

A LOW COST, HIGH PRECISION STAR SENSOR

by

M J JACOBS



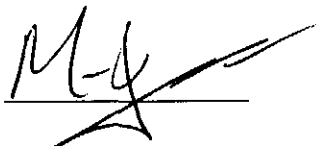
THESIS PRESENTED FOR THE FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE: MASTER OF ELECTRONIC ENGINEERING AT THE
UNIVERSITY OF STELLENBOSCH

December 1995

Study leader: Dr. W H STEYN

DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and has not previously been submitted at any university, in part or its entirety, for a degree.

A handwritten signature in black ink, appearing to be 'M. J.', written over a horizontal line.

Signed, December 1995 in Stellenbosch

SYNOPSIS

The Sunsat microsatellite is being developed at the University of Stellenbosch. It is equipped with a high resolution earth imager which requires a sophisticated attitude determination and control system. Precise control of the satellite is impossible without high precision attitude sensors. The most accurate attitude sensor presently used by satellites is the star sensor.

The aim of this thesis is to develop a low cost, high precision star sensor for the Sunsat microsatellite. The development included the selection of electronic and optical hardware followed by the design, construction and testing of the sensor. Software algorithms were developed and programmed to control the imager, extract stars from images and match the observed constellations to a star catalogue. A set of matching observed and reference vectors are passed to the attitude determination and control system which calculates the three axis orientation of the satellite.

OPSOMMING

Die Sunsat mikrosatelliet word tans by die Universiteit van Stellenbosch ontwikkel. Dit is toegerus met 'n hoëresolusie aardwaarnemingskamera en vereis 'n gevorderde oriëntasiebepaling-en beheerstelsel. Presiese beheer is onmoontlik sonder akkurate sensore. 'n Stersensor is die mees akkurate tipe sensor waarmee satelliete tans toegerus word.

Die doelwit van hierdie tesis is die ontwikkeling van 'n hoë resoluësie, laekoste stersensor vir die Sunsat mikrosatelliet. Die ontwikkeling behels die seleksie van elektroniese en optiese apparatuur wat gevolg is deur die ontwerp, opbou en toets van die sensor. Programmatuur algoritmes is ontwerp om die kamera te beheer, sterre in fotos te herken en waargenome konstellasies in die ster katalogus te vind. 'n Stel waargenome en verwysingsvektore stel die oriëntasiebepaling-en beheerstelsel in staat om die oriëntasie van die satelliet te bereken.

CONTENTS

List of Figures

Introduction	i
Acknowledgements	iii
Notation and Symbols	iv

1. COORDINATE SYSTEMS	1
1.1 The CCD coordinate systems	1
1.1.1 The column-row (<i>colrow</i>) coordinate system	1
1.1.2 The horizontal-vertical (<i>horvert</i>) coordinate system	2
1.1.3 The <i>uv</i> coordinate system	3
1.2 Star sensor (<i>ss</i>) coordinate system	4
1.3 Satellite body (<i>b</i>) coordinate system	5
1.4 The orbit (<i>o</i>) coordinate system	7
1.5 The celestial (<i>c</i>) coordinate system	9
1.6 The celestial sphere (<i>cs</i>) coordinate system	12
1.7 The complete transformation	12
2. SOFTWARE	14
2.1 Software summary	14
2.2 The <i>Region Growing</i> Module	15
2.2.1 Detail description of the region growing algorithm	15
2.2.2 Calculating the centroid	21
2.2.3 The shape of a region	23
2.2.4 Calculating the magnitude	25
2.2.5 Returning the results	26
2.3 Testing	26
2.4 Conclusion	27

3. THE MATCHING ALGORITHM	28
3.1 Calling the <i>Region Growing</i> module	28
3.2 Matching algorithms considered	29
3.2.1 The van Bezooijen algorithm	29
3.2.2 Other matching algorithms	30
3.3 Detail description of the <i>van Bezooijen</i> Algorithm	32
3.3.1 Generation of match groups through star pair matching	34
3.3.2 Elimination of small match groups	37
3.3.3 Geometry test	38
3.3.4 Elimination of redundant match groups	39
3.3.5 Extraction of the largest match groups	39
3.4 Transformation of centroids	39
3.5 Verifying and testing of the algorithm	40
3.6 Conclusion	44
 4. THE TRACKING ALGORITHM	 45
4.1 Detail description of the Tracking algorithm	45
4.1.1 Calculate transformation matrices	47
4.1.2 Finding catalogue stars in the FOV	48
4.1.3 Mapping reference stars from celestial sphere to CCD plane	52
4.1.4 Selecting the best three matches	52
4.1.5 Transformation of centroids	58
4.1.6 Attitude information for the next iteration	58
4.1.6.1 The Triad algorithm	58
4.1.6.2 Accuracy of the <i>Triad</i> algorithm	60
4.1.6.3 Determining the size of the search area	61
4.1.6.4 Conclusions regarding the use of the ADCS and <i>Triad</i> estimates	63
4.1.7 Returning the matching vectors	64
4.1.8 The predictor	64
4.2 Synchronization	65
4.2.1 Adjusting the frame rate	65
4.2.2 Synchronization of the Tracking algorithm and ADCS	65
4.3 Testing	66
4.4 Conclusion	68
 5. GROUND SUPPORT SOFTWARE	 69
5.1 Statistics	69
5.2 Making a sub catalogue	71
5.2.1 Method used to produce the sub catalogue	71

6. CALIBRATION	74
6.1 Measurement errors expected	74
6.1.1 Focal length error	74
6.1.2 Positional error of optical centre	77
6.2 Calibration procedure	79
6.2.1 Locating the optical centre	80
6.2.2 Estimating the focal length	81
6.2.3 Distortion model	82
6.2.3.1 Estimating the distortion parameters	84
6.2.3.2 Using the distortion model	86
6.3 Results	86
6.4 Conclusion	89
 7. EXTERNAL FACTORS AFFECTING THE STAR SENSOR	 90
7.1 Parallax	90
7.2 Precession of the equinoxes	91
7.3 Proper Motion	91
7.4 Variable stars	91
7.5 Aberration	93
7.5.1 Worst case aberration	94
7.5.2 Calculating and correcting aberration	94
7.5.2.1 Determining the velocity vector	95
7.5.2.2 Calculating the aberration	96
7.5.2.3 Correcting the aberration	97
7.6 Bright objects	98
7.7 Conclusion	99
 8. HARDWARE	 101
8.1 The Microprocessor	102
8.2 The RWSS (Reaction Wheel Star Sensor) Bus Interface	104
8.3 Memory	104
8.4 Power Supplies	106
8.5 CCD circuit	106
8.5.1 Clocking circuit	107
8.5.2 The CCD and signal processing circuit	108
8.7 Results	114
8.8 Conclusion	116

9. SELECTION OF CCD'S AND OPTICS	117
9.1 Selection of optics	117
9.2 Sony CCD's	119
9.2.1 The dark signal non uniformity voltage	120
9.2.2 The photogenerated voltage	120
10. CONCLUSION	125

References

APPENDIX

A Astronomy
B Spherical Geometry
C Distance and Magnitude Calculations
D <i>Matching</i> Algorithm
E <i>Tracking</i> Algorithm
F Calibration
G Drivers
H Selection of CCD's
I Programme listings
J Hardware Schematics

List of Figures

Figure 1-1 The <i>column-row</i> coordinate system	1
Figure 1-2 The <i>horizontal-vertical</i> and <i>uv</i> coordinate systems	2
Figure 1-3 Relation between the <i>uv</i> and <i>ss</i> coordinate systems	4
Figure 1-4 Star sensor in the satellite body coordinate system	5
Figure 1-5 An Euler 1-2-3 rotation	6
Figure 1-6 Definition of the orbit coordinate system in the celestial coordinate system	8
Figure 1-7 The satellite body in the orbit coordinate system	8
Figure 1-8 (x', y', z_c) after a rotation of $90^\circ + \Omega$ about z_c	9
Figure 1-9 (x'', y', z'') after a rotation of $-i$ about y'	10
Figure 1-10 (x'', y''', z''') after a rotation of -90° about x''	10
Figure 1-11 The <i>orbit</i> coordinate system reached after a rotation of ω' about y'''	11
Figure 2-1 A typical image	15
Figure 2-2 Top level flow diagram of the region growing algorithm	16
Figure 2-3 Region growing in the image. A jump size of three pixels is used to locate the first light pixel belonging to the star.	18
Figure 2-4 Flow diagram of the <i>grow region</i> block in Figure 2-2	19
Figure 2-5 Lower bound on the error expected using:	22
Figure 2-6 The long axis a , short axis b and direction λ of a region	23
Figure 2-7 Extraction time for a single star	27
Figure 3-1 Exponential increase in number of stars with increasing magnitude	30
Figure 3-2 Flow diagram of van Bezooijen algorithm	33
Figure 3-3 Explanation of matching example. Match numbers are encircled and star numbers appear next to the stars.	35
Figure 3-4 Generation of a match group with observed star 1 as kernel. The matching pairs are indicated in parenthesis.	36
Figure 4-1 Top level flow diagram of the Tracking algorithm	46
Figure 4-2 FOV on celestial sphere, projected onto a flat surface	49
Figure 4-3 FOV (a) near the celestial equator, (b) over the North pole and (c) over the edge of catalogue	50
Figure 4-4 Subdivision of the CCD plane into nine boxes, showing the positions of the reference stars and the search areas (grey rectangles)	52
Figure 4-5 Selection of the best three matches	53
Figure 4-6 Observed stars (solid) actually appearing in the image and reference stars (outlined) used to find three of them. Due to the orientation error, all the observed stars appear shifted to the left from the expected positions.	55
Figure 4-7 Synchronization of ADCS and <i>Track</i> is shown in the first and second lines and that of <i>Track</i> and the CCD circuit in the second and third lines. The sequence starts at time = 0 s.66	56
Figure 5-1 Histogram showing the number of images containing the indicated number of stars. A total of 1 million random FOV locations on the celestial sphere were used.	70
Figure 5-2 FOV on the celestial sphere over a period of two weeks (large white oval). Also shown are three single fields of view to illustrate how the projected size changes with declination. The FOV on the celestial equator (centre horizontal line) appears small. The one close to the South Pole appears larger while the FOV including the South Pole extends right around the projection.	72
Figure 5-3 Position of the FOV over a period of a year, for the extreme attitude values listed in Table 5-2, but the right ascension of the ascending node changing from 0° to 360° .	73
Figure 6-1 Actual angular distance d_a and measured distance d when incorrect focal length f is used. The error in focal length estimate is exaggerated.	75

Figure 6-2 Angular error as the pair of stars (separated by 0.5 mm) move around the CCD plane.	76
Figure 6-3 Angular error as the error in estimated focal length f varies from $-40\text{ }\mu\text{m}$ to $40\text{ }\mu\text{m}$, with true focal length $f_a = 25\text{ mm}$.	76
Figure 6-4 Angular error occurring due to incorrect position of optical centre.	77
Figure 6-5 Angular error when the estimated optical centre shifts from the actual position ($u_a = 0\text{ mm}$) to $u_a = 0.65\text{ mm}$. Star 1 moves from $(u_1, v_1) = (0, 1.5)$ to $(u_1, v_1) = (2, 1.5)$ while star 2 is fixed at $(u_2, v_2) = (0, -1.5)$	78
Figure 6-6 Error occurring due to tilted CCD plane	82
Figure 6-8 Histogram of the errors in angular distances between stars. Each bar of the histogram spans $10\text{ }\mu\text{rad}$ and a total of 4385 distances were measured.	87
Figure 6-9 Histogram of the magnitude error. Each bar spans 0.1 Mv and 354 errors were entered.	88
Figure 7-1 Parallax of close stars due the earth's rotation about the sun	90
Figure 7-2 Aberration due to velocity of the satellite. The star appears at S' instead of its true position S .	93
Figure 7-3 Rotation of the reference vector through angle α to correct for aberration	96
Figure 7-4 The baffle used to prevent direct light from bright objects reaching the lens	98
Figure 8-1 Block diagram of the hardware	102
Figure 8-2 Memory map	105
Figure 8-3 State diagram for storing image in memory	108
Figure 8-4 Signal path from the CCD to image memory	109
Figure 8-5 Signal waveforms	110
Figure 8-6 Image centred at $\text{RA} = 97^\circ$, $\text{Dec} = -55^\circ$	114
Figure 8-7 Section of the catalogue corresponding to the image of Figure 8-6, limiting magnitude is $\text{Mv} = 6$	115
Figure 8-8 Section of the catalogue corresponding to the image of Figure 8-6, limiting magnitude is $\text{Mv} = 7$	115
Figure 9-1 Path of light from the star to the CCD plane	118
Figure 9-2 Expected Gaussian distribution of a star on the CCD plane. The pixel centres are at the intersections of the lines. The standard deviation is 1.5 pixels.	121
Figure 9-3 Relation between peak amplitude and standard deviation of the Gaussian, for a magnitude 6 star	123
Figure 9-4 Determining σ for which the edge pixel detectable. The threshold of detectability is shown as the horizontal line at 1.13 mV	124

Introduction

The Sunsat microsatellite is being developed at the University of Stellenbosch. One of the goals of the project is to produce all possible sub systems at the university and in this way provide masters students with a variety of research topics.

Sunsat is equipped with a high resolution earth imager which requires a sophisticated attitude determination and control system (ADCS). The ADCS gathers data from earth horizon sensors, sun sensors and a magnetometer. The need for a higher precision attitude sensor was identified to improve the pointing accuracy of the earth imager. The most precise method of determining a satellite's orientation is by means of a star sensor. The principle of operation is to image stars and match the observed constellation to a star catalogue. The set of corresponding observed and reference vectors are used to determine the satellite's orientation. At least three matching pairs of stars are required for full three axis attitude determination.

A star sensor can be purchased for tens of thousands of US Dollars, but this does not agree with the Sunsat policy of developing all sub systems, nor does it agree with the Sunsat budget. The aim of this thesis was to develop a *high precision, low cost* star sensor for the Sunsat microsatellite. This includes the design and production of the electronic hardware and the development of software algorithms. The design should not be considered as specific to Sunsat and may be adapted to interface other processors.

The following design specifications were set

- An accuracy of at least 1 *mrاد* was desired. The accuracy of 1 *mrاد* means that the earth imager can be directed with an accuracy no better than 1 *mrاد*. From a height of 800 km, this converts to 800 m on the earth's surface.
- To provide a high probability of finding at least three stars in the image, magnitude 6 stars should be detectable in a 10° square field of view.
- ADCS requires an orientation update every second
- Overall cost should be minimized
- Power consumption should be minimized

During the past few decades, star sensors have been produced from various devices. Array CCD's (charged coupled devices) are presently the best suited imagers for star sensor applications. They are simple to operate, low power and high precision components.

This document elaborates on the development and testing of the software and hardware required to achieve the abovementioned specifications. It is presented as three sections:

The first section consists of Chapters 1 to 5 and discusses software and related topics. The software algorithms required to control the star sensor, recognize stars and

provide matching observed and reference vectors to the ADCS, were developed in parallel with the hardware.

- Chapter 1 defines the coordinate systems which form the basis of the software. All the software algorithms extensively use transformations between the coordinate systems.
- Chapter 2 introduces the software modules and discusses the technique used to extract stars from the image and estimate their centroids and magnitudes.
- Chapter 3 explains an algorithm which matches an imaged constellation to the star catalogue, with limited a-priori attitude information.
- Chapter 4 discusses the algorithm which was developed to update the ADCS orientation every second.
- Chapter 5 introduces the software which forms part of the star sensor's ground support.

The next section examines the accuracy expected from the star sensor as well as any external factors which may affect this accuracy.

- Chapter 6 describes the calibration of the star sensor and determines the accuracy.
- Chapter 7 compares the effect of various external factors on the accuracy of the star sensor. Those which have a significant effect are corrected for.

The last section discusses the electronic and optical hardware. The selection of components, design and testing is summarized. Throughout the design, attention is devoted to the cost of components. Cost is minimized by selecting mass produced commercial components, rather than custom made devices. Commercial components generally tend to be robust, reliable and inexpensive. Electronic components were, where possible, selected to minimize power consumption.

- Chapter 8 concentrates on the selection of the processor and other digital components. The design and testing of the complete sensor is discussed. Images are compared to a star catalogue to prove the sensor's ability to detect faint stars.
- Chapter 9 is devoted to the selection of the CCD and lens.

A summary of results and concluding remarks are presented in Chapter 10.

Acknowledgements

I would like to express my sincere appreciation towards:

- My Creator
- My study leader, W H Steyn, for his guidance and assistance
- Leon Heinkelein for his help and advice with the electronics
- John Jorgensen at Oersted for the name of the only company in the world which was prepared to sell us the Sony CCD's

Notation and Symbols

The following notation and symbols are used throughout the thesis

\hat{x}	unit vector
\underline{x}	vector
$x \rightarrow y$	transformation from coordinate system x to y
A	transformation matrix
A^T	transpose of matrix A
f	focal length
α	right ascension (RA)
δ	declination (Dec)
$\langle x \rangle$	true value of x i.e. not the estimated value
\approx	approximately equal to
\dot{x}	time derivative of x
L	luminance measured in <i>lumen.sr⁻¹.m⁻²</i>
I	luminous intensity measured in <i>lumen.sr⁻¹</i>
F	luminous flux measured in <i>lumen, lum</i>
E	illumination measured in <i>lux, lum.m⁻²</i>
B	lens transmission factor

1. Coordinate Systems

A set of coordinate systems and the transformations between them are defined in this Chapter. They are used throughout the thesis and in the software algorithms. Where applicable, the convention corresponds to that used by Steyn [1] for the Sunsat ADCS system. This conforms with the NASA standard which is defined in Wertz [2].

The following coordinate systems are defined:

1.1 The CCD coordinate systems

A point on the CCD may be referenced to one of three coordinate systems:

1.1.1 The column-row (*colrow*) coordinate system

This is a discrete coordinate system where the unit of measurement is pixels. The reference point is the bottom left of the dummy pixel area of the CCD, shown in Figure 1-1.

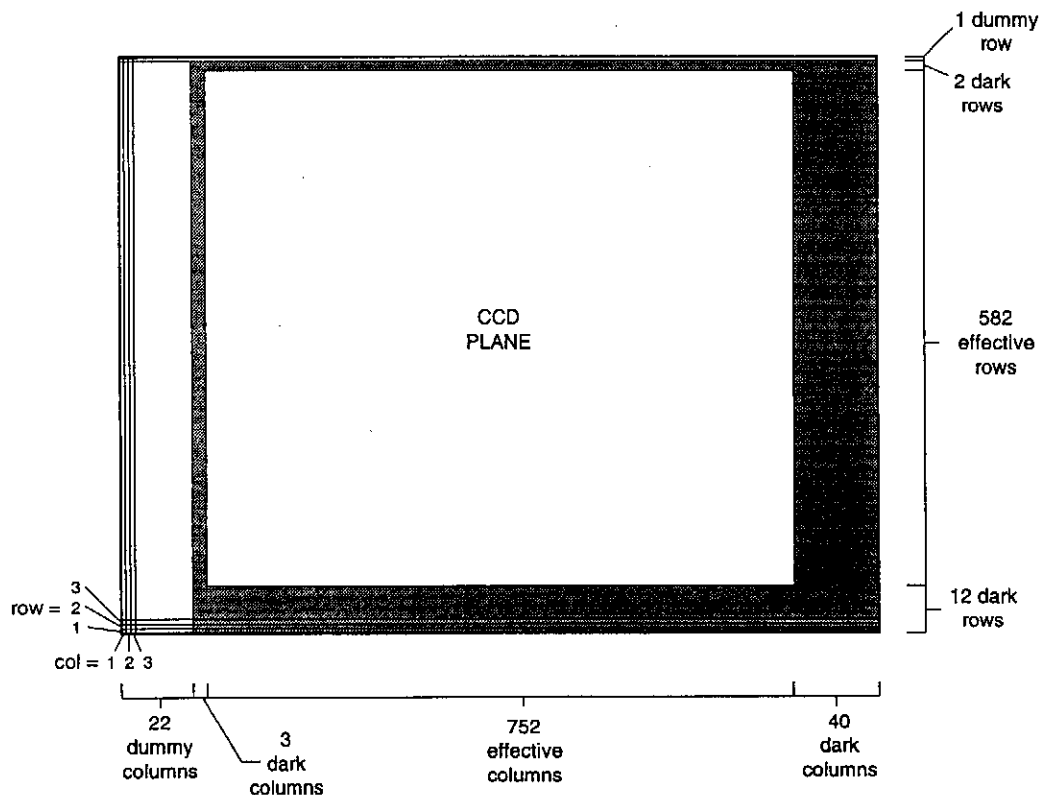


Figure 1-1 The column-row coordinate system

Its sole purpose is to facilitate the process of finding the pixel corresponding to a point on the CCD plane, in the image memory. For this reason the dummy and dark

reference pixels, which are also clocked into memory, are included and not only the image pixels. Any vector referenced to this coordinate system is denoted by the subscript *colrow*, where the '*col*' refers to columns and the '*row*' to rows on the CCD. As an example, the bottom left pixel of the actual light sensitive image area is located at $lightpixel_{colrow} = (26, 13)$.

1.1.2 The horizontal-vertical (*horvert*) coordinate system

This continuous coordinate system is required to locate a point specified in *mm* on the CCD plane. The reference point is identical to that of the *colrow* system. The top right corner of the pixel (*col*, *row*) is found by multiplying with the pixel unit cell size in *mm*:

$$\begin{aligned} hor &= col \times hunit \\ vert &= row \times vunit \end{aligned} \tag{1.1}$$

with

$$hunit = 0.0065 \text{ mm}$$

$$vunit = 0.00625 \text{ mm}$$

as quoted in the SONY data sheets for the *ICX059AL* CCD.

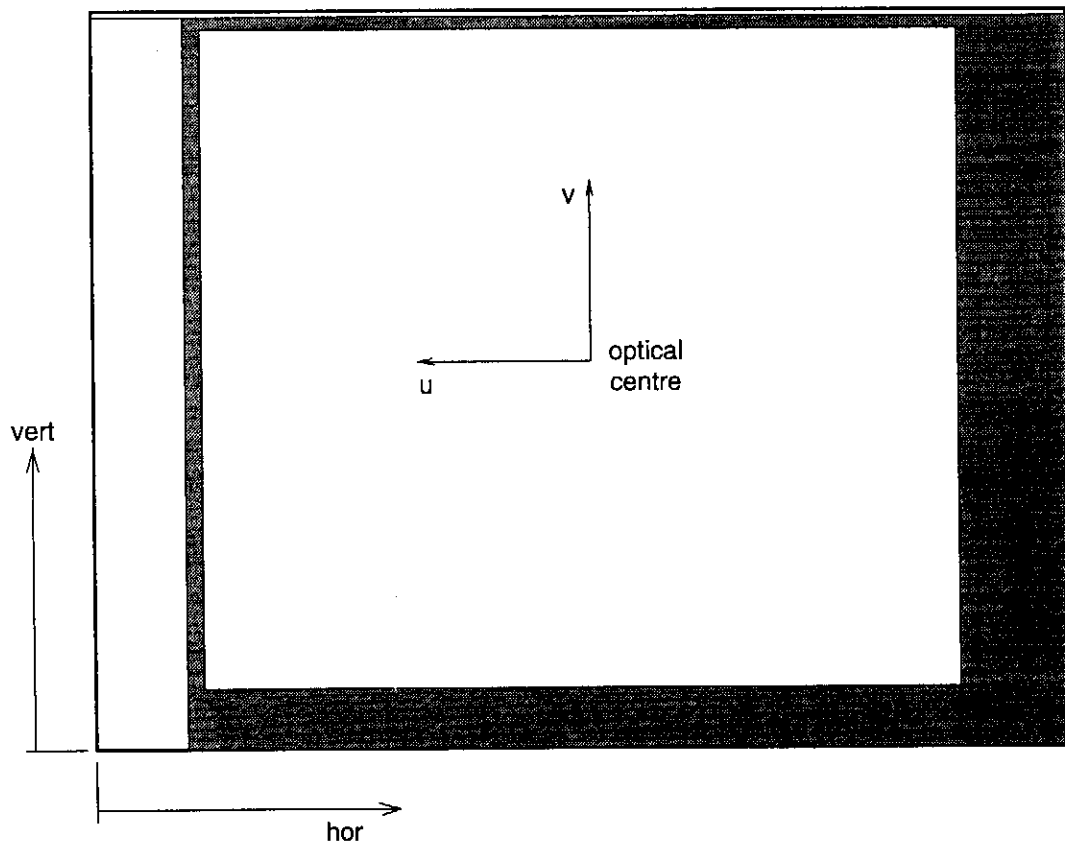


Figure 1-2 The *horizontal-vertical* and *uv* coordinate systems

The function¹ *horverttocolrow* uses these equations to map a point from the *horizontal-vertical* coordinate system to the *column-row* system, while *colrowtohorvert* uses

$$\begin{aligned} col &= \text{trunc}(hor / hunit) + 1 \\ row &= \text{trunc}(vert / vunit) + 1 \end{aligned} \quad (1.2)$$

where the *trunc* function truncates the value in parenthesis.

As an example, the top right corner of the first light sensitive pixel will be at:

$$\begin{aligned} lightpixel_{horvert} &= (26 \times 0.0065, 13 \times 0.00625) \\ &= (0.169, 0.08125) \text{ mm} \end{aligned}$$

The effective element's picture centre referred to in the SONY data sheets, which is the centre of the light sensitive image area, is at:

$$\begin{aligned} CCDcentre_{horvert} &= ((22 + 3 + 752 / 2) \times 0.0065, (12 + 582 / 2) \times 0.00625) \\ &= (2.6065, 1.89375) \text{ mm} \end{aligned}$$

1.1.3 The *uv* coordinate system

The coordinates *u* and *v* are measured in *mm* from the reference point where the optical axis intersects the CCD. It is virtually impossible to align the CCD centre point with the optical axis to the desired accuracy. As long as the exact point of intersection of the optical axis with the CCD can be determined, this perfect alignment is not necessary. The method of determining this point of intersection is described in Chapter 6. It will move each time the CCD PCB has been removed from the lens mount and will also vary from camera to camera. It has to be determined specifically for the camera set-up being used.

The optical centre measured in *horvert* coordinates is given by:

$$optcen_{horvert} = (horoptcen, vertoptcen)$$

and is determined during calibration.

Conversion from the *horvert* to *uv* coordinate system depends on this position:

$$\begin{aligned} u &= horoptcen - hor \\ v &= vert - vertoptcen \end{aligned} \quad (1.3)$$

¹ These functions are used in the Modula software and also referred to in other Chapters.

1.2 Star sensor (ss) coordinate system

This coordinate system is centred in the optical centre of the lens. It is denoted by the subscript ss . The y_{ss} axis coincides with the boresight of the sensor and y_{ss} intersects the CCD plane in the point $(u, v) = (0,0)$. The z_{ss} axis points in the direction of positive v and the x_{ss} axis towards positive u . The distance along the optical axis, between the lens and the CCD plane is the focal length f . The position of a star relative to the lens is directly related to its position on the CCD. The angles ϕ_{ss} and λ_{ss} are related to the coordinates u and v as follows:

$$\phi_{ss} = \arctan\left(\frac{u}{f}\right) \quad (1.4)$$

$$\lambda_{ss} = \arctan\left(\frac{v}{f} \cos \phi_{ss}\right) \quad (1.5)$$

where both angles are smaller than 90° .

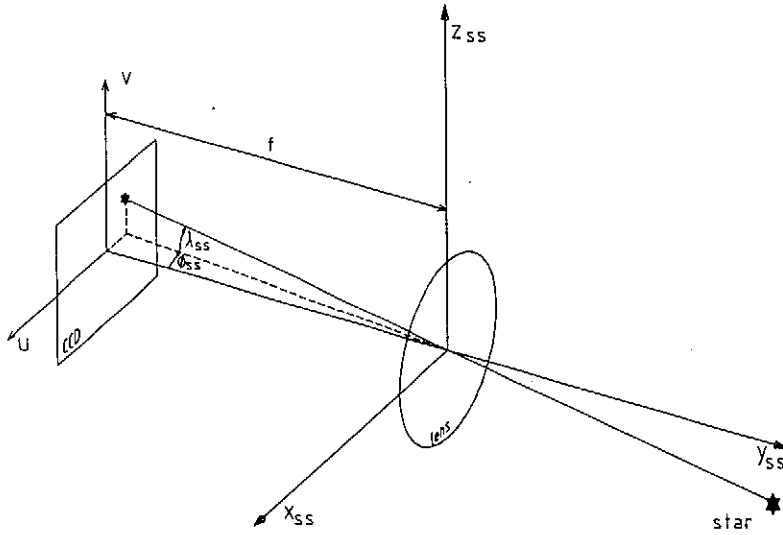


Figure 1-3 Relation between the uv and ss coordinate systems

The three dimensional unit vector in the star sensor coordinate system is then:

$$\hat{\underline{s}}_{ss} = \begin{bmatrix} -\sin \phi_{ss} \cos \lambda_{ss} \\ \cos \phi_{ss} \cos \lambda_{ss} \\ -\sin \lambda_{ss} \end{bmatrix} \quad (1.6)$$

The function *uvto ss* (uv coordinate system to ss coordinate system) uses this equation to map a point given in uv coordinates on the CCD plane into the three dimensional ss

coordinate system. Its inverse function *sstouv* maps a point given in the *ss* coordinate system onto the CCD plane with:

$$\phi_{ss} = \arctan\left(-\frac{x_{ss}}{y_{ss}}\right) \quad (1.7)$$

$$\lambda_{ss} = \arcsin(-z_{ss}) \quad (1.8)$$

$$\underline{S}_{uv} = \begin{bmatrix} -f \frac{x_{ss}}{y_{ss}} \\ f \frac{\tan \lambda_{ss}}{\cos \phi_{ss}} \end{bmatrix} \quad (1.9)$$

1.3 Satellite body (*b*) coordinate system

The satellite body coordinate system is centred in the satellite's centre of gravity. It is denoted by the subscript *b*. The positive z_b axis is parallel to the gravity boom but opposite to its direction of deployment. The x_b and y_b axes are perpendicular to the side solar panels.

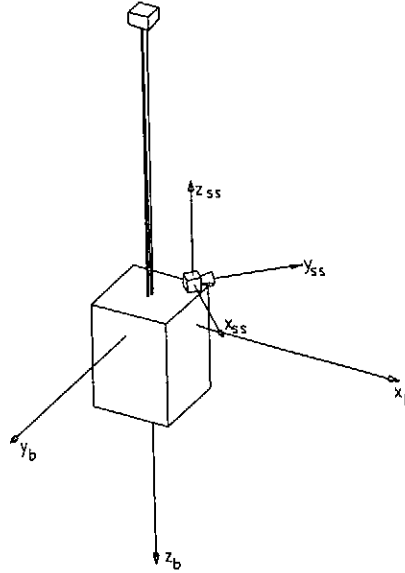


Figure 1-4 Star sensor in the satellite body coordinate system

The star sensor coordinate system has translation and rotation relative to the satellite body coordinate system. The translation has a negligible influence on coordinate transformations and is ignored. The boresight direction of the star sensor makes an angle of -45° with the x_b axis so that it passes between the x_b and $-y_b$ axes. An Euler 1-2-3 rotation is used to convert from the star sensor coordinate system to the body coordinate system. Rotation about the 1 or x axis is ϕ , about the 2 or y axis is θ and about the 3 or z axis is ψ . The three right hand positive rotations comprising an Euler

1-2-3 sequence are shown in Figure 1-5. The coordinate system without subscripts is transformed into coordinate system '2'. The corresponding transformation matrices, which are the direction cosine matrices for each of the rotations, are:

$$A_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \quad (1.10)$$

$$A_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (1.11)$$

$$A_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.12)$$

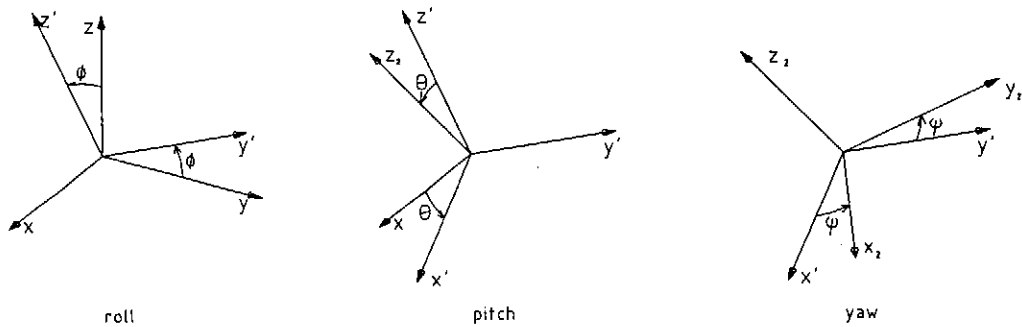


Figure 1-5 An Euler 1-2-3 rotation

The complete transformation matrix is the product of the three individual direction cosine matrices:

$$\begin{aligned} A(\phi, \theta, \psi) &= A_\psi A_\theta A_\phi \\ &= \begin{bmatrix} C\psi C\theta & C\psi S\theta S\phi + S\psi C\theta & -C\psi S\theta C\phi + S\psi S\theta \\ -S\psi C\theta & -S\psi S\theta S\phi + C\psi C\theta & S\psi S\theta C\phi + C\psi S\theta \\ S\theta & -C\theta S\phi & C\theta C\phi \end{bmatrix} \end{aligned} \quad (1.13)$$

with

$$C = \cos$$

$$S = \sin$$

Only two rotations are required to align the star sensor coordinate system with the body coordinate system: a rotation of $\phi = 180^\circ$ about the x axis and then $\psi = -45^\circ$ about the z axis. The general transformation matrix then simplifies to:

$$A_{ss \rightarrow b} = \begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.14)$$

Any vector in the star sensor coordinate system may be transformed to the satellite body coordinate system. The conversion is achieved by multiplication with the transformation matrix:

$$\underline{\hat{S}}_b = A_{ss \rightarrow b} \underline{\hat{S}}_{ss} \quad (1.15)$$

The inverse transformation is achieved by multiplication with the inverse of the transformation matrix. Since it is an orthogonal matrix, the transpose of the transformation matrix is also equal to its inverse. This greatly simplifies the calculation of inverse matrices.

1.4 The orbit (o) coordinate system

This coordinate system is denoted by the subscript o . The popular RPY (roll, pitch, yaw) system is used, where the axes always have the same orientation with respect to the earth, as shown in Figure 1-6. The yaw axis is denoted by z_o and is nadir or earth pointing. The pitch or y_o axis is directed opposite to the orbit normal. The roll or x_o axis completes the set of three right hand positive axes. In the case of a circular orbit it points in the direction of the velocity vector.

The three significant orbit parameters, also shown in Figure 1-6, are:

The right ascension of the ascending node is denoted by Ω . It is the right ascension, measured eastward from the First Point of Aries² to the point where the upward or North going orbit crosses the celestial equator. Vernal Equinox defined at epoch 1950 is used as reference.

The inclination, i , of the orbit describes the angle measured from the plane of the celestial equator northwards to the plane of the orbit. The plane of the celestial equator is in the plane of the earth's equator. The inclination of the Sunsat orbit is fixed at 97° .

The argument of perigee defines the angular rotation of the elliptical orbit within the orbit plane. The true anomaly describes the distance that the satellite has travelled

² Refer to Appendix A

from the ascending node towards perigee. The sum of these two angles is of significance to the star sensor coordinate rotations and is denoted by ω' .

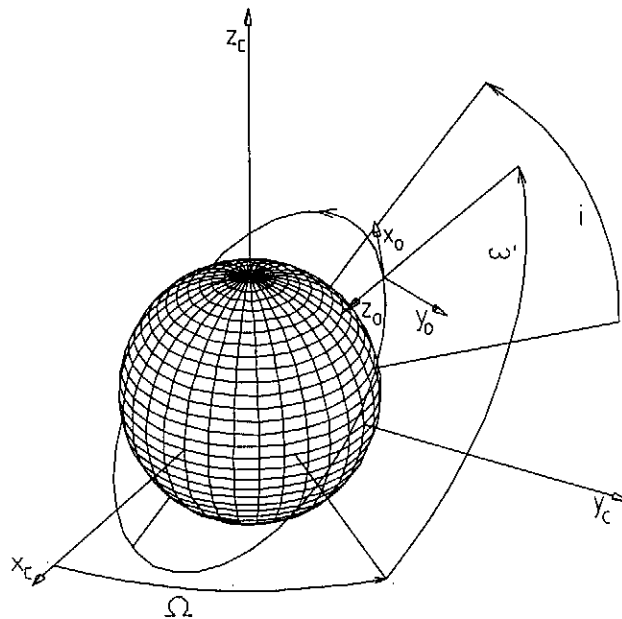


Figure 1-6 Definition of the orbit coordinate system in the celestial coordinate system

An Euler 1-2-3 rotation, as defined in Section 1.3, is used to describe the rotation of the satellite body in the orbit coordinate system.

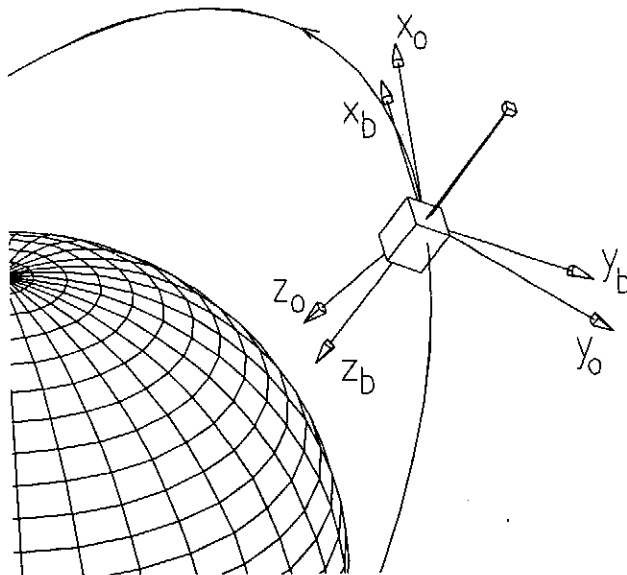


Figure 1-7 The satellite body in the orbit coordinate system

The matrix for transformations from the orbit to body coordinate systems is identical to that given in equation (1.13):

$$A_{o \rightarrow b} = A(\phi_{o \rightarrow b}, \theta_{o \rightarrow b}, \psi_{o \rightarrow b}) \quad (1.16)$$

A vector in the body coordinate system is transformed to the orbit coordinate system through multiplication by this matrix:

$$\hat{\underline{S}}_b = A_{o \rightarrow b} \hat{\underline{S}}_o \quad (1.17)$$

The inverse transformation is achieved by multiplication with the inverse or transpose of this matrix.

1.5 The celestial (c) coordinate system

The celestial coordinate system's centre coincides with the centre of the earth. It is denoted by the subscript c . The z_c axis is directed towards the celestial North pole. The x_c axis points towards the First Point of Aries and y_c is chosen to complete the right hand positive set.

When transforming from the orbit to celestial coordinate system, the translation of the base points of the two coordinate systems is neglected. Parallax of even the closest stars, due to motion of the satellite around the earth, is negligible. Once again only the rotation of the one system with respect to the other is considered. The orbit parameters are used to describe the position of the satellite and its orbit relative to the celestial coordinate system.

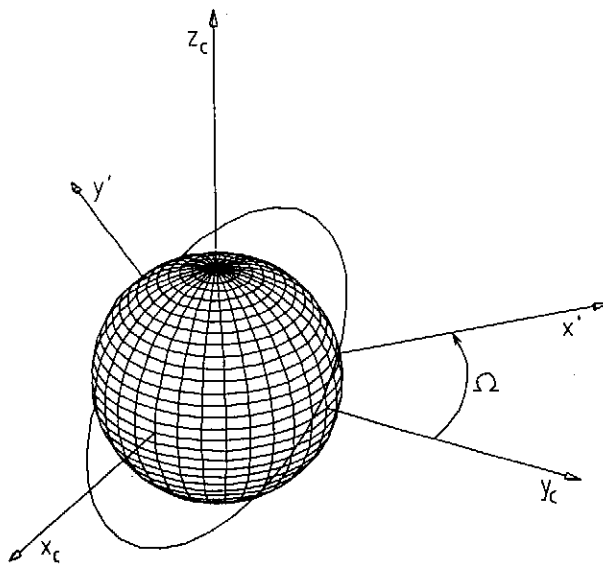


Figure 1-8 (x', y', z') after a rotation of $90^\circ + \Omega$ about z_c

To obtain the transformation from the orbit to the celestial coordinate system, four rotations are required. To describe the transformation, the four steps required to rotate the celestial coordinate system to obtain the orbit coordinate system are shown. This can be achieved with a variety of combinations of rotations, but only the sequence used in the software algorithms is shown in Figures 1-8 to 1-11.

Firstly, the celestial coordinate system is rotated (Figure 1-8) about the z_c axis by $\phi_{c \rightarrow o} = 90^\circ + \Omega$, which gives the intermediate (x', y', z_c) coordinate system. This is followed by a rotation (Figure 1-9) about the y' axis of $\psi_{c \rightarrow o} = -i$.

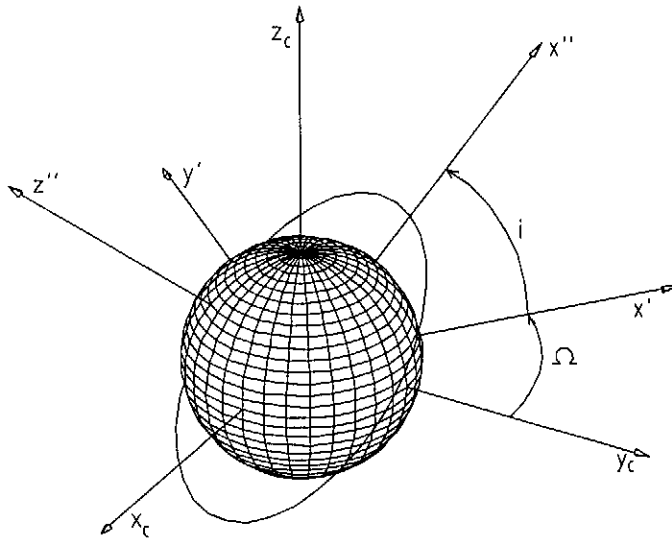


Figure 1-9 (x'', y', z'') after a rotation of $-i$ about y'

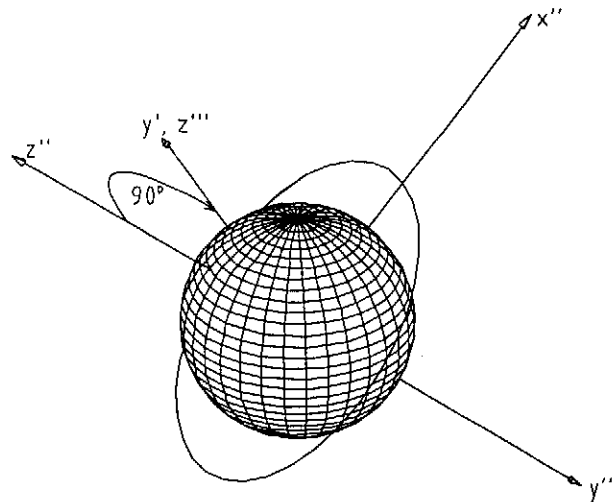


Figure 1-10 (x'', y''', z''') after a rotation of -90° about x''

This (x'', y', z'') coordinate system is rotated (Figure 1-10) about x'' by $\theta_{c \rightarrow o} = -90^\circ$ and finally this is rotated (Figure 1-11) about y''' by $\psi_{2c \rightarrow o} = \omega'$

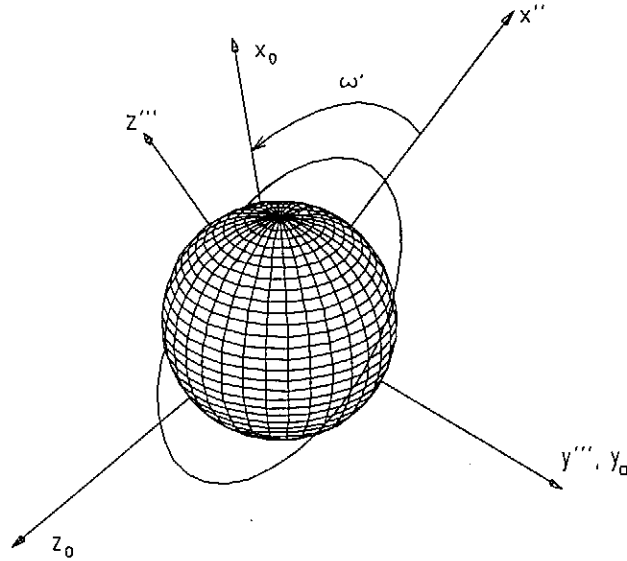


Figure 1-11 The *orbit* coordinate system reached after a rotation of ω' about y'''

to obtain the *orbit* coordinate system. The translation required to move this coordinate system to the orbit position shown in Figure 1-6 has a negligible effect on the transformations and is ignored in computations.

The complete transformation is once again the product of the individual matrices defined in equations (1.10) to (1.12):

$$A_{c \rightarrow o} = A_{\psi_{2c \rightarrow o}} A_{\theta_{c \rightarrow o}} A_{\psi_{c \rightarrow o}} A_{\phi_{c \rightarrow o}} \quad (1.18)$$

and the reverse transformation is accomplished by multiplication with the inverse of this matrix,

$$A_{o \rightarrow c} = A_{c \rightarrow o}^T \quad (1.19)$$

So that any vector in the orbit coordinate system can be mapped to the celestial coordinate system using

$$\underline{\hat{S}}_c = A_{o \rightarrow c} \underline{\hat{S}}_o \quad (1.20)$$

1.6 The celestial sphere (cs) coordinate system

Star catalogues contain the stars' right ascension and declination on the celestial sphere. To explain the celestial sphere concept³, all stars in the universe are visualised as being on a sphere of unit radius, centred at the observer. Since it has unit radius, any point on the sphere is fully defined by only two coordinates. The function *ctocs* maps a point from the three dimensional celestial coordinate system onto the two dimensional celestial sphere. The standard spherical to Cartesian transformation, with unit radius, is used:

$$\begin{aligned}x_c &= \cos\delta \cos\alpha \\y_c &= \cos\delta \sin\alpha \\z_c &= \sin\delta\end{aligned}\tag{1.21}$$

where

α = right ascension (RA)

δ = declination (Dec)

and the reverse transformation is achieved with the function *cstoc*:

$$\alpha = \begin{cases} \arctan \frac{y_c}{x_c} & : x_c \geq 0 \\ \pi + \arctan \frac{y_c}{x_c} & : x_c < 0 \end{cases}\tag{1.22}$$

$$\delta = \arcsin(z_c)$$

1.7 The complete transformation

The complete transformation from the star sensor to the celestial coordinate system is obtained by matrix multiplication of the individual transformation matrices:

$$A_{ss \rightarrow c} = A_{o \rightarrow c} A_{b \rightarrow o} A_{ss \rightarrow b}\tag{1.23}$$

and the sequence

$$\begin{aligned}\hat{\underline{S}}_{ss} &= uvtoSS(\underline{S}_{uv}) \\ \hat{\underline{S}}_c &= A_{ss \rightarrow c} \hat{\underline{S}}_{ss} \\ \underline{S}_{cs} &= ctocs(\hat{\underline{S}}_c)\end{aligned}\tag{1.24}$$

³ Refer to Appendices A and B

maps a point from the CCD plane onto the celestial sphere (star catalogue). The reverse transformation is accomplished by multiplication with the inverse matrix

$$A_{c \rightarrow ss} = A_{ss \rightarrow c}^T \quad (1.25)$$

and the sequence

$$\begin{aligned} \underline{\hat{S}}_c &= cstoc(\underline{S}_{cs}) \\ \underline{\hat{S}}_{ss} &= A_{c \rightarrow ss} \underline{\hat{S}}_c \\ \underline{S}_{uv} &= sstouv(\underline{\hat{S}}_{ss}) \end{aligned} \quad (1.26)$$

These transformations are used extensively in the software discussed in the following five Chapters.

2. Software

Software for the star sensor is grouped into three categories

- on board software
- ground support software
- calibration and testing

The first section of this chapter summarizes the software developed for the star sensor.

2.1 Software summary

On board software includes all programmes that will be executed from the star sensor's Flash RAM, while the satellite is in orbit. Two main programmes are run on the star sensor. The *Matching* algorithm finds any imaged constellation in the star catalogue without a-priori attitude information, using the van Bezooijen [3] algorithm. The *Tracking* programme assumes that an initial estimate of the attitude is available and updates the attitude information every second. Chapters 2 to 4 are devoted to on board software.

Ground support software for the star sensor is discussed in Chapter 5. These programmes assist users of the star sensor and include software which creates a sub catalogue from the entire star catalogue and shows the positions of the field of view (FOV) on the celestial sphere.

Chapter 6 explains the calibration software which is essential for calibrating each camera before launch.

Modula is the standard software language used on Sunsat. To comply with this standard, most star sensor software was developed and tested in Topspeed Modula. The final Topspeed version was converted to Rowley Parallel Modula 2 for the transputer. Developing the software in Topspeed allowed the use of its very powerful debugger and compiler environment. Rowley Modula is compiled and linked from the command line and debugging is done the primitive way by displaying the necessary variables on screen or writing them to a file by means of the *InOut* library. The afserver programme [4, p. 28] acts as interface between the root transputer and host PC. It enables the root programme to use MS-DOS functions such as file accesses and screen display. When debugging a programme which is running on the star sensor transputer, all debugging information has to be sent to the root transputer via a link.

Modula's syntax is similar to Pascal but it encourages the use of separately compiled modules which are linked into a single executable file. The *Matching* and *Tracking* algorithms discussed in Chapters 3 and 4 are main modules. Both call the *Region Growing* implementation module described in the following section. Its Rowley Modula listing *SSGrow.mod* appears in Appendix I.

2.2 The *Region Growing* Module

Figure 2-1 shows a typical image, consisting of a black background with a few stars. Between 5 and 15 stars, varying in size between 10 and 300 pixels appear among the $752 \times 582 = 437\,664$ pixels.

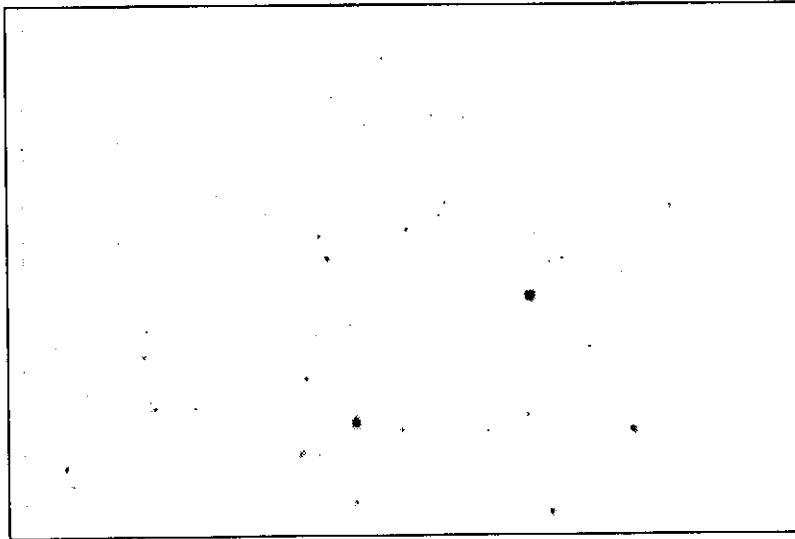


Figure 2-1 A typical image

To the human eye it is a trivial matter to distinguish the stars from the background. A significant amount of effort is required by a computer to do the same, but the result is subpixel accuracies of centroids and reliable magnitude estimates of all the stars.

The region growing algorithm is used to find stars in the image. It was mentioned by Greyling [5]. The basic principle of scanning an image and growing the region around light pixels is discussed by Ballard [6, p. 149-163]. These principles were used to develop the region growing algorithm for the star sensor.

The algorithm searches a specific area of the image, which may be the entire frame or only a fraction thereof. This area is referred to as the search area. The centroids and magnitudes of all stars found in the search area are returned.

2.2.1 Detail description of the region growing algorithm

A top level flow diagram of the region growing algorithm is shown in Figure 2-2.

The first step is to input the parameters defining the area to be searched. The threshold distinguishing a light pixel from the dark background is determined as discussed in Section 2.2.1.2. Then the search area is scanned (refer to Section 2.2.1.3) to find pixels brighter than the threshold. Once one has been found, the region surrounding it is checked to determine whether it is part of a star. Having scanned the entire region and traced all stars, the module returns the number of stars found (*nofstarsfound*), their centroids (in *uv* coordinates) and magnitudes (in *Mv*).

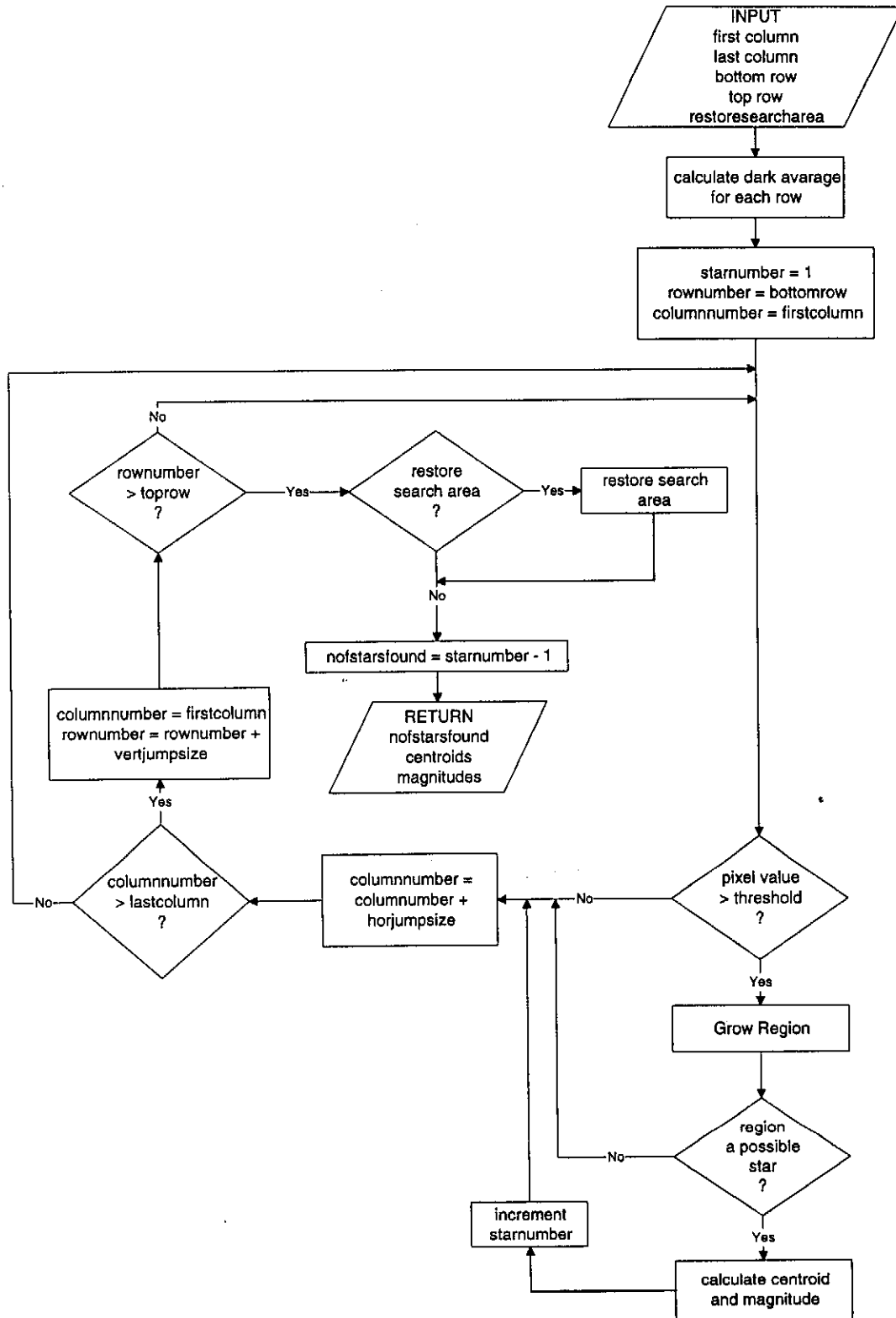


Figure 2-2 Top level flow diagram of the region growing algorithm

2.2.1.1 Input of variables

The module is called with the following parameters:

- The first and last columns and bottom and top rows of the search area. For the *Matching* algorithm these boundaries of the search area are the boundaries of the effective image area of the CCD. For the *Tracking* algorithm only a small section of the image is searched. The boundaries are a function of the estimated position and radius of the expected star.
- A flag to determine whether the searched area should be restored after all stars have been identified. By nature of the region growing algorithm, it erases all pixels detected from the image. If the Boolean value passed is *TRUE*, the entire area searched for stars will be restored on completion of the search. For the *Tracking* algorithm it is essential that the searched area be restored, as emphasized in Section 4.1.4.1. For the *Matching* algorithm the parameter is passed as *FALSE* since it is not necessary to restore the entire image once all the stars have been extracted.
- The previous and present orbit to body transformation matrices. These are used when verifying the shape of possible stars. It is only used by the *Tracking* algorithm since no orientation history exists when the *Matching* module is used.
- A flag which determines whether valid transformation matrices are being passed. This will only be set to true when the *Tracking* algorithm is used.

2.2.1.2 Determining the threshold

The image read from the CCD consists of a stream of analogue data representing consecutive pixels. Pixels are digitized to six bits (i.e. 64 scales of grey) and stored in the image memory. The bottom left pixel of the odd field¹ is the first to be clocked out and the top right one of the even field the last. The entire odd field is clocked into memory before the even field.

The first three and last forty columns of the frame are dark columns, as shown in Figure 1-1. These dark pixels are shielded from light completely. The dark average for each row of the image is calculated by averaging the values of all 43 dark pixels. Dark rows leading and lagging the frame are also available. Due to the ac coupled method used in the video amplifier, the dc offset of pixel values drops slowly during the readout of the image. Using the dark lines to calculate an average dark value for the entire image does not give the accuracy which is obtained by calculating an independent dark average for each line.

Background light in the image² may raise the threshold of detection above the dark average. In other words, effective pixels which are not part of detectable stars, have a value which is higher³ than that of the dark pixels (which are shielded from light completely). The difference between the dark average and threshold of detection

¹ Refer to Appendix G.3

² Due to millions of faint stars and reflected light

³ Refer to Appendix C.1 for details

varies, for example, when imaging with full moon and no moon. Any pixel brighter than the threshold may be part of a star and is defined as a light pixel, while a pixel equal to or below the threshold represents dark space and is defined as a dark pixel. Due to noise inherent to the system, it is impossible to reliably identify one pixel as a star. A star has to be spread over a specified minimum number of pixels. The minimum star size was determined by examining images taken with the star sensor. If the minimum size is selected too small, the probability of mistaking noise as a star is increased. If the minimum size is set too large, all the faint stars will be discarded as noise. As a compromise, the minimum star size was selected as 10 pixels. Only the very bright stars cover more than 300 pixels, in which case the centre pixels start saturating. The maximum star size was selected as 300 pixels.

2.2.1.3 Scanning the search area for stars

The search area is scanned from bottom left until the first pixel brighter than the threshold is found. Since the minimum size of a star is 10 pixels, each pixel does not have to be checked. A star of 10 pixels will have a diameter of at least 3 pixels, which means that only every third pixel needs to be checked.

The distance which the algorithm jumps from checked pixel to checked pixel is referred to as the jump size of the algorithm. The horizontal jump size (*horjumpsize*) is the number of columns jumped and the vertical jump size (*vertjumpsize*) the number of rows. Both variables are equal to three. In other words only one ninth of the total number of pixels in the search area are checked. When scanning the complete image, only 48 629 of the 437 664 pixels are checked. Once a light pixel has been identified, the region surrounding it is grown in an attempt to find a star. This growing procedure is shown in Figures 2-3 and 2-4.

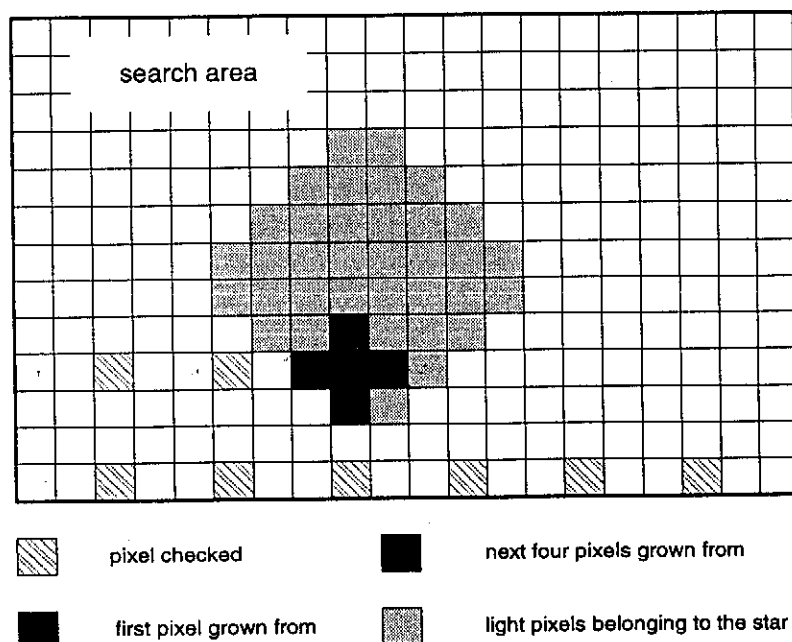
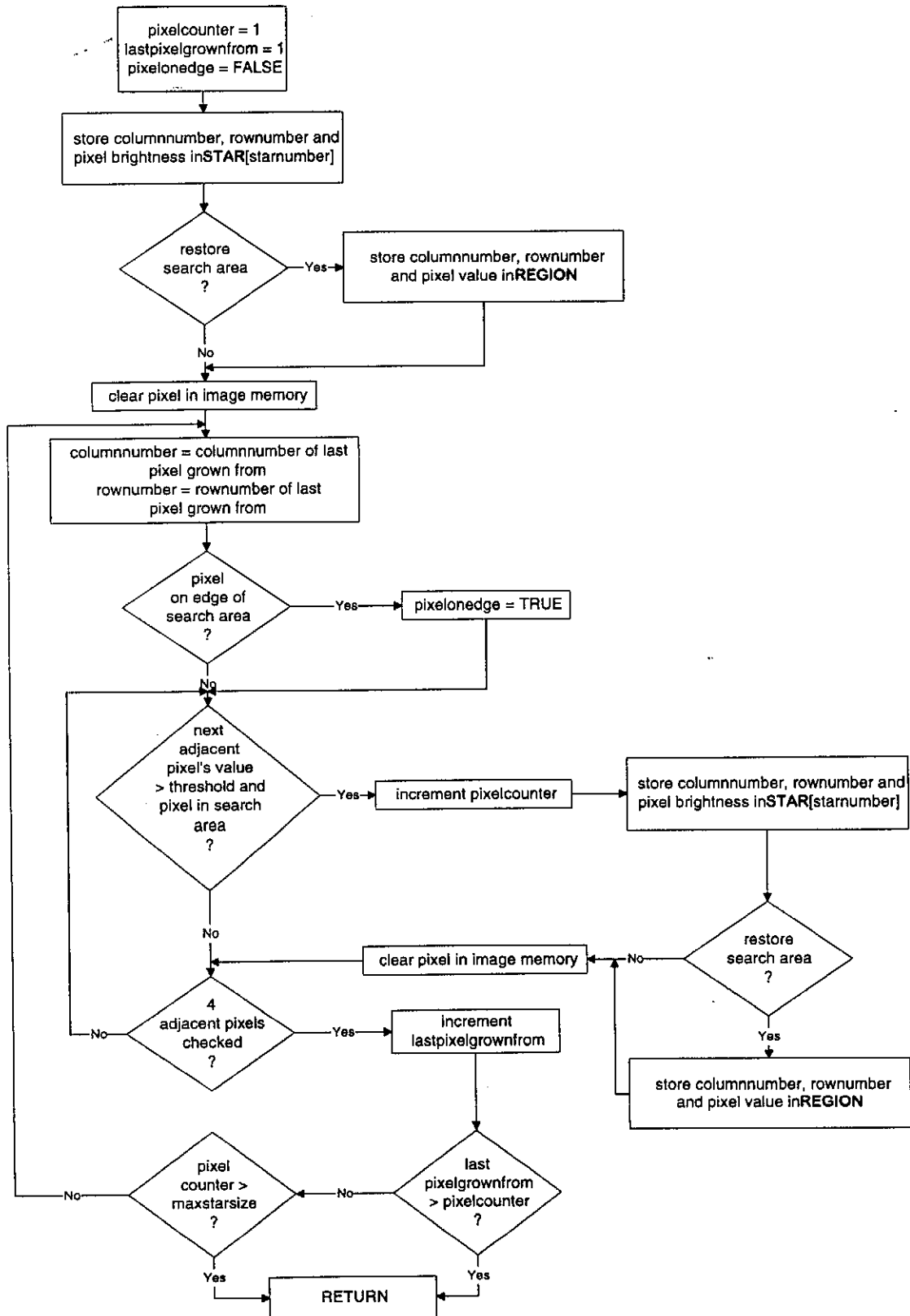


Figure 2-3 Region growing in the image. A jump size of three pixels is used to locate the first light pixel belonging to the star.

Figure 2-4 Flow diagram of the *grow region* block in Figure 2-2

The column number, row number and pixel brightness (i.e. the difference between the pixel value and threshold) of the light pixel are stored in the *star* array. All other light pixels belonging to the potential star will be stored in the same array. A pixel counter (*pixelcounter*) is used as index. The light pixel is erased (made dark) in memory to prevent it from being found again. If the search area has to be restored on completion of the search, each erased pixel is temporarily saved in the *region* array. On completion of the search, it is used to replace all erased pixels.

The four pixels directly adjacent to the light pixel are examined and any light pixel found is erased from memory and added to the potential star (the array *star*). Such a pixel is said to be connected to the first one. The four dark grey pixels in Figure 2-3 are all connected to the black one, which was the first light pixel found. The next step is to grow from all these pixels as well. In other words, the four pixels adjacent to each pixel connected to the first one are checked. This process is continued until no more light pixels are found belonging to the potential star. The last pixel number of which all the surrounding pixels were checked is stored in *lastpixelgrownfrom*. When it exceeds the total number of pixels collected in the *star* array, it means that all four pixels adjacent to each pixel in the region has been checked and no more light pixels were found.

Each potential star found in the image is not necessarily a star. The following criteria have to be met for a region to qualify as a star:

1. The size of the region has to fall within certain limits.
2. Any region bordering the edge of the effective image area is not accepted as a star.
3. The shape should correspond to that of a star.

If the region consists of fewer than the specified minimum number of pixels, the probability exists that it is noise and not a star. If it is larger than the maximum number of pixels, it is possibly a brighter or larger object than a star, such as a planet or a section of the moon.

If a region contains one or more pixels adjacent to the border of the search area, there may be more pixels outside the search area. Such a defective region used as a star will result in an unreliable centroid value. For this reason any region bordering the edge of the search area is discarded. It is important that the region growing continues when a pixel is found on the border. The entire region has to be located and erased from the image. If the growing is stopped once a pixel is located on the edge, the remainder of this region may be found in the next iteration and since the edge pixel has been removed, there is no way of knowing that it was part of another region which extended over the edge.

If criteria 1 and 2 are met, the region's *uv* centroid is calculated. The procedure is discussed in Section 2.2.2. If the shape criteria described in Section 2.2.3 is satisfied, the region is accepted as a star and the magnitude is calculated using the method of Section 2.2.4.

The search is repeated until another light pixel is found from which a new region is grown. Since all pixels belonging to the first region have been erased from memory,

the next region detected will be unique. This process is repeated until the entire search area has been scanned. On completion of the search, the section of the image is restored, if the module was called with *restoresearcharea TRUE*. The *Tracking* module uses the *Region Growing* module to find one star in a small search area. If a section of another star is located within this search area, it will be erased from the image. It is possible that this deformed star's centroid will be calculated at a later stage, leading to an inaccurate value. To avoid this situation, the entire search area is restored to its original state. Restoration is never required when the *Matching* module uses the *Region Growing* module to find all stars in the FOV.

2.2.2 Calculating the centroid

The centroid of a region accepted as a star is calculated with a basic centre of gravity equation, where the weight is the brightness of the pixels.

$$cen_{horvert}[1] = \frac{\sum_{i=1}^{pixelcounter} [(columnnumber[i] - 0.5) \times hunit] \times pixelbrightness[i]}{\sum_{i=1}^{pixelcounter} pixelbrightness[i]} \quad (2.1)$$

$$cen_{horvert}[2] = \frac{\sum_{i=1}^{pixelcounter} [(rownumber[i] - 0.5) \times vunit] \times pixelbrightness[i]}{\sum_{i=1}^{pixelcounter} pixelbrightness[i]} \quad (2.2)$$

where

- $cen_{horvert}[1]$ = horizontal centroid position, in *horizontal-vertical* coordinates
- $cen_{horvert}[2]$ = vertical centroid position, in *horizontal-vertical* coordinates
- $pixelcounter$ = once all pixels in the star have been found, $pixelcounter$ is equal to the total number of pixels in the star
- $hunit$ = horizontal pixel size in *mm* (0.0065 *mm*)
- $vunit$ = vertical pixel size in *mm* (0.00625 *mm*)
- $pixelbrightness$ = $pixelvalue - threshold$

The arrays *columnnumber*, *rownumber* and *pixelbrightness* collectively form the *star* array. They contain the column and row numbers and brightness of each pixel in the star. The subtraction of 0.5 and multiplication by pixel size gives the pixel's centre measured from the reference point of the *horizontal-vertical* coordinate system defined in Chapter 1, in *mm*. The horizontal-vertical coordinates are transformed to *uv* coordinates with equation (1.3).

It is useful to have some idea of the accuracy of the centroid calculated with (2.1) and (2.2). Many articles have appeared which examine the field of subpixel precision during imaging. Havelock [7] developed equations for the bounds on subpixel precision in noise free images. His example [7, pp. 1067, 1068] illustrates the

maximum positional error of a Gaussian shaped dot on a binary image. The significant results are summarized here:

The number of quantization bits per pixel strongly affect the precision. A lineal estimate of the lower bound on the rms error, in pixels, is

$$\varepsilon_{rms} \geq \frac{1}{12[A]} \quad (2.3)$$

where

A = centre amplitude of the Gaussian dot, in units of quantization levels

$[A]$ = integer truncation of the amplitude, i.e. the number of quantization levels covered by the dot

The lineal bound on the error, as well as the bound for the centroiding algorithm [7, p.1073] have been reproduced in Figure 2-5.

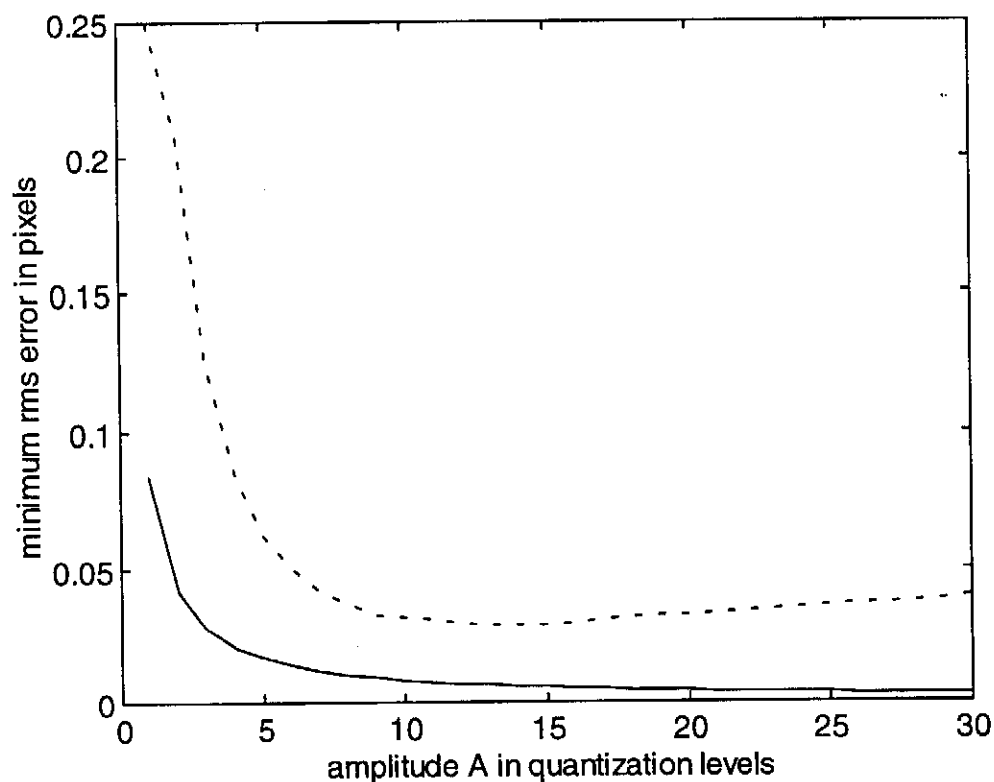


Figure 2-5 Lower bound on the error expected using:

solid line : the lineal bound

dotted line : the centroiding algorithm

The lineal bound's estimate is not very conservative. From Figure 2-5 it is concluded that the centroid of a star with an amplitude of one quantization level will not be

estimated with an accuracy better than a quarter of a pixel. Since the minimum star size is 10 pixels, the amplitude of the faintest stars should exceed one quantization level. Centroids of the faintest stars should be estimated with an accuracy of at least 0.2 pixels. This does not mean that angular distance between two stars can be measured with the same accuracy. Other factors affecting the angular precision are discussed in Chapter 6. Ideally the centre amplitude A of the star should be between 7 and about 30 quantization levels.

2.2.3 The shape of a region

An imaged star's brightness has a Gaussian distribution around the centroid. The brightness distribution function of bright stars may be checked, but it is impractical doing so for faint stars. Their pixels' brightnesses may be only one or two quantization levels above the threshold, which makes it impossible to determine the distribution function properly. The two dimensional shape of faint and bright stars are checked with the procedure described in this section.

When the satellite is rotating slowly, the stars will appear elongated. Since the radius of a faint star is smaller, the affect of the elongation will be greater.

The first step is to determine the longest axis of the region. The procedure described by Ballard [6, p. 255] uses the second moments of inertia of the region to determine the direction of the long axis. Figure 2-6 shows that the direction is defined as the angle between the long axis and the positive direction of the u axis.

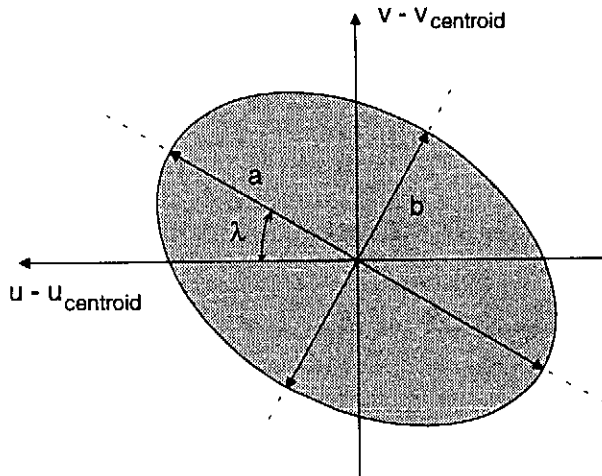


Figure 2-6 The long axis a , short axis b and direction λ of a region

The ij 'th moment is defined [6, p. 255 equation (8.18)] as

$$M_{ij} = \sum_{\text{all pixels}} (u_o - u)^i (v_o - v)^j \quad (2.4)$$

Only the moments around the principle axes are used here:

$$M_{20} = \sum_{\text{all pixels}} (u_0 - u)^2 \quad (2.5)$$

$$M_{02} = \sum_{\text{all pixels}} (v_0 - v)^2 \quad (2.6)$$

$$M_{11} = \sum_{\text{all pixels}} (u_0 - u)(v_0 - v) \quad (2.7)$$

Due to lens aberrations, the centroid calculated with equations (2.1) and (2.2) is not necessarily at the centre of the star's base. This phenomena increases as the star moves further from the optical axis. The base of the star remains circular, but the centroid is shifted towards the optical centre. All pixels with values considerably higher than the threshold, appear close to the inside (optic axis side) of the star. Using this centroid in equations (2.5) to (2.7) may lead to undesirable results. Instead, the centre of the base of the star is used. This is calculated with (2.1) and (2.2) as well, but each pixel value (*pixelbrightness[i]*) is set to 1.

The direction of the region is calculated from the moments with [6, p. 255 equation (8.19)]

$$\lambda = 0.5 \operatorname{atan} \frac{2M_{11}}{M_{20} - M_{02}} + n \frac{\pi}{2} \quad (2.8)$$

where

$$n = \begin{cases} 1 & : M_{11} \geq 0 \text{ and } M_{20} < M_{02} \\ -1 & : M_{11} < 0 \text{ and } M_{20} < M_{02} \\ 0 & : \text{otherwise} \end{cases}$$

The next step is to determine the lengths of the long and short axes. The short axis is defined as the axis perpendicular to the long axis, passing through the centroid. The length of the long axis *a* is determined as the distance between the two furthest pixels located along the axis. The length *b* of the short axis is the distance between the furthest pixels located along the perpendicular axis. If the region is round, the ratio *a/b* will be close to unity. *M*₂₀ will be more or less equal to *M*₀₂ so that the direction of the long axis will approach either 45° or -45°. This direction has no physical meaning, but since the star is round it does not need to be used any further.

One of the parameters passed to the *Region Growing* module is a flag which determines whether the direction and shape of the region should be checked. If this flag is set, the expected direction and elongation of the star is estimated to verify the measured values. The *uv* centroid calculated for the region is converted to orbit coordinates. The transformation matrix used, is calculated with the estimate of the present roll, pitch and yaw. These variables are received by the *Region Growing*

module from the *Tracking* algorithm, together with the roll, pitch and yaw of the previous second. The previous orientation is then used to map the centroid back onto the uv plane. The vector from the previous to the present centroid is \underline{d}_{elong} . Its magnitude is a measure of the elongation expected. Section 4.1.6.3 shows that it should not exceed 10 pixels. The angle λ_{est} which this vector makes with the positive u axis is an estimate of the star's direction. If $90^\circ < \lambda_{est} < 270^\circ$ the angle is converted by subtracting 180° . The measured direction λ can then be compared to the estimated angle λ_{est} since both vary between 90° and -90° .

The measured elongation and direction of the star is compared to the estimated values to determine whether the region is a star:

If the measured elongation is similar to the estimated value:

$$d_{elong} - 2hunit < a - b < d_{elong} + 2hunit \quad (2.9)$$

where

$hunit = 0.0065 \text{ mm}$, the width of a pixel

the direction of the region may be checked. If the measured and estimated directions correspond as well,

$$|\lambda - \lambda_{est}| < 30^\circ \quad (2.10)$$

the region is accepted as a star. If the elongation is small, it is not sensible to check the direction. The direction is only checked if the elongation exceeds 5 pixels. These specifications minimize the probability that two overlapping stars are mistaken as one star.

If the *Matching* algorithm calls the *Region Growing* module, no attitude history is available to estimate the elongation and direction of the region. If the measured elongation $a - b$ is smaller than the maximum possible elongation, the region is accepted as a star. This maximum elongation is set equal to $10 + 2 = 12$. The 10 is the value expected from Section 4.1.6.3 and the 2 added to compensate for possible inaccuracies.

2.2.4 Calculating the magnitude

The total brightness of the star is the sum of the individual pixel brightnesses, plus compensation for the difference between the threshold of detection and the dark average:

$$brightness_{star} = \sum_{i=1}^{pixelcounter} pixelbrightness[i] + c_{diff} \quad (2.11)$$

where

c_{diff} = compensation⁴ for the difference between the detection threshold and dark average

The magnitude of the star is calculated from its total brightness. The illuminance is directly proportional to measured brightness:

$$E_{star} = k_{illum} \times brightness_{star} \quad (2.12)$$

where

k_{illum} = conversion factor from $brightness_{star}$ to illuminance, determined experimentally⁵

Magnitude in M_v is calculated from the illuminance with

$$M_v = \log_{2.512} \left(\frac{2.65 \times 10^{-6}}{E_{star}} \right) \quad (2.13)$$

which is derived in Appendix C from the equation given by Simon [9, p. 66].

2.2.5 Returning the results

The final step of the algorithm is to return the number of stars found in the search area together with their centroids and magnitudes.

These stars are identified by the *Matching* or *Tracking* algorithm, which are discussed in the following two chapters.

2.3 Testing

The *Region Growing* algorithm was tested extensively using software simulated stars, imaged objects resembling stars and real stars.

The accuracy of estimated magnitude and the measured distances between stars are discussed in Chapter 6. Tests mentioned here were used to determine the search speed and reliability of the *Region Growing* algorithm. It was found to be a fast and accurate method of extracting stars, even in noisy images.

Since the 70 ns Flash RAM for the star sensor was not yet available, the final tests for this thesis were performed on a TR6 transputer board, equipped with a T800. The TR6 board has 100 ns memory and a 5 MHz clock, while the star sensor transputer operates

⁴ Details appear in Appendix C.1

⁵ Refer to Appendix C

with 70 ns memory and a 3.6864 MHz clock. Execution times on the two transputers are thus similar.

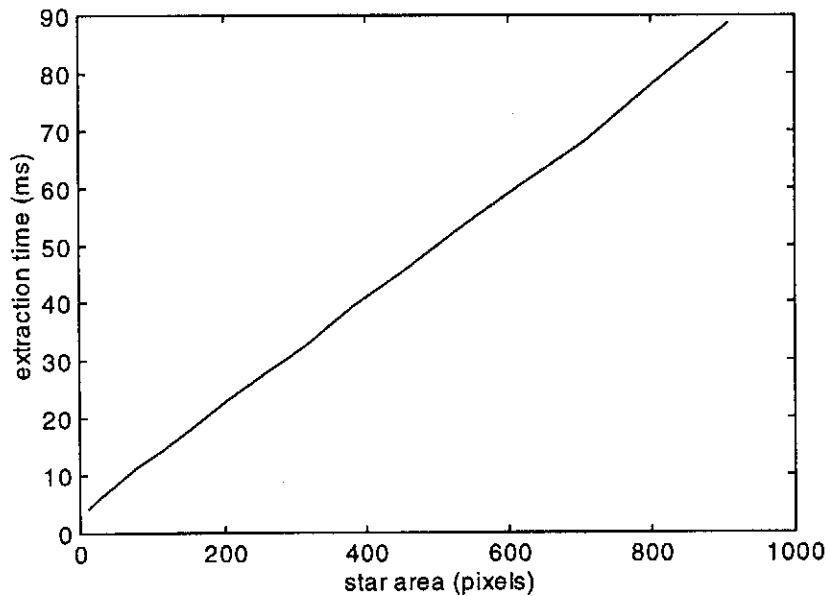


Figure 2-7 Extraction time for a single star

To establish whether the *Tracking* algorithm could detect and match at least three observed stars within one second, it was necessary to determine the average time required to extract one star from the image. This extraction time includes the time taken to locate and grow the star, calculate its centroid, check the shape and estimate the magnitude. The extraction time is function of the star area (in pixels), as shown in Figure 2-7.

The graph shows that even three stars of the maximum star size of 300 pixels can be extracted in under 0.1 s. The significance of these extraction times for the *Tracking* algorithm is discussed in detail in Chapter 4. In the case of the *Matching* algorithm, time is not as critical. In this case all stars in the image are extracted and the brightest ones selected for the matching process. In an extreme case an image will contain a few bright stars and no more than about 40 faint stars. To limit the extraction time, only the first 20 stars found in the image are extracted for the *Matching* algorithm.

2.4 Conclusion

The *Region Growing* module is a fast and reliable algorithm for extracting stars from an image. It operates successfully on noisy images. It should always extract the stars fast enough to ensure that the *Tracking* algorithm can find a match within 1 s. When limiting the number of stars extracted for the *Matching* algorithm to 20, the extraction time should be kept below 1 s. The *Matching* algorithm then selects the brightest of these stars for matching purposes.

3. The Matching Algorithm

This chapter discusses the *Matching* algorithm implemented on the Sunsat star sensor. Its purpose is full three axis attitude determination in circumstances where limited a-priori information is available. The minimum information required are the orbit parameters. It is assumed that the satellite's angular velocity is no more than $0.1^\circ/\text{s}$.

During normal operation of the star sensor, the *Tracking* algorithm will be used in conjunction with the other attitude sensors. In the case where the other sensors fail or are inaccurate, no or limited attitude information will be available. The *Matching* module is designed to determine the attitude under these unfavourable conditions.

A star image is taken and the observed constellation of stars is found in the star catalogue. The observed vectors are mapped from the CCD plane to body coordinates while the matching catalogue star vectors are mapped from the celestial sphere into orbit coordinates. Transformation from body to orbit coordinates involve rotations through the roll, pitch and yaw angles. It is accomplished by multiplication with the orbit to body transformation matrix $A_{o \rightarrow b}$ discussed in Chapter 1. In this instance the two sets of transformed vectors are known while the transformation matrix is required. Various algorithms¹ exist to estimate the matrix. The purpose of the star sensor is, however, to provide the ADCS with the set of matching observed and reference vectors and not to determine the attitude. For this reason the attitude determination will not be discussed in detail here. This Chapter gives details of the actual star matching process, performed by the *Matching* module. The Modula programme *Match.mod* appears in Appendix I.

The *Matching* module is the main programme. It controls² the CCD circuit to take an image, calls the *Region Growing* module to find all stars in the image and then finds the best catalogue match with the van Bezooijen algorithm. The matching vectors are transformed to the appropriate coordinate systems and returned to the ADCS.

3.1 Calling the *Region Growing* module

The image is taken as described in Appendix G and Section 8.5.1. Once the image is in memory, the *Region Growing* implementation module³ is called with the following parameters:

- the corners of the search area are the corners of the effective image area so that the entire image is scanned for stars
- the search area is not restored on completion of the search i.e. *restoresearcharea* is set to *FALSE*
- no attitude history is passed

¹ For example the *Triad* algorithm of Chapter 4

² Refer to Appendix G and Section 8.5.1 for details

³ See Chapter 2

It returns the centroids of all stars found in the image. A constellation is selected from this set. A maximum of eight stars, referred to as the observed stars, are used. The best matching set of stars is found in the catalogue. This is done by means of the van Bezooijen matching algorithm [3]. Its implementation for the Sunsat star sensor is discussed in detail in this chapter.

Detecting only one star in the image is not particularly useful. Identification would rely solely upon the magnitude of the star. Due to the uncertainty associated with the magnitude estimate, it is practically impossible to find a unique match. Even if the match was found, it can not provide any rotational attitude information. Two observed stars facilitate the identification by introducing the distance criteria. This realizes the possibility of finding a unique match in the catalogue. Orientation about all three axes can be derived from this match, except for a 180° uncertainty regarding rotation around the y_{ss} axis. The gravity boom keeps the satellite earth pointing, which resolves this unknown. Ideally three or more observed stars are required, ensuring a high probability of a unique match and providing full attitude information.

3.2 Matching algorithms considered

A wide variety of algorithms have been developed for the purpose of matching observed star constellations to those listed in catalogues. The following sections mention some of these algorithms.

3.2.1 The van Bezooijen algorithm

This algorithm was developed by R W H van Bezooijen [3] at the Jet Propulsion Laboratory, specifically to enable a star sensor to determine its attitude about all three axes without any a-priori attitude knowledge. Its purpose is to find the largest matching group of observed and matching guide (or catalogue) stars. It does so by comparing the observed angular distances separating *each* pair of observed stars to the distances between all pairs of catalogue stars. All possible groups of matching star pairs are considered as the possible best matching constellation. This method minimizes the probability of false stars corrupting the matching process. A false star is any object appearing to the image processing software as a star, which is not listed in the star catalogue. This may be a meteor, a planet, another satellite passing overhead, a UFO or even a freak star such as a variable star or a nova.

Various matching algorithms and combinations of algorithms were examined. Many are capable of finding a match faster than the van Bezooijen algorithm, but all have the disadvantage of not being able to find a match when even one star is a false star. Only the van Bezooijen algorithm is robust against one or more false stars appearing in the image.

3.2.2 Other matching algorithms

The constellation matching algorithm discussed by Baldini [10] contained useful information which has been implemented as modifications to the van Bezooijen algorithm. Figure 3-1 shows that the number of stars in the catalogue increases almost exponentially with magnitude. Selecting brighter stars from the image means fewer possible matches which decreases the probability of an incorrect match. A group of stars from the catalogue is associated with each observed star in the constellation. This group is referred to as a star vector. An example is shown in Table 3-1, where the observed and catalogue stars are sorted in order of decreasing magnitude. Star vector i contains all stars in the catalogue with matching magnitude. Observed⁴ and reference magnitudes are said to match if

$$Mv_{obs} - Mv_{uncert} \leq Mv_{cat} \leq Mv_{obs} + Mv_{uncert} \quad (3.1)$$

where

Mv_{obs} = observed magnitude

Mv_{cat} = catalogue magnitude

Mv_{uncert} = uncertainty in observed magnitude due to measurement, which is determined by hardware

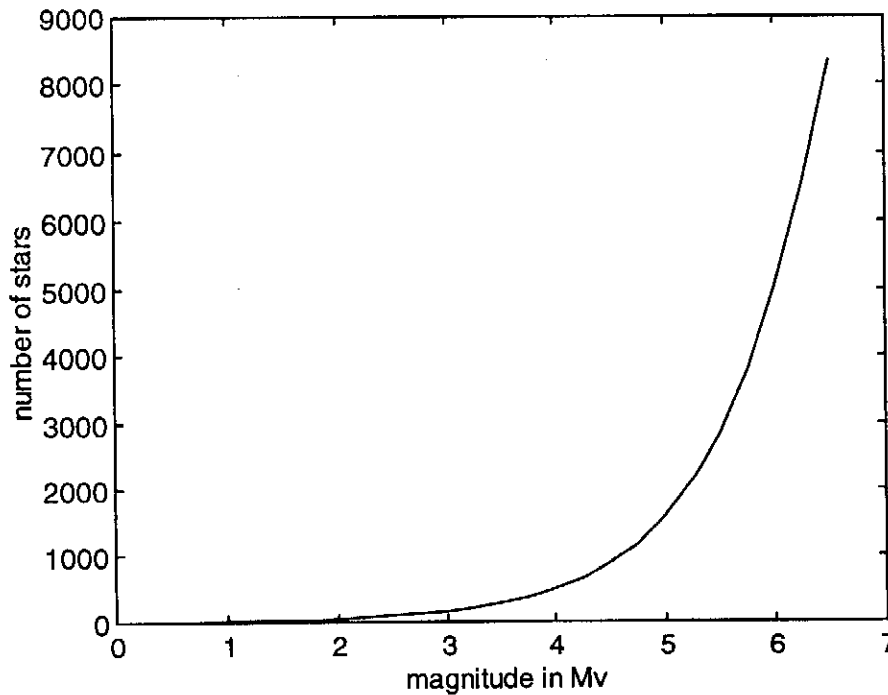


Figure 3-1 Exponential increase in number of stars with increasing magnitude number

⁴ Refer to Appendix C, equation (C.5)

The next step is to check for matching distances. Similar to magnitude matching, the observed and catalogue distances match when

$$d_{obs} - d_{uncert} \leq d_{cat} \leq d_{obs} + d_{uncert} \quad (3.2)$$

where

d_{obs} = angular distance⁵ between two observed stars

d_{cat} = angular distance⁶ between two catalogue stars

d_{uncert} = uncertainty in the observed distance due to measurement errors

Starting with the first entry in vector 1, the distances to all stars in vector 2 are calculated. If none of these distances match the measured distance between observed star 1 and 2, the first star in vector 1 is discarded. The process is repeated for all stars in vector 1. On completion, all stars in vector 2 which never met the distance constraint are discarded as well. The number of stars in vectors 1 and 2 is considerably reduced and the procedure is repeated for all the star vectors.

observed star number	Star vectors				
	1	2	3	4	5
reference stars	33	44	46	66	87
associated with	34	45	47	67	88
the observed star		46	48	68	89
.			49	69	90
.					91
.					92

Table 3-1 A set of five star vectors

Finally constellations are built from the stars remaining in the vectors. Again starting with the first star in vector 1, the distance to the stars in vector 2 are checked. Once a match is found, vector 3 is searched until a star is found for which the distances to the stars in vectors 1 and 2 match. This procedure is iterated. If no match is found while scanning vector 3, the star search is traced back to vector 2. If a match is found in vector 3, the procedure continues to vector 4. Iteration continues until all vectors have been scanned and an entire constellation has been built up. Each time a constellation has been found, it is recorded and the search restarted.

The merits of this procedure are its simplicity and speed of execution. The main drawback is that one false star is enough to ruin the complete algorithm. Assume for example that star 2 is a false star. In other words, no real match exists in the catalogue for this star and its vector will contain a random list of stars with magnitude close to that estimated for star 2. When scanning vector 2 for a match with stars in vector 1, the correct stars in vector 1 may be discarded while incorrect ones are kept. The same

⁵ Refer to Appendix C, equation (C.7)

⁶ Refer to Appendix C, equation (C.8)

is true for all vectors scanned for a match with star 2. Clearly this leads to the downfall of the algorithm, even though all the other stars are true stars.

It is unacceptable that one false star destroys the algorithm's ability to find the correct match. Consequently this algorithm could not be used, but the star vector concept may be implemented in the van Bezooijen algorithm to minimize memory use.

The approach followed by Königsmann [11] contains no novel ideas and is based on the same principle as the previous algorithm. Instead of using the star vector concept to minimize calculations and search time, the entire catalogue is searched to find a match for each star. This algorithm also fails hopelessly when a false star is imaged and its search time is very long.

Kan's [12] work is based on the van Bezooijen algorithm and does not contain much more useful information than [3]. It does however illustrate how the success ratio is seriously affected by the accuracy of imaged star positions, but that magnitude accuracy is not as important. Using as few as five imaged stars, a success rate of practically 100% is achieved when positional accuracy is better than 0.2 pixels. This means that the correct set of catalogue stars is almost always found, from the entire catalogue, to match the observed stars. With a positional accuracy of 1 pixel or better, the success rate of the algorithm is reduced to 50% when the magnitude accuracy is 0.5 Mv and 44% for a magnitude accuracy of 1 Mv. The FOV and CCD size are similar to Sunsat's. The average angle spanned by a pixel is $5.15^\circ/288 = 0.018^\circ$ (FOV size/number of pixels) compared to the Sunsat Star Sensor's $11^\circ/584 = 0.019^\circ$. Since these values are almost identical, the above results give a fair estimate of how the Sunsat star sensor's performance will be affected by positional accuracy of stars. This article highlighted the importance of precise hardware during the period when CCD's were being selected.

3.3 Detail description of the *van Bezooijen* Algorithm

The aim of the algorithm is to find the best match for a set of observed stars from a set of guide stars. The guide stars appear in the uncertainty area of the star catalogue. The uncertainty area is that section of the catalogue where the observed stars are expected. When absolutely no a-priori attitude information is available, the uncertainty area is the entire catalogue. This will only be the case if the gravity boom does not deploy, in which case the satellite will most probably be spinning wildly as well. The very bright stars will appear as lines in the star image while the fainter ones will be spread across so many pixels that they will be indistinguishable from the background. With the gravity boom deployed and the orbit parameters known, the uncertainty area will only be a fraction of the catalogue. Assuming that any one of the other attitude sensors are operational, the uncertainty area will be slightly larger than the size of the FOV (field of view). In this case the *Tracking* algorithm will be used. In other words, the van Bezooijen algorithm will only be employed if no other sensors are capable of providing attitude information of the desired accuracy to serve as initial estimate for the *Tracking* algorithm.

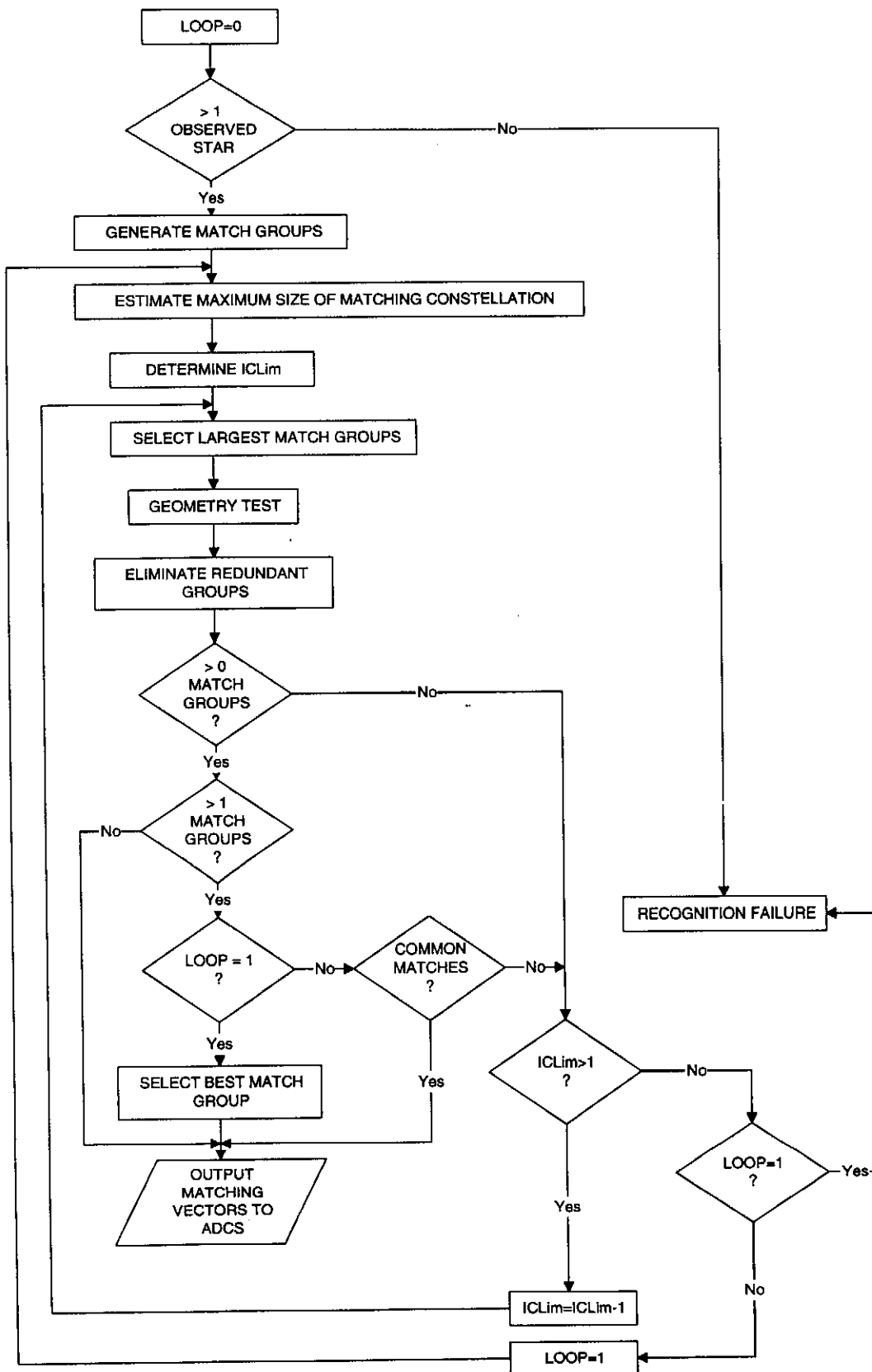


Figure 3-2 Flow diagram of van Bezooijen algorithm

A group of observed stars match a group of guide stars when:

- The magnitude estimated for each observed star matches that of the corresponding guide star. A magnitude match is defined in equation (3.1).
- The angular distance measured between each pair of observed stars in the group matches the distance calculated between the corresponding pair of guide stars. The distances match if they satisfy equation (3.2). The calculation of distances between pairs of observed stars and guide stars is described in Appendix C.
- The observed group is not the mirror image of the guide group.

The matching process consists of five steps:

1. Generation of match groups through star pair matching.
2. Elimination of the smaller match groups.
3. Validation of match groups by checking distances between all stars in the group.
4. Elimination of redundant match groups.
5. Extraction of the largest match group(s).

A flow diagram of the algorithm is given in Figure 3-2. Steps 1 to 4 appear between the first two decisions (diamonds) and are explained in the next sections. Following step five, one large match group usually remains. This is supposed to be the correct group and the algorithm is complete. The rest of the flow diagram describes the routine followed when no valid match or a few matches are found.

If no match is found, the minimum expected matchgroup size (*ICLim*, discussed in the following section) is reduced. If, after successive reductions until its minimum value is reached no match is found, the algorithm has failed to recognize the group. When more than one match groups of the same size are found, a quality criteria is used to select the best group. The differences in distances between all observed star pairs and corresponding guide star pairs are summed and the group with the smallest sum is selected as the matching group.

3.3.1 Generation of match groups through star pair matching

A match group is a selection of guide stars which may be a match for the group of observed stars. The first step in the generation of match groups is finding and tabulating all matching star pairs.

Consider the example illustrated in Figure 3-3, where a group of guide stars have been found to match 6 of the observed stars. The distance $od_{1,2}$ between the first pair of observed stars is measured. The catalogue is scanned for a pair of guide stars with matching distance between them. This is done in one of two ways, either by searching a table containing all the relevant distances or by calculating the distances while searching the catalogue. Both methods are discussed in Appendix D.

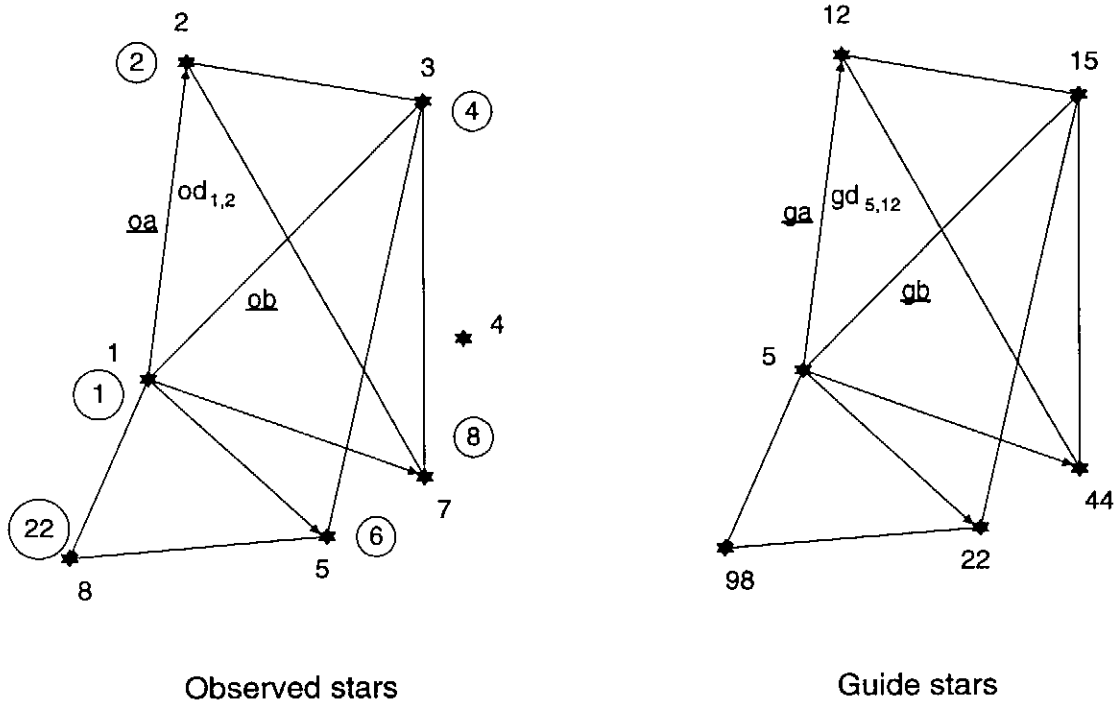


Figure 3-3 Explanation of matching example. Match numbers are encircled and star numbers appear next to the stars.

The guide star pairs which are found to have separations matching those of the observed star pairs are collected in the match vectors. Two match vectors, *matcho* for observed stars and *matchg* for guide stars are used as indicated in Table 3-2.

match no		<i>matcho</i>		<i>matchg</i>
1	ostar1	1	gstar1	5
2	ostar2	2	gstar2	12
3	ostar1	1	gstar1	5
4	ostar2	3	gstar2	15
5	ostar1	1	gstar1	5
6	ostar2	5	gstar2	22
.		.		.
.		.		.
21	ostar1	5	gstar1	22
22	ostar2	8	gstar2	98
		.		.
		.		.
		.		.

Table 3-2 The match vectors

Once a pair of guide stars (reference numbers 5 and 12 in the catalogue) have been found so that their separation matches the separation between observed stars 1 and 2, they are entered as the first two rows of *matchg*. The following notation is used: the

guide star matching the observed star in row i of *matcho* is entered in row i of *matchg*. Similarly, the guide star matching the observed star in row $i+1$ of *matcho* is entered in row $i+1$ of *matchg*. Each two entries in *matcho* form a pair with the corresponding two entries in *matchg*. Observed star *ostar1* matches guide star *gstar1* while *ostar2* matches *gstar2*.

All guide star pairs matching observed pair 1-2 are tabulated⁷. Once all matching pairs have been found, the match groups are created. Each unique⁸ match (of an observed star with a guide star) gets the opportunity to be the kernel match of a match group. In the example of Figure 3-3, match 1 in Table 3-2 (i.e. observed star 1 and guide star 5) is the kernel. The kernel match is the centre of the match group and all other matches are referred to as associated matches.

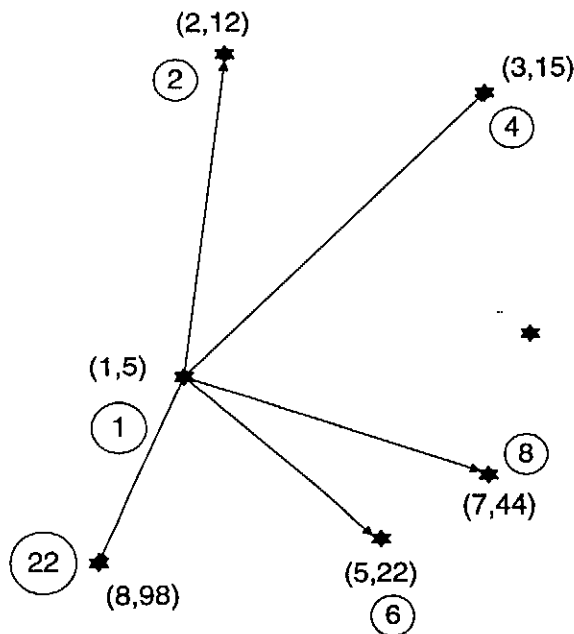


Figure 3-4 Generation of a match group with observed star 1 as kernel. The matching pairs are indicated in parenthesis.

Having identified the kernel of a match group, all associated matches are collected by scanning the match vectors *matcho* and *matchg*. For this trivial example it is a simple matter of selecting every pair (*ostar2*, *gstar2*) in Table 3-2 where *ostar1* = 1 and *gstar1* = 5. When more matches exist, the collection of associated matches becomes quite involved. A more complex example as well as the method used in *Match.mod* for building the match group vectors are covered in Appendix D.

⁷ Note that Table 3-2 only shows those matches indicated in Figure 3-3, while typically many more would be found

⁸ From Table 3-2 it is clear that many duplicate matches may appear in the match vector, but a match group is only generated for each unique match

The notation used for the match group vector *matchgroup*⁹ is indicated in Table 3-3

- the first column contains the kernel match
- the next seven hold the associated matches
- the last column is used for the confidence of the match group

		matchgroup							
group no	Kernel	associated match number						Confidence	
	1	2	3	4	5	6	7		8
1	1	2	4	6	8	22			5
2	2	1	4	8					3
3	4	1	2	8					3
4	8	1	2	4					3
.
.
.

Table 3-3 Configuration of the *matchgroup* vector

The confidence of a match group is the number of matches associated with the kernel. For the example of Figure 3-4, five matches are associated with kernel match 1 so that the confidence of the first group in Table 3-3 is 5.

The procedure is repeated until a matchgroup has been created for each unique combination of matches. This typically amounts to few thousand match groups varying in confidence from 1 to 7, but the majority having low confidence.

3.3.2 Elimination of small match groups

The minimum size of match groups that should remain following this step is estimated. The key to this estimate lies in the fact that for the maximum size of a matching constellation to be n , there needs to be at least n match groups with confirmation value $n-1$. If there are n match groups each containing n matches (i.e. confirmation value is $n-1$) and each of the n match groups contain the same n matches, then this constellation is probably the correct match. To simplify the elimination process the entries of the n match groups are not checked as yet, but only the confirmation value.

The maximum size of match groups to be eliminated is defined as *ICLim* which is set equal to $ICLim = n - 2$. All match groups with *confidence* $\leq ICLim$ are eliminated. The kernels of eliminated matchgroups are removed from all other match groups where they are associated matches.

⁹ Note that due to the method used to generate match groups in the *Match.mod*, all consecutive columns of *matchgroup* will not necessarily be filled as shown in Table 3-3.

3.3.3 Geometry test

All pairs of matching stars in a match group have the kernel match in common. In other words, all distances measured from the guide star of the kernel match to the guide stars of associated matches are within the tolerance from the distances between the corresponding observed stars.

The geometry test checks the distances between stars of the associated matches. The five distances for matches 1-2, 1-4, 1-6, 1-8 and 1-22 in Figure 3-3 are known to be correct. Ten distances remain to be checked. Starting with the first associated match, observed distance $od_{2,3}$ is compared to guide distance $gd_{12,15}$ and is found to match. Once a distance is found which does not match, such as $od_{2,5}$ and $gd_{12,22}$, the original match group is split in two. The one new group contains match 2 (observed star 2 and guide star 12) and all other matches in the original group except match 6 (observed star 5 and guide star 22) while the other contains match 6 together with all the original matches except match 2. After checking all associated matches, the new *matchgroup* vector of Table 3-4 has been created.

group no	matchgroup								Confidence
	Kernel	associated match number							
	1	2	3	4	5	6	7	8	
1	1	2	4	8					3
2	1	4	6						2
3	1	6	22						2
4	2	1	4	8					3
5	4	1	2	8					3
6	8	1	2	4					3
.
.
.

Table 3-4 *matchgroup* after geometrical test

The test is repeated until all match groups have been checked and, where necessary, split.

The last phase of the geometry test is the mirror test. It ensures that the observed and guide constellations are not mirror images of one another. If the cross products $oa \times ob$ and $ga \times gb$ of the vectors in Figure 3-3 have the same sign, the constellations are identical. To determine the cross products, it is necessary decompose the vectors into components. Only the direction and not the magnitude of the cross product is important. For this reason the right ascension and declination components are used. To determine if the signs are the same, the scalar value of the two cross products are multiplied. The geometry test reduces to the following equation:

$$(oa_{RA} \times ob_{Dec} - ob_{RA} \times oa_{Dec})(ga_{RA} \times gb_{Dec} - gb_{RA} \times ga_{Dec}) \geq 0 \quad (3.3)$$

If it is satisfied, the mirror test is passed.

3.3.4 Elimination of redundant match groups

All redundant groups in the match vector are removed. Match groups are redundant if they contain the same set of matches as a previous match group, such as groups 4, 5 and 6 in Table 3-4 which all contain the same matches as group 1.

3.3.5 Extraction of the largest match groups

Usually only one group is the largest and it is assumed to contain the correct matches. If two or more groups remain with at least two matches in common, any one of these groups may be used as the correct match. It is possible that either no match groups or a few match groups without any common matches remain. In this case the limit used during the elimination of smaller match groups (*ICLim*) is assumed to have been too high and is decremented. The algorithm fails if *ICLim* has been reduced to 1 without the observed constellation being recognized.

3.4 Transformation of centroids

The centroids of the matching observed stars are transformed to the orbit coordinate system:

$$\begin{aligned} obsstar[i]_{ss} &= uvtoos(obsstar[i]_{uv}) \\ obsstar[i]_o &= A_{ss \rightarrow o} obsstar[i]_{ss} \end{aligned} \quad (3.4)$$

with $i = 1 \dots \text{number of matching pairs found}$

The positions of the matching reference stars are converted from the star catalogue (celestial sphere) to orbit coordinates:

$$\begin{aligned} refstar[i]_c &= cstoc(refstar[i]_{cs}) \\ refstar[i]_o &= A_{c \rightarrow o} refstar[i]_c \end{aligned} \quad (3.5)$$

The aberration occurring due to the satellite's velocity in inertial space is corrected by shifting the reference vectors in the direction of the velocity. This process is described in detail in Section 7.5.

The set of corresponding observed and reference vectors are passed to the ADCS via the link.

3.5 Verifying and testing the algorithm

The algorithm was initially coded in Toplevel Modula. Its *VID* environment greatly facilitated the testing and debugging of the software. Since this software was developed prior to the availability of any imaging hardware, tests were conducted by selecting stars from the catalogue, calculating their uv coordinates on the CCD plane, intentionally introducing errors and then matching this simulated image to the catalogue.

Using simulated data instead of true images to test the algorithm had several advantages:

- The error in position and magnitude estimates could be varied to examine the effect on the algorithm
- Once the algorithm claimed to have found a match, no doubt existed concerning its validity. The constellation used to create the simulated image was initially selected from the catalogue and the exact stars used are known.
- Thousands of star patterns could be examined with much greater ease than taking such large numbers of images.

The final tests for this thesis were performed on a TR6 transputer board which uses a *T800*. The 70 ns Flash RAM required for the star sensor transputer was not available at the time these tests were performed. The static RAM on the star sensor is not sufficient to store the programme, catalogue, distance table and variables. Consequently the execution times of the algorithm could not be timed on the star sensor transputer. It operates with 70 ns memory and a 3.6864 MHz clock input, while the TR6 card has 100 ns memory and a 5 MHz clock. Execution times on the two transputers are thus similar.

The TR6 card does not have sufficient memory for storing a complete simulated image as well as the abovementioned data. For this reason a simulated image was not created, as was done when testing the *Tracking* algorithm of Chapter 4. The *Region Growing* algorithm was, however tested thoroughly as discussed in Chapters 2 and 4.

It is impractical to perform tests on the entire catalogue, due to the enormous number of entries in the distance table (288464 entries¹⁰ for the full catalogue of 5000 stars). A sub catalogue was generated with *MakeCat.mod*, discussed in Chapter 5. The extreme attitude values are shown in Table 3-5.

The catalogue contains 540 stars and a distance table with 16281 entries. The uncertainty area covers an area larger than one third of that shown in Figure 5-3. The boresight of the star sensor should remain within such an area for three to four months. It includes both sections of the densely populated Magellanic cloud and sparsely populated sections in between.

¹⁰ Refer to Appendix D

	Minimum	Maximum
roll	-10°	10°
pitch	-5°	5°
yaw	130°	140°
ω'	0°	360°
i	97°	97°
Ω	0°	120°

Table 3-5 Extreme values of attitude used for testing.

The two critical aspects to be examined by the tests were the execution time and success rate of the algorithm.

The execution time is affected by various factors, which include

- the 1σ distance measurement error and uncertainty in angular distance measured between stars, d_{uncert}
- the 1σ magnitude measurement error and the uncertainty in measured magnitude, Mv_{uncert}
- the number of stars used for matching
- the catalogue size and dynamic range

The selection of d_{uncert} and Mv_{uncert} is based on the error in distance and magnitude measurement, but they should at least equal or exceed the 1σ measurement errors. Their selection is discussed in the following section. If these values are large, a great number of star pair matches are selected according to equations (3.1) and (3.2). This leads to large arrays and long search times. Keeping these values small minimizes the search time, but increases the probability of recognition failure.

The number of observed stars used for matching also affects the search time. Van Bezooijen [3] found the optimal value to be 8 stars. The catalogue size and star density within the uncertainty area also affects the search time and memory requirements. For this reason the catalogue was limited to 550 stars.

The effect of varying d_{uncert} and Mv_{uncert} was examined by varying d_{uncert} while keeping Mv_{uncert} fixed and then fixing d_{uncert} and varying Mv_{uncert} . The results are shown in Appendix D, Figures D-1 and D-2. The graphs show that as long as d_{uncert} is limited to a few hundred μrad (one pixel is about 230 μrad) and Mv_{uncert} to a few tenths of a magnitude, the average matching times remain under 10 s. The uncertainty values were set equal to the 1σ errors.

The success rate, or probability that the algorithm finds the correct match, is affected by the abovementioned points as well.

A large catalogue includes many more false matching pairs for each observed star pair. It is thus desirable to keep the size of the catalogue to a minimum. The limiting magnitude of stars contained in the catalogue affects the success rate as well. If the dynamic range of the catalogue is too small, the probability of finding fewer than three

observed stars in the FOV is increased. The dynamic range should be large enough to ensure that at least three stars appear in a very high percentage of the images. Chapter 5 examines this further.

If the 1σ measurement errors are minimized¹¹, the success rate of the algorithm is improved. The ratio of the uncertainty in magnitude and distance used in equations (3.1) and (3.2) to the measurement errors (1σ) also affects the success rate. Extending the uncertainty values, increases the success rate but slows the algorithm down considerable. Many more matches are selected, which increases the search time but also increases the probability that a high sigma observed star (an observed star of which the measured magnitude or angular distance has an exceptionally large error) is included in the match groups.

The effect of different values of the measurement errors (1σ) and uncertainty values were examined by means of simulations. The position and magnitude of each observed star was distorted. A normally distributed positional error and magnitude error with standard deviations σ_p and σ_{Mv} (shown in Table 3-6) were added to the centroid and magnitude of each observed star. The catalogue was searched for a match for this distorted constellation. The process was repeated for 1000 random attitude values, within the limits listed in Table 3-5.

σ_p (μ rad)	σ_{Mv} (Mv)	$d_{uncert.}$ (μ rad)	$Mv_{uncert.}$ (Mv)
50	0.3	50	0.3
50	0.3	75	0.45
50	0.3	100	0.6
50	0.3	150	0.6
100	0.3	200	0.45

Table 3-6 Values used to generate the data of Table 3-7

Ave time (s)	Max time (s)	Recognition failure with		Wrong match	Success (%)
		< 3 stars	> 3 stars		
2.7	6.2	4	177	0	81.9
4	15	5	17	0	97.8
8	39	3	9	0	98.8
14	60	2	4	0	99.4
10	73	4	24	0	97.2

Table 3-7 Data obtained for the corresponding rows of Table 3-6

The results for the values listed in each row of Table 3-6 appear in the corresponding row of Table 3-7. Table 3-6 contains the standard deviation of the error in the positions of each star and the uncertainty in the distance measured between two stars.

¹¹ This is a function of the hardware

The average time and maximum time taken to match each of the 1000 constellations is given in Table 3-7. The number of times recognition failures¹² occurred are listed as well. When recognition failure occurred with fewer than 3 observed stars in the image, it is due to the insufficient dynamic range of the sensor. Recognition failure with more than three observed stars in the image only occurs when few stars or a large number of high sigma cases appear in a single image. Due to the exceptionally large positional or magnitude errors, the correct matches are not included in the match group and the algorithm fails to find the stars. The first four rows of the tables show that increasing the distance and magnitude uncertainties, improves the success rate of the algorithm. During all the simulations performed, it never found an incorrect match.

The relatively long search times pose a problem if the *Matching* algorithm needs to update the orientation every second. This is, however, not necessary and the update is performed by the *Tracking* algorithm. In a situation where all other sensors are inaccurate, the *Matching* algorithm will be used to determine the initial orientation estimate for the *Tracking* algorithm. In such a case an image will be taken and the orientation estimated. When the star sensor is to be used, the satellite will never be rotating at a rate of more than $0.1^\circ/\text{s}$ in each axis, or in the worst case about $0.141^\circ + 0.1^\circ = 0.241^\circ/\text{s}$ in total.

The initial orientation estimate required by the *Tracking*¹³ algorithm is at least one or two degrees, but preferable much better. It should find the correct match with even larger initial errors, but to limit the probability of finding the incorrect match, a maximum error in the initial estimate of 2° is allowed. When using the *Matching* algorithm to find an initial orientation estimate, the orientation may change considerably during the period from imaging until the correct match is found. The accuracy of the match will be fractions of a *mrاد*, but since the match was valid the moment the image was taken, an uncertainty in the orientation exists. During the few seconds that elapse since the image was taken until the match is found, denoted by t_{match} , the orientation may change by as much as

$$\Delta\text{orient} = 0.241 \times (t_{\text{match}} + 1) \text{ deg} \quad (3.6)$$

where

Δorient = maximum possible change in orientation

One second is added to t_{match} to represent the maximum time which the *Region Growing* module may take to extract the centroids from the image. When allowing a maximum change in orientation of 2° , t_{match} is solved from equation (3.6) as 7.3 s. It should be emphasized that this is the worst case value. Usually the satellite will not be rotating nearly as fast, and definitely not in all three axis at the same time. If it is rotating at the maximum rate in only one axis, t_{match} increases to 19 s.

The second entry in Table 3-7 demonstrates acceptable average search times. The maximum search time of 15 s does not pose a problem. The *Matching* algorithm will

¹² See Figure 3-2 for a definition

¹³ Refer to Chapter 4 for details

be used a few times in succession, until the matching time is shorter than 7 s. If the satellite is rotating at the maximum rate, a new image will definitely appear in the FOV each time, and by the second or third iteration of the algorithm a match should be found in under 7 s. Each time the *Matching* algorithm finds a match, it is used as initial value for the *Tracking* algorithm. Should it succeed in detecting the correct stars, the *Matching* algorithm is discarded. Otherwise the *Matching* algorithm is run once again. This process is repeated until the *Tracking* algorithm is initiated.

If the satellite is rotating slower than the maximum rate, the change in orientation occurring during the matching period will be less and the *Tracking* algorithm should find an initial match. Due to the gravity boom, the satellite can not continue rotating about all three axes at the maximum rate. At some point the pitch or roll will slow down, come to a halt and change direction. If all else fails, the *Tracking* algorithm should find a match at this stage.

As another option, uncertainty values equal to the 1σ errors are used, as shown in the first rows of Tables 3-6 and 3-7. Matching times are well within the specified limit of 7.3 s, but the matching success rate is only 81.9 %. Due to the fast matching time, the probability that the *Tracking* algorithm will find an initial match is very high. The overall probability of successfully initiating the *Tracking* algorithm should be close to 80 %.

3.6 Conclusion

If the accuracy of all the other sensors on board the satellite does not meet the specifications, the *Matching* algorithm will be used to find an initial orientation estimate for the *Tracking* algorithm.

When selecting the uncertainty values equal to the 1σ errors, the *Tracking* algorithm should be successfully initiated during the first iteration. The probability of successful initiation is close to 80 %. If recognition failure occurs, another iteration of the *Matching* algorithm is executed. The catalogue needs to be updated once every three to four months.

The Matching algorithm also forms part of the ground support software, but is not discussed in Chapter 5. It is used extensively to match images taken during calibration and testing of the star sensor.

4. The Tracking Algorithm

The prime objective of the star sensor is to update the orientation variables (pitch, roll and yaw) at one second intervals. The *Matching* algorithm discussed in Chapter 3 determines the orientation with limited a-priori attitude information. The Tracking algorithm, on the other hand, is optimized to update the orientation variables in the shortest possible time with the minimum calculations and memory accesses. To accomplish this, it utilizes the full attitude information available.

Normally the star sensor will be used in the period prior to earth imaging and during imaging. The angular rate of the satellite should practically be zero during this time. Any stars appearing in the star image will remain there for quite a few frames, either almost stationary on the CCD plane or moving slowly. A maximum angular velocity of $0.1^\circ/\text{s}$ per axis is expected.

One possible method of finding a match for the imaged stars is to use the van Bezooijen algorithm with a very small uncertainty area. The matrix transformations of Chapter 1, the orientation estimate and orbit parameters are used to locate the FOV on the celestial sphere. Only the stars within this FOV are used as the uncertainty area for the van Bezooijen algorithm. This will lead to a fast search, but all the information available is not utilized.

Since the same stars remain in the FOV for a long time, it is not necessary to search for a new matching constellation during each iteration. During the first iteration of the algorithm, the match is based on an attitude estimate obtained from the ADCS. This estimate is used to locate the FOV on the celestial sphere. The reverse transformation is used to estimate the current positions of all reference stars in the FOV, on the CCD plane. An area surrounding the expected centroid of each star is searched. This eliminates the need to scan the entire image memory to find the observed stars. No matching algorithm is required since each observed star found at the expected position is directly matched to the reference star used to locate it. Even though false stars may appear in the image, they will be overlooked since the areas of the image which are searched are based only on catalogue stars.

4.1 Detail description of the Tracking algorithm

The top level flow diagram of Figure 4-1 shows the sequence followed by the algorithm. The first step is to input orbit parameters (right ascension of ascending node, inclination and sum of the argument of perigee and true anomaly) and orientation (pitch, roll and yaw) from the ADCS. This is the orientation of the previous second and was determined by means of the magnetometer and possibly the horizon sensors. The error may be as much as 0.5° per axis. Due to the relatively large error, a large area on the CCD plane has to be searched for each expected star. The expected error on the CCD plane is discussed in Section 4.1.6.2 and the searching procedure in Section 4.1.4.1.

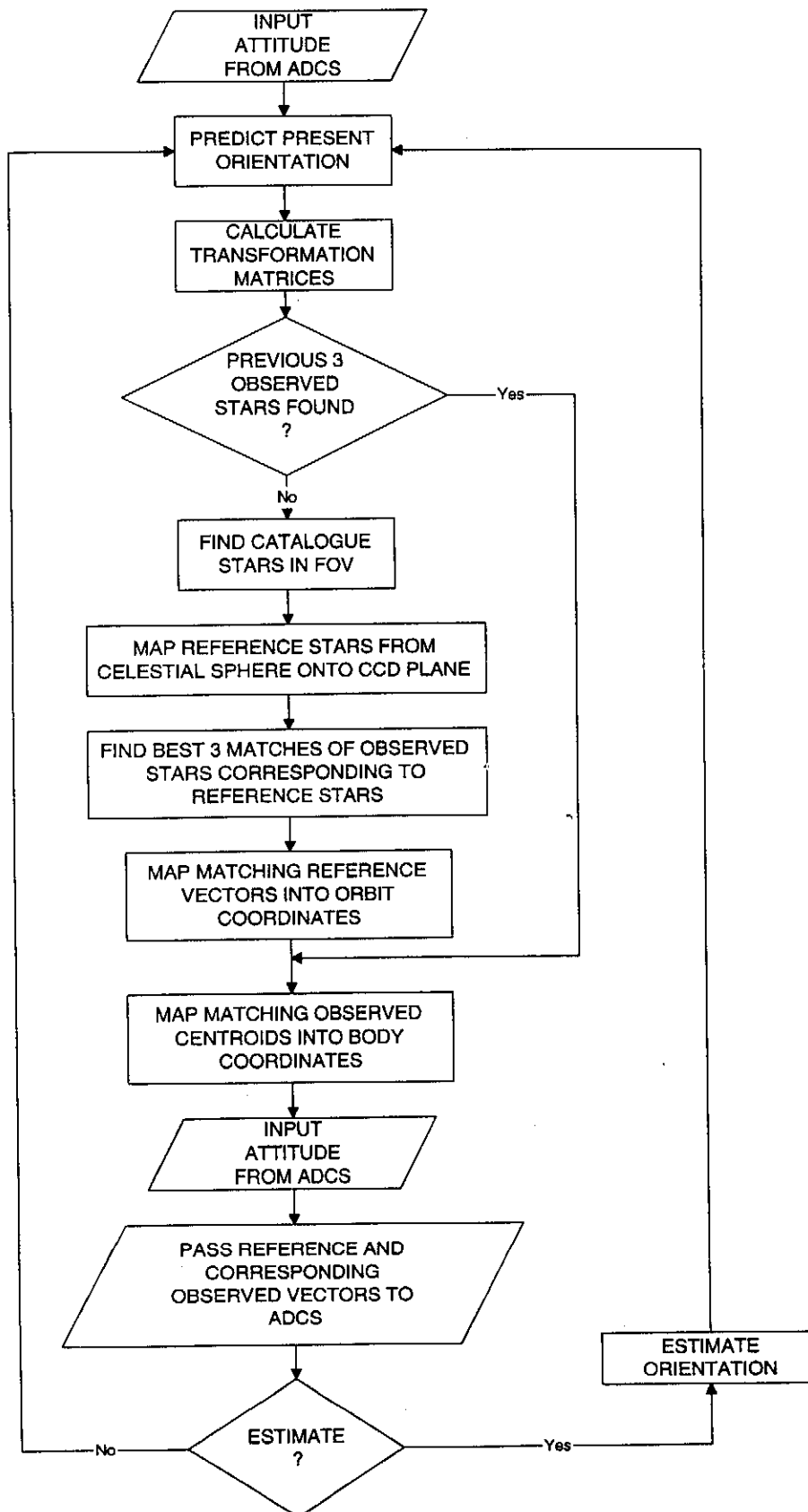


Figure 4-1 Top level flow diagram of the Tracking algorithm

The uncertainty of 0.5° dominates the error due to rotation of the satellite, which is smaller than $0.1^\circ/s$. During the following iterations of the algorithm, the accuracy of the orientation estimate will improve significantly and expected stars may be pinpointed exactly. Under these conditions, a predictor (discussed in Section 4.1.8) is used to predict the present orientation, but during the first iteration the orientation received from the ADCS is used directly as the predicted value.

The predicted value of the present orientation is used to calculate the coordinate transformation matrices discussed in Chapter 1.

If the algorithm is in the first iteration, no previous star matches will be available and a new set of reference stars has to be selected. The sequence of finding reference stars expected in the FOV and searching the image for observed stars is followed, as discussed in Sections 4.1.2 to 4.1.4.

If three matching observed and reference stars were found during the previous iteration of the algorithm, the same observed stars will most probably still appear in the image. The three reference vectors used previously are mapped onto the CCD plane using the new orientation information. The areas surrounding these expected centroids are searched as discussed in Section 4.1.4.1. If all three centroids are found the set remains complete and the new observed centroids are used. If one or more of the observed stars have moved off the FOV, a new set of matching stars is selected. The same procedure followed during the first iteration is repeated. The orientation is however, more accurate.

The set of matching observed and reference vectors are transformed to the required coordinate systems, as explained in Section 4.1.5.

At this stage the algorithm awaits input from the ADCS. Once the attitude has been received, the set of three matching observed and reference vectors are returned to the ADCS via the link. Section 4.2 elaborates on the synchronization of the star sensor and the ADCS.

If it has been decided to use a local orientation estimate rather than the ADCS value, the *Triad* algorithm of Section 4.1.6.1 determines the orientation. This signifies the end of the present iteration. The iteration number is kept in the variable *iterationnumber*, with *iterationnumber* = 1 for the first iteration. The first step of the next iteration (*iterationnumber* + 1) is to predict the present orientation. In other words, it predicts the orientation expected during iteration *iterationnumber* + 1. This orientation is used to find the observed and reference matches.

4.1.1 Calculate transformation matrices

Four transformation matrices are required. They are calculated using the present attitude and the equations of Chapter 1. The FOV corners need to be mapped from the CCD plane onto the celestial sphere. This requires the transformation matrix, $A_{ss \rightarrow c}$ which maps from the star sensor to celestial coordinates. It is calculated using equation (1.25).

The reference stars found in the FOV need to be mapped from the celestial sphere to the CCD plane. The transformation matrix $A_{c \rightarrow ss}$ is calculated with equation (1.23).

A transformation is required to map matching reference vectors from celestial to orbit coordinates. The matrix $A_{c \rightarrow o}$ is given by equation (1.18). The matching observed vectors are mapped from star sensor to body coordinates using $A_{ss \rightarrow b}$ given by equation (1.14).

4.1.2 Finding catalogue stars in the FOV

If the algorithm is in its first iteration or if the previously used observed stars have moved off the CCD plane, a new set of reference stars is selected. The first step is to locate the FOV on the celestial sphere (or the star catalogue).

The four corners of the FOV are found on the celestial sphere, using the transformation matrices calculated previously. The two dimensional positions of the four corners on the CCD plane are given by:

$$\begin{aligned} \text{top left:} \quad & FOVcorner[1]_{colrow} = (22 + 3, 12 + 584) = (25, 596) \\ \text{top right:} \quad & FOVcorner[2]_{colrow} = (22 + 3 + 752, 12 + 584) = (777, 596) \\ \text{bottom left:} \quad & FOVcorner[3]_{colrow} = (22 + 3 + 752, 12) = (777, 12) \\ \text{bottom right:} \quad & FOVcorner[4]_{colrow} = (22 + 3, 12) = (25, 12) \end{aligned}$$

The corner positions on the CCD plane are converted¹ to *horvert* and then to *uv* coordinates (with function *colrowtouv*). These are mapped into the three dimensional star sensor coordinate system with the *uvtoss* function.

$$\begin{aligned} FOVcorner[i]_{uv} &= colrowtouv(FOVcorner[i]_{colrow}) \\ FOVcorner[i]_{ss} &= uvtoss(FOVcorner[i]_{uv}) \end{aligned} \tag{4.1}$$

with $i = 1..4$

A single matrix multiplication places each of the corners in the three dimensional celestial coordinate system.

$$FOVcorner[i]_c = A_{ss \rightarrow c} FOVcorner[i]_{ss} \tag{4.2}$$

The function *ctocs* maps the three dimensional celestial coordinates onto the celestial sphere, where the coordinates are given as right ascension (RA) and declination (Dec).

$$FOVcorner[i]_{cs} = ctocs(FOVcorner[i]_c) \tag{4.3}$$

¹ Details of all functions and matrix transformations appear in Chapter 1.

Having found the corners of the FOV on the celestial sphere, the borders of the FOV are calculated. This is done by noting that a straight line between two corners on the CCD plane will map onto the celestial sphere as the great circle passing through these two points. A great circle between points a and b on a sphere is the line of intersection of a flat plane passing through the centre of the sphere and points a and b , and the sphere. The general equation for a great circle through two points is given in Appendix B.

Next all stars within the FOV are collected. The leftmost corner (with right ascension $\alpha = \alpha_{min}$) and the rightmost corner (with $\alpha = \alpha_{max}$) of the FOV are found.

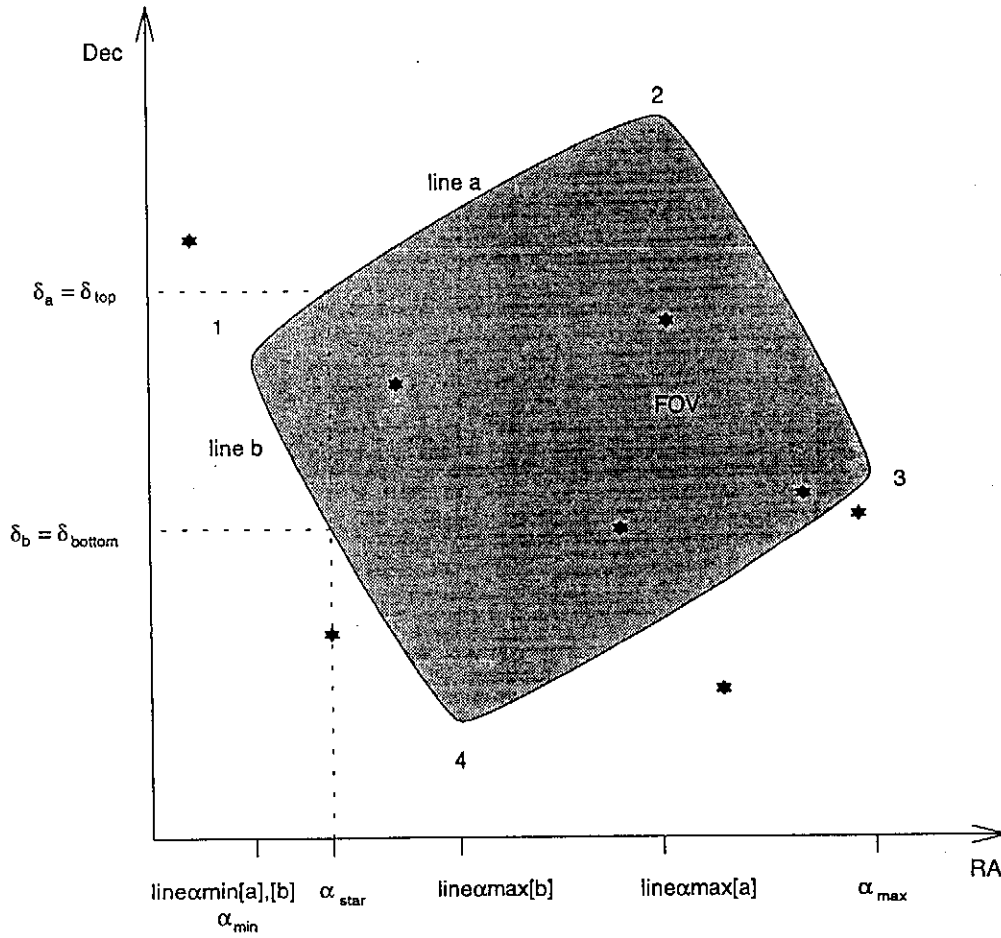


Figure 4-2 FOV on celestial sphere, projected onto a flat surface

For each line i forming an edge of the FOV, the α values of its endpoints (FOV corners) are stored as $line\alpha_{min}[i]$ (left end) and $line\alpha_{max}[i]$ (right end). The right ascension of all stars in the catalogue are checked until one is found for which

$$\alpha_{min} \leq \alpha_{star} \leq \alpha_{max} \quad (4.4)$$

Once such a catalogue star has been found (e.g. the second star from the left in Figure 4-2), the two lines (a and b) for which $line\alpha_{min}[i] \leq \alpha_{star} < line\alpha_{max}[i]$ is true, are used to calculate the declination of the FOV edges. The top and bottom values of the FOV are calculated using the equations of the great circles describing the lines.

$$\begin{aligned}\delta_{bottom} &= \min(\delta_a, \delta_b) \\ \delta_{top} &= \max(\delta_a, \delta_b)\end{aligned}\tag{4.5}$$

with

$$\begin{aligned}\delta_a &= GC(\text{point}(line\alpha_{min}[a]), \text{point}(line\alpha_{max}[a]), \alpha_{star}) \\ \delta_b &= GC(\text{point}(line\alpha_{min}[b]), \text{point}(line\alpha_{max}[b]), \alpha_{star})\end{aligned}\tag{4.6}$$

and where

$GC(\text{point1}, \text{point2}, \alpha)$ returns the δ (Dec) value on the great circle passing through points 1 and 2, where $RA = \alpha$. The equation is given in Appendix B. The next step is to check if the declination of the star is correct. For the star to fall within the FOV, the following test has to be satisfied:

$$\delta_{bottom} \leq \delta_{star} \leq \delta_{top}\tag{4.7}$$

The second star from the left will fail this test while the third will pass.

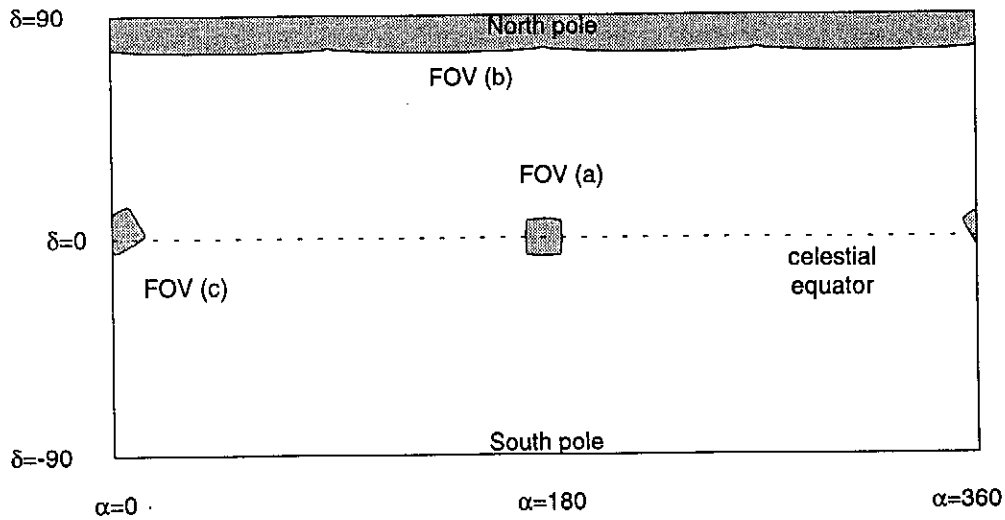


Figure 4-3 FOV (a) near the celestial equator, (b) over the North pole and (c) over the edge of catalogue

The FOV labelled (a) in Figure 4-3 is near the celestial equator and does not cross the line where $\alpha = 0^\circ$. It corresponds to the FOV shown in Figure 4-2. The other two special cases shown in Figure 4-3 have to be considered as well. FOV (b) includes a celestial pole. In this case the values for δ_{bottom} and δ_{top} have to be corrected. FOV (c)

crosses the line $\alpha = 0^\circ$. This creates a problem when searching the catalogue since right ascensions in the catalogue range from 0° to 360° .

To determine which one of the three cases are current, the number of times which the FOV boundaries cross the edge of the catalogue (i.e. the line $\alpha = 0^\circ$) is calculated as *nofzerocrossings*. This is done by noting that any line i which crosses $\alpha = 0^\circ$ will have $line\alpha_{max}[i] < line\alpha_{min}[i]$.

The *nofzerocrossings* indicate where the FOV is situated on the celestial sphere:

- No crossings indicate that the FOV is situated somewhere in the centre of the catalogue and does not cross $\alpha = 0^\circ$ or a celestial pole, e.g. FOV (a) in Figure 4-3.
- One crossing indicates that a celestial pole is included within the FOV. This situation is shown as FOV (b) in Figure 4-3.
- Two crossings occur when the FOV lies across $\alpha = 0^\circ$, e.g. FOV (c) shown in Figure 4-3.

In the case where the FOV includes the celestial North pole, the following values are used:

$$\begin{aligned}\alpha_{min} &= 0 \\ \alpha_{max} &= 2\pi \\ \delta_{bottom} &= GC(\text{point}(line\alpha_{min}[i]), \text{point}(line\alpha_{max}[i]), \alpha_{centre}) \\ \delta_{top} &= \pi/2\end{aligned}$$

where i is the number of the line for which $line\alpha_{min}[i] \leq \alpha_{star} < line\alpha_{max}[i]$

and if the celestial South pole is included

$$\begin{aligned}\alpha_{min} &= 0 \\ \alpha_{max} &= 2\pi \\ \delta_{bottom} &= -\pi/2 \\ \delta_{top} &= GC(\text{point}(line\alpha_{min}[i]), \text{point}(line\alpha_{max}[i]), \alpha_{centre})\end{aligned}$$

If the FOV crosses the catalogue edge $\alpha = 0^\circ$, it is split in two. The one section is situated to the left of the edge and the other to the right. Each is scanned individually for stars.

In the above discussion it was mentioned that all the stars in the catalogue were checked during each iteration to see which fall within the FOV. This wastes a lot of processor time and is prevented by creating an uncertainty area which is slightly larger than the FOV. Only the stars within this area are checked. Once the satellite has rotated significantly, a new uncertainty area is generated.

4.1.3 Mapping reference stars from celestial sphere to CCD plane

The number of reference stars found in the FOV may vary significantly². Depending on where it is situated on the celestial sphere. A band³ close to the celestial equator has a high star density while the celestial pole regions have a low density. The FOV may contain no stars or only one or two, sometimes there may be close to 50. Three matching observed and reference vectors need to be passed to the ADCS. As mentioned in Section 4.1.6.2, the accuracy of the orientation estimate increases considerably if these vectors are well separated.

Selecting a set of three vectors well spread across the FOV is more complex on the celestial sphere than on the CCD plane⁴. For this reason all the reference stars are transformed from the celestial sphere to the CCD plane. To map the reference star i from the celestial sphere to the CCD plane, the sequence of equation (1.26) is used:

$$\begin{aligned} \text{refstar}[i]_c &= \text{cstoc}(\text{refstar}[i]_{cs}) \\ \text{refstar}[i]_{ss} &= A_{c \rightarrow ss} \text{refstar}[i]_c \\ \text{refstar}[i]_{uv} &= \text{sstouv}(\text{refstar}[i]_{ss}) \end{aligned} \quad (4.8)$$

Having found all the reference stars' positions on the CCD plane, the best three matches are selected.

4.1.4 Selecting the best three matches

To obtain the highest orientation accuracy, a set of three well separated stars are selected. The FOV is divided into nine boxes, numbered as shown in Figure 4-4. The boxes are searched consecutively to find a well spread selection of reference stars.

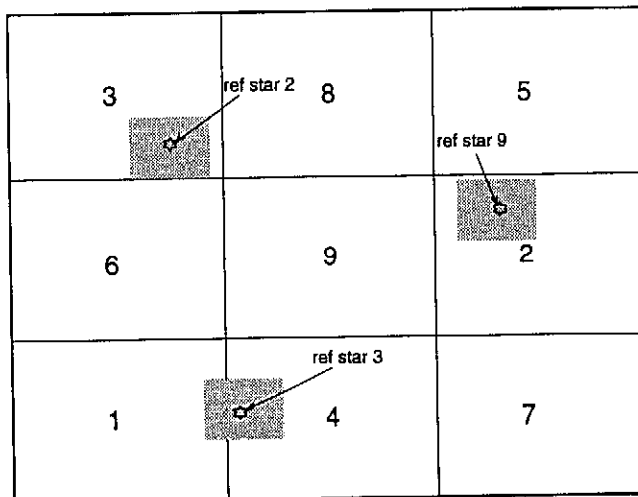


Figure 4-4 Subdivision of the CCD plane into nine boxes, showing the positions of the reference stars and the search areas (grey rectangles)

² Figure 5-1 shows a histogram of the number of stars expected in the FOV

³ Appendix A shows the stars on the celestial sphere up to magnitude 6.

⁴ The FOV may appear very warped on the celestial sphere while on the CCD plane it is rectangular

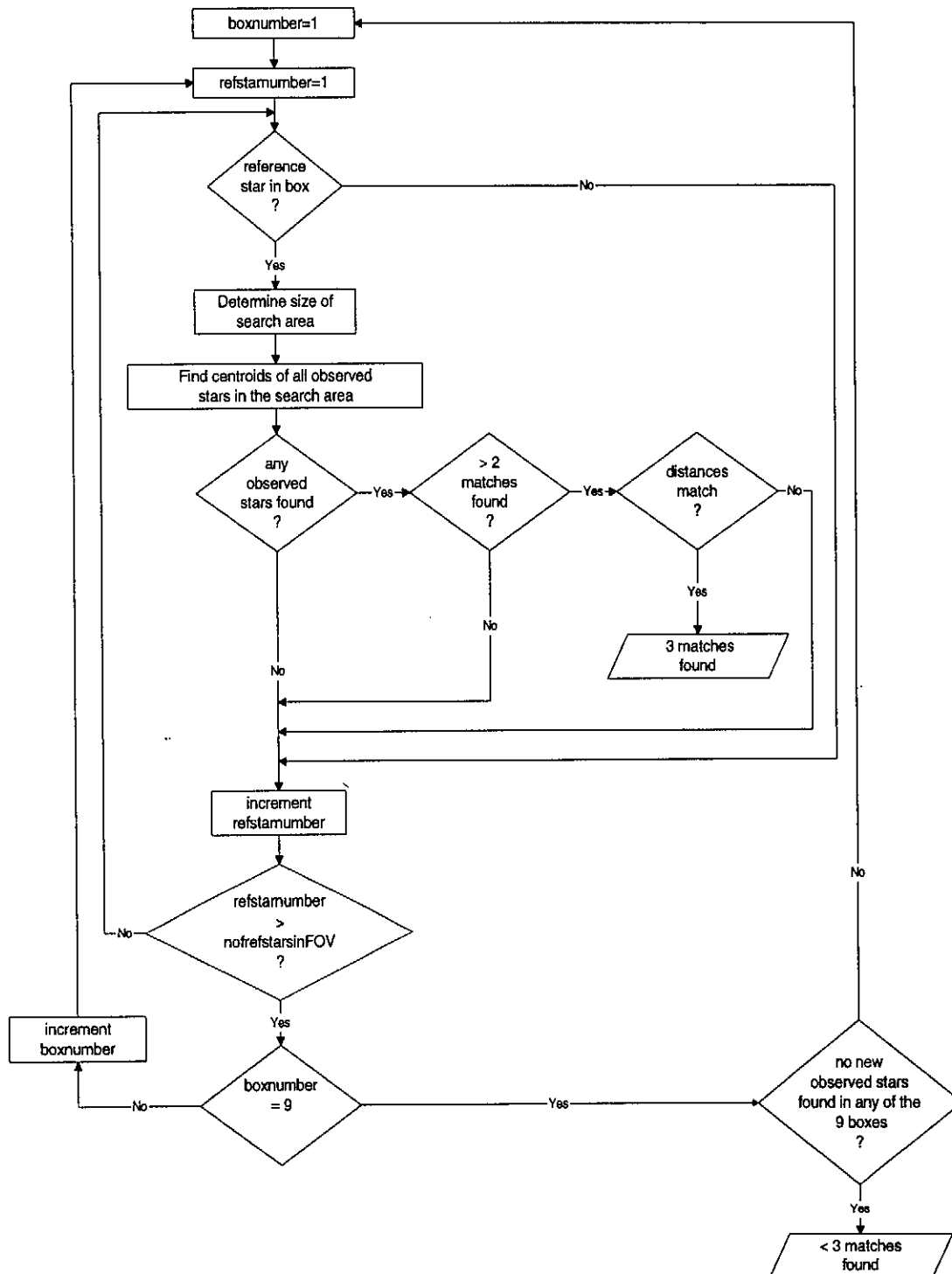


Figure 4-5 Selection of the best three matches

One star is selected from each box until a matching set of three reference and observed stars have been found. If all nine boxes have been searched and the three matches have not been found, the search is restarted at box one and only at this stage can another star be selected from a previously used box. The flow diagram of Figure 4-5 illustrates the selection process. The boxes method does not *guarantee* that the stars will be well spread, since two stars may be very close to one another but on opposite sides of the box boundaries. Due to the method of selecting the reference

stars, the left most star in each box will be the first one used. This minimises the probability of ending up with a cluster of close stars.

Consider the example of Figure 4-4. Box one is searched to see if it contains any reference stars. None are found and the search is repeated for box 2, where one reference star (number 9) is found. This reference star position is the point on the CCD plane where an observed star is expected.

4.1.4.1 Finding observed stars in the search area

An area surrounding the expected centroid position is searched to find the observed star in the image memory. This is the search area referred to in Chapter 2. The radius (or half the length of the square's side) of the search area (grey rectangle in Figure 4-4) is determined in Section 4.1.6.3.

The position of the reference star on the CCD plane is used to estimate the position in the image memory. The first step is to determine in which pixel of the CCD the centroid of the observed star is expected. The following two conversions determine the pixel:

$$\begin{aligned} \text{refstar}[i]_{\text{horvert}} &= \text{uvtohorvert}(\text{refstar}[i]_{\text{uv}}) \\ \text{refstar}[i]_{\text{colrow}} &= \text{horverttocolrow}(\text{refstar}[i]_{\text{horvert}}) \end{aligned} \quad (4.9)$$

This pin points the row and column of the CCD in which the centroid of star i should be located, which is the centre of the search area. The region growing technique described in Chapter 2 is used to find all imaged stars within the search area, in the image memory. The implementation module *SSGrow.mod* is called with the following parameters:

- the boundaries of the search area
- *restoresearcharea* set to *TRUE*
- the present estimate of the orientation and previous orientation
- a *TRUE* Boolean value which sets the flag for checking the shape and direction of stars

The boundaries of the search area are its first and last columns and bottom and top rows. For reference star i they are determined as follows:

$$\begin{aligned} \text{firstcolumn} &= \text{refstar}[i][1]_{\text{colrow}} - r_{\text{search area}} \\ \text{lastcolumn} &= \text{refstar}[i][1]_{\text{colrow}} + r_{\text{search area}} \\ \text{bottomrow} &= \text{refstar}[i][2]_{\text{colrow}} - r_{\text{search area}} \\ \text{toprow} &= \text{refstar}[i][2]_{\text{colrow}} + r_{\text{search area}} \end{aligned}$$

where

$r_{\text{search area}}$ = search area radius, in pixels, determined in Section 4.1.6.3

Any column or row number exceeding the effective image area⁵ is truncated to the edge of this area.

When the orientation estimate is poor, the search area will be large and more than one star may be found (as is the case for the search areas of reference star 3). Only a section of a star may be included as is the case with reference star 2 in Figure 4-6. The region growing algorithm is robust enough to cope with this situation. It determines that the star extends over the edge of the search area and does not use it as a valid star. The centroids of all the stars in the search area are returned.

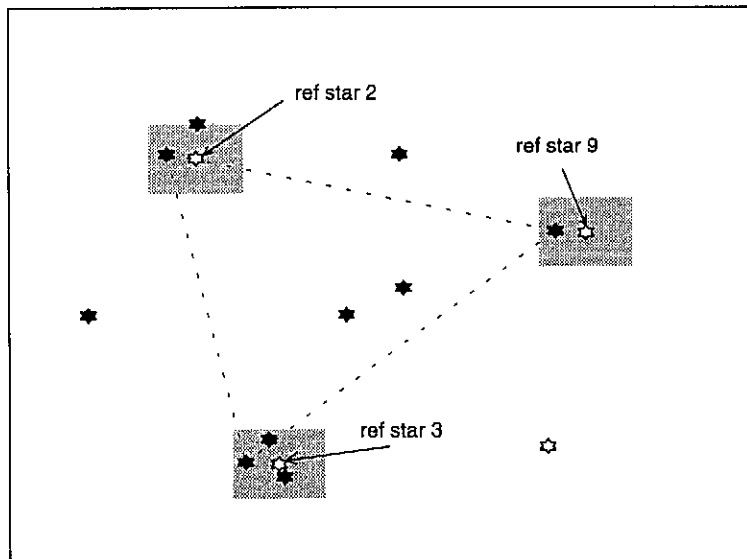


Figure 4-6 Observed stars (solid) actually appearing in the image and reference stars (outlined) used to find three of them. Due to the orientation error, all the observed stars appear shifted to the left from the expected positions.

It is important that the region growing algorithm restores the search area on completion of the search. Suppose for example that only three close stars appeared in the image. Three corresponding reference stars appear in the catalogue. The first time the *Region Growing* algorithm is used, all three stars are within the first reference star's search area. All three are erased and their centroids returned as possible matches for the first reference star. When the *Region Growing* algorithm is used again to find matches for the second reference star, no observed stars will be found. If the first search area included only part of one of the stars, it would be partly erased. The next time the region growing algorithm is called, the remaining section of the partly erased observed star will give an incorrect centroid. These errors are prevented by restoring the search area on completion of the search.

The observed stars which are returned by the region growing module, are checked. Only those with magnitudes matching⁶ that of the reference star are kept. If one or

⁵ See Figure 1-1

⁶ Refer to equation (3.1) for a definition of magnitude matching

more matching observed stars are found for the reference star, two arrays referred to as the *reference* array and *observed* array are updated. The reference star number and number of matching observed stars are stored in the *reference* array. The centroids of the matching observed stars are stored in the *observed* array.

The first reference star for which a match is found, is reference star 9. The two arrays are updated and the search continued. In box 3, reference star 2 is the first to be selected and one match is found for it. In box 4, reference star 3 is the first reference star used. Three matching observed stars are found.

The reference and observed arrays are shown in Tables 4-1 and 4-2.

	1 st reference star	2 nd reference star	3 rd reference star
reference star number	9	2	3
number of observed stars	1	1	3

Table 4-1 The *reference* array

	1 st reference star	2 nd reference star	3 rd reference star
1 st observed star	centroid _{uv,1}	centroid _{uv,2}	centroid _{uv,3}
2 nd observed star			centroid _{uv,4}
3 rd observed star			centroid _{uv,5}

Table 4-2 The *observed* array, containing the centroids of all the observed stars matching the reference star in the same column of the *reference* array.

Once the first three matches have been found, the distances between the observed stars are checked. The three distances between the reference stars are calculated, using their right ascensions and declinations listed in the star catalogue. Equation (C.6) of Appendix C is used. Starting with the first observed star associated with the first reference star and the first observed star associated with the second reference star, the angular distance is calculated. Equation (C.5) is used to calculate observed distances. If the reference and observed distances match, the search is continued to the third column. This process is repeated until a set of three observed stars have been found (one from each column) for which all three reference and observed distances match. This set is accepted as the matching set. The observed centroids are converted to body coordinates and the reference vectors to orbit coordinates. The matching set is returned to the ADCS.

Each column of the observed array may contain several stars. With a search area radius of 50 pixels, the area searched will cover approximately $100 \times 100 / (752 \times 584) \approx 1/44$ of the image. On average an image contains between 5 and 15 stars. Even in the worst case it may contain 50 stars. Quite often stars appear in clusters and five to ten (assuming the magnitudes of all stars match that of the reference star, which will not usually be true) may appear in the search area. The point to be made, however, is that the probability is negligible that all three columns of the

observed array will contain large numbers of stars, causing a lot of time to be wasted performing distance calculations.

Once the orientation estimate is accurate enough, the search area radius should be about 10 pixels and include only one star. In this case the distance calculations will verify the match based on the expected positions of the stars and the estimated magnitude. This is the usual mode in which the algorithm will operate. The orientation estimate is only poor when the algorithm starts initially or when the star sensor can detect only one observed star or two stars which are very close to one another.

If only two distances in the orientation array match the reference distances, these two matches are stored, but the search is continued for a new match to complete a set of three matches. This situation will occur if one of the observed stars expected on the CCD plane is a high sigma case. The measured position for the centroid of the observed star has a normal distribution, centred on the true position of the star. In a high sigma case, the centroid of the star can deviate from the true position by a considerable amount. The faulty centroid value of this high sigma star will cause all three matches to fail the distance test. An error may occur as well, if the observed magnitude deviates considerably from the expected value. Suppose the correctly observed star associated with reference star 3 failed the magnitude test and was discarded. Only the two incorrect observed stars may be associated with the reference star and will fail the distance test. In both these cases the search is continued for a set of three matches which satisfy the distance test.

Once a fourth matching reference and observed group have been found, the distances between all the combinations of the four matching groups are checked. This will eliminate the high sigma case. If more than one of the observed stars are high sigma cases, a matching set will not yet be found. A fifth reference star and matching observed stars are selected and again all combinations of distances are checked to find a set of three stars. If a set of three matching reference and observed stars have not yet been found, it means that at least three of the five reference stars have high sigma observed stars. A sixth reference star is selected and the process repeated. Due to the large number of combinations which need to be checked, the process can not be repeated for all the reference stars (which may be as many as 50). By the time the seventh reference star is to be selected, some unforeseen error has quite probably occurred. This situation should never be reached, but if it is, the first four reference stars are discarded and only the latest three used. This is the procedure followed until all remaining reference stars have been checked.

Three pairs of reference and matching observed stars should be found before all reference stars have been checked. The vectors are transformed to the appropriate coordinates and returned to the ADCS. If all reference stars have been checked and only two matching vectors have been found, their coordinates are transformed. If one match was found, it can only be based on magnitude and rotational information can be derived from it. The reference and observed vector will still be passed to the ADCS.

4.1.5 Transformation of centroids

The centroids are transformed to the appropriate coordinates exactly as described in Section 3.4.

4.1.6 Attitude information for the next iteration

The algorithm needs an estimate of the present orientation to predict the orientation during the next iteration. The orientation is imported from the ADCS and may be estimated with the *Triad* algorithm as well. The data received from the ADCS includes the present orbit parameters as well as the latest orientation. It synchronizes⁷ the two systems by signifying that the ADCS is ready to receive the matching vectors.

It is advantageous to use a local star sensor orientation update rather than relying on the ADCS value. A few algorithms were considered for this purpose. Most methods involved an iterative search for the transformation matrix, while the *Triad* algorithm described in [13] gives an analytical solution. The use of the *Triad* algorithm for the star sensor's local orientation update is examined in the following section.

4.1.6.1 The *Triad* algorithm

The orientation estimate generated by this algorithm is only used to locate stars on the CCD plane and not as an attitude update for the ADCS. It is thus important to obtain a relatively accurate estimate in as short a time as possible. A simple procedure referred to as the *Triad* algorithm is ideally suited for these purposes. Shuster [13] discusses this algorithm which is widely used by spacecraft to provide a closed form solution to the attitude transformation matrix. It uses two pairs of observed and reference vectors to calculate a nonoptimal solution for the transformation matrix. Since the equation is available in closed form, no iterations are required. In other words, a fast but less accurate estimate is obtained.

Since the roll, pitch and yaw angles which rotate between the body and orbit coordinate system are the variables in question, the transformation matrix $A_{o \rightarrow b}$ is required. From the selection part of the *Tracking* algorithm up to three pairs of observed and reference vectors are available. The observed vectors are in the uv coordinate system while their matching reference vectors are on the celestial sphere. The accuracy⁸ of the pitch estimate is strongly influenced by the separation between the vector pairs. Rotational information can not be obtained to a high accuracy if both pairs of vectors are close together. In the limit where they are parallel, no rotational information is available and the *Triad* algorithm fails. To maximize pitch accuracy, the two furthest separated stars are selected. From the CCD plane (uv coordinates) they are mapped⁹ into the body coordinate system to obtain the two observed unit

⁷ See Section 4.2

⁸ The errors in estimates are discussed in the next section

⁹ Refer to Chapter 1 for details

vectors $\underline{\hat{W}}_1$ and $\underline{\hat{W}}_2$. Their catalogue matches are mapped from the celestial sphere into the orbit coordinate system as the two reference unit vectors $\underline{\hat{V}}_1$ and $\underline{\hat{V}}_2$.

The desired transformation matrix satisfies

$$\begin{aligned}\underline{\hat{W}}_1 &= A_{o \rightarrow b} \underline{\hat{V}}_1 \\ \underline{\hat{W}}_2 &= A_{o \rightarrow b} \underline{\hat{V}}_2\end{aligned}\tag{4.10}$$

Two triads are constructed from the reference and observation vectors.

$$\begin{aligned}\underline{\hat{r}}_1 &= \underline{\hat{V}}_1 \\ \underline{\hat{r}}_2 &= \frac{\underline{\hat{V}}_1 \times \underline{\hat{V}}_2}{|\underline{\hat{V}}_1 \times \underline{\hat{V}}_2|} \\ \underline{\hat{r}}_3 &= \frac{\underline{\hat{V}}_1 \times (\underline{\hat{V}}_1 \times \underline{\hat{V}}_2)}{|\underline{\hat{V}}_1 \times \underline{\hat{V}}_2|}\end{aligned}\tag{4.11}$$

$$\begin{aligned}\underline{\hat{s}}_1 &= \underline{\hat{W}}_1 \\ \underline{\hat{s}}_2 &= \frac{\underline{\hat{W}}_1 \times \underline{\hat{W}}_2}{|\underline{\hat{W}}_1 \times \underline{\hat{W}}_2|} \\ \underline{\hat{s}}_3 &= \frac{\underline{\hat{W}}_1 \times (\underline{\hat{W}}_1 \times \underline{\hat{W}}_2)}{|\underline{\hat{W}}_1 \times \underline{\hat{W}}_2|}\end{aligned}\tag{4.12}$$

The unique orthogonal matrix $A_{o \rightarrow b}$ which satisfies

$$\underline{\hat{s}}_i = A_{o \rightarrow b} \underline{\hat{r}}_i\tag{4.13}$$

with

$$i = 1, 2, 3$$

is given by

$$A_{o \rightarrow b} = \sum_{i=1}^3 \underline{\hat{s}}_i \underline{\hat{r}}_i^T\tag{4.14}$$

This is identical to

$$A_{o \rightarrow b} = M_{obs} M_{ref}^T\tag{4.15}$$

with

$$\begin{aligned} M_{obs} &= [\hat{s}_1 \mid \hat{s}_2 \mid \hat{s}_3] \\ M_{ref} &= [\hat{r}_1 \mid \hat{r}_2 \mid \hat{r}_3] \end{aligned} \quad (4.16)$$

The matrices of equation (4.16) are 3×3 .

Equation (4.15) gives a unique solution for the attitude transformation matrix $A_{o \rightarrow b}$ which is also given by equation (1.13). By equating the right hand sides of equations (4.15) and (1.13), the pitch, roll and yaw angles are calculated:

$$\theta = \arcsin A_{o \rightarrow b \ 31} \quad (4.17)$$

$$\phi = \arcsin \left(-\frac{A_{o \rightarrow b \ 32}}{\cos \theta} \right) \quad (4.18)$$

$$\psi = \begin{cases} \arcsin \left(-\frac{A_{o \rightarrow b \ 21}}{\cos \theta} \right) & : \quad A_{o \rightarrow b \ 11} \geq 0 \\ \pi - \arcsin \left(-\frac{A_{o \rightarrow b \ 21}}{\cos \theta} \right) & : \quad A_{o \rightarrow b \ 21} \leq 0 \text{ and } A_{o \rightarrow b \ 11} < 0 \\ -\pi - \arcsin \left(-\frac{A_{o \rightarrow b \ 21}}{\cos \theta} \right) & : \quad A_{o \rightarrow b \ 21} > 0 \text{ and } A_{o \rightarrow b \ 11} < 0 \end{cases} \quad (4.19)$$

This orientation estimate is used to find the stars on the CCD plane. The accuracy of estimate is important since it determines the size of the area searched for each star.

4.1.6.2 Accuracy of the *Triad* algorithm

The accuracy with which stars can be pin-pointed on the CCD plane when using the *Triad* algorithm is discussed in detail in Appendix E. Both the accuracy of the *Triad* orientation estimate and the positional error on the CCD plane due to errors in this orientation are examined. The results are summarized here. All results were derived using roll and pitch angles close to zero and a yaw of either -45° or 135° . The orientation of the satellite should be close to this during earth imaging, which is the prime time for use of the star sensor.

An error was intentionally introduced in each of the observed vectors used to estimate the orientation. Equations were derived to describe these errors. The maximum error expected in the roll angle is given by equation (E-19), while that of the pitch estimate is given by equation (E-12) and the yaw error by equation (E-20). All three errors are inversely proportional to the distance separating the stars. As the stars approach one another, the error increases. This emphasizes the necessity of using well separated stars for the orientation estimate. The errors are directly proportional to the maximum error expected in any of the two observed star's centroids. The roll error increases as the star moves away from the optical centre in the horizontal (or u) direction while the

yaw error increases as the star's vertical (or v) displacement from the optical centre increases.

The error expected when calculating the positions of stars on the CCD plane with erroneous orientation was examined as well. Since a pitch of the satellite only rotates a star around the optical centre, the positional error of the star (equation (E.22)) is a function of its radial distance from the optical axis. A pitch error has no effect on the estimated position of a star on the optical axis, but increases as the star approaches the edge of the CCD plane. Errors in roll (equation (E.21)) and yaw (equation (E.23)) are proportional to the focal length and have a much greater influence on the positional error.

The maximum positional error of a star on the CCD plane due to orientation errors is given by equation (E.26). If this value is smaller than two pixels, an error of two pixels is used.

4.1.6.3 Determining the size of the search area

The FOV is located on the celestial sphere using the predicted orientation. The catalogue stars found within the FOV are mapped back onto the CCD plane. The imaged stars are expected in the vicinity of these uv centroids. The size of the area on the CCD plane which is searched around the expected centroid of each star, depends on three factors:

- the estimated radius of the star
- the uncertainty in the position of the uv centroid
- the speed of rotation

The radius is estimated using the magnitude of the star listed in the catalogue.

The uncertainty in the position of the uv centroid, depends on the method of determining the present orientation. The estimate obtained from the ADCS should be accurate to 0.5° or about 9 mrad [14] for each axis. The maximum error on the CCD plane is estimated with equations (E.21) to (E.23). The roll and yaw errors are in perpendicular directions while, in the worst case, the pitch will add to this.

$$\begin{aligned}
 \epsilon_{obs,max} &\approx \sqrt{2} f \epsilon_{\phi,max} + r \epsilon_{\theta,max} \\
 &= \sqrt{2} \times (0.009 \times 25) + 3 \times 0.009 \\
 &= 0.345 \text{ mm} \\
 &= 53 \text{ pixels}
 \end{aligned}$$

The *Triad* algorithm's worst case observational error¹⁰ is 46 pixels when the two stars are separated by only 20 pixels. When they are closer than this, the ADCS estimate will be used.

¹⁰ Refer to Appendix E

To summarize, the search area's uncertainty radius¹¹, in pixels, is

$$r_{\text{uncert}} = \begin{cases} 53 & : \text{ADCS estimate} \\ \frac{12 \epsilon_{uv}}{r_{12}} & : \text{Triad estimate} \\ 2 & : \text{Triad estimate, } \frac{12 \epsilon_{uv}}{r_{12}} < 2 \end{cases} \quad (4.20)$$

with

$$\epsilon_{uv} = 0.5 \text{ pixels}$$

r_{12} = distance separating two observed stars on the CCD in mm

Assuming that the satellite is rolling, pitching and yawing at the maximum rate of $0.1^\circ/s$ or 1.75 mrad/s , the maximum translation of a star on the CCD plane is calculated using the time derivatives of (E.1) and (E.2). The motion of the star due to roll and yaw are in perpendicular directions, so that the maximum velocity is

$$\begin{aligned} \dot{l}_{\phi+\psi} &= \sqrt{2} \times \dot{\phi} f \\ &= 0.062 \text{ mm/s} \end{aligned}$$

The maximum velocity due to pitch is reached by a star at the CCD corner, i.e. at a radial distance of 3 mm . The time derivative of equation (E.3) gives

$$\begin{aligned} \dot{l}_\theta &= \dot{\theta} r \\ &= 0.005 \text{ mm/s} \end{aligned}$$

In the worst case the motion due to roll and yaw will add to this. The maximum velocity of a star on the CCD plane is

$$\begin{aligned} \dot{l} &= \dot{l}_{\phi+\psi} + \dot{l}_\theta \\ &= 0.067 \text{ mm/s} \end{aligned}$$

which is slightly more than 10 pixels per second. A star on the CCD plane may be elongated by this amount in any direction. Even a dot covering one pixel on a stationary CCD will appear as a line when the sensor is rotated. The maximum increase in radius of the star due to rotation of the satellite is half the elongation

$$r_{\text{rotation}} = 5 \text{ pixels}$$

The predictor in Section 4.1.8 predicts the present orientation and eliminates any uncertainty regarding the star's position due to rotation of satellite.

¹¹ Refer to Appendix E for details

The rotation and observational error is assumed to be in any direction, so that the search area extends in all directions from the expected centroid. The total radius is the sum of the three

$$r_{searcharea} = r_{star} + r_{uncert} + r_{rotation} \quad (4.21)$$

The search area is a square, centred on the expected centroid. The radius (or half the length of a side of the square) is given by equation (4.21). The reason for using a square instead of a circle is the fact that searching a circle requires considerably more computational effort than searching a square.

4.1.6.4 Conclusions regarding the use of the ADCS and *Triad* estimates

The *Triad* algorithm is accurate enough, but a problem arises when only two very close stars or fewer than two stars appear in the image. Under these conditions the *Triad* algorithm can not function and no orientation update is provided. The *Tracking* algorithm fails to function without an update. In other words, once fewer than two stars appear in the image, the algorithm needs to switch to a mode where the ADCS's orientation is used. The advantage of using the *Triad* algorithm rather than relying solely upon the ADCS update, is that the *Triad* algorithm's first estimate is already very accurate. Once the first set of two or more matching vectors have been found, a very accurate orientation estimate is available. The Kalman filters [14] used by the ADCS have a long time constant of about 15 minutes. The time taken by the Kalman filter to converge to an accuracy of 0.1° depends on the sensors used prior to the star sensor. If the horizon sensors and sun sensors were used for a considerable time prior to the star sensor, the convergence should take about 50 samples (i.e. 50 seconds) and convergence accuracy is 0.1° . Relying only on the magnetometer before using the star sensor, the filter may take up to one orbit to converge to 0.5° .

The search area uncertainty radius is adjusted according to the orientation accuracy. For poor accuracy the area searched for observed stars is large and small for high accuracy. The *Tracking* algorithm is designed to function with poor orientation estimates, but the probability of finding incorrect matches is obviously minimized when the orientation estimate is accurate and the search area is minimized.

The attitude is imported from the ADCS during each iteration, but it is only used when necessary. The variable *estimate* is only set to false and the *Triad* algorithm not used,

- during the first iteration
- when only two very close stars are found in the image
- one star or no star matches are found

4.1.7 Returning the matching vectors

Once the orientation has been received from the ADCS, the matching vectors are returned. The first data returned across the link to the ADCS is the number of matching vectors that were found. This informs the ADCS how many vectors are to follow. If no matches were found, only a zero is returned which indicates that no vectors are following. The observed vectors are returned in body coordinates, followed by the reference vectors in orbit coordinates.

4.1.8 The predictor

The first step of the next iteration of the *Tracking* algorithm is to predict the new (present) orientation. The previous iteration's orientation was estimated using the *Triad* algorithm or obtained from the ADCS. When using the *Triad* algorithm, this present value will actually be the average orientation of the sampling period. If the satellite is not perfectly stationary during this time, the stars will move across the FOV. The CCD integrates for most of the sampling period. If the maximum integration time is used, it will be close to one second. The imaged stars will appear elongated or even in extreme cases as a line. The centroid of such a deformed star gives the average position of the star and the average orientation. The predictor will predict the average position of the orientation during the present iteration, which again gives the average position of the star.

One possibility is to use the previous orientation directly as the present orientation, but this may lead to a positional error of up to 10 pixels¹² in the expected positions of the stars. Another option is to use a highly accurate model of the satellite including the effect of the reaction wheels and magneto torquers. This would be duplicating a function already performed by the ADCS. The last option is to use a simple predictor based on the previous orientation to estimate the present value. A basic predictor assuming constant acceleration was derived to predict the present orientation from previous values. Each of the roll, pitch and yaw are predicted independently using equation (E.7) derived in Appendix E:

$$\begin{aligned}\phi(k) &= 2.5\phi(k-1) - 2\phi(k-2) + 0.5\phi(k-3) \\ \theta(k) &= 2.5\theta(k-1) - 2\theta(k-2) + 0.5\theta(k-3) \\ \psi(k) &= 2.5\psi(k-1) - 2\psi(k-2) + 0.5\psi(k-3)\end{aligned}\tag{4.22}$$

The present samples $\phi(k)$, $\theta(k)$ and $\psi(k)$ are calculated from the preceding three samples. During the first iteration of the Tracking algorithm no orientation history is available, but no calculations are performed either. During the second iteration the previous two values are used in a predictor assuming constant velocity, as shown in equation (4.23). Equation (4.22) is used for all the following predictions. Considering the long time constant of the satellite's dynamics, the assumption of constant acceleration over a period of one second is acceptable.

¹² See Appendix E

$$\begin{aligned}
 \phi(2) &= 2\phi(1) - \phi(0) \\
 \theta(2) &= 2\theta(1) - \theta(0) \\
 \psi(2) &= 2\psi(1) - \psi(0)
 \end{aligned}
 \tag{4.23}$$

4.2 Synchronization

The *Tracking* algorithm needs to be synchronized with the ADCS, which requires a new set of matching vectors every second. Obviously the execution of each iteration of the *Tracking* algorithm has to be shorter than one second. The ADCS has various sensors and actuators to control and under no circumstances may the star sensor delay the routine. This is ensured by proper synchronization.

In parallel with the *Tracking* algorithm's execution, the CCD circuit is controlled to ensure that the latest image is in memory by the time the image memory is scanned. The Sony circuit is used in free running mode. This means that integration and readout of images takes place continuously. The CCD integrates (collects charge) after which the image is clocked into memory. Integration of the next frame is started immediately and it is vital that:

- the duration of this integration and readout sequence is exactly one second
- it is synchronized with the *Tracking* algorithm

The duration of the integration and readout sequence is determined by the frame rate of the CCD.

4.2.1 Adjusting the frame rate

The integration time of the CCD can be varied between 2 and 510 fields. This is set by 8 bits serially written to the CCD control latch. The duration of a field is determined by the crystal. With a 28.375 MHz crystal and an integration time of one field, a new field is read out at precisely 20 ms intervals which is the CCIR television standard. The star sensor CCD circuit is used in frame readout mode¹³ with a frame rate of one second. On completion of integration, both fields are read out instead of the one used in the field readout (as for television) mode. Appendix G.2 gives details on the selection of the correct crystal.

4.2.2 Synchronization of the Tracking algorithm and the ADCS

To establish synchronization, ADCS passes the initial attitude to the star sensor transputer. Immediately zero is returned to ADCS, indicating that no matching vectors have been found yet. The timing diagram is shown in Figure 4-7.

¹³ Appendix G.4 explains the difference between field and frame readout mode

The CCD frame rate is set to 1 Hz as described in Appendix G. Readout of the first image is completed exactly one second later. In the mean time the *Tracking* algorithm waits to receive the attitude for this iteration. The transputer link is read and it waits until an attitude value appears at *time* = 1 s. This attitude is used to generate the transformation matrices. The reference stars are found on the celestial sphere and transformed to the CCD plane. The image memory may not yet be accessed since the analogue to digital converter is still writing the image into memory. The *RDOUT* (read out) bit of the status register¹⁴ is polled until it goes low. It is high when the image is being read out from the CCD into memory and low for the remainder of the cycle. Once *RDOUT* goes low, the image memory is accessed by the transputer and the observed stars detected. The matching observed and reference vectors are transformed to the appropriate coordinates. The link is again read which means that the transputer waits for the next attitude value from ADCS. At *time* = 2 s this value is received and the matching vectors are returned.

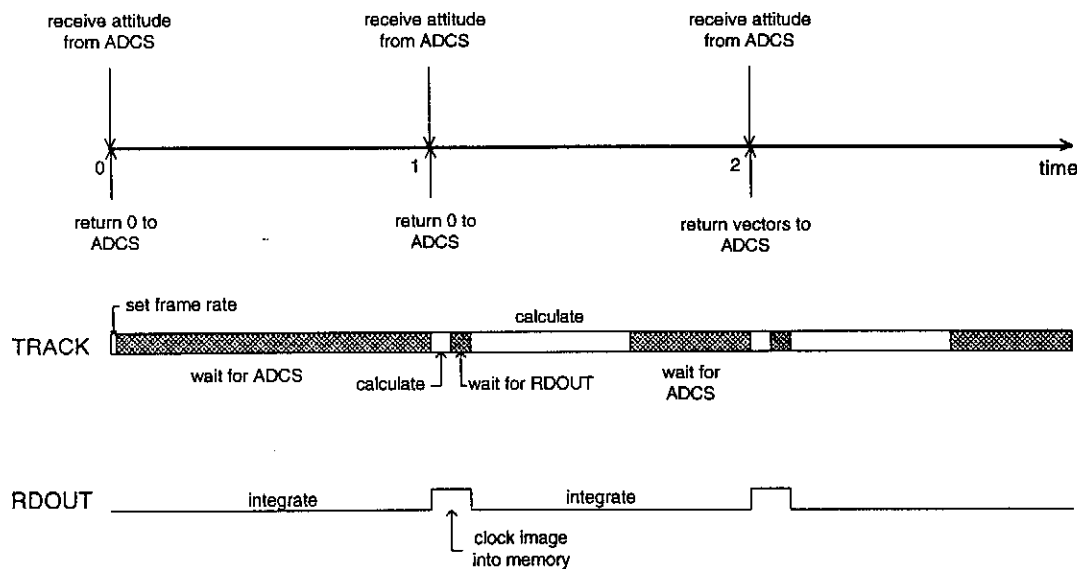


Figure 4-7 Synchronization of ADCS and *Track* is shown in the first and second lines and that of *Track* and the CCD circuit in the second and third lines. The sequence starts at time = 0 s.

This routine is repeated until the termination command is returned to the Tracking algorithm, which halts its execution.

4.3 Testing

It is very difficult to reproduce the rotation which the star sensor will experience in space, here on earth. On the other hand, the *Tracking* algorithm needs to be verified in some way before the satellite is launched. Instead of constructing a complicating test

¹⁴ Refer to Chapter 8 for details

set-up where the roll, pitch and yaw angles are controlled and the 'orbit' parameters are known, simulations for testing purposes are done by means of software.

Rather than taking star images while the sensor is being rotated, images are created by the simulation software. The values of roll, pitch and yaw are obtained from the Turbo Pascal programme *Animate.pas* written by W H Steyn to simulate the motion of Sunsat in space. These simulated orientation values are used together with simulated orbit parameters to find the true position of the FOV on the celestial sphere. The stars found in the star catalogue¹⁵ are mapped onto the CCD plane¹⁶. A radius is generated for the star using the catalogue magnitude. All pixels of which the centres fall within this radius from the centroid of the star are given a value of one, while all others remain zero. This is repeated for all stars found in the FOV to create a binary image. The simulated image has identical dimensions to a real image clocked out by the CCD and is loaded into memory. The *Tracking* algorithm operates exactly as it would when true images are being taken.

The critical aspects are once again the reliability and execution time of the algorithm. The reliability of the *Tracking* algorithm can not be defined as clearly as that of the *Matching* algorithm. It should never fail since it resorts to the ADCS orientation estimate when fewer than three stars appear in the image. A few thousand iterations of the algorithm were run to gain confidence in its operation. The simulated images were even contaminated with 100 false stars of about 100 pixels each and the algorithm still operated smoothly, only occasionally resorting to the ADCS orientation. This happened when fewer than three stars could be distinguished from the false stars.

The readout time of the CCD is about 80 *ms*, which leaves the *Tracking* algorithm with 920 *ms* to extract the stars from the image and match them. The execution time of the algorithm is only influenced by the catalogue size and magnitude of the stars used for matching. Even when the full catalogue of 5000 stars is used, the time taken to find the reference stars expected within the FOV seldom exceeds 70 *ms*, which means that the full catalogue can be loaded into the Flash RAM prior to the launch, requiring no in flight updates. The catalogue only needs to be searched when new stars are required for tracking, i.e. when the tracked stars move out of the image. The time needed to search the catalogue together with all other overheads, except the time required to find the stars in the image, is less than 0.1 *s*. About 800 *ms* are left to extract the stars from the image.

Typically three faint stars are used for matching and the average extraction time for all three is below 50 *ms*. When the previous orientation used by the *Tracking* algorithm is provided by the star sensor, the search area for each star is very small and seldom includes any other stars. In the worst case three stars of about 300 pixels can be used, requiring about 100 *ms* for extraction¹⁷.

The worst situation arises when an ADCS orientation estimate is used. The search area includes a large area surrounding each star, which may include a few other stars

¹⁵ Using the technique of Section 4.1.2

¹⁶ As discussed in Section 4.1.3

¹⁷ Refer to Figure 2-7 for details

as well. Using a search radius of 53 pixels, each search area covers only 2.5% of the total image and should not include too many stars. All stars in the search area need to be extracted and considered as possible matches. Assuming that three of the brightest allowed stars are being tracked, 700 *ms* are left to extract other stars appearing in the search areas. Figure 2-7 shows that at least 20 stars of 300 pixels can be extracted in this time.

The possibility exists that a star of area larger than the allowed maximum star size appears in the search area. This star needs to be eliminated from the search area as well. Even the brightest stars should never cover more than 900 pixels and if such a bright star appears in the image, there will almost certainly be only one. Figure 2-7 shows that the time required to extract such a large star is 90 *ms*. Since it is larger than the allowed maximum, it will not be accepted as a star, which means that its centroid will not be calculated and its shape does not need to be checked. Consequently the time required to eliminate the star from the image is about 40 *ms*.

In an extreme case where several large search areas with many bright stars need to be searched before the three matches can be found, the possibility can not be excluded that the extraction time exceeds 920 *ms*. In such a case ADCS may not be kept waiting. The internal timer of the transputer will be polled periodically to show when the allowed time has run out. Zero will be returned to ADCS, indicating that no match could be found.

4.4 Conclusion

Simulations showed that the *Tracking* algorithm is a reliable algorithm for tracking stars. The execution time should never come close to 1 *s*. ADCS will be provided with three matching observed and reference vectors whenever three stars appear in the image. When fewer than three stars appear in the image (of which the probability is shown to be 0.0056 in Section 5.1), the algorithm will not be able to provide matching vectors and will rely on the ADCS orientation to find a new set of stars. This situation may prevail for a number of seconds while the FOV crosses the sparsely populated section of the celestial sphere.

5. Ground Support Software

The previous three chapters concentrated on the on board software. Ground support software to assist users of the star sensor is described here.

To ensure reliable operation of the matching software, it is essential to find at least two stars in the FOV. Preferably three or more stars should appear in the image. While designing the hardware, the influence of the FOV size and visible magnitude on the number of the stars appearing in the FOV had to be determined. Software producing these statistics is described in Section 5.1.

The probability of finding incorrect matches is decreased and search speed of the matching software increased when a sub catalogue is used instead of the entire catalogue. The software used to generate the sub catalogue is discussed in Section 5.2.

5.1 Statistics

The number of stars appearing in the image is determined by the FOV size and the faintest visible magnitude. The faintest visible magnitude is a function of the lens F/number, type of CCD and the FOV size. A lens with a small F/number will collect a lot of light¹, making faint stars visible. If the FOV is enlarged, the image of the star becomes a smaller percentage of the total image. Very faint stars are visible when the FOV is very narrow, but as the FOV is widened the fainter stars fade away.

It is essential that the star sensor has a very high probability of detecting at least three stars in each image. Tests performed by Greyling [5] for the first version of the Sunsat star sensor showed that a 10° square FOV with a sensor capable of detecting magnitude 6 stars is required. The number of stars expected in an image taken by the present star sensor² was determined with the Modula programme *Stats.mod* which appears in Appendix I. Figure 5-1 shows a histogram of the number of stars expected in an image. It was generated by using a random orientation to locate the FOV on the celestial sphere. Identical techniques to those implemented in *Track.mod*³ were used to find and count the catalogue stars in the FOV. The histogram entry corresponding to the number of stars counted in the simulated image, was incremented. This process was repeated a million times to produce Figure 5-1. The probability of finding a specified number of stars in an image may be calculated from the histogram and Table 5-1, which shows the smaller values.

The probability of finding fewer than three stars is the sum of the probabilities of finding zero, one or two stars i.e. 0.0056. In other words, enough stars to enable reliable recognition should appear in the image most of the time.

¹ Chapter 9 contains details

² With FOV 11.17° × 8.32°

³ Refer to Chapter 4 for details

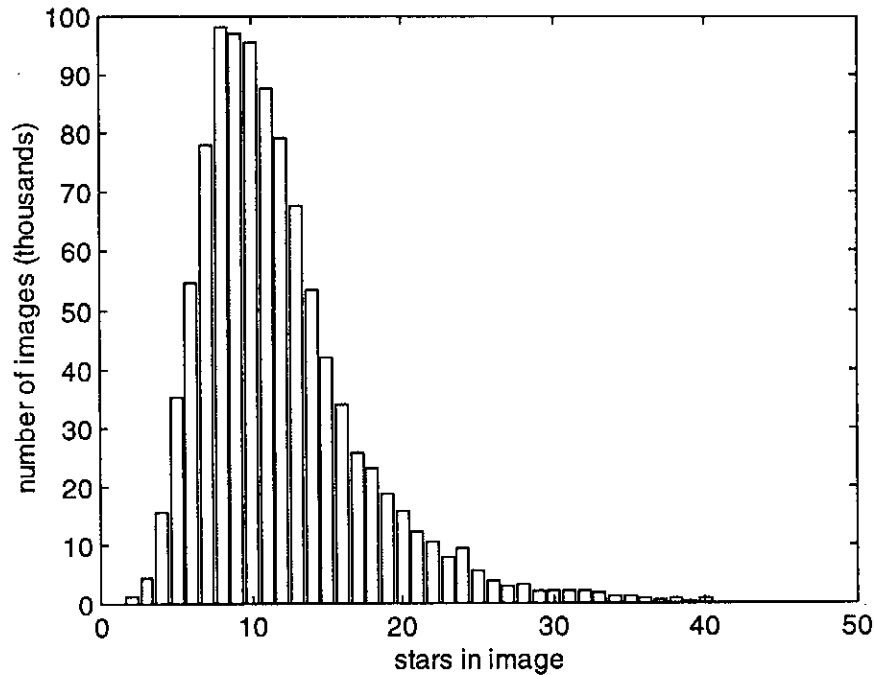


Figure 5-1 Histogram showing the number of images containing the indicated number of stars. A total of 1 million random FOV locations on the celestial sphere were used. The FOV size is 11.17° by 8.32° and the catalogue contains stars up to $M_v=6$.

For the 1 million images checked, a maximum number of 49 stars were found in a single image. This only occurred 8 times. Only 21 of the images contained no stars.

Stars in image	Probability (10^3)
0	0.021
1	1.05
2	4.53
3	15.46
4	35.13
:	:
40	0.808
41	0.331
42	0.273
43	0.238
44	0.258
45	0.107
46	0.283
47	0
48	0
49	0.008
50	0

Table 5-1 Probability of finding a specified number of stars in the image

The programme *Stats.mod* may be used to estimate the number of stars expected in the image for the sub catalogues as well. Since the sub catalogues should be centred around the celestial equator rather than the sparsely populated pole regions, the average number of stars expected in the image should increase slightly.

5.2 Making a sub catalogue

Chapter 3 emphasized the necessity of using a sub catalogue instead of the entire star catalogue. The programme *MakeCat.mod* produces the sub catalogue. It collects all the stars from the catalogue which appear in the FOV over a period of time. The duration of this period determines the variation in orbit parameters which influences the size of the sub catalogue. Due to the variation in orientation (roll, pitch and yaw) the FOV will move around on the celestial sphere as well.

5.2.1 Method used to produce the sub catalogue

The extreme values of the orientation and orbit parameters are input to the programme. A typical set of extreme values expected over a period of two weeks is given in Table 5-2.

	Minimum	Maximum
roll	-10°	10°
pitch	-5°	5°
yaw	130°	140°
ω'	0°	360°
i	97°	97°
Ω	133°	149°

Table 5-2 Typical set of extreme values expected for a period of two weeks

The right ascension of the ascending node Ω , inclination i and sum of argument of perigee and true anomaly ω' are defined in Section 1.4. Each of the variables in Table 5-2 is varied from its minimum to its maximum value, by a specified increment size. The increment with which the variable changes during each iteration seriously affects the algorithm's speed.

The celestial sphere is divided into $1^\circ \times 1^\circ$ square blocks. For each combination of the six variable values, the FOV is located on the celestial sphere. This is done as described in Chapter 4. All $1^\circ \times 1^\circ$ blocks included within the FOV are marked. The variables are iterated until all possible combinations have been checked and the blocks included in the fields of view have been marked. If the increment size is too small, checking all these possible combinations takes too long, but setting it too large means that gaps may appear in the marked area or that its edges appear very rough. The area marked for the variables of Table 5-2 is shown in Figure 5-2. Three single fields of

view are also shown to illustrate the varying size and shape of the FOV on the projection of the celestial sphere.

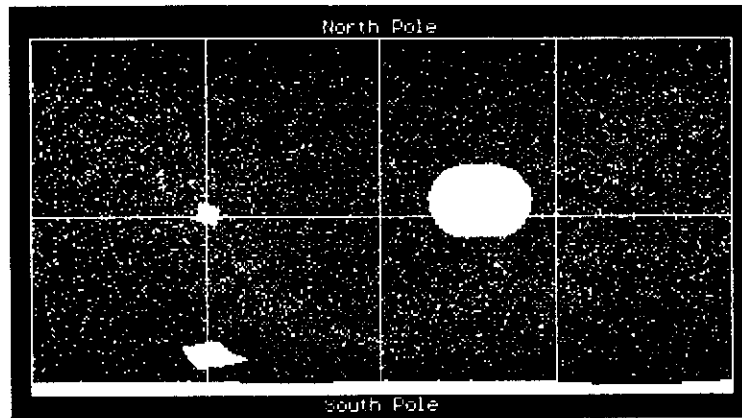


Figure 5-2 FOV on the celestial sphere over a period of two weeks (large white oval). Also shown are three single fields of view to illustrate how the projected size changes with declination. The FOV on the celestial equator (centre horizontal line) appears small. The one close to the South Pole appears larger while the FOV including the South Pole extends right around the projection.

Once the iteration of the attitude variables is complete and all possible $1^\circ \times 1^\circ$ blocks which may be included in a FOV have been marked, the catalogue stars appearing in the marked area are collected. These are stored as the sub catalogue. The sub catalogue generated from the marked area of Figure 5-2 contains 113 stars.

It is interesting to examine the effect of varying orbit parameters while the roll, pitch and yaw are kept within the limits listed in Table 5-2. The inclination determines the declination of the FOV. An inclination of 0° places the FOV over the South Pole. An inclination of 90° places it on the celestial equator and the 97° used to generate Figure 5-2 shifts it above the equator by 7° . The sum of the argument of perigee and true anomaly changes from 0° to 360° for each orbit. This only rotates the FOV around the boresight on the celestial sphere and does not cause any translation. During a period of two weeks the right ascension of the ascending node will change by about 15.4° (1.1° per day for 14 days). This shifts the FOV along the celestial equator. Over a period of a year, variation in the right ascension of the ascending node will cause the FOV to move right around the celestial sphere, as shown in Figure 5-3. This sub catalogue contains 1300 stars.

These positions of the FOV are verified by examining the coordinate transformations of Chapter 1.

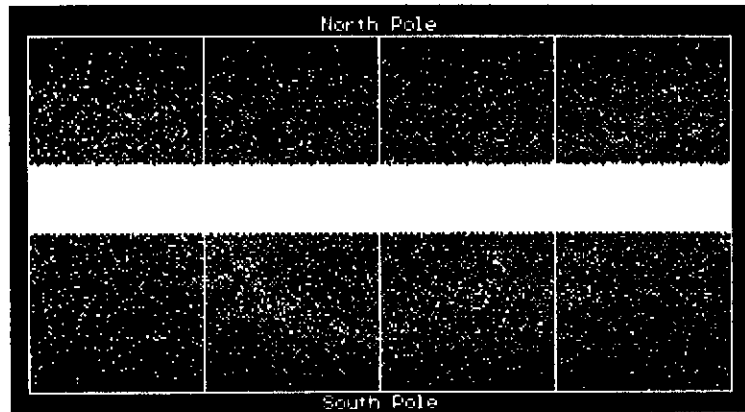


Figure 5-3 Position of the FOV over a period of a year, for the extreme attitude values listed in Table 5-2, but the right ascension of the ascending node changing from 0° to 360° .

Appendix G explains the procedure for downloading a catalogue to the Flash Ram of the star sensor. If required, the Modula programme *DistTab.mod* can import the catalogue and generate a distance table for use with the *Matching* module of Chapter 3.

6. Calibration

Calibration of the camera is essential since it determines the accuracy of angular measurement and the accuracy of pin pointing star centroids.

Ideally the CCD should be fixed at a perfect right angle to the optical axis and its effective image centre aligned with the axis, at precisely 25 mm (the focal length) from the optical centre of the lens. Even though this is unrealistic to achieve, the errors occurring due to misplacement of the CCD can be corrected. It is possible to estimate the position and orientation of the CCD relative to the optical axis and centre of the lens. Optical imperfections (lens aberrations) also contribute to errors. Coefficients describing these are estimated and used together with the CCD positional error to implement a mathematical camera model which corrects for measurement errors.

This Chapter describes a fast, simple and reliable method to calibrate the camera. No accurate test patterns or precise adjustments to equipment are required. No time consuming or complicated measurements need to be made. A single set of measurements on the optical bench and a few stellar images are required. One known constellation needs to be imaged to estimate the focal length. Once this has been done, only the lens aberrations need to be corrected. Angular distance measurement errors should have a standard deviation of no more than a pixel. The other images are matched to the catalogue with a Topspeed Modula version of the *Matching* algorithm. It returns all matching pairs of observed and catalogue stars. Data from these matches are used to correct the lens aberrations.

The first section of this chapter examines the errors expected due to misalignment of the CCD. The next section discusses the calibration techniques and camera model developed to correct the significant errors.

6.1 Measurement errors expected

This section examines the effects of mechanical positioning errors of the CCD on angular measurement. The effect of an incorrect focal length on the angle measured between two stars is examined in Section 6.1.1. Section 6.1.2 examines the effect of an error in the positional estimate of the optical centre of the CCD.

6.1.1 Focal length error

The actual focal length f_0 is the true distance from the optical centre of the lens, measured along the optical axis, to the point of intersection of the optical axis and CCD plane. During calibration the focal length f is estimated. An error in this estimate leads to an error in angles measured between stars, as illustrated in Figure 6-1.

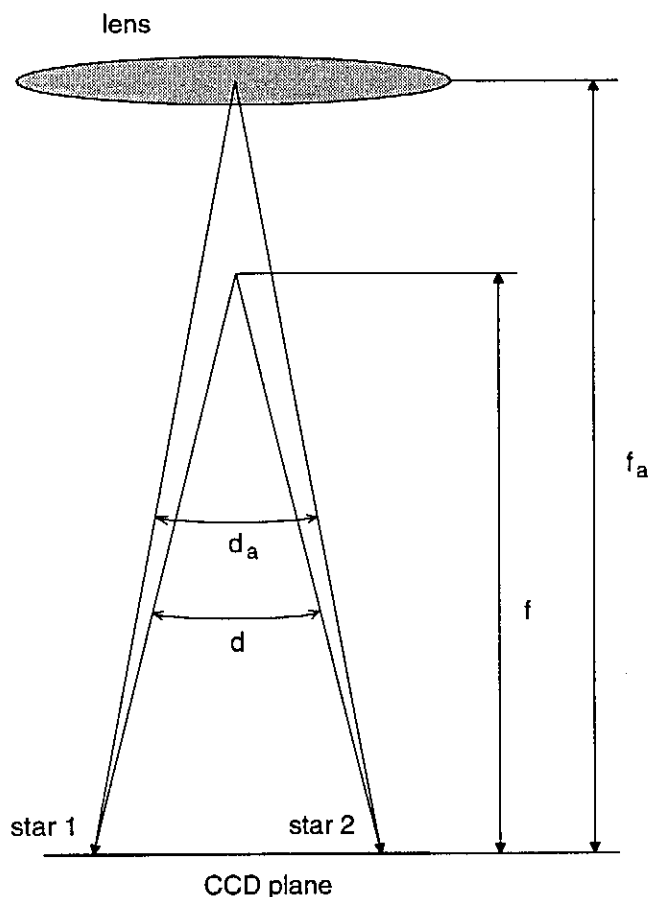


Figure 6-1 Actual angular distance d_a and measured distance d when incorrect focal length f is used. The error in focal length estimate is exaggerated.

The error is a function of the angular separation of the two stars as well as their positions on the CCD plane. To determine the effect of the position of the stars on the error, two imaginary stars with fixed separation were moved around the CCD plane and the error in their separation calculated. The absolute value of the angular error calculated for 2 stars separated by 0.5 mm is shown in Figure 6-2. Equation (C.7) was used to calculate the true angle d_a with focal length f_a , and measured angle d with focal length f , for each position. The error is the difference between the true and measured angle.

$$\varepsilon = d_a - d \quad (6.1)$$

For Figure 6-2, the error in focal length is $40\text{ }\mu\text{m}$ and the true focal length $f_a = 25\text{ mm}$. The pair of stars are moved around the CCD plane and the absolute value of the error is shown for each position. It reaches a maximum as the pair approaches the CCD centre. A pixel spans an average of about 0.23 mrad which means that the maximum error is only a fraction of a pixel. The variation in error is negligible.

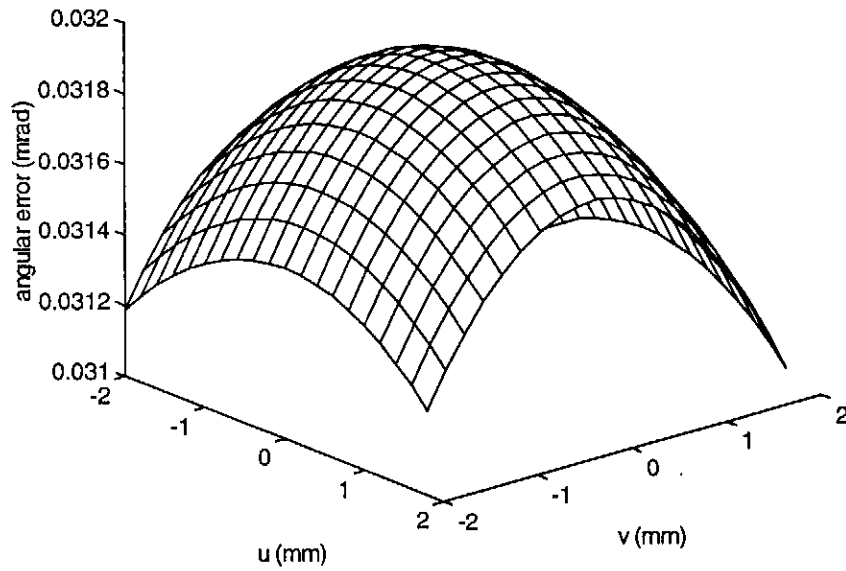


Figure 6-2 Angular error as the pair of stars (separated by 0.5 mm) move around the CCD plane.

To examine the effect of variation in focal length on the angular error, two stars were placed at opposite corners of the CCD i.e. one star at $(u_1, v_1) = (2.44, 1.82) \text{ mm}$ and the other at $(u_2, v_2) = (-2.44, -1.82) \text{ mm}$.

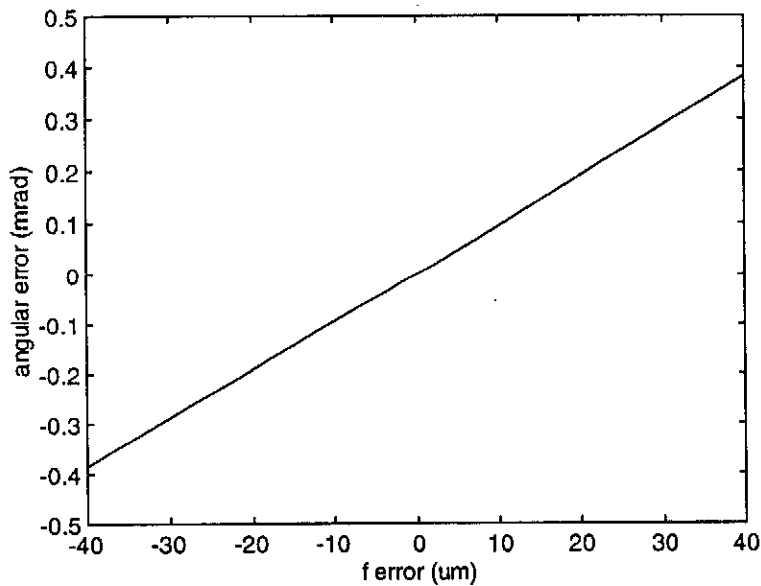


Figure 6-3 Angular error as the error in estimated focal length f varies from $-40 \mu\text{m}$ to $40 \mu\text{m}$, with true focal length $f_s = 25 \text{ mm}$.

The error in focal length is defined as the difference between true focal length f_a and the estimated focal length f .

$$f_{error} = f_a - f \quad (6.2)$$

This error is varied from $-40 \mu m$ to $40 \mu m$. Figure 6-3 shows that for a fixed position and separation between the pair of stars, an error in the focal length has a more or less linear effect on the angular error. Due to the large angular separation between the stars, the angular error is considerably larger than that of Figure 6-2. An error of only $25 \mu m$ in the focal length will lead to an angular error of 0.23 mrad or one pixel. This emphasizes the importance of estimating the focal length accurately.

6.1.2 Positional error of optical centre

During calibration the position of the optical centre is estimated. The effect of an incorrect estimate is examined here. The optical centre is calculated in *horizontal-vertical* coordinates and the uv coordinate system is then centred on this point. As described in Chapter 1, the uv coordinate reference point is related to the optical centre in *horizontal-vertical* coordinates by

$$(0,0)_{uv} = (\text{horoptcentre}, \text{vertopticentre})_{\text{horvert}} \quad (6-3)$$

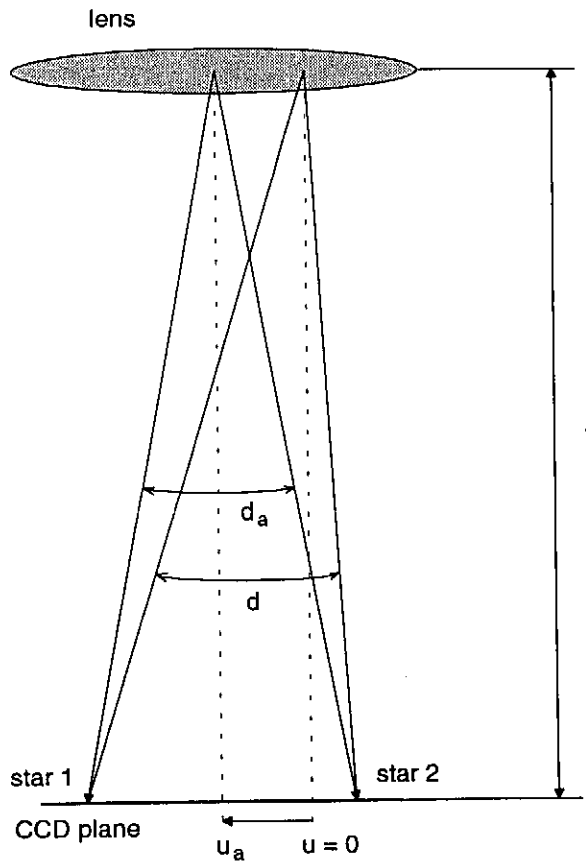


Figure 6-4 Angular error occurring due to incorrect position of optical centre.

The true optical centre is assumed to be located at (u_a, v_a) . Figure 6-4 illustrates why a measurement error occurs.

The angular distance d differs from the true angular distance d_a , due to the incorrect positional estimate of the optical centre. A larger angular separation leads to a larger error. This is shown in Figure 6-5. Star 1 is moved from $(u_1, v_1) = (0, 1.5)$ to $(u_1, v_1) = (2, 1.5)$ while star 2 is fixed at $(u_2, v_2) = (0, -1.5)$. The error in the positional estimate of the optical centre is increased from 0 mm to $u_a = 0.65$ mm or 100 pixels.

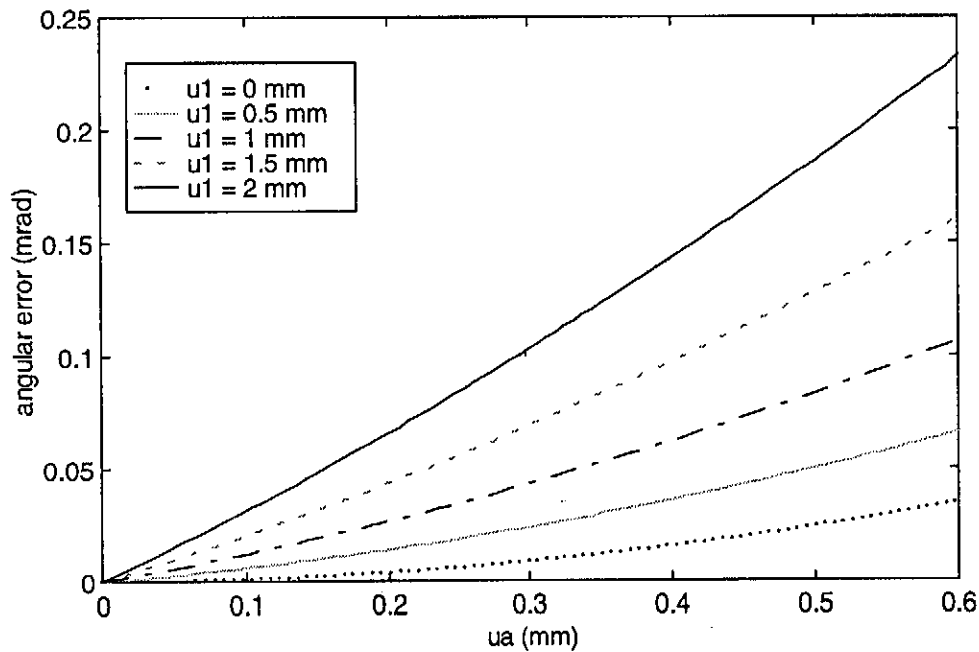


Figure 6-5 Angular error when the estimated optical centre shifts from the actual position ($u_a = 0$ mm) to $u_a = 0.65$ mm. Star 1 moves from $(u_1, v_1) = (0, 1.5)$ to $(u_1, v_1) = (2, 1.5)$ while star 2 is fixed at $(u_2, v_2) = (0, -1.5)$

The angular error increases as the positional error of the optical centre increases. An error was only introduced in the u direction, but due to the symmetry of the system, an error in the v direction will have a similar effect. The errors may add or cancel one another, depending on the error directions and star positions. Figure 6-5 shows that an error of $u_a = 0.3$ mm may cause an angular error of about half a pixel. It is concluded that the position of the optical centre needs to be estimated with great accuracy to ensure precise angular measurement.

The effects of focal length errors and optical centre errors have been examined independently, but the combined effects may once again add or cancel. This section has pointed out that both the focal length and the optical centre position need to be determined precisely. The next section explains how this is done and discusses a camera model to correct for lens aberrations and a tilted CCD.

6.2 Calibration procedure

Calibration of most high precision optical instruments is performed with sophisticated equipment. As an example, the test setup used by Weng [15, p. 976] has a test pattern which is accurate to $50\text{ }\mu\text{m}$ and the position can be set to an accuracy of $50\text{ }\mu\text{m}$. Weng derives a complete mathematical camera model which includes parameters such as focal length and lens aberration parameters. Data obtained from images taken with this test setup is used to estimate all the coefficients of the camera. This camera model is used to correct the measured positions of objects mathematically. Since Sunsat does not have access to such sophisticated testing equipment, a more basic approach to calibration was followed. It relies upon a normal optical bench and imaging of known stellar constellations. Instead of solving all the coefficients of a camera model simultaneously, they are solved individually. Factors causing the greatest errors are examined first and those causing smaller errors are examined after these have been solved.

As a first attempt to measure distortion, an array of dots with 20 mm separation was printed on a laser printer and imaged at a distance of 1 m . The object distance could not be measured precisely and consequently it was not possible to estimate the object size (or true distance between two dots) with great precision. The distance between the *same* pair of dots on the CCD plane was, however, measured with great precision. When rotating the camera (and keeping the object distance fixed) the variation in distance between the dots on the CCD plane was found to be a fraction of a pixel. In other words, the true distance between dots could not be measured precisely, but the variation in distance between dots on the CCD plane could be determined with sub pixel accuracy.

When imaging the array of dots, the differences in distances between dots on the CCD plane was considerably larger than a pixel. Since the difference did not occur when the same pair of dots was imaged on different sections of the CCD, the variation in distance could only be ascribed to printer inaccuracies. This variation in the distances between dots was absolutely random and completely overwhelmed any form of lens aberrations. The lack of an accurate test pattern ruled out the possibility of measuring lens aberration with an array of dots.

The following calibration procedure was developed for the Sunsat star sensor, using principles introduced by Weng [15] where possible.

1. Determine the point of intersection of the optical axis and CCD, as described in Section 6.2.1.
2. Determine the focal length of the camera with the procedure discussed in Section 6.2.2.
3. Calculate the distortion coefficients to implement a distortion model, discussed in Section 6.2.3.

Steps 1 and 2 should already provide angular accuracy in the order of a pixel size (i.e. 0.23 mrad).

6.2.1 Locating the optical centre

Due to the lack of an accurate test setup, a basic method of determining the optical centre was used. This was done by noting that any stationary imaged object rotates around the optical centre of the CCD when the camera (lens and CCD) is rotated around the optical axis.

To rotate the camera about the optical axis, two aluminium disks were machined to fit tightly around the front of the lens and the lens mount (between the lens and CCD). Two 2" water pipe clamps are used to clamp the camera into a V-clamp which is mounted on the optical bench. The pipe clamps are tightened so that the camera can rotate smoothly, but with the minimum axial motion.

The ideal imaged object is a bright dot on a black background. Photographic negative (used for PCB layouts) with a small transparent dot is mounted on the opposite end of the optical bench and a desk lamp shone from behind. The negative is large enough to fill the entire image, which simplifies the process of extracting the imaged dot from the background. The dot is covered with white paper on the back side (the desk lamp side) to smoothen the image. The dot should cover at least 30 pixels and not more than a few hundred. Ensuring that it does not saturate will give the greatest centroid accuracy."

By repeatedly imaging the dot and adjusting its position, its rotation radius around the optical centre can be decreased until it hardly shifts when the camera is rotated. At this point it should be close to the optical centre. Without much effort it can be adjusted to remain within an area of about 12 by 12 pixels. The optical centre should be somewhere near the centre of this area. Measurements taken with the engineering model of the Sunsat star sensor are shown in Appendix F, Figure F-2.

The following sequence is used to locate the optical centre:

1. Move the dot until it is close to the optical centre. Save a few centroids of the dot, while each time rotating the camera. The imaging programme run on the star sensor transputer extracts the centroids of the dots and returns them to the root transputer. The root transputer saves the centroids to a file.
2. Estimate the centre by examining the positions of the imaged dot.
3. The position of the optical centre (*horoptcen*, *vertoptcen*) is set in the definition module of the star sensor library *StarLib.def*. This defines the centre of the *uv* coordinate system which is used by all other software modules.

The positional accuracy of the optical centre is expected to be a few pixels or better. Figure 6-5 shows that the angular distance measured between stars will not be affected significantly by such a small error in the position of the optical axis. Due to this small error, all star positions will be referenced to a point which deviates from the true optical centre. Limitations on the accuracy with which the camera is mounted on the roof of the satellite, will cause a further alignment error between the boresight of the actual star sensor and that shown in Figure 1-4. These two errors will not have any

effect on the star matching algorithms, but will lead to incorrect orientation calculations. The error is fixed and may be estimated when a known object is imaged by the earth pointing imager, which will also have some mounting error.

6.2.2 Estimating the focal length

The focal length may be estimated using an optical bench setup, but it is not possible to adjust the object distance with the desired accuracy. An error in the object distance will lead to an error in focal length estimate as well. The greatest accuracy is achieved by imaging known stars and measuring the angular separation between two of them. Since lens aberrations and errors due to a slanted CCD¹ are a minimum close to optical axis, it is preferable to use two stars close to the CCD centre. Their angular separation is given by (C.7). In this case the angular separation θ is known from the star catalogue and equation (C.8), while the focal length f is required. Equation (C.7) is solved for f which is the only unknown. This gives a fourth order equation

$$af^4 + bf^2 + c = 0 \quad (6.4)$$

with

$$\begin{aligned} a &= \frac{1}{(\cos\theta)^2} - 1 \\ b &= \frac{2(u_1u_2 + v_1v_2)}{(\cos\theta)^2} - u_1^2 - v_1^2 - u_2^2 - v_2^2 \\ c &= \frac{(u_1u_2 + v_1v_2)^2}{(\cos\theta)^2} - u_1^2u_2^2 - u_1^2v_2^2 - v_1^2u_2^2 - v_1^2v_2^2 \end{aligned}$$

which is solved as

$$\begin{aligned} f_+^2 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ f_-^2 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \end{aligned} \quad (6.5)$$

Only the first of these values is positive so that

$$f = \sqrt{f_+^2} \quad (6.6)$$

The Modula programme *Estf.mod* uses equations (6.4) to (6.6) to estimate the focal length. An initial estimate is obtained by imaging a known constellation, with two stars close to the image centre. The *Matching* algorithm can then be used to find the catalogue matches for all other images. To minimize the error in the estimate, a few images should be taken with stars close to the centre. The average value of the focal

¹ These are discussed in the next section.

length calculated from all the images is taken. The estimated value of f is set in the library definition module *StarLib.def*. This is the last variable required to define the relation between the uv and star sensor (ss) coordinate systems.

6.2.3 Distortion model

Having determined the optical centre and the focal length, errors may still occur due to variation in image magnification. Lens aberrations and a slanted CCD may cause variation in image magnification on different sections of the CCD. The size of an imaged object close to the CCD edge will differ from the image size close to the optical centre. This section describes a camera model developed to correct these errors.

The CCD may be soldered in such a way that it is not perfectly perpendicular to the optical axis. This slant in the CCD plane will lead to errors in the distance measured from the optical axis to an imaged dot. Figure 6-6 shows a CCD which is tilted through angle θ from the vertical. Appendix F.1 contains the details.

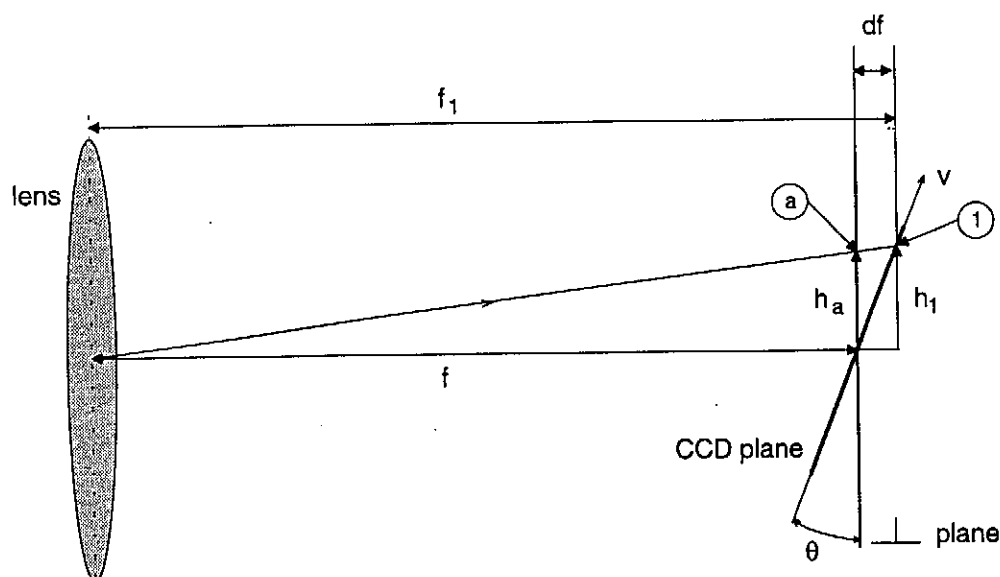


Figure 6-6 Error occurring due to tilted CCD plane

Consider a dot of which the image appears at point 1 on the CCD plane. Its position in uv coordinates on the CCD plane is $(0, v_1)$. The u axis is directed into the paper. If the CCD plane is perpendicular to the optical axis, the dot will appear at point a , or $(0, v_a)$. When calculating the distance between the dot and the optical centre, the distance $v_a = h_a$ is desired. This distance will be calculated incorrectly as v_1 for two reasons. Firstly the greater focal length will cause larger magnification and the image will appear further from the optical axis. Consequently,

$$\frac{h_1}{h_a} = \frac{f_1}{f} \quad (6.7)$$

Secondly the v coordinate (i.e. the distance measured along the CCD plane) calculated for the dot at point 1 will be

$$v_1 = \frac{h_1}{\cos\theta} \quad (6.8)$$

Replacing (6.7) in (6.8) with $v_a = h_a$ gives

$$v_1 = \frac{v_a}{\cos\theta} \frac{f_1}{f} \quad (6.9)$$

Equation (6.9) gives the relation between the measured v coordinate v_1 and the desired v_a , which is the perpendicular distance measured from the optical axis. It holds for any point on the CCD for which the focal length of the dot (f_1) differs from the focal length of the optical centre (f). The CCD may of course be tilted in any direction, not necessarily the vertical. Replacing the v coordinate, used for illustrative purposes, with the radial distance from the optical centre, r , gives

$$r_1 = \frac{r_a}{\cos\theta} \frac{f_1}{f} \quad (6.10)$$

where θ is the angle² between r_1 (the radial distance measured to a point on the CCD plane) and r_a (the radial distance to the point on the plane perpendicular to the optical axis). This gives the distortion in the radial distance due to a slanted CCD. Optical imperfections in the lens cause further distortion of the image. At a given radius from the optical centre, this distortion should be a fixed value. The measured radial distance will then be

$$r = \frac{r_a}{\cos\theta} \frac{f_1}{f} + \Delta r_{lens} \quad (6.11)$$

where

Δr_{lens} = lens aberration and is a function of r

Appendix F.1 shows how this is rewritten as

$$r = \frac{\frac{r_a}{\cos\theta} + \Delta r_{lens}}{1 - \frac{r_a}{f} \tan\theta} \quad (6.12)$$

and

² Appendix F and Figure F-1 contain details

$$r_a = \frac{\cos\theta(r - \Delta r_{lens})}{1 + \frac{r}{f} \sin\theta} \quad (6.13)$$

with

$$\theta = \arcsin[\sin\theta_{max} \cos(\phi - \phi_{max})] \quad (6.14)$$

where ϕ is the angle measured clockwise from the positive u axis to the point. The maximum angle between the CCD and perpendicular plane is θ_{max} , which occurs where $\phi = \phi_{max}$.

If the parameters θ_{max} , ϕ_{max} and Δr_{lens} can be determined, equation (6.13) gives the true radial distance of a point from the optical axis when its measured radial distance is known. The next section explains the procedure used to estimate these parameters.

6.2.3.1 Estimating the distortion parameters

Lens aberrations may vary with object distance and lens aperture. While practically no lens aberrations could be observed on the optical bench with a small lens aperture, considerable distortion was measured when imaging stars with a large aperture. The two types of distortion in equation (6.12) could thus be examined independently³. The slant of the CCD is determined from measurements on the optical bench, where lens aberration is a minimum. This is corrected, after which stars are imaged and the lens aberrations determined.

The slant of the CCD is determined in a similar manner as was used to estimate the optical centre. A pair of dots is imaged, with one dot close to the optical centre and the other close to the CCD edge. A large radial distance between the dots is desirable since it increases the distortion. It is only necessary to measure a single set of data to locate the optical centre and estimate the slant. The centre dot is used to locate the centre and the distance between the dots gives an indication of the slant.

The variation in distance can be measured with sub pixel accuracy. If, while rotating the pair of dots, the camera is accidentally twisted slightly around the optical axis, the distance r_{12} between the dots will not change, even though their position will change slightly. The variation in r_{12} as the camera is rotated, gives an indication of the slant of the CCD. Since lens aberration is neglected, equation (6.12) reduces to

³ Optical distortion is of course not zero when imaging on the optical bench, but the effect of optical distortion on distortion due the CCD slant and vice versa, is negligible. This justifies the method of examining the two types of distortion independently.

$$r_{12} = \frac{\frac{r_a}{\cos \theta}}{1 - \frac{r_a}{f} \tan \theta} \quad (6.15)$$

This curve is used to approximate the measured data. A least squares approximation is used to estimate the unknowns θ_{max} , ϕ_{max} (of which θ is a function through equation (6.14)) and r_a . The value determined for r_a is of no further significance, but those estimated for θ_{max} and ϕ_{max} are used in the distortion model. Measured values and the slant parameters estimated for the engineering model of the star camera are given in Appendix F-2.

Once the tilt has been corrected, the lens aberration is determined. This is done by examining data obtained from stellar images. The angular distance measured between stars can be determined with sufficient precision to enable the *Matching* algorithm to find star matches. A large number of star matches can be obtained in this way without much effort.

Lens aberration is usually a function of radial distance from the optical centre. In most cases it can be described by a high order polynomial of the form:

$$\Delta r_{lens} = k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4 + \dots \quad (6.16)$$

This equation does not include the effect of tangential distortion, but should describe radial distortion, coma and field curvature. Tangential distortion [15, p.969] causes an imaged point to appear displaced in the direction normal to the radial line to the point. The effect of tangential distortion is hard to measure and should be minimal. Radial distortion [15, p.968] displaces a point in the radial direction. If the point appears closer to the optical centre than it should, it is referred to as barrel distortion. If it is displaced outward, it is referred to as pincushion distortion. For an off-axis object, coma [9, p.212] is observed when various zones of the lens magnify images by different amounts. Field curvature [9, p.213] causes a plane normal to the optical axis to be imaged as a curved surface.

To determine the lens aberrations, a few images with a star close to the optical centre are required. These are obtained by imaging a densely populated section of the celestial sphere. At least half the images have a star sufficiently close to the optical centre. The angular distances between the centre star and all other stars in the image are measured. The catalogue distances between the centre star the other stars are used as the correct distances. The error is defined as the difference between the measured distance and the catalogue distance

$$\Delta r_{lens} = r - r_a \quad (6.17)$$

where

r = measured angular distance
 r_a = catalogue distance

The measured error is plotted for a large number of values and approximated by the polynomial of equation (6.16). Data measured with the engineering model star sensor is shown in Appendix F.2.

All the parameters of the distortion model have been determined and the complete model is implemented to correct the errors.

6.2.3.2 Using the distortion model

The following sequence is used to find the correct uv coordinate point (u_a, v_a) of an imaged point (u, v) :

The radial distance

$$r = \sqrt{u^2 + v^2} \quad (6.18)$$

and the angle between the positive u axis and the point is calculated as

$$\tan \phi = \frac{v}{u} \quad (6.19)$$

These measured values are used in equation (6.13) to estimate the true radial distance r_a . The correct uv coordinates are then

$$\begin{aligned} u_a &= r_a \cos \phi \\ v_a &= r_a \sin \phi \end{aligned} \quad (6.20)$$

This correction is not applied to each imaged pixel of a star, but only to the centroids of stars.

6.3 Results

The final results obtained after calibration are presented in this section. The data was generated from 20 images. The 4385 distances between 354 stars were measured and compared to the corresponding catalogue distances. The error was calculated from equation (6.21) and the histogram showing the distribution of the errors is shown in Figure 6-8. Each column of the histogram spans $10 \mu rad$. The distribution of the error is expected to have a Gaussian shape. The histogram is approximated by the normal distribution shown as the solid line. It was generated by calculating the standard deviation of the measured data and equating the area under the curve to that under the histogram.

$$\varepsilon_{dist} = d_m - d_{cat} \quad (6.21)$$

where

d_m = measured angular distance

d_{cat} = corresponding catalogue distance

The standard deviation of the distance error is:

$$\begin{aligned}
 \sigma_{dist} &= 78.2 \mu rad \\
 &= \frac{78.2}{230} pixels \\
 &= 0.34 pixels
 \end{aligned} \tag{6.22}$$

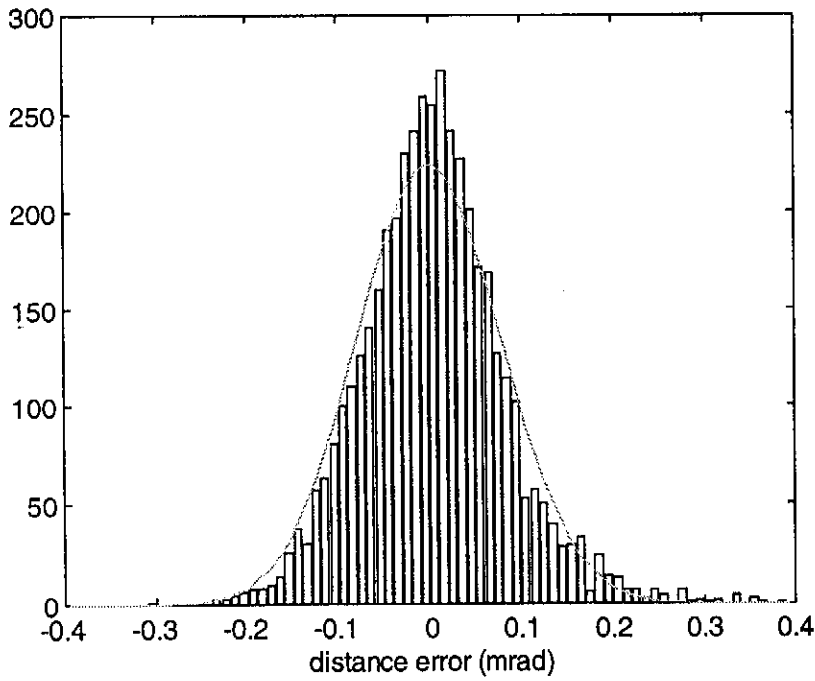


Figure 6-8 Histogram of the errors in angular distances between stars. Each bar of the histogram spans $10 \mu rad$ and a total of 4385 distances were measured.

If the standard deviation of the distance between two points is σ_{dist} , it means that the error in the position of each point (when referenced to some fixed position such as the optical centre) is $1/\sqrt{2}$ times σ_{dist} . The standard deviation of the positional error of a point is:

$$\begin{aligned}
 \sigma_{pos} &= \frac{\sigma_{dist}}{\sqrt{2}} \\
 &= 55.3 \mu rad \\
 &= 11.4'' \\
 &= 0.24 pixels
 \end{aligned} \tag{6.23}$$

When referenced to the optical centre, the position of a star can be pin pointed with an accuracy (1σ) of better than $12''$.

The magnitude is estimated with the sequence described in Appendix C.1. The sample to illuminance conversion factor was calculated as

$$k_{illum} = 16 \times 10^{-9} \quad (6.24)$$

The dynamic range of the star sensor is limited to M_v 2.5 to 6. This eliminates the use of the 35 brightest stars. Stars brighter than magnitude 3 start to saturate and consequently their magnitudes are estimate too high (i.e. too faint). Using only stars within this dynamic range, the magnitudes of 354 imaged stars were compared to their corresponding catalogue values. The magnitude error is defined as

$$\epsilon_{mag} = M_{v_m} - M_{v_{cat}} \quad (6.25)$$

where

M_{v_m} = measured magnitude

$M_{v_{cat}}$ = corresponding catalogue magnitude

The histogram of the error distribution is shown in Figure 6-9. The Gaussian approximation to the histogram is shown as a solid line. Each bar spans $0.1 M_v$ and 354 errors are entered in the histogram. Two errors of magnitude greater than one occurred but are not shown in the histogram.

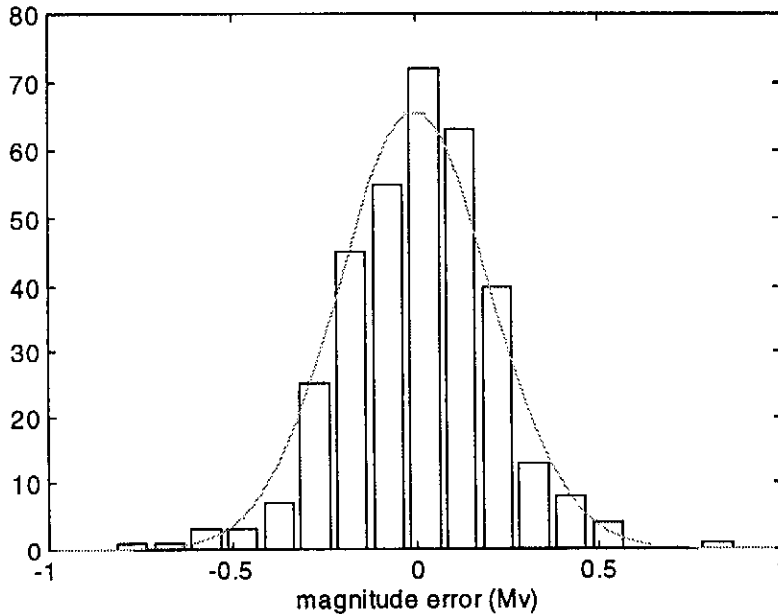


Figure 6-9 Histogram of the magnitude error. Each bar spans $0.1 M_v$ and 354 errors were entered.

The standard deviation of the magnitude error is

$$\sigma_{mag} = 0.2 M_V \quad (6.26)$$

With the dynamic range of the sensor limited to M_V 2.5 to 6, the standard deviation of the magnitude error is $0.2 M_V$.

6.4 Conclusion

The factors causing major errors in the angular distance are

- misalignment of the CCD centre and optic centre
- incorrect focal length

Once these have been corrected, remaining errors are corrected with a distortion model which takes lens aberrations and a tilted CCD into account. The entire calibration procedure is completed with a single set of measurements taken on the optical bench, one image of a known constellation and a few more stellar images. Results obtained from the distortion model yield a positional error (1σ) of a quarter of a pixel and a magnitude error of $0.2 M_V$.

7. External Factors Affecting the Star Sensor

Many external factors may affect the normal operation of the star sensor and a few actually do. It is essential to determine which factors have an effect on the sensor's operation and which are negligible. This chapter examines these external factors which may influence the star sensor. Procedures are developed to correct for those which do have a significant influence.

The previous chapter showed that the angular position of a star can be measured with an accuracy¹ (1σ) of at least $60\ \mu\text{rad}$ or $12''$.

7.1 Parallax

Parallax is the apparent motion of a star against the celestial sphere, when viewed from different directions. As illustrated in Figure 7-1, near stars appear to move relative to the fixed background of very distant stars when the observer moves large distances. Parallax is largest when the observer moves the largest distance and the star is the closest. The greatest parallax is experienced when the closest star, Alpha Centauri, is viewed from opposite sides of the sun. Even this is only $0.76''$ or 2×10^{-4} degrees [16, p. 198], which is not significant in comparison with the accuracy of the star sensor. Parallax of further stars and parallax due to motion of the satellite around the earth is negligible as well.

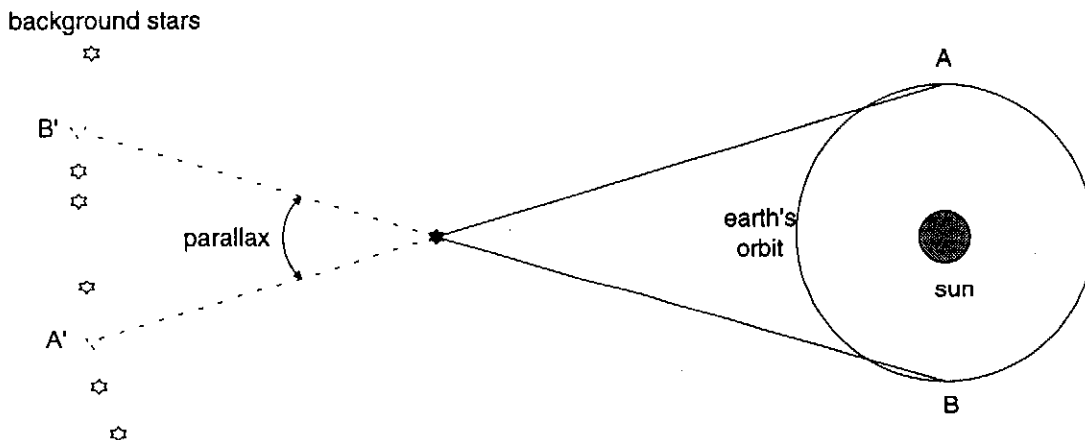


Figure 7-1 Parallax of close stars due the earth's rotation about the sun

¹ $100\ \mu\text{rad} = 0.0057^\circ = 0.34' = 20.63''$

7.2 Precession of the equinoxes

The celestial poles and equator do not remain fixed on the celestial sphere, but move around in a predictable fashion. The gravitational influence² of the sun and moon on the equatorial bulge of the earth causes its spin axis to wobble. Instead of remaining fixed in inertial space, it describes a circle on the celestial sphere. The period of this oscillation is 25765 years [16, p. 20]. This phenomena is known as precession of the equinoxes³, since it results in the First Point of Aries sliding along the ecliptic at a rate of 50.3" per year. The date at which the position of the First Point of Aries is used as the right ascension $\alpha = 0$, is referred to as the Epoch. It is usually updated every 50 years. The next update is due in the year 2000 and consequently Sunsat will still use Epoch 1950.

Both the star catalogue and the orbit coordinate system are referenced to Epoch 1950. As long as both are referenced to the same date, no corrections are required.

7.3 Proper Motion

Although all stars move through the heavens, the effect is only apparent for the nearest stars which move large distances. The motion of a star can further be resolved into two components, one in the line of sight and the other perpendicular to it. Only the component perpendicular to the line of sight is visible and is known as the proper motion of the star. Barnard's star has the largest proper motion, of 10.25" per year [16, p. 227]. Its magnitude is however 13.4 which is well outside the detectable range of the star sensor. When proper motion of a star is known, it can be updated linearly over a period of hundreds of years. Only 5% of the stars brighter than ninth magnitude have proper motion larger than 10" and only 0.1% have proper motion more than 1' per century [2, p. 144]. A table of the stars with the most significant proper motion is given in the Appendix A.

The maximum proper motion of a detectable star is 3.68" which is less than the accuracy expected from the star sensor. Proper motion will have no significant influence on the operation of the star sensor.

7.4 Variable stars

The magnitude of most stars remains constant with time, but that of variable stars changes with time. The period of variation as well as the fluctuation in magnitude is often fixed and predictable for a specific star, but in many instances it is not. Appendix A contains a list of the most prominent variable stars.

² The equatorial plane of the earth is slanted by 23.5° relative to the ecliptic plane. Since the sun and moon are in the plane of the ecliptic, they tend to draw the equatorial bulge toward this plane. This sideward force on the spinning earth causes its axis to wobble.

³ Refer to Appendix A

A few examples [17, p. 53] of variables stars should explain why it is virtually impossible to compensate for all cases:

1. **Cepheids variables**

These variables vary over a period of only a few days but the amplitude variation is never more than 0.9 Mv. Delta Cephei's magnitude varies as regular as clockwork, between 3.48 and 4.37 in 5.366 days.

2. **Mira Ceti variables**

The variation in magnitude is much larger for these stars, but the period is much longer as well. The magnitude of O Ceti varies between 2 and 10 during a period of about 331 days. The period does not vary by more than a few days.

3. **Semi regular variables**

The period of a semi regular star may vary by months or even years. The magnitude of Beutelgeuse changes from 0.4 to 1.3 in about 6 years.

4. **Irregular variables**

They suddenly change their magnitude in a totally unpredictable fashion. The R Coronae Borealis stars shine steadily most of the time and then suddenly dims by 5 to 8 magnitudes. η Car changes magnitude from -0.8 to 7.9.

5. **Novae**

A novae is a star which explodes. The magnitude may suddenly increase up to ten fold and remain so for months or years, fading all the time. On average, two or three bright novae occur a year.

6. **Recurrent novae**

These stars explode more than once. T Coronae Borealis is normally magnitude 10, but in 1866 it reached magnitude 2 and in 1946 magnitude 3.

7. **Supernovae**

During the past millennium only three have been recorded. An increase of up to 20 magnitudes may occur.

Only variables belonging to the first two categories are practical to be corrected. The amplitude of variation of those in category one is not very large and there are only about six stars belonging to the second category which are worth updating. These magnitudes may be corrected before sending the sub catalogue to the satellite. Stars belonging to categories 3 to 7 are unpredictable and it is impossible to correct for these cases. They will appear on the image as false stars, since their observed magnitudes may not always match the catalogue value.

7.5 Aberration

Since the speed of light is finite, the positions of stars appear shifted [2, p. 144] when observed from an object moving with a speed which is significant compared to that of light. Referring to Figure 7-2, assume that the light from S enters the lens at time $t = 0$. At this instant, the lens is at point C and the CCD at point A . After time t_1 has elapsed, the light has covered the distance ct_1 and strikes the CCD, which has moved vt_1 to the right and is situated at A' . Consequently the light appears to be coming from S' instead of from S . If the satellite were stationary in inertial space, the observed star vector would be pointing in the direction $A'C$, but since the satellite is moving the observed star vector is pointing in the direction $A'C'$. The star appears shifted towards the direction of motion by angle α .

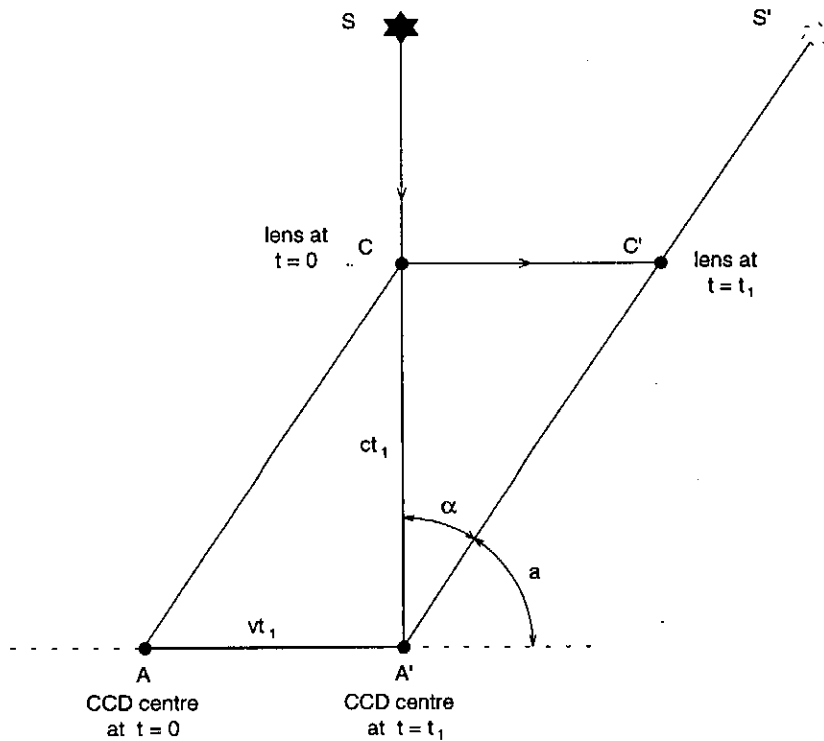


Figure 7-2 Aberration due to velocity of the satellite. The star appears at S' instead of its true position S .

The angle α is computed from the sine law:

$$\frac{\sin \alpha}{vt_1} = \frac{\sin a}{ct_1} \quad (7.1)$$

so that

$$\alpha = \sin^{-1}\left(\frac{v}{c} \sin a\right) \approx \frac{v}{c} \sin a \quad (7.2)$$

Two velocities are significant when compensating for aberration, i.e. the velocity of the earth rotating around the sun and the velocity of the satellite orbiting the earth. The worst case where $a = 90^\circ$ is examined to determine the effect of aberration.

7.5.1 Worst case aberration

The velocity of the earth rotating around the sun is $v_{earth,sun} = 29.785 \text{ km/s}$ [16, p 71]. For light incident perpendicular to the direction of motion, \underline{v}_s , the angle $\alpha = 90^\circ$ which gives an aberration of

$$\alpha_{earth,sun} = \frac{v_{earth,sun}}{c} = \frac{29.785}{299688} = 99.39 \times 10^{-6} \text{ rad} = 20.5'' \quad (7.3)$$

The earth orbiting velocity of the satellite has to be taken into account as well. This velocity is estimated as follows: Adding the equatorial radius of the earth and the altitude of the satellite, the orbiting radius of satellite is approximated as $R_{sat} = 6378 + 700 = 7078 \text{ km}$. The circumference of the orbit is $2\pi R_{sat} = 44472 \text{ km}$. Assuming that the orbit period is about 100 minutes, the velocity is estimated as $v_{sat,earth} \approx 26700 \text{ km/h} = 7.4 \text{ km/s}$. Again considering the worst case of perpendicular light, the aberration is

$$\alpha_{sat,earth} = \frac{v_{sat,earth}}{c} = \frac{7.4}{299688} = 24.69 \times 10^{-6} \text{ rad} = 5.1'' \quad (7.4)$$

From the above results it is clear that significant errors may occur due to aberration and need to be corrected. The next section formulates a procedure to determine the aberration, and correcting for it.

7.5.2 Calculating and correcting aberration

The angle a in (7.2) is obtained from the dot product of the velocity vector and the observed star vector. In the plane of these two vectors, the observed star vector is shifted through angle α , opposite to the direction of the velocity vector. Instead of estimating the orientation, calculating the velocity in body coordinates and correcting the observed star vectors, the reference vectors are corrected. Each reference vector needs to be shifted through angle α in the direction of the velocity vector.

7.5.2.1 Determining the velocity vector

The first step is to determine the velocity vector in orbit coordinates. It is the vector sum of the velocity of the earth around the sun and the satellite around the earth, both in the orbit (o) coordinate system.

$$\underline{v}_{sat,o} = \underline{v}_{sat,earth,o} + \underline{v}_{earth,sun,o} \quad m/s \quad (7.5)$$

The direction of the satellite's velocity around the earth, in orbit coordinates, is approximately in the x_o axis direction⁴.

$$\begin{aligned} \underline{v}_{sat,earth,o} &= v_{sat,earth} \hat{x}_o \\ &\approx 7400 \hat{x}_o \quad m/s \end{aligned} \quad (7.6)$$

The direction of the velocity of the earth around the sun is determined from the cross product of the sun pointing unit vector (\hat{p}_{sun}) and ecliptic normal (\hat{p}_{en}). The sun pointing unit vector is the vector pointing from the earth to the sun. It is obtained by converting the right ascension and the declination of the sun to celestial coordinates.

$$\hat{p}_{sun,c} = cstoc(\alpha_{sun}, \delta_{sun}) \quad (7.7)$$

The ecliptic is slanted at 23.5° relative to the celestial equator. The normal to the ecliptic $\hat{p}_{en,c}$ is obtained by rotating z_c through -23.5° around the axis pointing toward the First Point of Aries, x_c .

$$\begin{aligned} \hat{p}_{en,c} &= \sin(-23.5^\circ) \hat{y}_c + \cos(-23.5^\circ) \hat{z}_c \\ &= -0.3987 \hat{y}_c + 0.9171 \hat{z}_c \end{aligned} \quad (7.8)$$

The direction of the earth's velocity around the sun, in celestial coordinates (c) is then

$$\hat{v}_{earth,sun,c} = \hat{p}_{sun,c} \times \hat{p}_{en,c} \quad (7.9)$$

so that the velocity of the earth around the sun is

$$\begin{aligned} \underline{v}_{earth,sun,c} &= v_{earth,sun,c} \hat{v}_{earth,sun,c} \\ &= 29785 \hat{v}_{earth,sun,c} \end{aligned} \quad (7.10)$$

This velocity vector is transformed to orbit coordinates.

$$\underline{v}_{earth,sun,o} = A_{c \rightarrow o} \underline{v}_{earth,sun,c} \quad (7.11)$$

⁴ See Section 1.4

Equation (7.5) sums the velocity of the satellite around the earth with the velocity of the earth around the sun to obtain the total velocity. Once the velocity is known, the aberration is calculated.

7.5.2.2 Calculating the aberration

The angle a in (7.2) is calculated from the dot product of the satellite's velocity and the reference vector:

$$a = \arccos\left(\frac{\underline{v}_{sat,o} \cdot \underline{ref}_o}{v_{sat,o} \, ref_o}\right) \quad (7.12)$$

and (7.2) is used to calculate the aberration

$$\alpha = \frac{v_{sat,o}}{299\,688 \times 10^3} \sin a \quad (7.13)$$

7.5.2.3 Correcting the aberration

The reference vector is shifted through an angle α in the plane of the velocity and reference vectors. To accomplish this, a new coordinate system referred to as the *velocity* coordinate system, is defined. The z_v axis is perpendicular to the plane of the velocity and the reference vector. It is used as the rotation axis.

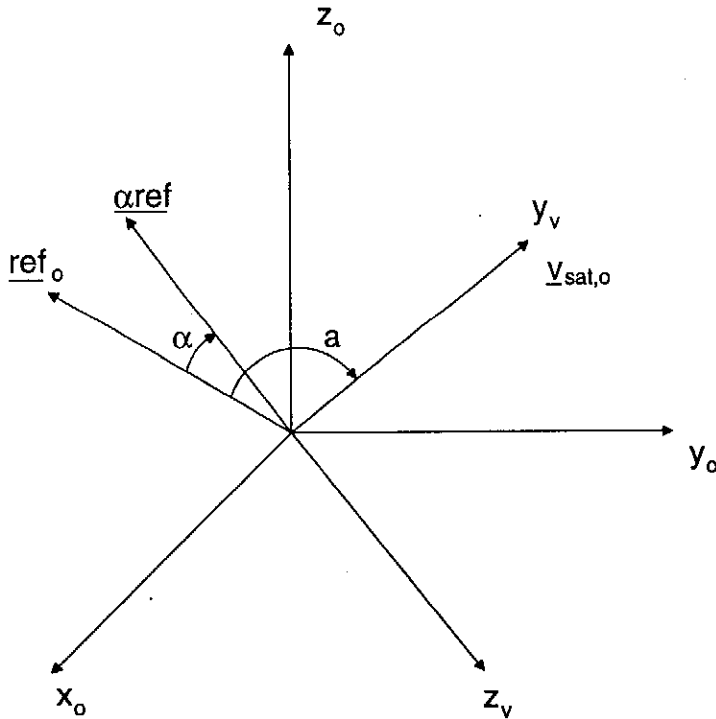


Figure 7-3 Rotation of the reference vector through angle α to correct for aberration

$$\hat{\underline{z}}_v = \frac{\underline{ref}_o \times \underline{v}_{sat,o}}{\underline{ref}_o \cdot \underline{v}_{sat,o}} \quad (7.14)$$

The y_v axis is parallel to the velocity vector

$$\hat{\underline{y}}_v = \frac{\underline{v}_{sat,o}}{v_{sat,o}} \quad (7.15)$$

and the x_v axis is chosen to complete the right hand positive set of axes

$$\hat{\underline{x}}_v = \hat{\underline{y}}_v \times \hat{\underline{z}}_v \quad (7.16)$$

The reference vector is converted from *orbit* to *velocity* coordinates with (7.17). The columns of the transformation matrix from *velocity* to *orbit* coordinates are the unit vectors of the *velocity* coordinate system in *orbit* coordinates [18, p.227].

$$\underline{ref}_v = A_{o \rightarrow v} \underline{ref}_o \quad (7.17)$$

where

$$A_{v \rightarrow o} = \begin{bmatrix} \hat{\underline{x}}_v & \hat{\underline{y}}_v & \hat{\underline{z}}_v \end{bmatrix} \quad (7.18)$$

and its inverse is

$$A_{o \rightarrow v} = A_{v \rightarrow o}^T \quad (7.19)$$

A positive rotation of the reference vector through α around z_v corrects for the aberration. The rotation is accomplished by

$$\underline{\alpha ref}_v = A_\alpha \underline{ref}_v$$

where

$$A_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.21)$$

The shifted vector is transformed back to orbit coordinates

$$\underline{\alpha ref}_o = A_{v \rightarrow o} \underline{\alpha ref}_v \quad (7.22)$$

which is the desired result.

Each of the reference vectors is corrected in this way to compensate for aberration. The position of the sun can either be sent to the star sensor from the ADCS or the correction performed by the ADCS.

7.6 Bright objects

Bright objects appearing close to the FOV may cause reflections which will make it impossible to detect the stars. The sun, moon and earth are bright objects which may have this effect. Even when they do not appear within the FOV, the light radiated or reflected by these objects onto the lens is powerful enough to overwhelm the light gathered from the stars. The light may be reflected directly into the FOV by atmospheric particles, or it may be collected by the lens and reflected internally onto the CCD.

One method of minimising the amount of stray light reaching the CCD, is to add a baffle around the lens. As shown in Figure 7-4, the baffle prevents light from objects outside the FOV from directly reaching the lens surface.

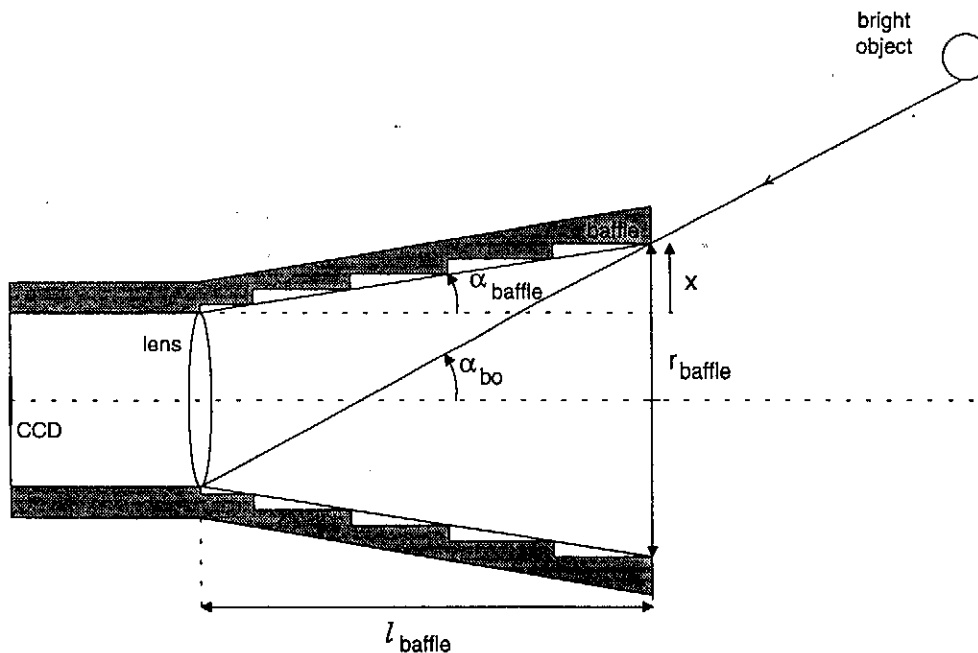


Figure 7-4 The baffle used to prevent direct light from bright objects reaching the lens

The baffle should be as narrow as possible to block out the maximum stray light, but it should never enter the FOV. The diagonal of the CCD effective image area is 6.1 mm and the focal length 25 mm, which gives a FOV diagonal of

$$\begin{aligned}\alpha_{FOV} &= 2 \operatorname{atan} \frac{3.05}{25} \\ &= 14^\circ\end{aligned}\tag{7.23}$$

The baffle length is 60 *mm*. This length should be sufficient to block stray light. A baffle which is too long becomes heavy and needs a very strong mount on the lens. The distance x shown in Figure 7-4 is

$$\begin{aligned}
 x &= l_{baffle} \tan \alpha_{baffle} \\
 &= l_{baffle} \tan \frac{\alpha_{FOV}}{2} \\
 &= 60 \tan 7^\circ \\
 &= 7.3 \text{ mm}
 \end{aligned} \tag{7.24}$$

To ensure that the baffle does not enter the FOV, its front radius should be at least

$$\begin{aligned}
 r_{baffle} &\geq r_{lens} + x \\
 &= 11 + 7.3 \\
 &= 18.3 \text{ mm}
 \end{aligned} \tag{7.25}$$

To allow for a CCD centre which does not coincide with the optical centre, and a possible misalignment of the baffle and optical axis, a value of $x = 10 \text{ mm}$ is used. This gives a baffle radius of $r_{baffle} = 21 \text{ mm}$. The minimum angle at which light from a bright object outside the FOV can reach the lens surface is α_{bo} .

$$\begin{aligned}
 \alpha_{bo} &= \text{atan} \frac{2r_{lens} + x}{l_{baffle}} \\
 &= \text{atan} \frac{32}{60} \\
 &= 28.1^\circ
 \end{aligned} \tag{7.26}$$

To prevent light from being reflected along the baffle, it has a black, stepped inner surface. This also simplifies the machining of the baffle, since it only needs to be drilled out instead of using a lathe to remove the inner material. The baffle will not prevent all light from outside the FOV from reaching the lens, but it will minimize the effect of bright objects outside the FOV.

7.7 Conclusion

Various external factors which may affect the operation of the star sensor have been examined. Software and hardware solutions have been presented for those causing significant errors.

Parallax and proper motion of close stars should not affect operation of the star sensor significantly. Most types of variable stars will not have a significant effect either. Those which may affect normal operation, are generally unpredictable and virtually impossible to correct. These stars will be discarded as false stars by the recognition software.

The effect of precession of the equinoxes is eliminated by using Epoch 1950 as reference for both the star catalogue and the orbit coordinate system. Aberration due to the finite speed of light is calculated and corrected as discussed in Section 7.5.

The effect of bright objects are limited by the use of a baffle with a black, stepped inner surface.

8. Hardware

The hardware for the star sensor consists of the star imager, a processor and the required interface and peripheral circuits. The selection of the optics and CCD components are discussed in Chapter 9, while this chapter concentrates on the electronics. Details of the design and operation of the star sensor are presented.

The star sensor was designed according to the following specifications:

- a microprocessor is required to control the imager and perform image processing as well as orientation calculations
- the image, star catalogue and software have to be stored in memory and the star matching algorithms require large amounts of processor memory
- the images taken by the CCD (Charge Coupled Device) circuit have to be digitized and stored
- the processor as well as the RWSS (Reaction Wheel Star Sensor) bus should have access to the image memory and be able to control the imager independently
- power consumption should be kept to a minimum
- cost should be a minimized
- angular accuracy of at least 1 mrad is desired

All circuit diagrams were drawn in *PCAD*¹ and appear in Appendix J. The complete circuit consists of three printed circuit boards:

- the *CCD PCB* which includes the CCD and video amplifier
- the *timing generation PCB* where all the video related clocking signals are generated
- the *processor PCB* for the microprocessor, memory and interfaces

Since the Sony data sheets do not provide all the necessary details, a prototype version of the hardware was manufactured. It included a large number of solder pads and test points. This was used to finalise the design and do the initial tests for star identification and calibration. Most of the software was developed and tested on the prototype version as well. The engineering and flight models are the next versions of the circuits.

A block diagram of the star sensor is shown in Figure 8-1. The components shown are discussed in detail in the following sections.

¹ A product of the Altium Corporation, 1994

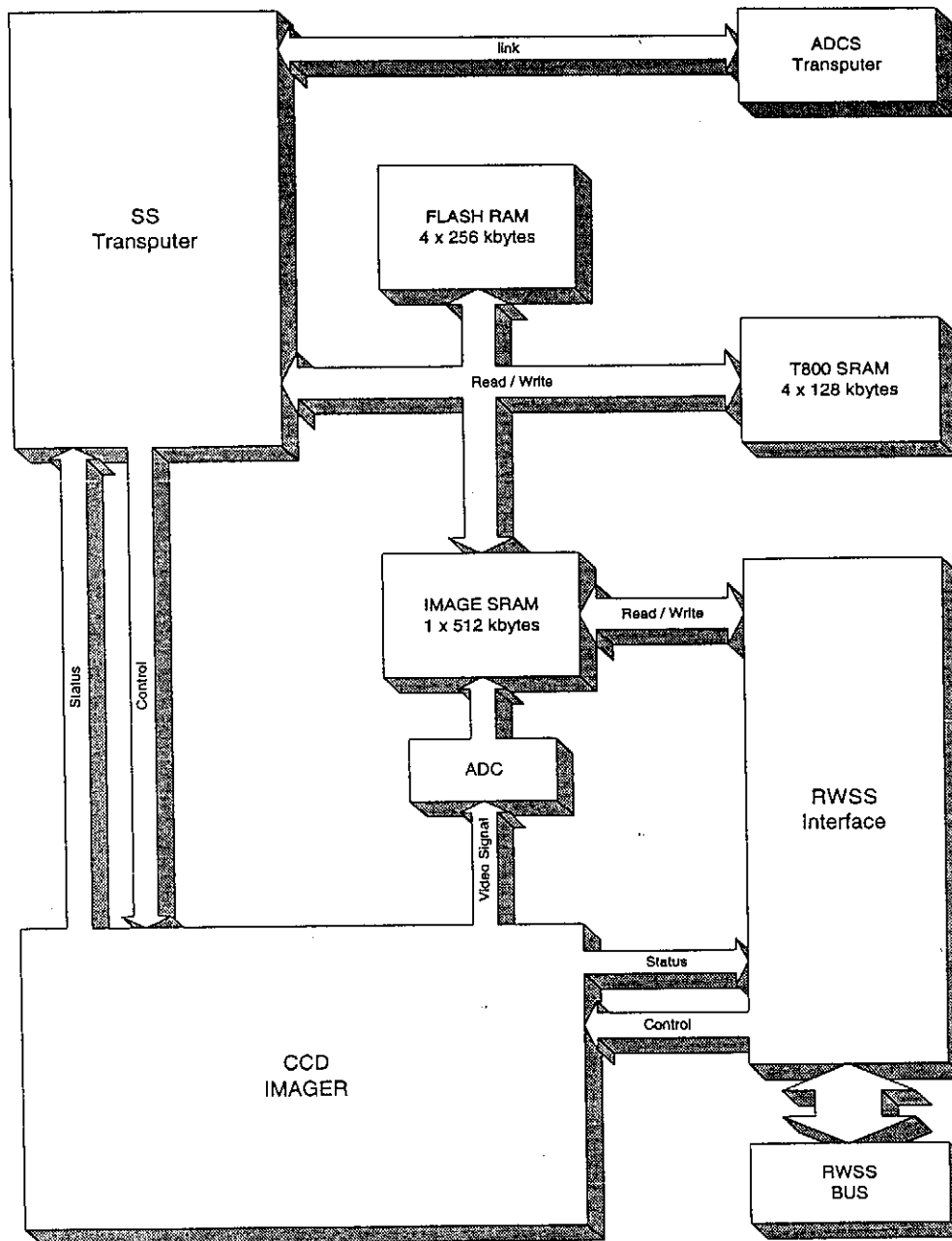


Figure 8-1 Block diagram of the hardware

8.1 The Microprocessor

The microprocessor controls the imager by setting the integration time and sending the command to start imaging. Once the image has been stored in memory, it performs the image processing and orientation calculations.

All these calculations have to be performed within one second, since the attitude determination and control system of the satellite requires an orientation update at one second intervals. Further bearing in mind that the star sensor is only a secondary

system which will be operating during the power intensive period when the earth imager is being used, it has to consume the minimum power.

These two constraints, high speed and low power consumption, narrowed the options down to two processor types: transputers and DSP's. Much thought was given to the selection and the final decision was to use a *T800* transputer.

In the DSP category, Motorola and Texas Instruments' devices were best suited for the application. Even though extremely fast, the TI DSP's consume too much power to be considered. Motorola manufactures a low power DSP operating from a 3.3V supply, the DSP56002. Unfortunately the low power devices do not have floating point units and have to simulate floating point operations. The calculations performed by the star sensor involve a great deal of matrix transformations and other floating point operations, making fast floating point abilities essential.

Since the *T414* transputer does not have a floating point unit either, only the *T800* was considered. This narrowed the choice down to the Inmos *T800* transputer or the Motorola *DSP56002*.

Tests were run to determine the speed at which the processors could perform basic floating point arithmetic. A large number of iterations were performed and the total operation time was divided by the number of iterations to obtain the average time for one iteration. Consequently the results include the time to perform the 'FOR' statement in the loop as well, but this was found to be much shorter than the time spent performing the arithmetic. The *T800* test was compiled in Rowley Modula and run on a *TR6* board while the DSP test was done in C by Alex Basios at Peralex. The results are listed in Table 8-1.

FUNCTION	TIME (μ s)	
	<i>DSP56002</i>	<i>T800</i>
sin	65	39
tan	103	88
asin	82.5	91
atan	77.5	34
log	90	64
sqrt	63.25	47
x/y	0.75	1.9
x+y	0.75	1.3
xy	0.75	1.6

Table 8-1 Comparison of calculation speed for the DSP56002 and the T800

From the table it is clear that the even though the DSP simulates its floating point operations, its calculation speed is similar to that of the transputer. The speed criteria could thus be eliminated from the selection process. The only advantage of the DSP is that its power consumption of 165 *mW* is well below the transputer's 500 *mW*.

The advantages of the transputer outweighed the few hundred milliwatts which the DSP would save. A number of people at the US have developed transputer systems and their expertise would be available in the event of any problems arising. This is a major advantage of the *T800* above the DSP, considering the short time span allowed for the development of the star sensor. Due to fact that the *T800* has been widely used at the US, the software packages for programming the transputer are available while those required for high level programming of the DSP would have to be purchased at great expense. This expense by far exceeds the relatively small saving realised when purchasing the DSP hardware. Since the DSP would be running off 3.3V the peripheral IC's, including memory and logic IC's, would either have to operate from 3.3V as well or all the control, data and address lines would be converted from 3.3 to 5V.

In summary, the *T800* is a well known, space proven microprocessor with high level programming software available and running off the same 5 V supply as its peripherals.

8.2 The RWSS (Reaction Wheel Star Sensor) Bus Interface

In the event of the star sensor and/or ADCS transputers failing, the star sensor will remain operational via the RWSS bus. The interface allows the star imager to be controlled and its memory accessed by other on board processors. On board processing of images may be possible on the 386, or alternatively the star images may be relayed to earth via the downlink.

Since the RWSS bus was previously designed with only 15 address lines, another four address lines are created by latching the data lines. These extra four address lines allow the RWSS bus to access the 512 *kbyte* image memory directly, as 16 blocks of 32 *kbytes* each. The other option was to clock the image out sequentially. This would simplify the hardware involved, but would restrict the speed of image processing by any processor connected to the memory. For example assuming that the coordinates of a star were roughly known and thus its position in memory as well, direct addressing of the memory makes a fast search possible. This would be slower with only sequential access.

Local star sensor buses are isolated from the RWSS bus when the *T800* controls the star sensor. Ultra fast and low power *QS3L384* bus switches are used.

8.3 Memory

The ability to store the software algorithms and the star catalogue necessitated some form of non volatile memory. Since the catalogue may need to be updated in flight, the use of EPROM and EEPROM were ruled out. Even though it would provide very fast access, battery backed up RAM could not be used due to the fact that nickel cadmium batteries tend to leak out in a vacuum. The main battery of the satellite could be connected to RAM directly but this would mean that all software would be loaded

up in flight since the battery will be disconnected prior to launch. A dedicated voltage regulator would be required to supply the RAM. A much simpler solution is the use of Flash RAM. Access times of 75 ns and low power requirements made Flash RAM the best option.

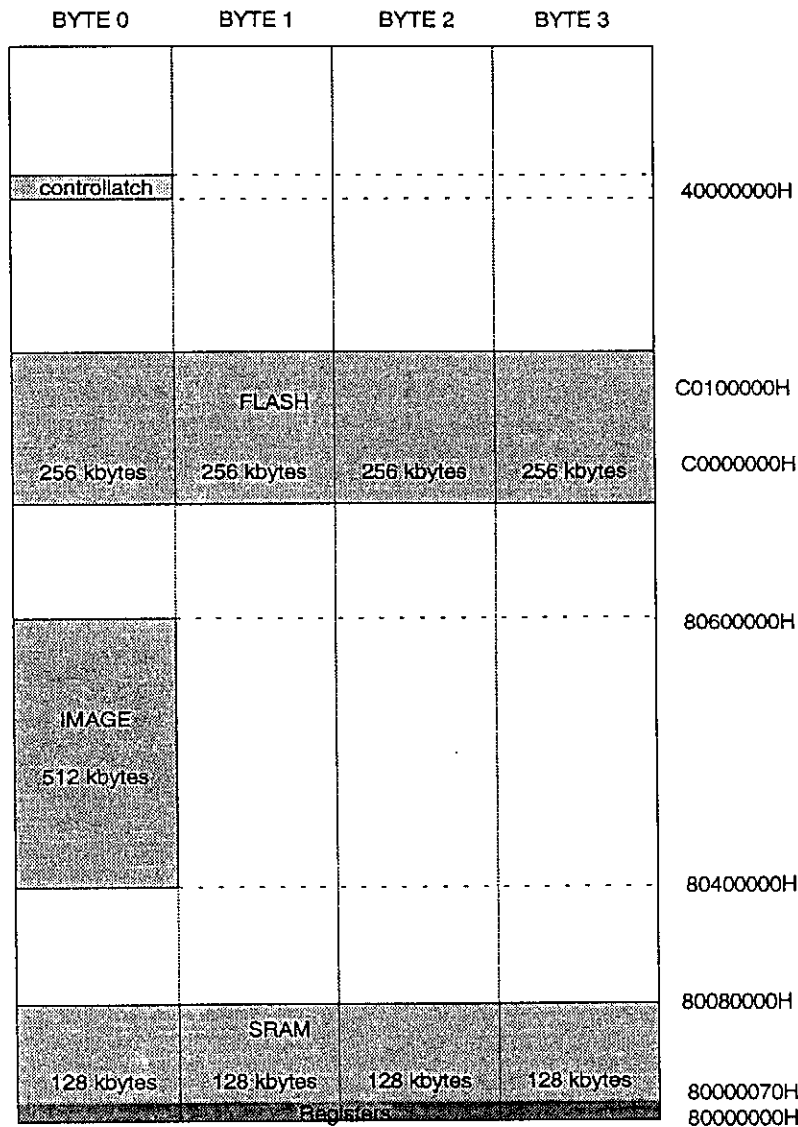


Figure 8-2 Memory map

Since the star matching algorithms require a lot of memory, a total of 512 kbytes of static RAM is connected to the transputer. A further 512 kbytes is used to store the image. A total of $817 \times 598 = 488566$ pixels clocked out of the CCD almost fills this memory.

A large quantity of *HM628128LFP-7* and *HM628512LFP-7* were to be purchased to test and build the RAM tray for the earth imager. These IC's are very fast, low power devices which were readily available and were consequently selected for the star sensor as well.

The memory map of the star sensor is shown in Figure 8-2. A *HC273* latch and *QS3L384* bus switch are both mapped to location 40 000000H. Writing to this address latches data which controls the CCD circuit. The sequence for imaging is described in Section 8.5.1 while the procedure used to set the integration time is discussed in Appendix G. Data regarding the status of the CCD circuit is read from the same address through the bus switch.

8.4 Power Supplies

The star sensor operates off six supply voltages:

- *ADCS+5V* to supply all logic IC's interfacing with the ADCS board
- *STAR+5V* for all remaining logic, memory and analogue circuitry
- *T800+5V* for the transputer
- *STAR+12V* for programming Flash RAM
- *STAR+15V* required by the CCD
- *STAR-8V* for the CCD

All voltages, except *STAR-8V*, are generated on the ADCS board. Since *STAR-8V* supplies a minimal current (approximately 7.5 mA) to the CCD circuitry, it is obtained from *STAR-15V* by means of a linear regulator. This means a dissipation of 50 mW, but is more power economical than using a switching regulator. The switching regulators tend to have high efficiency at high current, but this drops off dramatically at low current.

The separate five volt supplies ensure that the ADCS board does not drive dead logic when the star sensor is off. *ADCS+5V* is on as long the ADCS board is on, supplying all star sensor logic and bus switches connected to ADCS. *STAR+5V* is only on while the star sensor is on. The transputer supply, *T800+5V*, is turned off when the star sensor is operated via the RWSS bus. No power is wasted by keeping components on while they are not being used.

The switching regulators cause a 40 mV, 600 kHz ripple on the supply lines. This is unacceptable for the CCD circuit since the noise appears directly on the image in the form of vertical lines. To alleviate the problem, LC filters are incorporated in the CCD supplies. Various techniques of filtering were examined. Normal linear regulators would dissipate too much power even to be considered. A low dropout linear regulator's response does not extend to 600 kHz and consequently most of the ripple is simply let through. Capacitive filtering is not sufficient to remove the noise, but simple LC filters were found to be adequate.

8.5 CCD circuit

This section discusses the electronics for the Sony CCD and peripheral circuits. Chapter nine is devoted to the optical properties of the CCD.

The CCD circuit consists of two sections:

- the clocking circuit
- the CCD and signal processing circuit

Complete circuit diagrams appear in Appendix J.

8.5.1 Clocking circuit

The Sony CCD is supported by a complete set of hybrid IC's. The *CXD1261* signal generator generates all the necessary clock pulses for driving the CCD. The horizontal transfer (pixel readout) pulses require no buffering, but the vertical transfer (line readout) pulses are level shifted and buffered by the *CXD1250*.

The CCD circuit is controlled by the microprocessor (*T800* or RWSS bus) via a *HC273* latch. The signals *ED0*, *ED1*, *ED2* and *LADC* are set or reset by writing to the memory mapped latch. A *QS3L384* bus switch is used to read the status (*ADCaccess*, *RDOUT* and *ADC*) of the CCD circuit.

The CCD is clocked by the same pulses all the time, even though no image is being read out. The state diagram sequence for imaging, shown in Figure 8-3, is implemented with three D flip-flops. On power-up */RDOUTCMP* is taken low for a few milliseconds. This resets the star sensor to state *a*. It remains in *a* as long as *LADC* stays unchanged. *LADC* is the signal which latches *ADC* high. If an image is to be clocked into memory, *LADC* is taken low and high again with software. *ADC* is set which means that the ADC will have access to the image memory once the image is ready. The star sensor is now in state *b* and remains here until *XSG1* pulses low. *XSG1* is a Sony signal which indicates that the odd field is about to be clocked out. The period since *XSG1* goes high until the first pixel of the odd field is read out of the CCD is unspecified by Sony. In this time the horizontal shift register of the CCD is read out, but it contains no sensible data. To prevent this data from being placed in memory, it remains in state *c* until the first CCD line readout pulse *XV4* goes low and high. This places the star sensor in state *d* which means that the next pixel to be clocked from the horizontal shift register is the first pixel of the odd field. *RDOUT* is high and the pixels clocked from the CCD are converted by the ADC. The *HC590* counters increment the memory address for each consecutive pixel.

If *LADC* has been reset to 0 in the mean time, *RDOUT* will still be active but the transputer will have access to the image. The signal *ADCaccess* is the logical *AND* of *RDOUT* and *LADC*. When it is high, the ADC has access to the memory and the transputer or RWSS bus has access while it is low. If *RDOUT* is high but *LADC* has been reset low, the image will be converted and the counters incremented, but no values written into memory. This should only be necessary in unforeseen circumstances, but ensures that the transputer (or RWSS bus) *always* remains in control of access to the image memory. During normal operation, *LADC* will still be 1 and *ADCaccess* will be 1 as well. The ADC will have access to the image memory and all pixels are stored in memory.

The CCD is operated in the frame readout mode. Both the even and odd fields are integrated at the same time and are clocked out directly one after the other. Once the memory counters have counted to 512k, both fields have been read out. $\overline{\text{RDOUTCOMP}}$ is set low by the 21st address line going high. The star sensor is reset to state *a*.

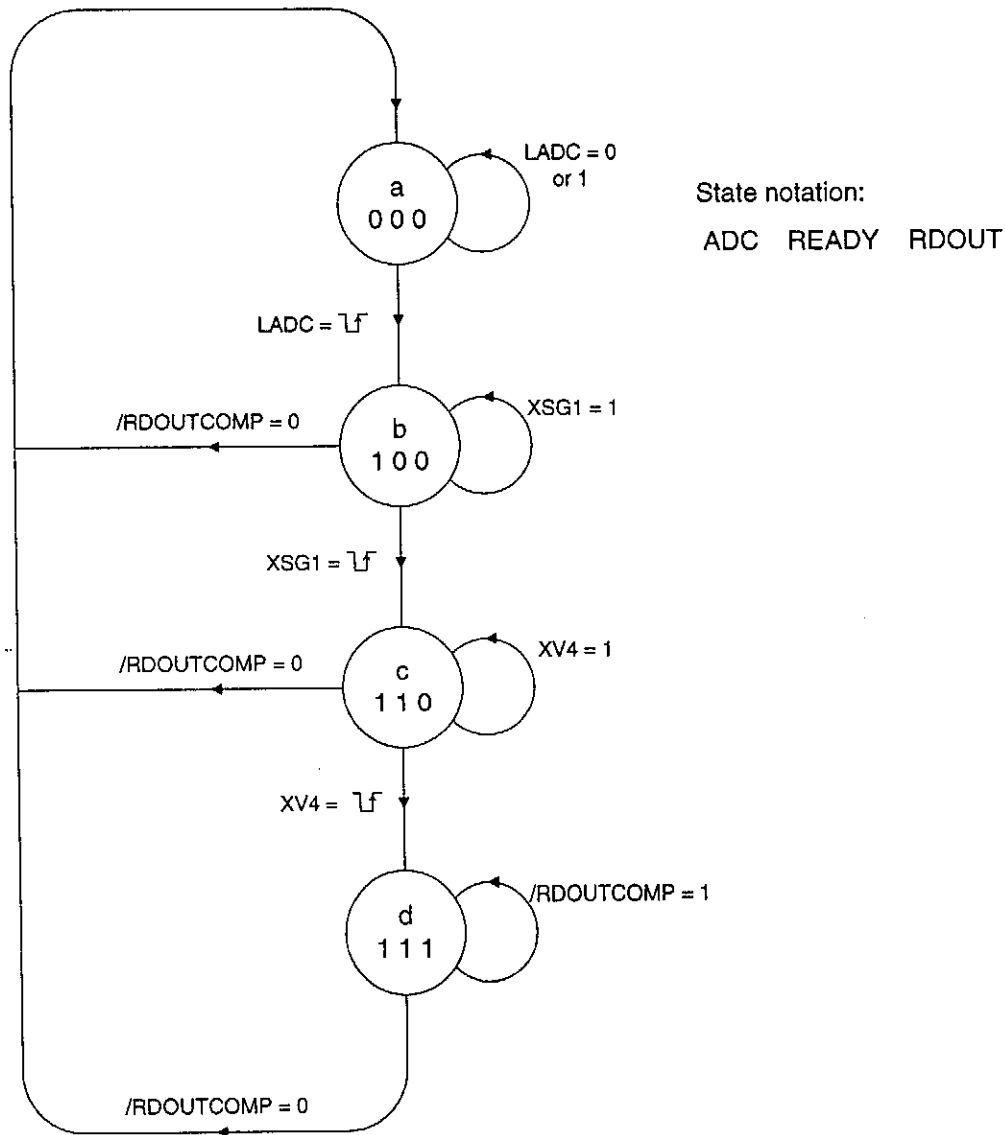


Figure 8-3 State diagram for storing an image in memory

Reading out both fields provides double the resolution in the vertical direction. The resolution in the horizontal and vertical directions are then almost identical. The pixels are almost square measuring $6.5\ \mu\text{m}$ horizontally by $6.25\ \mu\text{m}$ vertically.

8.5.2 The CCD and signal processing circuit

This circuit is mounted directly behind the CCD. As suggested in the Sony data sheets, the CCD output is fed into a source follower FET and again buffered by a

bipolar emitter follower which drives the large capacitive input to the sample and hold. The signal path from the CCD to image memory is illustrated in Figure 8-4.

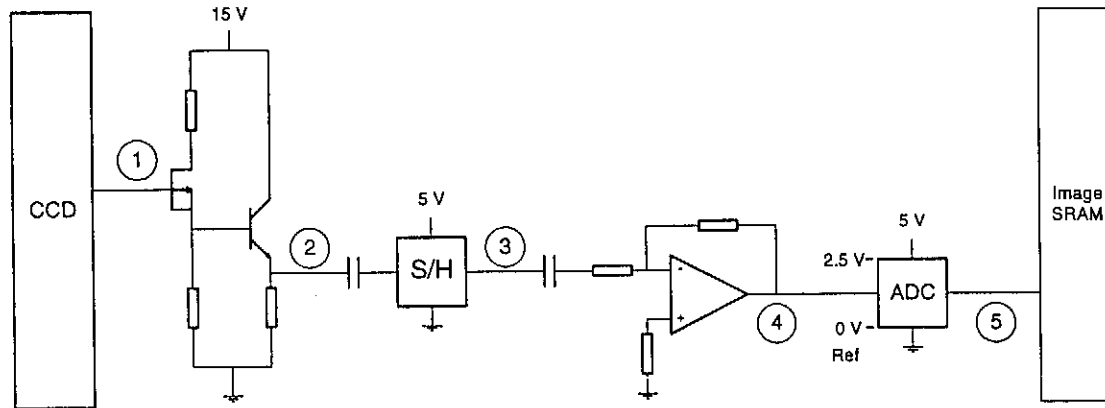


Figure 8-4 Signal path from the CCD to image memory

The typical waveform of the CCD output is shown as signal number 1 in Figure 8-5. Within the CCD the gate reset pulse (*RG* in Figure 8-5 and the circuit diagram in the appendix) is coupled through to appear on the video output, as shown in the diagram. The downward going section represents the pixel output. This value is directly proportional to the light² incident on the photo element². A longer integration time or brighter light incident on the photosensor increases the amplitude of this deviation. It is a negative going pulse and saturates at approximately 500 mV.

The coupling capacitors remove the dc offset introduced by the CCD and sample and hold. Only the ac signal is passed through the capacitors to be amplified and digitized. Large capacitors ensure that the dc offset does not shift from its constant (black) value when brighter pixels are clocked out. The largest capacitors allowed on Sunsat (10 μ F tantalums) are used as the coupling capacitors. The input resistance of the sample and hold is 10k and that of the amplifier 820 Ω . Increasing the input resistance of the operational amplifier increases the offset voltage. It can be corrected with external circuitry, but it is preferred to keep the input resistance to a minimum to prevent large offset variations with temperature. The shortest time constant is at point 3 in Figure 8-4, where

$$\begin{aligned}
 \tau &= RC \\
 &= 820 \times 10\mu F \\
 &= 8.2\text{ ms} \\
 &\approx 57800 T_{\text{pixel}}
 \end{aligned}
 \tag{8.1}$$

where

T_{pixel} = the time taken to clock one pixel out, about 140 ns

² See equation (9.9)

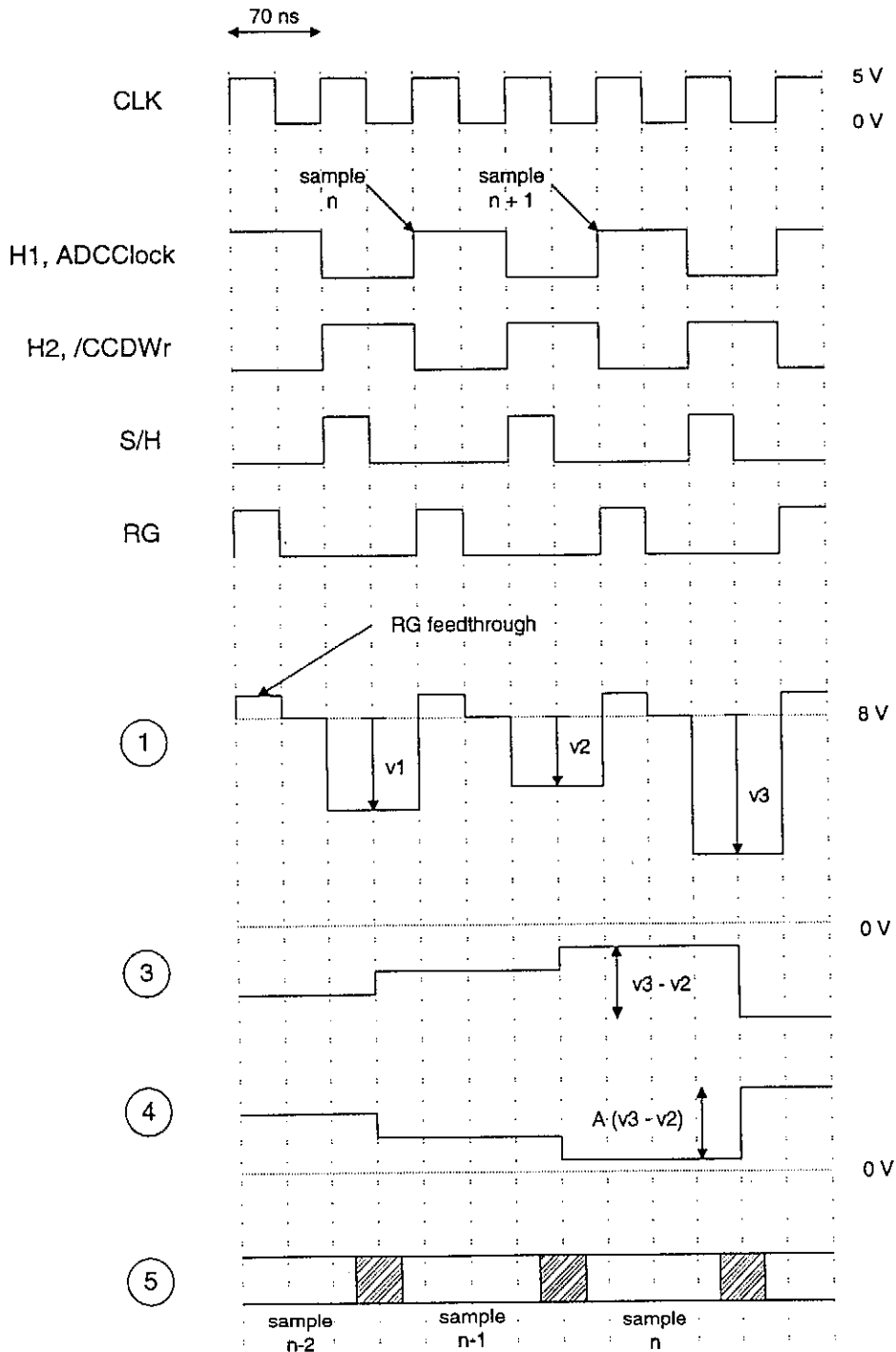


Figure 8-5 Signal waveforms

Twelve dark rows are clocked out of the CCD before the effective image. This charges the capacitor at point 3 to some dc value representing the dark voltage. The time constant τ should be long enough to ensure that this dc voltage does not change significantly when pixels are clocked out. The *Region Growing* module of Chapter 2 uses the 43 dark pixels of each row to determine the dark reference voltage for the row. The brightness of each pixel in the row is the difference between the pixel voltage and the threshold. It is thus unacceptable that the dc voltage changes significantly while a row is being clocked out, but it may change slightly from row to

row. When imaging stars, most pixels are no brighter than the dark average of the row. A few bright pixels may appear which slowly discharge the capacitor. They are followed by dark pixels which again slowly charge the capacitor.

The change in voltage on the capacitor when n pixels of voltage V_{pixel} are clocked out is

$$\Delta V_c = V_{pixel} \left(1 - e^{-\frac{nT_{pixel}}{\tau}} \right) \quad (8.2)$$

Equation (8.2) is solved for the number of pixels n which will bring about a significant change in voltage. A significant change is considered to be a change of one quantization level. Each of the n pixels clocked out are assumed to have the maximum voltage which can be converted by the ADC.

$$\begin{aligned} n &= -\frac{\tau}{T_{pixel}} \ln \left(1 - \frac{\Delta V_c}{V_{pixel}} \right) \\ &= -57800 \ln \left(1 - \frac{1}{64} \right) \\ &= 909 \text{ pixels} \end{aligned} \quad (8.3)$$

In other words, 909 consecutive pixels with the maximum digitizable voltage can be clocked out before the dc voltage changes by one quantization level. This is more than the number of effective pixels in a row (752). Usually a row of pixels belonging to a star will contain no more than 20 or 30 pixels. Even if a few stars appear in a row, the change will not be significant. The small drop in the dc voltage will be recovered when the dark pixels between the stars are read out. The voltage of an object appearing in the image may exceed the maximum digitizable voltage. The CCD output saturates at 500 mV and one quantization level (before amplification) is about 1.5 mV. Replacing these values in equation (8.3), it is found that 174 consecutive saturated pixels can be clocked out before the dc voltage changes by 1.5 mV. Which means that the magnitude of any star appearing after this exceptionally bright object may be calculated incorrectly. Due to the smoothening effect of the optical transfer function³, an object which saturates 174 consecutive pixels will cover the entire image and no stars will be visible anyway. During normal operation of the star sensor, the time constant is long enough to ensure reliable magnitude estimates.

The sample and hold signal S/H is generated by logically *ANDing* the horizontal driver $H2$ and the input clock to the system. This produces a pulse which samples the video output during the period of maximum deviation. At the video frequency used, S/H is delayed significantly by the *AND* gate. Due to the delay, the video signal is sampled during the centre of the pixel output, and not in the first section as shown in Figure 8-5.

³ Refer to Chapter 9

The sample and hold output (signal 3 in Figure 8-5) is ac coupled into the amplifier stage. A wide bandwidth video operational amplifier is used to amplify and change the polarity of the sampled output before driving the analogue to digital converter. The amplification provided by this stage is indicated as A for signal 4 in Figure 8-5. Offset is added from a reference diode to ensure that the output remains above zero.

Various options were considered for this signal processing stage, but the abovementioned configuration of a sample and hold and operational amplifier was the simplest circuit yielding the best results.

One option was to use the Sony *CXA1310* signal processing IC. This includes a sample and hold and buffer of which the output can not exceed 0.5 V peak to peak. Therefore another amplifier is needed before the signal is input to the ADC, leading to a more complex circuit than the abovementioned one.

A common emitter amplifier was tested as well. The video output was amplified directly, together with the feedthrough of the reset gate pulse. This scheme worked well, except for the fact that the dc offset of the amplified signal varied with up to 20 quantization levels from one image to the next. The fluctuation was mainly due to the varying feedthrough of the gate reset pulse on the video signal. Obviously the dc level input to the ADC can not be allowed to fluctuate that much. Using clamping circuits to dc-restore the signal made this circuit very complex. Power supply noise was amplified by the common emitter amplifier as well, which ruled out the use of this circuit.

Problems were experienced with noise in all the above circuits, the greatest problem being random electronic noise generated in the CCD. Using the sample and hold and op amp circuit, a constant 5 V dc reference was sampled with less than 200 μV of noise. The measurements were made with all clocks in the circuit running, but the CCD output was disconnected from the source follower (at point 1 in Figure 8-4) and the 5 V reference connected instead. Once the CCD was connected and a dark image taken, the peak to peak noise levels increased to 3 mV. This lead to assumption that the noise was coming from the CCD. Noise generated in the CCD is of two types:

- fixed pattern noise caused by the dark current non uniformity
- electronic noise

Fixed pattern noise remains identical in consecutive images and increases linearly in magnitude with increasing integration time, while electronic noise is random. The noise measured was not fixed pattern but electronic noise. The fixed pattern noise became clearly visible with integration times exceeding one second, while the 3 mV random noise was measured with integration times as short as 20 ms. This lead to the conclusion that the noise is electronic noise introduced by the clocking signals in the circuit.

8.6 The Analogue to Digital Converter Circuit

The CA3306 satisfies all the specifications set for the star sensor ADC. Conversion of the 7 MHz pixel stream is well within its range. It has a tristate output which makes the bus connection simple. A quoted maximum power consumption of 50 mW probably makes it the lowest power Flash ADC available at the moment.

Conversion rate of about 7 MHz was selected as a compromise between power consumption and the speed at which the image is read out. The maximum pixel readout frequency of 14 MHz means that the entire image is read out in approximately 40 ms, while 7 MHz gives a readout time of 80 ms. A 14 MHz readout would place unnecessarily stringent demands on the memory and consume more power. Reducing the clock speed much below 7 MHz causes the readout time to become excessive. Bearing in mind that an update on the orientation of the satellite is required once a second, the integration time can not exceed 1 s minus the readout time. A longer integration time is desirable since the magnitude of visible stars is increased.

The horizontal register clocking pulse *H1* is buffered and used as the ADC clock, *ADCClock*. The ADC latches the sample internally at the rising edge of *ADCClock*. The samples appear as shown in signal 5 of Figure 8-4.

With enough amplification provided in the previous stage, a six bit ADC provides sufficient accuracy to observe faint stars just visible above noise. With too large an amplification, all bright stars would totally saturate the ADC. They would still be visible on the image, but it would be impossible to accurately estimate their magnitudes or calculate their centroids. Using an amplification of 25 and input range ($V_{REF+} - V_{REF-}$) of 2.5 V digitized to 64 levels, means that each quantization level represents 1.5 mV on the CCD output. Stars of magnitude 3 and brighter saturate while stars of magnitude 6 are still visible above the noise. Even though magnitude 3 stars start saturating, this is only to a slight degree and magnitudes and centroids are still estimated accurately. This does not pose too much of a problem since a very small percentage of stars are brighter than magnitude 3. In most cases enough fainter stars are available to allow the brighter ones to be discarded.

If the ADC is allowed access to the memory (i.e. *ADCaccess* is high), the converted pixels are stored in image memory. Three HC590 8-bit counters address the image memory. They are reset by *RDOUT*. As long as it remains low, the counters are kept in the reset state. When *RDOUT* is high, indicating that the pixels are being read out, the same pulse clocking the ADC (*ADCClock*) is used to increment the counters. This ensures that the address is stable well before the pixel value is written to memory. The second horizontal shift register clock *H2* is buffered before being used as the image memory write pulse, */CCDWr*. Data is latched into memory on the rising edge of */CCDWr*. This ensures the maximum memory write setup time for the converted pixel, as shown in Figure 8-5.

8.7 Results

The first set of results prove that the star sensor can actually detect magnitude 6 stars, while the second set examines the power consumption.

The image shown in Figure 8-6 was taken within 75° of the full moon, in Stellenbosch. Reflections of city light and the full moon made the night sky appear very bright. Even in these unfavourable conditions all magnitude 6 stars were clearly visible. Due to the bright sky, the threshold of detection was raised above the dark average by 2 quantization levels. The integration time was 0.984 s, which is practically the same as the 1 s period which will be used on the flight model.

A minimum star size of 10 pixels was allowed. When the minimum size is reduced to 7 pixels, 11 more stars appear in the image. To reduce the probability of mistaking noise as stars, the minimum size of 10 pixels is used instead.

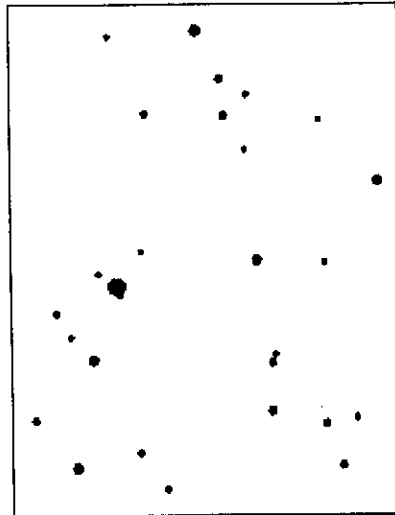


Figure 8-6 Image centred at RA = 97° , Dec = -55°

Figure 8-6 is not the actual image which was taken. It displays the centroids detected by the *Region Growing* algorithm. The sizes of the dots vary according to the measured magnitude. Where two stars appear merged, they were extracted as separate stars, but due to the representation used, they seem to overlap.

The corresponding section of the star catalogue appears in Figure 8-7. It contains all stars down to magnitude 6. By comparing Figure 8-6 and 8-7, it is clear that all magnitude 6 stars which should appear in the image, do indeed. More dots appear in the image than the catalogue. To prove that these dots are stars and not noise, the limiting magnitude $M_v = 7$ is used in Figure 8-8. All remaining dots in the image can be traced in Figure 8-8.

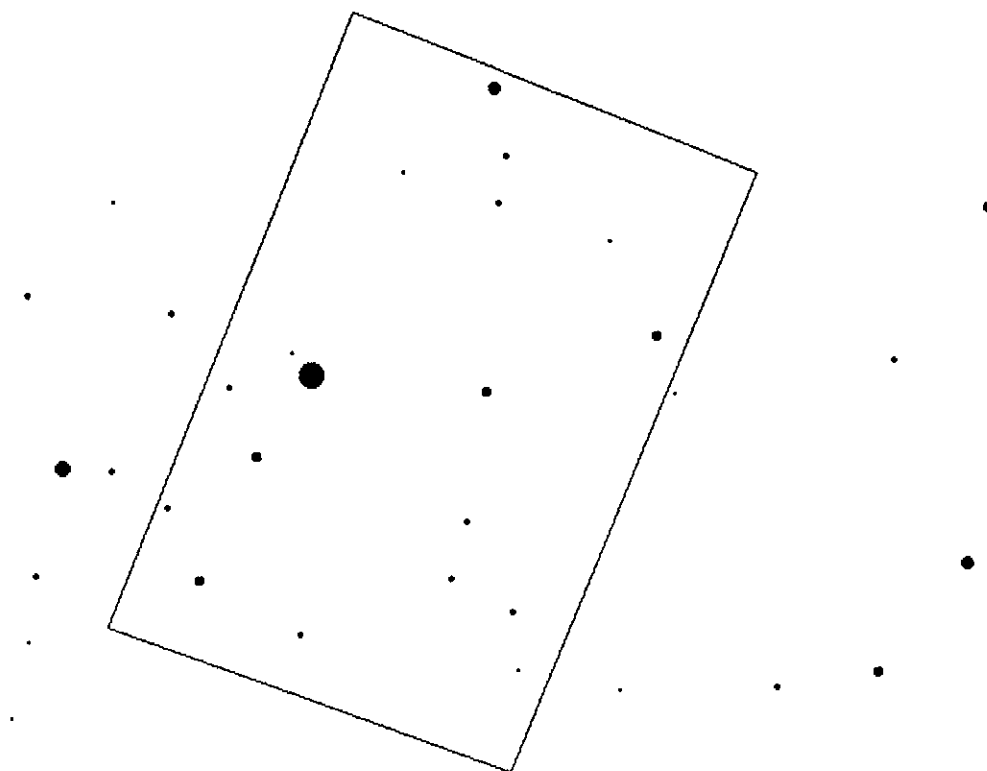


Figure 8-7 Section of the catalogue corresponding to the image of Figure 8-6, limiting magnitude is $M_v = 6$

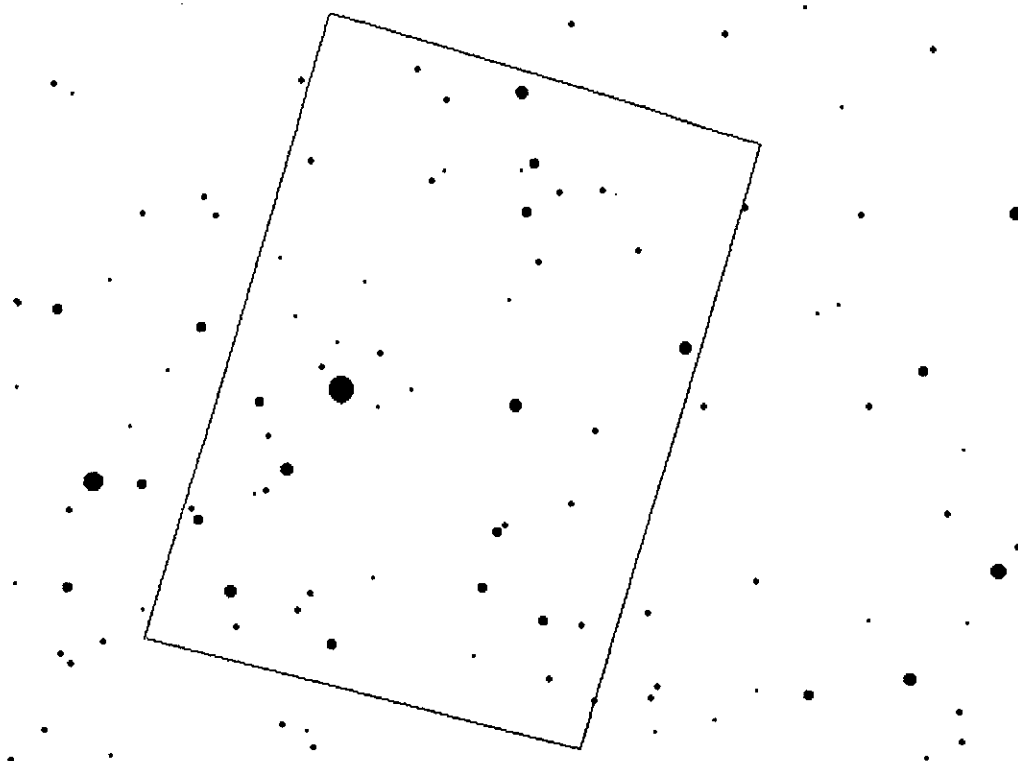


Figure 8-8 Section of the catalogue corresponding to the image of Figure 8-6, limiting magnitude is $M_v = 7$

Imaging conditions at an altitude of 800 km should be much more favourable than these worst case conditions. Reflections from the atmosphere should be minimal, providing a very dark night sky. The difference between the detection threshold and dark average should be 0 or maybe 1 quantization level. Under these circumstances, much fainter stars should be visible.

The currents drawn from and power delivered by the three supplies are listed in Table 8-2.

Supply (V)	Current (mA)		Power (mW)	
	Average	Maximum	Average	Maximum
-15	-10	-10	150	150
5	188	270	940	1350
15	11.8	11.8	177	177
total			1267	1677

Table 8-2 Power delivered to the star sensor

The average power consumed by the entire star sensor was measured as 1.3 W and the peak consumption as 1.7 W. The peak value and average value drawn by the Sony components are equal, since all the clock pulses run continuously. Considering the fact that the T800 consumes about 500 mW, this is a reasonable consumption.

8.8 Conclusion

With a relatively low power consumption, the star sensor manages to detect magnitude 6 and fainter stars. The tests were performed in unfavourable conditions, which prove that at least magnitude 6 stars should be visible from the orbit altitude of 800 km.

The next Chapter discusses the selection of the lens and the CCD components.

9. Selection of CCD's and Optics

The first version of the Sunsat star sensor used a *TC211* array CCD. This sensor was found to be inadequate for detecting stars of visual magnitude 6 and a more sensitive CCD had to be selected for the next iteration of the star sensor.

The fundamental specifications set for the CCD and optics were:

1. With a 10° field of view, stars of visual magnitude 6 have to be detectable. This requires a CCD which is very sensitive and can operate at very low light intensities.
2. The abovementioned criteria should be met without exceeding an integration time of one second, since the orientation information has to be updated at a rate of 1 *Hz*.
3. Star positions should be determined to an accuracy of at least 1 *mrاد*.

Three manufacturers offered devices which seemed suitable and justified further investigation. These were EEV, Sony and Thomson. Each manufacturer offers a wide range of array CCD's and hybrid support IC's. A specific set of components was selected from each manufacturer and evaluated in detail. The advantages and disadvantages of each manufacturer's components was weighed up after which a selection was made. The decisive factor was eventually the major price difference between the Sony components and the EEV and Thomson components. A Sony set could be purchased for about R700 while an EEV or Thomson set would be in excess of R7000. The detectability of magnitude 6 stars and optics required for Sony CCD's is examined in this chapter.

A brief introduction to the types of CCD's and the basic principles of CCD operation are presented in Appendix H.

9.1 Selection of optics

The diagonal of the CCD effective imaging area is 6.1 *mm*, referred to as the field stop. A FOV diagonal of approximately 14° is required. The focal length required is determined from the relation between FOV diagonal and field stop diagonal.

$$\begin{aligned}
 f &= \frac{fs_{diag}}{2 \tan\left(\frac{\beta_{FOV}}{2}\right)} \\
 &= \frac{6.1}{2 \tan(7^\circ)} \\
 &= 24.84 \text{ mm}
 \end{aligned} \tag{9.1}$$

where

$f_{S_{diag}}$ = field stop diagonal, 6.1 mm
 β_{FOV} = field of view diagonal, 14°

The closest commercially available lens has a 25 mm focal length.

The only remaining parameter is the aperture or F/number (F_N) of the lens. The F/number [9, p. 211] of the lens is defined as the ratio of focal length to effective lens diameter. It gives an indication of the lens's ability to collect light. A lens with a small F/number has a greater ability to collect light and is better suited for low light level imaging.

$$F_N = \frac{f}{D} \quad (9.2)$$

where

D = entrance pupil diameter of the lens

The amount of light reaching the image plane (CCD) is inversely proportional to the square of the F/number. A short derivation will prove this point. The illumination from a star reaching the lens surface is denoted by E_{star} . It is calculated from the magnitude of the star with (C.4). Illumination or luminous flux density is measured in *lux* or *lumen/m²*. Luminous flux F is measured in *lumen* and is equivalent to the light energy incident on the surface per second. The flux incident on the lens surface is collected by the lens and focused on the CCD plane. The luminous flux collected by the lens is the product of the flux density at the lens surface and the lens area. This flux is concentrated on the image plane (CCD). The total flux reaching the CCD plane is a function of the lens transmission factor B , which is close to unity:

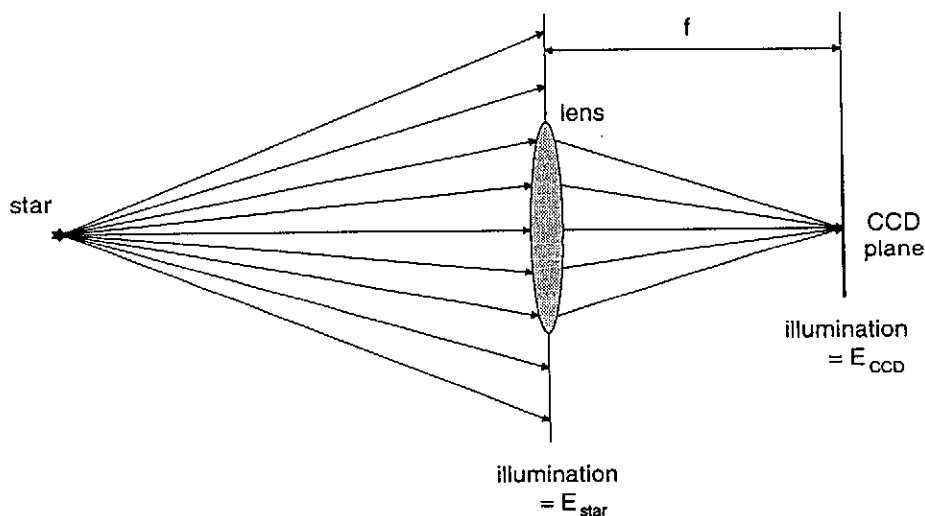


Figure 9-1 Path of light from the star to the CCD plane

$$F_{CCD} = BE_{star} A_{lens} \quad (9.3)$$

Due to the smoothing effect of the optical transfer function, the illumination (flux density) will have a Gaussian distribution on the CCD plane. This is examined in detail in the next section. For the purpose of illustration, the illumination is assumed to have a uniform distribution, covering an area $A_{star,CCD}$. The illumination on the CCD plane is then

$$E_{CCD} = BE_{star} \frac{A_{lens}}{A_{star,CCD}} \quad (9.4)$$

The area of the lens is related to the F/number by

$$\begin{aligned} A_{lens} &= \pi \left(\frac{D}{2} \right)^2 \\ &= \frac{\pi}{4} \left(\frac{f}{F_N} \right)^2 \end{aligned} \quad (9.5)$$

Replacing this in (9.4) gives

$$E_{CCD} = \frac{\pi BE_{star} f^2}{4 A_{star,CCD}} \frac{1}{F_N^2} \quad (9.6)$$

Equation (9.6) shows that the illumination on the CCD plane due to a star is inversely proportional to the square of the lens F/number. The only other variable in (9.6) is the area covered by the star on the CCD plane. It is determined by the focus and is independent of the F/number i.e. a well focused star will cover a small area while a defocused star will cover a larger area.

The voltage generated by the CCD is directly proportional to the illumination of its pixels. The smallest possible F/number is thus desirable. Unfortunately price and size of a lens increases together with the aperture. The custom made lens used by the Oersted star sensor has an F/number of 0.6 and costs close to US \$7000. The best commercially available lens is the Cosmocar/Pentax B2514D: a 25 mm, F/number 1.4 lens which costs about R740. This means that the Sunsat star sensor CCD will only receive $(0.6/1.4)^2 = 0.18$ times the amount of light the Oersted CCD receives from the same star, but at a fraction of the price.

9.2 Sony CCD's

This section describes the calculations used for evaluation purposes of the Sony CCD. These were performed during the component selection phase of the design and are based on the manufacturer data sheets.

Two factors will determine the ability of the star sensor to detect faint stars:

- electronic noise generated due to clock pulses, power supply ripple, etc.
- the dark signal non uniformity (DSNU) mentioned in Appendix H

No data existed on the CCD's ability to reject electronic noise and this could only be determined by building the hardware. The data sheets do, however, contain DSNU measurements and test conditions. This information was used to estimate the CCD's ability to detect magnitude 6 stars. The following two sections will determine the photogenerated voltage due to a magnitude 6 star and the dark signal non uniformity voltage. If the photogenerated voltage exceeds the DSNU voltage by a significant amount, it means that the star sensor should be able to detect a magnitude 6 star.

9.2.1 The dark signal non uniformity voltage

The dark current is a thermally generated current. As shown in Appendix H, it varies with temperature according to the diode law. The dark signal itself is not a problem when imaging. The difference between the actual pixel value and that of the average of the dark reference pixels¹ is used as the brightness of the pixel. The non uniformity in the dark signal is, however, a problem. The peak DSNU for the Sony ICX059 is given as 1 mV at 60°C with an integration time of 20 ms. Both the dark signal voltage and dark signal non uniformity voltage increase linearly with the integration time until saturation is reached. The DSNU with an integration time T_{int} is

$$\begin{aligned} V_{DSNU}(60^\circ C) &= 1 \times \frac{T_{int}}{0.02} \\ &= 50 T_{int} \text{ mV} \end{aligned} \quad (9.7)$$

During operation, the star sensor will always be on the cold side of the satellite (facing away from the sun). The temperature should usually be below 0°C. Equation (H.2) is used to calculate the DSNU at 5°C

$$\begin{aligned} V_{DSNU}(5^\circ C) &= 50 T_{int} e^{6374 \left(\frac{1}{333} - \frac{1}{278} \right)} \\ &= 1.133 T_{int} \text{ mV} \end{aligned} \quad (9.8)$$

For an integration time of 1 s, the DSNU voltage is 1.133 mV.

9.2.2 The photogenerated voltage

The voltage generated by each pixel due to incident photons is a function of the CCD responsivity, the incident illumination and the integration time. The responsivity R is an indication of the CCD's ability to convert luminous flux to voltage and is measured in units of Volts per incident lux per second. The responsivity of the Sony ICX059 is derived in Appendix H from the test data quoted in the data sheets. Using an integration time of 1 s, the photogenerated voltage is

¹ See Figure 1-1

$$\begin{aligned}
 V_{phot} &= RT_{int} E_{CCD} \\
 &= 1.385 E_{CCD}
 \end{aligned}
 \tag{9.9}$$

From (9.8) and (9.9) it is clear that increasing the integration time will not raise the photogenerated voltage above the DSNU. It will raise the level of the photogenerated voltage above the randomly generated electronic noise, which is not a function of integration time. For the purpose of these calculations, the maximum allowable integration time of 1 s is used in both (9.8) and (9.9).

As mentioned in the previous section, the illumination on CCD plane has a Gaussian distribution [7, p. 1067 equation (1)]:

$$E_{uv} = E_{max} e^{-\frac{(u-u_0)^2 + (v-v_0)^2}{2\sigma^2}} \tag{9.10}$$

where

E_{max} is the peak illumination at the centroid, (u_0, v_0)

To simplify calculations, the star centroid is aligned with the optical centre. The Gaussian distribution of the illumination is shown in Figure 9-2.

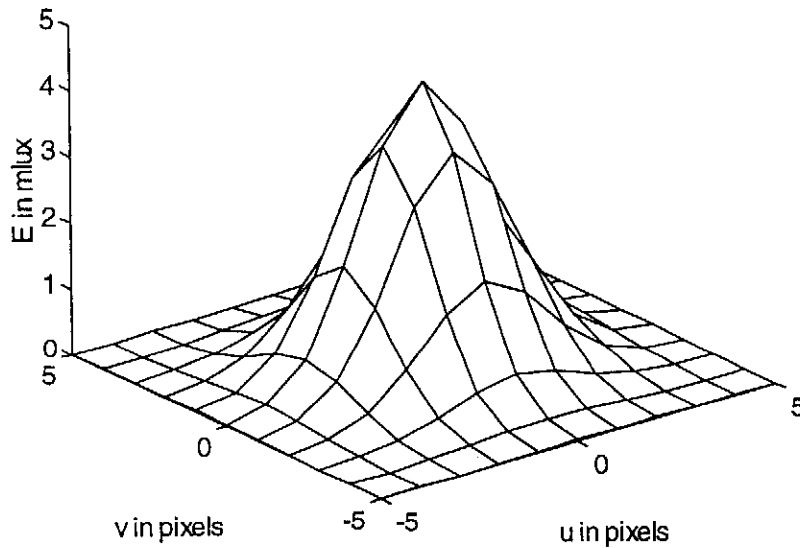


Figure 9-2 Expected Gaussian distribution of a star on the CCD plane. The pixel centres are at the intersections of the lines. The standard deviation is 1.5 pixels.

The total incident flux of a magnitude 6 star is known from (9.3).

$$\begin{aligned}
 F_{CCD} &= 1.05 \times 10^{-8} \frac{\pi}{4} \left(\frac{0.025}{1.4} \right)^2 \\
 &= 2.63 \times 10^{-12} \text{ lum}
 \end{aligned} \tag{9.11}$$

where

$$\begin{aligned}
 E_{\text{star}} &= 1.05 \times 10^{-8} \text{ lux [9, p. 66 Table 6.5]} \\
 B &= 1 \text{ as for an ideal lens} \\
 F_N &= 1.4 \text{ for the Cosmimar lens} \\
 f &= 25 \text{ mm}
 \end{aligned}$$

It is also known to be the surface integral of (9.10)

$$F_{CCD} = E_{\text{max}} \int_{-\infty-\infty}^{\infty \infty} \int e^{-\frac{u^2+v^2}{2\sigma^2}} du dv \tag{9.12}$$

The standard deviation of the Gaussian is determined by the focus of the image. A perfectly focused dot will have a small σ while a defocused image will have a larger value. Assuming a value for σ , the peak amplitude of the Gaussian may be calculated by equating (9.11) and (9.12). No closed form solution exists for (9.12) and a numerical method is used. As a first order approximation, the illumination of each pixel is set equal to the illumination at the pixel centre and these are summed. The result is solved for the peak amplitude of the Gaussian, E_{max}

$$\begin{aligned}
 E_{\text{max}} &= \frac{F_{CCD}}{h_{\text{unit}} v_{\text{unit}} \sum e^{-\frac{u_{\text{pixel}}^2 + v_{\text{pixel}}^2}{2\sigma^2}}} \\
 &= \frac{0.0647}{\sum e^{-\frac{u_{\text{pixel}}^2 + v_{\text{pixel}}^2}{2\sigma^2}}}
 \end{aligned} \tag{9.13}$$

Figure 9-3 shows this relation between peak amplitude and standard deviation.

A star with a minimum of 9 pixels will have a diameter of three pixels. The furthest any pixel centre could be located from the centroid is 1.5 pixels in each direction (if the centroid is located close to the edge of a centre pixel). This edge pixel is the worst case pixel, with the very minimum amount of illumination. Measuring in units of pixels, the illumination of this pixel is given by (9.10)

$$\begin{aligned}
 E_{\text{edge}} &= E_{\text{max}} e^{-\frac{1.5^2 + 1.5^2}{2\sigma^2}} \\
 &= E_{\text{max}} e^{-\frac{2.25}{\sigma^2}}
 \end{aligned} \tag{9.14}$$

The photogenerated voltage of the edge pixel is given by (9.9)

$$V_{photo,edge} = 1.385E_{edge} \quad (9.15)$$

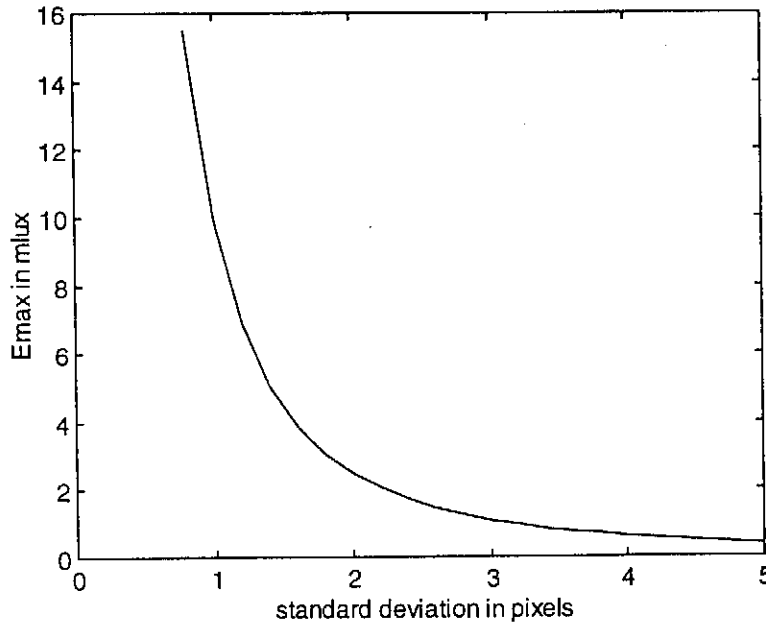


Figure 9-3 Relation between peak amplitude and standard deviation of the Gaussian, for a magnitude 6 star

To ensure reliable detection, the amplitude of this edge voltage should be at least 3 dB above the DSNU peak amplitude.

$$\frac{1.385 E_{max} e^{\frac{2.25}{\sigma^2}}}{\sqrt{2}} \geq 1.133 mV \quad (9.16)$$

E_{max} is a function of σ and can only be determined numerically. The values of σ which satisfy (9.16) are determined graphically. The quantity on the left of (9.16) is plotted in Figure 9-4 together with the threshold.

From Figure 9-4 it is concluded that when the standard deviation is too small the star is too well focused and the illumination on the edge pixel is not sufficient. Only the centre pixel(s) are detectable. For a standard deviation varying between about one and 2.5 pixels the edge pixel is detectable. This means that the star is well spread and sufficient illumination is still present on all the pixels up to the edge one. When the star is defocused even further, the illumination of the edge pixel (and the centre ones) drops below the threshold as the luminous flux is spread over too a large area.

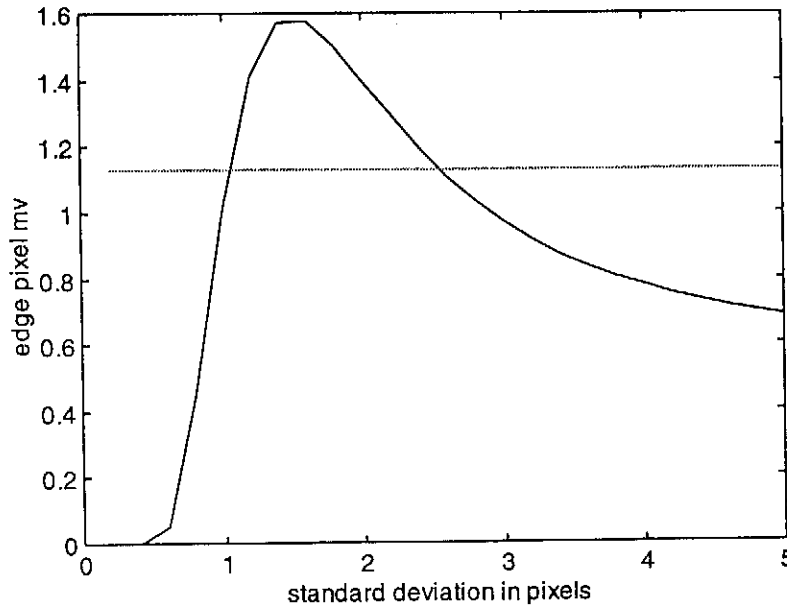


Figure 9-4 Determining σ for which the edge pixel detectable. The threshold of detectability is shown as the horizontal line at 1.13 mV

Figure 9-4 shows the level which is 3 dB below the voltage generated by the edge pixel of a magnitude 6 star, compared to the maximum amplitude of the DSNU at 5°C. This is the very worst case since the DSNU amplitude of the majority of the pixels is well below the maximum value used here. Only a few single pixels throughout the image will have such large deviations. Since the star is spread over a few pixels and the extreme DSNU pixels are usually isolated, they should never hinder the detection of stars as seriously as assumed possible in this case. Using the typical response of a CCD, the peak amplitude of the curve in Figure 9-4 will be at 2 mV instead of 1.6 mV. During normal operation the temperature should also be below 5°C.

These calculations prove that a magnitude 6 star should be detectable above the DSNU at 5°C. The assumption was made that the Sony ICX059 should be sufficient for the purposes required by the Sunsat star sensor.

10. Conclusion

This chapter summarizes the results presented throughout the thesis. It shows that the desired design specifications, listed in the Introduction, have been met.

The *Region Growing* module of Chapter 2 was found to be a fast and reliable method of extracting stars from a noisy image. The probability of mistaking a false object as a star, is minimised by verifying that the shape and size of the object corresponds to that of a star.

Chapter 3 showed that the *Matching* algorithm will match an imaged constellation to the star catalogue within an acceptable period of time. If the other attitude sensors do not maintain their specified accuracy, it will be used to initialise the *Tracking* algorithm. The *Matching* algorithm is also used extensively during calibration and testing of the camera.

The *Tracking* algorithm of Chapter 4 enables the star sensor to provide ADCS with matching observed and reference vectors, well within the desired 1 s period. Simulations show it to be a reliable means of updating the orientation.

The feasibility study presented in Chapter 9 indicated that, in the worst case, magnitude 6 stars should be detectable above the fixed pattern noise. Only tests could show whether the electronic noise levels were low enough. Chapter 8 proves beyond any doubt that magnitude 6 stars are visible with a one second integration time.

A fast and simple calibration procedure was developed and summarized in Chapter 6. Proper calibration of the camera allows angular distances between stars to be measured with an accuracy (1σ) of $78\ \mu\text{rad}$. The position of a star can be pinpointed with an accuracy (1σ) of $55.4\ \mu\text{rad}$. This is better than $12''$, which justifies the classification of the star sensor as a *high precision* instrument. The magnitude can be estimated with an accuracy (1σ) of $0.2\ \text{Mv}$. The dynamic range of the sensor is, however limited to the range $\text{Mv } 2.5$ to 6 . This only eliminates the 35 brightest stars from the catalogue of 5000, which is not a significant reduction.

The electronic circuitry was designed to minimize power consumption. Chapter 8 showed that the maximum power delivered to the circuit is $1.7\ \text{W}$ and the average power is $1.3\ \text{W}$. Approximately $500\ \text{mW}$ is drawn by the *T800*, which means that the camera, memory and interface circuitry draw $800\ \text{mW}$. The peak power is consumed for very short periods when the memory is accessed repeatedly.

Mass produced commercial components kept the overall cost of the star sensor to an absolute minimum. The reliability of these components will only be determined once the satellite is in orbit. Table 10-1 summarizes the cost of components.

Component	Cost
CCD and peripherals	R700.00
lens	R740.00
T800	R400.00
Static RAM (512 k)	R450.00
Static RAM (128 k)	R160.00
Flash RAM	R400.00
other components	R500.00
total	<u>R3350.00</u>

Table 10-1 Summary of costs

The cost to initially develop and manufacture the PCB's and mechanical housing for the star sensor is a few thousand Rand. Once these components are produced in larger quantities, the cost is significantly reduced. All the components for the star sensor should cost in the vicinity of R5000.00, which justifies its classification as a *low cost* instrument.

In conclusion, all specifications set for the star sensor were met or exceeded.

References

1. Steyn W H, *A Multi-Mode Attitude Determination and Control System for Small Satellites*, PhD Thesis, University of Stellenbosch, Stellenbosch, December 1995.
2. Wertz J R, *Spacecraft Attitude Dermination and Control*, Kluwer Academic Publishers, Dordrecht, 1991.
3. van Bezooijen R W H (1989), "A Star Pattern Recognition Algorithm for Autonomous Attitude Determination", *11'th IFAC Symposium on Automatic Control in Space*, Tsukuba, Japan, 1989, pp. 51-58.
4. *Rowley Parallel Modula-2 User Manual*, Third Edition, Rowley Associates, 32 Rowley, Cam, Dursley, Gloucester, GL11 5NT, May 1992.
5. Greyling B, *A Charge Coupled Device Star Sensor System for a Low Earth Orbit Microsatellite*, Masters thesis, University of Stellenbosch, Stellenbosch, December 1995.
6. Ballard D H, Brown C M, *Computer Vision*, Prentice Hall, New Jersey, 1982.
7. Havelock D I, "Geometric Precision in Noise-Free Digital Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11 No. 10, October 1989, pp. 1065-1075.
8. Emsley H H, *Aberrations of Thin Lenses*, University Press, Glasgow, 1956 .
9. Simon R E, *Electro-Optics Handbook*, RCA Corporation, USA, 1974.
10. Baldini D, Barni M, Foggi A, Benelli G, Mecocci A, "A New Star-Constellation Matching Algorithm for Satellite Attitude Determination", *ESA Journal*, Vol 17, 1993, pp. 185-198.
11. Königsmann H J, Rath H J, Renken H, "Attitude Determination by Image Processing Algorithms", ZARM, University of Bremen, 1992.
12. Kan D, Yang J, Ye P, Zhu Z, Guo R, "Star Reference Autonomous Attitude Determination", Beijing Institute of Control Engineering.
13. Shuster M D, Oh S D, "Three-Axis Attitude Determination from Vector Observations", *AIAA Journal of Guidance, Control and Dynamics*, Vol. 4, No. 1, Silver Springs, 1980, pp. 70-77.
14. Conversation with W H Steyn during September 1995.

15. Weng J, Cohen P, Herniou M, "Camera Calibration with Distortion Models and Accuracy Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No 10, October 1992, pp. 965-980.
16. van Zyl J E, *Ontsluier die Heelal, 'n Inleiding tot Sterrekunde*, Die Potchefstroom Herald, Potchefstroom, August 1993.
17. Kerrod R, *Stars and Planets*, Kingfisher Books, London, 1983.
18. Anton H, *Elementary Linear Algebra*, 5'th Edition, Wiley, New York, 1987.
19. Sony *ICX059* data sheet, 1993.
20. Sony *CXD1261* data sheet, 1993.
21. *Flash Memory*, Volume I, Intel data book, 1994.
22. Conversation with D Griffith, optical designer at the CSIR, during December 1995.
23. *CCD Imaging III*, Guidelines published for imaging with EEV CCD sensors, Essex, 1987.
24. Pierret R F, *Field Effect Devices*, Second Edition, Addison-Wesley, New York, 1990.

APPENDIX A

Astronomy

This Appendix introduces the basic concepts of astronomy and lists the tables of variable stars and stars with significant proper motion, referred to in Chapter 7.

A.1 Introduction to astronomy

A wide variety of books cover the basic concepts of astronomy. This section briefly explains the celestial sphere, right ascension, declination and magnitude.

All celestial bodies are visualised to be situated on the celestial sphere [17, p. 21]. It is centred on the earth's centre. The celestial equator lies in the plane of the earth's equator. The line drawn through the earth's poles intersects the celestial poles, with the celestial North pole located above the earth's North pole and the celestial South pole below the earth's South pole. The celestial sphere is stationary in the celestial coordinate system (or inertial space). The earth revolves around its own axis, at the centre of the sphere, once a day and the sun completes one revolution around the celestial sphere annually.

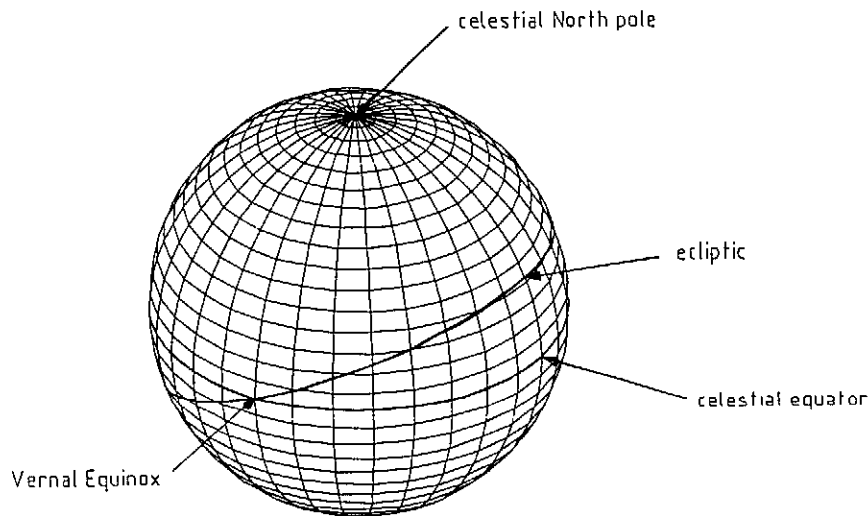


Figure A-1 The celestial sphere

The path traced by the sun on the celestial sphere is known as the ecliptic. The plane of the ecliptic is the plane of the earth's orbit around the sun and is slanted relative to the celestial equator by 23.5° . The ecliptic intersects the celestial equator twice a year at the time of the equinoxes. Vernal equinox occurs on about 21 March when the sun is travelling north and intersects the celestial equator in the First Point of Aries. On

about 23 September it is travelling south and intersects the equator again. Due to precession of the equinoxes, the First Point of Aries shifts slowly. About 2000 years ago it was located in Aries, but today it is in Pisces. A band of about 9° on either side of the ecliptic contains the paths of the moon and most planets and is called the zodiac.

The position of any celestial body on the sphere is specified in terms of its right ascension (RA) and declination (Dec). The RA of a star is its celestial longitude. It is measured eastward from the First Point of Aries, along the celestial equator, to the great circle passing through the star and the celestial poles. It ranges from 0° to 360° . The Dec is the celestial latitude of the star. It is the angular distance measured from the celestial equator towards the poles. The North pole is located at 90° and the South pole at -90° .

The apparent brightness of a star is measured on a scale developed by the ancient Greeks. According to this, a star of magnitude 0 is 100 times brighter than a star of magnitude 5. The brightest star in the sky is Sirius, with a magnitude of -1.45. Appendix B describes magnitude calculations.

Various projections exist to represent the celestial sphere on a flat surface. The basic rectangular projection of Figure A-2 shows all 5026 stars which are brighter than magnitude six. Since the polar region has been stretched in the projection (the actual point where the pole is located has been stretched to a line) the stellar density appears to be very low towards the poles. The polar regions are sparsely populated compared to the equatorial belt, but the projection exaggerates this phenomena.

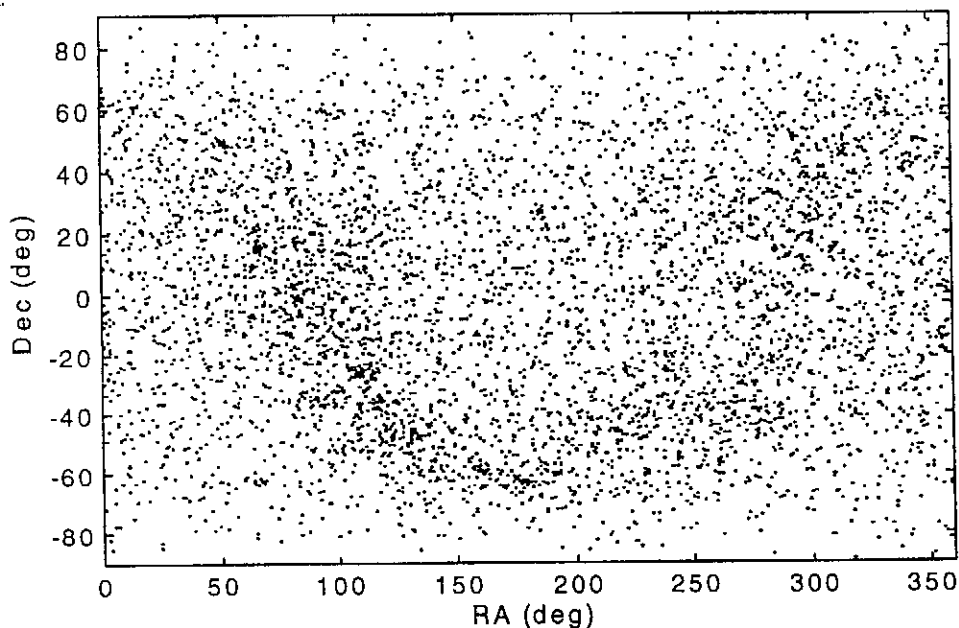


Figure A-2 All stars mapped onto a flat plane

A.2 Tables of stars with proper motion and variable stars

The stars with significant variation in magnitude and position have been extracted from the tables presented by van Zyl [16, p. 366-372 and p.200] and listed in Tables A-1 and A-2.

RA (hours min)	Dec (deg min)	Magnitude (Mv)		Period (days)
		Maximum	Minimum	
Algol type				
03 07	+40 55	2.12	3.4	2.867
04 00	+12 28	3.3	3.8	3.9
05 29	-00 20	2.4	2.6	5.7
07 58	-49 13	4.7	5.2	1.45
15 00	-08 28	4.92	5.9	2.37
Beta Lyrea type				
07 17	-24 28	4.7	5.5	4.39
15 18	-40 45	5.4	6.0	2.16
16 48	-37 58	3.0	3.3	1.44
17 17	+33 07	4.6	5.3	2.05
18 50	+33 21	3.34	4.34	12.93
Delta Cephei type				
22 29	+58 21	3.48	4.37	5.366
06 28	+30 30	5.0	5.82	3.7
07 03	+20 30	3.66	4.16	10.15
09 44	-62 17	3.28	4.18	35.53
18 18	-18 53	5.4	6.1	5.77
18 52	-67 18	3.91	4.78	9.09
19 50	+00 53	3.48	4.39	7.17
19 54	+16 30	5.28	6.4	8.38
Delta Scuti type				
18 40	-09 06	4.6	4.79	0.19
W Virginis type				
18 56	-62 15	3.94	4.75	9.08
Long period variables				
02 19	-03 02	2.0	10.1	331
02 53	-49 56	4.7	14.3	403.97
09 47	11 29	4.4	11.3	312.43
16 53	-30 30	5.0	12.2	279.45
19 50	+32 53	3.3	14.2	406.93
Irregular type				
10 43	-59 25	-0.8	7.9	
RV Tauri				
18 47	-05 43	4.45	8.2	140.05

RA (hours min)	Dec (deg min)	Magnitude (Mv)		Period (days)
		Maximum	Minimum	
Semi regular type				
04 37	-62 06	4.8	6.6	338
06 14	+22 31	3.2	3.9	232.9
07 13	-44 37	2.6	6.2	140.42
05 55	+07 24	0.4	1.3	2110
16 29	-26 24	0.88	1.8	1733
17 14	+14 24	3.0	4.0	2170
23 01	+27 49	2.4	2.8	36

Table A-1 Table of significant variable stars

RA (hours min sec)	Dec (deg min sec)	Proper motion (sec)	Magnitude (Mv)
17 57 49	+04 41 36	10.25	13.4
14 39 37	-60 50 0	3.68	4.4
14 39 37	-60 50 0	3.68	5.7
06 45 09	-16 43 0	1.32	1.4
07 39 18	+05 13 30	1.25	2.6
01 43 32	-15 59 54	1.92	5.7

Table A-2 Table of stars with significant proper motion

APPENDIX B

Spherical Geometry

All catalogue star coordinates are given as right ascension (α) and declination (δ) on the celestial sphere. It is an imaginary sphere of unit radius centred on the earth. All stars are visualized to be located on the sphere. It is often necessary to calculate the angular distance between two catalogue stars. When collecting the stars in the FOV on the celestial sphere, the great circle passing through the FOV corners is used. These two concepts are briefly discussed in the following sections.

B.1 Angular distance between points

The angular separation d_{12} between any points P_1 and P_2 on the celestial sphere is given by Wertz [2, p. 728 equation (A-1)]

$$\cos d_{12} = \sin \delta_1 \sin \delta_2 + \cos \delta_1 \cos \delta_2 \cos(\alpha_1 - \alpha_2) \quad (\text{B.1})$$

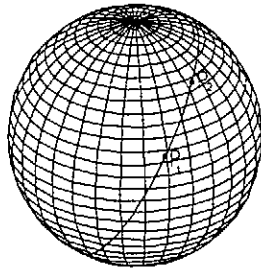


Figure B-1 Two points P_1 and P_2 on the celestial sphere.

B.2 Great circle through two points

The great circle passing through two points P_1 and P_2 is shown in Figure B-1. It is the line of intersection of a plane through the centre of the sphere, P_1 and P_2 and the sphere. Once the FOV corners have been located on the celestial sphere, it is necessary to calculate the declination δ of any point on the FOV edge. The right

ascension α of the point is known¹. The equations are once again given by Wertz [2, pp. 728,729 equations (A-7a) and (A-8)] but repeated here for convenience

$$\cot \beta_1 = \left(\frac{\tan \delta_1}{\tan \delta_2} \right) \frac{1}{\sin(\alpha_2 - \alpha_1)} - \cot(\alpha_2 - \alpha_1) \quad (\text{B.2})$$

$$\tan \delta = \begin{cases} \frac{\tan \delta_1}{\sin \beta_1} \sin(\alpha - \alpha_1 + \beta_1) & : \delta_1 \neq 0 \\ \frac{\tan \delta_2}{\sin(\alpha_2 - \alpha_1)} \sin(\alpha - \alpha_1 + \beta_1) & : \delta_1 = 0 \end{cases} \quad (\text{B.3})$$

These equations are implemented in the function *calcdelta* used in the software. It is called with $\alpha_1, \delta_1, \alpha_2, \delta_2$ and α . The declination of the point on the great circle is returned.

¹ Refer to paragraph 4.1.2

APPENDIX C

Magnitude and Distance Calculations

C.1 Magnitude calculations

The magnitude of an imaged star is estimated from the sum of the brightnesses of all the pixels. The brightness of a pixel is defined as the difference between the pixel value read from the image memory and the threshold of detection. The threshold of detection may be higher than the dark average, which is calculated from the leading and lagging dark columns¹. Background light from millions of fainter stars and light reflected by the atmosphere raise the detection threshold above the dark average. When imaging with full moon in Stellenbosch, for example, the threshold is about two to four quantization levels above the dark average. The brightness of the star is the sum of all the pixels' brightnesses plus an expression to compensate for the difference between the dark average and threshold of detection.

$$brightness_{star} = \sum_{all\ pixels} pixelbrightness + k_{area} n \Delta th \quad (C.1)$$

where

n = number of pixels in the star

k_{area} = factor compensating for the increase in area of the star (or number of pixels forming part of the star) below the threshold

Δth = difference between the threshold of detection and dark average, measured in units of quantization levels.

The difference between the threshold of detection and dark average varies between about 0 and 4 quantization levels. The last term in equation (C.1) compensates for the section of the star between the dark average and threshold of detection. The area of the star, or number of pixels included in the star, increases as the threshold approaches the dark average. The factor k_{area} corrects for this increase in area of the star below the threshold and is determined experimentally. Ideally stars should always be imaged in conditions where the background light level is as dark as the dark pixels of the CCD. This is usually not possible and the correction improves the accuracy of the magnitude estimate under imperfect imaging conditions.

The magnitude of the star is calculated from its total brightness. The illuminance is directly proportional to measured brightness:

$$E_{star} = k_{illum} \times brightness_{star} \quad (C.2)$$

where

¹ Refer to Figure 1-1.

k_{illum} = conversion factor from *brightness_{star}* to illuminance

This conversion factor is determined experimentally by comparing the brightness of an imaged star to its illuminance calculated from the catalogue. To do so, the conversion between illuminance and magnitude M_v (listed in the catalogue) is required.

The concept of magnitude of a star dates back centuries to the early astronomers. A star of magnitude 0 is defined to be 100 times brighter than one of magnitude 5. The ratio of the illuminance of two stars differing in magnitude by 1 is $\sqrt[5]{100} = 2.512$. The equation given by Simon [9, p.66 equation (6.3)] relates the illuminance of two objects with magnitudes m and n :

$$\frac{E_m}{E_n} = (2.512)^{n-m} \quad (C.3)$$

Letting the object of illuminance E_n be a star of magnitude 0, the relation between any star's illuminance and magnitude in M_v is determined

$$E_{star} = 2.65 \times 10^{-6} \times (2.512)^{-M_v} \quad (C.4)$$

where the illuminance of a magnitude 0 star is quoted by Simon [6, p.66 Figure 6-5] as $2.65 \times 10^{-6} lux$. The inverse of this relation is

$$M_v = \log_{2.512} \left(\frac{2.65 \times 10^{-6}}{E_{star}} \right) \quad (C.5)$$

Equation (C.4) is used to determine the illuminance of the catalogue star corresponding to the imaged one. The constant k_{illum} is calculated by equating the catalogue illuminance of (C.4) and the observed illuminance of (C.2). This process is repeated for a large number of stars and the average value used. The value of k_{area} used in equation (C.1) is set before calculating k_{illum} . The best value of k_{area} will minimize the variation of k_{illum} calculated from different stars. A value of $k_{area} = 2$ was found to deliver the best results.

C.2 Distance calculations

Angular distances need to be calculated between pairs of observed stars on the CCD plane as well as catalogue stars on the celestial sphere.

C.2.1 Angular distance between two observed stars

The angular distance measured between two stars on the CCD plane² is determined from the dot product between the vectors in the star sensor (ss) coordinate system.

$$d_{12} = \arccos\left(\frac{\underline{S}_{1,ss} \cdot \underline{S}_{2,ss}}{|\underline{S}_{1,ss}| |\underline{S}_{2,ss}|}\right) \quad (C.6)$$

which simplifies to

$$d_{12} = \frac{u_1 u_2 + v_1 v_2 + f^2}{\sqrt{(u_1^2 + v_1^2 + f^2)(u_2^2 + v_2^2 + f^2)}} \quad (C.7)$$

when the vectors are broken down into components. Equation (C.7) is implemented in software as the *uvdist* function, accepting the *uv* coordinates of two stars and returning the angular distance in radians.

C.2.2 Angular distance between two catalogue stars

The coordinates of catalogued stars are given as right ascension (α) and declination (δ) on the celestial sphere. The cross section of the FOV restricts the angle to the first quadrant. The angular separation between stars 1 and 2 is given by equation (B.1)

$$d_{12} = \arccos[\sin \delta_1 \sin \delta_2 + \cos \delta_1 \cos \delta_2 \cos(\alpha_1 - \alpha_2)] \quad (C.8)$$

The function *addist* (*alpha-delta* distance) accepts the alpha and delta coordinates of two points and returns their angular separation.

² Refer to Figure 1-3

APPENDIX D

Matching algorithm

D.1 Storage and searching of star catalogues in memory

The method of storing and searching the star catalogue referred to in Chapter 3 is done in one of two ways, depending on execution speed required and memory available. Assuming that high speed is required and enough memory is available, or that the uncertainty area is limited to a small enough section of the catalogue, the procedure suggested in [3] is followed. Distances between all possible pairs of guide stars are calculated and tabulated. The table is sorted according to increasing distance as shown in Table D-1.

guide star 1	guide star 2	Distance d_{12}°
233	487	0.2502
201	500	0.2514
878	944	0.2512
⋮	⋮	⋮
⋮	⋮	⋮
444	634	13.7895
33	212	13.8433
178	179	13.8884

Table D-1 Storing the catalogue as sorted pairs in a distance table

It is only necessary to tabulate those pairs of which the distance is smaller than the field of view. To further limit the size of the catalogue, no two stars which are closer than 0.25° (about 15 pixels) are used. This is justified by the fact that the minimum diameter of an observed star is 3 pixels and many stars are much larger. Therefore it is reasonable to assume that it should not be necessary to use observed stars with separation smaller than 0.25° . No two stars with separation exceeding the FOV diagonal belong in the distance table either. This angle is 13.9° .

Using a sorted catalogue simplifies the selection of guide star pairs. The catalogue is searched until the first distance larger than $d_{12} - d_{\text{uncert}}$ is found. All consecutive guide star pairs are then selected until the distance between the selected pair is larger than $d_{12} + d_{\text{uncert}}$. The disadvantage of this method is the immense memory requirements.

The entire catalogue including all stars to magnitude $M_v = 6$ (5026 stars) has 288464 pairs with separation varying between 0.25° and 13.9° . Each entry in the distance table has to contain the following information:

- catalogue number of guide star i : cardinal (2 bytes)
- catalogue number of guide star j : cardinal (2 bytes)
- angular distance between guide star i and j : real (4 bytes)

requiring a total of $288464 \times 8 \text{ bytes} \approx 2.3 \text{ Mbytes}$. This is an unrealistic amount of memory for the star sensor. It can be minimized by encoding the data. The unique number of a star in a catalogue containing 5026 stars needs a minimum of 13 bits. Since the observed angular distance can be measured to an accuracy of say $50 \mu\text{rad}$ or 0.0029° , the guide star distances require no greater accuracy. The number of quantization levels needed is $13.9^\circ/0.0029^\circ = 4878$ or at least 13 bits. The two unique numbers and angular distance still require a total of 39 bits. The encoding becomes very involved when the different entries in the table are allowed to share a byte for storage. For each table entry (row) to have its own set of bytes, at least five bytes are required per entry. The minimum amount of memory for storing the distance table is then $288464 \times 5 \text{ bytes} \approx 1.44 \text{ Mbytes}$.

The catalogue referenced by the distance table is stored in memory as well. Since the star sensor only has 1 Mbyte of Flash RAM for the catalogue, distance table and software code, another option was considered instead.

The uncertainty area is greatly reduced by the a-priori attitude information. The sensible option is to create a sub-catalogue containing only those stars in the area of the catalogue where the star sensor is known to be directed. Using a sub-catalogue has the following advantages:

- Memory requirements are reduced. Both memory required for storing the catalogue and distance table and that needed for the large arrays (especially *matcho*, *matchg* and *matchgroup* discussed in Section 3.3.1) used by the matching algorithm are reduced.
- Search speed is increased.
- Reliability is improved. Fewer stars are used and consequently the probability of selecting an incorrect match is reduced.

In-flight updates of the catalogue is the only complication involved when using sub-catalogues. The orbit parameters vary continuously, causing the star sensor to scan a different section of the celestial sphere. The Topspeed Modula programme *MakeCat.mod* (listed in Appendix I and discussed in Chapter 5) determines the uncertainty area for a given orientation and orbital parameters. It generates a sub-catalogue while displaying the position of the FOV on the celestial sphere. The programme *DistTab.mod* generates the distance table.

Hardware and software was specifically designed to allow in-flight updates. Flash RAM enables the catalogue to be updated in flight but to remain unaffected during periods when the star sensor power supply is off. Software for downloading the catalogue to Flash RAM is described in Appendix G. The same sub-catalogue should remain in use for a period of at least one to two weeks.

Instead of using a distance table to find all guide star pairs matching the first observed pair, they can be selected directly from the catalogue. The memory used by the distance table is saved, but an immense number of calculations are required. As an example, consider the case where 8 observed stars and a sub-catalogue of 1000 stars are used. For each observed star pair, the entire catalogue has to be scanned for pairs of stars with matching separation. In the catalogue there are $999 + 998 + 997 + \dots + 1 = 999 \times 1000 / 2 = 499500$ pairs for which the separation has to be calculated. Since values are not stored in a table, this process has to be repeated for each pair of observed stars i.e. $7 + 6 + 5 + \dots + 1 = 7 \times 8 / 2 = 28$ times. Each one of these $28 \times 499500 = 13.986\text{M}$ distance calculations involve several references to memory, trigonometric functions and multiplications (see Appendix C).

Such a multitude of calculations do not only waste time, but even more critical on the satellite is the unnecessary power consumption they bring about. The memory for storing the distance table is available and the extra power dissipated and time wasted by this method can not be justified.

The star vector concept mentioned in Section 3.2.2 can be used to reduce the number of calculations. Instead of searching the entire catalogue, a star vector is associated with each observed star. It contains all guide stars of which the magnitude lies within the magnitude tolerance from that measured for the observed star. Each star vector contains only a fraction of the catalogue and only these vectors are searched for matching pairs of guide stars. Suppose that on average a star vector contains as many as 100 of the 1000 catalogue stars mentioned in the above example, the number of calculations is reduced to $(99 \times 100 / 2) \times 28 = 138600$, a 100 fold reduction. Immediately this option becomes viable, but the distance table method remains the first choice. If for any reason it is found before launch that in-flight updating of the catalogue is impossible, the entire catalogue of 5026 stars will be placed in the Flash RAM. No space will be available for the distance table, which will necessitate the implementation of this method.

D.2 Building the *matchgroups* in *Match.mod*

This section briefly explains the method used to generate the *matchgroup* vector in the module *Match.mod*. It is illustrated by means of match vectors *matcho* and *matchg* of Table D-2 and *matchgroup* of Table D-3.

match no	<i>matcho</i>	<i>matchg</i>
1	1	5
2	2	12
3	1	5
4	2	33
5	1	5
6	3	15
7	1	5
8	3	40
9	1	5
10	5	22
11	1	5
12	5	54
13	1	5
14	5	56
.	.	.
.	.	.
.	.	.

Table D-2 The match vectors

The first observed star is selected as *kmatcho* (*kernel match observed star*). The second observed star is selected as *secmatcho* (*secondary match observed star*). The match vectors of Table D-2 are searched until a match is found with either *ostar1=kmatcho* and *ostar2=secmatcho* or vice versa. The first match is entered in the first row of *matchgroup* and consecutive matches in the following rows. Once all n ($n=2$ for the first case) matches have been found, this set of $2 \times n$ entries is referred to as a block (indicated with double frame in Table D-3) of the *matchgroup*.

The counter *secmatch* is incremented and the search repeated. For each match found with this new value of *secmatch*, the entire block found in the previous step is duplicated and the new match appended to all rows of the block.

This process of incrementing *secmatch*, searching for matches and duplicating blocks is repeated until all observed stars, excluding *kmatcho*, have been used as *secmatch*. Next *kmatch* is incremented and the entire sequence repeated. Since *kmatch* is no longer the first star, repeated increments of *secmatch* will wrap around the last observed star to become the first observed star.

matchgroup									
group no	Kernel	associated match number							Confidence
	1	2	3	4	5	6	7	8	
1	1	2	6	10					
2	1	4	6	10					
3	1	2	8	10					
4	1	4	8	10					
5	1	2	6	12					
6	1	4	6	12					
7	1	2	8	12					
8	1	4	8	12					
9	1	2	6	14					
10	1	4	6	14					
11	1	2	8	14					
12	1	4	8	14					
13	2	...							
.
.
.

Table D-3 Building the match group vector

This procedure ensures that all match groups are unique and none are omitted.

D.3 Graphs of the matching times

The graphs referred to in Section 3.5 are shown here. The uncertainty values and 1σ errors are equal.

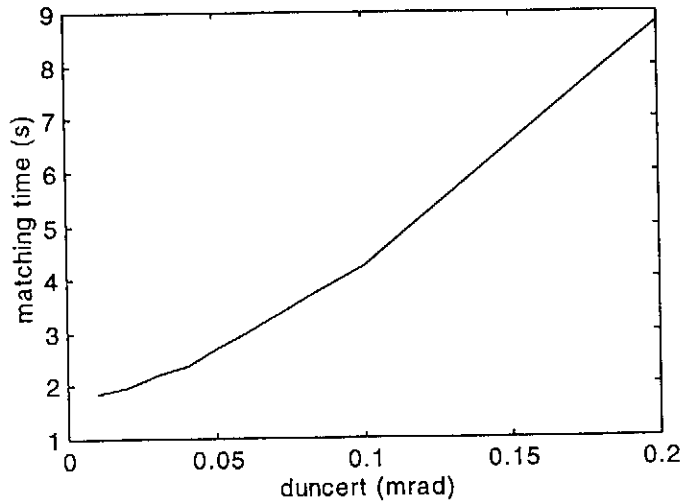


Figure D-1 Average matching times for different values of d_{uncert} , $Mv_{uncert} = 0.3$

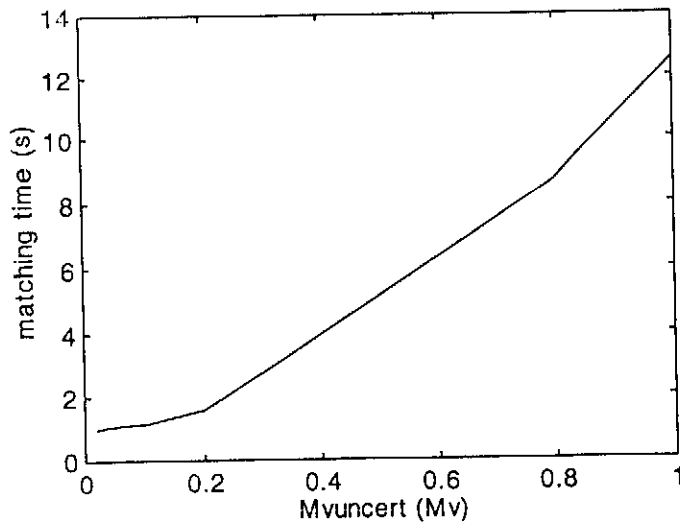


Figure D-2 Average matching times for varying Mv_{uncert} , $d_{uncert} = 0.05$ mrad

APPENDIX E

Tracking Algorithm

E.1 Accuracy of the Triad algorithm

This Appendix determines the maximum error expected from the *Triad* algorithm. Before estimating the *Triad* error, the motion of a star on the CCD plane due to roll, pitch and yaw of the satellite is examined.

E.1.1 Motion of a star due to roll, pitch and yaw

During and prior to earth imaging, the roll and pitch of the satellite are close to zero while the yaw is either -45° or 135° . Under these conditions, the star sensor's boresight is parallel to the pitch axis. A roll or yaw of the satellite will cause a star in the image to translate, while pitch will rotate it. A small roll of the satellite will translate the star in the vertical (v) direction by

$$\Delta v \approx \Delta \phi f \quad (\text{E.1})$$

where

Δv = small motion of the star in the v direction in mm

$\Delta \phi$ = small change in roll angle of the satellite in rad

f = focal length in mm

while yaw will shift a star in the horizontal (u) direction by

$$\Delta u \approx \Delta \psi f \quad (\text{E.2})$$

The distance calculated here is actually the arc length and not the straight line distance. The equations are only used for the error angles, which are assumed to be in the order of milli radians and smaller. For these small angles the results are accurate.

A change in pitch of the satellite will cause the star to rotate about the optical centre. The angle of rotation will equal the pitch angle and the translation on the CCD plane is a function of the radial distance

$$\Delta l \approx \Delta \theta r \quad (\text{E.3})$$

where

Δl = translation on CCD plane

$\Delta\theta$ = small change in pitch

r = radial distance from optical centre to star

The radial distance r can never exceed half the CCD diagonal, or 3 mm . The u and v components of the translation may be calculated if the angle between the radial line to the star and the u axis is known.

The ratio of translation of the star due to pitch compared to roll (or yaw) is approximately the ratio of radius r to focal length f . This is always smaller than $3/25$. In other words, an error in pitch will move the star by a smaller distance on the CCD plane than a roll or yaw error. On the other hand, a small positional error of the star may lead to a much greater error in the pitch estimate than the roll or yaw estimate.

E.1.2 Estimating the total error

The aim of this section is to determine the maximum error in the positional estimate of a star's centroid on the CCD plane. The error occurs due to incorrect roll, pitch and yaw angles calculated using the *Triad* algorithm. These angles are calculated incorrectly due to errors in the observed positions of the two stars used by the *Triad* algorithm.

Section E.1.2.1 determines the maximum error in roll, pitch and yaw when calculating orientation with the *Triad* algorithm. Section E.1.2.2 determines the effect of the orientation errors on the positional estimate of a star's centroid on the CCD plane.

E.1.2.1 Estimating the orientation error

Star catalogues give star positions with greater precision than that obtained from the star sensor. Consequently it is assumed that the error in the observed vectors dominate those of the reference vectors.

Only the two dimensional positional error of the observed star on the CCD plane is taken into account, so that

$$\begin{aligned} u &= \langle u \rangle + \varepsilon_u \\ v &= \langle v \rangle + \varepsilon_v \end{aligned} \tag{E.4}$$

where

u = measured u position of observed star

v = measured v position of observed star

$\langle u \rangle$ = true u position where star should have been observed if no measurement errors were made

$\langle v \rangle$ = true v position of observed star

ε_u = measurement error in mm along the u axis

ε_v = measurement error in mm along the v axis

The error in the roll, pitch and yaw angles estimated with the *Triad* algorithm, due to the observation error, will be determined here. The following procedure is followed:

As a first step, two true observation vectors are generated on the CCD plane. From these, the true observation vectors are calculated in body coordinates. The true roll, pitch and yaw rotations ($\langle\phi\rangle, \langle\theta\rangle$ and $\langle\psi\rangle$) are used to calculate the true reference vectors in orbit coordinates.

The next step is to generate the observed vectors according to (E.4) where an error is intentionally introduced. The two observed vectors in body coordinates are calculated from these. The observed vectors and the true reference vectors are used with the *Triad* algorithm to estimate the orientation. The estimated orientation is subtracted from the true orientation (used to calculate the true reference vectors). The absolute values of the differences are the errors in the estimates. The roll, pitch and yaw errors are:

$$\begin{aligned}\varepsilon_\phi &= |\langle\phi\rangle - \phi| \\ \varepsilon_\theta &= |\langle\theta\rangle - \theta| \\ \varepsilon_\psi &= |\langle\psi\rangle - \psi|\end{aligned}\tag{E.5}$$

For the purposes of these calculations, any error in the focal length is ignored. The aim of this section is purely to determine the accuracy with which a star can be pinpointed on the CCD plane. An error in the focal length will cause an error in the orientation estimate¹, but when using the estimated error to find stars on the CCD plane, the same faulty focal length will be used again. Consequently an error in the focal length will have no net effect on the estimated position of stars.

Since the error of (E.5) is a function of the roll, pitch and yaw, the orientation of the satellite is assumed to be close to that when the star sensor is in use. In other words roll and pitch are close to zero while yaw is either -45° or 135° .

Having fixed the orientation, the error still depends on the observation error as well as the relative positions of the observed vectors. To determine the effects of these variables, the two situations were examined independently. Results for various cases were examined in Matlab.

The following conclusions were drawn from these results:

If the two stars used by the *Triad* algorithm are well separated, the worst roll error occurs when both observed vectors are shifted in the v (vertical) direction.

$$\begin{aligned}u_i &= \langle u_i \rangle \\ v_i &= \langle v_i \rangle \pm \varepsilon_v\end{aligned}\tag{E.6}$$

¹ The error is introduced in the *uvto* transformation

$$\begin{aligned} u_2 &= \langle u_2 \rangle \\ v_2 &= \langle v_2 \rangle \pm \varepsilon_v \end{aligned} \quad (\text{E.7})$$

The roll error ε_ϕ is calculated using equation (E.1).

$$\varepsilon_\phi \approx \frac{\varepsilon_v}{f} \text{ rad} \quad (\text{E.8})$$

The worst yaw error occurs when both observed vectors are shifted in the u (horizontal) direction.

$$\begin{aligned} u_1 &= \langle u_1 \rangle \pm \varepsilon_u \\ v_1 &= \langle v_1 \rangle \end{aligned} \quad (\text{E.9})$$

$$\begin{aligned} u_2 &= \langle u_2 \rangle \pm \varepsilon_u \\ v_2 &= \langle v_2 \rangle \end{aligned} \quad (\text{E.10})$$

and the error ε_ψ is given by

$$\varepsilon_\psi \approx \frac{\varepsilon_u}{f} \text{ rad} \quad (\text{E.11})$$

Equations (E.8) and (E.11) only hold when the observed stars are very well separated. If they move closer, another error dominates. This error occurs when the pitch error is a maximum. The errors of (E.8) and (E.11) are halved when only one of the observed stars has an observed error, while the other star's position is accurate.

The maximum error in the pitch estimate occurs when the two stars appear shifted in the direction perpendicular to the line joining their centres. This is illustrated in Figure E-1, which shows that the pitch error is

$$\varepsilon_\theta \approx \frac{2\varepsilon_{uv}}{r_{12}} \text{ rad} \quad (\text{E.12})$$

where

r_{12} = distance in mm separating the two stars

ε_{uv} = error on the uv plane, in any direction, measured in mm

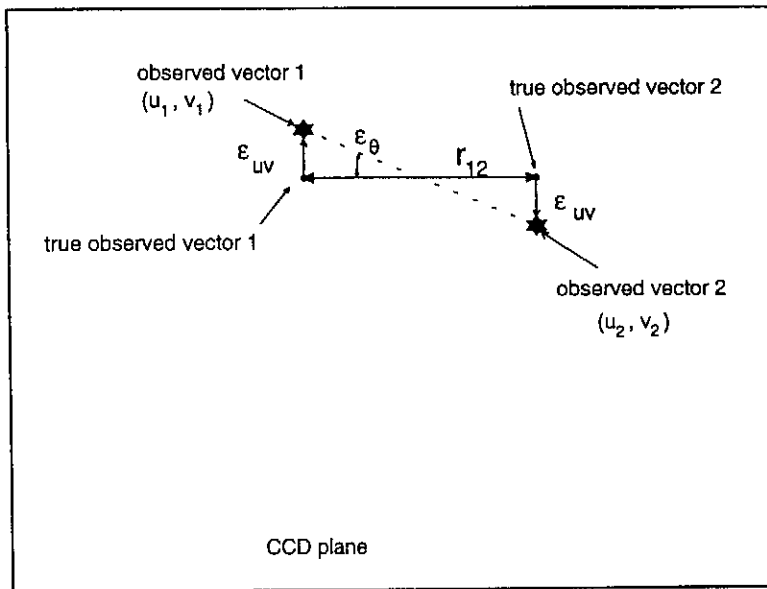


Figure E-1 Maximum pitch error occurs when stars appear shifted perpendicular to the line joining their centroids

Equation (E.12) holds for any position of the observed stars on the CCD plane. As the two observed stars approach one another, the pitch error increases. In the limit where the stars are on top of one another, no pitch estimate is possible.

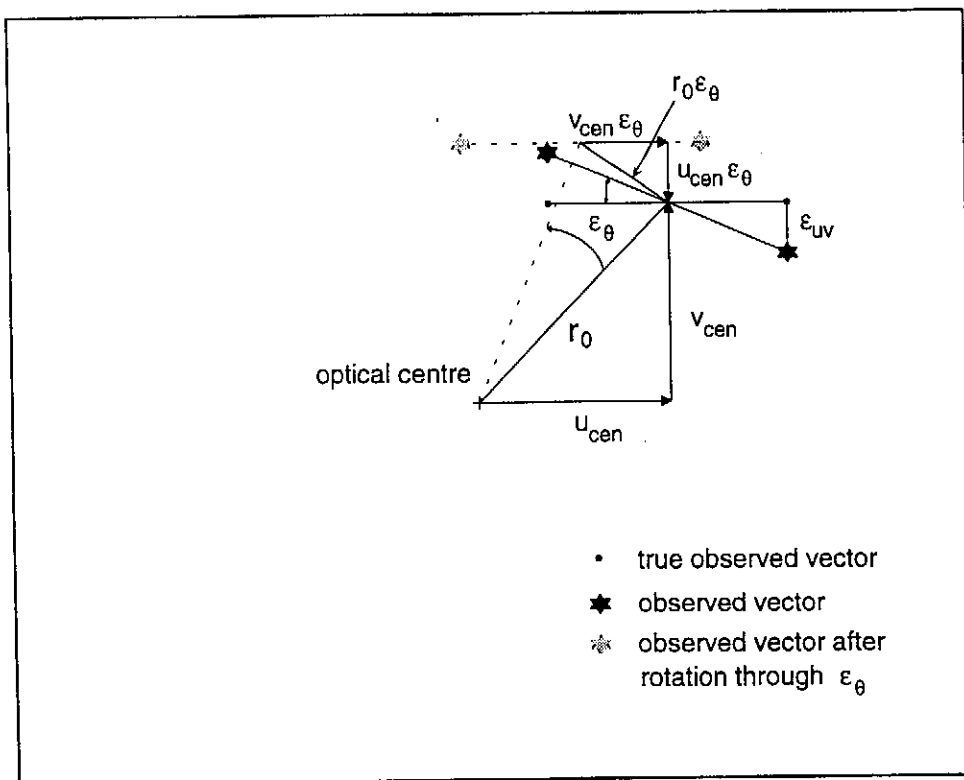


Figure E-2 Relation between pitch error ε_θ and the roll and yaw errors

When using the incorrect observed positions shown in Figure E-1 to estimate the orientation, the roll and yaw angles will be incorrect as well. Figure E-2 illustrates how the observed vectors need to be shifted to place them on the true observed vectors. The observed vectors are rotated around the optical centre, through ε_θ , which aligns them with the true observed vectors. A yaw of the satellite through angle $v_{cen}\varepsilon_\theta$ and roll through $u_{cen}\varepsilon_\theta$ corrects for the yaw and roll errors.

The error in the roll angle associated with the error in pitch is given by equation (E.8) with $\varepsilon_v = u_{cen}\varepsilon_\theta$:

$$\varepsilon_\phi \approx \frac{u_{cen}\varepsilon_\theta}{f} \quad rad \quad (E.13)$$

where

$$u_{cen} = \left| \frac{u_1 + u_2}{2} \right| \quad (E.14)$$

The yaw error is given by (E.11) with $\varepsilon_u = v_{cen}\varepsilon_\theta$

$$\varepsilon_\psi \approx \frac{v_{cen}\varepsilon_\theta}{f} \quad rad \quad (E.15)$$

where

$$v_{cen} = \left| \frac{v_1 + v_2}{2} \right| \quad (E.16)$$

The values of u_{cen} and v_{cen} given in equations (E.14) and (E.16) are only used when both observed stars have equal errors ε_{uv} . When only one star has an error while the other is accurate, the values of u_{cen} and v_{cen} are the uv coordinates of the accurate star. Since it is not known which of the two observed stars have the largest error, the maximum u and v coordinates of the two stars are used to calculate the maximum error. Replacing (E.12) in (E.13) and using the maximum uv coordinates, the maximum error in roll associated with the pitch error is determined as

$$\varepsilon_\phi \approx \frac{2u_{max}\varepsilon_{uv}}{f r_{12}} \quad rad \quad (E.17)$$

and the maximum error in yaw is

$$\varepsilon_\psi \approx \frac{2v_{max}\varepsilon_{uv}}{f r_{12}} \quad rad \quad (E.18)$$

with

$$u_{max} = \max(|u_1|, |u_2|)$$

$$v_{max} = \max(|v_1|, |v_2|)$$

When the separation between the observed stars is large and the stars are close to the optical centre, the maximum roll and yaw errors are given by (E.8) and (E.11). Otherwise the errors given by (E.17) and (E.18) dominated. In summary, the upper bounds on the roll and yaw errors are

$$\epsilon_{\phi, max} \approx \max\left(\frac{2u_{max}\epsilon_{uv}}{f r_{12}}, \frac{\epsilon_v}{f}\right) \text{ rad} \quad (\text{E.19})$$

$$\epsilon_{\psi, max} \approx \max\left(\frac{2v_{max}\epsilon_{uv}}{f r_{12}}, \frac{\epsilon_u}{f}\right) \text{ rad} \quad (\text{E.20})$$

The maximum pitch error is given by (E.12).

E.1.2.2 Determining the positional error

Having determined the upper bound on the orientation error, the next step is to determine the error in the position of the star. As a result of the incorrect orientation, the expected position of the star on the CCD plane will be calculated incorrectly.

The positional error on the CCD plane due to an orientation error is derived from equations (E.1) to (E.3)

$$\epsilon_{pos, \phi} \approx f \epsilon_{\phi, max} \quad (\text{E.21})$$

$$\epsilon_{pos, \theta} \approx r \epsilon_{\theta, max} \quad (\text{E.22})$$

$$\epsilon_{pos, \psi} \approx f \epsilon_{\psi, max} \quad (\text{E.23})$$

The positional error due to roll, $\epsilon_{pos, \phi}$, is in the v direction and that of yaw, $\epsilon_{pos, \psi}$, is in the u direction. The error due to pitch is perpendicular to the radial line to the expected star and is a function of the radial distance r .

To obtain the maximum positional error, the first bracketed values in equations (E.19) and (E.20) are replaced in (E.21) and (E.23). These roll and yaw errors are in perpendicular directions. The maximum pitch error given by equation (E.12), is replaced in (E.22). This error may add to the roll and yaw errors.

$$\epsilon_{pos, max} \approx \frac{2\epsilon_{uv} \sqrt{u_{max}^2 + v_{max}^2}}{r_{12}} + r \frac{2\epsilon_{uv}}{r_{12}} \quad (\text{E.25})$$

In the worst case:

$$u_{max} = 2.44 \text{ mm}$$

$$v_{max} = 1.82 \text{ mm}$$

$$\sqrt{u_{max}^2 + v_{max}^2} = 3 \text{ mm}$$

$$r = 3 \text{ mm}$$

The upper bound on the positional error is then

$$\epsilon_{pos,max} \approx \frac{12\epsilon_{uv}}{r_{12}} \text{ mm} \quad (\text{E.26})$$

The error in the positional estimate increases linearly with the observed error of the two stars used by the *Triad* algorithm. It is inversely proportional to the distance r_{12} separating them on the CCD plane. Table E-1 shows a few significant values of the positional error assuming a maximum observed error of 0.5 of a pixel, $\epsilon_{uv} = 0.0032 \text{ mm}$. The value selected for this maximum error should be between the 1 σ measurement error² of 0.25 *pixels* and the maximum error detected during measurements. Using the maximum measured value will lead to unnecessarily large search areas most of the time. When using the 1 σ measurement error as the maximum, stars may often not appear within the search area. As a compromise, a maximum *uv* error of 0.5 *pixels* was selected. Using the histogram of Figure 6-8, the probability of a star's positional³ accuracy being better than 0.5 *pixels*, is estimated as 98.22 %. The probability of both observed stars having an accuracy better than 0.5 *pixels* is 96.5 %. It should also be kept in mind that equation (E.26) gives the error in the position estimated for a star when the two observed stars used to determine the orientation are displaced as illustrated in Figure E-2. If the two observed stars are displaced in any other directions, the error in attitude and the positional estimate for the new star will be smaller than that of (E.26).

r_{12}		$\epsilon_{pos,max}$	
<i>mm</i>	<i>pixels</i>	<i>mm</i>	<i>pixels</i>
0.13	20	0.295	46
0.5	77	0.078	12
1	154	0.039	6
2	308	0.0195	3

Table E-1 Maximum positional error for an observed error of $\epsilon_{uv} = 0.5 \text{ pixel}$.

When using the last bracketed values in (E.19) and (E.20) together with (E.12), the maximum positional error is

² Refer to Section 6.3

³ The histogram contains entries for the distances between two stars. The 0.5 *pixels uv* error in the position of a star converts to a distance error 0.163 *mrad*, as shown in equation (6.23). This value is used in the histogram of Figure 6-8 to estimate the probability.

$$\begin{aligned}
\varepsilon_{pos,max} &\approx \sqrt{2}\varepsilon_{uv} + r \frac{2\varepsilon_{uv}}{r_{l2}} \\
&= \varepsilon_{uv} \left(\sqrt{2} + \frac{6}{r_{l2}} \right) mm
\end{aligned} \tag{E.27}$$

It only exceeds the error of (E.26) when $r_{l2} > 4.24\text{ mm}$. Consequently its maximum value is 1.4 pixels . Equation (E.26) is used to estimate the maximum error. If the error is smaller than 2 pixels , a conservative value of 2 pixels is used.

E.2 Derivation of the predictor

A predictor which assumes constant acceleration to determine the future position from the previous three values. The linear equations of motion are used as starting point:

$$x(t) = x_0 + \dot{x}_0 t + \frac{1}{2} \ddot{x}_0 t^2 \quad (\text{E.28})$$

where the subscript 0 denotes the variable at time $t = 0$. The initial position is the last position sampled, while the initial velocity and acceleration are estimated from previous samples. A discrete equation will be derived which estimates the position at time t , where t is taken as one sample interval later than $t = 0$, i.e. at $t = T$. T is the sampling period.

Equation (E.12) is discretized by letting $t = kT$. This gives:

$$x(kT) = x_0 + \dot{x}_0 kT + \frac{1}{2} \ddot{x}_0 (kT)^2 \quad (\text{E.29})$$

The last three sampled values of x are known while the present value is desired. Let these known values be $x(k-3)$, $x(k-2)$ and $x(k-1)$. The value of x expected at the present sample is required, which is $x(k)$. The position at sample $k-1$ replaces x_0 . The velocity is estimated as

$$\dot{x}(k) = \frac{x(k) - x(k-1)}{T} \quad (\text{E.30})$$

and acceleration as

$$\ddot{x}(k) = \frac{\dot{x}(k) - \dot{x}(k-1)}{T} \quad (\text{E.31})$$

where kT is replaced by k to simplify the notation.

Replacing (E.14) in (E.15) gives

$$\ddot{x}(k) = \frac{x(k) - 2x(k-1) + x(k-2)}{T^2} \quad (\text{E.32})$$

The initial velocity and acceleration used in (E.13) is estimated with (E.14) and (E.16).

$$x(k) = x(k-1) + \frac{x(k-1) - x(k-2)}{T} T + \frac{1}{2} \frac{x(k-1) - 2x(k-2) + x(k-3)}{T^2} T^2 \quad (\text{E.33})$$

which simplifies to

$$x(k) = 2.5x(k-1) - 2x(k-2) + 0.5x(k-3) \quad (\text{E.34})$$

Equation (E.18) is the desired estimate of the present position based on the previous three samples.

A simpler equation is derived for the second iteration of the algorithm when only two previous samples are available. Replacing (E.14) in (E.13) gives

$$x(k) = 2x(k-1) - x(k-2) \quad (\text{E.35})$$

APPENDIX F

Calibration

F.1 Derivation of equations describing a tilted CCD

This section contains details of the effect of a tilted CCD plane, discussed in Section 6.2.3. Figure F-1 shows a CCD plane which is slanted relative to the plane normal to the optical axis. The CCD is slanted at an angle of θ_{max} in the direction where $\phi = \phi_{max}$.

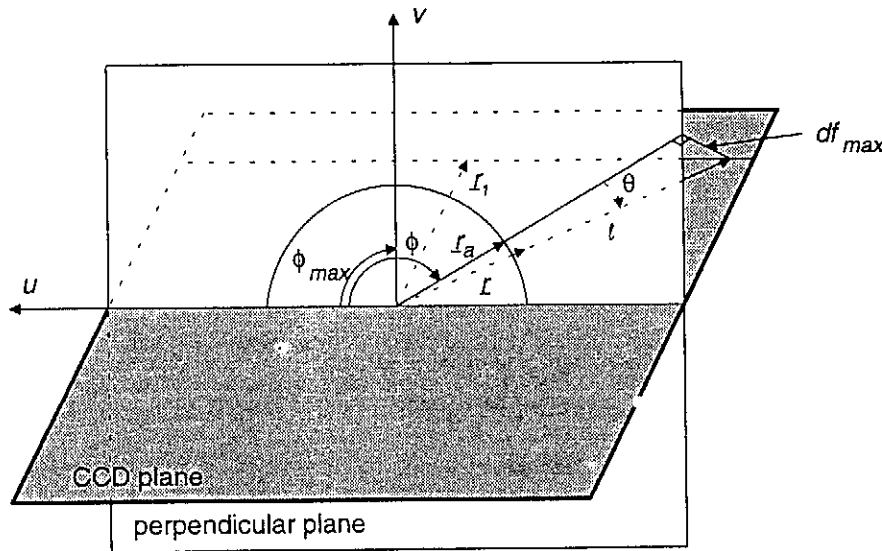


Figure F-1 Variation in radial distance on the CCD plane

The angle θ is measured between the vector r_l on the CCD plane and the undistorted vector r_a on the plane perpendicular to the optic axis.

$$\begin{aligned}
 \sin \theta &= \frac{df_{max}}{l} \\
 &= \frac{r_l \sin \theta_{max}}{l} \\
 &= \sin \theta_{max} \cos(\phi - \phi_{max})
 \end{aligned} \tag{F.1}$$

where

ϕ = the angle between the positive u axis and r_a
 $= \phi_{max}$ where $\theta = \theta_{max}$

so that

$$\theta = \text{asin}[\sin \theta_{\max} \cos(\phi - \phi_{\max})] \quad (\text{F.2})$$

In equation (6.11) the radius was shown to be

$$r = \frac{r_a}{\cos \theta} \frac{f_l}{f} + \Delta r_{\text{lens}} \quad (\text{F.3})$$

which is rewritten as

$$\begin{aligned} r &= \frac{r_a}{\cos \theta} \left(1 + \frac{df}{f} \right) + \Delta r_{\text{lens}} \\ &= \frac{r_a}{\cos \theta} \left(1 + \frac{r \sin \theta}{f} \right) + \Delta r_{\text{lens}} \end{aligned} \quad (\text{F.4})$$

Solving (F.4) for r gives

$$r = \frac{\frac{r_a}{\cos \theta} + \Delta r_{\text{lens}}}{1 - \frac{r_a}{f} \tan \theta} \quad (\text{F.5})$$

and solving for r_a gives

$$r_a = \frac{(r - \Delta r_{\text{lens}}) \cos \theta}{1 + \frac{r}{f} \sin \theta} \quad (\text{F.6})$$

F.2 Measurements for calibration

The first measurements were made to determine the position of intersection of the optical axis and the CCD. The same data was used to measure the tilt of the CCD plane. Two dots were imaged, one close to the optical centre and the other close to outside of the CCD. The set of dots were moved around until the centre one remained almost stationary in the image with rotations of the camera.

The camera was rotated and the dots imaged at 10° intervals i.e. 36 images. The positions of the centre dot on the CCD plane is shown in Figure F-2. It jumps around at random from one image to the next as the camera is accidentally twisted around the optical centre during the rotation sequence.

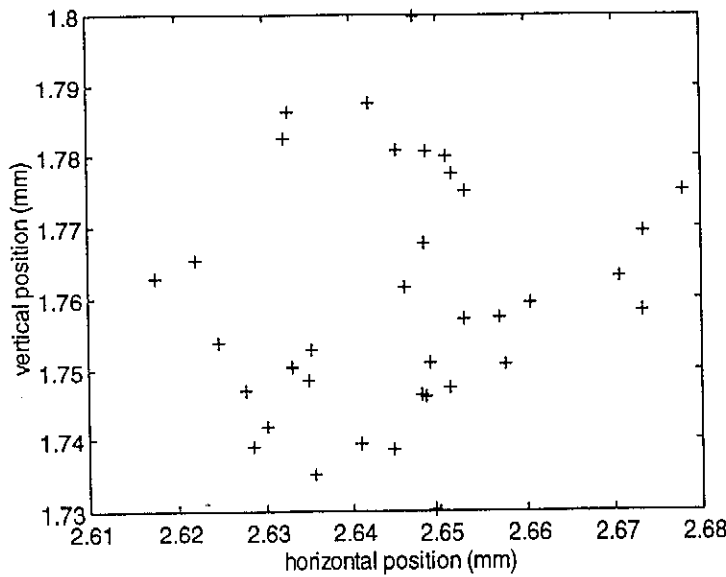


Figure F-2 Positions of a dot close to the optical centre when the camera is rotated around the optical centre.

The centre dot remains within an area of about 70 by $70\ \mu\text{m}$, or 12 by 12 pixels. The optical centre should be located within this area. It was estimated to be at the centre of the area i.e.

$$(horoptcen, vertoptcen) = (1.77, 2.64)\ \text{mm} \quad (\text{F.7})$$

The distance r_{12} measured between the centre and outer dot as function of the rotation angle ϕ (measured from the u axis toward the outer dot), is shown in Figure F-3.

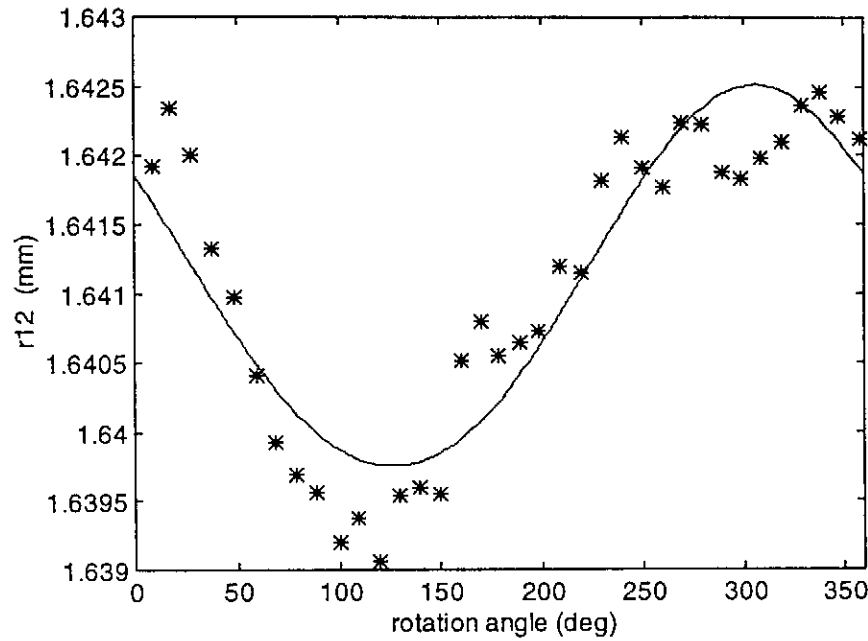


Figure F-3 Distance between the centre and outer dots, as a function of the angle measured towards the outer dot, ϕ : solid line indicates the curve given by equation (6.15) and the stars indicate the measured values

The 36 measured values are indicated as stars while the least squares fit of the mathematical approximation given by equation (6.15), is indicated as the solid line. The following parameters give the least squares fit:

$$\begin{aligned}\theta_{\max} &= 0.733^\circ \\ \phi_{\max} &= 306^\circ\end{aligned}\tag{F.8}$$

This means that the CCD is tilted relative to the normal plane by a maximum angle of 0.733° . This maximum slant occurs at an angle of 306° , measured from the positive u axis toward the positive v axis. These values were verified by imaging a set of dots at different radial distances. The slant parameters calculated for the shorter radii were all similar to those given in (F.8). The values estimated with the longest radius should be the most accurate and are used in the distortion model.

Having located the optical centre and corrected for the slant of the CCD, the lens aberrations are determined. This is done by measuring the angular distances from a star close to the image centre, to all other stars in the image. The true distance (obtained from the star catalogue) is subtracted from the measured distance. The angular error is converted to μm and plotted in Figure F-4.

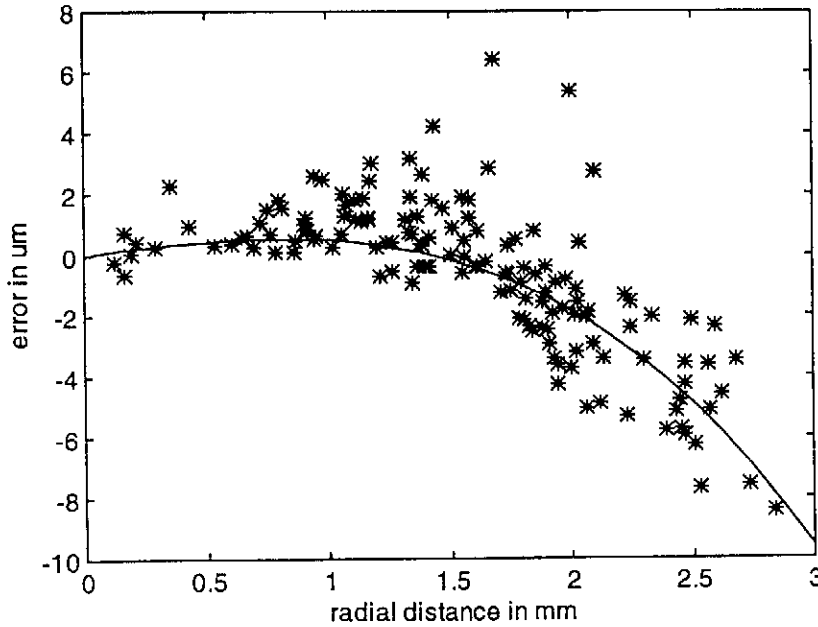


Figure F-4 Error in distances measured from a centre star in the image to all other stars: measured values indicated as stars and the third order approximation shown as a solid line

The maximum error is slightly larger than a pixel ($8.4 \mu m$ compared to the pixel size of $6.5 \mu m$). This means that the error in the distance between two stars on opposite sides of the CCD, may be as large as 2.5 to 3 pixels. The third order polynomial of equation (6.16) is used to approximate the data. This is shown as the solid line in Figure F-4. The coefficients used for the approximation are:

$$\begin{aligned} k_1 &= 1 \times 10^{-3} \\ k_2 &= -8 \times 10^{-5} \\ k_3 &= -4.4 \times 10^{-4} \end{aligned} \tag{F.9}$$

According to Derek Griffith [22] the strange spread in the data is a function of the optical design of the lens. Without the design specifications or data characterising the lens, it is hard to say which combination of aberrations is responsible. The best solution is to obtain the manufacturer's data on the lens or otherwise approximate the data with a high order polynomial. At the time of completing this thesis, no data on the lens had been obtained.

APPENDIX G

Drivers

The drivers used to set the integration time of the CCD and software for loading the star catalogue into the Flash RAM are discussed here.

G.1 Setting the integration time

The Sony *CXD1261* timing generator is used to generate all clock pulses for the CCD. The sequence used to set the integration time is summarized in the data sheets [20, p. 9]. Its implementation in Rowley Modula is discussed here.

An *HC273* latch is used to programme the three bits *ED0*, *ED1* and *ED2* which control the *CXD1261*. The latch is memory mapped at address 40000000H. The timing of the three signals used to set the integration time is shown in Figure G-1. Details of the restrictions on the pulse widths appear in the data sheet [20, p. 10]. These specifications can easily be met by the transputer and even the slower processors such as an *8031*.

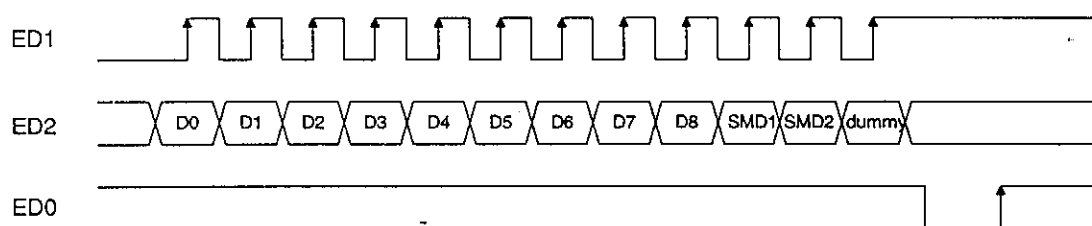


Figure G-1 Timing when setting the integration time

Eight data bits are written serially as *ED2*. The eight bits are the binary representation of the load value which determines the integration time. The next section discusses the relation between the load value, crystal frequency and integration time. Before each data bit has been written to *ED2*, *ED1* is taken low. *ED2* is set or reset according to the value of the data bit and then *ED1* is set. Once all 8 data bits have been written, *SMD1* and *SMD2* are written. These two variables determine the mode in which the integration time is set [20, p. 9]. For the star sensor *SMD1* = 1 and *SMD2* = 0 which selects the low speed shutter (long integration time).

Since other signals controlling the Flash RAM and CCD are programmed by the same latch, these values may not change when the integration time is set. If a bit needs to be programmed, the new value to be written to the latch is determined as follows:

- to set a bit it is logically OR'd with 1
- to reset a bit it is logically ANDed with 0
- all other bits are ANDed with 1, so that they remain unchanged

The value to be written to the latch is stored in *latch*, which is declared as a *BITSET* type and based in SRAM. Once the new value of *latch* has been determined, it is written to the *HC273* latch. The latch is declared as a *BITSET*, *controllatch*, based at 40000000H. Writing data to *controllatch* latches the data to *ED0*, *ED1*, *ED2*, *LADC* and *SFLASHVPP*. Reading *controllatch* reads the status of the CCD (*ADCACCESS*, *RDOUT* and *ADC*) and this is the reason for using the separate variable *latch* to store an accessible copy of the value present on the latch.

G.2 Relation between the load value, crystal frequency and integration time

The integration time measured in fields, for different load values, is calculated from equation¹ (G.1) and listed in Table G-1. The duration of a field is a function of the crystal used to drive the *CXD1261*. When using a 28.375 MHz crystal with the *CXD1261* in CCIR mode, the duration of a field is precisely 20 ms.

$$T_{int} = 2 \times (256 - L) \text{ FLD} \quad (\text{G.1})$$

Load value <i>L</i>	Integration time <i>T_{int}</i> (FLD)
254	4
253	6
:	:
1	510
0	512

Table G-1 Integration time measured in fields for different load values.

Equation (G.1) and the fact that a 28.375 MHz clock gives a 20 ms integration time, is used to calculate the integration time in seconds as

$$T_{int} = 2 \times (256 - L) \left(\frac{28.375}{f_{crystal}} \times 0.02 \right) s \quad (\text{G.2})$$

where

$f_{crystal}$ = crystal frequency in MHz

and solving for the crystal frequency

$$f_{crystal} = 1.135 \frac{(256 - L)}{T_{int}} \text{ MHz} \quad (\text{G.2})$$

¹ This is not the same as the equation listed in the *CXD1261* data sheet [20, p. 10], which is only true for the field readout mode. The star sensor uses the CCD in frame readout mode which provides double the resolution obtained from the field readout mode.

Section 8.6 showed that an analogue to digital conversion rate of about 7 to 8 *MHz* is desired. The readout frequency of pixels from the CCD is half the clock input frequency to the *CXD1261*. This means that a crystal of 14 to 16 *MHz* is required. Selecting a load value of 242 and an integration time of 1 *s*, the crystal frequency is solved as $f_{crystal} = 15.89 \text{ MHz}$. In other words, using a 15.89 *MHz* crystal and a load value of 242, the integration time will be set to 1 *s*. The duration of this period will change slightly as the crystal frequency drifts with variation in temperature. This change in frequency should be very small and similar to the drift in the ADCS crystal frequency. Consequently the synchronization between ADCS and the star sensor should not be affected.

G.3 Field and frame readout

As shown in Figure 1-1, the CCD's effective image area consists of 752 columns by 582 rows of pixels. The concept of interlacing two fields of an image was introduced by television in the 1930's. The entire frame (or image) is divided into two fields. The odd field consists of all the columns but only the odd numbered rows, while the even field only contains the even numbered rows. In a television, the odd field and even field are interlaced to form the image. The field rate, or rate at which fields are updated, is 50 Hz in the RSA. A new field is displayed every 20 ms, first the odd field followed by the even field during the next period. The frame rate, or rate at which the entire image is updated, is 25 Hz or at intervals of 40 ms. In other words, instead of updating the entire frame every 20 ms, only half of it is updated. The human eye can not distinguish the small shift or jump between the fields.

For this reason the Sony CCD, which is often used in video cameras, is clocked out as two separate fields. When used in the field readout mode, the odd field is clocked out first, followed by the even field during the next period. This conforms to the television standard. The CCD is, however, also capable of providing the entire frame (i.e. both fields or the entire effective image) after each integration period. For the star sensor, the *CXD1261* timing generator is programmed to operate in the frame readout mode. While imaging, the clocking pulse *XSG1* pulses low, which indicates that the odd field is about to be read out. Once all the odd numbered rows have been clocked out, *XSG2* pulses low before the even field is read out. After a single integration period, both fields are clocked out consecutively. The odd field appears first in the image memory, followed by the even field.

G.4 Loading the catalogue into Flash RAM

Flash RAM is a non-volatile read/write memory which means that it retains data when the supply voltage is turned off. To ensure that data is not altered during the transient conditions occurring with power up and power down, a sequence is used to programme the IC's. The sequences used to erase and programme the Flash RAM is described in detail in [21, p. 5.8 - 5.12]. The basic sequence was programmed in

Rowley Modula by M Prinsloo² for ADCS. This programme was adjusted for the star sensor. The 12V supply required to programme the Flash is turned on by setting the *SFLASHVPP* bit of the *HC273* latch. The bit is programmed as mentioned above for *ED0*, *ED1* or *ED2*.

The star catalogue contains the right ascension (RA), declination (Dec) and magnitude of the stars. These values are stored as REAL variables which means that each value requires 4 bytes. A problem occurs when a REAL variable needs to be written to the Flash RAM. The IC's can only be programmed one at time, but writing a REAL to memory means that 4 IC's are accessed simultaneously. The solution is to break the REAL value down into its 4 bytes and programme each one individually.

This may be done by determining the binary representation Modula uses to store a REAL. Knowing the representation, each variable is converted to its binary representation and then each of the four bytes are written individually to the Flash RAM. A much simpler method is used to achieve the same result. Rowley Modula allows variables to be declared at specific memory locations. To write a catalogue of 1000 REAL values to the Flash RAM, an array (*Cat*) of 1000 REAL values is declared at a base address in SRAM. The catalogue is downloaded from the link and stored as REAL numbers in this array. Another array (*tempCat*) of 4000 CARD_1's (which is a single byte variable type) is declared at the same base address in SRAM.

A base address (*FlashCatOffset*) is selected for the catalogue in the Flash RAM and an array (*tempFlashCat*) of 4000 CARD_1's is declared here. The bytes of *tempCat* are read one by one and programmed into the Flash Ram in the array *tempCatFlash*. Once all bytes have been programmed the programme can be terminated and the power turned off. The values remain in the Flash. When the catalogue needs to be accessed, an array of 1000 REAL's is declared at base address *FlashCatOffset*.

This method is implemented in the Rowley Modula programmes *SSCatFsh.mod* which appears in Appendix I.

² M Prinsloo is a masters engineering student who worked on the ADCS for Sunsat. The software appears on the Sunsat network drive: *J:\SW\Sat\Drv\ADCS\Flash\FlashTx.mod*

APPENDIX H

Selection of CCD's

Before performing the calculations, a brief introduction to the types of CCD's which are to be examined and their basic principles of operation is presented.

H.1 Types of CCD's

Two types of array CCD's are presently widely used. These are:

- **Interline transfer CCD's**

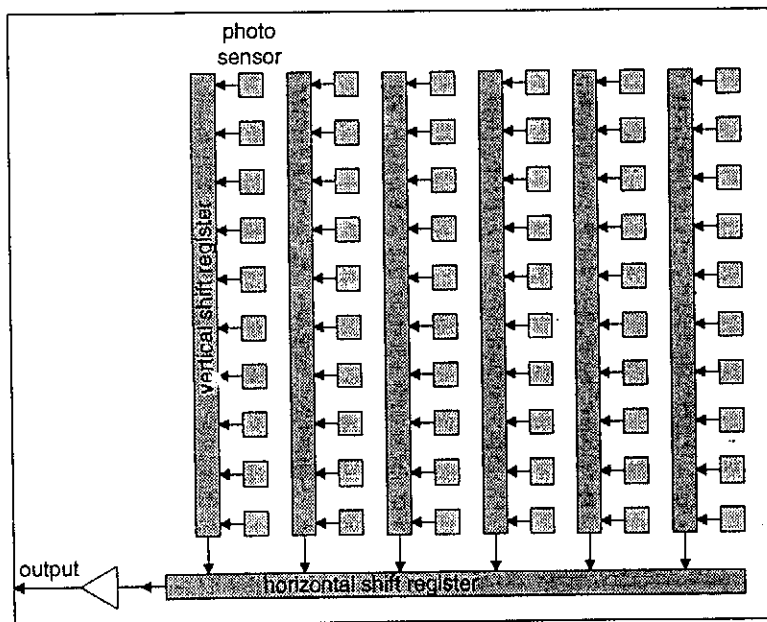


Figure H-1 Interline transfer CCD

The array consists of a series of vertical imaging sections and opaque vertical transfer registers. Charge is integrated in the photosensors of the imaging sections after which it is shifted horizontally to the transfer register. Once in the shielded vertical transfer register, the integrated charge is no longer affected by light. It is stored here before being clocked out. Each row is transferred to the horizontal shift register from where it is clocked out pixel by pixel. The Sony CCD's are of the interline type.

- **Frame transfer CCD's**

Charge is integrated in the imaging section of the CCD, after which it is very quickly shifted to the storage section. In the storage section the charge is shielded from light. As in the case of the interline device, rows are shifted down and clocked out pixel by pixel. EEV, Thomson and Phillips manufacture frame transfer CCD's.

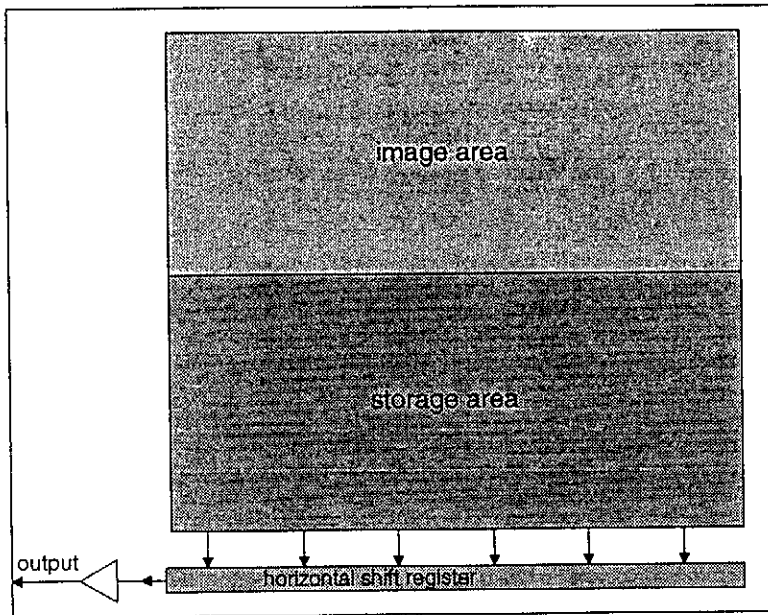


Figure H-2 Frame transfer CCD

H.2 Principles of operation

The CCD essentially consists of an array of MOS cells [24, p. 30]. As illustrated in Figure H-3, the device consists of an n-type silicon substrate, insulated from the metal electrode by an oxide layer. The electrodes are similar in function to the gate of a MOS transistor. When a positive potential is applied to an electrode, all electrons in the vicinity accumulate under it.

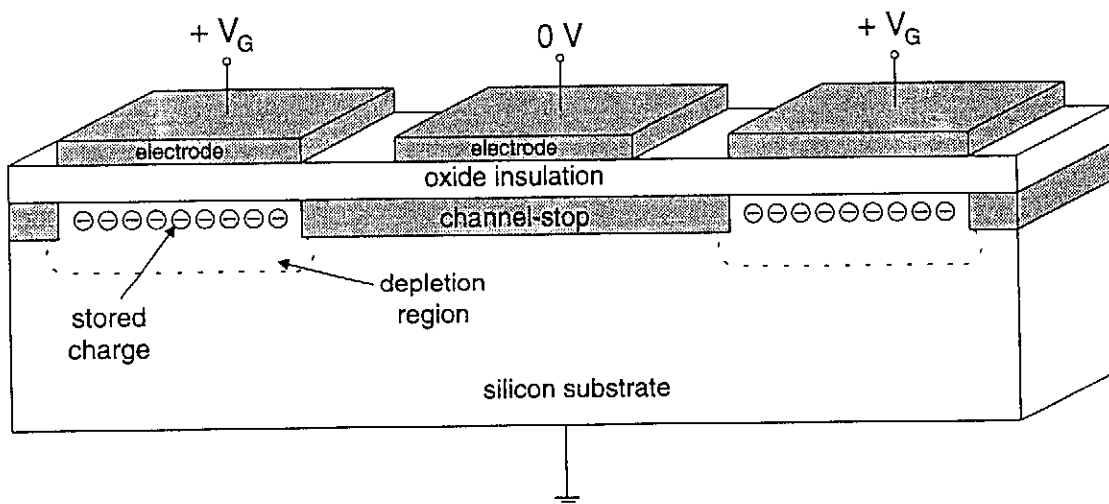


Figure H-3 Structure of MOS cells forming the CCD

Typically a positive potential is first applied to the electrode of the photosensor. All photogenerated electrons accumulate under this electrode. On completion of the

charge integration process (imaging), the voltage on the adjacent electrode is increased and that on the photosensor is decreased. Consequently charge is transferred to the more positive adjacent electrode. In a line transfer device, this adjacent cell would be in the opaque vertical shift register and in the case of the frame transfer device it is the next cell in the vertical column. In a similar fashion, charge is transferred through the entire CCD, from the photosensor to the output gate. Drive pulses to move the charge is generated by external circuitry.

The channel is the section under the electrode where the charge is accumulated. To prevent charge being transferred from one column to another, the channels are isolated by p-type silicon channel stops.

H.3 Dark current

Photogenerated charge accumulates in the photocells due to incident photons which generate electron-hole pairs. Together with the photogenerated charge, thermally generated electrons accumulate in the pixels. This thermally generated charge is known as the dark signal. Dark reference pixels are clocked out at the beginning and end of each line and this reference value is subtracted from the signal value to obtain the true photogenerated value. If dark signal remained constant throughout the CCD, it would not have been a problem when imaging. Due to nonuniformities in the device, the dark signal varies throughout the array. This variation in dark signal is known as the dark signal non uniformity (DSNU) or fixed pattern noise. The DSNU is the factor setting the lower limit on the dynamic range of presently manufactured CCD's.

The dark signal voltage is obtained by integrating the dark current over time. Over the range $+75^{\circ}$ to -60° [23, p. 18], the dark current follows the diode law, which means that non uniformities will follow the same pattern:

$$V_{DSNU}(T) = T_{int} C e^{-\frac{V_{BG}q}{2kT}} \quad (H.1)$$

with

$$\begin{aligned} T_{int} &= \text{integration time} \\ T &= \text{temperature in Kelvin} \\ V_{BG} &= 1.1 \text{ V} \\ q &= 1.6 \times 10^{-19} \text{ C} \\ k &= 1.38066 \times 10^{-23} \text{ J / K} \\ C &= \text{constant} \end{aligned}$$

When the voltage is known at one temperature, it can easily be calculated at any other temperature within the given range. The constant C is solved at the given temperature and replaced in the equation to calculate the signal at a new temperature. The relation between the voltages at two temperatures is given by:

$$V_{DSNU}(T_2) = V_{DSNU}(T_1) e^{6374 \left(\frac{1}{T_1} - \frac{1}{T_2} \right)} \quad (H.2)$$

H.4 Responsivity

The responsivity of the CCD is its ability to convert incident photons to voltage. It is derived from the information given in the data sheets for the Sony *ICX059*.

The luminance of the scene is quoted as [19, p. 10]

$$L_{scene} = 706 \text{ cd / m}^2 \quad \text{or} \quad \text{lux.sr}^{-1}.\text{m}^{-2} \quad (\text{H.3})$$

A Halogen source of colour temperature 3200 K was used. The peak radiance of this lamp is shown [9, p. 38 Figure 4-1] to be about 700 nm . The spectral irradiance of most stars peak between 500 and 1000 nm [9, p. 67 Figure 6-6]. Tests performed with this lamp give a good indication of the respons expected from stars.

When using a lens with F/number of 8, the illumination on the CCD is [23, p.18 equation 15]

$$\begin{aligned} E_{CCD} &= \frac{\pi B R_{scene} L_{scene}}{4 F_N^2} \\ &= 8.66 \text{ lux} \end{aligned} \quad (\text{H.4})$$

if the lens transmission factor B and scene reflectance R_{scene} are taken as unity.

The sensitivity of the CCD in volts is [19, p. 9]

$$S = V_s \frac{f_{int, test}}{f_{int}} \quad (\text{H.5})$$

The test was performed at 250 Hz [19, p. 10] and the value given for 50 Hz . Using the minimum sensitivity, the voltage is

$$V_s = 240 \text{ mV} \quad (\text{H.6})$$

Responsivity R is defined as the ratio of the generated voltage to the product of integration time and illumination on the CCD plane. Using (H.4) and (H.6) it is calculated as

$$\begin{aligned} R &= \frac{V_s}{E_{CCD} T_{int}} \\ &= \frac{0.24}{8.66 \frac{1}{50}} \\ &= 1.385 \text{ Vs}^{-1} \text{ lux}^{-1} \end{aligned} \quad (\text{H.7})$$

This is the minimum responsivity expected from the CCD. Assuming that it remains constant throughout the linear range of the CCD, (9.9) may be used to calculate the photogenerated voltage for any integration time and illumination. The typical sensitivity is $V_s = 300 \text{ mV}$, giving a typical responsivity of

$$\begin{aligned}
 R &= \frac{0.3}{8.66 \frac{1}{50}} \\
 &= 1.732 \text{ Vs}^{-1}\text{lux}^{-1}
 \end{aligned}
 \tag{H.8}$$

APPENDIX I

Software Listings

The software referred to throughout this document is listed in here. Due to the amount of software, it is impractical to print the listings and they are kept on disk instead. The following listings appear:

Modula files:

- *SSGrow.mod* Rowley Modula, *Region Growing* implementation module for star sensor transputer, discussed in Chapter 2
- *Match.mod* Rowley Modula, main *Matching* module discussed in Chapter 3
- *Track.mod* Rowley Modula, main *Tracking* module discussed in Chapter 4
- *Stats.mod* Topspeed Modula programme used to determine the number of stars expected in the FOV in Chapter 5
- *MakeCat.mod* Topspeed Modula programme used to generate sub catalogues, discussed in Chapter 5
- *DistTab.mod* Topspeed Modula programme which generates the distance table mentioned in Appendix D
- *SSCatFsh.mod* Rowley Modula programme used to download the catalogue to the Flash RAM
- *StarLib.mod* Rowley Modula library file containing all frequently used functions

Definition modules:

- *StarLib.def* definition file for the *StarLib* library
- *SSGrow.def* definition file for the Region Growing module *SSGrow.mod*

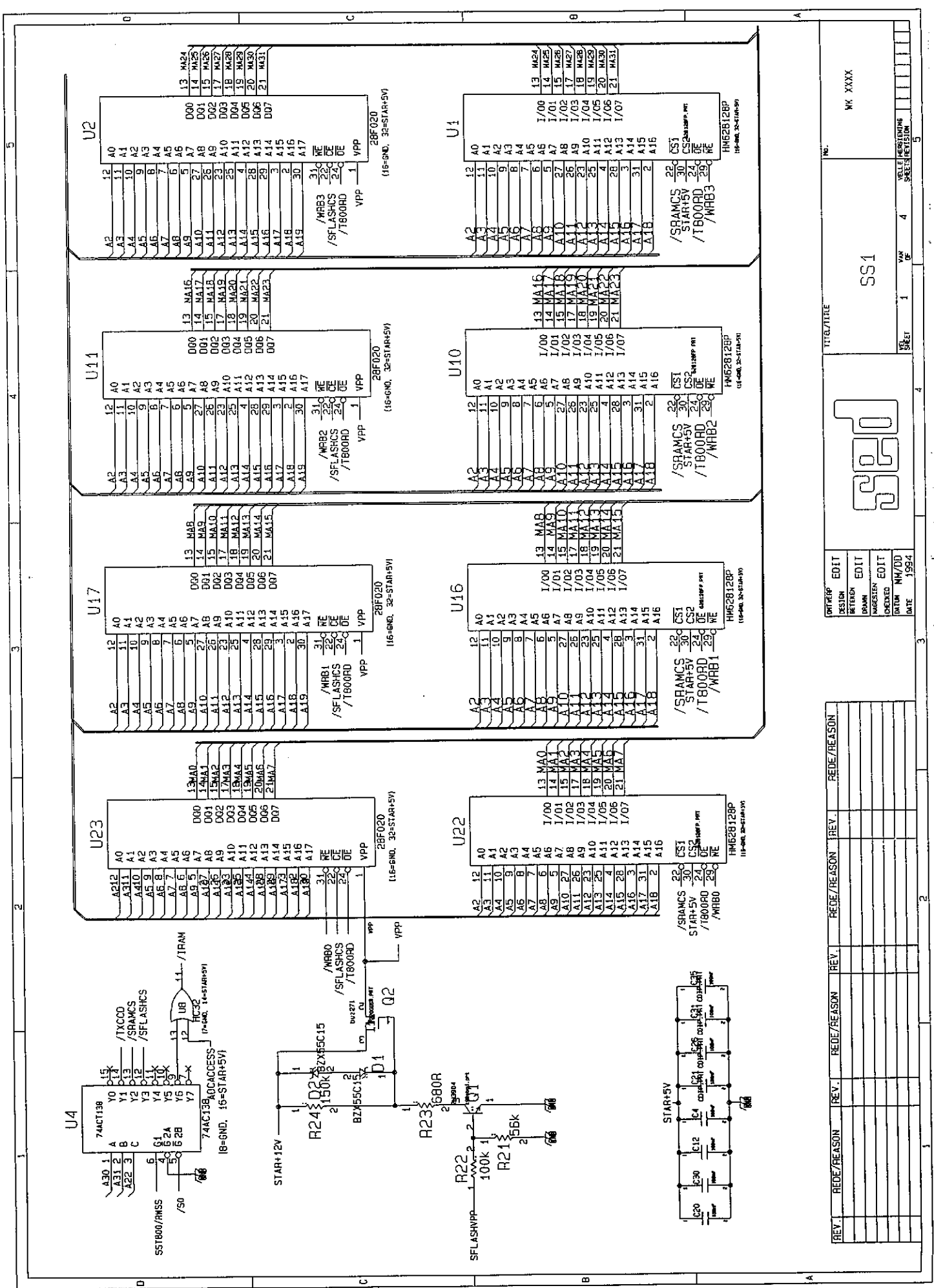
APPENDIX J

Schematics

This Appendix contains the PCAD schematics of the hardware designed for star sensor and discussed in Chapter 8.

The entire circuit consists of three printed circuit boards (PCB's):

- The design labelled SS1, SS2, SS3 and SS4 is the PCB containing the transputer, ADC and memory. This board is mounted as a piggyback on the ADCS board.
- The SONY PCB contains the timing generation and buffer IC's for the CCD. It is mounted in the star sensor box on the roof of the satellite.
- The PCB labelled CCD contains the CCD, sample and hold and amplifier. It is mounted directly behind the lens.



REV.	REDE/REASON	REV.	REDE/REASON	REV.	REDE/REASON

DATE	DATE	DATE	DATE	DATE	DATE
1994	1994	1994	1994	1994	1994

DATE	DATE	DATE	DATE	DATE	DATE
1994	1994	1994	1994	1994	1994

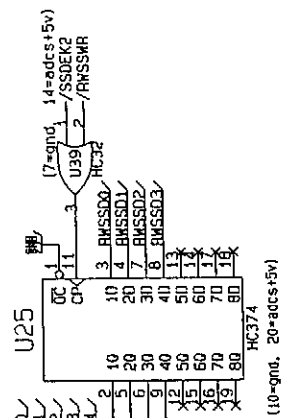
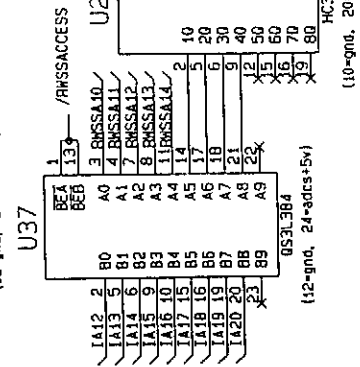
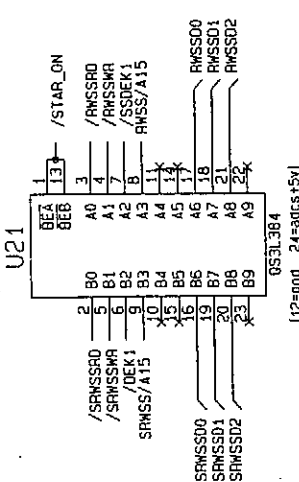
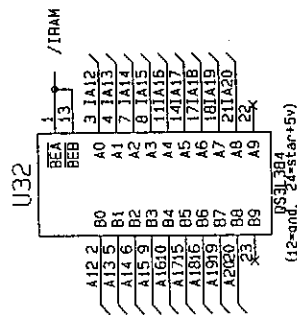
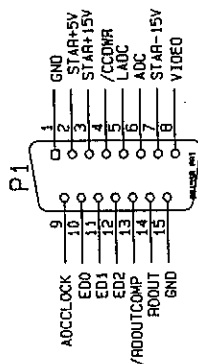
REV. REDE/REASON REV. REDE/REASON REV. REDE/REASON

DATE DATE DATE DATE DATE DATE

DATE DATE DATE DATE DATE DATE

DATE DATE DATE DATE DATE DATE

DATE DATE DATE DATE DATE DATE

[illegible]

ONTREAP	EDIT
DESIGN	
GETEXEN	EDIT
DEMAN	
SAGESIEN	EDIT
CHECKED	
DATUM	MM/DD
DATE	1994

554

TIME	Mo.
------	-----

XXXXX

VEHICLE	4	OF	4	LEAF
SHEET	4			
REMARKS				

