SUBPIXEL MULTIFRAME REGISTRATION FOR FORMATION FLYING SPACECRAFT

BY

EVAN WIDLOSKI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Farzad Kamalabadi

ABSTRACT

This thesis develops a multiframe image registration algorithm to accurately estimate the motion parameters of a sequence of images with constant linear translation between frames. The algorithm is developed with formation flying applications in mind, where such motion is common. The algorithm is non-iterative (obtaining a registration estimate in a fixed amount of time), efficient (requiring only FFT and spatial scaling operations), and parameterless (requiring no tuning for image classes). Additionally, the algorithm performs well under extreme levels of noise, obtaining an accurate motion estimate even when individual frames are so severely degraded that high-frequency structures are no longer visible in the individual frames. The algorithm is tested against synthetic noisy frames which have been generated by a computational pipeline designed to simulate observations that will be made by the *VIrtual Super-resolution Optics with Reconfigurable Swarms* (VISORS) Cubesat mission, set to be launched in 2023.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would like to extend my thanks to Ulas Kamaci, for his collaboration on the simulation pipeline implementation as well as reviewing the algorithm derivation section to make sure it is sound. I am also grateful for the support of my friends and family over the past few years of my degree.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

ADC	analog-to-digital converter
AIA	Atmospheric Imaging Assembly
ASE	absolute sum of errors
CMOS	complementary metal-oxide semiconductor
CPSD	cross power spectral density
DSC	detector spacecraft
EBTEL	Enthalpy-based Thermal Evolution of Loops
EUV	extreme ultraviolet
FFT	fast Fourier transform
FOV	field of view
IDFT	inverse discrete Fourier transform
IFFT	inverse fast Fourier transform
NCC	normalized cross correlation
OSC	optics spacecraft
PSF	point spread function
RANSAC	RAndom SAmple Consensus
SDO	Solar Dynamics Observatory
SSDA	Selective Similarity Detection Algorithm
VISORS	VIrtual Super-resolution Optics with Reconfigurable Swarms
CS_k	correlation sum

i_k	kth observation image in a set of K images
I	input scene in numerical experiments
μ	noiseless input scene in Section 3.2 derivations
$\boldsymbol{x}$	length 2 coordinate vector
$\boldsymbol{c}$	length 2 interframe drift vector
$\boldsymbol{\omega}$	length 2 spatial frequency vector
f	coordinate transform
R_θ	rotation matrix by angle θ
S_s	scaling matrix by factor s
T	VISORS frame integration time
r_{aia}	AIA angular resolution (mas)
r_{visors}	VISORS angular resolution (mas)
N_{visors}	VISORS detector width (pixels)
$(\cdot \star \cdot)$	circular cross correlation
$(\cdot \star_p \cdot)$	phase correlation
K	number of observation frames
a_{max}	photon arrival rate of brightest CMOS pixel

CHAPTER 1

INTRODUCTION

Image registration is the process of transforming multiple snapshots so that subjects or features common to two or more snapshots are aligned. The images may be stitched into a composite image to get a wider field of view, higher resolution, or reduced noise, or may be simply be aligned as in the case of video stabilization. Depending on the problem, registration algorithms often need to contend with changes in the scene being imaged (due to elapsed time between snapshots), perspective changes (changes in camera position), and illumination changes (from different imaging equipment) [1].

Motion estimation, a related field, is the process of identifying motion captured in a series of images (usually frames of a video). This motion may be due to motion of the camera which causes the whole scene to appear to move (*apparent motion*), or individual objects moving independently within the frame. In motion fields, a velocity vector is associated with each pixel in a particular region of the image (*local* motion estimation) or the image as a whole (*global* motion estimation). These motion vectors usually represent 2D motion across the image, but may also be 3D to capture movement in 3D space. When a motion field for individual pixels has been computed it is common to group motion vectors that belong to the same moving object, a process known as *motion segmentation* [2].

Registration is an important step in the image processing pipeline for countless fields. For example, in remote sensing applications registration is used in change detection, image mosaicing, and super-resolution. In medical imaging applications, registration is used for overlaying patient images from multiple channels, such as CT and MRI, which the caregiver can cross-reference for diagnoses, and to compare patient data to physiological atlases.

The next sections mathematically describe the problem of image registration, provide a categorization framework for registration methods, and explain the motivating problem of this thesis.

1.1 Registration Problem Model

Let i_1 and i_2 be two images captured of a scene. These are often called the *reference* and *sensed* images. In image registration, we want to find a mapping from regions in the sensed image to regions in the reference image. More formally, we want to find f such that

$$i_2[\boldsymbol{x}] = g(i_1[f(\boldsymbol{x})], \boldsymbol{x}) \ \forall \boldsymbol{x} \in X$$

where $\boldsymbol{x}$ is a coordinate vector in the discrete grid of coordinates X being registered, f is some unknown coordinate transform, and g is an unknown intensity mapping function. Function f can take a number of forms depending on the type of misalignment between i_1 and i_2 and is usually determined by a small number of registration parameters, which will be explored later in this chapter. When $f(\boldsymbol{x})$ maps to a point not on grid X, an interpolation operation is implied by $i_1[f(\boldsymbol{x})]$. Function g is often assumed to be unitary, but can be a very complicated function in multimodal applications like medical imaging where i_1 and i_2 are captured from different instruments. This thesis assumes i_1 and i_2 are 2D vectors containing image data.

1.2 Categorizing Registration Methods

While there is a wide variety of approaches to the problem of image registration, many algorithms can be broken down into four steps which aid in their classification [3].

1. *Feature detection* - Distinct features (points, edges, closed regions, intersections, corners, etc.) are detected in both images. These features may be represented by coordinates (intersections, corners, etc.), coordinate pairs (edges) or a more complex parameterization. This step is omitted in non-feature-based registration methods.

2. *Feature Matching* - Correspondence is established between features detected in the images. Feature similarity measures or feature positions within the images may be used to do this. This step is omitted in non-feature-based registration methods.

3. *Transform model estimation* - In feature-based methods, the parameters of the coordinate mapping function f are computed using the previously matched features. In non-feature-based methods, model parameters can be estimated from image statistics, iterative cost minimization, or image spectra, to name a few. This step is where most variability between registration methods lies.

4. *Image transformation* - The sensed image is transformed using the estimated parameters and optionally fused with the reference image. Interpolation may be necessary if the mapping function contains non-integer coordinates.

   The registration algorithms reviewed later in this thesis perform a single pass of these steps to arrive at the registered result, but some other algorithms, especially those used in the process of super-resolution, repeat steps through several iterations and only stop when some criterion is met [4].

1.3 A Comment on Notation

This document contains many types of variables which can represent transform parameters, 1D vectors of parameters and 2D images. This thesis follows these notational conventions for easier reading:

- Bold for variables which represent 1D vectors. For example $\boldsymbol{x}$ is a coordinate vector representing position within an image.
- Superscript $*$ for ground truth parameters of coordinate transform f. For example s^* and θ^* are parameters controlling scaling and rotation.
- Hat $\hat{}$ for algorithmic estimates of ground truth parameters. For example $\hat{\theta}$ represents the estimates for θ^* found by a particular algorithm.
- Parenthesis ( ) when indexing 1D functions or 2D surfaces defined over rational numbers. For example $f(\boldsymbol{x})$ is a coordinate transform function.
- Brackets [ ] when indexing 1D functions or 2D surfaces defined over integers. This may imply an interpolation if the argument is non-integer. For example $i[\boldsymbol{x}]$ is a pixel of a discrete 2D image.

1.4 Types of Image Transforms

The coordinate transform is a fundamental component of any registration algorithm. Most registration algorithms describe a specific class of coordinate transforms which can be completely described by a handful of parameters that are searched over during the *transform model estimation* step. This section describes a few of the most common classes of coordinate transforms and their parameters, and gives some examples of where they are used.

1.4.1 Similarity and Affine

The simplest and most common type of coordinate transform is translation

$$f(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{c}$$

where $\boldsymbol{c}$ is a length 2 vector whose elements correspond to the shift in each dimension. Some of the oldest registration methods operate over this class of transforms.

Another type of registration method is rotation, in which the sensed image is rotated about some point.

$$f(\boldsymbol{x}) = R_\theta \boldsymbol{x}$$

where R_θ is known as a *rotation matrix*. R_θ has orthogonal columns and can be entirely parameterized by θ, the rotation angle.

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

These two transform classes might be used together when stitching images from a digital microscope to get a larger field of view where the specimen slide is allowed to translate or rotate in a fixed plane.

A third type of coordinate transform is scaling, where the sensed image origin and orientation remain fixed, but coordinates are scaled.

$$f(\boldsymbol{x}) = S_s \boldsymbol{x}$$

where S_s is a scaling matrix parameterized by the scaling factor s.

$$S_s = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$$

These three coordinate transforms taken together are often called a *similarity transform*. Similarity transforms are rigid, meaning they do not change the shape of features in the reference image, parallel lines remain parallel, and angles and lengths are preserved. For example, a triangle in the sensed image will map to a similar triangle in the reference image.

Similarity transforms can be written generally as

$$f(\boldsymbol{x}) = R_\theta S_s \boldsymbol{x} - \boldsymbol{c}$$

Some authors allow for the first or second column of R_θ to be negated which corresponds to a geometric reflection, though this is less common in registration settings.

A generalization of the similarity transform is the *affine transform*, where the rotation matrix is replaced by an arbitrary matrix and the scaling factor is incorporated into it.

$$f(\boldsymbol{x}) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \boldsymbol{x} - \boldsymbol{c}$$

This transform can also account geometric skew, where angles and lengths are no longer preserved but parallel lines remain parallel.

1.4.2 Perspective

Another coordinate transform is the *perspective transform*, which occurs when a 3D scene is viewed through an idealized optical system and projected onto a 2D plane. Perspective distortion causes objects which are farther away from the camera lens to appear smaller in a process known as *foreshortening* [1]. If the coordinates of a visible point in 3D space are known, for example $[x', y', z']$, then its location within the image can be computed as

$$\boldsymbol{x} = [x, y]^T = \left[\frac{-dx'}{z' - d}, \frac{-dy'}{z' - d} \right]^T$$

where d is the distance of the camera lens from the image plane. This is

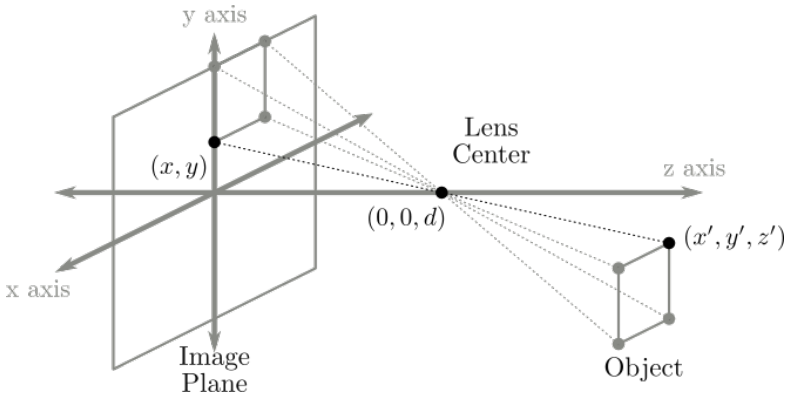

Figure 1.1: Perspective projection model.

illustrated in Figure 1.1.

In the context of image registration, we would like to know how changes in the relative position and orientation between the camera and observed scene affect projected coordinates. However, it is not possible to express the coordinate transform between i_1 and i_2 as a function of $\boldsymbol{x}$, as the transform depends on the physical location of features in 3D space which is generally unknown. However, assumptions can sometimes be made which simplify the transform significantly.

In many remote sensing applications, the distance to the scene is much larger than the distance to the lens center ($z' \gg d$) and the spacecraft's field of view is narrow. In this setting, illustrated in Figure 1.2, it is possible to show that small changes in attitude manifest as apparent translational motion of the scene.

In other words

$$f(\boldsymbol{x}) = R_\theta \left(\boldsymbol{x} - [d\tan(\alpha),\, d\tan(\phi)]^T \right)$$

where θ, ϕ and α are the spacecraft roll, pitch and azimuth.

This reduces the registration problem in many space science applications from a coordinate transform model of perspective projection, to a simple rigid model with translation and rotation in the image plane.

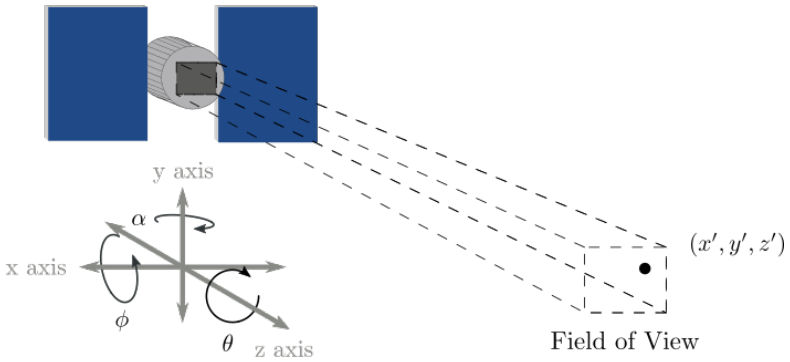

Figure 1.2: Distance field of view of spacecraft, along with 6 degrees of freedom.

1.4.3 Elastic

When the type of transform is unknown or more complicated, elastic transforms can be used to correct for misalignment. This can arise in situations where a 3D scene is projected through an optical system onto a 2D plane, but unlike perspective projection, there are large depth variations which occlude parts of the scene. Objects which appear in one image may be completely obscured in the other, which is increasingly difficult to account for as the number of occlusions increases. In these settings, a more general coordinate transform which can map more arbitrary distortions is preferred.

Elastic methods can either operate over the image as a whole (global), or apply different transforms to regions of the image separately (local). An example of a global elastic method is the *bivariate polynomial* transform, which feeds the x and y coordinates through a pair of polynomial functions.

$$u = \sum_{l=0}^{m} \sum_{j=0}^{i} a_{ij} x^l y^{j-l}$$

$$v = \sum_{l=0}^{m} \sum_{j=0}^{i} b_{ij} x^l y^{j-l}$$

where $[x, y]$ are the original coordinates, $[u, v]$ the transformed coordinates, and a_{ij} and b_{ij} are constant parameters controlling the transform.

Local methods are more general than global methods and can handle distortions that global methods cannot, such as deformable objects, complex 3D surfaces, and object motion within the scene. While these methods are more powerful, there is a tradeoff with computational complexity as the number

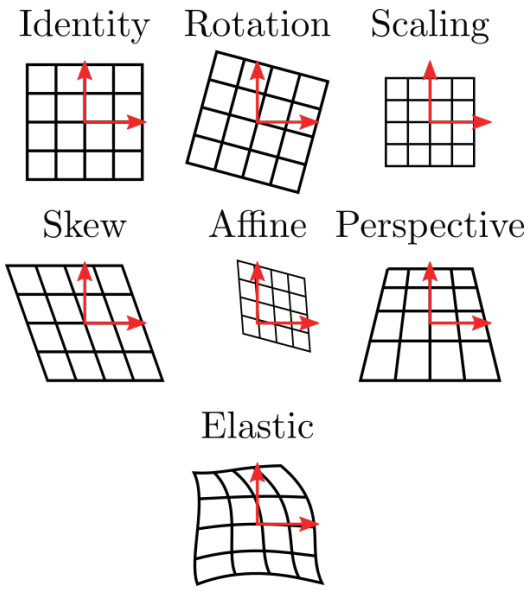

Figure 1.3: Visualizations of several coordinate transformations of a grid with origin prior to transformation shown with red arrows.

of parameters increases. An example of a local elastic coordinate transform is piecewise spline interpolation.

Figure 1.3 is a diagram which illustrates examples of some of these transform classes.

1.5 VISORS Mission

The VIrtual Super-resolution Optics with Reconfigurable Swarms (VISORS) mission, due to be launched by NASA in 2023, is a heliophysics CubeSat mission designed to study the Sun's atmosphere (corona) at a finer scale than has been achieved in previous missions. Its primary mission is to reveal the heating process which causes the corona to be over 1000 times hotter than its surface (photosphere), which has been a central problem in heliophysics for decades.

Observations made in x-ray and extreme ultraviolet (EUV) have provided hints on this heating process, but the structure and precise mechanism of the heating have yet to be observed directly. A leading hypothesis is that the heating is constrained to narrow sheets which transfer energy from the Sun's surface into the corona. The expected scale of these sheets is on the order of 100km [5], or about 150 milli-arcseconds (mas) when viewed from

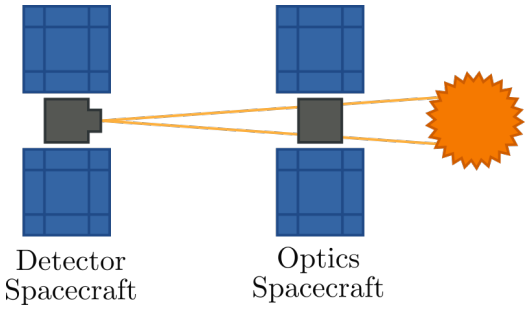

Figure 1.4: Pictorial representation of VISORS in formation flying configuration.

Earth's orbit, but no existing coronal imagers have been able to reach this resolution.

VISORS consists of two 3U spacecraft known as the optics spacecraft (OSC) and detector spacecraft (DSC), which carry instrumentation for taking measurements in extreme ultraviolet (EUV) range. These two spacecraft will fly in formation 40 meters apart aligned along an axis pointed at the region of interest on the Sun during science mode, as in Figure 1.4. The OSC focuses incoming light using a novel diffractive element known as a photon sieve while simultaneously using its solar panels to block off-axis light from entering the DSC. The DSC will be positioned on the focal plane corresponding to He II emission line at 30.4 nm. In particular, the OSC uses a diffractive optical element known as a photon sieve, which can outperform equivalent reflective optics due to tighter manufacturing tolerances [6].

VISORS is what is known as a *virtual* telescope. In contrast to other non-virtual space telescopes such as Hubble (visible light) and the Solar Dynamic Observatory (EUV), the focusing optics and detector fly on separate spacecraft which allows the design to support large focal lengths without significantly increasing spacecraft volume and to reconfigure the focused wavelength after launch by adjusting spacecraft separation.

In addition to its contributions to heliophysics, VISORS will serve as a technology demonstration of diffractive, distributed telescopy and precision satellite formation flying.

1.6 Thesis Outline

Chapter 2 introduces classes of image registration and describes popular registration methods from each class, and details the method of subpixel registration. Chapter 3 introduces the idea of multiframe registration and relates it mathematically to the two-frame problem. This chapter also presents a new multiframe registration method and explains how it is approximately optimal in an ML sense. Chapter 4 contains numerical registration experiments under various settings, a description of the pipeline used to generate the test images, and some tests involving other classes of images unrelated to the VISORS project.

CHAPTER 2

PRELIMINARIES

2.1 Review of Global Registration Methods

Most registration algorithms can be assigned to two broad categories known as area-based and feature-based methods. These registration classes vary primarily in the steps leading up to transform estimation, while the image transformation step is generally unchanged for a given motion model. In feature-based methods, localized structures such as lines, regions or points, known as *features*, are detected and related in the reference and sensed images during the feature detection and feature estimation steps. Area-based methods omit these steps entirely.

This section focuses primarily on the first three steps before performing the coordinate transform, namely feature detection, feature matching, and transform model estimation.

2.1.1 Correlation-Based Methods

Cross-Correlation

The most straightforward of all methods, direct correlation, is conceptually simple and works for many classes of transforms.

Note that the normalization here is crucial so that the intensities of i_1 and i_2 do not influence the maximum. We call this measure normalized cross-correlation (NCC) [1].

$$\begin{aligned}\hat{f} &= \arg\max_{f\in F} \mathrm{NCC}(i_1, f(i_2)) \\ &= \arg\max_{f\in F} \frac{\sum_{\boldsymbol{x}\in X} i_1(\boldsymbol{x}) i_2(f(\boldsymbol{x}))}{\sqrt{\sum_{\boldsymbol{x}\in X} i_1(\boldsymbol{x})^2}}\end{aligned}$$

where F is some class of coordinate transforms. The term $f(i_2)$ is shorthand for a transformed imaged such that $f(i_2)(\boldsymbol{x}) = i_2(f(\boldsymbol{x}))$. Practically, the time required to search the space of all possible transforms for a given application makes this approach infeasible, so F is often restricted to translations for this method.

$$\hat{c} = \arg\max_{\boldsymbol{c}} \frac{\sum_{\boldsymbol{x}\in X} i_1(\boldsymbol{x}) i_2(\boldsymbol{x}-\boldsymbol{c})}{\sqrt{\sum_{\boldsymbol{x}\in X} i_1(\boldsymbol{x})^2}}$$

Related similarity measures which are sometimes used in place of normalized cross-correlation are sum of squared error (SSE)

$$\mathrm{SSE}(i_1, f(i_2)) = \sum_{\boldsymbol{x}\in X} (i_1(\boldsymbol{x}) - i_2(f(\boldsymbol{x})))^2$$

and correlation coefficient

$$\mathrm{Corr}(i_1, f(i_2)) = \frac{\mathrm{cov}(i_1, f(i_2))}{\sigma_1 \sigma_2} = \frac{\sum_{\boldsymbol{x}\in X} (i_1(\boldsymbol{x}) - \mu_1)(i_2(f(\boldsymbol{x})) - \mu_2)}{\sqrt{\sum_{\boldsymbol{x}\in X} (i_1(\boldsymbol{x}) - \mu_1)^2 \sum_{\boldsymbol{x}\in X} (i_2(f(\boldsymbol{x})) - \mu_2)^2}}$$

where μ_1, μ_2, σ_1, σ_2 are the means and variances of i_1 and i_2 over X.

While cross-correlation methods are very old, they continue to see widespread use because of ease of implementation in hardware (NCC can be efficiently implemented using multiply-accumulate hardware) and because limiting f to translations is not significantly restrictive for many scenarios.

In practice though, cross-correlation still succeeds on natural images in the presence of slight rotational, scalar or even non-linear distortions.

Selective Similarity Detection Algorithm

In standard correlation methods, the sum over X for each candidate f must be computed in full before a maximum is found. Barnea and Silverman [7] propose a class of alternative algorithms which greatly improve computation time in two ways. The paper calls these algorithms selective similarity detection algorithms (SSDAs), of which one is presented here.

First, the paper uses absolute sum of errors (ASE) as a similarity measure, which requires no costly multiplications unlike NCC or SSE.

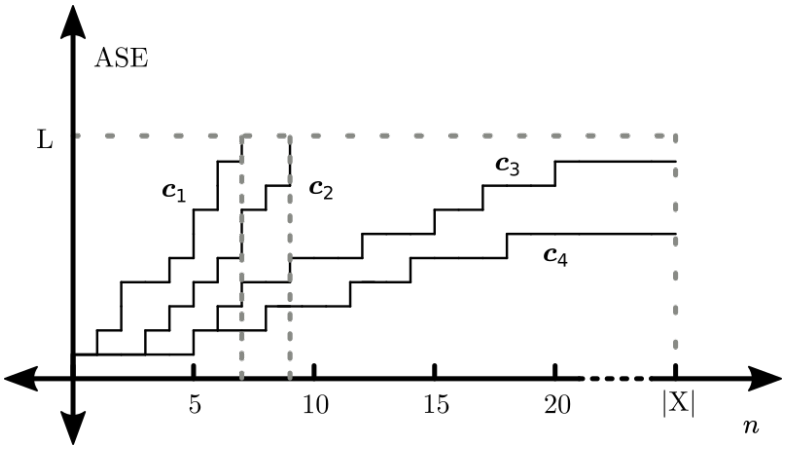

Figure 2.1: Error accumulation curves for candidates $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ and $\boldsymbol{c}_4$. Error computation for $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ terminated early at 7 and 9 iterations. $\boldsymbol{c}_4$ is the best estimate for offset, followed by $\boldsymbol{c}_3$, $\boldsymbol{c}_2$ and $\boldsymbol{c}_1$.

$$ASE(i_1, f(i_2)) = \sum_{\boldsymbol{x} \in X} |i_1(\boldsymbol{x}) - i_2(\boldsymbol{x} - \boldsymbol{c})|$$

The second optimization uses early stopping and requires that the sum over X be implemented sequentially (e.g. as an iterative software loop). For a particular candidate $\boldsymbol{c}$, the current value of the sum is compared to a threshold parameter L after each iteration. If the ASE surpasses this threshold, the number of iterations is recorded and the algorithm moves on to the next candidate, as illustrated in Figure 2.1. If an candidate computation never exceeds L, then the final ASE is recorded instead.

Finally, the candidate with the lowest ASE is selected. If all candidates surpassed the threshold, then the candidate with the most number of iterations before passing the threshold is selected.

SSDA requires selection of parameter L. A choice of L too high limits efficiency gains while a choice of L too low can lead to suboptimal results. However, this algorithm offers potentially orders of magnitude in speed improvements over correlation methods which makes it appropriate for time-sensitive or computationally under-powered applications.

2.1.2 Feature-Based Methods

Feature-based methods involve a preprocessing step known as *feature detection*, where notable structures in both images are located. These structures can come in a variety of forms and can correspond to different kinds of objects depending on the context. For example, images being registered in a remote sensing context could contain features such as amorphous regions (forests, lakes, fields), line segments (roads, buildings), or single points (street intersections, region corners). There are many algorithms capable of extracting these features and the optimal choice of algorithm is highly dependent on the target scene. In general, a desirable property of these algorithms is that the same features can be detected in both images and that these features are robust against corruption introduced by intensity mapping function g or noise [3]. In particular, the SIFT [8] and MSER [9] algorithms have gained widespread popularity in computer science as feature detection algorithms.

Some features in the sensed image may not have a matching counterpart in the reference image due to occlusions or because their counterpart is outside of the field of view. A desirable property of the feature matching algorithm is that incorrect matches are eliminated or assigned a low score so that parameter estimates in the later transform estimation step are not skewed [3].

RANSAC

RANSAC, or RAndom SAmple Consensus, is an iterative parameter estimation method which is robust to outliers [10]. In the context of image registration, RANSAC is sometimes used to estimate the parameters of a perspective transform, where the outliers are erroneously matched pairs of features.

RANSAC begins by selecting a random subset of matched feature pairs and computing a hypothetical set of transform parameters. In the case of perspective projection, this is simply solving a linear system. The quality of this estimated coordinate transform is evaluated by transforming the coordinates of all features in the reference image and aggregating the error for all pairs. This estimated transform is then stored along with its aggregate error, and the process is repeated for a new randomly selected subset of feature pairs until a predetermined number of transforms has been estimated. The

transform with the smallest aggregate error is then chosen as the algorithm's result.

RANSAC is unique among the methods described in this chapter in that it is non-deterministic and repeated trials on the same dataset can yield different results. Additionally, while RANSAC is robust to outliers, it is not especially immune to measurement noise which causes the number of incorrectly paired features to rise [10].

2.1.3 Information-Based Methods

Viola and Wells [11] introduced a new class of registration methods in 1994 based on entropy or information content of image pairs. This class of methods has proven effective in multimodal registration so it has achieved significant popularity in medical imaging.

The number of unique messages that can be encoded given a message length of n and s unique symbols is s^n. However, a desirable property of information is that information should grow linearly with message length. A message which is twice as long should contain twice as much information. Therefore, in the context of information theory, information is defined as

$$H = \log s^n = n \log s$$

From the first formulation, it is apparent that information grows linearly with n. Another interesting feature of this measure is that if there is only one symbol, we know exactly what the message will be and so it contains no information ($H = n \log 1 = 0$). This suggests that entropy can also be viewed as a measure of uncertainty.

A disadvantage of this definition of entropy is that it assumes all symbols are equally likely to occur in a message, which is generally not true in physical systems.

An alternative measure of information takes this into account by weighting the entropy by the probability that symbols occur. This is known as Shannon entropy. For a set of s symbols with probabilities $p_1, ..., p_s$ of occurring, Shannon entropy is given as

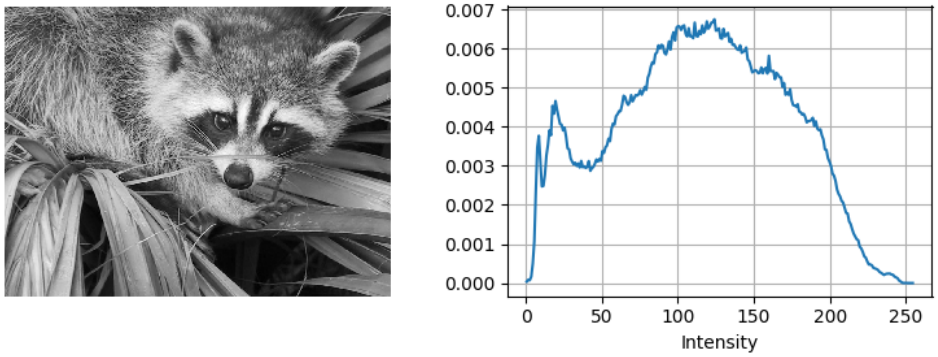

Figure 2.2: An image and its intensity histogram

$$H = \sum_k p_k \log \frac{1}{p_k} = -\sum_i p_k \log p_k$$

Like the unweighted entropy measure, Shannon entropy can be viewed as a measure of uncertainty. If a particular symbol has a very high probability of occurring, our uncertainty about the message decreases and hence information decreases. If all symbols have an equal probability of occurring, entropy is maximized. Thus Shannon entropy may also be considered as a measure of spread of a probability distribution [12]. A distribution with most mass concentrated around a few peaks will have low entropy while a more uniform distribution will have higher entropy.

To compute Shannon entropy of an image, all possible intensity values of the pixels can be interpreted as symbols in the message. For an image with bit depth of 8 bits, one can collect all the intensity values into a histogram in order to compute $p_0, ..., p_{255}$, as shown in Figure 2.2.

Now that we can compute entropy for images we must introduce one more concept before registration can occur, joint histograms. A joint histogram is a 2D function which, for all possible pairs of intensities, describes how many times intensity pairs occur for a pair of registered images. For example, if a joint histogram has value 17 at coordinate [33, 34], then for this particular registration there are 17 pixels in which the first image has intensity 33 and the second has intensity 34. In the case of two 8 bit images, the joint histogram is a 256x256 image. An example joint histogram for two images is shown in Figure 2.3.

The joint histogram changes with the alignment of the images. For a

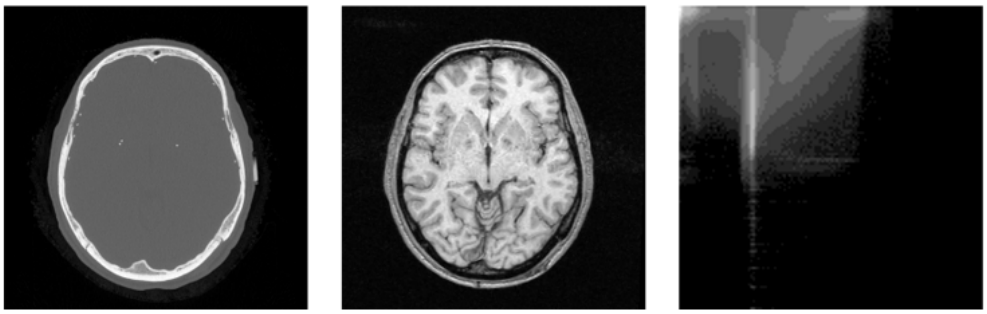

Figure 2.3: Two registered images and their joint histogram, or feature space.

correctly aligned pair of images, structures within the image align and vary with each other, so we expect the intensities to correlate which manifests as clustering in the joint histogram. As the image pair becomes misaligned, more greyscale combinations are introduced and the joint histogram exhibits more uniformity. By measuring this uniformity we now have a similarity measure for registration.

Formally, the joint Shannon entropy for a pair of registered images

$$H(i_1, f(i_2)) = -\sum_{m,n} p(m,n) \log p(m,n)$$

where $p(i,j)$ is the joint histogram of i_1 and candidate registered $f(i_2)$ in the region of overlap X.

However, a problem that can occur when joint entropy is used directly is that low entropy (high degree of reported alignment) can occur for invalid registrations if the images contain large regions of uniform intensity. For example, if the images in the figure above are aligned so that only their corners containing background overlap, the joint histogram will have approximately a single peak and the joint entropy will be very low. To account for this, one can make use of the marginal entropies to penalize alignments where the region X contains little information in the images. This is known as mutual information.

$$MI(i_1, f(i_2)) = H(i_1) + H(f(i_2)) - H(i_1, f(i_2))$$

With this new measure, if the overlap region contains little information,

terms $H(i_1)$ and $H(f(i_2))$ will be small and counteract joint entropy. Also note that since mutual information contains $-H(i_1, f(i_2))$, minimizing joint entropy is related to maximizing mutual information.

The ability of mutual information based methods to handle images with differing distributions of pixel intensities makes it uniquely suited for multimodal image registration applications. Especially in medical applications, mutual information based registration can be used to register images of the same organ made by different instruments (e.g. PET and CT).

2.1.4 Frequency-Based Methods

If an acceleration over correlation-based methods is needed or the images were acquired under frequency dependent noise, Fourier methods are often preferred. These methods exploit the Fourier representation of images in the frequency domain and have shown better robustness against illumination differences between i_1 and i_2.

Phase Correlation

Phase correlation was originally proposed for registering linearly translated images. It takes advantage of the Fourier shift theorem, which states that translating an image and taking its Fourier transform is equivalent to multiplying the Fourier transform of the original untranslated image by a complex exponential.

Computing the cross-power spectral density (CPSD) we can directly obtain this complex exponential.

$$i_2(\boldsymbol{x}) = i_1(\boldsymbol{x} - \boldsymbol{c}^*)$$

$$CPSD(i_1, i_2)(\boldsymbol{\omega}) = \frac{\tilde{i}_1(\boldsymbol{\omega})\overline{\tilde{i}_2(\boldsymbol{\omega})}}{|\tilde{i}_1(\boldsymbol{\omega})\overline{\tilde{i}_2(\boldsymbol{\omega})}|} = \frac{\tilde{i}_1(\boldsymbol{\omega})\overline{\tilde{i}_1(\boldsymbol{\omega})e^{-j\langle\boldsymbol{\omega},\boldsymbol{c}^*\rangle}}}{|\tilde{i}_1(\boldsymbol{\omega})\overline{\tilde{i}_1(\boldsymbol{\omega})e^{-j\langle\boldsymbol{\omega},\boldsymbol{c}^*\rangle}}|} = e^{j\langle\boldsymbol{\omega},\boldsymbol{c}^*\rangle}$$

where $\tilde{i}_1$ and $\tilde{i}_2$ are the Fourier transforms of i_1 and i_2. The final estimate for $\boldsymbol{c}_0$ is obtained by a final inverse Fourier transform of the CPSD, yielding a delta at location $\boldsymbol{c}_0$.

$$PC(i_1, i_2)(\boldsymbol{x}) = \mathcal{F}^{-1}\left[CPSD(i_1, i_2)\right](\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{c}^*)$$
$$\hat{c} = \arg\max_{\boldsymbol{x}} PC(i_1, i_2)(\boldsymbol{x})$$

An important consideration here is that the Fourier shift theorem only holds when translation is circular. In practice, the phase correlation method still works if the region of overlap is sufficiently large. Foroosh et al. [13] propose a prefilter which can be applied to both images before phase correlation to reduce these effects.

2.2 Subpixel Registration

We will now turn our attention to a special class of translational registration methods known as *subpixel* methods. Unlike non-subpixel translation methods which return a vector of integers representing the coordinate shift between images in each axis, subpixel methods assume the scene being observed exists at a higher resolution than the observations and seek to find a fractional value for this shift. In real imaging systems, the resolution ratio between the scene and observations is usually infinite as the scene is a continuous function. However, in practice this ratio is often assumed to be finite and the scene assumed to exist on a discrete high-resolution grid which captures the features of the scene being studied in sufficient detail.

This requires us to adjust the image registration model given in the introduction to include some notion of downsampling. First we define a pair of high-resolution images h_1 and h_2 which live on a high-resolution grid. The observed images i_1 and i_2 are downsampled versions of these high-resolutions images corresponding to the same field of view. The downsample factor between the high-resolution grid and low-resolution grid is denoted by positive integer d.

$$i_2[\boldsymbol{x}] = h_2[d\boldsymbol{x}]$$
$$i_1[\boldsymbol{x}] = h_1[d\boldsymbol{x}]$$
$$\boldsymbol{x} \in X$$

Now instead of aligning i_1 and i_2 like non-subpixel methods, subpixel registration seeks to align the high-resolution images without directly observing them:

$$h_2[\boldsymbol{x}_h] = h_1[\boldsymbol{x}_h - \boldsymbol{c}] \ \forall \boldsymbol{x}_h \in X_h$$

where $\boldsymbol{x}_h \in X_h$ is a high-resolution coordinate of the high-resolution set of points X_h.

This section introduces two subpixel registration methods and highlights advantages and disadvantages of each.

2.2.1 Interpolation and Registration

A straightforward way to achieve subpixel registration is to make use of interpolation techniques to boost the accuracy of the misalignment estimate.

Some methods involve interpolating the observed images before applying a standard non-subpixel registration technique such as cross-correlation to the upsampled images [14].

Other methods make use of the correlation surface obtained from the low-resolution images, then upsample or fit a continuous function to the surface before searching for a maximum. Many types of interpolation have been employed, such as nearest neighbor, bilinear, spline and sinc interpolation [15]. A classic technique involves zero-padding the correlation result from one of the Fourier-based methods described previously before inversion, which effectively applies sinc interpolation while benefitting from the efficiency of FFT algorithms.

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{x}_h \in X_h} \text{IFFT}\left(\text{Zeropad}_d\left(\text{FFT}(i_1) \cdot \overline{\text{FFT}(i_2)}\right)\right)[\boldsymbol{x}_h]$$

However, a major downside to these subpixel methods is that computational cost often scales rapidly with image size and downsample factor d. For example, using the zero-padded IFFT technique on a square image 1024 pixels wide requires computation and storage of an IFFT of size 102400 on either side, not feasible on modern workstations [16].

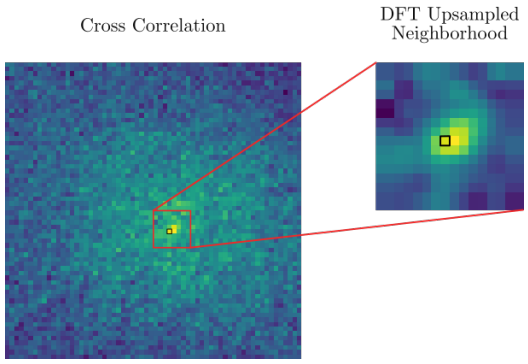

Figure 2.4: two-steps of Guizar-Sicairos's method. A coarse peak location is found in the original correlation surface, then a small neighborhood around this peak is upsampled via IDFT to get the fine estimate.

2.2.2 Accelerated Interpolation via IDFT

To get around the computational challenges associated with the zero-padded IFFT method, Guizar-Sicairos et al. [16] developed a mathematically equivalent algorithm which is faster and much more resource efficient. It uses a two-step coarse-to-fine approach to first find the approximate peak location on the original low-resolution grid, then applies an upsampled IDFT to a region of pixels around this peak to obtain a fine registration estimate.

$$\hat{\boldsymbol{c}}_{\text{coarse}} = \arg\max_{\boldsymbol{x}\in X} \text{IFFT}\left(\text{FFT}\left(i_1\right)\cdot\overline{\text{FFT}\left(i_2\right)}\right)[\boldsymbol{x}]$$

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{x}_h\in \text{N}(\hat{\boldsymbol{c}}_{\text{coarse}})} \text{UpsampIDFT}\left(\text{FFT}\left(i_1\right)\cdot\overline{\text{FFT}\left(i_2\right)}\right)[\boldsymbol{x}_h]$$

where $N(\hat{\boldsymbol{c}}_{\text{coarse}})$ is a neighborhood of pixels around $\hat{\boldsymbol{c}}_{\text{coarse}}$ and UpsampIDFT is an inverse DFT operation with an output size larger than its input size.

This two-step process is illustrated in Figure 2.4.

CHAPTER 3

MULTIFRAME REGISTRATION

Multiframe registration is a generalization of image registration from a pair of images to a set of images. This image set may be ordered, as in the case of a sequence of frames, or unordered. The idea remains the same as classical registration in that the goal is to align template images in the sequence with a reference image.

There are many applications of multiframe registration, such as chromatic aberration correction [17], motion stabilization in video [18], super-resolution [4], noise reduction and image compositing [19].

This chapter introduces a mathematical model for the problem of multiframe registration, describes a multiframe registration algorithm developed for the constant translational motion model in the VISORS project, then shows how this method is derived from the maximum likelihood solution to multiframe registration.

3.1 Multiframe Registration Problem Model

Let i_j and i_k be two images from a set of K images. In the classical registration problem, we would like to relate these two images such that

$$i_k[\boldsymbol{x}] = i_j[f_{j,k}(\boldsymbol{x})] \ \forall \boldsymbol{x} \in X$$

by finding the coordinate transform $f_{j,k}$ from the jth frame to the kth frame. In the multiframe setting, not only do we wish to find $f_{j,k}$, but also some or all of the $K(K-1)$ coordinate transforms which relate any two images in the set.

Multiframe registration methods take varying approaches when choosing template and reference images. Methods taking the *anchoring* approach choose a single anchor frame from the set of images and register all other

images against this frame. These methods are sometimes used when motion between frames is random, such as the case of random vibrations affecting a camera system.

An alternative is the *progressive* approach, where pairs of adjacent frames in an ordered image sequence are taken to be the reference and sensed images. This is often used in situations where there is some prior knowledge on the motion between adjacent image pairs. For example, in a handheld camera one would expect motion between adjacent frames to change slowly over time, as the movements induced by hand tremors are relative slow compared to the shutter frame rate [18].

In both cases, it is usually desirable that the found coordinate transforms exhibit properties of consistency known as the *Jacobi identity* and *skew anti-symmetry* properties [20].

The Jacobi identity property states that the application of the coordinate transform from frame j to frame k, then the transform from k to l, should be the same as the transform from j to l.

$$f_{j,k}(f_{k,l}(\boldsymbol{x})) = f_{j,l}(\boldsymbol{x}) \tag{3.1}$$

for $j, k, l \in 1, ..., K$.

In a similar vein, the skew anti-symmetry property states that applying coordinate transform $f_{k,j}$ immediately after $f_{j,k}$ should yield the original coordinate.

$$f_{k,j}(f_{j,k}(\boldsymbol{x})) = \boldsymbol{x} \tag{3.2}$$

for $j, k \in 1, ..., K$.

These properties (illustrated in Figure 3.1) serve to constrain the set of found coordinate transforms to a solution which is physically realizable [21].

We may impose additional constraints on the coordinate transform solution set if there is extra prior knowledge about the class of transforms. In the VISORS project, we assume that motion is purely translational and constant between frames. which translates mathematically to

$$f_{j,k}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{c}(k - j) \tag{3.3}$$

$$f_{j,k}(\boldsymbol{x}) = f_{l,m}(\boldsymbol{x}) \tag{3.4}$$

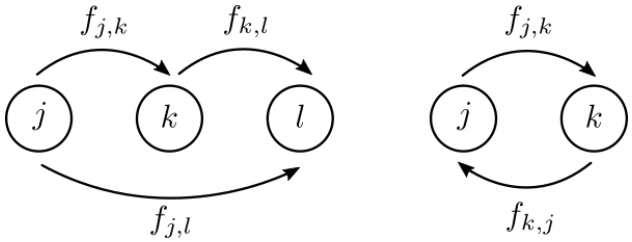

Figure 3.1: Jacobi identity and skew anti-symmetry properties.

for $j, k, l, m \in 1, ..., K$ when $k - j = m - l$. $\boldsymbol{c}$ is the interframe drift vector in pixels.

Plugging these constraints into the left-hand side of Equations 3.1 and 3.2, it is easy to see the Jacobi and skew anti-symmetry properties are satisfied.

3.2 Multiframe Registration with All Frames

A naive extension of one of the algorithms described in Section 2.1 appropriate for constant translational motion might be repeated application of the algorithm to adjacent frames in a progressive fashion, then an averaging of the estimated motion parameters to get a single estimate. For example, one might extend the algorithm given by Guizar-Sicairos for subpixel registration to multiple frames like so:

$$\hat{\boldsymbol{c}} = \frac{\sum_{k=1}^{K-1} \text{Register}(i_k, i_{k+1})}{K-1}$$

where Register$(\cdot, \cdot)$ represents one of the algorithms given in Section 2.1.

However, since only two frames are considered at a time, in low SNR settings Register$(\cdot, \cdot)$ may return an estimate which is wildly inaccurate. As the mean is sensitive to outliers, this can seriously degrade the accuracy of the estimate. Instead, one can use methods which jointly consider all adjacent image pairs to suppress noise before a motion estimate is made.

The method proposed in this section works by fusing correlation surfaces from many image pairs into a single surface before finding the maximum to make an estimate. The method not only uses correlation surfaces from adjacent image pairs, but also all pairs separated by two or more frames for greater noise immunity. One might believe that these higher order correlation surfaces contain redundant information which is already present in the

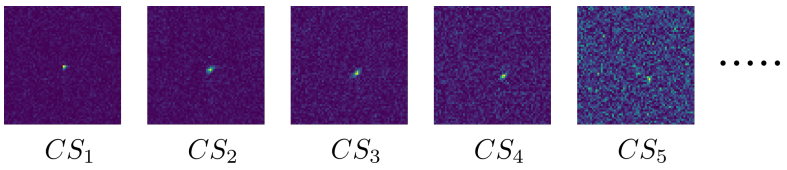

Figure 3.2: Sample correlation sum surfaces.

surfaces for adjacent images, but the analysis in Section 3.3 suggests this is the right approach.

First define *correlation sum* (denoted CS_m) as the sum of all phase correlation surfaces of image pairs separated by m.

$$CS_m[\boldsymbol{c}] = \sum_{k=1}^{K-m} (i_k \star_p i_{k+m})[\boldsymbol{c}]$$

where $\star_p$ is phase correlation. Examples of correlation sums for experimental data are given in Figure 3.2.

The location of the peak of the correlation sum varies with $\boldsymbol{c}m$, so in order to add all the correlation sums into a single surface they must be scaled spatially so that their peaks coincide. Finally, the method adds these scaled correlation sums and finds the peak location. This algorithm can be summarized mathematically as

$$\hat{\boldsymbol{c}} = \frac{1}{K-1} \arg\max_{\boldsymbol{c}} \sum_{m=1}^{K-1} CS_m \left[\frac{m}{K-1}\boldsymbol{c}\right] \tag{3.5}$$

$$= \frac{1}{K-1} \arg\max_{\boldsymbol{c}} \sum_{m=1}^{K-1} \sum_{k=1}^{K-m} (i_k \star_p i_{k+m}) \left[\frac{m}{K-1}\boldsymbol{c}\right] \tag{3.6}$$

Figure 3.3 illustrates the algorithm and is color-coded to correspond to Equation 3.6.

As we will see in the numerical experiments section, this algorithm yields significantly more accurate results than a naively extended application of the Guizar-Sicairos method due to stronger assumptions about interframe motion.

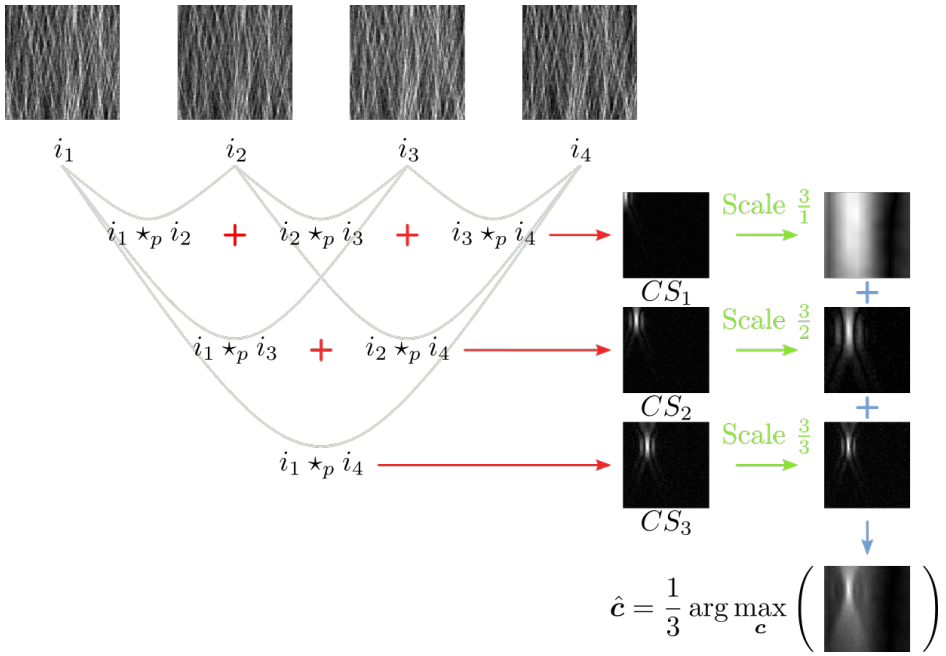

Figure 3.3: Illustration of multiframe registration method.

3.3 Optimality

This section shows that the above algorithm is derived from the maximum-likelihood (ML) solution to the multiframe registration problem with a particular set of assumptions. The result is first shown in the simple case of a two frame registration problem, then extended to multiple frames.

3.3.1 Two Frames

The ML solution derivation for the two frame registration problem makes the following simplifying assumptions:

- Images are circularly shifted rather than linearly shifted.
- Observations are corrupted with additive white Gaussian noise with variance σ^2.

Let $\mu = [\mu_1, ..., \mu_N]^T$ be a vector representing the ground truth input scene.
Let $C_c(\cdot)$ be a circular shift operator by c positions.
Let $i_1 = [i_{1,1}, ..., i_{1,N}]^T \sim \mathcal{N}(C_c(\mu), \sigma^2)$ be a noisy observation of μ shifted by integer c.

The likelihood of i_1 given c is

$$P(i_1|c) = \prod_{n=1}^{N} P(i_{1,n}|c) = \prod_{n=1}^{N} \phi\left(\frac{i_{1,n} - C_c(\mu)_n}{\sigma}\right)$$

where ϕ is the probability density function of the standard normal distribution. Thus the ML solution for c is

$$\begin{aligned}
\hat{c} &= \arg\max_c \ln P(Y|c) \\
&= \arg\max_c \sum_{n=1}^{N} \ln \exp\left[-\frac{(Y_n - C_c(\mu)_n)^2}{\sigma^2}\right] \\
&= \arg\min \sum_{n=1}^{N} [Y_n - C_c(\mu)_n]^2 \\
&= \arg\min_c \cancel{\sum_{n=1}^{N} Y_n^2} - 2\sum_{n=1}^{N} Y_n C_c(\mu)_n + \cancel{\sum_{n=1}^{N} \mu_n^2}
\end{aligned}$$

Since the first and third terms are constant over c:

$$= \arg\max_c \sum_{n=1}^{N} Y_n C_c(\mu)_n = \arg\max_c \langle Y,\, C_c(\mu)\rangle$$

where $\langle \cdot, \cdot \rangle$ is inner product.

Recognizing that the above inner product is a circular cross correlation, we conclude that argmax of cross correlation is the ML solution for two frames.

$$\hat{c} = \arg\max_c \langle Y,\, C_c(\mu)\rangle = \arg\max_c (Y \star \mu)[c]$$

where $\star$ is circular cross correlation.

Note that the solution still depends on the ground truth μ. This will be addressed in the next section.

3.3.2 Multiple Frames

The ML solution for a multiframe registration problem is derived using the same approach as the two frame case. The outline of this proof is as follows:

1. Derive joint probability of observing all frames given the drift between frames.

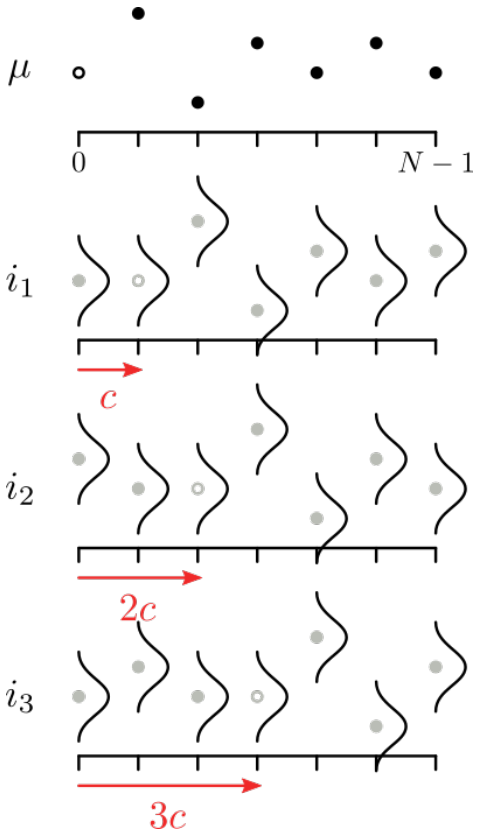

Figure 3.4: Circularly shifted 1D observations corrupted with i.i.d. Gaussian noise.

2. Derive ML solution to finding the drift assuming the ground truth scene is known.

3. Since the ground truth is generally not known, substitute it with the average of the registered frames.

4. Expand terms and recognize the expression is a sum of correlations between frames.

5. Substitute sum of correlations with *correlation sum* as defined in Section 3.2.

Let $i_k = [i_{k,1}, ..., i_{k,N}]^T \sim \mathcal{N}(C_{ck}(\mu), \sigma^2)$ be the kth noisy observation of μ shifted by ck in a sequence of K frames of length N. This is illustrated in Figure 3.4.

The likelihood of $i_1, ..., i_K$ is then

$$P(i_1, ..., i_K|c) = \prod_{k=1}^{K} P(i_k|c) = \prod_{k=1}^{K}\prod_{n=1}^{N} P(i_{k,n}|c)$$
$$= \prod_{k=1}^{K}\prod_{n=1}^{N} \phi\left(\frac{i_{k,n} - C_{ck}(\mu)_n}{\sigma}\right)$$

And the ML solution for c is

$$\begin{aligned}
\hat{c} &= \arg\max \ln P(i_1, ..., i_K|c) \\
&= \arg\max_c \sum_{k=1}^{K}\sum_{n=1}^{N} \ln \exp\left[-\frac{(i_{k,n} - C_{ck}(\mu)_n)^2}{\sigma^2}\right] \\
&= \arg\min_c \sum_{k=1}^{K}\sum_{n=1}^{N} [i_{k,n} - C_{ck}(\mu)_n]^2 \\
&= \arg\min_c \sum_{k=1}^{K}\sum_{n=1}^{N} i_{k,n}^2 - 2i_{k,n}C_{ck}(\mu)_n + \mu_n^2 \\
&= \arg\max_c \sum_{k=1}^{K}\sum_{n=1}^{N} i_{k,n}C_{ck}(\mu)_n = \arg\max_c \sum_{k=1}^{K} \langle i_k,\, C_{ck}(\mu)\rangle
\end{aligned}$$

However, since the ground truth scene μ is generally not known in image registration problems, this solution is not directly usable. Instead we can make an additional assumption that μ is equal to the average of the corrected frames.

$$\mu = \frac{C_{-c}(i_1) + ... + C_{-Kc}(i_K)}{K}$$

.

Substituting this definition of μ into the above result yields

$$\begin{aligned}
\hat{c} &= \arg\max_c \sum_{k=1}^{K} \langle i_k,\, C_{ck}(\mu)\rangle \\
&= \arg\max_c \sum_{k=1}^{K} \left\langle i_k,\, C_{ck}\left(\frac{C_{-c}(i_1) + ... + C_{-Kc}(i_K)}{K}\right)\right\rangle
\end{aligned}$$

Since C is a linear operator and since K does not depend on c:

$$= \arg\max_c \sum_{k=1}^{K} \langle i_k,\, C_{ck}\left(C_{-c}(i_1)\right)\rangle + ... + \langle i_{k,n},\, C_{ck}\left(C_{-Kc}(i_K)\right)\rangle$$

Since $C_a(C_b(\cdot)) = C_{a+b}(\cdot)$:

$$= \arg\max_c \sum_{k=1}^{K} \left\langle i_k,\, C_{(k-1)c}(i_1)\right\rangle + ... + \left\langle i_k,\, C_{(k-K)c}(i_K)\right\rangle$$

$$= \arg\max_c \sum_{k=1}^{K} (i_k \star i_1)[(k-1)c] + ... + (i_k \star i_K)[(k-K)c]$$

Without loss of generality, assume the number of frames $K = 4$:

$$\begin{aligned} = \arg\max_c \quad & (i_1 \star i_1)[0] + (i_1 \star i_2)[-c] + (i_1 \star i_3)[-2c] + (i_1 \star i_4)[-3c] \\ & + (i_2 \star i_1)[c] + (i_2 \star i_2)[0] + (i_2 \star i_3)[-1c] + (i_2 \star i_4)[-2c] \\ & + (i_3 \star i_1)[2c] + (i_3 \star i_2)[c] + (i_3 \star i_3)[0] + (i_3 \star i_4)[-1c] \\ & + (i_4 \star i_1)[3c] + (i_4 \star i_2)[2c] + (i_4 \star i_3)[c] + (i_4 \star i_4)[0] \end{aligned}$$

Using the fact that for any two frames i_a and i_b, $(i_a \star i_b)[c] = (i_b \star i_a)[-c]$ and $(i_a \star i_b)[0]$ is a constant:

$$\begin{aligned} = \arg\max_c \quad & \cancel{(i_1 \star i_1)[0]} + \\ & + 2(i_2 \star i_1)[c] + \cancel{(i_2 \star i_2)[0]} \\ & + 2(i_3 \star i_1)[2c] + 2(i_3 \star i_2)[c] + \cancel{(i_3 \star i_3)[0]} \\ & + 2(i_4 \star i_1)[3c] + 2(i_4 \star i_2)[2c] + 2(i_4 \star i_3)[c] + \cancel{(i_4 \star i_4)[0]} \end{aligned}$$

Reorganizing terms by the argument within the braces:

$$\begin{aligned} = \arg\max_c \quad & (i_2 \star i_1)[c] + (i_3 \star i_2)[c] + (i_4 \star i_3)[c] + \\ & + (i_3 \star i_1)[2c] + (i_4 \star i_2)[2c] \\ & + (i_4 \star i_1)[3c] \\ = \arg\max_c \, & CS_1[c] + CS_2[2c] + CS_3[3c] \end{aligned}$$

We see that the result for arbitrary K is a sum of scaled correlation sums.

$$\hat{c} = \arg\max_c \sum_{m=1}^{K-1} CS_m[mc]$$

where a correlation sum is defined as

$$CS_m[c] = \sum_{k=1}^{K-m} (i_k \star i_{k+m})[c]$$

In order to arrive at this solution, it was necessary to make several assumptions in the problem formulation:

1. Images are circularly shifted rather than linearly shifted.

2. Observations are corrupted with additive white Gaussian noise with variance σ^2.

3. The ground truth is equal to the average of all registered frames.

The purpose of the circular image assumption was to simplify the ML solution derivation by making many terms cancel. In reality, images obtained of a scene are generally going to be linearly translated as new, unobserved parts of the scene drift into the field of view. However, this assumption holds approximately so long as the total drift across the sequence of frames is small relative to the size of the field of view. Additionally, there are windowing techniques that can be applied to the frames before registration to reduce edge effects.

The second assumption on additive Gaussian noise simplifies the derivation of the likelihood function. The noise model for the VISORS project consists of both additive Gaussian noise and Poisson noise. With the use of the Anscombe transform the observations can be adjusted to approximate additive Gaussian noise with better approximations as the image intensity increases. As will be seen in the numerical experiments, the algorithm still seems to perform well on unadjusted frames generated with this more complex noise model.

As explained previously, the last assumption was necessary because the ground truth μ is not known during registration. In general, the ground truth is not exactly equal to average of registered frames. However, this approximation holds as the number of frames increases or the noise variance decreases.

Finally, two small tweaks have been experimentally found to improve performance. First, instead of downsampling the correlation sums to CS_1 to get

the peaks to coincide, upsampling to CS_{K-1} improves registration precision.

$$\hat{c} = \frac{1}{K-1} \arg\max_{c} \sum_{m=1}^{K-1} CS_m \left[\frac{m}{K-1} c \right]$$

This gives the algorithm subpixel precision down to $\frac{1}{K-1}$. This fractional upsampling implicitly requires interpolation which can be implemented a number of ways, but sinc interpolation works well.

Aside from its registration accuracy, which will be explored in the next section, the algorithm has some particular advantages which should be pointed out.

Firstly, the algorithm is non-iterative. Compared to an iterative algorithm with some stopping criterion, this algorithm will take a fixed amount of time to obtain a registration estimate for a fixed number of frames of a fixed size. This can be useful in real-time applications where computational resources are tight and determining a time budget is critical.

Next, the algorithm should be able to be implemented fairly efficiently since it consists of FFTs, a spatial scaling, and elementwise multiplication. Again, this is important for resources constrained and time sensitive applications.

Lastly, the algorithm needs no parameter tuning or training since it is derived from maximum likelihood. For the VISORS project in particular, this is helpful because the corona has not been observed in such high resolution before and the structure (or existence) of nanoflares is not known for certain.

CHAPTER 4

NUMERICAL EXPERIMENTS

In order to evaluate the performance of the registration algorithm in the previous section, it is necessary to have a sequence of frames with known interframe coordinate transform parameters so that the algorithm's computed parameters can be compared. In other publications involving development of registration methods, this sequence is sometimes captured from a physical system with additional sensors to capture ground truth information of the interframe motion. However, more often the authors make use of theoretical models of their instrumentation and of the scene being imaged to create simulated data where the ground truth is decided during data synthesis. While the accuracy of this approach depends on the fidelity of these models, it comes with the advantage that parameters within these models like noise, exposure time, motion type and others can all be changed freely to evaluate algorithm performance in a variety of conditions.

In the VISORS project, we must take this approach because measurements made from the hardware will not be available until the spacecraft launches. Additionally, since VISORS will be making measurements at higher resolutions than previously achieved, it is not known for certain what structures (if any) are present at these spatial scales.

This chapter describes tests developed to evaluate the registration algorithm's efficacy on the VISORS project. Section 4.1 describes how the test frames are generated, including descriptions of the models used to create the input scenes, blurring due to optical and motion effects, various noise sources, and physical properties of the camera. Section 4.2 evalutes the algorithm while sweeping various model parameters and compares it against Guizar-Sicairos's registration algorithm for both speed and accuracy, additionally investigating effects of motion model mismatch on the accuracy of the registration.

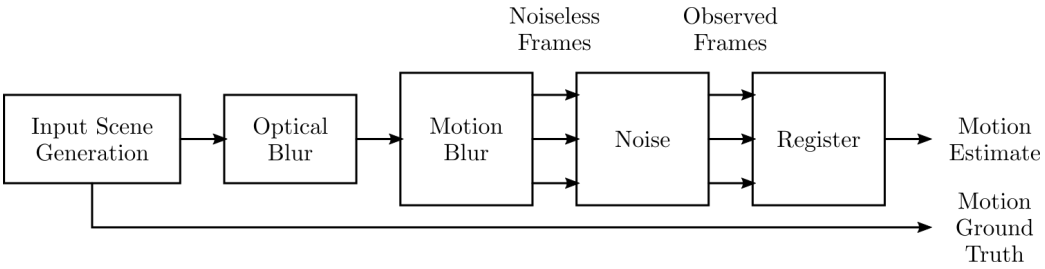

Figure 4.1: Registration algorithm evaluation pipeline.

4.1 Generation of Test Data

As mentioned previously, a simulation which is as high-fidelity as possible is important to ensure the algorithm performance adequately when VISORS is deployed. The simulation should be able to both generate a plausible ground truth scene and capture effects of distortions that the imaging system will experience while making a measurement. To this end, this section develops a simulation pipeline which can generate a sequence of observed frames. The pipeline parameters may be swept over a range of plausible values, which will be utilized in a later section to validate the registration algorithm.

The blocks of this pipeline are illustrated in Figure 4.1 and described in detail in following sections.

4.1.1 Input Scene Generation - Simplified Nanoflare Model

VISORS team member Klimchuk has studied coronal loops extensively and has used a hydrodynamical model called Enthalpy-based Thermal Evolution of Loops (EBTEL) to create a video of a changing scene containing many loops on a small patch of the Sun [22]. Within this small patch, nanoflares appear as superposed wide stripes across the entire scene which fade in and out at various inclinations and positions.

Instead of using the EBTEL frames directly, the pipeline uses a custom Python function which generates frames that visually approximate the EBTEL frames, but at any field of view and resolution. A comparison of an EBTEL frame and the Python approximation is given in Figure 4.2. Additionally, any dynamic evolution of the nanoflares is ignored, as the total change expected over VISORS 10 second exposure is expected to be minimal, as shown in Figure 4.3. More detail is given about VISORS exposure time

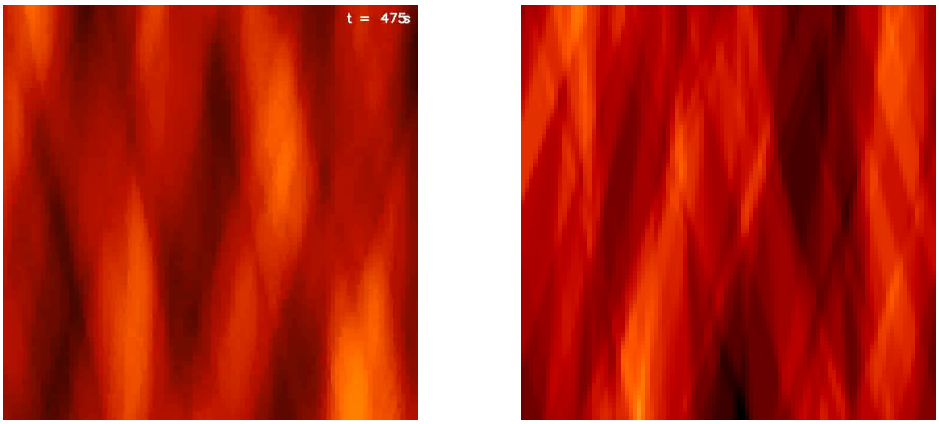

Figure 4.2: Simulated field of views from EBTEL-generated video sequence and our Python approximation, respectively. Field of view corresponds to square patch on Sun with sides about 2000km.

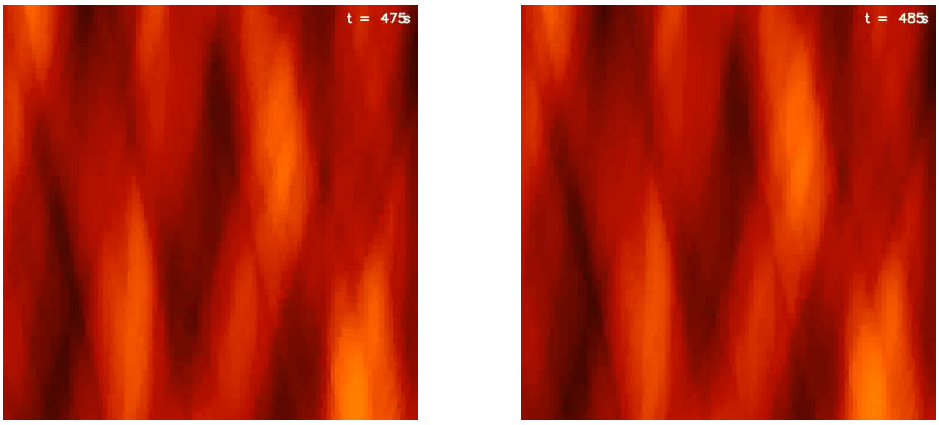

Figure 4.3: Two frames from EBTEL video separated by 10 s. Very little evolution occurs over such a short window, which justifies the use of a static scene for simulation purposes.

and shutter speed in Section 4.1.5

4.1.2 Input Scene Generation - AIA Upsampled Data Model

The accuracy of registration algorithms is dependent on the frequency content of the images being registered, so the presence or absence of nanoflares in the observed data is an important consideration in the simulation pipeline. In order to establish worst-case bounds on the registration accuracy, it is possible to use lower-resolution measurements from previous missions with artificial fine structure added and evaluate the effect on the registration ac-

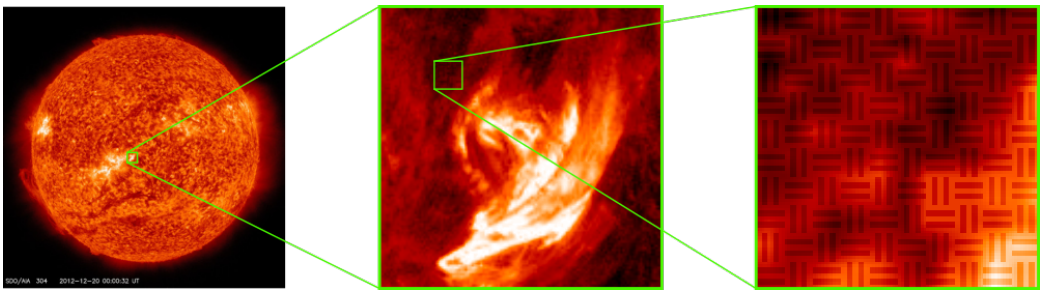

Figure 4.4: Full photosphere capture from AIA, cropped region corresponding to input scene, and zoomed view showing synthetic modulated structure.

Table 4.1: AIA and VISORS detector parameters.

Sensor Parameters		
r_{visors}	72 mas	VISORS angular resolution
r_{aia}	600 mas	AIA angular resolution
N_{aia}	750 pixels	VISORS detector width

curacy.

The Solar Dynamics Observatory (SDO) is one such mission with a resolution of 600 mas on its Atmospheric Imaging Assembly (AIA) instrument [23]. The AIA captures images of the full photosphere at many wavelengths, including several in the EUV spectrum.

The angular resolutions and detector sizes of VISORS and AIA (summarized in Table 4.1) determine the FOV of VISORS within the AIA image to be 90 pixels as in Equation 4.1, which is upsampled to the VISORS detector size of 750 pixels. Finally, small scale structure of size 2 pixels (computed in Equation 4.2) is artifically added. This process is illustrated in Figure 4.4.

$$\text{(upsample factor)} = r_{aia}/r_{visors} = 8.33$$

$$\text{(FOV size)} = N_{visors}/\text{(upsample factor)} = 90 \text{ pixels} \tag{4.1}$$

$$\text{(Nanoflare size)} = \text{(Nanoflare angular size)}/r_{visors} = 2.08 \text{ pixels} \tag{4.2}$$

With two different modles for generating input scenes, the next step in the pipeline is to simulate physical distortions as the scene propagates through space and the imaging instrument.

Table 4.2: Expected photon rates in VISORS mission for 60 arcsecond FOV pointed at brightest active solar regions.

Expected VISORS photon rates		
a_{mean}	1.2 photons/s/pixel	average photon rate
a_{max}	20 photons/s/pixel	photon rate for brightest pixel

4.1.3 Input Scene Scaling

The input scenes generated thus far have dimensionless intensities ranging from 0 to 255. In later sections, especially the noise analysis in Section 4.1.6, it is important that the intensities of the simulated observed frames are realistic as this has a direct impact on the level of detail visible and thus registration accuracy.

The VISORS detector spacecraft carries a complementary metal-oxide semiconductor (CMOS) sensor which is sensitive to photons emitted by the solar photosphere. Due to the discrete nature of light, a countable number of photons will impinge upon each pixel on the detector within a given window. Since the integration time T is known, the measured intensity is usually given as photons/s/pixel, which describes the average number of photons detected by a pixel per second.

VISORS team member Daw performed an analysis of multiple years of image data from AIA (shown in Figure 4.5) and determined both the average and maximum photon arrival rates of bright regions on the photosphere, accounting for the smaller size of VISORS pixels and losses due to VISORS optical path. These computed values are shown in Table 4.2.

In order to scale the unitless generated scenes to have a intensities measured in photons/s/pixel with the desired average a_{avg} and maximum a_{max} photon arrival rates, the pipeline uses the affine mapping

$$(a_{max} - \operatorname{avg}(I)) \frac{I - \operatorname{avg}(I)}{\max I - \operatorname{avg}(I)} + a_{avg} \to I$$

where I is the simulated input scene and $\to$ is an assignment operator.

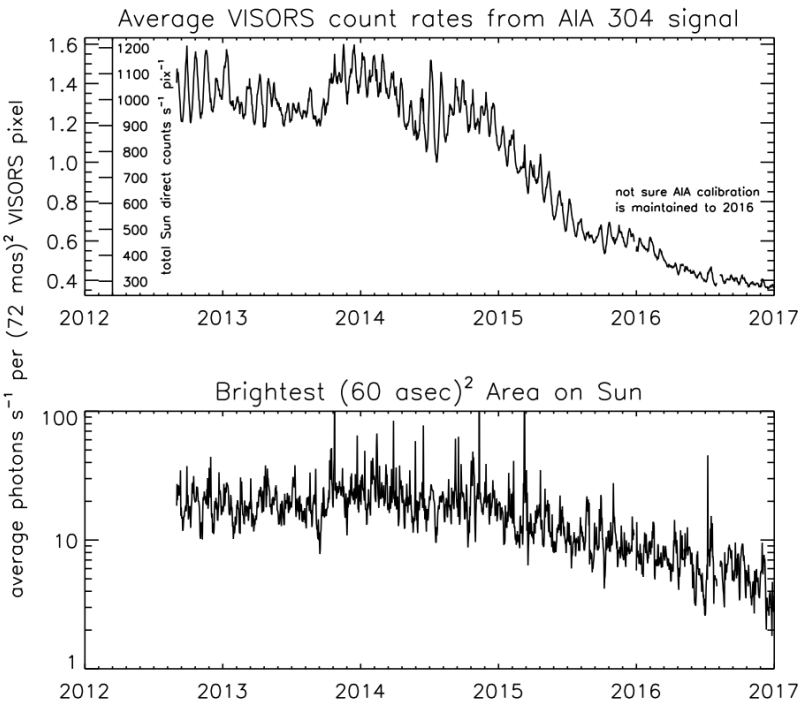

Figure 4.5: Multi-year analysis of AIA photon arrival rates.

4.1.4 Optical Propagation

In the VISORS project, light propagates from the Sun to the optics spacecraft, where it is diffracted by the photon sieve and again propagates to the sensor within the detector spacecraft. This has the effect of slightly blurring the input scene even when the optics and detector spacecraft are in perfect focus. An exact simulation of this process is intractable as it would require propagation of each pixel within the input scene through each hole in the photon sieve and propagation through free space to the detector. Oktem et al. [24] use the Fresnel approximation to obtain a closed form expression which relates the input scene to the image observed at the sensor in the detector spacecraft. The transfer function is given terms of simple functions requiring no numerical integration techniques.

$$\tilde{h}_{sieve}(\omega_x, \omega_y) = \frac{(\lambda d_i)^4}{2} e^{i\pi\Delta\lambda d_i^2(\omega_x^2+\omega_y^2)} \mathrm{circ}\left(\frac{\lambda d_i}{D}\omega_x, \frac{\lambda d_i}{D}\omega_y\right) + \frac{(-1)}{\pi}(\lambda d_i)^4 \mathrm{circ}\left(\frac{\lambda d_i}{D}\omega_x, \frac{\lambda d_i}{D}\omega_y\right)$$

Table 4.3: VISORS noise parameters. Dark current and read noise are determined by the CMOS specifications, while background noise has been computed by modelling light leakage around optics spacecraft.

VISORS Photon Sieve Parameters		
λ	30.4 nm	source wavelength
w	17 μ m	sieve smallest hole diameter
D	75 mm	sieve diameter
d_s	$147.5 \cdot 10^6$ km	distance to Sun

where $\Delta = 1/d_i + 1/d_s$, ω_x and ω_y represent spatial frequency of the transfer function, d_i the distance from the photon sieve to the image plane, d_s the distance from the photon sieve to the source plane, λ the wavelength of the incoming light, and circ is a 2D unit circular aperture function. The numerical values for sieve parameters are given in Table 4.3.

This transfer function can be easily computed numerically and transformed into a point spread function (PSF) like the one shown in Figure 4.6, which can be convolved with the input scene to model optical path blurring. Since we want to measure at the point at which the image is in focus, the distance to the image plane should be equal to the focal length so that

$$d_i = f = \frac{Dw}{\lambda}$$

Now the transfer function can be computed and applied to the input scene using

$$I * h_{sieve} \rightarrow I$$

where I is the input scene and h_{sieve} is the inverse Fourier transform of the transfer function $\tilde{h}_{sieve}$.

The effect of this optical blur is shown in Figure 4.7 on a zoomed patch of the AIA input scene. The synthetically added nanoflare modulation pattern is visibly degraded after the blurring kernel has been applied.

4.1.5 Motion Blurring

In order for the CMOS sensor on board the VISORS detector spacecraft to capture enough signal in these photons and overcome various sources of noise

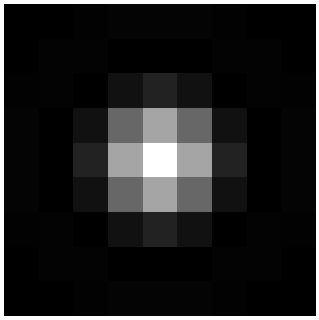

Figure 4.6: Point spread function representing distortions due to VISORS optical path.

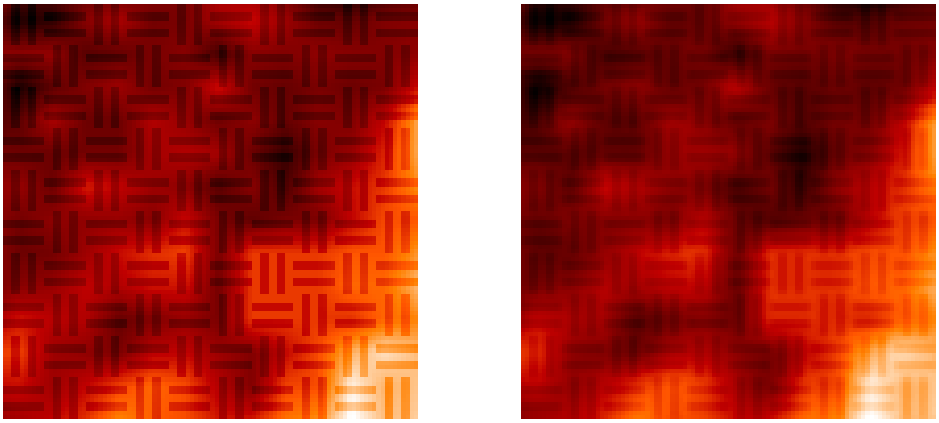

Figure 4.7: Zoomed patch of AIA input scene before and after blurring by photon sieve point spread function.

described in Section 4.1.6, it is necessary to hold the shutter open on the order of 1-10 seconds. During this period, solar features within the VISORS FOV will have drifted significantly, causing motion-induced blurring and further degrading the captured frames. From navigational requirements established by team members Koenig and D'Amico [25], at a 4 Hz capture rate solar features can be blurred by over 3 pixels, [1] corresponding to an angular resolution of 216 mas. The ability to simulate the effects that frame integration time and spacecraft drift have on the observed frames is therefore extremely valuable both for evaluating performance of the registration algorithm and further constraining mission design parameters of the detector.

To simulate motion blur effects, the pipeline uses the concept of a high-resolution grid described in Section 2.2 where the pixel pitch of the observed frames is less than the pixel pitch of the underlying ground truth image. In this thesis, the ratio of the high-resolution grid pitch and detector pixel pitch is $d = 2$, shown in Figure 4.8.

First, the pipeline computes the path taken during each frame by multiplying the drift rate $\boldsymbol{c}$ and total experiment duration, then divides this line into K segments of equal length. Since this path is expressed in continuous coordinates, it must be discretized to the high-resolution grid. This is achieved using Bresenham's line algorithm, which approximates a straight line drawn on a grid [26]. The algorithm result and subdivided line are illustrated in Figure 4.9.

After the FOV path for each frame is calculated, the FOV is simply integrated for each pixel in each frame to get the observed frame, as shown in Figure 4.10. This is summarized mathematically as

$$i_k[\boldsymbol{x}] = \frac{T}{|B_k(\boldsymbol{c})|} \sum_{\boldsymbol{x}_b \in B_k(\boldsymbol{c})} I[\boldsymbol{x} + \boldsymbol{x}_b] \tag{4.3}$$

where $B_k(\boldsymbol{c})$ is the set of pixel offsets for the kth frame and a drift of $\boldsymbol{c}$ determined by Bresenham's algorithm. In order to keep an image intensity which is physically meaningful (photons/s/pixel), the integrated frames are scaled by the frame integration period T and divided by the number of summands in the path segment.

An alternative to the above strategy might have been to derive a blurring

[1] (0.2 mm/s drift) / (14 micron CMOS pixel pitch) / 4 Hz = 3.57 pixels

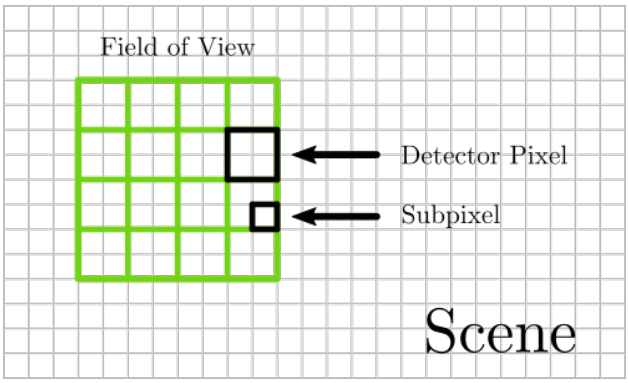

Figure 4.8: Detector instantaneous field of view and simulation high-resolution grid.

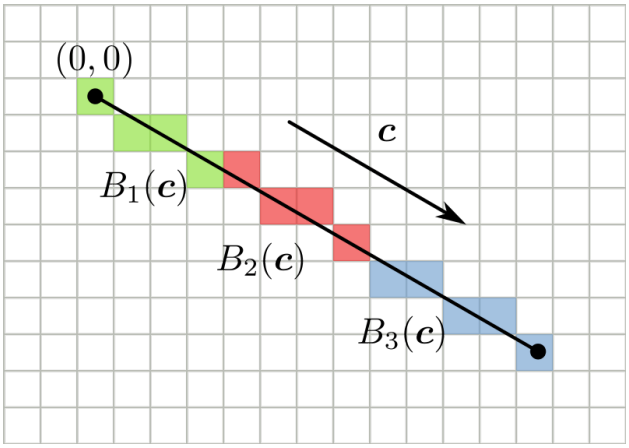

Figure 4.9: Discrete path traced out by frames using Bresenham's line algorithm.

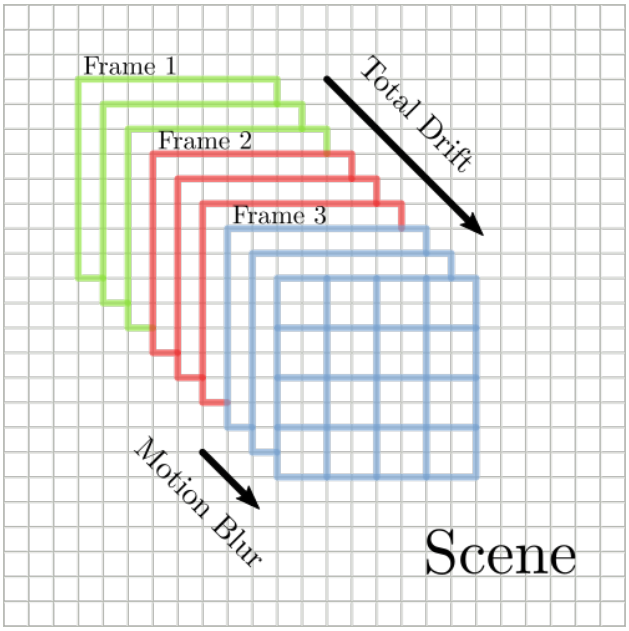

Figure 4.10: Frame integration of high-resolution grid under constant linear motion.

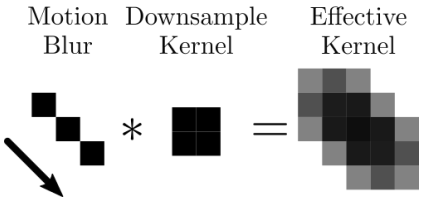

Figure 4.11: Effective kernel due to motion and downsampling.

kernel to apply to all pixels in a cropped region of the input scene representing instantaneous position of the VISORS FOV. This would be computed by convolving the FOV motion path within 1 frame with a unit function representing the size of a detector pixel on the high-resolution grid to get an effective blurring kernel, as shown in Figure 4.11. However, the pipeline utilizes the above technique for flexibility of motion prior.

In order for each pixel to have the same relative motion path, motion must be strictly translational. Figure 4.12 shows motion with a rotational component, where the shape of the path of each pixel is unique and the motion blurring is no longer spatially invariant. While the multiframe registration method proposed models spacecraft motion as translational and linear, simulating motion blur in this way allows us to test motion model mismatch on the registration result. Figure 4.13 shows an example of the effects of motion

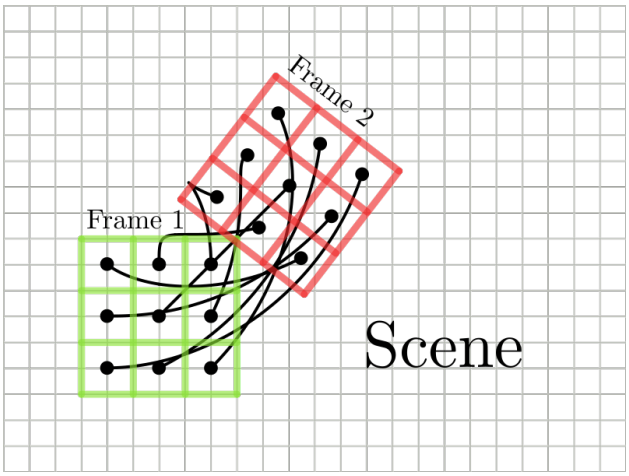

Figure 4.12: Under non-translational motion, each detector pixel traces a unique path, so motion blurring is no longer spatially invariant.

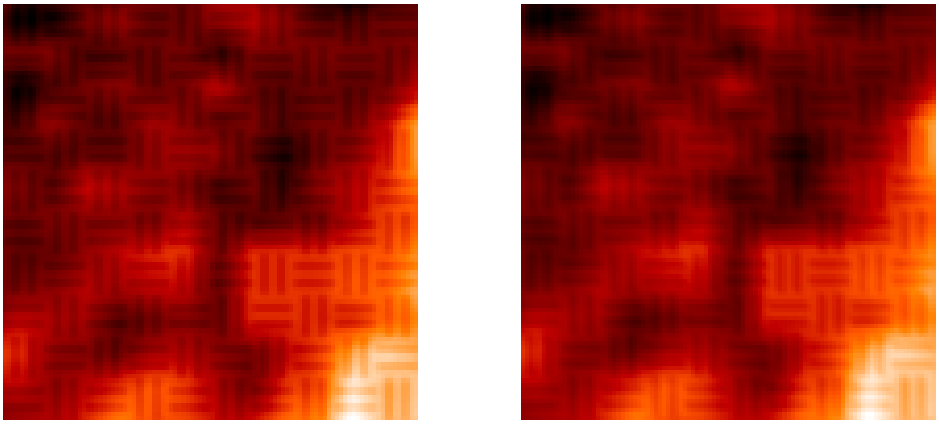

Figure 4.13: Zoomed patch of AIA input scene before and after motion-induced blurring.

induced blurring modelled in this way.

4.1.6 Noise Sources

CMOS sensors are susceptible to several types of noise. This section models three types of noise.

The first source of noise is referred to as *dark current noise*. Dark current noise is created by random movement of charge within the sensor which falsely registers as captured photons and is often caused by leakage current in the semiconductors used to fabricate the sensor [27]. It is present even when no photons enter the detector (a dark field of view) and is characterized

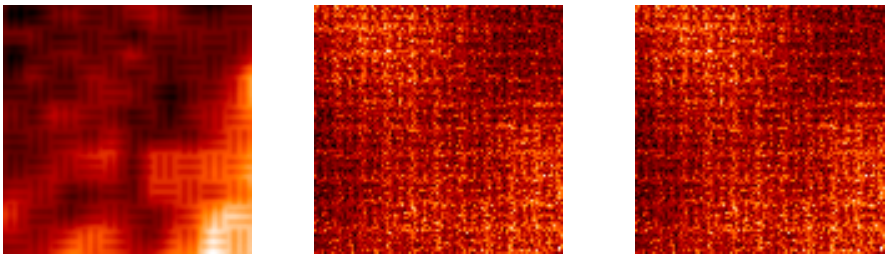

Figure 4.14: Zoomed patch of observed frame with no noise, after shot noise, and after shot noise and read noise.

Table 4.4: VISORS noise parameters. Dark current and read noise are determined by the CMOS specifications, while background noise has been computed by modelling light leakage around optics spacecraft.

VISORS noise parameters		
n_d	0.8 photons/s/pixel	dark current noise
n_b	2 photons/s/pixel	background noise
n_r	1 photons/s/read	read noise

by the sensor manufacturer. Dark noise is especially important to consider in low light conditions when the useful signal generated by impinging photons is overtaken by random signal generated by electrons. Dark noise increases with increasing integration time.

Another source of noise is *background noise* which is caused by light entering the detector from outside the optical path. While this background light is caused by real photons hitting the detector, it is generally categorized as noise because these photons do not originate from the field of view being studied. Background noise also increases with integration time.

These sources of noise are due to the quantum nature of energy and follow a Poisson process. Together with the photons from the source itself they are referred to as *shot noise* (though sometimes the photons coming from the source itself are named separately as *photon shot noise*).

The final source of noise is *read noise*, which occurs at the end of the frame integration interval. Total accumulated charge is read from each pixel on the detector by an analog to digital converter (ADC). Read noise is caused by interference from internal electronics and environmental sources and does not depend on integration time.

The observation model used for generating each observed noisy frame i_k is

$$\mathcal{N}\Big(\text{Pois}(i_k + T(n_d + n_b)),\, n_r^2\Big) \to i_k$$

An example of a noisy observation frame after shot noise and read noise is shown in Figure 4.14

4.2 Registration Results

This section characterizes the error in the registration algorithm by running it on simulated noisy frames generated by the forward model and computing the Euclidean distance between the estimated drift $\hat{\boldsymbol{c}}$ and actual drift $\boldsymbol{c}$.

$$\text{err} = \|\hat{\boldsymbol{c}} - \boldsymbol{c}\|_2$$

While error in the drift estimated by the registration algorithm can be used as a metric for algorithm performance, it is also informative to look at the reconstruction derived from this drift estimate. Below is a simple frame coadding technique to obtain a reconstruction.

$$I_{recon} = \sum_{k=1}^{K} i_k \left[\boldsymbol{x} - \text{round}(k\hat{\boldsymbol{c}})\right]$$

This method serves as a baseline only and can certainly be improved by techniques such as super-resolution, denoising, motion deblurring, and regularization.

The rest of this section evaluates reconstruction performance under the two types of input scenes presented previously, investigates algorithm accuracy while varying noise levels and motion in a Monte Carlo simulation, and examines the effect on reconstructions when the constraint of constant translational motion between frames is relaxed.

Unless otherwise stated, experiments were performed with $T = 0.25$, $K = 40$, $a_{max} = 20$ and $\boldsymbol{c} = [.675, .675]$.

4.2.1 Input Scene Model

Sections 4.1.1 and 4.1.2, presented two methods for generating an input scene. The first is based upon hydrodynamical simulations of nanoflare evolution

Table 4.5: VISORS noise parameters. Dark current and read noise are determined by the CMOS specifications, while background noise has been computed by modelling light leakage around optics spacecraft.

VISORS noise parameters		
T	0.25 s	frame integration time
K	40 frames	number of frames
a_{max}	20 photons/s	photon rate of brightest pixel
$\boldsymbol{c}$	$[0.141, 0.141]$ mm/s $[2.53, 2.53]$ pixels/s	interframe drift

and has been scaled to the expected nanoflare size. The second is derived from low-resolution images from AIA which has been synthetically altered to add some fine-grained structure.

Figure 4.15 is an illustration of a test with the nanoflare input scene that shows the ground truth, a single noisy frame out of a set of 40, and the coadded reconstruction after running the registration algorithm. The algorithm was successful and the 40 input frames were registered and coadded to obtain a reconstruction which is much higher fidelity than any single frame.

Figure 4.16 is a similar test using an upsampled AIA input scene, but has a second row so that the synthetic structure is visible. The results from this experiment are notable, because even though the fine synthetic structure is no longer visible in a single frame due to noise, the algorithm is able to register the frames accurately enough so that these structures become visible again in the reconstruction. The parameters used in these experiments are given in Table 4.5.

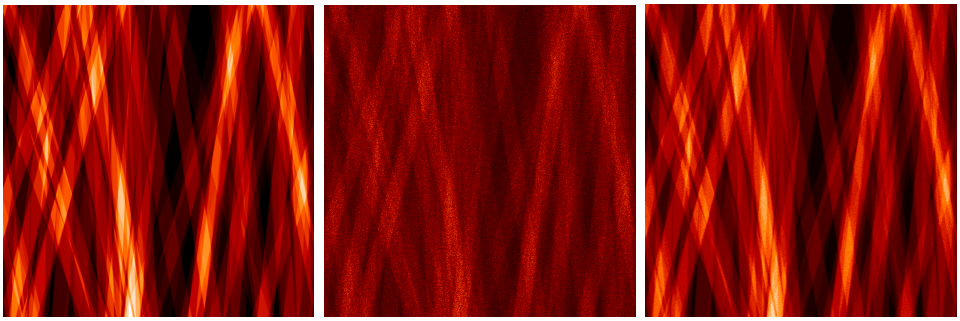

Figure 4.15: Modulated nanoflare input scene, noised + blurred observation frame, coadded reconstruction.

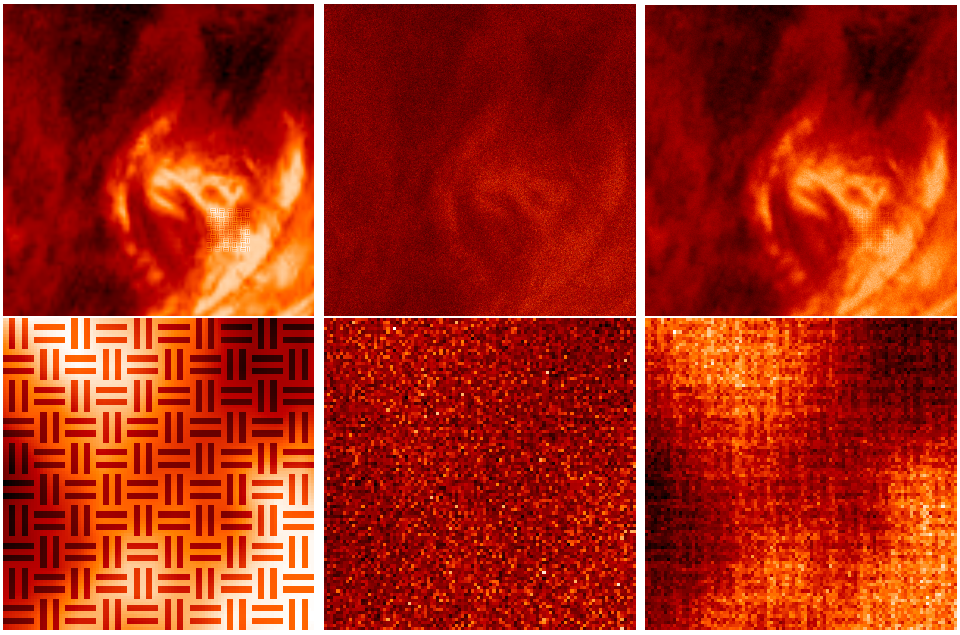

Figure 4.16: Top row, from left: Modulated AIA input scene, noised + blurred observation frame, coadded reconstruction. Bottom row, from left: Same as top row, but zoomed to show modulation detail.

4.2.2 Noise Variation

Maximum photon count a_{max} is a major contributor to shot noise in the observation model and dominates read noise as shown in Figure 4.14. The expected value of 20 photons/s/pixel for the maximum photon count in VISORS was derived by analyzing bright regions on the photosphere from many AIA images and compensating for differences in detector pixel size, quantum efficiency and efficiency of instrument optical paths. However, this is a worst case estimate and in reality may be higher if small scale structures are observed. Figure 4.17 illustrates reconstruction quality at several noise levels by sweeping the maximum photon count from 10 to 100. At 20 photon/s/pixel, fine structure is just barely visible in the reconstruction, and at 100 photons/s/pixel the modulated grid is quite clear.

4.2.3 Monte Carlo Simulation

Other system parameters besides noise can affect the registration result, such as drift velocity and drift angle.

Figure 4.18 shows a Monte Carlo simulation in which these three parame-

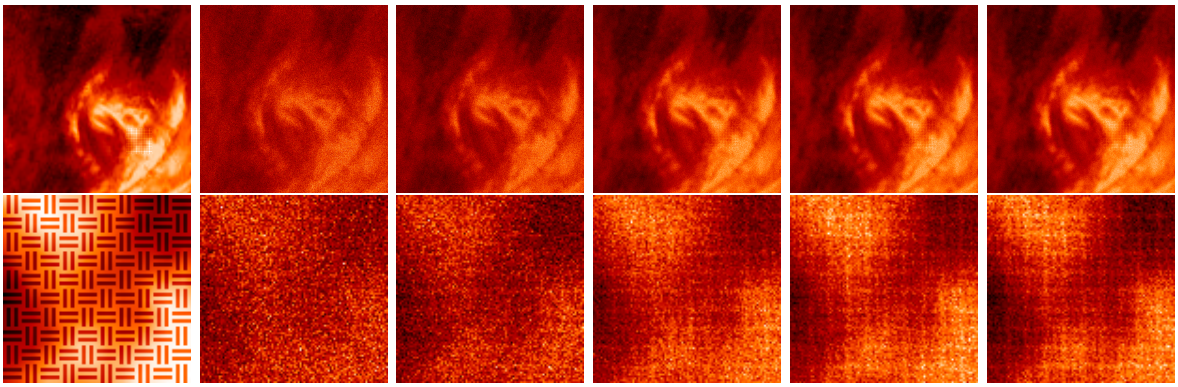

Figure 4.17: Top: AIA input scene ground truth, and reconstructions for input scaling (a_{max}) of 10, 20, 50, 80 and 100 photons/s/pixel. Bottom: Same as Top, but zoomed to show modulation detail.

ters were swept across a range of realistic values and the drift estimate error computed. Each test point was repeated 10 times to get an estimate of the mean and standard deviation of the error. Several conclusions can be drawn from this plot. First, mean estimation error decreases as noise decreases (increasing a_{max}) and converges to a mean of around 0.05. This number is corroborated by the reconstruction experiments in the previous section because an interframe drift estimate error of 0.05 corresponds to a total alignment error of $39 \times 0.05 \approx 1$ pixels between the first and last frames in the sequence, which is less than the nanoflare feature size of 2 pixels derived in Section 4.1.2.

Next, drift estimation error improves slightly with a decreasing drift velocity. This is possibly due to a decrease in the amount of motion blur which better preserves high frequency structure that the registration algorithm uses for accurate drift estimation.

Finally, the drift error is fairly independent of the spacecraft's angle of drift. This is expected for the modulated AIA scene which has approximately equal frequency content in both the x and y directions, but the algorithm may perform differently for different drift angles if this is not the case.

4.2.4 Motion Model Mismatch

While the relative motion of solar features within VISORS field of view is expected to be constant and translational, it is useful to know the effect a

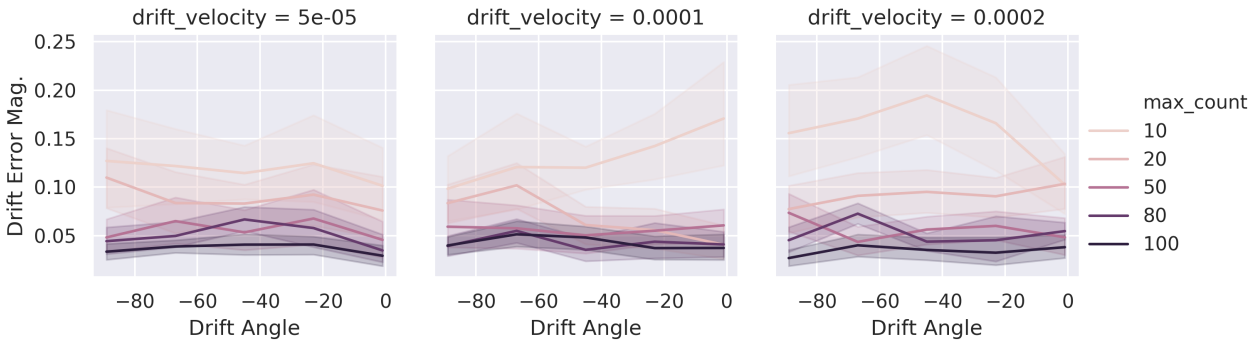

Figure 4.18: Monte Carlo simulation showing error in the drift estimate under varying drift rates, drift angles, and maximum photon counts (a_{max}). The solid line is the mean while the shaded region is ± 1 standard deviation.

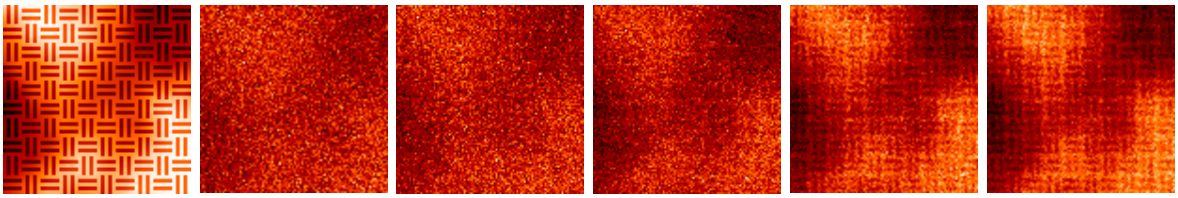

Figure 4.19: Zoomed reconstructions from upsampled AIA scene under different rates of roll θ.

violation of this motion model has on the registration result. Here the motion model has been adjusted to include a constant roll on the boresight axis of the formation. Tweaking the model given in Section 3.1, we now have the more general coordinate transform relations

$$f_{j,k}(\boldsymbol{x}) = R_{\theta(k-j)}\left(\boldsymbol{x} - \boldsymbol{c}(k-j)\right) \tag{4.4}$$

$$f_{j,k}(\boldsymbol{x}) = f_{l,m}(\boldsymbol{x}) \tag{4.5}$$

for $j, k, l, m \in 1, ..., K$ when $k - j = m - l$. θ is the roll rate of the formation.

Figure 4.19 shows zoomed patches from reconstructions of the upsampled AIA scene under several rates of roll. As θ is increased, misalignment which cannot be accounted for by shifting and coadding dominates and the fine scale structure is no longer visible. The formation roll rate should be constrained to be below this threshold to successfully image such fine structures.

CHAPTER 5

CONCLUSION

This chapter summarizes the work presented in this thesis and describes potential future work.

5.1 Registration Algorithm and Forward Model

Chapter 1 introduced the VISORS project and the problem of image registration. It detailed several types of coordinate transforms that may be used in image registration and explained why a simple translational model is appropriate for the VISORS project. Chapter 2 described a framework for categorizing image registration methods and illustrated several algorithms capable of registering two frames, as well as algorithms capable of registering images on a subpixel level. Chapter 3 introduced the problem of multiframe image registration and related it to the two-frame problem with a mathematical model. Chapter 3 then detailed a new parameterless, non-iterative multiframe registration method which should have more robust performance than a simple extension of a two-frame method to multiple frames. This chapter showed that this method is similar to the ML solution when the drift between frames is assumed to be constant and linear.

5.2 Forward Model and Registration Results

Section 4.1 began describing a forward optical pipeline which simulates frames captured by VISORS starting from generating photosphere scenes to capture by the detector. The first step of this pipeline was the generation of the solar scene using a physics-based model, and another more conservative model based on data from previous orbiters. Next, these scenes were scaled

from dimensionless images to a map of photon arrival rates, then propagated through the VISORS optical system. These propagated scenes were then motion blurred using a method which is more sophisticated than global convolution which can account for spatially variant blurring in the case of complex spacecraft motion. Finally, sensor readout noise and various types of shot noise are modeled and applied to the images.

Section 4.2 contains an array of numerical experiments which illustrate the performance of the multiframe registration algorithm. This section showed that images can be recovered even when no structure is clearly visible in individual observation frames, and specifically that fine-grained structure in the modulated nanoflare scene is preserved given a sufficiently high SNR. Furthermore, Monte Carlo simulations characterizing the mean and standard deviation of the error in the algorithm's drift estimate showed that small amounts of model mismatch in the form of a rotational component in the coordinate transform model can be tolerated by the algorithm.

5.3 Future Work

The primary goal of this work was to develop a registration algorithm for the purposes of determining the presence of small scale nanoflares within the VISORS observations. As a result, the reconstruction scheme is simply to coadd the observed frames using the registration algorithm's estimate of interframe drift. There is much room for improvement to boost the quality of the reconstructions.

The motion blur introduced in Section 4.1 is not accounted for during reconstruction. There are many motion deblurring algorithms available, from methods removing simple Gaussian blurring to more sophisticated methods that can account for complex motion, all of which should sharpen the reconstructions. Furthermore, the motion blurring kernel can be computed from the final drift estimate, which means that non-blind deblurring methods may be used.

Finally, super-resolution techniques may be applied which may reveal more fine structures within the reconstruction. Many super-resolution methods either depend on accurate image registration as an initialization step or use the registration algorithm in an iterative fashion. The registration algorithm

proposed in this thesis could be substituted into these super-resolution techniques.

REFERENCES

[1] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.

[2] A. C. Bovik, *Handbook of Image and Video Processing*. Academic Press, 2010.

[3] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[4] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[5] J. A. Klimchuk, "Key aspects of coronal heating," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2042, p. 20140256, 2015.

[6] F. S. Oktem, F. Kamalabadi, and J. M. Davila, "High-resolution computational spectral imaging with photon sieves," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 5122–5126.

[7] D. I. Barnea and H. F. Silverman, "A class of algorithms for fast digital image registration," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 179–186, 1972.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[10] M. Rais, "Fast and accurate image registration. applications to on-board satellite imaging." Ph.D. dissertation, University of the Belearic Islands, 2016.

[11] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.

[12] O. Rioul, "This is it: A primer on Shannon's entropy and information," *L'Information, Seminaire Poincare*, vol. 23, pp. 43–77, 2018.

[13] H. Foroosh, J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 188–200, 2002.

[14] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 2, pp. 220–233, 1986.

[15] J. Li and A. D. Heap, "A review of spatial interpolation methods for environmental scientists," *Geoscience Australia Canberra*, 2008.

[16] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics Letters*, vol. 33, no. 2, pp. 156–158, 2008.

[17] D. Wüstner and N. J. Faergeman, "Chromatic aberration correction and deconvolution for UV sensitive imaging of fluorescent sterols in cytoplasmic lipid droplets," *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, vol. 73, no. 8, pp. 727–744, 2008.

[18] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150–1163, 2006.

[19] E. Barrett and M. R. eds., *Contextual Media: Multimedia and Interpretation*. MIT Press, 1995.

[20] S. Farsiu, M. Elad, and P. Milanfar, "Constrained, globally optimal, multi-frame motion estimation," in *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005.* IEEE, 2005, pp. 1396–1401.

[21] P. Milanfar, *Super-Resolution Imaging*. CRC Press, 2017.

[22] J. Klimchuk, S. Patsourakos, and P. Cargill, "Highly efficient modeling of dynamic coronal loops," *The Astrophysical Journal*, vol. 682, no. 2, p. 1351, 2008.

[23] J. R. Lemen, D. J. Akin, P. F. Boerner, C. Chou, J. F. Drake, D. W. Duncan, C. G. Edwards, F. M. Friedlaender, G. F. Heyman, N. E. Hurlburt et al., "The atmospheric imaging assembly (aia) on the Solar Dynamics Observatory (SDO)," in *The Solar Dynamics Observatory*. Springer, 2011, pp. 17–40.

[24] F. S. Oktem, F. Kamalabadi, and J. M. Davila, "Analytical Fresnel imaging models for photon sieves," *Optics Express*, vol. 26, no. 24, pp. 32 259–32 279, 2018.

[25] A. Koenig and S. D'Amico, "Orbit and GNC design for VISORS mission," 2020, unpublished.

[26] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[27] P. Merken and R. Vandersmissen, "Dark current and influence of target emissivity," *Photonics Imaging Technol*, 2016.