Durham E-Theses

Durham All-Sky Camera (DAC)

ZHANG, LIN

Durham All-Sky Camera (DAC)

Lin Zhang

Department of Physics
University of Durham

MSc Thesis

March 2009

Abstract

In this work, the prototype of a Durham all-sky camera (DAC) was designed and constructed for the purpose of monitoring the night sky of Durham. The DAC consisted of a Fujinon fish-eye lens coupled to a monochrome 640-by-480-pixel CCD camera (DMK-21BF04 model from the ImagingSource) housed in an 8-inch Perspex dome. After construction, the images were taken of the night sky in order to determine the properties of DAC. The field of view of the DAC was 185˚ at average scale of 0.2˚ per pixel.

The astrometric characteristics of the camera were investigated by measuring the relationship between star positions on the 3-D celestial hemisphere and their projected 2-D pixel positions on the DAC CCD images. The derived relationship (the mapping) of the stars onto the CCD images achieved an average uncertainty of 1 pixel. For the reverse process, the uncertainties were 0.2˚ in elevation and 0.7˚ in azimuth. The relationship was tested for the robustness and was found to be stable at the level of 1 pixel.

The photometric characteristics of the camera were studied by investigating how well the magnitude of a star could be measured by DAC. The results showed that, under the sky background condition in Durham, the camera was able to determine a 6th magnitude star at zenith within 1 magnitude uncertainty, but a 4th magnitude star within the similar uncertainty when the stars were at an elevation of 30˚. The brightness of the sky background of Durham was determined to be 18 magnitudes per square arcsecond. Subsequently, the comparison was drawn between the sky background in Durham and in Hawaii, thereby estimating that the accuracy of measuring a 6th magnitude star would be about 7 times better if the camera was deployed under the darker sky background of Hawaii.

Declaration

I declare that all the contents, including the texts, figures, tables and results, presented on this thesis are original except the ones being referenced. It states the majority of work I did for making the prototype of the Durham All-Sky Camera (DAC) during the year when I was studying in the Department of Physics at Durham University.

Lin Zhang

Acknowledgments

Contents

Chapter 1

Introduction

1.1   All-Sky Camera

An all-sky camera, commonly abbreviated to ASC, is a camera, which has a field of view of about 180° and is capable of presenting an image of the whole sky on a single frame, as indicated by its name. The images, or the all-sky images as being normally called, taken by such a camera has a round shape, with the centre coinciding at zenith and the edge imaging the horizon.

Two optical designs are commonly used for all-sky cameras. One uses a fish-eye lens attached to a camera and the other has a convex mirror with the camera suspended above looking downwards. Both types can achieve a field of view of 180°. The comparison drawn between the two shows that the mirror configuration losses the sky at zenith since it is blocked by the camera; however, the light losses are relative smaller and costs less than the fish-eye lens configuration. For both configurations, a CCD sensor is very often used as the imager on today's all-sky cameras.

The primary function of an all-sky camera is being as a sky monitor as a result of the special images it can produce. The meteorologists may like to know the weather condition without actually going outside themselves, or the star observers may feel unpleasant when frequently getting their eyes adapted to the dark to identify the stars. These problems can be solved by implementation of an all-sky camera.

An all-sky camera is generally capable of presenting the live images of the sky while keeping the records of the sky views from the past. It makes the camera extremely useful when people want to trace the phenomena, like the meteor showers or aurora, or to track the objects, like a star or satellite. Furthermore, it reduces the difficulty of quantitative analysis of such a sky phenomenon and makes the measurement easier to carry out. An all-sky camera has been widely used in many institutes and observatories in the world.

Here are some examples. An all-sky camera was implemented to monitor the meteor trails at the Cloudbait Observatory in the US①. In Korea, people used it to observe the mesospheric waves②. At Michigan Technological University, it was used to detect the bright optical transient③. In Finnish Meteorological Institute, three all-sky cameras were used to study the auroral electrodynamics at mesoscale④.

1.2　About The Work of Making An All-Sky Camera

In the previous section, we have introduced the background of an all-sky camera. Based on the functions that such a camera can perform, my work was carried out to construct a prototype of the all-sky camera (Chapter 2) and to investigate its performance after the real-time operation (Chapter 3 for the real-time operation).

Two parts of analysis were carried out for the investigation of the performance of the camera. The first analysis was done to achieve the mapping between the 3-D celestial hemisphere, i.e. the sky dome, and its 2-D image, thereby empirically establishing a numerical relation (Chapter 4). The second part of the analysis was to investigate how well the brightness of the stars could be measured by the camera under the sky background condition in Durham (Chapter 5). In the end, the possible methods of making an improvement on the camera's performance were discussed (Chapter 6).

Since it was made in Durham, the prototype was named DAC to stand for Durham all-sky camera.

---

①. More information is available at the Cloudbait Observatory website at http://www.cloudbait.com/projeccts/allskycamera.html.

②. Details can be found from *Observation of Mesospheric Waves with An All-Sky Camera in Korean Peninsula* by J-K. Chung, Y. H. Kim and Y-I. Won. The abstract is available at http://adsabs.harvard.edu/abs/2003AdSpR..32..825C.

③. The relevant paper is *Transient Detection Using Panoramic All-Sky Cameras* by Shamir and Lior. A copy of the abstract can be found out at http://www.ingentaconnect.com.

④. The detailed information is available on the website of the Finnish Meteorological Institute at http://www.geo.fmi.fi/MIRACLE/ASC.

Chapter 2

Design and assembly of Durham all-sky camera

2.1 Type of Design

Taking account of the purpose of utilization, price, ease of construction and flexibility of modification, the prototype of the Durham all-sky camera (DAC) was designed to be a CCD camera implementing a fish-eye lens. It avoids the obstacle to viewing the central sky, in contrast to the mirror configuration (Chapter 1), while the central sky is darker and more stars are detectable than the horizon in a city where the sky background is contaminated by the streetlights.

2.2 Lens and Camera

2.2.1 General information

The DAC lens was the FUJINON YV2.2✕1.4A-SA2 1/3-inch vari-focal lens. The field of view was set to the maximum, i.e. 185˚, after the construction of DAC. The lens contains an inner iris driven by the switch of the connected camera, so it keeps closed when the camera is set off, thereby protecting the camera sensor from exposure to the sunlight when not being put into operation in the daytime.

The DAC camera was the DMK–21BF04 from IMAGINGSOURCE. It implements a Sony CCD sensor (ICX098BL), which is ¼-inch in format and has 640-by-480-pixel spatial resolution. The operating temperature indicated on the manual is between -5˚C and 45˚C, which is suitable for DAC to operate under the environmental condition in Durham. The camera is able to produce live monochrome images progressively.

2.2.2 Operating parameters

The camera has its own operating software, IC Capture 2.0, installed on a computer, by which the operating parameters of the camera were set. The parameters consist of the camera's frame rate, exposure time, gain, brightness and gamma.

The frame rate determines how fast an image is made and the exposure time is the time when the sensor is active during each frame. The camera in DAC has the exposure time ranging from 1/10,000 to 30 seconds. The gain and brightness

parameters are able to intensify the image contrast by multiplying every pixel by a constant and by adding a constant to all pixels, respectively, as indicated in the program instruction; however, none of them has a unit shown on the program or operating instruction. Parameter gamma has nothing to do with the quality of the image but it improves the viewing quality of a display by neutralising the nonlinearity of the monitor, as explained on the instruction.

2.2.3 Investigation in dark current

Dark current is a characteristic feature of a CCD camera. It affects the image quality and determines the performance of the sensor. Therefore, the images of dark current of the camera of DAC were investigated in order to obtain a general idea about quality of the camera we used.

In Fig. 2.1 shows a 3-D plot of the dark frame, which was made by covering the lens with a cap so as to avoid any light coming into the camera. The exposure time of the dark frame was set to be 14 seconds while other parameters were set to zero. The spikes (Fig. 2.1) indicate the pixels that had signal response. The histogram of the pixel values of the frame is shown in Fig. 2.2, on which the peak falls on 0 ADU. The pixels of the value 0 ADU indicates they were not affected by the dark current. Such pixels were found to have a normalised frequency of 0.995, i.e. being 99.5% of total number of pixels on the sensor. It implies the sensor's performance was reasonable since we expect the majority of the pixels not to respond under the dark-frame circumstances. From Fig. 2.1, we can see a hot corner of the dark current, being at the bottom left of the frame.
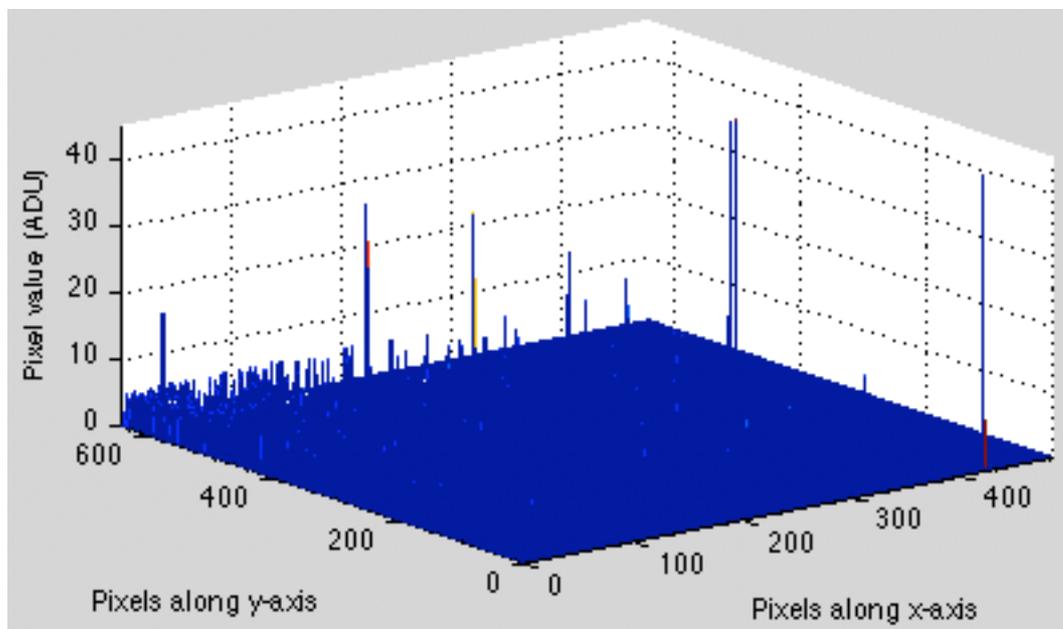
Fig. 2.1   The 3-dimentional plot of a dark frame. The spikes show the image position of the pixels that had signal response. A high spike corresponds to a strong signal.

4
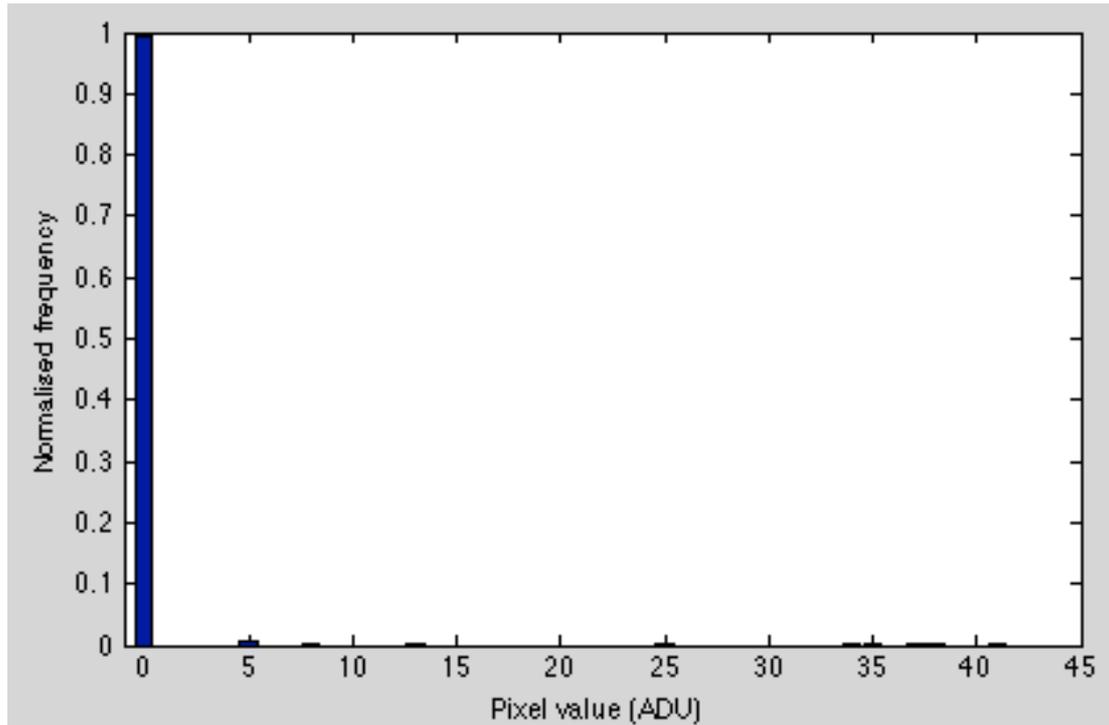
Fig. 2.2   The histogram of the pixel values of the dark frame. The normalised frequency of the pixels of 0 ADU is about 0.9951.

The subsequent investigation was done into how the number of the pixels of 0 ADU changes with the camera's operating time, exposure time, gain and brightness parameters, respectively. In Fig. 2.3, the vertical axis, labelled normalised frequency of 0 ADU, indicates the percentage of the total number of pixels being 0 ADU, but the percentage was expressed in the normalised frequency. In other words, every point in a graph of Fig. 2.3 corresponds to a histogram like Fig. 2.2, and only the normalised frequency of the pixels of 0 ADU was considered. The result shows, during one-hour operation, the normalised frequency of 0 ADU pixels fluctuated slightly around 0.995 (Fig. 2.3(a)), which indicates the number of the pixels unaffected by the dark currents did not change with the operating time, thereby implying the activity of the dark currents on the sensor was independent of the camera's operating time.

In Fig. 2.3(b), it shows the number of the pixels unaffected by the dark currents fell exponentially with the increase of the exposure time of the camera. It is reasonable since the sensor became hotter and hotter during a long exposure, thus triggering more and more dark currents. As a result, an exposure time below 20 seconds was considered to be adequate to use.

The result of how the dark currents fluctuating with the gain parameter (Fig. 2.3(c)) shows the increase of the gain stimulated the growth of the dark currents so that the normalised frequency of the unaffected pixels dropped down to about 0.1 when gain was maximum. Since no instruction of the gain parameter was found, it is unclear what the gain really indicates; however, according to the result, the gain seems to behave similarly to the intensifier gain①, which produces noise image but allows the exposure time to be short.

---

①. More information is available on the website of Space Physics Research Group, University of California, Berkeley, at http://sprg.ssl.berkeley.edu/atmos/sondre.html.
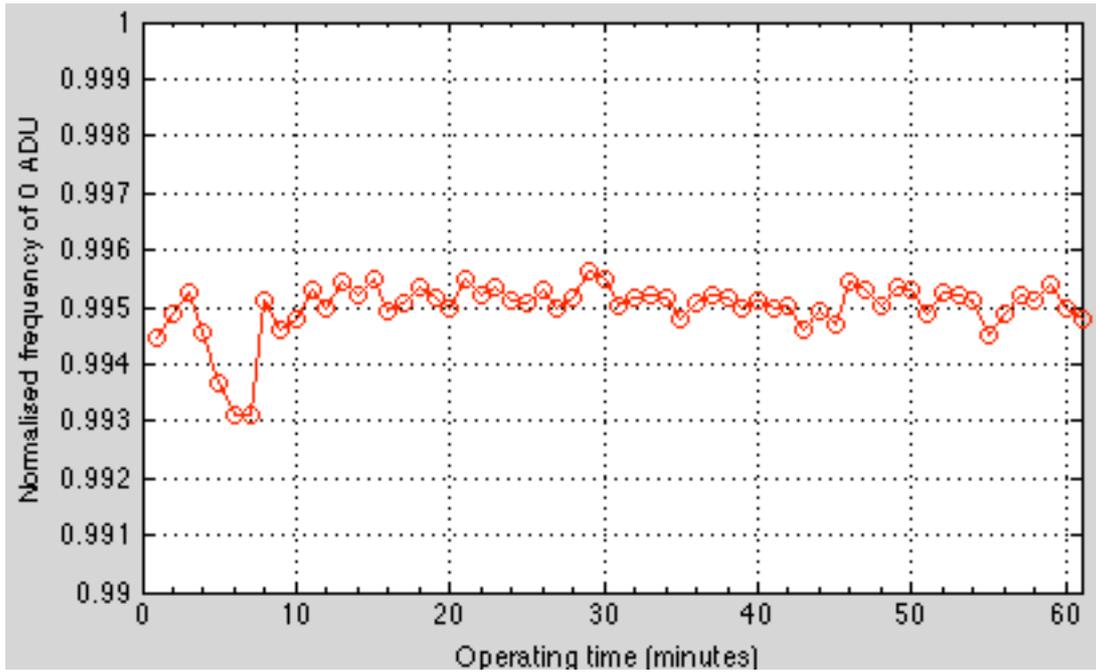
(a)
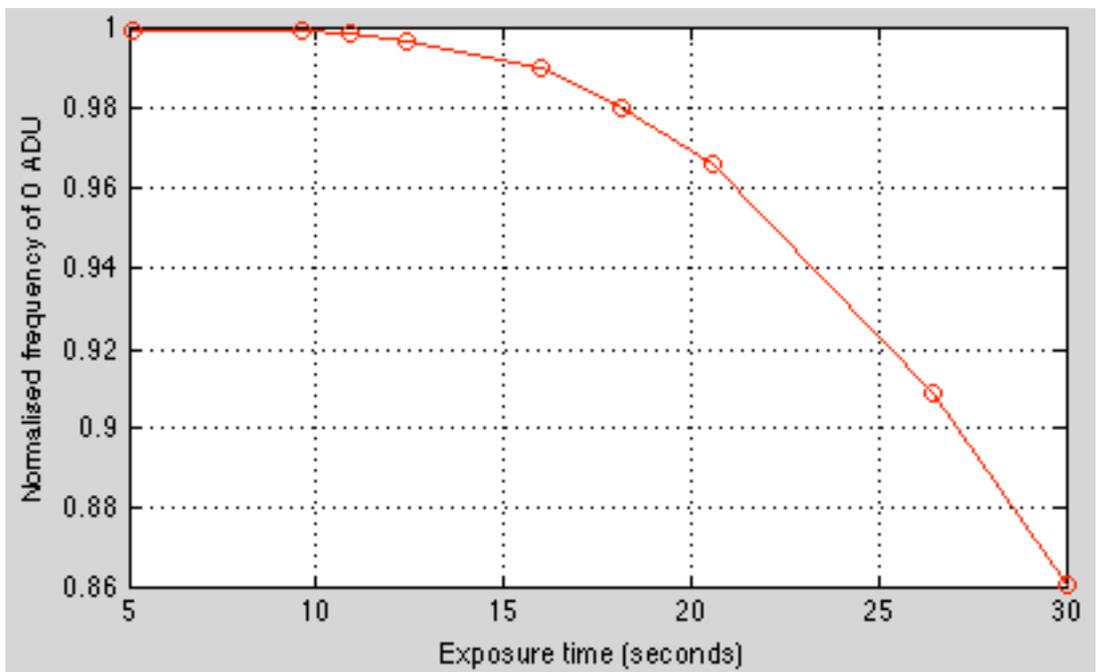
(b)
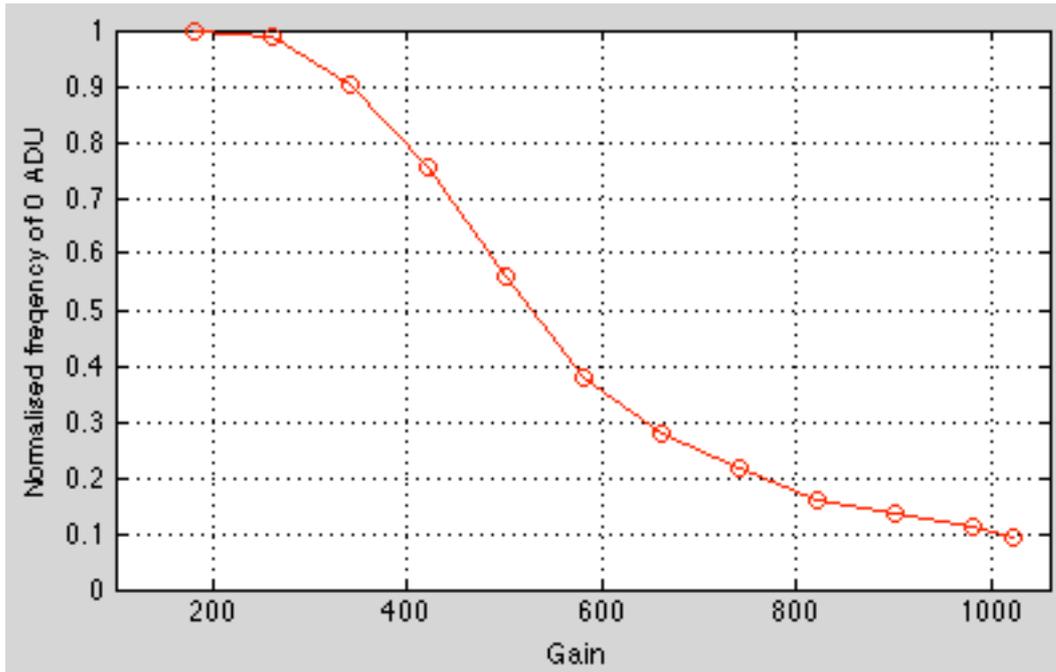
(c)

Fig. 2.3   The normalised frequency of 0 ADU pixels against (a) the operating time of the camera, (b) the exposure time and (c) the gain parameter, respectively. These graphs show how the number of the unaffected pixels by the dark currents changed with each setting parameter.

Similar to gain, the exact indication of the brightness parameter on the operating software is also uncertain. In Fig. 2.4 shows the analysed result, on which the vertical axis represents the pixels value that has the highest normalised frequency, i.e. the counts of the majority of the pixels. From Fig. 2.4, we can see that the counts of the majority of the pixels went up with increase of the brightness parameter, which implies the activity of the dark currents was intensified. Therefore, we conclude that the magnitude of the dark currents increases with the increase of the brightness parameter.
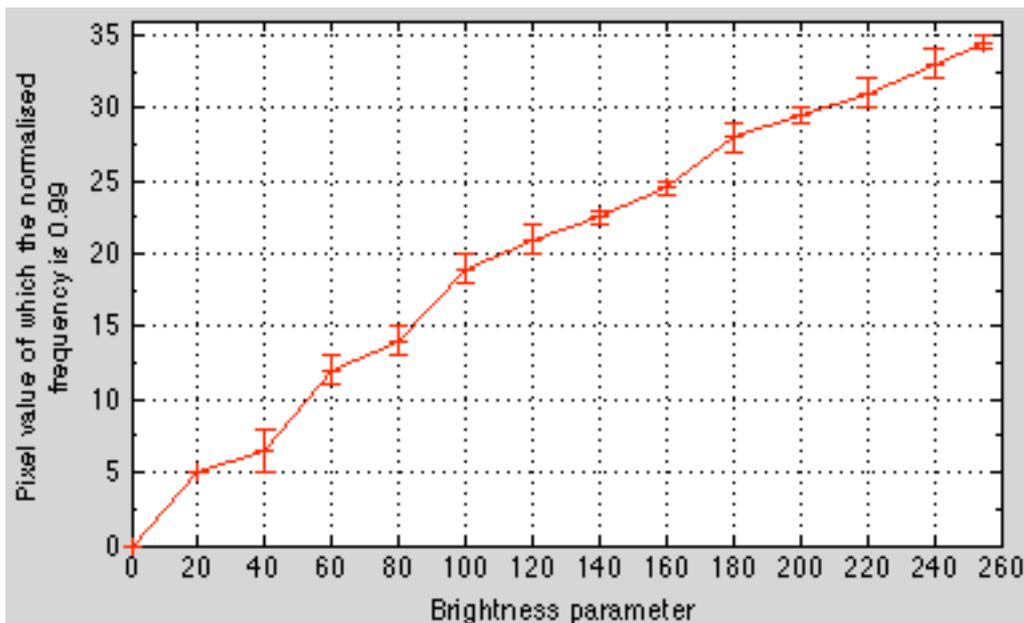
Fig. 2.4   The increase of the dark currents with respect to the brightness parameter. The vertical axis shows the pixel value of which the normalised frequency of the number of pixels is 0.99.

7

Along with the investigation in the dark currents, 7 hot pixels were found. Their pixel values rose sharply with the increase of the exposure time, gain or brightness parameter. The image positions of the hot pixels were identified to be (368, 317), (422, 11), (56, 30), (382, 332), (87, 402), (224, 471) and (13, 595).

In conclusion, for a better image quality, the settings of the exposure time, gain and brightness parameter should not exceed or be far beyond 20 seconds, 400 and 120, respectively. In order to discount the pixel counts of the dark currents, a dark frame should be subtracted from the images.

2.3   Housing and Assembly

The camera was housed inside an aluminium case on top of which was a Perspex dome with a diameter of 8 inches. The case was earthed and a circular rubber mat was placed between the dome and case to seal the connection between the two rigid bodies so that the water droplets or dusts cannot easily get through.

The original design of the camera case was attached in Appendix A. Although several other designs were made, only this one was accepted on account of ease of manufacture. The case was designed with extra inner space for the add-ons to fit in since other devices may be needed for the improvement of the camera's performance.

The silver box inside the case (Fig. 2.5(a)) was a control box, which was used to deliver the signals between the lens, camera and a control site. The signals were transmitted to the control set via a FireWire cable. After construction, DAC was fixed on the roof of the Physics Department (Fig.2.5 (b)).

(a)   (b)

Fig. 2.5   The pictures of the Durham all-sky camera. In (a), it shows the components inside the case. In (b) shows what the camera looks like after construction.

Chapter 3

General implementation

3.1 Imaging by DAC

After camera construction, a set of all-sky images were taken. Three of them are shown in Fig. 3.1 below. The white structures, at the bottom right and left of the images, are the west and east domes, respectively. The structure between them along the edge is a corridor with the entrance to the roof. From these images, we can see some parts of an image, being the side view of the sky, were lost. It was the result of the size of the image focal point being larger than the CCD sensor. As described in Chapter 2, the camera implements a 1/3-inch vari-focal lens but a ¼ -inch sensor. However, it does not affect much on an observation. Because the night sky near the horizon, as shown on the edges of an image, is illuminated brightly by the streetlights. At a long exposure, the pixels on these areas saturate thereby revealing little information about the sky views.

It was found that a long exposure, about 14 or 15 seconds, was essential for making a good image when the sky is clear (Fig. 3.1(a)). A trade-off is that this long exposure allows more stars at zenith to be detected at the expense of saturating the pixels around edge. A short exposure time, 12 seconds, was necessary for an observation at a cloudy night (Fig. 3.1(b)). The clouds reflect streetlights, thus appearing to be very bright on an image. The even shorter exposure, about 9 or 10 seconds, was needed for making an image on a night when the moon was in the night sky (Fig. 3.1(c)). This shorter exposure was essential to reduce the ghosting and scattered light of the moon from dominating the image. The gain and brightness parameter were set at 232 and 0, respectively, when taking the images.

N

(a)

E   W

(b)

S
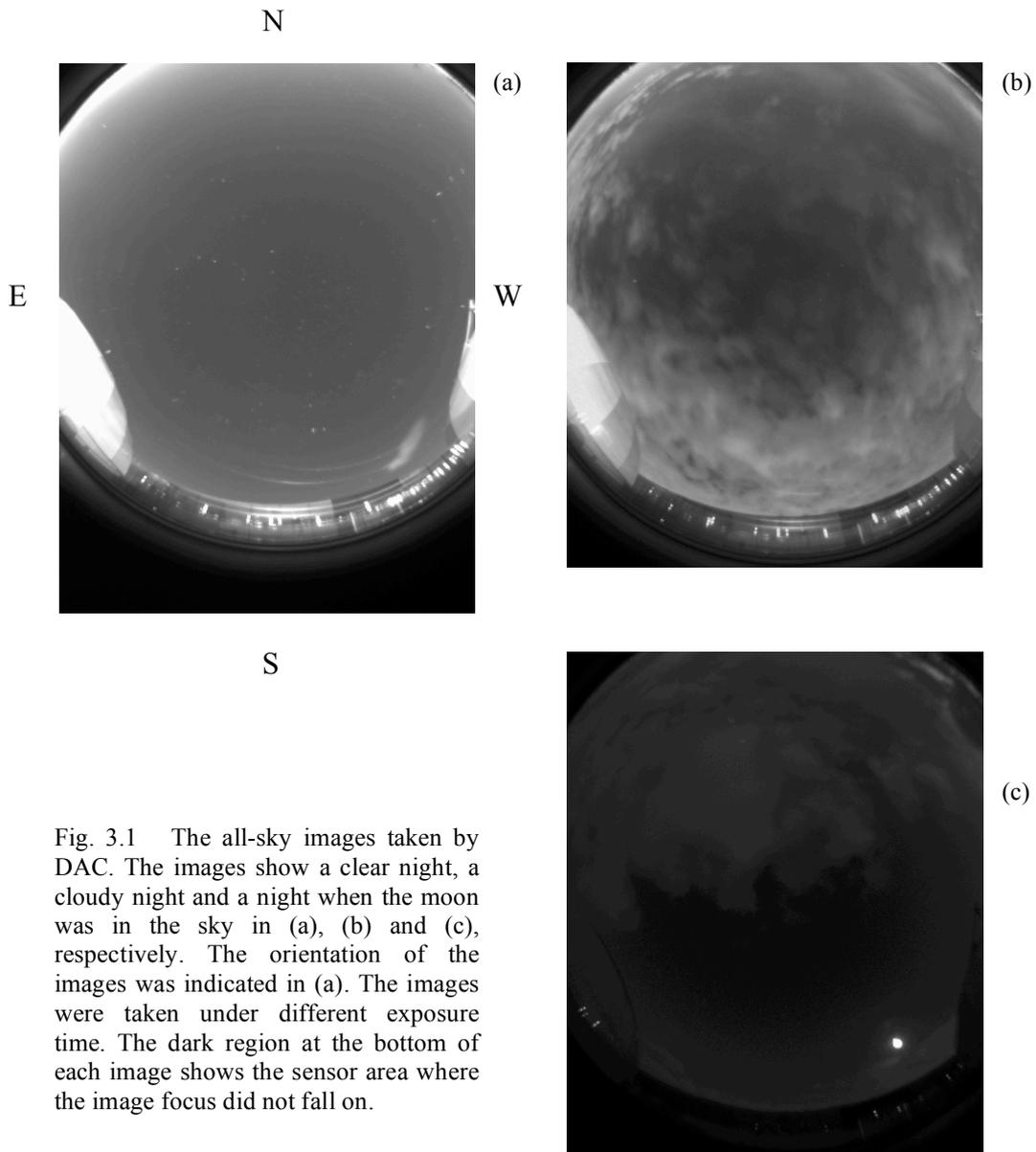
Fig. 3.1   The all-sky images taken by
DAC. The images show a clear night, a
cloudy night and a night when the moon
was in the sky in (a), (b) and (c),
respectively. The orientation of the
images was indicated in (a). The images
were taken under different exposure
time. The dark region at the bottom of
each image shows the sensor area where
the image focus did not fall on.

(c)

3.2   Image Features and Factors Causing Defects

In Fig. 3.2 shows the enlarged images of several stars from the same frame. The position of every star in the figure was arranged according to their relative positions on the frame. It implies, as a result of the fish-eye view, the star images were distorted along the radial direction, and the distortion became large when a star was further away from the zenith.
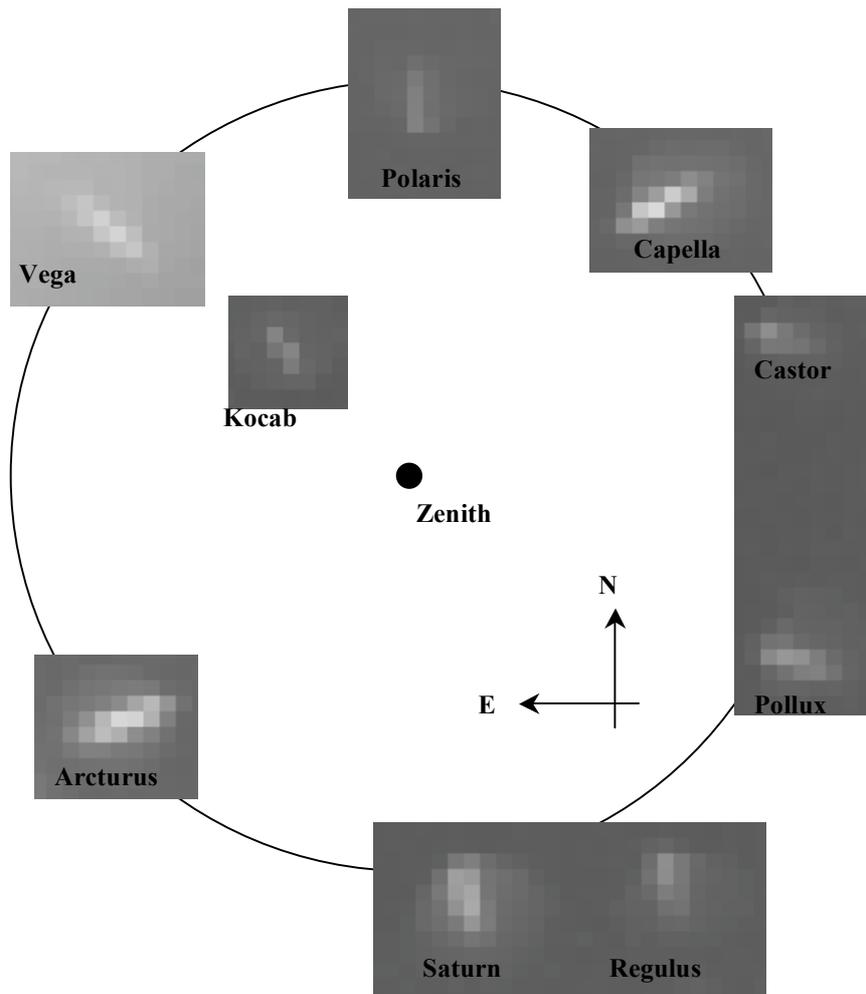
Fig. 3.2   The enlarged image of stars, of the same frame, distorted as a result of the fish-eye view. It seems that each point star was stretched along the radial line through it. The collection of stars shown above was from the frame taken at 22.00 on 4 April 2008.

The image of a hot pixel or dust on a single frame is very likely to be confused with the image of a star (Fig. 3.3 & 3.4). However, they do not move, like the stars, from frames to frames. Consequently, we can distinguish between a star and a hot pixel or dust by checking their positions on the subsequent frames. Another way to distinguish them is by the feature of the star images. A star image occupies several pixels with the brightest one being around the middle; by contrast, a hot pixel usually appears to be a single bright pixel and the image of dust has an irregular shape. However, it may cause confusion when a star was partially blocked by the dusts or when its image moves near off a hot pixel.

Apart from the hot pixels and dusts, it was found that the water vapour condensation is another factor causing the image imperfection. After several hours' operation in a winter night, the vapour was condensed on the dome, thus partly blocking the sky, scattering the streetlights and making the images brighter.
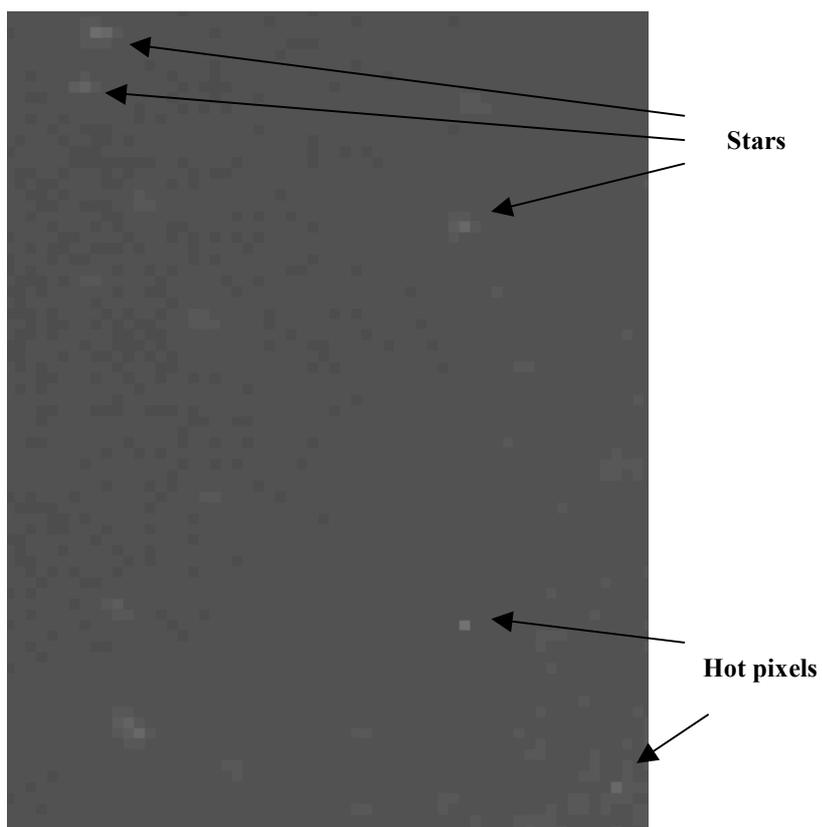
Fig. 3.3   The image of stars and hot pixels. The two hot pixels shown in the figure were at the pixel position of (368, 371) and (382, 332), respectively.
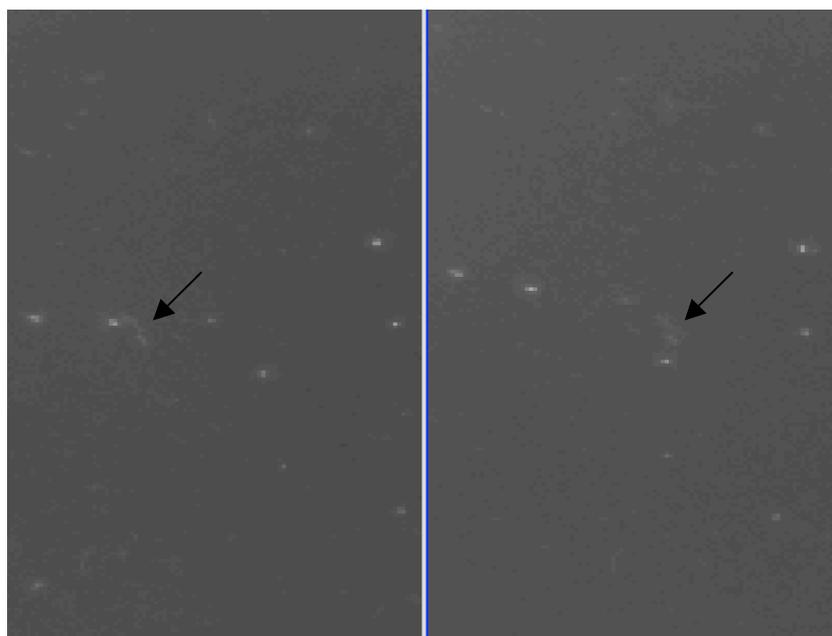
Fig. 3.4   The enlarged image of Plough. The arrows point to the image of the same piece of dust on the dome or lens. The two images were made about 1-hour apart. The image on the right was earlier in time than the left one.

12

3.3 Conclusions

In this Chapter, we have shown some all-sky images produced by DAC and described the features of the images while discussed the factors that may affect the image quality. We draw the conclusion that the camera performed well and is capable of monitoring the night sky of Durham despite slight image loss at horizon.

In comparison with the images taken by a different all-sky camera implemented at a different observing site, like the one from the Institute for Astronomy in the University of Hawaii[1], the DAC images presented less stars. However, it does not necessarily mean that DAC has a relatively lower resolution than the all-sky camera implemented at Hawaii. The depth of the sky or the number of the stars that can be detected is subject to the brightness of sky background. In the following chapters, we will show the astrometry and photometry analysis of the DAC images, exploring the capabilities and function of DAC and demonstrating the reason why less stars were seen by DAC.

---

[1]. Their all-sky images can be seen at http://uh22data2.ifa.hawaii.edu/public/allsky.

Chapter 4

Astrometry analysis

4.1 Purpose and Introduction

In the astrometry analysis, the mapping of the points between the 3-D celestial hemisphere and a 2-D image was carried out. It aimed to predict the image position of a star mathematically when its sky position was known, and vice versa. As a result, a star can be quickly identified on an image and we are able to predict its image position at any time, thus tracking the star whenever necessary. The mapping was considered as an essential function of DAC and a primary step to the following analysis. In this chapter, we will introduce how the mapping was done empirically and show the results in the end.

4.2 Theory and Method

Depending on the applied coordinates, the position of a point (artificial) on the dome of the sky can be measured in terms of RA/Dec or Az/El. The former change their values with time as the earth rotates but the latter are time-independent (RA/Dec are fixed for a particular star but the artificial points on the sky dome were considered here). Since the pixel position of every point, in terms of (x, y), on an image is also time-independent, it is possible to establish a mathematical relationship between the values of Az/El and (x, y), thereby mapping the points from the 3-D celestial hemisphere to the 2-D images.

Among all the image points that need to be mapped, i.e. those showing the view of the sky, two points are special and can be used as the reference. One is the pixel position of the projection of zenith, which indicates the position of the centre of an all-sky image; the other is the pixel position of the projection of a zero-azimuth point. Theoretically, the projection of the zero azimuth is a line formed by a set of zero-azimuth points; however, in practice, the image points are represented by the pixel positions, i.e. the integer points, and consequently only the point that gave the best accuracy of the position of a zero azimuth was considered. When the pixel positions of such two special points are found, the line of the projection of the zero azimuth can be determined by joining the two points together. Subsequently, the Cartesian coordinates (Fig. 4.1) can be set up for the relevant calculations.

The pixel position of zenith can be determined empirically by investigating the arithmetic relation between the zenith distance of the points in the sky and the pixel distance of their corresponding image positions. The zenith distance, Z, of a point is

defined to be the smallest curvature of the line joining this point with the point of zenith on the celestial hemisphere. It follows

$$Z = \frac{\pi}{2} - El \tag{4.1}$$

where El is the elevation. The pixel distance, P, of the corresponding image point is, therefore, the length of the projection of Z and is measured in the unit of pixels. The mathematical expression is

$$P = \sqrt{(x - X_Z)^2 + (y - Y_Z)^2} \tag{4.2}$$

where (x, y) and (X_Z, Y_Z) are the pixel positions of the point and the zenith, respectively. As shown in Eq. (4.2), the pixel distance of a point is dependent upon the choice of the image position of zenith. When such a position is properly selected, the pixel distance and the zenith distance of those points of high elevations can be approximated to have linear relations (Fig. 4.2); however, the relation is at best approximation of quadratics or higher power when the points of low elevations are added. Such a mathematical relation between the zenith and pixel distances was established, as the first step of mapping, by assuming the image position of zenith. The image position of zenith is, therefore, best represented by the pixel to which the relationship between the zenith and pixel distances holds under the smallest uncertainties of the pixel distance, δP, and the zenith distance, δZ. As a result, we can find not only the image position of zenith, but also an equation to convert between the zenith and pixel distances.
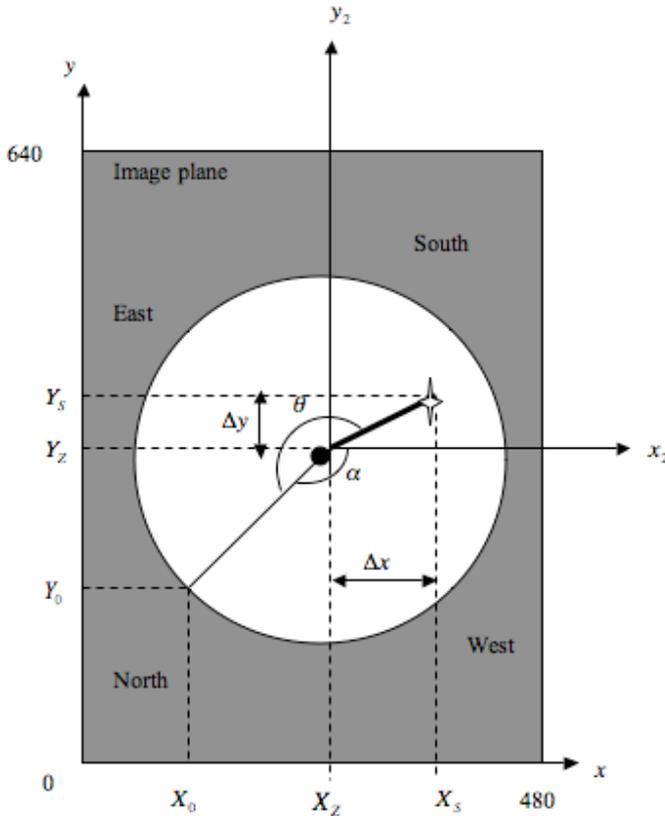
Fig. 4.1 A schematic representation of an all-sky image. The white disk at the centre represents the image of the sky. The dark dot at the centre of the disk represents the pixel position of the zenith, (X_Z, Y_Z). The star-shaped symbol on the right hand inside of the disk represents the image of a star which is at the pixel position of (X_S, Y_S). The bold solid line joining the pixel position of zenith with the star image indicates the pixel distance of the image point. The point, (X_0, Y_0), represents the pixel position of a zero-azimuth point. The angle α is the between the line of the projection of the zero-azimuth points and positive x_2–axis. The angle θ is the calculated azimuth angle of the image point. The distances Δx and Δy represent the components of the pixel distance along x- and y- axis, respectively. The axis x_2 intersects y_2 at the pixel position of zenith, and it is parallel to the x-axis which intersects y at zero pixel position.

15

When *El* is large, we have:
if $2AB = AC$, $2AD \approx AE$;
hence $2ab \approx ac$. Therefore, the
curve *AC* and the line *ac* are
approximately linearly related.

Image (2-D)
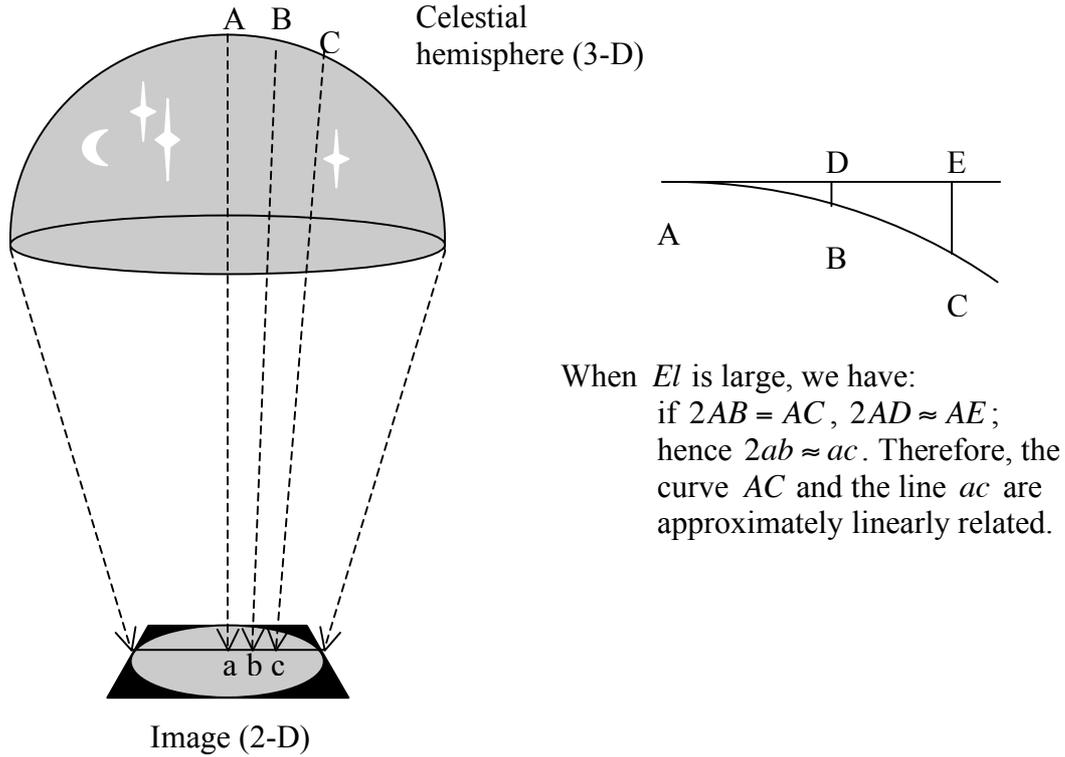
Fig. 4.2   The schematic representation of mapping from 3-D celestial hemisphere onto the 2-D image plane. The dash lines represent the light rays and the arrows indicate their travelling directions. The points A and a represent the sky position and image position of zenith, respectively. The points A, B and C locate on the same great circle of the celestial sphere. The points b and c are the image positions of B and C, respectively.

The pixel position of the best approximate zero-azimuth point can be determined by assessing the differences between the actual values of the azimuth, Az, of a set of points from the values, θ (Fig. 4.1), calculated as a result of image projection. The image position of a zero-azimuth point is, therefore, the pixel to which the two values, Az and θ, agree well under the minimum uncertainty of θ. The value of the angle θ of an image point can be obtained with the help of the pixel of the zenith. For example, as shown in Fig. 4.1, the image of a star is at the pixel position (X_S, Y_S) and a randomly chosen pixel, (X_0, Y_0), is assumed to be the pixel position of the best approximate zero-azimuth point. These two points and the pixel position of zenith, (X_Z, Y_Z), form a triangle, and the angle θ can be calculated by applying the cosine rule. As it is shown, the determination of the pixel position of the zero-azimuth point requires the pixel position of zenith to be determined first. As a result, the accuracy of the zero-azimuth pixel position, namely $\delta\theta$, is dependent upon the accuracy of the pixel position of zenith.

The determination of the pixel position of the zero-azimuth point serves for obtaining the value of angle α (Fig. 4.1) which defines the relative position between the line of the zero-azimuth points and positive x_2-axis and helps with the conversion between the Cartesian coordinates on an image and horizontal coordinates on the celestial hemisphere. The size of the angle α is fixed once the pixel position of the zero-azimuth point is determined since the other side of the angle is x_2-axis whose

position only depends on the pixel position of zenith. Therefore, the angle α has the same uncertainty as the angle θ.

When α is determined, we can define an angle β as the angle from the positive x_2-axis to the line joining the image position of a star with the image position of zenith. It follows

$$\beta = 2\pi - |\alpha| - Az \tag{4.3}$$

Hence, getting the x- and y- components of the pixel distance of the star (Fig. 4.1) by

$$\Delta x = P \cdot \cos\beta$$
$$\Delta y = P \cdot \sin\beta \tag{4.4}$$

and finally, the pixel position (X_S, Y_S) of the star is

$$X_S = X_Z + P \cdot \cos\beta$$
$$Y_S = Y_Z + P \cdot \sin\beta \tag{4.5}$$

Therefore, the mapping between Az/El and (X_S, Y_S) has been achieved. The following flow charts summarise the procedure of the conversions.
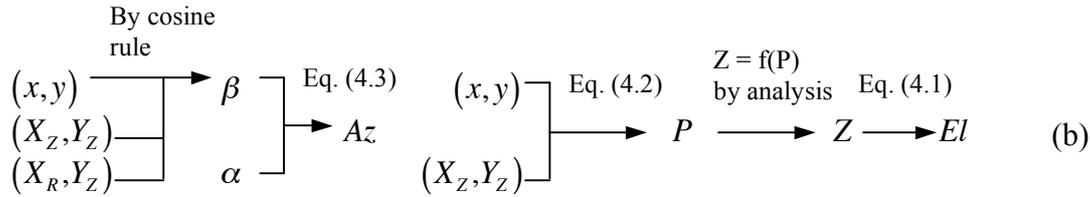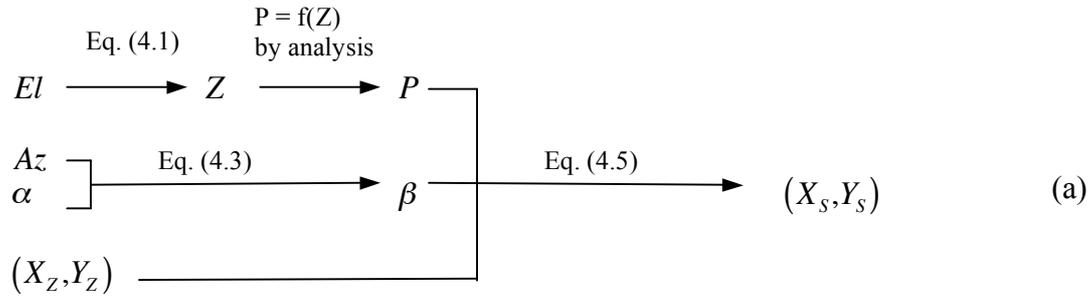
Fig. 4.3   The flow charts showing the procedure of the conversions (a) from Az/El to (x, y) and (b) from (x, y) to Az/El, respectively. The arrow indicates the flow direction and the content above the arrow tells the method used on each step. The (X_R, Y_Z) denotes a random point on the positive x_2-axis.

4.3 Mapping Process

4.3.1 Points sampling

It is not realistic to investigate all the points in the sky and their corresponding image positions so a set of points were sampled for investigation. The relation between the sampled points in the sky and their image projections was used to approximate the relation for all the points. The sampled points were chosen to be a group of stars since their Az/El and (x, y) were measurable.

Thirty-one stars were sampled from the all-sky image taken at 22:00 on 4 April 2008 (Fig. 4.5). The number of the sampled stars, 31, was not particularly chosen but it was considered to be sufficient for achieving a reasonable accuracy of mapping which was preferred to be within 1 pixel or 1 degree depending on the direction of conversion. The stars were sampled over the whole image in order that all the image points can be well represented, thus producing a good mapping accuracy. Meanwhile, the sampled stars were chosen to be the ones showing a sharp image against the sky background so as to easily identify their pixel positions. The choice of the all-sky image for sampling was based on the preference of a clear sky image where many stars were presented, thereby having more choices of the sampled points.

As shown in Fig. 3.2 in Chapter 3, the image of a star occupies a certain area, which consists of a number of pixels. The pixel position of a star was chosen to be the pixel that showed the highest pixel value among all the pixels making up the star's image. The pixel-value centroid position was also considered and was investigated separately for a couple of points. It was found that it did not make a difference since the brightest pixel was so sharp, hence being heavily weighted, so that the centroid star image position fell on the brightest pixel. Regarding the minimum accuracy of a star's image position being practically 1 pixel, the brightest pixel was chosen to be the image position of a star for the subsequent analysis.
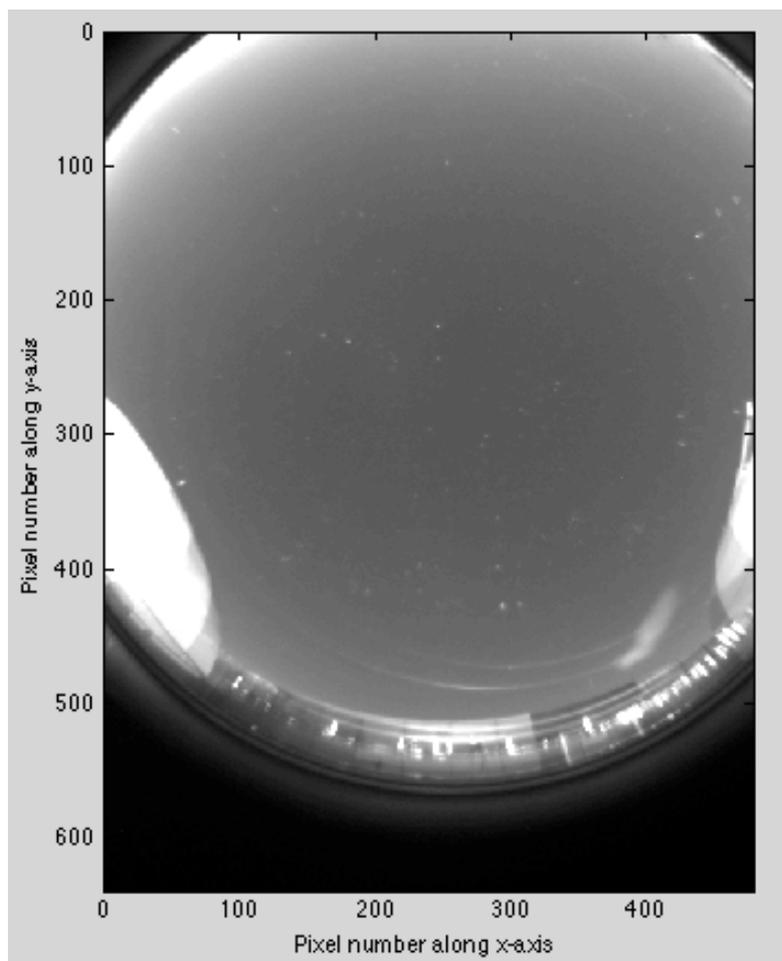
Fig. 4.5   The all-sky image taken by DAC at 22:00 on 4 April 2008. Thirty-one stars on the image were sampled to study the relation between the points on the hemisphere and their image positions on the image.

18

4.3.2 RA/Dec to Az/El

The problem of using the horizontal coordinates for mapping is that the values of Az/El of a star was not directly measurable from any reference source. However, every star is associated with unique RA/Dec from which Az/El of the samples were converted. The equations for the conversion are not discussed here but they can be found from *Astronomy with Your Personal Computer* by P.Duffett-Smith. The terrestrial position of DAC was also needed for the calculation of Az/El. In Fig. 4.5 shows a flow chart of the conversion.

Universal Time (UT)

Sidereal Time (ST)

Longitude of DAC

Local Sidereal Time (LST)

Latitude of DAC
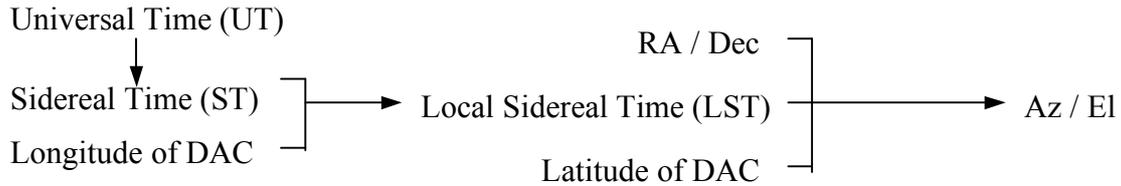
RA / Dec

Az / El

Fig. 4.5   The flow chart of the process of converting RA / Dec to Az / El. The arrow indicates the flow direction.

In the process, the values of RA/Dec of the sampled stars were read from Stellarium 0.9.0. The terrestrial position of the Department of Physics at Durham was taken to be $54°46'1''N$ $1°34'59''W$, as being suggested by the Astronomy and Astrophysics Group of Durham University.

4.3.3 Computation

4.3.3.1 Computing the pixel position of zenith

As discussed in section 4.2, in order to determine the pixel position of zenith, we need to investigate the numerical relation between the zenith distance and pixel distance. The zenith distance of every sampled point is independent of the choice of the image position of zenith and they were calculated through Eq. (4.1) after Az/El were obtained. In contrast, the length of a pixel distance depends on the choice of the image position of zenith. Therefore, in the process, a random pixel on the sampled image was chosen to be the image position of zenith and the pixel distances of the sampled points were calculated subsequently (Eq. (4.2)). To assess if the chosen pixel was appropriate to represent the image position of zenith, the pixel distance was plotted as a polynomial function of the zenith distance. Applying the least square fits, the best fit function of the plot was determined, thereby calculating the predicted values of the pixel distance of the samples. The deviations between the measured values and predicted values of the pixel distance were calculated subsequently, and so was the root mean square (RMS) of the pixel distance, δP. This procedure was repeated for every pixel on the sampled image. The image position of zenith was at best approximate pixel position when the RMS was the minimum. The mathematical equation established according to this image point of zenith was, therefore, at best approximate description of the relation between the zenith distance and pixel distance.

Following a similar procedure, the zenith distance was expressed as a polynomial function of the pixel distance. The minimum RMS of the zenith distance, δZ, was calculated and the image position of zenith was determined again at the minimum δZ to compare with the position determined at the minimum δP. The polynomial function Z of P was used when converting P to Z while the polynomial

function Z of P was for the reverse conversion. The two functions were all at best description of the mathematical relation between the zenith distance and pixel distance. The reason why two equations were used was because it was troublesome to simplify the high degree of polynomial factors when converting each reversely.

In the process, the uncertainties of the measured values of the zenith and pixel distances were presumed to be insignificant and were neglected. It was because the uncertainties of RA/Dec and the uncertainties of other parameters for calculating Az/El were extremely small, thus the uncertainty of the measured zenith distance; meanwhile, since the image position of a point was defined to be on a pixel, according to Eq. (4.2), it made the measured pixel distance of a star almost definite.

A computer programme was written in MATLAB to carry out all the calculations, finding out the pixel position of zenith while determining the coefficients of the function P of Z, and vice versa. A flow chart of the functions performed by the programme was shown below.

For first (X_z, Y_z), $y = Y_z$.

Select manually:
1. if and whose RMS (P's or Z's) is displayed for each plot of Z vs P at every (X_z, Y_z) within the range;
2. if the graphs, Z vs P, are plotted.

Program calculates the order of the layout of the graphs, Z vs P, according to the input range of (X_z, Y_z). Graph Z vs P is plotted for each possible (X_z, Y_z) within the range.

Input manually:
1. the possible range of (X_z, Y_z) for program to scan;
2. the value of the degree of a polynomial for the program to determine a function that describes the relation between Z and P.

Program starts with importing Z and the image position (x,y) of the sampled stars from the data files.

Check if value y is within the range. — No

Yes

Initialise x; i.e. x = first X_z.

Check if value x is within the range. — No / Yes

The projection of zenith is at (x, y).

Calculate P of each star to the point (x, y).

Find two polynomial functions of given degree:
1. P = f(Z); 2. Z = g(P)

Calculate: 1. RMS of P; 2. RMS of Z

Display RMS of P or Z if any of the two was selected to display

When x ≠ first X_z, the RMS was compared with the values according to other (x, y), the smallest ones are found for P and Z, respectively.

Plot and display the graphs if plot was selected.

$x = X_z + 1$

Display the current smallest RMS of P or Z if display RMS was selected previously.

$y = Y_z + 1$

Display results:
1. summary of input and number of sampled data;
2. the smallest RMS of P and Z, respectively;
3. the (X_z, Y_z) corresponding to the smallest RMS;
4. the coefficients of function P = f(Z) and Z = g(P), respectively, corresponding to (X_z, Y_z).

Program ends

Fig. 4.6 The flow chart of the process of the computer programme. It shows what functions the programme performed and what output it generated. The arrows indicate the process direction.

20

4.3.3.2 Computing the pixel position of a zero-azimuth point

The process of finding the best approximate pixel position of a zero-azimuth point was similar to that of finding the pixel position of zenith. However, this time, the linear relationship was established between the azimuth angles of the sampled stars and the azimuth image projection, i.e. the angle θ (Fig. 4.1). The angle θ was calculated geometrically, by use of the cosine rule, according to the triangle whose vertices were made up by the pixel position of zenith, the pixel position of a sampled star and the assumed pixel position of the zero-azimuth point.

The deviation between the azimuth Az and the angle θ was calculated for each sampled point, and the RMS of θ was computed subsequently. The RMS of θ was used to assess the agreement between the two angles. The pixel to which the minimum RMS of θ appeared was searched so as to determine the pixel position of the zero-azimuth point.

Except the pixel occupied by zenith, every pixel within the all-sky image was assumed to be the pixel position of the zero-azimuth point and, one by one, they were used to calculate the angle θ. Another computer program was written to scan the points and to execute the calculations. After determining the pixel position of the zero-azimuth point, the value of angle α (Fig. 4.1) was calculated.

4.4 Results

4.4.1 Pixel position of zenith

When the minimum RMS of the pixel distance or the one of the zenith distance was computed by scanning all the pixels to find the pixel position of the zenith, it was found the minimum RMS was also dependent on the value of the degree of polynomial. The analysed result of how the minimum RMS changed with respected to the degree of polynomial was shown in Fig. 4.6. The figure shows the minimum RMS of the pixel distance and the one of the zenith distance all decreased exponentially with the increase of the degree of polynomial. In Fig. 4.6(a), the minimum RMS of the pixel distance reached bottom and became steady with the increase of the degree of polynomial when the degree was three. The degree was four for the minimum RMS of the zenith distance, as shown in Fig. 4.6 (b). Despite the small RMS at the degree of polynomial 4, 5 and 6, the polynomial function P of Z and Z of P were both chosen to be 3 degrees polynomial for the usage of mapping. It was due to the concern that the polynomial was not well conditioned when the degree of freedom was further reduced. Meanwhile, the value of the minimum RMS of the pixel distance or the zenith distance at the 3 degrees did not differ significantly from the value at 4 degrees. Therefore, the polynomial function with the degree of 3 was considered to be good enough for the purpose of mapping.
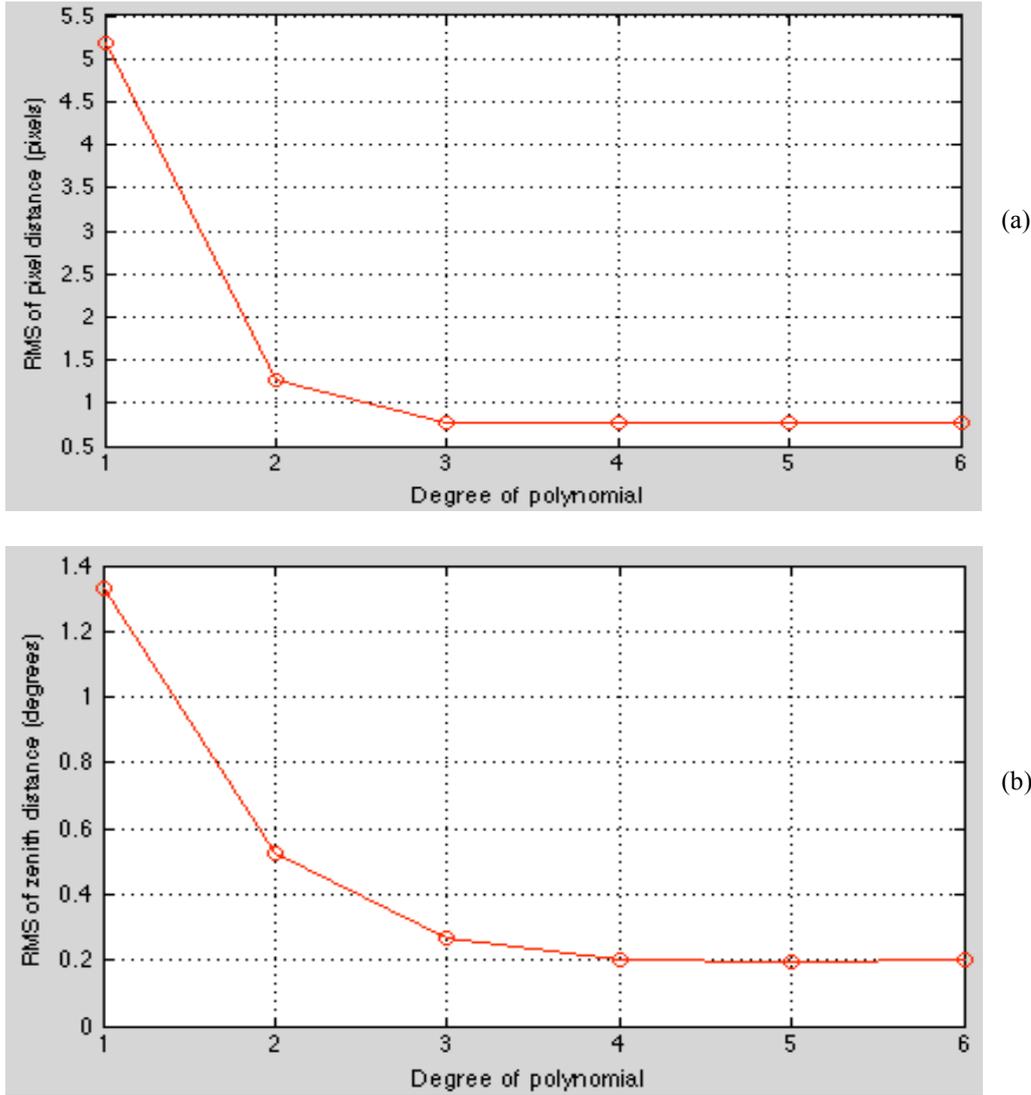
Fig. 4.7   The variation of (a) the minimum RMS of the pixel distance with respect to the degree of polynomial and (b) the minimum RMS of the zenith distance with respect to the degree of polynomial. In (a), the RMS was calculated when the pixel distance was expressed as a polynomial function of the zenith distance while, in (b), the zenith distance was a polynomial function of the pixel distance.

At 3 degrees of the polynomial, the data was plotted and was shown in Fig. 4.7. The minimum RMS of the pixel distance was 0.8 pixels. It was obtained when the pixel position of the zenith was at the pixel (256, 251). Since the image positions of sampled stars were defined for certain, the minimum RMS of the pixel distance was also the uncertainty of the image position of zenith. The function P in terms of Z was derived to be

$$P = -1.57 \times 10^{-4} \cdot Z^3 + 2.43 \times 10^{-3} \cdot Z^2 + 4.368 \cdot Z + 0.594 \qquad (4.6)$$

The minimum RMS of the zenith distance at the polynomial of 3 degrees was 0.3 degrees. It was also obtained when the pixel position of the zenith was at the pixel (256, 251), and the function Z was

$$Z = 1.34 \times 10^{-6} \cdot P^3 - 2.94 \times 10^{-4} \cdot P^2 + 0.251 \cdot P - 0.627 \qquad (4.7)$$

Therefore, the pixel position of zenith was determined to be the pixel (256, 251) which was used for all the subsequent calculations.
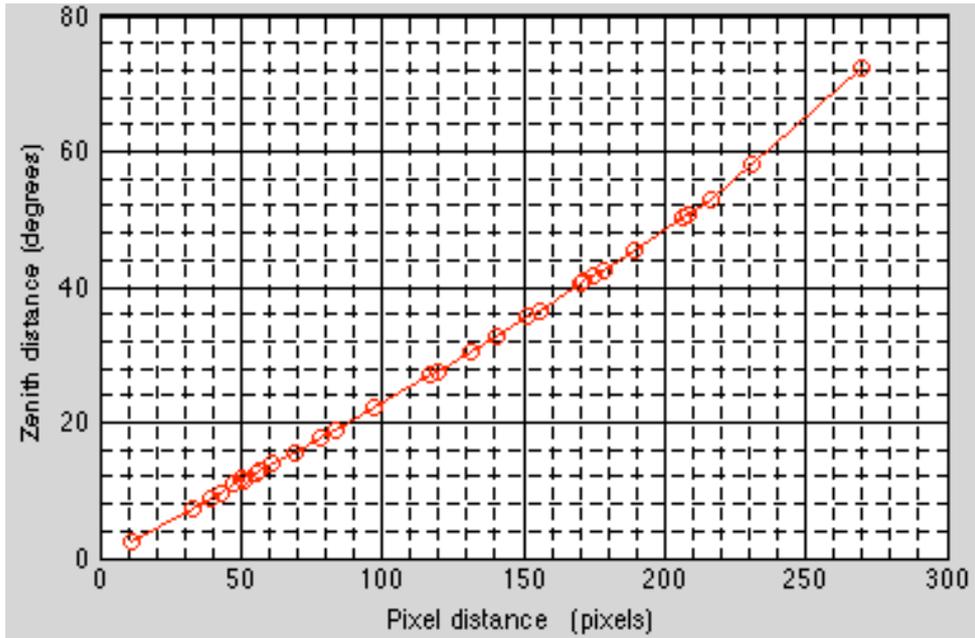
Fig. 4.8   The plot of Z versus P when the RMSs of Z and P were minimum. The minimum RMSs of Z and P were 0.3 degrees and 0.8 pixels, respectively. The data was fitted by a polynomial function of the degree of 3. The pixel position of zenith was at point (256, 251) when the minimum RMS was obtained.

4.4.2   Pixel position of a zero-azimuth point

The minimum RMS of the angle θ was 0.5 degrees and it was obtained when the pixel position of the best approximate zero-azimuth point was at the pixel position of (254, 196). In Fig. 4.9 shows the plot of the azimuth against the angle θ when the minimum RMS of θ was obtained. The angle α was calculated to be 92.1 degrees. As discussed in section 4.2, the uncertainty of the angle α is the same as the one of the angle θ, hence $\alpha = 92.1 \pm 0.5$ degrees.
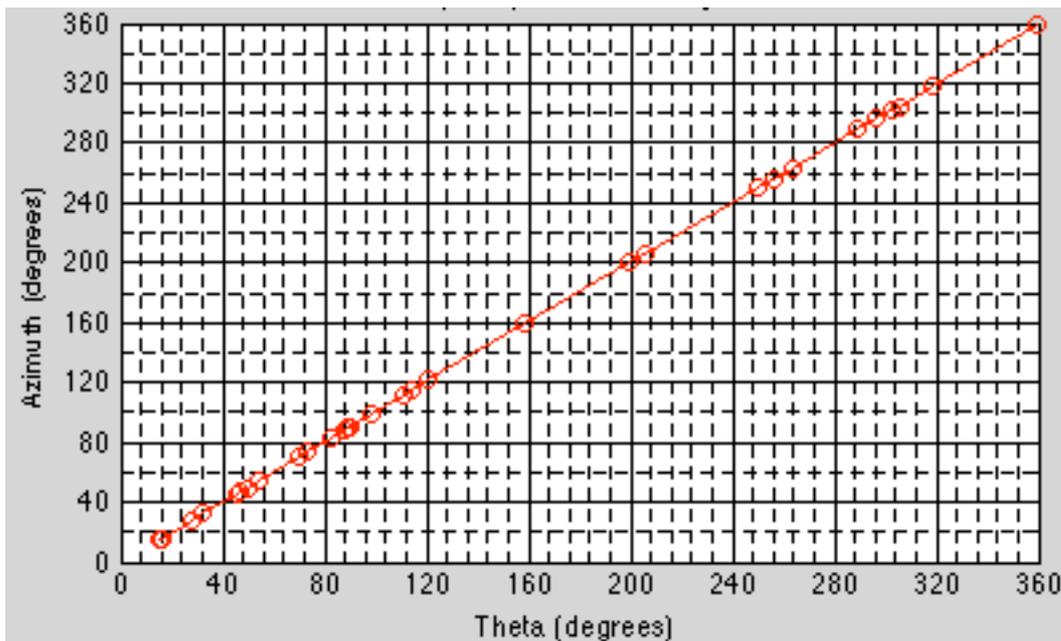
Fig. 4.9   The plot of azimuth against the angle θ at the minimum RMS of θ. The line was found to have the gradient value 1 and the y-intercept -0.006. The pixel position of the zero-azimuth used for the plot was (254, 196).

23

4.5 Mapping Tests

4.5.1 Purpose and process

There were two tests for mapping, one being aimed to assess the accuracy of mapping and the other being to test the mapping robustness. In order to assess the accuracy of mapping, the angle α and the equations (4.6) & (4.7) of the mapping results were written in a programme by which the conversion between Az/El and (x,y) was carried. The tested points were chosen to be the positions of the same sampled stars on the all-sky image that was taken at a later time, 22:49, on the same day. As the Earth rotated, each star, except Polaris, moved to a new point which had the different values of Az/El and (x,y). Therefore, the tested points were unrelated to the derivation of the mapping functions and parameters.

The predicted pixel positions of the tested star points, (X_S, Y_S), were calculated and they were compared with the observed values, (x,y). The deviations of them were used to calculate the RMS of the predicted pixel positions along x- and y-axis, respectively. From the RMS value, it was able to see how accuracy the mapping was. Such a process was repeated when assessing the accuracy of the predicted Az/El.

When testing the robustness of mapping, a group of stars was sampled from another image being taken at 21:46 on 4 April 2008, and the mapping was repeated. The consequent functions and angle were used to predict the pixel position of the stars on an image. The predicted stars and the image were the same as the ones being used for the previous test of mapping accuracy, namely the image taken at 22:49 on 4 April 2008. As a result, the difference between the predicted pixel positions of the same star was analysed to see how well the two predictions agreed.

4.5.2 Results of testing

The observed and the predicted pixel positions of every tested point were listed in the table shown in Table 4.1. The RMS of the predicted pixel position along the x-axis was calculated to be 1 pixels and so was the RMS along the y-axis. Therefore, when the azimuth and elevation of a point object in the sky is known, the pixel position of the object is likely to be found within 1 pixel of the predicted position.

The table in Table 4.2 shows the result of the predicted azimuth and elevation of the tested points along with the values that were converted from RA/Dec. The modulus of the difference between the two sets of the azimuth and elevation were listed in the columns titled ΔAz and ΔEl, respectively. The RMS of the azimuth was 0.7 degrees and the RMS of the elevation was calculated to be 0.2 degrees.

Table 4.1   The table of the pixel positions of the stars on the image taken at 22:49 on 4 April 2008. The x and y denote the observed pixel positions of the stars along x and y axis, respectively. The Xs and Ys are the predicted pixel positions. The deviations between two sets of values are listed under the column denoted by Δx and Δy.

Stars	x	y	Xs	Ys	ΔX	Δy
Vega	41	112	41	112	0	0
Capella	440	119	439	118	1	1
Menkalinan	441	152	442	150	1	2
Alphekka	51	288	51	289	0	1
Pollux	457	275	457	274	0	1

Castor	453	255	454	253	1	2
Arcturus	97	365	100	365	3	0
Izar	84	319	86	319	2	0
Eta Dra	133	154	134	155	1	1
Polaris	253	98	253	98	0	0
Seginus	115	283	116	284	1	1
Edasich	141	186	142	187	1	1
Pherkad	186	149	185	149	1	0
Kocab	199	147	199	148	0	1
Muscida	345	189	345	189	0	0
Cor Caroli	189	316	190	316	1	0
Alkaid	166	257	166	257	0	0
Pi UMA	337	243	338	243	1	0
Mu UMA	318	300	319	299	1	1
23 UMA	314	198	315	198	1	0
Lambda UMA	320	292	322	292	2	0
Mizar	188	239	190	240	2	1
Upsilon UMA	313	218	314	219	1	1
Giausar	255	186	256	187	1	1
Psi UMA	278	295	279	295	1	0
Alioth	208	241	209	241	1	0
Dubhe	272	218	272	218	0	0
Chi UMA	249	282	250	282	1	0
Megrez	231	240	232	240	1	0
Merak	276	241	278	242	2	1
Phad	244	255	245	256	1	1

Table 4.2   The table of the measured and predicted values of Az/El of the sampled stars at the frame taken at 22:49 on 4 April 2008. The values listed under Az and El are obtained by converting RA/Dec. The Az(pred) and El(pred) show the predicted values as a result of mapping. Under ΔAz and ΔEl list the modulus of the difference between the converted and predicted azimuth and elevation, respectively.

Stars	Az	El	Az(pred)	El(pred)	ΔAz	ΔEl
Vega	55.09	23.30	55.02	23.17	0.07	0.13
Capella	303.93	33.49	303.56	33.31	0.37	0.18
Menkalinan	296.50	38.09	296.05	38.53	0.45	0.44
Alphekka	98.30	39.03	98.13	38.99	0.17	0.04
Pollux	261.51	40.83	261.09	40.75	0.42	0.08
Castor	267.27	42.20	266.74	42.34	0.53	0.14
Arcturus	123.89	43.47	123.54	42.74	0.35	0.73
Izar	109.76	46.25	109.47	45.78	0.29	0.47
Eta Dra	49.64	53.54	49.64	53.37	0.00	0.17
Polaris	359.13	54.23	359.02	54.30	0.11	0.07
Seginus	101.00	56.45	100.69	56.43	0.31	0.02
Edasich	58.51	59.55	58.42	59.51	0.09	0.04
Pherkad	32.49	61.28	32.36	61.53	0.13	0.25
Kocab	26.64	62.92	26.63	62.76	0.01	0.16
Muscida	302.78	64.94	302.76	65.15	0.02	0.21
Cor Caroli	132.84	68.83	132.03	68.66	0.81	0.17
Alkaid	91.82	69.43	91.71	69.39	0.11	0.04
Pi UMA	273.36	71.30	273.54	71.42	0.18	0.12
Mu UMA	230.61	71.90	229.58	71.96	1.03	0.06

23 UMA	310.06	72.04	310.32	72.07	0.26	0.03
Lambda UMA	235.96	72.40	235.26	72.66	0.70	0.26
Mizar	78.44	74.72	77.89	74.25	0.55	0.47
Upsilon UMA	297.32	75.01	297.97	74.98	0.65	0.03
Giausar	357.99	75.42	358.78	75.18	0.79	0.24
Psi UMA	206.10	78.82	204.47	78.83	1.63	0.01
Alioth	75.90	79.05	76.13	78.87	0.23	0.18
Dubhe	331.32	81.76	332.03	81.75	0.71	0.01
Chi UMA	167.84	82.88	165.18	82.90	2.66	0.02
Megrez	63.19	84.18	64.15	83.96	0.96	0.22
Merak	291.55	84.76	294.47	84.14	2.92	0.62
Phad	112.02	87.34	106.33	87.50	5.69	0.16

The result of testing the robustness of the mapping was given in Fig. 4.10. It showed it was about 60% likely that the two sets of the predicted pixel positions of the stars along x-axis were 1 pixel away from each other. The maximum pixel position difference was 2 pixels which was about 10% of the total tested stars. For the predicted pixel position difference along y-axis, it showed that about 65% of the two results agreed and the rest differed by 1 pixel.

The pixel position of the zenith, functions and the angle α obtained by the second mapping was slightly different from the first. In Table 4.3 listed the results from each mapping.


(a)

(b)
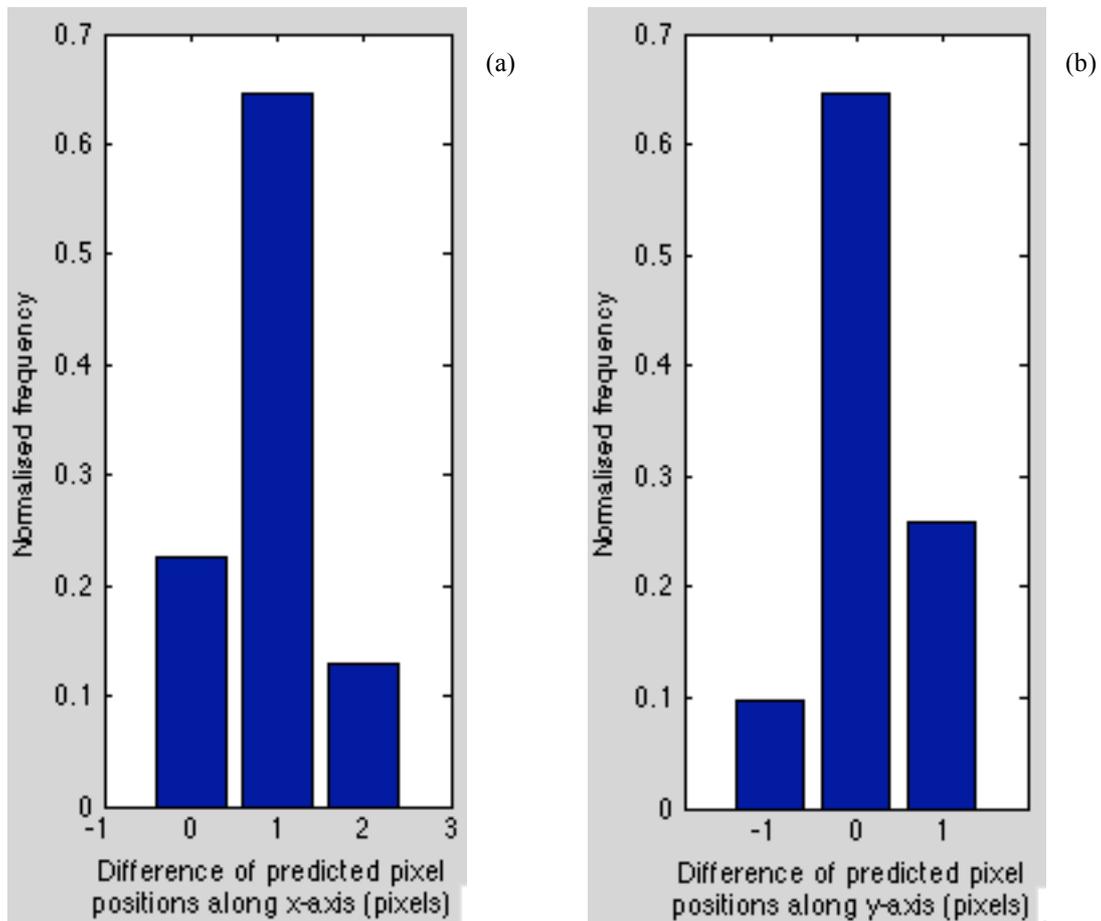
Fig. 4.10 Histogram of the difference between the two sets of the predicted pixel positions of the same stars. Each set of the predicted result was given by a different mapping. The position differences along x- and y-axis were shown in (a) and (b), respectively.

Table 4.3   The table of results from the two separate mappings. The results of the first and second mapping were shown in the row of Mapping 1 and Mapping 2, respectively. Under the Coeff 1, Coeff 2, Coeff 3 and Coeff 4 listed the coefficients of the polynomial function P (pixel distance) of Z (zenith distance) for Z to the power of 3 to 0, respectively. The (Xz, Yz) showed the pixel position of the zenith.

	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Xz	Yz	Alpha (degrees)
Mapping 1	-0.000157	0.00243	4.368	0.594	256	251	92.1
Mapping 2	-0.000166	0.00379	4.313	1.123	255	251	91.8

4.6   Conclusions and Discussions

In this chapter, we have determined the image position of zenith is at the pixel position of (256,251) within an uncertainty of 0.8 pixels. The best approximate zero-azimuth image position is at (254,196) and the azimuth angles measured according to this point are within an uncertainty of 0.5 degrees. We have demonstrated the image positions of stars can be determined to an accuracy of 1 pixel and, for the reverse process, the sky positions can be determined to the accuracy of 0.2 degrees in elevation and of 0.7 degrees in azimuth. The mapping was tested to be robust. The testing result suggests the mapping was stable at the level of 1 pixel.

In Fig. 4.8, the plot shows the zenith distance against the pixel distance of the sampled stars were quite linearly related when the zenith distance was below 50 degrees which corresponded to the elevations higher than 40 degrees. It agrees with the earlier approximation (Fig. 4.2) that the zenith distance and pixel distance could be considered to have linear relation when the points were at the high elevations. Therefore, it indicates that the data was analysed appropriately and the subsequent results were reasonable.

The predicted pixel position of the zero-azimuth point was lined with the predicted pixel position of zenith. The line was found to pass through the pixel nearby the image of the star Polaris. It strongly suggests the result of the pixel position of the zero-azimuth point was at a correct position relative to the predicted pixel position of zenith and other points on an image.

The 1-pixel uncertainty of the mapping from the points in the sky to the ones on an image was small in contrast to the area of a star image that normally occupied 4 to 25 pixels. According to the method by which the image position of a star was sampled, the resulted 1-pixel uncertainty was the uncertainty of the predicted position of the brightest pixel that was approximately at the centre of a star image. Therefore, the predicted pixel position of a star was almost certain to be at a pixel making up the star image.

In Table 4.1, the x and y were listed in the ascending order of the elevation from the top to bottom. It was expected the deviation of the predicted image position from the observed image position would decrease with the increase of the elevation since the image distortion became stronger and stronger toward the edge. However, as shown in Table 4.1, the deviations along x or y-axis fluctuates randomly and keeps small even the elevation goes down. The result suggests the accuracy of mapping is not affected by the image distortion.

The RMS of the predicted azimuth of tested points was 0.7 degrees, being threefold larger than the RMS of the predicted elevation. The 0.7-degree RMS in predicted azimuth is reasonable since it cannot be smaller than the uncertainty of the

angle α, 0.5 degrees, based on which the prediction was made. Small RMS of the predicted elevation was a result of the accurately determined pixel position of zenith.

The results of mapping are reasonable and satisfactory. It contributes a useful function to DAC's implementation and provides an essential means for the further image analysis when star positioning is required.

Chapter 5

Photometry analysis

5.1 Purpose and Overall Sky Brightness Condition in Durham

The photometry analysis was aimed at measuring how well the magnitude of the stars could be determined through the DAC's images. It helps to understand quantitatively the sky brightness condition in Durham, and by comparing it with the sky brightness from another observatory site, we can estimate how well the magnitude of the stars can be measured when DAC is deployed there.

Before carrying out any analysis for the task, we first take a look at the overall sky brightness condition in Durham to obtain a general idea of how it looks like at night. Taking the clear night sky framed at 22:00 on 4th April 2008 as an example, a dark frame was subtracted from the image frame to remove the dark current and hot pixels, thereafter the 3-D sky flux distribution and the corresponding histogram were plotted (Fig. 5.1 & 5.2).

In Fig. 5.1, the overall flux distribution of the all-sky image is bowl-shaped and the flux of the stars appears to be the spikes sticking out from the bottom. The sharp increase of the sky flux from zenith to horizon, i.e. from the middle of the bowl to the edge, indicates the sky pollution by the streetlights was marked. In Fig. 5.2, the frequency distribution shows three distinct peaks. One is at the saturated value, 255 ADUs, being as a result of the intense light reflected from the surrounding buildings. Another peak is at around 10 ADUs, indicating the sensor area on which the image did not focus. The last peak is at the centre of the range, representing the frequency distribution of the actual sky flux. The mode value, about 75 ADUs, suggests a possible value of the sky background brightness and the tail on the right of the mode also implies the sky was contaminated by other light sources.
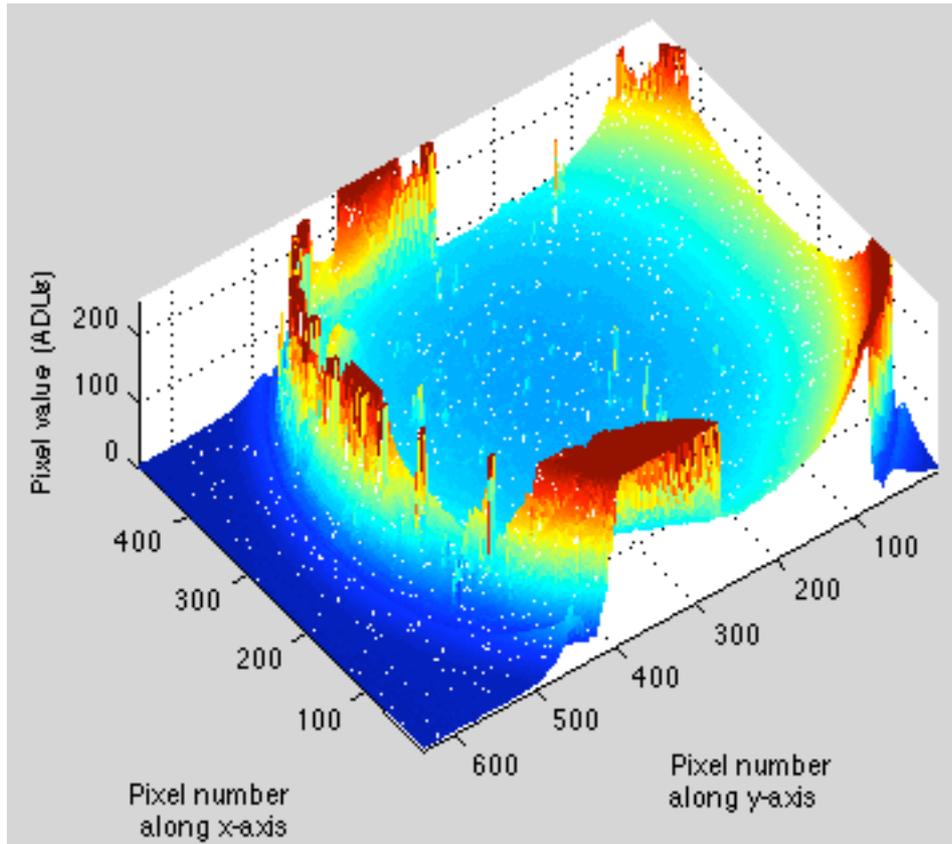
Fig. 5.1   The 3-D plot of the flux of the all-sky image taken at 22:00 on 4 April 2008, from which the dark counts were deducted. The plot was inverted along the x-axis due to the default of the MATLAB for the 3-D plot. The colour bands, from the blue to red on the plot, indicate the ascending order of the pixel values.
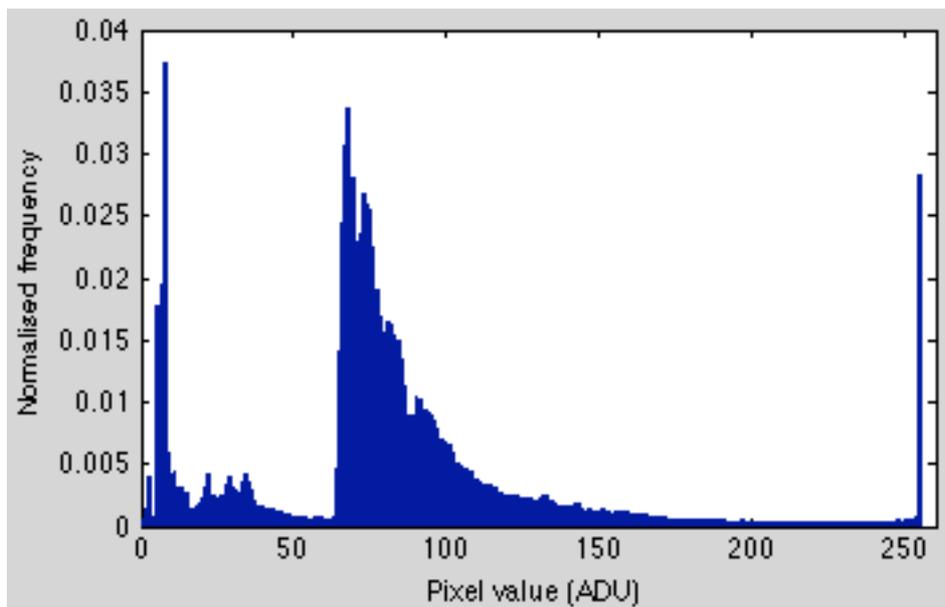

Fig. 5.2   The histogram of the flux of the all sky image taken at 22:00 on 4 April 2008. The dark counts were subtracted from the total counts before plotting.

30

5.2   General Method of Star Magnitude Measurement

The most commonly applied method to calculate a star's apparent magnitude is by finding the star's flux and putting it into the following equation.

$$magnitude = const - 2.5\log_{10}(flux) \qquad (5.1) \ ①$$

In this work, we applied this equation in the following analysis.

Since the counts of a pixel is proportional to the incident photons, the flux of a star, in DAC image analysis, was considered to be the counts of the sum of pixel values of the star's image subtracting the counts of the background of the same area, so its unit being ADU or counts. The background counts refer to the pixel values produced by the sky brightness and dark current. In Eq. (5.1), *const* is the zero reference constant whose value depends on the flux of the star that has zero apparent magnitude. In process, the value of *const* was not measured literally by finding the flux of the star of zero apparent magnitude, because there may not be a zero-magnitude star in a frame at all times and, if there is one, the flux changes when the star moves across the frame as a result of flux reduction by airmass. Regarding the convenience and accuracy of the magnitude measurements, the value of *const* was set arbitrarily to be 50 for all the calculations in our analysis when *const* was used. A magnitude calculated by Eq. (5.1) was, therefore, named as instrumental magnitude, which was the magnitude being actually measured by reading a star's flux in terms of pixel counts. A star's apparent magnitude, being the value when *const* in Eq. (5.1) is set according to a zero-magnitude star, was determined when establishing its mathematical relation with the instrumental magnitude. This relation was derived empirically by means of investigating how the flux of a star changed with the airmass. The apparent magnitude of a star is not constantly the same amount larger or smaller than the instrumental magnitude of the same star on account of reduction of the flux by airmass. A detailed analysis of finding the relation is given in the following section where airmass is discussed. Therefore, in the following contents, the magnitude of a star refers to the instrumental magnitude unless being specified.

The uncertainty of a star's magnitude can be estimated by use of the signal-to-noise ratio of the image. The following equations were used in our analysis to calculate the signal-to-noise ratio (Eq. (5.2)), star flux (Eq. (5.3)) and magnitude uncertainty (Eq. (5.4)), respectively.

$$\frac{S}{N} \approx \frac{F_S}{\sqrt{F_S/G + n\sigma_A^2 + n\sigma_A^2/p}} \qquad (5.2) \ ②$$

$$F_S = P_S - n\overline{P}_A \qquad (5.3)$$

$$\Delta m \approx 2.5\log(\frac{S+N}{N}) \approx 1.0857\frac{N}{S} \qquad (5.4) \ ③$$

---

①. The derivation of Eq. (5.1) is not discussed here but it can found from most textbooks of star magnitude calculation.

②&③. The derivations of the equations (5.2) & (5.4) are beyond the scope of the thesis. A discussion of the two equations can be found on *Measuring the Signal-to-Noise Ratio S/N of the CCD Image of a Star or Nebula* by J.H. Simonetti at http://www.phys.vt.edu/~jhs/phys3154/snr20040108.pdf.

where S and N are the signal and noise values, respectively, F_S being the star flux, G being the gain of the camera, n being the number of pixels within the aperture that samples the star image, p being the number of the pixels on the annulus that samples the sky brightness, σ_A being the RMS of the pixel value in annulus, P_S being the total counts within the aperture, $\overline{P}_A$ being the mean pixel value of sky brightness sampled by annulus and Δm being the instrumental magnitude uncertainty.

When applying the above equations to calculate a star's magnitude and uncertainty, we first need to establish a method of sampling the star and background flux. We take the image of the star Alioth as an example to show the method that was used. The brightest pixel was chosen to be the centre of the star image and a circular aperture was drawn. Around it, a circular annulus was defined with the same centre (Fig. 5.3(a)).

When sampling, the aperture's radius should be large enough for an aperture to cover the most star image; meanwhile it should also be small so that there is no more than one star image in the aperture. The size of an annulus should be chosen to avoid falling itself on the anything other than sky flux, and so the gap between an aperture and annulus keeps the annulus away from the tail of a star flux, which is normally bell-shaped (Fig. 5.3(b)). Due to this reason, the radii of the aperture, inner and outer annulus were initialised to be 4, 6 and 9 pixels long, respectively.

| (a) | (b) |
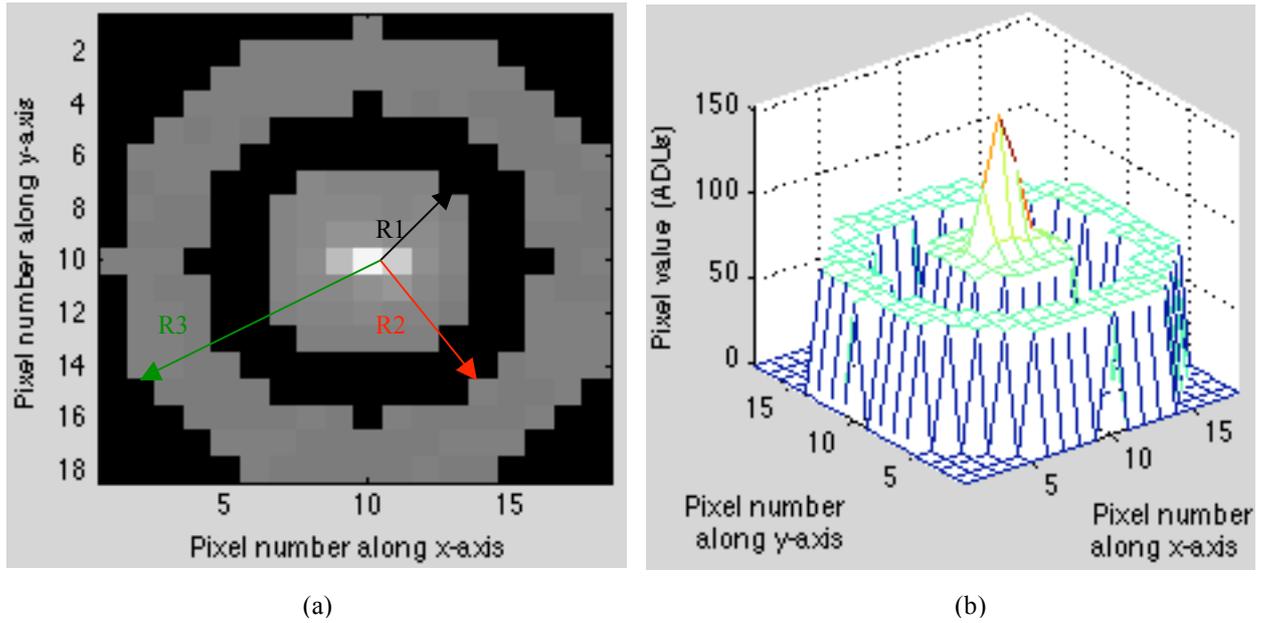
Fig. 5.3   The image of star Alioth in (a) and the corresponding 3-D plot in (b). In (a), the star's image is presented in the centre aperture and the grey annulus shows the brightness of sampled sky. The dark area, showing zero counts, is the part of the image being taken away when sampling. R1, R2 and R3 indicate the radii of the aperture, inner and outer annulus, respectively. In (b) shows the flux strength on each part of the sample.

As a result, we can calculate the values of n and p, which equal the round-off integer of the areas of the aperture and annulus, respectively. Knowing n and p, the values of P_S, $\overline{P}_A$ and F_S can be calculated subsequently. For ease of process, the pixel values on an image frame had subtracted the ones on a dark frame before the sampling was carried out. It is why there is no term regarding the dark counts in Eq. (5.3). The pixel values within the annulus were used to assess the sky brightness around the star. The frequency distribution of the values (Fig. 5.4) was analyzed by

Gaussian statistics since the distribution was expected to follow the Gaussian profile, providing no strong interference from another light source presented on the annulus. The value of σ_A can, therefore, be computed from FWHM of the distribution.
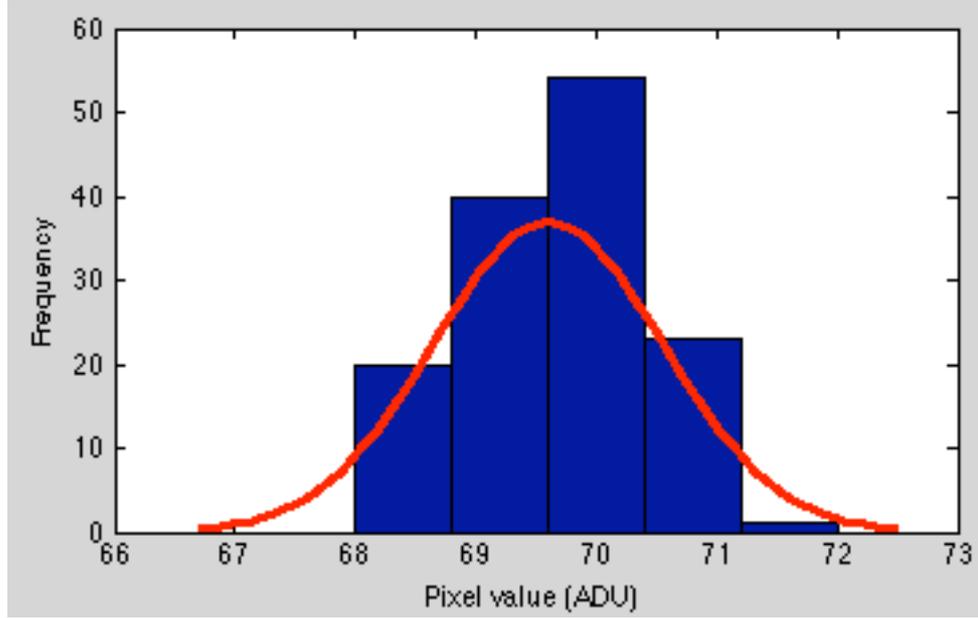
Fig. 5.4   The histogram of the pixel value within the annulus of Alioth. The distribution was approximated by Gaussian statistics and the curve indicates the best fit of the approximate Gaussian profile. The top of the curve falls on 69.6 ADU and the FWHM is about 2.25 ADU.

Apart from G, the values of the parameters in equations (5.2) and (5.3) depend exclusively on how the flux is sampled and can be measured directly from the image. The gain, G, has to be determined through other means. The gain in Eq. (5.2) is measured in the unit of number of electrons per ADU. Unfortunately, the same term is shared by a setting parameter, which does not have a unit, on the camera's operating software, as described in section 2.2.2. The relation between the two gains is unknown. In the following contents of the photometry analysis, the term gain means the parameter in Eq. (5.2). In the next section, we will discuss how the gain being determined and its significance to the star magnitude uncertainties.

5.3   The Gain

5.3.1   Determination of the gain of the camera

The value of the gain was not given on the manual of the camera. It was estimated empirically by investigating the sky background and was calculated by assuming the electrons released by the incident photons from the sky obey Poisson distribution. The following equation was used.

$$\mu_{sky}G = \left(\sigma_{sky}G\right)^2 \qquad\qquad (5.5)\ \text{①}$$

---

①. The number of electrons released by the photons from the sky is $\mu_{sky}G$ and the RMS is $\sigma_{sky}G$ since the gain is measured in number of electrons per ADU. Because "all electrons released by incident photons obey Poisson statistics" (*Measuring the Signal-to-Noise Ratio S/N of the CCD Image of a Star or Nebula* by J.H. Simonetti), we have $\sigma_{sky}G = \sqrt{\mu_{sky}G}$, thus having Eq. (5.5).

where G is the gain, μ_{sky} and σ_{sky} being the mean and RMS of the pixel values of the selected patch of an sky image, respectively.

Dividing the sum of the all the pixel values within the patch of sky by the pixel numbers, we calculated the mean pixel value, μ_{sky}, in ADU or counts. The corresponding RMS, σ_{sky}, was estimated by assessing FWHM of the frequency distribution of the pixel values. As we can see, the accuracy of the gain is mathematically dependent upon how well the value σ_{sky} can be determined. As a result, the Gaussian distribution of the pixel values was expected. In order to approach it, the patch of the sky was selected to be free from, or weakly contaminated by, the any sort of intense light sources, for example, the visible stars, streetlights or clouds.

A processing example is shown as the following. The image frame, being taken at 22:00 on 4 April 2008, was used to sample the sky background. A circular patch of the sky, having its centre at the pixel position (241, 269) and a radius of 20 pixels, was chosen. The fluctuation of the pixel values within the patch, in Fig. 5.5, shows a wavy pattern on which there is no apparent distinctive spike. Therefore, we assumed the pixel values from the sampled sky patch were adequate to use for our purposes. The result of further analysis of the pixel values was shown in Fig. 5.6, from which we can see that the distribution of the pixel values can be approximated by the Gaussian statistics. After calculation, the sky flux of the sample was 66.0 ± 0.1 ADUs, and the corresponding gain was 54.5 electrons/ADU.
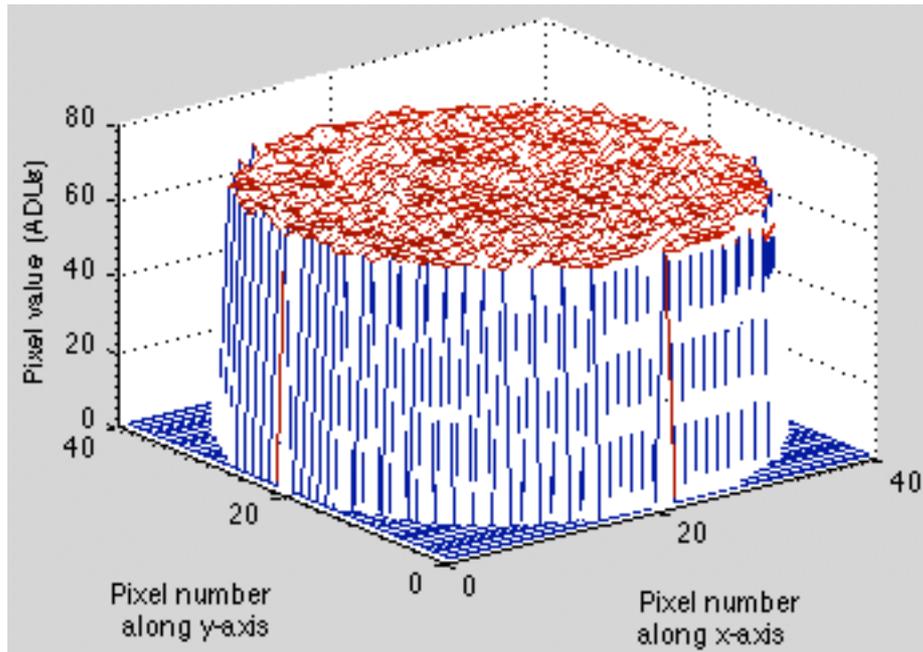
Fig. 5.5   The 3-D plot of the flux of a patch of the sky. It centred at the pixel position (241, 269) on the frame captured at 22:00 on 4 April 2008, and had a radius of 20 pixels, so around 1200 pixels were sampled.

Five circular sky patches at the different positions of the same frame were investigated. They were all chosen to be the ones whose centre was close to the zenith position of the image, because the influence from the streetlights to the sky background within this region was relatively weaker than the sky near the horizon, as shown in Fig. 5.1. The result from each patch of sky is listed in Table 5.1.

34
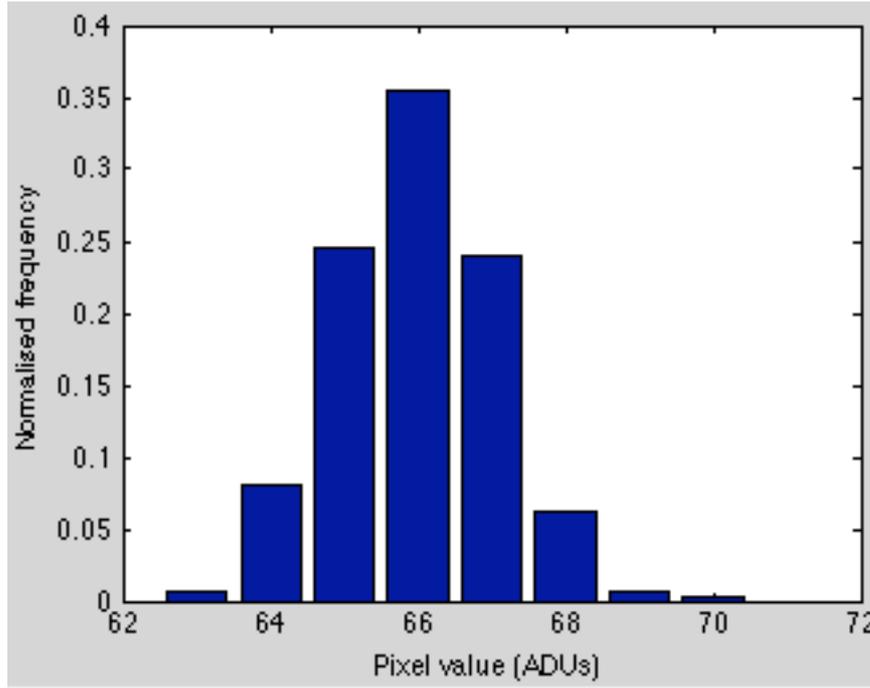
Fig. 5.6   Histogram of the normalised frequency of the pixel values within the sampled sky patch. The graph shows a mode at 66 ADUs and the FWHM of the distribution is 2.591.

Table 5.1   The table listing the results from the sampled five patches of sky. The values at Xc and Yc show the pixel positions of the centre of each sky patch. The column R is the radius. The mean flux and RMS of flux are μ_{sky} and σ_{sky}, respectively. G shows the gain derived from each patch.

Xc	Yc	R (pixels)	Mean flux (ADU/pixel)	RMS of flux (ADU/pixel)	G (electrons/ADU)
241	269	20	66.0	1.1	54.5
189	290	20	66.8	1.0	66.8
194	202	20	70.4	1.2	48.9
335	231	10	66.5	1.0	66.5
228	225	10	68.0	0.9	84.0

The mean and RMS of the gain was calculated subsequently, and the gain was empirically estimated to be 64 ± 12 electrons/ADU. Apparently, the error is large; however, it does not necessarily mean the error is significant, nor does it mean the determined gain value is inadequate for the photometry analysis of the camera. A test was carried out afterwards to investigate how well the gain we measured.

5.3.2 Gain test

After the gain estimate, we need to assess if the value is reasonable and adequate for our measuring purposes. The method of assessment we used was to measure the magnitude and uncertainty of the same stars on two adjacent frames. The magnitude of a star on one frame was expected to be nearly the same as itself on an adjacent frame on account of the fact that the star moved very little across the frame after a short time. Therefore, if the estimated gain is adequate, the measurement of the same star magnitude on adjacent frames will agree within the uncertainty.

To examine the agreement between such two sets of values, we have

$$Delta = \frac{mag_1 - mag_2}{\sqrt{error_1^2 + error_2^2}} \qquad (5.6)$$

where mag_1, $error_1$ and mag_2, $error_2$ are the magnitudes and uncertainties of the star on two frames, respectively. We expected $-1 \le Delta \le 1$ since $|mag_1 - mag_2| \le \sqrt{error_1^2 + error_2^2}$ when the two sets of magnitudes agree with each other. In other words, we expect $Delta$ to be normal distributed with a mean of 0 and RMS of 1 if the errors are correctly determined by gain.

In the process, the frames were chosen to be the ones taken at 22:00 and 22:02, respectively, so the frames were two-minute apart. The thirty-one stars used for mapping in Chapter 4 were selected again to be the sample for the photometry analysis because of their large position variation in the sky. According to Eq. (5.6), a value of Delta was computed from every sampled star. The values were then arranged in bins and were plotted against its occurring frequency, as shown in Fig. 5.7.
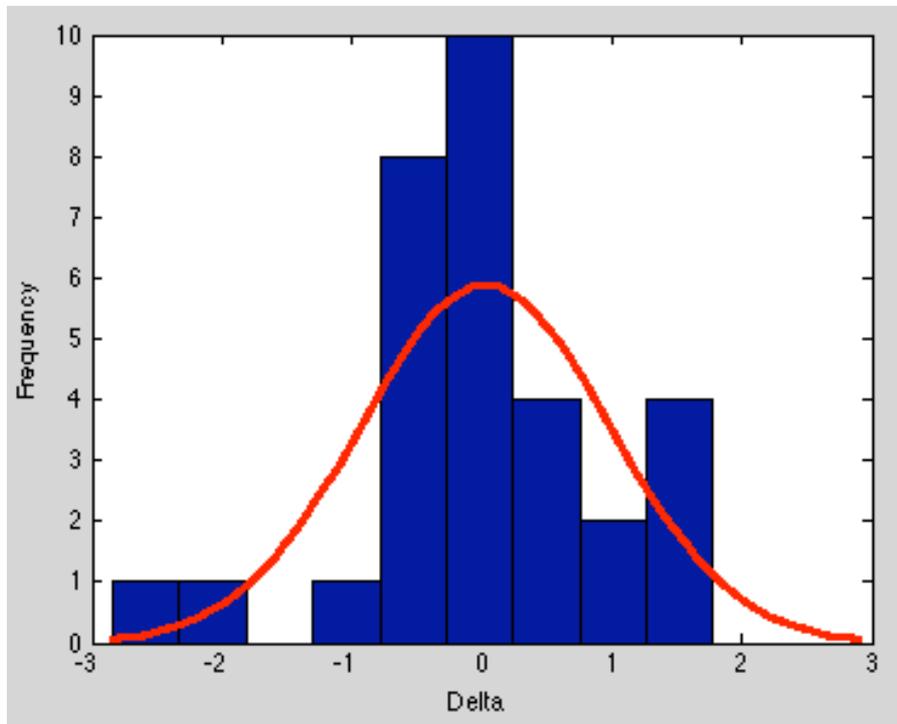
Fig. 5.7 The histogram of Delta. Since the values of Delta were not discrete, they were grouped into 9 bins as shown in the graph. The blue bars indicate the frequency of the occurrence of delta while the redline shows the best fit of the Gaussian approximation. The FWHM of the best fit is 2.25 and the mean of delta is 0.03.

According to the plot in Fig. 5.7, the mean of the delta was found to be 0.03 and FWHM of the frequency distribution was about 2.25; therefore, Delta = 0 ± 1, which satisfies our early expectation and it suggests that the empirically determined value of the gain is adequate for the purpose of photometry analysis of the star magnitudes. In Fig. 5.8 shows a comparison of the star magnitudes at the two frames. As an extra check of the agreement of the two sets of magnitudes, we found the best fit of the data was a straight line with a gradient value of 1. We tried the reduced chi-square and it was about 0.87, which suggests the gain estimate is reasonable.
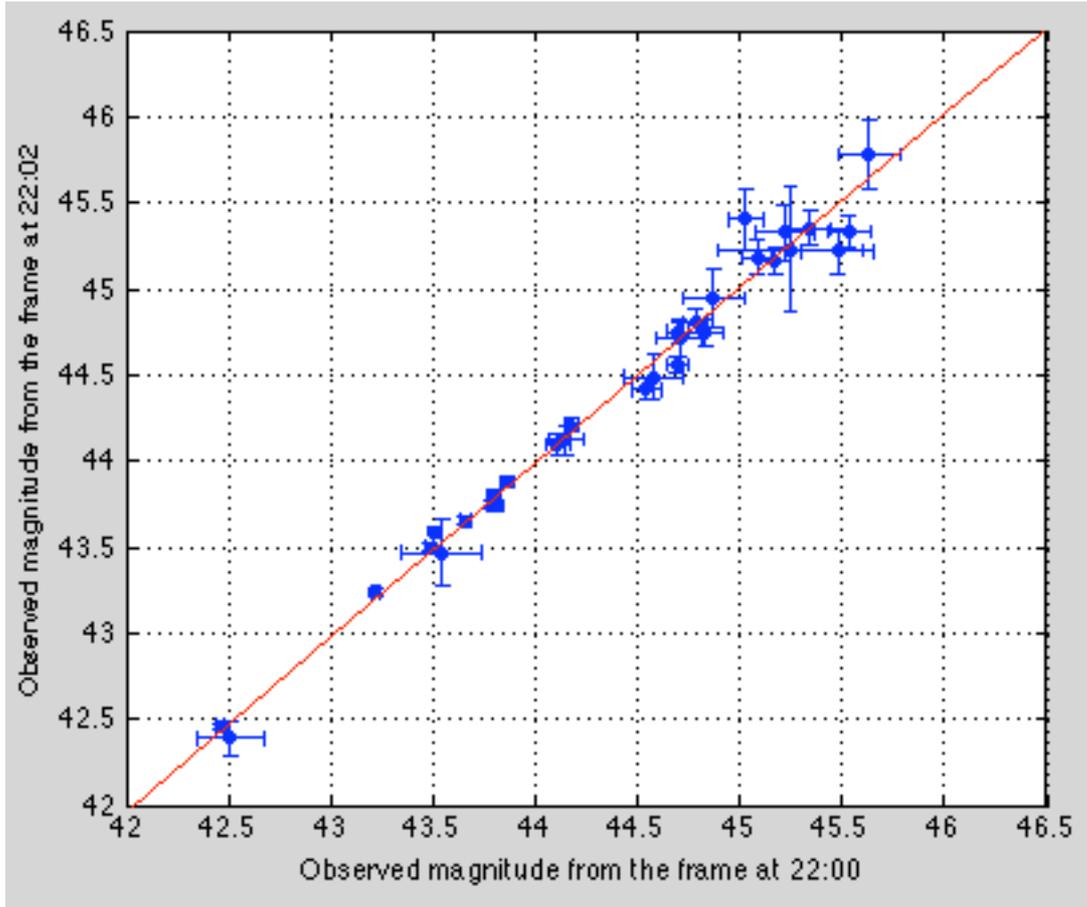
Fig. 5.8   The plot of the magnitudes of the sampled stars of the two frames, which were taken two-minute apart. The redline going through the data points shows the best fit of the linearity of the data. It has a gradient about 1.01.

5.3.3   Gain contribution to star magnitude errors

The previous testing result suggested the determined value of the gain was adequate to do photometry analysis for the images though it had a large uncertainty. The question that arose from it was why it was so since it might contradict our expectation that slight variation of the gain would cause a big change to the signal-to-noise ratio.

According to eq (5.2), we see the gain is not the only factor to determine the signal-to-noise ratio. There are another two terms on the denominator, being $n\sigma_A^2$ and $n\sigma_A^2/p$, respectively. The contribution of each term to the denominator and so to the signal-to-noise ratio was investigated. The results are listed in Table 5.2.

On the table, the column under 1st, 2nd and 3rd term refers to F_S/G, $n\sigma_A^2$ and $n\sigma_A^2/p$ on the denominator of the Eq. (5.2), respectively, and the column under 'mag error' shows the magnitude error of each star. From the table, we can see the value of the second term is about a few ten or hundred times greater than the each corresponding 1st term. Therefore, it was the second term that dominated the signal-to-noise ratio and so the magnitude error of the stars. It explains why the seemingly inaccurate value of gain succeeded the previous test. When reducing the gain to 14 electrons per ADU, which is far less than the measured value of gain, we see from the table the second term still dominating the signal-to-noise ratio and the magnitude error of the stars being almost unchanged.

Table 5.2　A table of stars and their magnitude errors, etc from the frame taken at 22:00. The 1st term, 2nd term and 3rd term in the table refer to F_S/G, $n\sigma_b^2$ and $n\sigma_A^2/p$, respectively, which were given in the Eq. (5.2). The 'mag error' refers to the magnitude uncertainty of the stars. Here are listed two sets of the first term and magnitude errors, each of which was calculated under a different gain value, i.e. 64 and 14 electrons / ADU. Since only the 1st term and magnitude error were gain dependent, the values of the 2nd and 3rd terms keep unchanged and are not listed when the gain is 14 electrons / ADU.

Stars	G = 64 electrons / ADU				G = 14 electrons / ADU	
	1st term	2nd term	3rd term	mag error	1st term	mag error
Vega	6.40	4930.40	35.73	0.20	29.27	0.20
Alphekka	1.65	274.79	1.99	0.15	7.52	0.15
Arcturus	17.34	10572.00	76.61	0.16	79.25	0.16
Izar	2.68	95.98	0.70	0.07	12.25	0.08
Capella	16.18	191.18	1.39	0.01	73.97	0.02
Menkalinan	4.41	73.56	0.53	0.04	20.14	0.04
Pollux	7.92	84.30	0.61	0.02	36.20	0.02
Eta Dra	2.03	153.39	1.11	0.11	9.30	0.11
Castor	4.73	80.58	0.58	0.03	21.62	0.03
Seginus	1.28	92.84	0.67	0.18	5.84	0.18
Edasich	1.16	110.67	0.80	0.14	5.29	0.15
Polaris	3.59	154.34	1.12	0.06	16.43	0.06
Pherkad	1.97	72.69	0.53	0.09	9.00	0.09
Kocab	4.98	81.40	0.59	0.03	22.78	0.03
Alkaid	4.96	51.45	0.37	0.03	22.67	0.03
Cor Caroli	1.86	53.34	0.39	0.06	8.52	0.06
Mizar	5.40	66.55	0.48	0.03	24.67	0.03
Muscida	1.35	38.05	0.28	0.08	6.15	0.08
Alioth	6.23	52.07	0.38	0.02	28.47	0.02
Giausar	1.32	50.23	0.36	0.08	6.02	0.08
Mu UMA	2.50	334.97	2.43	0.14	11.43	0.14
Lambda UMA	1.27	732.42	5.31	0.36	5.79	0.36
23 UMA	1.15	38.29	0.28	0.11	5.24	0.11
Megrez	1.08	127.58	0.92	0.08	4.93	0.09
Pi UMA	2.34	42.46	0.31	0.05	10.68	0.06
Chi UMA	1.13	41.72	0.30	0.09	5.17	0.10
Psi UMA	1.97	50.79	0.37	0.06	8.98	0.06
Phad	3.50	317.37	2.30	0.08	16.01	0.09
Upsilon UMA	0.76	75.35	0.55	0.15	3.48	0.16
Dubhe	5.74	52.20	0.38	0.02	26.22	0.02
Merak	3.23	44.11	0.32	0.04	14.78	0.04

Since the second term is related to the fluctuation of sky brightness, an interpretation to such a result is the background sky noise being the dominant source of magnitude errors under the sky condition of Durham. If the camera was moved to a dark sky observatory where the stars were the main source of the brightness of sky background, the gain would be more likely to dominate the star magnitude errors. In that case, a better gain value is needed. After all, the determined value of gain is good enough for measurement of the star magnitudes when the camera is used under the sky condition like Durham.

5.4 Magnitude Calibration

In order to assess how good the applied method is for measuring the star magnitudes, the measured magnitudes of previously sampled stars were calibrated by the readings from GAIA – Graphical Astronomy and Image Analysis Tool ①.

When taking the readings from GAIA on the same frame, the semi-major axis of the aperture, annulus inner and outer scale were set to be 4, 1.5 and 2.2, respectively, so as to produce about the same size of the aperture and annulus as the ones used in the previous measurements. The options of mean flux of sky background and sky variance in GAIA settings were ticked in order that the star flux and background flux in GAIA mean the same as the ones in our method.

The result is shown below (Fig. 5.9). We can see the line of best fit, indicated by the red line on the graph, goes through almost all the points. The gradient of the line of the best fit was calculated to be 1. Therefore, in conclusion, the two sets of data are consistent with each other and the applied method is adequate for the measurement of the star magnitudes.
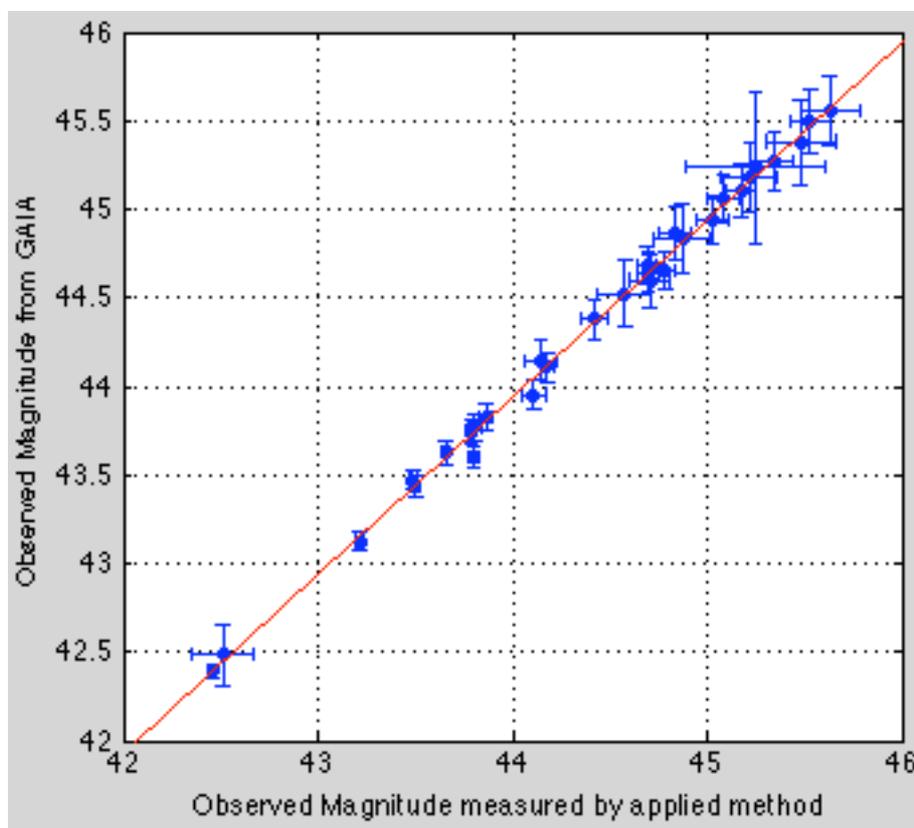

Fig. 5.9   The plot of the instrumental magnitudes measured by our method against the ones by GAIA. The red line, being the best fit of the data, has gradient value of 1. The zero reference constant in GAIA for magnitude calculation was set to be 50, being the same as the value used in our method.

5.5 Airmass and Star Magnitude Measurement

5.5.1 General theory

A star's flux is absorbed as it travels through the Earth's atmosphere. The amount of absorbed flux depends on the length of the atmosphere it travels through. The airmass is commonly used to measure such a length and it was defined as

$$airmass = 1/\cos(\frac{\pi}{2} - El) \qquad (5.7)$$

where *El* is the elevation. Therefore, the airmass has the minimum value of 1 to the direction of zenith and increases its value with the decrease of the elevation.

As a result of light reduction by the airmass, a star can be observed with changing brightness when it moves across the sky. Ideally, we assume the earth's atmosphere is homogenous, and so the reduction on the magnitude of a star is approximated to be proportional to the airmass. Therefore, we have

$$\Delta mag = a \cdot Airmass + b \qquad (5.8)$$

where Δmag is the reduction on the magnitude of a star, *a* and *b* being the coefficients of the function. Since

$$\Delta mag = Mag(obs) - Mag(cat) \qquad (5.9)$$

where *Mag(obs)* represents the instrumental magnitude of a star measured on earth and *Mag(cat)* the catalogue value of the star's apparent magnitude when the atmosphere is absent, hence

$$Mag(obs) - Mag(cat) = a \cdot Airmass + b \qquad (5.10)$$

It means, by studying the change in magnitude with airmass, we establish a function, by which the magnitude category that an image star falls on can be estimated in terms of catalogue magnitude; meanwhile, the instrumental magnitude of a star of any magnitude category, at any sky position within the scope, can be predicted, and consequently, the flux of the star is also predicable by reversing the equation (5.1).

5.5.2 Change in magnitude with airmass

To empirically determine the function of the change in star magnitude with respect to airmass (Eq. (5.10)), the instrumental magnitudes of the previously sampled stars for magnitude calibration were used and their catalogue magnitude values were read from the Astronomical Almanac for the year 2007. Thus, we can determine the left hand side of the Eq. (5.10). The airmass of each star was calculated from the corresponding elevation (Eq. (5.7)) that had been obtained in the previous chapter of astrometry. When analysing, the sampled star Vega having an airmass greater than 2, which is less than 30˚ in elevation, was not used on account of the heavy background light pollution near the horizon. The result is shown below.
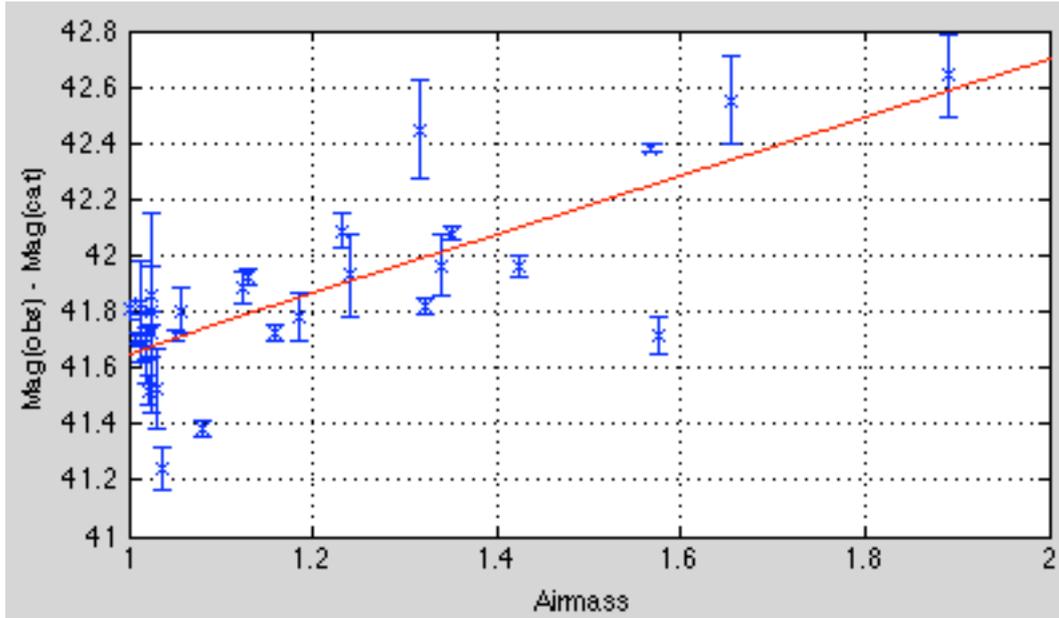
Fig. 5.10   The plot of the change in star magnitude with respect to airmass. The red line represents the best fit of the linear relation of the data points. The y-label shows Mag(obs) minus Mag(cat), i.e. the left hand side of Eq. (5.10).

In Fig. 5.10, the line of the best fit of the data points was found to have a gradient of 1.05 magnitudes per airmass and the reduced chi-square is 17.8, which indicates the pattern of the resulting points seems not reveal any correlation between the two axes.

Further investigation was done to understand the reason for such a result. We first examined the term of the reduced chi-square of every sampled star to find the ones that contributed most to value. Such stars whose rows were shaded are listed in Table 5.3.

Table 5.3   The table of the sampled stars with the corresponding value contributing to the reduced chi-square for the data in Fig. 5.10. The stars were listed in descend order of their airmass. The Mag_obs, Error_obs, Mag_cat (V) and Mag_diff refer to the instrumental magnitude, uncertainty of the instrumental magnitude, catalogue magnitude and difference between the instrumental and catalogue magnitude, respectively. The shaded rows show the stars whose reduced chi-square terms are large.

Stars	Reduced chi-square term	Mag_obs	Error_obs	Mag_cat (V)	Mag_diff
Alphekka	0.0042	44.87	0.15	2.23	42.64
Arcturus	0.0586	42.51	0.16	-0.04	42.55
Izar	2.2456	44.42	0.07	2.7	41.72
Capella	2.5975	42.46	0.01	0.08	42.38
Menkalinan	0.4199	43.86	0.04	1.9	41.96
Pollux	0.2144	43.22	0.02	1.14	42.08
Eta Dra	0.0062	44.70	0.11	2.74	41.96
Castor	1.0409	43.80	0.03	1.98	41.82
Seginus	0.2277	45.48	0.18	3.03	42.45
Edasich	0.0008	45.22	0.14	3.29	41.93
Polaris	0.3476	44.11	0.06	2.02	42.09
Pherkad	0.0206	44.83	0.09	3.05	41.78

41

Kocab	0.3072	43.81	0.03	2.08	41.73
Alkaid	0.8302	43.79	0.03	1.86	41.93
Cor Caroli	0.1080	44.79	0.06	2.9	41.89
Mizar	5.7738	43.66	0.03	2.27	41.39
Muscida	0.0439	45.17	0.08	3.37	41.80
Alioth	0.0113	43.49	0.02	1.77	41.72
Giausar	1.1201	45.08	0.08	3.84	41.24
Mu UMA	0.0426	44.58	0.14	3.05	41.53
Lambda UMA	0.0035	45.25	0.36	3.45	41.80
23 UMA	0.0967	45.53	0.11	3.67	41.86
Megrez	0.0087	45.03	0.08	3.31	41.72
Pi UMA	0.2852	44.69	0.05	3.17	41.52
Chi UMA	0.0062	45.34	0.09	3.71	41.63
Psi UMA	0.0020	44.70	0.06	3.01	41.69
Phad	0.0066	44.15	0.08	2.44	41.71
Upsilon UMA	0.0390	45.63	0.15	3.8	41.83
Dubhe	0.1483	43.50	0.02	1.8	41.70
Merak	0.5903	44.18	0.04	2.37	41.81

The table shows the stars that were far off the line of the best fit in Fig. 5.10, thus having a large reduced chi-square term, are Izar, Capella, Menkalinan, Castor, Mizar, Giausar and Merak. Most of them are not a single star, Izar, Castor and Mizar being the binary stars, Capella consisting of four stars in two binary pairs, and Menkalinan being a triple star. It suggests that the magnitude catalogue values of these stars should be reconsidered to take account of the flux contribution from the constituted stars. It is reasonable since the aperture for flux collection has a certain size of 4 pixels in radius, which is about 0.8 degrees. Therefore, their catalogue values were re-collected from the SKY2000 Catalog Version 4 (Myers+ 2002) on SIMBAD database. The new catalogue magnitudes, $Mag(cat)$, being used in our analysis, were calculated by adding the flux of the constituted stars in a star system, as shown by the equations below.

$$\sum_i flux = 10^{-0.4\,mag_1} + 10^{-0.4\,mag_2} + ... + 10^{-0.4\,mag_i} \qquad (5.11)$$

$$Mag(cat) = 0 - 2.5\log_{10}(\sum flux) \qquad (5.12)$$

where mag_i represents the catalogue magnitude of a constituted star.

When reading the catalogue magnitude of the binary star Castor from the SKY2000, we found that the two values were given to be 1.580 and 2.880. It was different from what we got from other reference sources, like the Astronomical Almanac for the year 2007 or the teaching contents[①] in astronomy on the web of the University of Illinois, which indicated one of the two stars or the brighter one had a value of around 1.95 or 1.98. The magnitude 1.580 was more likely to be the value of the binary star system of Castor since we got 1.587 after combining the 1.98 and 2.88. However, we were still not sure if 1.580 was a mistake though it was very likely to be. As a result, the star Castor was eliminated in the subsequent analysis.

---

① It was written by J. B. Kaler and can be found at http://www.astro.uiuc.edu/~jkaler/sow/castor.html.

No suggestion was found to explain the reason why the star Merak had a large reduced chi-square term. However, for star Giausar, "there is some indication that it changes brightness erratically by about a tenth of the magnitude"① – by Prof. Kaler from the University of Illinois. Therefore, Giausar was also removed from the following analysis.

After changing the catalogue magnitudes of the stars, the graph of magnitude change against airmass was plotted again (Fig. 5.11). The new graph shows the gradient of the best fit to be 1.11 magnitudes per airmass and $\tilde{\chi}^2 = 3.5$. Thus, the result was largely improved but it is still unsatisfactory as we expect $\tilde{\chi}^2 \leq 1$. Another possible source that could affect the result is the quantum efficiency of the CCD sensor of DAC when detecting the lights with different wavelengths, so the colour terms were investigated.
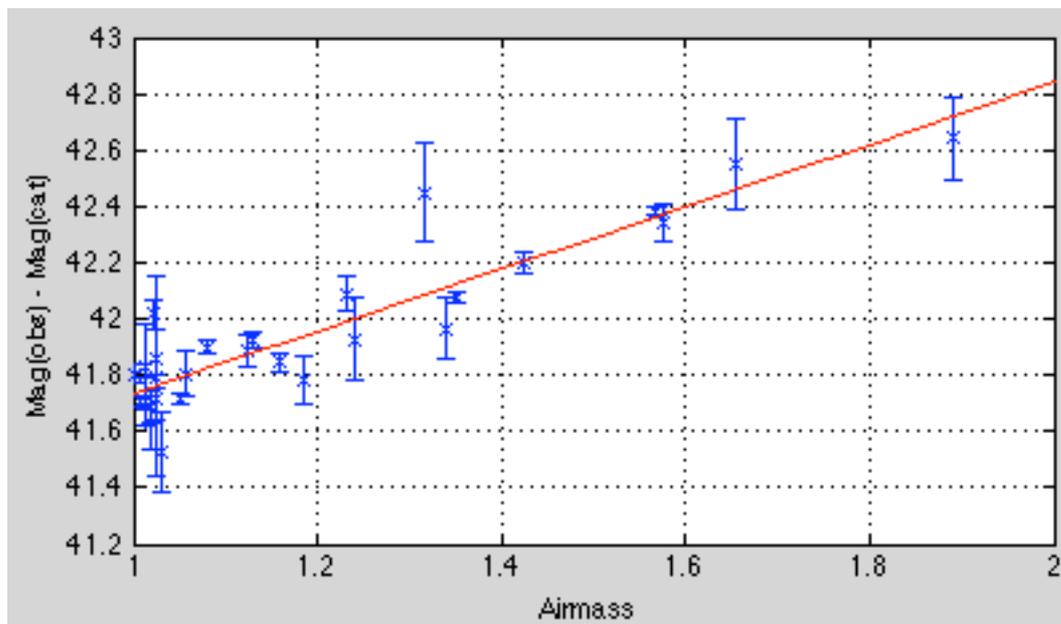

Fig. 5.11   The new plot of the change in star magnitude with respect to airmass. The red line represents the best fit of the data.

5.5.3   Investigation in colour terms

The investigation of the colour terms aimed at finding out if the CCD sensor of DAC is biased toward blue or visual band of frequencies. As we know, a star emits light of a broad band of frequencies. A hot star predominately radiates blue light and so has a low B-V around 0, whereas a cool star gives off more red light and has a large B-V about 1. The catalogue magnitudes that were collected from the Astronomical Almanac or SKY2000 were the V-band magnitudes. Therefore, if the sensor is biased toward red light, the change in magnitude, i.e. $Mag(obs) - Mag(cat)$, of the stars that emits more blue light is expected to be consistently larger than the predicted change, i.e. above the line of the best fit in Fig. 5.11. The analysed result is shown in Fig. 5.12, of which the vertical axis is the difference between the measured and predicted change in magnitude.

---

① The contents can be found on the teaching web in astronomy from the University of Illinois at http://www.astro.uiuc.edu/~jkaler/sow/giausar.html.
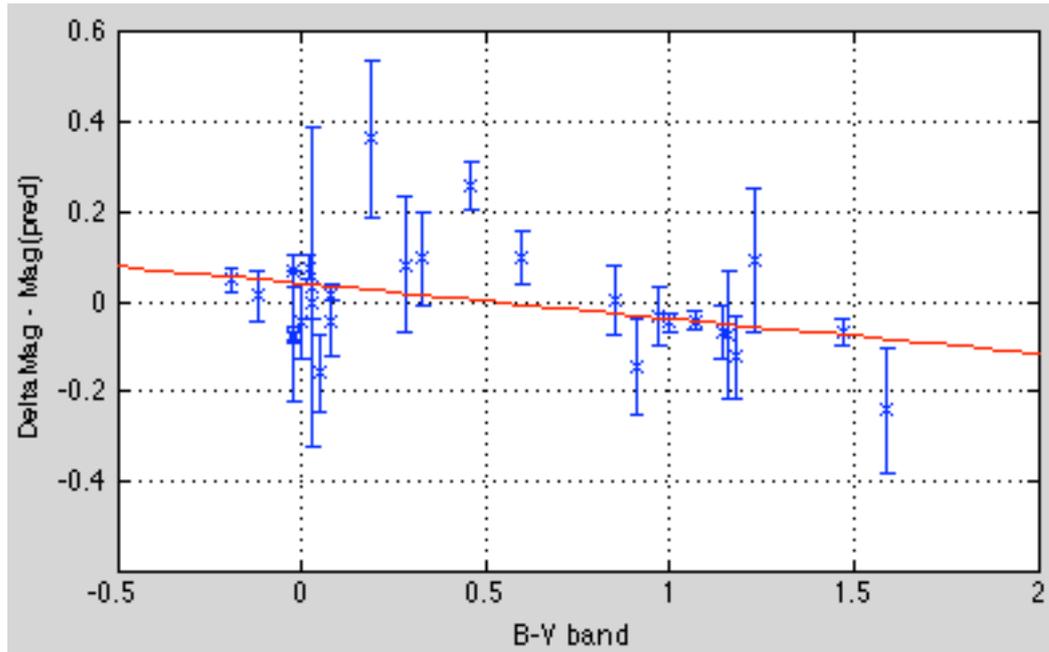
Fig. 5.12   The plot of the deviation in magnitude change against B-V colour term. In the label of vertical axis, DeltaMag represents the measured change in the magnitude of a star and Mag(pred) is the predicted change value that is on the line of the best fit in Fig. 5.11. The B-V of the sampled stars were collected from the Astronomical Almanac for the year 2007 and SKY2000.

In Fig. 5.12, the points seem to equally distribute around the points (0, 0) and (1, 0). The gradient of the line of the best fit is -0.08 and $\tilde{\chi}^2 = 3.3$. It does not matter how well the data points follow such a line of best fit; however, what the small gradient and seemingly equal point distribution around (0, 0) and (1, 0) suggest is the sensor of the camera was not biased toward red or blue colour. Thus, the effect of the colour term on the magnitude analysis can be discounted.

5.5.4 Result and discussion

The previous analysis shows that the colour terms do not affect the result in Fig. 5.11. If Fig. 5.11 shows the best result that could be obtained by the current analysing technique, the gradient of the line of the best fit in Fig. 5.11 would be about 5 times greater than the expected value since the rate of change in V-band magnitude with respect to airmass is approximately 0.2 magnitudes per airmass[①]. The large gradient implies the magnitude deduction of the stars at a low elevation is significantly large. There are four possible causes, which could result a large gradient.

First, the image of a star, which was at a low elevation, was largely distorted along the radial direction as a result of fisheye view. The aperture that was used to take the flux measurements had a constant size, i.e. not changing with the elevation. Consequently, the measured flux was partly lost when the distorted image spread out of the area of the aperture. The reason why we cannot use bigger aperture to solve this problem is because it introduces more noises and produces distinct large error bars.

[①] More information about the change in magnitude with respect to airmass is available on Magnitude Corrections for Atmospheric Extinction, written by Daniel. W. E. Green. A copy can be found at http://www.cfa.harvard.edu/icq/ICQExtinct.html.

The second possible reason that the gradient is large is because the fisheye lens absorbs or reflects more light away from the optical axis, i.e. when light coming through the lens from the edge. To test it, we can use a laser beam scanning from the top to the edge and find out the light loss. The third possibility could be the nonlinearity of the CCD chip on which the quantum efficiency varies from pixel to pixel. The fourth possibility is, after a long time operation, the Perspex dome was partly covered with condensed water vapour or dusts, which scattered the light away before they entered camera.

The large reduced chi-square is likely to be the result of such a change in magnitude being significantly large at small elevations. After all, it is the best result we could achieve at this stage so, from the plot in Fig. 51.4, we obtain

$$Mag(obs) - Mag(cat) = 1.11 \cdot Airmass + 40.63 \qquad (5.13)$$

by which the instrumental and catalogue magnitude of a star are predicable and derivable from each other when the star's sky position is known.

5.6 Magnitude Accuracy Determined by DAC under Durham Sky Condition

By use of Eq. (5.13), we can estimate the magnitude category that an image star falls on. The degree of accuracy of the estimation determines how well the DAC can perform in the measurement of photometry. To outline such a degree of accuracy for the stars of different magnitude categories and at different image position, the following investigation was done.

The brightness of sky background and its RMS, on the area of about zenith position and 30° elevation, were measured, in the unit of ADU per pixel, by the same method described in section 5.3.1. The results are listed in Table 5.4(a). Thereafter the instrumental magnitudes of the stars of a range of catalogue magnitudes representing 1st to 6th magnitude were calculated by Eq. (5.13), and so were corresponding flux (Eq. (5.1)). Substituting the results of the background RMS and star flux for F_S and σ_A, respectively, into the Eq. (5.2), we get the signal-to-noise ratios and, subsequently, the magnitude uncertainties by Eq. (5.4). Therefore, as shown in Table 5.4(b), the DAC is capable of determining the brightness of 6th magnitude stars within 1 magnitude accuracy when the stars are near zenith position; whereas only the 4th magnitude stars can be determined within the same accuracy when the stars move down to 30° in elevation.

The degree of accuracy that a magnitude can be measured is subject to the condition of the sky background brightness, i.e. the value of RMS listed in Table 5.4(a). In Fig. 5.14 shows the analysed fluctuation of the RMS of sky brightness against the azimuth position at airmass = 2. Thus, the RMS at airmass = 2 listed in Table 5.4(a) is only a special case where the sky was lightly contaminated by the streetlights. The degree of accuracy that the DAC can measure could be even worse when a star is observed at another azimuth position. As a result, in Table 5.4(b) lists the best accuracy of each magnitude category that can be determined by DAC under the sky brightness condition in Durham.

Table 5.4   The table of, (a), the sky brightness and its RMS in Durham, and (b), the best magnitude accuracy, that can be determined by DAC, under the sky condition in Durham, for different categories of star magnitude.

Position	Sky brightness (ADU per pixel)	RMS (ADU per pixel)	
At zenith (airmass = 1)	66	1.06	(a)
At El = 30 degrees (airmass = 2)	83	1.65	

Catalogue magnitude	1.00	2.00	3.00	4.0	5.0	6	
Magnitude uncertainty at zentith	0.01	0.03	0.06	0.16	0.4	1	(b)
Magnitude uncertainty at El = 30 degrees	0.04	0.11	0.27	0.7	1.7	4	
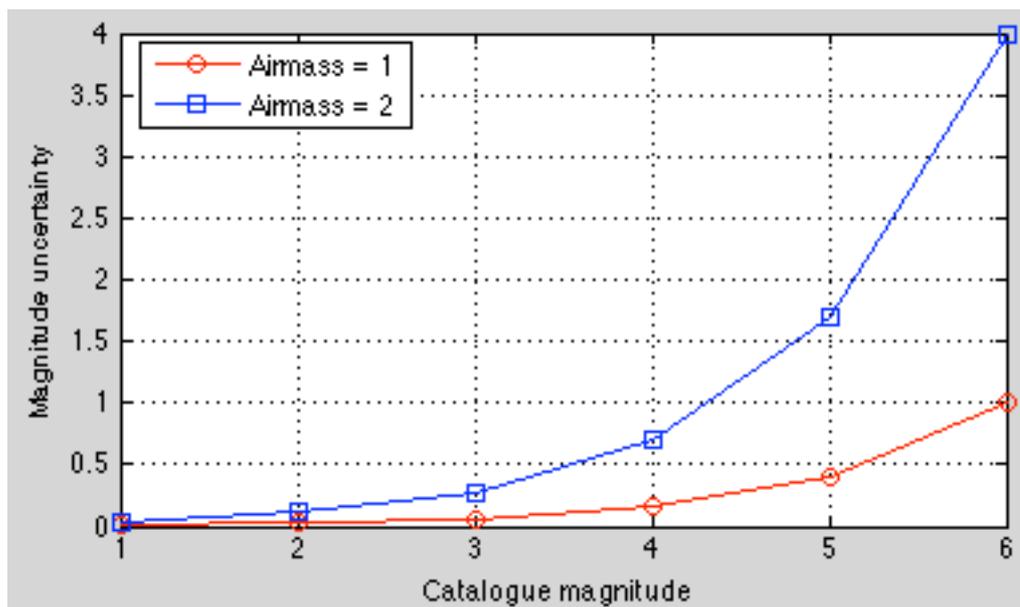
Fig. 5.13   The plot of magnitude uncertainty versus the catalogue magnitude from Table 5.4(b).
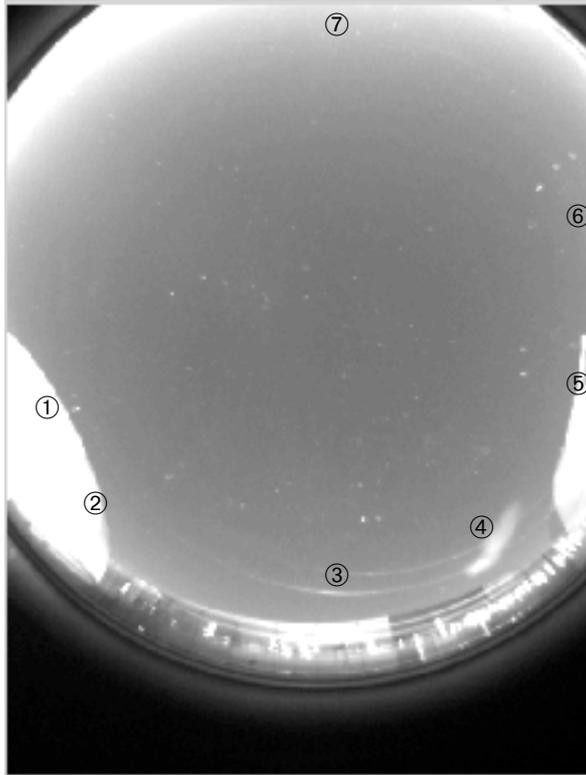
Fig. 5.14   An all-sky image, (a), and the plot of the fluctuation of the RMS of brightness of the sky background against azimuth at airmass = 2, (b). The markers ① to ⑤ show the positions where large RMS was produced. It was caused by the reflected streetlights from:
① & ②. east dome,
③. the DAC dome (double reflection),
④. cloud,
⑤. west dome.
The markers ⑥ & ⑦ show these areas had a scope less than airmass = 2, so the RMS was not plotted for these area in (b).

(a)

(b)
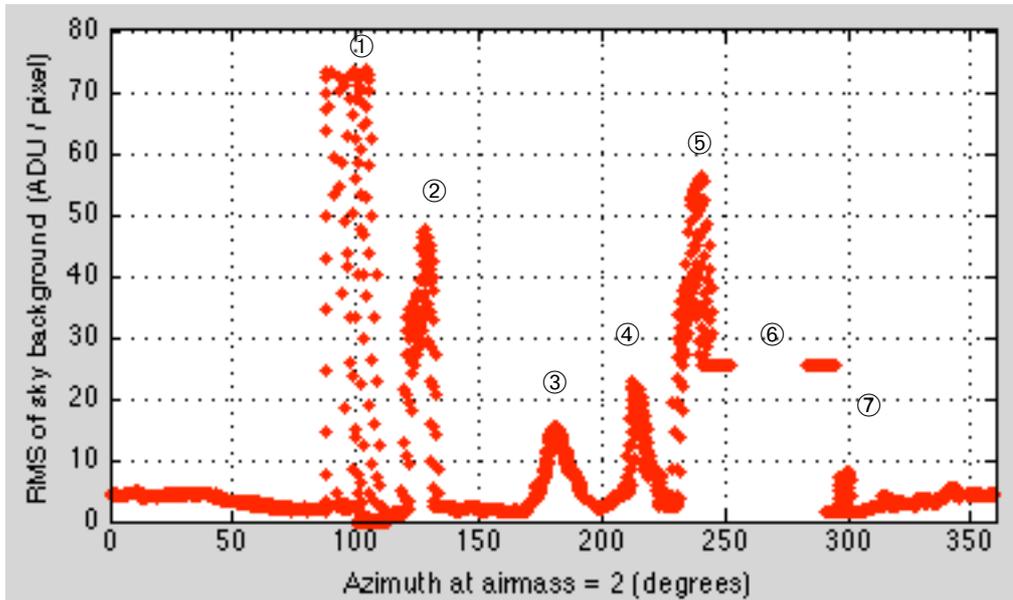
5.7   Durham Sky Brightness Compared with Hawaii

After determining the best degree of accuracy that can be measured by DAC under the Durham sky condition, it is interesting to know what the measurement accuracy would be when the DAC is implemented elsewhere, for example, an observatory on Hawaii island. Since the magnitude accuracy is subject to the sky

brightness, we need to know how bright the Durham sky is comparing with the sky brightness in Hawaii.

The V-band sky brightness in Hawaii was found to be 21.5 magnitudes per square arcsecond ①. For comparison, we need to find the Durham sky brightness in the same unit. Thus, the angular size of a pixel needs to be determined primarily. In the process, the zenith distance per pixel was used to approximate the angular size and it was estimated to be about 0.2 degrees. For an angular size with a good accuracy, we need to take account of image distortion, which results in a scale variation of 4% from centre to edge. However, for our purpose, the 0.2 degrees estimate is adequate. The sky brightness was then calculated in counts per square arcsecond and was converted into catalogue magnitude per square arcsecond by Eq. (5.1 & 5.13).

As a result, the Durham sky brightness was roughly 18 magnitudes per square arcsecond, which was 25 times brighter than the sky in Hawaii.

5.8 Magnitude Accuracy Determined by DAC in Hawaii

To estimate a star's magnitude accuracy in Hawaii, the error of magnitude of the Hawaii sky brightness is needed. Since the sky brightness in Hawaii was 25 times fainter than in Durham, the detected sky pixel counts there would also be 25 times smaller than the sky counts in Durham. Then we can calculate the sky counts in Hawaii and put it into the Eq. (5.5) to find out the error of magnitude. The results are shown in Table 5.5 and Fig. 5.15.

Table 5.5   The table of, (a), the sky brightness and its RMS in Hawaii after being converted to pixel counts, and (b), the magnitude accuracy, that can be determined by DAC, under the sky condition in Hawaii.

Position	Sky brightness (counts per pixel)	RMS (counts per pixel)
At zenith (airmass = 1)	2.64	0.20
At El = 30 degrees (airmass = 2)	3.32	0.23

(a)

Catalogue magnitude	1.0	2.0	3.0	4.0	5.0	6.0
Magnitude uncertainty at zentith	0.005	0.009	0.016	0.04	0.08	0.19
Magnitude uncertainty at el = 30 degrees	0.01	0.02	0.04	0.1	0.24	0.6

(b)

① . The value of the sky brightness in Hawaii was found on Further Measurements of Extinction and Sky Brightness on the Island of Hawaii, by K. Krisciunas.
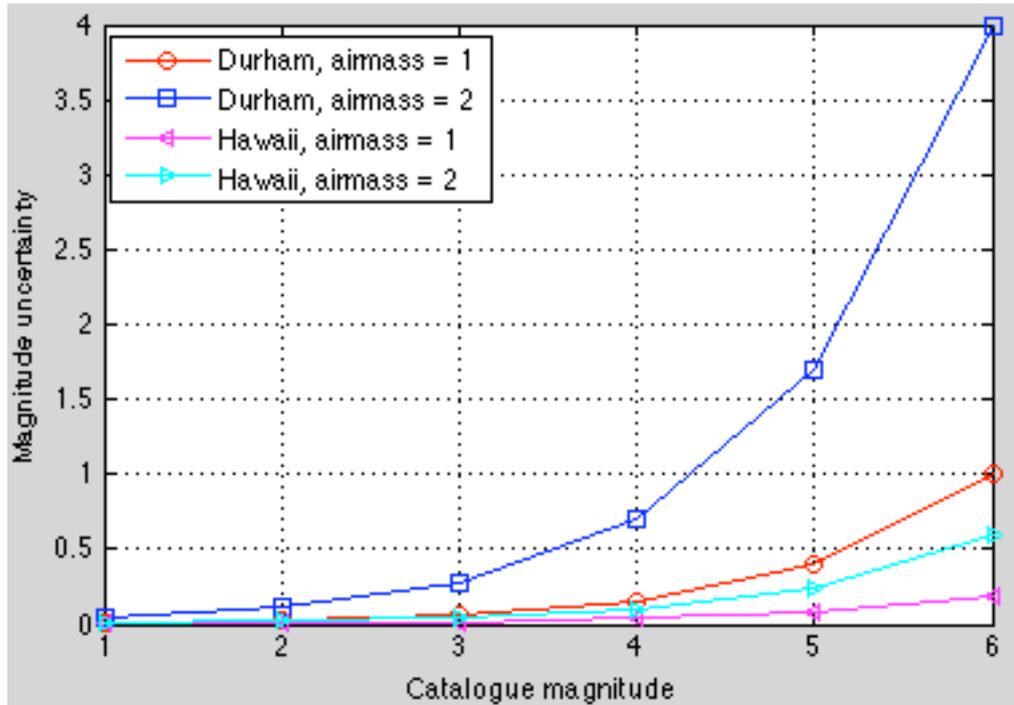
Fig. 5.15   The plot of magnitude accuracy of the stars of 1st to 6th magnitude determined in Hawaii in contrast to in Durham.

Fig. 5.15   The plot of magnitude accuracy of the stars of 1st to 6th magnitude determined in Hawaii in contrast to in Durham.

As the result shows, DAC can determine the 6th magnitude stars within about 0.2 magnitudes accuracy at zenith position and within 0.6 magnitudes at the elevation of 30˚ in Hawaii, which are 7 to 10 times better than the result in Durham.

5.9   Conclusions

In this Chapter we investigated the photometry characteristics of DAC. We empirically determined the value of gain of the camera to be 64 ± 12 electrons per ADU, and demonstrated it was not the gain that dominated the error of a star's magnitude but the sky background. We proved that the method we used to sample a star's flux was reasonable and adequate.

The rate of change of a star's magnitude with airmass was determined to be 1.11 magnitude per airmass. We discussed the reasons why the result was unexpectedly large and gave suggestion for testing. We discussed how to determine a star's apparent magnitude in terms of the instrumental value and established a mathematical relation between them (Eq. (5.13)).

In the end, as a result of all the work that was done previously, it was calculated that, within 1 magnitude of uncertainty, DAC could determine a 6th magnitude star at zenith, a 4th at 30˚ elevation. We demonstrated the performance of DAC would be much better if it was implemented in a darker place like Hawaii where the sky brightness was estimated to be 25 times dimmer than the sky brightness, 18 magnitudes per airmass, in Durham. It may answer the question that raised in Chapter 3 about the reason why less stars being shown on the DAC's image of a clear sky (Fig. 3.1).

Chapter 6

Summary and Future Work

6.1 Summary

Throughout the thesis, various points have been discussed and a number of characteristic features of DAC have been determined. They were summarised as the followings.

1. The reason why the fish-eye lens configuration was preferred as the design of DAC rather than a convex mirror was because the camera suspended above the mirror would block the central area of the sky. According to Durham sky background condition, the central sky area was least contaminated by the streetlights and so an observation at this area was more desirable.

2. The dark currents of the DAC sensor was found to be independent of the camera's operating time but rose exponentially with the increase of the exposure time. The dark currents also rose with the increase of the gain and brightness parameter settings. The analysed results suggest that the exposure time, gain and brightness parameter should be set below 20 seconds, 400 and 120, respectively, in order to achieve a good quality image with acceptable noise characteristics.

3. The gain and brightness parameter on the camera setting program were not clearly defined by the instruction. Further investigation was necessary to understand how they were related to the camera gain that was measured in the unit of the number of electrons per ADU.

4. A hot corner of the sensor was found but it was outside the area covered by the night sky.

5. Seven hot pixels were found. They were at the pixel positions of (368, 317), (422, 11), (56, 30), (382, 332), (87, 402), (224, 471) and (13, 595), respectively.

6. Although the DAC CCD images did not cover all the night sky hemisphere, the area lost was at low elevation where the light pollution from bright street lights saturated the pixels.

7. For a better image, different exposure times, 14, 12 and 10 seconds, were set for imaging at a clear night, a cloudy night, and a night when the moon was in the sky, respectively.

8. The star images distorted along the radial direction of an all-sky image. The distortion was large when a star approached horizon.

9. The image of dusts or hot pixels could cause confusion over stars identification, especially, when a star's image moved close to them.

10. The image position of zenith was determined to be at (256, 251). The uncertainty was 0.8 pixels.

11. The mapping between the celestial hemisphere and an image was achieved empirically. It was tested that the uncertainty was 1 pixel when mapping a celestial point onto an image. The tested results of the reverse process, i.e. from an image point to a celestial point, showed the uncertainties were within 0.2˚ in elevation and 0.7˚ in azimuth.

12. The robustness of mapping was tested. The result showed the mapping was stable at the level of 1 pixel.

13. The gain of the camera was empirically determined to be 64 ± 12 electrons per ADU. It was found the gain was not a dominant source of error of the instrumental magnitudes.

14. The brightness of the sky background in Durham was found to dominate the error of the magnitudes measured by DAC.

15. The rate of change of star magnitudes with respect to the airmass was empirically determined to be 1.11 magnitudes per airmass. Various reasons have been discussed in section 5.5.4. For example, the attenuation of the star flux by the camera lens biased toward the large incident angles.

16. It was found, under the sky background condition of Durham, DAC was able to measure a 6th magnitude star at zenith within 1 magnitude of uncertainty. Within the same uncertainty, DAC was only able to measure a 4th magnitude star when the star's elevation was 30˚.

17. The brightness of sky background of Durham was estimated to be 18 magnitudes per square arcsecond, which was about 25 times brighter than the sky background in Hawaii.

18. It was estimated that if DAC was deployed under the darker sky conditions of Hawaii, stars of 6th magnitude could be measured to an uncertainty of 0.19 magnitudes at the zenith.

6.2 Future Exploitation

The work of investigating or exploiting the prototype of DAC did not finish with the end of the photometry analysis. As a consequence of the previous analyses, there have been many things we can do to maximise the DAC's functions. Below shows two examples, which I felt interesting.

First, we can use DAC to measure the apparent magnitude of a stars simply by knowing its name. What we need to achieve it is basically a combination of the results of the astrometry and photometry analysis written in a computer program for execution. The only thing we need to do manually, apart from writing the programme, is to set up a database for keeping the names of stars and their corresponding RA/Dec. Therefore, after indicating the name of a star, a computer can read the corresponding RA/Dec to find out the star on an image, then calculating the flux and airmass and finally determining the magnitude.

The reason why this is interesting is not because we can obtain the apparent magnitude of a star by a straightforward operation, since it would be much easier if we search the star on the websites. What it suggests is, as a result of the straightforward measurement and by comparing the measured magnitude with the catalogue value, we can monitor the environmental condition around DAC almost instantly. Since it may not be always noticeable when thin clouds drifting across the sky or the water vapours being condensed on the dome, but if the measured magnitude is unexpectedly high, i.e. low in flux, it would suggest their existence. The

following shows an example, which can also be used to test the result of the photometry analysis.

The stars, Kappa UMA and Procyon, were framed at 20:57 on 4 April 2008, and we would like to know their apparent magnitude measured by DAC. Instead of writing a new program to combine the astrometry and photometry results for analysis, we simply executed two previously used programs for each part of calculation. The RA/Dec of each star were typed into the code, and consequently the image positions of the stars were found and marked by the program (Fig. 6.1). After running another program, we obtained the results listed in Table 6.1.

As determined in section 5.6, DAC is able to measure a 3rd magnitude star within 0.06-magnitude uncertainty when the airmass is 1. Therefore, the measured magnitude of Kappa UMA is consistent with the catalogue value. However, the measured magnitude of Procyon is significantly higher than the expected value, 0.4 magnitudes, if we consider DAC can determine a 0th magnitude star within an uncertainty similar to the one when determining a 1st magnitude star, as suggested by Fig. 5.13. When using the annulus RMS to calculate the magnitude uncertainty of Procyon, we got 0.02 magnitudes. This uncertainty is reasonable in comparison with 0.04 magnitudes for a 1st magnitude star (Table 5.4). However, it makes the magnitude of Procyon even more significant. As a result, it suggests there was an unexpected transmission loss of the flux of Procyon though nothing can be seen from the image. After playing a sequence of the frames, from this one to the one at 22:15, it showed something was drifting around the area of Procyon and a cloud was formed on that area after an hour of time (Fig. 6.2). It may not be the true reason why the Procyon was significantly dim but astro-physically it is unlikely that Procyon would change in brightness and hence dimming by clouds is a very likely cause.

Another interesting application of DAC is to help the users to identify the stars by drawing the constellation lines. An example is shown in Fig. 6.3. The mapping functions were applied and the RA/Dec database described previously was also essential to perform the task. The function of plotting the constellation lines on an all-sky image was considered to be particularly useful for the first year undergraduate students to study the positions of stars.
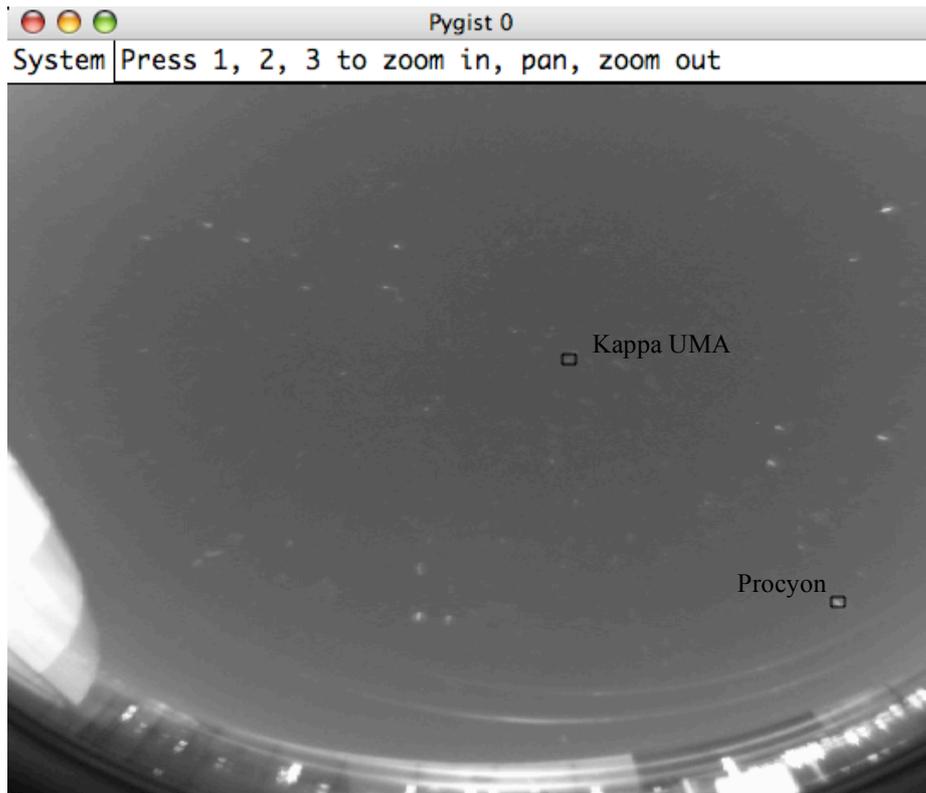
Fig. 6.1 The part of an all-sky image showing the stars of Kappa UMA and Procyon. The image was taken at 20:57 on 4 April 2008. Kappa UMA and Procyon were not previously sampled for the analysis.

Table 6.1 The table lists the measured and catalogue values of the apparent magnitude of the stars. When sampling the flux, different aperture size was used for each star to take account of the image distortion. The catalogue values were found from SKY2000 Catalog, Version 4, from VizieR at http://vizier.u-strasbg.fr/viz-bin/VizieR-4.

Stars	Airmass	Measured Magnitude	Catalogue value
Kappa UMA	1.02	3.3	3.58
Procyon	1.77	0.58	0.4
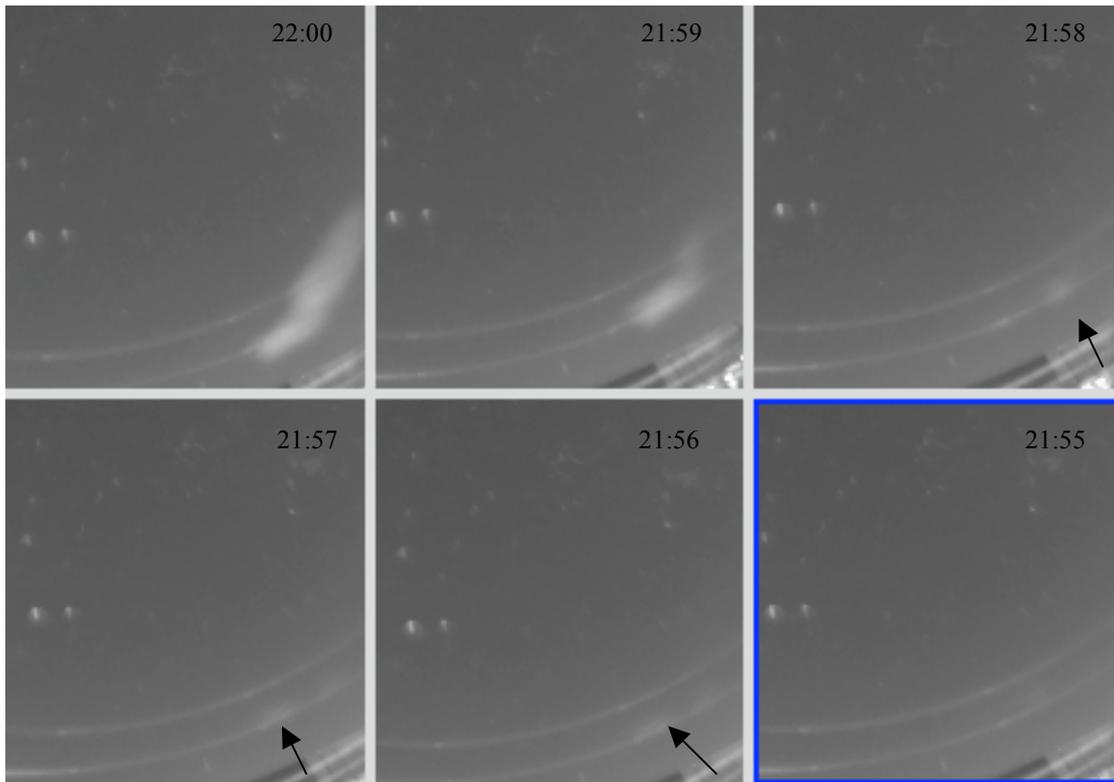
Fig. 6.2 The images showing the formation of a cloud. The arrows point to foggy area, which seemed to be floating when the image were played progressively. All six frames were taken on the same day as Fig. 6.1. The time when they were taken are shown on the top right of each image.
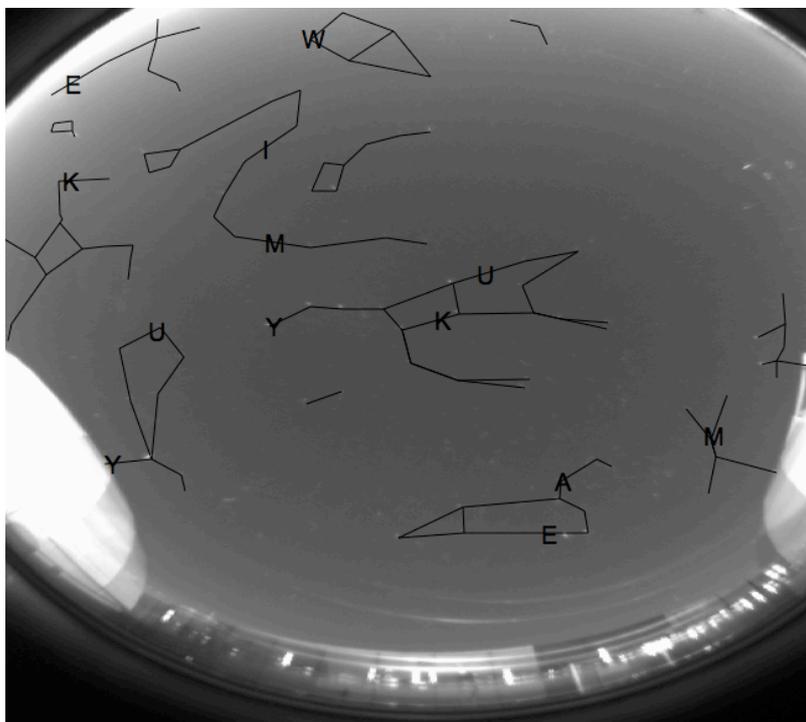
Fig. 6.3 The all-sky image on which the constellation lines were drawn as a result of mapping. The letters on the line are the results of coding. They do not represent anything on the image.

6.3 Further Improvements

There are at least four aspects of DAC that would warrant further investigations. Firstly, the amount of light intensity attenuated by the lens from all possible incident angles should be investigated. It helps to understand the reason why the rate of change of the magnitudes with respect to airmass (section 5.5) had a large experimental result. As already been suggested in section 5.5.4, a laser beam can be used to scan the lens and find out the light losses.

Secondly, the DAC's operation needs to be automated and a webpage should be set up for viewing the live all-sky images. The automation is our final goal for a completed all-sky camera. However, I do not suggest it to be done at last since it would be rather difficult when more and more facilities, like filter, heater, etc., are added to DAC for a new feature. The program that runs DAC would be easily modified for a new feature if the automation is done at an early stage. Setting the webpage and watching the images online are good for monitoring the performance of DAC. Meanwhile, we can collect the ideas from other users for the DAC's further developments.

Thirdly a heater could be added to DAC to prevent the water vapour condensations on the dome or lens. It has been discovered that the heat generated by camera itself was not sufficient to prevent the condensation and the condensed water vapour was clearly observable on a DAC image after 2-hour operation in a wet cold night. When the case of DAC was designed, it was left the space for putting a heater and fan inside. A fan helps to circulate the air inside DAC. We can also use something like a duvet to wrap the case inside or outside to trap the hot air.

Fourthly reducing the influence of the streetlights to DAC during an observation is essential for the improvement of the DAC images. We have demonstrated that the dominant source of error of the instrumental magnitudes of stars in Durham was the brightness of sky background. In Durham, the sky background is marked illuminated by the streetlights, which are the sodium light predominantly emitting a spectral of the wavelength from 589.0 to 589.9 nm when the lamp is at low pressure. Therefore, the filters or other optical elements should be added to the camera to attenuate the streetlights. An example of the light pollution filter is Lumicon Deep Sky Filter①, which is able to block all high and low pressure Mercury and Sodium light.

In addition, the maintenance work of the housing should also be carried. The spider nets have been frequently found inside case of DAC.

---

① . The details are available from the website of Rother Valley Optics Ltd at
http://www.rothevalleyoptics.co.uk/lumicon-deep-sky-filter-fits-all-1-25-eyepices_d907.html.

Appendix A

Design and dimensions of the case of DAC

       The following diagrams are the original plots of the design and dimensioning of the components of the DAC case. According to the order of layout, they are the post, the camera stand, the ring that connects with the dome on the top of the case and the base, respectively. The last figure shows the relative position of the components.

Post for M6 counterbore

Unit: mm

Height: 148

Camera stand
Unit: mm

Four clearance holes
of M6 counterbore

57
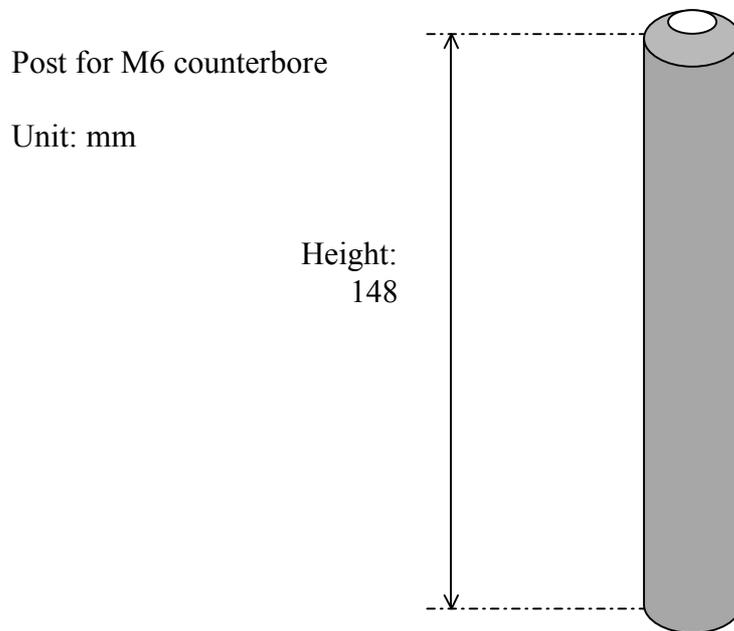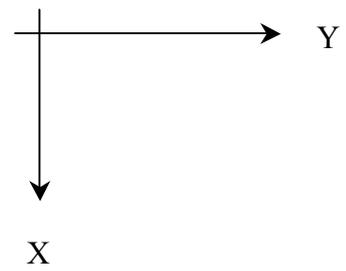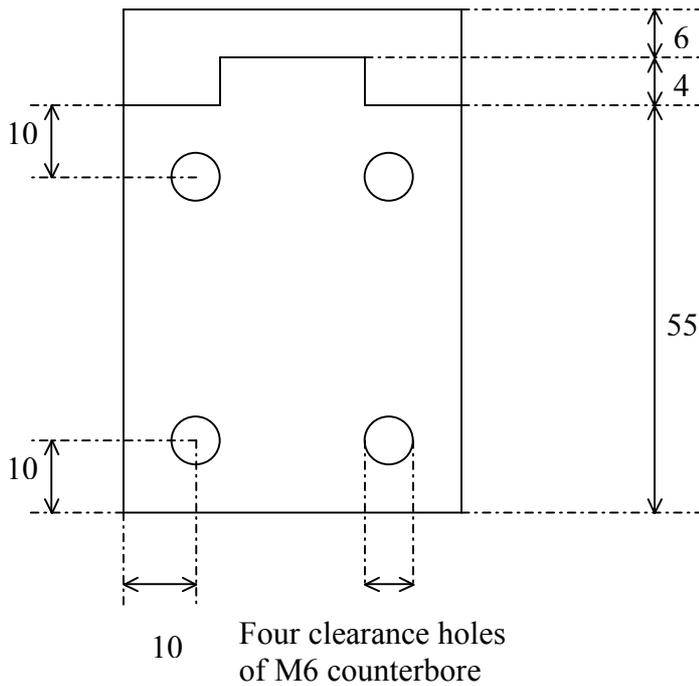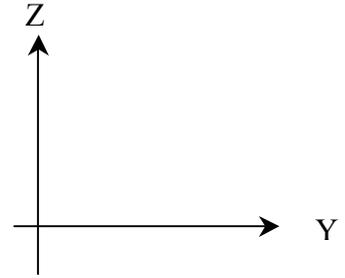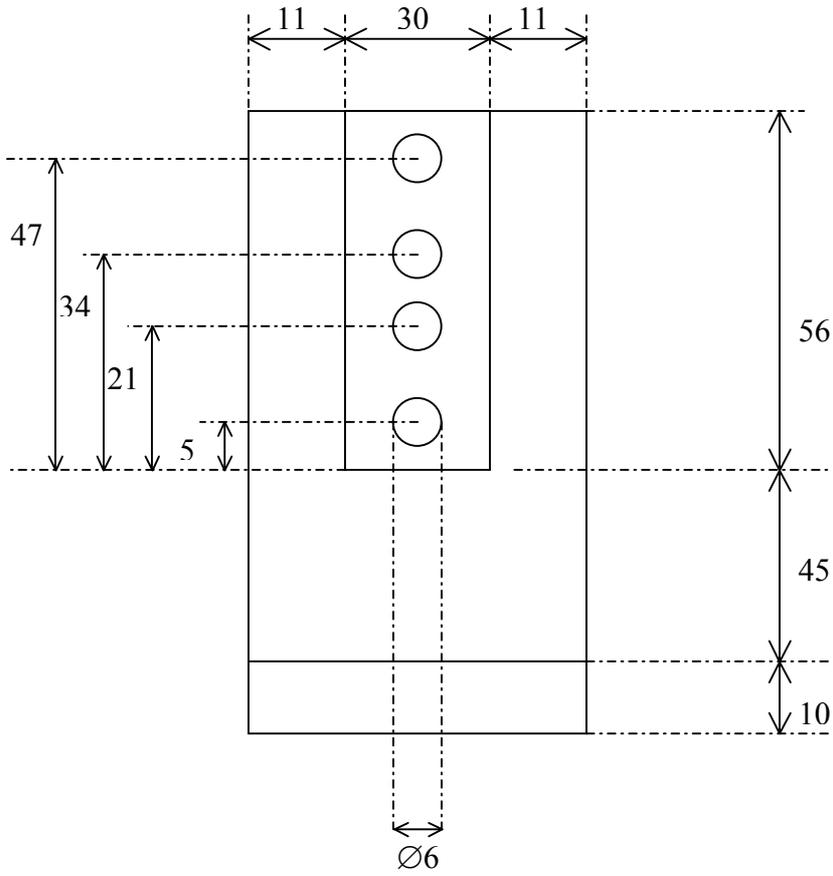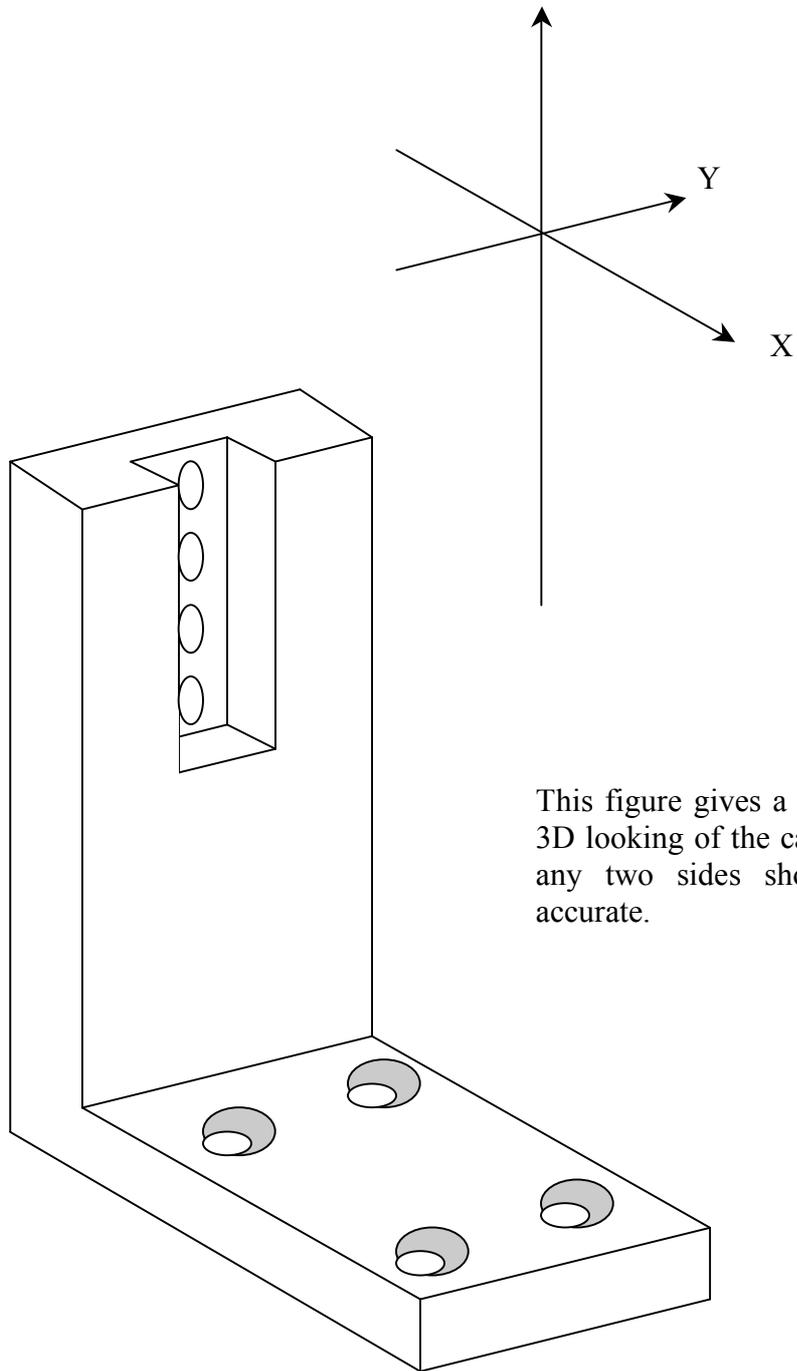
This figure gives a general indication of the 3D looking of the camera stand. The ratio of any two sides shown here may be not accurate.

Z

Y

X

π/4

Clearance
hole of M6
counterbore
(hole A).

87

98

∅ 226

Y

X

Z

X

10

59
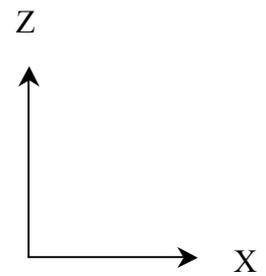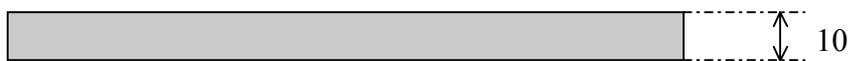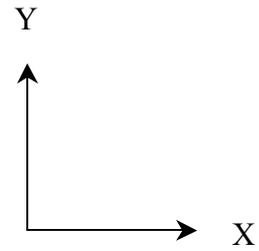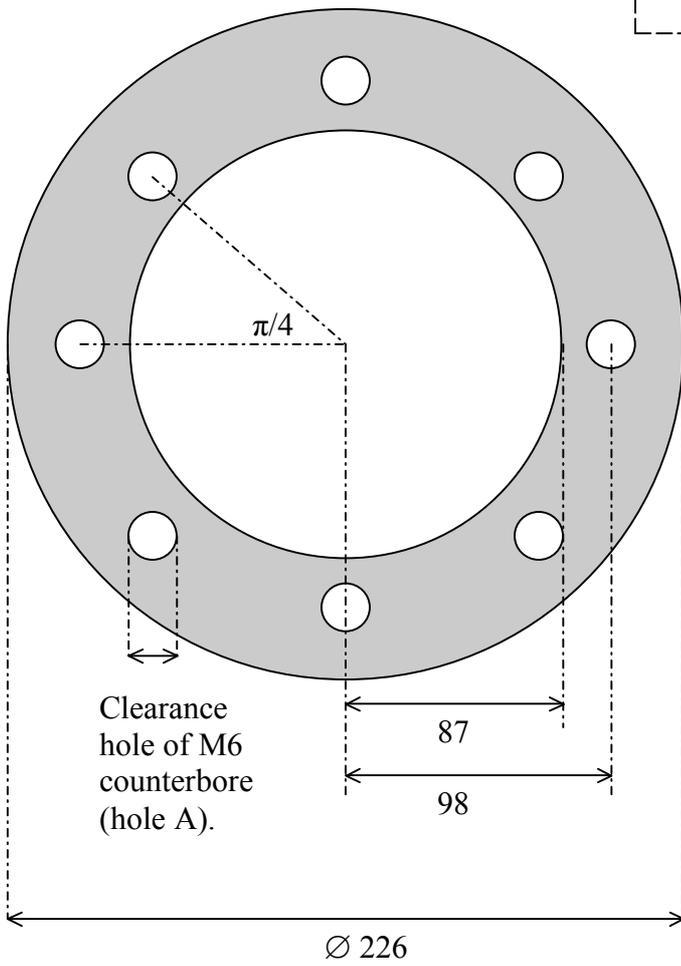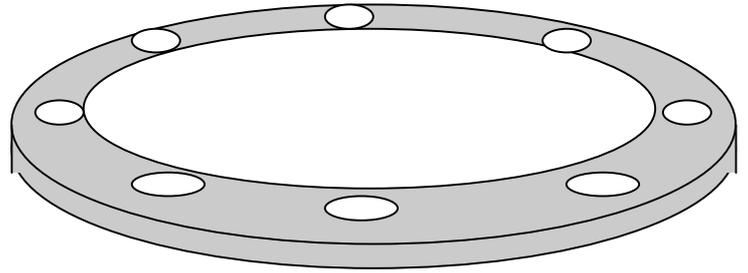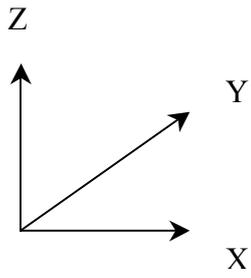
Around the edge are eight clearance holes (hole C) for M6 counterbore.

Near the centre are four M6 thread holes (hole D) for connecting the disk to the camera stand.

98

3

18

19

$\pi/4$

16

$\pi/8$

112

The base
Unit: mm

14   25   16   16

Four M6 thread holes (hole B) close to edge for fixing the disk to a surface.

Radius 120

10
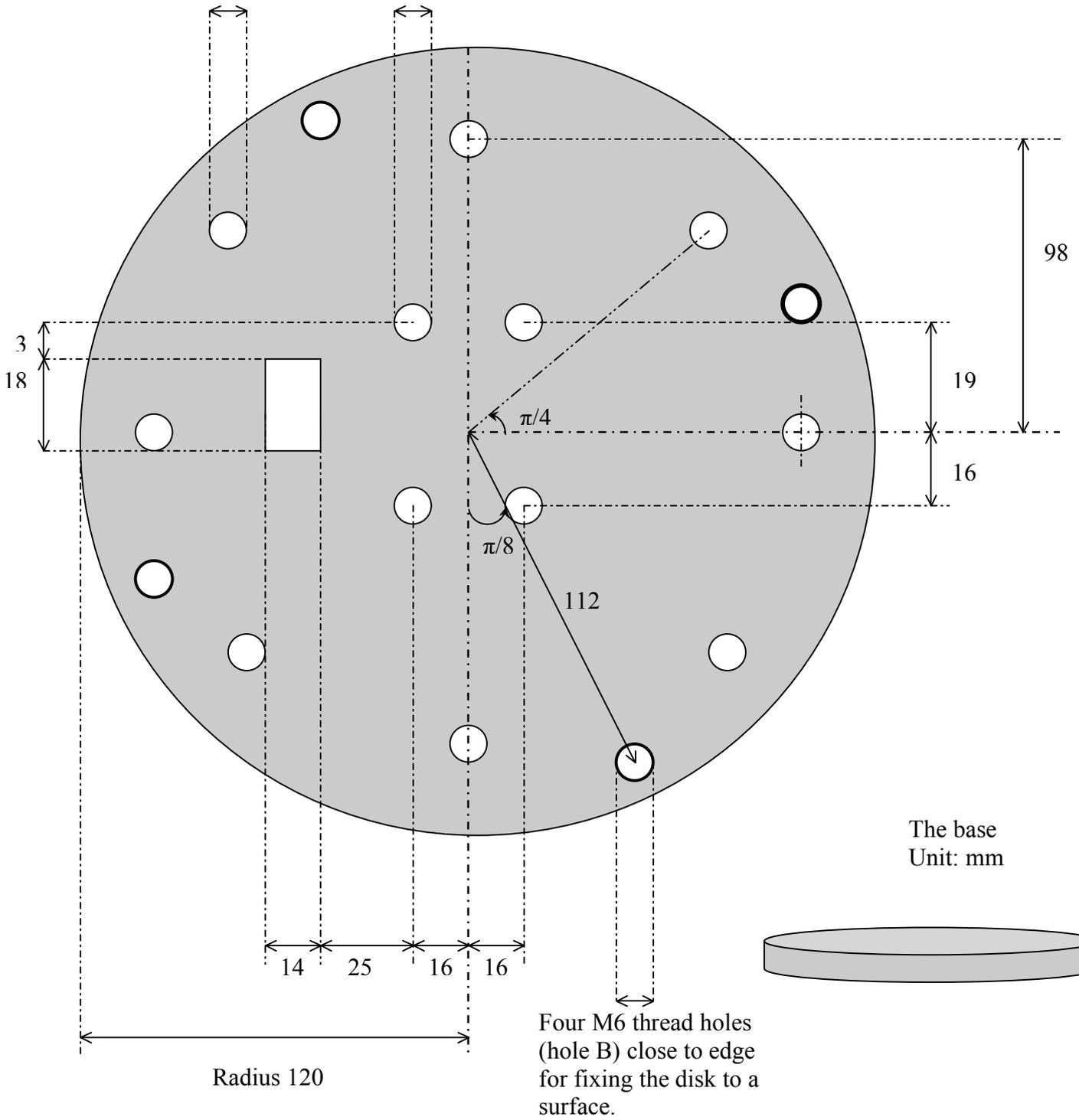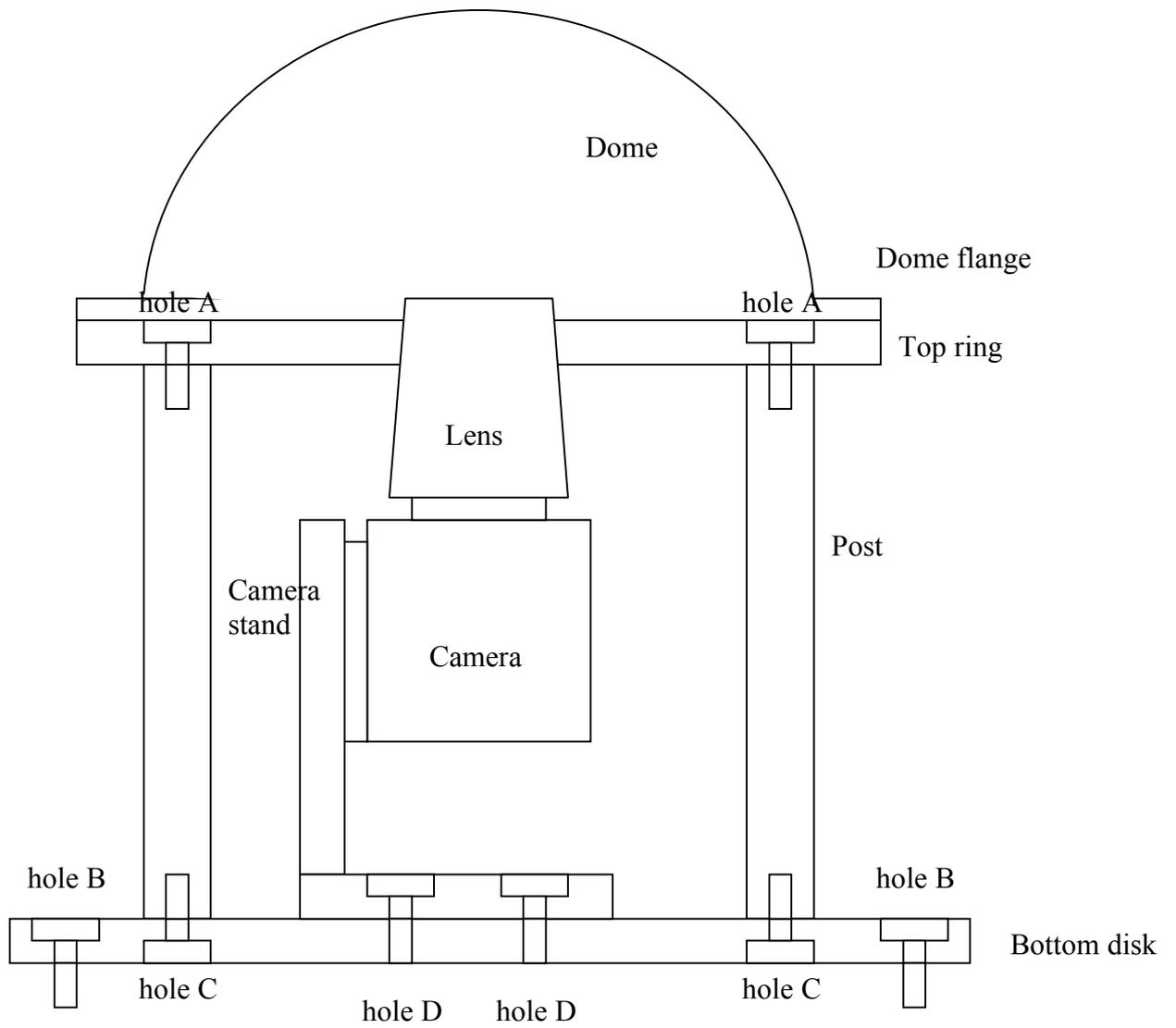
The relative position of the components.

Appendix B

Some computer programs used for the analysis

When analysing the data, more than 20 programs were written in MATLAB or python for computation. Below shows two important ones written for the astrometry and photometry analysis, respectively.

B.1 The Program for Mapping

```matlab
clear all;
disp(' ');
disp('*****************');
%First part is to get Julian day number and hence ST.
x = load('x6.txt'); %load the pixel positions of stars
y = load('y6.txt');
RA_hrs = load('RA_hrs.txt'); %load sky positions
RA_mins = load('RA_mins.txt');
RA_secs = load('RA_secs.txt');
Dec_hrs = load('Dec_hrs.txt');
Dec_mins = load('Dec_mins.txt');
Dec_secs = load('Dec_secs.txt');
disp('Input a new date and time?');
temp = input('y or n: ','s');
if temp == 'n'
    Y = 2008;
    M = 4;
    D = 4;
    ut_hours = 21 + 46/60 + 0/3600;
elseif temp == 'y'
    Y = input('Year (i.e. 2007) = ');
    M = input('Month = ');
    D = input('Day = ');
    H = input('Hour (i.e. 16) = ');
    Min = input('Minute = ');
    %ut_hours = H + Min/60 + 30/3600;
    ut_hours = H + Min/60 + 0/3600;
end
if M <= 2
    Y = Y-1;
    M = M+12;
end
a = floor(Y/100);
b = 2 - a + floor(a/4);
if Y < 0
    c = floor(365.25*Y - 0.75);
else
    c = floor(365.25*Y);
```

62

```matlab
    end
d = floor(30.6001*(M+1));
jd = b + c + d + D + 1720994.5;
t = (jd - 2451545)/36525;
t0 = 6.697374558 + (2400.051336*t) + (0.000025862*(t*t));
t0 = (t0 - floor(t0/24)*24);
if t0 < 0
    t0 = t0+24;
end
ut = 1.002737909*ut_hours;
tmp = floor((ut + t0)/24);
gst = ut + t0 - tmp*24;
gst_hour = floor(gst);
gst_min = floor((gst - gst_hour)*60);
gst_sec = floor((gst - gst_hour - gst_min/60)*3600);
disp(' ');
disp(['ST = ',num2str(gst_hour),' hrs']);
disp(['      ',num2str(gst_min),' mins']);
disp(['      ',num2str(gst_sec),' secs']);
%%
%This 2nd part is for getting LST.
Durham_Long_Deg = 1.583;
Durham_Lat_Deg = 54.767;
LST = gst - Durham_Long_Deg/15;
if LST > 24
    LST = LST - 24;
end
if LST < 0
    LST = LST + 24;
end
disp(' ');
disp(['LST = ',num2str(LST),' hrs']);
%%
%This 3rd part is doing the conversion from RA/Dec to Az/El
disp(' ');
phi = Durham_Lat_Deg;
disp('Use the imported RA/Dec?');
temp = input('y or n: ','s');
if temp == 'n'
    disp('Input the RA/Dec of the star:');
    RA_hrs_Single = input('  RA hrs = ');
    RA_mins_S = input('  RA mins = ');
    RA_secs_S = input('  RA secs = ');
    Dec_hrs_Single = input('  Dec hrs = ');
    Dec_mins_S = input('  Dec mins = ');
    Dec_secs_S = input('  Dec secs =');
    RA_S = RA_hrs_Single + RA_mins_S/60 + RA_secs_S/3600;
    ha_S = LST - RA_S;
    Dec_S = Dec_hrs_Single + Dec_mins_S/60 + Dec_secs_S/3600;
    ha_S = 15*ha_S;
    sh = sind(ha_S);
    ch = cosd(ha_S);
    sd = sind(Dec_S);
    cd = cosd(Dec_S);
    sp = sind(phi);
    cp = cosd(phi);
    x_factor = -1*ch*cd*sp + sd*cp;
    y_factor = -1*sh*cd;
    z_factor = ch*cd*cp + sd*sp;
    r = sqrt(x_factor.^2 + y_factor.^2);
    if r == 0
        a = 0;
    else
```

```matlab
            a = atan2(y_factor,x_factor) * 180/pi;
        end
        if a < 0
            a = a + 360;
        end
        az_S = a;
        el_S = atan2(z_factor,r) * 180/pi;
        disp(' ');
        disp(['az = ',num2str(az_S),' degrees']);
        disp(['el = ',num2str(el_S),' degrees']);
    elseif temp == 'y'
        disp(' ');
        disp('No.  x,    y,
RA_hrs,RA_mins,RA_secs,Dec_hrs,Dec_mins,Dec_secs');
        for i = 1:length(RA_hrs)
            disp([num2str(i),'    ',num2str(x(i)),'  ',num2str(y(i)), ...
                  '     ',num2str(RA_hrs(i)),'      ',num2str(RA_mins(i)),...
                  '     ',num2str(RA_secs(i)),'
',num2str(Dec_hrs(i)),...
                  '          ',num2str(Dec_mins(i)),'
',num2str(Dec_secs(i))]);
        end
        disp(' ');
        disp('No.   x        y     Az(degrees)      El(degrees)');
        RA = RA_hrs + RA_mins/60 + RA_secs/3600;
        ha = LST - RA;
        Dec = Dec_hrs + Dec_mins/60 + Dec_secs/3600;
        ha = 15*ha;
        sh = sind(ha);
        ch = cosd(ha);
        sd = sind(Dec);
        cd = cosd(Dec);
        sp = sind(phi);
        cp = cosd(phi);
        x_factor = -1*ch.*cd.*sp + sd.*cp;
        y_factor = -1*sh.*cd;
        z_factor = ch.*cd.*cp + sd.*sp;
        r = sqrt(x_factor.^2 + y_factor.^2);
        if r == 0
            a = 0;
        else
            a = atan2(y_factor,x_factor) * 180/pi;
        end
        for i = 1:length(a)
            if a(i) < 0
                a(i) = a(i) + 360;
            end
        end
        az = a;
        el = atan2(z_factor,r) * 180/pi;
        for i = 1:length(RA_hrs)
            disp([num2str(i),'     ',num2str(x(i)),'      ',num2str(y(i)),
...
                  '      ',num2str(az(i)),'        ',num2str(el(i))]);
        end
    end
end
disp(' ');
clear x_factor y_factor z_factor r sh ch sd cd sp cp a ha_S el_S az_S
Dec_S Dec_hrs_Single Dec_mins_S Dec_secs_S RA_hrs_Single RA_mins_S
RA_secs_S RA_S b c d temp tmp phi t t0 ut_hours gst gst_hour gst_min
gst_sec ut Dec RA ha i x4 y4 Y M D;

%%
```

```matlab
%To put El in order
disp(' ');
disp('Arrange El in ascending order:');
disp(' ');
temp = 0;
tempx = 0;
tempy = 0;
n = length(el) - 1;
for j = 1:n
    for i = 1:n
        if el(i) > el(i+1)
            temp = el(i);
            tempx = x(i);
            tempy = y(i);
            tempAz = az(i);
            tempRA_hrs = RA_hrs(i);
            tempRA_mins = RA_mins(i);
            tempRA_secs = RA_secs(i);
            tempDec_hrs = Dec_hrs(i);
            tempDec_mins = Dec_mins(i);
            tempDec_secs = Dec_secs(i);
            el(i) = el(i+1);
            x(i) = x(i+1);
            y(i) = y(i+1);
            az(i) = az(i+1);
            RA_hrs(i) = RA_hrs(i+1);
            RA_mins(i) = RA_mins(i+1);
            RA_secs(i) = RA_secs(i+1);
            Dec_hrs(i) = Dec_hrs(i+1);
            Dec_mins(i) = Dec_mins(i+1);
            Dec_secs(i) = Dec_secs(i+1);
            el(i+1) = temp;
            x(i+1) = tempx;
            y(i+1) = tempy;
            az(i+1) = tempAz;
            RA_hrs(i+1) = tempRA_hrs;
            RA_mins(i+1) = tempRA_mins;
            RA_secs(i+1) = tempRA_secs;
            Dec_hrs(i+1) = tempDec_hrs;
            Dec_mins(i+1) = tempDec_mins;
            Dec_secs(i+1) = tempDec_secs;
        end
    end
end
disp(' ');
disp('No.   x       y    Az(degrees)     El(degrees)');
for i = 1:length(el)
    disp([num2str(i),'     ',num2str(x(i)),'     ',num2str(y(i)), ...
        '     ',num2str(az(i)),'        ',num2str(el(i))]);
end
clear n i j temp tempx tempy tempAz tempDec_hrs tempDec_mins
tempDec_secs tempRA_hrs tempRA_mins tempRA_secs

%%
for NextStage = 1:1
    disp(' ');
    disp('Do you want to continue the program?');
    ProgramContinue = input('Type "n" for no: ','s');
    if ProgramContinue == 'n'
        break;
    end
%%
%Finding out the pixel position of zenith:
```

```matlab
    Z = 90 - el;
    %*** 1.1. INPUT
    disp(' ');
    disp('START *********************');
    Xcinput = input('Centre X =  ');
    %Xcinput is the assumed centre pixel value of x.
    Xi = input('Increment value of X =  ');
    %Xi is minimum step of movement of centre along +ve x direction.
    if Xi == 0
        n = 1; %return n = 1 not 0 is for the loop below
    else
        n = input('Number of increments of Xi =  ');
        %n is the number of steps, i.e. number of Xi or Yi.
        if n == 0
            n = 1;
        end
    end
    XcFirst = Xcinput;
    Yc = input('Centre Y =  '); %assumed centre pixel value of y.
    Yi = input('Increment value of Y =  ');
    if Yi == 0
        ny = 1; %ny = 1 is for the loop used below
    else
        ny = input('Number of increments of Yi =  ');
        if ny == 0
            ny = 1;
        end
    end
    YcFirst = Yc;
    dop = input('Degree of polyfit =  ');
    %*** 1.2. DETERMINE SUBPLOT'S COLUMN & ROW(k) NUMBER
    for k = 1:100
        if k >= n/5
            break;
        end
    end
    row = k;
    if k == 1
        column = n;
    else
        column = 5;
    end
    %k gives the number of rows for a subplot used below.
    %*** 1.3. DETERMINE IF DISPLAY EACH RMS
    disp(' ');
    disp('Press "2" to display the RMS of zenith distance of all
figures;');
    disp('press "1" to display the RMS of pixel distance of all
figures;');
    disp('press "0" to get the smallest RMS for both directly, .');
    disprms = input('Your choice: ');
    for temp  = 1:50
        if (disprms == 2 || disprms == 1 || disprms == 0)
            break;
        else
            disp(' ');
            disp('Press "2" to display all the RMS of zenith
distance;');
            disp('press "1" to display all the RMS of pixel distance
;');
            disp('press "0" to get the smallest RMS for both
directly, .');
            disprms = input('Your choice: ');
```

```matlab
        end
    end
    %*** 1.4. DETERMINE IF PLOT THE GRAPHS
    disp(' ');
    disp('Do you want to have the graphs plotted?');
    dispfig = input('Input "1" for yes, "0" for no: ');
    for temp  = 1:50
        if (dispfig == 1 || dispfig == 0)
            break;
        else
            dispfig = input('Input "1" for yes, "0" for no: ');
        end
    end
    %*** 2. START CACULATIONS
    %*** 2.1. LOOP j FOR CHANGE OF Yc
    for j = 1 : ny
        if (disprms == 1 || disprms == 2)
            disp(' ');
            disp(num2str(Yc));
        end
        if dispfig == 1
            figure;
        end
        Xc = Xcinput;
        %Below for scanning the all points with the determined pixel
range.
        %*** 2.2. LOOP i FOR CHANGE OF Xc
        for i = 1 : n
            %*** 2.3. PIXEL DIST CALCULATION
            dx = x - Xc;
            %Having been imported, x is the stars'
            %pixel positions along x-axis.
            dy = y - Yc;
            %y along y-axis.
            P = sqrt(dx.^2 + dy.^2); %P is the Pixel distance.
            %*** 2.4. RMS OF PIXEL DIST CALCULATION
            %rms calcuation begins
            f = polyfit(P,Z,dop);
            g = polyfit(Z,P,dop);
            %*** RMS of Pixel Distance
            dopInLoop = dop;
            Pb = 0;   %Initialise Pb
            for temp = 1 : (dopInLoop+1)
                PbTerm = g(temp)*Z.^(dopInLoop);
                Pb = PbTerm + Pb;
                dopInLoop = dopInLoop - 1;
            end
            %
            Pd = Pb - P; % Pd = deviation of pixel dist
            SumPdSquared = 0;
            for h = 1:length(Pd)
                SumPdSquared = Pd(h).^2+SumPdSquared;
                %Pvariance = SumPdSquared/length(Pd); %variance for P
            end
            Prms = sqrt(SumPdSquared/(length(Pd)-(dop+1))); %RMS for
P
            if disprms == 1
                disp(['Prms =  ',num2str(Prms), ' pixels']);
            end
            %*** RMS of Zenith Distance

            %***test begins
            dopInLoop = dop;
```

```matlab
                Zb = 0; %initialise Zb
                for temp = 1 : (dopInLoop+1)
                    ZbTerm = f(temp)*P.^(dopInLoop);
                    Zb = ZbTerm + Zb;
                    dopInLoop = dopInLoop - 1;
                end
                %***test ends
                Zd = Zb - Z; % Zd = deviation of zenith dist
                SumZdSquared = 0;
                for h = 1:length(Zd)
                    SumZdSquared = Zd(h).^2+SumZdSquared;
                    %Zvariance = SumZdSquared/length(Zd); %variance for Z
                end
                Zrms = sqrt(SumZdSquared/(length(Zd)-(dop+1))); %RMS for
Z
                if disprms == 2
                    disp(['Zrms =  ',num2str(Zrms), ' degrees']);
                end
                %rms ends
                %*** 2.5.FIND THE SMASLLEST RMS OF EACH FIGURE AND OF ALL
                %compare begins
                %*** Prms comparison for finding the smallest.
                if i > 1;
                    compareP = Psmaller - Prms;
                    if compareP > 0
                        Psmaller = Prms;
                            %% what if compare = 0?????????/
                        XspInGroup = Xc;
                        YspInGroup = Yc;
                        PPolyCoeffofGroup = g; %Pixel distance is on y
axis.
                    end
                else
                    Psmaller = Prms;
                    XspInGroup = Xc;
                    YspInGroup = Yc;
                    PPolyCoeffofGroup = g;
                end
                if j > 1;
                    FindSmallestPrms = Psmallest - Prms;
                    if FindSmallestPrms > 0
                        Psmallest = Prms;
                        Xsp = Xc;
                        Ysp = Yc;
                        PPolyCoeff = g;
                    end
                else
                    Psmallest = Psmaller;
                    Xsp = XspInGroup;
                    Ysp = YspInGroup;
                    PPolyCoeff = PPolyCoeffofGroup;
                end
                %*** Zrms comparison for finding the smallest.
                if i > 1;
                    compareZ = Zsmaller - Zrms;
                    if compareZ > 0
                        Zsmaller = Zrms;
                        XszInGroup = Xc;
                        YszInGroup = Yc;
                        ZPolyCoeffofGroup = f;
                    end
                else
                    Zsmaller = Zrms;
```

```matlab
                XszInGroup = Xc;
                YszInGroup = Yc;
                ZPolyCoeffofGroup = f;
            end
            if j > 1;
                FindSmallestZrms = Zsmallest - Zrms;
                if FindSmallestZrms > 0
                    Zsmallest = Zrms;
                    Xsz = Xc;
                    Ysz = Yc;
                    ZPolyCoeff = f;
                end
            else
                Zsmallest = Zsmaller;
                Xsz = XszInGroup;
                Ysz = YszInGroup;
                ZPolyCoeff = ZPolyCoeffofGroup;
            end
            %*** 2.6.PLOT THE GRAPHS
            if dispfig == 1
                if n <= 25
                    subplot(row,column,i); %******for subplot
                    plot (P,Z,'-xr');
                    ylabel('Zenith distance (degrees)');
                    xlabel('Pixel distance    (pixels)');
                    title({['Centre (', num2str(Xc), ', ',
num2str(Yc),...
                        '); no of points: ',num2str(length(Z))];...
                    ['Prms = ',num2str(Prms), ' pixels'];...
                    ['Zrms = ',num2str(Zrms),' degrees.'];...
                    ['degree of Polynomial: ',num2str(dop),')']});
                    grid on;
                    axis([0 300 0 80]);
                    set(gca, ...
                        'XTick',0:50:300, ...
                        'YTick',0:20:80, ...
                        'XMinorTick', 'on', ...
                        'YMinorTick', 'on', ...
                        'XMinorGrid', 'on', ...
                        'YMinorGrid', 'on', ...
                        'GridLineStyle','-',...
                        'MinorGridLineStyle','--')

                end
            end
            XcLast = Xc;
            Xc = Xc + Xi;
        end %END OF PIXEL DISTANCE
        %*** 2.7. DISPLAY SMALLEST RMS OF EACH FIGURE
        if (n ~= 1 && disprms == 1)
            disp(['The smallest Prms of figure [',num2str(j),'] =
',...
                num2str(Psmaller),' pixels,']);
            disp(['of which the centre is (',num2str(XspInGroup),...
                ', ',num2str(YspInGroup),').']);
            disp(['The smallest Prms of all figures=
',num2str(Psmallest),...
                ' pixels.']);
            disp(['end of ',num2str(Yc)]);
            disp(' ')
        end
        if (n ~= 1 && disprms == 2)
            disp(['The smallest Zrms of figure [',num2str(j),'] =
```

```matlab
        ',...
                      num2str(Zsmaller),' degrees,']);
                  disp(['of which the centre is (',num2str(XszInGroup),...
                      ', ',num2str(YszInGroup),').']);
                  disp(['The smallest Zrms of all figures=
        ',num2str(Zsmallest),...
                      ' degrees.']);
                  disp(['end of ',num2str(Yc)]);
                  disp(' ')
            end
            %*** BACK TO Yc
            YcLast = Yc;
            Yc = Yc + Yi;
        end %END OF ZENITH PROJECTION DETERMINATION
        %*** 3. OUTPUT
        disp(' ');
        disp(' ');
        disp('>>>Summary');
        disp('>>>The scanning range of the centre points (in pixels): ');
        disp(['    Xc from ', num2str(XcFirst),' to ',num2str(XcLast)]);
        disp(['   increment: ',num2str(Xi)]);
        disp(['    Yc from ', num2str(YcFirst),' to ',num2str(YcLast)]);
        disp(['   increment: ',num2str(Yi)]);
        disp(['>>>The smallest Prms = ',num2str(Psmallest), ' pixels
        which']);
        disp(['   appears at the centre (',num2str(Xsp),',
        ',num2str(Ysp),').']);
        disp(['>>>The smallest Zrms = ',num2str(Zsmallest), ' degrees
        which']);
        disp(['   appears at the centre (',num2str(Xsz),',
        ',num2str(Ysz),').']);
        disp(['>>>Degree of polynomial: ',num2str(dop)]);
        disp(['   Number of sampled points(stars):
        ',num2str(length(Z))]);
        disp(' --When Zenith distance is the variable,');%Assumed that
        variables
                                          %have uncertainties too
        small to be
                                          %considered.
        disp('   at smallest Prms, the polynommial coefficients are:');
        disp(' ');
        disp(num2str(PPolyCoeff));
        disp(' ');
        disp(' --When Pixel distance is the variable,');
        disp('   at smallest Zrms, the polynommial coefficients are:');
        disp(' ');
        disp(num2str(ZPolyCoeff));
        disp(' ');
        disp(['>>>Yc = ',num2str(Yc)]); %test value; should be deleted in
        the end.
        disp(' ');
        clear FindSmallestPrms FindSmallestZrms PPolyCoeffofGroup;
        clear Pb PbTerm Pd Prms Psmaller SumPdSquared SumZdSquared;
        clear XcFirst XcLast Xcinput Xi Xc XspInGroup ZPolyCoeffofGroup
    Zb Zd ZbTerm Zrms Zsmaller column compareP compareZ dispfig disprms
    dopInLoop dx dy f g h i j k n ny row temp Yc YcFirst YcLast Yi
    YspInGroup YszInGroup Zb Zd ZbTerm Zrms Zsmaller XszInGroup;

    %%
        disp(' '); %Asking if continue for zero-azimuth analysis.
        disp('Do you want to continue the program?');
        ProgramContinue = input('Type "n" for n: ','s');
        if ProgramContinue == 'n'
```

```matlab
            break;
    end
    %To put the data in the accending order of Az.
    disp(' ');
    disp('Arrange Az in ascending order:');
    disp(' ');
    Az = az;
    temp = 0;
    tempx = 0;
    tempy = 0;
    n = length(Az) - 1;
    for j = 1:n
        for i = 1:n
            if Az(i) > Az(i+1)
                temp = el(i);
                tempx = x(i);
                tempy = y(i);
                tempAz = Az(i);
                tempRA_hrs = RA_hrs(i);
                tempRA_mins = RA_mins(i);
                tempRA_secs = RA_secs(i);
                tempDec_hrs = Dec_hrs(i);
                tempDec_mins = Dec_mins(i);
                tempDec_secs = Dec_secs(i);
                el(i) = el(i+1);
                x(i) = x(i+1);
                y(i) = y(i+1);
                Az(i) = Az(i+1);
                RA_hrs(i) = RA_hrs(i+1);
                RA_mins(i) = RA_mins(i+1);
                RA_secs(i) = RA_secs(i+1);
                Dec_hrs(i) = Dec_hrs(i+1);
                Dec_mins(i) = Dec_mins(i+1);
                Dec_secs(i) = Dec_secs(i+1);
                el(i+1) = temp;
                x(i+1) = tempx;
                y(i+1) = tempy;
                Az(i+1) = tempAz;
                RA_hrs(i+1) = tempRA_hrs;
                RA_mins(i+1) = tempRA_mins;
                RA_secs(i+1) = tempRA_secs;
                Dec_hrs(i+1) = tempDec_hrs;
                Dec_mins(i+1) = tempDec_mins;
                Dec_secs(i+1) = tempDec_secs;
            end
        end
    end
    disp(' ');
    disp('No.   x       y    Az(degrees)      El(degrees)');
    for i = 1:length(el)
        disp([num2str(i),'    ',num2str(x(i)),'    ',num2str(y(i)),
...
            '    ',num2str(Az(i)),'        ',num2str(el(i))]);
    end
    disp('end');
    clear n i j temp tempx tempy tempAz tempDec_hrs tempDec_mins
tempDec_secs tempRA_hrs tempRA_mins tempRA_secs
%%
    if ProgramContinue == '1'
        break;
    end
%%
    % Finding the best approximate zero-azimuth point.
```

```matlab
disp(' ');
disp('**********');
Xofirst = input('Xo first = ');
Xolast = input('Xo last = ');
Yofirst = input('Yo first = ');
Yolast = input('Yo last = ');
disp(' ');
disp('Type "2" for g = polyfit(Az,theta,1);');
disp('     "1" for f;');
fit = input('Your choice: ');
for temp  = 1:50
    if (fit == 2 || fit == 1)
        break;
    else
        disp(' ');
        disp('Type "2" for g');
        disp('     "1" f;');
        fit = input('Your choice: ');
    end
end
disp(' ');
disp('Type "1" to display RMS for each set of data;');
disp('     "0" to ignore;');
disprms = input('Your choice: ');
for temp  = 1:50
    if (disprms == 0 || disprms == 1)
        break;
    else
        disp(' ');
        disp('Type "1" to display all RMS;');
        disp('     "0" to ignore;');
        disprms = input('Your choice: ');
    end
end
disp(' ');
disp('Type "1" to display graphs;');
disp('     "0" to ignore;');
dispfig = input('Your choice: ');
for temp  = 1:50
    if (dispfig == 0 || dispfig == 1)
        break;
    else
        disp(' ');
        disp('Type "1" to display graphs;');
        disp('     "0" to ignore;');
        dispfig = input('Your choice: ');
    end
end
Xc = Xsp;
Yc = Ysp;
theta = x - x;
%*** 1.2. DETERMINE SUBPLOT'S COLUMN & ROW(k) NUMBER
n = Xolast - Xofirst + 1;
for k = 1:100
    if k >= n/5
        break;
    end
end
row = k;
if k == 1
    column = n;
else
    column = 5;
```

```matlab
        end
    %k gives the number of rows for a subplot used below.
    for Yo = Yofirst:1:Yolast
        if disprms == 1
            disp(' ');
            disp(num2str(Yo));
        end
        if dispfig == 1
            figure;
        end
        j = 0; %j is used for subplot
        for Xo = Xofirst:1:Xolast
            j = j + 1; %j for subplot
            a = sqrt((Xo - Xc).^2+(Yo - Yc).^2);
            b = sqrt((x - Xc).^2+(y - Yc).^2);
            c = sqrt((x - Xo).^2+(y - Yo).^2);
            for i = 1:length(b);
                EachTheta = acosd((a.^2+b(i).^2-c(i).^2)/(2*a*b(i)));
                if Az(i) > 180
                    EachTheta = 360 - EachTheta;
                end
                theta(i) = EachTheta;
            end
            %*** Finding the RMS of theta
            thetaD = Az - theta;
            Sum = 0;
            for temp = 1:length(thetaD)
                Sum = thetaD(temp).^2+Sum;
            end
            thetaRMS = sqrt(Sum/(length(thetaD)-2)); %RMS for theta
            if disprms == 1
                disp(['RMS(theta) =  ',num2str(thetaRMS), '
degrees']);
            end
            if Xo > Xofirst
                if (SmallerRMS - thetaRMS > 0)
                    SmallerRMS = thetaRMS;
                    XoInGroup = Xo;
                    YoInGroup = Yo;
                end
            else
                SmallerRMS = thetaRMS;
                XoInGroup = Xo;
                YoInGroup = Yo;
            end
            if Yo > Yofirst
                if (SmallestRMS - thetaRMS > 0)
                    SmallestRMS = thetaRMS;
                    Xos = Xo;
                    Yos = Yo;
                end
            else
                SmallestRMS = SmallerRMS;
                Xos = XoInGroup;
                Yos = YoInGroup;
            end
            if dispfig == 1
                subplot(row,column,j);
                plot (theta,Az,'-ro');
                ylabel('Azimuth (degrees)');
                xlabel('Theta (degrees)');
                title({['Reference (', num2str(Xo), ', ',
num2str(Yo),...
```

```matlab
                    '); no of points: ',num2str(length(Az))];...
                ['RMS(theta) = ',num2str(thetaRMS), ' degrees']});
                grid on;
                axis([0 360 0 360]);
                set(gca, ...
                    'XTick',0:40:360, ...
                    'YTick',0:40:360, ...
                    'XMinorTick', 'on', ...
                    'YMinorTick', 'on', ...
                    'XMinorGrid', 'on', ...
                    'YMinorGrid', 'on', ...
                    'GridLineStyle','-',...
                    'MinorGridLineStyle','--')
            end
        end
        if disprms == 1
            disp(['>>> The smallest of Group =
',num2str(SmallerRMS),...
                ' degrees']);
            disp(['    when the point at
(',num2str(XoInGroup),',',...
                num2str(YoInGroup),').']);
            disp(['>>> The smallest RMS = ',num2str(SmallestRMS),'
degrees']);
            disp(['    when the point at (',num2str(Xos),',',...
                num2str(Yos),').']);
        end
    end
    disp(' ');
    disp(' ');
    disp('>>> SUMMARY');
    disp('>>> The range of scan: ');
    disp(['    Xo from ',num2str(Xofirst),' to ',num2str(Xolast)]);
    disp(['    Yo from ',num2str(Yofirst),' to ',num2str(Yolast)]);
    disp(['>>> The smallest RMS = ',num2str(SmallestRMS),'
degrees']);
    disp(['    when the point at
(',num2str(Xos),',',num2str(Yos),').']);
    disp(' ')
    Ya = Ysp;
    sum = 0;
    count = 0;
    Xa = 300;
        a = sqrt((Xos - Xc).^2+(Yos - Yc).^2);
        b = sqrt((Xa - Xc).^2+(Ya - Yc).^2);
        c = sqrt((Xa - Xos).^2+(Ya - Yos).^2);
        angle = acosd((a.^2+b.^2-c.^2)/(2*a*b));
    mean = sum/count;
    disp(' ');
    disp(['Mean angle = ',num2str(angle),' degrees']);
end
disp('end');
disp(' ');
```

B.2 The Program for Measuring The Instrumental Magnitude of A Star

```matlab
clear all;
disp('***********');
disp(' '); %load the frame
a = imread('sky20080404_2200.fits');
a = double(a);%load the dark frame
b = imread('sky20080501_1342.fits');
b = double(b);
c = a - b; %substract dark frame from all-sky image.
G = 64;
const = 50;
Xc = 180; %Image position of star 'Alioth'
Yc = 231;
mag_obs = Xc - Xc;
error_mag = Xc - Xc;
error_mag2 = Xc - Xc;
R = input('Radius of the aperture = ');
r_inner = input('Inner radius of the annulus = ');
r_outer = input('Outer radius of the annulus = ');
for loop = 1:length(Xc)
    x = Xc(loop) - R;
    count = 0;
    for sub_x = 1:2*R
        y = Yc(loop) - R;
        for sub_y = 1:2*R
            if (x-Xc(loop))^2 + (y-Yc(loop))^2 <= R^2
                subframe(sub_y,sub_x) = c(y,x);
                count = count + 1;
                star(count) = c(y,x);
            else
                subframe(sub_y,sub_x) = 0;
            end
            y = y + 1;
        end
        x = x + 1;
    end
    %%
    %Annulus
    x = Xc(loop) - r_outer;
    count = 0;
    for sub_x = 1:2*r_outer
        y = Yc(loop) - r_outer;
        for sub_y = 1:2*r_outer
            if (x-Xc(loop))^2 + (y-Yc(loop))^2 > r_inner^2 && (x-Xc(loop))^2 + (y-Yc(loop))^2 <= r_outer^2
                annulus(sub_y,sub_x) = c(y,x);
                whole_sample(sub_y,sub_x) = c(y,x);
                count = count + 1;
                base(count) = c(y,x);
            else
                annulus(sub_y,sub_x) = 0;
                if (x-Xc(loop))^2 + (y-Yc(loop))^2 < R^2
                    whole_sample(sub_y,sub_x) = c(y,x);
                else
                    whole_sample(sub_y,sub_x) = 0;
                end
            end
            y = y + 1;
        end
        x = x + 1;
    end
    lincount = count;
```

```matlab
    [mu_annulus,sigma_annulus] = normfit(base,0);
    disp(' ');
    disp(['mu(annulus) = ',num2str(mu_annulus),', sigma(annulus) =
',...
        num2str(sigma_annulus),' ADUs']);
    disp(' ');
%%
    figure;
    mesh(whole_sample);
    xlabel({'Pixel number';'along x-axis'});
    ylabel({'Pixel number';'along y-axis'});
    zlabel('Pixel value (ADUs)');
    axis tight;
    figure;
    image(whole_sample/2.5);
    colormap('gray');
    xlabel('Pixel number along x-axis');
    ylabel('Pixel number along y-axis');
%%
    %count the frequency of every pixel value of the star.
    count = 0;
    k = 0;
    star_pixel_number = length(star);
    total_star_pv = 0;
    for j = 1:length(star)
        total_star_pv = total_star_pv + star(j);
        newvalue = 0;
        count = count + 1;
        if count > 1
            for k = 1:count-1
                if star(j) == star_pv(k);
                    star_freq(k) = star_freq(k) + 1;
                    count = count - 1;
                    newvalue = 1;
                    break;
                end
            end
            if newvalue == 0
                star_freq(count) = 1;
                star_pv(count) = star(j);
            end
        else
            star_freq(count) = 1;
            star_pv(count) = star(j);
        end
    end

    %%
    %disp(' ');
    %disp('Sampled star');
    %disp(['Star pixel number = ',num2str(star_pixel_number)]);
    %disp(['frequency =   ',num2str(star_freq)]);
    %disp(['pixel value = ',num2str(star_pv)]);
    %%
    star_norm_freq = star_freq - star_freq;
    for temp = 1: length(star_freq)
        star_norm_freq(temp) = star_freq(temp)/star_pixel_number;
    end
    %disp(['norm_freq   = ',num2str(star_norm_freq)]);
    %figure;
    %bar(star_pv,star_norm_freq);
    %title({'Star histogram';' '});
    %xlabel('Pixel value (ADUs)');
```

```matlab
    %ylabel('Normalised frequency');
%%
    %Annulus histogram
    count = 0;
    k = 0;
    annulus_pixel_number = length(base);
    total_annulus_pv = 0;
    for j = 1:length(base)
        total_annulus_pv = total_annulus_pv + base(j);
        newvalue = 0;
        count = count + 1;
        if count > 1
            for k = 1:count-1
                if base(j) == annulus_pv(k);
                    annulus_freq(k) = annulus_freq(k) + 1;
                    count = count - 1;
                    newvalue = 1;
                    break;
                end
            end
            if newvalue == 0
                annulus_freq(count) = 1;
                annulus_pv(count) = base(j);
            end
        else
            annulus_freq(count) = 1;
            annulus_pv(count) = base(j);
        end
    end

    figure;
    histfit(base,length(annulus_freq));
    %title('annulus pv');
    xlabel('Pixel value (ADU)');
    ylabel('Frequency');
    %disp(' ');
    %disp(['Annulus pixel number = ',num2str(annulus_pixel_number)]);
    %disp(['annulus frequency =   ',num2str(annulus_freq)]);
    %disp(['annulus pixel value = ',num2str(annulus_pv)]);

    annulus_norm_freq = annulus_freq - annulus_freq;
    for temp = 1: length(annulus_freq)
        annulus_norm_freq(temp) =
annulus_freq(temp)/annulus_pixel_number;
    end
    %disp(['annulus norm_freq   = ',num2str(annulus_norm_freq)]);
    figure;
    bar(annulus_pv,annulus_norm_freq);
    title({'Annulus histogram';' '});
    xlabel('Pixel value (ADUs)');
    ylabel('Normalised frequency');

%%
    %mean_background
    mean_background = total_annulus_pv/annulus_pixel_number;

    disp(' ');
    disp(['Mean background flux = ',num2str(mean_background),'
ADU']);

    %%
    %FWHM of sky background
    n = length(annulus_freq) - 1;
```

```matlab
    for i = 1:n %Range annulus_freq in descending order
        for j = 1:n
            if annulus_freq(j) < annulus_freq(j+1)
                temp = annulus_freq(j);
                temp2 = annulus_pv(j);
                annulus_freq(j) = annulus_freq(j+1);
                annulus_pv(j) = annulus_pv(j+1);
                annulus_freq(j+1) = temp;
                annulus_pv(j+1) = temp2;
            end
        end
    end
    mode = annulus_pv(1);
    disp(['mode = ',num2str(mode),' ADU']); %mode
%%
    %star flux
    star_flux = total_star_pv - mean_background*star_pixel_number;
    disp(['Star flux = ',num2str(star_flux),' ADU']);
    HM = annulus_freq(1)/2;
    n = length(annulus_freq) - 1;
    for i = 1:n %Range annulus_pv in ascending order
        for j = 1:n
            if annulus_pv(j) > annulus_pv(j+1)
                temp = annulus_freq(j);
                temp2 = annulus_pv(j);
                annulus_freq(j) = annulus_freq(j+1);
                annulus_pv(j) = annulus_pv(j+1);
                annulus_freq(j+1) = temp;
                annulus_pv(j+1) = temp2;
            end
        end
    end
    count = 0;
    for i = 1:length(annulus_freq)
        if annulus_freq(i) >= HM
            count = count + 1;
            width(count) = annulus_pv(i);
        end
    end
    FWHM = width(count) - width(1);
    disp(['FWHM = ',num2str(FWHM),' ADU']);
%%
    %uncertainty of magnitude
    denominator = sqrt(star_flux/G +
star_pixel_number*sigma_annulus^2 +
star_pixel_number*sigma_annulus^2/annulus_pixel_number);
    ratio_SignalToNoise = star_flux/denominator;
    error_mag(loop) = 1.0857/ratio_SignalToNoise;
    mag_obs(loop) = const - 2.5*log10(star_flux);
end
```

References

Many texts have been suggested throughout the thesis for further readings. For convenience, they are also collected here.

[1].P.Duffett-Smith, Astronomy with Your Personal Computer, 2nd Edition, Cambridge University Press, Page 15-20,29-32,37-50.

[2].John R. Taylor, An Introduction to Error Analysis, Oxford University Press (1982), Chapter 8, 9, 11 & 12.

[3].P.K.Seidelmann, Explanatory Supplement to the Astronomical Almanac, University Science Books (1992), Page 2-5,48-56,600-604.

[4].The Astronomical Almanac For The Year 2007, The Stationery Office, Section H (V band stars' magnitudes).

[5].M. Syrjasuo, All-Sky Camera, Helsinki University of Technology (1996)

[6].J-K. Chung, Y. H. Kim and Y-I. Won, Observation of Mesospheric Waves with An All-Sky Camera in Korean Peninsula, Advances in Space Research, Volume 32, Issue 5, p.825-830.

[7].Shair, Lior, Transient Detection Using Panoramic All-Sky Camera, Astrophysics and Space Science (2006), Volume 305, Number 2. pp. 165 – 168(4)

[8].D.W.E.Green, Magnitude Corrections for Atomspheric Extinction, International Comet Quarterly, Vol. 14, p.55.

[9].K. Krisciunas, Further Measurements of Extinction and Sky Brightness on the Island of Hawaii, Astronomical Society of the Pacific (1990), vol. 102, p. 1052-1063.

[10].T.E.Pickering, The MMT All-Sky Camera, MMT Observatory, 933 N. Cherry Ave., Tucson, AZ 85721, USA

[11].J.H.Simonetti, Measuring the Signal-to-Noise Ratio S/N of the CCD Image of a Star or Nebula, Department of Physics, VirginiaTech.
http://www.phys.vt.edu/~jhs/phys3154/snr20040108.pdf.

[12].J.Kaler, Stars, Astronomy Department University of Illinois,
http://www.astro.uiuc.edu/~jkaler/sow/sowlist.html.

[13].H.U.Frey, All-Sky Camera, Space Physics Researh Group, UC Berkeley,
http://sprg.ssl.berkeley.edu/atmos/sondre.html.

[14].C.L.Peterson, Allsky Meteor Camera, Cloudbait Observatory,
http://www.cloudbait.com/projeccts/allskycamera.html .

[15].H.Yamada, C.Aspin, UH 2.2 Telescope – Public Area (All-Sky Camera), Institute for Astronomy, University of Hawaii, http://uh22data2.ifa.hawaii.edu/public/allsky.

[16].F. Chereau, M.Gates, et al., Stellarium 0.9.0, http://www.stellarium.org.

[17].GAIA::SkyCat, Extragalactic Astronomy & Cosmology Research Group, Department of Physics, Durham University, http://astro.dur.ac.uk/~pdraper/gaia/gaia.html.

[18].SKY2000 Catalog, Version 4 (Myers+2002), VizieR, Centre de Donnees Astronomiques de Strasbourg, http://vizier.u-strasbg.fr/viz-bin/VizieR-3.

[19].Lumicon Deep Sky Filter, Astronomy & Optical Centre, Rother Valley Optics LTD, http://www.rothevalleyoptics.co.uk/lumicon-deep-sky-filter-fits-all-1-25-eyepices_d907.html.

Notations and abbreviations

ASC	All-sky camera
DAC	Durham all-sky camera
ADU	Analogue-to-digital unit
RMS	Root mean square
ST	Sidereal Time
LST	Local Sidereal Time
UT	Universal Time
RA/Dec	Right ascension and Declination
Az/El	Azimuth and Elevation
Z	Zenith distance
P	Pixel distance
δZ	RMS of the zenith distance
δP	RMS of the pixel distance
(x,y)	Observed pixel position of a star
(X_S, Y_S)	Calculated pixel position of a star
(X_Z, Y_Z)	Pixel position of zenith
(X_0, Y_0)	Pixel position of zero-azimuth
θ	Azimuth angle of a star measured an the image
G	The gain of DAC
F_S	Flux of a star
n	The number of the pixels within an aperture
p	The number of the pixels within an annulus
$Mag(cat)$	Catalogue value of the apparent magnitude of a star
$Mag(obs)$	Instrumental magnitude of a star