

# Detection Strategies for High Slew Rate, Low SNR Star Tracking

by

Laila Kazemi, B.Eng

Aerospace Engineering

Ryerson University, 2013

*A thesis presented to Ryerson University*

*in partial fulfilment of the requirements for the degree of*

Master of Applied Science

*in the Program of*

Aerospace Engineering

Toronto, Ontario, Canada 2015

©Laila Kazemi

# Declaration of Authorship

I, Laila Kazemi, hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I confirm that:

- I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.
- I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.
- I understand that my thesis may be made electronically available to the public.

Signed:

---

Date:

---

## **Detection Strategies for High Slew Rate, Low SNR Star Tracking**

**Laila Kazemi**

Master of Applied Science, Department of Aerospace Engineering  
RYERSON UNIVERSITY(2015)

### *Abstract*

This research is aimed to improve star tracker performance in presence of dynamic conditions. It offers an assessment of various image thresholding and centroiding algorithms to improve star tracker centroiding accuracy at moderate slew rates ( $< 10^\circ/s$ ). Star trackers generally have arc-second accuracy in stationary conditions, however their accuracy degrades as slew rate increases. In dynamic conditions, blur effects add to the challenges of star detection. This work presents an image processing algorithm for star images that preserves star tracker detection accuracy and is able to detect dim stars up to slew rates less than  $10^\circ/s$ . A number of algorithms from literature were evaluated and their performance in motion and simulations were measured. The primary performance metrics are false positive ratio, and false negative ratio of star pixels. This Work introduced a new algorithm for star acquisition in moderate slew rates that combines positive features of existing algorithms.

## *Acknowledgements*

I would like to thank my advisor, Dr. John Enright, whose patience and forethought has guided this research in the right direction.

This gratitude also extends to my fellow lab mates, both past and present, Tom, Dmitri, Geoff, Martin, and Ilija that continue to make SAIL a special place to be part of.

# Contents

<b>Declaration of Authorship</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>Abbreviations</b>	<b>x</b>
<b>Symbols</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Star Tracker Processing Framework . . . . .	2
1.2 Star Detection . . . . .	2
1.3 The ST-16 Star Tracker . . . . .	4
1.4 Research Objective . . . . .	5
1.5 Outline . . . . .	5
<b>2 Static Thresholding</b>	<b>7</b>
2.1 ST-16 Detection Routine . . . . .	8
2.2 Local Thresholding Methods . . . . .	9
2.3 Cluster-based Thresholding Methods . . . . .	10
2.3.1 Otsu Method . . . . .	10
2.3.2 Kittler-Illingworth Method . . . . .	12
2.4 Attribute-based Method . . . . .	13
<b>3 Simulation, Lab Testings and Static Thresholding Results</b>	<b>15</b>

---

3.1	Challenges of Detection Performance . . . . .	15
3.1.1	The Impact of Slew Rate on Centroiding Performance . . . . .	16
3.1.2	Stray Light . . . . .	17
3.1.3	Star Trackers' Noise Sources . . . . .	18
3.2	Simulation . . . . .	19
3.3	Simulation Testing . . . . .	20
3.4	Static Thresholding Simulation Results . . . . .	22
3.5	Static Thresholding Results Summary . . . . .	27
<b>4</b>	<b>Dynamic Thresholding</b>	<b>29</b>
4.1	Existing Algorithms . . . . .	29
4.2	New Algorithms . . . . .	31
4.3	Simulation Tests and Results . . . . .	33
4.4	Dynamic Thresholding Results Summary . . . . .	37
<b>5</b>	<b>Star Localization</b>	<b>39</b>
5.1	Centroiding Simulation Results . . . . .	39
5.1.1	Centroiding Performance for Different Stellar Magnitude . . . . .	40
5.2	Laboratory Set Up . . . . .	41
5.3	Lab Testing Validation . . . . .	43
<b>6</b>	<b>Conclusion</b>	<b>45</b>
	<b>Bibliography</b>	<b>47</b>

# List of Tables

1.1	Key Parameters of the ST-16 Star Tracker . . . . .	5
3.1	Summary of Static Algorithms Results . . . . .	27
4.1	Summay of Dynamic Algorithms Results . . . . .	38





# List of Figures

1.1	Star tracker's processing framework . . . . .	2
1.2	Simple illustration of star tracker process. . . . .	3
1.3	Processed star images from left to right: raw image, image processed by pixel level thresholding, image processed by object level thresholding. . . . .	4
1.4	Star detection steps. . . . .	4
2.1	Segmentation process. . . . .	8
2.2	Otsu threshold. . . . .	12
2.3	Minimum Error Threshold . . . . .	12
2.4	Moment Preserving Thresholding . . . . .	14
3.1	Effect of slew rate on star intensity distribution . . . . .	17
3.2	Change in background level across 100 detector columns. . . . .	18
3.3	Star images simulation sequence. . . . .	19
3.4	Sample simulated image. . . . .	20
3.5	Sample Simulated Image with Background Gradient . . . . .	21
3.6	Single star binarized. . . . .	21
3.7	FNR versus slew rate. . . . .	23
3.8	FPR versus slew rate. . . . .	23
3.9	Ratio of correctly detected stars. . . . .	24
3.10	Average false star detected. . . . .	24
3.11	FNR versus slew rate with background gradient. . . . .	25
3.12	FPR versus slew rate with background gradient. . . . .	25
3.13	Ratio of correctly detected stars. . . . .	26
3.14	Average false star detected. . . . .	26
4.1	Star Tracker . . . . .	32
4.2	False negative ratio versus slew rate. . . . .	33
4.3	False positive ratio versus slew rate. . . . .	34
4.4	Ratio of Correctly Detected Stars . . . . .	35
4.5	Average False Star Detected . . . . .	35
4.6	FNR versus Slew Rate with background gradient. . . . .	36
4.7	FPR versus Slew Rate with background gradient. . . . .	36
4.8	Ratio of Correctly Detected Stars . . . . .	37
4.9	Average False Star Detected . . . . .	37

---

5.1	Centroiding accuracy. . . . .	40
5.2	Centroiding accuracy of Local-Kittler algorithm for various stellar magnitude. . .	41
5.3	Centroiding accuracy of Shape-Based algorithm for various stellar magnitude . .	41
5.4	Laboratory set up. . . . .	42
5.5	triangular pattern in static and dynamic condition. . . . .	42
5.6	Triangular pattern length error in lab tests. . . . .	43

# Abbreviations

<b>CMOS</b>	<b>C</b> omposite- <b>M</b> etal- <b>O</b> xide- <b>S</b> emiconductor
<b>FOV</b>	<b>F</b> ield <b>O</b> f <b>V</b> iew
<b>FNR</b>	<b>F</b> alse <b>N</b> egative <b>R</b> atio
<b>FPR</b>	<b>F</b> alse <b>P</b> ositive <b>R</b> atio
<b>PSF</b>	<b>P</b> oint <b>S</b> pread <b>F</b> unction
<b>ROI</b>	<b>R</b> egion <b>O</b> f <b>I</b> nterest
<b>SAIL</b>	<b>S</b> pace <b>A</b> vionics and <b>I</b> nstrumentation <b>L</b> aboratory
<b>SNR</b>	<b>S</b> ignal-to- <b>N</b> oise <b>R</b> atio



# Symbols

$A$	Area	$m^2$
$B(x, y)$	Estimated Background Level in Image	[ADU]
$C(x, y)$	Local Contrast in Image	[ADU]
$\mathbf{D}$	Dark Signal	[ADU]
$f(x, y)$	Initial Image	[ADU]
$FDR$	False Detection Rate	[-]
$g(x, y)$	Thresholded Image	[-]
$I$	Pixel Intensity	[ADU]
$m$	Stellar Magnitude	[-]
$\bar{m}$	Local Mean	[ADU]
$t_{\text{exp}}$	Exposure Time	$s$
$T$	Threshold Level	[ADU]
$\mu$	Gaussian Mean Parameter	$ph$
$\bar{\sigma}$	Local Standard Deviation	[-]
$\sigma_{\alpha}$	Normalized Standard Deviation Curve	[-]



*I dedicate this to my friends, family and loved ones.*





# Chapter 1

## Introduction

Star trackers are attitude sensors for small satellites that provide three-axis information. Maintaining a star tracker's attitude accuracy becomes more difficult during attitude maneuvers. Small satellites are becoming the preferred platform for scientific and commercial missions and they require high performance sensors. The need for both precision and agility requires a star tracker to estimate both attitude and rate to operate at stationary and high slew rates.

From a fundamental standpoint, star trackers are autonomous camera-based sensors that use the unique patterns formed by stars to determine spacecraft attitude. These sensors acquire star field images, then identify stars in the sensed frame using an image segmentation and object localization method. After matching the detected stars with an on-board star catalog, the spacecraft's attitude is estimated. The accuracy of a Star tracker's attitude solution depends on its sensitivity to star light, field of view (FOV), internal star catalog, number of stars detected and how accurately the centroids of these stars can be determined [1]. Angular motion (i.e. slewing) of the sensor during imaging decreases the signal-to-noise ratio (SNR) of imaged stars and impairs centroid accuracy. The aim of this thesis is to analyze different star detection algorithms and evaluating better methodologies for star centroiding. Multiple+9 image processing techniques are used to develop new star detection algorithms that maintains star centroiding accuracy in dynamic conditions. It is hoped that this study of star detection algorithms in high slew rates enables more agile small spacecraft to be able to track their absolute attitude during angular maneuvers without hardware changes.

## 1.1 Star Tracker Processing Framework

This section presents a general overview of the star tracker's regular processing framework. Steps that are studied in this thesis are explained in more details. The processing framework of the star trackers can be simplified to following steps in Fig. 1.1 :

- **Image Acquisition** The camera system captures the star scene visible through the FOV of the optical system. The image intensities are combination of star light and image noise.
- **Star Detection** Images are processed to detect stars in the frame. The detection algorithms aims to estimate the location of star centroid.
- **Star Vector Computation** The centroids are then converted to body-referenced vectors using the camera model of the sensor [2].
- **Star Matching** The patterns formed by star vectors are compared to the catalog on-board to find a matching pattern. Star matching identifies stars included in the pattern.
- **Attitude Solution** Star trackers calculate attitude solution by comparing vectors in inertial frame and sensor frame. The attitude output of the star tracker can be a cosine matrix, quaternion or Euler angles.

The star detection step is explained in more details in the next section.

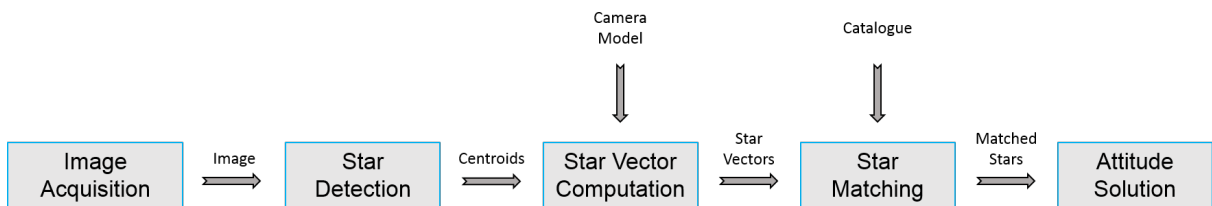


FIGURE 1.1: Star tracker's processing framework

## 1.2 Star Detection

The first part of the work in this research is focused on the star detection step of the framework. Most star detection algorithms in literature employ the two steps of thresholding and centroiding. Thresholding is the most common method for image segmentation. It is used to

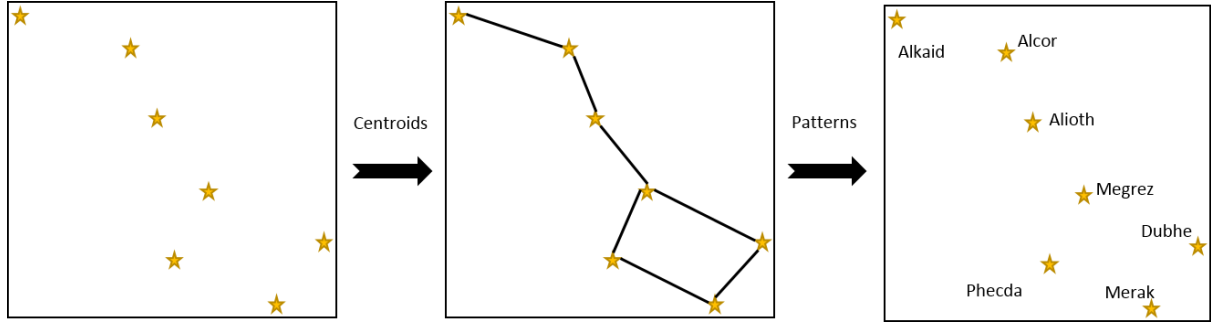


FIGURE 1.2: Simple illustration of star tracker process.

classify the pixels in a frame as background(noise) and foreground(stars). During thresholding all the pixels with intensity above a certain level (threshold) are counted as objects and other pixels are considered as background. The centroiding step, estimates the location of star centroid on the focal plane. Common thresholding algorithms define centroids as the first moment of brightness distribution so the calculated centroid lies in the middle of the imaged star.

The thresholding step could be divided into multiple parts. The thresholding step applied in this work consists of two parts, pixel-level thresholding and object-level thresholding. The pixel-level thresholding classifies the pixels into ones lit by star light and ones containing noise only. The pixels above the threshold are assigned as lit pixels. The object-level thresholding groups the lit pixels together and applies a second thresholding criteria on the grouped pixels. The object-level thresholding applies a minimum number of contiguous pixel to the object to insure grouped noise pixels are not classified as stars. Then the valid objects are tested for a minimum integrated intensity to eliminate any area of bright noise pixels. The impact of the second step can be seen in the Fig. 1.3 . The window centers around an imaged star. After the pixel level thresholding, there are still bright noise pixels remaining around the star, that object-level thresholding can often eliminate.

This thesis investigated several pixel-level thresholding algorithms within the star detection process. Studying thresholding literature , a number of promising cluster-based, attribute-based, and local thresholding methods are synthesized and analyzed. Based on the results, new algorithms that combine advantages of existing algorithm are developed. The new algorithms are then compared to other methods in the literature applied to dynamic star images. Throughout this analysis, object-level thresholding and the centroiding algorithm are kept constant.

These detection algorithms are tested in different scenarios. Initially their detection ability are tested with angular velocities from 0 °/s to 10 °/s. Then same algorithms are tested with simulated star images that have uneven background level, this case studies the effect of bright

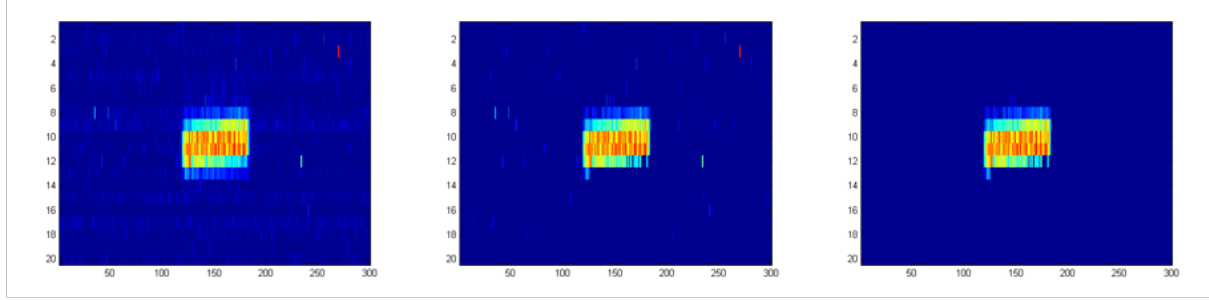


FIGURE 1.3: Processed star images from left to right: raw image, image processed by pixel level thresholding, image processed by object level thresholding.

bodies such as plants close to star tracker FOV. The last scenario looks at the centroiding accuracy of the dim stars in dynamic conditions (low SNR).

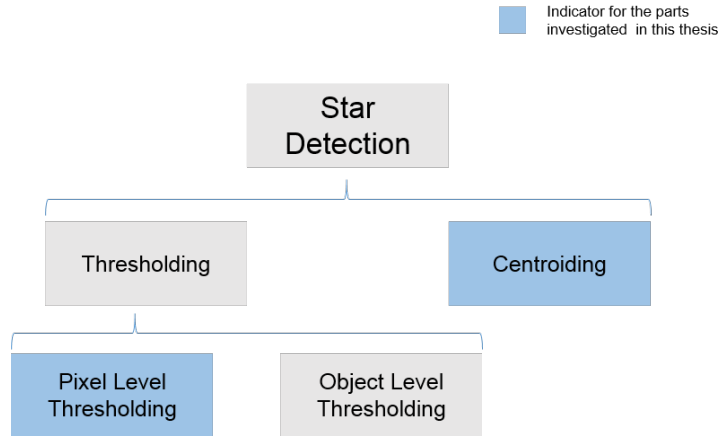


FIGURE 1.4: Star detection steps.

### 1.3 The ST-16 Star Tracker

The baseline sensor used for this work is ST-16 star tracker from Sinclair Interplanetary. The ST-16 star tracker is a relatively new nanosatellite-class star tracker that became available in 2011 [6]. Some key specifications are listed in table 1.1. In this work, ST-16's lens diameter, lens F#, detector pixel size and field of view were used in image simulations. The ST-16 star catalog contains the stars of a visual magnitude of 5.75 or brighter, hence the detection threshold at static condition is set to 5.75. The ratio of the focal length to the aperture diameter is the F# of the lens. This value represents the optical collecting power of the system.

TABLE 1.1: Key Parameters of the ST-16 Star Tracker

<b>Accuracy</b>	0.01° with 85% confidence
<b>Size</b>	59 × 56 × 31.5 mm
<b>Mass</b>	≈90 g
<b>Field of View</b>	7.5° half axis
<b>Exposure Time</b>	100 ms
<b>Detection Threshold</b>	5.75
<b>Catalog</b>	3746 stars
<b>Availability</b>	> 99.9%
<b>Lens Diameter</b>	D = 12 mm
<b>Lens F#</b>	1.2
<b>Detector Pixel Size</b>	2.2 μm

Although the test images are simulated using optical and detector properties of the ST-16, we expect that the approach is applicable to any star tracker with only minor modifications.

## 1.4 Research Objective

The objective of this thesis is to improve the detection algorithm of a star tracker for operating at high slew rates. Maintaining accurate attitude estimates while slewing complicates the regular star tracker process. This research concentrates on image processing and star tracker redesign. This work presents an image processing algorithm for star images that preserves star tracker detection accuracy and is able to detect dim stars up to slew rates less than 10s. Most of star detection algorithms in literature are designed to work in stationary conditions. Multiple thresholding and centroiding methods have been considered and two new window-based algorithms that dynamically size their region of interest (ROI) based on the spacecraft motion projection are developed. These methods combine local and histogram-based thresholding and are robust to both motion and stray light. This algorithm increases the star detection accuracy in moderate slew rates and is robust to stray light. The star tracker's localization is then improved by looking at different localization methods such as derivative, Gaussian and moment based.

## 1.5 Outline

In chapter 2-3, existing star detection algorithms are studied. Chapter 2 discusses existing reviews of thresholding algorithms, and offers an introduction to promising algorithms that are studied in this work. This section also summarizes the base-line sensor star detection

---

algorithm. We present an overview of star simulation environment in Chapter 3. The main part of that chapter focuses on the performances of the methodologies summarized in chapter 2. Chapter 4 studies algorithms intended for high slew rate star tracking. It discusses the outline of three new algorithms developed. This chapter tests the developed algorithms against some of the existing star detection methods to compare their performances. Chapter 5 highlights the detection improvements of the suggested algorithms by showing their centroiding accuracy performance at high slew rates. This chapter also examines other localization methodologies and tests each algorithm for stars of different visual magnitude. Concluding chapter summarizes the findings of the research and suggest future works and improvements.

## Chapter 2

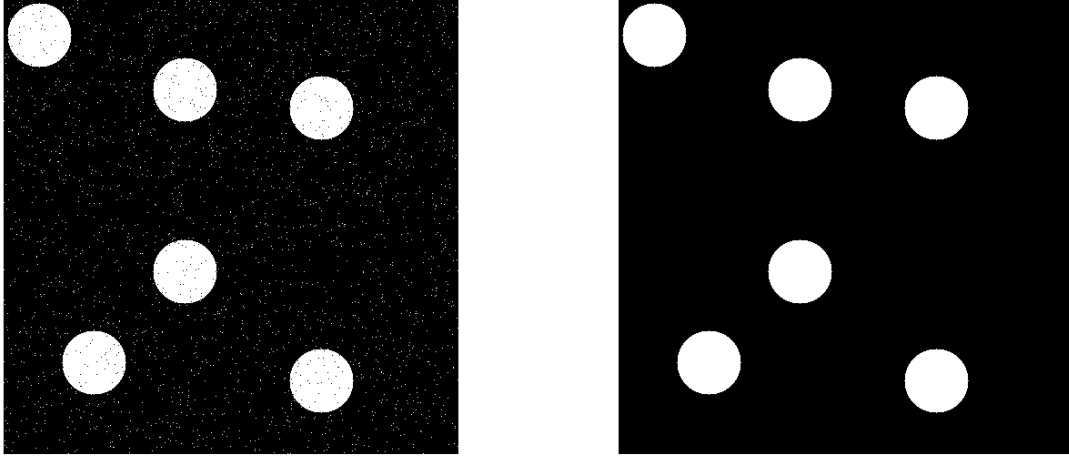
# Static Thresholding

This chapter studies several static thresholding algorithms in the literatures. Static thresholding algorithms don't rely on previous knowledge of the spacecraft rate. These methods make no assumption about expected shapes of PSF or effects of motion. Although these algorithms are not specially designed for motion images, they may be usable for dynamic images. Image segmentation is the process of classifying the pixels into objects and background. Segmentation has application in many computer vision projects and problems, including feature detection, three dimensional reconstructions, and stereo calibrations [7]. The most common method of segmentation is thresholding and achieves its purpose by setting pixels above a threshold value into the object and the rest as background. If  $g(x, y)$  is the processed version of  $f(x, y)$  then:

$$g(x, y) = \left\{ \begin{array}{ll} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{array} \right\} \quad (2.1)$$

Thresholding can be local or global. A global threshold method considers all the pixels in an image to calculate one threshold value for all the pixels in the image. Local thresholding methods consider the local attributes and property of the image to calculate a threshold for each pixel relative to its surrounding pixels. An ideal thresholding algorithm for the star tracker must be robust to background noise, stray light and blur effects from sensor slew. The main sources of background noise in star trackers are the read noise and dark current noise. Stray light from bright bodies in space (Sun, Earth, Moon) cause image gradients [4]. To improve star tracker attitude solution, star detection should be maximized in presence of motion and stray light.

Most thresholding static algorithms can be grouped into three categories: clustering-based, attribute-based and local methods. Clustering-based algorithms classify pixels into two Gaussian



(A) Initial Image with Noise

(B) Segmented Image

FIGURE 2.1: Segmentation process.

distributions. Attribute-based algorithms look for a similarity between gray-level and binarized image [8]. Local methods adjust the threshold based on local image characteristics. To narrow down the options among thresholding algorithms, previous literature review like Tier and Jain [7] and Sezgin and Sankur[8] were consulted. To evaluate the performance of each category, we selected five algorithms: local thresholding Niblack and Bernsen, Cluster-based thresholding Otsu and Kittler-Illingworth, and attribute-based thresholding Tsai method. In thresholding survey papers Tier and Jain [7] and Sezgin and Sankur [8], these five algorithms were found to be superior in their respective categories. This section summarizes existing thresholding algorithms and evaluates them based on their ability to detect stars. A testing procedure have been developed to analyze the effectiveness of various thresholding methods on star images.

## 2.1 ST-16 Detection Routine

The detection routine onboard the ST-16 utilizes an adaptive thresholding algorithm to detect stars. This algorithm uses a local averaging thresholding to estimate the background level. Then the pixels classified as foreground will be grouped together and each group will be tested for an integrated threshold test. Finally ST-16 implements a moment centroiding technique. This routine can be summarized by three basic operations:

- **Local Averaging Filter:** An averaging filter is used to estimate the average background illumination,  $B(x, y)$ , for each pixel. Using a  $m \times n$ -sized window, centered on pixel  $(x, y)$ ,



the average background illumination for each pixel calculated as:

$$B(x, y) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N I(x, y) \quad (2.2)$$

where  $I(x, y)$  is the image intensity, and  $(x, y)$  are the row and column image coordinates. Default ST-16 processing utilizes a window size of  $M = 1, N = 129$  pixels.

- **Lit Pixel.** After defining the average background illumination, the algorithm steps through each pixel and compares it against a static threshold  $T$ . If the  $I(x, y) - B(x, y)$  is larger than  $T$ , the pixel is labeled as 'lit', implying that it is lit by star light, as opposed to just background illumination. Pixels with intensity values below  $T$  are discarded as background. Default ST-16 processing utilizes a value of  $T = 80$  detector counts.
- **Contiguous Pixel Threshold:** Lit pixels are then grouped together into objects using 4-connectivity. Blobs with at least 6 contiguous pixels are considered possible stars. The minimum integrated intensity for the objects to be counted as star is 1000 detector counts.

Afterward the centroid of each star can be calculated as the first moment of illumination [9]. The centroid  $(x_c, y_c)$  is given as:

$$x_c = \frac{1}{\sum_x \sum_y f(x, y)} \sum_x \sum_y x f(x, y) \quad (2.3)$$

$$y_c = \frac{1}{\sum_x \sum_y f(x, y)} \sum_x \sum_y y f(x, y) \quad (2.4)$$

## 2.2 Local Thresholding Methods

The local thresholding algorithms assign a unique threshold value to a region or each individual pixel based on the characteristics of their neighboring pixels. The size of the window in the local method are significant as they must be small enough to preserve local characteristics but large enough to be resilient to noise. Two of the most commonly used local adaptive thresholding methods are Niblack's [11] and Bernsen's [10] methods. Niblack proposes a thresholding method based on the local mean  $\bar{m}$  and the standard deviation  $\bar{\sigma}$  of the image,  $I(x, y)$ . Niblack's threshold value  $T$  for pixel  $(x, y)$  is calculated as:

$$T(x, y) = \bar{m}(x, y) + k \bar{\sigma}(x, y) \quad (2.5)$$

where  $\bar{m}$  and  $\bar{\sigma}$  are computed across a window of size  $w_r \times w_c$  and  $k$  is a design parameter that determines how much of the surrounding pixels to include in the threshold calculation [11].

Bernsen's method makes use of the local image contrast to determine a threshold value. The local image contrast  $C(x, y)$  is calculated based on a window  $w_r \times w_c$  as:

$$C(x, y) = I_{high} - I_{low} \quad (2.6)$$

where  $I_{high}$  and  $I_{low}$  are the highest and the lowest intensity value in the local window. If  $C(x, y) < T_c$ , a predetermined contrast threshold, all the values inside of the window are considered background. If  $C(x, y) > T_c$ , the local threshold is defined as:

$$T(x, y) = \frac{1}{2}(I_{high} + I_{low}) \quad (2.7)$$

## 2.3 Cluster-based Thresholding Methods

Cluster-based methods calculate the optimal threshold based on histogram analysis of the image intensity. The histogram of images are assumed to be bimodal in these cases; each cluster represents the objects or the background pixels. Many of the cluster based algorithms takes one of the two approach; they either set the optimal threshold at the local minimum that separates the two clusters in intensity histogram or the algorithm fits two Gaussian distributions to the histogram data. There are a few challenges that cluster-based methods face; due to image noise there may be multiple local minimum and maximum, also the distributions may not produce two distinctive modes and overlap. The cluster based methodologies examined in this section are Otsu [12] and Kittler-Illingworth [13] methods; these methods are applied globally to the image.

### 2.3.1 Otsu Method

Otsu's method attempts to separate the image's intensity histogram into two distinctive modes: background and objects. This is achieved by using a threshold,  $T$ , that is optimized to minimize the within-class variance of intensities. Minimizing within class variances is the same as maximizing between-class variances. As a result, the optimal threshold value corresponds to the greatest separation of gray levels. Otsu's method is the most commonly used thresholding algorithm in the literature. This algorithm is based on the following assumptions [12]:

- The histogram of the image is bimodal
- Background illumination is uniform throughout the image
- Stationary objects and scene

To solve for the Otsu threshold the expression for the within class variance is first derived. The Otsu threshold  $T$  divides the pixels in the image into two class of background  $C_1$  and object  $C_2$ . Pixels with intensity in range of  $[1, \dots, k]$  are place in  $C_1$  and pixels in range of  $[k + 1, \dots, L]$  are placed in  $C_2$ . So the weighted within-class variance  $\sigma_w$  is computed as:

$$\sigma_w^2(T) = q_1(T) \sigma_1^2(T) + q_2(T) \sigma_2^2(T) \quad (2.8)$$

where the class probabilities  $q_1, q_2$  and the class means  $\mu_1, \mu_2$  are computed from:

$$q_1(T) = \sum_{i=1}^T P(i) \quad (2.9)$$

$$q_2(T) = \sum_{i=k+1}^L P(i) \quad (2.10)$$

$$\mu_1(T) = \sum_{i=1}^T \frac{iP(i)}{q_1(t)} \quad (2.11)$$

$$\mu_2(T) = \sum_{i=t+1}^L \frac{iP(i)}{q_2(t)} \quad (2.12)$$

The total variance of the intensity can be expressed as sum of the within-class variance and between-class variance. The total is constant and independent of the threshold  $T$ . So the between class variance can be expressed as Eq.2.13, and  $T$  can be calculated from optimization problem that maximizes  $\mu_B$ . The between class variance can be calculated from:

$$\sigma_B^2(t) = \sigma^2 - \sigma_w^2(t) \quad (2.13)$$

Fig. 2.2 represents the histogram of a window containing one star. The distribution approximated by Otsu algorithm and the threshold value in between valley of those distribution is marked on the figure.

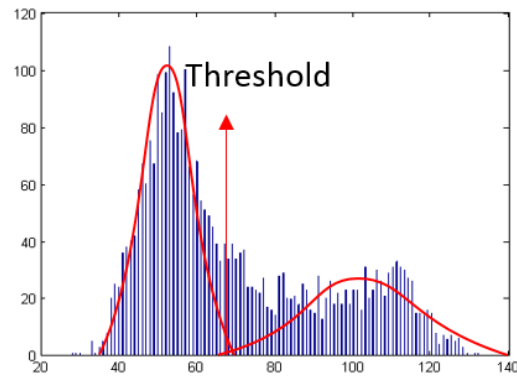


FIGURE 2.2: Otsu threshold.

### 2.3.2 Kittler-Illingworth Method

In Kittler-Illingworth's model, the threshold value,  $T$ , divides the gray histogram of an image into approximation of two probability density function each representing either population of background or objects. This method assumes that the object and background's pixel gray level values are normally distributed. The minimum error  $T$ , or optimal threshold, minimizes the error in fitting the Gaussian models into the histogram data and the overlap between the two density functions as it seen in Fig. 2.3.

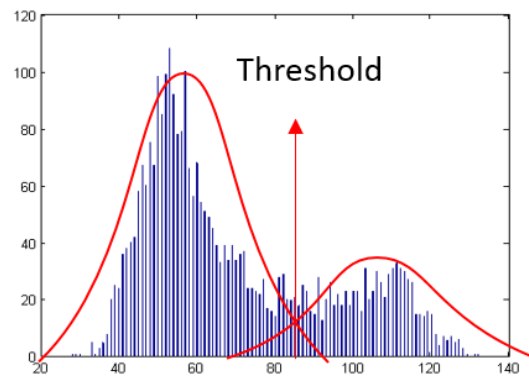


FIGURE 2.3: Minimum Error Threshold

The conditional probability  $e$  of gray level  $i$  being replaced in the image correctly is given by:

$$e(i, t) = P(i | n, t) \frac{P_n(t)}{q_n(t)} \quad (2.14)$$

$$n = \begin{cases} 1 & \text{if } i \leq T \\ 2 & \text{if } i > T \end{cases} \quad (2.15)$$

Since  $P(n)$  is independent of both  $n$  and  $T$ , the denominator in 2.14 will be ignored. Using the numerator of Eq. 2.14 and taking logarithm of it; an alternative index of correct classification performance is defined as:

$$\epsilon(i, t) = \left[ \frac{i - \mu_n(t)}{\sigma_n} \right]^2 + 2 \log \sigma_n^2(t) - 2 \log P_n^2(t) \quad (2.16)$$

and finally the criterion function is defined as Eq. 2.17 and the  $t$  that minimizes this expression is the ideal threshold.

$$J(t) = \sum_i P(i) \epsilon(i, t) \quad (2.17)$$

The value of threshold  $T$  yielding the lowest value of criterion  $J(t)$  will give the best fit model and therefore the minimum error threshold.

## 2.4 Attribute-based Method

The attribute-based thresholding method considered in this research is the Tsai's moment preserving method. In this method the gray level moments of the image is computed, and threshold level is calculated in a way that the processed image's first three moments are equivalent to the original image's moment[14].

The image moments  $m_n$  and binary image moments  $b_n$  are defined, respectively, as:

$$m_n = \sum_{i=0}^L P(i) i^n \quad (2.18)$$

$$b_n = P_o z_o^n + P_b z_b^n \quad (2.19)$$

Here  $z_o$  and  $z_b$  represents the value that intensities in the image that replace pixels below the threshold and pixels above the threshold.  $z_o$  and  $z_b$  are the fractions of the pixels classified as objects and background. The solution to this system is offered in [14].

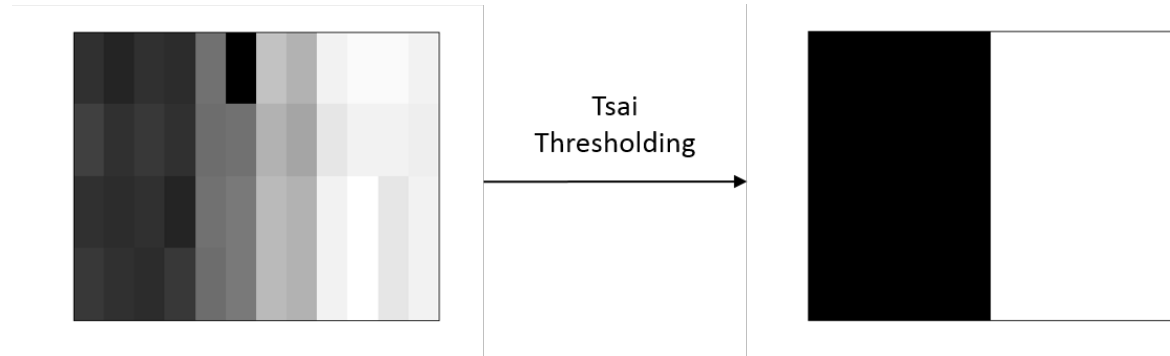


FIGURE 2.4: Moment Preserving Thresholding

## Chapter 3

# Simulation, Lab Testings and Static Thresholding Results

This chapter evaluates the performance of the static algorithms. The challenges that star detection algorithms face are discussed here. Then the Simulation environment used to synthesize the test star images are explained and the layout for the testing methodologies and parameters are discussed. For initial evaluation of each thresholding method a set of 50 star scenes were simulated. Each scene was examined in rates between 0-10 °/s. After discussion of preliminary results from the thresholding methods, the superior algorithms were selected and further testes for centroiding accuracy. The results for centroiding accuracy is then verified using laboratory testings.

### 3.1 Challenges of Detection Performance

There are various factors that can impair the star tracker's to accuracy estimate the attitude of the star tracker. The star tracker accuracy is dependent on how many stars the sensor is able to detect withing its field of view (FOV) and how accurately their centroid on image frame is calculated. In this section factors that are effecting detection performance of the star tracker are studied. Detection performance of star tracker is set as ability of the algorithms to detect all the available stars in FOV, while rejecting non-star pixels. Star detection algorithms should also be able to localize the star centroids with sub-pixel accuracy. Three main problems with threshold-based star detection routine are sensor noise from various sources, spacecraft dynamic conditions, and inconsistence background light level.

### 3.1.1 The Impact of Slew Rate on Centroiding Performance

Sensor slew is due to angular motion of the spacecraft during star tracker exposure. The primary objective of this work is investigating star detection robustness to slew rate.

The point spread function (PSF) is the response of an imaging system to a point source of light. In static conditions, a common model for the intensity of stars in the image is a symmetric Gaussian distribution [3]. In dynamic conditions the star centroid moves on the image plane during the exposure. In the presence of the spacecraft's angular rate, the Star PSF is elongated on the direction of the motion (blur effect). To model star's track on the image plane, we use the model by Dzamba and Enright [3]. The body-referenced star vectors at the beginning and end of an image exposure are defined as  $\mathbf{b}$  and  $\mathbf{b}'$ , respectively. The motion of the spacecraft can be described through an angle  $\Delta t_e \boldsymbol{\omega}$ , where  $\Delta t_e$  is the exposure time and  $\boldsymbol{\omega}$  is the spacecraft angular velocity.

$$\mathbf{b}' = (\mathbf{I}_{3 \times 3} - \Delta t_e \boldsymbol{\omega}^\times) \mathbf{b} \quad (3.1)$$

Star streak length  $\delta \mathbf{b}$  in the image is the:

$$\delta \mathbf{b} = \mathbf{b}' - \mathbf{b} = -\Delta t_e [\boldsymbol{\omega}^\times] \mathbf{b} \quad (3.2)$$

In Eq.3.1,  $\boldsymbol{\omega}^\times$  represents the skew-symmetric matrix of the vector  $\boldsymbol{\omega}$ . Where  $\boldsymbol{\omega}$  is:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.3)$$

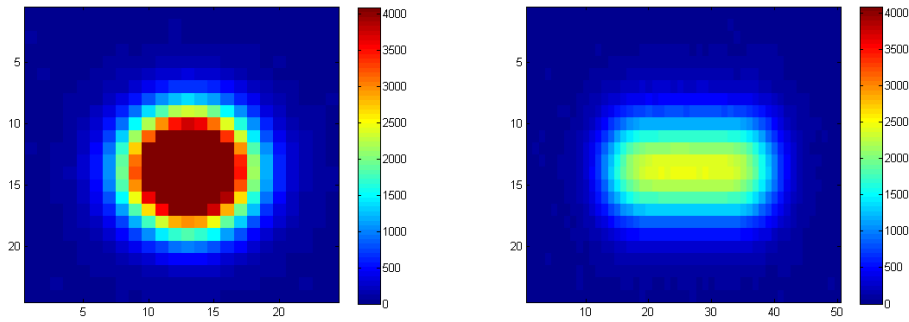
We then convert the vector difference  $\delta \mathbf{b}$  in Eq.3.2 to a displacement of the centroid across the focal plane,  $\Delta \mathbf{b}$ :

$$\Delta \mathbf{b} = \frac{f \Delta t_e}{\gamma b_z} \begin{bmatrix} \omega_z b_y - \omega_y b_z \\ \omega_x b_z - \omega_z b_x \end{bmatrix} \quad (3.4)$$



where  $f$  is the focal length of the star tracker optics,  $\gamma$  is the detector pixel size,  $t_e$  is the exposure time,  $\omega$  is the satellite angular velocity and  $\mathbf{b}$  is the star vector in the body frame at the beginning of the exposure.

The overall light collected from a star during exposure is the same for both static and dynamic conditions, however in dynamic condition this light spread across larger number of pixels compare to static conditions as it can be seen in Fig. 3.1. While the same star is visibly brighter compare to its background in static conditions, it is faint in dynamic condition and its intensity is close to background level.



(A) Star intensity in static conditions. (B) Star intensity in dynamic conditions.

FIGURE 3.1: Effect of slew rate on star intensity distribution

Blur effects as modeled by Eq.3.4 cause the light from a star to be smeared across a larger region of the image detector compare to static conditions. which decreases the signal-to-noise ratio (SNR). In images with low SNR, the signals from dimmer stars are close to the background noise level, and they are difficult to detect or localize. Blur effects impair both centroiding and thresholding performance. As the slew rates increases, star PSF' become longer and dimmer. As the stars grow dimmer, centroid noise tends to increase until stars' centroids can no longer be reliably detected.

### 3.1.2 Stray Light

Stray light is another challenge for star detection algorithms. Any undesirable light within the image and FOV is referred to as stray light. Celestial objects such as Sun, Moon, Earth and other planets are main sources of stray light. The light from these objects causes the background intensity variation in form of a gradient. When background intensity is not constant, the detection scheme can work properly on the darker side, while it will collect the well of the pixels

lit by stray light as false stars. Baffles are a stray light shielding element for star trackers. It is common to see an increase in background level at the baffle edge[4].

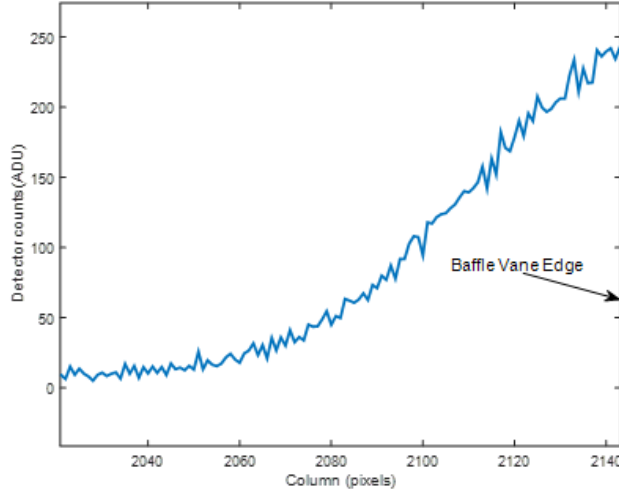


FIGURE 3.2: Change in background level across 100 detector columns.

### 3.1.3 Star Trackers' Noise Sources

The dominant noise sources in star trackers' images could be listed as dark current noise, shot noise and read noise. Dark response is the detector output while it is not illuminated which is the collection of thermal electrons in pixel wells. Dark response is usually consists of two parts, the bulk response part and random variation. The bulk response level could be averaged by reading pixel levels on a dark frame. The average then can be deducted from an illuminated frame to remove most of the dark response [5]. The random variation in the dark response are called dark noise. Shot noise is described as random variation in the observed amount of photoelectrons, due to the discrete, quantum nature of light. Shot noise is more dominant when collecting a relatively small number of photons. Its effect can be reduced by collecting more photons though with a longer exposure [1]. Read noise is essentially a summation of typical image detector noise sources. A pixel read out of the detector is a tiny voltage, on the order of microvolts per electron. The camera's electronics then pass that voltage to an Analog to Digital Converter (ADC) to be converted into a digital pixel value. During this process additional noise are added to the image. Read noise is generally modeled as a zero-mean, normally distributed random variable  $3.5e^-$ , where  $e^-$  is the electron count. In this case the electron count  $e^-$  is approximated as 7 detector count.

## 3.2 Simulation

To evaluate the performance of the algorithms studied, a set of simulated star scenes are examined. The existing star generation code in SAIL was used to produce the star images. The common part used for each frame in the scene generation is the sensor properties from Table 1.1 and the star catalog. The angular rate and attitude are changed for each scene. The attitude for each star scene are generated randomly. This section summarizes the function of the software:

- **Scene Generation** Using the baseline sensor properties and SKY2000 star catalog, centroids and brightness of the stars in field of view (FOV) are found at different attitude and angular rates.
- **PSF Model** The shape and the intensity profile of each star was simulated using the method presented in Dzamba and Enright [3].
- **Star Image** The ideal signal is then combined with two typical imaging noise sources: shot noise and detector read noise.

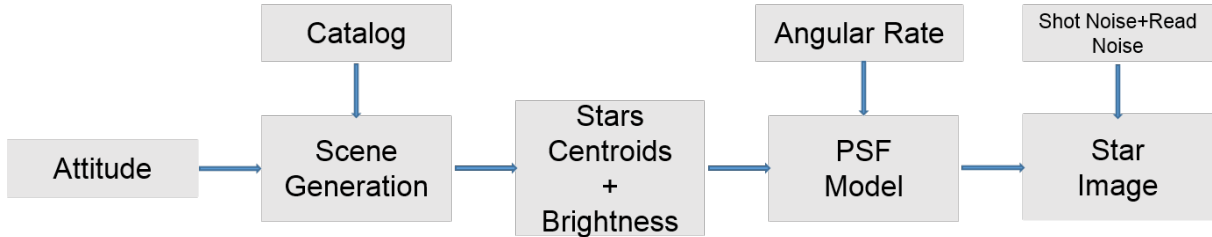


FIGURE 3.3: Star images simulation sequence.

Shot noise is typically modeled as a Poisson distribution with variance of 25 detector count out of 4095. Read noise is essentially a summation of typical image detector noise sources and is generally modeled as a zero-mean, normally distributed random variable  $3.5e^-$ .  $e^-$  is the electron count and in this case the electron count as approximately 7 detector count.

Two sets of star scenes were simulated: one with background gradient variation, and one without to test robustness of algorithms to stray light. Variation in background light, is one of the major causes of deficient centroiding accuracy. In Figure 3.2, the background level across 100 column of the detector is plotted. The background intensity has rapidly increased and peaked at 250 detector count, which is higher than most of the dim star peak intensity.

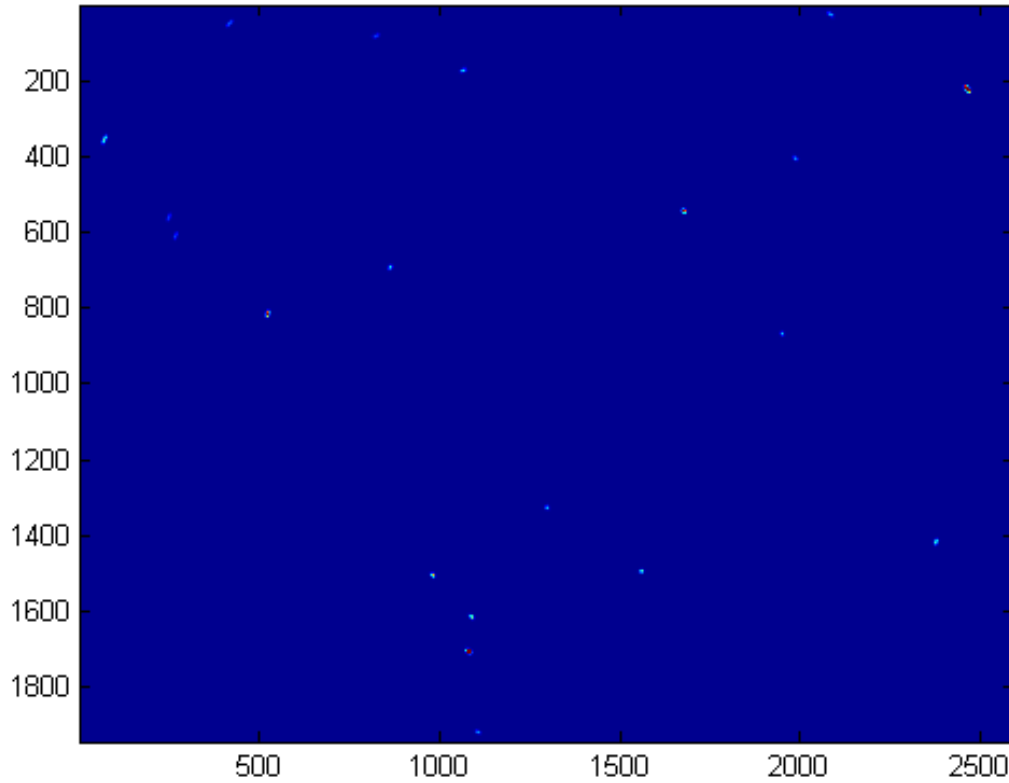


FIGURE 3.4: Sample simulated image.

### 3.3 Simulation Testing

The proposed thresholding algorithms are tested in multiple steps. The thresholding algorithms discussed in this thesis are initially examined by their ability to separate lit pixels from the background noise. This ability is then tested in presence of uneven background level. The test image prior to addition of noise and background gradient is binarized such that any nonzero pixel is set to 1, representing lit pixels. This image becomes the ideal binarized image and thresholded images will be compared to it.

The metrics that will be used to evaluate the thresholding techniques are false positive ratio (FPR) and false negative ratio (FNR). FPR is the ratio of incorrectly detected lit pixels to number of actual lit pixels in the image, and false negative rate is the ratio of lit pixels missed to be detected to number of actual lit pixels in the image. If the ideal binarized image is subtracted from the processed image, the count of pixels with value of one are false positive detection while the pixels with value of -1 are false negative detection. The second step of

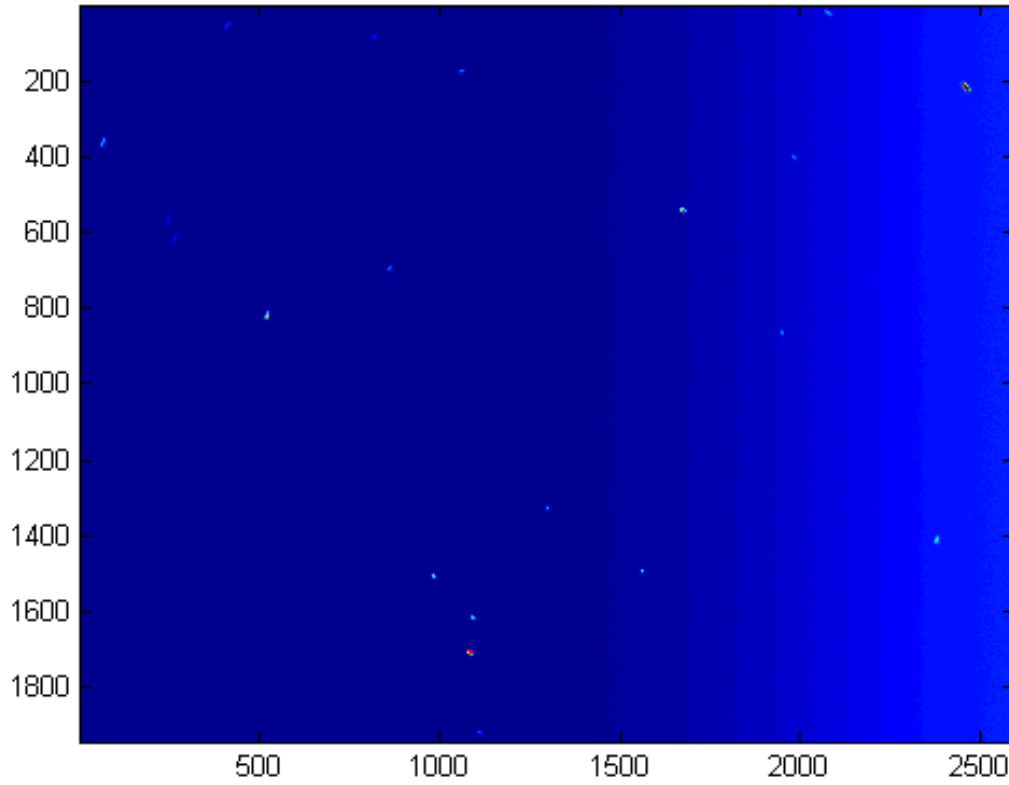


FIGURE 3.5: Sample Simulated Image with Background Gradient

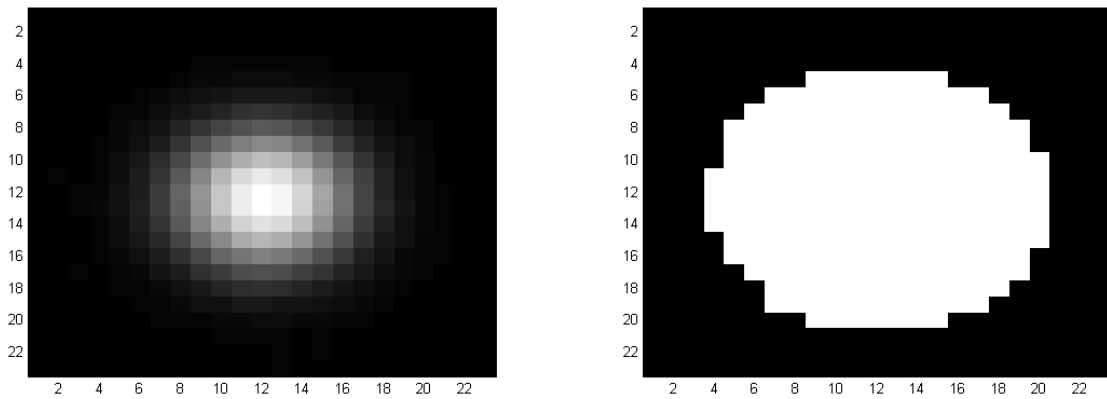


FIGURE 3.6: Single star binarized.

evaluation groups the pixels together and applies the contiguous pixel threshold from ST-16 detection algorithms. From then it determines the correctly detected stars, missed stars, and the number of bright noise spots misidentified as stars. An ideal segmentation algorithm should

result in no to low number of false stars and high star detection rate. To solve for the attitude problem from star tracker data, most of the algorithms available in literature require four stars for matching step.

The best performing algorithms in previous tests are then tested for their centroiding accuracy using first moment centroiding introduced in chapter 2. The centroid of the correctly detected stars are calculated and then the results are compared to the centroids from the ideal image. Finally the centroiding accuracy of these algorithms versus stars of different brightness is studied. It is important for the detection algorithm to estimate the centroid of the deem star as accurately as the brighter stars. The results are verified through lab testing in chapter 5 and discussion of further verification using field tests is presented in chapter 6.

### 3.4 Static Thresholding Simulation Results

To compare the performance of each thresholding algorithm, their ability to separate the background pixels from lit pixels in simulated star scenes were examined. Scenes were simulated for randomly generated orientations while slew rates in ranged of 0-10  $^{\circ}/s$  in steps of 0.5  $^{\circ}/s$  were considered. For all of the local methods, a  $1 \times 129$ -sized window was implemented for consistency with baseline sensor ST-16.

The three adaptive thresholding algorithms: Win, Niblack, Bernsen were tested in this paper. The results are seen in Fig. 3.7 and Fig. 3.8. Please note the difference in Y axis scale. Win thresholding has FNR of 35-40%, and 0% FPR. The FNR drops slightly as the slew rate increases, since as the star smears across a greater number of pixels and becomes dimmer, the mean value for the window also drops and more pixels are marked as lit. Niblack's method has a high FNR of 70%. Niblack's threshold depends on the local image mean  $\bar{m}$  and  $\bar{\sigma}$  standard deviation. Since  $\bar{m}$  and  $\bar{\sigma}$  are relatively high in star regions, many lit pixels were dropped. In the background region,  $\bar{m}$  and  $\bar{\sigma}$  are relatively low, and therefore the threshold is lower in these regions causing the FPR to increase. The FPR for Niblack stays negligible. Bernsen's method resulted in a FNR as high as 70% too. Bernsen's method depends on the local contrast, which could be relatively high in a window containing star, hence the high pixel rejection rate. Bernsen's method resulted in overall negligible FPR.

The global methods considered in this section are the two cluster-based algorithms of Otsu and Kittler-Illingworth, and the moment-based method describe by Tsai. Otsu's method does not present satisfactory results. Otsu's method assumes the histogram of the images are almost bimodal distribution, in other words number of pixels in background and objects are almost

equal. In star images, object areas are very small compared to background, so global Otsu thresholding did not perform well. Likewise the performance of Tsai's method was also limited by small object areas, and large variance of the objects. At higher rate when the contrasts between stars and background level are negligible and Tsai algorithm fails to detect any lit pixels. Kittler-Illingworth method performed the best with a FNR below 10%, and a negligible FPR.

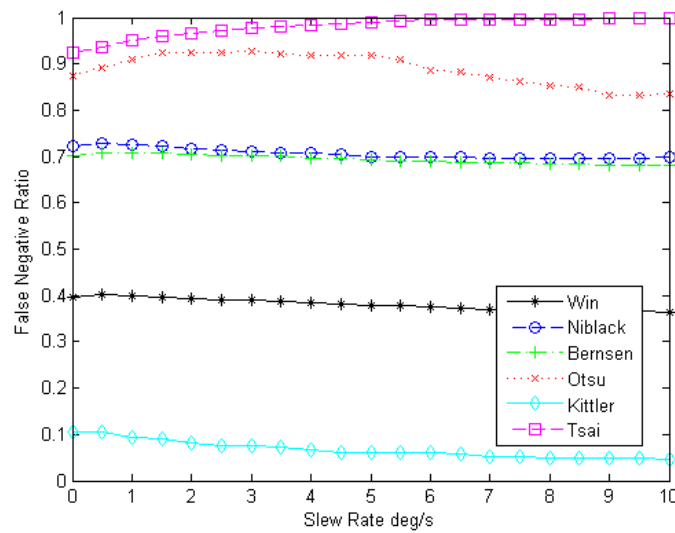


FIGURE 3.7: FNR versus slew rate.

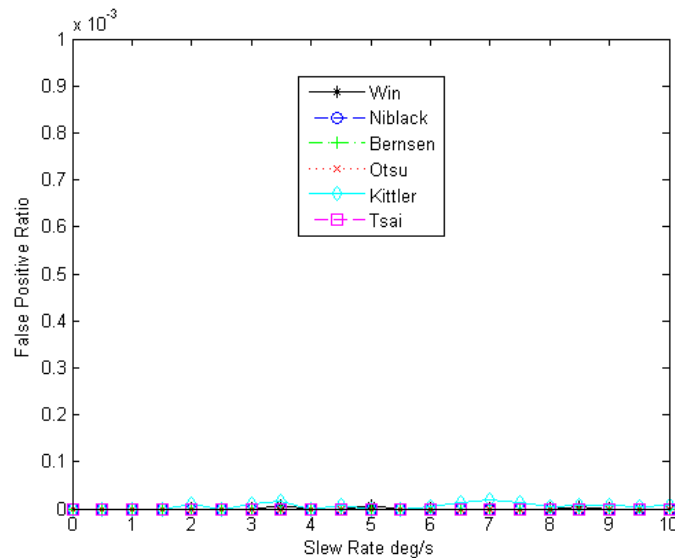


FIGURE 3.8: FPR versus slew rate.

The Fig. 3.9 presents the ratio of correctly detected stars to all the stars in the scene plotted versus rate. A star is marked as correctly detected if the calculated centroid error is less than

0.2 pixels. While most of the algorithms are capable of detecting majority of the stars, the detection rate drops fast with rate for Otsu and Tsai method. The Fig. 3.10 is the average number of incorrectly detected objects in star scenes versus angular rate. In each rate for each different frame number of the false detected objects are obtained then averaged for each rate.

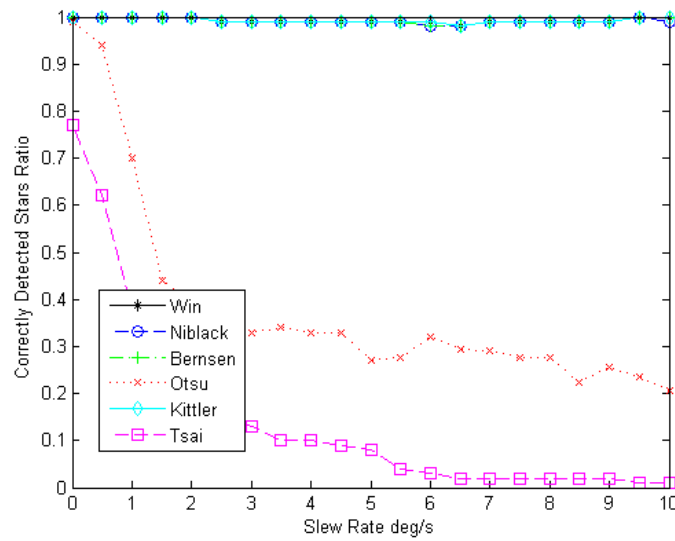


FIGURE 3.9: Ratio of correctly detected stars.

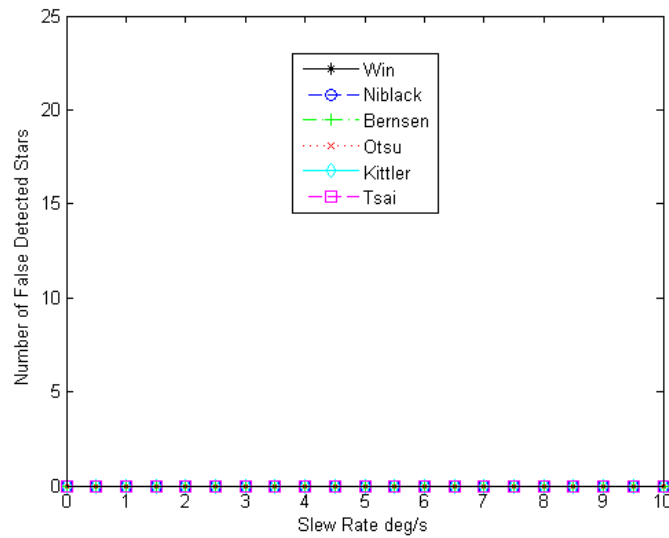


FIGURE 3.10: Average false star detected.

In second case of tests, a gradient was added to the image background, simulating the presence of a bright body (i.e. Earth, Sun, Moon) near the sensor field of view (FOV). This test was used to examine the robustness of each algorithm to stray light. The results of this test are



shown in Fig 3.11 and Fig 3.12. The performance of all three local thresholding algorithms was not affected by background gradient. Otsu's and Tsai methods were again unsuccessful in segmenting the images effectively. Unlike the previous test without the background gradient, Kittler-illingworth's method was unable to adjust to the non-uniform background level. Since the Kittler-illingworth method tries to fit the data into two Gaussian distributions pertaining to the background and image objects, the addition of gradient to background limits this process.

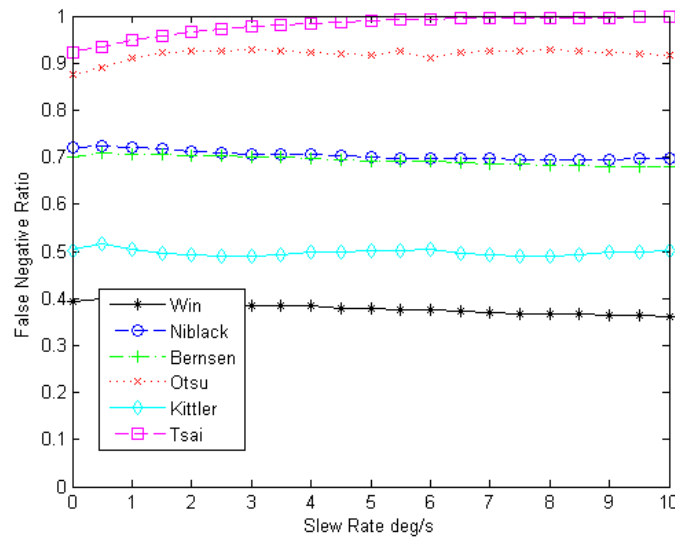


FIGURE 3.11: FNR versus slew rate with background gradient.

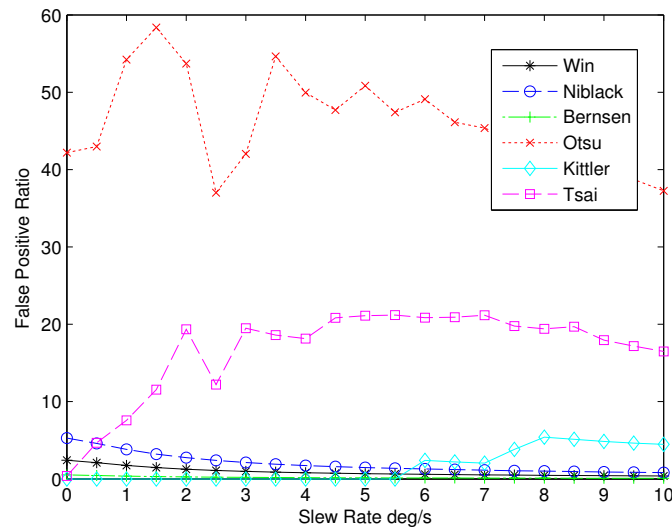


FIGURE 3.12: FPR versus slew rate with background gradient.

The results in Fig. 3.13 and Fig. 3.14 are also for the case where the background light is not constant. The Fig. 3.13 below present the ratio of correctly detected stars to all the stars in the scene plotted versus rate. The Fig. 3.14 is the average number of incorrectly detected objects in star scenes versus angular rate. In each rate for each different frame number of the false detected objects are obtained then averaged for each rate. The local algorithms performances are not affected by stray light in terms of star detection, while global algorithms resulted in detecting a number of false stars in the image areas illuminated by stray light.

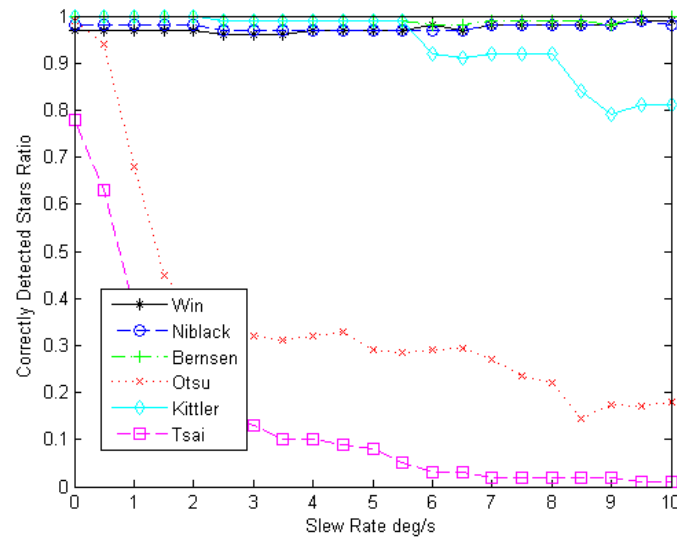


FIGURE 3.13: Ratio of correctly detected stars.

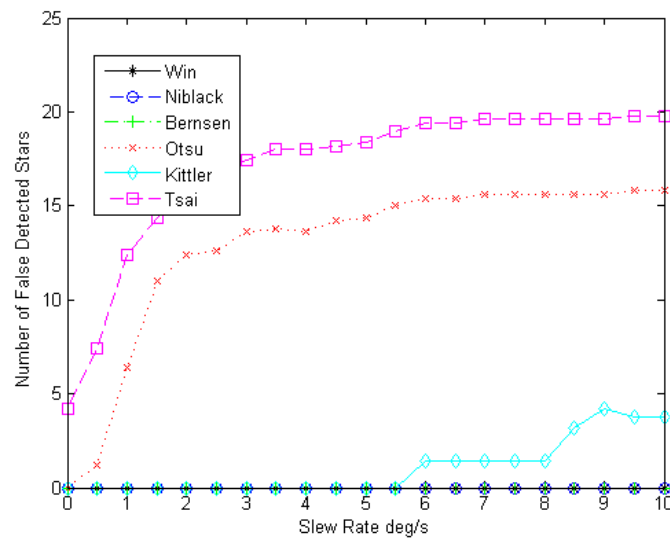


FIGURE 3.14: Average false star detected.

TABLE 3.1: Summary of Static Algorithms Results

Method	Performance in Slew	Performance in Slew + Stray Light	Summary
<b>Window</b>	Low FPR, Modest FNR	Low FPR, Modest FNR	Not Sensitive stray light and slew
<b>Niblack</b>	Low FPR, High FNR	Low FPR, High FNR	Sensitive to Stray Light
<b>Bernsen</b>	Low FPR, High FNR	Low FPR, High FNR	Does Not detect dim stars due to low contrast
<b>Otsu</b>	Low FPR, High FNR	High FPR, High FNR	Not acceptable due to High FNR
<b>Kittler-Illingworth</b>	Low FPR, Low FNR	Modest FPR, Modest FNR	Not Sensitive to slew
<b>Tsai</b>	Low FPR, High FNR	High FPR, High FNR	Does Not detect dim stars due to low contrast

Among algorithms discussed in this section, the Kittler-Illingworth algorithm performed the best with negligible FPR and low FNR; however this algorithm was not robust to stray light. Win thresholding was the most robust to background gradient; the main drawback was the FNR.

### 3.5 Static Thresholding Results Summary

In this section, the static thresholding algorithms were testing using synthetic star images. The algorithms are tested based on their ability to identify lit pixels. Two conditions were tested in this section, the performance in presence of motion and stray light. The results are summarized in Table 3.1. Considering these results, a new routine combining Win and Kittler algorithms was developed to be robust to gradients in images and have low detection errors. This routine is presented in next section.



## Chapter 4

# Dynamic Thresholding

An ideal thresholding algorithm for a star tracker must identify as many star pixels as possible close to noise level, while rejecting all the background pixels. The routine must also maintain the performance accuracy in presence of spacecrafts angular velocity and light gradients from other bright objects in sky. The thresholding algorithms investigated in this chapter uses In the new algorithms outlined in this section, the scenes are first processed using Win thresholding to estimate background level, and then areas of interest are processed using the Kittler-Illingworth algorithm. This method is expected to have low FNR and FPR while be very robust to varying background levels.

### 4.1 Existing Algorithms

Several adaptive star acquisition algorithms can be found in literature [15, 16]. The method presented by Sun *et al.* [15] includes denoising, signal enhancement and background removal. Sun *et al.* method is consisted of linear filters and operations so the wighted centroid of the star is not shifted. A correlation filter with a Gaussian kernel is applied to the original image  $I$  for the purpose of denoising the signal. The size of the kernel  $G$  was chosen to be  $6 \times 6$  in order to be consistent with the size of the point spread function (PSF) for the baseline sensor. The denoised image  $H$  can be obtained mathematically by:

$$H(x, y) = I(x, y) \circ G(x, y) \quad (4.1)$$

$$H(x, y) = \sum_{m=1}^M \sum_{n=1}^N I(m, n) H(m - i, n - j) \quad (4.2)$$

Sun *et al.* suggest that the correlation filter is efficient for filtering Gaussian noise, and also results in the PSF being extended and brighter without changing the centroid position.

Morphological operations (erosion and dilation) are implemented to smooth image background levels. Erosion and dilation operation are processed using a disk kernel  $D$  with radius of 25, values inside are 1 and outside are 0. To compute for  $M1$ , images  $S$  eroded by  $D$  we compute:

$$M1(x, y) = \min[S(x + i, y + j) - D(i, j)] \quad (4.3)$$

Then the resulting image is dilated by  $D$  to obtain an estimate of the varying background level  $M2$ .

$$M2(x, y) = \max[M1(x - i, y - j) + D(i, j)] \quad (4.4)$$

The background value  $B$  is estimated by applying an averaging window to  $M2$ . A constant value  $d$ , with a value between 10 to 20 is added to  $B$  to obtain threshold level:

$$T(x, y) = B(x, y) + d \quad (4.5)$$

This is then used to obtain the processed image  $F$  by:

$$F(x, y) = S(x, y) - T(x, y) \quad (4.6)$$

Sun *et al.* implemented their routine on star images obtained from a LEO satellite with estimated angular velocity of  $0.8^\circ/\text{s}$ . The experiment was repeated on simulated images of  $0.4$   $0.8^\circ/\text{s}$  slew rates.

The second star acquisition algorithm we considered was presented by Zhang *et al.* [16] and utilizes wavelet denoising to threshold star images in dynamic conditions. The specific wavelet thresholding algorithm implemented in Ref [16] is called Normal Shrink, originated from Zhang *et al.* [17]. Zhang *et al.* claims that it is easier to threshold star images in wavelet domain since light from the star are distributed as a smooth signal while the background noise is distributed at high frequency. Star images are transferred into wavelet domain using a 2-D orthogonal discrete wavelet transform (DWT). The noise variance in Normal Shrink routine is computed by following formula, where HH1 is the diagonal sub-band of level 1

$$\sigma^2 = \left[ \frac{\text{median}(\|HH1\|)}{0.6745} \right]^2 \quad (4.7)$$

The scale parameter  $\beta$  is calculated using  $L_k$ , which is the length of sub-band  $k$  level and  $J$  is the largest level.

$$\beta = \sqrt{\log \left( \frac{L_k}{J} \right)} \quad (4.8)$$

Finally the threshold for each pixel is calculated by:

$$T_N = \frac{\beta \sigma^2}{\sigma_y} \quad (4.9)$$

where  $\sigma_y$  is the standard deviation of each sub-band. Zhang *et al.* used SkyMap star simulation software to simulate images with various noise and up to angular rates of 5  $s$  for purpose of their thresholding test.

## 4.2 New Algorithms

In this section new methodologies for extracting and centroiding stars in dynamic conditions are discussed. This work offers two novel routines to improve star centroiding performance in dynamic imaging conditions. Both methods utilize a dynamically sized window based on the slew rate of the star tracker to refine the list of pixels selected for centroiding. Each method assumes typical ST-16 processing has been run to report a coarse guess for the centroid location.

Knowledge of the spacecraft slew rate can be obtained by external sensors such as a rate gyro, or directly from the star tracker, by comparing the change in attitude, or star centroid position across multiple exposures. Using the knowledge of the slew rate and Eq.3.4 from section 1, we can derive an estimate for star streak length. Utilizing this estimate of streak length, and the results from the previous examination of thresholding performance, two new methods for star detection was developed and they are based on dynamically-sized regions of interest (ROIs). Considering the motion model, the reference window is adopted to perform centroid determination.

The baseline processing of ST-16 was modified for this section. In cases where the slew is in direction of columns, the size of the row averaging must be modified. When star light occupies more than 30% of the averaging window pixels, the local threshold will be higher than the extended part of the star. Due to this effect, part of the star will not be detected and the centroid of the star is shifted. Using Eq.3.4 and an estimate of angular velocity, the extend of star streak in column direction can be calculated. The window size then can be resized to ensure the star will only occupy 30% or fewer pixels in the averaging window.

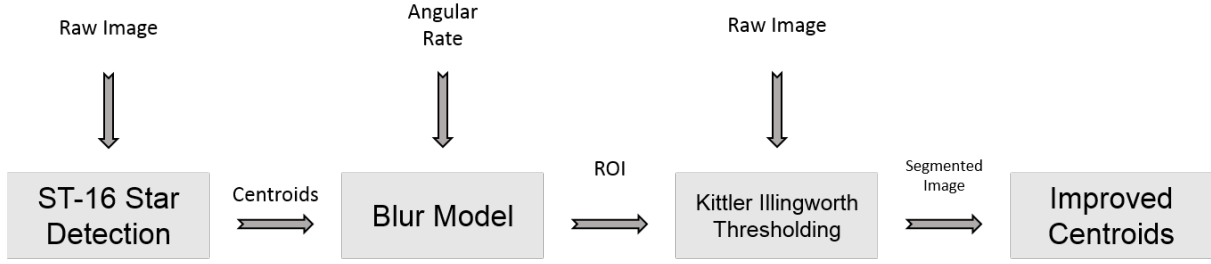


FIGURE 4.1: Star Tracker

In the Local-Kittler method, we define a box-shaped region of interest (ROI), centered on the initial estimate of the star position given by the modified ST-16 processing. The size of the ROI is calculated using Eq.3.4, that is  $\Delta \mathbf{b}$ . We then use the pixels in this ROI to calculate Kittler-Illingworth minimum error threshold. After processing, non-zero pixels are grouped together again as stars and the centroid of each new lit region is recalculated.

In the shape-based routine, we refine the ROI to more closely follow the shape of the star image. Using the initial centroid from the baseline sensor processor, we can find the star vector  $\mathbf{b}$  using the camera model, where  $\mathbf{b}$  is defined as:

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (4.10)$$

After obtaining  $\mathbf{b}$  and an estimate of  $\omega$  the satellite angular velocity; a line segment approximating star path on the image can be estimated from Eq.3.4. Given the the initial centroid, and  $\Delta \mathbf{b}$  length of the star streak, initial and final points of star in frame are calculated and a line path between them is obtained. we define a selection filter based on the convolution of a line,  $\Delta \mathbf{b}$  in length, with a circular kernel  $C$  of fixed diameter (empirically determined to match the sensor's PSF).

$$F(x, y) = \Delta \mathbf{b} \circ C(x, y) \quad (4.11)$$

For the ST-16, this circular kernel is chosen to be 12 pixels in diameter. All the pixels inside this ROI are classified as stars, and their centroid is computed by taking the first moment.

Alternatively a second central moment matrix of pixels selected in local adaptive threshold step could be used to fit an ellipse shape to estimate the cluster of the pixels. This method has



the advantage of being independent of angular rate knowledge and elliptical model if easy to formulate and fit. Since a blurred star image is not elliptical in shape, this method is just an estimation of ROI. Pixels in the new estimated ellipse are used to recalculate the centroid of the spots. The second central moment matrix of pixels belonging to a star could be computed as:

$$M = \begin{bmatrix} \sum_{\alpha=1}^N (x - \bar{x})^2 & \sum_{\alpha=1}^N (x - \bar{x})(y - \bar{y}) \\ \sum_{\alpha=1}^N (x - \bar{x})(y - \bar{y}) & \sum_{\alpha=1}^N (y - \bar{y})^2 \end{bmatrix} \quad (4.12)$$

where  $(\bar{x}, \bar{y})$  is estimated centroid. Eigenvalues and eigenvector of  $M$  corresponds to size and direction of major axis and minor axis of the estimated ellipse [18].

### 4.3 Simulation Tests and Results

The dynamic thresholding algorithms are tested in two cases again. Case one tests the algorithms for slew rate tolerance, and case two tests their robustness to stray light. To simply reference the different algorithms, the first routine developed will be referred to as Local-Kittler. The second routine will be then labeled 'Shape Ellipse' or 'Shape Footprint', depending on whether the shape was estimated using an ellipse or circle and line combination. The existing algorithm by Sun *et al.* will be referred to as 'Sunxing', and the algorithm developed by Zhang *et al.* will be labeled 'Wave'.

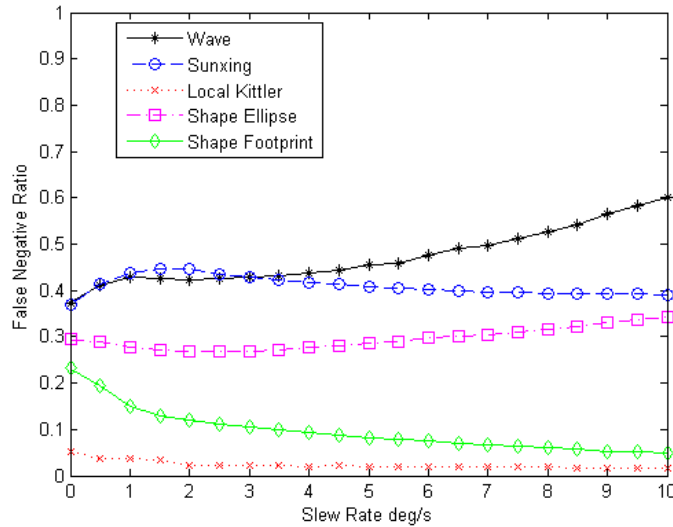


FIGURE 4.2: False negative ratio versus slew rate.

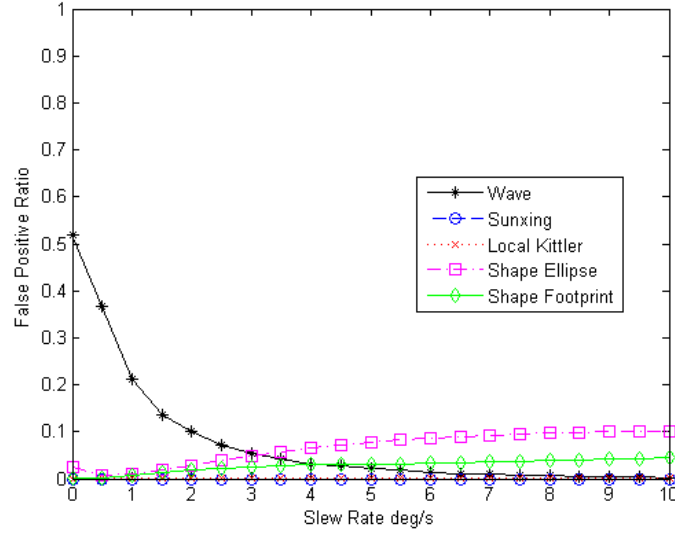


FIGURE 4.3: False positive ratio versus slew rate.

The result of the first case experiment is seen in Fig. 4.2 and Fig. 4.3. As it is seen in Eq.4.9, the normal Shrink threshold is dependent on the noise variance, which is estimated from the wavelet coefficient in sub-band HH1.

The Sunxing algorithm improves slightly as the slew rates increases. The size disk kernel  $C$  used in this method was selected so that it could capture the star within its boundaries. The radius of the disk was set to 25 pixels, so as the slew rates increases and the star cover more areas, the size of star blur get closer to disk, hence the better performance. However this algorithm still results in 40% FNR.

Among the Shape thresholding algorithms discussed, the Shape-Footprint outperformed the Ellipse shape as expected since an elongated circle is closer shape of blurred PSF than an ellipse. In slew rates lower than 2 deg/s, the false negative rate for the Shape footprint algorithm is higher, since the size of the region is dependent on the star trail length. Since the star trial length estimated is relatively is small, the area considered does not capture all of the pixels in a lower rates. Overall this algorithm has FNR of 10 % or less. The Local-Kittler algorithm outperformed the other four algorithms. This algorithm had FNR and FPR of less than 5%.

The Fig. 4.4 presents the ratio of correctly detected stars to all the stars in the scene. The Fig. 4.5 is the average number of incorrectly detected objects in star scenes versus angular rate. Most of the algorithms were capable of detecting almost all the star with no false detection. However the Wave algorithm detection rate drops with increasing angular velocity and it detects higher number of false objects.

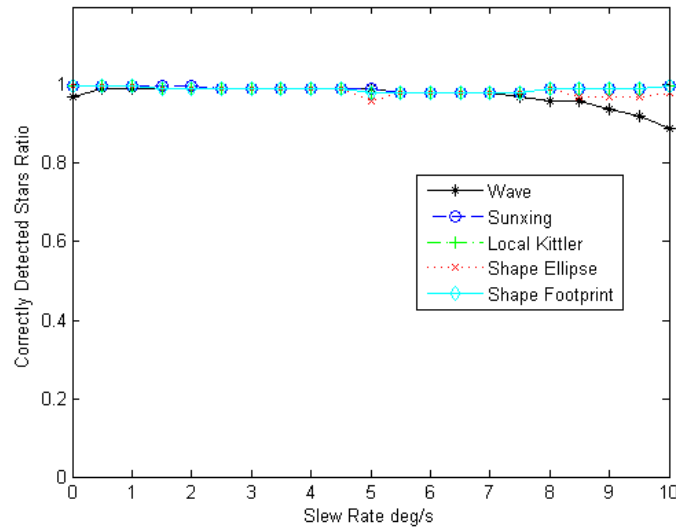


FIGURE 4.4: Ratio of Correctly Detected Stars

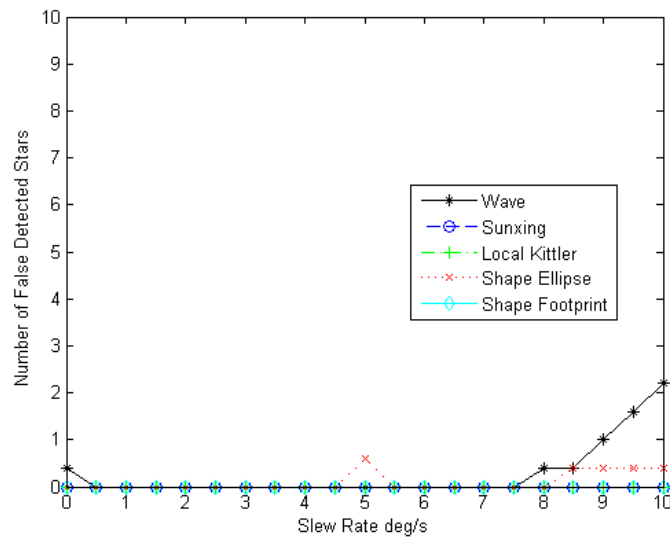


FIGURE 4.5: Average False Star Detected

In the second case, a light gradient is added to the frames. Most of the algorithms tested in this part are not affected by addition of extra background illumination except for the Sunxing algorithm. The Sunxing algorithm uses a Gaussian kernel to intensify the signal, but in case of the background illumination, the light from sources other than stars is also intensified and this causes a high FPR.

The Fig. 4.8 presents the ratio of correctly detected stars to all the stars in the scene plotted versus rate. The Fig. 4.9 is the average number of incorrectly detected objects in star scenes

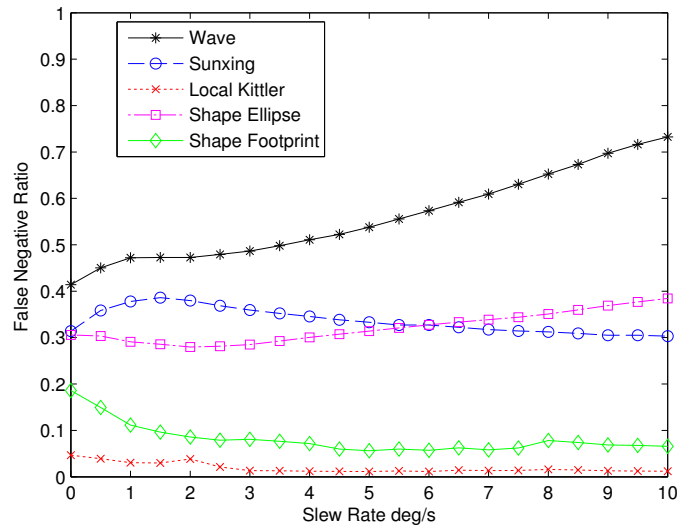


FIGURE 4.6: FNR versus Slew Rate with background gradient.

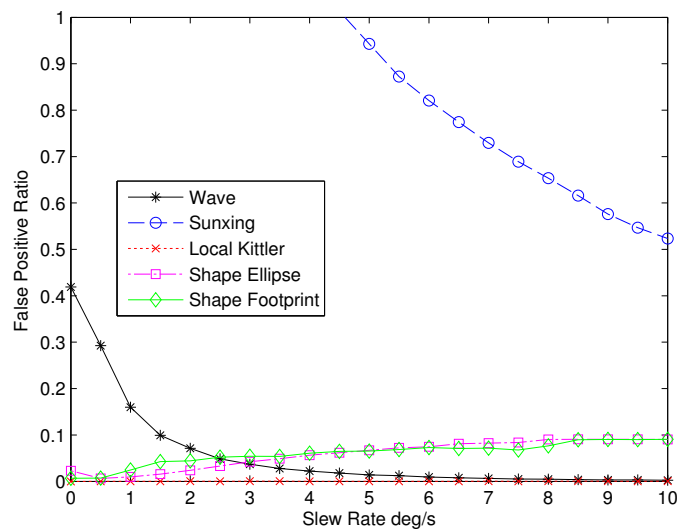


FIGURE 4.7: FPR versus Slew Rate with background gradient.

versus angular rate. In the case of background gradient, the Sunxing algorithm detects large number of false stars as the rate goes higher in the image area illuminated by stray light.

Based on the results of thresholding algorithms, Local-Kittler and Shape Footprint were determined to be superior algorithms in rejection of stray light and detection accuracy. In the next test, the centroiding accuracy of these two algorithms are discussed and compared.

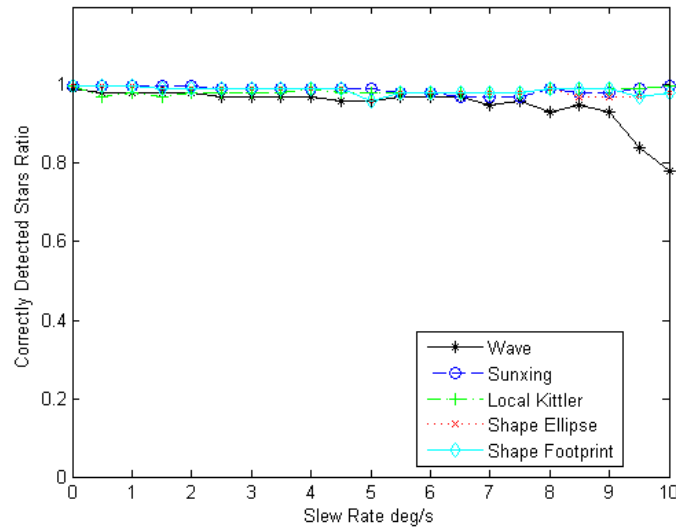


FIGURE 4.8: Ratio of Correctly Detected Stars

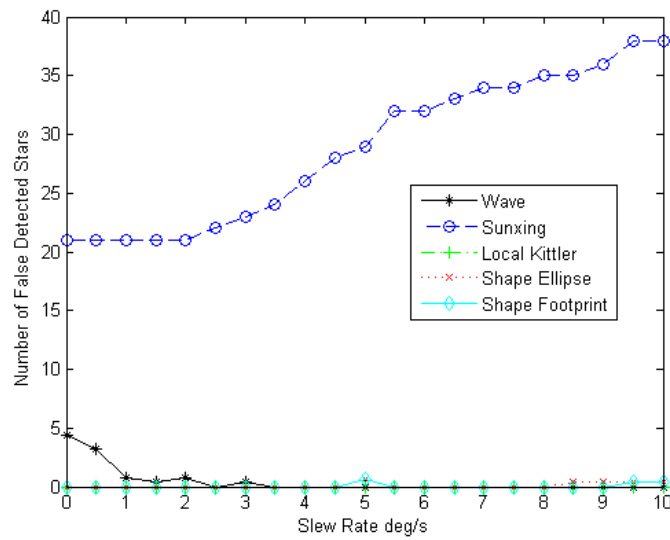


FIGURE 4.9: Average False Star Detected

## 4.4 Dynamic Thresholding Results Summary

In this section, the static thresholding algorithms were testing using synthetic star images. The algorithms are tested based on their ability to identify lit pixels. Two conditions were tested in this section, the performance in presence of motion and stray light. The results are summarized in Table 4.1. Considering these results, a new routine combining Win and Kittler algorithms

was developed to be robust to gradients in images and have low detection errors. This routine is presented in next section.

TABLE 4.1: Summay of Dynamic Algorithms Results

Method	Performance in Slew	Performance in Slew + Stray Light	Summary
<b>Local-Kittler</b>	Low FPR, Low FNR	Low FPR, Low FNR	Not sensitive to slew rate and stray light
<b>Shape-Ellipse</b>	Low FPR, medium FNR	High FPR, Medium FNR	Sensitive stray light
<b>Shape-Footprint</b>	Low FPR, Low FNR	Modest FPR, Low FNR	Not sensitive to slew rate and stray light
<b>Wave</b>	Medium FPR, High FNR	Medium FPR, High FNR	Sensitive stray light and slew
<b>Sunxingk</b>	Low FPR, Medium FNR	High FPR, Medium FNR	Not acceptable due to High FPR

## Chapter 5

# Star Localization

This chapter examines the centroiding accuracy of the local Kittler and Shape based algorithms. These two thresholding methods showed superior abilities to segment the image at dynamic conditions with non-uniform background. Centroiding is a common method of localization that uses the intensity values in pixels of the detected object to calculate the centroid of the objects of interest. Since in dynamic condition the centroid of the star moves on the frame, the localization methods are aimed at finding the centroid of the star half-way through the exposure. In this chapter the pixel-level thresholding are tested with ST-16 object-level and centroiding algorithm to compare their localizing accuracy. Afterward the detection schemes are tested on the group of stars with different brightness. These tests are there to ensure that the chosen algorithm maintain its localizing accuracy for dim star too. Then other localization schemes such as Gaussian distribution fitting and derivative methods and they are compared to the first moment centroiding. Finally these results are verified by lab tests and images taken by ST-16 star tracker. It is aimed to show these methods exhibit sub-pixel accuracy in both simulation and lab testing.

### 5.1 Centroiding Simulation Results

In this part, the Local Kittler and Shape Footprint algorithm are tested for centroiding accuracy. The pixels that were classified as star after each process were grouped together, and their first moment centroid was calculated. The root mean squared(RMS) errors in pixel for both algorithms were calculated from the centroids' location that was simulated. The results for this test are shown in Fig. 5.1. The centroiding errors from shape algorithm are slightly higher in lower slew rates, since star displacement in focal plane is small, the window size for these cases

are small and consequently some lit pixels were missed. The errors from both algorithms in  $x$  and  $y$  directions were lower than 0.15 pixel, so these routines are successful in segmenting and locating stars on the focal plane.

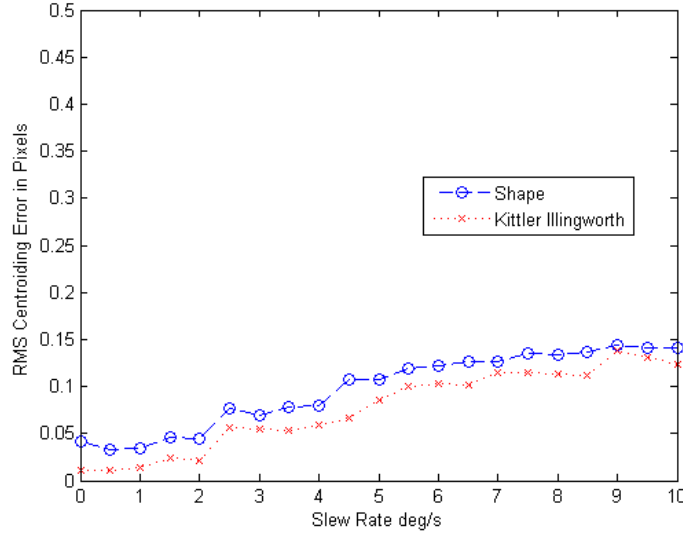


FIGURE 5.1: Centroiding accuracy.

### 5.1.1 Centroiding Performance for Different Stellar Magnitude

The stellar magnitude is a basic observable quantity for a star is its brightness. Because stars can have a very broad range of brightness, astronomers commonly introduce a logarithmic scale called a magnitude scale to classify the brightness [1]. In this scale, the lower the value of stellar magnitude the brighter the star is. The stars are grouped into three groups of brightness with magnitude  $M$  ranges of bright  $2.5 - 3.5$ , medium  $3.5 - 4.5$  and dim  $4.5 - 5.5$ . The stars in these magnitude range are slewed in lower than  $10^\circ/\text{s}$  and algorithms detection accuracy for each range is examined.

Both algorithms in all three magnitude range were able to achieve sub-pixels accuracy. For stars with high to medium brightness level, the accuracy of the detection increases insignificantly with increase in angular velocity. The dimmer star range however shows a strong dependency of accuracy to rate. Stars in magnitude range of  $4.5 - 5.5$  are closer to detection magnitude threshold of ST-16 which is 5.75. The detection magnitude threshold of star tracker are determined based on the static conditions. The stars above these magnitude are reliably detected in static conditions. However as the dim stars smear and its light spread across larger number of pixels, its



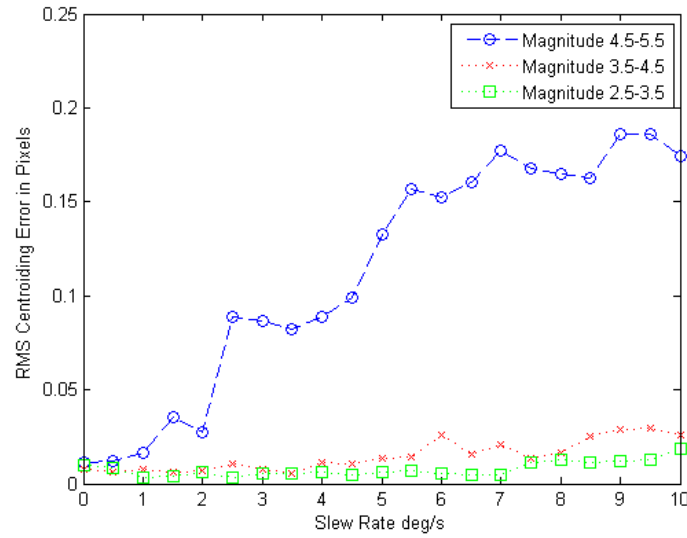


FIGURE 5.2: Centroiding accuracy of Local-Kittler algorithm for various stellar magnitude.

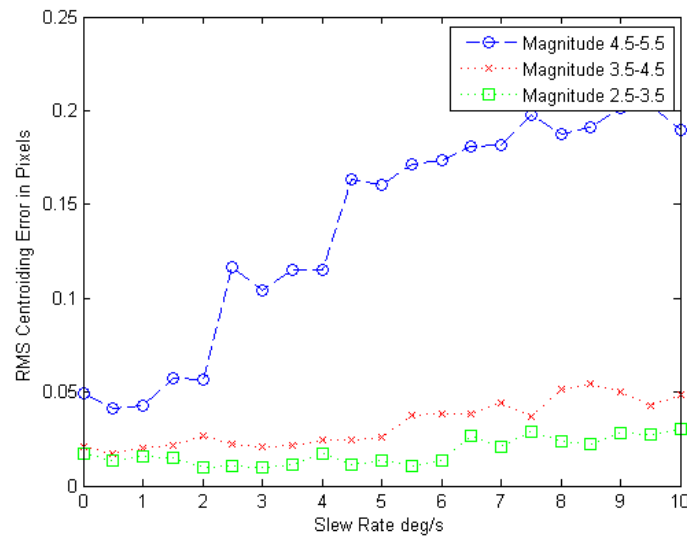


FIGURE 5.3: Centroiding accuracy of Shape-Based algorithm for various stellar magnitude

peak intensity becomes closer to background level, hence the noise have a larger impact on the stars centroid.

## 5.2 Laboratory Set Up

The algorithm developed in this study, was tested in labratory using ST-16 star tracker and a motorized three-axis gimbal capable of slewing at a prescribed rate. Fig. 5.4, shows the general

laboratory setup for a three pinhole telescope and the star tracker mounted on the gimbal. The gimbal is used to control the sensor's orientation with respect to the projected star light and while sensor is slewing, images are taken of a star source.

In these tests, a triangular three star pattern was projected on projection screen and star tracker was slewed from 0 to 5  $deg/s$  in  $x$  axis with 0.5 increments. Each step was repeated 20 times to insure the repeatability of the test. In the beginning of each tests stars were imaged 20 times in static conditions and the length of the triangular pattern side was estimated and averaged. Since the thresholding and centroiding algorithms are the most accurate in static conditions, the side length at static images are set to be the ground truth for this test. As the stars slew and their light smear across the large number of pixels, the relative distance between the star vectors should stay the same. To evaluate the centroiding error in higher rates, we will compare the estimated length of triangle side to the ground truth values. The difference in length is averaged as the centroiding error for his rate. These results should be studied further to remove the effects of ERS on the centroids of the stars.

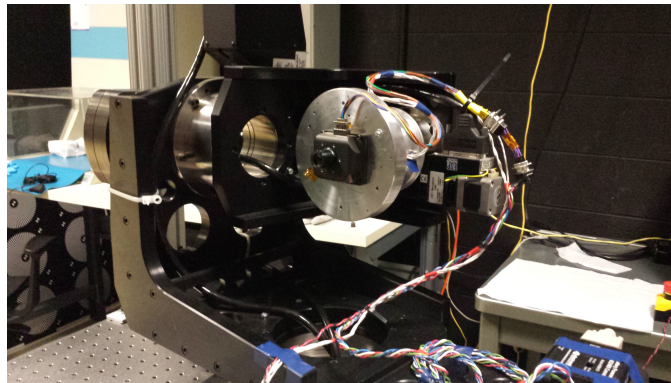


FIGURE 5.4: Laboratory set up.

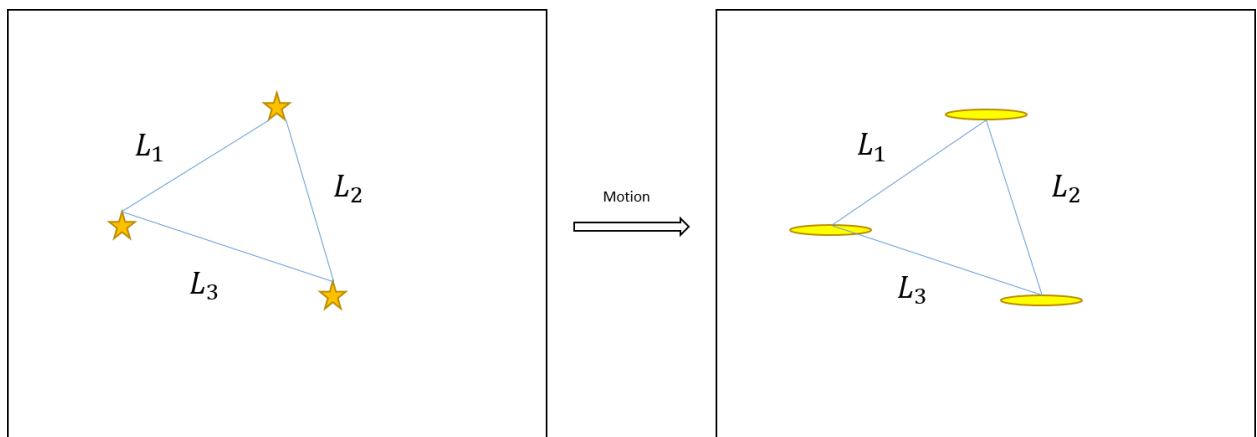


FIGURE 5.5: triangular pattern in static and dynamic condition.

### 5.3 Lab Testing Validation

The algorithm developed in this study, was tested in laboratory using ST-16 star tracker and a motorized three-axis gimbal. Fig. 5.4, represents the general laboratory setup for a three pinhole telescope and the star tracker mounted on the gimbal. In these tests, a triangular three star pattern was projected on projection screen and star tracker was slewed from 0 to 5  $^{\circ}/s$  in  $x$  axis with 0.5  $^{\circ}/s$  increments. Each step was repeated 10 times to insure the repeatability of the test. In the beginning of each tests stars were imaged 20 times in static conditions and the length of the triangular pattern side was estimated and averaged. Since the thresholding and centroiding algorithms are the most accurate in static conditions, the side length at static images are set to be the ground truth for this test. As the stars slew and their light smear across the large number of pixels, the relative distance between their centroids should stay the same. To evaluate the centroiding error in higher rates, we will compare the estimated length of triangle side to the ground truth values. The difference in length is averaged as the centroiding error for his rate. These results should be studied further to remove the effects of ERS on the centroids of the stars.

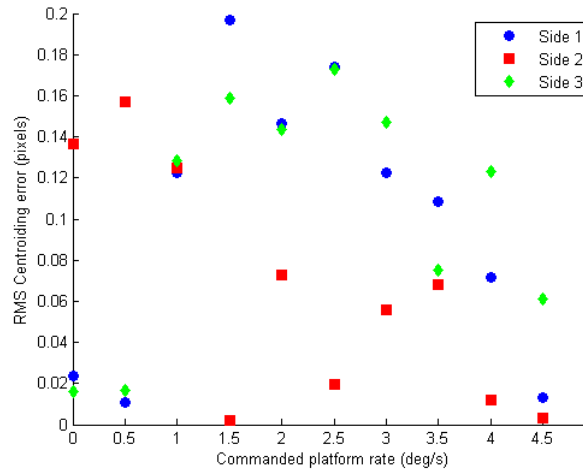


FIGURE 5.6: Triangular pattern length error in lab tests.

Unlike previous section, the images obtained in this section are influenced with electronic rolling shutter of the detector. The results for this test are presented in Fig. 5.6. However because of the sensor ERS, a centroid displacement in row and column direction of the image can also be seen. Due to the ERS effect the error in star allocations are higher.



## Chapter 6

# Conclusion

In this work thresholding algorithms for star tracker images are reviewed to improve star centroiding accuracy in high slew rates. The common thresholding algorithms in literature were identified and studied. Static thresholding algorithms in literature do not make an assumption regarding image motion and segment the images without prior knowledge of the angular rate. The promising static thresholding algorithm in each subcategory of local, cluster-based and attribute based was selected and synthesized. These algorithms were applied to star images in angular rates from  $0-10^\circ/\text{s}$ . In the second scenario these algorithms were tested in presence of background gradient. It was observed that although local algorithms are robust to nonuniform background level, while they had a higher FNR. On the other hand, the global method, Kittler-Illegworth had a low FNR and PNR and was robust to slew rate, however in presence of varying background illumination, it did not exhibit the same results. Based on these results, two new routines for star detection were developed. The new thresholding algorithms were categorized as dynamic thresholding algorithms as they require prior knowledge of the spacecraft angular rate. The Local-Kittler algorithm applies the Kittler-Illegworth at the local level in a ROI confining a star detected by the regular the baseline sensor detection process. The Shape-based methods use a common blur model follow the shape of star PSF more closely to select the lit pixels. The performance of these novel algorithms are then compared to other dynamic thresholding algorithms in the literature. The dynamic algorithms were tested to verify the slew rate tolerance and robustness to stray light. After confirming the effectiveness of the new thresholding routines, their centroiding accuracy was checked. Based on the simulation results the centroiding accuracy of the both Local-kittler and Shape-Footprint was kept below 0.2 pixel, which is the ST-16 centroiding noise at static conditions. A lab experiment was designed to verify the performance of these algorithms using the baseline sensor. Unlike

simulated images, the actual sensor images require ERS correction for an accurate analysis of the detection algorithms. Since the lab test set up, does not provide an output of angular rate, the commanded angular velocity to the platform was used to correct for ERS effects. Despite the correction, the ERS effect on the streak length is still observed on the final results. this suggest that these algorithms require a more accurate estimate of angular velocity than what is currently available. As future work it is recommended to add a rate gyro to the testing platform to be able to measure the angular velocity of the platform at the time of imaging more accurately. Computational efficiency has not been considered in this evaluation; it is recommended to reexamine these algorithms based on their computational cost. Despite the uncertainties in the effective ERS correction, the Local-Kittler algorithm showed promising results in detecting and centroiding the dim stars in presence of high angular rate. To validate the efficiency of Local-Kittler algorithms, night sky field tests using ST-16 must be conducted. Given the results from the lab tests, the novel detection algorithm offered in this work can enable small spacecrafts with a star tracker on-board to track their absolute attitude without any hardware changes.

# Bibliography

- [1] Carl Christian Liebe. Accuracy performance of star trackers-a tutorial. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(2):587–599, 2002.
- [2] K. Gromov C. C. Liebe and D. M. Meller. Toward a stellar gyroscope for spacecraft attitude determination. *JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS*, 27: 91–00, 2004.
- [3] Tom Dzamba and John Enright. Ground testing strategies for verifying the slew rate tolerance of star trackers. *Sensors*, 14(3):3939–3964, 2014. ISSN 1424-8220. doi: 10.3390/s140303939. URL <http://www.mdpi.com/1424-8220/14/3/3939>.
- [4] Martin Marciniak, John Enright, Doug Sinclair, and Tom Dzamba. Microsatellite star tracker baffles: Validation and testing. 2013.
- [5] Tom Dzamba. *IMPROVING THE MEASUREMENT QUALITY OF SMALL SATELLITE STAR TRACKERS*. PhD thesis, Ryerson University, Department of Aerospace Engineering, 2013.
- [6] Tom Dzamba and John Enright. Optical trades for evolving a small arcsecond star tracker. In *Aerospace Conference, 2013 IEEE*, pages 1–9. IEEE, 2013.
- [7] Oivind Due Trier and Anil K. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:1191–1201, December 1995.
- [8] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Electronic Imaging*, 13:146–165, January 2004.
- [9] Carl Christian Liebe, Edwin W. Dennison, Bruce Hancock, Robert C. Stirbl, and Bedabrata Pain. Active pixel sensor (aps) based star tracker. In *Aerospace Conference, 1998 IEEE*, pages 119–127. IEEE, 1998.
- [10] J. Bernsen. Dynamic thresholding of grey-level images. In *International Conference on Pattern Recognition*, 1986.

- 
- [11] Wayne Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, Birkerød, Denmark, 1985.
  - [12] NOBUYUKI OTSU. A threshold selection method from gray-level histograms. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 9(1):62–66, January 1979.
  - [13] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19:41–47, 1986.
  - [14] Wen-Hsiang H. Tsai. Moment-preserving thresholding: a new approach. *Pattern Recognition*, 29(3):377–393, March 1985.
  - [15] Ting Sun, Fei Xing, Zheng You, and Minsong Wei. Motion-blurred star acquisition method of the star tracker under high dynamic conditions. *OPTICS EXPRESS*, 21(17), August 2013.
  - [16] Weina Zhang, Wei Quan, and Lei Guo. Motion-blurred star acquisition method of the star tracker under high dynamic conditions. *Sensors*, 21, April 2012.
  - [17] Lakhwinder Kaur, Savita Gupta, and R. C. Chauhan. Image denoising using wavelet thresholding. In *INDIAN CONFERENCE ON COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING, AHMEDABAD*, 2002.
  - [18] Kenichi Kanatani. *Statistical Optimization for Geometric Computation*. Dover Publications, INC, Mineola, New York, 1996.
  - [19] D. Accardo G. Fasano, G. Rufino and M. Grassi. Satellite angular velocity estimation based on star images and optical flow techniques. (13):12771–12793.
  - [20] T. Tuytelaars H. Bay, A. Ess and L. Van Gool. Speeded-up robust features (surf). pages 346–359.
  - [21] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 381–395.
  - [22] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry,.
  - [23] J. Lewis S. Roth M. J. Black S. Baker, D. Scharstein and R. Szeliski. A database and evaluation methodology for optical flow.