obtained through the use of a digital terrain elevation database (DTED). DTED is a table indexed by latitude and longitude that provides the elevation of the ground above sea level. Global coverage DTED based on the Shuttle Radar Topography Mission (SRTM) is publicly available in 3 arc-second (level 1) or 1 arc-second (level 2) resolution from the Earth Resources Observation and Science Center (EROS) [91]. Using DTED, the GPS position of the platform is all that is needed to accurately calculate ground range. See the geometry in Figure 3-1.

Using DTED to find range to ground requires a search algorithm. The on-board GPS provides the sea-level altitude of the platform. From that position in space, the algorithm needs to search along the known direction $\widehat{\boldsymbol{d}} = \mathbf{R}_{\mathbf{G}}^{\mathbf{NED}}(\mathbf{x}', \mathbf{y}', \mathbf{k}) / \left\|\mathbf{R}_{\mathbf{G}}^{\mathbf{NED}}(\mathbf{x}', \mathbf{y}', \mathbf{k})\right\|$ until it finds the first intersection with the ground.

3.2 On-Line Calibration

The problem of accurately aligning an inertial navigation sensor to a visual sensor is similar to the well-studied problem of transfer alignment between an aircraft inertial navigation sensor and a secondary inertial navigation sensor hosted on a peripheral device, such as a missile. Classical transfer alignment is discussed in [92-94]. Classical transfer alignment is relatively straightforward as it is based upon the direct, one-to-one comparison of the angular velocity and linear acceleration outputs of two inertial navigation sensors. In contrast, the two-dimensional outputs of an imaging sensor and the three dimensional outputs of an inertial navigation sensor are not one-to-one comparable; however, for a given motion of the aircraft, they are related through (3.1.5). Transfer alignment methods are also autonomous. The only constraint is that the platform

must execute a multi-axis maneuver in order to provide observability of all of the alignment parameters. This is typically not a problem for an airborne platform where appropriate motions tend to happen naturally throughout the course of the flight; i.e., the coupled roll and yaw motion experienced during any turn maneuver.

Recently, additional work has been performed specifically on the topic of aligning inertial sensors to visual sensors on UAS platforms using either stellar observations [95] or tracking of known ground monuments [96]. Unfortunately, both of these approaches require the operator to setup special conditions for the alignment to take place. Therefore, this paper develops an approach similar to the classical transfer alignment problem which achieves observability through the natural motions of the aircraft during flight.

Assuming that intrinsic errors in both the inertial navigation sensor and the imaging sensor are minimized through factory calibration, the remaining error sources of interest are 1) misalignment in the installed orientation of the imaging sensor and 2) mis-synchronization between the image and inertial sensor data. Depending upon the specifics of the platform data bus and rate of maneuvers, the time synchronization error may be negligible. References to simultaneous angle and time delay estimation do not appear in most of the classical transfer alignment literature for alignment between components on the aircraft. Time delay estimation is, however, explicitly considered in [93] with regard to shipboard alignment, that is, the alignment between components on a ship. The technique used in [93] is to augment the parameter estimation states with an unknown time delay. Due to the possibility of large angle rate maneuvers that are possible in a UAS platform, explicit estimation of time delay is considered in the development below. The alignment and time synchronization errors may be written as small angle modifications

to both the DCM relating the attitude of the platform relative to the NED coordinate system as well as the DCM relating the orientation of the imaging sensor relative to the platform. That is,

$$\left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right] = (\boldsymbol{I} - \boldsymbol{\omega}_{\boldsymbol{x}}\boldsymbol{\tau})\left[\widehat{\boldsymbol{T}}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right], \text{ and} \tag{3.2.1}$$

$$\left[\boldsymbol{T}_{\boldsymbol{P}}^{\boldsymbol{S}}\right] = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{P}}^{\boldsymbol{S}}\right](\boldsymbol{I} - \boldsymbol{\delta}_{\boldsymbol{x}}), \tag{3.2.2}$$

where $\boldsymbol{I}$ represents the 3x3 identity matrix, ω is the 3-element angular velocity vector of the platform relative to NED (as returned by the inertial navigation sensor), τ represents the temporal mis-synchronization between the inertial and image data, and $\boldsymbol{\delta}$ is a three element vector representing the roll, pitch, and yaw misalignment of the platform to image sensor DCM. The subscript "x" applied to the vectors ω and $\boldsymbol{\delta}$ in (3.2.1) and (3.2.2) is an operator that converts them into the 3x3 skew-symmetric cross-product matrix; i.e., for a general, 3-element vector $\boldsymbol{\delta}$,

$$\boldsymbol{\delta}_{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{0} & -\boldsymbol{\delta}_{\boldsymbol{3}} & \boldsymbol{\delta}_{\boldsymbol{2}} \\ \boldsymbol{\delta}_{\boldsymbol{3}} & \boldsymbol{0} & -\boldsymbol{\delta}_{\boldsymbol{1}} \\ -\boldsymbol{\delta}_{\boldsymbol{2}} & \boldsymbol{\delta}_{\boldsymbol{1}} & \boldsymbol{0} \end{pmatrix}. \tag{3.2.3}$$

With this definition, the 3x3 matrices $(\boldsymbol{I} - \boldsymbol{\delta}_{\boldsymbol{x}})$ and $(\boldsymbol{I} - \boldsymbol{\omega}_{\boldsymbol{x}}\boldsymbol{\tau})$ are small angle approximations to the DCMs generated by rotations about the x, y, and z axes given by the elements of ω and $\boldsymbol{\delta}$ respectively.

In (3.2.1) and (3.2.2), the DCMs with the ^ symbol indicate the uncorrected matrices and the DCMs without the ^ represent the post-correction matrices. The goal is to estimate τ and $\boldsymbol{\delta}$ such as to improve the correspondence calculation. The technique for doing so is to apply a traditional image based optical flow algorithm to a small sub-set of the pixels on each video frame and find the values of τ and $\boldsymbol{\delta}$ that minimize the discrepancy between the correspondence vectors as measured via optical flow versus the correspondence predicted by the inertial sensor. Only a small number of data points are required to solve for the four unknowns contained in τ and $\boldsymbol{\delta}$.

Therefore, it is not necessary to generate a dense correspondence map using a computationally expensive, optical flow algorithm such a Lucas-Kanade [63]. Instead, the optical flow algorithm only needs to return relatively few correspondence vectors per frame. Once the error τ and $\boldsymbol{\delta}$, are resolved through the periodic on-line calibration, equation (3.1.5) is all that is required to find the dense flow field.

The parameter estimation for τ and $\boldsymbol{\delta}$ may be linearized to form

$$\boldsymbol{\alpha}\left[\begin{pmatrix}\boldsymbol{\Delta x}\\ \boldsymbol{\Delta y}\end{pmatrix}_{\boldsymbol{IP}} - \begin{pmatrix}\boldsymbol{\Delta x}\\ \boldsymbol{\Delta y}\end{pmatrix}_{\boldsymbol{IS}}\right] = \begin{pmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \mathbf{0}\end{pmatrix}[\boldsymbol{A} \quad \boldsymbol{A\omega}]\begin{pmatrix}\boldsymbol{\delta}\\ \boldsymbol{\tau}\end{pmatrix}, \tag{3.2.4}$$

where $\boldsymbol{IP}$ and $\boldsymbol{IS}$ refer to the correspondence vectors determined by the image processing and inertial system, respectively. $\boldsymbol{\alpha}$ is the normalization scale-factor defined in (3.1.4). Matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{P}}^{\boldsymbol{S}}\right]\left[\frac{\left\|\boldsymbol{R}_{\boldsymbol{G}}^{\boldsymbol{NED}}(x',y',k)\right\|}{\left\|[x' \quad y' \quad 1]^T\right\|}\left(\left(\boldsymbol{y}_{\boldsymbol{k+1}}^{\boldsymbol{P}}\right)_{\boldsymbol{x}} - \left[\widehat{\boldsymbol{T}}_{\boldsymbol{k}}^{\boldsymbol{k+1}}\right]\left(\boldsymbol{y}_{\boldsymbol{k}}^{\boldsymbol{P}}\right)_{\boldsymbol{x}}\right) - \boldsymbol{V}_{\boldsymbol{k+1}}^{\boldsymbol{P}}\boldsymbol{\Delta t}\right], \tag{3.2.5}$$

where

$$\boldsymbol{y}_{\boldsymbol{k+1}}^{\boldsymbol{P}} = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{k}}^{\boldsymbol{k+1}}\right]\left[\widehat{\boldsymbol{T}}_{\boldsymbol{P}}^{\boldsymbol{S}}\right]^{\boldsymbol{T}}\begin{pmatrix}\boldsymbol{x'}\\ \boldsymbol{y'}\\ \mathbf{1}\end{pmatrix}, \tag{3.2.6}$$

$$\boldsymbol{y}_{\boldsymbol{k}}^{\boldsymbol{P}} = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{P}}^{\boldsymbol{S}}\right]^{\boldsymbol{T}}\begin{pmatrix}\boldsymbol{x'}\\ \boldsymbol{y'}\\ \mathbf{1}\end{pmatrix}, \tag{3.2.7}$$

$$\left[\widehat{\boldsymbol{T}}_{\boldsymbol{k}}^{\boldsymbol{k+1}}\right] = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k+1}}\left[\widehat{\boldsymbol{T}}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k}}^{\boldsymbol{T}}, \text{ and} \tag{3.2.8}$$

$$\boldsymbol{V}_{\boldsymbol{k+1}}^{\boldsymbol{P}} = \left[\widehat{\boldsymbol{T}}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k+1}}\left(\frac{\boldsymbol{V}_{\boldsymbol{k+1}}^{\boldsymbol{NED}} + \boldsymbol{V}_{\boldsymbol{k}}^{\boldsymbol{NED}}}{\mathbf{2}}\right). \tag{3.2.9}$$

According to (3.2.4), each image correspondence data point generates two equations (horizontal and vertical displacements). Therefore, a minimum of two data points is required for a

solution to the four error parameters τ and $\boldsymbol{\delta}$. For a robust solution, many more data points are necessary yielding an over constrained linear relationship which may be solved using a standard least squares approach. Or, if *a priori* probability distributions are available for the measurements and unknown parameters, a maximum likelihood or maximum a posteriori (MAP) estimation method may be used.

In order to perform the calculations from this section, it is necessary to solve the full optical flow problem for a subset of points in the image. However, it is not necessary to expend the computational resources to find the optical flow for every point in the image. In [97], Shi and Tomasi propose a method to select the optimal points to track in an arbitrary image. The work is motivated by the closely related problem of extracting three-dimensional shape from motion, as discussed in [98], where a parameter estimation problem is solved using optical flow measurements of an image. In both applications, it is necessary to supply the filter only with measurements from points for which the optical flow estimate is accurate. Fundamentally, the accuracy of any image based optical flow estimate is improved in regions of the image with more diverse texture. Optical flow can not be determined reliably at all in regions of an image where there is little texture and can only be determined in a single direction in regions of the image with a near constant gradient direction. The approach in [97] is to quantitatively identify the regions of the image for which there is a large variation in the local texture gradient as these regions will produce the most accurate flow estimates. Only the optical flow vectors generated from pixels for which an accurate optical flow measurement is predicted are passed to the on-line calibration algorithm. Alternately, points obtained from the well-known SIFT or SURF algorithms may be used.

3.3 Monte-Carlo Simulation Experiment for Direct Correspondence Estimation

In this section, we use a Monte-Carlo based simulation to test our results.

Simulation Architecture

The ideal method of testing the above algorithms is to collect data from an aircraft or UAS equipped with an imaging sensor and inertial navigation sensor. However, for the purpose of this section, a simulation is used to maneuver a virtual UAS over a landscape in order to test the performance of the proposed algorithm. A top level structure of the simulation architecture is shown in Figure 3-2. The simulation consists of six interconnected modules: a six degree-of-freedom (6 d.o.f.) aircraft dynamics and autopilot model, a terrain model, an imaging sensor model, an inertial sensor model, and two alternate image processing modules. Image processing modules exist for both the classical Lucas-Kanade optical flow algorithm as well as the proposed closed-form inertial sensor algorithm. In principle therefore, to within the fidelity of the models, the inputs and outputs of the image processing modules are the same as if the data were generated from a real flight.

For this chapter, the platform selected to model for the simulation is the Silver Fox UAS manufactured by British Aerospace (BAE). The Silver Fox [99] is a gasoline powered UAS with a 2.4 m wingspan and weight of 11.4 kg. It is able to operate for up to 8 hours and cruise at a mission airspeed speed of 18 to 23 m/s. The platform contains the lightweight Piccolo avionics package and autopilot which includes the integrated GPS/INS navigation system as described in section 2. The UAS carries both an adjustable 2 to 46 degree FOV visual imaging sensor and a 36 degree FOV long wave infrared (7.5 to 13.5 micron) imaging sensor. The following paragraphs describe each module of the Silver Fox simulation in detail.

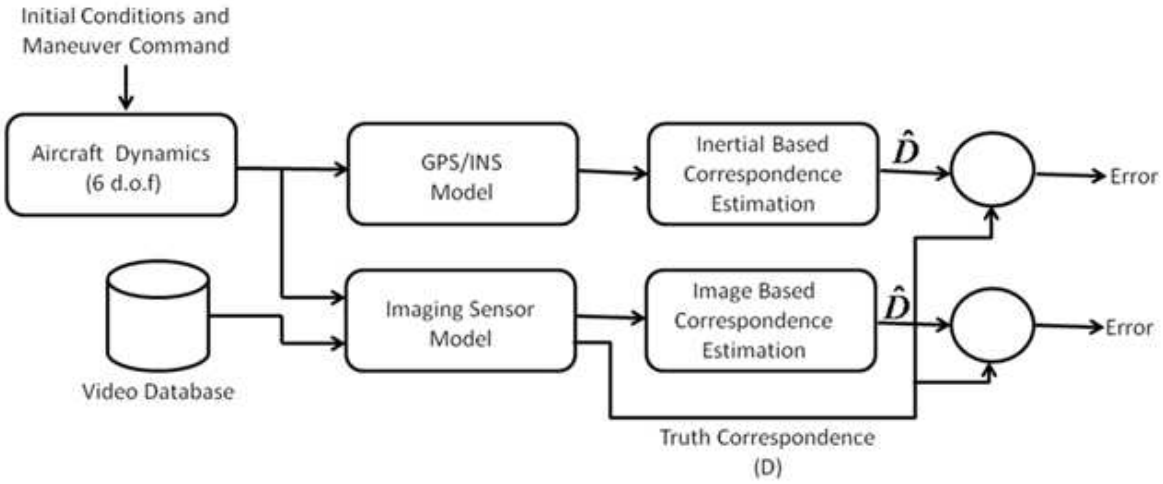


Figure 3-2: Top-Level simulation architecture

The 6 d.o.f. UAS model is utilized in order to provide a natural and physically realistic motion for the simulated aircraft. The term "six degrees of freedom" refer to the three translational (north, east, down) and three angular (roll, pitch, yaw) parameters required to fully specify the state of the aircraft in space. Although many more state variables are ultimately used in modern models to represent effects such as actuator dynamics, propeller dynamics, autopilot internal control variables, etc., the term "6 d.o.f." is historically used in aircraft simulation literature to indicate a high fidelity simulation. General techniques for modeling aircraft dynamics are presented in [100,101]. Reference [102] derives the specific parameters and equations of motion to model the Silver Fox UAS.

In order to present the image processing algorithms with realistic texture, Google Earth is utilized to supply a ground truth image based upon satellite terrain imagery. The section of terrain used for the simulation is a 1.16 km x 0.6 km region centered at latitude 41 degrees north and longitude 89 degrees west (near Peoria, Illinois). The image sensor model uses both the ground truth data as well as the aircraft's position and attitude (provided from the 6 d.o.f.) to render the

image sensed by the Silver Fox visual imaging sensor. The rendering is performed using the geometry of Figure 3-1 and projecting each pixel of the image sensor to the corresponding pixel(s) in the higher resolution ground truth image. Bi-cubic interpolation is used to handle the non-integer relationship between the pixels of the virtual sensor and the ground image. Figure 3-3 illustrates the ground image (left) and an example of the simulated sensor image (right). The example image in Figure 3-3 is generated with the simulated aircraft located at the center of the ground truth image, heading East at 450 m altitude, and with a 20 degree right bank (positive roll) angle. These values were selected for the example in the figure, at runtime of the simulation, the rendered sensor image is based, instead, upon the current output of the 6 d.o.f. The visual sensor on the Silver Fox UAS has a 30 degree field-of-view on a 640x480 pixel grid and runs at 100 Hz. The outline of the ground projection of the sensor's field-of-view on the larger ground plane is shown by the dashed white line in Figure (left). Although the image plane of the visual sensor is rectangular, the ground projection of the field-of-view is asymmetric due to the 20 degree roll angle of the aircraft.

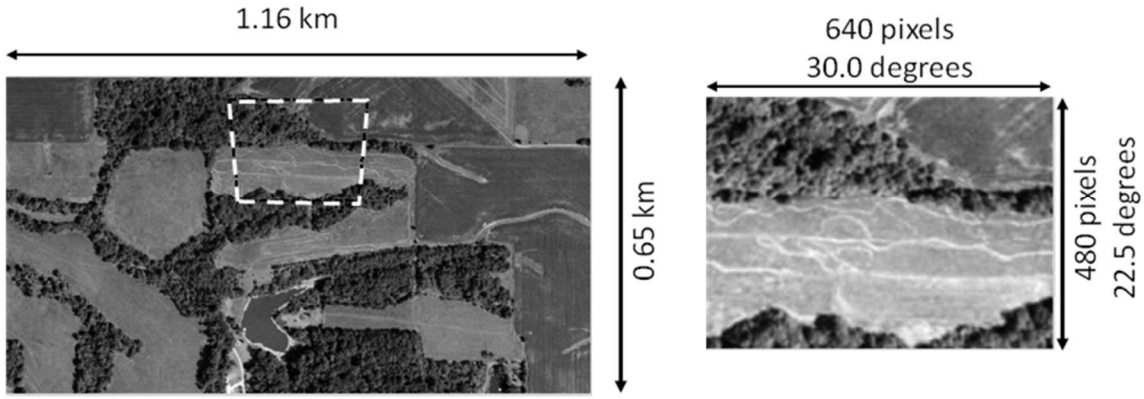


Figure 3-3: Simulated aircraft sensor imagery. Ground truth image (left). Projection into Silver Fox visual sensor (right). Imagery copyright Google Earth 2016.

The silver fox UAS contains both a GPS and INS. In order to model the typical errors associated with this set of sensors, [79] develops an integrated GPS/Inertial filter using a low-cost

Crossbow AHRS-DMU-HDX inertial measurement unit (IMU) and an Ashtech Z-XI1 GPS. The output of the blending Kalman filter has an empirically measured attitude error of 0.04 degrees one-sigma in the roll and pitch attitude axes and an error of 0.36 degrees one-sigma in the yaw attitude axis. For the purpose of this report, the simulation utilizes the model in [79] for the blended GPS/Inertial navigation sensor error characteristics.

The outputs of the GPS/Inertial model as well as the imaging sensor model are passed into the two alternative correspondence estimation algorithm modules for evaluation. The correspondence vector fields, $\widehat{\boldsymbol{D}}$, output by each of the estimation methods is compared to the true vector field, $\boldsymbol{D}$, as computed by (3.1.5) using truth motion data. The correspondence error per pixel is then defined as

$$\boldsymbol{\varepsilon} = \sqrt{(\boldsymbol{\Delta x_{EST}} - \boldsymbol{\Delta x_{TRUE}})^2 + (\boldsymbol{\Delta y_{EST}} - \boldsymbol{\Delta y_{TRUE}})^2}, \tag{3.3.1}$$

where the subscripts "$\boldsymbol{EST}$"and "$\boldsymbol{TRUE}$" correspond to the estimated and the truth correspondence respectively.

Simulation Scenario

To perform the evaluation, the simulation will execute for a total of ten seconds of virtual flight. The UAS will start out at t_{sim} = 0.0 s flying straight and level at a heading of East (90 degrees), an altitude of 450 m, and a cruise speed of 20 m/s. From t_{sim} = 2.0 s to t_{sim} = 8.0 s, the UAS will execute a six second, constant speed, constant altitude turn to a heading of North (0 degrees). The UAS will then resume straight and level flight until the simulation terminates at t_{sim} = 10.0 s. Due to the highly coupled nature of aircraft dynamics, as expressed by the equations of motion in [102], the simple turn maneuver will induce motion in all three of the aircraft attitude axes. A simplified illustration of the dynamics of an aircraft turn appears in Figure 3-4. The figure

shows the aircraft as it would appear to an observer behind the tail. The aerodynamic force generated by the wings acts in a direction perpendicular to the wings. During straight and level flight, this force is equal and opposite to the force of gravity acting on the body. In order to execute a turn, the autopilot rolls the aircraft in the direction of the turn such that the force normal to the wings has a horizontal component. The horizontal force causes the aircraft to turn in the correct direction. When the turn is complete, the autopilot rolls the aircraft back to level and resumes straight and level flight. However, if the autopilot were to only execute the roll maneuver, the aircraft would also start to descend due to the fact that the vertical component of the aerodynamic force is no longer equal to the weight. In order to maintain both constant altitude and airspeed during the roll, the autopilot must also simultaneously increase both the aircraft thrust and pitch. Therefore, in order to execute even the simple turn maneuver, all three of the aircraft attitude axes are exercised. As mentioned in 3.2, this coupled motion is necessary to provide observability for the on-line alignment calibration.

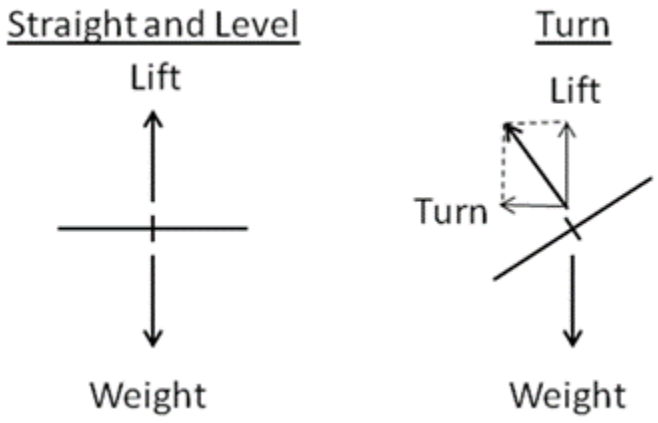


Figure 3-4: Rear-view illustration of an aircraft turn maneuver

Figure 3-5 shows the top level block diagram of the autopilot. Given the desired reference trajectory, the autopilot has the ability to control the deflection angles of the three control surfaces:

ailerons (∂_a), rudder (∂_r), and elevator (∂_e). It also has the ability to control the thrust, T, via the propeller power. At any given time, the six degrees of freedom of the aircraft are given by the state vector $\underline{\boldsymbol{x}}(\boldsymbol{t}) = \left[\boldsymbol{\phi} \quad \boldsymbol{\theta} \quad \boldsymbol{h} \quad \left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]\right]^{\boldsymbol{T}}$ (where the symbols for latitude, longitude, altitude, and the attitude DCM were defined in section 3.1) and its time derivative $\boldsymbol{d\underline{x}(t)}/\boldsymbol{dt}$. The acceleration of the state vector, $\boldsymbol{d^2\underline{x}(t)}/\boldsymbol{dt^2}$, is a coupled non-linear function of both the control inputs as well as the state itself. For simplification of notation, the time reference, $(\boldsymbol{t})$, is removed from the state variable $\underline{\boldsymbol{x}}$ in the block diagram.

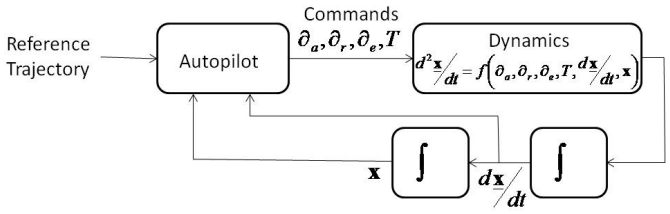


Figure 3-5: Top level autopilot block diagram

The maneuvers performed by the aircraft for the 10.0 s simulation appear in Figure 3-6. For the first two seconds, the aircraft is in its trim condition with a pitch of 2.9 degrees relative to the horizon and the propeller spinning at 3850 RPM. Under these conditions, at a velocity of 20 m/s, the forces and moments acting on the body are balanced such that it experiences zero linear or angular acceleration. In order to execute the turn maneuver at t_{sim} = 2.0 s, the aircraft begins banking to the left (negative roll angle) and the resulting lateral component of acceleration (see Figure 3-4) causes the aircraft to turn towards the North. The roll angle reaches its extreme deflection midway through the turn, at t_{sim} = 5.0 s, with a value of -30 degrees. At this point, in order to maintain the altitude and airspeed, the pitch has increased up to +10 degrees and the propeller speed has increased to 4340 RPM. After the mid-point, to slow the rate of turn, the bank

angle decreases back to 0.0 at t_{sim} = 8.0 s and the pitch and propeller RPM return to their straight and level trim values. Figure 3-7 shows the turn maneuver from a top-down view.

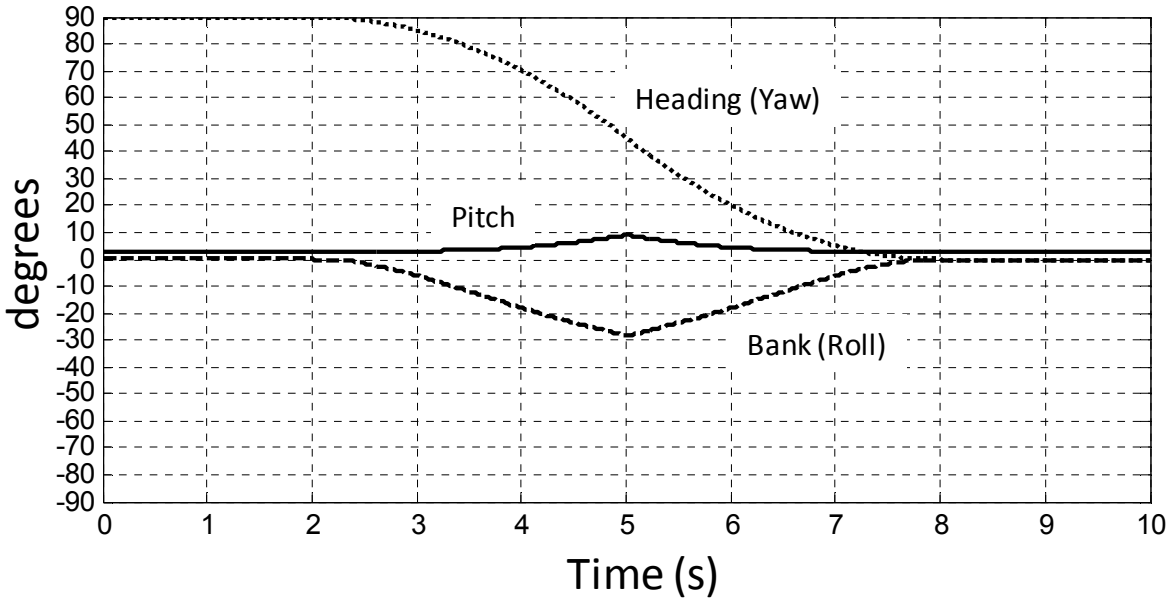


Figure 3-6: Aircraft maneuver during simulation

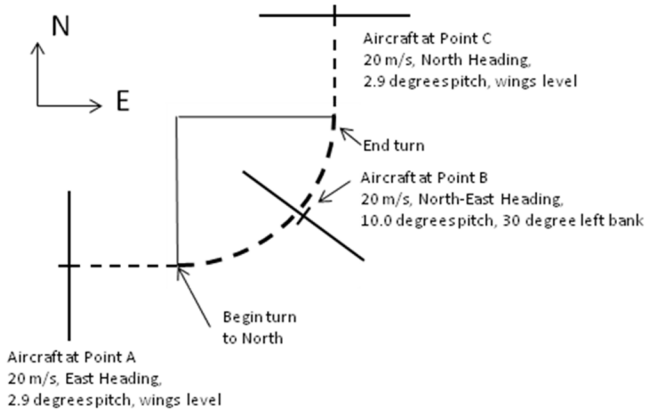


Figure 3-7: Top-Down view of simulated aircraft turn maneuver

Monte-Carlo

The closed-form inertial based solution, in reality, is not error free. The dominant errors affecting the performance are the velocity error, the absolute attitude error, and the attitude error drift between consecutive video frames. References [103,104] investigate GPS velocity error. Both of these references predict errors that are less than 1 m/s RMS.

Additionally, the Crossbow IMU, being used as the model for this investigation, has an angular readout noise of 8.5e-2 degrees / second (0.00085 degrees over a 1/100 second frame). In order to quantify the performance of the closed-form, inertial based solution in the presence of these random errors, a 50 run Monte-Carlo is utilized. For each run of the monte-carlo, the dense correspondence field of the entire 640x480 pixel image is computed using (3.1.5) with input parameters degraded as discussed below. After the 50 Monte-Carlo runs are completed, the error statistics for the closed-form inertial solution are based upon all the pixels over all the runs.

The error form of (3.1.5) is created by making the following replacements:

$$\left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k}} \leftarrow [\boldsymbol{D1}]\left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k'}} \tag{3.3.2}$$

$$\left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k+1}} \leftarrow [\boldsymbol{D2}]\left[\boldsymbol{T}_{\boldsymbol{NED}}^{\boldsymbol{P}}\right]_{\boldsymbol{k+1'}} \tag{3.3.3}$$

$$\boldsymbol{\Delta R} \leftarrow \boldsymbol{\Delta R} + \boldsymbol{D_v \Delta t}, \tag{3.3.4}$$

where the random disturbance matrices $\boldsymbol{D1}$, $\boldsymbol{D2}$, and the disturbance vector $\boldsymbol{D_v}$ are given by,

$$[\boldsymbol{D1}] = \boldsymbol{I} - \frac{\boldsymbol{\pi}}{\boldsymbol{180}}[\boldsymbol{0.04r_1} \quad \boldsymbol{0.04r_2} \quad \boldsymbol{0.36r_3}]_{\boldsymbol{x}}, \tag{3.3.5}$$

$$[\boldsymbol{D2}] = \boldsymbol{I} - \frac{\boldsymbol{\pi}}{\boldsymbol{180}}[\boldsymbol{0.0028r_4} \quad \boldsymbol{0.0028r_5} \quad \boldsymbol{0.0028r_6}]_{\boldsymbol{x}}, \text{ and} \tag{3.3.6}$$

$$\boldsymbol{D_v} = \left(\boldsymbol{1}\frac{\boldsymbol{m}}{\boldsymbol{s}}\right)[\boldsymbol{r_7} \quad \boldsymbol{r_8} \quad \boldsymbol{r_9}]^{\boldsymbol{T}}. \tag{3.3.7}$$

The subscript "x" operator was defined in (3.2.3). The values r_1 through $\mathbf{r_9}$ are independent draws from a zero-mean, unity variance normal distribution. The numerical values above are based upon the specific inertial sensor and GPS error parameters discussed previously.

Comparison of Analytic and Optical Flow Methods

The simulation architecture developed above allows for a quantitative, statistical comparison between the closed-form inertial correspondence estimation algorithm and the classical optical flow algorithm using representative sensor error characteristics, ground imagery, and flight dynamics for a UAS. The closed-form inertial algorithm was developed in section 3.1. Optical flow algorithm code is available in the public domain. The optical flow code used in this chapter is a Matlab implementation of the Lucas-Kanade method written by Sohaib Khan. Two fundamental metrics are important for comparison. The first is accuracy and the second is the execution time. A Matlab implementation for the optical flow algorithm is selected to allow a direct execution time comparison between the optical flow and the closed-form inertial method which was also written in Matlab.

Figure 3-9 shows an overlay of the cumulative distribution of the error, $\boldsymbol{\varepsilon}$, from (3.3.1) for both the Lucas-Kanade and closed-form inertial methods. The evaluation in Figure 3-9 is performed at both point A and point B in the simulated aircraft trajectory as shown in Figure 3-8. These represent extreme conditions as point B features both angular motion (a yaw rate of approximately 30 degrees/s) as well as 20 m/s linear motion. It is more stressing than point A which features only linear motion. The output of the Lucas-Kanade algorithm for a pair of 100 Hz frames at point B is shown as a quiver plot in Figure 3-8 (each line shows the local flow vector direction and magnitude). Both the Lucas-Kanade as well as closed-form inertial estimates show

a marginally degraded performance in the presence of the larger motion as apparent in Figure 3-9 and Figure 3-10. In both cases, the estimated correspondence is based on two consecutive 100 Hz video frames. As discussed before, for the Lucas-Kanade algorithm, the cumulative distribution is of the error $\boldsymbol{\varepsilon}$ evaluated on each of the 640x480 image pixels. For the closed-form inertial estimate, the cumulative distribution is of the error $\boldsymbol{\varepsilon}$ evaluated on each of the 640x480 image pixels for each of the 50 Monte-Carlo runs.

From Figure 3-10, the closed-form inertial estimate consistently out-perform the Lucas-Kanade method. Additionally, within the Matlab environment, the run time of the closed-form inertial estimate is approximately 1/160 of the processing time of the Lucas-Kanade code.

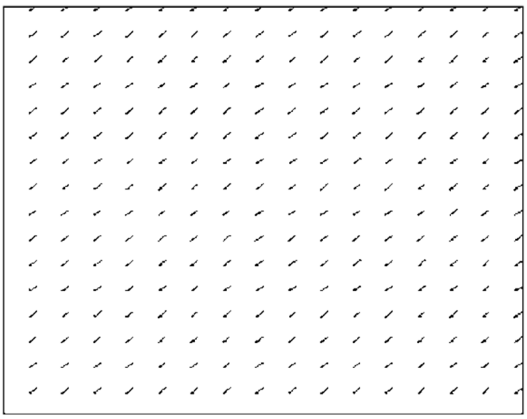


Figure 3-8: Quiver plot of optical flow vectors from the Lucas-Kanade algorithm for a pair of 100 Hz frames during the bank and turn maneuver at point B

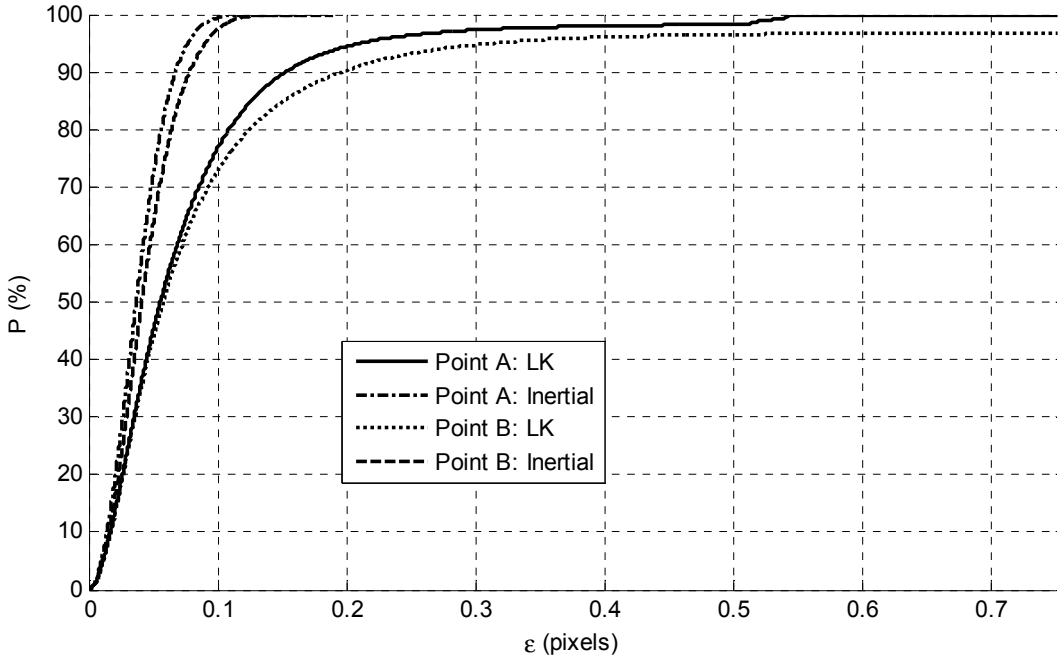


Figure 3-9: Overlay of the cumulative error distributions of the Lucas-Kanade (LK) and closed-form inertial correspondence algorithms

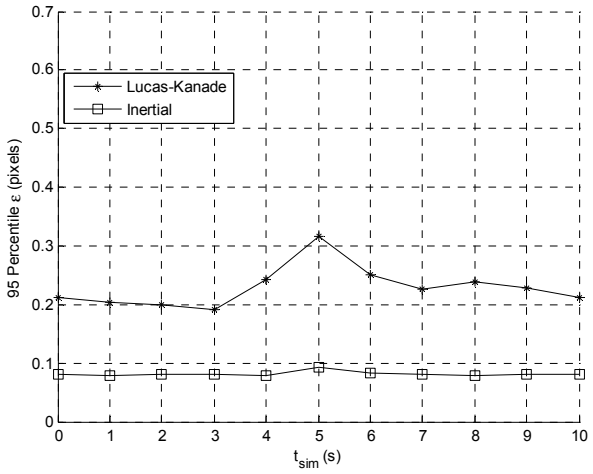


Figure 3-10: Comparison of 95 percentile correspondence errors for the Lucas-Kanade and closed-form inertial algorithms over the duration of the simulated flight

3.4 Conclusion

This chapter has introduced a computationally efficient means of calculating a dense correspondence vector field for a video sequence in airborne applications where an inertial navigation sensor is available. The method bypasses computationally expensive image processing methods of estimating the vector field and, instead, uses a closed-form solution to the geometric mapping from the inertial sensor measurements to the image. Furthermore, the chapter develops and analyzes an approach to the online estimation of the synchronization and misalignment between the inertial and image sensors. Accuracy of these parameters is required for making sub-pixel measurements of the correspondence vector field. Simulation based results show that a typical low-cost GPS/inertial sensor system is able to measure the correspondence an order of magnitude faster than a typical image-based optical flow code while achieving a significant improvement in accuracy.

CHAPTER 4

A SPATIAL FREQUENCY METRIC FOR MEASURING SUPER-RESOLUTION PERFORMANCE

In this chapter, we address a shortcoming in the current literature regarding SR in that there is no established metric that directly assesses an SR algorithm's ability to perform its principal objective of increasing resolution as defined in terms of spatial frequency response (SFR) by the ISO standard [49]. SR has the potential to allow engineers to specify lower resolution and, therefore, less expensive cameras for a given task by mathematically enhancing the camera's resolution. This is especially true for the resolution-critical, photogrammetric application class for captured imagery as summarized in 1.1. Performing each of these tasks requires a minimum image "sharpness" which is quantified by a maximum resolvable spatial frequency which is, in turn, a complex function of the camera optics, pixel sampling density, and signal-to-noise ratio (SNR).

SR image processing algorithms aim to increase the effective $\boldsymbol{M \; x \; N}$ pixel sampling density of a base camera such that it emulates the capabilities of an equivalent camera with a $\boldsymbol{kM \; x \; kN}$ pixel sampling density ($\boldsymbol{k}$ is a value greater than 1.). The exact meaning of capabilities in the preceding sentence is application dependent. In doing so, the SR algorithm improves image sharpness and, therefore, should improve the performance of any resolution-critical application.

In a typical engineering application, camera specifications, such as minimum pixel sampling density, will flow down as derived requirements from upper-level performance requirements. For example, the resolution requirements for a detection and classification task may be derived by the Johnson criteria or equivalent as discussed in 2.2. In this chapter, we examine the question of under what conditions can a requirement for a $\boldsymbol{kM \; x \; kN}$ pixel density camera be

satisfied by a less expensive $\boldsymbol{M\ x\ N}$ camera that is then up-sampled to $\boldsymbol{kM\ x\ kN}$ by an SR algorithm?

Much of the existing SR literature focuses on performance metrics for algorithms such as perceived image quality or peak signal to noise ratio SNR (PSNR). Other visual quality metrics are based on assessing characteristics of the processed output image that make it "better" from the perspective of human perception. In some cases, the objective of improving perceived quality can even be made at the expense of reconstruction fidelity [4]. Although appropriate for photographic or human visualization applications, loss of scene fidelity is undesirable for photogrammetric applications.

Unlike PSNR, however, the visual quality metrics do have the benefit that they may be made on any output image without the need to know the ground truth or the correct high resolution image. As such, a class of SR algorithms may also be derived based solely on optimizing these metrics [4]. Prevalent visual quality metrics include entropy based [105,106], edge contrast measure [106-108], and absolute mean brightness error [106,109].

PSNR and other image quality metrics can be misleading because most SR methods are simultaneously coupled with other enhancements such as de-blurring and/or de-noising. These latter enhancements can increase both the perceived image quality as well as PSNR without truly increasing the effective spatial frequency response of the camera. Alternate measures, such as the triangle orientation discrimination (TOD) method [110] which measures the ability of an observer to determine the orientation of a triangle from an image, are more directly traceable to the ability of the camera (and any associated image processing) to enable a resolution critical task. However, for the purpose of SR evaluation, these metrics are still undesirable because they measure broad

band spatial frequency characteristics of the enhanced image as opposed to the specific ability of SR to recover higher frequencies.

In this chapter, we propose a new, spatial frequency metric where the performance of a "black-box" SR algorithm is directly tied to the probability of successfully detecting critical spatial frequencies within the scene. Most importantly, the metric looks at detecting spatial frequencies aliased by the unenhanced camera. We show that the penalty of applying SR to an $\boldsymbol{M\ x\ N}$ pixel camera is an effective loss of SNR at higher frequencies relative to a true $\boldsymbol{kM\ x\ kN}$ camera and that this penalty is reflected in our proposed spatial frequency metric. We then use our metric to compare a set of standard SR algorithms on both simulated as well as real camera imagery.

The remainder of the chapter is organized as follows. In section 4.1, we discuss exiting methods of directly quantifying camera resolution. In section 4.2, we look at automation of those methods. In section 4.3, we extend the resolution measurement methods to introduce a new metric for measuring SR algorithmic performance based upon probability of successfully measuring high frequency content in the scene. In sections 4.4 through 4.6, we perform a set of experiments using both simulated and real imagery. In section 4.7, we provide our conclusions.

4.1 Existing Methods of Quantifying Camera Resolution

Once a critical spatial frequency, $\boldsymbol{\omega_c}$, requirement is established for a specific task, the Johnson criteria [41,42], as introduced in 2.2, loosely states that the camera's ability to perform the task may be reduced to the simpler, surrogate problem of detecting line-pairs in a standard chart such as the USAF 1951 resolution chart shown in Figure 4-1 (or others charts as mentioned in Appendix B). The resolution chart contains blocks of line pairs of decreasing size

(corresponding to higher spatial frequencies). Using the USAF 1951 chart, the resolution limit of the camera is defined as the smallest block for which it can not resolve the individual lines [45,50].

With modern, digital cameras, it is a common misconception that the spatial resolution is a function only of the camera's pixel sampling density. In truth, the digital camera's capability is determined by three dominant factors: pixel density, pre-sample modulation transfer function (MTF), and noise.

Figure 4-1: USAF 1951 resolution bar chart [50]

During the time Johnson published his results, the resolution of the camera would have been considered exclusively a function of the camera hardware (optics and electronics). For modern digital imaging systems, the final resolution, and, thus, the task performance capability, is the result of both the hardware and any associated signal processing, such as SR.

4.2 Automatic Measurement of Resolution from Bar Targets

The original work by Johnson was performed with the idea of a human observer. This introduces complicating factors, beyond the quality of the camera itself, which include quality, brightness, and magnification of the display device, MTF of the human eye, etc. [43]. In this section, we simplify the determination of camera resolution to include only the capabilities

intrinsic to the camera itself by casting the determination into an equivalent problem of machine detection of the critical frequency in a resolution bar chart.

Sample Camera (Tau-640) Model to Facilitate Discussion

To facilitate the discussion, we will introduce the specific characteristics of an example camera that will be used later in the paper. For the example, we use the Tau-640 long-wave infrared (LWIR) camera [111] which is representative of an inexpensive class of thermal imagers. The Tau-640 has 640 x 512 pixels, an aperture diameter, $\boldsymbol{D}$, of 6.4 cm, a $\boldsymbol{FOV}$ of 6.25 x 5.00 deg, and a spectral bandpass from 8μm to 12μm.

Even though images and their corresponding frequency spectra are 2-dimensional, it is often convenient for notational and conceptual clarity to consider effects in just a single dimension. We will use this simplification technique, when possible, throughout the paper. For the below discussion of the Tau-640 camera, we will consider characteristics only in the horizontal dimension. From (2.1.7) and the camera parameters listed above, the $\boldsymbol{IFOV}$ in the horizontal dimension is given by $\boldsymbol{IFOV}$ = (6.25 deg) / (640 pixels) = $9.766e^{-3}$ deg = 170 micro-radians.

Using (2.1.5) to approximate the diffraction-limited optical response, the spatial frequency response of the camera is shown in Figure 4-2 ($\boldsymbol{\lambda}$ is set to the center of the bandpass; i.e., 10μm). For visual clarity, throughout this chapter, frequency response curves will be shown on a semi-log plot and, unless otherwise specified, normalized to unity at DC. Figure 4-2 also shows the maximum frequency $\boldsymbol{\omega_{max}}$ = 2.9 cycles / milli-radian given by the pixel sampling density, the Nyquist-Shannon sampling theorem, and (2.1.9).

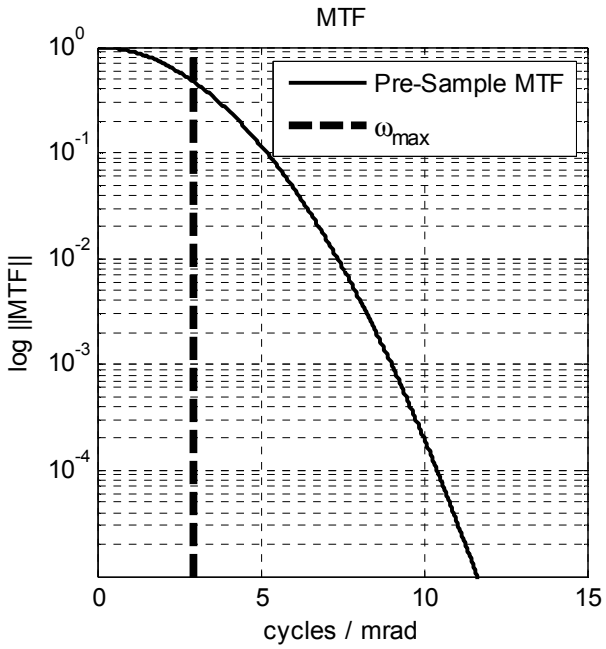


Figure 4-2: Theoretical, diffraction limited MTF of the Tau-640 (Gaussian Approximation)

It is readily apparent that any image processing task requiring detection of spatial frequencies above the 2.9 cycles / milli-radian $\boldsymbol{\omega_{max}}$ will mandate some form of SR to unroll these aliased frequencies. Typically, these SR algorithms, as summarized in 2.4, require multiple images of the scene with some change in the observation parameters, such as camera motion, between them in order to remove the ambiguity in aliased frequency components. However, it is also apparent from Figure 4-2 that any SR algorithm will have increasing difficulty recovering higher spatial frequencies due to signal attenuation. Information at spatial frequencies of approximately 11 cycles / milli-radian and higher is extinguished by the optical MTF by over 4 orders of magnitude. Although there are special cases where this high-frequency attenuation can be overcome by techniques such as lengthening the exposure time (e.g. astronomical imagery against a cold space background), these methods will typically be limited by the dynamic range of the camera and supporting electronics. That is, lower frequency signals present in the imagery will

push the camera into saturation before the higher frequency components become resolvable from the noise. Task requiring resolution beyond the practical MTF cutoff can not be accomplished with SR and will, instead, require an investment of a higher quality and, likely, more expensive camera.

Beating the MTF and Rayleigh Limit

It is noteworthy, at this point, to mention SR signal processing techniques described as achieving resolution beyond the limits of the optical MTF. These methods are sometimes referred to as "beating the Rayleigh limit." They appear in microscopy [112] where they can capitalize on the instrumentation's ability to manipulate the illumination source. For example, physical modulation of a scene by a spatially, sinusoidally varying illumination source will create sum and difference spatial frequencies. The differencing effect can shift higher frequencies to lower frequencies prior to being attenuated by the optical MTF. Another example is from astronomy [113,114] where the presence of a single or binary star may be determined from the resolution limited image. In general, these methods do not actually improve the MTF but rather find ways to better use the information that is captured within the existing MTF. This is possible because most all objects produce a broad spectral signature when imaged. That is, they contain characteristic spectral information at frequencies below the Rayleigh cutoff as well as above. For example, in the specific case of the binary star vs. single star detection, a naïve requirement flow down would suggest that, if the stars are separated by an angular displacement $\boldsymbol{\delta}$, the imaging system would require a minimum resolution of $\mathbf{2}/\boldsymbol{\delta}$. However, as there are only two, prior-known hypothesis, and the spectral signature of a point source is broad-band, a statistical inference separating the single star vs. two star case may be made using much more limited spectral information at frequencies much less than $\mathbf{2}/\boldsymbol{\delta}$.

The characteristic shown in Figure 4-2, where the optical cutoff frequency is high enough to permit some degree of aliasing is common in image system design where considerations of aliasing must be balanced with other system characteristics, such as $\boldsymbol{FOV}$ and sensitivity [59]. Indeed, for applications such as video forensics, images with a degree of aliasing are preferred over perfectly smooth images [115]. The presence of aliasing is often particularly true in infrared imaging systems where, in contrast to visible band sensors (0.39 – 0.75 μm), the construction of an FPA with smaller and more closely spaced detector elements is very difficult or may be prohibitively expensive due to fabrication complexity and quantum efficiency problems [2]. This fact makes infrared cameras good candidates for SR enhancement. In general, the above observation suggest that, if it is known that a particular camera will be used in an application where its output will always be subjected to SR signal processing, the optical design should be tailored to maximize as opposed to minimize aliasing.

The automatic detection of the frequency of a bar target is accomplished by identifying the peak of the Fourier spectrum of the bar image received by the camera. This concept is illustrated in Figure 4-3 and Figure 4-4. Figure 4-3 shows a simulated scene of a 2 line-pair (lp) / milli-radian bar target and its corresponding frequency spectrum. Because it is a bar target, there are spikes in the frequency spectrum at the base frequency as well as at all odd harmonics. Note, again, because the scene, $\boldsymbol{s(x,y)}$ in (2.1.1), is in analog space, the units of the image are in milli-radians as opposed to pixels. Figure 4-4 shows the simulated blurred and sampled version of the scene using the parameters of the 640 x 480 pixel density Tau-640 LWIR camera previously introduced.

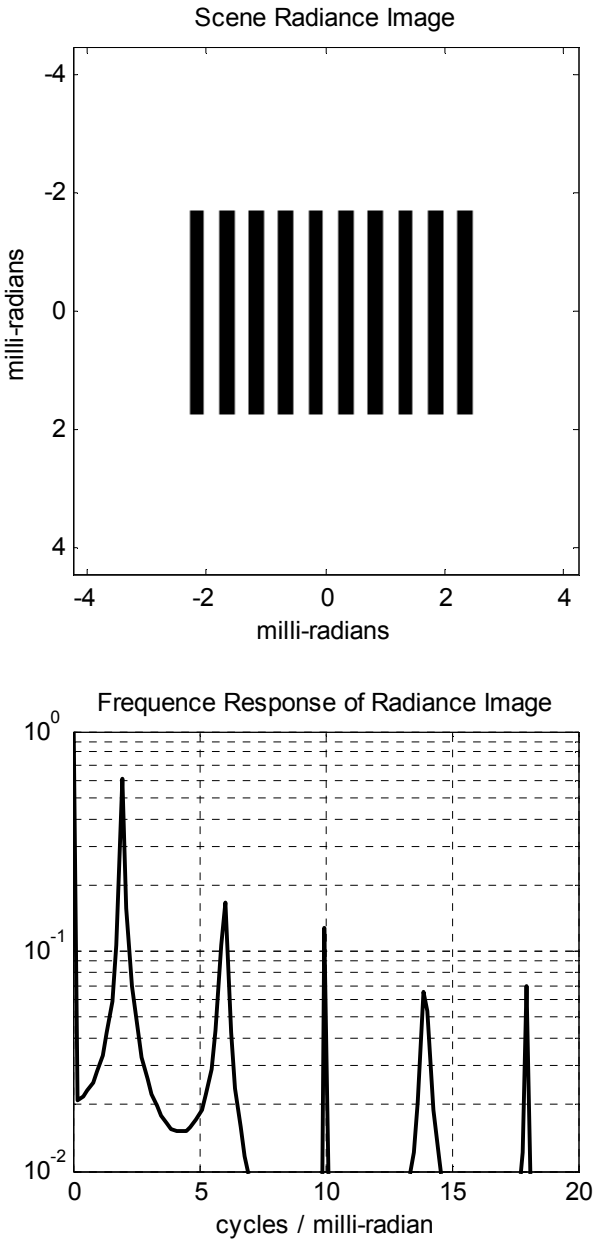


Figure 4-3: Simulated, external scene image containing bar target (top) and its corresponding continuous spatial frequency spectrum (bottom) for 2 lp/milli-radian bar target

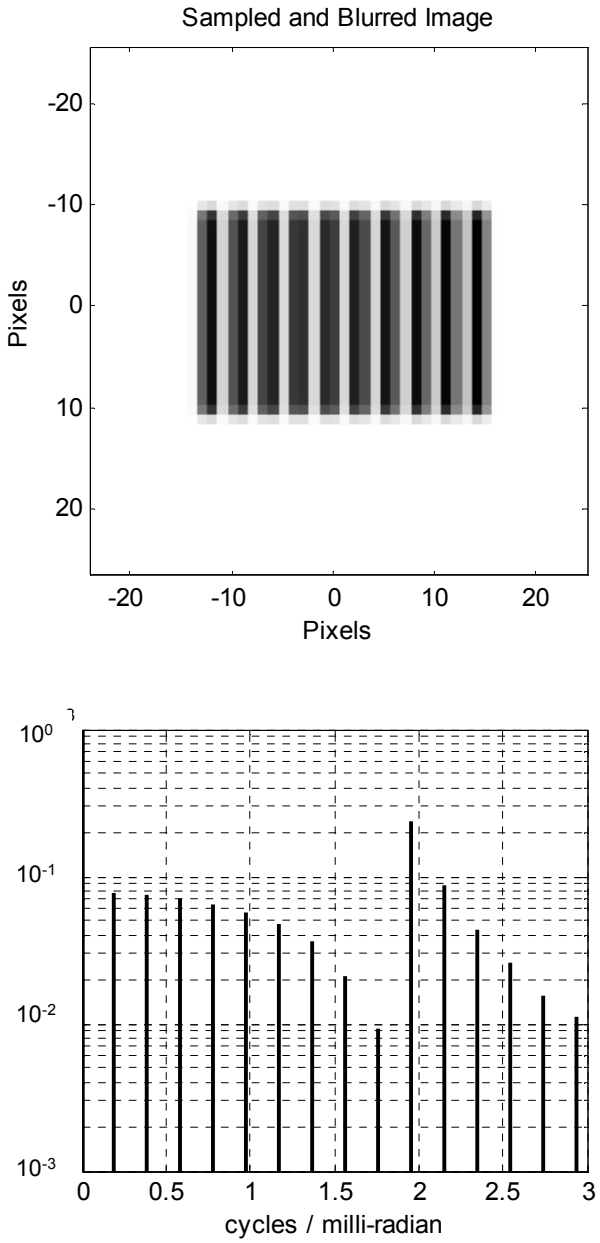


Figure 4-4: Simulated, sampled image (top) for 2 lp/milli-radian bar target and its discrete frequency spectrum (bottom) for the Tau-640 camera

As expected, because the first frequency mode of the bar target, at 2 cycles/milli-radian, is below the sampling cutoff of $\boldsymbol{\omega_{max}}$ = 2.9 cycles/milli-radian, there is no problem distinguishing the corresponding 2 cycles/milli-radian peak in the discrete Fourier spectrum (bottom of Figure 4-4). By identifying the peak in the discrete Fourier spectrum of the sampled image, a computer is able to generate a measurement of the frequency of the bar target. A "correct" measurement is one that matches the true frequency of the bar target. The probability of the computer measuring the correct frequency of the bar target, $\boldsymbol{P_m}$, is a function of the camera's frequency response as well as the image signal to noise ratio (SNR). The resolution of the camera, for a given SNR, is then the maximum bar target frequency for which the probability of making a correct measurement is above a threshold.

As a practical consideration, due to the fact that the DFT only exist at discrete frequencies, it is necessary to define an acceptance band when determining if the peak in the image DFT matches the known bar target frequency. Throughout this chapter, we utilize an acceptance band of 0.25 cycles/milli-radian. This effect may be seen in Figure 4-4 where the peak frequency is not exactly at 2.0 cycles/milli-radian.

Figure 4-5 and Figure 4-6 show the scene and image of a simulated 4 lp/milli-radian bar target. As the first frequency mode of 4 cycles/milli-radian is above $\boldsymbol{\omega_{max}}$, it is aliased to approximately 2 cycles/milli-radian (per the aliasing property given in (2.1.10)). Therefore, in this case, the computer would make an incorrect measurement of the true frequency of the bar target. In fact, because the discrete frequency spectrum is limited to $\boldsymbol{\omega_{max}}$, by definition, $\boldsymbol{P_m(\omega_{bar} > \omega_{max}) = 0}$ for any SNR.

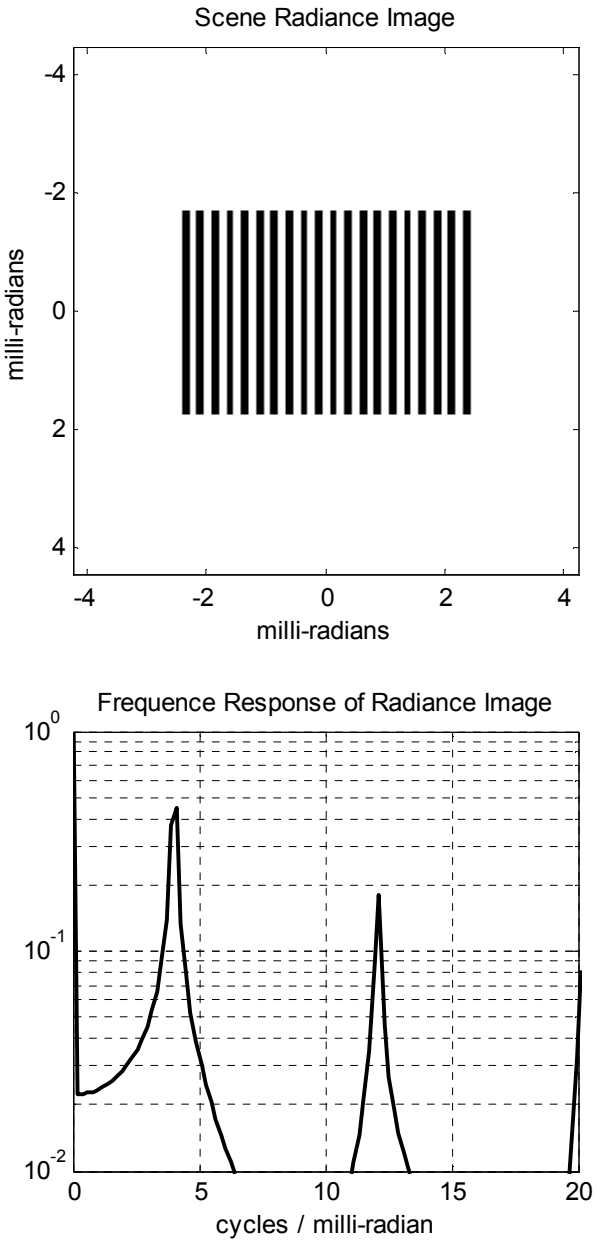


Figure 4-5: Simulated, external scene image containing a bar target (top) and its corresponding continuous spatial frequency spectrum (bottom) for 4 lp/milli-radian bar target

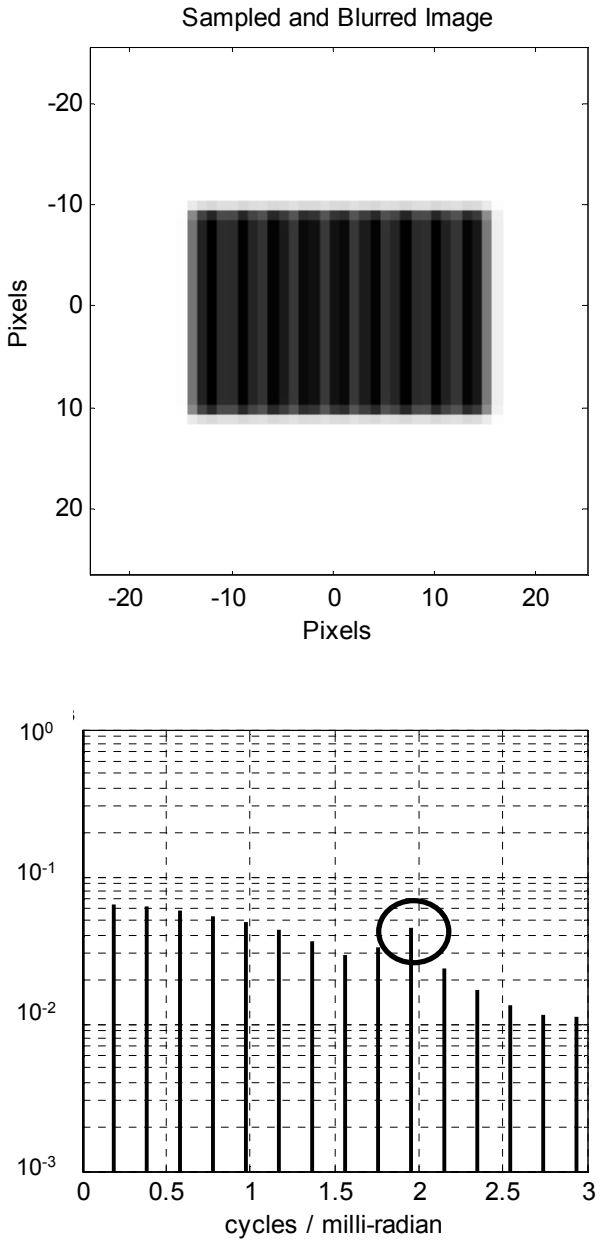


Figure 4-6: Simulated, sampled image (top) for 4 lp/milli-radian bar target and its discrete frequency spectrum (bottom) for the Tau-640 camera

4.3 Adapting Bar Target Measurement Probability Into a Metric for Super-Resolution

The problem with measuring the 4 lp/milli-radian bar target, discussed in the last section, would be resolved if we had a camera that was optically identical to the Tau-640 but had 2x the pixel density; i.e., 1280 x 960 pixels vs. 640 x 480 pixels. The sampled image and associated discrete spectrum of such a "Virtual-1280" camera is shown in Figure 4-7. As expected, because $\boldsymbol{\omega_{max}}$ = 4.8 cycles/milli-radian for the Virtual-1280 camera, a computer would have a non-zero probability of measuring the correct frequency of the 4 lp/milli-radian bar target (still dependent upon SNR).

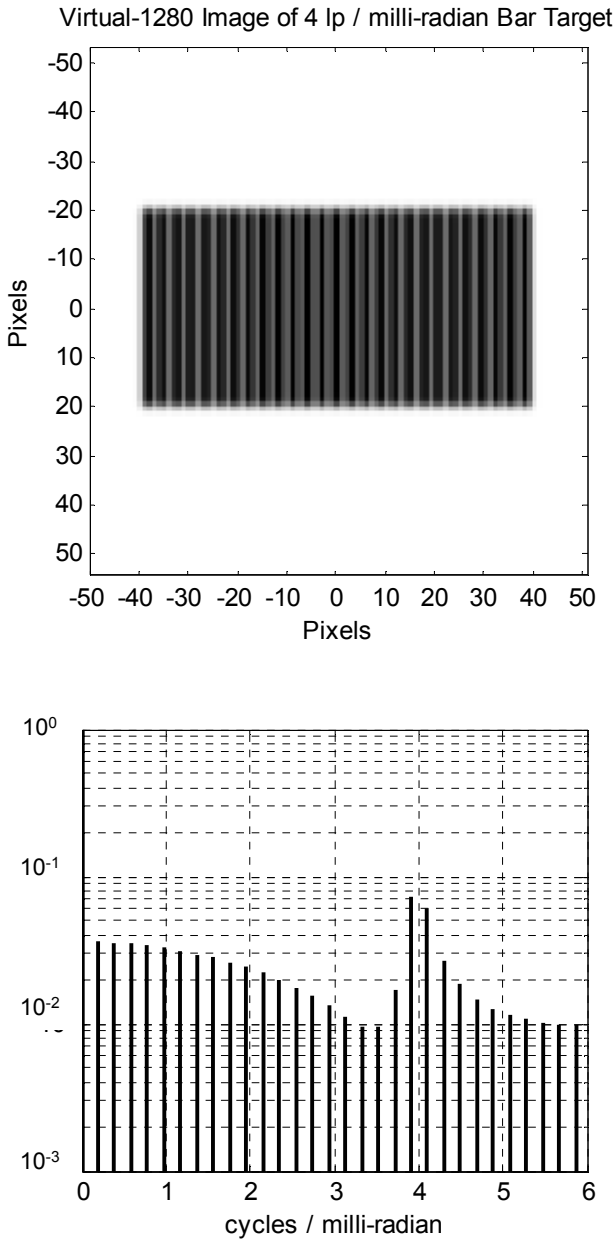


Figure 4-7: Simulated, sampled image (top) for 4 lp/milli-radian bar target and its discrete frequency spectrum (bottom) for the Virtual-1280 camera

Alternately, we should be able to get a similar increase in performance by mathematically increasing the effective pixel density of the Tau-640 using an effective SR algorithm.

Put back into general terms, an SR algorithm attempts to increase the pixel sampling density of a base $\boldsymbol{M\ x\ N}$ camera to that of a $\boldsymbol{kM\ x\ kN}$ camera ($\boldsymbol{k}$ is some value > 1). The effectiveness of the SR algorithm is determined by its ability to increase the measurement probability of bar targets, $\boldsymbol{P_m(\omega_{bar}, SNR)}$, from that of the base camera to that of a virtual $\boldsymbol{kM\ x\ kN}$ camera. We claim this metric is more relevant than traditional metrics, such as PSNR, for evaluating SR algorithm performance. Also, experimental measurement of the $\boldsymbol{P_m(\omega_{bar}, SNR)}$ metric is straightforward in that it does not require knowledge of the "ground truth" of the scene. It only requires the presence of a calibrated bar target chart in the scene.

Enhancements Due to De-blurring

Above, we made the claim that some existing metrics for resolution image enhancement were problematic because they would incorrectly credit image de-blurring along with true SR. Here we explain in more detail why de-blurring is not helpful for performance enhancement in terms of the SNR of image frequency components. If we receive a blurred image, $\boldsymbol{y(x,y)} + \boldsymbol{n(x,y)}$ (where $\boldsymbol{n(x,y)}$ is noise), we can represent it in the frequency domain as $\boldsymbol{Y(m)} + \boldsymbol{N(m)}$ (where $\boldsymbol{m}$ is the spatial frequency wave-number). Using the standard method, we then recover the un-blurred image spectrum $\boldsymbol{Z(m)}$ and, consequently, the un-blurred image $\boldsymbol{z(x,y)}$ as

$$\boldsymbol{Z(m) = R(m)[Y(m) + N(m)]}, \tag{4.2.1}$$

where $\boldsymbol{R(m)}$ is the recovery kernel which may be found either by direct inversion of the optical blur through use of a Wiener filter, or adapted from the image itself using one of several "blind deconvolution" methods such as in [116,117]. In any case, the pre-recovered SNR of each frequency component $\boldsymbol{m}$ is $\boldsymbol{Y(m)/\ N(m)}$. After applying the blur recovery of (4.2.1), the SNR

will be $\boldsymbol{R(m)Y(m)/\,R(m)N(m) = Y(m)/\,N(m)}$. In other words, even though de-blurring produces a perceived improvement in image quality and sharpness, the SNR of each frequency component and, hence, its utility for classification purposes is not enhanced by de-blurring. It will, however, show up as an improvement to a metric such as PSNR.

4.4 Evaluation (Noise-Free Case)

In order to evaluate the proposed, spatial frequency metric for SR algorithms introduced in the previous section, we begin testing using simulated resolution bar targets along with simulations of the Tau-640 LWIR camera. We will simulate imaging bar targets of spatial frequency 1, 2, 3, and 4 lp / milli-radian at varying SNR levels and compare $\boldsymbol{P_m(\omega_{bar}, SNR)}$ of the super-resolved imagery to that of both the base camera as well as the Virtual-1280 camera discussed above. For comparison, we utilize four spatial, SR techniques which are available on-line and run within a Matlab environment [115]. The first, used as a control, is simple BiCubic up-sampling. The BiCubic upsampling utilizes only a single LR image as input. The remaining three techniques all use a Variational Bayesian Inference (VBI) approach with different image prior models. The three priors were the Total Variation (TV) prior [116], the Simultaneous Autoregressive (SAR) prior [117], and the L1-Norm prior [117]. In all of these trials, we are attempting to super-resolve the image by a factor of 2x; i.e. increase from a pixel density of 640 x 480 to a pixel density of 1280 x 960. In order to do so, the SR algorithms are provided with two input images, shifted horizontally by half of the base camera's $IFOV$ (half a pixel).

Figure 4-8 shows, for the noise free case, the 2x super-resolved image and discrete spectrum for the simulated 4 lp/milli-radian bar target shown in Figure 4-6. The evaluation is carried out for each of the four SR techniques listed above using two images with a relative

horizontal shift of half a pixel. The resulting super-resolved images and their corresponding discrete frequency spectrums are shown in Figure 4-8. All three SR algorithms were able to unroll the aliased 4 cycles / milli-radian frequency component such that a computer could measure it whereas the BiCubic algorithm was not. The BiCubic algorithm still shows the dominant peak at the aliased 2 cycles/milli-radian frequency as does the base camera.

In general, multiple parameters can affect the performance of the SR algorithms. These include the total number of input video frames, the translational/rotational shift between frames, as well as the selection of various control constants contained within the algorithm implementations. In order to reduce the number of variations and focus on the utility of the P_m metric, we have chosen to uniformly run each SR algorithm in the simplest possible manner. Based on the results of Figure 4-8, two input frames is sufficient to enable each algorithm to achieve its SR capability in the horizontal direction. Consequently, in this work, we will show all SR results using two input frames, a best-case 0.5 pixel translational shift, no rotational shift, and the default constants as contained in the released implementation of the algorithm. We also did not enable the simultaneous SR and blur removal option for these algorithms.

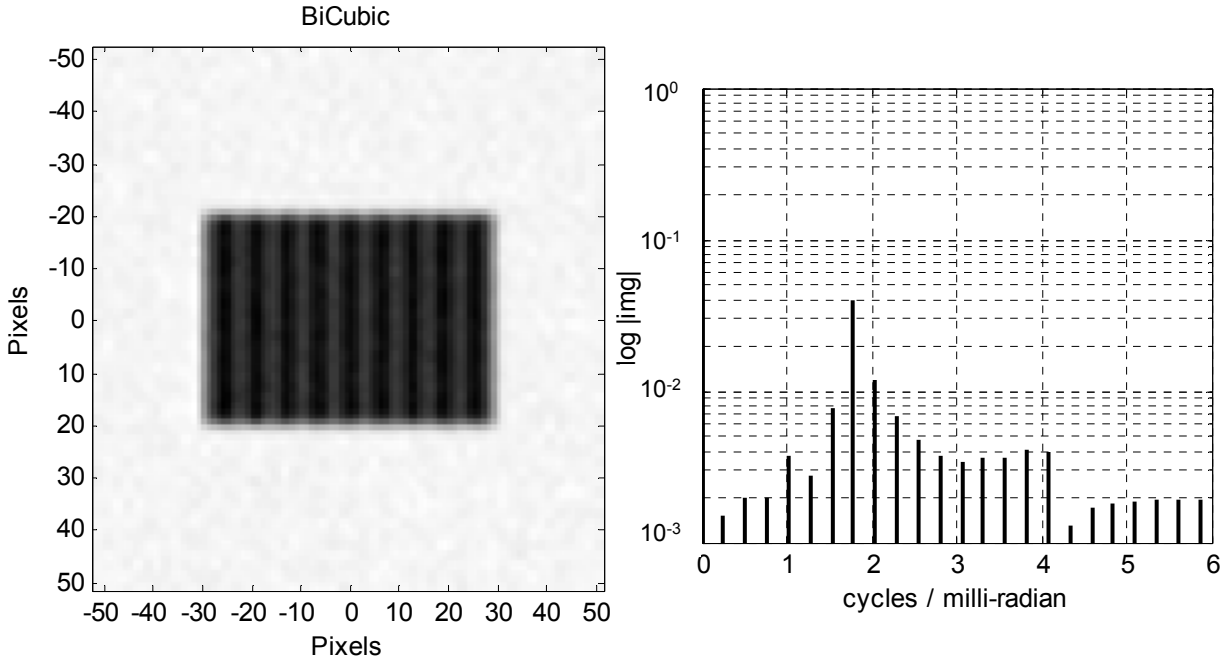
Relating the observation back to the Johnson criteria, the results of Figure 4-8 mean that increased pixel density imagery resulting from the three SR algorithms would enable a task with a critical frequency, $\boldsymbol{\omega_c}$, of 4 cycles/milli-radian while the BiCubic algorithm would not. Therefore, at least in the noise-free case, an engineer would be able to specify a less expensive 640x480 camera plus one of the three tested SR algorithms to perform a task for which the initial requirements flow down would have mandated a more expensive, higher pixel density camera.

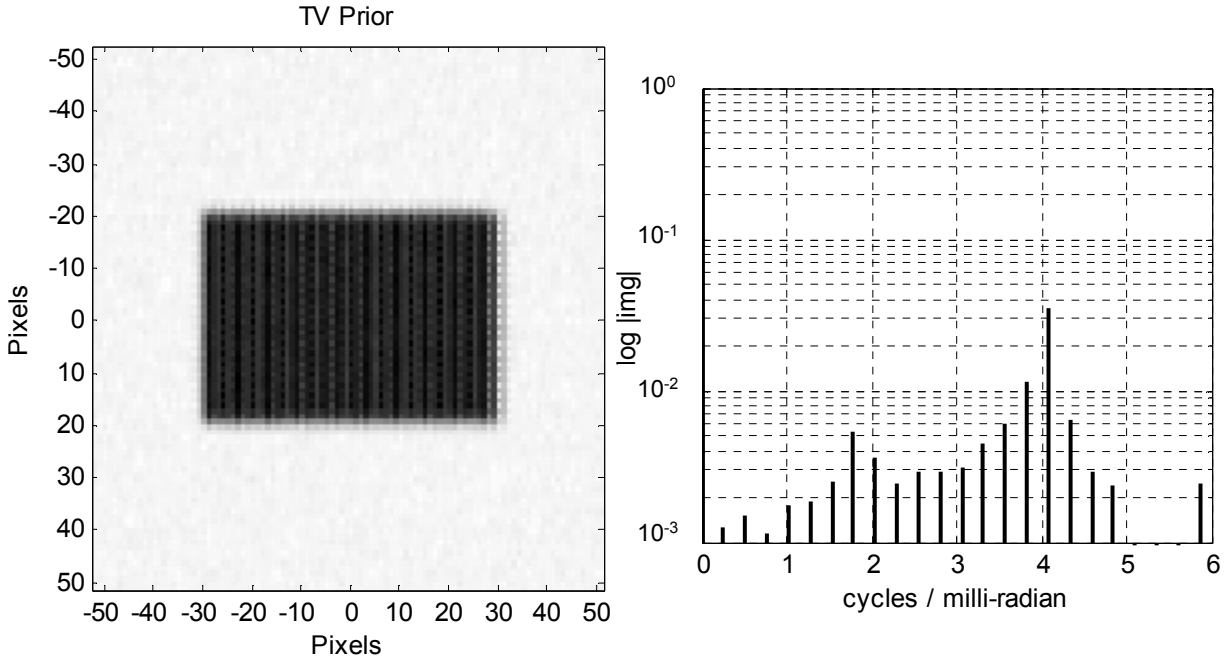
The corresponding PSNR values for this test were BiCubic = 16.8 dB, TV Prior = 18.3 dB, L1 Prior = 18.5 dB, SAR Prior = 18.0 dB. From the PSNR numbers, although slightly less, it is not clear that the BiCubic algorithm is fundamentally ineffective whereas the other three SR algorithms are effective. PSNR, for these cases, is defined as

$$\boldsymbol{PSNR = 20log_{10}\frac{max(ref)}{\sqrt{MSE(ref,img)}}} \text{ , and} \tag{4.4.1}$$

$$\boldsymbol{MSE(ref,img) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\big(ref(u,v) - img(u,v)\big)^2,} \tag{4.4.2}$$

where $\boldsymbol{ref}$ and $\boldsymbol{img}$ refer to the reference and super-resolved images respectively. For evaluation, the reference image is the output of the Virtual-1280 camera, shown in Figure 4-9. The evaluation is also only performed within the region of interest (ROI) shown in the figure. Prior to comparison, the super-resolved image is scaled to have the same maximum pixel value as the reference in order to eliminate any scaling changes that occurred during the SR process.


BiCubic
Pixels
Pixels
log |img|
cycles / milli-radian


TV Prior
Pixels
Pixels
log |img|
cycles / milli-radian

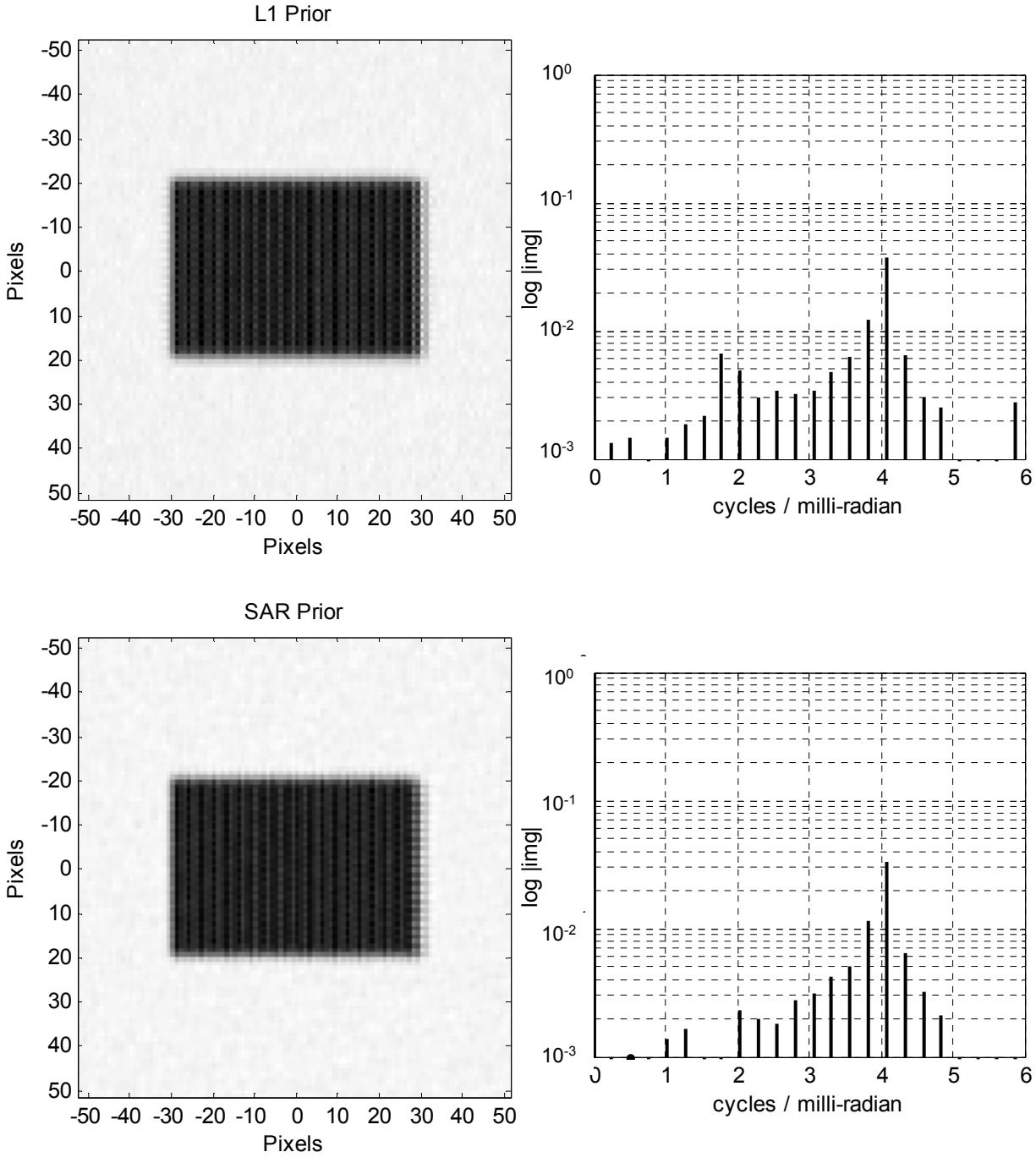


Figure 4-8: Noise-free evaluation of SR algorithms on simulated 4 lp/milli-radian bar target. From top to bottom, the SR techniques are BiCubic, TV Prior, L1 Prior, and SAR Prior

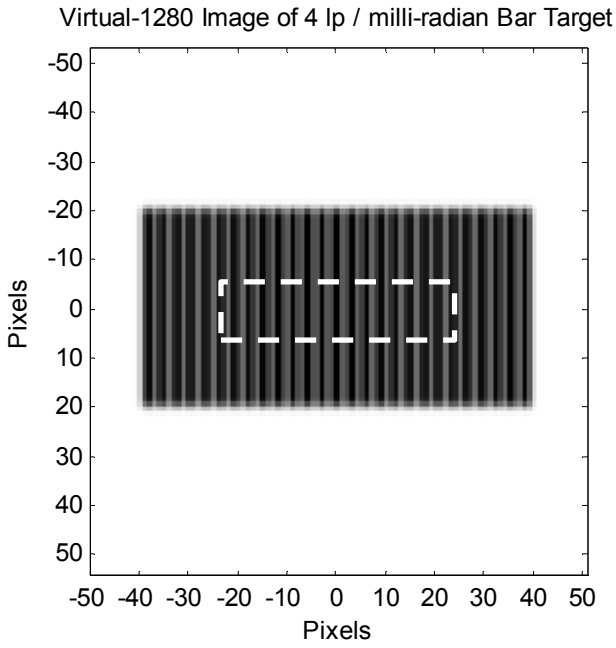


Figure 4-9: Reference image and ROI (dotted box) used for PSNR calculation

It is, of course, noteworthy that the SR images shown in Figure 4-8 are not as visually pristine as the Virtual-1280 camera image as shown in Figure 4-7 and Figure 4-9 (again, the goal of the SR algorithms is to emulate the Virtual-1280 camera). Part of this difference is attributed to the simplified method we used to run the algorithms as described at the beginning of the section. We would be able to further improve the visual quality of the intensity images in Figure 4-8 by any or a combination of: using greater than 2 LR frames with vertical as well as horizontal shifts, enabling the SR algorithm's option to simultaneously remove blur, and/or applying standard non-SR image enhancement techniques such as local contrast improvement and edge sharpening [4]. However, even without these enhancements, frequency analysis of the image output from the algorithm showed that it was sufficient to achieve the objective of exposing the high frequency content in the horizontal direction.

4.5 Introduction of Noise

Without consideration of camera noise, the results of the last section would suggest that use of the three tested SR algorithms is, indeed, equivalent to physically increasing the pixel density of the camera. However, we must also evaluate how the SR algorithms perform relative to the virtual camera over SNR.

Theoretical Treatment of the Effect of SR on SNR

Before looking at numerical results, we turn back to the frequency domain perspective of SR (section 2.4) to predict the effect of SR on SNR. The fundamental frequency domain perspective of SR comes, again, from considering the Fourier transform $\boldsymbol{X(\omega_x, \omega_y)}$ of the analog, (pre-sampled) image (see 2.1). A spatial translation of $\boldsymbol{(\delta_x, \delta_y)}$ produces an image spectrum

$$\boldsymbol{X'(\omega_x, \omega_y) = exp[j\omega(\delta_x\omega_x + \delta_y\omega_y)]X(\omega_x, \omega_y).} \tag{4.5.1}$$

The discrete Fourier transform of the digital image $\boldsymbol{I(m, n)}$, where $\boldsymbol{(m, n)}$ are discrete pixel indices, is related to the analog spectrum $\boldsymbol{X(\omega_x, \omega_y)}$ by the aliasing property

$$\boldsymbol{I\left(\frac{r_x}{M}\omega_s, \frac{r_y}{N}\omega_s\right) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} X\left(\frac{r_x}{M}\omega_s - k\omega_s, \frac{r_y}{N}\omega_s - l\omega_s\right),} \tag{4.5.2}$$

where $\boldsymbol{M}$ and $\boldsymbol{N}$ are are the width and height of the discrete spectrum. The discrete Fourier transform only exist at integer values of the indicies $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$ which range between $\boldsymbol{\pm M/2}$ and $\boldsymbol{\pm N/2}$ respectively ($\boldsymbol{M}$ and $\boldsymbol{N}$ even). Each element, therefore, of the discrete transform is a linear superposition of a base analog frequency components between $\boldsymbol{\pm\omega_s/2}$ and all of the aliased analog frequency components displaced by integer multiples of $\boldsymbol{\omega_s}$. When the shift operator,

$\boldsymbol{exp}\left[\boldsymbol{j\omega}\left(\boldsymbol{\delta_x\omega_x + \delta_y\omega_y}\right)\right]$, is applied in the analog domain, each of the aliased frequency components is given a different phase shift than the base frequency component. If multiple images are available with different translational shifts, their discrete transforms lead to a system of linearly independent equations which can be solved to unroll the aliased frequencies [45].

Equations (4.5.1) and (4.5.2) also illustrate why spatial frequencies beyond the optical cutoff can not be recovered by SR algorithms. Any component of the scene that is extinguished by $\boldsymbol{H(\omega)}$ will still be unrecovered independent of the application of the shift operator. From an information perspective, optical blur represents a true loss of information whereas aliasing is only a scrambling of information. SR algorithms can only unscramble the existing information.

Along with this concept, it is instructive to reconsider the rectangular pixel integration term, $\frac{\boldsymbol{1}}{\boldsymbol{ab}}\boldsymbol{rect}\left(\frac{\boldsymbol{x}}{\boldsymbol{a}},\frac{\boldsymbol{y}}{\boldsymbol{b}}\right)$, from equation (2.1.1). In the common case of an imager with 100% fill-factor; i.e. the pixel size is equal to the pixel spacing or $\boldsymbol{a = \Delta x, b = \Delta y}$, its frequency spectrum is a $\boldsymbol{sinc}$ function as shown in Figure 4-10. Again, for notational and conceptual clarity, we will consider effects in just a single dimension. At frequencies below $\boldsymbol{\omega_s}$, it has the effect of adding additional attenuation. However, the gain drops to 0 for frequencies at near integer multiples of $\boldsymbol{\omega_s}$, meaning that there is no information transferred to the digital image at these frequencies. These regions of unrecoverable information loss are a factor when performing SR with a pixel density increase greater than a factor of 2x.

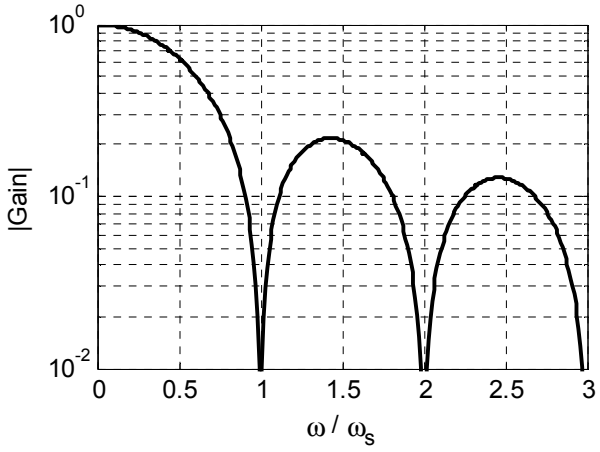


Figure 4-10: Normalized frequency response of the rectangular pixel integration term from equation (2.1.1)

Given the model expressed in equations (4.5.1) and (4.5.2), we can obtain a prediction of the SNR penalty associated with SR. To simplify, we will limit the problem to that of performing an SR magnification of 2x. That is, given a, low-resolution (LR), N element pixel array with pixel $\boldsymbol{IFOV = 1/\omega_s}$, we will attempt to recover the signal that would be observed by a virtual, high-resolution (HR) $\boldsymbol{2N}$ element pixel array with a smaller pixel $\boldsymbol{IFOV = 1/2\omega_s}$. Furthermore, we will assume that the combination of the scene and the optics band limit the signal reaching the image plane such that there is no aliasing in the HR image. Under these conditions, the discrete Fourier transform of the HR signal is given by

$$\boldsymbol{I_{HR}\left(\omega = \frac{r}{N}\omega_s\right) = H(\omega)sinc\left(\frac{\omega}{2\omega_s}\right)S(\omega), -N \le r \le N,} \tag{4.5.3}$$

where $\boldsymbol{H(\omega)}$ and $\boldsymbol{S(\omega)}$ are the discrete Fourier transforms of the optical blur and scene, respectively (see model in equation (2.1.1)). By the aliasing property in (4.5.3), the corresponding spectrum of the LR image is given by

$$\boldsymbol{I_{LR}\left(\omega = \frac{r}{N}\omega_s\right) =}$$

$$\boldsymbol{H(\omega)sinc\left(\frac{\omega}{\omega_s}\right)S(\omega)}$$

$$\boldsymbol{+H(\omega-\omega_s)sinc\left(\frac{\omega}{\omega_s}-1\right)S(\omega-\omega_s)} \tag{4.5.4}$$

$$\boldsymbol{,0 \le r \le \frac{N}{2}.}$$

Note that equation (4.5.4) is only valid for the positive half of the LR spectrum where $\boldsymbol{0 \le r \le \frac{N}{2}}$. As we are working with all real values signals, the negative half of the spectrum is redundant in that $\boldsymbol{I_{LR}(-\omega)}$ is the complex conjugate of $\boldsymbol{I_{LR}(\omega)}$.

Additionally, if we capture a second image translated by a displacement, $\boldsymbol{\delta}$, the transform of the shifted LR image is given by

$$\boldsymbol{I'_{LR}\left(\omega = \frac{r}{N}\omega_s\right) = exp(2\pi j\omega\delta)H(\omega)sinc\left(\frac{\omega}{\omega_s}\right)S(\omega) +}$$
$$\boldsymbol{exp(2\pi j(\omega-\omega_s)\delta)H(\omega-\omega_s)sinc\left(\frac{\omega}{\omega_s}-1\right)S(\omega-\omega_s).} \tag{4.5.5}$$

For compactness, we define a gain term, $\boldsymbol{G(\omega)}$, to reflect the relative signal loss between the LR and HR signals due to the pixel integration term. That is

$$\boldsymbol{G(\omega) = \frac{sinc\left(\frac{\omega}{\omega_s}\right)}{sinc\left(\frac{\omega}{2\omega_s}\right)} = cos\left(\frac{\pi\omega}{2\omega_s}\right).} \tag{4.5.6}$$

Combining equations (4.5.3), (4.5.4), (4.5.5), and (4.5.6), we get the following linear equation relating the transform of the pair of LR signals to that of the HR signal

$$\begin{pmatrix} \boldsymbol{I_{LR}(\omega)} \\ \boldsymbol{I'_{LR}(\omega)} \end{pmatrix} =$$

$$\begin{bmatrix} \boldsymbol{G(\omega)} & \boldsymbol{G(\omega-\omega_s)} \\ \boldsymbol{exp(2\pi j\omega\delta)G(\omega)} & \boldsymbol{exp(2\pi j(\omega-\omega_s)\delta)G(\omega-\omega_s)} \end{bmatrix} \tag{4.5.7}$$

$$\begin{pmatrix} \boldsymbol{I_{HR}(\omega)} \\ \boldsymbol{I_{HR}(\omega-\omega_s)} \end{pmatrix}.$$

Equation (4.5.7) may be solved to unroll the aliased frequency components in the HR signal by

$$\begin{pmatrix} \boldsymbol{I_{HR}(\omega)} \\ \boldsymbol{I_{HR}(\omega-\omega_s)} \end{pmatrix} = \frac{\boldsymbol{1}}{\boldsymbol{exp(-2\pi j\omega_s\delta)-1}}$$

$$\begin{bmatrix} \frac{\boldsymbol{exp(-2\pi j\omega_s\delta)}}{\boldsymbol{G(\omega)}} & -\frac{\boldsymbol{exp(-2\pi j\omega\delta)}}{\boldsymbol{G(\omega)}} \\ -\frac{\boldsymbol{1}}{\boldsymbol{G(\omega-\omega_s)}} & \frac{\boldsymbol{exp(-2\pi j\omega\delta)}}{\boldsymbol{G(\omega-\omega_s)}} \end{bmatrix} \begin{pmatrix} \boldsymbol{I_{LR}(\omega)} \\ \boldsymbol{I'_{LR}(\omega)} \end{pmatrix}. \tag{4.5.8}$$

Therefore, in the absence of noise and perfect knowledge of $\boldsymbol{\delta}$, the HR frequency components may be recovered exactly. If the LR signal is subject to broadband noise of magnitude, $\boldsymbol{\sigma_{LR}}$, then the noise amplification in the recovered frequency components is given by

$$\boldsymbol{\sigma^2_{HR-SR}(\omega)} = \frac{\boldsymbol{|\exp(-2\pi j\omega_s\delta)|^2+|\exp(-2\pi j\omega\delta)|^2}}{\boldsymbol{G(\omega)^2|\exp(-2\pi j\omega_s\delta)-1|^2}}\boldsymbol{\sigma^2_{LR}}, \text{ and} \tag{4.5.9}$$

$$\boldsymbol{\sigma^2_{HR-SR}(\omega-\omega_s)} = \frac{\boldsymbol{1+|exp(-2\pi j\omega\delta)|^2}}{\boldsymbol{G(\omega-\omega_s)^2|exp(-2\pi j\omega_s\delta)-1|^2}}\boldsymbol{\sigma^2_{LR}}, \tag{4.5.10}$$

where the subscript "$\boldsymbol{HR-SR}$" indicates the high resolution image based upon SR processing. Simplifying, we get

$$\boldsymbol{\sigma^2_{HR-SR}(\omega)} = \frac{\boldsymbol{1}}{\boldsymbol{G(\omega)^2\left(1-cos(-2\pi\omega_s\delta)\right)^2}}\boldsymbol{\sigma^2_{LR}}, \tag{4.5.11}$$

which is valid at discrete samples ranging from $\boldsymbol{-\omega_s \leq \omega \leq \omega_s}$. Based on (4.5.11), we can define a loss function as the ratio of the SNR of the recovered HR frequency components to the SNR of

those same components as would be collected by the virtual HR imager. As we would be able to perform (4.5.8) on both $\boldsymbol{\omega}$ and - $\boldsymbol{\omega}$, as we know the two results form a complex conjugate pair, we may further reduce the variance of the recovered HR frequencies, from (4.5.11), by a factor of ½ by averaging. We define an aggregate loss function for the SR process as

$$\boldsymbol{L(\omega) = \frac{SNR_{HR-SR}}{SNR_{HR}}.} \tag{4.5.12}$$

In order to provide an expression for $\boldsymbol{L(\omega)}$, we have to make the explicit assumption that the raw noise characteristics of the virtual HR camera are identical to those of the LR camera; i.e. $\boldsymbol{\sigma_{HR} = \sigma_{LR}}$**.** As a practical consideration, if we were to actually fabricate a real HR camera, various design compromises would likely invalidate this equality assumption. For example, if the physical size of each individual pixel were reduced in order to fit the larger number of pixels on the same FPA substrate, the sensitivity of each pixel would be reduced resulting in $\boldsymbol{\sigma_{HR} > \sigma_{LR}}$. Other factors affecting $\boldsymbol{\sigma_{HR}}$ for a real camera would be integration time and electronic readout characteristics. However, both because we lack specific design modification details for a real HR camera and because we are interested in SNR affects due to the SR process, we will proceed with the assumption of equivalent noise for the virtual HR camera. Given that assumption, and substituting (4.5.11) and equation (4.5.12), we can write

$$\boldsymbol{L(\omega) = \sqrt{\frac{2\sigma_{LR}^2(\omega)}{\sigma_{HR-SR}^2(\omega)}} = \sqrt{2}G(\omega)\big(1 - cos(-2\pi\omega_s\delta)\big).} \tag{4.5.13}$$

Our final loss function, $\boldsymbol{L(\omega)}$, is shown in Figure 4-11 for translation displacements between the two LR signals of 0.50, 0.25, and 0.10 pixels.

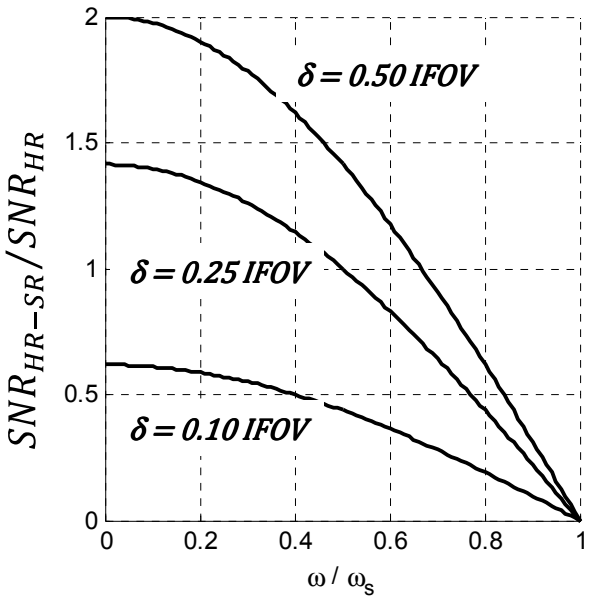


Figure 4-11: Predicted SNR loss over normalized frequency due to SR

As expected, Figure 4-11 shows that the ability to recover the HR spectrum, from an SNR perspective, improves with larger frame to frame sub-pixel translation, with the maximum at 0.50 pixels, or $\boldsymbol{\delta = 0.50\,IFOV}$. Also, even though (4.5.9) shows that we are able to unambiguously unroll aliased frequencies over the entire spectrum, Figure 4-11 shows that, for higher frequencies, we pay an increasing penalty in SNR. Note that, for lower frequencies, with large displacement, we actually get a boost in SNR as the fact that we have two measurements dominates the increase in noise. Consequently, because the optics are fixed, the super-resolved LR camera, will require a higher overall scene SNR than the virtual HR camera to meet the same application performance requirements. Because the results in this section assume the displacement, $\boldsymbol{\delta}$, is known perfectly, Figure 4-11 is an optimistic result. The inevitable uncertainty in the exact value of $\boldsymbol{\delta}$ for any real implementation will lead to an even lower SNR for the super-resolved image.

For some application, with plenty of signal margin, the impact of the SNR penalty may be very low. However, many applications that are attempting to push the capabilities of the camera to its limit, such as remote detection and classification, will be stressing both from a resolution as well as SNR perspective. For these later applications, the reduction in SNR due to SR must be carefully taken into account when establishing the practicality of using SR in lieu of a more expensive, higher-resolution camera.

Numerical Performance Assessment of SR Algorithms

Due to the iterative and non-linear nature of the SR algorithms, we do not attempt to generate a theoretical derivation of the noise propagation during the estimation process but rather employ a Monte-Carlo approach. Using a total of 32 runs for each point in a discrete set of $(\boldsymbol{\omega_{bar}}, \boldsymbol{SNR})$ combinations, we are able to measure $\boldsymbol{P_m}(\boldsymbol{\omega_{bar}}, \boldsymbol{SNR})$. Assuming that the primary contributor to noise in the camera is in the pixel readout electronics as well as photon shot noise (which is proportional to the mean signal level [118]), we model it as broadband Gaussian with constant amplitude across the spectrum. For each trial, the SNR defines the 1-sigma value for a Gaussian noise distribution relative to the peak amplitude of the image spectrum. For the cyclic bar target images, the peak amplitude will occur at $\boldsymbol{\omega} = 0$ and be equal to $\boldsymbol{P/2}$ where $\boldsymbol{P}$ is the peak pixel value of the image relative to the background. The SNR is related to the pixel noise by

$$\boldsymbol{SNR} = \frac{\boldsymbol{P}}{\boldsymbol{\sigma_{pix}}} \frac{\sqrt{\boldsymbol{N}}}{\boldsymbol{2}}, \tag{4.5.14}$$

where $\boldsymbol{\sigma_{pix}}$ is the 1-sigma noise level of each pixel in the image and $\boldsymbol{N}$ is the total number of pixels used to compute the Fourier transform. Equation (4.5.14) accounts for the fact that the Fourier

transform operation averages out noise. If the evaluation is performed over a fixed spatial rectangle of angular dimension $\theta_x \; x \; \theta_y$, then

$$N = \frac{\theta_x \theta_y}{ifov^2}. \tag{4.5.15}$$

In our mechanization of adding noise, we use (4.5.14) and (4.5.15) to compute the value of σ_{pix}. Then, after generating the noise-free LR images as in section 4.4, we add a Gaussian pseudorandom value based on a distribution with standard deviation equal to σ_{pix} to each individual pixel.

Note, the $P/2$ term in (4.5.14) applies only to the special case of bar targets. In order to get the equivalent SNR for a general 2D object, the calculation must be revised to

$$SNR = \left(\bar{I}_{fore} - \bar{I}_{bkg}\right) \frac{\sqrt{N}}{\sigma_{pix}}, \tag{4.5.16}$$

where $\bar{I}_{fore}$ and $\bar{I}_{bkg}$ represent the mean foreground and local background intensities of the object respectively.

For comparison, we include the performance of the base Tau-640 camera, the Virtual-1280 camera, the base camera up-sampled with the BiCubic method, and the base camera enhanced with the three SR methods (TV Prior, L1 Prior, and SAR Prior). For the 1 lp/milli-radian and 2 lp/milli-radian bar targets, all cases provide a P_m = 100% for SNR values as low as 1.5. This trivial result illustrates that the SR algorithms adhere to the important property that they, in no cases, degrade the original base camera image. That is, $P_m(\omega_{bar}, SNR)$ after super-resolving should always be greater than or equal to $P_m(\omega_{bar}, SNR)$ for the base camera.

The results for the 3 lp/milli-radian and 4 lp/mili-radian cases are shown in Figure 4-12. For the 3 lp/milli-radian case, which is just slightly above ω_{max} = 2.9 cycles/milli-radian of the

base camera, the virtual-1280 camera is able to provide $\boldsymbol{P_m}$ = 100% for SNR values as low as 1.5. The base camera, however, gradually loses performance below an SNR of 4.5. The SR enhancements largely follow the same trend as the base camera with the algorithm using the SAR prior being the only one to consistently outperform the base camera. The BiCubic up-sampling clearly degrades the frequency information as it is unable to reliably measure the 3 lp/milli-radian frequency even with SNR as high as 10.

For the 4 lp/milli-radian case, which is well above the $\boldsymbol{\omega_{max}}$ = 2.9 cycles/milli-radian of the base camera, the base camera performance is, by definition, $\boldsymbol{P_m}$ = 0% because the critical frequency is beyond the spectrum of the discrete Fourier transform. Similarly, $\boldsymbol{P_m}$ = 0% for the BiCubic up-sampling method as it does not unroll any of the aliased frequencies. The Virtual-1280 camera achieves $\boldsymbol{P_m}$ = 100% at an SNR of 5.5 and beyond but its performance rolls off below that. The three SR algorithms are relatively consistent in their performance and achieve a $\boldsymbol{P_m}$ = 100% at SNR of 8.5 and greater. As expected, across the board, the SR algorithmic performance is less than or equal to that of the Virtual-1280 camera they are trying to emulate. Put another way, the cost of using SR to make an $\boldsymbol{M\ x\ N}$ pixel density camera emulate a $\boldsymbol{kM\ x\ kN}$ pixel resolution camera, provided the optical blur characteristics are adequate, is that a higher SNR is required. For thermal imagers, this would typically correspond to a larger required temperature differential between the object and background. For a visible camera, it would typically correspond to brighter required lighting conditions.

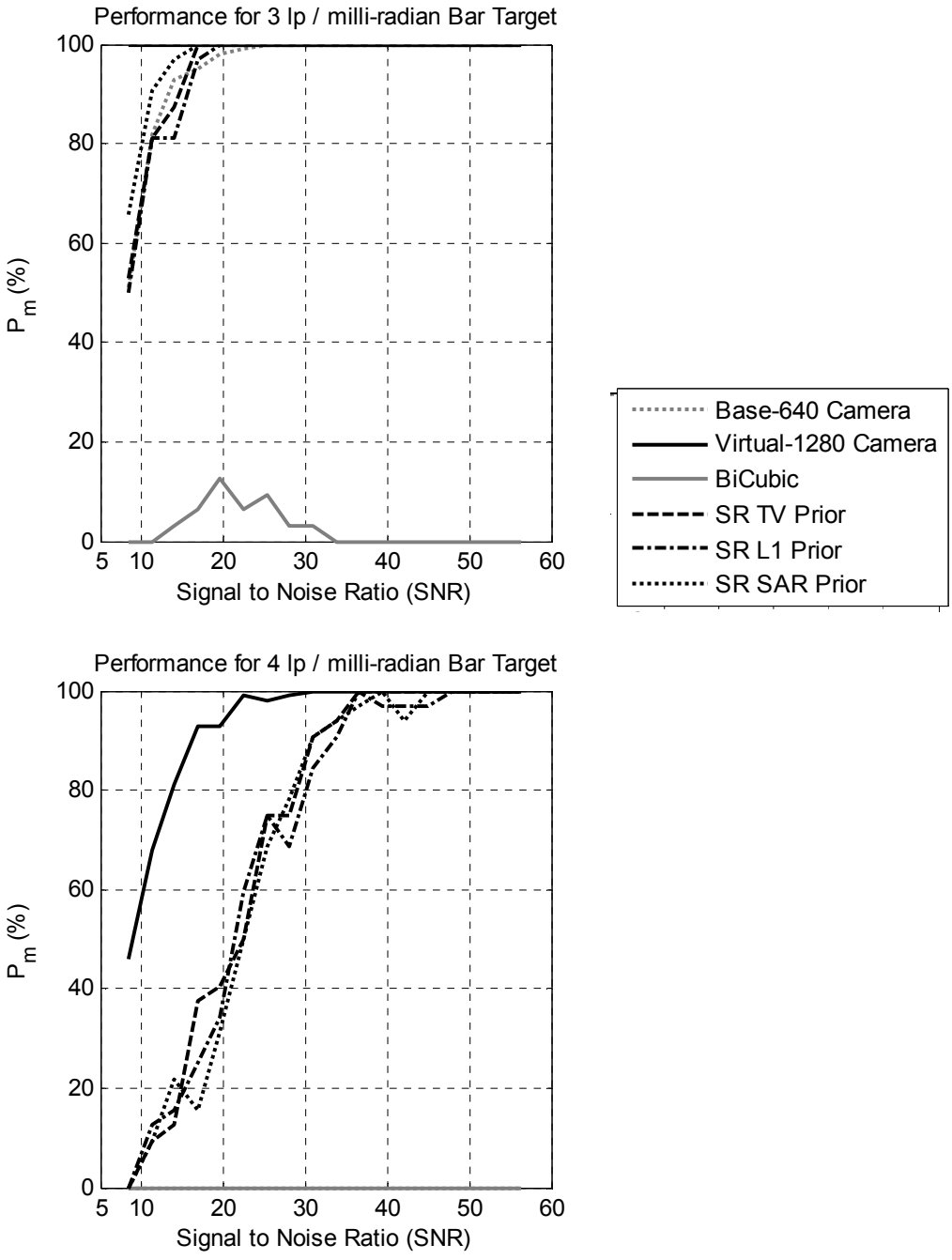


Figure 4-12: Probability of correct measurement (P_m) results for 3 and 4 lp/milli-radian bar targets

Note, in the top pane of Figure 4-12, both the base camera as well as the BiCubic algorithm have some performance success for the $\boldsymbol{\omega_{bar}}$ = 3 lp/milli-radian case even though $\boldsymbol{\omega_{max}}$ = 2.9 cycles/milli-radian. This is due to the 0.25 cycle/milli-radian acceptance band used in the determination of $\boldsymbol{P_m}$ as described in section 4.3. The band is wide enough that the slightly aliased 3 lp/milli-radian input frequency is still within the acceptance band.

Comparing the result of Figure 4-12 to the predictions of Figure 4-11, for the case of $\boldsymbol{\delta = 0.50\ IFOV}$ as used in the simulated experiments, the prediction is a conservative estimate of the achieved performance. For instance, the $\boldsymbol{\omega_{bar}}$ = 4.0 cycles/milli-radian case corresponds to $\boldsymbol{\omega_{bar}/\omega_s}$ = 0.69 for which, per Figure 4-11, should create an SNR loss of ~0.9. This would suggest an SR enhanced image would require ~11% greater SNR to achieve the same performance as the virtual high-resolution camera. In Figure 16, the virtual camera achieves a $\boldsymbol{P_m}$ of 90% at an SNR of ~15 whereas the SR algorithms achieve a $\boldsymbol{P_m}$ of 90% for SNR values between ~30 and 35. The lower performance of the actual reconstruction is attributed to the combination of additional error distribution in the estimate of the image shift as well as high-frequency smoothing resulting from the regularization.

As a final point, Figure 4-13 shows the performance of the SR algorithms, for the 4 lp / milli-radian case only, from a PSNR perspective. PSNR is defined as in (4.4.1) and the graph shows the average PSNR over the 32 trials in the Monte-Carlo experiment described above. As noted in Figure 4-12, the BiCubic algorithm is completely unable to unroll the aliased 4 lp / milli-radian frequency component. Yet, it consistently shows up as providing the highest PSNR. This is because, as mentioned earlier, PSNR is a better measure of de-blurring and de-noising than true resolution enhancement. Even though the BiCubic algorithm doesn't explicitly perform de-

blurring and de-noising, as with any interpolation based method, it intrinsically has a smoothing effect on the signal and, thereby, reduces noise.

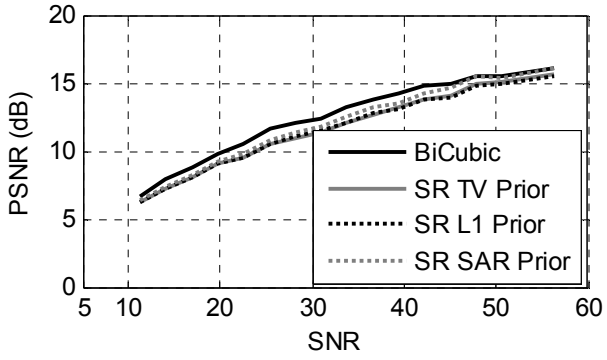


Figure 4-13: PSNR for the 4 lp/milli-radian recovery

4.6 Generalization of Results

To generalize the results of this chapter, we show the results of the Monte-Carlo in terms of a non-dimensional spatial frequency and camera blur. The non-dimensional spatial frequency is expressed as the ratio $\boldsymbol{\omega/\omega_{max}}$, where $\boldsymbol{\omega_{max}}$ corresponds to the base, LR camera. Spatial frequencies where $\boldsymbol{\omega/\omega_{max}} > 1$ correspond to those that are undetectable by the base camera due to aliasing, which we expect to recover via an SR algorithm. For the case where we are using SR to double the pixel density, ideal SR performance would, therefore, correspond to recovering spatial frequencies up to $\boldsymbol{\omega/\omega_{max}} = 2$.

In order to provide a non-dimensional measure of camera blur, we utilize the concept of "ensquared energy" (ESE), which is a fairly commonly used metric for infrared systems due to the fact that it provides a comprehensive measure of the optical performance of an imaging system based on a single value.as well as its ease of empirical measurement on a test station [119,120]. Referencing the illustration in Figure 4-14, the ESE of an optical system is defined as the percentage of the energy of the point spread function received by a single pixel at the phasing that

provides the maximum. ESE is applicable for any blur function. To simplify the analysis; however, we again assume a Gaussian blur. In that case, the MTF is related to the ESE by

$$\boldsymbol{H(\omega_x, \omega_y) = exp\left(-2\pi^2\sigma^2\left(\omega_x^2 + \omega_y^2\right)\right)}$$
$$\boldsymbol{\sigma = \sqrt{2}\frac{IFOV}{4erf^{-1}(\sqrt{ESE})}}, \tag{4.6.1}$$

where $\boldsymbol{erf^{-1}(x)}$ represents the inverse of the Gaussian error function given by $\boldsymbol{erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt}$. For reference, the diffraction limited, Tau-640 LWIR camera simulated in this section has an ESE of ~65%. A high ESE value for a camera suggest that the camera is under-sampled relative to the blur and, therefore, amenable to SR enhancement.

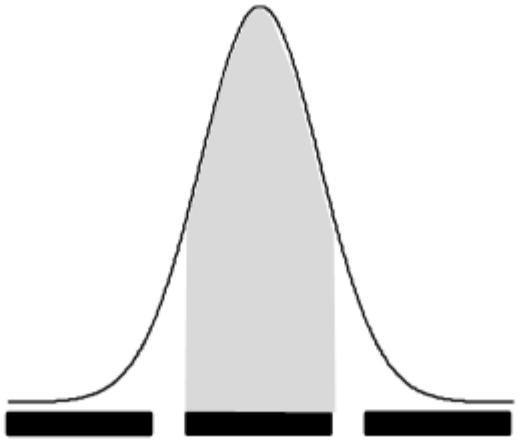


Figure 4-14: Illustration of "EnSquared Energy" definition

Using these definitions, we can use the Monte-Carlo method to generate the $\boldsymbol{P_m(\omega_{bar}, SNR)}$ metric for the variational Bayesian SR algorithm with TV prior over a range of ESE values (20% to 90%). The test was only performed for the TV prior given the similarity of performance of the various algorithms in Figure 4-12. The results of this multi-dimensional analysis are sown in Figure 4-15 with SNR on the y-axis and $\boldsymbol{\omega/\omega_{max}}$ on the x-axis. There is one

curve for each value of ESE. The interpretation of each curve is that it shows the minimum spectral SNR required to resolve each $\boldsymbol{\omega/\omega_{max}}$ spatial frequency with probability $\boldsymbol{P_m}$ = 50% (top graph) and $\boldsymbol{P_m}$ = 90% (bottom graph).

As expected, even at high SNR, a camera with poor focus (ESE < 50%) is unable to be practically super-resolved. As the ESE increases, with sufficient SNR, the algorithm gradually approaches the ideal capability of recovering $\boldsymbol{\omega/\omega_{max}}$ near 2. As shown in back in Figure 4-11, the SR algorithm will never be able to actually recover $\boldsymbol{\omega/\omega_{max}}$ = 2 because this frequency is extinguished by the pixel integration response predicted by equation (4.6.13).

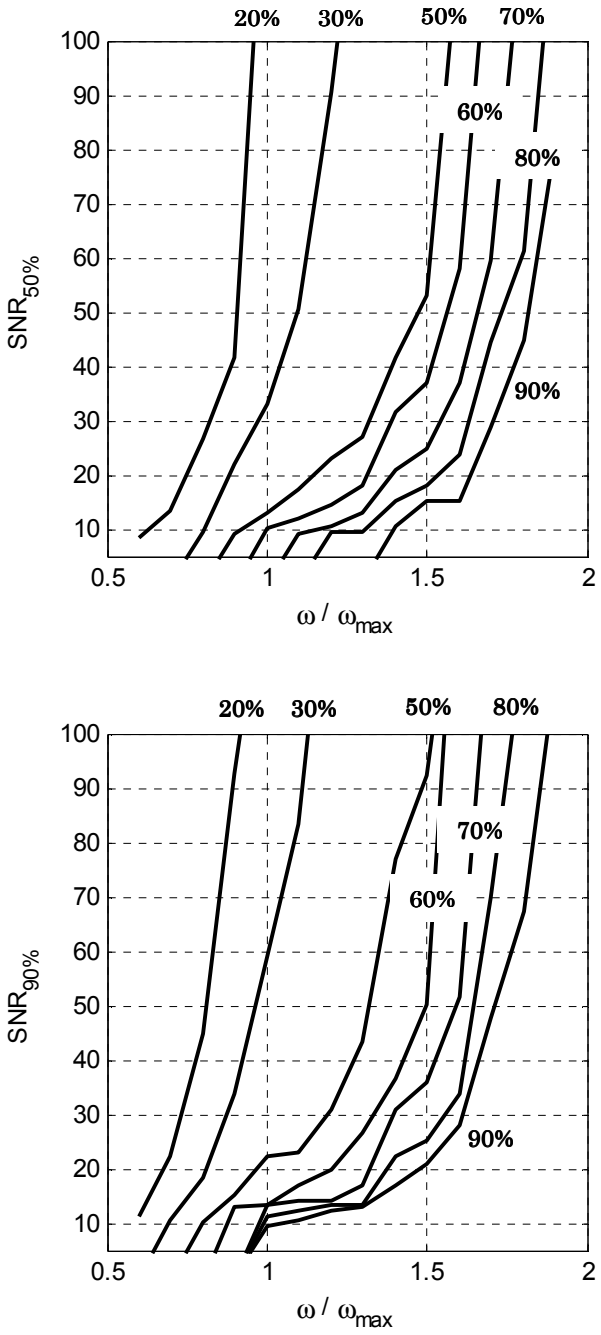


Figure 4-15: Non-dimensionalized probability of measurement results from variational Bayesian inference SR with TV Prior

Figure 4-15 provides a very general prediction of the capability limits of using SR in an application. In a typical vision task, the critical frequency, the signal level, the proposed camera sensitivity, and proposed camera ESE are all known and, therefore, Figure 4-15 may be used to assess the feasibility of the solution. Or, if camera design parameters have not yet been determined, Figure 4-15 may be used to derive the requirements for the sensitivity and/or ESE of the camera.

4.7 SR Spatial Matric Results on a Samsung 5 Galaxy Inexpensive Camera

In order to reinforce our simulation based results, we complement with actual camera imagery. The camera used is the Samsung Galaxy 5 smartphone (see Appendix A). This device falls into a class of inexpensive, visible band imaging devices. For the experiments, we created a resolution bar target closely mimicking that of the simulations. The target is shown in Figure 4-16. When printed on a standard legal 21.6 x 27.9 cm (8.5 x 11 in) paper, the rows correspond to spatial frequencies of 39.4 lp / meter, 78.7 lp / meter, and 196.9 lp / meter. When viewed by the camera at varying ranges, these produce a variety of sample frequencies, $\boldsymbol{\omega_{bar}}$, in terms of lp / milli-radian. The single black bar at the bottom allows us to measure the low frequency depth of modulation as well as the camera noise. It also allows us to coarsely register multiple images prior to running the SR algorithms.

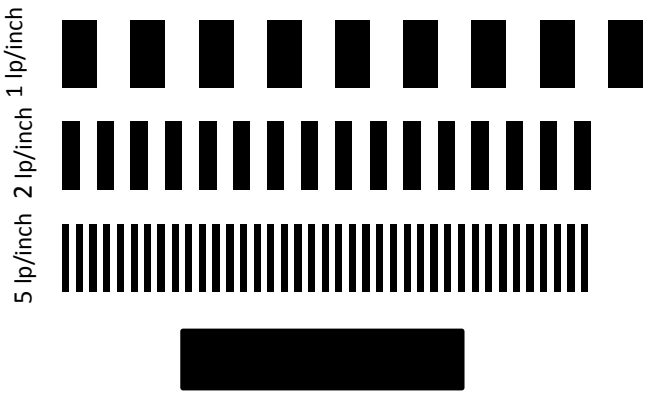


Figure 4-16: Resolution bar target used for real, visible band camera experiments

The printout of Figure 4-16 was imaged by the camera, in outside, daytime lighting conditions, at ranges of 5.4, 9.0, 12.0, and 17.4 meters. This produced 12 samples of the bar targets with spatial frequencies ranging from 0.2 lp / milli-radian to 3.4 lp / milli-radian. Four representative samples are shown in Figure 4-17. The measured pixel level SNR for these outdoor images is ~30 with a measured IFOV of 0.23 milli-radians. We evaluate the Fourier spectrum using an approximately 27.5 milli-radian x 3.5 milli-radian rectangle which, per equation (4.6.16), results in a very high spectral SNR of ~640.

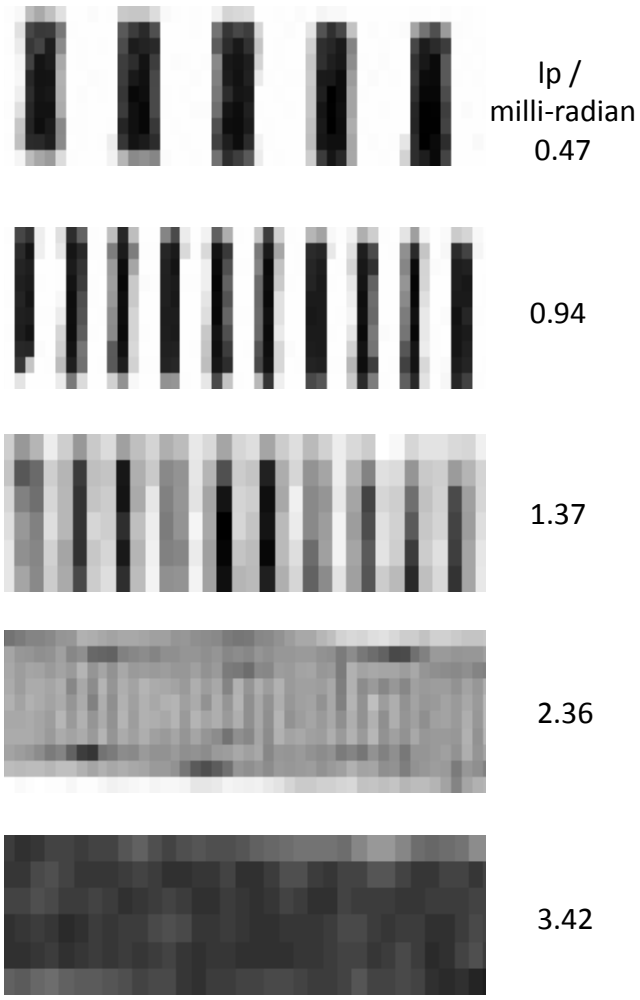


Figure 4-17: Samples of bar target images at various spatial frequencies from the Samsung Galaxy 5 camera

Calibration of the camera's geometry and MTF is discussed in Appendix B. In Appendix B, we measure the IFOV of the camera to be 0.23 milli-radians which produces $\boldsymbol{\omega_s}$ = 4.35 cycles / milli-radian and $\boldsymbol{\omega_{max}}$ = 2.17 cycles / milli-radian. Therefore, we expect frequencies below $\boldsymbol{\omega_{max}}$ =2.17 cycles/milli-radian to pass through to the digital image whereas frequencies from $\boldsymbol{\omega_{max}}$ to just over 3 cycles/milli-radian will be both attenuated and aliased. Frequencies much above 3

cycles/milli-radian will be significantly attenuated. This is illustrated in Figure 4-18 which shows the frequency response of the base camera for $\boldsymbol{\omega_{bar}}$= 0.94, 2.36, and 3.42 lp / milli-radian bar targets. The $\boldsymbol{\omega_{bar}}$ = 2.36 first peak is noticeably aliased to $\boldsymbol{\omega_s - \omega_{bar}}$= 2.0 cycles/milli-radian (per (4.5.2)). Although greatly attenuated, the $\boldsymbol{\omega_{bar}}$ = 3.42 cycles/milli-radian first peak also shows up as a maximum on the frequency response at $\boldsymbol{\omega_s - \omega_{bar}}$ = 0.9 cycles/milli-radian.

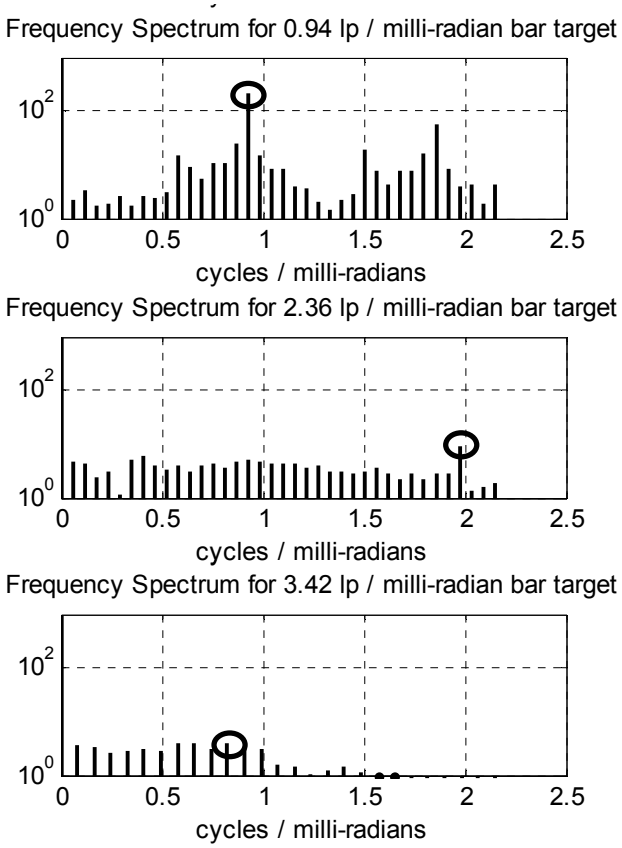


Figure 4-18: Spatial frequency spectrums for images of 0.94, 2.36, and 3.42 lp/milli-radian bar targets Aliasing is apparent for both the 2.36 and 3.42 lp/milli-radian targets.

Evaluation of SR Algorithms on the Samsung Imagery

In Figure 4-19 and Figure 4-20, we show the resulting images and spectrums of the SR algorithm with SAR prior on the stressing 2.36 and 3.24 lp / milli-radian bar targets. As we could not precisely control the sub-pixel phasing for the real camera samples, we utilized a total of 4 LR images with effectively random, horizontal translational shifts achieved through small rotations of the camera. Both of these represent cases where $\boldsymbol{\omega_{bar} > \omega_{max}}$. The results are nearly identical for the SR algorithms with TV prior and L1 prior. In order to corroborate the result with simulation, we use the MTF curve in Appendix B and equation (2.1.1) to generate simulations of the bar targets with SNR = 640 and run the simulated images through the SR algorithms. The simulations agree with the actual results in Figure 4-20. For clarity, the vertical axis of the spectrum plots in Figure 4-20 are clipped to 10^{-3}. The SR algorithms perform well in the 2.36 lp / milli-radian case, due to the fact that the camera optics have sufficient response, but do not perform well in the 3.42 lp / milli-radian case. The result is also consistent with the predictions for a 30% ESE camera, even at high SNR, from Figure 4-15.

Unlike for the simulated experiment in section 4.6, for the real experiment, we did not create a $\boldsymbol{P_m(\omega_{bar}, SNR)}$ plot. This plot could be generated for real data. It would require repeating the experiment a statistically significant number of times. During the experiment, SNR could be varied either by changing the ambient lighting conditions or by using multiple targets with different contrast between the black and white bars. Also, as noted above, simulations of the process using the estimated camera MTF and SNR agreed with the results for the real data. Consequently, an alternate method for the generation of the $\boldsymbol{P_m(\omega_{bar}, SNR)}$ plots is to use

simulation over SNR based on the camera parameters and then corroborate the results with real measurements based on availability.

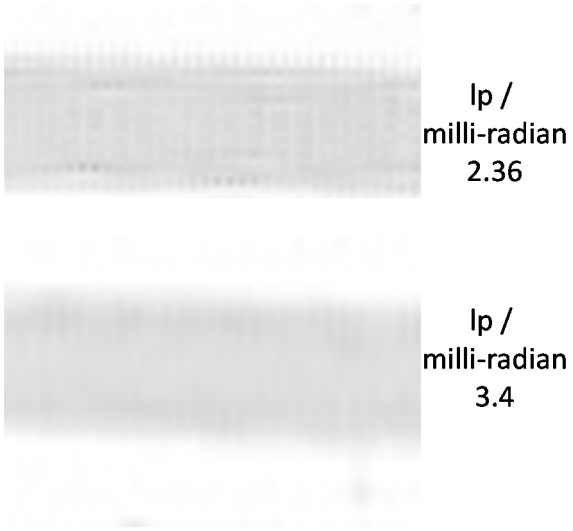


Figure 4-19: SR with SAR Prior recovered images of 2.4 and 3.4 lp/milli-radian bar targets

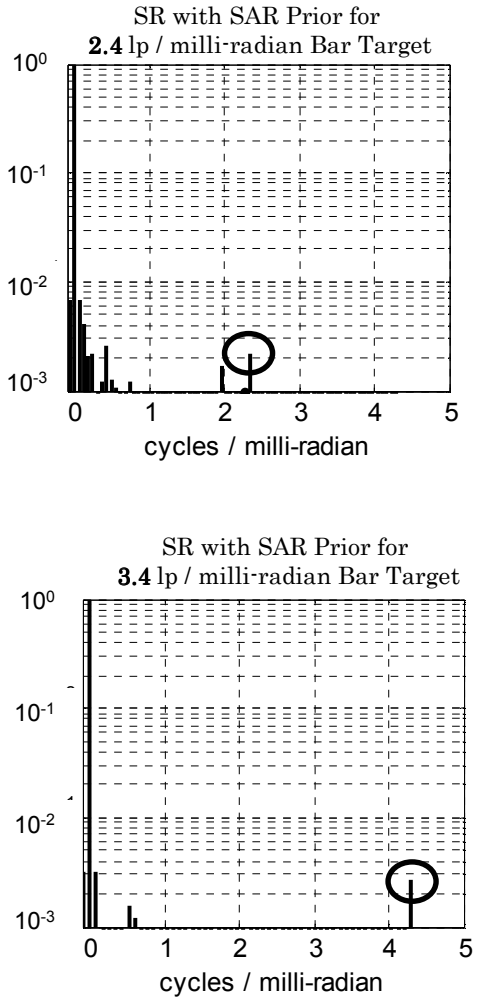


Figure 4-20: SR with SAR Prior spatial frequency spectrum of recovered 2.4 and 3.4 lp/milli-radian bar targets

4.8 Conclusion

In this chapter, we have first reinforced the importance of considering both optics and pixel integration effects of a camera when generating an expectation of the success of applying an SR algorithm to make a native $\boldsymbol{M\ x\ N}$ pixel density camera emulate a $\boldsymbol{kM\ x\ kN}$ pixel density camera. If the higher spatial frequencies that the SR algorithm needs to recover are attenuated to the extent that they are buried within the camera's noise, then no candidate SR algorithm will be able to recover them. Secondly, we showed, both analytically and via Monte-Carlo simulation that the principal penalty of an SR approach is an effective reduction in the camera's MTF such that it will require a higher scene SNR in order for the super-resolved lower pixel density camera to perform as well as a true higher pixel density system. Third, we illustrated that performance metrics based on either perceived image quality or, even quantitatively on PSNR, can produce potentially misleading results as they simultaneously credit image enhancements such as de-blurring or de-noising which do improve the perceptual quality of the imagery but do not recover aliased frequencies. Instead, we proposed the probability of correctly measuring a spatial frequency $\boldsymbol{\omega_{bar}}$ contained within the external scene with signal to noise ratio SNR, $\boldsymbol{P_m(\omega_{bar}, SNR)}$, as a metric to properly establish the capability of an SR algorithm. Additionally, based on the historic Johnson criteria for detection and classification, we claimed that it is sufficient to establish this metric using, exclusively, either real or simulated images of resolution bar targets. That is, we can use this general metric for an initial assessment of the effectiveness of SR algorithms for any specific detection and classification problem domain; e.g. remote face recognition, remote text recognition, etc., The effectiveness of an SR algorithm on a given camera is then assessed by comparing achieved $\boldsymbol{P_m(\omega_{bar}, SNR)}$ between the base $\boldsymbol{M\ x\ N}$ pixel density camera, the base camera super-

resolved to an equivalent $\boldsymbol{kM\ x\ kN}$ pixel density, and a virtual camera with identical optics as the base camera but with a true $\boldsymbol{kM\ x\ kN}$ pixel density. Finally, using the new metric, we generated a set of general design curves that may be used to derive requirements for sensitivity and or MTF of cameras that are going to be supplemented with SR.

In this chapter, we have used bar targets for their simplicity and traceability to the historic Johnson criteria. However, the same principle of assessing SR performance based upon its ability to recover aliased frequency components on a resolution target could be extended to a Siemens star target [121,122] or any of the other spatial frequency response (SFR) targets defined in the ISO 12233 [49] standard.

The $\boldsymbol{P_m(\omega_{bar}, SNR)}$ metric is flexible in the sense that it can be evaluated through Monte-Carlo simulations of a camera or on real camera data. One practical restriction, in both cases, the scene must contain a calibrated resolution bar target. An extension of this work could attempt to alleviate this requirement, particularly in the case of simulation based evaluation, by permitting evaluation on an arbitrary scene.

CHAPTER 5

EXTENDING SUPER-RESOLUTION TO THE AIRBORNE DOMAIN

In this chapter, we derive a new formulation of the SR algorithm using the Hierarchical Bayesian approach that has the advantage of being simpler than the Variational Bayesian Inference (VBI) solution discussed in 2.4. As mentioned in Chapter 2, the VBI methods are powerful; however, the derivation is advanced, tedious, and model specific [71,72]. It is difficult to quickly experiment with different forward models as the VBI solution must be carefully re-derived each time. This frequently leads practitioners to abandon the statistically optimal VBI methods in favor of conceptually simpler methods such as obtaining a single point MAP solution. Unfortunately, solutions such as MAP solution can not exploit the full potential offered by probabilistic modeling as only the posterior mode is sought [68]. Here, we capitalize on observations made based on previous VBI solutions, namely the fact that several of the posteriors turn out to have a Gaussian functional form, to derive a more readily adaptable form using Gaussian filters. We then use this simplified solution to accommodate more complex forward models, relevant to the airborne domain, that include factors such as wide field-of-view, significant lens distortion, and oblique viewing geometries.

5.1 An Information-Filter Formulation of Super-Resolution

It is well known that the super-resolution (SR) problem is best approached as one of inferring a high-resolution image based on a combination of measured low-resolution data as well as prior information [65,66]. Without this composite information, which serves to regularize the problem, the fundamental image reconstruction problem is ill-posed and, consequently, attempts to solve the forward image formation model (2.1) by direct inversion will perform poorly in

realistic scenarios corrupted by noise and other degradations. The desire to both incorporate prior information in a principled fashion as well as to pose the SR problem as one of statistical inference leads naturally to a hierarchical Bayesian formulation.

We start with the image formation model in (2.4.1); i.e. $\boldsymbol{y_k = AH_kC(s_k)x + n_k}$. For notational convenience, we combine the down-sampling, blur, and warping matrices into a single matrix $\boldsymbol{B_k}$ where $\boldsymbol{B_k = AHC_k(s_k)}$. At this point, we can apply the hierarchical Bayesian model from 2.4

$$\boldsymbol{Pr}\left[\boldsymbol{x}, \{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha} | \{\boldsymbol{y_k}\}, \{\boldsymbol{\Omega_{sk}}\}\right] \propto \prod_{k=1}^{K} \left(\boldsymbol{\beta_k^{mn/2}} \boldsymbol{exp}\left(-\frac{\beta_k}{2} \|\boldsymbol{y_k} - \boldsymbol{B_k(s_k)x}\|^2\right) \boldsymbol{exp}\left(-\frac{1}{2}\boldsymbol{s_k^T \Omega_{sk} s_k}\right)\right) \boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}], \tag{5.1.1}$$

where $\{\boldsymbol{y_k}\}$ is the set of $\boldsymbol{K}$ measured, $\boldsymbol{(m\ x\ n)}$ low-resolution images represented as a column vector in lexicographic order. $\boldsymbol{x}$ is the unknown, $\boldsymbol{(M\ x\ N)}$ high-resolution image represented as a column vector in lexicographic order. $\{\boldsymbol{s_k}\}$ is the set of $\boldsymbol{K}$ unknown warping parameter corrections (relative to initial estimates), $\boldsymbol{\Omega_{sk}}$ is the measurement precision matrix of the warping parameters for frame $\boldsymbol{k}$, $\boldsymbol{\beta_k}$ is a hyper-parameter for the likelihood of measured image $\boldsymbol{k}$, $\boldsymbol{\alpha}$ is a hyper-parameter for high-resolution image prior model, and the term $\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}]$ represents the high-resolution image prior model. The ideal objective is to find the expected value of the marginal distribution of $\boldsymbol{x}$; that is,

$$\boldsymbol{E}[\boldsymbol{x}] = \int_{\boldsymbol{x},\{\boldsymbol{s_k}\},\{\boldsymbol{\beta_k}\},\boldsymbol{\alpha}} \boldsymbol{x Pr}\left[\boldsymbol{x}, \{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha} | \{\boldsymbol{y_k}\}, \{\boldsymbol{\Omega_{sk}}\}\right] \boldsymbol{dx d}\{\boldsymbol{s_k}\} \boldsymbol{d}\{\boldsymbol{\beta_k}\} \boldsymbol{d\alpha}. \tag{5.1.2}$$

By taking the expected value as opposed to an alternative such as the single-point MAP estimate, we properly account for the, possibly large, variation in the auxiliary parameters $\{\boldsymbol{s_k}\}, \{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$.

Unfortunately, solving (5.1.2) analytically for the marginal posterior is, in general, not tractable. This leaves a set of options. The first is to employ sampling methods such as Markov Chain Monte Carlo (MCMC) [68]. These work in principle but are computationally expensive. They do, however, lend themselves to mass parallelization. So, it is possible that with the increase in computing power, particularly using devices such as GPUs that exploit mass parallelization, these methods may see a reemergence. The second category is to use approximation methods such as Variational Bayesian Inference (VBI) [69]. The VBI approach is used in state-of-the-art solutions [65-67].

As discussed in 2.4, prior work on VBI for SR has shown that, without prior assumption, the distribution models which minimize the Kulback-Leibler (KL) divergence for both the unknown high-resolution image x as well as unknown image registration parameters $\{\boldsymbol{s_k}\}$ are, indeed, Gaussian distributions [65,67]. Given this result, we propose, as an alternative and analytically simpler approach which still preserves the statistical benefits of VBI, to solve the SR problem using the well-established tools and theory surrounding Bayesian Gaussian filters [22]. The extended Kaman filter (EKF), which supports non-linear models via local linearization, is the most popular of these and has been previously examined for SR [123]. However, the EKF solution creates several issues. We offer two changes to address these issues. The most significant issue arises from the required size of the state covariance matrix. An HR image of size $\boldsymbol{(M\ x\ N)}$ pixels requires a very large covariance matrix with $\boldsymbol{(MN)^2}$ elements. However, even though the covariance matrix is large, it is very sparse. The actual required size can be judged by estimating the number of likely non-zero correlations for each image pixel. Each pixel will be correlated to its immediate neighbors due to motion and blur as well as to the augmented parameters such as

motion, etc. If there are $\boldsymbol{P}$ of these non-trivial correlation elements, we expect the total size to be $\boldsymbol{PMN} \ll (\boldsymbol{MN})^{\boldsymbol{2}}$. Consequently, by using modern numerical linear algebra software which is able to utilize sparse matrices, we are able to overcome this limitation. Support for sparse matrix operations is available in such packages as Matlab as well as NVIDIA's Cuda library [124].

Secondly, the EKF in its native form has numerical difficulties with the SR problem. The EKF is characterized by a relatively easy time update and a more difficult measurement update. In contrast, its dual formulation, the information filter (IF) has a difficult time update and a relatively easy measurement update [22]. In the SR formulation, we are effectively only using the measurement update of the Bayesian filter. Therefore, the IF turns out to be a computationally simpler and more numerically stable solution. In this section, we derive the solution for the Gaussian unknowns $\boldsymbol{x}, \{\boldsymbol{s_k}\}$ using the IF method. We also investigated the Unscented Kalman Filter (UKF) which is a Gaussian filter closely related to the EKF [22,125] and has some precedent in image enhancement applications such as film-grain removal [126]. In principle, the UKF is better able to handle non-linear models than the EKF because it handles non-linear input/output functions applied to Gaussian variables directly through the unscented transform (UT) as opposed to though approximate, local linearization. There is, however, no existing dual formulation of the UKF akin to the IF and we find the UKF suffers from the same numerical difficulties with SR as the EKF.

Derivation of the IF Solution

We start with a general description of the IF as a method that, given an unknown state vector $\boldsymbol{\theta}$ and a series of $\boldsymbol{P}$ measurements $\{\boldsymbol{z_p'}\}$, solves for posterior probability distribution of the form

$$\boldsymbol{Pr}\left[\boldsymbol{\Delta\theta}|\{\boldsymbol{z'_p}\}\right] = \frac{\prod_{p=1}^{P} \boldsymbol{Pr}[\boldsymbol{z'_p}|\boldsymbol{\Delta\theta}]\boldsymbol{Pr}[\boldsymbol{\Delta\theta}]}{\boldsymbol{Pr}[\{\boldsymbol{z'_p}\}]}. \tag{5.1.3}$$

where $\Delta\boldsymbol{\theta}$ is a correction to the state $\boldsymbol{\theta}$, allowing us to handle non-linear models through local linearization. An analytic solution to (5.1.3) is available for the special case that all of the probability functions are Gaussian. The solution is given by

$$\boldsymbol{Pr}\left[\boldsymbol{\Delta\theta}|\{\boldsymbol{z'_p}\}\right] \sim \boldsymbol{N}\left(\overline{\boldsymbol{\Delta\theta}}, \boldsymbol{\Omega}_{\boldsymbol{\Delta\theta}}^{-\mathbf{1}}\right), \tag{5.1.4}$$

$$\boldsymbol{Pr}[\boldsymbol{\Delta\theta}] \sim \boldsymbol{N}\left(\mathbf{0}, \boldsymbol{\Omega}_{\mathbf{0}}^{-\mathbf{1}}\right), \tag{5.1.5}$$

$$\boldsymbol{z'_p} = \boldsymbol{z_p} - \boldsymbol{h_p}(\boldsymbol{\theta_0}) \cong \left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]\boldsymbol{\Delta\theta}, \text{ and} \tag{5.1.6}$$

$$\boldsymbol{Pr}\left[\boldsymbol{z'_p}|\boldsymbol{\Delta\theta}\right] \sim \boldsymbol{N}\left(\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]\boldsymbol{\Delta\theta}\,, \boldsymbol{Q_p}\right). \tag{5.1.7}$$

where $\boldsymbol{\theta_0}$ indicates our initial, best-estimate of the state, $\overline{\boldsymbol{\Delta\theta}}$ and $\boldsymbol{\Omega}_{\boldsymbol{\Delta\theta}}^{-\mathbf{1}}$ are the mean and covariance of the posterior of $\boldsymbol{\Delta\theta}$, $\boldsymbol{Q_p}$ is the covariance of the linearized measurement $\boldsymbol{z'_p}$. Note, in (5.1.6) and (5.1.7), we allow for a non-linear measurement function $\boldsymbol{h}(\boldsymbol{\theta})$. As we will see, this allows us to both accommodate complex image formation models as well as introduce shaping transformations to handle non-Gaussian priors. The price of this is that we have to adopt an iterative solution based on the first order Taylor series expansion of $\boldsymbol{h}(\boldsymbol{\theta})$.

The Kalman filter paradigm is centered around the concept of measurements. However, we also need to incorporate prior models which are based exclusively on the state vector and don't truly have an associated measurement. We handle these through the concept of "pseudo-measurements" [127] which has been applied successfully in the Kalman filter based tracking domain but not yet, to our knowledge, in the image processing domain. A pseudo-measurement

term enforces (5.1.6 and 5.1.7) with $\boldsymbol{z_p \equiv 0}$; thus, imposing a probabilistic constraint based exclusively on the state vector itself.

In order to cast the Bayesian model of (5.1.1) into a series of "measurements" and "pseudo-measurements", as required by (5.1.4 to 5.1.7), we address each term separately. The data likelihood for each image, given by

$$\boldsymbol{Pr[y_k|x, s_k, \beta_k] = \beta_k^{mn/2} exp\left(-\frac{\beta_k}{2}\|y_k - B_k(s_k)x\|^2\right),} \tag{5.1.8}$$

is represented by setting

$$\boldsymbol{z_p = y_k,} \tag{5.1.9}$$

$$\boldsymbol{h_p(\theta) = B_k(s_k)x}, \text{ and} \tag{5.1.10}$$

$$\boldsymbol{Q_p^{-1} = \beta_k I_{mn\,x\,mn}.} \tag{5.1.11}$$

In (5.1.11), $\boldsymbol{I_{mn\,x\,mn}}$ is the $\boldsymbol{(mn\,x\,mn)}$ identity matrix.

Likewise, the prior on the warping parameter corrections, given by

$$\boldsymbol{Pr[s_k] = exp\left(-\frac{1}{2}s_k^T \Omega_{sk} s_k\right),} \tag{5.1.12}$$

is represented by setting

$$\boldsymbol{z_p = 0,} \tag{5.1.13}$$

$$\boldsymbol{h_p(\theta) = s_k}, \text{ and} \tag{5.1.14}$$

$$\boldsymbol{Q_p^{-1} = \Omega_{sk}.} \tag{5.1.15}$$

In this formulation of the SR problem, we define the augmented state vector $\boldsymbol{\theta}$ to be the union of image $\boldsymbol{x}$ and the warping vectors $\{\boldsymbol{s_k}\}$; i.e.,

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{x} \\ \{\boldsymbol{s_k}\} \end{bmatrix}. \tag{5.1.16}$$

With this definition of $\boldsymbol{\theta}$, we can now define the matrix $\left[\frac{\delta \boldsymbol{h_p}}{\delta \boldsymbol{\theta}}\right]$ for both the low-resolution image and warping parameter correction measurements, as defined by (5.1.10) and (5.1.14) respectively, as

$$\left[\frac{\delta \boldsymbol{h_p}}{\delta \boldsymbol{\theta}}\right] = \begin{bmatrix} \boldsymbol{B_k(s_k)} & \frac{\delta \boldsymbol{B_k(s_k)}}{\delta \boldsymbol{s_{k1}}} \boldsymbol{x} & \cdots & \frac{\delta \boldsymbol{B_k(s_k)}}{\delta \boldsymbol{s_{kp}}} \boldsymbol{x} \end{bmatrix} \tag{5.1.17}$$

for the low-resolution image measurement model, and

$$\left[\frac{\delta \boldsymbol{h_p}}{\delta \boldsymbol{\theta}}\right] = [\boldsymbol{0} \quad \boldsymbol{I_{pxp}}] \tag{5.1.18}$$

for the warping parameter corrections. In (5.1.17), $\boldsymbol{s_{kj}}$ represents the j'th element of the p-element, model specific parameterization of the warping vector $\boldsymbol{s_k}$. In (5.1.18), $\boldsymbol{I_{pxp}}$ represents the $\boldsymbol{pxp}$ identity matrix.

The final item we need to address is handling common forms of the image prior model $\boldsymbol{Pr}[\boldsymbol{x}|\boldsymbol{\alpha}]$ using pseudo-measurements. By setting $\boldsymbol{z_p} = 0$ in (5.1.6) and (5.1.7), we add a probability term which is exclusively a function of the state $\boldsymbol{\theta}$. That is,

$$\boldsymbol{Pr}\left[\boldsymbol{z_p'}|\boldsymbol{\theta}\right] = \boldsymbol{Pr}[\boldsymbol{\theta}] \propto \boldsymbol{exp}\left(-\frac{\boldsymbol{1}}{\boldsymbol{2}} \boldsymbol{h_p(\theta)^T Q_p^{-1} h_p(\theta)}\right). \tag{5.1.19}$$

With this development, we can now represent the common image priors.

Pseudo-Measurements for Common Image Priors

Total Variation (TV) Prior

The TV prior is commonly used for image reconstruction problems due to its inherent ability to retain sharp gradients at image edges [65,67]. The TV prior is given by

$$\boldsymbol{Pr}\left[\boldsymbol{z_p'}|\boldsymbol{\theta}\right] = \boldsymbol{\alpha}^{\boldsymbol{MN/2}}\boldsymbol{exp}\left(-\frac{\boldsymbol{\alpha}}{\boldsymbol{2}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{MN}}\sqrt{\boldsymbol{\Delta h_i}^{\boldsymbol{2}} + \boldsymbol{\Delta v_i}^{\boldsymbol{2}}}\right), \tag{5.1.20}$$

where $\boldsymbol{\Delta h_i}$ and $\boldsymbol{\Delta v_i}$ are the horizontal and vertical gradients, respectively, for element $\boldsymbol{i}$ in the $\boldsymbol{(M\ x\ N)}$ high-resolution image $\boldsymbol{x}$. $\boldsymbol{\Delta h_i}$ and $\boldsymbol{\Delta v_i}$ may be found using any reasonable gradient estimator. For our work, we use the method of first-order-differencing as in [65]. The probability term (5.1.20) is represented in the IF through a pseudo-measurement where

$$\boldsymbol{h_p(\theta)} = \begin{bmatrix} \left(\boldsymbol{\Delta h_1}^{\boldsymbol{2}} + \boldsymbol{\Delta v_1}^{\boldsymbol{2}}\right)^{\boldsymbol{1/4}} \\ \vdots \\ \left(\boldsymbol{\Delta h_{MN}}^{\boldsymbol{2}} + \boldsymbol{\Delta v_{MN}}^{\boldsymbol{2}}\right)^{\boldsymbol{1/4}} \end{bmatrix}, \text{ and} \tag{5.1.20A}$$

$$\boldsymbol{Q_p^{-1}} = \boldsymbol{\alpha I_{MN\ x\ MN}}. \tag{5.1.20B}$$

In this case, we derive the Jacobian

$$\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right] = \frac{\boldsymbol{1}}{\boldsymbol{2}}\begin{bmatrix} \left(\boldsymbol{\Delta h_1} \quad \frac{\boldsymbol{\delta\Delta h_1}}{\boldsymbol{\delta\theta}} + \boldsymbol{\Delta v_1} \quad \frac{\boldsymbol{\delta\Delta v_1}}{\boldsymbol{\delta\theta}}\right) \\ \vdots \\ \left(\boldsymbol{\Delta h_{MN}} \quad \frac{\boldsymbol{\delta\Delta h_{MN}}}{\boldsymbol{\delta\theta}} + \boldsymbol{\Delta v_{MN}} \quad \frac{\boldsymbol{\delta\Delta v_{MN}}}{\boldsymbol{\delta\theta}}\right) \end{bmatrix}. \tag{5.1.20C}$$

Simultaneous Auto-Regressive (SAR) Prior

The SAR prior is also commonly used in the image recovery literature due to its simplicity. It is fundamentally a Gaussian prior. However, it is known to not preserve image edges as well as the TV prior above. The SAR prior is given by

$$\boldsymbol{Pr}\left[\boldsymbol{z}_{\boldsymbol{p}}'|\boldsymbol{\theta}\right] = \boldsymbol{\alpha}^{\boldsymbol{MN/2}}\boldsymbol{exp}\left(-\frac{\boldsymbol{\alpha}}{\mathbf{2}}\|\boldsymbol{Cx}\|^{\mathbf{2}}\right), \tag{5.1.21}$$

where $\boldsymbol{C}$ is the Laplacian matrix. The SAR prior penalizes high-frequency content, such as noise, in the high-resolution image. The probability term (5.1.21) is represented in the IF through a pseudo-measurement where

$$\boldsymbol{h_p(\theta) = Cx}, \text{ and} \tag{5.1.21A}$$

$$\boldsymbol{Q_p^{-1} = \alpha I_{MN\,x\,MN}}. \tag{5.1.21B}$$

In this case, we derive the Jacobian

$$\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right] = [\boldsymbol{C} \quad \mathbf{0}]. \tag{5.1.21C}$$

IF Solution

With the modelling complete, we can now proceed to find the expected value of the state correction, $\overline{\boldsymbol{\Delta\theta}}$, through the standard solution to the linear information filter from [22] given by,

$$\boldsymbol{\Omega} = \sum_{\boldsymbol{p=1}}^{\boldsymbol{P}}\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]^{\boldsymbol{T}}\boldsymbol{Q_p^{-1}}\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right], \tag{5.1.22A}$$

$$\boldsymbol{\xi} = \sum_{\boldsymbol{p=1}}^{\boldsymbol{P}}\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]^{\boldsymbol{T}}\boldsymbol{Q_p^{-1}}\left[\boldsymbol{z'}_{\boldsymbol{p}}\right], \text{ and} \tag{5.1.22B}$$

$$\overline{\boldsymbol{\Delta\theta}} = \boldsymbol{\Omega^{-1}\xi}. \tag{5.1.22C}$$

Equations (5.1.22A) to (5.1.22C) gives us both the expected value of the state as well as posterior distribution. However, as mentioned before, the result isn't exact due to the fact that we needed to

linearize the model via the Taylor approximation $\boldsymbol{h_p(\theta\ )} = \boldsymbol{h_p(\theta_0)} + \left[\frac{\delta h_p}{\delta \theta}\right]\Delta\boldsymbol{\theta}$. We, therefore, need to iterate.

The only computationally complex part of (5.1.22A) to (5.1.22C) is the inversion of the precision matrix $\boldsymbol{\Omega^{-1}}$ in (5.1.22C). However, $\boldsymbol{\Omega}$ is sparse, symmetric, and positive-definite allowing the factorization of $\overline{\boldsymbol{\Delta\theta}} = \boldsymbol{\Omega^{-1}\xi}$ to be performed efficiently. In fact, there are optimized GPU functions to factorize (5.1.22C) available in packages such as the sparse cuSPARSE library from NVIDIA CUDA [124].

Automatic Differentiation

One common challenge encountered in SR solutions utilizing the model of 5.1.1 is the need to compute derivatives, particularly of the $\boldsymbol{B_k(s_k)}$ matrix which is, in general, a complex, non-linear function. We encountered this derivative above in the IF solution (e.g. (5.1.17)); however, it is also required for VIB and MAP solutions. One obvious solution is to derive a closed-form expression for the derivative as is done in [65]. However, this can be tedious and error prone as the model gets more complex. The second common method, used in [69] is to numerically evaluate the derivative by central-differencing [128]; i.e. for a general function $\boldsymbol{y(x)}$, $\frac{\boldsymbol{dy}}{\boldsymbol{dx}} \cong \frac{\boldsymbol{y(x+\Delta x)-y(x-\Delta x)}}{\boldsymbol{2\Delta x}}$. This is a viable option; however, for complex and non-linear models can be very sensitive to the choice of the $\boldsymbol{\Delta x}$ value. There is a third option, called exact automatic differentiation (AD) [128]. This term is often confused with, but completely different from, numerical differentiation and is guaranteed to calculate the exact derivative for any function that can be calculated by a computer program. Due to this confusion, the potential value of the AD method is under-used in the general community. It works off the principle that any computed function, no matter how complex, is

implemented in the computer by a series of simple, atomic operations with known analytic derivatives; e.g. addition, subtraction, multiplication, exponentiation, etc. Employing the chain-rule for differentiation, the derivatives of each atomic operation are propagated from the function input to the function output. There are two variants of AD referred to the forward and backwards methods. The forward method is the most easily understood; however, the backwards method is more efficient.

AD is integrated with many programming languages both as freeware as a commercial product [60]. Some variants use a special code compiler [74] where as others capitalize upon language extensions such as operator overloading. In all cases, the objective is that the programmer only needs to worry about coding the function itself and does not have to make any special provisions for the derivative. In this way, even legacy source code can be easily integrated in the solution. For our work, we use the commercial toolkit TOMLAB MATLAB Automatic Differentiation (TOMLAB MAD) [129] which provides AD capability in Matlab.

5.2 Expectation-Maximization for non-Gaussian Parameters

The information filter (IF) solution described in the previous section requires all of the parameters in the state vector $\boldsymbol{\theta}$ to be Gaussian. Therefore, it is able to successfully estimate the unknown high-resolution image $\boldsymbol{x}$ as well as unknown image registration parameters $\{\boldsymbol{s_k}\}$ as long as the hyper-parameters $\{\boldsymbol{\beta_k}\}$ and $\boldsymbol{\alpha}$ are provided by the user. However, as mentioned in 2.4, this is undesirable because it often requires a long parameter-tuning process and can limit the applicability of the solution [67]. In this section, we show that, if non-Gaussian unknowns are present, they may be solved by augmenting the IF methodology using the well-known Expectation-Maximization (EM) technique [69,130,131].

In summary, the EM method allows us to maximize the general distribution

$$\boldsymbol{Pr}[\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{\lambda}], \tag{5.2.1}$$

where $\boldsymbol{Y}$ is a set of inputs, $\boldsymbol{\theta}$ is a set of latent unknowns, and $\boldsymbol{\lambda}$ represents a set of unknown parameters. The difference between the unknowns in $\boldsymbol{\theta}$ and in $\boldsymbol{\lambda}$ is that we will keep track of the full posterior distribution of $\boldsymbol{\theta}$; i.e., $\boldsymbol{Pr}[\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{\lambda}]$. Whereas, we will treat the elements of $\boldsymbol{\lambda}$ as single point unknowns. The EM method breaks the solution into two iterative steps [69]:

E-Step (Expectation)

Evaluate $\boldsymbol{Pr}\left[\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{\lambda}^{\boldsymbol{old}}\right]$ based upon the current estimate of the parameters $\boldsymbol{\lambda}^{\boldsymbol{old}}$.

M-Step (Maximization)

Update the parameters in $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda}^{\boldsymbol{new}} = \boldsymbol{\underset{\lambda}{argmax}}\, \boldsymbol{E_{\theta}}[\boldsymbol{ln}(\boldsymbol{Pr}[\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{\lambda}])]. \tag{5.2.2}$$

In order to apply the EM method to SR, we need to define the vectors $\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{\lambda}$ and the probability function $\boldsymbol{Pr}[\boldsymbol{\theta}, \boldsymbol{Y}|\boldsymbol{\lambda}]$. The vector $\boldsymbol{Y}$ is the set of low-resolution measurements $\{\boldsymbol{y_k}\}$ defined in section 5.1. The vector $\boldsymbol{\theta}$ is the augmented Gaussian state vector containing the unknown high-resolution image $\boldsymbol{x}$ and the set of registration correction parameters $\{\boldsymbol{s_k}\}$ as defined in (5.1.16). The parameter vector $\boldsymbol{\lambda}$ contains the set of likelihood and image prior hyper-parameters, $\{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$, as defined in section 5.1.

We know from prior VBI analysis of the SR problem that the hyper-parameters are optimally modeled as a Gamma distribution [65] which, unfortunately, we can not simply transform the Gamma into an equivalent Gaussian using the pseudo-measurement method described in 5.1. Consequently, we can not directly include $\{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$ into the augmented state vector $\boldsymbol{\theta}$. A complete VBI solution, such as in [65] is able to determine the full posterior distributions of

the hyper-parameters at the expense of analytic complexity. In our simplified solution, using the EM method, we settle for finding point solutions to the non-Gaussian parameters.

Applying the EM method to the SR problem, we first observe that the E-step is exactly the solution given to us by the IF method in 5.1 for a given set of known values for $\{\boldsymbol{\beta_k}\}, \boldsymbol{\alpha}$. Also, recall the algorithm in section 5.1 is naturally iterative due to the Taylor series linearization of the measurement function in (5.1.6). In order to include an update of the hyper-parameters in the iteration cycle, we add the M-step, which is equivalent to solving the minimization of the expectation shown in (5.2.2).

Taylor Series Expansion of the Expectation Function

Before proceeding, it will be useful to derive a general solution to the expectation

$$\boldsymbol{E_{\Delta\theta}[\|v(\theta_0 + \Delta\theta)\|^2]}, \tag{5.2.3}$$

where $\boldsymbol{v}$ is an arbitrary function, $\boldsymbol{\theta_0}$ represents our current best estimate of the unknown $\boldsymbol{\theta}$, and $\boldsymbol{\Delta\theta}$ represents the uncertainty in $\boldsymbol{\theta_0}$. Note, $\boldsymbol{\Delta\theta}$ has a mean of zero and, for the Gaussian case, a covariance of $\boldsymbol{\Omega^{-1}}$. Then, applying a Taylor series expansion to (5.2.3), we get the approximation

$$\boldsymbol{E_{\Delta\theta}[\|v(\theta_0 + \Delta\theta)\|^2] \cong \|v(\theta_0)\|^2 + E_{\Delta\theta}[\Delta\theta^T V^T V \Delta\theta]}, \tag{5.2.4}$$

where $\boldsymbol{V} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{\theta}}$. In the derivation of (5.2.4), we have used the fact that $\boldsymbol{\Delta\theta}$ is zero mean to eliminate any terms containing $\boldsymbol{E_{\Delta\theta}[\Delta\theta]}$. Furthermore, if we assume $\boldsymbol{\Delta\theta}$ has a Gaussian distribution (as it does for the SR problem) and ignore the off-diagonal, cross-correlation elements of $\boldsymbol{\Omega^{-1}}$, we can efficiently compute the expectation on the right-hand-side of (5.2.4) as

$$\boldsymbol{E_{\Delta\theta}[\Delta\theta^T V^T V \Delta\theta] \cong [V.^2]\, diag(\Omega^{-1}),} \tag{5.2.5}$$

where, the nomenclature $[\boldsymbol{V.^2}]$ indicates a matrix formed via squaring each element of $\boldsymbol{V}$ and the function $\boldsymbol{diag}$ converts a square matrix into a column vector containing its diagonal elements.

To solve for each hyper-parameter $\boldsymbol{\beta_k}$, we use the joint distribution function in (5.1.1) to evaluate (5.2.2). In doing so, we are able to simplify the expression by ignoring any terms that do not explicitly depend on $\boldsymbol{\beta_k}$. Substituting (5.1. 1) into (5.2.2) and replacing $\boldsymbol{\lambda}$ by $\boldsymbol{\beta_k}$, we find

$$\boldsymbol{\beta_k^{new} = \underset{\beta_k}{argmax}\left(-\frac{mn}{2} ln(\beta_k) + \frac{\beta_k}{2} E_{\Delta\theta}[\|y_k - B_k(s_k)x\|^2]\right).} \tag{5.2.6}$$

We solve the minimization in (5.2.6) analytically by the standard method of setting the derivative of the expression with respect to $\boldsymbol{\beta_k}$ to zero yielding

$$\boldsymbol{\beta_k^{new} = \frac{mn}{E_{\Delta\theta}[\|y_k - B_k(s_k)x\|^2]}} \quad . \tag{5.2.7}$$

The denominator of (5.2.7) has the form of (5.2.3) and, therefore, we apply the approximation of (5.2.4) to get

$$\boldsymbol{\beta_k^{new} = \frac{mn}{\|y_k - B_k(s_k)x\|^2 + E_{\Delta\theta}[\Delta\theta^T V^T V \Delta\theta]}} \quad , \tag{5.2.8}$$

where

$$\boldsymbol{V = -\left[B_k(s_k) \quad \frac{\delta B_k(s_k)}{\delta s_{k1}} x \quad \cdots \quad \frac{\delta B_k(s_k)}{\delta s_{kp}} x\right].} \tag{5.2.9}$$

In (5.2.9), $\boldsymbol{s_{kj}}$ represents the j'th element of the total p values in the warping vector $\boldsymbol{s_k}$ (the value of $\boldsymbol{p}$ depends upon the specific warping model). Also, as described in section 5.1, we utilize automatic differentiation to obtain the, likely complex, derivatives of the warping matrix $\boldsymbol{B_k(s_k)}$.

Note, once V is determined, we have everything we need to compute $\boldsymbol{\beta_k^{new}}$ from (5.2.7) as we have already derived an efficient method of calculating the expectation in the denominator using (5.2.5).

It is instructive to consider the intuition behind (5.2.8). Based on the model in (5.1.1), $\boldsymbol{\beta_k}$ is the precision of the data likelihood. If we believed we had perfect knowledge of x and s_k, we would calculate $\boldsymbol{\beta_k}$ by its definition as $\boldsymbol{\beta_k} = \frac{\boldsymbol{mn}}{\|\boldsymbol{y_k - B_k(s_k)x}\|^2}$, which is equivalent to (5.2.8) without the expectation term in the denominator. Equivalently, we would get the same result from (5.2.8) if we set the precision of the state vector $\boldsymbol{\Omega^{-1}}$ to 0 (indicating perfect knowledge of $\boldsymbol{x}$ and $\boldsymbol{s_k}$). Therefore, we can interpret the effect of the expectation term in the denominator of (5.2.8) as reducing the precision of the data likelihood due to the current uncertainty in the state.

Following an identical approach, we derive the analytic solution for the image prior hyper-parameter $\boldsymbol{\alpha}$. The solution, which is very similar in form to (5.2.8), is

$$\boldsymbol{\alpha^{new}} = \frac{\boldsymbol{MN}}{\|\boldsymbol{h_p(\theta_0)}\|^2 + \boldsymbol{E_{\Delta\theta}[\Delta\theta^T V^T V \Delta\theta]}} \quad , \tag{5.2.10}$$

where $\boldsymbol{V} = \left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]$. The form of $\boldsymbol{h_p(\theta_0)}$ and $\left[\frac{\boldsymbol{\delta h_p}}{\boldsymbol{\delta\theta}}\right]$ depends upon the choice of image prior model and is given, for the TV and SAR priors by (5.1.18) and (5.1.19) respectively. The same intuition as to the interpretation of the two terms in the denominator of (5.2.10) is the same as described above for (5.2.8).

5.3 Comparison of Information-Filter / Expectation-Maximization Method with Variational Inference

As stated before, our motivation in deriving the IF approach to the SR problem was to create an alternative methodology that avoids the analytic complexity of the VBI solution (particularly when expanding the complexity of the model) while retaining the benefits of Bayesian solutions over the alternative, single-point MAP solutions. In this section we compare our results to a state-of-the-art implementation of the VBI method from [65] released as a Matlab SW package [132].

Both the referenced VBI method as well as our information-filter / expectation-maximization (IFEM) method use a 3 d.o.f. parameterization $\boldsymbol{s_k} = [\boldsymbol{s_{kx}} \quad \boldsymbol{s_{ky}} \quad \boldsymbol{s_{k\phi}}]^{\boldsymbol{T}}$ for image warping. This is sufficient to produce a global, affine transformation allowing for translation in both the x and y axes as well as a rotation $\boldsymbol{\phi}$ about the center of the image. For the IFEM solution, we utilize exact AD to compute the derivatives of $\boldsymbol{B_k}$ with respect to the warping parameters as required in the derivation.

In this section, we use three methods to compare the two variants of the SR algorithm. The first will use a simulated scene from DIRSIG with an embedded resolution target, the second will use a real scene containing a resolution target, and the third will use the spatial frequency metric of chapter 4.

For the DIRSIG comparison, we use a stock airport scene as described in section 2.6 with an embedded pair of Siemens star resolution targets. We use the Siemens star for resolution measurement (see Appendix B) as it is known to be more robust to image enhancement through non-linear processing (such as SR) [49]. The simulated camera is a 640 x 512 pixel LWIR with a bandpass of 7.5 to 13.5 μm, a focal length of 19.0 mm, a pixel pitch of 16.0 μm, and a Gaussian

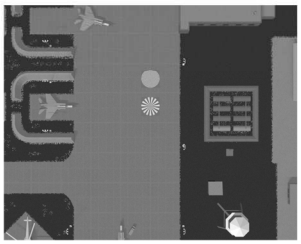
blur with σ = 0.25 pixels. We collect a series of 8 image frames from an altitude of 300 m with horizontal and vertical translations between frames such as to create sub-pixel shifts. We collect the frames with SNRs of 40dB and 20dB.

In Figure 5-1a/b and Figure 5-2a/b, for the SNR = 40 dB and SNR = 20 dB cases respectively, we compare and contrast the SR results. Figure 5-1a and Figure 5-2a show the input, LR airport scene from DIRSIG rendered with our simulated LWIR camera in the upper left. The LR image has dimensions 640 x 512 pixels and is rendered on the page with a width of 5.04 cm (2 inches). In the remaining three panes, we up-sample the LR image by a factor of 2x to create 1280 x 1024 pixel HR images using the non-SR BiCubic method, the state-of-the-art VBI method, and our new IFEM method. The HR images are rendered on the page with a width of 10.08 cm (4 inches).

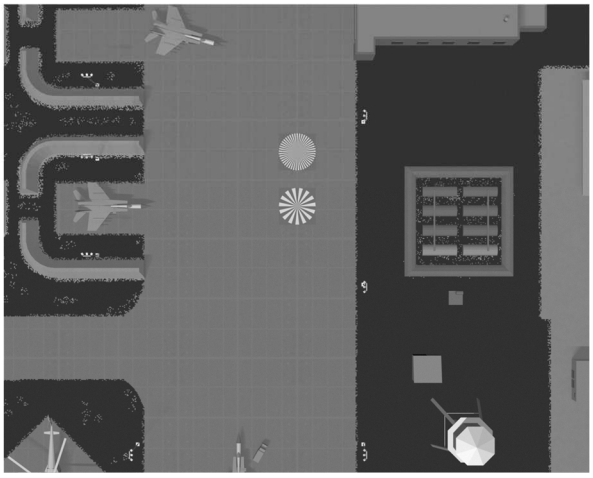
In the full size image, it can be difficult to discern the differences. This is partially due to anti-aliasing filtering (which is effectively a blur filter) that is applied to the images for rendering on the page. In Figure 5-1b and Figure 5-2b, we zoom in on the aircraft on the left of the scene as well as the Siemens star target. From the images, we can observe that the IFEM and VBI methods produce similar sharpening results and that both out-perform the non-SR BiCubic method.

As stated in chapter 4; however, qualitative assessment can be misleading. In order to generate a quantitative comparison, we use the Siemens star resolution targets embedded in the images to measure the MTF of the recovered images and assess if the SR algorithms are properly increasing the resolution by revealing previously aliased spatial frequencies (frequencies above 0.5 cycles / pixel in the LR images). Figure 5-3 shows a comparison of the measured MTF from the two resolution targets for the three algorithms.

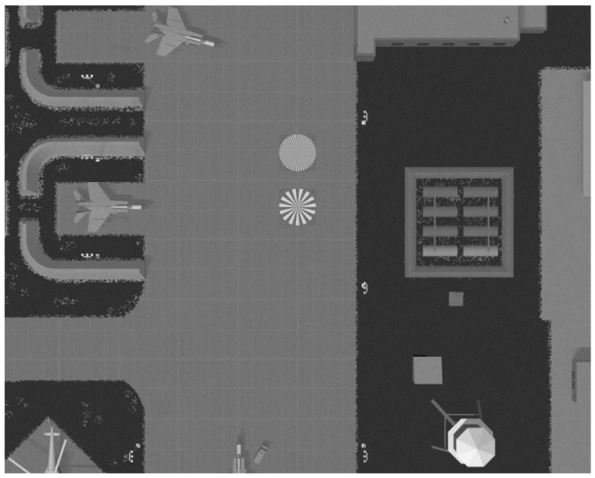
LR Airport Image



IFEM Up-Sampling



BiCubic Up-Sampling



VBI Up-Sampling

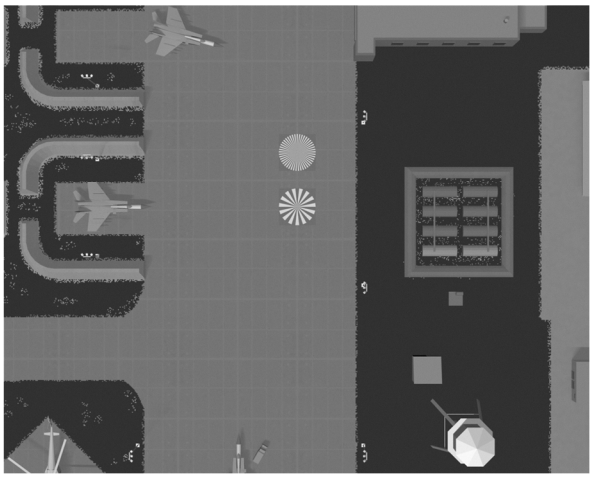


Figure 5-1a: Airport scene from DIRSIG rendered in the LWIR. LR image, with SNR = 40 dB, is up-sampled using BiCubic, IFEM, amd VBI methods

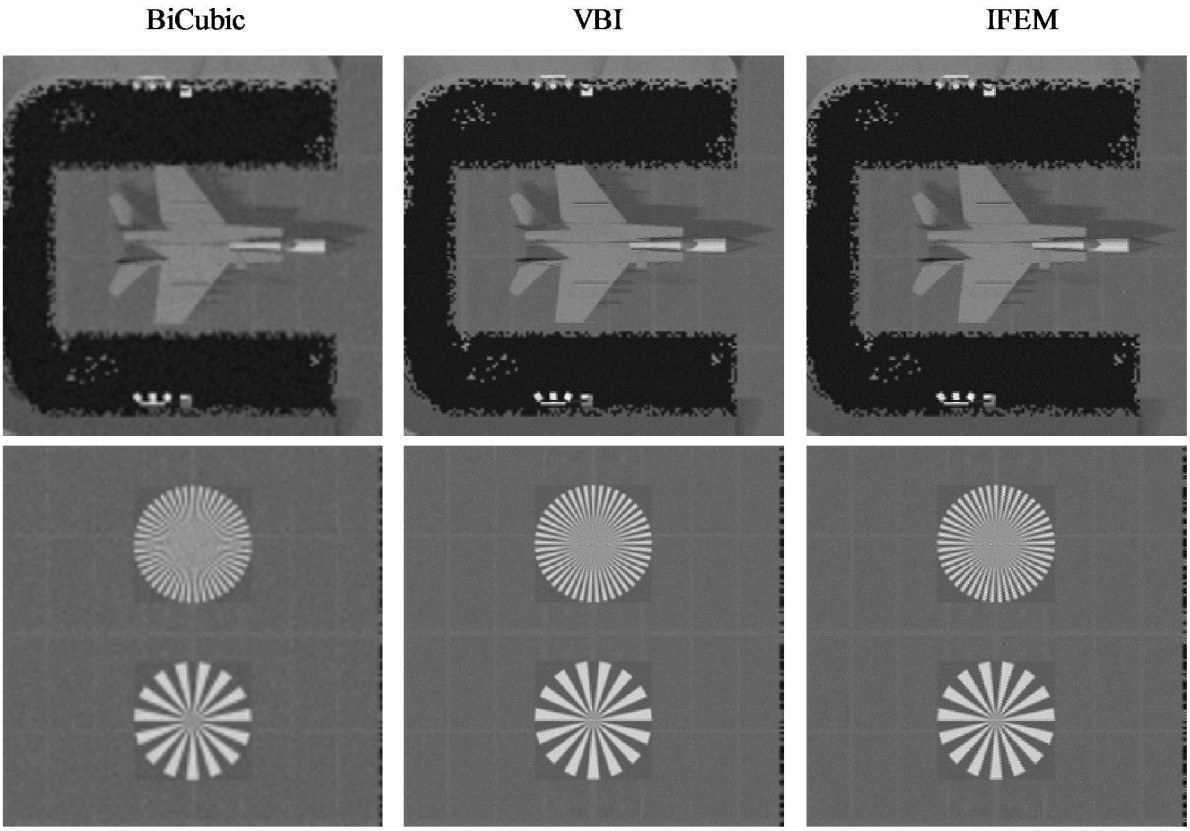


Figure 5-1b: Zooming in on aircraft and Siemens star target for the BiCubic, VBI, and IFEM results. SNR = 40 dB

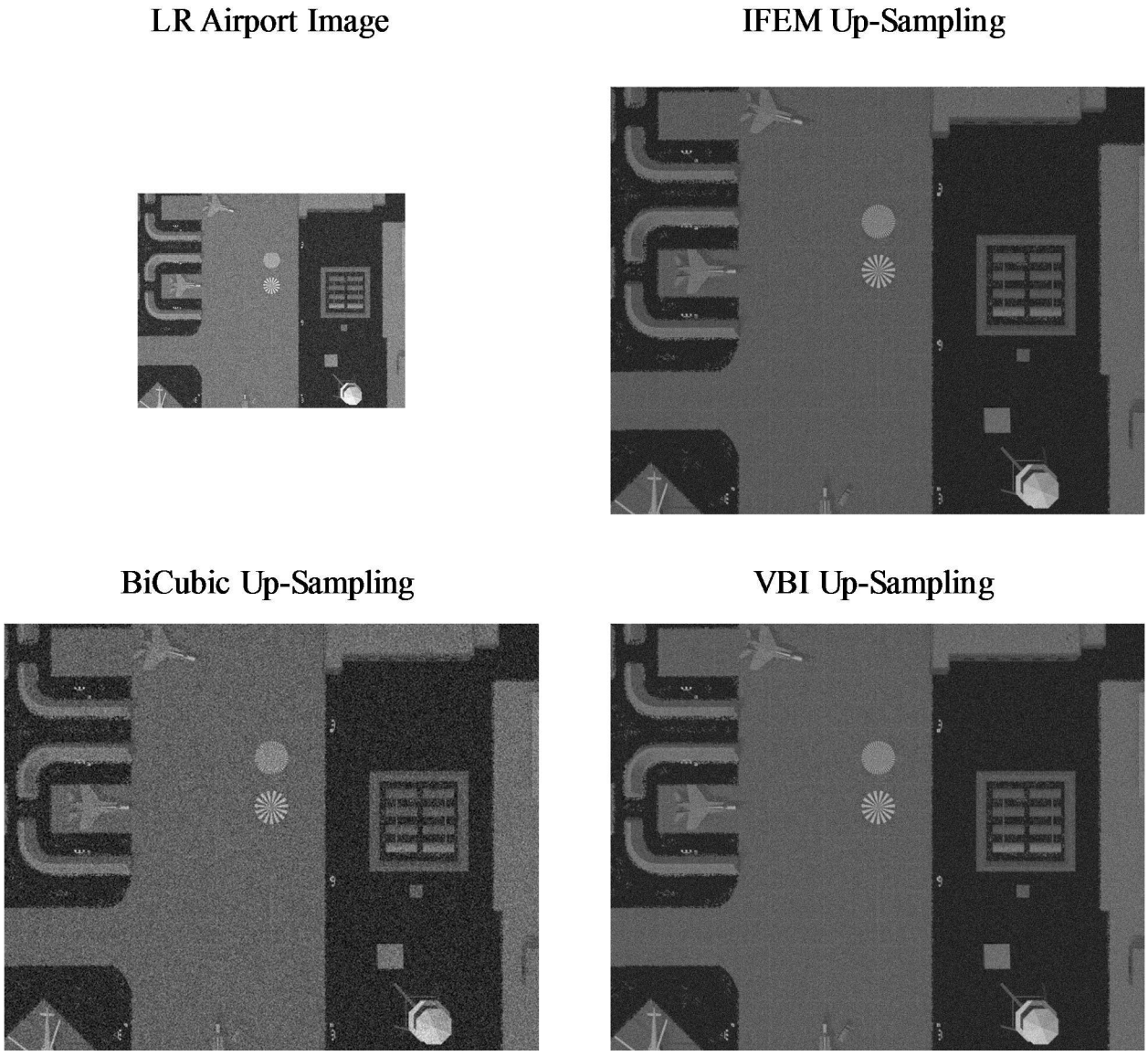


Figure 5-2a: Airport scene from DIRSIG rendered in the LWIR. LR image, with SNR = 20 dB, is up-sampled using BiCubic, IFEM, amd VBI methods.

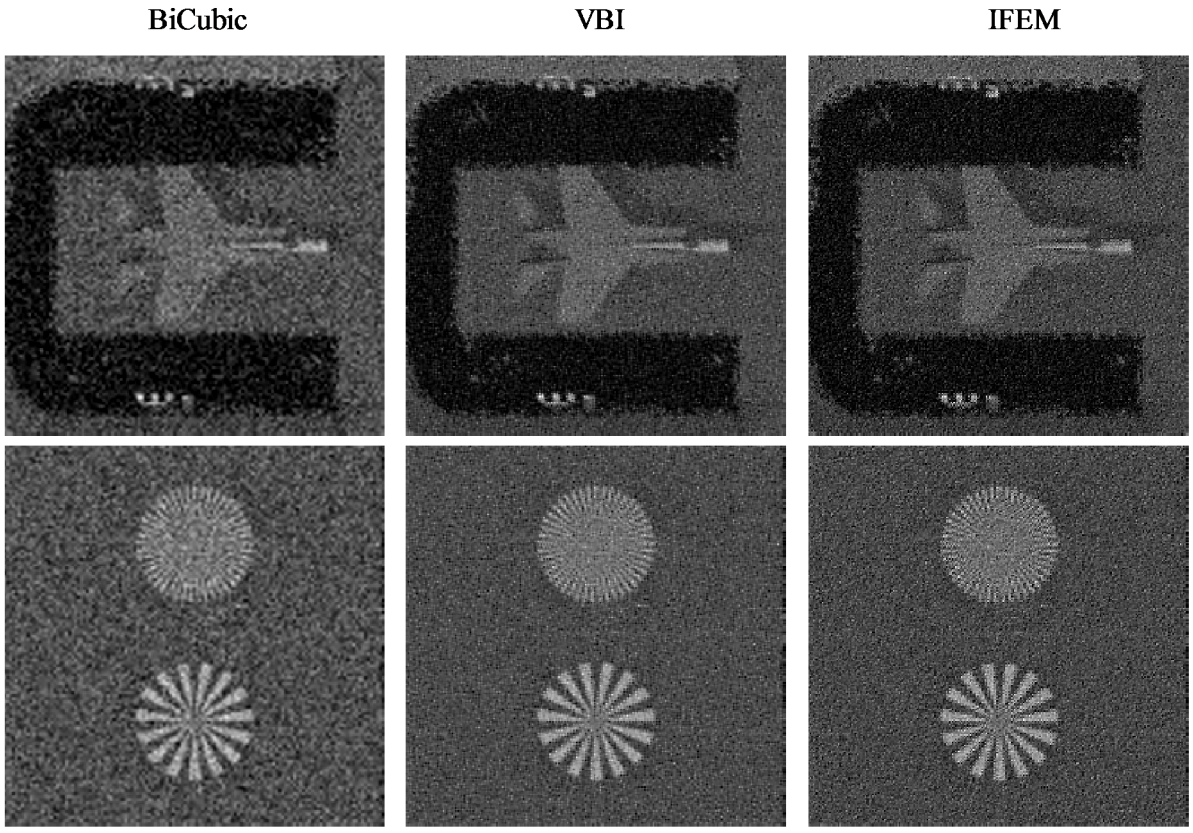


Figure 5-2b: Zooming in on aircraft and Siemens star target for the BiCubic, VBI, and IFEM results. SNR = 20 dB

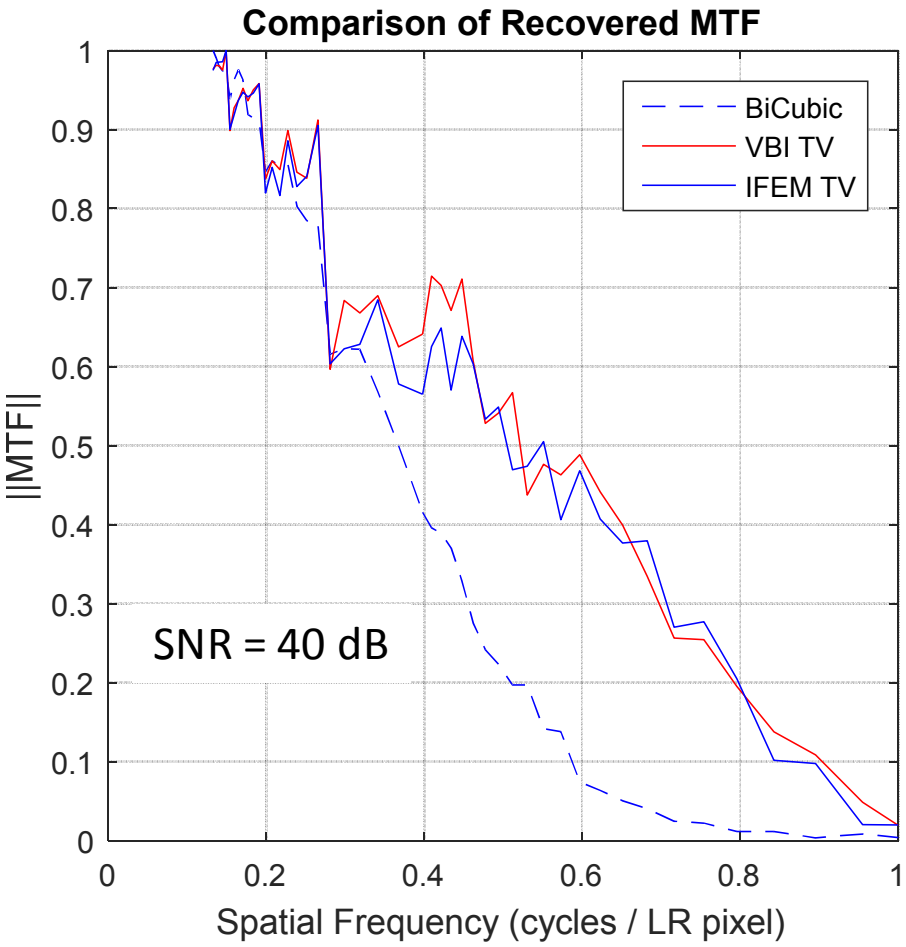


Figure 5-3: MTF comparison between different SR methods based on resolution Siemens star target in DIRSIG imagery at SNR = 40 dB

The MTF measurements are only shown for the SNR = 40 dB case as the extra noise in the SNR = 20 dB case makes direct measurement difficult. The result shows that both the IFEM and VBI solutions perform similarly and recover a significant portion of the aliased frequencies.

The next comparison we make is based on the spatial frequency metric introduced in chapter 4. For this, we create a synthetic bar target scene with a frequency of 5.0 line-pairs / milli-radian (see Figure 5-4). We select a camera with IFOV = 125 micro-radians which leads to a sample frequency of 8 cycles / milli-radian and an $\boldsymbol{\omega_{max}}$ = 4 cycles / milli-radian. We set the

camera's blur as Gaussian with a $\boldsymbol{\sigma}$ = 0.25 pixels. For these settings, the 5 line-pair / milli-radian bar target image will be aliased down to a false spatial frequency of 3 cycles / milli-radian, but should be full recovered with an ideal SR algorithm with a 2x magnification factor that increases $\boldsymbol{\omega_{max}}$ to 8 cycles / milli-radian.

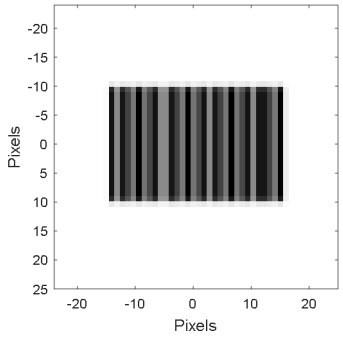


Figure 5-4: Sampled and blurred 5 line-pair / milli-radian bar target

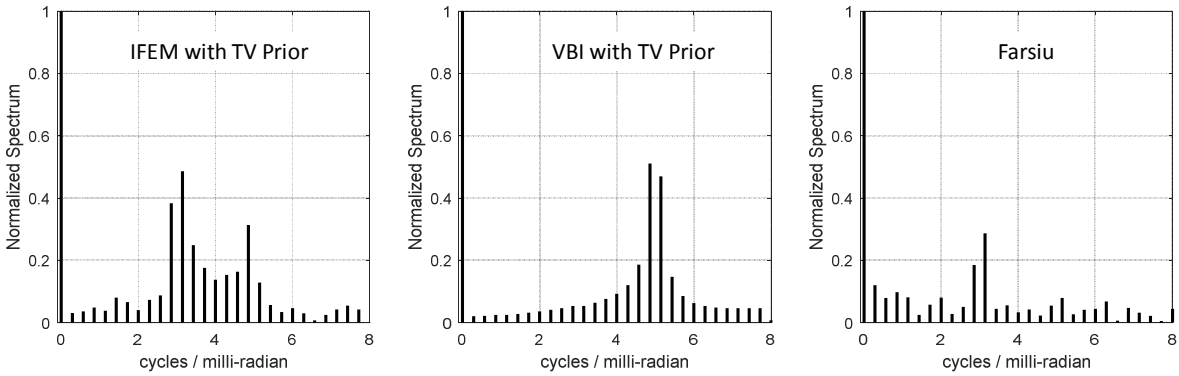


Figure 5-5: Comparison of spatial frequency recovery of three different SR algorithms

For comparison, we run the synthetic bar target through three SR algorithms. The first two are the VBI and IFEM, with TV image prior, as discussed above. The third is the popular algorithm of Faisiu [44] as publicly available in the Open Computer Vision (OpenCV) library [133,134]. As prescribed in the method of chapter 4, we provide each SR algorithm two input images,

horizontally shifted by an ideal 0.5 pixels. The spectra of the resulting SR images are shown in Figure 5-5. The VBI algorithm clearly does the best job of transferring all of the content from the aliased position of 3 cycles / milli-radian to the correct position of 5 cycles / milli-radian. The IFEM algorithm is next in effectiveness. Although it does transfer significant content to the correct position, it still retains a majority of the content at the aliased position. The same trend is true of the Farsiu algorithm, although it transfers significantly less content to the correct spatial frequency position than either of the two Bayesian based algorithms.

Finally, we use real data collected from the Phantom 3 drone (see Appendix A). Figure 5-6 shows an image of a test board used for remote recognition (this will be covered in Chapter 6) with two embedded resolution Siemens star targets. The board was imaged at an altitude of 20 m with the Phantom 3 drone in a hover configuration. Even in a hover, buffeting and limit cycles produced by the flight control system result in sufficient frame to frame motion to accomplish SR.

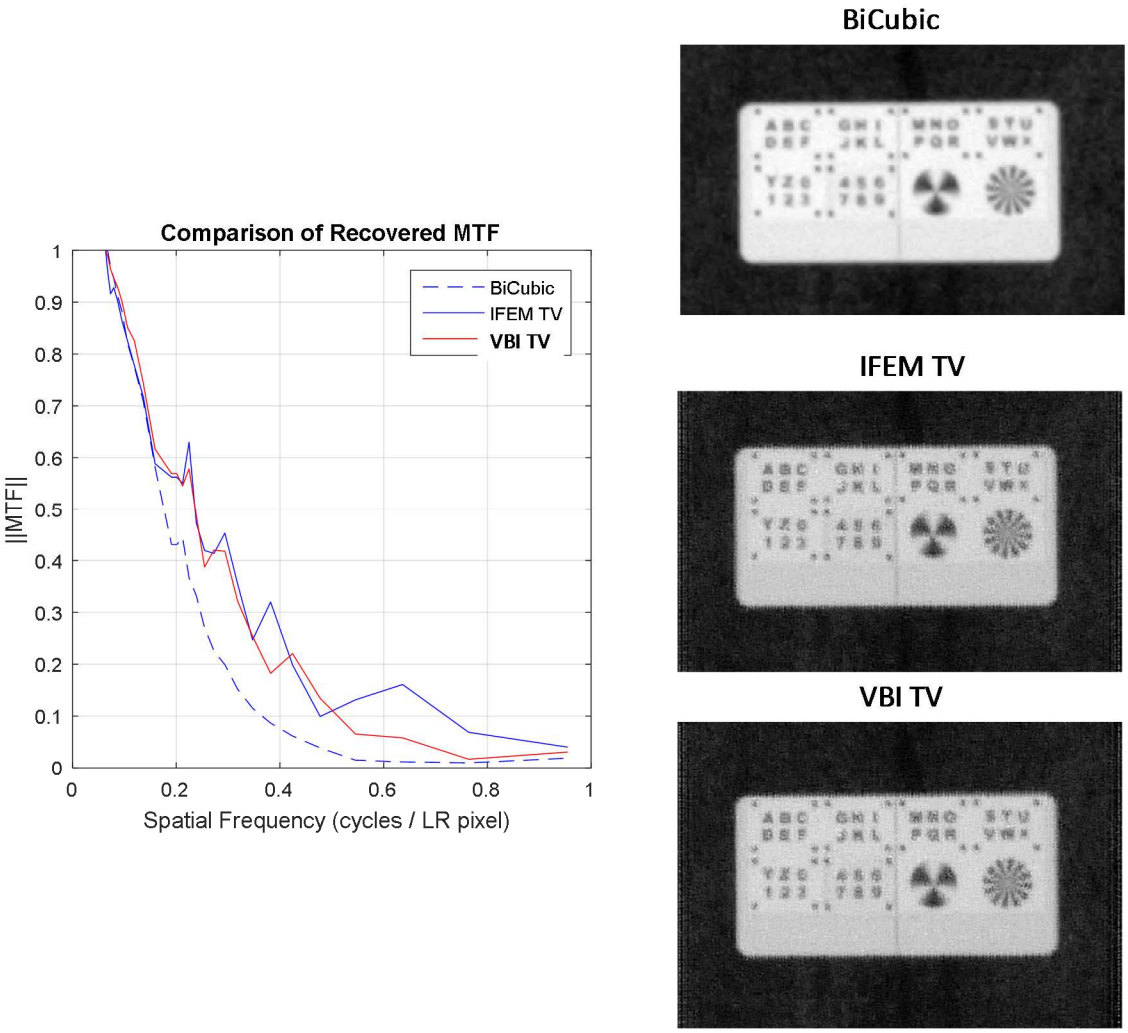


Figure 5-6: MTF measurements based on data collected by the DJI Phantom 3 drone at 20m altitude

Again, a comparison of the VBI and IFEM outputs shows them to be similar and perceptually sharper than the BiCubic output. A quantitative measurement of the recovery of aliased spatial frequencies shows, again, that both SR algorithms performed in a similar fashion.

Based on these results, the IFEM algorithm provides similar benefits to the full VBI method with a simpler mathematical framework. Out of three methods of comparison between the VBI and IFEM algorithms, it is only the spatial frequency method of chapter 4 that exposes, in a quantitative fashion, that the more complex VBI algorithm provides a SR performance benefit;

although, the IFEM algorithm still outperformed the alternative Farsiu algorithm. Consequently, in subsequent sections, we proceed to take advantage of the reduced analytic complexity of the IFEM algorithm and use it as the basis for re-deriving the solution for the more complex image formation forward model associated with oblique viewing geometries.

5.4 Distortion and Blur Recovery Independent of SR

Before proceeding with additional SR development, we look in detail at a couple of degradations typical of WFOV camera used in airborne imagery. Again, due to SWaP and cost requirements, particularly in the case of UAS applications, airborne WFOV cameras may use a simplified, single lens optical design which can produce significant geometric distortion. Geometric distortion is typically also coupled to significant, spatially varying blur. Fortunately, both of these characteristics can be measured for a particular camera serial number through calibration methods as discussed in Appendix B.

A generic distortion model is described in (2.1.12) and (2.1.13) in terms of a function $\boldsymbol{f}(\cdot)$ that maps a position $(\boldsymbol{x}', \boldsymbol{y}')$ on the idealized normalized image plane to a pixel position $(\boldsymbol{u}, \boldsymbol{v})$ on the actual camera; i.e. $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{f}(\boldsymbol{x}', \boldsymbol{y}')$. The functional form and parameters describing the function $\boldsymbol{f}(\cdot)$ are the outcome of the, above mentioned, camera calibration. If the distortion characteristics of the original camera are undesirable (e.g., fish-eye distortion), the image can be mapped to a virtual camera that has a more desirable distortion $\boldsymbol{f_v}(\cdot)$ by mapping pixel locations $(\boldsymbol{u}, \boldsymbol{v})_{\boldsymbol{v}}$ in the virtual camera's image back to pixel locations $(\boldsymbol{u}, \boldsymbol{v})$ in the original camera's image by

$$(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{f}(\boldsymbol{f_v^{-1}}(\boldsymbol{u}, \boldsymbol{v})_{\boldsymbol{v}}). \tag{5.4.1}$$

As the mapped pixel location $(\boldsymbol{u}, \boldsymbol{v})$ will be non-integer, an interpolation method such as bi-linear interpolation is required to map pixel intensity values.

For a typical application, the desired distortion of the output image, $\boldsymbol{f_v}(\cdot)$, is the ideal pin-hole distortion as discussed in 2.1. This corresponds to the perspective projection distortion correction method from [135,136]. No distortion correction method can simultaneously eliminate all distortion induced artifacts, but the perspective projection is robust and close to optimal [1345,136].

In addition to distortion, we have to remove the spatially varying blur from the original camera's image. This step must precede the distortion removal because our calibration measurement of blur will correspond to the camera's raw, distorted image. Once we remap the distortion via (5.2.10), we will alter the blur characteristics. Due to the spatially varying nature of the blur for single lens, WFOV sensors, classic image recovery methods, such as Wiener filtering, which assume a constant and linear blur, are sub-optimal. An alternate technique, well suited for the spatially varying recovery problem, is the Van Cittert iterative technique provided in [137,138]. The Van Cittert iteration equation is given by,

$$\boldsymbol{i_{k+1}} = \boldsymbol{i_k} + \boldsymbol{\alpha\phi}(\boldsymbol{i_k}), \text{ and} \tag{5.4.2A}$$

$$\boldsymbol{\phi}(\boldsymbol{i_k}) = \boldsymbol{g} - \boldsymbol{T}\{\boldsymbol{i_k}\}, \tag{5.4.2B}$$

where $\boldsymbol{g}$ is the original, blurred image, $\boldsymbol{i_k}$ is the recovered image after iteration $\boldsymbol{k}$, $\boldsymbol{\alpha}$ is a learning rate parameter that must be tuned by the user, and $\boldsymbol{T}\{\boldsymbol{i_k}\}$ represents the non-linear and spatially varying transform to apply the known blur to the recovered image.

Evaluation Through Simulation

In order to fully test the image recovery technique described in this section, it is first tested on a known truth image where the blur and distortion models are synthetically applied. The recovery methods are then used to try and recover the original, non-degraded image. The truth image selected is the standard United States Air Force (USAF) bar target for accessing sensor resolution as shown in Figure 5-7a. Figure 5-7b shows the same bar target image with the fish-eye lens distortion and spatially varying blur synthetically applied. The distortion function and blur data used to degrade the image are identical to those measured during calibration of a real, airborne MWIR sensor from FLIR discussed in Appendix A.

Figure 5-8 shows the results of applying the technique above to the degraded image in Figure 5-7a. Figure 5-8a shows the result of only applying distortion inversion from (5.4.1) without any attempt to first remove the spatial varying blur. Comparing Figure 5-8a to Figure 5-7a shows that the distortion inversion process is adversely affected by the presence of blur in the original image. Figure 5-7b shows the result of first applying the blur recovery prior to attempting the distortion inversion. The resulting image is improved considerably. As a comparison, Figure 5-7c shows the result of performing inverse distortion on an image that contained no synthetic blur. Comparing Figure 5-7b to Figure 5-7c illustrates that they are nearly identical. This indicates that the blur recovery step completely eliminated any adverse effects due to blur.

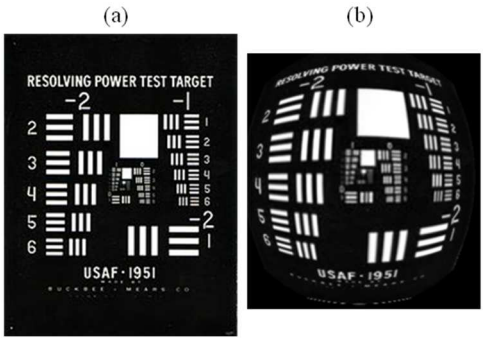
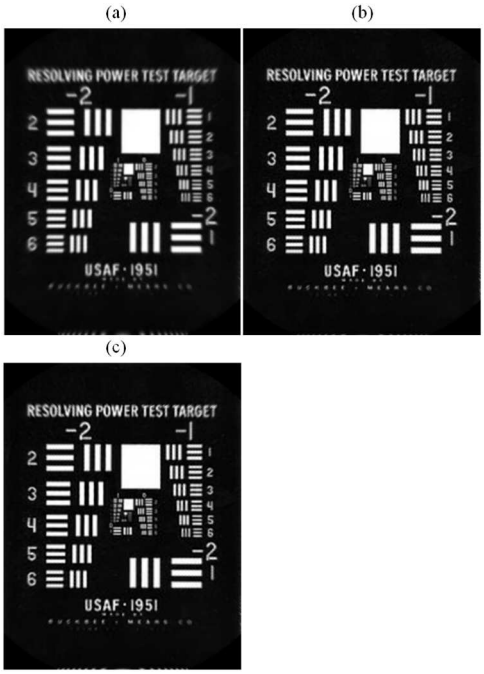


Figure 5-7: US Air Force Bar Target [50]

(a) Original
(b) Simulated Blur and Distortion Degradation



(a) Distortion inversion without blur recovery
(b) Blur recovery applied prior to distortion inversion
(c) Reference: Distortion inversion only of an un-blurred image

Figure 5-8: Recovered images of the bar target

Evaluation on a Real Airborne Image

The simulation based results above provide sufficient confidence in the technique to apply it to the real degraded image in Figure 5-9. This image was collected from the airborne MWIR camera described in Appendix A. Its calibration parameters for distortion and blur are provided by

the manufacturer. Figure 5-9b shows the result of applying only blur recovery to Figure 5-9a using (5.4.2). Figure 5-10 shows the result of applying distortion inversion to Figure 5-9b.

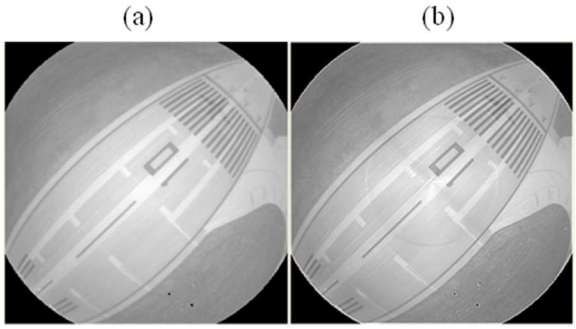


Figure 5-9: Blur recovery

(a) Original Image
(b) Blur recovery

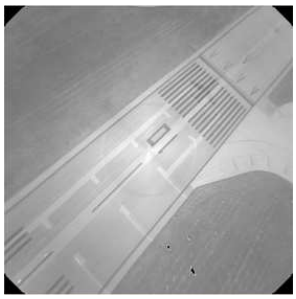


Figure 5-10: Inverse distortion applied after blur recovery

The results in this section show an effective technique, independent of SR, to correct the degradations of distortion and spatially-varying blur typically seen in WFOV images from airborne cameras. These techniques may be used in applications where additional resolution enhancement is not required. In the subsequent section, we will develop a solution to jointly remove these artifacts along with SR.

5.5 Oblique Viewing Geometries and Mosaic Projection

Thus far, we have been considering the 3 d.o.f. affine warping model present in [65]. This model works well for cameras with a relatively narrow field of view and a viewing geometry for which the camera's viewing plane is roughly parallel to the ground plane. However, either or both of these assumptions may be violated from an airborne sensor's perspective. In Figure 5-11, for example, the observer on the right would meet the narrow field of view and parallel viewing geometry assumptions. Consequently, the 3 d.o.f. warping model is likely adequate. However, the observer on the left is viewing the ground with a more oblique geometry. In this later case, the truly 6 d.o.f. rigid motion of the observer (3 translation d.o.f. and 3 rotational d.o.f.) will induce a more complex variation in the perceived pixel motion of the captured images. There will also be a significant difference in perceived pixel motion at different angles across the field of view. However, all of these geometric effects are deterministic and predictable. In order to integrate them into the image formation model of (2.4.1), the warping vector s_k must be expanded to contain the full 6 d.o.f. motion of the observer; i.e., $\boldsymbol{s_k} = [\boldsymbol{s_x} \quad \boldsymbol{s_y} \quad \boldsymbol{s_z} \quad \boldsymbol{s_{\theta x}} \quad \boldsymbol{s_{\theta y}} \quad \boldsymbol{s_{\theta z}}]^{\boldsymbol{T}}$ where $\boldsymbol{x, y, z}$ represent the translational motion of the observer relative to an appropriate world coordinate system and $\boldsymbol{\theta x, \theta y, \theta z}$ represent the three Euler rotations [101] of the observer relative to the same world coordinate system.

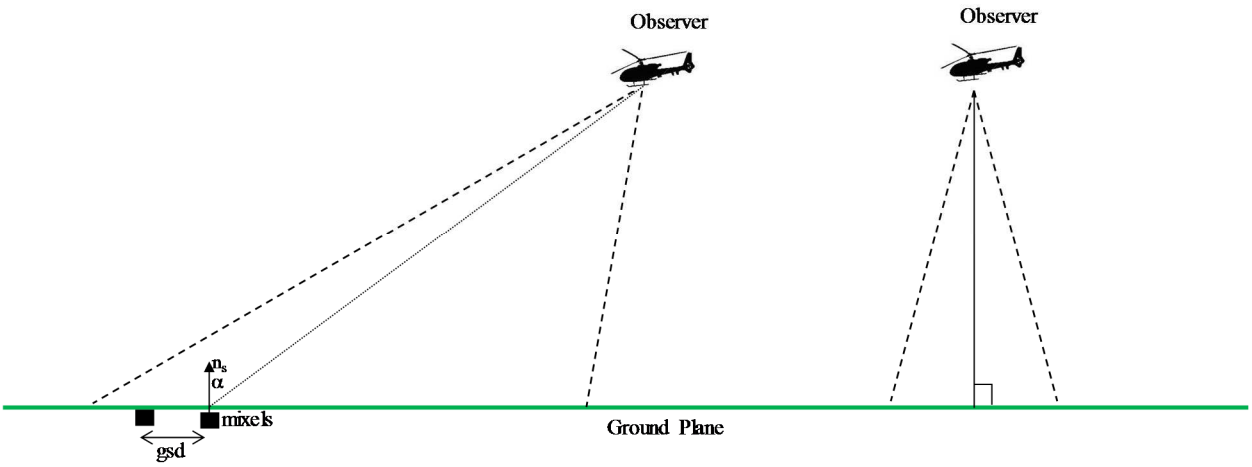


Figure 5-11: Illustration of oblique viewing geometry

The depiction in Figure 5-11 suggest two different interpretations of the SR problem. The first is the classic interpretation where, based on the sequence of captured LR images, we infer an HR image that would have been captured from a higher-resolution camera in the same geometry. With the exception of the expanded 3 d.o.f to 6 d.o.f. warping model, this is the identical problem we examined in the previous sections. The second interpretation is that the HR image we wish to recover is fixed to the ground and sampled in object space at a regular ground sample distance (gsd). This second interpretation is the mosaicking problem. In keeping with prior work [36-38], we refer to the discrete samples of the ground image as "mixels" to distinguish them from "pixels".

Modification of the Image Formation Model for Mosaics

For the mosaicing problem, we have to resort to a more complex image formation model which borrows from both 3D computer graphics [83-85,89] and radiometry [56]. Unlike the classic SR problem, there is no longer a fixed up-sampling ratio between the pixel density in the LR image and the ground sample density in the HR mosaic. Instead, the ratio increases proportionally to the secant of the angle of incidence, α, (see Figure 5-11) which is the angle between the local surface

normal, $\mathbf{n_s}$, and the projected ray of each individual LR pixel. If we assume that, from the camera's perspective, we are far enough away and the gsd is small enough that the spatial extend of each mixel is less than an LR pixel, the irradiance reaching camera from each mixel, using a Lambertian assumption for the surface [56], is

$$\boldsymbol{H = \frac{1}{R^2} L_{mixel} gsd^2 cos\alpha cos\beta,} \tag{5.5.1}$$

where $\boldsymbol{L_{mixel}}$ is the radiance of the mixel (including both emissive and reflected components), $\boldsymbol{\alpha}$ is the angle of incidence defined above, $\boldsymbol{\beta}$ is the angle off-boresight of the camera, and $\boldsymbol{R}$ is the slant range distance between the camera and the mixel. Note, for the flat surface geometry shown in Figure 5-11, if we define the ground plane to correspond to the plane $\mathbf{z} = 0$, then we can write $\boldsymbol{R = Z_C^W / cos\alpha}$, where $\boldsymbol{Z_C^W}$ is the vertical position of the observer in the world coordinate system.

In general, the camera will be pre-calibrated such that the pixel counts are proportional to $\boldsymbol{H/cos\beta}$. That is, the signal loss or "shading" due to off-boresight angles is already accounted for by calibration. We will lump the proportionality constant of the camera's response as well as any other scale factors we apply during the acquisition process (e.g., normalizing the maximum output to 1) into a single gain term $\boldsymbol{G_{resp}}$. Note, if we have a more accurate model of the camera's physical conversion process from irradiance to pixel counts, we can easily add that information into our image formation model. Again, because we are using exact automatic differentiation in our IFEM solution, we will get any derivatives associated with this more complex model for free.

For implementation, we will define the $\boldsymbol{(M\ x\ N)}$ mixel image, $\boldsymbol{x}$, such that

$$\boldsymbol{x_{ij} = \frac{1}{\bar{z}^2} G_{resp} L_{ij} gsd^2,} \tag{5.5.2}$$

where $\boldsymbol{i, j}$ are the 2D row and column indices of the mixel and $\bar{\boldsymbol{z}}$ is the mean observer height, $\boldsymbol{s_{zk}}$, for the K, LR image frames. This definition provides an intuitive and convenient scaling such that the numerical values of the mixels in the image $\boldsymbol{x}$ are consistent with the numerical values of the pixels in the LR images $\boldsymbol{y_k}$. The relationship is exact if the mixel were to be imaged directly below the observer; i.e., with $\boldsymbol{\alpha} = 0$.

Although we have complicated the image formation model, the Bayesian model (5.1.1) still applies without modification. All of our alterations to the image formation process are contained within the warping matrix $\boldsymbol{B_k(s_k)}$ and the expansion of s_k from a 3 d.o.f. representation to a 6 d.o.f. representation. As we will be computing the matrices $\frac{\boldsymbol{\delta B_k(s_k)}}{\boldsymbol{\delta s_{kj}}}$, for use in equation (5.2.9), via exact automatic differentiation, the derivation of the IFEM solution for the mosaic case is essentially unchanged from that of the 3 d.o.f. affine case. This would not be the case if we attempted to reformulate the variational Bayesian inference solution for the mosaic case.

We do still have to properly formulate $\boldsymbol{B_k(s_k)}$ as the break-out in equation (2.4.1) no longer applies (there is no longer a fixed down-sampling matrix $\boldsymbol{A}$). Instead, we simultaneously model the effects of sampling and blur for LR image $\boldsymbol{y_k}$ by first projecting the mixel $\boldsymbol{x_{ij}}$ to a sub-pixel location in the LR image and then using the blur model to determine the contributions of $\boldsymbol{x_{ij}}$ to the surrounding LR pixels (see Figure 5-12). The projection of each mixel $\boldsymbol{x_{ij}}$ to the $(\boldsymbol{m\ x\ n})$ LR image results in the creation of a single column of the $\boldsymbol{(mn)}$ x $\boldsymbol{(MN)}$ matrix $\boldsymbol{B_k(s_k, \sigma_{kh})}$, where we have added a functional dependence on the sigma parameter of the Gaussian blur (we will ultimately augment our filter to simultaneously estimate $\{\boldsymbol{\sigma_{kh}}\}$ along with $\boldsymbol{x}$ and $\{\boldsymbol{s_k}\}$).

We solve the projection problem based on the geometric model described in 2.1. We first determine the vector of the mixel $\boldsymbol{x_{ij}}$ relative to the sensor in the world coordinate system as

$$\boldsymbol{R_{M/S}^{W}} = \begin{pmatrix} \boldsymbol{X_{M/S}^{W}} \\ \boldsymbol{Y_{M/S}^{W}} \\ \boldsymbol{Z_{M/S}^{W}} \end{pmatrix} = \boldsymbol{gsd} \begin{pmatrix} \boldsymbol{j - j_0} \\ \boldsymbol{i - i_0} \\ \boldsymbol{0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{X_S^W + sx_k} \\ \boldsymbol{Y_S^W + sy_k} \\ \boldsymbol{Z_S^W + sz_k} \end{pmatrix}, \tag{5.5.3}$$

where, $\boldsymbol{i_0}$ and $\boldsymbol{j_0}$ represent the center of the mixel array; i.e., the point that corresponds to $[0\ 0\ 0]^T$ in world coordinates. $[\boldsymbol{X_S^W} \quad \boldsymbol{Y_S^W} \quad \boldsymbol{Z_S^W}]^{\boldsymbol{T}}$ is our initial estimate of the observer's 3D position from, likely, an inertial navigation system as discussed in 2.5 and $[\boldsymbol{sx_k} \quad \boldsymbol{sy_k} \quad \boldsymbol{sz_k}]^{\boldsymbol{T}}$ are the position corrections from the first three elements of the vector $\boldsymbol{s_k}$. We then rotate the vector into the camera's coordinate system

$$\boldsymbol{R_{M/S}^{C}} = \begin{pmatrix} \boldsymbol{X_{M/S}^{C}} \\ \boldsymbol{Y_{M/S}^{C}} \\ \boldsymbol{Z_{M/S}^{C}} \end{pmatrix} = \left[\boldsymbol{T_W^C}\right] \boldsymbol{R_{M/S}^{W}}, \tag{5.5.4}$$

where $\left[\boldsymbol{T_W^C}\right]$ is the DCM constructed from our initial estimate of the camera pose Euler angles corrected by the last three elements of $\boldsymbol{s_k}$; i.e., $[\boldsymbol{\theta x + s_{\theta x}} \quad \boldsymbol{\theta y + s_{\theta y}} \quad \boldsymbol{\theta z + s_{\theta z}}]^{\boldsymbol{T}}$. In computing $\left[\boldsymbol{T_W^C}\right]$, we must ensure that we have a correct understanding of the specific Euler convention. Once computing $\boldsymbol{R_{M/S}^{C}}$, we can compute the projection into the normalized image plane, as described in 2.1, as

$$\begin{pmatrix} \boldsymbol{u'} \\ \boldsymbol{v'} \end{pmatrix} = \frac{1}{\boldsymbol{Z_{M/S}^{C}}} \begin{pmatrix} \boldsymbol{X_{M/S}^{C}} \\ \boldsymbol{Y_{M/S}^{C}} \end{pmatrix}. \tag{5.5.5}$$

Once we have the normalized projection, we use the inverse of the camera distortion function, which we have determined through calibration, to get the final, sub-pixel projection location $(\boldsymbol{u, v})$.

Given the center of projection of the mixel $\boldsymbol{x_{ij}}$, we use the Gaussian blur assumption to compute its contribution to each of the neighboring LR pixels as shown in Figure 5-12.

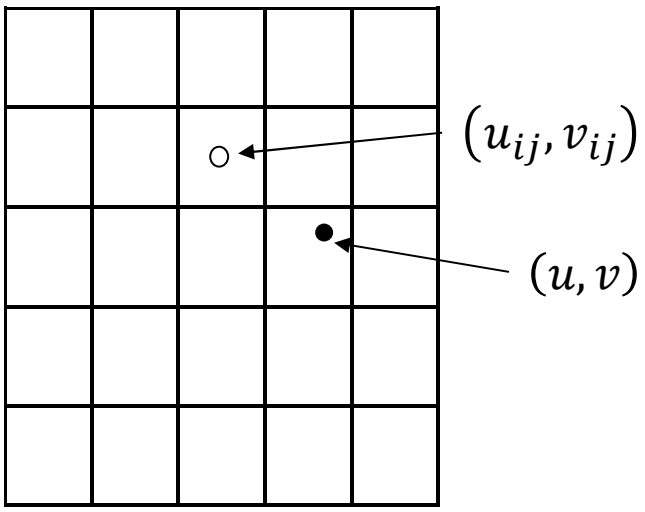


Figure 5-12: Computation of the contribution of mixel x_{ij} to LR pixel ij

If the LR pixel indexed by $\boldsymbol{ij}$ is centered at pixel position $(\boldsymbol{u_{ij}, v_{ij}})$, then the contribution from mixel $\boldsymbol{x_{ij}}$ which projects to LR location $(\boldsymbol{u, v})$ is given by

$$\boldsymbol{y_{kij}(x_{ij}) = \left[\frac{\bar{z}^2}{\left(Z^W_{M/C}\right)^2} x_{ij} cos^3\alpha\right] \frac{1}{2\pi\sigma^2_{hk}} \left[\int_{-0.5}^{0.5} exp\left(-\frac{1}{2\sigma^2_{hk}}(u_{ij} + \zeta - u)^2\right) d\zeta\right] \left[\int_{-0.5}^{0.5} exp\left(-\frac{1}{2\sigma^2_{hk}}(v_{ij} + \zeta - v)^2\right) d\zeta\right],} \quad (5.5.6)$$

where we have used the results of (5.5.2), the relationship $\boldsymbol{R = Z^W_C / cos\alpha}$ for a flat ground plane at $\boldsymbol{Z^W_M}$ = 0, and the seperability property of the Gaussian distribution. The integrals in (5.5.6) may be written and computed efficiently in terms of the Gaussian error function, $\boldsymbol{erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt}$, to simplify the expression to

$$\boldsymbol{y_{kij}(x_{ij}) = \frac{1}{4}\left[\frac{\bar{z}^2}{\left(Z^W_{M/C}\right)^2} x_{ij} cos^3\alpha\right] \left[erf\left(\frac{u_{ij}+0.5-u}{\sqrt{2}\sigma^2_{hk}}\right) - erf\left(\frac{u_{ij}-0.5-u}{\sqrt{2}\sigma^2_{hk}}\right)\right] \left[erf\left(\frac{v_{ij}+0.5-v}{\sqrt{2}\sigma^2_{hk}}\right) - erf\left(\frac{v_{ij}-0.5-v}{\sqrt{2}\sigma^2_{hk}}\right)\right].} \quad (5.5.7)$$

Equation (5.5.7) is used to set each column of the $\boldsymbol{B_k(s_k, \sigma_{kh})}$ matrix. Additionally, TOMLAB [129] as well as most implementations of exact automatic differentiation support the Gaussian error function so that all of the partial derivatives $\frac{\delta B_k(s_k,\sigma_{hk})}{\delta s_{kj}}$ and $\frac{\delta B_k(s_k,\sigma_{hk})}{\delta \sigma_{hk}}$, as needed for the implementation of the IFEM filter, are available.

The final step in the derivation is to expand the augmented state vector in (5.1.16) to include the blur parameters; that is

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{x} \\ \{\boldsymbol{s_k}\} \\ \{\boldsymbol{\sigma_{hk}}\} \end{bmatrix}. \tag{5.5.8}$$

Results

In order to test the mosaic solution, we again use synthetic imagery from DIGSIG. We use the same scene as used in 5.3; however, we pitch the camera up by 30 degrees (see Figure 5-13). In order to maintain the resolution Siemens star target at the center of the image, we also translate the camera 170m along the negative y-axis. For the recovery, we have to define the parameters of the mosaic. We solve for a $\boldsymbol{M}$=512 x $\boldsymbol{N}$=800 mixel mosaic with a gsd of 0.20m and centered just below the location of the resolution target. Based on the relevant parameters of the simulated camera (altitude=300m, focal length = 19mm, pixel pitch = 16μm), the selected gsd requires an effective SR up sampling of between 1x to 2x throughout the image (with points closer to the horizon requiring a larger up sample factor). This produces the result in Figure 5-14 and Figure 5-15. Again, we supplement the qualitative comparison with a quantitative measurement of the recovered MTF based on the embedded resolution targets as shown in Figure 5-16.

Being iterative and non-linear in nature, the IFEM mosaic algorithm has to be initialized with a best guess as to the value of the parameters. The image $\boldsymbol{x}$ is initialized using the set of LR images $\{\boldsymbol{y_k}\}$ and a simplified nearest-neighbor inversion of the geometric projection model. The initial estimate for $\boldsymbol{x}$ is referenced in Figures 5-14, 5-15, and 5-16 as "initial estimate". The registration parameters $\{\boldsymbol{s_k}\}$ are initialized using an external inertial sensor as described in 2.5 (DIRSIG includes a model of an external inertial sensor). The image blur $\{\boldsymbol{\sigma_k}\}$ is initialized using calibration information for the sensor. The hyper-parameters are initialized to their maximum likelihood values which come from (5.2.8) and (5.2.10) without the expectation term in the denominator; i.e. $\boldsymbol{\beta_k} = \frac{\boldsymbol{mn}}{\|\boldsymbol{y_k} - \boldsymbol{B_k}(\boldsymbol{s_k})\boldsymbol{x}\|^2}$ and $\boldsymbol{\alpha} = \frac{\boldsymbol{MN}}{\|\boldsymbol{h_p}(\boldsymbol{\theta})\|^2}$, where the form of $\boldsymbol{h_p}(\boldsymbol{\theta})$ depends upon the choice of prior.

LR Image
Viewed with 30 deg Camera Pitch
640 x 512 Image
SNR = 40 dB

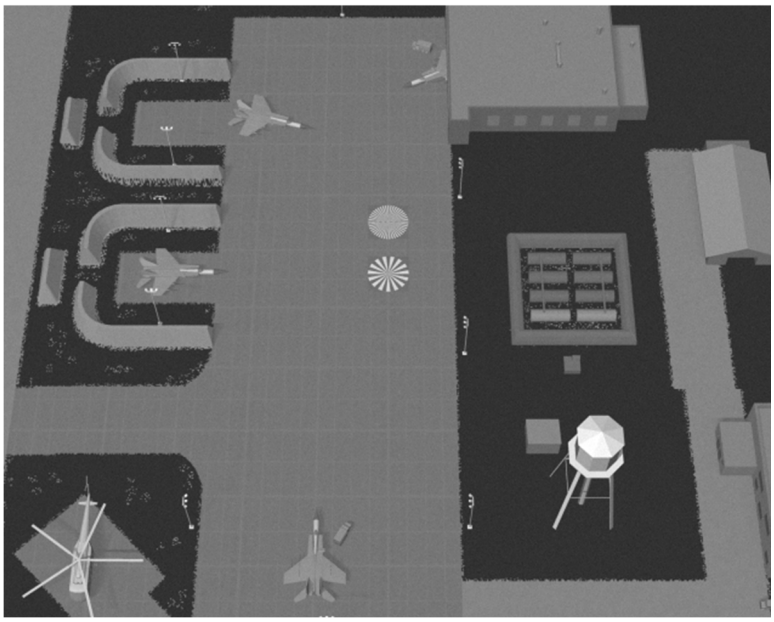


Figure 5-13: DIRSIG rendered image with camera boresight angled 30 degrees from vertical

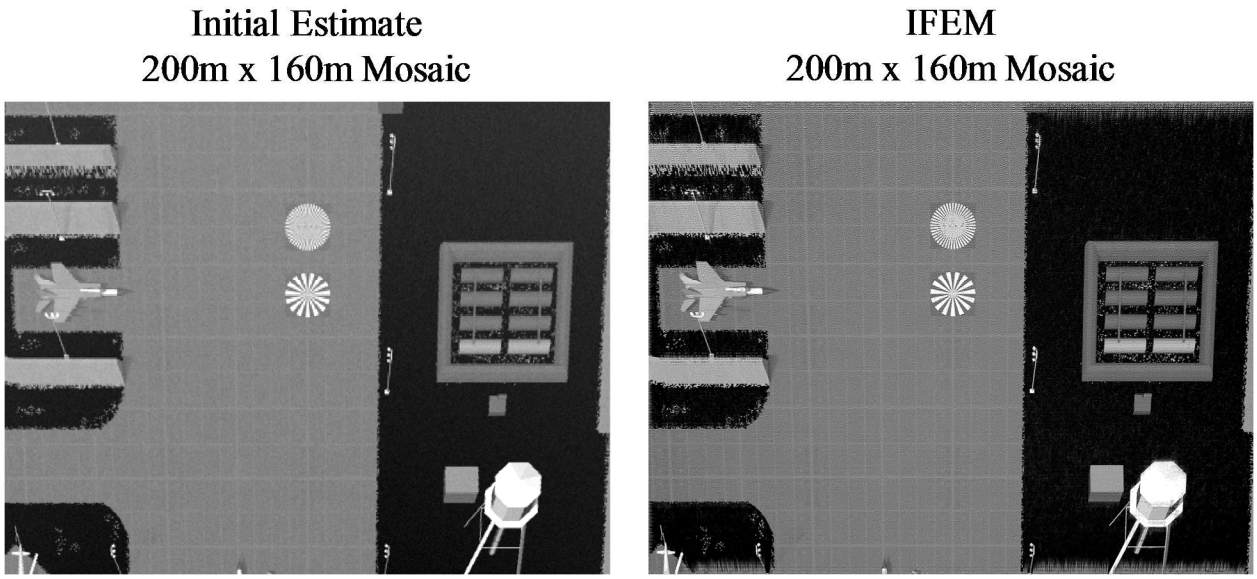


Figure 5-14: Initial mosaic estimate (left) and final IFEM mosaic solution (right)

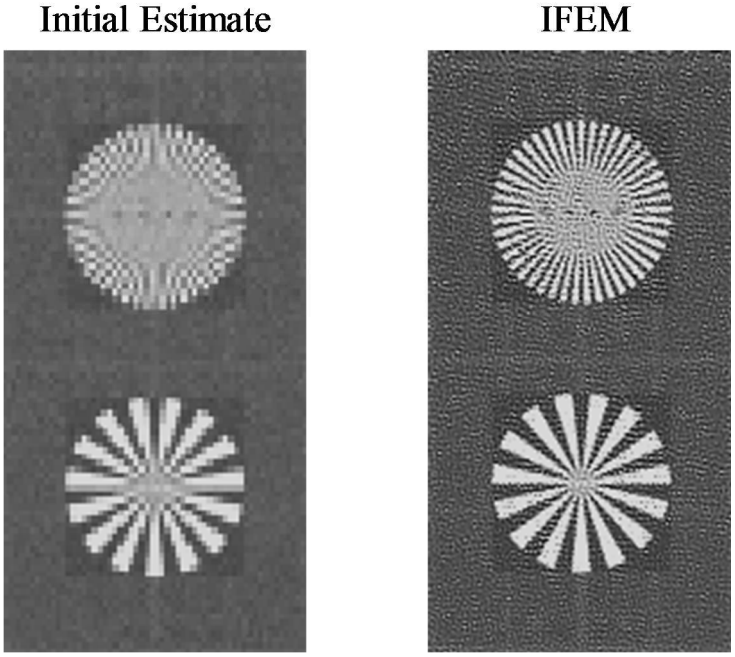


Figure 5-15: Comparison of Siemens star target in initial estimate (left) and final IFEM solution (right)

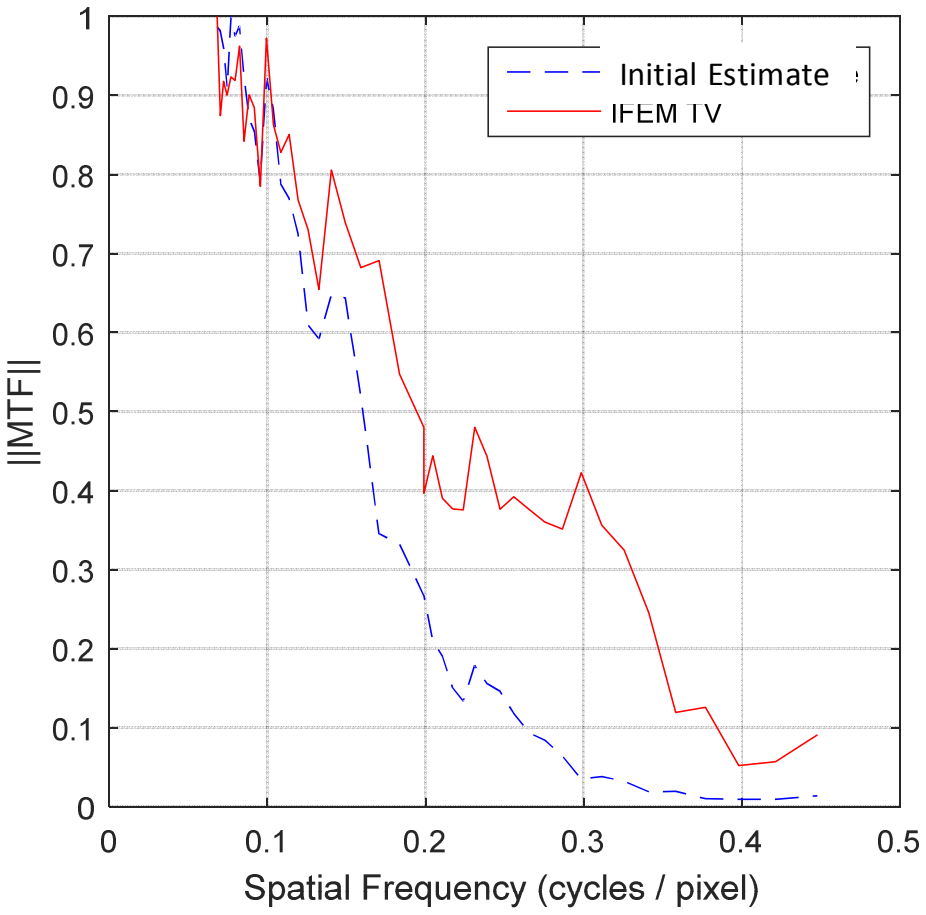


Figure 5-16: MTF estimate of the initial estimate and recovered mosaic image

5.6 Airborne Mosaicing In the Mid-Wave Infrared Domain

This experiment demonstrates the benefits of our IFEM algorithm with 6 d.o.f. motion model on a video sequence collected from a FLIR Systems [139] Mid-Wave Infrared camera rigidly mounted on the bottom of an aircraft. The camera outputs uncompressed video data at a rate of 30 Hz. The video data is time synchronized to measurements from the aircraft inertial navigation system (INS) via the Inter-Range Instrumentation Group (IRIG) IRIG-B timecode. Alignment information for the camera, INS, as well as intrinsic camera parameters such as center pixel, distortion, and MTF are supplied by the manufacturer so that they can be incorporated into the forward image formation model. The camera is focused to provide an MTF modeled by a Gaussian with $\sigma = 0.4$ pixels. Note, this is a significantly tighter blur relative to the pixel size than typically seen with visual cameras (see Appendix B). As discussed in 2.4, this property, which is favorable to successful SR, is unique to IR cameras due to the difficulty in fabricating large pixel format arrays.

Figure 5-17 shows a raw image frame from the camera captured while flying approximately 300 meters above the runway of Birmingham, AL. Figure 5-18 shows a Google Earth plot of the same region. The camera has a wide field-of-view, similar to the 95 degree FOV of the Phantom 3 quadcopter. Figure 1 shows that the camera possesses significant barrel distortion (the straight runway appears curved) caused by the challenge typical for wide field of view (WFOV) infrared cameras when trying to compress a wide viewing angle into a relatively small pixel format. For this MWIR camera, the pixel format is 320 x 320 pixels. The figure also shows that the distortion causes the image to be projected as a conic, as opposed to square.

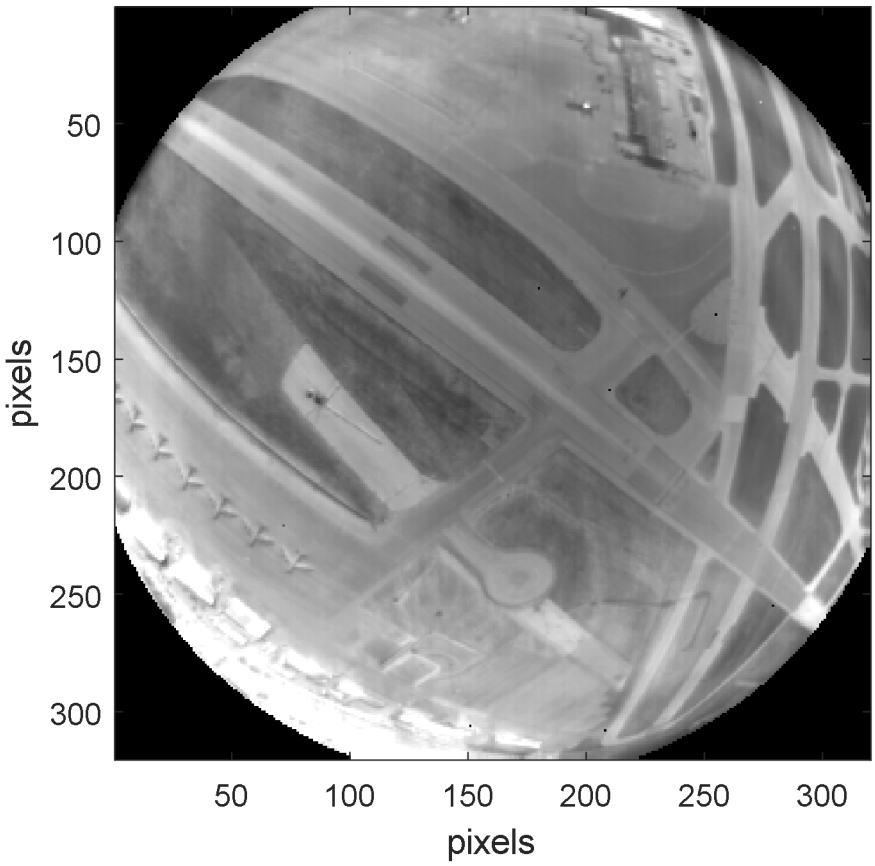


Figure 5-17: A single raw frame captured by the MWIR camera

Figure 5-18: Google Earth map of the imaged area, Birmingham, AL. Aircraft was positioned over north-east / north-west runway heading north-east

Using the SR mosaic version of the IFEM algorithm from 5.3, we use a sequence of 8 video frames to generate a 640m x 640m super-resolved ground map with ground sample distance (gsd) = 1.0 m. At the altitude of 300 meters, this corresponds to ~2x resolution increase for pixels near the center of the raw image increasing to ~4x for pixels near the edge. Figure 5-19a shows the result of the initial estimate of the mosaic using a simple nearest neighbor inversion of the image formation model. This is also the image used to initialize the iterative SR algorithm. We found the best results using the Simultaneous Auto-Regressive (SAR) prior, which are shown Figure 5-19b.

Time-synchronized INS data, including position, velocity, and attitude, for each captured video frame is also used to set the initial estimate of the 6 d.o.f. warping parameter vector $\{\boldsymbol{s_k}\}$. We set the initial error distribution on the parameters in $\{\boldsymbol{s_k}\}$ measured from the INS as Gaussian with a 1 meter, 1-sigma error in each of the three position estimates and a 1 degree, 1-sigma error in each of the three attitude estimates. Refinements to both $\{\boldsymbol{s_k}\}$ as well as to the camera's blur (initialized with the manufacturers value of $\boldsymbol{\sigma}$ = 0.4 pixels) are determined simultaneously with the output mosaic image $\boldsymbol{x}$.

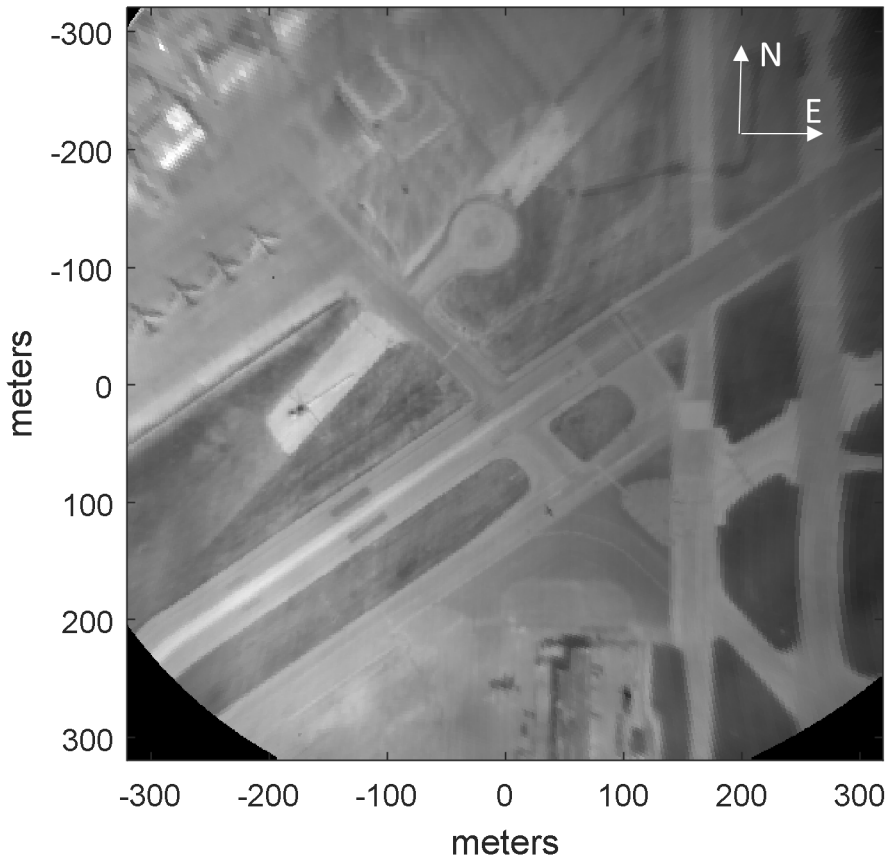


Figure 5-19a: Initial mosaic image estimate using a nearest neighbor method of inverting the image formation model

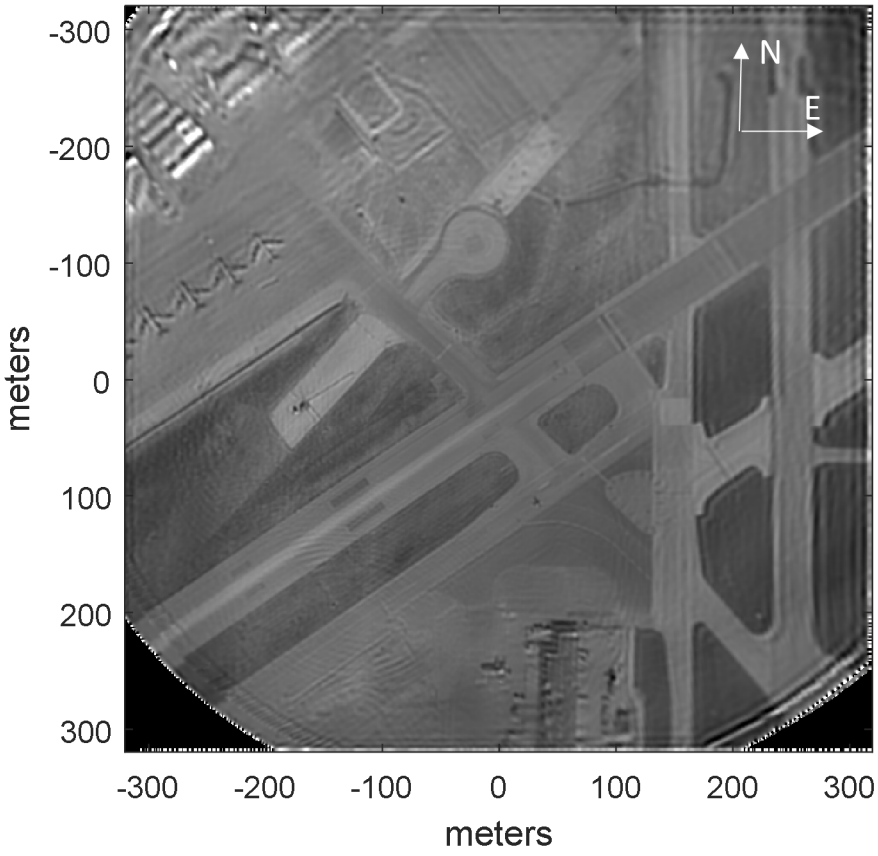


Figure 5-19b: SR mosaic image with IFEM method and SAR prior

Because the forward model included the effects of blur and optical distortion, these effects are both simultaneously removed from the mosaic, resulting in both visible flattening and sharpening of the ground image. There was no ground-truth resolution chart present in the scene; so, metrics such as that proposed in chapter 4 can't be applied and assessment of the resulting imagery must be qualitative. Comparing Figure 5-19b to both Figure 5-19a and Figure 5-18 shows visually both that the distortion artifacts have been removed and the fundamental resolution increased. The only noticeable artifact in Figure 5-19b is the presence of light ringing, particular as we get further from the center where the SR magnification factor get increasingly more aggressive.

As a final comparison, Figure 5-20 shows what would happen if we attempted to recover the HR image using the affine 3 d.o.f. registration model and ignored the calibration distortion. The results was generated with the non-mosaicing, variational Bayesian inference SR approach with SAR prior based on [65]. Clearly, without using a proper image formation model, the SR processing actually appears to degrade the image relative to the raw input in Figure 5-17.

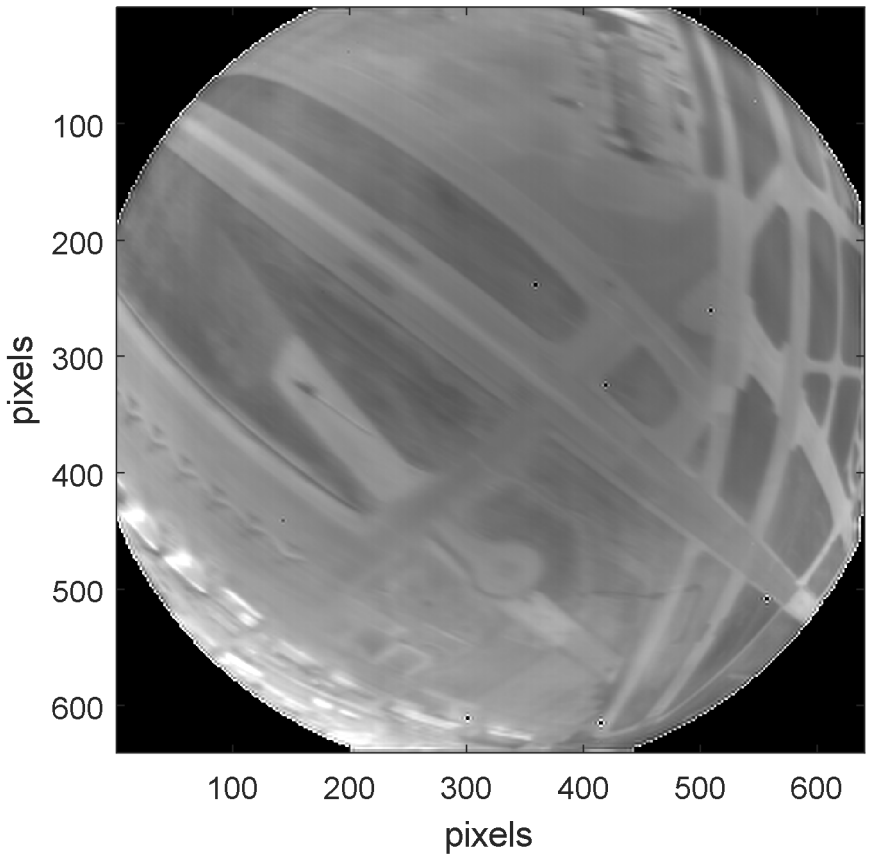


Figure 5-20: Direct SR using variational Bayesian inference with 3 d.o.f. affine motion model and SAR prior

CHAPTER 6

REMOTE IMAGE CLASSIFICATION USING SUPER-RESOLUTION

In this chapter, we examine the ability of image based super-resolution (SR) to enhance the ability of an imaging system to classify objects in the environment from a passive, airborne camera mounted on a manned or unmanned air vehicle. In order to limit the scope of the study, we focus on the problem of remote classification of text. As all classification problems are fundamentally tied to the resolution of the image [41-43,113], we claim these results will also apply to the closely related problems of remote face, vehicle, aircraft, etc. classification.

Much of the existing literature on image based classification focuses on the pixel-density, or number of pixels across the image of the targets of interest, as the primary predictor of performance. However, as we showed in chapter 4 and will confirm below, the other factors of Modulation Transfer Function (MTF) and Signal-to-Noise Ratio (SNR) are equally as important considerations. We find that the majority of the literature on SR algorithms briefly acknowledges these analog effects but then falls short of quantifying the degree to which a designer must adjust expectations for the ultimate effectiveness of SR based upon them. In this chapter, we quantify performance as a function of MTF, SNR, as well as pixel-density on classification performance.

The domain of remote sensing and, particularly, remote sensing from an airborne camera brings unique considerations to the problem. Some elements of SR applied to remote sensing are covered in [7,20,140]. First, unlike reading text in a scanner, the operator has no control of the ambient light conditions and, therefore, no control over the SNR. Low SNR scenarios are quite possible. Second, the motion of the air vehicle due to wind buffeting and lightly damped oscillatory modes in the control loop [141] will reduce the higher-frequency MTF gain; thereby, introducing unrecoverable degradation. This additional, motion induced blur may be mitigated by reducing the

frame exposure time at the expense of SNR. On the other hand, this natural motion provides the ego-motion required for typical multi-frame based SR algorithms without the necessity of specialized hardware such as micro-scanners or the need for deliberate flight maneuvers. Third, the orientation of the target relative to the camera (both in terms of position and pose) may be arbitrary. Fourth, particularly for streaming video, the video stream may undergo lossy compression before either on-board storage or off-board transmission (SWaP constrains may force off-board processing). Fifth, size, weight, and power (SWaP) restrictions limit the size of optics, therefore, limiting the MTF and potentially introducing distortion.

In this chapter, we use a combination of Monte-Carlo simulation and real data collections to examine remote text classification capability as a function of the relevant parameters of pixel-density, MTF, and SNR. For real data collections, we use the Phantom 3 quadcopter with attached visible RGB camera described in Appendix A.

6.1 Performance Predictions Through Simulation

All experiments, simulated and real, are based upon the target board shown in Figure 6-1.

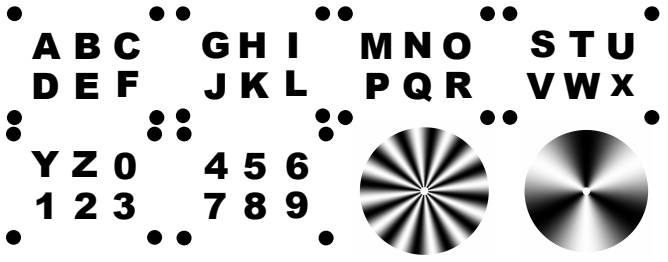


Figure 6-1: Test board for simulation and experiment

The 36 characters comprise the capital letters A-Z and the numerals 0-9. The characters are printed in Aerial bold, 176 point font. Each character is centered inside a 6.9 x 6.9 cm (2.7 x 2.7 in) square. The circular dots on each pane with characters are 2.5 cm in diameter and used as fiducial points

for locating and segmenting the characters from aerial imagery. The two circular, sinusoidal Siemens star patterns on the lower right are used for direct, *in-situ* MTF measurement as is described in Appendix B.

Using a black on white target has an additional benefit since we are imaging with an RGB color camera. For color images, the standard Bayer mask coding of the image array has the effect of reducing the pixel density for each color component relative to the stated pixel density of the image. It then further degrades the MTF by performing demosaicking [142]. With a greyscale target, however, the signal to all color pixels is highly correlated. Therefore, we retain the stated, nominal pixel sampling density of the array.

Text-Recognition Using the Sub-Space Method

Text classification is a well-studied and diverse field. In [16], the problem is broken down into 4 levels of complexity:

> **Level 0.** Low variation in shape; e.g., printed characters of a specific font or highly constrained handwritten symbols. Low noise.
>
> **Level 1.** Medium variation in Shape; e.g. printed characters of multiple fonts or loosely constrained handwritten symbols. Medium noise.
>
> **Level 2.** High variation in shape; e.g. printed characters with a large variety of fonts, unconstrained handwritten characters, significant affine shape transformations. High noise.
>
> **Level 3.** Non-segmented string of characters; e.g., touching/broken characters, cursive handwriting, characters on a textured background.

Levels 1-3 are currently active areas of research, including state of the art machine learning methods such as Artificial Neural Networks, Support Vector Machines, and Deep Learning [16,17,143,144,145]. However, as our objective is to studying the effects of image enhancement on the classification problem, we limit ourselves to only considering level 0 complexity. This means that we ignore the challenges of unknown fonts and large, varying character rotations so

that we can decouple the effects of pixel-density, MTF, and SNR from the artificial intelligence complexities of the more advanced algorithms. The only exception is that, given the drone environment, we have to accommodate some of the characteristics of level 1. These include medium to high noise as well as medium variation in scaling and rotation.

For classifying text, we use the common sub-space method as described in [18]. This method is a simple machine learning approach that consists of a training stage and a recognition stage. The method is known to be robust to small sample sizes as well as to small transformations such as shifting, scale, and rotation. The latter is important in considering imagery from a drone. In the training phase, we learn an orthogonal basis for the training data by constructing a matrix for each class $\boldsymbol{c}$ (each character is a class for our application)

$$\boldsymbol{X^{(c)} = [x_1 \quad \cdots \quad x_K],} \tag{6.1.1}$$

where $\boldsymbol{K}$ is the number of training samples for class $\boldsymbol{c}$. Each vector, $\boldsymbol{x_i}$, in (6.1.1) is the i'th training image converted to an $\boldsymbol{n}$-element column vector in raster order and normalized to have zero mean and unity magnitude ($\boldsymbol{n}$ is the total number of pixels in the sample image). Once the matrix $\boldsymbol{X^{(c)}}$ is generated, it is factorized, using its singular value decomposition into

$$\boldsymbol{X^{(c)} = S^{(c)} \Sigma V^T.} \tag{6.1.2}$$

The matrix $\boldsymbol{S^{(c)}}$ is an $\boldsymbol{n\,x\,n}$ matrix with each column containing an orthogonal basis vector representing the class $\boldsymbol{c}$. The ordered singular values in the diagonal matrix $\boldsymbol{\Sigma}$ provide a measure of how well the entire training set is represented by taking only the first $\boldsymbol{r < n}$ columns of the matrix $\boldsymbol{S^{(c)}}$.

For classification, we again convert a new image of an unknown target into a zero mean, unity magnitude vector $\boldsymbol{y}$ using the same method as in the training images. We then assign sample $\boldsymbol{y}$ to class $\hat{\boldsymbol{c}}$ based on

$$\hat{c} = \boldsymbol{arg}\,\boldsymbol{\max_{c}} \sum_{j=1}^{r} \left(\left[s_j^{(c)} \right]^T \boldsymbol{y} \right)^2, \tag{6.1.3}$$

where $\boldsymbol{s}_{\boldsymbol{j}}^{(\boldsymbol{c})}$ represents the $\boldsymbol{n}$-element, $\boldsymbol{j}^{\text{th}}$ column of the matrix $\boldsymbol{S}^{(\boldsymbol{c})}$ learned from the training matrix $\boldsymbol{X}^{(\boldsymbol{c})}$**.** The general concept of the sub-space method, that is of representing a training set by a reduced set of basis vectors, is common in other domains such as face classification [19].

As mentioned above, the sub-space method is robust to only slight variations in scale or rotation. Many classification methods attempt to first "correct" the image by shifting, rotating, and scaling as a pre-processing step. The problem is the required interpolation manifest itself directly as an equivalent MTF attenuation and creates an irrecoverable loss of information comparable to that caused by optical or motion blur. As will be seen below, reduced system MTF will degrade its classification capability. Consequently, for our current study, we avoid any such pre-processing transformations.

Simulation

A Monte-Carlo simulation is used to generate predictions of the capability of the sub-space method to classify characters as well as the ability of SR to improve the results. The simulation starts with a high-resolution version of the "truth" board shown in Figure 6-1. It then performs optical blur, sampling, and noise addition using the image formation model from 2.1. Figure 6-2 shows the original template and examples of the simulated image with different Gaussian blur and pixel-density parameters. The Gaussian blur is parameterized by a single parameter, the standard deviation $\boldsymbol{\sigma}$, expressed in pixels. SNR, measured in decibels (dB), is defined as $\boldsymbol{20log_{10}(MOD/\sigma_N)}$ where $\boldsymbol{MOD}$ is the depth of modulation defined as the difference between the peak pixel value in the character and the background and $\boldsymbol{\sigma_N}$ is the standard deviation of a Gaussian white noise.

For the Monte-Carlo, we simulate the board being imaged at several different ranges such as the number of pixels across each, fixed-size character decreases from 10 to 3. At each discrete range, we step SNR from 0.0 dB to 32.0 dB in steps of 4 dB. At each discrete range and SNR combination, we generate a set of 40 different image-sequences of the board, each with an independent draw on the additive Gaussian pixel noise and sub-pixel translational motion. An image-sequence is defined as a set of 7-images which will be fed into the SR algorithms to produce a single enhanced image. We use 20 of these samples to train a sub-space classifier based on (6.1.2) and use the other 20 to test the performance of the classifier based on (6.1.3). At each discrete range and SNR combination, we then score the classification capability by creating a 36 x 36 element confusion matrix. The diagonal of the confusion matrix shows the probability of each character being correctly classified whereas the off-diagonal terms show the probability of making each possible mis-classification error; e.g. the letter "O" will be more frequently confused with the

letter “Q” than with the number “9” and this will be reflected in the corresponding off-diagonal elements. For summary purposes, we compress the confusion matrix into a single, scalar metric by taking the average of the diagonal.

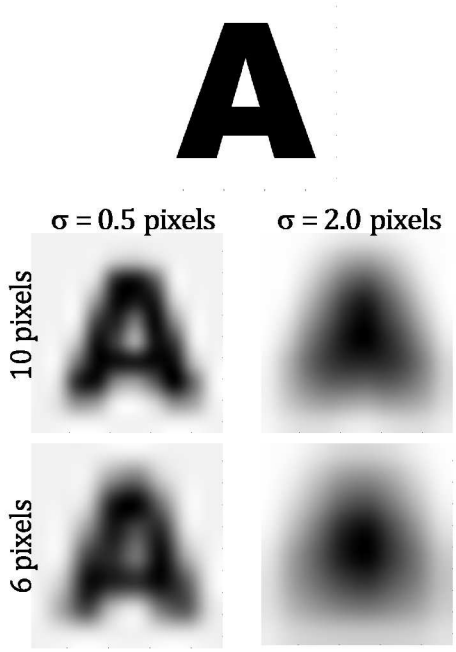


Figure 6-2: Simulated blurred and sampled character “A” for different Gaussian blur kernels and pixel-densities

Note, in Figure 6-2, particularly for the bottom right image corresponding to a $\sigma = 2.0$ Gaussian blur and 6 pixel-density, the transformation caused by the imaging process makes the character unrecognizable from a human viewing perspective. However, this is not necessarily a problem for the machine based classification algorithm as the transformed image is still information rich and the algorithm works directly in the transformed space.

Figure 6-3 through Figure 6-5 show, for a Gaussian blur of σ equal to 0.50, 1.50, and 2.50 pixels respectively, the number of pixels across the character, or pixel-density, required to get a 50% classification capability score (defined as the mean of the diagonal of the confusion matrix) as a function of SNR. As shown in Appendix B., a Gaussian blur of $\sigma = 1.50$ is the closest match

to the blur of our experimental Phantom 3 drone. The different lines in the figures correspond to different enhancement algorithms as introduced in 2.4. “Raw” means no SR is applied, “Babacan” uses the variational Bayesian inference method in [58], “BiCubic” is a non-SR up-sampling by bicubic interpolation, “Farsiu” uses the SR method in [44], and, for reference, “Ideal SR” shows the capability for an ideal SR algorithm that achieves a magnification effectiveness of exactly 2. The Babacan method is available in a Matlab package from Northwestern University [115] and the Farsiu method is incorporated in the popular open-source image software package Open Computer Vision (OpenCV) [133,134]. Note that, in all figures, all curves trend to flatten out at 3 pixel-density at sufficiently high SNR. This is due to the fact that we limited the domain of the Monte-Carlo to ≥ 3 pixels based on the belief that images with fewer than 3 pixels, while they may show performance in the simulation, are likely not of practical interest. Also, note that the visible discontinuities in the curves are artifacts attributed to the limited total number of trials and sampling density of the input parameter space of the Monte-Carlo. For example, the fact that some of the curves, e.g. the “Raw” curve in Figure 6-3, stop to the right of the SNR = 0 dB point is due to the fact that, at these lower SNR values, more than 10 pixels across the character would have been required to achieve a 50% probability of classification. However, as stated above, the Monte-Carlo was limited to the range of 10 to 3 pixels across each character; so, in these cases, the 50% performance point was never observed. Limitations on the Monte-Carlo were necessary for practical run-times and we did not attempt to extrapolate performance beyond the limits.

To clarify the interpretation of the figures, consider, in Figure 6-3, at an SNR of 10.0 dB, we required 6.5 pixels across each of the characters to achieve 50% classification rate on the raw, unenhanced image. If we had an “ideal” SR algorithm, with a magnification factor of 2, it would

have only required 6.5/2 = 3.25 pixels across the character to achieve the equivalent 50% correct classification rate in our simulated experiment (as shown by the "Ideal SR" line). However, the SR algorithms are not ideal. At this set of conditions, the Babacan algorithm required the input image to have a pixel-density of 4.5 to achieve the same 50% correct classification rate as achieved by the raw image with 6.5 pixels across the character. This means that, even though the SR algorithm did increase the total number of pixels by a factor of 2, it did not actually increase the classification task effectiveness by the same factor. Likewise, at these same conditions, the Farsiu algorithm required the input image to have 6.5 pixels across the character to achieve a 50% correct classification rate. Since, at these conditions, the raw image also required 6.5 pixels across the character to achieve a 50% correct classification rate, the Farsiu algorithm did not provide any benefit (although it does provide a benefit at higher SNR).

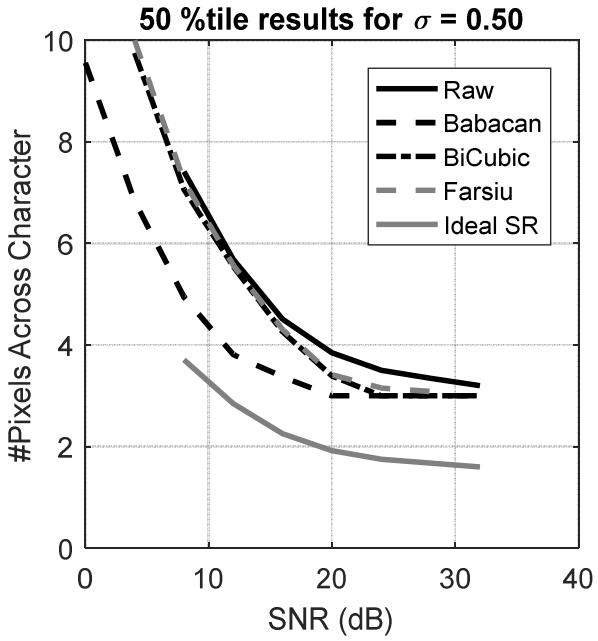


Figure 6-3: Simulation Results for σ = 0.50 Gaussian blur

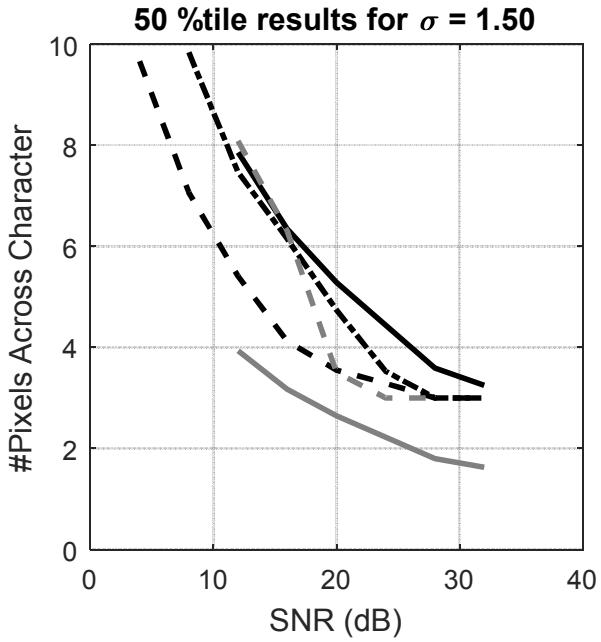


Figure 6-4: Simulation Results for σ = 1.50 Gaussian blur

As an alternate perspective, we can define an effective magnification metric, $\mathbf{M_{eff}}$,for a particular SR algorithm, as

$$\boldsymbol{M_{eff}} = \frac{\boldsymbol{npix_{raw}}}{\boldsymbol{npix_{SR}}}, \tag{6.1.4}$$

where $\boldsymbol{npix_{SR}}$ and $\boldsymbol{npix_{raw}}$ are the number of pixels across the character required, in the raw input image, to produce an equivalent probability of correct classification, $\boldsymbol{P_{class}}$, in the case where the input image is enhanced by the SR algorithm and the case where it is not. That is, under the constraint

$$\boldsymbol{P_{class,SR}(npix_{SR}) = P_{class,raw}(npix_{raw})}, \tag{6.1.5}$$

for a given SNR. Thus, even though the SR algorithms will always produce an output image with twice the pixel density of the raw input image, $\mathbf{M_{eff}}$, is a measure of the actual achieved benefit. Figure 6-6 shows $\mathbf{M_{eff}}$ for both the Babacan and Farsiu algorithms at various values of SNR for the cases of 3 and 6 pixels across the character in the input, pre-enhanced images.

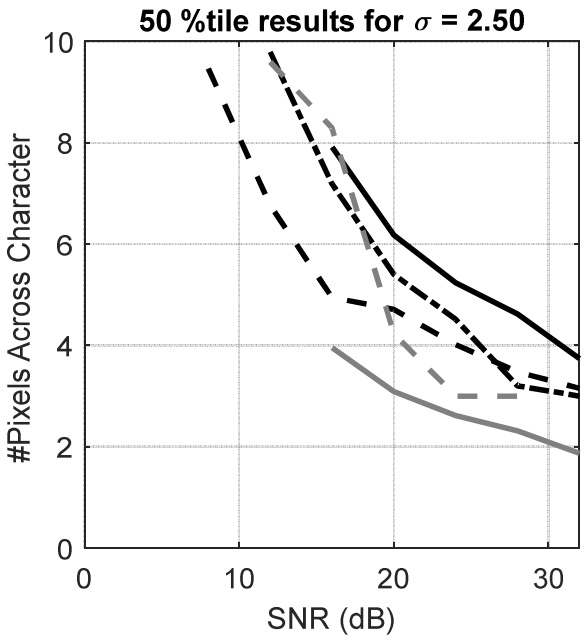


Figure 6-5: Simulation Results for σ = 2.50 Gaussian blur

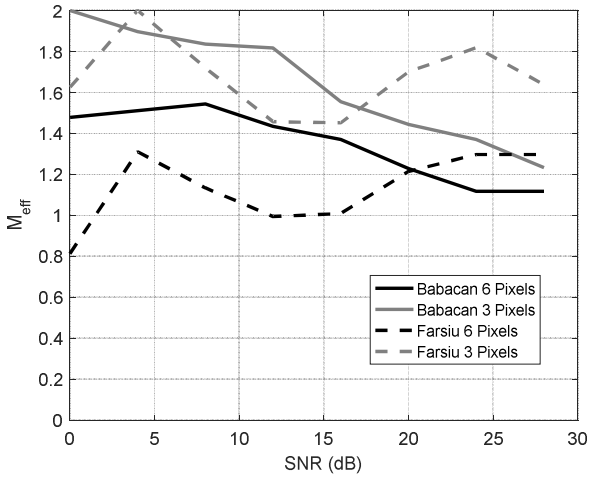


Figure 6-6: Simulation results showing SR magnification effectiveness for σ = 1.50 Gaussian blur

As expected, Figures 6-3 through 6-5 show that achievable performance, both in the case of raw images as well as with images enhanced with SR, decreases the higher the σ of the Gaussian blur. The second interesting observation is that the BiCubic method, which in principle should not add any information, actually does improve performance relative to raw, unenhanced images. We

attribute this to the fact that interpolation in the BiCubic process reduces noise. The third observation is that, while the Babacan algorithm tends to show a relatively consistent trend, the behavior of the Farsiu algorithm tends to vary more as a function of the specific condition parameters. For the case of less optical blur, such as shown in Figure 6-3, the Farsiu algorithm performs no better than the non-SR BiCubic algorithm. In contrast, for the cases of greater optical blur, such as in Figures 6-4 and 6-5, the Farsiu algorithm shows the best performance of any algorithm when the SNR is greater than about 17 dB. Also, as shown in Figure 3-5, for greater optical blur, the Farsiu algorithm tends to get better performance than the other methods in the region of 8-10 pixel-density and SNR < 10 dB.

Figure 6-6 shows a similar comparison trend. With a Gaussian blur of $\sigma = 1.50$, the Babaan algorithm follows a relatively smooth trend whereas the Firsiu algorithm provides better effectiveness at both low and high SNR but shows less effectiveness at moderate SNR.

In all cases, the $\mathbf{M_{eff}}$ of the SR algorithms tends to decrease at higher SNR due to the simple fact that, in these low noise conditions, the pixel-density becomes increasingly less critical and, therefore, we get good performance even out of the unenhanced images. Similarly, and intuitively, Figure 6-6 shows the overall benefit of performing SR on the raw images decreases as the pixel-density of the raw image increase. This is again because, with sufficient pixel-density, we already get good performance from the raw images; thereby, limiting the potential for the SR algorithm to add additional improvement.

6.2 Comparison to Spatial-Frequency Metric

As a means to confirm the differences seen between the BiCubic, Babacan, and Farsiu algorithms in the previous section, they are re-evaluated using the application agnostic, spatial-frequency metric introduced in chapter 4. For this metric, we examine the probability of correctly measuring the spatial-

frequency of a bar target that has been subjected to the camera's analog blur and digital sampling. The motivation for using bar targets comes from the historical work of Johnson [41-43] which concludes that the performance of an imaging system in recognizing high spatial-frequencies in a simple image, such as a bar target, is a strong indicator of its performance in a large set of specific, more complex task, such as text classification.

An example of a 4 cycle bar target image is shown in Figure 6-7. The left image represents the analog scene, $\boldsymbol{s(x, y)}$, from (2.1.1). The right image represents the output of the camera after the scene is subjected to the modeled blur and sampling such that there $\boldsymbol{n_{pixels}} = 20$ across the image (no noise is included in this example). In general, the spectrum of the image of an $\boldsymbol{n_{cycles}}$ bar target will show up with peaks at spatial frequencies 0 and $\frac{\boldsymbol{n_{cycles}}}{\boldsymbol{n_{pixels}}}$ cycles/pixel in the DFT as shown in Figure 6-8. Technically, for the bar target, there are additional peaks at the odd harmonics given by $\boldsymbol{(2n+1)\frac{n_{cycles}}{n_{pixels}}}$, where $\boldsymbol{n}$ is an integer > 1. However, we typically assume, since we're interested in higher frequencies, that these harmonics are sufficiently attenuated by the camera's MTF to be ignored.

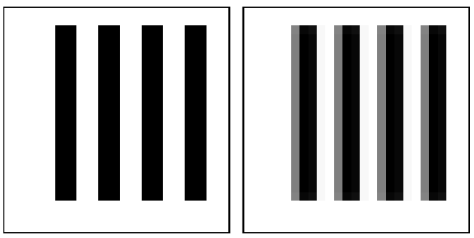


Figure 6-7: Example of $n_{cycles} = 4$ bar target. The analog scene is on the left and the simulated camera image, with $n_{pixels} = 20$, is on the right. No noise is present in this example.

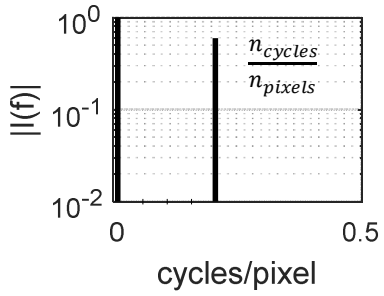


Figure 6-8: DFT of the sampled and blurred image (un-aliased case). Principal component occurs at $\frac{n_{cycles}}{n_{pixels}}$ = 0.2 cycles/pixel.

Given an image of a bar target as input, in the presence of noise, we measure the spatial-frequency of the bar target, $\hat{\boldsymbol{n}}_{\boldsymbol{cycles}}$, by finding the frequency (> 0) corresponding to the maximum peak in the DFT. Then, the spatial frequency performance metric, $\boldsymbol{P_m}$, as defined in chapter 4, is given by the probability of measuring the correct frequency. That is,

$$\boldsymbol{P_m}\left(\boldsymbol{n_{cycles}}, \boldsymbol{SNR}\right) = \boldsymbol{Pr}\left\{\hat{n}_{cycles} = \boldsymbol{n_{cycles}}\right\}. \tag{6.2.1}$$

Of particular interest, for evaluating an SR algorithm, is the case where $\boldsymbol{n_{cycles}} \leq \boldsymbol{n_{pixels}} < \boldsymbol{2n_{cycles}}$. In this case, the principal spatial frequency will be > 0.5 cycles/pixel and, therefore, aliased in the raw image into the wrong location as given by the aliasing property (2.1.6). Consequently, for the raw image, the metric $\boldsymbol{P_m}\left(\boldsymbol{n_{cycles}}, \boldsymbol{SNR}\right) = 0$. However, if an SR algorithm properly expands the effective number of pixels to $\boldsymbol{2n_{pixels}}$ then, in the SR output image, the peak will appear at $\frac{\boldsymbol{n_{cycles}}}{\boldsymbol{2n_{pixels}}}$ which is ≤ 0.5 cycles/pixel and will, therefore, be detectable with $\boldsymbol{P_m}\left(\boldsymbol{n_{cycles}}, \boldsymbol{SNR}\right) > 0$. The better the SR algorithm, the higher the probability.

We are able to evaluate the $\boldsymbol{P_m}$ metric using the same Monte-Carlo procedure as used for the text classification simulation. For each Monte-Carlo trial, we generate 7 instances of the bar target image with different noise draws and random sub-pixel translational offsets. Each value of $\boldsymbol{P_m}\left(\boldsymbol{n_{cycles}}, \boldsymbol{SNR}\right)$ is then computed based upon 32 Monte-Carlo trials. All simulations are generated with a $\boldsymbol{\sigma}$ = 1.50 Gaussian blur

because this corresponds most closely to the blur in our later experiments with the Phantom 3 drone. We vary $\boldsymbol{n_{pixels}}$ from 4 to 10. Results are shown in Figure 9 for bar targets with $\boldsymbol{n_{cycles}}$ equal to 3 and 4. The graphs show, for each SNR and algorithm combination, the pixel-density of the raw image required to get a 50% $\boldsymbol{P_m}$ in the enhanced image.

Again, of particular note are the cases where the SR enhancement allows us to recognize initially aliased bar target frequencies where $\boldsymbol{n_{pixels} < 2n_{cycles}}$. These are the cases where the SR algorithm is clearly doing its job of properly unrolling the aliased frequency components. For $\boldsymbol{n_{cycles}}$ = 3, this corresponds to allowing us to make a correct measurement with $\boldsymbol{n_{pixels}} \leq 6$ and, for $\boldsymbol{n_{cycles}} = 4$, allowing us to make a correct measurement with $\boldsymbol{n_{pixels}} \leq 8$.

From Figure 6-9, we see the same trend as in the character recognition simulations from Figure 6-3 through Figure 6-6 that the Babacan algorithm has a relatively smooth trend whereas the Farsiu algorithm tends to have a more complex dependence on the specific conditions. As before, visible discontinuities in the graphs are artifacts attributed to practical limits in the number of trials and input space sampling of the Monte-Carlo.

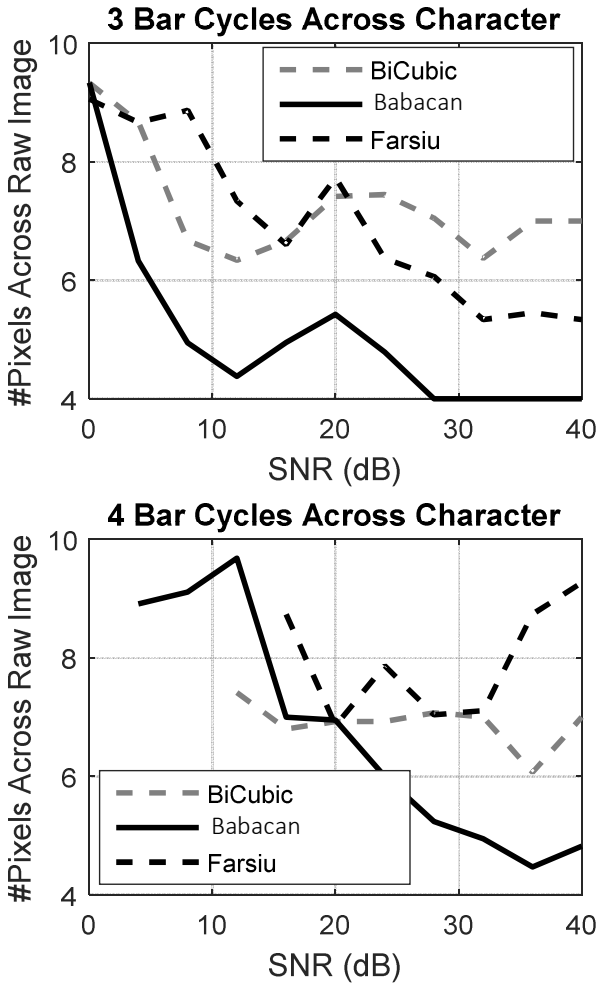


Figure 6-9: $P_m(n_{cycles}, SNR)$ = 50% for bar targets with n_{cycles} of 3 and 4 (σ = 1.50 Gaussian blur)

The typical difference between the Babacan and Farsiu algorithms, when applied to the bar target, is shown in Figure 6-10 for a single Monte-Carlo instance with $\boldsymbol{n_{cycles}}$ =4, $\boldsymbol{n_{pixels}}$= 4, and SNR=40 dB. In this case, in the raw image, the spatial frequency of the bar target is 1.0 cycles/pixel which is undetectable. In the SR image, however, since there will 2x, or 8 pixels, across the image, the spatial frequency of the bar target is located at a detectable frequency of 0.5 cycles/pixel.

As expected, the BiCubic enhancement is unable to unroll the aliased frequency and, therefore, creates no detectable peak at 0.5 cycles/pixel (it is possible for noise to, coincidentally, create a false at the correct frequency resulting in a $\boldsymbol{P_m}$ > 0). Both the Babacan and Farsiu

algorithms are able to recover the energy at 0.5 cycles/pixel; however, they also introduce some artifacts at other frequencies. Of the two, the Farsiu algorithm introduces the most artifacts, increasing the probability of measuring the wrong peak and, consequently, decreasing $\boldsymbol{P_m}$.

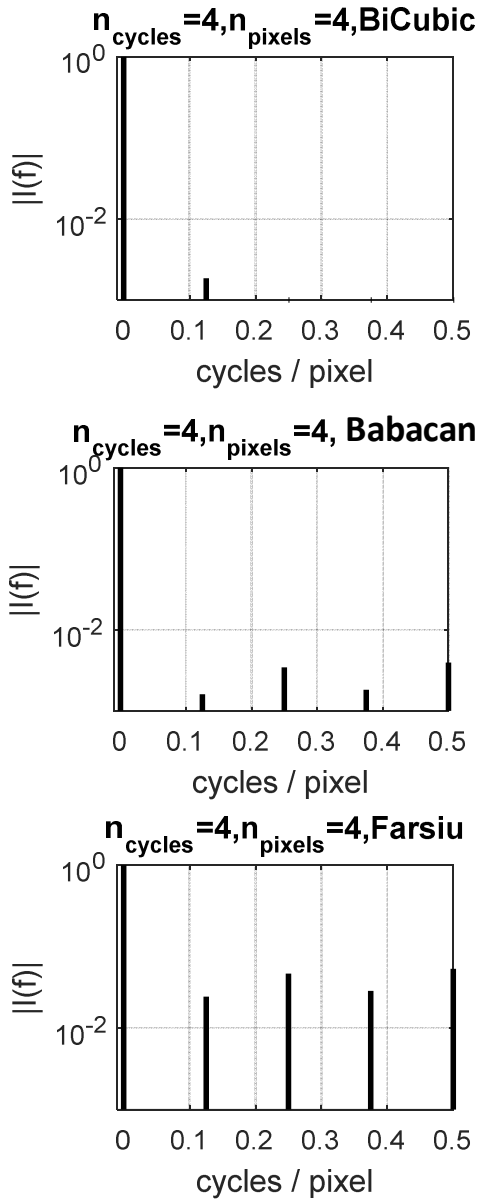


Figure 6-10: Example DFT spectrums of a n_{cycles} =4, n_{pixels}= 4, SNR=40 dB bar target image after SR enhancement

Although the absolute value of the $\boldsymbol{P_m}$ metric does not match one-to-one to the probability of successful performance of each algorithm on the specific task of character classification, it does

predict the same general observations we make when comparing performance. Most notably, it predicts

i. Over the parameter space, the Babacan SR algorithm outperforms both BiCubic and Farsiu algorithms.

ii. The performance trend of the Babacan algorithm is relatively smooth over the input conditions whereas that of the Farsiu SR algorithm has a more complex and varying dependence on the specific input conditions.

6.3 Performance on Real Data

For real data experiments, we image the board in Figure 6-1 with the Phantom 3 drone in flight. During flight, motion of the vehicle, even though it is stabilized by the GPS, prevents us from maintaining perfectly consistent geometric conditions image to image. In practice, during hover, we achieve a rotation consistency of +/- 1.5 degrees and an altitude consistency of +/- 1 meter. The relative effect of variation of scale due to sample to sample altitude variation diminishes as the altitude increases. All imagery for classification performance analysis are taken with the Phantom 3 in a hover condition located at a fixed altitude directly above the target board. Ideally, during hover, the vehicle would have a ground speed of 0. In practice, the Phantom 3 is able to maintain its speed < 0.25 meters/second. SNR is largely based upon external lighting conditions. With the 1/1000 s shutter speed, we achieve SNR levels of 25-30 dB in direct sunlight and 10-20 dB in daytime overcast conditions. SNR drops rapidly during sunset.

As mentioned before, a characteristic unique to an air vehicle is that buffeting in the wind as well as natural, lightly damped frequencies of the internal control loop provide a continuous and guaranteed source of frame-to-frame motion. Consequently, specialized hardware, such as mechanical micro-scanners, are not required to provide the frame-to-frame motion needed by

typical multi-frame SR algorithms. The further benefit of using a small drone for experimentation is that it is convenient and inexpensive to use and re-used numerous times under varying conditions. This would not be possible using a more expensive or higher overhead collection platform, such as a real aircraft.

Results

Modeling the Phantom 3 blur based on the measured Gaussian blur with $\sigma = 1.20$ (see Appendix B), we can compare the directly measured characters from the Phantom 3 in flight to the simulated characters. A comparison is shown in Figure 6-11.

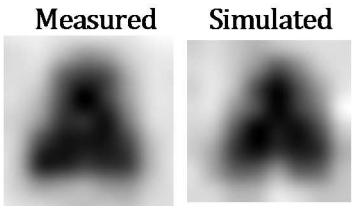


Figure 6-11: Comparison of measured and simulated character "A" from a 20m altitude hover. Simulation uses Gaussian blur with $\sigma = 1.20$

In Figure 6-12, we show a sample of the target board captured by the Phantom 3. This is one image in a 7-image burst, we also show the SR output, using the Babacan algorithm, after processing all 7 images in the burst. In Figure 6-13, we zoom in on the character "A" and show the raw measurement compared to its SR enhanced image based on the Babacan (top) and Farsiu (bottom) algorithms respectively. Prior to processing the boards for classification performance statistics, it is necessary to manually locate the fiducial markers on the target board such that the individual characters can be properly segmented. Once segmented, the modulation depth can be measured directly by differencing the minimum and maximum intensity values and the noise level can be measured as the standard deviation of the digital counts in a homogeneous white section of the board. Combining these two measurements provides the SNR of the extraction. The example

shown in Figure 6-12 and Figure 6-13, which was taken in direct sunlight illumination, has a measured SNR of 25.8 dB.

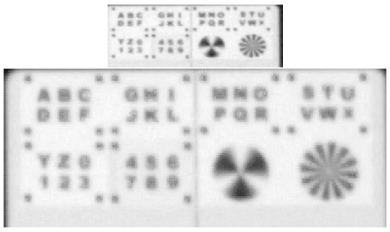


Figure 6-12: Target board captured from Phantom 3 at 20m altitude. Shown are raw image (top) and Babacan enhanced image (bottom)

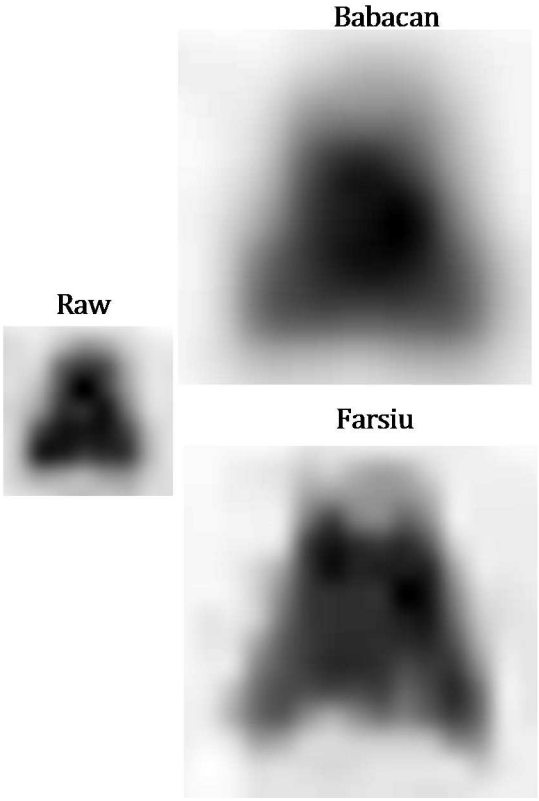


Figure 6-13: Comparison between raw and super-resolved image of the character "A" taken at 20m altitude during hover

Table 6-1 shows a comparison, for different altitudes of the drone above the target board (which results in different pixel-densities across the characters) and lighting conditions (which results in different SNR values), between measured classification performance results from the

Phantom 3 data and the classification results from the Monte-Carlo simulation. The simulation results corresponding to a Gaussian blur with $\sigma = 1.50$ as this was the closest match between the conditions used in the Monte-Carlo of section 6.1 and the actual MTF of the Phantom 3. For the Phantom 3, each row in Table 1 is generated by taking 20 sets of 7-image burst of the target board. Each image, therefore, contains all 36 of the characters. From these, 10 sets are used to train the sub-space classifier via (6.1.2) and the remaining 10 are used to test the classification via (6.1.3). The scalar performance metric is generated in the same manner as for the simulations; i.e., by taking the mean of the diagonal of the resulting confusion matrix. The reason for 20 total events is because that is the maximum number we can capture during each flight of the Phantom 3 (battery limited to 30 min).

		Performance (Measured % / Predicted %)			
# Pixels Across Character	Average SNR (dB)	Raw	BiCubic	Babacan	Farsiu
8.1	27.5	97/100	100/100	100/100	100/100
5.7	19.9	32/56	66/63	77/79	74/76
5.4	27.3	84/90	97/97	98/96	98/99
4.1	17.5	25/20	53/30	62/55	58/40
4.1	13.9	4/11	15/18	37/38	25/16

Table 6-1: Measured vs. Predicted Performance on Phantom 3

In the first three rows of Table 6-1, corresponding to altitudes where the number of pixels across each character was > 5, the measured and predicted performance for the images enhanced with the BiCubic, Babacan, and Farsiu algorithms agree very well (to within 3%). In the last two rows, corresponding to the 4.1 pixels across character cases, the differences are larger, with the measured performance better than the predicted performance. We attribute these larger differences,

in the case of lower pixel resolution, to the fact that, as noted in Appendix B, the simulated, Gaussian MTF is marginally, particularly at higher frequencies, worse than that directly measured for the Phantom 3.

We also see, in Table 6-1, a general trend that the measured performance on the raw, unenhanced images is worse than predicted. We attribute this to the fact that, in the measured images, we have to use the fiducial points shown in Figure 6-1 to segment the characters. The segmentation is more difficult in the smaller, raw images than in the larger, enhanced images (including the BiCubic enhancement). Also, the measured data is subject to small but present scale and rotation changes due to the drone's motion. These two factors, geometric variation and segmentation errors, are not present in the simulations and appear to have a more pronounced effect on the performance of the smaller, raw images; thereby, creating an additional, decrease in performance not accounted for in the simulation.

Overall, of the 20 cases compared in Table 1, 17/20 of the cases showed less than a 10% difference between the measured and predicted results. Given the noted blur difference and additional complexities associated with the measured data as cited above, we consider this a reasonable match showing that the measured data corroborates the predicted trends of the simulation.

6.4 Operational Considerations

In this chapter, we focused on quantifying the ability of SR algorithms to enhance classification capability of imagery from a UAS with the claim that results for text recognition generalize to other common recognition task. Consequently, the methodology for generating

results contained constraints that do not translate directly into a useable operational scenario for remote classification of text.

The first constraint is the fact that we use measured samples of the characters to both train as well as classify. In an operational scenario, where there is only a single sample of the character that is to be classified, this would not be possible. The solution is to utilize the generative method [146]. In the generative method, once we see the unknown character, we use the calibrated camera model (specifically the measured MTF) to generate, online, a simulated training database of the character library projected to the same scale and orientation as the measured character. We then use this simulated training database to train the sub-space method via (6.1.2) and use (6.1.3) to classify the measurement. One significant advantage of the generative method is it handles arbitrary rotation of the measured character. This is because the online training set is generated to match the measured character's rotation as opposed to rotating the measured character to match the training set. The high-resolution templates from the library are rotated prior to applying the blur and down-sampling. Because, in the generative method, the high-resolution templates are rotated to match the measured orientation of the captured image as opposed to rotating the captured image to match the templates, the method avoids the additional MTF degradation that would, otherwise, result from sub-pixel interpolation associated with rotating the captured image.

The second fundamental constraint is that we utilized only burst of uncompressed images from the drone. At minimum, the use of uncompressed imagery requires more storage capacity and bandwidth than may be present on a typical, small drone. In addition, as with the Phantom 3, there may not be an option to record continuous video in an uncompressed format. To overcome this limitation requires investigating SR algorithms for compressed video [147] which brings

additional dimensions, such as compression algorithm and quality, into the performance trade-space.

The third constraint of our method is that, as stated in section 6.1, we assume a fixed set of classes. In this paper, we limited ourselves to 36 characters from the Aerial font; however, the set of classes could be arbitrarily increased based on the computation resources of the signal processor. Otherwise, the constraint of *apriori* class templates is applicable to a large set of problems including face classification against a library, text classification of a constrained font (road signs, license plates, etc.), vehicle classification, and so on.

6.5 Conclusion

In this chapter, we have evaluated the ability of SR algorithms to genuinely enhance remote classification capabilities of a UAS mounted imaging system. To limit the scope of the evaluation, we focused on the well-defined problem of remote text classification of a fixed font. However, we claim the methodology and results are applicable to a broad range of remote classification problems. We generated our detailed statistical performance predictions through the use of a generic image formation model and Monte-Carlo simulation. We corroborated these results both with the general spatial frequency metric for SR algorithm as well as with real experimental data taken from a DJI Phantom 3 quadcopter.

Our prediction results and methodology were sufficiently consistent with experimental data to be used to predict the performance for text or other remote classification task for any observer platform where the MTF of the imaging system and the expected SNR of the targets can be characterized.

We find that, even in a hover condition, the natural buffeting and control-loop oscillations of an air vehicle create sufficient motion to enable multi-frame SR algorithms without any special hardware or deliberate in-flight maneuvering. We also find that, through the use of typical SR algorithms, we can perform practical character recognition (achieving > 50% correct recognition rate) on characters from a fixed font with representative, low-cost drone optics down to 4 pixels across the character with an SNR > 17 dB. Quantifiably reduced performance is available for conditions of either lower pixel-density and/or lower SNR if acceptable to the higher-level application.

The current work leaves open a number of questions to be addressed in future studies. The results showed that performance is highly dependent upon the imaging system MTF. Our current testing with blind-deconvolution algorithms for measuring MTF without a dedicated calibration target showed this method to be inadequate both for performance prediction as well as for use in the generative algorithm. However, having an acceptable method to measure MTF without the need for a dedicated target is of immense practical value, particularly for conditions where it may be necessary to have a lower shutter speed and capture images during motion vs. hover. Similarly, although performance on compressed imagery and video will be less than performance on uncompressed imagery, it would be useful to quantify the difference in a side-by-side comparison using the metrics introduced in chapter 4.

CHAPTER 7

CONCLUSION

In this thesis, we have extended the capabilities of the general technology of image super-resolution (SR) to include the domain of airborne sensors.

In chapter 3 of the thesis, we provided an efficient solution for image correspondence estimation by fusing supplementary information from an inertial navigation system, such as would be present on either a manner or unmanned aircraft system. Even though many state-of-the-art SR methods jointly estimate registration parameters along with the high-resolution image, many still expect registration as an external input. The alternate method of basing registration estimates exclusively on the low-resolution images, using traditional optical-flow techniques, is problematic as aliasing will corrupt the estimates (aliased components don't move as expected). Also, in addition to SR applications, this new method can support other image processing techniques that depend upon knowing correspondence.

In chapter 4, we addressed a shortcoming in the literature in that no existing metric truly assesses SR on its ability to perform its primary function of increasing image resolution. There is no clear traceability between existing metrics and the specific definition of resolution as defined by the international standards organization (ISO) in the ISO-12233 standard. Such a metric is arguably not critical for the photographic application class of imaging; i.e., where the final output is meant for a human observer. Indeed, most existing metrics for evaluating SR are human perception centric. A true, resolution based metric is important for assessing SR performance when it is applied to the photogrammetric application class of imaging. In chapter 4, we provided a rationale for a new metric that fills this omission, developed the metric, and showed its use through both simulation and real data.

In chapter 5, we derived modifications to existing, state-of-the-art SR solutions in order to allow them to handle unique characteristics and challenges of airborne camera applications. These additional challenges are wide field-of-view, possibility of significant lens distortion, and oblique viewing geometries. In order to accomplish the extension, we first derived an SR solution, called the information-filter / expectation-maximization (IFEM) solution, which is comparable to the state-of-the-art Bayesian based, SR solutions. However, the IFEM is much less complicated to derive and implement that other solutions in this class, such as variational Bayesian inference, making it more amenable to rapid expansion and experimentation. As part of this activity, we also showed how to take advantage of the exact automatic differentiation computing technique which, at present, we have not found to be in wide use for image processing applications.

In chapter 6, we demonstrated the utility of SR for remote sensing problems by showing a marked improvement in the probability of remote text classification from imagery captured by a small aerial drone with an attached camera. Although, to limit scope, the study used remote text classification as the surrogate problem, we claim that our results and observations extend to a general class of applications such as remote face or vehicle classification.

REFERENCES

1. T. Huang and R. Tsai, “Multi-frame image restoration and registration,” Adv. Comput. Vis. Image Process. **1**, 317–339 (1984).
2. M. Alam, J. Bognar, R. Hardie, and B. Yasuda, “Infrared Image Registration and High-Resolution Reconstruction Using Multiple Translationally Shifted Aliased Video Frames,” Instrumentation and Measurement, IEEE Trans. on **49**, 915-923 (2000).
3. W. O’Neil, “Recent Progress in Microscan Resolution Enhancement”, Northrop Grumman Corporation, Electronic Systems and Sensors Sector, (1999).
4. F. Liu, J. Wang, S. Zhu, M. Gleicher, and Y. Gong, "Visual-Quality Optimizing Super Resolution", Computer Graphics Forum **28**, 127-140 (2009).
5. “Photogrammetry”, https://en.wikipedia.org/wiki/Photogrammetry.
6. H. Ren, Q. Du, J. Wang, C. Chang, J.O. Jensen, and J.L. Jensen, “Automatic Target Recognition for Hyperspectral Imagery using High-Order Statistics,” Aerospace and Electronic Systems, IEEE Trans. On **42**, 1372-1385 (2006).
7. A. van Eekeren, *Super-Resolution of Moving Objects in Under-Sampled Image Sequences*, PhD Thesis (Acedemic 2009).
8. T. Zhao, R. Nevatia, “Car Detection in Low Resolution Aerial Image,” *Proceedings of Eight ICCV* (IEEE, 2001), pp. 710-717.
9. C. Gronwall, F. Gustafsson, and M. Millnert, “Ground Target Recognition Using Rectangle Estimation,” Image Proc., IEEE Trans. On **15**, 3400-3408 (2006).
10. M. Ulmke and W. Koch, “Road-Map Assisted Ground Moving Target Tracking,” Aerospace and Electronic Systems, IEEE Trans. On **42**, 1264-1274 (2006).
11. O. Arandjelovic and R. Cipolla, “A Manifold Approach to Face Recognition from Low Quality Video Across Illumination and Pose using Implicit Super-Resolution,” *Proceedings of ICCP* (IEEE, 2007), pp. 1-8.
12. Z. Wang and X. Xie, “An Efficient Face Recognition Algorithm Based on Robust Principal Component Analysis,” *Proceedings of the ICIMCS* (2010), pp. 99-102.
13. S. Wang, L. Cui, D. Liu, R. Huck, P. Verma, J. Sluss, and S. Cheng, “Vehicle Identification via Sparse Representation,” Intelligent Transportation Systems, IEEE Trans. On **13**, 955-962 (2012).
14. K. Krapels, R. Driggers, and J. Garcia, “Performance of infrared systems in swimmer detection for maritime security,” Optics Express **15**, 12296-12305 (2007).
15. E. Bilgazyev, B. Efraty, S. Shah, and I. Kakadiaris,, “Improved Face Recognition Using Super-Resolution,” Biometrics, International Joint Conference on, 1-7 (2011).
16. S. Mori, H. Nishida, and H. Yamada, *Optical Character Recognition* (Acedemic, 1999).
17. C. Jacobs, P. Simard, P. Viola, and J. Rinker, "Text Recognition of Low-Resolution Document Images," *Eighth ICDAR* (IEEE,2005), pp. 695-699.
18. A. Ohkura, D. Deguchi, T. Takahashi, I. Ide, and H. Murase, "Low-resolution Character Recognition by Video-based Super-resolution", *10th International Conference on Document Analysis and Recognition* (2009), pp. 191-195.
19. J. Wright, A.Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," PAMI, IEEE Trans. on **31**, 1-18 (2009).
20. R. Wagner, D. Waagen, and M. Cassabaum, "Image Super-Resolution for Improved Automatic Target Recognition," *Proc. Of SPIE* (2004).
21. J. Verly, R. Delanoy, and D. Dudgeon, “Machine Intelligence Technology for Automatic Target Recognition,” The Lincoln Laboratory Journal **2**, 277- 311 (1989).
22. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (Academic 2006).
23. A. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera,” *Proc. Of the Intl. Sym. Of Robot Research* (2011), pp. 1-16.

24. A. Yol, B. Delabarre, A. Dame, J. Dartois, and E. Marchand, "Vision based Absolute Localization for Unmanned Aerial Vehicles," Int. Conf. on IROS (IEEE/RSJ, 2014).
25. Y. Li, Q. Pan, Z. Jin, and C. Zhao, "Scene matching based visual SLAM navigation for small unmanned aerial vehicle," *15th International Conf. on Information Fusion* (2012), pp. 2256-2262.
26. F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs," Journal of Intelligent and Robotic Systems **54**, 137-161 (2008).
27. J. Engel, J. Sturm, and D. Cremers, "Accurate Figure Flying with a Quadrocopter Using Onboard Visual and Inertial Sensing," in *Proc. Of the Workshop on ViCoMoR* (IEEE, 2012).
28. Thermography, http://www.infratec-infrared.com/thermography/application-area/building-thermography.html.
29. R. Ishimwe, K. Abutaleb, F. Ahmed, "Applications of Thermal Imaging in Agriculture – A Review", Advances in Remote Sensing **3**, 128-140 (2014).
30. O. Kung, C. Strecha, P. Fua, D. Gurdan, M. Achtelik, K. Doth, and J. Stumpf, "Simplified Building Models Extraction from Ultra-Light UAV Imagery," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **38**, (2011).
31. C. Strecha, R. Zoller, S. Rutishauser, B. Brot, K. Schneider-Zapp, V. Chovancova, M. Krull, and L. Glassey, "Terrestrial 3D Mapping Using Fisheye and Perspective Sensors," PiX4D (2016).
32. "Pix4D Mapper," https://pix4d.com/product/pix4dmapper-pro/
33. J. Casana and J. Kantner, "Drones are the latest archaeological tool," https://www.elsevier.com/connect/drones-are-the-latest-archaeological-tool
34. M. Pollefeys and L. Van Gool, "From Images to 3D Models," Communications of the ACM **45**, 50-55 (2002).
35. F. Jiang, Y. Wu, and A. Katsaggelos, "Abnormal Event Detection from Surveillance Video," *ICIP* (IEEE, 2007), pp. 145-148.
36. A. Zomet and S. Peleg, "Efficient Super-Resolution and Application to Mosaics," *15th ICVR* (IEEE, 2000), pp. 579-583.
37. A. Smolic and T. Wiegand, "High-Resolution Video Mosaicing," *Proc. Of ICIP* (IEEE, 2001), pp. 872-875.
38. A. Camargo, Q. He, and K. Palaniappan, "Performance Evaluations for Super-Resolution Mosaicing on UAS Surveillance Videos," International Journal of Advanced Robotic Systems (2013).
39. S. Han, J. Bae, and M. Lee, "Intelligent control of a robot manipulator by visual feedback," Artificial Life and Robotics **4**, 156-161 (2000).
40. S. Hutchinson, G. Hager, P. Corke, "A Tutorial on Visual Servo Control," Robotics and Automation, IEEE Trans. On **12**, 1996, 651-670 (1996).
41. J. Johnson, "Analysis of Imaging Forming Systems," *Proc. Image Intensifier Symp.* (1958), pp. 249-273.
42. J. Donohue, *Introductory Review of Target Discrimination Criteria* (Academic, 1991).
43. R. Vollmerhausen, E. Jacobs, *The Targeting Task Performance (TTP) Metric: A New Model for Predicting Target Acquisition Performance* (Academic, 2006).
44. S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and Robust Multiframe Super Resolution," Image Processing, IEEE Transactions on **13**, 1327-1344 (2004).
45. J. Yang and T. Huang, "Image Super-Resolution: Historical Overview and Future Challenges," in *Super-Resolution Imaging*, Milanfar ed. (Academic, 2010), pp. 1-34.
46. D. Forsyth and J. Ponce, *Computer Vision* (Academic, 2013).
47. Z. Zhang, *A Flexible New Technique for Camera Calibration* (Academic, 2008).
48. B. Lambert, J. Ralph, L. Wren, and J. Dale, "Spherical Alignment of Imagers using Optical Flow Fields", *Proc. SPIE* **6238**, (2006).
49. "Photography – Electronic still picture imaging – Resolution and spatial frequency response," ISO 12233 (2014).
50. USAF-1951 Target, https://en.wikipedia.org/wiki/1951_USAF_resolution_test_chart.
51. D. Shumaker, J. Wood, and C. Thacker, *Infrared Imaging Systems Analysis*, (Academic, 1988).
52. "The Extended Yale Face Database B", University of California San Diego, http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.

53. P. Belhumeur, P. Hespanha, and D. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” PAMI, IEEE Transactions on **19**, 711-720 (July 1997).
54. A. Georghiades, P. Belhumeur, and D. Kriegman, “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose,” PAMI, IEEE Transactions on **23**, 643-660 (2001).
55. P. Jacobs, *Thermal Infrared Characterization of Ground Targets and Background* (Academic, 1996).
56. W. McCluney, *Introduction to Radiometry and Photometry* (Academic, 2014).
57. G. Villiers, N. Gordon, D. Payne, I. Proudler, I. Skidmore, K. Ridley, C. Bennett, R. Wilson, and C. Slinger, “Sub-pixel super-resolution by decoding frames from a reconfigurable coded-aperture camera: theory and experimental verification,” *Proc. SPIE* **7468**, (2009).
58. W. O’Neil, *Recent Progress in Microscan Resolution Enhancement* (Academic, 1999).
59. R. Hardie, “Super-Resolution Using Adaptive Wiener Filters,” in *Super-Resolution Imaging*, Milanfar ed. (Academic, 2010), pp. 35-61.
60. “Community Portal for Automatic Differentiation”, http://www.autodiff.org/.
61. E. Watson, R. Muse, and F. Blommel, “Aliasing and blurring in microscanned imagery,” in *Proc. SPIE* **1689**, 242-250 (1992).
62. A. Schaum and M. McHugh, “Analytical methods of image registration: Displacement Estimation and Resampling,” Naval Res. Lab **9298**, (1991).
63. L. Siong, S. Morki, A. Hussain, and N. Ibrahim, "Motion Detection Using Lucas Kanade Algorithm and Application Enhancement," *International Conference on Electrical Engineering and Informatics* (2009), pp. 537-542.
64. S. Babacan, R. Molina, and A. Katsaggelos, “Variational Bayesian super resolution,” Image Processing, IEEE Transactions on **20**, 984-999 (2011).
65. S. Villena, M. Vega, S. Babacan, . R. Molina, and A. Katsageelos, “Bayesian combination of sparse and non-sparse priors in image super resolution,” Digital Signal Processing **23**, 530-541 (2013).
66. S. Villena, M. Vega, R. Molina, and A. Katsaggelos, “A non-stationary image prior combination in super-resolution,” Digital Signal Processing **32**, 1-10 (2014).
67. S. Babacan, R. Molina, and A. Katsaggelos, “Variational Bayesian Super-Resolution Reconstruction,” in *Super-Resolution Imaging*, Milanfar ed. (Academic, 2010), pp. 285-313.
68. H. Zhang, Y. Zhang, H. Li, and T. Huang, “Generative Bayesian Image Super-Resolution with Natural Image Prior,” Image Processing IEEE Trans. On **21**, 4054-4067 (2012).
69. C. Bishop, *Pattern Recognition and Machine Learning* (Academic 2006).
70. L. Pickup, S. Roberts, and A. Zisserman, Capel, “Multiframe Super-Resolution from a Bayesian Perspective,” in *Super-Resolution Imaging*, Milanfar ed. (Academic, 2010), pp. 247-284.
71. R. Ranganath, S. Gerrish, and D. Blei, “Black Box Variational Inference,”
72. A. Kucukelbir, R. Ranganath, A. Gelman, and D. Blei, “Automatic Variational Inference in Stan,” *Proceedings of the 17th International Conference on AISTATS*, (2014).
73. B. Carpenter, D. Lee, M. Brubaker, A. Riddell, A. Gelman, B. Goodrich, J. Guo, M. Hoffman, M. Betancourt, and P. Li, “STAN: A Probabilistic Programming Language,” Journal of Statistical Software (2014).
74. *Stan Modeling Language User’s Guide and Reference Manual* (Academic, 2015).
75. M. Elad, *Sparse and Redundant Representations* (Academic, 2010).
76. S. Yu, W. Kang, S. Ko, J. Paik, “Single image super-resolution using locally adaptive multiple linear regression,” J. Opt. Soc. Am. A **32**, 2264-2275 (2015).
77. Yu, Kang, Ko, and Paik, “Single image super-resolution using locally adaptive multiple linear regression,” J. Opt. Soc. Am. A **32**, 2264-2275 (2015).
78. D. Gosgen-Neskin and I. Bar-Itzhack, “Unified Approach to Inertial System Error Modeling”, Journal of Guidance, Control, and Dynamics **15**, 648-653 (1992).
79. C. Hide, T. Moore, and M. Smith, "Adaptive Kalman Filtering Algorithms for Integrating GPS and Low Cost INS," *Position, Location, and Navigation Symposium* (Academic, 2004), pp. 227-233.
80. I. Rhee, M. Abdel-Hafez, and J. Speyer, “Observability of an Integrated GPS/INS During Maneuvers,” Aerospace and Electronic Systems IEEE Trans. On **40**, 526-535 (2004).

81. D. Gebre-Egziabher, R. Hayward, and J. Powell, "A Low-Cost GPS/Inertial Attitude Heading Reference System (AHRS) for General Aviation Applications," *Position Location and Navigation Symposium* (IEEE, 1998), pp. 518-525.
82. DIRSIG, http://www.dirsig.org/.
83. E. Lentilucci and S. Brown, "Advances in Wide Area Hyperspectral Image Simulation," *Proc. SPIE* **5075**, (2003).
84. "DIRSIG Generic Radiometry Solver," DIRSIG Wiki, http://www.dirsig.org/docs/d4-generic-radsolver.pdf.
85. J. Schott, C. Salvaggio, S. Brown, and R. Rose, "Incorporation of texture in multispectral synthetic image generation tools," *Proc. SPIE* **2469**, 189-196 (1995).
86. E. Ientilucci, *Synthetic Simulation and Modeling of Image Intensified CCDs (IICCD)*, PhD Thesis, (Academic, 2000).
87. J. Mason, J. Schott, C. Salvaggio, J. Sirianni, "Validation of contrast and phenomenology in the Digital Imaging and Remote Sensing (DIRS) lab's Image Generation (DIRSIG) model," *Proc. SPIE* **2269**, pp. 622-633 (1994).
88. K. Barcomb, J. Schott, S. Brown, and T. Hattenberger, "High-resolution slant-angle scene generation and validation of concealed targets in DIRSIG," *Proc. SPIE* **5546**, pp. 300-311 (2004).
89. OpenGL, https://www.opengl.org/
90. M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide* (Academic, 1999).
91. "Performance Specification Digital Terrain Elevation Data (DTED)", MIL-PRF-89020B (Academic, 2000).
92. K. Chen, G. Zhao, Z. Meng, J. Yan, and H. Lu, "Equivalent Approaches to Equations of Traditional Transfer Alignment and Rapid Transfer Alignment", *Proceedings of the 7th World Congress on Intelligent Control and Automation* (Academic, 2008), pp. 892-895.
93. L. Joon and L. You-Chol, "Transfer alignment considering measurement time delay and ship body flexure," Journal of Mechanical Science and Technology **23**, 195-203 (2009).
94. K. Shortelle, W. Graham, and C. Rabourn, "F-16 Flight Test of a Rapid Transfer Alignment Procedure," *IEEE Position, Location, and Navigation Symposium* (IEEE, 1998), pp. 379-386.
95. M. Veth and J. Raquet, *Alignment and Calibration of Optical and Inertial Sensors Using Stellar Observations* (Academic, 2005).
96. G. Fasano, D. Accardo, A. Moccia, and A. Rispoli, "An Innovative Procedure for Calibration of Strapdown Electro-Optical Sensors Onboard Unmanned Air Vehicles," Sensors **10**, 639-654 (2010).
97. J. Shi and C. Tomasi, "Good Features to Track", *CVPR* (IEEE, 1994), pp. 593 - 600.
98. C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: A Factorization Method: Full Report on the Orthographic Case," CMU Technical Report **CMU-CS-92-104**, (1992).
99. Silverfox, http://www.satnews.com/cgi-bin/story.cgi?number=1554529420.
100. B. Stevens, and F. Lewis, *Aircraft Control and Simulation* (Academic, 1992).
101. J. Blakelock, *Automatic Control of Aircraft and Missiles* (Academic, 1991).
102. M. Lizarraga, *Autonomous Landing System for a UAS* (Academic, 2004).
103. J. Herbert, J. Keith, S. Ryan, M. Szannes, G. Lachapelle, and M. Cannon, "DGPS Kinematic Carrier Phase Signal Simulation Analysis for Precise Aircraft Velocity Determination", Navigation **44**, 231-245 (1997).
104. L. Serrano, D. Kim, and R. Langley, "A GPS Velocity Sensor: How Accurate Can it Be? – A First Look," *ION NTM* (Academic, 2004).
105. S. Gabarda and G. Cristobal, "Blind image quality assessment through anisotropy," J. Opt. Soc. Am. A **24**, 42-51 (2007).
106. T. Celik and T. Tjahjadi, "Automatic Image Equalization and Contrast Enhancement Using Gaussian Mixture Modeling," Image Processing, IEEE Trans. On **21**, 145-156 (2012).
107. K. Panetta, E. Wharton, and S. Agaian, "Human Visual System Based Image Enhancement and Logarithmic Contrast Measure," Systems, Man, and Cybernetics Part B: Cybernetics **38**, 174-188 (2008).
108. A. Beghdadi and A. Negrate, "Contrast enhancement technique based on local detection of edges," Computer Vis., Graphics, and Image Processing **46**, 162-174 (1989).
109. S. Chen and A. Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement," Consum. Electron., IEEE Transactions on **49**, 1310-1319 (2003).

110. P. Bijl and M. Valeton, "Triangle orientation discrimination: the alternative to minimum resolvable temperature difference and minimum resolvable contrast," Opt. Eng. **37**, 1976-1983 (1998).
111. *Tau 640 Slow Video Camera User's Manual*, FLIR Commercial Systems (2011).
112. Y. Shechtman, Y. Eldar, A. Szameit, M. Segev, "Sparsity based sub-wavelength imaging with partially incoherent light via quadratic compressed sensing," Optics Express **19**, 14807-14822 (2011).
113. J. Harris, "Resolving power and decision theory," J. Opt Soc. Am. **54**, 606-611 (1964).
114. L. Lucy, "Statistical limits to superresolution," Astronomy and Astrophysics **261**, 706-710 (1992).
115. S. Villena, M. Vega, S. Babacan, J. Mateos, R. Molina and A. Katsaggelos, "Superresolution software manual," (2007).
116. R. McCluney, *Introduction to Radiometry and Photometry* (Academic, 2014).
117. S. Villena, M. Vega, R. Molina, and A. Katsaggelos, "Bayesian super-resolution image reconstruction using an l1 prior," *6th International Symposium on Image and Signal Processing and Analysis* (IEEE, 2009), pp. 152-157.
118. T. Lomheim and E. Hernandez-Baquero, "Translation of spectral radiance levels, band choices, and signal-to-noise requirements to focal plane specifications and design constraints," *Proc. SPIE* **4486**, 263-307 (2002).
119. G. Koretsky, J. Nicoll, and M. Taylor, *A Tutorial on Electro-Optical/Infrared (EO/IR) Theory and Systems* (Academic, 2013).
120. V. Melzer, H. Heckmanna, C. Ritter, J. Barenz, and M. Raab, "Fast and precise point spread function measurements of IR optics at extreme temperatures based on reversed imaging conditions," *Proc. SPIE* **7662**, 766213-1 - 766213-18 (2010).
121. C. Loebich, D. Wueller, B. Klingen, and A. Jaeger, "Digital Camera Resolution Measurement Using Sinusoidal Siemens Star," *Proc. SPIE* **6502**, 1-11 (2007).
122. G. Birch and J. Griffin, "Sinusoidal Siemens star SFR measurement errors due to misidentified target centers," Opt. Eng. **54**, (2015).
123. F. Dellaert, C. Thorpe, and S. Thrun, "Super-Resolved Texture Tracking of Planar Surface Patches," *Proc. International Conf. on Intelligent Robots and Systems* (IEEE, 1998), pp. 197-203.
124. NVIDIA CUDA Sparse Matrix library (cuSPARSE), https://developer.nvidia.com/cusparse.
125. E. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Proc. Of AS-SPCC* (IEEE, 2000), pp. 153-158.
126. G. Subrahmanyam, A. Rajagopalan, and R. Aravind, "Unscented Kalman Filter for Image Estimation in Film-Gain Noise," *ICIP* (IEEE, 2007), pp. 17-20.
127. S. Julier and J. LaViola, "On Kalman Filtering with Nonlinear Equality Constraints," Signal Proc. IEEE Trans. On **55**, 2774-2784 (2007).
128. J. Nocedal and S. Wright, *Numerical Optimization* (Academic, 2006).
129. TOMLAB MATLAB Automatic Differentiation, http://tomopt.com/tomlab/products/mad/.
130. F. Mufti, R. Mahony, and J. Kim, "Super-Resolution of Speed Signs in Video Sequences," *9th Biennial Conf. of the Australian Pattern Recognition Society* (IEEE, 2007), pp.278-285.
131. N. Woods, N. Galatsanos, and A. Katsaggelos, "EM-Based Simultaneous Registration, Restoration, and Interpolation of Super-resolved Images," *Proc. ICIP* (IEEE, 2003), pp. 303-306.
132. Variational Bayesian Inference Super-Resolution Matlab Software, http://decsai.ugr.es/pi/superresolution/software.html.
133. Open Computer Vision (OpenCV) library, http://opencv.org.
134. G. Bradski and A. Kaehler, *Learning OpenCV* (Academic 2008).
135. R. Carroll, M. Agrawala, A. Agarwala, "Optimizing content-preserving projections for wide-angle images," *ACM Transactions on Graphics – Proceedings of ACM SIGGRAPH* (Academic, 2009).
136. D. Zorin and A. Barr, ``Correction of geometric perceptual distortions in pictures," *Proc. SIGGRAPH* (Academic, 1995), pp. 257-264.
137. A. Katsaggelos, "Iterative Image Restoration Algorithms", *Digital Signal Processing Handbook*, Ed. Vijay K. Madisetti and Douglas B. Williams (Academic, 1999).
138. R. Schafer, R. Mersereau, and M. Richards, "Constrained Iterative Restoration Algorithms," Proc. IEEE **69**, 700-703 (1981).
139. FLIR Systems, http://www.flir.com.

140. Q. He and R. Schultz, "Super-Resolution Reconstruction by Image Fusion and Application to Surveillance Videos Captured by a Small Unmanned Aircraft System", in *Sensor Fusion and Its Applications*, Ciza Thomas ed. (Academic, 2010), pp. 475-485
141. F. Azevedo, *Complete System for Quadcopter Control*, Computer Engineering Thesis (Academic, 2014).
142. J. Simpkins and R. Stevens, "An Introduction to Super-Resolution Imaging", Mathematical Optics: Classical, Quantum, and Computational Methods (Academic, 2012).
143. E. Tautu and F. Leon, "Optical Character Recognition System Using Support Vector Machines", (2010).
144. A. Thome, "SVM Classifiers - Concepts and Applications to Character Recognition", in *Advances in Character Recognition*, Xiaoqing Ding eds. (Academic, 2012).
145. Y. Bengio, A. Courville, P. Vincent, "Representation Learning: A Review and New Perspectives", PAMI , IEEE Trans on **35**, 1798-1828 (2013).
146. H. Ishida, S. Yanadume, T.Takahashi, I. Ide, Y. Mekada, and H. Murase, "Recognition of low-resolution characters by a generative learning method," (2005), pp. 45-51.
147. C. Segall, A. Katsaggelos, R. Molina, and J. Mateos, "Bayesian Resolution Enhancement of Compressed Video," Image Processing, IEEE Transactions on **13**, 898-911 (2004).
148. DJI Phantom 3 4K User's Manual, http://download.dji-innovations.com/downloads/phantom_3/en/Phantom_3_Professional_User_Manual_v1.0_en.pdf.
149. DJI Phantom 2 User's Manual, http://dl.djicdn.com/downloads/phantom-2-vision/en/Phantom_2_Vision_User_Manual_v1.8_en.pdf.
150. FLIR Vue^{Pro} User's Guide, http://www.flir.com/uploadedFiles/sUAS/Products/Vue/FLIR-VUE-Users-Guide.pdf.
151. Imatest software, http://www.imatest.com.
152. O. Sengul, M. Turkmenoglu, and L. Yalciner, "MTF Measurements for the Imaging System Quality Analysis," Gazi University Journal of Science **25**, 19-28 (2012).
153. G. Boreman, *Modulation Transfer-Function in Optical and Electro-Optical Systems* (Academic, 2001).
154. X. Zhang, T. Kashti, D. Kella, T. Frank, D. Shaked, R. Ulichney, M. Fischer, and J. Allebach, "Measuring the Modulation Transfer Function of Image Capture Devices: What Do the Numbers Really Mean?" *Proc. SPIE* **8293**, (2012).
155. D. Williams and P. Burns, "Diagnostics for Digital Capture using MTF," *Proc. IS&T PICS Conf.* (2001), pp. 227-232.
156. V. Sukumar, H. Hess, K. Noren, G. Donohoe, and S. Ay, "Imaging System MTF – Modeling with Modulation Functions," IECON (IEEE,2008), pp. 1748-1753.
157. Slanted Edge MTF in ImageJ, https://imagej.net/plugins/se-mtf/index.html.
158. B. Lei, E. Hendriks, and A. Katsaggelos, "Camera calibration for 3D reconstruction and view transformation," in *3D Modeling and Animation: Synthesis and Analysis Techniques for the Human Body*, N. Sarris and M. Strintzis, eds. (Academic, 2005), pp. 70–129.

APPENDIX A

DRONES AND CAMERAS FOR PERFORMANCE EVALUATION

In order to perform our testing of algorithms for enhancement of imagery collected from airborne sensors, we make extensive use of quadcopter drones. The two available, shown in Figure A-1, are the DJI Phantom 3 [148] with an attached visible band RGB camera on a 2-axis stabilized gimbal and a DJI Phantom 2 [149] custom fit with a FLIR Vue^{Pro} long wave infrared (LWIR) uncooled micro bolometer [150]. Figure A-2 shows a comparison of a common scene viewed by both drones in flight.

Figure A-1: DJI Drones Used for Experiments. Phantom 3 with RGB Visible Band (left). Phantom 2 with LWIR Micro Bolometer (right)

Figure A-2: Comparison of RGB visible and LWIR images

One of the most notable differences between the visible and LWIR bands is the LWIR performance at night. Figure A-3 shows a comparison of a common scene imaged both during the day (bottom row) and night (top row) with both the visible RGB camera (right column) and the LWIR camera (left column). Although the day picture for the LWIR camera shows a few hot spots which are being heated by direct sunlight, the two images are very similar. In contrast, of course, the RGB picture at night shows very little detail.

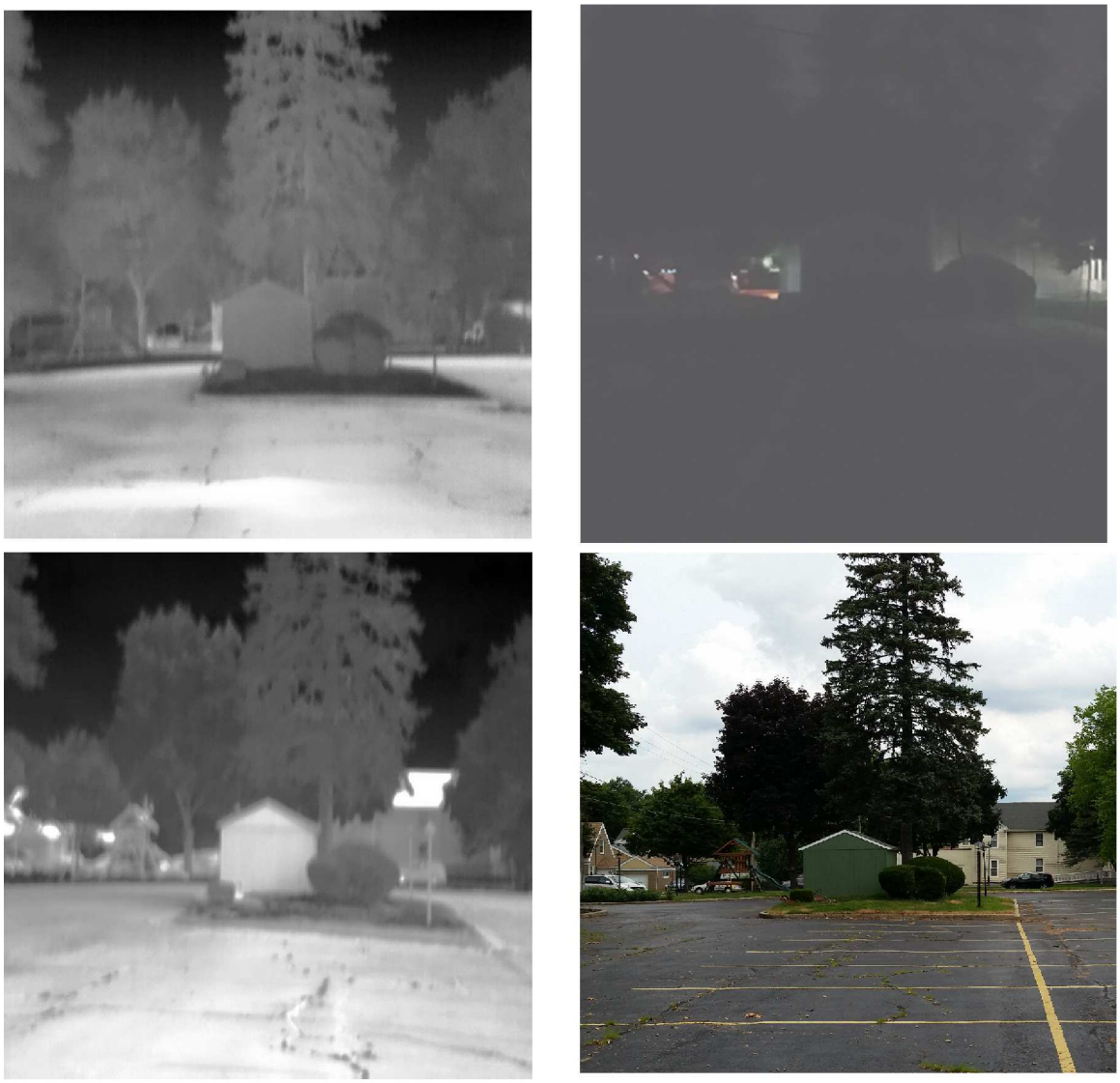


Figure A-3: Comparison of LWIR (left) and RGB (right) cameras during night (top) and day (bottom)

A.1 Phantom 3 Quadcopter with attached visible RGB Camera

For visible band, in-flight data collections we use the commercially available Phantom 3 quadcopter drone provided by DJI corporation [148]. This is a small, battery-powered, relatively inexpensive drone (approximately $1200 US). The Phantom 3 has several advantages for scientific data collection over other options commercial options:

1. It allows manual control of the exposure time and gain, or ISO. Automatic adjustment, which is typically the only option in less expensive systems, can over-expose or under-expose the critical part of the image.

2. Video is captured using MPEG-4 compression (up to 30 Hz); however, a single or a series of still images may be collected in raw, uncompressed digital negative (DNG) format. The raw data maintains the full dynamic range of the 16-bit per channel camera.

3. It is portable and inexpensive to operate; thereby, allowing for convenient and repeated experimentation.

4. Data from the flight controller, including 6 d.o.f. position and attitude, is recorded and available for data processing.

As the purpose of the device is to enable high-quality aerial imaging, the camera is mounted on a 3-axis gimbal which, using feedback from inertial sensors, is able to counter attitude motion and stabilize the line of sight of the camera as the vehicle buffets in the wind. In addition, the Phantom 3 utilizes an on-board Global Positioning System (GPS) to autonomously stabilize its position; thereby, allowing the operator to focus on photography.

The Phantom 3 possesses a 4K Sony EXMOR camera. The fixed f/2.8 lens has a diagonal field-of-view of 94 degrees, a 20mm focal length (35mm equivalent), and is permanently focused at infinity. ISO sensitivity and electronic shutter speed may be either manually or automatically controlled. During testing, we are able to avoid motion blur degradation by selecting a 1/1000 s frame exposure time.

The Phantom 3 allows for 30 fps video compressed as MPEG4. It also has the ability to capture uncompressed single-frame (or single-frame burst) images in a raw, 16-bit RGB format. During testing, we avoid additional image degradation artifacts due to compression by collecting only uncompressed image sequence in the auto 7-frame burst mode. The 7 captured frames are then passed to the SR algorithms. The Phantom 3 battery allows for ~30 minutes of continuous

hover time which is sufficient to capture about 20 sets of 7-frame burst uncompressed images. The principal time limitation is time required to transfer the 7 uncompressed images from the device memory to the onboard micro-SD card prior to capturing the next burst.

The coordinate systems for both the Phantom 3 aircraft platform and the independent, 3-axis gimbal which houses the camera are shown in Figure A-4. The six instantaneous rotations about these two axis systems, relative to a local north-east-down (NED) coordinate system are captured during flight and available for post-processing. The combination of the two allows us to compute the direction cosine matrix (DCM) relating the camera pose to the world coordinate system as discussed in 2.5.

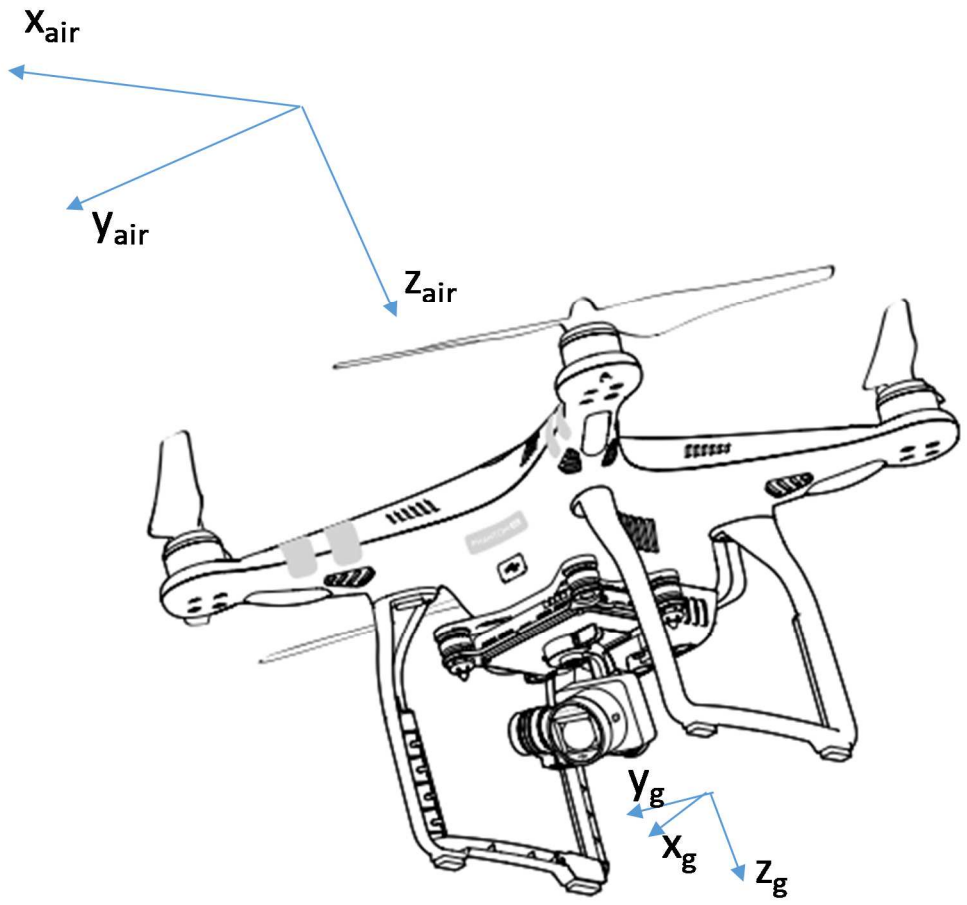


Figure A-4: Phantom 3 Coordinate Systems for the aircraft platform ("air") and the gimbal ("g")

A.2 DJI Phantom 2 with FLIR Vue^{Pro} LWIR Camera

Our second data collection platform is the DJI Phantom 2 drone custom modified to house a long wave infrared camera (the FLIR Vue^{Pro} [150]). The Phantom 2 is a little bit older version of the DJI Phantom 3 discussed in the previous section. Otherwise, the flight capabilities of the Phantom 2 are very similar to those of the Phantom 3.

The FLIR Vue^{Pro} is a 640 x 512 pixel, uncooled LWIR micro-bolometer sensitive in the 7.5 to 13.5 micron wavelength band. We use a 19 mm aperture lens which produces a 32° x 26°

field of view. The camera is able to either capture raw 16-bit, uncompressed images in a Tagged Image File Format (TIFF) format or 30 Hz video is a compressed H264 format. The images and video are stored, during flight, on a microSD card and available for post-flight extraction.

A.3 Commercial Samsung Galaxy 5 Camera

We supplement our results with visible band imagery captured from the RGB camera embedded in the Samsung Galaxy 5 smartphone (shown in Figure A-5). This camera has characteristics typical of inexpensive grade cameras typically found in commercial electronics such as cell phones.

Figure A-5: Samsung Galaxy 5 smartphone with embedded RGB camera

The embedded camera has a 16 mega-pixel, f/2.2, 31 milli-meter format with an ~90 degree field of view. It collects 24-bit RGB imagery which is only available in a Joint Photographic Experts Group (JPG) compressed format.

A.4 FLIR Cooled Mid-Wave Infrared Camera

The final camera that is used for our testing is a high quality, cooled Mid-Wave Infrared (3.5 to 5.0 micron spectral bandpass) available from FLIR systems [139] and mounted on an aircraft. This is a wide field of view camera of ~110 degrees. However, as is typical of infrared

cameras, as mentioned in 2.3, fabrication cost and complexity limits the pixel format to 320 x 320. The camera outputs uncompressed video data at a rate of 30 Hz and is time synchronized to measurements from the aircraft inertial navigation system (INS) via the Inter-Range Instrumentation Group (IRIG) IRIG-B timecode. Alignment information for the camera, INS, as well as intrinsic camera parameters such as center pixel, distortion, and MTF are supplied by the manufacturer.

APPENDIX B

CAMERA CALIBRATION MEASUREMENTS

In order to use cameras for experimental and developmental purposes, it is most often necessary to perform a level of calibration measurement. The reasons are threefold:

1. **Performance Prediction:** Regardless if performed using rule-of-thumb equations or detailed, simulation based analysis, performance will be a function of camera physical characteristics.

2. **Explicit Use by Algorithms:** The most powerful of image processing and analysis algorithms employ some explicit mathematical model of the camera which are provided as an input.

3. **Test and Evaluation:** Even if camera parameters aren't explicitly input to an algorithm, the algorithm may internally estimate them. Independent calibration and measurement of the parameters provides a "ground truth" for comparison.

The set of parameters needed to support the image formation models of chapter 2 as well as other standard models fall into two categories. The first are the set of geometric projection properties commonly referred to as the "camera intrinsic parameters" [46,134]. These include pixel pitch, focal length, optical center, and radial/tangential distortion. The category is a model of the camera MTF. Unfortunately, particularly for low and moderate cost systems, there can be significant variety in both the availability and detail of information provided by the manufacturer. Frequently, when information is available, it is design parameters as opposed to measurements for a specific serial number. As a consequence, it is often necessary for the end user to measure the parameters directly.

In general, calibration is typically performed by imaging a "target" with known characteristics and then inverting the image formation model to estimate the parameters that are responsible for generating the observed image. For the set of geometric projection parameters, this is most easily accomplished by using the "checkerboard" target imaged at various, arbitrary positions and orientations. Various algorithms [47,134,157] are able to use the images to simultaneously estimate both the extraneous position/orientation information ("extrinsic parameters") as well as the desired intrinsic parameters.

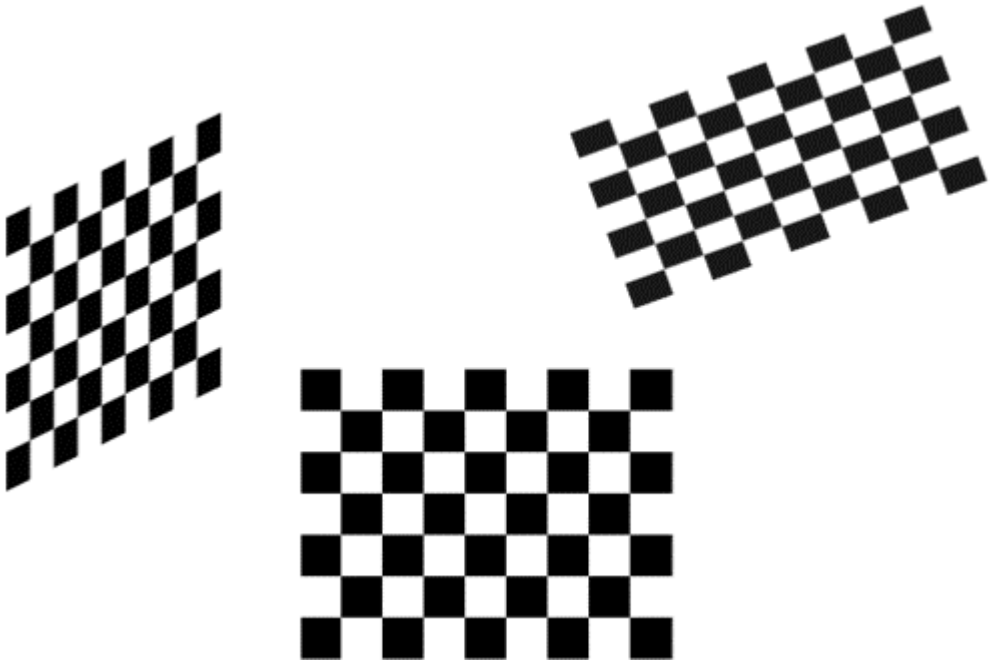


Figure B-1: Chessboard calibration target for geometric projection parameter calibration

Similarly, there are a variety of standard targets available for estimating the camera MTF. Figure B-2 shows several standard targets. The upper left is a version of the ISO 12233 target available from Cornell University [49] which supports a variety of measurements, the bottom left is the Sinusoidal Siemens star target as used by the Imatest software suite [151], and the upper

right is the 1951-USAF standard resolution target [50]. Overall, MTF measurement is more difficult than the geometric perspective measurements discussed above. Each of the targets in Figure B-2 and, if any, associated software packages vary in their applicability to different measurement conditions, camera types, and amenability to automation.

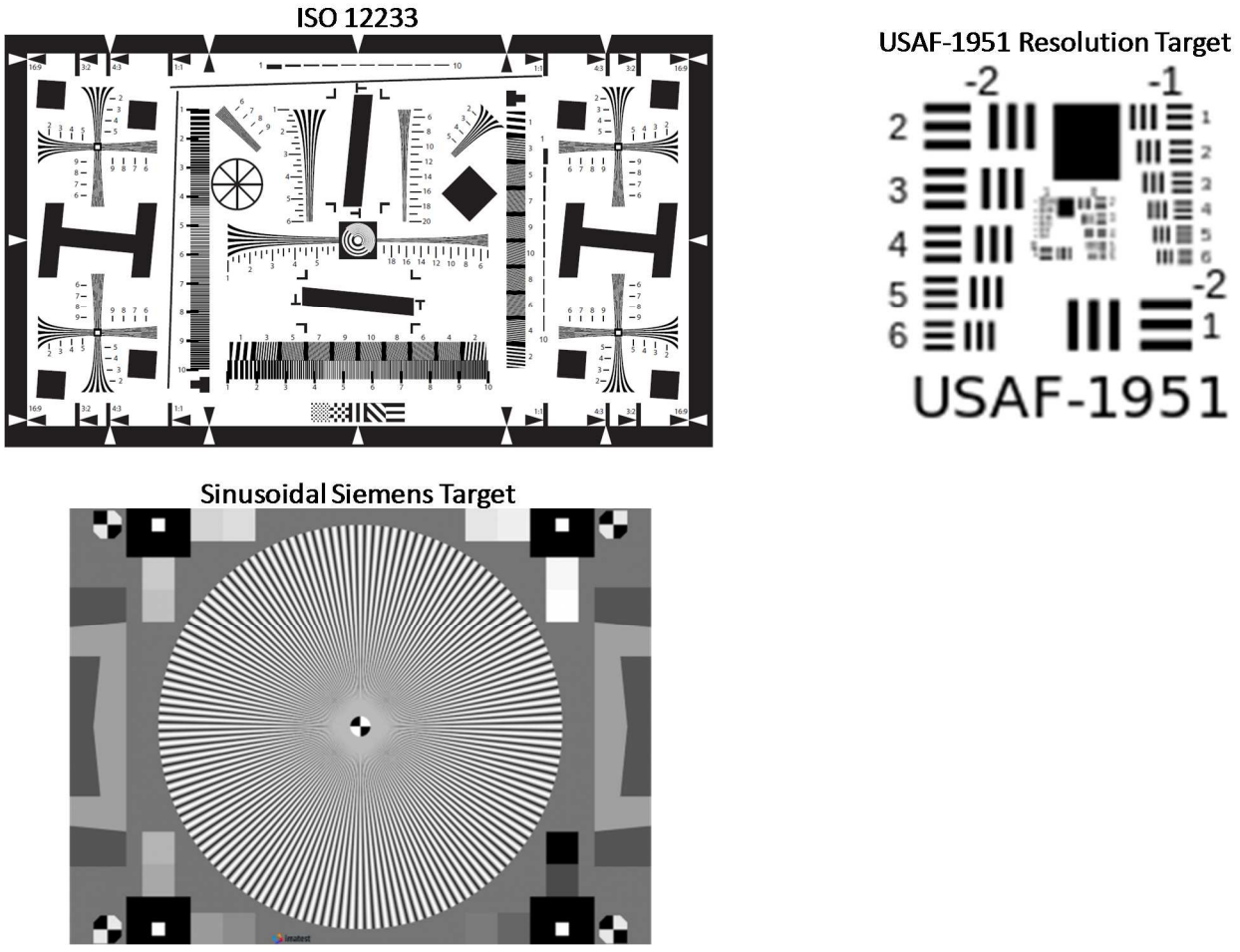


Figure B-2: Standard MTF measurement targets

In this appendix, we provide results for the calibration of the cameras used for our experiments. The four cameras are: DJI Phantom 3 drone with gimbaled 4000 x 3000 pixel RGB visible camera, DJI Phantom 2 drone with gimbaled 640 x 512 pixel Longwave Infrared (7.5 –

13.5 μm), Samsung Galaxy 5 smartphone with embedded m x n RGB visible camera, and a wide field-of-view (WFOV) 320 x 320 pixel MidWave Infrared camera (3.5 – 5.0 μm).

B.1 Characterization of the Phantom 3 Drone

Geometric Projection

As the RGB camera on the Phantom 3 drone has a wide FOV of 95 degrees, it is necessary to measure its geometric properties, particularly distortion. To do so, we use the OpenCV calibration package [133,134] which involves imaging the chessboard target of Figure B-1 at a variety of positions and poses. The geometric model used by OpenCV first comprises of a "pin-hole" camera model where the transformation of a point $[X \quad Y \quad Z]^T$ in a 3D Cartesian space fixed to the camera (with the Z-direction parallel to the optical axis) to a location (x, y) in 2D pixel space is given by

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{pmatrix} f_x & 0 & cx \\ 0 & f_y & cy \\ 0 & 0 & 1 \end{pmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{B.1.1}$$

where (cx, cy) is the location of the center of the optical axis in camera pixel coordinates and f_x, f_y are the normalized ratios focal_length/pixel pitch in the horizontal (x) and vertical (y) directions of the camera. s is an arbitrary scale factor used to ensure the third element of the vector on the left-hand side of (B.1.1) is equal to unity.

The model in (B.1.1) is valid for an idealized, pin-hole camera. Real cameras, particularly wide FOV cameras, possess both radial and tangential distortion [46,134]. One artifice to handle distortion is to apply a correction to the raw, measured pixel location (x', y') into the corrected pixel location (x, y) which is then used in (B.1.1) as well as any geometric manipulations on the

camera. In the model used by OpenCV, raw pixel locations are first corrected for radial distortion by

$$\boldsymbol{x_c = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)}, \text{ and}$$
$$\boldsymbol{y_c = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)} \tag{B.1.2}$$

where $\boldsymbol{(x_c, y_c)}$ is the raw pixel location $\boldsymbol{(x', y')}$ corrected for radial distortion and $\boldsymbol{(k_1, k_2, k_3)}$ are the camera specific parameters describing the distortion. The radius $\boldsymbol{r}$ is the distance to the optical center; i.e. $\boldsymbol{r = \sqrt{(x' - cx)^2 + (y' - cy)^2}}$. Once corrected for radial distortion, the model allows for a correction of tangential distortion by

$$\boldsymbol{x = x_c + 2p_1 y_c + p_2(r^2 + 2x_c^2)}, \text{ and}$$
$$\boldsymbol{y = y_c + p_1(r^2 + 2y_c^2) + 2p_2 x_c} \tag{B.1.3}$$

where the polynomial coefficients $\boldsymbol{(p_1, p_2)}$ are unique to the camera. Figure B-3 shows 1 out of 17 total chessboard images used for the Phantom 3 calibration. The final calibration results were:

$\boldsymbol{f_x}$ = 2313.4
$\boldsymbol{f_y}$ = 2319.4
$\boldsymbol{(cx, cy)}$ = (1977.8,1508.8)
$\boldsymbol{(k_1, k_2, k_3)}$ = 0, 0, 0)
$\boldsymbol{(p_1, p_2)}$ = (0, 0)

Note than $\boldsymbol{f_x}$ and $\boldsymbol{f_y}$ may also be interpreted as the sample frequency of the camera, in units of 10^{-3} cycles / milli-radian, at the center of the camera; i.e., the sample frequency is ~2.3 cycles / milli-radian. Radial distortion will alter the effective sample frequency away from the center in a predictable fashion per (B.1.1 to B.1.3). For the Phantom 3 camera, the calibration software found that both the radial distortion as well as tangential distortion coefficients were close enough to 0 as to be ignored.

Figure B-3: Chessboard target imaged by Phantom 3 for geometric calibration

MTF

Unlike geometric calculation, there are many methods described in the literature for measuring MTF [49,121,122,152-156]. We use two techniques. The first is a static measurement on the ground using a slant edge target and the ImageJ MTF plugin [157]. The second is in flight using a custom version of the Sinusoidal Siemens Target [49,121,122] and custom reduction method.

The slant edge method works by directly measuring the edge spread function (ESF) across the step discontinuity in the target. The algorithm then computes the line spread function (LSF) as the derivative of the ESF. The MTF, in a direction perpendicular to the edge, is the Fourier transform of the LSF. The method requires an edge that is slanted relative to the pixel rows and columns. This guarantees different pixel phasing along the edge. With a principle similar to SR, this has the effect of allowing measurement all the way up to a spatial frequency of 1.0 cycles / pixel. For the for the Phantom 3 drone, the slant-edge was printed on paper and imaged with the drone stationary on the ground. The relevant measurements are shown in Figure B-4. The image is in the upper-left, the ESF is in the upper-right, the computed LSF is in the lower-left, and the computed MTF is in the lower-right. The imageJ algorithm computed the “pre-sample” MTF meaning that it doesn’t include the frequency response associated with pixel integration. For comparison, the measured MTF in Figure B-4 is plotted along with a pure Gaussian MTF with σ

= 1.2 pixels. In general, most real camera responses are longer tailed than the Gaussian. Consequently, a typical Gaussian model will tend to show a higher response at lower frequencies and lower response at higher frequencies.

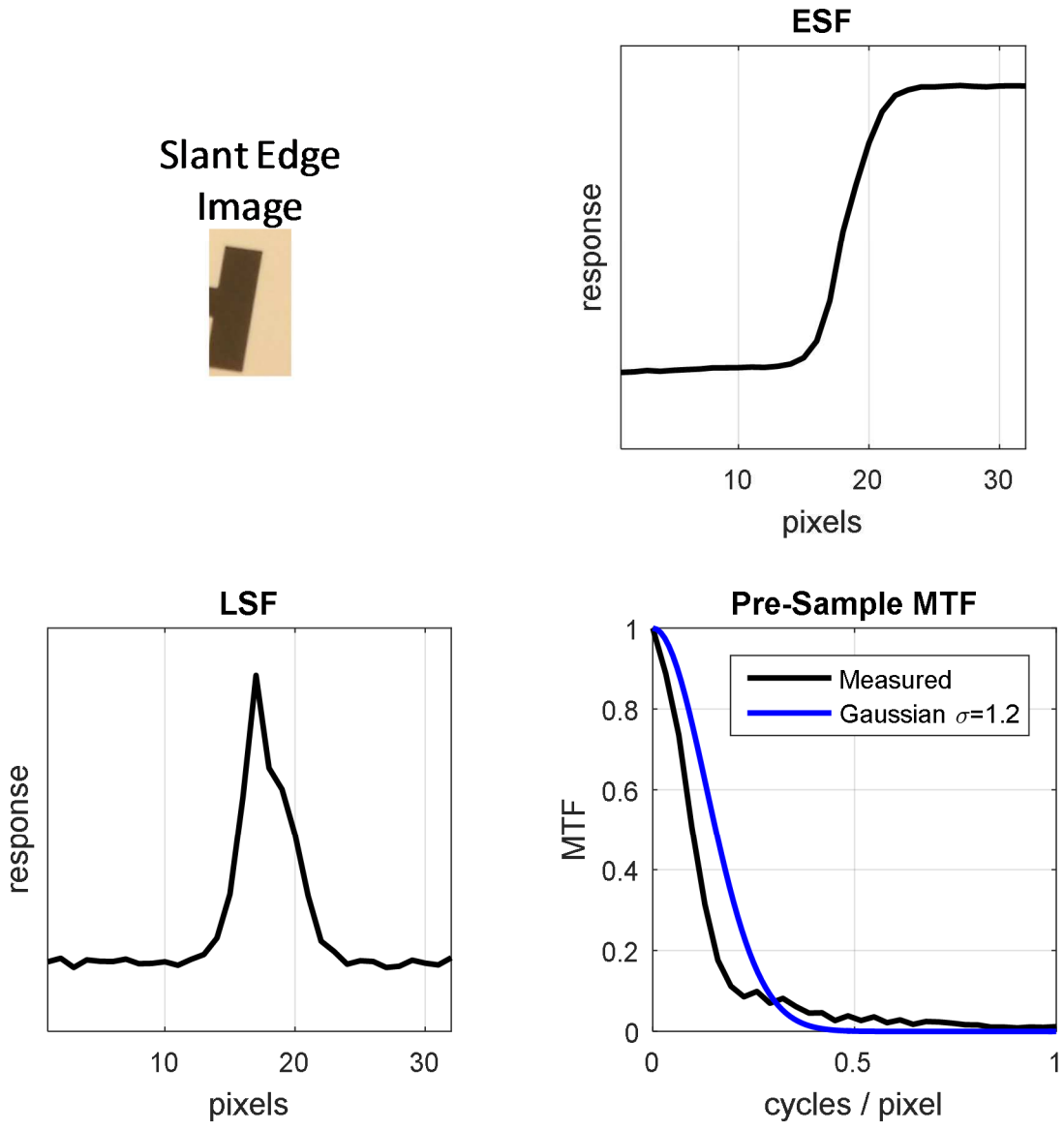


Figure B-4: Slant edge MTF measurement for the Phantom 3 drone

The second method, which was used to measure MTF while the drone was in flight, was to use a printed version of the sinusoidal Siemens star chart, as shown in Figure B-5, as a calibration target. In order to accommodate a variety a viewing distances as the drone maneuvers, the target board contains 2 star patterns of 20.3 cm diameter each. The pattern on the left contains 12 total cycles around the circumference and the pattern on the right contains 3 total cycles around the circumference. A sample of the target board viewed from the drone at 20m altitude directly above is shown in Figure B-5 (additional elements on the board are used for remote recognition testing in Chapter 6). The translation between cycles per circumference and a spatial frequency of cycles per pixel depends upon both the altitude of the drone above the target board as well as the pixel radius r in the image where we extract pixel intensities. Using both target patterns extends the range of spatial frequencies we can measure from a single image.

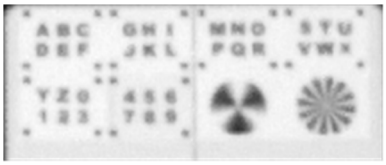


Figure B-5: Target board with Siemens star targets imaged from 20m altitude

To mechanize the MTF measurement, we locate the center of the star target in the image and then scan around the target at a radius of r pixels and extract intensities of adjacent pixels as shown in Figure 11. An example extraction is shown in Figure B-7 which corresponds to a spatial frequency of 0.11 cycles/pixel.

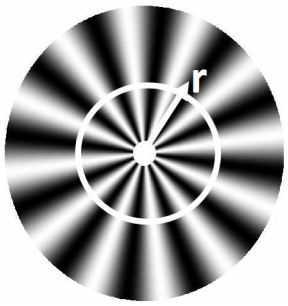


Figure B-6: Measurement of MTF at radius r. Spatial frequency for this target will be $12/2\pi r$ cycles/pixel.

At a radius $\boldsymbol{r}$, we have $\boldsymbol{2\pi r}$ pixels. If there are $\boldsymbol{k}$ cycles around the circumference of the star target, the spatial frequency measured by extracting the series of pixels at radius $\boldsymbol{r}$ is $\boldsymbol{k/2\pi r}$ cycles/pixel. We can measure the spatial frequency response up to $\boldsymbol{r = k/\pi}$ pixels from the center (corresponds to a max spatial frequency of 0.5 cycles/pixel) without concerns of aliasing.

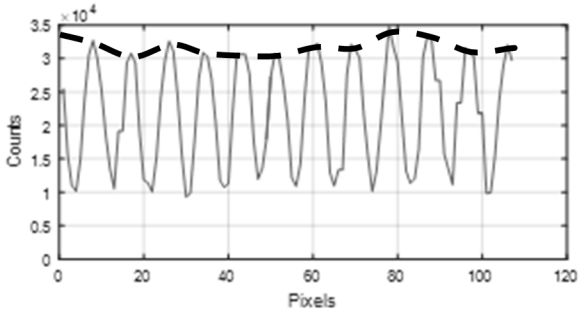


Figure B-7: Sample star target extraction

The common method in the literature is to extract a measurement of the contrast ratio at a particular spatial frequency $\boldsymbol{f}$ as

$$\boldsymbol{H(f) = \frac{1}{H(0)} \frac{I_{max}(f) - I_{min}(f)}{I_{max}(f) + I_{min}(f)}} \quad \text{(B.1.4)}$$

where $\boldsymbol{I_{max}(f)}$ and $\boldsymbol{I_{min}(f)}$ correspond to the maximum and minimum intensities as observed in the extraction from Figure B-7. $\boldsymbol{H(0)}$ represents the DC gain of the image and typically must be measured from a region of the image with very low frequency. The problem, as can be seen by the dashed line in Figure B-7 is that there are other, superfluous frequency components in the

extraction other than the principal frequency we are trying to measure. These are caused by non-uniform shading of the target, etc. We, therefore, instead use a method of first performing a DFT on the image, which separates the response at the principal frequency from the other components. This is illustrated for the 0.11 cycles/pixel case in Figure B-8.

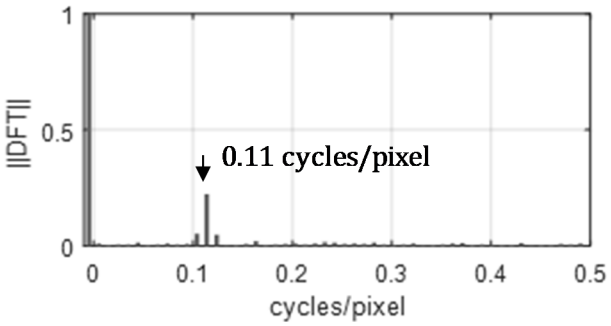


Figure B-8: DFT of MTF star target extraction where the spatial frequency measurement is at 0.11 cycles/pixel

We performed the MTF measurement of the Phantom 3 while the UAS was in a hover configuration 20m directly above the target board. For the collection, the shutter speed was set to the same 1/1000 s as will be used for the text character collections; so, any additional motion induced blur is consistent between the MTF measurement and the later character measurement experiments. In this geometry, the rightmost star target in Figure B-5, labeled "Target 1" in Figure B-9, provides measurements at spatial frequencies ranging from 0.07 to 0.16 cycles/pixel. The leftmost star target, labeled "Target 2" in Figure B-9, provides spatial frequency measurements ranging from 0.21 to 0.50 cycles/pixel.

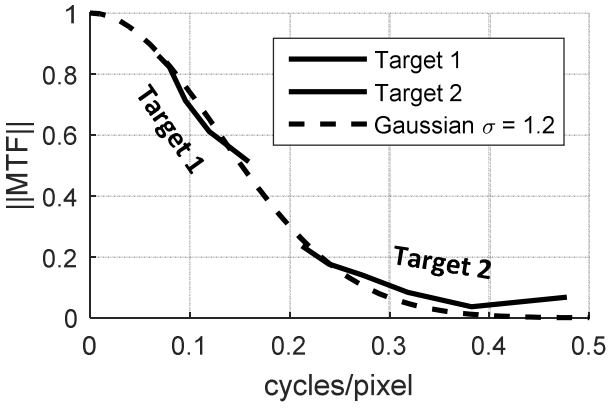


Figure B-9: Measured MTF at 20m altitude during hover

The best fit blur model to the measurements is given by the dashed line in Figure B-9 which is a combination of a Gaussian blur with $\boldsymbol{\sigma}$ = 1.20 pixels and the 100% fill factor rectangular pixel integration function. In the frequency domain, the image formation model is, therefore, given by

$$\boldsymbol{I(f_x, f_y) = exp\left(-2\pi^2\sigma^2\left(f_x^2 + f_y^2\right)\right)\ sinc(\pi f_x) sinc(\pi f_y),} \qquad \text{(B.1.5)}$$

where $\boldsymbol{f_x}$ and $\boldsymbol{f_y}$ represent spatial frequencies in units of cycles/pixel. Equation (B.1.5) matches the imaging model as introduced in chapter 2.

Figure B-9 shows that the sinusoidal Siemens measurement is consistent with the slant edge method. One subtle difference between the two methods is that the slant edge method measures the pre-sample MTF whereas the star target method measures the post-sample MTF. The post-sample MTF is accounted for in the Gaussian overlay in Figure B-9 by incorporating the additional term $\boldsymbol{sinc(\pi f_x) sinc(\pi f_y)}$ representing the MTF of the pixel integration. Both methods are consistent with respect to the observation that at spatial frequencies of 0.25 cycles/pixel and above, the measured blur has higher gain than the Gaussian model.

As a comparison of methodology, we also attempted to measure the MTF using the Richardson-Lucy method [117] as made available in the Mathworks Matlab image processing

toolbox function ‘deconvblind’. Our observation was that this method severely underestimated the higher frequency MTF gain relative to what is shown in Figure B-9. Without further investigation, as stated above, we believe this error was due to the fact that this algorithm does not handle the presence of aliasing in the source image.

B.2 Characterization of the Samsung Galaxy 5 Camera

Geometric Projection

All of our experiments using the Samsung Galaxy 5 camera are performed with images near the center of the field-of-view. Therefore, we did not need to characterize the geometic projection as completely as that of the Phantom 3 drone in B.1. The only geometric parameter of interest is the IFOV which is measured based on the number of spanning a known angle in the image. It is measured at 0.23 milli-radians. This produces $\boldsymbol{\omega_s}$ = 4.35 cycles / milli-radian and $\boldsymbol{\omega_{max}}$ = 2.17 cycles / milli-radian using the definitions of chapter 2.

MTF Measurement

For MTF measurement, we use the same ImageJ slant edge method as used in section B.1 for the Phantom 3 drone. The measurement is shown in Figure B-10. One difference between the Samsung camera and the image output from the drones is that the Samsung only provides images in a compressed JPG format whereas the drones provide raw, uncompressed images.

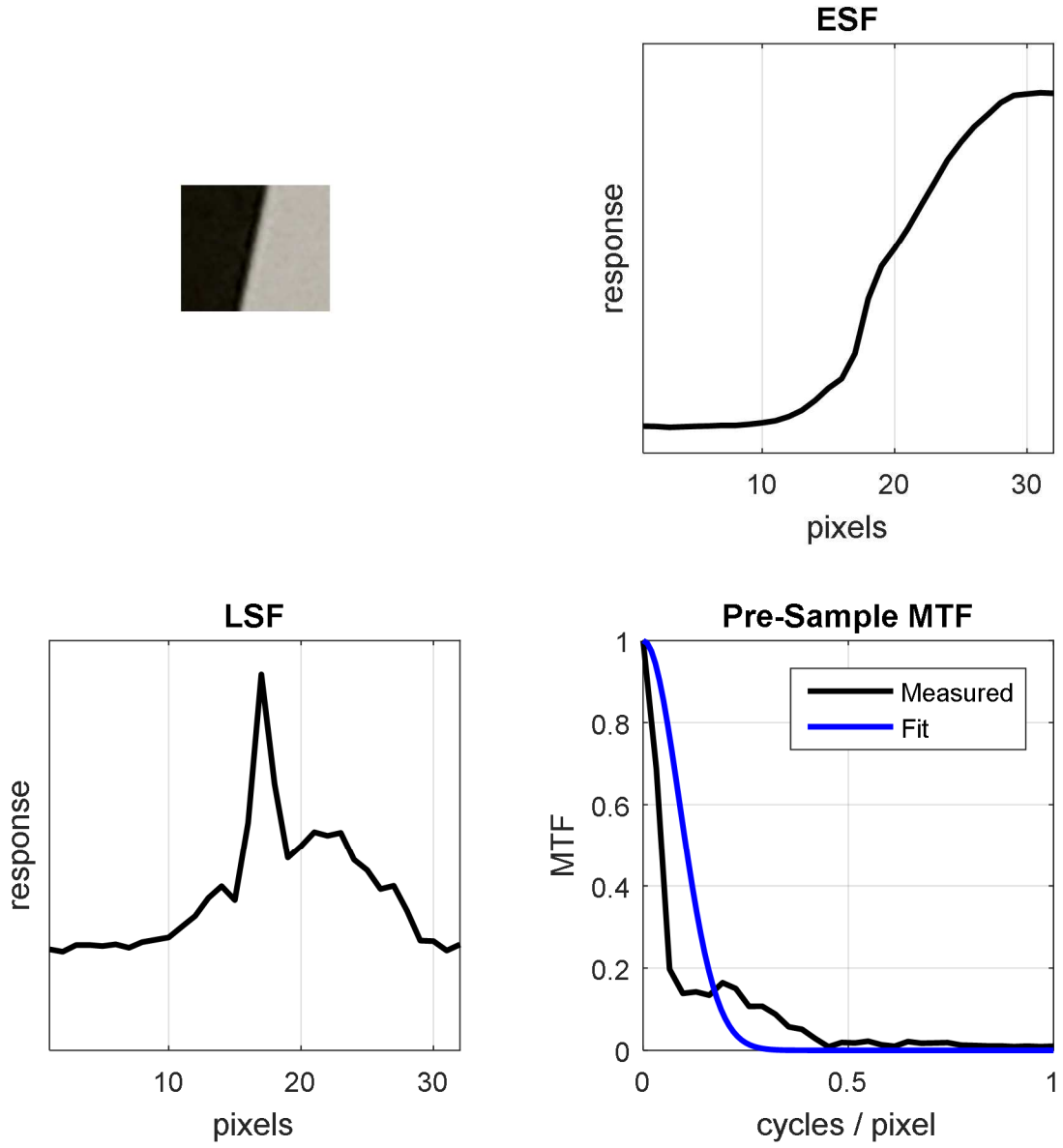


Figure B-10: Slant edge MTF measurement for the Samsung Galaxy 5 camera

The image is in the upper-left of Figure B-10, the ESF is in the upper-right, the computed LSF is in the lower-left, and the computed MTF is in the lower-right. Again, the imageJ algorithm computed the "pre-sample" MTF meaning that it doesn't include the frequency response associated with pixel integration. The Samsung MTF shows most clearly the trend to have a longer

tail (greater response at higher frequencies) than a Gaussian model. For the Samsung camera, we instead fit a slightly revised model to the measured pre-sample MTF, $\boldsymbol{H(\omega)}$, (shown as an overlay in the bottom-right of Figure B-10. We get a good fit to the measurements with $\boldsymbol{H(\omega)}$ taking the functional form

$$\boldsymbol{H(\omega) = exp(2\pi^2 q^2 \omega^2 - 2\pi r \omega),} \tag{B.2.1}$$

with $\boldsymbol{q}$ = 0.5762 milli-radians and $\boldsymbol{r}$ = 1.7382 milli-radians.

B.3 Characterization of the Phantom 2 FLIR Vue^{Pro} LWIR Camera

Being thermal in nature, calibration of infrared cameras (particularly in the long wave) using printed test targets, as is performed with the visible band cameras, is challenging due to the fact that it is difficult to create a detailed pattern with contrast. Contrast in the LWIR can be achieved either through temperature or emissivity differences of the materials. Printed ink on paper, such as used to form the targets of Figure B-1 and Figure B-2, presents an isothermal image with little emissivity differences. Detailed targets for IR camera calibration require expensive, specialized materials and equipment. These differences require us to take a slightly modified approach to calibration.

Geometric Projection

Compared to our visible cameras, the FLIR Vue^{Pro}, with 19 mm focal length, has a relatively small FOV (32° x 26°). Therefore, distortion is less significant and, due to the challenges mentioned above in constructing test targets, we not proceed with the assumption of a constant IFOV model. Based on the FOV dimensions and the 640 x 512 pixel format, we calculate the IFOV as 0.87 milli-radians. This produces $\boldsymbol{\omega_s}$ = 1.15 cycles / milli-radian and $\boldsymbol{\omega_{max}}$ = 0.57 cycles

/ milli-radian using the definitions of chapter 2. Note, as discussed in 2.4, we expect, in general, the sampling frequency of an IR camera to be less than that of a visible band camera.

MTF Measurement

We proceed to use the ImageJ slant edge method for MTF measurement. In order to produce a slant edge image to the camera, it is necessary to produce either a sharp difference in emissivity or temperature. Many common materials have an emissivity near 1.0 in the LWIR. Unfortunately, those that have a lower emissivity also tend to have a higher reflectivity which means they present an image of the background. Without careful controls, gradients associated with the background reflection will be falsely measured as MTF contributions. Generating temperature differences is also a challenge due to heat flow resulting between two contacting materials at different temperature. The temperature gradient due to heat flow will also manifest itself as a false MTF reading. Our solution was to use the temperature method. We solved the thermal gradient problem by placing a uniform hot object in the background and then inserting a sharp edge partition at a sufficient distance in front of it such as to eliminate heat transfer. Results are shown in Figure B-11.

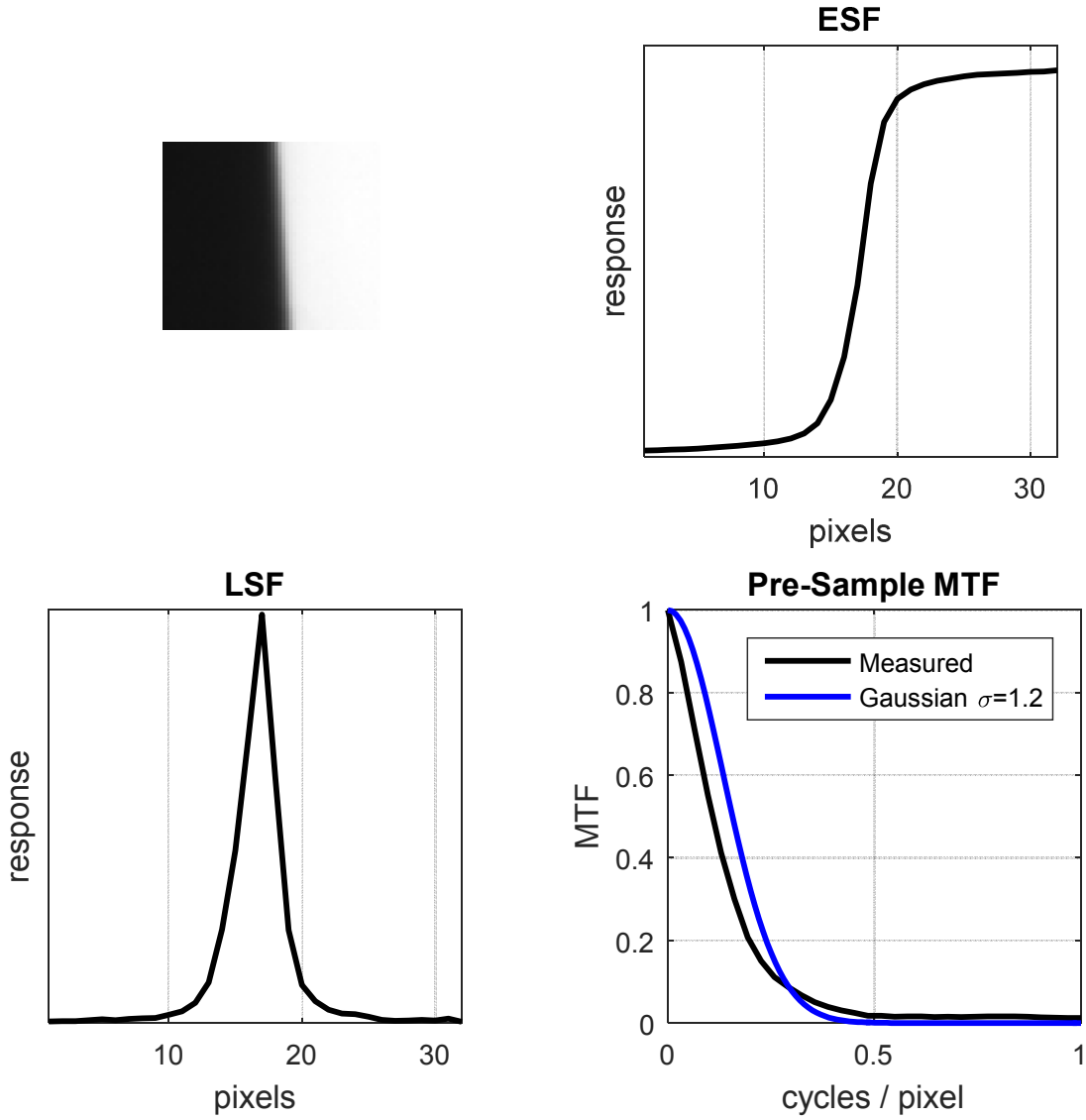


Figure B-11: Slant edge MTF measurement for the Phantom 2 FLIR Vue^{Pro} LWIR camera

The pre-sample MTF of the FLIR Vue^{Pro} LWIR camera is well represented by a Gaussian model with $\boldsymbol{\sigma}$ = 1.2 pixels. This is about twice the estimated diffraction limit (see 2.1) of $\boldsymbol{\sigma = 0.42\lambda/D}$ = 0.5 pixels.

VITA

Education

- Bachelor of Science in Mechanical Engineering, Purdue University, received 1997
- Masters of Science in Mechanical Engineering, Stanford University, received 1998
- Doctor of Philosophy in Electrical Engineering and Computer Science, Northwestern University, expected 2016

Professional Experience

- Systems Engineering, Passive Electro-Optical/Infrared, Northrop Grumman Systems Corporation, Rolling Meadows, Illinois, 2004 – present
- Systems Engineering, Guidance, Navigation, and Control, Raytheon Missile Systems, Tucson, Arizona, 1998 – 2004

Publications (Professional and Academic)

1. M. Woods and A. Katsaggelos, "Efficient Image Correspondence Measurements In Airborne Applications Using Inertial Navigation Sensors", in *18th European Signal Processing Conference (EUSIPCO)* (Academic, 2010), pp.606-610.

2. M. Woods and A. Katsaggelos, "Efficient method for the determination of image correspondence in airborne applications using inertial sensors," J. Opt. Soc. Am. A **30**, 102-111 (2013).

3. G. Mirsky, M. Woods, and J. Grasso, "Detection of Dim Targets in Multiple Environments," *Proc. SPIE* **8898**, (2013).

4. M. Woods and A. Katsaggelos, "Spatial-frequency-based metric for image superresolution," J. Opt. Soc. Am. A **32**, (2015) 2002-2020.

5. M. Woods and A. Katsaggelos, "Remote Classification from an Airborne Camera Using Image Super-Resolution," submitted to JOSA-A August 2016.

6. M. Woods and A. Katsaggelos, "Extending super-resolution to the airborne domain," to be submitted to JOSA-A.

Broad Agency Announcement (BAA) Proposals

1. M. Woods and A. Katsaggelos, "Super-Resolution Enhancement of Small-Format EO-IR Sensor Imagery", submitted in response to Office of Naval Research (ONR) BAA 12-005 (2012).

2. M. Woods and A. Katsaggelos, "Super-Resolution for Enhanced IR Target Recognition," submitted in response to Air Force FY 2013 Rapid Innovation Fund BAA-AFLCMC-2013-0001 (2013).