



By far the greatest amount of data was collected from area 'a', nearly 74,000, the numbers are in the table below.

I collected data from one area in Auriga also, centred at radec: 05h 18m 0s, 41°50' 0". Labelled area 'p'

The number of images collected only depends on the weather and varies from just a handful, 30 say, to about 600 on a good long night in Cygnus, a few more are possible in Auriga. The short runs can add a little useful data for short period stars. The images are collected with ra horizontal and dec vertical, i.e. dec=y and ra=x. This is only exact in the middle of the image because the lines visibly curve of course. I used a Losmandy mount and I want to keep the stars in approximately (but not exactly) the same place on the chip all night (that reduces systematic errors) so the scope has to be rotated through 180 degrees at 'switchover'. Also, that switchover must take minimum time so there has to be a mechanical arrangement to maintain the ra/dec alignment. I used adjustable end stops which with care gave consistent results for months.

The 9 imaging locations in Cygnus are labelled 'a' to 'i' .

A list of the nominal centres of the imaging locations together with the number of images taken and the number of variables found (as of 01-01-2015) is in the table below:

Table 1

	RA			DEC			
Location	hr	min	sec	deg.	min	no of images	no of vars
a	21	08	30	46	30.0	73675	1023
b	20	51	30	43	30.0	413	36
c	21	08	30	43	30.0	584	229
d	21	25	30	43	30.0	1936	260
e	21	25	30	46	30.0	2490	287
f	20	51	30	46	30.0	2057	284
g	20	51	30	49	30.0	421	167
h	21	08	30	49	30.0	642	159
i	21	25	30	49	30.0	750	180
p	5	18	0	41	50.0	20967	178

The total number of images taken as listed above was 103,935. The good ones that are used number about 90% of those. The total number of data values analysed so far in area 'a' is 5,600 million, that is on 76,000 stars but more could be done.

The total number of variables in the list above is 2,803 but the numbers will change a bit as more confirming work is done. The number in Auriga (area 'p') will certainly increase.

The chip data and FOV are as below:

4080 pixels at 9mu=36.72mm chip—with a 750mm focal length = 0.04896 radians square or 2.805° square

Fits file size: 32.676Mb, each pixel 2.475 arc-sec

## 2. Data Reduction

### 2.1 Programming

Except for the initial file conversion all the data analysis is done with programs written by me (SW) in Dyalog APL. I used apl because I was very familiar with it from my previous working life! APL is an interpreted

language so not the fastest but it is highly compact and mathematically adept at handling large arrays so is ideal for the analysis of large numbers of stars or pixels in a simultaneous way. Some examples of apl programmes will be given. Any of the analysis routines can be made available for other people to use.

## 2.2 File Rename and File Conversion

The next stage is a file rename to change from the 3/4 digit file number that Maxim uses to the five that I need so file a123 becomes file a00123. Also when the numbers get to 9999 in Maxim one has to start again at 1 so from that point onward 10000 is added to the file number made by Maxim.

The next step is to extract time and other info and convert the Fits files to an image data only file. This is the only non-apl program, called Fitsconvert and written by a kind friend, Nigel Eke, who now lives in Australia. The data is then converted to APL format. At this point a quick check program is used to assess the quality of the images. This is particularly important on a night with cloud about. The program assesses the images only in terms of average brightness and discards those that fail pre-set criteria. On a night with a lot of cloud this saves a lot of wasted later processing but useful data can still be obtained under such conditions from those that are left. It also removes bright aircraft trails (I live just 6 miles from Luton Airport).

Then the images, in an apl workspace, are dark subtracted and divided by the flat field. Both the darks and the ffs are prepared by other programmes, the darks are averages of up to 50 but the flats usually less than that, depending how they are made. The flat fields are obtained from the sky or from a flat-field generator which is essentially a large baffled sheet of opal Perspex, abraded and illuminated at a small angle (four degrees) by a source some 20 metres away, the assembly being just outside my observatory. The small angle eliminates the possibility of direct rays getting into the telescope. Although the flats generated are, in ways I don't understand, not as good as infinite sky ones I do get the same pattern of vignetting from them as from the sky and no correction for shutter movement is required because the exposures are much longer than for sky flats. There are no stars to be removed either. This system is entirely satisfactory in practice because I only ever use close reference stars- I'm not trying to precisely flat-field the whole of the FOV- and an accuracy of a few 10ths of 1% is achievable over the distances required.

The flats have their hot pixels removed and replaced by average values. The image files are 4090 by 4090 pixels. The leftmost 10columns and the topmost 10 rows are removed, they are of poor quality anyway, and that leaves 4080 by 4080 which has many more divisors than 4090.

In the course of the apl importing process the time and date are noted and removed from the data file, the number of pixels above a threshold and the picture sky background across the image are calculated. These data are then assessed by a subroutine which flags those images that are doubtful for continuing with for whatever reason, usually high background variability due to cloud. The elevation of the image centre is also computed. Some of these data sets are used to form a 'control and information variable', Table 2.1 shows a few lines near to image 1051 from 2004. The expression 'variable' in apl refers to objects, often numbers, created by a computing routine (or by hand).

These lists are quite long (19,322 lines for the 2006/7 season for example!) but very useful as they give an immediate view of the image situation: time, elevation, background and brightness, quality

Table 2.1

Pic id	date	time	Julian Time*	no>5000	no sat	av p-p rms	flat no dark no	pass	elevation	HJD
z-01048a	2004 9 1	21 36 24	1705.40028	1330	613	60 6.2 1.2	38 36	1	80.9	1705.40328
z-01049a	2004 9 1	21 37 19	1705.40091	1246	545	60 6.5 1.3	38 36	1	81	1705.40391
z-01050a	2004 9 1	21 38 14	1705.40155	1279	591	60 6.4 1.3	38 36	1	81.1	1705.40455
z-01051a	2004 9 1	21 39 8	1705.40218	1340	631	60 6.2 1.2	38 36	1	81.2	1705.40518
z-01052a	2004 9 1	21 40 4	1705.40282	1340	618	60 6.3 1.3	38 36	1	81.4	1705.40582

\*The Julian time is from Jan 1<sup>st</sup> 2,000. The heading 'av p-p rms' refer to the sky background in ADU, average , peak to peak and rms over the image.

The data output (i.e. The adu in each pixel) from the importing routine is multiplied by 10 and made integer to improve the resolution and speed up processing but that increases the file size to 65.026 Mb per image. These are the fundamental apl picture files with a scaling of 1 unit per 0.1 adu.

### 2.3 Finding the stars (and other objects)

This needs a mention here because it's a pre-requisite of the picture coefficient process.

At this stage the apl picture files have an average value close to zero between the more than 100,000 clusters of above average pixels which are the stars and other luminous objects scattered across each one. There are also a number of hot pixels, and, depending on the definition, there are 10,000 ish of these, mostly very faint.

I've used two methods of star-finding. One is the agglomeration method where the program finds all the pixels above a threshold and then, by using a nested loop, finds all the pixels joined to each of those that are above a base threshold. It works but is slow (several seconds for a few thousands stars) and a change of threshold needs a recalculation- the brightness can be obtained by adding all the touching pixels in each object. The second and much faster method finds the maxima across the image. This can find 150,000 stars, their centroids and approximate brightness in a typical image in under a second. Apart from speed the other advantage of this method is that it can separate melds, even very close and of very different brightness.

The method uses a simple differential comparison of the image compared to itself shifted left, right and up and down by one element. This enables one to find maxima which may or may not be linked by bridges to other objects or be part of complex melds. Many objects are complex in Cygnus and cannot be evaluated by simple circles but they can be found and studied by more subtle approaches.

A threshold controls the flux sensitivity, too low and a million 'stars' will be seen but over half will be noise spikes! In practice in the Cygnus images over 300,000 stars can be found in each image. Most are too faint to be worth investigating and I have limited my catalogue in area 'a' to 76,000. Finding the stars is an essential preliminary to identifying them so they can be tracked from image to image.

So, the first step in the picture coefficient process is to use this method to find the co-ordinates of the 6000 brightest stars in each image.

### 2.4 Picture Coefficients (pcs)

So the next essential is to know where you are on the image in relation to the sky and this connection is formed by the picture coefficients.

I call this process 'radeccing'. It is the set of algorithms which generates the relationship between the picture co-ordinates of the stars (x and y) and their sky co-ordinates, i.e.: RA and DEC so that the stars can be uniquely identified and tracked over a long period of time. The picture coefficients are simply that mathematical relationship between the co-ordinates on the chip and the co-ordinates in the sky. The program does this by locating guide stars and looking up their RA and DEC in a catalogue and then fitting a

mathematical relationship between RA and DEC and x and y. I find a fourth order polynomial (and two dimensional in x,y or ra,dec) works best with my present setup giving very small errors (<0.2 arc-sec) over the whole field. Because there are always small alignment errors and the lines of equal ra and dec are not exactly linear across 2.8 degrees both ra and dec are functions of x and y which is why it is necessary to use a two dimensional fit.

To control a 4th order polynomial it needs good constraints so the program chooses (iteratively) 144 guide stars, one in each of 144 squares spaced across the image and fits the 4th order polynomials to them. The stars are chosen from the USNO B1.0 catalogue. Two sets of coefficients are calculated , ie x,y to RA, DEC and vice versa.

I used simple polynomials because APL is brilliant at them and inverting the matrices to solve them! The chip is assumed to be uniform, all pixels the same size and my x co-ordinate is zero is the left edge of the left most column of pixels used, y starts at the top of the top row of pixels. The chip is 4090 by 4090 but the left few are very poor anyway and 4080 is a much more useful size (many divisors) so the left 10 and top 10 are removed to leave a 4080 by 4080 pixel set so that x and y run from 0 to 4080.

The picture coefficients are of the form:

$$1) \text{ ra} = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 + a_{10} y^3 + a_{11} x^4 + a_{12} x^3 y + a_{13} x^2 y^2 + a_{14} xy^3 + a_{15} y^4$$

$$2) \text{ dec} = b_1 + b_2 x + b_3 y + b_4 x^2 + b_5 xy + b_6 y^2 + \dots \text{ similarly...}$$

A first order fit uses the 1<sup>st</sup> three terms, a second order one would be the terms shown up to a6 and so on. I use a 4<sup>th</sup> order fit which is 15 terms. One can write similar expressions for x and y as polynomials in ra and dec:

$$x = c_1 + c_2 \text{ra} + c_3 \text{dec} + c_4 r^2 + c_5 d^2 + c_6 rd + \dots \text{ where r and d are ra and dec ....}$$

$$y = d_1 + d_2 r + d_3 d + d_4 r^2 + d_5 d^2 + d_6 rd + \dots$$

To find the coefficients ( $a_n$ ,  $b_n$ ,  $c_n$  and  $d_n$ , above) one needs to have a set of good data. Equation 1 for example one needs a minimum 15 pairs of x and y to go with 15 values of ra so creating 15 simultaneous equations with 15 unknowns. In practice I use 144 sets of values, see below, and this matrix is very easily solved (with just 3 symbols in one line) in APL.

The values of ra and dec come from a catalogue but to which star among the 100,000 or so on the image does each of those ra's correspond? I've used two methods to find those vital x's and y's. Both use a subset of data from about 6,000 stars covering the picture plus a similar number from a catalogue such as USNO B1.0. One knows the nominal centre and the fov so selecting the 6,000 brightest stars from any catalogue is an easy matter. Method (a) is slow, but needs no manual input and is a multistar pattern matching approach. Method (b) is much faster and uses a seed triangle of 3 brightish stars towards the middle of each image

### Method a, Star Pattern Recognition

A star is identified by studying and recognising the positions of the stars around it.

This is the most fundamental way and requires no external input other than a catalogue of star coordinates. For each of the brightest 6,000 stars in the catalogue the three brightest in the vicinity of

each (which are also in that set of 6,000) are identified (this is a slightly vague but sufficient definition) from the image data. Obviously, this often goes wrong because what my program thinks is the three brightest 'near' stars is frequently not the same as the 3 brightest 'near' stars in a catalogue which is why I start with 6,000 rather than a few hundred- however, the process is self correcting. The distance (in arc-sec) from the target star to each of those three and the 3 distances between the three are calculated. So each triangle of stars surrounding the target star is defined by those 6 numbers. The corresponding set of 6 numbers for each of the 6,000 stars in the catalogue have been pre-calculated using the scale factor between pixel coordinates and arc-seconds which can be fairly well estimated (from chip data and focal length) and then refined in the general calculations. The catalogue arc-sec values are thus converted to distances in pixel co-ordinates (not x and y but the radial distance 'r' as we don't at this stage know the exact orientation between the pixel axes and the sky axes) A set of 6 such numbers are attached to each of the 6,000 brightest stars in the catalogue which covers the patch of sky being imaged. This is just done once for each area of sky.

For each image, similar calculations (directly in picture co-ordinates) are done for a few thousand of the brightest stars in the image. This set of n by 6 numbers is then compared to the starlist set. When all 6 numbers are equal (to a certain tolerance) between a star in the image and a star in the catalogue that creates a provisional identification. So, after a first comparison a number of the objects now have both known values of ra and dec and x and y and these values are inserted into the set of simultaneous equations above, which are then solved. In practice it is solved first as a 2<sup>nd</sup> order equation.

This is a matrix inversion which is one symbol in APL! (this is why I use APL, it is extremely concise and mathematically orientated- it handles multidimensional arrays of numbers as one object. For example, the program to solve a set of 10 or 10,000 simultaneous equations is just one line, with a few bits and pieces around it.)

The equations are evaluated, the image stars recalculated and poorly fitting ones discarded. The polynomials are recalculated iteratively through several 2<sup>nd</sup> order, 3<sup>rd</sup> and 4<sup>th</sup> order fits. Eventually a 1,000 or so of the stars are identified with a close agreement between the 6 numbers in the image and the 6 in the catalogue, for each of those 1,000 stars. The image is notionally divided into 144 equal squares, 340 pixels square, and the best guide star (the one with the least error from the polynomial) in each square is selected. The reason for this is so that the equations are evenly balanced over the picture to minimise the errors everywhere. Typically using the USNO B1.0 catalogue the rms error of the fit of the stars is less than 0.1 arc-sec over the image but corners are sometimes a little more than that. I found, after a lot of experimenting that a 4<sup>th</sup> order final fit was best. A lower order fit is not accurate enough and higher orders may fit the 144 points very slightly better but they can move more in between so the overall accuracy is worse. The program starts with a 2<sup>nd</sup> order fit and progresses through several of those and several 3<sup>rd</sup> and 4<sup>th</sup> order fits as the number of well fitted stars increases. Eventually all 144 guide stars are within the tolerance (set to be 0.5 arc-sec) to give a final answer.

This method is almost totally reliable but is slow, taking around 30seconds cpu time per image. It is though, better than method b) below for a constantly changing fov centre because identifying a reliable triangle may sometimes be a problem.

### **Method b, The triangle seed**

This works a bit like the growth of a bacterial colony on a Petrie dish! Three bright stars near the image centre have been pre-chosen for each data area. The list of the ra and dec of those three stars is the seed.

This is much faster, taking less than a second per image but it does require the extra 'hand' input of the triangle data in addition to the trimmed catalogue, but both are only required once for each sky area. As I was only imaging in 10 areas the preliminary work was quite small but the extra speed made it possible to analyse even 600 images in the 12 hours following their capture.

Once only: a 'seed' triangle of bright stars near the centre is chosen by inspection and the stars identified from a planetarium program and their ra/dec co-ordinates noted. The seed for area 'a' (see document 'Project Outline') is shown below in a 201 by 201 pixel sample (the 3 bright stars in the middle), this happens to be a particularly good one, though it is a little small:

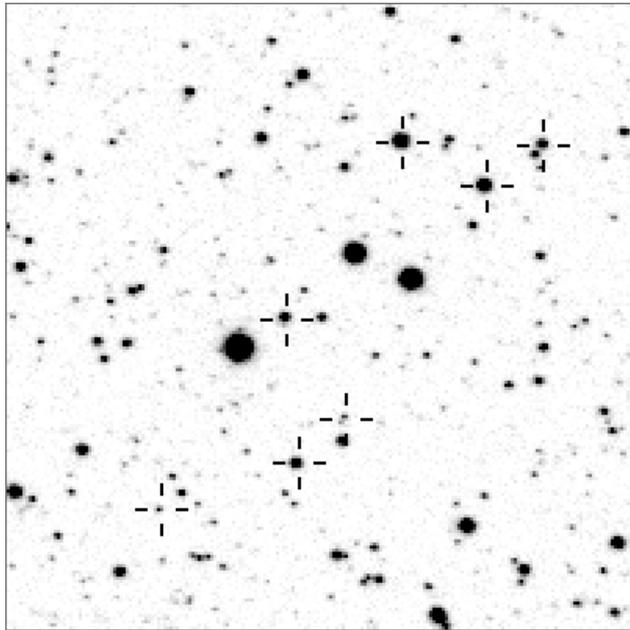


Figure 2

(This is from my standard bitmap program, the stars with crosses have been found to show variability). The six seed numbers for the triangle of bright stars are simply:

21	8	5.20	46	26	24.66
21	8	17.93	46	25	32.52
21	8	13.76	46	25	12.61

There are thus 6 values of ra and dec used as an input for each area of sky. The scale factor (arc-secs in the sky to microns or pixels on the chip) is known pretty well so the lengths of the three sides of the triangle on the chip can be well estimated. Call these three numbers 'tri'.

The picture-coefficient routine starts with a small subroutine which looks at the central 1% of each image (10% in x and y) to match the side lengths of the 100 brightest triangles with the specified triangle, the numbers 'tri'. There are of course many, many triangles of stars but this takes less than a milli-second and, even though we are only matching 3 numbers, the correct triangle always has a very much better match than the closest of the others. Very poor quality images are automatically rejected because the matching triangle cannot be found. Having done that and taking the triangle vertices, there are now known 3 matching pairs of x,y with three matching pairs of ra/dec, ie six matching number pairs.

So, having identified that 'seed' triangle the job is almost done because those numbers are just enough to compute the six 1<sup>st</sup> order (i.e. straight- line) coefficients ( $a_1, a_2, a_3, b_1, b_2, b_3$  above ). This approximates the ra and dec axes to straight lines all over the image, which they are not, but it is

sufficient to accurately identify a further 30 or so stars near the seed triangle. They are identified from that same list of 6,000 stars but this time, because the program predicts the ra/dec positions of the stars (from the progressively improving polynomial), the identification is immediate. The ra and dec values from those are more than enough to compute a second order fit, which is repeated. As is a third and fourth order several times until 144 evenly spaced best fit and low error 'guide stars' are found. They are then used to compute the final fourth order coefficients. Because the program is merely predicting and comparing the spread of identifications happens in a flash.

These 15 4<sup>th</sup> order coefficients are computed for each image and stored and then used in all subsequent calculations on that image. The ra and dec of any pixel (or any point on a pixel) on that image can be computed. Similarly, any ra and dec in the sky area covered by the image can be located on the image.

So any event in the image can be located on the sky and the position of any sky object, even if nothing is visible on the image, can be located in the image.

That enables me, if necessary, to average the pixels of an invisible object thousands of times to, hopefully, make it visible!

So, any image can be probed at a specified ra/dec to retrieve data corresponding to that point in the sky for that 30 seconds of time. Next, we need to decide what objects to locate and study.

### 3. Data Analysis

#### 3.1 Introduction

The APL picture files are the initial basis for the data pipeline but they are not used throughout the process for obvious practical reasons: just to get the data on one star throughout the time period would otherwise need files retrievals on 70,000 images in area 'a', some 4.35 Tb. It was obvious early on that the data would need considerable pre-processing but it was essential not to lose the original data in the process.

The most efficient method is to divide the images up in some way, using x,y s tied to ra,dec co-ords and then assemble each set of related ones into a long string and have lots of those strings. Each string will then contain data from a sky locked area.

There are many objects on each image, including many groups of pixels with maxima and the data to be analysed is the flux recorded by those groups of pixels, most of which are stars. The single pixels that stand out are the so-called hot pixels which are easily identified because they are single and stay in the same x,y position from image to image. The stars stay very nearly in the same ra/dec position, though some movement of the nearer stars can be seen. Much rarer objects are cosmic ray hits which are also single pixels and are only on one image. More annoying are aircraft trails, which can make an image hard to use. There are also the occasional satellite trail and presumably asteroid movements but I've not looked for any of the latter.

I've used two quite different ways of studying the image objects, one is the starlist approach, the other is the 'everything' or 'area' approach. The first which I have followed for most of the time since I started in 2003 either uses a starlist created from a chosen set of images, which I used exclusively from 2003 to 2011, or a starlist derived from a published one, such as the USNO ones. The area approach divides every image into a set of equal squares, each centred on a pre-set ra/dec (and overlapping 10% with each other) and aims to log every object above a threshold flux in each square. These are then correlated with the objects from the same square in the next image and so on. The area method is much more complex and cpu intensive than the starlist approach but, in principle, anything that happens anywhere in any image at anytime can be seen.

I used this method to search for transient happenings, in particular flares which may well happen in objects too faint to be in my starlist. Most of the time I have used the starlist approach. The two methods are described in a bit more detail below.

### 3.2 Starlists

I've found starlists to be a perennially thorny problem and I've used all sorts of different approaches. When the new USNO catalogue (B1.0) came out (some years ago now!) I thought, great I can use that to quickly create starlists with which to probe the pictures. Sadly, that proved grossly over-optimistic! Having downloaded the info for areas 'a' to 'i' and 'p', I spent weeks trying to create a clean starlist for area 'a' and finally gave up. There are so many unreal objects and duplicates it was a nightmare trying to sort it out.

The other big problem with this approach is the missing stars. One has to use a cut-off because down to mag 22 there are a million objects in my area 'a' so I used magnitude R16 which gives me a list of 60,000 in an area slightly bigger than 'a' to allow for drift and pointing errors over the season. Of course, limiting the magnitude means that some long period variables were too faint at the epoch of the images used to get in that list so some particularly interesting ones are automatically missed! However that will eventually be corrected by examining the data for another 50,000 stars fainter than the main starlist.

With hindsight, I realise now that I was being naive in expecting a cleaner list when it includes hundreds of millions of stars and was never intended for the purpose I was essaying, especially in view of the density of stars in Cygnus.

I'm sure some of the more recent catalogues would be brilliant but they were not available in 2003.

So, I went back to the best method, which is to make your own list from your own images but that is not that simple a process, one has to eliminate irregular pixels and cosmic ray hits and have a policy on melds etc. I don't yet have a program that will do this entirely reliably and automatically and it took two weeks to make my first starlist of 21,394 for area 'a'. Of course having made a list, one can then cross-reference it back to a published catalogue and I used USNO B1.0\* for a long time, though again there are complications of course. A pair I can't resolve, so classify as a meld and measure as a meld may be two separate stars in the catalogue. One needs a fairly intelligent program so sort all that out, which takes time, again - but there is always a lot of eyeballing to do as well, even more time!

#### 3.2.1 Making the Starlist

The starlists are made by identifying the N brightest objects on a set of 30 good images spread over as long a period as is available and, using the picture coefficients, the ra,decs of the N objects are calculated and the lists from each picture are compared and gradually amalgamated into one. At this stage the 'star' has to be in over 50% of the images to be counted as real. The ra and dec co-ordinates (hereinafter called radecs) of the objects in this list are then compared to a published catalogue to get mag values and other IDs. When I started this work in 2003 the aim was to search for extra-solar planets so my catalogue was limited to the brightest 12,001 stars (later, I noticed that star 12001 varied by over 4.4 magnitudes!). I thought at the time that finding a 2% dip (as in HD209458- the first extra-solar planet found by transit) would not be possible for fainter stars. As time progressed and I studied the increasing pile of data, the noise in the system soon became of great interest (it had dawned on me that this 'noise' was variable stars!), At that point I contacted Roger Pickard, head of the BAA variable Star Section and increased the catalogue to 17,000 and later again to 75,000 in area 'a'.

I have so much data in area 'a' that, with averaging, at least 200,000 objects can be seen, but I've not studied all of them, yet.

- A much higher quality comparison list for my area 'a' was very kindly made by Chris Lloyd using the CMC catalogue

#### 3.2.2 Data Reduction with a Starlist

All the data analysis from this point is made with reduced images which are those that have been dark subtracted, flat divided, converted to apl, multiplied by 10 and made integer.



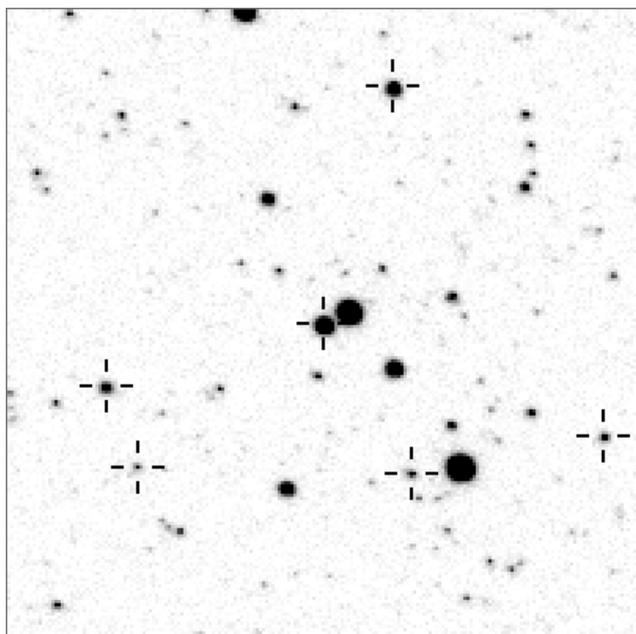
on averaging the pixel sets not averaging flux totals. In addition for melded stars a template can be used to extract a subset of pixels to minimise neighbour interference. Removing a subset of outer pixels close to a neighbouring star will reduce the interference and then, knowing the growth curve, the missing contribution can be replaced. Another, and usually better way with a bright interferer, is to compute the contribution of the interferer within the collecting circle of the target star and subtract it. I'm intending to do that with the important star mentioned above, a00656, because it's variability is only a few % anyway.

Stars are particularly closely packed in Cygnus (which is why I chose it as a target) so the luxury of using a large collecting circle is not often available. In order to choose an optimum diameter (which may change with time during a night and seeing conditions) a choice of diameters for every star in every image gives great flexibility. Apart from choosing a diameter, the data to predict the loss as a function of diameter is also available so that if a small diameter has to be used the asymptotic value can be calculated. This leads to a set of data with a smaller spread, or lower noise.

So, growth curves are essentially a set of light curves for a range of probe diameters around each star in the starlist. The project generated starlist for area 'a' is 75,000 long so each image is probed at that number of ra-dec points with a set of collecting circles of different collecting areas. After some initial variations I have settled on a smallest area of 3 by 3 pixels square and then 13 different circles increasing from 4 pixels diameter to 16 diameter. The small sizes are used for stars in close proximity to others. Below are a few growth curves from stars in Fig 3. The central star is a00656 in my catalogue and it is a very interesting variable. There will be lots about elsewhere on the website. This, again, is a 201 by 201 bitmap:

**Fig 3**

The next picture, Fig 4 is a key to the stars in the bitmap, the crosses indicate that I have found the star to have variability. I'll show below growth curves of the variable star a00656 which is star 'o' in the centre of the image and close to a brighter star, a00100 in my catalogue and the fainter star 'c' or 941 just below and to the right.



The radial distance between the centroids of 656 and its brighter companion, star 100, is 8.8 pixels or 21.8 arc-sec. One might perhaps expect to see an effect from star 100 when the circle around 656 gets to a diameter of about 9 and that is about what happens, in Fig 5, it can be seen that it very nearly levels out at a diameter of 9 or 10 but then the influence of its companion begins to be felt. Compare that to the curve for the star c (941), Fig 6.

Fig.4

variables & brighter stars

· q ······	· O	656v	
· ······	· a	27	
· ······	· d ······	· b	100
· ··· r ······	· o ······	· c	941
· ······	· ······	· d	1952v
· ······	· h ······	· e	1982
· ······	· f ······	· f	2958
· ······	· ······	· g	3943v
· ······	· ······	· h	7482
· ······	· i ······	· i	8577
· ······	· O b ······	· j	8720
· ······	· c ······	· k	9895v
· ··· g ······	· p ······	· l	11061
· ······	· j ··· l ······	· m	11873v
· ······	· ······	· k · n	12777
· ··· t ······	· e ··· m a ······	· o	12920
· ······	· ······	· p	14231
· ······	· s ······	· q	15731
· ······	· ······	· r	16080
· n ······	· ······	· s	16218
· ······	· ······	· t	18874v

FIG. 5

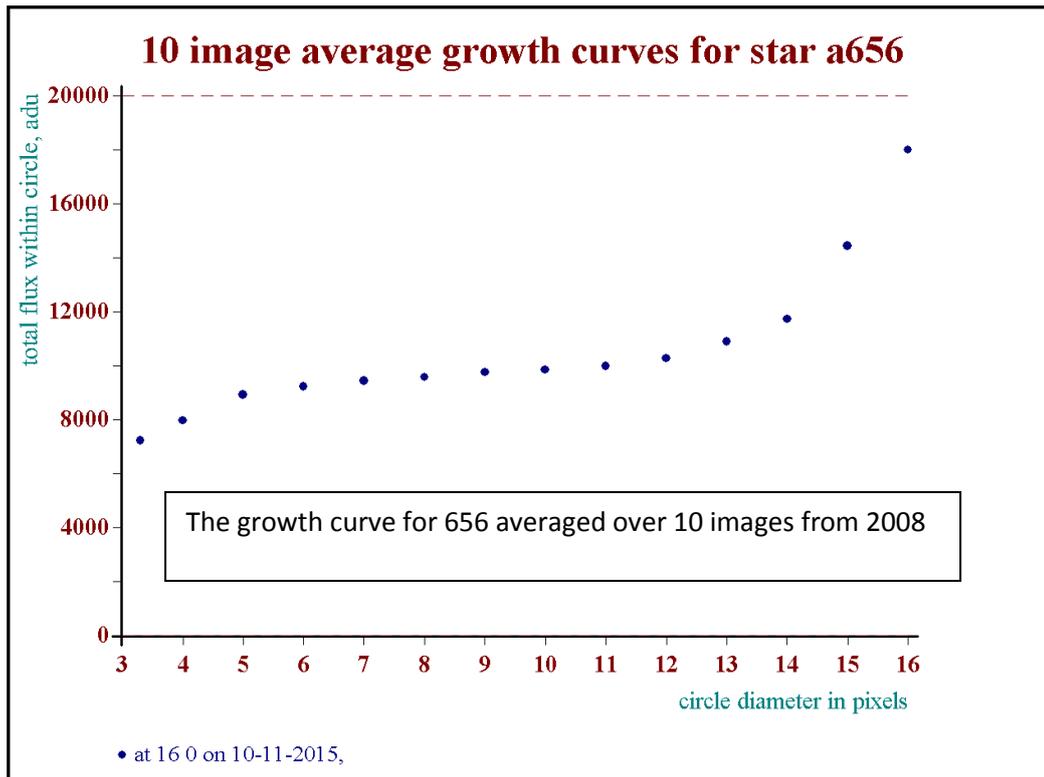
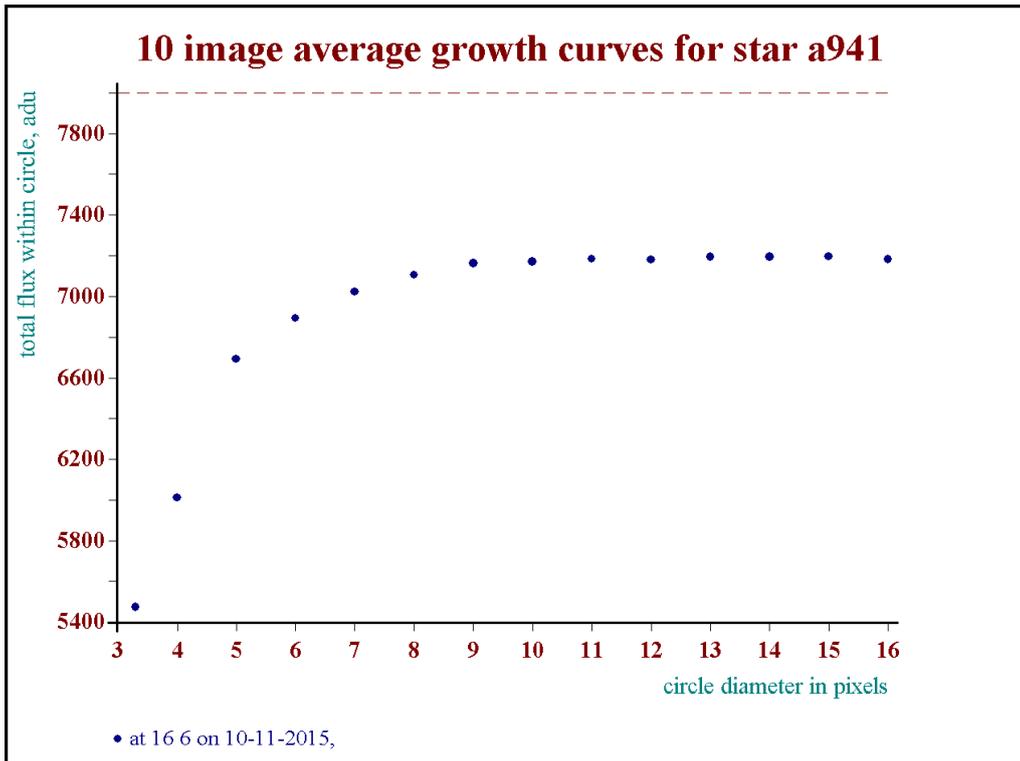


Fig 6



This only varies by 0.1% from 11 to 16 diameter. If we take the mean of the last 6 as being the right answer then by observing that that mean is 4.3, 2.4, 1.2 and 0.36% above the values at diameters 6, 7, 8 and 9 we can estimate what 656 should be on that basis.

### 3.2.5 Data Reduction with Image Samples

The area strand uses group sets, these are explained below.

Although it cover most of what one would want to do the starlist approach has two shortcomings:

- a) Most of each image is ignored so if something interesting happened between the catalogue stars it would not be noticed.
- b) There may be an interesting star not in the list! In area 'a' there are in fact at least 150,000 measurable stars down to about magnitude 18. Not for me accurately measurable but some useful information can be obtained by averaging 20 plus images.

This approach is to divide every image into a number of small and equal areas and form sets of these sub-images. I used this method to search for transient happenings, in particular flares which may well happen in objects too faint to be in my starlist.

I've not in fact attempted to stack whole images but have found it easier to divide the images into squares, in fact 49 by 49 squares or 2501 of them, each 91 pixels on a side. Each square is centred on a precise ra/dec so that even over years (if I keep the orientation within one degree) the subsquares line up with a worst error of 2.4

arc-sec in a corner and that would very rarely happen. The subsquares are averaged in sets of 25 and the 91 pixels allows for a little overlap so the raw images are effectively reduced in storage requirement by about 20 times. Any of these star field averages can then be instantly compared with any other, from the same night to nights years apart. This system overcomes both problems, a and b. I've not taken full advantage of it yet because of other priorities but eventually all the variable stars in that 150,000 set will be listed and any events that happened also logged.

### 3 Finding Variability

Having got the data into a study-able form the interesting work starts, looking for variability. This essentially boils down to looking at stars that have more 'noise' than others. Knowing little about variable stars I classified them into very short period variables (VSPVs), SPVs, medium period variables (MPVs) and long period variables (LPVs). Of course there are sub categories, some stars produce spikes and dips, and some vary continuously. There are different techniques for each. The continuous VSPVs and SPVs are best found by Fourier and overlay<sup>1</sup> methods, the spiky or dippy ones overlay only. The analysis is done with varying degrees of averaging before analysis, from none up to about 30 samples. Great care is required when analysing averaged results or serious errors can result. The averaging must be by time interval of course. For MPVs and LPVs I just average each whole night's worth of data (for initial finding at least) and then compare the variability (i.e. the noise) of each set (with a variety of comparison stars for each) with the average for that brightness.

If the variable has a small range the 'fast' ones are much easier to find because one can do a periodogram using a week's data or a month or even a year which is a huge amount of averaging for a 0.1day period say. Of course, if the period changes, then a smaller window is necessary.

A big problem is aliases. I'm sure I have many errors due to that and there will be a separate report detailing several alias investigations.

I could write a great deal more about this, a great deal of my time has been spent on it. I'll do a detailed report on finding variability if I get requests to do so!

### 4. Reporting

I only gradually developed a standard report format for each star so many of the reports in this website will be in different forms initially because the software helping to produce them evolved. Hopefully they will be standardised eventually. The standard form starts with a bitmap, or set of zoom bitmaps showing the environs of the star and listing those around it. Then there is basic data about the star and that is followed by comments, deductions and plots of light-curves, periodograms, phase plots etc.

Note 1. What I call 'overlay' is generally called 'folding'.